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}@iaik.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 we prove secure in the SXDH setting where EUF-CMA security is proven against generic forgers in the generic group model and the so called class hiding property is proven 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. We present our ABC system along with a suitable security model and rigorously prove its security.

Keywords: Public key cryptography, structure-preserving signatures, attribute-based anonymous credentials, polynomial commitments

1 Notes

This is an updated version of the paper in response to the paper [38] by Fuchsbauer. He provides an attack which invalidates the claimed EUF-CMA security of the candidate construction of an SPS-EQ-$\mathcal{R}$ scheme included in the proceedings version of ASIACRYPT 2014 (and also in this extended version). Subsequently, we discuss the implications and updates in this version.

1. For our original candidate construction of an SPS-EQ-$\mathcal{R}$ we falsely claimed EUF-CMA security (as shown in [38]). In this paper we show that the original construction at least provides unforgeability against random message attacks (RMA) (however, due to 3) we consider the original candidate construction as obsolete).
2. The attack does not affect the multi-show attribute-based anonymous credential (ABC) system construction, as it can be instantiated in a black-box way from any EUF-CMA secure SPS-EQ-$\mathcal{R}$ scheme.
3. In a recent work [39], together with Fuchsbauer, we present an EUF-CMA secure SPS-EQ-$\mathcal{R}$ scheme, which is even more efficient than the original construction in every respect. As a consequence, our ABC system can be efficiently instantiated.

2 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

* This is an updated full version of a paper appearing in the proceedings of ASIACRYPT 2014.

hash function is usually required to be modeled as a random oracle (thus, one signs random group elements). In contrast, structure-preserving signatures [36,6,1,2,23,5,4] 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 [26,27,20,49]. We emphasize that although these schemes are randomizable, they are still secure digital signatures in the standard sense ($\mathsf{EUF\text{-}CMA}$ 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.

2.1 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. 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 along with a security model in which we 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. We prove the proposed ABC system secure in a security model adapted from [25,8,29,30] and, finally, we compare our system to other existing multi- and one-show ABC approaches. We note that 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 [22] (and, thus, also its provably secure generalization [12]).

2.2 Related Work

In [18], Blazy et al. present *signatures on randomizable ciphertexts* (based on linear encryption [20]) using a variant of Waters' signature scheme [49]. 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 [34] which is used to build *self-blindable* certificates by Verheul in [48]. 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 [48], but it does not represent a secure signature scheme (as also observed in [48]) due to its homomorphic property and the possibility of efficient existential forgeries.

Homomorphic signatures for network coding [21] 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., [21,10] 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 [45].

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 [32], 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 [43] 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 [32] relies on malleable-ZKPs [31] and is not really efficient, even when instantiated with Groth-Sahai proofs [41]. 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 $\mathsf{SigEval}$ 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, [32] 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 [26,27,20] 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 [26,27]. 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 [37], 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.

2.3 Organization

Section 3 discusses preliminaries and Section 4 presents our signature scheme. In Section 5, we propose the polynomial commitment scheme. Section 6 shows how to build an efficient ABC system from the previously introduced signature scheme and the previously introduced polynomial commitment scheme. Finally, we discuss other possible applications of the proposed signature scheme and future work in Section 7.

3 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 $e : G_1 \times G_2 \rightarrow G_T$ *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 $\Psi : G_2 \rightarrow G_1$, whereas for Type-3 pairings no such efficient isomorphism is assumed to exist. Note that Type-3 pairings are currently the optimal choice [33] 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

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

Definition 3 (Symmetric External Diffie Hellman 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 Diffie Hellman Assumption (t-SDH) [19]). 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 [33]:

Definition 6 (co-t-Strong Diffie Hellman Assumption (co-t-SDH$_i^*$)). Let p be a prime of bitlength κ, G_1 and G_2 be two groups of prime order p 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

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

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$_i^* \le_p$ t-SDH in group G_i. The t-SDH assumption was originally proven to be secure in the generic group model in [19, Theorem 5.1] and further studied in [35]. 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 [19, 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.

Finally, note that later on we will use the co-t-SDH$_1^*$ assumption in a static way, as we fix the value t a priori as a system parameter.

3.1 Proofs of Knowledge

In a proof of knowledge (PoK) [16], we consider a binary relation $R = \{(y, w) : y \in L, w \in W(y)\}$, for which membership $y \in L$ with $L = \{y : \exists w \text{ such that } R(y, w) = 1\}$ can be tested in polynomial time (here $W(y)$ denotes the set of witnesses associated to y). On common input y to a prover and a verifier, the prover with additional secret input w can convince the verifier that it knows some $w \in W(y)$, such that $(y, w) \in R$ holds and without disclosing any information about w. An example for this would be $R_{DL} = \{(Y, x) : Y \in G, Y = xP\}$ for group $G = \langle P \rangle$ of a prime order p. This can be efficiently proven using three-move honest-verifier zero-knowledge proofs of knowledge (Σ-protocols) with proofs of the form (α, β, γ). We recall the special soundness property, which states that for two transcripts of the form $t = (\alpha, \beta, \gamma)$ and $t' = (\alpha, \beta', \gamma')$ such that $\beta \neq \beta'$, there is a polynomial-time knowledge extractor $\mathcal{E}$ that on input (t, t') outputs w' such that $R(y, w') = 1$. As it is common, we use the notation of [28] and denote a proof of knowledge of a discrete logarithm $x = \log_P Y$ as $\mathsf{PoK}\{\alpha : Y = \alpha P\}$ and a transcript as (K_Y, c, s) where c is the challenge, $K_Y = kP$ and $s = k + xc \mod p$.

3.2 Digital Signatures

Definition 7 (Digital Signature Scheme). A *digital signature scheme* is a tuple $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ of polynomial time algorithms:

$\mathsf{KeyGen}(\kappa)$**:** Is a probabilistic algorithm that takes input a security parameter $\kappa \in \mathbb{N}$ and outputs a private key sk and a public key pk (we assume that pk includes a description of the message space $\mathcal{M}$).

$\mathsf{Sign}(M, \mathsf{sk})$**:** Is a (probabilistic) algorithm that takes input a message $M \in \mathcal{M}$, a secret key sk and outputs a signature σ.

$\mathsf{Verify}(M, \sigma, \mathsf{pk})$**:** Is a deterministic algorithm that takes input a message $M \in \mathcal{M}$, a signature σ, a public key pk and outputs `true` if σ is a valid signature for M under pk and `false` otherwise.

A digital signature scheme is secure, if it is *correct* and existentially unforgeable under adaptively chosen-message attacks ($\mathsf{EUF\text{-}CMA}$) [40]. A signature scheme can also be secure under the weaker notions of non-adaptive chosen-message attacks ($\mathsf{NA\text{-}CMA}$) or random message attacks (RMA). We define the properties below:

Definition 8 (Correctness). A digital signature scheme $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ is called *correct*, if

$$\forall \kappa > 0\ \forall (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(\kappa)\ \forall M \in \mathcal{M} : \quad \mathsf{Verify}(M, \mathsf{Sign}(M, \mathsf{sk}), \mathsf{pk}) = \mathtt{true}$$

Definition 9 (RMA). A digital signature scheme $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ is called secure under *random message attacks*, if for all PPT algorithms $\mathcal{A}$ having access to a signing oracle $\mathcal{O}(\mathsf{sk})$ which on i-th call randomly chooses a message M_i from the message space, issues a signature σ_i on M_i and returns (M_i, σ_i), there is a negligible function $\epsilon(\cdot)$ such that:

$$\Pr\left[(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(\kappa), (M^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}(\mathsf{sk})}(\mathsf{pk}) : \quad M^* \notin Q \quad \wedge \quad \mathsf{Verify}(M^*, \sigma^*, \mathsf{pk}) = \mathtt{true}\right] \leq \epsilon(\kappa).$$

where Q is the set of answers from the signing oracle $\mathcal{O}(\mathsf{sk})$.

Definition 10 (NA-CMA). A digital signature scheme $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ is called *existentially unforgeable under non-adaptive chosen-message attacks*, if for all PPT algorithms $\mathcal{A}$ having access to a signing oracle $\mathcal{O}(\mathsf{sk})$ which on i-th call issues a signature for the i-th message in $(M_i)_{i \in [\mathsf{poly}(\kappa)]}$, there is a negligible function $\epsilon(\cdot)$ such that:

$$\Pr\left[\begin{array}{c}(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(\kappa), (\mathsf{state}, (M_i)_{i \in [\mathsf{poly}(\kappa)]}) \leftarrow \mathcal{A}(\mathsf{pk}), \\ (M^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}(\mathsf{sk})}(\mathsf{state}, \mathsf{pk}) : \quad M^* \notin (M_i)_{i \in [\mathsf{poly}(\kappa)]} \quad \wedge \quad \mathsf{Verify}(M^*, \sigma^*, \mathsf{pk}) = \mathtt{true}\end{array}\right] \leq \epsilon(\kappa).$$

Definition 11 (EUF-CMA). A digital signature scheme $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ 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:

$$\Pr\left[(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(\kappa), (M^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}(\mathsf{sk}, \cdot)}(\mathsf{pk}) : \quad M^* \notin Q \quad \wedge \quad \mathsf{Verify}(M^*, \sigma^*, \mathsf{pk}) = \mathtt{true}\right] \leq \epsilon(\kappa),$$

where Q is the set of queries which $\mathcal{A}$ has issued to the signing oracle $\mathcal{O}(\mathsf{sk}, \cdot)$.

4 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.

4.1 Defining the Signature Scheme

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

Definition 12 (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 13 (Correctness). An SPS-EQ-$\mathcal{R}$ scheme $(\mathsf{BGGen}_{\mathcal{R}}, \mathsf{KeyGen}_{\mathcal{R}}, \mathsf{Sign}_{\mathcal{R}}, \mathsf{ChgRep}_{\mathcal{R}}, \mathsf{Verify}_{\mathcal{R}})$ 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 $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}_{\mathcal{R}}(\mathsf{BG}, \ell)$, for all $M \in (G_1^*)^\ell$ and $\rho \in \mathbb{Z}_p^*$ it holds that

$$\Pr\left[\mathsf{Verify}_{\mathcal{R}}(M, \mathsf{Sign}_{\mathcal{R}}(M, \mathsf{sk}), \mathsf{pk}) = \mathtt{true}\right] = 1 \quad \text{and}$$
$$\Pr\left[\mathsf{Verify}_{\mathcal{R}}(\mathsf{ChgRep}_{\mathcal{R}}(M, \mathsf{Sign}_{\mathcal{R}}(M, \mathsf{sk}), \rho, \mathsf{pk}), \mathsf{pk}) = \mathtt{true}\right] = 1.$$

Furthermore, we require a notion of $\mathsf{EUF\text{-}CMA}$ security. In contrast to the standard definition of $\mathsf{EUF\text{-}CMA}$ 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 14 (EUF-CMA). An SPS-EQ-$\mathcal{R}$ scheme $(\mathsf{BGGen}_{\mathcal{R}}, \mathsf{KeyGen}_{\mathcal{R}}, \mathsf{Sign}_{\mathcal{R}}, \mathsf{ChgRep}_{\mathcal{R}}, \mathsf{Verify}_{\mathcal{R}})$ on $(G_1^*)^\ell$ 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:

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

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

Note that as in case of traditional signatures we can analogously define $\mathsf{NM\text{-}CMA}$ as well as RMA security. 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 key pair sk, pk, a bit b and a message M. If $M \notin Q$, it returns $\perp$. On the first valid call, it records M; if later called on a different message, it returns $\perp$. Otherwise, it picks $R \xleftarrow{R} (G_i^*)^\ell$ and $\rho \xleftarrow{R} \mathbb{Z}_p^*$, sets $(M_0, \sigma_0) \leftarrow \mathsf{ChgRep}_{\mathcal{R}}(M, \sigma, \rho, \mathsf{pk})$ and $(M_1, \sigma_1) \leftarrow (R, \mathsf{Sign}_{\mathcal{R}}(R, \mathsf{sk}))$ and returns (M_b, σ_b).

Definition 15 (Class Hiding). An SPS-EQ-$\mathcal{R}$ scheme $(\mathsf{BGGen}_{\mathcal{R}}, \mathsf{KeyGen}_{\mathcal{R}}, \mathsf{Sign}_{\mathcal{R}}, \mathsf{ChgRep}_{\mathcal{R}}, \mathsf{Verify}_{\mathcal{R}})$ 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

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

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. Moreover, in the class hiding definition we have implicitly assumed that one can check adversarial keys for correctness.

We note that in the recent $\mathsf{EUF\text{-}CMA}$ secure SPS-EQ-$\mathcal{R}$ construction [39], a stronger variant of class hiding is defined, in which the $\mathcal{O}^{RoR}$ oracle is given a message-signature pair instead of solely a message. This requires, that an adversary is not able to distinguish a re-randomization of a message-signature pair from a random one (the construction in this paper and in [39] provide this stronger notion). Moreover, in [39] the definition of SPS-EQ-$\mathcal{R}$ is extended by an explicit key verification algorithm $\mathsf{VKey}_{\mathcal{R}}$, which when given an adversarially generated key pair $(\mathsf{sk}, \mathsf{pk})$ efficiently checks the keys for consistency. We denote this stronger class hiding property as *class hiding** henceforth, as we require this property later in the black-box construction of the ABC system.

Definition 16 (Security). An SPS-EQ-$\mathcal{R}$ scheme $(\mathsf{BGGen}_{\mathcal{R}}, \mathsf{KeyGen}_{\mathcal{R}}, \mathsf{Sign}_{\mathcal{R}}, \mathsf{ChgRep}_{\mathcal{R}}, \mathsf{Verify}_{\mathcal{R}})$ is X-*secure*, if it is correct, class hiding and $\mathsf{X} \in \{\mathsf{EUF\text{-}CMA}, \mathsf{NA\text{-}CMA}, \mathsf{RMA}\}$.

4.2 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. 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 $y \cdot \hat{y}$).

4.3 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 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. Therefore, we chose to prove the 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].

Now, we state the security of the signature scheme. The proofs will be given in Appendix B.

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

In the original version we falsely stated the subsequent theorem with respect to $\mathsf{EUF\text{-}CMA}$ security (which has been shown to be incorrect in [38]). However, the following can be shown to hold:

Theorem 2. *In the generic group model for SXDH groups, Scheme 1 is an* RMA *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.*

$\mathsf{BGGen}_{\mathcal{R}}(\kappa)$**:** 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

5 Polynomial Commitments with Factor Openings

In [44], 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 [44]. Now, we briefly present this construction, which we denote by $\mathsf{PolyCommitFO}$.

$\mathsf{Setup}_{\mathsf{PC}}(\kappa, t)$**:** 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 $(g(X), \mathcal{C}_h)$.

[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.

$\mathsf{VerifyFactor_{PC}}(\mathsf{pp}, \mathcal{C}, g(X), \mathcal{C}_h)$**:** 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 [44], 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. We introduce a security model and give security proofs in Appendix A. 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 $\mathsf{VerifyFactor_{PC}}$, we define it differently to reduce the computational complexity of the prover in the ABC system in Section 6.3. 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.

6 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 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 [22,12]. 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 [26,27]. 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 6.2 and present the scheme in Section 6.3. In Section 6.4, we discuss the security of the construction. Finally, we give a performance comparison with other existing approaches in Section 6.5.

6.1 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 17 (Attribute-Based Anonymous Credentials System). An *attribute-based anonymous credentials 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.

$\mathsf{OrgKeyGen}$: 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.

$\mathsf{UserKeyGen}$: 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.

$(\mathsf{Obtain}, \mathsf{Issue})$: 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.

$(\mathsf{Show}, \mathsf{Verify})$: 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 attribute-based anonymous credential system is called *secure* if it is *correct*, *unforgeable* and *anonymous* (for a formal definition of these properties, we refer the reader to Appendix C).

6.2 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 for guaranteeing freshness. In contrast to this, the number of proofs of knowledge in other ABC systems, like [25,22] 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 [29,30]. 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 $\{"01.01.1980", "> 16", "> 18", "> 21"\}$ 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 4.2). 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 $\mathsf{PolyCommitFO}$ (cf. Section 5) 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 [42]). 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 $\mathsf{PolyCommitFO}$ 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 proof of knowledge 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:

$$\mathbb{A} = \{(\mathtt{gender}, \{\mathtt{male}\}), (\mathtt{birthdate}, \{\mathsf{01.01.1980}, > 18, > 21\}), (\mathtt{drivinglicense}, \{\#, \mathtt{car}, \mathtt{truck}\})\}.$$

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 $\overline{\mathbb{A}'}$:

$$\mathbb{A}' = \{(\mathtt{birthdate}, \{> 21\}), (\mathtt{drivinglicense}, \{\#\})\}$$
$$\overline{\mathbb{A}'} = \{(\mathtt{gender}, \{\mathtt{male}\}), (\mathtt{birthdate}, \{\mathsf{01.01.1980}, > 18\}), (\mathtt{drivinglicense}, \{\mathtt{car}, \mathtt{truck}\})\}.$$

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.

6.3 The Construction of the ABC System

Now, we present our ABC system. Subsequently, 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 $\mathtt{true}$, the respective algorithm terminates with a failure and the algorithm Verify accepts only if $\mathsf{VerifyFactor}_{\mathsf{PC}}$ and $\mathsf{Verify}_{\mathcal{R}}$ return $\mathtt{true}$ 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. Moreover, we emphasize that in an honest issuer model the costs for the obtainer can be made constant. This can be achieved by moving the computation of $\mathsf{enc}(\mathbb{A})(\alpha)P$ into the Issue algorithm (at the expense of an additional round for including r_i into C_1) and removing the verification of σ.

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 $r_i \xleftarrow{R} \mathbb{Z}_p^*$, compute $R_i \leftarrow r_i P$ 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:

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

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

Scheme 2: A Multi-Show ABC System

6.4 Security

In Appendix C, we introduce a security model for attribute-based anonymous credentials and we formally prove the following. We note that in the original version our theorems were formulated with respect to the candidate construction, but we actually can formulate all of them in a black-box fashion with respect to any EUF-CMA secure SPS-EQ-$\mathcal{R}$ scheme (as done below) that provides the stronger class hiding notion (*class hiding**).

Theorem 4. *Scheme 2 is correct.*

Theorem 5. *If* PolyCommitFO *is factor-sound, H is a collision-resistant hash function, the SPS-EQ-$\mathcal{R}$ scheme is* EUF-CMA *secure and the DLP is hard in G_1, then Scheme 2 is unforgeable.*

Theorem 6. *If an SPS-EQ-$\mathcal{R}$ provides class hiding*, then Scheme 2 is anonymous.*

Taking everything together, we obtain the following corollary:

Corollary 1. *Scheme 2 is a secure attribute-based anonymous credential system.*

Note that in the proof of Theorem 5, we can distinguish whether a forgery goes back to a signature forgery of the SPS-EQ-$\mathcal{R}$ scheme 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 security game.

6.5 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 [25,26,27] as well as schemes from BBS$^+$ signatures [20,11] 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 [29,30] as well as Brands' approach [22] (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 [24] for CL credentials in the strong RSA setting (encoding attributes as prime numbers) or in a pairing-based setting using BBS^+ credentials [47] (encoding attributes using accumulators) where the latter additionally requires very large public parameters (one F-secure 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 subgroup of order q of $\mathbb{Z}_p^*$ with $p = 2q + 1$ or an elliptic curve group of order q). 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 [29]. We emphasize that, in contrast to other approaches, such as [27,30], our construction when instantiated with the SPS-EQ-$\mathcal{R}$ scheme from [39] 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.

Scheme	**Parameter Size** (L attributes)				**Issuing**			**Showing** (k-of-L attributes)		
	Setting	Public Params	Credential Size		Issuer	User	Comm	Verifier	User	Comm
CaLy [25,26]	sRSA	$O(L)$	$O(1)$	$3\|\mathbb{Z}_N\|$	$O(L)$	$O(L)$	$O(L)$	$O(L)$	$O(L)$	$O(L-k)$
CaLy [27]	Type-1	$O(L)$	$O(L)$	$(2L+2)\|G_1\|$	$O(L)$	$O(L)$	$O(L)$	$O(L)$	$O(L)$	$O(L)$
BBS [20]	Type-2	$O(L)$	$O(1)$	$\|G_1\| + 22\|\mathbb{Z}_q\|$	$O(L)$	$O(L)$	$O(1)$	$O(L)$	$O(L)$	$O(L)$
CaLe [29]	Type-2	$O(1)$	$O(L)$	$L\|G_1\| + c + \|G_2\|$	$O(L)$	$O(L)$	$O(L)$	$O(L)$	$O(1)$	$O(1)$
CaLe [30]	XDH	$O(L)$	$O(L)$	$(2L+2)(\|G_1\| + \|\mathbb{Z}_p\|)$	$O(L)$	$O(L)$	$O(L)$	$O(k)$	$O(k)$	$O(k)$
Br [22]	G_q	$O(L)$	$O(1)$	$2\|G_q\| + 2\|\mathbb{Z}_q\|$	$O(L)$	$O(L)$	$O(1)$	$O(k)$	$O(k)$	$O(L-k)$
Scheme 2	SXDH	$O(L)$	$O(1)$	$4\|G_1\| + \|G_2\|$	$O(L)$	$O(L)$	$O(1)$	$O(k)$	$O(L-k)$	$O(1)$

7 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 and Georg Fuchsbauer 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. Springer (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. Springer (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. Springer (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. Springer (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. Springer (2014)

6. Abe, M., Haralambiev, K., Ohkubo, M.: Signing on Elements in Bilinear Groups for Modular Protocol Design. IACR Cryptology ePrint Archive (2010), `http://eprint.iacr.org/`
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. Springer (2012)
8. Akagi, N., Manabe, Y., Okamoto, T.: An Efficient Anonymous Credential System. In: FC. LNCS, vol. 5143, pp. 272–286. Springer (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. Springer (2012)
10. Attrapadung, N., Libert, B., Peters, T.: Efficient completely context-hiding quotable and linearly homomorphic signatures. In: Public Key Cryptography. LNCS, vol. 7778, pp. 386–404. Springer (2013)
11. Au, M.H., Susilo, W., Mu, Y.: Constant-Size Dynamic k-TAA. In: SCN. LNCS, vol. 4116, pp. 111–125. Springer (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), `http://eprint.iacr.org/`
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. Springer (2009)
15. Belenkiy, M., Chase, M., Kohlweiss, M., Lysyanskaya, A.: P-signatures and Noninteractive Anonymous Credentials. In: TCC. LNCS, vol. 4948, pp. 356–374. Springer (2008)
16. Bellare, M., Goldreich, O.: On Defining Proofs of Knowledge. In: CRYPTO. LNCS, vol. 740, pp. 390–420. Springer (1992)
17. Blanton, M., Hudelson, W.: Biometric-Based Non-transferable Anonymous Credentials. In: ICICS. LNCS, vol. 5927, pp. 165–180. Springer (2009)
18. Blazy, O., Fuchsbauer, G., Pointcheval, D., Vergnaud, D.: Signatures on Randomizable Ciphertexts. In: PKC. LNCS, vol. 6571, pp. 403–422. Springer (2011)
19. Boneh, D., Boyen, X.: Short Signatures Without Random Oracles. In: EUROCRYPT. LNCS, vol. 3027, pp. 56–73. Springer (2004)
20. Boneh, D., Boyen, X., Shacham, H.: Short Group Signatures. In: CRYPTO. LNCS, vol. 3152, pp. 41–55. Springer (2004)
21. 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. Springer (2009)
22. Brands, S.: Rethinking public-key Infrastructures and Digital Certificates: Building in Privacy. MIT Press (2000)
23. Camenisch, J., Dubovitskaya, M., Haralambiev, K.: Efficient Structure-Preserving Signature Scheme from Standard Assumptions. In: SCN. LNCS, vol. 7485, pp. 76–94. Springer (2012)
24. Camenisch, J., Groß, T.: Efficient Attributes for Anonymous Credentials. ACM Trans. Inf. Syst. Secur. 15(1), 4 (2012)
25. Camenisch, J., Lysyanskaya, A.: An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. In: EUROCRYPT. LNCS, vol. 2045, pp. 93–118. Springer (2001)
26. Camenisch, J., Lysyanskaya, A.: A Signature Scheme with Efficient Protocols. In: SCN. LNCS, vol. 2576, pp. 268–289. Springer (2002)
27. Camenisch, J., Lysyanskaya, A.: Signature Schemes and Anonymous Credentials from Bilinear Maps. In: CRYPTO. LNCS, vol. 3152, pp. 56–72. Springer (2004)
28. Camenisch, J., Stadler, M.: Efficient Group Signature Schemes for Large Groups (Extended Abstract). In: CRYPTO. LNCS, vol. 1294, pp. 410–424. Springer (1997)
29. Canard, S., Lescuyer, R.: Anonymous credentials from (indexed) aggregate signatures. In: DIM. pp. 53–62. ACM (2011)
30. Canard, S., Lescuyer, R.: Protecting privacy by sanitizing personal data: a new approach to anonymous credentials. In: ASIACCS. pp. 381–392. ACM (2013)
31. Chase, M., Kohlweiss, M., Lysyanskaya, A., Meiklejohn, S.: Malleable proof systems and applications. In: EUROCRYPT. LNCS, vol. 7237, pp. 281–300. Springer (2012)
32. Chase, M., Kohlweiss, M., Lysyanskaya, A., Meiklejohn, S.: Malleable Signatures: Complex Unary Transformations and Delegatable Anonymous Credentials. IACR Cryptology ePrint Archive (2013), `http://eprint.iacr.org/`
33. Chatterjee, S., Menezes, A.: On cryptographic protocols employing asymmetric pairings - the role of ψ revisited. Discrete Applied Mathematics 159(13), 1311–1322 (2011)
34. Chaum, D., Pedersen, T.P.: Wallet Databases with Observers. In: CRYPTO. LNCS, vol. 740, pp. 89–105. Springer (1992)
35. Cheon, J.H.: Security Analysis of the Strong Diffie-Hellman Problem. In: EUROCRYPT. LNCS, vol. 4965, pp. 1–11. Springer (2006)
36. Fuchsbauer, G.: Automorphic Signatures in Bilinear Groups and an Application to Round-Optimal Blind Signatures. IACR Cryptology ePrint Archive (2009)
37. Fuchsbauer, G.: Commuting signatures and verifiable encryption. In: EUROCRYPT. LNCS, vol. 6632, pp. 224–245. Springer (2011)

38. Fuchsbauer, G.: Breaking Existential Unforgeability of a Signature Scheme from Asiacrypt 2014. Cryptology ePrint Archive, Report 2014/892 (2014), `http://eprint.iacr.org/`
39. Fuchsbauer, G., Hanser, C., Slamanig, D.: EUF-CMA-Secure Structure-Preserving Signatures on Equivalence Classes. Cryptology ePrint Archive, Report 2014/944
40. Goldwasser, S., Micali, S., Rivest, R.L.: A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks. SIAM J. Comput. 17(2), 281–308 (1988)
41. Groth, J., Sahai, A.: Efficient Non-interactive Proof Systems for Bilinear Groups. In: EUROCRYPT. LNCS, vol. 4965, pp. 415–432. Springer (2008)
42. Hanser, C., Slamanig, D.: Blank Digital Signatures. IACR Cryptology ePrint Archive (2013)
43. Johnson, R., Molnar, D., Song, D.X., Wagner, D.: Homomorphic Signature Schemes. In: CT-RSA. LNCS, vol. 2271, pp. 244–262. Springer (2002)
44. Kate, A., Zaverucha, G.M., Goldberg, I.: Constant-size commitments to polynomials and their applications. In: ASIACRYPT. LNCS, vol. 6477, pp. 177–194. Springer (2010)
45. 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. Springer (2013)
46. Lysyanskaya, A., Rivest, R.L., Sahai, A., Wolf, S.: Pseudonym Systems. In: SAC. LNCS, vol. 1758, pp. 184–199. Springer (2000)
47. Sudarsono, A., Nakanishi, T., Funabiki, N.: Efficient Proofs of Attributes in Pairing-Based Anonymous Credential System. In: Privacy Enhancing Technologies. LNCS, vol. 6794, pp. 246–263. Springer (2011)
48. Verheul, E.R.: Self-Blindable Credential Certificates from the Weil Pairing. In: ASIACRYPT. LNCS, vol. 2248, pp. 533–551. Springer (2001)
49. Waters, B.: Efficient Identity-Based Encryption Without Random Oracles. In: EUROCRYPT. LNCS, vol. 3494, pp. 114–127. Springer (2005)

A Security of PolyCommitFO

In this section, we discuss the security properties of polynomial commitment schemes with factor openings and prove the security of the PolyCommitFO construction presented in Section 5.

Definition 18 (Security of Polynomial Commitment Schemes with Factor Openings). A polynomial commitment scheme with factor openings is *secure*, if the following properties hold:

Correctness: $\forall \kappa > 0\, \forall t > 0\, \forall \mathsf{pp} \leftarrow \mathsf{Setup}_{\mathsf{PC}}(\kappa, t)\, \forall$ monic, reducible $f(X) \in \mathbb{Z}_p[X]\, \forall \mathcal{C} \leftarrow \mathsf{Commit}_{\mathsf{PC}}(\mathsf{pp}, f(X))$ (using an arbitrary $\rho \in \mathbb{Z}_p^*$), we require that

- $\mathsf{Verify}_{\mathsf{PC}}(\mathsf{pp}, \mathcal{C}, \mathsf{Open}_{\mathsf{PC}}(\mathsf{pp}, \mathcal{C}, \rho, f(X))) = \mathtt{true}$, and
- $\mathsf{VerifyFactor}_{\mathsf{PC}}(\mathsf{pp}, \mathcal{C}, \mathsf{FactorOpen}_{\mathsf{PC}}(\mathsf{pp}, f(X), g(X), \rho)) = \mathtt{true} \quad \forall g(X) \mid f(X)$.

Polynomial Binding: For all PPT adversaries $\mathcal{A}$, we require that there is a negligible function $\epsilon(\cdot)$ such that:

$$\Pr\left[\begin{matrix} \mathsf{pp} \leftarrow \mathsf{Setup}_{\mathsf{PC}}(\kappa, t), (\mathcal{C}, \rho_0, f_0(X), \rho_1, f_1(X)) \leftarrow \mathcal{A}(\mathsf{pp}) : \\ f_0(X) \neq f_1(X) \quad \wedge \quad \mathsf{Verify}_{\mathsf{PC}}(\mathsf{pp}, \mathcal{C}, \rho_i, f_i(X)) = \mathtt{true} \ \text{ for } i = 0, 1 \end{matrix}\right] \leq \epsilon(\kappa)$$

Factor Binding: For all PPT adversaries $\mathcal{A}$, we require that there is a negligible function $\epsilon(\cdot)$ such that:

$$\Pr\left[\begin{matrix} \mathsf{pp} \leftarrow \mathsf{Setup}_{\mathsf{PC}}(\kappa, t), (\mathcal{C}, \mathcal{C}_h, g_0(X), g_1(X)) \leftarrow \mathcal{A}(\mathsf{pp}) : \\ g_0(X) \neq g_1(X) \quad \wedge \quad \mathsf{VerifyFactor}_{\mathsf{PC}}(\mathsf{pp}, \mathcal{C}, g_i(X), \mathcal{C}_h) = \mathtt{true} \ \text{ for } i = 0, 1 \end{matrix}\right] \leq \epsilon(\kappa)$$

Factor Soundness: For all PPT adversaries $\mathcal{A}$, we require that there is a negligible function $\epsilon(\cdot)$ such that:

$$\Pr\left[\begin{matrix} \mathsf{pp} \leftarrow \mathsf{Setup}_{\mathsf{PC}}(\kappa, t), (\rho, f(X), g(X), \mathcal{C}_h) \leftarrow \mathcal{A}(\mathsf{pp}) : \\ \mathsf{VerifyFactor}_{\mathsf{PC}}(\mathsf{pp}, \rho f(\alpha) P, g(X), \mathcal{C}_h) = \mathtt{true} \quad \wedge \quad g(X) \nmid f(X) \quad \wedge \quad \deg f > 0 \end{matrix}\right] \leq \epsilon(\kappa)$$

Witness Soundness: For all PPT adversaries $\mathcal{A}$, we require that there is a negligible function $\epsilon(\cdot)$ such that:

$$\Pr\left[\begin{matrix} \mathsf{pp} \leftarrow \mathsf{Setup}_{\mathsf{PC}}(\kappa, t), (\rho, f(X), g(X), \mathcal{C}_h) \leftarrow \mathcal{A}(\mathsf{pp}) : \\ \mathsf{VerifyFactor}_{\mathsf{PC}}(\mathsf{pp}, \rho f(\alpha) P, g(X), \mathcal{C}_h) = \mathtt{true} \quad \wedge \quad g(X) \mid f(X) \quad \wedge \quad \mathcal{C}_h \neq \rho \cdot \frac{f}{g}(\alpha) P \end{matrix}\right] \leq \epsilon(\kappa)$$

Hiding: Given $(\mathsf{pp}, \mathcal{C}, \{(g_i(X), \mathcal{C}_{h_i}) \leftarrow \mathsf{FactorOpen}_{\mathsf{PC}}(\mathsf{pp}, \mathcal{C}, f(X), g_i(X), \rho)\})$ for $\rho \in_R \mathbb{Z}_p^*, f(X) \in_R \mathbb{Z}_p[X]$ such that $\exists\, g(X) : g(X) \mid f(X) \quad \wedge \quad \deg g > 0 \quad \wedge \quad \gcd(g(X), \prod_i g_i(X)) = 1$ no computationally unbounded adversary $\mathcal{A}$ obtains any information about $g(X)$.

Now, we prove $\mathsf{PolyCommitFO}$ secure under the co-t-SDH$_1^*$ assumption (cf. Section 3). As already outlined in Section 5, one can analogously define a scheme based on the co-t-SDH$_2^*$ assumption with $\mathcal{C} \in G_1$ and $\mathcal{C}_h \in G_2$, if the performance of the $\mathsf{VerifyFactor_{PC}}$ algorithm is important.

Theorem 7. $\mathsf{PolyCommitFO}$ *is correct.*

Proof. The correctness of the scheme is easy to see and, therefore, the proof is omitted here. □

Theorem 8. *If the co-t-SDH$_1^*$ assumption holds, then* $\mathsf{PolyCommitFO}$ *is polynomial binding.*

Proof. We show that if $\mathcal{A}$ is able to find a commitment $\mathcal{C}$, two scalars $\rho_0, \rho_1 \in \mathbb{Z}_p^*$ and two distinct polynomials $f_0(X), f_1(X) \in \mathbb{Z}_p[X]$ such that $\mathsf{Verify_{PC}}(\mathsf{pp}, \mathcal{C}, \rho_i, f_i(X)) = \mathtt{true}$ for $i = 0, 1$, we construct an adversary $\mathcal{B}$ against the co-t-SDH$_1^*$ problem.

$\mathcal{B}$ gets input an instance $((\alpha^i P)_{i=0}^t, (\alpha^i P')_{i=0}^t)$ to the co-t-SDH$_1^*$ problem as well as the corresponding bilinear group description BG, sets $\mathsf{pp} \leftarrow (\mathsf{BG}, (\alpha^i P)_{i=1}^t, (\alpha^i P')_{i=1}^t)$ and runs $\mathcal{A}(\mathsf{pp})$. If $\mathcal{A}$ outputs a forgery $(\mathcal{C}, \rho_0, f_0(X), \rho_1, f_1(X))$, then we know that

$$\rho_0 f_0(\alpha) P = \mathcal{C} = \rho_1 f_1(\alpha) P$$
$$\rho_0 f_0(\alpha) P - \rho_1 f_1(\alpha) P = 0_{G_1}$$

holds. This implies that $\rho_0 f_0(\alpha) - \rho_1 f_1(\alpha) = 0$. Hence, α is a root of the polynomial $t(X) = \rho_0 f_0(X) - \rho_1 f_1(X)$. As factoring of $t(X)$ yields α, $\mathcal{B}$ can efficiently obtain α and by choosing $c \in_R \mathbb{Z}_p \setminus \{-\alpha\}$, $\mathcal{B}$ can output a solution $(c, \frac{1}{\alpha+c}P)$ to the co-t-SDH$_1^*$ problem. □

Theorem 9. *If the co-t-SDH$_1^*$ assumption holds, then* $\mathsf{PolyCommitFO}$ *is factor binding.*

Proof. We show that if $\mathcal{A}$ outputs a commitment $\mathcal{C}$, two distinct polynomials $g_0(X), g_1(X)$ of positive degree and a witness $\mathcal{C}_h$ such that $\mathsf{VerifyFactor_{PC}}(\mathsf{pp}, \mathcal{C}, g_i(X), \mathcal{C}_h) = \mathtt{true}$ for $i = 0, 1$, then we can construct an adversary $\mathcal{B}$ against the co-t-SDH$_1^*$ problem.

Adversary $\mathcal{B}$ works as follows. $\mathcal{B}$ obtains an instance $((\alpha^i P)_{i=0}^t, (\alpha^i P')_{i=0}^t)$ to the co-t-SDH$_1^*$ problem as well as the corresponding bilinear group description BG, sets $\mathsf{pp} \leftarrow (\mathsf{BG}, (\alpha^i P)_{i=1}^t, (\alpha^i P')_{i=1}^t)$ and runs $\mathcal{A}(\mathsf{pp})$. If $\mathcal{A}$ returns a forgery $(\mathcal{C}, \mathcal{C}_h, g_0(X), g_1(X))$, we know that

$$e(\mathcal{C}_h, g_0(\alpha) P') = e(\mathcal{C}, P') = e(\mathcal{C}_h, g_1(\alpha) P')$$
$$e(\mathcal{C}_h, g_0(\alpha) P' - g_1(\alpha) P') = 1_{G_T}$$

It follows that $g_0(\alpha) - g_1(\alpha) = 0$. Consequently, α is a root of the polynomial $t(X) = g_0(X) - g_1(X) \in \mathbb{Z}_p[X]$. By factoring $t(X)$, $\mathcal{B}$ can efficiently obtain α and solve the instance of the co-t-SDH$_1^*$ problem given by pp by choosing $c \in \mathbb{Z}_p \setminus \{-\alpha\}$ and outputting $(c, \frac{1}{\alpha+c}P)$. □

Theorem 10. *If the co-t-SDH$_1^*$ assumption holds, then* $\mathsf{PolyCommitFO}$ *is factor sound.*

Proof. We show that if $\mathcal{A}$ is able to find a polynomial $f(X)$, a scalar ρ, a polynomial $g(X)$ and a witness $\mathcal{C}_h$ such that $g(X) \nmid f(X)$, $\deg g, \deg f > 0$ and $\mathsf{VerifyFactor_{PC}}(\mathsf{pp}, \rho f(\alpha) P, g(X), \mathcal{C}_h) = \mathtt{true}$, we construct an adversary $\mathcal{B}$ against the co-t-SDH$_1^*$ problem.

Adversary $\mathcal{B}$ works as follows. $\mathcal{B}$ obtains an instance $((\alpha^i P)_{i=0}^t, (\alpha^i P')_{i=0}^t)$ to the co-t-SDH$_1^*$ problem as well as the corresponding bilinear group description BG, sets $\mathsf{pp} \leftarrow (\mathsf{BG}, (\alpha^i P)_{i=1}^1, (\alpha^i P')_{i=1}^1)$ and runs $\mathcal{A}(\mathsf{pp})$. If $\mathcal{A}$ returns a forgery $(\rho, f(X), g(X), \mathcal{C}_h)$, then it holds that $f(X) = g(X)\hat{h}(X) + \xi(X)$ with $\xi(X) \neq 0$ and $\mathcal{C}_h$ must have the form $\mathcal{C}_h = \rho(\hat{h}(\alpha) + \frac{\xi(\alpha)}{g(\alpha)})P$. Since $\mathcal{B}$ knows ρ, it can now compute

$$\rho^{-1}\mathcal{C}_h - \hat{h}(\alpha) P = \frac{\xi(\alpha)}{g(\alpha)} P.$$

As $\deg f, \deg g > 0$ and the public parameters pp restrict the maximum degree of polynomials to 1, we have $\deg f = 1$ and $\deg g = 1$. Hence, $g(X) = X + c$ for some $c \in \mathbb{Z}_p$ and $\deg \xi = 0$, i.e., $\xi(X) = \omega \in \mathbb{Z}_p^*$ (Note that $\mathcal{B}$ can compute both values c and ω from $g(X)$ and $h(X)$). Therefore, we obtain

$$\frac{\xi(\alpha)}{g(\alpha)} P = \frac{\omega}{\alpha + c} P.$$

As the latter is a valid group element, $c \neq -\alpha$ must hold and $(c, \frac{1}{\alpha+c}P)$ is a solution to the co-t-SDH$_1^*$ problem.
□

Theorem 11. *If the co-t-SDH$_1^*$ assumption holds, then* $\mathsf{PolyCommitFO}$ *is witness sound.*

Proof. We show that if $\mathcal{A}$ is able to find a polynomial $f(X)$, a scalar ρ, a polynomial $g(X)$ and a witness $\mathcal{C}_h \neq \rho \cdot \frac{f}{g}(\alpha)P$ such that $g(X) \mid f(X)$, $\deg g > 0$, $\mathsf{VerifyFactor_{PC}}(\mathsf{pp}, \rho f(\alpha)P, g(X), \mathcal{C}_h) = \mathtt{true}$, we construct an adversary $\mathcal{B}$ against the co-t-SDH$_1^*$ problem.

Adversary $\mathcal{B}$ works as follows. $\mathcal{B}$ obtains an instance $((\alpha^i P)_{i=0}^t, (\alpha^i P')_{i=0}^t)$ to the co-t-SDH$_1^*$ problem as well as the corresponding bilinear group description BG, sets $\mathsf{pp} \leftarrow (\mathsf{BG}, (\alpha^i P)_{i=1}^t, (\alpha^i P')_{i=1}^t)$ and runs $\mathcal{A}(\mathsf{pp})$. If $\mathcal{A}$ returns a forgery $(\rho, f(X), g(X), \mathcal{C}_h)$, then it holds that $\deg g > 0$ and that

$$e(\mathcal{C}_h, g(\alpha)P') = e(\mathcal{C}, P') = e(\rho \cdot \frac{f}{g}(\alpha)P, g(\alpha)P')$$

$$e(\mathcal{C}_h - \rho \cdot \frac{f}{g}(\alpha)P, g(\alpha)P') = 1_{G_T}$$

As $\mathcal{C}_h \neq \rho \cdot \frac{f}{g}(\alpha)P$, it follows that $g(\alpha) = 0$ (observe that $g(X) \mid f(X)$ and, hence, $\frac{f}{g}(\alpha)$ is defined). Consequently, α is a root of the polynomial $g(X)$. By factoring $g(X)$, $\mathcal{B}$ can efficiently obtain α and solve the instance of the co-t-SDH$_1^*$ problem given by pp by choosing $c \in \mathbb{Z}_p \setminus \{-\alpha\}$ and outputting $(c, \frac{1}{\alpha+c}P)$. ☐

Theorem 12. $\mathsf{PolyCommitFO}$ *is hiding.*

Proof. W.l.o.g. we assume that the only unrevealed factor of $f(X)$ is $g(X) = (X - \lambda) \in \mathbb{Z}_p[X]$. Therefore, $\mathcal{C}$ and all the values $\mathcal{C}_{h_i}$ include the values $\rho \neq 0$ and $\alpha - \lambda$ where we neither know ρ nor λ (nevertheless, observe that an unbounded adversary can obtain α). Thus, for the commitment $\mathcal{C}$ and all witness values $\mathcal{C}_{h_i}$, there are $p - 1$ equally likely, valid pairs $(\rho, \lambda) \in \mathbb{Z}_p^* \times \mathbb{Z}_p$. Consequently, $g(X)$ remains unconditionally hidden within them. ☐

Corollary 2. $\mathsf{PolyCommitFO}$ *is a secure polynomial commitment scheme with factor openings.*

B Security of the Signature Scheme on Equivalence Classes

The proof of security of Scheme 1 consists of three parts, namely, correctness, unforgeability (RMA) and class hiding*.

B.1 Proof of Theorem 1 (Correctness)

We have to show that for all $\kappa \in \mathbb{N}$, 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 $M \in (G_1^*)^\ell$, for all $y, \rho \in \mathbb{Z}_p^*$ it holds that

$$\mathsf{Verify}_{\mathcal{R}}(M, \mathsf{Sign}_{\mathcal{R}}(M, \mathsf{sk}), \mathsf{pk}) = \mathtt{true}\big] = 1 \quad \text{and}$$

$$\mathsf{Verify}_{\mathcal{R}}(\mathsf{ChgRep}_{\mathcal{R}}(M, \mathsf{Sign}_{\mathcal{R}}(M, \mathsf{sk}), \rho, \mathsf{pk}), \mathsf{pk}) = \mathtt{true}.$$

Recall that $(Z_1, Z_2, Y, Y') \leftarrow \mathsf{Sign}_{\mathcal{R}}(M, \mathsf{sk})$ is such that

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

The verification relations look as follows:

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

Plugging M, (Z_1, Z_2, Y, Y') and the public keys $(X', (X_i')_{i=1}^{\ell}) = (xP', (x_i x P')_{i=1}^{\ell})$ into the first two relations yields:

$$\prod_{i=1}^{\ell} e(M_i, x_i x P') = e(x \sum_{i=1}^{\ell} x_i M_i, P') \quad \text{and} \quad e(x \sum_{i=1}^{\ell} x_i M_i, yP') = e(y \sum_{i=1}^{\ell} x_i M_i, xP').$$

Due to the bilinearity of e it is now obvious that the verification relations are correct. Moreover, as we have $e(P, Y') = e(P, yP') = e(Y, P) = e(yP, P')$, the third verification relation is also correct.

Now, we have that $(\rho M, (\rho Z_1, \hat{y}\rho Z_2, \hat{y}Y, \hat{y}Y')) \leftarrow \mathsf{ChgRep}_{\mathcal{R}}(M, (Z_1, Z_2, Y, Y'), \rho, \mathsf{pk})$ for $\rho, \hat{y} \in \mathbb{Z}_p^*$ and see that the output of $\mathsf{ChgRep}_{\mathcal{R}}$ is a message ρM and a signature $(\rho Z_1, \hat{y}\rho Z_2, \hat{y}Y, \hat{y}Y')$, which is identical to a signature produced by $\mathsf{Sign}_{\mathcal{R}}(\rho M, \mathsf{sk})$ for message ρM and internally using randomness $\hat{y}y$. ☐

B.2 Proof of Theorem 2 (Unforgeability)

Proof. In the generic group model, an adversary only performs generic group operations (operations in G_1, G_2 and G_T, bilinear pairings and equality tests) by querying the respective group oracle.

In the first part of this proof, we consider all signature and message elements as formal polynomials in $m_{1,1}, \ldots, m_{1,\ell}, \ldots, m_{q,1}, \ldots, m_{q,\ell}, x, x_1, \ldots, x_\ell, y_1, \ldots, y_q$ and show that in this case an adversary is unable to produce valid forgeries. Then, in the second part, we show that the probability for an adversary to produce a forgery by incident is negligible.

Note that the adversary is unaware of the values $x, x_1, \ldots, x_\ell$ used in the public keys $(X', (X_i)_{i=1}^{\ell}) \in (G_2^*)^{\ell+1}$ and also unaware of the values y_j, $j \in [q]$ used for the computation of the signature

$$(Z_{1,j}, Z_{2,j}, Y_j, Y_j') = (x \sum_{i=1}^{\ell} x_i M_{j,i}, y_j \sum_{i=1}^{\ell} x_i M_{j,i}, y_j P, y_j P')$$

in the j-th signature query for equivalence class $[(M_{j,i})_{i=1}^{\ell}]_{\mathcal{R}}$. Moreover, the adversary is unaware of the values $m_{j,i}$ $j \in [q], i \in [\ell]$ from signing queries, as the messages are sampled by the challenger. To obtain a forgery $(Z_1^*, Z_2^*, Y^*, Y'^*)$, the generic adversary is restricted to choosing

$$\pi_{z_1}, \pi_{z_2}, \pi_y, \pi_{y'}, \eta_{z_1,j,i}, \eta_{z_2,j,i}, \eta_{y,j,i}, \rho_{z_1,j}, \rho_{z_2,j}, \rho_{y,j}, \psi_{z_1,j}, \psi_{z_2,j}, \psi_{y,j}, \phi_{z_1,j}, \phi_{z_2,j}, \phi_{y,j}, \phi_{y',j}, \chi_i, \chi \in \mathbb{Z}_p$$
$$\text{for } j \in [q] \text{ and } i \in [\ell]$$

and computing the forgery $(Z_1^*, Z_2^*, Y^*, Y'^*)$ for message $(M_i^*)_{i=1}^{\ell}$ as

$$Z_1^* = \pi_{z_1} P + \sum_j^q \sum_i^{\ell} \eta_{z_1,j,i} M_{j,i} + \sum_j^q \rho_{z_1,j} Z_{1,j} + \sum_j^q \psi_{z_1,j} Z_{2,j} + \sum_j^q \phi_{z_1,j} Y_j$$

$$Z_2^* = \pi_{z_2} P + \sum_j^q \sum_i^{\ell} \eta_{z_2,j,i} M_{j,i} + \sum_j^q \rho_{z_2,j} Z_{1,j} + \sum_j^q \psi_{z_2,j} Z_{2,j} + \sum_j^q \phi_{z_2,j} Y_j$$

$$Y^* = \pi_y P + \sum_j^q \sum_i^{\ell} \eta_{y,j,i} M_{j,i} + \sum_j^q \rho_{y,j} Z_{1,j} + \sum_j^q \psi_{y,j} Z_{2,j} + \sum_j^q \phi_{y,j} Y_j$$

$$Y'^* = \pi_{y'} P' + \chi X' + \sum_i^{\ell} \chi_i X_i' + \sum_j^q \phi_{y',j} Y_j'.$$

The queries $(M_{j,i})_{i=1}^{\ell}$ are chosen by the challenger and the message $(M_i^*)_{i=1}^{\ell}$, for which the forgery $(Z_1^*, Z_2^*, Y^*, Y'^*)$ is obtained, is computed as linear combination of

$$P, M_{1,1}, \ldots, M_{1,\ell}, \ldots M_{j,1}, \ldots, M_{j,\ell}, Z_{1,1}, Z_{2,1}, Y_1, \ldots, Z_{1,q}, Z_{2,q}, Y_q.$$

By considering all these group elements and taking their discrete logarithms to the basis P and P', respectively, we obtain

$$1, m_{1,1}, \ldots, m_{1,\ell}, \ldots, m_{j,1}, \ldots, m_{j,\ell}, x, x_i, z_{1,j}, z_{2,j}, y_j$$

and, consequently, we can express these discrete logarithms as the following linear combinations:

$$z_1^* = \pi_{z_1} + \sum_j^q \sum_i^{\ell} \eta_{z_1,j,i} m_{j,i} + \sum_j^q \rho_{z_1,j} z_{1,j} + \sum_j^q \psi_{z_1,j} z_{2,j} + \sum_j^q \phi_{z_1,j} y_j$$

$$z_2^* = \pi_{z_2} + \sum_j^q \sum_i^{\ell} \eta_{z_2,j,i} m_{j,i} + \sum_j^q \rho_{z_2,j} z_{1,j} + \sum_j^q \psi_{z_2,j} z_{2,j} + \sum_j^q \phi_{z_2,j} y_j$$

$$y^* = \pi_y + \sum_j^q \sum_i^{\ell} \eta_{y,j,i} m_{j,i} + \sum_j^q \rho_{y,j} z_{1,j} + \sum_j^q \psi_{y,j} z_{2,j} + \sum_j^q \phi_{y,j} y_j$$

$$y'^* = \pi_{y'} + \chi x + x \sum_i \chi_i x_i + \sum_j \phi_{y',j} y_j$$

$$m_i^* = \pi_{m_i^*} + \sum_j^q \sum_i^{\ell} \eta_{m_i^*,j,i} m_{j,i} + \sum_j^q \rho_{m_i^*,j} z_{1,j} + \sum_j^q \psi_{m_i^*,j} z_{2,j} + \sum_j^q \phi_{m_i^*,j} y_j$$

Plugging the forgery into the verification relations yields:

$$\prod_i e(M_i^*, X_i') = e(Z_1^*, P') \quad \wedge \quad e(Z_1^*, Y'^*) = e(Z_2^*, X') \quad \wedge \quad e(P, Y'^*) = e(Y^*, P').$$

By taking discrete logarithms to the basis $e(P, P')$ in G_T, we obtain the following equations:

$$x \sum_i^{\ell} m_i^* x_i = z_1^* \tag{1}$$
$$z_1^* y'^* = z_2^* x \tag{2}$$
$$y^* = y'^* \tag{3}$$

Before we begin, we recall that elements $z_{1,j}$, $z_{2,j}$ and y_j, $j \in [q]$ are of the form $m_{j,i} x x_i$, $m_{j,i} y_j x_i$ and y_j respectively. We start by considering Equation (1):

$$x \sum_i m_i^* x_i = z_1^*$$

$$x\Big(\sum_i^{\ell} \big(\pi_{m_i^*} + \sum_j^q \sum_i^{\ell} \eta_{m_i^*,j,i} m_{j,i} + \sum_j^q \rho_{m_i^*,j} z_{1,j} + \sum_j^q \psi_{m_i^*,j} z_{2,j} + \sum_j^q \phi_{m_i^*,j} y_j\big) x_i\Big) =$$
$$\pi_{z_1} + \sum_j^q \sum_i^{\ell} \eta_{z_1,j,i} m_{j,i} + \sum_j^q \rho_{z_1,j} z_{1,j} + \sum_j^q \psi_{z_1,j} z_{2,j} + \sum_j^q \phi_{z_1,j} y_j$$

We first investigate terms on the RHS. By comparing coeffcients in 1 we get that $\pi_{z_1} = 0$. Furthermore, by comparing coeffcients in $m_{j,i}$, we see that $\eta_{z_1,j,i} = 0$ for all $i \in [\ell], j \in [q]$. By comparing coefficients in y_j we see that $\phi_{z_1,j} = 0$ for all $j \in [q]$. Moreover, by comparing coefficients in $m_{j,i} y_j x_i$ we see that $\psi_{z_1,j} = 0$ for all $j \in [q]$. This yields a simplified Equation (1):

$$\sum_i \big(\pi_{m_i^*} x x_i + \sum_j^q \sum_i^{\ell} \eta_{m_i^*,j,i} m_{j,i} x x_i + \sum_j^q \rho_{m_i^*,j} z_{1,j} x x_i + \sum_j^q \psi_{m_i^*,j} z_{2,j} x x_i + \sum_j^q \phi_{m_i^*,j} y_j x x_i\big) = \sum_j^q \rho_{z_1,j} z_{1,j}$$

Now, we investigate terms on the LHS. Comparing coefficients in $x x_i$ yields that $\pi_{m_i^*} = 0$ for all $i \in [\ell]$. Comparing coefficients in $m_{j,i} x^2 x_i x_{i'}$ yields that $\rho_{m_i^*,j} = 0$ for all $i, i' \in [\ell], j \in [q]$. Comparing coefficients in $m_{j,i} y_j x x_i x_{i'}$ yields that $\psi_{m_i^*,j} = 0$ for all $ii', \in [\ell], j \in [q]$. Moreover, comparing coefficients in $y_j x x_i$ gives that $\phi_{m_i^*,j} = 0$ for all $i \in [\ell], j \in [q]$. Consequently, we obtain that:

$$z_1^* = \sum_j^q \rho_{z_1,j} z_{1,j}. \tag{4}$$

and

$$m_i^* = \sum_j^q \sum_i^{\ell} \eta_{m_i^*,j,i} m_{j,i}. \tag{5}$$

By looking at Equation (3):

$$\pi_y + \sum_j^q \sum_i^{\ell} \eta_{y,j,i} m_{j,i} + \sum_j^q \rho_{y,j} z_{1,j} + \sum_j^q \psi_{y,j} z_{2,j} + \sum_j^q \phi_{y,j} y_j = \pi_{y'} + \chi x + x \sum_i^{\ell} \chi_i x_i + \sum_j^q \phi_{y',j} y_j$$

and comparing coefficients in x and $x x_i$, we derive that $\chi = 0$ and $\chi_i = 0$ for all $i \in [\ell]$. Moreover, by comparing coefficients in $m_{j,i}$, $m_{j,i} x x_i$ and $m_{j,i} y_j x_i$ we derive that $\eta_{y,j,i} = 0$ for all $i \in [\ell], j \in [q]$ as well as $\rho_{y,j} = 0$ and $\psi_{y,j} = 0$ for all $j \in [q]$. By comparing coefficients in 1 and y_j we further see that $\pi_y = \pi_{y'}$ and $\phi_{y,j} = \phi_{y',j}$ for all $j \in [q]$ respectively. This yields the simplified representation

$$y'^* = \pi_y + \sum_j^q \phi_{y,j} y_j. \tag{6}$$

Using (4) and (6) in Equation (2), we obtain:

$$(\sum_j^q \rho_{z_1,j} z_{1,j})(\pi_y + \sum_k^q \phi_{y,k} y_k) = x(\pi_{z_2} + \sum_j^q \sum_i^\ell \eta_{z_2,j,i} m_{j,i} + \sum_j^q \rho_{z_2,j} z_{1,j} + \sum_j^q \psi_{z_2,j} z_{2,j} + \sum_j^q \phi_{z_2,j} y_j)$$

$$\pi_y \sum_j^q \rho_{z_1,j} z_{1,j} + \sum_j^q \rho_{z_1,j} z_{1,j} \sum_k^q \phi_{y,k} y_k = \pi_{z_2} x + \sum_j^q \sum_i^\ell \eta_{z_2,j,i} m_{j,i} x + \sum_j^q \rho_{z_2,j} z_{1,j} x + \sum_j^q \psi_{z_2,j} z_{2,j} x + \sum_j^q \phi_{z_2,j} y_j x$$

Starting on the RHS, by comparing coefficients in x we see that $\pi_{z_2} = 0$. Moreover, by comparing coefficients in $y_k x$ we see that $\phi_{z_2,k} = 0$ for all $k \in [q]$. Additionally, comparing coefficients in $m_{j,i} x^2 x_i$ and $m_{j,i} x$ yields that $\rho_{z_2,j} = 0$ for all $j \in [q]$ and $\eta_{z_2,j,i} = 0$ for all $i \in [\ell], j \in [q]$ respectively. This gives us the following simplified equation:

$$\pi_y \sum_j^q \rho_{z_1,j} z_{1,j} + \sum_j^q \rho_{z_1,j} z_{1,j} \sum_k^q \phi_{y,k} y_k = x \sum_j^q \psi_{z_2,j} z_{2,j}$$

Looking at the LHS, by equating coeffcients in $m_{j,i} x x_i$, we see that $\pi_y \rho_{z_1,j} = 0$ for all $j \in [q]$. As, however, it needs to hold that $\rho_{z_1,j} \neq 0$ for some $j \in [q]$ (as otherwise z_1^* in (4) would vanish, but the all 0_{G_1} message is not a valid forgery), it follows that $\pi_y = 0$. Consequently, we obtain that $\pi_{y'} = 0$. All together, this gives us that:

$$y^* = y'^* \qquad \sum_j^q \phi_{y,j} y_j = \sum_j^q \phi_{y',j} y_j \tag{7}$$

and

$$z_2^* = \sum_n^q \psi_{z_2,n} z_{2,n} \tag{8}$$

Using (4), (7) and (8) in Equation (2) gives us that:

$$z_1^* y'^* = z_2^* x \qquad \sum_j^q \rho_{z_1,j} z_{1,j} \sum_k^q \phi_{y,k} y_k = \sum_n^q \psi_{z_2,n} z_{2,n} x \tag{9}$$

We know that y^* (by definition) and z_1^* (as the all 0_{G_1} message is not a valid forgery) cannot vanish. Hence, also z_2^* cannot vanish. As we know that $z_{2,n} x = z_{1,n} y_n$ by definition, we can substitute in (9) and obtain

$$\sum_j^q \rho_{z_1,j} z_{1,j} \sum_k^q \phi_{y,k} y_k = \sum_n^q \psi_{z_2,n} z_{1,n} y_n \tag{10}$$

We observe that for each $j \in [q]$ and all $i \in [\ell]$ terms with variables $m_{j,i}$ can only occur in $z_{1,j}$ on the LHS and in $z_{1,n}$ with $n = j$ on the RHS. Consequently, $\rho_{z_1,j} \phi_{y,k} = 0$ for all $j, k \in [q], j \neq k$. As the LHS however cannot vanish, we know that there exists $k \in [q]$ such that $\rho_{z_1,k} \phi_{y,k} \neq 0$. Considering the matrix $C = (\rho_{z_1,j})_{j \in [q]} (\phi_{y,k})_{k \in [q]}^\top$ we thus know that C is a non-zero diagonal matrix. As from basic linear algebra we have that $\mathrm{rank}((\rho_{z_1,j})_{j \in [q]}) = \mathrm{rank}((\phi_{y,k})_{k \in [q]}^\top) = 1$ and $\mathrm{rank}(C) \leq \min\{(\rho_{z_1,j})_{j \in [q]}, (\phi_{y,k})_{k \in [q]}^\top\}$, we obtain that all but one of the elements of the diagonal $(\rho_{z_1,k} \phi_{y,k})_{k \in [q]}$ (henceforth indexed by k^*) are zero. Consequently, we can simplify Equation (10) to

$$\rho_{z_1,k^*} z_{1,k^*} \phi_{y,k^*} y_{k^*} = \sum_n^q \psi_{z_2,j} z_{1,j} y_j$$

and when comparing coefficients in $m_{k^*,i}$ it follows that $\psi_{z_2,j} = 0$ for all $j \in [q], j \neq k^*$. Going back to Equation (1), using (5) and substituting the definition of z_{1,k^*} in the second line, we obtain:

$$x\big(\sum_i^{\ell}\big(\sum_j^{q}\sum_n^{\ell}\eta_{m_i^*,j,n}m_{j,n}\big)x_i\big) = \rho_{z_1,k^*}z_{1,k^*}$$

$$x\big(\sum_i^{\ell}\big(\sum_j^{q}\sum_n^{\ell}\eta_{m_i^*,j,n}m_{j,n}\big)x_i\big) = \rho_{z_1,k^*}x\sum_r^{\ell}m_{k^*,r}x_r$$

We observe, that on the RHS there is exactly one monomial $m_{k^*,r}x_r$ for each $r \in [\ell]$ (note that $\rho_{z_1,k^*} \neq 0$). Investigating the LHS, we see the following: For each $i \in [\ell]$ in the double-sum, all monomials $m_{j,i}$ with $j \neq k^*, n \neq r$ when comparing coefficients in $m_{j,i}x_r$ cannot appear on the RHS. Thus, $\eta_{m_r^*,j,n} = 0$ for all $j \neq k^*, n \neq r$. Consequently, the double-sum collapses and we can simplify Equation (1) to

$$x\big(\sum_i^{\ell}\big(\eta_{m_i^*,k^*,i}m_{k^*,i}\big)x_i\big) = \rho_{z_1,k^*}x\sum_i^{\ell}m_{k^*,i}x_i$$

Moreover, when comparing coefficients in $m_{k^*,i}$, we see that $\eta_{m_i^*,k^*,i} = \rho_{z_1,k^*}$ for all $i \in [\ell]$ and consequently, we can replace all $\eta_{m_i^*,k^*,i}$ by η and we can further simplify Equation (1) to

$$\eta x\sum_i^{\ell}m_{k^*,i}x_i = \rho_{z_1,k^*}x\sum_i^{\ell}m_{k^*,i}x_i$$

Hence, the only forgeries the adversary can produce are only representatives of classes, for which the adversary has already obtained a signatures via a signing oracle query.

Now, in the second part of this proof, we show that the probability for an adversary to produce an existential forgery by "incident" is negligible, i.e., that two formally different polynomials evaluate to the same value or actually that the difference polynomial evaluates to zero. All involved formal polynomials resulting from querying the group oracles are of degree $O(1)$, when we assume that the adversary makes $O(q)$ queries to the group oracles. Then, by using the Schwartz-Zippel lemma and a collision argument, we know that the probability of such an error in the simulation of the generic group is $O(\frac{q^2}{p})$ and is, therefore, negligible. $\square$

B.3 Proof of Theorem 3 (Class Hiding)

Proof. For the stronger notion of class hiding*, we need to be able to check a key pair $(\mathsf{sk}, \mathsf{pk})$ output by the adversary for consistency. It is easy to see that given $\mathsf{sk} = (x_i)_{i\in[\ell]} \in \mathbb{Z}_p^{\ell}$ and $\mathsf{pk} = (X_i')_{i\in[\ell]} \in (G_2^*)^{\ell}$ one can efficiently check whether $x_iP' = X_i'\ \forall i \in [\ell]$. Moreover, we see from the correctness proof that signatures obtained from $\mathsf{ChgRep}_{\mathcal{R}}$ are identically distributed as signatures obtained from $\mathsf{Sign}_{\mathcal{R}}$.

We now use the following sequence of games to show that Scheme 1 is class hiding. We define $\mathsf{Game}_{\mathsf{real}}$ to be the game where the $\mathcal{O}^{RoR}$ oracle returns re-randomizations of messages (with corresponding signatures) that have already been queried to the $\mathcal{O}^{RM}$ oracle, and $\mathsf{Game}_{\mathsf{random}}$ to be the game where the real-or-random oracle $\mathcal{O}^{RoR}$ returns purely random messages (with corresponding signatures). More precisely, in $\mathsf{Game}_{\mathsf{real}}$ the $\mathcal{O}^{RoR}$ oracle re-randomizes a message $(M_i)_{i=1}^{\ell}$ drawn by the $\mathcal{O}^{RM}$ oracle by returning $\mathsf{ChgRep}_{\mathcal{R}}((M_i)_{i=1}^{\ell}, \mathsf{Sign}_{\mathcal{R}}((M_i)_{i=1}^{\ell}, \mathsf{sk}), \rho, \mathsf{pk})$ for $\rho \xleftarrow{R} \mathbb{Z}_p^*$, whereas in $\mathsf{Game}_{\mathsf{random}}$ the $\mathcal{O}^{RoR}$ oracle just returns $(R_i)_{i=1}^{\ell} \xleftarrow{R} (G_1^*)^{\ell}$ together with $\sigma \leftarrow \mathsf{Sign}_{\mathcal{R}}((R_i)_{i=1}^{\ell}, \mathsf{sk})$. For $j = 1, \ldots, \ell$, we define Game_j to be the game where the $\mathcal{O}^{RoR}$ oracle returns $(\rho M_1, \ldots, \rho M_j, R_{j+1}, \ldots, R_{\ell})$ with corresponding signature for $\rho \xleftarrow{R} \mathbb{Z}_p^*$ and $R_{j+1}, \ldots, R_{\ell} \xleftarrow{R} G_1^*$ after $(M_1, \ldots, M_{\ell})$ has been randomly sampled by the $\mathcal{O}^{RM}$ oracle. Moreover, we define $\mathsf{Game}'_{\mathsf{real}}$ to be a modification of $\mathsf{Game}_{\mathsf{real}}$ where the $\mathcal{O}^{RoR}$ oracle replaces the execution of $(\rho M, \sigma') \leftarrow \mathsf{ChgRep}_{\mathcal{R}}(M, \sigma, \rho, \mathsf{sk})$ by $(\rho M, \mathsf{Sign}_{\mathcal{R}}(\rho M, \mathsf{sk}))$. Consequently, instead of re-randomizing the message-signature pair, it computes a fresh signature on the randomized message. Thereby, the message-signature pair output by $\mathcal{O}^{RoR}$ in $\mathsf{Game}'_{\mathsf{real}}$ is perfectly indistinguishable from the message-signature pair output by $\mathcal{O}^{RoR}$ in $\mathsf{Game}_{\mathsf{real}}$. We analogously derive Game'_j from Game_j for $j \in [\ell]$.

Now, given a DDH instance (P, aP, bP, cP) we simulate the following game for the adversary for $j \in [\ell]$ as follows. The $\mathcal{O}^{RM}$ oracle returns messages of the form $(m_1P, \ldots, m_{j-1}P, m_jaP, M_{j+1}, \ldots, M_{\ell})$, whereas the $\mathcal{O}^{RoR}$ oracle returns a message of the form $(m_1bP, \ldots, m_{j-1}bP, m_jcP, R_{j+1}, \ldots, R_\ell)$ and its corresponding signature σ. If (P, aP, bP, cP) is a valid DDH instance (that is $c = ab$), then the simulation of Game'_j is perfect, whereas we have perfectly simulated Game'_{j-1} otherwise. Furthermore, we have $\mathsf{Game}'_1 = \mathsf{Game}_{\mathsf{random}}$ and given that (P, aP, bP, cP) is a valid DDH instance, we have $\mathsf{Game}'_\ell = \mathsf{Game}'_{\mathsf{real}}$. Therefore, if the adversary is able to distinguish Game'_{j-1} from Game'_j for any $j \in [2, \ell]$ with non-negligible probability $\epsilon(\kappa)$, then it can also be used to break the DDH assumption with non-negligible probability $\epsilon(\kappa)$. All in all, the probability to decide the class hiding property, i.e., to distinguish $\mathsf{Game}_{\mathsf{random}}$ from $\mathsf{Game}_{\mathsf{real}}$, is bound by $(\ell - 1) \cdot \epsilon(\kappa)$. □

C Security of the ABC System

In the following, we provide a security model for attribute-based anonymous credentials. Then, we prove the unforgeability and the anonymity of our proposed scheme. The proof of correctness is omitted, as the correctness of Scheme 2 can easily be verified by the construction.

C.1 Security of ABCs

The subsequent security model is adapted from [25,8,29,30]. Before we present it in all detail, we give a high-level overview of the required security properties and we note that we consider only a single organization (identified by $j = 1$) in our model of unforgeability and anonymity (since all organizations have independent signing keys, the extension is straightforward):

Correctness: A showing of a credential with respect to a set $\mathbb{A}'$ of attributes and values must always verify if the credential was issued honestly with respect to $\mathbb{A}$ and it holds that $\mathbb{A}' \sqsubseteq \mathbb{A}$.

Unforgeability: A user can not show a valid credential for some $\mathbb{A}$, unless this credential was issued to him by an organization for $\mathbb{A}$. Furthermore, it must not be possible to succeed in a showing protocol without having access to the user's secret key and the credential (replay).

Anonymity: Given a showing, no verifier and no organization (even if they collude) should be able to identify the user or find anything about the user, except for the fact that he owns a valid credential. Furthermore, different showings of a user with respect to the same credential must be unlinkable.

In the following, we provide formal definitions of these properties. To do so, we introduce several global variables and oracles. In order to keep track of all, honest and corrupt users as well as users, whose secret keys and credentials have leaked, we introduce the sets $\mathtt{U}$, $\mathtt{HU}$, $\mathtt{CU}$ and $\mathtt{KU}$, respectively. All these sets are maintained by the environment and available to the adversary for read access. We use the lists $\mathtt{UPK}$, $\mathtt{USK}$, $\mathtt{CRED}$, $\mathtt{SCRED}$ and $\mathtt{ATTR}$ to track issued user keys, credentials, shown credentials and corresponding attributes, which are only accessible to the environment.

We introduce the subsequent oracles and assume that the public parameters pp are implicitly available to them:

$\mathcal{O}^{\mathtt{HU}+}(i)$**:** It takes input a user identity i. If $i \in \mathtt{U}$ return $\perp$. Otherwise, it creates a new user i by running $(\mathtt{USK}[i], \mathtt{UPK}[i]) \leftarrow \mathsf{UserKeyGen}(\mathsf{pp}, i)$, adding i to $\mathtt{U}$ and to $\mathtt{HU}$ and returning $\mathtt{UPK}[i]$.

$\mathcal{O}^{\mathtt{CU}+}(\mathsf{pk}, i)$**:** It takes input a user public key pk and a user i. If $i \notin \mathtt{U}$ or $i \in \mathtt{CU}$ return $\perp$. Otherwise, it adds user i to the set of corrupted users $\mathtt{CU}$, removes i from $\mathtt{HU}$, and sets $\mathtt{UPK}[i] \leftarrow \mathsf{pk}$.

$\mathcal{O}^{\mathtt{KU}+}(i)$**:** It takes input a user i. If $i \notin \mathtt{U}$ or $i \in \mathtt{KU}$ return $\perp$. Otherwise, it reveals the credentials and the secret key of user i by returning $\mathtt{USK}[i]$ and all credentials in $\mathtt{CRED}$, which have been issued for i. Finally, it adds i to $\mathtt{KU}$.

$\mathcal{O}^{U_I O_O}(\mathsf{osk}, \mathsf{opk}, i, \mathbb{A})$**:** It takes input the organization key pair $(\mathsf{osk}, \mathsf{opk})$, a user i and a set of attributes $\mathbb{A}$. If $i \notin \mathtt{HU}$ return $\perp$. Otherwise, it issues a credential cred_i on $\mathbb{A}$ for an honest user $i \in \mathtt{HU}$. Here, the oracle plays the role of the user as well as the organization. It runs

$$(\mathsf{cred}_i, \emptyset) \leftarrow (\mathsf{Obtain}(\mathsf{pp}, \mathtt{USK}[i], \mathsf{opk}, \mathbb{A}), \mathsf{Issue}(\mathsf{pp}, \mathtt{UPK}[i], \mathsf{osk}, \mathbb{A})).$$

Finally, it appends $(\mathsf{cred}_i, \mathbb{A})$ to $(\mathtt{CRED}, \mathtt{ATTR})$ where the caller does not get any output.

$\mathcal{O}^{U_I}(\mathsf{osk}, \mathsf{opk}, i, \mathbb{A})$**:** It takes input the organization key pair $(\mathsf{osk}, \mathsf{opk})$, a user i and a set of attributes $\mathbb{A}$. If $i \notin \mathtt{HU}$ return $\perp$. Otherwise, it plays the role of an honest user who gets issued a credential for $\mathbb{A}$. It runs

$$(\mathsf{cred}_i, \emptyset) \leftarrow (\mathsf{Obtain}(\mathsf{pp}, \mathtt{USK}[i], \mathsf{opk}, \mathbb{A}), \mathsf{Issue}(\mathsf{pp}, \mathtt{UPK}[i], \mathsf{osk}, \mathbb{A})),$$

where Obtain is run on behalf of honest user i and Issue is executed by the caller (the dishonest organization). Finally, it appends $(\mathsf{cred}_i, \mathbb{A})$ to $(\mathtt{CRED}, \mathtt{ATTR})$.

$\mathcal{O}^{O_\mathsf{O}}(\mathsf{osk}, \mathsf{opk}, i, \mathsf{usk}_i, \mathbb{A})$**:** It takes input the organization key pair $(\mathsf{osk}, \mathsf{opk})$, a user i, a user secret key usk_i and a set of attributes $\mathbb{A}$. If $i \notin \mathtt{CU}$ return $\perp$. Otherwise, it plays the role of the organization when interacting with a dishonest user, i.e., a corrupted user whose public key has been replaced (thus the corresponding secret key usk_i is not stored in $\mathtt{USK}$). It runs

$$(\mathsf{cred}_i, \emptyset) \leftarrow (\mathsf{Obtain}(\mathsf{pp}, \mathsf{usk}_i, \mathsf{opk}, \mathbb{A}), \mathsf{Issue}(\mathsf{pp}, \mathtt{UPK}[i], \mathsf{osk}, \mathbb{A})),$$

where Obtain is executed by the caller. Finally, it appends $(\mathsf{cred}_i, \mathbb{A})$ to $(\mathtt{CRED}, \mathtt{ATTR})$.

$\mathcal{O}^{U_\mathsf{V}}(\mathsf{opk}, j, \mathbb{A}')$**:** It takes input the organization public key opk, an index of an issuance j and a set of attributes $\mathbb{A}'$ certified to the user i_j. If $i_j \notin \mathtt{HU}$ return $\perp$. Otherwise, it plays the role of an honest user i_j and runs

$$(\emptyset, b) \leftarrow (\mathsf{Show}(\mathsf{pp}, \mathtt{USK}[i_j], \mathsf{opk}, (\mathtt{ATTR}[j], \mathbb{A}'), \mathtt{CRED}[j]), \mathsf{Verify}(\mathsf{pp}, \mathsf{opk}, \mathbb{A}')),$$

where Verify is executed by the caller (the dishonest verifier). If $b = \mathtt{true}$, then it appends the shown credential cred to $\mathtt{SCRED}[j]$.

$\mathcal{O}^{RoR}(\mathsf{osk}, \mathsf{opk}, b, j_0, \mathbb{A}')$**:** It takes input the organization key pair $(\mathsf{osk}, \mathsf{opk})$, a bit b, an index of an issuance j_0 and a set of attributes $\mathbb{A}'$. If this oracle has already been queried for some $j_0' \neq j_0$, or if $\mathtt{CRED}[j_0] = \perp$, $i_{j_0} \notin \mathtt{HU}$ or $\mathbb{A}' \not\sqsubseteq \mathtt{ATTR}[j_0]$ return $\perp$. Otherwise, on the first call, it generates a credential for some new, random user i_{j_1} and the attribute set $\mathtt{ATTR}[j_0]$. It plays the role of user i_{j_b} and interacts with the adversary during an execution of the $(\mathsf{Show}, \mathsf{Verify})$ protocol for the attributes $\mathbb{A}'$.

Now, we are ready to give an exact definition of a secure attribute-based anonymous credential system:

Definition 19 (Correctness). An anonymous credential system is *correct*, if

$$\begin{gathered}\forall \kappa > 0\ \forall t > 0\ \forall \mathbb{A} : \#\mathbb{A} \leq t\ \forall j\ \forall i\ \forall \mathsf{pp} \leftarrow \mathsf{Setup}(\kappa, t),\\ (\mathsf{osk}_j, \mathsf{opk}_j) \leftarrow \mathsf{OrgKeyGen}(\mathsf{pp}, j), (\mathsf{usk}_i, \mathsf{upk}_i) \leftarrow \mathsf{UserKeyGen}(\mathsf{pp}, i),\\ (\mathsf{cred}_i, \emptyset) \leftarrow (\mathsf{Obtain}(\mathsf{pp}, \mathsf{usk}_i, \mathsf{opk}_j, \mathbb{A}), \mathsf{Issue}(\mathsf{pp}, \mathsf{upk}_i, \mathsf{osk}_j, \mathbb{A})) \text{ it holds that}\\ (\emptyset, \mathtt{true}) \leftarrow (\mathsf{Show}(\mathsf{pp}, \mathsf{usk}_i, \mathsf{opk}_j, (\mathbb{A}, \mathbb{A}'), \mathsf{cred}_i), \mathsf{Verify}(\mathsf{pp}, \mathsf{opk}_j, \mathbb{A}'))\ \forall \mathbb{A}' \sqsubseteq \mathbb{A}.\end{gathered}$$

Definition 20 (Unforgeability). We call an attribute-based anonymous credential system *unforgeable*, if for all PPT-adversaries $\mathcal{A}$ there is a negligible function $\epsilon(\cdot)$ such that

$$\Pr\left[\begin{array}{c}\mathsf{pp} \leftarrow \mathsf{Setup}(\kappa, t), (\mathsf{osk}, \mathsf{opk}) \leftarrow \mathsf{OrgKeyGen}(\mathsf{pp}, 1),\\ \mathcal{O} \leftarrow \{\mathcal{O}^{\mathtt{HU}+}(\cdot), \mathcal{O}^{\mathtt{CU}+}(\cdot,\cdot), \mathcal{O}^{\mathtt{KU}+}(\cdot), \mathcal{O}^{U_\mathsf{I} O_\mathsf{O}}(\mathsf{osk}, \mathsf{opk}, \cdot, \cdot), \mathcal{O}^{O_\mathsf{O}}(\mathsf{osk}, \mathsf{opk}, \cdot, \cdot, \cdot), \mathcal{O}^{U_\mathsf{V}}(\mathsf{opk}, \cdot, \cdot)\},\\ (\mathbb{A}'^*, \mathsf{state}) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{pp}, \mathsf{opk}),\ \ (\emptyset, b^*) \leftarrow (\mathcal{A}(\mathsf{state}), \mathsf{Verify}(\mathsf{pp}, \mathsf{opk}, \mathbb{A}'^*)) :\\ b^* = \mathtt{true}\ \wedge\\ \Big(j^* = \perp\ \vee\ \Big(j^* \neq \perp\ \wedge\ \big(\mathbb{A}'^* \not\sqsubseteq \mathtt{ATTR}[j^*]\ \vee\ (i^*_{j^*} \in \mathtt{HU} \setminus \mathtt{KU}\ \wedge\ \mathsf{cred}^* \in \mathtt{SCRED}[j^*])\big)\Big)\Big)\end{array}\right] \leq \epsilon(\kappa),$$

where cred^* is the credential shown by the adversary, $i^*_{j^*}$ is the user, who obtained the corresponding credential in the j^*-th issuing query. Thereby, $\perp$ indicates that no such index j^* exist. We note that the reduction needs to be able to efficiently determine j^* given the shown credential.

The winning conditions in the unforgeability game are chosen following the subsequent rationale. The first condition ($j^* = \perp$) captures showings of credentials, which have never been issued (existential forgeries). The second condition ($j^* \neq \perp \wedge \mathbb{A}'^* \not\sqsubseteq \mathtt{ATTR}[j^*]$) captures showings with respect to existing credentials, but invalid attribute sets. The third condition ($j^* \neq \perp \wedge i^*_{j^*} \in \mathtt{HU} \setminus \mathtt{KU}\ \wedge\ \mathsf{cred}^* \in \mathtt{SCRED}[j^*]$) covers replays of showings with respect to users where the adversary does neither know the credentials nor the respective secrets.

Definition 21 (Anonymity). We call an attribute-based anonymous credential system *anonymous*, if for all PPT-adversaries $\mathcal{A}$ there is a negligible function $\epsilon(\cdot)$ such that

$$\Pr\left[\begin{array}{c}\mathsf{pp} \leftarrow \mathsf{Setup}(\kappa, t),\ b \xleftarrow{R} \{0,1\},\ (\mathsf{osk}, \mathsf{opk}) \leftarrow \mathsf{OrgKeyGen}(\mathsf{pp}, 1)\\ \mathcal{O} \leftarrow \{\mathcal{O}^{\mathtt{HU}+}(\cdot), \mathcal{O}^{U_\mathsf{I}}(\mathsf{osk}, \mathsf{opk}, \cdot, \cdot), \mathcal{O}^{U_\mathsf{V}}(\mathsf{opk}, \cdot, \cdot), \mathcal{O}^{RoR}(\mathsf{osk}, \mathsf{opk}, b, \cdot, \cdot)\}\\ b^* \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{osk}, \mathsf{opk}, \mathsf{pp}, \mathsf{osk}, \mathsf{opk}) :\\ b^* = b\end{array}\right] - \frac{1}{2} \leq \epsilon(\kappa).$$

Definition 22 (Security). An attribute-based anonymous credential system is *secure*, if it is correct, unforgeable and anonymous.

We emphasize that we *do not* consider non-transferability of credentials in our model, since this is typically achieved by other means. This could, for instance, be achieved by requiring users to use an already existing valuable secret as secret key in order to prevent them from sharing the credential (PKI-assured [46]). Other approaches are that sharing the credential of one organization implies sharing all credentials (all-or-nothing [25]) or to require the presence of biometric features in order to use credentials (biometrics based [17]). In practice, a standard way to achieve non-transferability is to embed the user's secrets into a tamper proof hardware such as a smart card.

C.2 Proof of Theorem 5 (Unforgeability)

Proof. We assume that there is an efficient adversary $\mathcal{A}$ winning the unforgeability game with non-negligible probability, then we are able to use $\mathcal{A}$ for reductions in the following way.

Type 1: Adversary $\mathcal{A}$ manages to conduct a showing protocol accepted by the verifier such that $j^* = \perp$ holds. Then, we construct an adversary $\mathcal{B}$ that uses $\mathcal{A}$ to break the unforgeability of the SPS-EQ-$\mathcal{R}$ scheme.

Type 2: Adversary $\mathcal{A}$ manages to conduct a showing protocol accepted by the verifier using the j^*-th issued credential of user i^* with respect to $\mathbb{A}'^*$ such that $\mathbb{A}'^* \not\sqsubseteq \mathtt{ATTR}[j^*]$ holds. Then, we construct an adversary $\mathcal{B}$ that uses $\mathcal{A}$ to break

Type 2A: the hash function used in the encoding of attributes.

Type 2B: the factor soundness property of $\mathsf{PolyCommitFO}$.

Type 3: Adversary $\mathcal{A}$ manages to conduct a showing protocol accepted by the verifier reusing a showing based on the j^*-th issued credential of user i^* with $i^* \in \mathtt{HU} \setminus \mathtt{KU}$, whose secret usk_{i^*} and issued credentials it does not know. This means that in any case $\mathcal{A}$ is able to produce a valid PoK. Then, we construct an adversary $\mathcal{B}$ that uses $\mathcal{A}$ to break

Type 3A: the DLP in G_1 (with respect to Q).

Type 3B: the DLP in G_1 (with respect to C_2).

In the following, $\mathcal{B}$ guesses $\mathcal{A}$'s strategy, i.e., the type of forgery $\mathcal{A}$ will conduct. We are now going to describe the setup, the initialization of the environment, the reduction and the abort conditions for each type.

Type 1: Here, $\mathcal{B}$ consists of adversary $\mathcal{A}$ playing the unforgeability game with a challenger $\mathcal{S}$. $\mathcal{B}$ is interacting with the challenger $\mathcal{C}$ in the unforgeability game of the SPS-EQ-$\mathcal{R}$ scheme. Here, $\mathcal{B}$ runs algorithm $\mathcal{A}$ and plays the challenger $\mathcal{S}$ for $\mathcal{A}$ in the unforgeability game. Subsequently, we describe how $\mathcal{S}$ simulates the environment for $\mathcal{A}$ and interacts with the challenger $\mathcal{C}$ for the $\mathsf{EUF\text{-}CMA}$ game.

$\mathcal{C}$ is in possession of $(\mathsf{sk}, \mathsf{pk})$ for the signature scheme with $\ell = 2$ and gives pk to $\mathcal{B}$. Then, $\mathcal{S}$ sets $\mathsf{opk} \leftarrow \mathsf{pk}$ and generates pp in way compatible to opk (note that $\mathcal{S}$ has no direct access to sk and therefore must access the signing oracle of $\mathcal{C}$). Next, $\mathcal{S}$ runs $\mathcal{A}(\mathsf{pp}, \mathsf{opk})$ and simulates the environment and the oracles. All oracles are as in the real game, except for the oracles $\mathcal{O}^{U_\mathrm{I} O_\mathrm{O}}$ and $\mathcal{O}^{O_\mathrm{O}}$, which are simulated as follows:

$\mathcal{O}^{U_\mathrm{I} O_\mathrm{O}}(\mathsf{osk}, \mathsf{opk}, i, \mathbb{A})$: $\mathcal{S}$ runs this oracle as in the real game, with the only difference that whenever $\mathcal{S}$ requires a signature during the issuing protocol, it calls the signing oracle $\mathcal{O}(\mathsf{osk}, \cdot)$ of $\mathcal{C}$ with respective message (C_1, P).

$\mathcal{O}^{O_\mathrm{O}}(\mathsf{osk}, \mathsf{opk}, i, \mathsf{usk}_i, \mathbb{A})$: $\mathcal{S}$ runs this oracle as in the real game, with the only difference that whenever $\mathcal{S}$ requires a signature during the issuing protocol, it calls the signing oracle $\mathcal{O}(\mathsf{osk}, \cdot)$ of $\mathcal{C}$ with respective message (C_1, P).

If $\mathcal{A}$ outputs $(\mathbb{A}'^*, \mathsf{state})$, then $\mathcal{S}$ runs $\mathcal{A}(\mathsf{state})$ and interacts with $\mathcal{A}$ as verifier in a showing protocol. Now, if $\mathcal{A}$ delivers a valid showing using a credential cred'^* and, thus, wins the game, then $\mathcal{S}$ rewinds $\mathcal{A}$ to the step after sending the commitments (K_Q, K_{C_2}) in PoK and restarts $\mathcal{A}$ with a new challenge $c' \neq c$. Then, by the knowledge extractor of PoK, $\mathcal{S}$ obtains ρ (such that $C_2 = \rho P$). $\mathcal{S}$ now computes $\mathsf{cred}^* \leftarrow \rho^{-1} \cdot \mathsf{cred}'^*$. If $\mathsf{cred}^* \in \mathtt{CRED}$ then $\mathcal{S}$ and, in further consequence, $\mathcal{B}$ abort. In this case, the credential was honestly computed and a signing query was issued to the signing oracle $\mathcal{O}$ of $\mathcal{C}$. Otherwise, $\mathcal{B}$ outputs $\mathsf{cred}^* = ((C_1^*, C_2^*), \sigma^*)$ as a forgery to $\mathcal{C}$ and $\mathcal{B}$ wins the unforgeability game.

Type 2A: Here, $\mathcal{B}$ plays the role of the challenger for $\mathcal{A}$. $\mathcal{B}$ runs the setup by generating public parameters pp, generates the organization key pair $(\mathsf{osk}, \mathsf{opk})$, runs $\mathcal{A}(\mathsf{pp}, \mathsf{opk})$ and simulates the oracles as in the real game.

If $\mathcal{A}$ outputs $(\mathbb{A}'^*, \mathsf{state})$, then $\mathcal{B}$ runs $\mathcal{A}(\mathsf{state})$ and interacts with $\mathcal{A}$ as verifier in a showing protocol. Now, if $\mathcal{A}$ delivers a valid showing using a credential cred'^*, then $\mathcal{B}$ rewinds $\mathcal{A}$ to the step after sending the commitments (K_Q, K_{C_2}) in PoK and restarts $\mathcal{A}$ with a new challenge $c' \neq c$. Then, by the knowledge extractor of PoK, $\mathcal{B}$ obtains ρ (such that $C_2 = \rho P$). $\mathcal{B}$ now computes $\mathsf{cred}^* \leftarrow \rho^{-1} \cdot \mathsf{cred}'^*$. If $\mathsf{cred}'^* \notin \mathtt{CRED}$, then $\mathcal{B}$ aborts. Otherwise, we know that cred^* was the result of the j^*-th issue step during the simulation. Consequently, $\mathcal{B}$ knows the set of attributes $\mathbb{A}^* = \mathtt{ATTR}[j^*]$ corresponding to cred^*. If $\mathbb{A}'^* \sqsubseteq \mathbb{A}^*$, then $\mathcal{B}$ aborts. Otherwise, $\mathcal{B}$ can now compute the corresponding polynomial $\mathsf{enc}(\mathbb{A}^*)$. If $\mathsf{enc}(\mathbb{A}'^*) \nmid \mathsf{enc}(\mathbb{A}^*)$, then $\mathcal{B}$ aborts. Otherwise, we have that $\mathsf{enc}(\mathbb{A}'^*) \mid \mathsf{enc}(\mathbb{A}^*)$, but $\mathbb{A}'^* \not\sqsubseteq \mathbb{A}^*$. Therefore, there is at least one factor $X - H(\mathtt{attr}^*{}_\ell \| M^*)$ of $\mathsf{enc}(\mathbb{A}'^*)$ and one factor $X - H(\mathtt{attr}_\ell \| M)$ of $\mathsf{enc}(\mathbb{A}^*)$ such that $H(\mathtt{attr}^*{}_\ell \| M^*) = H(\mathtt{attr}_\ell \| M)$ and $\mathtt{attr}^*{}_\ell \| M^* \neq \mathtt{attr}_\ell \| M$. Consequently, $\mathcal{B}$ outputs the pair $(\mathtt{attr}^*{}_\ell \| M^*, \mathtt{attr}_\ell \| M)$ as a collision for H.

Type 2B: Here $\mathcal{B}$, consists of adversary $\mathcal{A}$ playing the unforgeability game with a challenger $\mathcal{S}$. $\mathcal{B}$ is interacting with the challenger $\mathcal{C}$ in the factor soundness game of the $\mathsf{PolyCommitFO}$ scheme.

$\mathcal{C}$ chooses the public parameters pp' of $\mathsf{PolyCommitFO}$ and runs $\mathcal{B}$ on pp'. Then, $\mathcal{S}$ completes the setup by generating public parameters pp based on pp' and generates the organization key pair $(\mathsf{osk}, \mathsf{opk})$. $\mathcal{S}$ runs $\mathcal{A}(\mathsf{pp}, \mathsf{opk})$ and simulates the oracles as in the real game.

If $\mathcal{A}$ outputs $(\mathbb{A}'^*, \mathsf{state})$, then $\mathcal{S}$ runs $\mathcal{A}(\mathsf{state})$ and interacts with $\mathcal{A}$ as verifier in a showing protocol. Now, if $\mathcal{A}$ delivers a valid showing using a credential cred'^*, then $\mathcal{S}$ rewinds $\mathcal{A}$ to the step after sending the commitments (K_Q, K_{C_2}) in PoK and restarts $\mathcal{A}$ with a new challenge $c' \neq c$. Then, by the knowledge extractor of PoK, $\mathcal{S}$ obtains ρ (such that $C_2 = \rho P$). $\mathcal{S}$ now computes $\mathsf{cred}^* \leftarrow \rho^{-1} \cdot \mathsf{cred}'^*$. If $\mathsf{cred}^* \notin \mathtt{CRED}$, then $\mathcal{S}$ and, in further consequence, $\mathcal{B}$ abort. Otherwise, we know that cred^* was the result of the j^*-th issue step during the simulation. Consequently, $\mathcal{S}$ knows the set of attributes $\mathbb{A}^* = \mathtt{ATTR}[j^*]$ corresponding to cred^*. If $\mathbb{A}'^* \sqsubseteq \mathbb{A}^*$, then $\mathcal{B}$ aborts. Otherwise, $\mathcal{S}$ can now compute the corresponding polynomial $\mathsf{enc}(\mathbb{A}^*)$. If $\mathsf{enc}(\mathbb{A}'^*) \mid \mathsf{enc}(\mathbb{A}^*)$, then $\mathcal{S}$ and, in further consequence, $\mathcal{B}$ abort. Otherwise, $\mathcal{B}$ outputs $(\rho, \mathsf{enc}(\mathbb{A}^*), \mathsf{enc}(\mathbb{A}'^*), \mathcal{C}_{\overline{\mathbb{A}'^*}})$. It is easy to verify that this is a valid output to win the factor soundness game of $\mathsf{PolyCommitFO}$.

Type 3A: Here, $\mathcal{B}$ plays the role of the challenger for $\mathcal{A}$. $\mathcal{B}$ obtains the instance (P, aP) to the DLP in G_1. Then, $\mathcal{B}$ runs the setup by generating public parameters pp and setting $Q \leftarrow aP$, generates the organization key pair $(\mathsf{osk}, \mathsf{opk})$, runs $\mathcal{A}(\mathsf{pp}, \mathsf{opk})$ and simulates the oracles as in the real game.

If $\mathcal{A}$ outputs $(\mathbb{A}'^*, \mathsf{state})$, then $\mathcal{B}$ runs $\mathcal{A}(\mathsf{state})$ and interacts with $\mathcal{A}$ as verifier in a showing protocol. Now, if $\mathcal{A}$ delivers a valid showing, then $\mathcal{B}$ rewinds $\mathcal{A}$ to the step after sending the commitments (K_Q, K_{C_2}) in PoK and restarts $\mathcal{A}$ with a new challenge $c' \neq c$. Then, by the knowledge extractor of PoK (for the Q-part of the proof), $\mathcal{B}$ obtains a value $a' \in \mathbb{Z}_p^*$. If $a'P \neq aP$, then $\mathcal{B}$ aborts. Otherwise $\mathcal{B}$ outputs (a', aP) as a solution to the DLP in G_1.

Type 3B: Here, $\mathcal{B}$ plays the role of the challenger for $\mathcal{A}$. $\mathcal{B}$ obtains the instance (P, aP) to the DLP in G_1. Then, $\mathcal{B}$ runs the setup by generating public parameters pp where it chooses $q \xleftarrow{R} \mathbb{Z}_p^*$ and sets $Q \leftarrow qP$. Next, $\mathcal{B}$ generates the organization key pair $(\mathsf{osk}, \mathsf{opk})$, runs $\mathcal{A}(\mathsf{pp}, \mathsf{opk})$ and initializes the environment. Additionally, $\mathcal{B}$ creates two secret lists $\mathtt{CRED}'$ and $\mathtt{CRED}''$. Furthermore, $\mathcal{B}$ simulates the oracles as in the real game, except for the oracles $\mathcal{O}^{U_\mathsf{I} O_\mathsf{O}}$, $\mathcal{O}^{U_\mathsf{V}}$ and $\mathcal{O}^{O_\mathsf{O}}$, which are simulated as follows:

$\mathcal{O}^{U_\mathsf{I} O_\mathsf{O}}(\mathsf{osk}, \mathsf{opk}, i, \mathbb{A})$**:** $\mathcal{B}$ runs this oracle as in the real game and produces a credential $\mathsf{cred}_i = ((C_1, P), \sigma)$. Additionally, $\mathcal{B}$ computes $((C_1, aP), \sigma')$ with $\sigma' \leftarrow \mathsf{Sign}_{\mathcal{R}}((C_1, aP), \mathsf{osk})$ and appends it to $\mathtt{CRED}'$.

$\mathcal{O}^{U_\mathsf{V}}(\mathsf{opk}, j, \mathbb{A}')$**:** $\mathcal{B}$ runs this oracle as in the real game, with the difference that $\mathcal{B}$ performs the showing with respect to the credentials stored in the list $\mathtt{CRED}'$. It computes the shown credential as $\mathsf{cred}' \leftarrow \mathsf{ChgRep}_{\mathcal{R}}(\mathtt{CRED}'[j], \rho, \mathsf{opk})$ (for some $\rho \xleftarrow{R} \mathbb{Z}_p^*$) and appends the tuple (cred', ρ) to $\mathtt{CRED}''$. Note that under the class hiding property of Scheme 1, $\mathcal{A}$ will not detect that the showings are performed with respect to a different obtained credential. Finally, $\mathcal{B}$ also performs PoK with respect to the known discrete logarithm q of Q and simulates the remainder. This produces a valid showing, since by the witness indistinguishability of the OR proof PoK, $\mathcal{A}$ cannot distinguish whether the honest part of the proof was conducted for C_2 or $Q = qP$.

$\mathcal{O}^{O_\mathsf{O}}(\mathsf{osk}, \mathsf{opk}, i, \mathsf{usk}_i, \mathbb{A})$**:** $\mathcal{B}$ runs this oracle as in the real game, with the only difference that $\mathcal{B}$ appends $\perp$ to $\mathtt{CRED}'$ (in order to preserve the same order as in $\mathtt{CRED}$).

If $\mathcal{A}$ outputs $(\mathbb{A}'^*, \mathsf{state})$, then $\mathcal{B}$ runs $\mathcal{A}(\mathsf{state})$ and interacts with $\mathcal{A}$ as verifier in a showing protocol. Now, if $\mathcal{A}$ delivers a valid showing using a credential cred'^*, then $\mathcal{B}$ rewinds $\mathcal{A}$ to the step after sending the commitments (K_Q, K_{C_2}) in PoK and restarts $\mathcal{A}$ with a new challenge $c' \neq c$. Then, by the knowledge extractor of PoK (for the C_2-part of the proof), $\mathcal{B}$ obtains $\rho' \in \mathbb{Z}_p^*$. If $\mathsf{cred}'^* \notin \mathtt{SCRED}$, then $\mathcal{B}$ aborts. Otherwise, let j^* be the index such that $\mathsf{cred}'^* \in \mathtt{SCRED}[j^*]$. Then, $\mathcal{B}$ retrieves the pair (cred', ρ) from $\mathtt{CRED}''$ such that $\mathsf{cred}'^* = \mathsf{cred}'$ and computes $a' \leftarrow \rho' \rho^{-1} \in \mathbb{Z}_p^*$. If $a'P \neq aP$ or $i^*_{j^*} \notin \mathtt{HU} \setminus \mathtt{KU}$, then $\mathcal{B}$ aborts. Otherwise $\mathcal{B}$ outputs (a', aP) as a solution to the DLP in G_1.

We emphasize that we do not consider re-randomized replays, i.e., replays, whose credentials and polynomial commitment witnesses have been re-randomized by the adversary, as this cannot be considered as an attack being easier than simple replays of showings.

In front of an adversary $\mathcal{A}$, we randomly pick an adversary of Type 1, 2A, 2B, 3A or 3B with equal probability. Thus, the global security loss is 1/5. $\square$

C.3 Proof of Theorem 6 (Anonymity)

Proof. For the sake of contradiction, we assume that there is an efficient adversary $\mathcal{A}$ winning the anonymity game with non-negligible probability and we assume that the used SPS-EQ-$\mathcal{R}$ scheme is class hiding*. We will show that this implies an efficient adversary against the class hiding* property of the SPS-EQ-$\mathcal{R}$ scheme for message length $\ell = 2$ giving the desired contradiction.

We construct an adversary $\mathcal{B}$, consisting of adversary $\mathcal{A}$ playing the anonymity game with a challenger $\mathcal{S}$. $\mathcal{B}$ is interacting with the challenger $\mathcal{C}$ in the class hiding game. Subsequently, we describe how $\mathcal{S}$ simulates the environment for $\mathcal{A}$ and interacts with the challenger of the class hiding* game.

Initially, $\mathcal{C}$ runs $\mathsf{BG} \leftarrow \mathsf{BGGen}_{\mathcal{R}}(\kappa)$, chooses $b \xleftarrow{R} \{0,1\}$ and runs $\mathcal{B}(\mathsf{BG}, \ell)$, who internally generates the organization key pair $(\mathsf{osk}, \mathsf{opk})$, and outputs $(\mathsf{state}, \mathsf{sk}, \mathsf{pk})$ with $(\mathsf{sk}, \mathsf{pk}) \leftarrow (\mathsf{osk}, \mathsf{opk})$. Here, the public parameters pp were generated by $\mathcal{B}$ (respectively $\mathcal{S}$) based on BG and by choosing $q \xleftarrow{R} \mathbb{Z}_p^*$ to compute $Q \leftarrow qP$ (instead of picking Q at random). Note that state includes $(\mathsf{osk}, \mathsf{opk})$, pp as well as the trapdoors α and q. Then, $\mathcal{C}$ runs $\mathcal{B}(\mathsf{state}, \mathsf{sk}, \mathsf{pk})$ and $\mathcal{S}$ initializes the environment (i.e., the lists and sets) and runs $\mathcal{A}(\mathsf{pp}, \mathsf{osk}, \mathsf{opk})$. $\mathcal{S}$ simulates $\mathcal{A}$'s oracle calls as follows:

$\mathcal{O}^{\mathtt{HU}+}(i)$: $\mathcal{S}$ runs the oracle as in the real game.

$\mathcal{O}^{U_\mathsf{I}}(\mathsf{osk}, \mathsf{opk}, i, \mathbb{A})$: $\mathcal{S}$ runs the oracle as in the real game, but discards the so obtained credential. Additionally, on the j-th call, $\mathcal{S}$ calls the random message oracle $\mathcal{O}^{RM}(\ell)$ from $\mathcal{C}$, gets in response a message vector (M_1, M_2) and sets $\mathsf{cred}_i \leftarrow ((M_1, M_2), \sigma)$ where it computes $\sigma \leftarrow \mathsf{Sign}_{\mathcal{R}}((M_1, M_2), \mathsf{osk})$. Finally, it appends $(\mathsf{cred}_i, \mathbb{A})$ to $(\mathtt{CRED}, \mathtt{ATTR})$.

$\mathcal{O}^{U_\mathsf{V}}(\mathsf{opk}, j, \mathbb{A}')$: $\mathcal{S}$ runs the oracle as in the real game and follows the Show algorithm, except for the computation of the value $\mathcal{C}_{\overline{\mathbb{A}'}}$ and the PoK. Let $((C_1, C_2), \sigma) \leftarrow \mathtt{CRED}[j]$. Then, in the case of $\mathcal{C}_{\overline{\mathbb{A}'}}$, $\mathcal{S}$ knows the trapdoor α and computes $\mathcal{C}_{\overline{\mathbb{A}'}} \leftarrow \frac{1}{\mathsf{enc}(\mathbb{A}')(\alpha)} C_1$, whereas in the latter case $\mathcal{S}$ simulates the proof part for C_2 and conducts an honest proof for the knowledge of q. This produces a valid showing, since by the witness indistinguishability of the OR proof PoK, $\mathcal{A}$ cannot distinguish whether the honest part of the proof was conducted for C_2 or $Q = qP$.

$\mathcal{O}^{RoR}(\mathsf{osk}, \mathsf{opk}, b, j_0, \mathbb{A}')$: $\mathcal{S}$ runs the oracle as in the real game, but in order to obtain a credential, it obtains $((C_1, C_2), \sigma) \leftarrow \mathtt{CRED}[j_0]$ and calls the oracle $\mathcal{O}^{RoR}(\mathsf{osk}, \mathsf{opk}, b, (C_1, C_2))$ of $\mathcal{C}$. Then, it simulates a showing using this response and the trapdoor information (similar to the oracle $\mathcal{O}^{U_\mathsf{V}}$).

Note that under the class hiding* property of the signature scheme, $\mathcal{A}$ will not recognize that the credentials presented in an interaction with the oracle $\mathcal{O}^{U_\mathsf{V}}$ consist of random messages (obtained from the oracle $\mathcal{O}^{RM}$) and not randomized versions of the discarded credentials obtained in the interaction with $\mathcal{O}^{U_\mathsf{I}}$. Moreover, by class hiding*, $\mathcal{A}$ is not able to distinguish a re-randomization of a message-signature pair from a random one. Also note that $\mathcal{C}_{\overline{\mathbb{A}'}}$ does not leak any information on attributes (except for the shown attributes), as it is derived from the random, fake commitment value C_1 in a way that only the verification relations work out. Since this value has the same distribution as the corresponding value in the real game, this part of the simulation is indistinguishable from the respective part of the real game.

If $\mathcal{A}$ outputs b^* to $\mathcal{S}$, then $\mathcal{B}$ outputs b^* to $\mathcal{C}$. It is now obvious that if $\mathcal{A}$ wins the anonymity game played with $\mathcal{S}$ with non-negligible probability, then also $\mathcal{B}$ wins the class hiding game played with $\mathcal{C}$ with the same probability, which contradicts the assumed hardness of the class hiding property. $\square$