A New Approach To Efficient Revocable Attribute-Based Anonymous Credentials*

David Derler, Christian Hanser, and Daniel Slamanig

IAIK, Graz University of Technology, Austria
{david.derler|christian.hanser|daniel.slamanig}@tugraz.at

Abstract. Recently, a new paradigm to construct very efficient multi-show attribute-based anonymous credential (ABC) systems has been introduced in Asiacrypt'14. Here, structure-preserving signatures on equivalence classes (SPS-EQ-$\mathcal{R}$), a novel flavor of structure-preserving signatures (SPS), and randomizable polynomial commitments are elegantly combined to yield the first ABC systems with $O(1)$ credential size and $O(1)$ communication bandwidth during issuing and showing. It has, however, been left open to present a full-fledged revocable multi-show attribute-based anonymous credential (RABC) system based on the aforementioned paradigm. As revocation is a highly desired and important feature when deploying ABC systems in a practical setting, this is an interesting challenge.

To this end, we propose an RABC system which builds upon the aforementioned ABC system, preserves its nice asymptotic properties and is in particular entirely practical. Our approach is based on universal accumulators, which nicely fit to the underlying paradigm. Thereby, in contrast to existing accumulator-based revocation approaches, we do not require complex zero-knowledge proofs of knowledge (ZKPKs) to demonstrate the possession of a non-membership witness for the accumulator. This is in part due to the nice rerandomization properties of SPS-EQ-$\mathcal{R}$. Thus, this makes the entire RABC system conceptually simple, efficient and represents a novel direction in credential revocation. We also propose a game-based security model for RABC systems and prove the security of our construction in this model. Finally, to demonstrate the value of our novel approach, we carefully adapt an efficient existing universal accumulator approach (as applied within Microsoft's U-Prove) to our setting and compare the two revocation approaches when used with the same underlying ABC system.

1 Introduction

Credential systems have been envisioned by Chaum [Cha85], with the motivation to develop a concept that allows users to interact anonymously with multiple

* The authors have been supported by EU Horizon 2020 through project Prismacloud (GA No. 644962) and by EU FP7 through project MATTHEW (GA No. 610436). This is the full version of a paper to appear at IMA Conference on Cryptography and Coding 2015.

organizations online. Thereby, a user can obtain a credential for a pseudonym (nym) from one organization (issuer) and demonstrate possession of the credential to other organizations (verifiers), without revealing his nym. Later on, this idea has been formalized as pseudonym systems in [LRSW00] and has, subsequently, been further extended and formalized as anonymous credential (AC) systems in [CL01]. As privacy in digital interactions has become more and more important over the last decades, various AC systems with different properties and targeting different environments have been proposed [AMO08, BL13, BCC+09, Bra00, CL01, CL02a, CL04, CLNR14, CMZ14, GGM14, HM12, Ver01, CDHK]. Today, the most prevalent approaches are IBM's idemix [CH02] and Microsoft's U-Prove [PZ13]. The former is based on CL signatures [CL02a] supporting an unlimited number of unlinkable showings of a credential (multi-show), where the latter is based on Brands' blind signatures [Bra00] and all showings are linkable (one-show).

While early ACs, such as [CL01], did not put focus on how credentials should look like, nowadays credentials in ACs are typically viewed as being a collection of users' attributes, e.g., birth date, nationality, sex. In such a setting, users obtain credentials on attributes (issued by some organization). Then, users can prove possession of these credentials anonymously (and in an unlinkable fashion) to any verifier. Thereby, they reveal only (the possession of) some attributes and nothing beyond. Such AC systems are also known as privacy-ABC systems (or simply ABC systems).

Revocation of ABCs. Efficient revocation of credentials is especially important and challenging in practical applications of multi-show ABCs. Unfortunately, this is no trivial task at all. It is clearly not possible to simply blacklist credentials as it can be conveniently done in PKIs. To realize revocable ABCs (RABCs), various different credential revocation mechanisms have been introduced over the years (cf. [LKDN11] for an exhaustive discussion). The idea is that a revocation authority (which may be run by the credential issuer) publishes revocation information which allows verifiers to decide whether a credential has been revoked. Ideally, such revocation mechanisms are conceptually simple, scale well and do not add significant additional burden to users and verifiers. However, simple mechanisms are either inflexible or far from practical. Examples are the inclusion of the validity period as attribute into credentials or the re-issuing of all unrevoked credentials triggered by the replacement of the issuer's key material. Obviously, such mechanisms either get insecure due to too long validity periods (and, thus, too long revocation intervals) or require to frequently re-issue a large amount of credentials. More importantly, they do not allow to selectively revoke single credentials in case of loss, theft or fraud.

More sophisticated revocation mechanisms supporting the selective revocation of single credentials are either based on whitelists or blacklists. Whitelist approaches require users to prove that unrevoked credentials are contained in a list. The effort for users (during showings) is typically linear in the number of valid credentials and/or it requires users to download revocation information each time a new credential gets issued. Thus, whitelist approaches do not

scale well and cannot be considered practical in general. In contrast, blacklist revocation usually scales far better. The main reason for this is that revocation list updates are only required on revocation (which usually can be considered a rare event in comparison to the issuing of new credentials). Thereby, blacklisting approaches based on verifier-local revocation (VLR) [BS04] do not require any updates from the users, but require an effort for the verifier that is linear in the number of revoked credentials. Many of the VLR techniques also have the problem of missing the property of backward unlinkability [Son01], i.e., the revocation of a credential implies the linkability of all past showings (e.g., as it is the case in [HM12, LAHV15]). Furthermore, techniques to add backward unlinkability to VLR either induce a significant additional computational burden on users and verifiers [Son01] or require frequent updates and computational overhead for verifiers [NF05]. Another blacklist approach [NFHF09] represents blacklists as signatures on ordered credential identifier pairs. This is elegant, since the computational costs for users and verifiers are constant and quite small. Yet, the user and the verifier have to update a significant amount of revocation information on each revocation, as the blacklist has to be recomputed entirely (number of signatures linear in the number of revoked credentials). The remaining and popular choice is to use blacklists based on universal accumulators [ACN13, ATSM09, CKS09, CL02b, Ngu05]. This approach scales well and requires only constant computational effort for users and verifiers. Although updates of the accumulator and the non-membership witnesses are required on revocation, these are small and often constant in size.

Design paradigms of existing (R)ABCs. ABC systems are typically constructed in the following way (with few exceptions [CL11, CL13]). A user obtains a signature on (commitments to) attributes using a suitable signature scheme. Then, on a showing, the user randomizes the signature (such that the resulting signature is unlinkable to the issued one) and proves in zero-knowledge the possession of a signature. Thereby, attributes may be selectively revealed and/or relations among attributes may be proven. In one-show ABCs, blind signature schemes are used, and—instead of randomizing the signatures—the same unblinded signature is presented on each showing. A standard way to turn ABCs into RABCs is to add a credential identifier (revocation handle) as an additional never-revealed attribute. Then, for the aforementioned approaches which use explicit ZKPKs, the choice of the revocation mechanisms is somewhat arbitrary. It only has to be guaranteed that the identifier in the credential and the one used for blacklisting (or whitelisting) are identical. Hence, the showing in such an RABC system amounts to providing the ZKPK for the underlying ABC and the ZKPK of the used revocation mechanism plus an additional ZKPK that the identifier in the credential coincides with the identifier used for revocation.

Design paradigm of the ABC from [HS]. The ABC system proposed in [HS] is conceptually significantly different from the aforementioned approach. Its main building block are structure-preserving signatures on equivalence classes (SPS-EQ-$\mathcal{R}$). An SPS-EQ-$\mathcal{R}$ signs equivalence classes defined on group element vectors and allows to consistently randomize messages and signatures in the public by

changing representatives of the signed class. It is used to sign rerandomizable, constant-size commitments to polynomials. Thereby, the rerandomization of the commitment is compatible with the rerandomization of the SPS-EQ-$\mathcal{R}$. To perform a showing for a subset of the attributes, the (rerandomized) commitment is partially opened and the rerandomization property of SPS-EQ-$\mathcal{R}$ provides unlinkability, while authenticity is still ensured. Additionally, the approach requires a single, constant-size ZKPK to prevent replays of already conducted interactive showings. Consequently, the so obtained ABC system does not need costly ZKPKs to prove possession of the attributes. In particular, [HS] provides the first ABC system with $O(1)$ credential size and $O(1)$ communication bandwidth during both issuing and showing and is thus very efficient. The communication costs of other existing approaches are at least linear in the number of shown/encoded attributes in the ABC system (or constant-size showings can only be achieved for special cases [SNF11, BNF12], e.g., very small attribute domains, at the cost of huge public parameters—linear in the number of all potential values over all attribute domains).

Contribution. The efficiency of the ABC system from [HS], e.g., when instantiated with the EUF-CMA secure SPS-EQ-$\mathcal{R}$ scheme from [FHS14], makes it very attractive for practical use. Thus, obtaining an RABC system following the same paradigm is an important step towards highly efficient and practical RABCs. We construct an RABC system based on the ABC system in [HS] (which can e.g. be instantiated with the SPS-EQ-$\mathcal{R}$ from [FHS14]), and, thereby, rely on a universal accumulator-based blacklist approach. In contrast to all previous applications of universal accumulators to blacklist revocation [LLX07, ATSM09, ACN13, NP14], we do, however, not require explicit ZKPKs of non-membership witnesses satisfying the accumulator verification equation. We achieve this by rerandomizing the used universal accumulator, which is a novel way of proving possession of a particular non-membership witness.

In order to evaluate our approach, we, in addition, carefully adapt an existing universal accumulator revocation mechanism [ACN13, NP14] (applied within Microsoft's U-Prove) to the ABC system from [HS]. Contrary to our first construction, this revocation mechanism represents a traditional ZKPK approach for demonstrating knowledge of a non-membership witness that satisfies the accumulator verification equation. Thereby, it turns out that regarding the most time critical part, i.e., the showing protocol performed by a (potentially resource constrained [UW14]) user, our approach outperforms the revocation approach adpoted from U-Prove.

As our revocation mechanisms preserve the asymptotic optimality of the ABC system in [HS], our RABC constructions are also the first RABC system with $O(1)$ credential size and $O(1)$ communication costs during issuing as well as showing.

Revocation in ABC systems is typically considered as an add-on and, thus, not considered in the security models of ABCs. To overcome this issue, another contribution of this paper is a comprehensive game-based security model for RABC systems, which explicitly considers backward-unlinkability. We prove our

proposed approach secure in this model. Independently to our work, another formal model for ABC systems has been introduced in [CKL+15]. It also considers revocation but also additional features such as auditing [CLNR14]. However, the model in [CKL+15] aims at constructing ABCs by means of a generic composition of numerous building blocks (commitment schemes, NIZKPs, privacy-enhancing attribute-based signatures, revocation schemes and pseudonym schemes), considers only non-interactive protocols (using the notion of tokens) and uses stronger simulation-based security definitions. In particular the stronger security notions add a non-trivial overhead in terms of efficiency to the constructions, which, in turn, makes it less attractive for highly efficient and practical ABC systems.[1]

2 Preliminaries

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

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

We use lower-case boldface letters for elements in $\mathbb{G}_T$, e.g., $\mathbf{g} = e(P, \hat{P})$.

Definition 2 (Bilinear Group Generator). Let BGGen be an algorithm which takes a security parameter κ and generates a bilinear group $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$ in the Type-3 bilinear group setting, where the common group order p of the groups $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ is a prime of bitlength κ, e is a pairing and P and $\hat{P}$ are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively.

Definition 3 (Discrete Logarithm Assumption). The DL assumption in $\mathbb{G}_i$ states that for all probabilistic polynomial-time (PPT) adversaries $\mathcal{A}$ there is a negligible function $\epsilon(\cdot)$ such that

$$\Pr\left[\mathsf{BG} \leftarrow \mathsf{BGGen}(1^\kappa), r \xleftarrow{R} \mathbb{Z}_p, r^* \leftarrow \mathcal{A}\big(\mathsf{BG}, rP_i\big) \;:\; r^* = r\right] - \frac{1}{2} \leq \epsilon(\kappa),$$

where $P_1 = P$ and $P_2 = \hat{P}$ and $i \in \{1, 2\}$.

Definition 4 (Decisional Diffie-Hellman Assumption). The DDH assumption in $\mathbb{G}_i$ states that for all probabilistic polynomial-time (PPT) adversaries $\mathcal{A}$ there is a negligible function $\epsilon(\cdot)$ such that

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

where $P_1 = P$ and $P_2 = \hat{P}$ and $i \in \{1, 2\}$.

[1] We, however, note that the efficiency of our scheme comes at the cost of more complex proofs.

Definition 5 (Symmetric External Diffie Hellman Assumption). Let BG be a bilinear group. The SXDH assumption states that the DDH assumption holds in $\mathbb{G}_1$ and $\mathbb{G}_2$.

The following assumption [HS] is the Type-3 bilinear group counterpart of the strong Diffie-Hellman assumption.

Definition 6 (t-co-Strong Diffie Hellman Assumption). The t-co-SDH$_i^*$ assumption states that for all probabilistic polynomial-time (PPT) adversaries $\mathcal{A}$ there is a negligible function $\epsilon(\cdot)$ such that

$$\Pr\left[\begin{array}{ll} \alpha \xleftarrow{R} \mathbb{Z}_p, \ \ \mathsf{BG} \leftarrow \mathsf{BGGen}(1^\kappa), & c \in \mathbb{Z}_p \setminus \{-\alpha\} \\ (c, T_i) \xleftarrow{R} \mathcal{A}(\mathsf{BG}, (\alpha^j P_1)_{j=0}^t, (\alpha^j P_2)_{j=0}^t) : & \wedge \ \ T_i = \frac{1}{\alpha+c} P_i \end{array}\right] \leq \epsilon(\kappa),$$

where $P_1 = P$ and $P_2 = \hat{P}$ and $i \in \{1, 2\}$.

We will use the t-co-SDH$_1^*$ assumption statically, as we will fix t a priori as a system parameter and assume that it is bounded by $\mathsf{poly}(\kappa)$. Then, the security loss which applies when using t-co-SDH$_1^*$ in a non-static way [Che06] does not apply.

2.1 Universal Accumulators

Cryptographic accumulators [BdM93] represent a finite set $\mathcal{X}$ as a single succinct value $\Pi_{\mathcal{X}}$ and for each $x \in \mathcal{X}$ one can compute a witness ω_x, certifying membership of x in $\mathcal{X}$. Universal accumulators additionally support non-membership witnesses ω_y that certify non-membership of a value $y \notin \mathcal{X}$. Henceforth, we write Π if we do not want to make $\mathcal{X} = \{x_1, \ldots, x_n\}$ explicit. To blacklist credentials, we require a universal accumulator. Subsequently, we restate the accumulator of Au et al. [ATSM09] for the Type-3 bilinear group setting and in the model of [DHS15], where we omit the algorithms that are not required in our context, i.e., the dynamic features. The formal model is given in Appendix A.2.
For the Type-3 bilinear setting, in analogy to [ATSM09], we can straightforwardly prove the following (where we omit the proof):

Theorem 1. *Scheme 1 is collision-free under the t-co-SDH$_i^*$ assumption, where t is the maximum number of values to be accumulated.*

2.2 Structure-Preserving Signatures on Equivalence Classes

The notion of structure-preserving signature schemes on equivalence classes (SPS-EQ-$\mathcal{R}$) has been introduced in [HS]. The authors consider elements of a vector $(M_i)_{i\in[\ell]} \in (\mathbb{G}_1^*)^\ell$ (where $\mathbb{G}_1^* = \mathbb{G}_1 \setminus \{0_{\mathbb{G}_1}\}$, for some prime order group $\mathbb{G}_1$) which share different mutual ratios. These ratios depend on their discrete logarithms and are invariant under the operation $\gamma : \mathbb{Z}_p^* \times (\mathbb{G}_1^*)^\ell \to (\mathbb{G}_1^*)^\ell$ with $(s, (M_i)_{i\in[\ell]}) \mapsto s(M_i)_{i\in[\ell]}$. Thus, one can use this invariance to partition $(\mathbb{G}_1^*)^\ell$ into equivalence classes using the relation $\mathcal{R} = \{(M, N) \in (\mathbb{G}_1^*)^\ell \times (\mathbb{G}_1^*)^\ell : \exists s \in$

$\mathsf{Gen}_{\mathsf{Acc}}(\mathsf{BG}, t)$: Given a bilinear group BG and an upper bound t for the number of elements to be accumulated, pick $\lambda \xleftarrow{R} \mathbb{Z}_p^*$, compute $\mathsf{pk}_{\Pi} \leftarrow ((\lambda^i P)_{i\in[t]}, (\lambda^i \hat{P})_{i\in[t]})$ and return $(\emptyset, \mathsf{pk}_{\Pi})$.

$\mathsf{Eval}_{\mathsf{Acc}}(\mathcal{X}, (\emptyset, \mathsf{pk}_{\Pi}))$: Given a set $\mathcal{X} = \{x_1, \ldots, x_n\}$ and an accumulator public key pk_{Π}, compute $\pi(X) \leftarrow \prod_{i\in[n]}(X - x_i) = \sum_{i=0}^{n} a_i \cdot X^i$ and $\Pi_{\mathcal{X}} \leftarrow \sum_{i=0}^{n} a_i(\lambda^i P)$ and return $\Pi_{\mathcal{X}}$ together with $\mathsf{aux} \leftarrow \mathcal{X}$.

$\mathsf{WitCreate}_{\mathsf{Acc}}(\Pi_{\mathcal{X}}, \mathsf{aux}, y, (\emptyset, \mathsf{pk}_{\Pi}))$: Given an accumulator $\Pi_{\mathcal{X}}$, some auxiliary information $\mathsf{aux} = \mathcal{X} = \{x_1, \ldots, x_n\}$, a non-member y and an accumulator public key pk_{Π}, this algorithm checks whether $y \in \mathcal{X}$ and if so returns $\perp$. Otherwise, it computes $\pi(X) \leftarrow \prod_{i\in[n]}(X - x_i)$ and $d \in \mathbb{Z}_p^*$ such that $\pi(X) = g(X)(X - y) + d$ holds. With $g(X) = \sum_{i=0}^{n-1} a_i \cdot X^i$ it computes $\hat{W} \leftarrow \sum_{i=0}^{n-1} a_i(\lambda^i \hat{P})$ and returns $\omega_y \leftarrow (\hat{W}, d)$.

$\mathsf{Verify}_{\mathsf{Acc}}(\Pi, \omega_y, y, \mathsf{pk}_{\Pi})$: Given an accumulator Π, a non-membership witness ω_y and some corresponding y, this algorithm parses ω_y as $(\hat{W}, d)$, checks if $d \neq 0$ and $e(\Pi, \hat{P}) = e(\lambda P - yP, \hat{W}) \cdot e(dP, \hat{P})$ holds and if so returns 1 and 0 else.

Scheme 1: Universal Accumulator from [ATSM09] tailored to Non-Membership Witnesses.

$\mathbb{Z}_p^*$ such that $N = s \cdot M\} \subseteq (\mathbb{G}_1^*)^{2\ell}$. When signing an equivalence class $[M]_{\mathcal{R}}$ with such a scheme, one actually signs a representative $(M_i)_{i\in[\ell]}$ of class $[M]_{\mathcal{R}}$. The scheme, then, allows to switch to different representatives of the same class and to update corresponding signatures in the public, i.e., without any secret key. The initial instantiation proposed in [HS] turned out to only be secure against random-message attacks (cf. [Fuc14] and the updated full version of [HS]), but together with Fuchsbauer [FHS14] they subsequently presented a scheme that is secure against chosen-message attack (EUF-CMA) in the generic group model.

For our RABC, we need a Type-3 bilinear group setting based, EUF-CMA-secure SPS-EQ-$\mathcal{R}$ that perfectly adapts signatures (cf. Appendix A.4 for the definitions). Scheme 3, presented in Appendix A.4, restates the SPS-EQ-$\mathcal{R}$ construction from [FHS14], which satisfies all our requirements.

3 An Efficient RABC System

In an RABC system there are different organizations issuing credentials for different users under different pseudonyms.[2] Furthermore, there are revocation authorities which can selectively revoke credentials. Such a system requires that issuings and showings of the same user are unlinkable and is called multi-show RABC system when multiple showings carried out by the same user cannot be linked and one-show RABC system otherwise. A credential cred for user i under pseudonym nym 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

[2] We stress that in our context pseudonyms are solely used for revocation and not for showing purposes (as e.g., in the model of [CKL+15]) and thus one might call ours revocation pseudonyms (but we simply call them pseudonyms henceforth).

mean the size of $\mathbb{A}$, which is defined to be the sum of cardinalities of all second components $\mathtt{attrV}_k$ of all tuples in $\mathbb{A}$. Moreover, we denote by $\mathbb{A}' \sqsubseteq \mathbb{A}$ a subset of the credential attributes. In particular, for every $k \in [n]$, 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). Below, we present our formal RABC model which is based on the ABC model in [HS].

Definition 7 (RABC System). A *revocable attribute-based anonymous credential (RABC) system* consists of the following polynomial time algorithms:

Setup: A probabilistic algorithm that takes a security parameter κ and some optional auxiliary information aux (which may fix an universe of attributes and attribute values and other parameters).

RAKeyGen: A probabilistic algorithm that takes input the public parameters pp and outputs a key pair (rsk, rpk) for the revocation authority.

OrgKeyGen: A probabilistic algorithm that takes input the public parameters pp and outputs an organization key pair (osk, opk).

UserKeyGen: A probabilistic algorithm that takes input the public parameters pp and outputs a user key pair (usk, upk).

(Obtain, Issue): These (probabilistic) algorithms are run by user i and organization j, who interact during execution. Obtain takes input the public parameters pp, the user's secret key usk_i, an organization's public key opk_j, a pseudonym nym and an attribute set $\mathbb{A}$. Issue takes input the public parameters pp, the public key of the revocation authority rpk, the user's public key upk_i, an organization's secret key osk_j, a pseudonym nym and an attribute set $\mathbb{A}$. At the end, Obtain outputs a credential $\mathsf{cred}_{\mathsf{nym}}$ for $\mathbb{A}$ for user i with respect to nym.

(Show, Verify): These (probabilistic) algorithms are run by user i and a verifier, who interact during execution. Show takes input public parameters pp, the public revocation key rpk, the user's secret key usk_i, the organization's public key opk_j, a credential $\mathsf{cred}_{\mathsf{nym}}$ for the attribute set $\mathbb{A}$, a second set $\mathbb{A}' \sqsubseteq \mathbb{A}$ and some information $\mathbb{R}_S^{\mathsf{nym}}$ to prove that $\mathsf{cred}_{\mathsf{nym}}$ has not been revoked. Verify takes input pp, rpk, opk_j, a set $\mathbb{A}'$ and some revocation information $\mathbb{R}_V$. At the end, Verify outputs 1 or 0 indicating whether the credential showing was accepted or not.

Revoke: This (probabilistic) algorithm takes input the public parameters pp, the revocation key pair (rsk, rpk) and two disjoint lists NYM and RNYM holding valid and revoked pseudonyms, respectively. It outputs the revocation information $\mathbb{R} = (\mathbb{R}_V, \mathbb{R}_S)$. $\mathbb{R}_V$ is needed for verifying the revocation status and $\mathbb{R}_S$ is a list holding the revocation information per nym.

3.1 Security Model for RABCs

The subsequent security model is adapted from [HS]. We note that we consider only a single organization (identified by $j = 1$) in our model (since all organizations have independent signing keys, the extension is straightforward). Basically,

an RABC system needs to be *correct*, *unforgeable* and *anonymous*. To provide formal definitions of these properties we introduce several global variables and oracles. 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. Furthermore, we introduce the sets $\mathtt{N}$ and $\mathtt{RN}$ for keeping track of all pseudonyms and all revoked pseudonyms, respectively. We use the variables $\mathtt{RI}$ and $\mathtt{NYM}_{LoR}$ (initially set to $\perp$) to store the globally maintained revocation information $\mathbb{R}$ and the pseudonyms used in the $\mathcal{O}^{LoR}$ oracle. All these sets as well as $\mathtt{RI}$ and $\mathtt{NYM}_{RoR}$ are maintained by the environment and are available to the adversary for read access. We use the lists $\mathtt{UPK}$, $\mathtt{USK}$, $\mathtt{CRED}$ and $\mathtt{ATTR}$ to track issued user keys, credentials and corresponding attributes (per pseudonym). These lists are only accessible to the environment. We introduce the subsequent oracles and assume the public parameters pp to be 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 $(\mathtt{USK}[i], \mathtt{UPK}[i]) \leftarrow \mathsf{UserKeyGen}(\mathsf{pp})$, 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 $i \notin \mathtt{U}$, $i \in \mathtt{CU}$, or $\mathtt{NYM}_{LoR} \cap \mathtt{N}[i] \neq \emptyset$ 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 $i \notin \mathtt{U}$, $i \in \mathtt{KU}$, or $\mathtt{NYM}_{LoR} \cap \mathtt{N}[i] \neq \emptyset$ return $\perp$. Otherwise, it reveals the credentials and the secret key of user i by returning $\mathtt{USK}[i]$ and the credentials $\mathtt{CRED}[\mathsf{nym}]$ for all $\mathsf{nym} \in \mathtt{N}[i]$. Finally, it adds i to $\mathtt{KU}$.

$\mathcal{O}^{\mathtt{RN}+}(\mathsf{rsk}, \mathsf{rpk}, \mathtt{REV})$**:** It takes input the revocation secret key rsk, the revocation public key rpk and a list $\mathtt{REV}$ of pseudonyms to be revoked. If $\mathtt{REV} \cap \mathtt{RN} \neq \emptyset$ or $\mathtt{REV} \not\subseteq \mathtt{N}$ return $\perp$. Otherwise, set $\mathtt{RN} \leftarrow \mathtt{RN} \cup \mathtt{REV}$ and $\mathtt{RI} \leftarrow \mathsf{Revoke}(\mathsf{pp}, \mathsf{rsk}, \mathsf{rpk}, \mathtt{N} \setminus \mathtt{RN}, \mathtt{RN})$.

$\mathcal{O}^{U_{\mathsf{I}}O_{\mathsf{O}}}(\mathsf{osk}, \mathsf{opk}, \mathsf{rsk}, \mathsf{rpk}, i, \mathsf{nym}, \mathbb{A})$**:** It takes input the organization key pair (osk, opk), the revocation key pair ($\mathsf{rsk}, \mathsf{rpk}$), a user i, a pseudonym nym and a set of attributes $\mathbb{A}$. If $i \notin \mathtt{HU}$ or $\mathsf{nym} \in \mathtt{N}$ return $\perp$. Otherwise, it issues a credential cred on $\mathbb{A}$ and nym for an honest user $i \in \mathtt{HU}$. Here, the oracle plays the role of the user as well as the organization. It runs

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

Finally, it sets $(\mathtt{CRED}[\mathsf{nym}], \mathtt{ATTR}[\mathsf{nym}]) \leftarrow (\mathsf{cred}, \mathbb{A})$, appends nym to $\mathtt{N}[i]$ and runs $\mathtt{RI} \leftarrow \mathsf{Revoke}(\mathsf{pp}, \mathsf{rsk}, \mathsf{rpk}, \mathtt{N} \setminus \mathtt{RN}, \mathtt{RN})$, but returns nothing to the caller.

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

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

where Obtain is run on behalf of honest user i and Issue is executed by the caller (the dishonest organization). Finally, it sets $(\mathtt{CRED}[\mathsf{nym}], \mathtt{ATTR}[\mathsf{nym}]) \leftarrow (\mathsf{cred}, \mathbb{A})$, appends nym to $\mathtt{N}[i]$ and runs $\mathtt{RI} \leftarrow \mathsf{Revoke}(\mathsf{pp}, \mathsf{rsk}, \mathsf{rpk}, \mathtt{N} \setminus \mathtt{RN}, \mathtt{RN})$.

$\mathcal{O}^{O_O}(\mathsf{osk}, \mathsf{opk}, \mathsf{rsk}, \mathsf{rpk}, i, \mathsf{nym}, \mathsf{usk}_i, \mathbb{A})$**:** It takes input the organization key pair $(\mathsf{osk}, \mathsf{opk})$, the revocation key pair $(\mathsf{rsk}, \mathsf{rpk})$, a user i, a pseudonym nym, a user secret key usk_i and a set of attributes $\mathbb{A}$. If $i \notin \mathtt{CU}$ or $\mathsf{nym} \in \mathtt{N}$ 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

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

where Obtain is executed by the caller and sets $(\mathtt{CRED}[\mathsf{nym}], \mathtt{ATTR}[\mathsf{nym}]) \leftarrow (\mathsf{cred}, \mathbb{A})$, appends nym to $\mathtt{N}[i]$ and runs $\mathtt{RI} \leftarrow \mathsf{Revoke}(\mathsf{pp}, \mathsf{rsk}, \mathsf{rpk}, \mathtt{N} \setminus \mathtt{RN}, \mathtt{RN})$.

$\mathcal{O}^{U_V}(\mathsf{opk}, \mathsf{rpk}, \mathsf{nym}, \mathbb{A}', \mathbb{R}_V)$**:** It takes input the organization public key opk, the public revocation key rpk, a user i, a pseudonym nym, a set of attributes $\mathbb{A}'$ certified to the user i_{nym} (that is the index such that $\mathsf{nym} \in \mathtt{N}[i_{\mathsf{nym}}]$) and the revocation information $\mathbb{R}_V$. If $\mathsf{nym} \notin \mathtt{N}$, $i_{\mathsf{nym}} \notin \mathtt{HU}$, $\mathbb{A}' \not\sqsubseteq \mathtt{ATTR}[\mathsf{nym}]$ or $\mathsf{nym} \in \mathtt{RN}$ return $\perp$. Otherwise, it plays the role of an honest user i_{nym} and runs

$$(\emptyset, b) \leftarrow \big(\mathsf{Show}(\mathsf{pp}, \mathsf{rpk}, \mathtt{USK}[i_{\mathsf{nym}}], \mathsf{opk}, \mathtt{CRED}[\mathsf{nym}], \mathtt{ATTR}[\mathsf{nym}], \mathbb{A}', \mathtt{RI}[2][\mathsf{nym}]), \mathsf{Verify}(\mathsf{pp}, \mathsf{rpk}, \mathsf{opk}, \mathbb{A}', \mathbb{R}_V)\big),$$

where Verify is executed by the caller (the dishonest verifier).

$\mathcal{O}^{LoR}(\mathsf{osk}, \mathsf{opk}, \mathsf{rsk}, \mathsf{rpk}, b, \mathsf{nym}_0, \mathsf{nym}_1, \mathbb{A}', \mathbb{R}_V)$**:** It takes input the organization and revocation key pairs $(\mathsf{osk}, \mathsf{opk})$ and $(\mathsf{rsk}, \mathsf{rpk})$, a bit b, two pseudonyms nym_0 and nym_1 and a set of attributes $\mathbb{A}'$. It returns $\perp$ if for $j \in \{0, 1\}$

$$\mathsf{nym}_j \notin \mathtt{N} \ \vee \ i_{\mathsf{nym}_j} \notin \mathtt{HU} \ \vee \ i_{\mathsf{nym}_j} \in \mathtt{KU} \ \vee \ \mathbb{A}' \not\sqsubseteq \mathtt{ATTR}[\mathsf{nym}_j] \ \vee \ \mathsf{nym}_j \in \mathtt{RN},$$

where i_{nym_j} is such that $\mathsf{nym}_j \in \mathtt{N}[i_{\mathsf{nym}_j}]$. Else, it adds nym_0 and nym_1 to $\mathtt{NYM}_{LoR}$ and interacts with the adversary during an execution of the (Show, Verify) protocol for the credential with the pseudonym nym_b and attributes $\mathbb{A}'$.

Now, we are ready to introduce an exact definition of a *secure* RABC system:

Definition 8 (Correctness). An RABC system is *correct*, if

$\forall \kappa > 0, \ \forall \mathsf{aux}, \ \forall \mathbb{A}, \ \forall \mathbb{A}' \sqsubseteq \mathbb{A},$
$\forall \mathtt{NYM}, \mathtt{RNYM} \subseteq \mathtt{N} : \mathtt{NYM} \cap \mathtt{RNYM} = \emptyset, \ \forall \mathsf{nym} \in \mathtt{NYM},$
$\forall \mathsf{pp} \leftarrow \mathsf{Setup}(1^\kappa, \mathsf{aux}), \ \forall (\mathsf{rsk}, \mathsf{rpk}) \leftarrow \mathsf{RAKeyGen}(\mathsf{pp}),$
$\forall (\mathsf{osk}, \mathsf{opk}) \leftarrow \mathsf{OrgKeyGen}(\mathsf{pp}), \ \forall (\mathsf{usk}, \mathsf{upk}) \leftarrow \mathsf{UserKeyGen}(\mathsf{pp}) :$
$(\mathsf{cred}, \emptyset) \leftarrow (\mathsf{Obtain}(\mathsf{pp}, \mathsf{usk}, \mathsf{opk}, \mathsf{nym}, \mathbb{A}), \mathsf{Issue}(\mathsf{pp}, \mathsf{upk}, \mathsf{osk}, \mathsf{nym}, \mathbb{A})),$
$(\mathbb{R}_S, \mathbb{R}_V) \leftarrow \mathsf{Revoke}(\mathsf{pp}, (\mathsf{rsk}, \mathsf{rpk}), \mathtt{NYM}, \mathtt{RNYM})$ it holds that
$(\emptyset, 1) \leftarrow (\mathsf{Show}(\mathsf{pp}, \mathsf{usk}, \mathsf{opk}, \mathsf{cred}, \mathbb{A}, \mathbb{A}', \mathbb{R}_S[\mathsf{nym}]), \mathsf{Verify}(\mathsf{pp}, \mathsf{opk}, \mathbb{A}', \mathbb{R}_V)).$

Definition 9 (Unforgeability). We call an RABC system *unforgeable*, if for all PPT adversaries $\mathcal{A}$ there is a negligible function $\epsilon(\cdot)$ such that

$$\Pr\left[\begin{array}{l}\mathsf{pp} \leftarrow \mathsf{Setup}(1^\kappa, \mathsf{aux}), (\mathsf{rsk}, \mathsf{rpk}) \leftarrow \mathsf{RAKeyGen}(\mathsf{pp}),\\ (\mathsf{osk}, \mathsf{opk}) \leftarrow \mathsf{OrgKeyGen}(\mathsf{pp}), \mathcal{O} \leftarrow \{\mathcal{O}^{\mathtt{HU}+}(\cdot), \mathcal{O}^{\mathtt{CU}+}(\cdot,\cdot),\\ \mathcal{O}^{\mathtt{KU}+}(\cdot), \mathcal{O}^{\mathtt{RN}+}(\mathsf{rsk}, \mathsf{rpk}, \cdot), \mathcal{O}^{U_\mathsf{I} O_\mathsf{O}}(\mathsf{osk}, \mathsf{opk}, \mathsf{rsk}, \mathsf{rpk}, \cdot, \cdot, \cdot),\\ \mathcal{O}^{U_\mathsf{V}}(\mathsf{opk}, \mathsf{rpk}, \cdot, \cdot, \mathtt{RI}[0]), \mathcal{O}^{O_\mathsf{O}}(\mathsf{osk}, \mathsf{opk}, \mathsf{rsk}, \mathsf{rpk}, \cdot, \cdot, \cdot, \cdot)\},\\ (\mathbb{A}'^*, \mathsf{state}) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{pp}, \mathsf{opk}, \mathsf{rpk}),\\ (\emptyset, b^*) \leftarrow (\mathcal{A}(\mathsf{state}), \mathsf{Verify}(\mathsf{pp}, \mathsf{opk}, \mathsf{rpk}, \mathbb{A}'^*, \mathtt{RI}[1])) :\\ \qquad b^* = 1 \ \wedge\ \big(\mathsf{nym}^* = \bot \ \ \vee\ \ \big(\mathsf{nym}^* \neq \bot \wedge\\ \qquad \big(\mathbb{A}'^* \not\sqsubseteq \mathtt{ATTR}[\mathsf{nym}^*] \ \vee\ (i^*_{\mathsf{nym}^*} \in \mathtt{HU} \setminus \mathtt{KU}) \ \vee\ \mathsf{nym}^* \in \mathtt{RN}\big)\big)\end{array}\right] \leq \epsilon(\kappa),$$

where the credential shown by $\mathcal{A}$ in the second phase corresponds to pseudonym nym^* and to user $i^*_{\mathsf{nym}^*}$ (that is the index such that $\mathsf{nym}^* \in \mathtt{N}[i^*_{\mathsf{nym}^*}]$). Thereby, $\bot$ indicates that no such index nym^* exists.

The winning conditions in the unforgeability game are chosen following the subsequent rationale. The first condition ($\mathsf{nym}^* = \bot$) captures showings of credentials, which have never been issued (existential forgeries). The second condition ($\mathsf{nym}^* \neq \bot \wedge \mathbb{A}'^* \not\sqsubseteq \mathtt{ATTR}[\mathsf{nym}^*]$) captures showings with respect to existing credentials, but invalid attribute sets. The third condition ($\mathsf{nym}^* \neq \bot \wedge i^*_{\mathsf{nym}^*} \in \mathtt{HU} \setminus \mathtt{KU}$) covers showings with respect to honest users, whose credentials and respective secrets the adversary does not know. This essentially boils down to replayed showings. Finally, the last condition ($\mathsf{nym}^* \neq \bot \wedge \mathsf{nym}^* \in \mathtt{RN}$) covers that showings cannot be performed with respect to revoked pseudonyms.

Definition 10 (Anonymity). We call an RABC system *anonymous*, if for all PPT adversaries $\mathcal{A}$ there is a negligible function $\epsilon(\cdot)$ such that

$$\Pr\left[\begin{array}{l}\mathsf{pp} \leftarrow \mathsf{Setup}(1^\kappa, \mathsf{aux}),\ b \xleftarrow{R} \{0,1\},\\ (\mathsf{osk}, \mathsf{opk}) \leftarrow \mathsf{OrgKeyGen}(\mathsf{pp}),\\ (\mathsf{rsk}, \mathsf{rpk}) \leftarrow \mathsf{RAKeyGen}(\mathsf{pp}),\\ \mathcal{O} \leftarrow \{\mathcal{O}^{\mathtt{HU}+}(\cdot), \mathcal{O}^{\mathtt{CU}+}(\cdot,\cdot), \mathcal{O}^{\mathtt{KU}+}(\cdot), \mathcal{O}^{\mathtt{RN}+}(\mathsf{rsk}, \mathsf{rpk}, \cdot), \ : b^* = b\\ \mathcal{O}^{U_\mathsf{I}}(\mathsf{osk}, \mathsf{opk}, \mathsf{rsk}, \mathsf{rpk}, \cdot, \cdot, \cdot), \mathcal{O}^{U_\mathsf{V}}(\mathsf{opk}, \mathsf{rpk}, \cdot, \cdot, \mathtt{RI}[0]),\\ \mathcal{O}^{LoR}(\mathsf{osk}, \mathsf{opk}, \mathsf{rsk}, \mathsf{rpk}, b, \cdot, \cdot, \cdot, \mathtt{RI}[0])\},\\ b^* \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{pp}, \mathsf{osk}, \mathsf{opk}, \mathsf{rsk}, \mathsf{rpk})\end{array}\right] - \frac{1}{2} \leq \epsilon(\kappa).$$

Observe, that the pseudonyms contained in $\mathtt{NYM}_{LoR}$ can later be revoked using the $\mathcal{O}^{\mathtt{RN}+}$ oracle. This explicitly requires that even if pseudonyms get revoked and the adversary has access to all previous showing transcripts, users still remain anonymous (backward unlinkability).

4 Construction of the RABC System

We first recall the intuition behind the ABC system in [HS]. Then, we present the intuition behind our construction and finally we present our RABC system.

4.1 Intuition of the ABC System

The ABC construction in [HS] requires an EUF-CMA secure SPS-EQ-$\mathcal{R}$ scheme with perfect adaption of signatures and DDH holding on the message space (subsumed as class-hiding property in [HS]; e.g., Scheme 3). It further requires randomizable polynomial commitments with factor openings (PolyCommitFO, cf. [HS]) and one single, constant-size ZKPK to prevent replays of previously shown credentials. Below, we recall how the building blocks are combined.

In [HS], a credential cred_i for user i is a vector of two group elements (C_1, P) together with a signature of the organization under the SPS-EQ-$\mathcal{R}$ scheme, where C_1 is a polynomial commitment to a polynomial that encodes the attribute set $\mathbb{A}$ of the credential. The encoding of the attribute set $\mathbb{A} = \{(\mathtt{attr}_k, \mathtt{attrV}_k)\}_{k=1}^{n}$ to a polynomial in $\mathbb{Z}_p[X]$ is defined by the following encoding function, where $H : \{0,1\}^* \to \mathbb{Z}_p^*$ is a collision-resistant hash function:

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

Additionally, C_1 includes the private key r_i corresponding to the public key $R_i = r_i P$ of user i.

On a showing for some attribute set $\mathbb{A}' \sqsubseteq \mathbb{A}$, a credential owner proceeds as follows. To achieve unlinkability, the user randomizes the credential using a random scalar ρ. This is simply done by changing the representative of (C_1, P) with signature σ to the representative $\rho(C_1, P)$ and signature σ' (using $\mathsf{ChgRep}_{\mathcal{R}}$ of SPS-EQ-$\mathcal{R}$). Then, a user provides the randomized credential together with a selective opening of the polynomial commitment ρC_1 with respect to the encoding of the revealed attributes $\mathsf{enc}(\mathbb{A}')$. This so called factor opening includes a consistently randomized witness (by using ρ), attesting that $\mathbb{A}' \sqsubseteq \mathbb{A}$ while hiding the unrevealed attribute set $\overline{\mathbb{A}'}$.[3] Thereby, the rerandomization of PolyCommitFO is compatible with the rerandomization of the SPS-EQ-$\mathcal{R}$ scheme. Additionally, the user provides a ZKPK (denoted PoK) to demonstrate knowledge of ρ in ρP with respect to P to guarantee freshness, i.e., to prevent replaying of past showings.

Now, to verify a credential, the verifier starts by checking the signature σ' on the obtained credential $(\rho C_1, \rho P)$ (using the organization's SPS-EQ-$\mathcal{R}$ public key). Then, it verifies whether the factor opening to $\mathsf{enc}(\mathbb{A}')$ is correct with respect to the randomized polynomial commitment ρC_1 (via $\mathsf{VerifyFactor}_{\mathsf{PC}}$ [HS]). In particular, it checks whether the polynomial that encodes $\mathbb{A}'$ is indeed a factor of the polynomial committed to in ρC_1 by using the witness to $\overline{\mathbb{A}'}$ and without learning anything about $\overline{\mathbb{A}'}$. By construction this also guarantees that the prover knows the respective secret key (without revealing it). Furthermore, the verifier only accepts if PoK holds to guarantee that the showing is fresh (and no replay).

[3] Such a witness is basically a consistently randomized commitment (by using ρ) to $\overline{\mathbb{A}'}$.

Example: To illustrate the attribute sets, we restate a short example from [HS]. Suppose that we are given a user with the following attribute set: $\mathbb{A} = \{(\texttt{age}, \{> \texttt{16}, > \texttt{18}\}), (\texttt{drivinglicense}, \{\#, \texttt{car}\})\}$, where # indicates an attribute value that proves the possession of an attribute without revealing any concrete value. A showing could involve the attributes $\mathbb{A}' = \{(\texttt{age}, \{> \texttt{18}\}), (\texttt{drivinglicense}, \{\#\})\}$ and its hidden complement $\overline{\mathbb{A}'} = \{(\texttt{age}, \{> \texttt{16}\}), (\texttt{drivinglicense}, \{\texttt{car}\})\}$.

4.2 Incorporating Blacklist Revocation

To enable revocation, we need to augment the credentials in the ABC construction of [HS] to include a unique nym. Recall that in our context pseudonyms are more or less credential identifiers that are never being revealed during showings and solely used for revocation purposes. In a nutshell, the revocation authority holds a list of revoked nyms $\texttt{RNYM} = \{\mathsf{nym}_i\}_{i\in[n]}$ and unrevoked nyms $\texttt{NYM} = \{\mathsf{nym}_i\}_{i\in[m]}$, respectively. It publishes an accumulator $\mathit{\Pi}$, which represents the list of revoked pseudonyms $\texttt{RNYM}$. Additionally, the revocation authority maintains a public list $\texttt{WIT}$ of non-membership witnesses $\{\omega_{\mathsf{nym}_i}\}_{i\in[m]}$ for unrevoked users. An unrevoked user then demonstrates that the nym encoded in the credential has not been blacklisted, i.e., nym is not contained in the accumulator, during a showing. We assume that two dummy nyms are initially inserted into the accumulator so that the accumulator $\mathit{\Pi}$ as well as witnesses ω_{nym_i} match the form, which is required for the respective algorithms to work. We emphasize that, in contrast to existing accumulator-based approaches, we avoid to prove in zero-knowledge the possession of such a non-membership witness which satisfies the accumulator verification relation. Furthermore, we note that one could also allow the users to update their witnesses on their own by using the dynamic features of the accumulator construction in [ATSM09].

4.3 Our Construction

Our revocation mechanism is based on the observation that the accumulator in Scheme 1 is compatible with the rerandomizations of the credentials (due to similarities between Scheme 1 and $\mathsf{PolyCommitFO}$ in [HS]). In particular, we extend the original credential by two values C_2 and C_3, resulting in a credential $\mathsf{cred} = ((C_1, C_2, C_3, P), \sigma)$. We choose the second credential component C_2 to be $C_2 = u_i(\lambda P - \mathsf{nym} \cdot P)$ (which can directly be used in the $\mathsf{Verify}_{\mathsf{Acc}}$ algorithm). Here, u_i is an additional user secret key that is required for anonymity (similar to the secret r_i in C_1) and corresponds to $U_i = u_i P$ in the augmented public key (R_i, U_i). Furthermore, for technical reasons, we include a third credential component $C_3 = u_i Q$, where Q (as in the original scheme) is a random element in $\mathbb{G}_1$ with unknown discrete logarithm. During showings, rerandomized versions of the credential will be presented, which is due to the nature of the credential scheme in [HS]. To preserve the correctness of the accumulator verification relation, the prover must present consistently rerandomized versions of the accumulator $\mathit{\Pi}$

Setup: Given $(1^\kappa, \mathsf{aux})$, parse $\mathsf{aux} \leftarrow (t, t')$, runs $\mathsf{pp}' = (\mathsf{BG}, (\alpha^i P)_{i\in[t]}, (\alpha^i \hat{P})_{i\in[t]}) \leftarrow \mathsf{Setup}_{\mathsf{PC}}(1^\kappa, t)$ and $\mathsf{pp}'' = ((\lambda^i P)_{i\in[t']}, (\lambda^i \hat{P})_{i\in[t']}) \leftarrow \mathsf{Gen}_{\mathsf{Acc}}(\mathsf{BG}, t')$. Then, let $\mathbf{g} \leftarrow e(P, \hat{P})$ and $H_s : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$ be a collision-resistant keyed hash function used inside $\mathsf{enc}(\cdot)$, drawn uniformly at random from a family of collision-resistant keyed hash functions $\{(H_s, s)\}_{s\in S}$. Finally, choose $Q \xleftarrow{R} \mathbb{G}_1$ and output $\mathsf{pp} \leftarrow (H_s, \mathsf{enc}, Q, \mathbf{g}, \mathsf{pp}', \mathsf{pp}'')$.

RAKeyGen: Given pp return $(\mathsf{rsk}, \mathsf{rpk}) \leftarrow (\emptyset, \mathsf{pp}'')$.

OrgKeyGen: Given pp, return $(\mathsf{osk}, \mathsf{opk}) \leftarrow \mathsf{KeyGen}_{\mathcal{R}}(1^\kappa, \ell = 4)$.

UserKeyGen: Given pp, pick $r, u \xleftarrow{R} \mathbb{Z}_p^*$, compute $(R, U) \leftarrow (rP, uP)$ and return $(\mathsf{usk}, \mathsf{upk}) \leftarrow ((r, u), (R, U))$.

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

$\mathsf{Issue}(\mathsf{pp}, \mathsf{rpk}, \mathsf{upk}_i, \mathsf{osk}_j, \mathsf{nym}, \mathbb{A})$		$\mathsf{Obtain}(\mathsf{pp}, \mathsf{usk}_i, \mathsf{opk}_j, \mathsf{nym}, \mathbb{A})$
$e(C_1, \hat{P}) = e(R_i, \mathsf{enc}(\mathbb{A})(\alpha)\hat{P})$	$\xleftarrow{C_1, C_2, C_3}$	$(C_1, C_2, C_3) \leftarrow (r_i \mathsf{enc}(\mathbb{A})(\alpha)P,$
$e(C_2, \hat{P}) = e(U_i, \lambda\hat{P} - \mathsf{nym} \cdot \hat{P})$	$\xleftrightarrow{\mathsf{PoK}}$	$u_i(\lambda P - \mathsf{nym} \cdot P), u_i Q)$
$\sigma \leftarrow \mathsf{Sign}_{\mathcal{R}}((C_1, C_2, C_3, P),$	$\xrightarrow{\sigma}$	$\mathsf{Verify}_{\mathcal{R}}((C_1, C_2, C_3, P), \sigma, \mathsf{opk}_j) = 1$
$\mathsf{osk}_j)$		$\mathsf{cred}_{\mathsf{nym}} \leftarrow ((C_1, C_2, C_3, P), \sigma)$

where PoK is: $\mathsf{PoK}\{(\psi) : C_3 = \psi Q \ \ \wedge \ \ U_i = \psi P\}$.

(Show, Verify): Show and Verify interact in the following way, where $\mathbb{R}_V = \mathit{\Pi} \leftarrow \mathbb{R}[1]$ and $\mathbb{R}_S^{\mathsf{nym}} = (\mathit{\Pi}, (\hat{W}, d)) \leftarrow (\mathbb{R}[1], \mathbb{R}[2][\mathsf{nym}])$:

$\mathsf{Verify}(\mathsf{pp}, \mathsf{rpk}, \mathsf{opk}_j, \mathbb{A}', \mathbb{R}_V)$		$\mathsf{Show}(\mathsf{pp}, \mathsf{rpk}, \mathsf{usk}_i, \mathsf{opk}_j, \mathsf{cred}_{\mathsf{nym}}, \mathbb{A}, \mathbb{A}', \mathbb{R}_S^{\mathsf{nym}})$
		$\rho, \nu \xleftarrow{R} \mathbb{Z}_p^*$
		$(\hat{W}', \mathbf{d}') \leftarrow (\nu\hat{W},\ e(\rho\nu u_i d P, \hat{P}))$
		$\mathit{\Pi}' \leftarrow \rho\nu u_i \mathit{\Pi}$
		$\mathsf{cred} \leftarrow \mathsf{ChgRep}_{\mathcal{R}}(\mathsf{cred}_{\mathsf{nym}}, \rho, \mathsf{opk}_j)$
$\Big[\mathsf{Verify}_{\mathcal{R}}(\mathsf{cred}, \mathsf{opk}_j) \ \wedge$	$\xleftarrow{\mathsf{cred}, \mathcal{C}_{\overline{\mathbb{A}'}}, \mathit{\Pi}', \hat{W}', \mathbf{d}'}$	$\mathcal{C}_{\overline{\mathbb{A}'}} \leftarrow (\rho \cdot r_i) \cdot \mathsf{enc}(\overline{\mathbb{A}'})(\alpha)P$
$\mathbf{d}' \neq 1_{\mathbb{G}_T} \ \wedge \ \mathsf{VerifyFactor}_{\mathsf{PC}}(\mathsf{pp}',$		
$C_1, \mathsf{enc}(\mathbb{A}'), \mathcal{C}_{\overline{\mathbb{A}'}}) \ \wedge \ \mathsf{Verify}_{\mathsf{Acc}}(\mathit{\Pi}',$		
$(\hat{W}', \mathbf{d}'), C_2, \mathsf{pp}'')\Big] = 1$	$\xleftrightarrow{\mathsf{PoK}}$	

where $\mathsf{cred} = ((C_1, C_2, C_3, C_4), \sigma)$ and PoK is: $\mathsf{PoK}\{(\gamma, \delta, \eta, \zeta, \psi) :\ Q = \eta P \vee (C_3 = \psi Q \ \wedge \ C_4 = \gamma P \ \wedge \ \mathbf{d}' = \mathbf{g}^\delta \ \wedge \ \mathit{\Pi}' = \zeta \mathit{\Pi})\}$.

Revoke: Given pp, $(\mathsf{rsk}, \mathsf{rpk})$, $\mathtt{NYM}$ and $\mathtt{RNYM}$, this algorithm computes $\mathit{\Pi} \leftarrow \mathsf{Eval}_{\mathsf{Acc}}(\mathtt{RNYM}, (\emptyset, \mathsf{pp}''))$. Then, for all $\mathsf{nym} \in \mathtt{NYM}$ it computes $(W'_{\mathsf{nym}}, d_{\mathsf{nym}}) \leftarrow \mathsf{WitCreate}_{\mathsf{Acc}}(\mathit{\Pi}, \mathtt{RNYM}, \mathsf{nym}, (\emptyset, \mathsf{pp}''))$, sets $\mathtt{WIT}[\mathsf{nym}] \leftarrow (W'_{\mathsf{nym}}, d_{\mathsf{nym}})$ and returns $\mathbb{R} \leftarrow (\mathit{\Pi}, \mathtt{WIT})$.

Scheme 2: Our Multi-Show RABC System.

(and of the non-membership witnesses as well). Apparently, the prover must be restricted to present only honestly rerandomized versions thereof.[4]

[4] To ensure the authenticity of the rerandomized revocation information, we require users to prove knowledge of the randomizer used for randomizing the original accu-

Scheme 2 presents our RABC system, where we require t, t' to be bounded by $\mathsf{poly}(\kappa)$. If a check does not yield 1 or a PoK is invalid, the respective algorithm terminates with a failure and the algorithm Verify accepts only if $\mathsf{Verify}_{\mathcal{R}}$, $\mathsf{VerifyFactor}_{\mathsf{PC}}$, $\mathsf{Verify}_{\mathsf{Acc}}$ return 1. Note that in Scheme 2, we use a slightly modified version of the algorithm $\mathsf{Verify}_{\mathsf{Acc}}$, which directly takes $\mathbf{d} = e(dP, \hat{P})$ instead of a scalar d as part of the witness (as done in Scheme 1). This version uses the verification relation $e(\Pi, \hat{P}) = e(\lambda P - yP, \hat{W}) \cdot \mathbf{d}$. Also note that the prover can compute the commitment of the $\mathbf{d}'$-part of the proof using a pairing, which is typically faster than a corresponding exponentiation in $\mathbb{G}_T$ in state-of-the-art pairing implementations. In addition to PoK on the discrete logarithm of $\mathbf{d}'$, we must also check whether $\mathbf{d}' \neq 1$ to ensure the correct form of the presented witness $(\hat{W}', \mathbf{d}')$ (recall that $d \neq 0$ is required). Furthermore, the accumulator Π needs to be available in an authentic fashion. Finally, we note that the first move in the showing protocol can be combined with the first move of PoK. Thus, a showing consists of a total of three moves.

4.4 Security of the RABC System

Theorem 2. *The RABC system in Scheme 2 is correct.*

The correctness of Scheme 2 follows from inspection.

Theorem 3. *If* $\mathsf{PolyCommitFO}$ *is factor-sound,* $\{(H_s, s)\}_{s \in S}$ *is a collision-resistant hash function family, the underlying SPS-EQ-*$\mathcal{R}$ *is* $\mathsf{EUF\text{-}CMA}$ *secure and perfectly adapts signatures,* Acc *is collision-free and the DDH assumption holds in* $\mathbb{G}_1$*, then Scheme 2 is unforgeable.*

We prove Theorem 3 in Appendix B.1. Now, for anonymity of Scheme 2 we introduce two plausible assumptions in the Type-3 bilinear group setting.

Definition 11. Let BG be a bilinear group with $\log_2 p = \lceil \kappa \rceil$. Then, for every PPT adversary $\mathcal{A}$ there is a negligible function $\epsilon(\cdot)$ such that

$$\Pr\left[\begin{matrix} b \xleftarrow{R} \{0,1\},\ r,s,t,u,v \xleftarrow{R} \mathbb{Z}_p, b^* \leftarrow \mathcal{A}(\mathsf{BG}, rP, r\hat{P}, sP, s\hat{P}, \\ tP, ru\hat{P}, stuP, \mathbf{g}^{(1-b)\cdot v + b \cdot ut}) \end{matrix} : b^* = b \right] - \frac{1}{2} \leq \epsilon(\kappa).$$

We emphasize that the assumption in Definition 11 can easily be justified in the uber-assumption framework [Boy08], i.e., by setting $\mathsf{R} = \langle 1, r, s, t, stu \rangle, \mathsf{S} = \langle 1, r, s, ru \rangle, \mathsf{T} = \langle 1 \rangle, f = ut$. The subsequent assumption is closely related to the assumption in Definition 11, but does not fit the uber-assumption framework due to the decision-part being in $\mathbb{G}_2$. Consequently, we analyze the assumption in the generic group model.

mulator and for proof-technical reasons we require the user to prove knowledge of $\log_Q C_3$.

Definition 12. Let BG be a bilinear group with $\log_2 p = \lceil \kappa \rceil$. Then, for every PPT adversary $\mathcal{A}$ there is a negligible function $\epsilon(\cdot)$ such that

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

Proposition 1. *The assumption in Definition 12 holds in generic Type-3 bilinear groups and reaches the optimal, quadratic simulation error bound.*

The proof of the above proposition is given in Appendix B.2.

Theorem 4. *If the underlying SPS-EQ-$\mathcal{R}$ perfectly adapts signatures, DDH in $\mathbb{G}_1$ and the assumptions in Definition 11 and 12 hold, then Scheme 2 is anonymous.*

We prove Theorem 4 in Appendix B.3.

5 Discussion

The presented revocation mechanism for the RABC system uses similar building blocks as the original ABC system. In particular, it does not use a complex ZKPK for demonstrating the knowledge of a non-membership witness, which satisfies the verification relation of the accumulator. It only requires a simple ZKPK of the dicrete logarithms in $\mathbf{d}', \Pi', C_3$ (and C_4 which is already required in the original ABC system from [HS]) for technical reasons. Consequently, this concept yields a new direction for revocation in ABC systems.

To evaluate our approach, we additionally adapted the accumulator-based blacklist revocation from Microsoft's U-Prove to our setting (see Appendix C for the adapted scheme and a security proof of this approach in the model proposed in this paper). Here, one uses a (relatively complex) ZKPK of a nym encoded in a credential and a non-membership witness for the same value nym such that the verification relation of the accumulator holds. Since in most settings the user is the only resource-constrained entity, it is most reasonable to compare the two proposed approaches based on the user's computational effort. Even though both proposed RABCs can be instantiated with any Type-3 bilinear-group-setting based, $\mathsf{EUF\text{-}CMA}$-secure SPS-EQ-$\mathcal{R}$ that perfectly adapts signatures, we instantiate both of them with the one from [FHS14] in our comparison. Table 1 shows the number of revocation-induced operations on the user side: revocation-induced pairing operations, scalar multiplications in $\mathbb{G}_1$ and $\mathbb{G}_2$ and exponentiations in $\mathbb{G}_T$. To obtain a comparable representation, we convert all operations to their "$\mathbb{G}_1$ equivalents" (based on the computation times on an ARM Cortex-M0+ with a drop-in hardware accelerator [UW14]) and sum them up, which shows that our approach is up to a factor of 1.65 faster than the classical approach on constrained devices. We emphasize that this factor even increases to 2, when using the performance values from the plain ARM-Cortex-M0+-based implementation in [UW14].

Scheme 2		$\mathbb{G}_1$	$\mathbb{G}_2$	$\mathbb{G}_T$	e	**Sum**
Obtain	Commit	3				
	PoK	2				
	$\mathsf{Verify}_{\mathcal{R}}$				2	
Sum		5			2	
$\mathbb{G}_1$ **equivalents**		5			10	**+15**
Show	Blind	2	1		1	
	ChgRep	2				
	PoK	3			1	
Sum		7	1		2	
$\mathbb{G}_1$ **equivalents**		7	3		10	**+20**

Scheme 4		$\mathbb{G}_1$	$\mathbb{G}_2$	$\mathbb{G}_T$	e	**Sum**
Obtain	Commit	2				
	PoK	2				
	$\mathsf{Verify}_{\mathcal{R}}$				2	
Sum		4			2	
$\mathbb{G}_1$ **equivalents**		4			10	**+14**
Show	Blind	8				
	ChgRep	2				
	PoK	23				
Sum		33				
$\mathbb{G}_1$ **equivalents**		33				**+33**

Table 1. Number of revocation-induced operations for the user. To obtain the $\mathbb{G}_1$ equivalents for comparison, we use the performance values of a BN-pairing implementation (254-bit curves) on an ARM-Cortex-M0+ with a drop-in hardware accelerator, operating at 48MHz [UW14]. This delivers the following performance values 33ms-101ms-252ms-164ms ($\mathbb{G}_1$-$\mathbb{G}_2$-$\mathbb{G}_T$-pairing), which we norm to 1-3-7.6-5.

Acknowledgements. We would like to thank the anonymous reviewers for their valuable comments.

References

[ACN13] Tolga Acar, Sherman S. M. Chow, and Lan Nguyen. Accumulators and U-Prove Revocation. In *Financial Cryptography*, LNCS. 2013.

[AMO08] Norio Akagi, Yoshifumi Manabe, and Tatsuaki Okamoto. An Efficient Anonymous Credential System. In *Financial Cryptography*, LNCS. Springer, 2008.

[ATSM09] Man Ho Au, Patrick P. Tsang, Willy Susilo, and Yi Mu. Dynamic Universal Accumulators for DDH Groups and Their Application to Attribute-Based Anonymous Credential Systems. In *CT-RSA*, LNCS. Springer, 2009.

[BCC+09] Mira Belenkiy, Jan Camenisch, Melissa Chase, Markulf Kohlweiss, Anna Lysyanskaya, and Hovav Shacham. Randomizable Proofs and Delegatable Anonymous Credentials. In *CRYPTO*, LNCS. Springer, 2009.

[BdM93] Josh Cohen Benaloh and Michael de Mare. One-Way Accumulators: A Decentralized Alternative to Digital Signatures (Extended Abstract). In *EUROCRYPT*, LNCS. Springer, 1993.

[BG92] Mihir Bellare and Oded Goldreich. On Defining Proofs of Knowledge. In *CRYPTO*, LNCS. Springer, 1992.

[BL13] Foteini Baldimtsi and Anna Lysyanskaya. Anonymous Credentials Light. In *ACM CCS*. ACM, 2013.

[BNF12] Nasima Begum, Toru Nakanishi, and Nobuo Funabiki. Efficient Proofs for CNF Formulas on Attributes in Pairing-Based Anonymous Credential System. In *ICISC*, LNCS. Springer, 2012.

[Boy08] Xavier Boyen. The Uber-Assumption Family – A Unified Complexity Framework for Bilinear Groups. In *PAIRING*, LNCS. Springer, 2008.

[Bra00] Stefan Brands. *Rethinking public-key Infrastructures and Digital Certificates: Building in Privacy.* MIT Press, 2000.

[BS02] Emmanuel Bresson and Jacques Stern. Proofs of Knowledge for Non-Monotone Discrete-Log Formulae and Applications. In *ISC*, LNCS, 2002.

[BS04] Dan Boneh and Hovav Shacham. Group Signatures with Verifier-Local Revocation. In *ACM CCS*, 2004.

[CDHK] Jan Camenisch, Maria Dubovitskaya, Kristiyan Haralambiev, and Markulf Kohlweiss. Composable & modular anonymous credentials: Definitions and practical constructions. IACR Cryptology ePrint Archive.

[CH02] Jan Camenisch and Els Van Herreweghen. Design and implementation of the *idemix* anonymous credential system. In *ACM CCS*. ACM, 2002.

[Cha85] David Chaum. Security without identification: Transaction systems to make big brother obsolete. *Commun. ACM*, (10), 1985.

[Che06] Jung Hee Cheon. Security Analysis of the Strong Diffie-Hellman Problem. In *EUROCRYPT*, LNCS. Springer, 2006.

[CKL+15] Jan Camenisch, Stephan Krenn, Anja Lehmann, Gert Læssøe Mikkelsen, Gregory Neven, and Michael Ostergaard Pedersen. Formal Treatment of Privacy-Enhancing Credential Systems, 2015.

[CKS09] Jan Camenisch, Markulf Kohlweiss, and Claudio Soriente. An Accumulator Based on Bilinear Maps and Efficient Revocation for Anonymous Credentials. In *Public Key Cryptography*, LNCS. Springer, 2009.

[CL01] Jan Camenisch and Anna Lysyanskaya. An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. In *EUROCRYPT*, LNCS. Springer, 2001.

[CL02a] Jan Camenisch and Anna Lysyanskaya. A Signature Scheme with Efficient Protocols. In *SCN*, LNCS. Springer, 2002.

[CL02b] Jan Camenisch and Anna Lysyanskaya. Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials. In *CRYPTO*, LNCS. Springer, 2002.

[CL04] Jan Camenisch and Anna Lysyanskaya. Signature Schemes and Anonymous Credentials from Bilinear Maps. In *CRYPTO*, LNCS. Springer, 2004.

[CL11] Sébastien Canard and Roch Lescuyer. Anonymous credentials from (indexed) aggregate signatures. In *DIM*. ACM, 2011.

[CL13] Sébastien Canard and Roch Lescuyer. Protecting privacy by sanitizing personal data: a new approach to anonymous credentials. In *ASIA CCS*. ACM, 2013.

[CLNR14] Jan Camenisch, Anja Lehmann, Gregory Neven, and Alfredo Rial. Privacy-Preserving Auditing for Attribute-Based Credentials. In *ESORICS 2014, Part II*, LNCS. Springer, 2014.

[CMZ14] Melissa Chase, Sarah Meiklejohn, and Gregory M. Zaverucha. Algebraic MACs and Keyed-Verification Anonymous Credentials. In *ACM CCS*. ACM, 2014.

[CS97] Jan Camenisch and Markus Stadler. Efficient Group Signature Schemes for Large Groups (Extended Abstract). In *CRYPTO*, LNCS. Springer, 1997.

[DHS15] David Derler, Christian Hanser, and Daniel Slamanig. Revisiting Cryptographic Accumulators, Additional Properties and Relations to other Primitives. In *CT-RSA*, LNCS. Springer, 2015.

[FHS14] Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig. EUF-CMA-Secure Structure-Preserving Signatures on Equivalence Classes. IACR Cryptology ePrint Archive, 2014.

[FHS15] Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig. Practical Round-Optimal Blind Signatures in the Standard Model. In *CRYPTO*, LNCS. Springer, 2015.

[Fuc14] Georg Fuchsbauer. Breaking Existential Unforgeability of a Signature Scheme from Asiacrypt 2014. IACR Cryptology ePrint Archive, 2014.

[GGM14] Christina Garman, Matthew Green, and Ian Miers. Decentralized Anonymous Credentials. In *NDSS*, 2014.

[Gol08] Oded Goldreich. *Computational Complexity - A Conceptual Perspective*. Cambridge University Press, 2008.

[HKK14] Lucjan Hanzlik, Kamil Kluczniak, and Miroslaw Kutylowski. Attack on a U-Prove Revocation Scheme from FC'13 - Exploiting the Weakness of the Underlying Accumulator Scheme (Short Paper) . In *Financial Cryptography*, LNCS. 2014. to appear.

[HKK15] Lucjan Hanzlik, Przemysław Kubiak, and Mirosław Kutyłowski. Tracing Attacks on U-Prove with Revocation Mechanism: Tracing Attacks for U-Prove. ASIA CCS, 2015.

[HM12] Jan Hajny and Lukas Malina. Unlinkable Attribute-Based Credentials with Practical Revocation on Smart-Cards. In *CARDIS*, LNCS. Springer, 2012.

[HS] Christian Hanser and Daniel Slamanig. Structure-Preserving Signatures on Equivalence Classes and their Application to Anonymous Credentials. In *ASIACRYPT*. Full Version: Cryptology ePrint Archive, Report 2014/705.

[Kat10] Jonathan Katz. *Digital Signatures*. Springer, 2010.

[LAHV15] Wouter Lueks, Gergely Alpr, Jaap-Henk Hoepman, and Pim Vullers. Fast revocation of attribute-based credentials for both users and verifiers. SEC, 2015.

[LKDN11] Jorn Lapon, Markulf Kohlweiss, Bart De Decker, and Vincent Naessens. Analysis of Revocation Strategies for Anonymous Idemix Credentials. In *CMS*, LNCS. Springer, 2011.

[LLX07] Jiangtao Li, Ninghui Li, and Rui Xue. Universal Accumulators with Efficient Nonmembership Proofs. In *ACNS*, LNCS. Springer, 2007.

[LRSW00] Anna Lysyanskaya, Ronald L. Rivest, Amit Sahai, and Stefan Wolf. Pseudonym Systems. In *SAC*, LNCS. Springer, 2000.

[NF05] Toru Nakanishi and Nobuo Funabiki. Verifier-Local Revocation Group Signature Schemes with Backward Unlinkability from Bilinear Maps. In *ASIACRYPT*, LNCS. Springer, 2005.

[NFHF09] Toru Nakanishi, Hiroki Fujii, Yuta Hira, and Nobuo Funabiki. Revocable Group Signature Schemes with Constant Costs for Signing and Verifying. In *Public Key Cryptography*, 2009.

[Ngu05] Lan Nguyen. Accumulators from Bilinear Pairings and Applications. In *CT-RSA*, 2005.

[NP14] Lan Nguyen and Christian Paquin. U-Prove Designated-Verifier Accumulator Revocation Extension. Technical report, Microsoft Research, 2014.

[PZ13] Christian Paquin and Greg Zaverucha. U-prove cryptographic specification v1.1, revision 3. Technical report, Microsoft Corporation, 2013.

[SNF11] Amang Sudarsono, Toru Nakanishi, and Nobuo Funabiki. Efficient Proofs of Attributes in Pairing-Based Anonymous Credential System. In *PETS*, LNCS. Springer, 2011.

[Son01] Dawn Xiaodong Song. Practical forward secure group signature schemes. In *ACM CCS*. ACM, 2001.

[UW14] Thomas Unterluggauer and Erich Wenger. Efficient Pairings and ECC for Embedded Systems. In *CHES*, LNCS. Springer, 2014.

[Ver01] Eric R. Verheul. Self-Blindable Credential Certificates from the Weil Pairing. In *ASIACRYPT*, LNCS. Springer, 2001.

A Security Models

A.1 Collision Resistant Hashing

Subsequently, we present the model for hash functions from [Kat10], which we tailor to the bilinear group setting.

Definition 13. A hash function is a pair of PPT algorithms (Gen, H), which are defined as follows:

$\mathsf{Gen}(\mathsf{BG})$**:** A probabilistic algorithm that takes a bilinear group description $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$ and outputs a key s (where BG is implicitly contained in s).

$H_s(x)$**:** A probabilistic algorithm that takes a key s and some input $x \in \{0,1\}^*$, and outputs a string $H_s(x) \in \mathbb{Z}_p$.

For security, we require a hash function to be collision resistant, i.e.,

Definition 14 (Collision Resistance). A hash function (Gen, H) is said to be collision resistant, if for all PPT adversaries $\mathcal{A}$, there is a negligible function $\epsilon(\cdot)$ such that:

$$\Pr\left[s \leftarrow \mathsf{Gen}(\mathsf{BG}), (x, x') \leftarrow \mathcal{A}(s) : x \neq x' \ \wedge \ H_s(x) = H_s(x')\right] \leq \epsilon(\kappa).$$

A.2 Formal Accumulator Model

We present the model for bounded accumulators from [DHS15], where we omit algorithms that we do not require.

Definition 15 (Accumulator). An *accumulator* is a tuple of efficient algorithms $(\mathsf{Gen}_{\mathsf{Acc}}, \mathsf{Eval}_{\mathsf{Acc}}, \mathsf{WitCreate}_{\mathsf{Acc}}, \mathsf{Verify}_{\mathsf{Acc}})$, which are defined as follows:

$\mathsf{Gen}_{\mathsf{Acc}}(\mathsf{BG}, t)$**:** This algorithm takes input the bilinear group parameters BG and an upper bound t for the number of elements to be accumulated and returns an accumulator keypair $(\mathsf{sk}_{\Pi}^{\sim}, \mathsf{pk}_{\Pi})$.

$\mathsf{Eval}_{\mathsf{Acc}}(\mathcal{X}, (\mathsf{sk}_{\Pi}^{\sim}, \mathsf{pk}_{\Pi}))$**:** Given a set $\mathcal{X}$ of values $\{x_1, \ldots, x_k\}$ to be accumulated and a keypair $(\mathsf{sk}_{\Pi}^{\sim}, \mathsf{pk}_{\Pi})$, this algorithm returns the accumulator $\Pi_{\mathcal{X}}$ together with the auxiliary information aux.

$\mathsf{WitCreate}_{\mathsf{Acc}}(\Pi_{\mathcal{X}}, \mathsf{aux}, y, (\mathsf{sk}_{\Pi}^{\sim}, \mathsf{pk}_{\Pi}))$**:** This algorithm takes an accumulator $\Pi_{\mathcal{X}}$ with corresponding auxiliary information aux, a value y and a keypair $(\mathsf{sk}_{\Pi}^{\sim}, \mathsf{pk}_{\Pi})$. It returns $\perp$, if y is in the accumulator and a witness ω_y, attesting that y is not contained in the accumulator, otherwise.

$\mathsf{Verify}_{\mathsf{Acc}}(\Pi, \omega_y, y, \mathsf{pk}_{\Pi})$**:** This algorithm takes an accumulator Π, a witness ω_y with corresponding non-member y and an accumulator public key pk_{Π} and verifies whether y is a non-member of Π. If so, it returns 1 and 0 otherwise.

We note that $\mathsf{sk}_{\Pi}^{\sim}$ indicates that the secret accumulator key is an optional parameter and correctness also needs to hold without having access to $\mathsf{sk}_{\Pi}^{\sim}$. The security requirements for an accumulator are *correctness* and *collision freeness*. Informally, correctness states that the $\mathsf{Verify}_{\mathsf{Acc}}$ algorithm returns 1 given an accumulator and a corresponding, correctly formed witness. Collision freeness in our setting requires that it is computationally infeasible to find non-membership witnesses for accumulated values. While we do not formally state correctness here, collision freeness is formally defined as follows, where we only consider non-membership witnesses.

Definition 16 (Collision Freeness). An accumulator ($\mathsf{Gen}_{\mathsf{Acc}}$, $\mathsf{Eval}_{\mathsf{Acc}}$,$\mathsf{WitCreate}_{\mathsf{Acc}}$,$\mathsf{Verify}_{\mathsf{Acc}}$) is said to be *collision-free*, if for all PPT adversaries $\mathcal{A}$ having oracle access to $\mathcal{O} \leftarrow \{\mathcal{O}^{\mathsf{E}(\cdot,\cdot)}, \mathcal{O}^{\mathsf{W}(\cdot,\mathsf{aux},\cdot,\cdot)}\}$, security parameters κ there is a negligible function $\epsilon(\cdot)$ such that:

$$\Pr\left[\begin{array}{c}(\mathsf{sk}_{\Pi},\mathsf{pk}_{\Pi}) \leftarrow \mathsf{Gen}_{\mathsf{Acc}}(\mathsf{BG},t), (\omega_{x_i}, x_i, \mathcal{X}^*, r) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{pk}_{\Pi}) : \\ \mathsf{Verify}_{\mathsf{Acc}}(\mathsf{pk}_{\Pi}, \Pi^*, \omega_{x_i}, x_i) = 1 \quad \wedge \quad x_i \in \mathcal{X}^*\end{array}\right] \leq \epsilon(\kappa),$$

where $\Pi^* \leftarrow \mathsf{Eval}_{\mathsf{Acc}}(\mathcal{X}^*, (\mathsf{sk}_{\Pi}, \mathsf{pk}_{\Pi}))$ and the oracles $\mathcal{O}^{\mathsf{E}}$ and $\mathcal{O}^{\mathsf{W}}$ allow the adversary to execute the $\mathsf{Eval}_{\mathsf{Acc}}$ and $\mathsf{WitCreate}_{\mathsf{Acc}}$ algorithms, respectively.

A.3 Proofs of Knowledge

In a proof of knowledge (PoK) [BG92], 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 [CS97] 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 $s = k + xc \mod p$.

A.4 Structure-Preserving Signatures on Equivalence Classes

Here, we discuss the abstract model and the security model of an SPS-EQ-$\mathcal{R}$ scheme, as presented in [HS, FHS14] and restate the SPS-EQ-$\mathcal{R}$ construction from [FHS14].

Definition 17 (Structure-Preserving Signature Scheme on Equivalence Classes (SPS-EQ-$\mathcal{R}$)). An *SPS-EQ-$\mathcal{R}$ scheme over* $\mathbb{G}_i$ (for $i \in \{1,2\}$) consists of the following PPT algorithms:

$\mathsf{BGGen}_{\mathcal{R}}(1^{\kappa})$: A deterministic bilinear-group generation algorithm, which on input a security parameter κ outputs an asymmetric bilinear group BG.

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

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

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

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

$\mathsf{VKey}_{\mathcal{R}}(\mathsf{sk}, \mathsf{pk})$ is a deterministic algorithm, which given a secret key sk and a public key pk outputs 1 if the keys are consistent and 0 otherwise.

From an SPS-EQ-$\mathcal{R}$, we require the following correctness property.

Definition 18 (Correctness). An SPS-EQ-$\mathcal{R}$ scheme on $(\mathbb{G}_i^*)^{\ell}$ is called *correct* if for all security parameters $\kappa \in \mathbb{N}$, $\ell > 1$, $\mathsf{BG} \leftarrow \mathsf{BGGen}_{\mathcal{R}}(1^{\kappa})$, $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}_{\mathcal{R}}(\mathsf{BG}, \ell)$, $M \in (\mathbb{G}_i^*)^{\ell}$ and $\rho \in \mathbb{Z}_p^*$:

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

For EUF-CMA security, outputting a valid message-signature pair, corresponding to an unqueried equivalence class, is considered to be a forgery.

Definition 19 (EUF-CMA). An SPS-EQ-$\mathcal{R}$ scheme on $(\mathbb{G}_i^*)^{\ell}$ is *existentially unforgeable under adaptively chosen-message attacks*, if for all PPT algorithms $\mathcal{A}$ with access to a signing oracle $\mathcal{O}$, there is a negligible function $\epsilon(\cdot)$ such that:

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

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

Besides EUF-CMA security, an additional security property for SPS-EQ-$\mathcal{R}$ was introduced in [FHS15].

Definition 20 (Perfect Adaption of Signatures [FHS15]). An SPS-EQ-$\mathcal{R}$ scheme $(\mathsf{BGGen}_{\mathcal{R}}, \mathsf{KeyGen}_{\mathcal{R}}, \mathsf{Sign}_{\mathcal{R}}, \mathsf{ChgRep}_{\mathcal{R}}, \mathsf{Verify}_{\mathcal{R}}, \mathsf{VKey}_{\mathcal{R}})$ on $(\mathbb{G}_i^*)^{\ell}$ *perfectly adapts signatures* if for all tuples $(\mathsf{sk}, \mathsf{pk}, M, \sigma, \rho)$:

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

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

$\mathsf{BGGen}_{\mathcal{R}}(1^\kappa)$**:** Given a security parameter κ, output $\mathsf{BG} \leftarrow \mathsf{BGGen}(1^\kappa)$.

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

$\mathsf{Sign}_{\mathcal{R}}(M, \mathsf{sk})$**:** Given a representative $M = (M_i)_{i\in[\ell]} \in (\mathbb{G}_1^*)^\ell$ and secret key $\mathsf{sk} = (x_i)_{i\in[\ell]}$, choose $y \xleftarrow{R} \mathbb{Z}_p^*$ and compute $Z \leftarrow y\sum_{i\in[\ell]} x_i M_i, \quad (Y, \hat{Y}) \leftarrow \frac{1}{y} \cdot (P, \hat{P})$. Then, output $\sigma = (Z, Y, \hat{Y})$ as signature for $[M]_{\mathcal{R}}$.

$\mathsf{ChgRep}_{\mathcal{R}}(M, \sigma, \rho, \mathsf{pk})$**:** Given a representative $M = (M_i)_{i\in[\ell]} \in (\mathbb{G}_1^*)^\ell$, the corresponding signature $\sigma = (Z, Y, \hat{Y})$, $\rho \in \mathbb{Z}_p^*$ and public key pk, pick $\psi \xleftarrow{R} \mathbb{Z}_p^*$ and return (M', σ'), where $\sigma' \leftarrow (\psi\rho Z, \frac{1}{\psi}Y, \frac{1}{\psi}\hat{Y})$ is the updated signature for the new representative $\rho \cdot (M_i)_{i\in[\ell]}$.

$\mathsf{Verify}_{\mathcal{R}}(M, \sigma, \mathsf{pk})$**:** Given a representative $M = (M_i)_{i\in[\ell]} \in (\mathbb{G}_1^*)^\ell$, a signature $\sigma = (Z, Y, \hat{Y})$ and public key $\mathsf{pk} = (\hat{X}_i)_{i\in[\ell]}$, check if $\prod_{i\in[\ell]} e(M_i, \hat{X}_i) = e(Z, \hat{Y}) \ \wedge \ e(Y, \hat{P}) = e(P, \hat{Y})$ and if so output 1 and 0 else.

$\mathsf{VKey}_{\mathcal{R}}(\mathsf{sk}, \mathsf{pk})$**:** Given $\mathsf{sk} = (x_i)_{i\in[\ell]}$ and $\mathsf{pk} = (\hat{X}_i)_{i\in[\ell]}$, output 1 if $x_i\hat{P} = \hat{X}_i \ \ \forall i \in [\ell]$ and 0 otherwise.

Scheme 3: The SPS-EQ-$\mathcal{R}$ Scheme from [FHS14].

Scheme 3 has been proven correct and EUF-CMA-secure in [FHS14] and further proven to fulfill Definition 20 in [FHS15].

B Security of Scheme 2

B.1 Proof of Theorem 3

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 valid showing so that $\mathsf{nym}^* = \perp$. Then, we construct an adversary $\mathcal{B}$ that uses $\mathcal{A}$ to break the EUF-CMA security of the SPS-EQ-$\mathcal{R}$.

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

- **Type 2A:** collision-resistance of the hash function used in the encoding $\mathsf{enc}(\cdot)$ of attributes.
- **Type 2B:** factor soundness of PolyCommitFO.

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

- **Type 3A:** we construct an adversary $\mathcal{B}$ that uses $\mathcal{A}$ to break the DLP in $\mathbb{G}_1$.

Type 3B: we show that the success probability of $\mathcal{A}$ is bounded by $\epsilon_{\mathrm{DDH}}(\kappa)+\epsilon_{\mathrm{DL}}(\kappa)$, where $\epsilon_{\mathrm{DDH}}(\kappa)$ and $\epsilon_{\mathrm{DL}}(\kappa)$ are the success probabilities for DDH and DLP in $\mathbb{G}_1$.

Type 4: Adversary $\mathcal{A}$ manages to conduct a showing protocol accepted by the verifier using some credential corresponding to a revoked pseudonym $\mathsf{nym}^* \in \mathtt{RN}$. Then, we construct an adversary $\mathcal{B}$ that uses $\mathcal{A}$ to break the collision-freeness of the accumulator scheme Acc.

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. For the PoK, we assume that the reduction always aborts if the respective discrete logarithm cannot be extracted because the wrong part of the OR statement was honestly computed. In Type 3B we make the abort probability explicit, whereas it is omitted in the other cases.

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 $\ell = 4$ and gives pk to $\mathcal{B}$. Then, $\mathcal{S}$ sets $\mathsf{opk} \leftarrow \mathsf{pk}$ and generates the public parameters pp and the revocation key pair $(\mathsf{rsk}, \mathsf{rpk})$ in way compatible to opk. Next, $\mathcal{S}$ runs $\mathcal{A}(\mathsf{pp}, \mathsf{opk}, \mathsf{rpk})$ and simulates the environment and the oracles. All oracles are as in a real game, but whenever $\mathcal{S}$ requires a signature it uses the signing oracle $\mathcal{O}(\mathsf{osk}, \cdot)$ of $\mathcal{C}$.

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 $(K_Q, K_{C_3}, K_{C_4}, \mathbf{k}_{\mathbf{d}'}, K_{\Pi'})$ in PoK and restarts $\mathcal{A}$ with a new challenge $c' \neq c$. Then, by the knowledge extractor of PoK, $\mathcal{S}$ obtains ρ. $\mathcal{S}$ now computes $\mathsf{cred}_0^* \leftarrow \rho^{-1} \cdot \mathsf{cred}'^*[0]$ on the message part of the credential. If there is some $\mathsf{cred}' \in \mathtt{CRED}$ such that $\mathsf{cred}'[0] = \mathsf{cred}_0^*$, then $\mathcal{S}$ and, in further consequence, $\mathcal{B}$ aborts. 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 the message-signature pair cred'^* as a forgery to $\mathcal{C}$ and $\mathcal{B}$ wins the unforgeability game.

Type 2A: Here, $\mathcal{B}$ consists of an adversary $\mathcal{A}$ playing the unforgeability game with a challenger $\mathcal{S}$. $\mathcal{B}$ is interacting with a challenger $\mathcal{C}$ in the collision freeness game of the hash function. $\mathcal{B}$ runs the setup as in the real game, except for the choice of (H_s, s), where it hands BG to $\mathcal{C}$ and obtains s. Then it runs $\mathcal{A}(\mathsf{pp}, \mathsf{opk}, \mathsf{rpk})$, where it simulates the oracles as in a 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 $(K_Q, K_{C_3}, K_{C_4}, \mathbf{k}_{\mathbf{d}'}, K_{\Pi'})$ 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 $C_4 = \rho P$). $\mathcal{B}$ now computes $\mathsf{cred}_0^* \leftarrow \rho^{-1} \cdot \mathsf{cred}'^*[0]$ on the message part of the credential. Let $\mathsf{cred}' \in \mathtt{CRED}$ be such that $\mathsf{cred}'[0] = \mathsf{cred}_0^*$. If there is no such cred', then $\mathcal{S}$ and, in further consequence, $\mathcal{B}$ aborts. Otherwise, let nym^* be such that $\mathsf{cred}' = \mathtt{CRED}[\mathsf{nym}^*]$ with $\mathsf{cred}'[0] = \mathsf{cred}_0^*$. Then, we know that $\mathsf{cred}_{\mathsf{nym}^*}^* \leftarrow \mathsf{ChgRep}_{\mathcal{R}}(\mathsf{cred}'^*, \rho^{-1}, \mathsf{pk})$ is—modulo a rerandomization of the signature part of the credential—identical to cred', which arises from an issue step during the simulation. Consequently, $\mathcal{B}$ knows the set of attributes $\mathbb{A}^* = \mathtt{ATTR}[\mathsf{nym}^*]$ corresponding to $\mathsf{cred}_{\mathsf{nym}^*}^*$. If $\mathbb{A}'^* \sqsubseteq \mathbb{A}^*$, then $\mathcal{B}$ aborts. Else, $\mathcal{B}$ can compute the corresponding polynomial $\mathsf{enc}(\mathbb{A}^*)$. If $\mathsf{enc}(\mathbb{A}'^*) \nmid \mathsf{enc}(\mathbb{A}^*)$, then $\mathcal{B}$ aborts. Else, $\mathsf{enc}(\mathbb{A}'^*) \mid \mathsf{enc}(\mathbb{A}^*)$ holds, but $\mathbb{A}'^* \not\sqsubseteq \mathbb{A}^*$. Thus, there is at least one factor $X - H_s(\mathtt{attr}_\ell^* \| M^*)$ of $\mathsf{enc}(\mathbb{A}'^*)$ and one factor $X - H_s(\mathtt{attr}_\ell \| M)$ of $\mathsf{enc}(\mathbb{A}^*)$ such that $H_s(\mathtt{attr}_\ell^* \| M^*) = H_s(\mathtt{attr}_\ell \| M)$ and $\mathtt{attr}_\ell^* \| M^* \neq \mathtt{attr}_\ell \| M$. Then, $\mathcal{B}$ outputs $(\mathtt{attr}_\ell^* \| M^*, \mathtt{attr}_\ell \| M)$ as collision in H_s.

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', generates the organization and revocation key pairs $(\mathsf{osk}, \mathsf{opk})$ and $(\mathsf{rsk}, \mathsf{rpk})$. $\mathcal{S}$ runs $\mathcal{A}(\mathsf{pp}, \mathsf{opk}, \mathsf{rpk})$ and simulates the oracles as in a 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 $(K_Q, K_{C_3}, K_{C_4}, \mathbf{k}_{\mathbf{d}'}, K_{\Pi'})$ 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 $C_4 = \rho P$). $\mathcal{S}$ now computes $\mathsf{cred}_0^* \leftarrow \rho^{-1} \cdot \mathsf{cred}'^*[0]$ on the message part of the credential. Let $\mathsf{cred}' \in \mathtt{CRED}$ be such that $\mathsf{cred}'[0] = \mathsf{cred}_0^*$. If there is no such cred', then $\mathcal{S}$ and, in further consequence, $\mathcal{B}$ aborts. Otherwise, let nym^* be such that $\mathsf{cred}' = \mathtt{CRED}[\mathsf{nym}^*]$. Then, we know that $\mathsf{cred}_{\mathsf{nym}^*}^* \leftarrow \mathsf{ChgRep}_{\mathcal{R}}(\mathsf{cred}'^*, \rho^{-1}, \mathsf{pk})$ is—modulo a rerandomization of the signature part of the credential—identical to cred', which arises from an issue step during the simulation. Consequently, $\mathcal{S}$ knows the set of attributes $\mathbb{A}^* = \mathtt{ATTR}[\mathsf{nym}^*]$ corresponding to $\mathsf{cred}_{\mathsf{nym}^*}^*$. If $\mathbb{A}'^* \sqsubseteq \mathbb{A}^*$, then $\mathcal{S}$ and, in further consequence, $\mathcal{B}$ aborts. Else, $\mathcal{S}$ can 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. Else, $\mathcal{B}$ outputs $(\rho, \mathsf{enc}(\mathbb{A}^*), \mathsf{enc}(\mathbb{A}'^*), \mathcal{C}_{\overline{\mathbb{A}'^*}})$, and wins 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 (BG, aP) with $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$ to the DLP in $\mathbb{G}_1$. Then, $\mathcal{B}$ runs the setup by generating public parameters pp based on BG and setting $Q \leftarrow aP$. It generates the organization key pair $(\mathsf{osk}, \mathsf{opk})$ as well as the revocation key pair $(\mathsf{rsk}, \mathsf{rpk})$, runs $\mathcal{A}(\mathsf{pp}, \mathsf{opk}, \mathsf{rpk})$ and simulates the oracles as in a 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 $(K_Q, K_{C_3}, K_{C_4}, \mathbf{k}_{\mathbf{d}'}, K_{\Pi'})$ 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 the proof was honestly computed with respect to Q, $\mathcal{B}$ outputs a as a solution to the given DLP instance and aborts otherwise.

Type 3B: In the following, we will show that the success probability of a Type-3B adversary is bounded by $\epsilon_{\mathrm{DDH}}(\kappa) + \epsilon_{\mathrm{DL}}(\kappa)$. In all games, the setup is as in the original game, with the following differences. Upon generation of the public parameter pp—instead of choosing Q at random—one chooses $q \xleftarrow{R} \mathbb{Z}_p^*$ and sets $Q \leftarrow qP$. Then, the environment stores q as well as the trapdoors α and λ used for generating the tuples $(\alpha^i P)_{i=0}^t, (\alpha^i \hat{P})_{i=0}^t$ and $(\lambda^i P)_{i=0}^t, (\lambda^i \hat{P})_{i=0}^t$ contained in pp.

Game 0: The original unforgeability game.

Game 1: As Game 0, but the PoK in all showings is conducted by honestly proving knowledge of q and simulating the proof part for the remainder.

Transition 1 - Game 0 → Game 1: Since the witness indistinguishability of the OR proof is unconditional, we have that $\Pr[S_1] = \Pr[S_0]$.

Game 2: As Game 1, except that all calls to $\mathsf{ChgRep}_{\mathcal{R}}(M, \sigma, \rho, \mathsf{pk})$ are replaced by $(\rho M, \mathsf{Sign}_{\mathcal{R}}(\rho M, \mathsf{sk}))$.

Transition 2 - Game 1 → Game 2: Since Scheme 3 perfectly adapts signatures, we have $\Pr[S_2] = \Pr[S_1]$.

Game 3: As Game 2, but if $\mathcal{A}$ delivers a valid showing using a credential cred^*, then $\mathcal{B}$ rewinds the experiment to the step after sending the commitments $(K_Q, K_{C_3}, K_{C_4}, \mathbf{k}_{\mathbf{d}'}, K_{\Pi'})$ in PoK and restarts $\mathcal{A}$ with a new challenge $c' \neq c$. Then, by the knowledge extractor of PoK (for the C_4-part of the proof), $\mathcal{B}$ obtains $\rho^* \in \mathbb{Z}_p^*$ and is able to determine the pseudonym nym^* of the shown credential cred'^* via computing $\mathsf{cred}_0^* \leftarrow \rho^{-1} \cdot \mathsf{cred}'^*[0]$ on the message part of the credential. Let F denote the event that there is no $\mathsf{cred}' \in \mathtt{CRED}$ such that $\mathsf{cred}'[0] = \mathsf{cred}_0^*$ or if $i^*_{\mathsf{nym}^*} \notin \mathtt{HU} \setminus \mathtt{KU}$. If F occurs, then $\mathcal{B}$ aborts.

Transition 3 - Game 2 → Game 3: Game 3 is equivalent to Game 2, unless abort event F happens. Event F occurs if and only if $\mathcal{A}$ is no Type-3B adversary, thus, $\Pr[F] = 6/7$. Thus, $\Pr[S_3] = \Pr[\neg F] \cdot \Pr[S_2] = (1 - \Pr[F]) \cdot \Pr[S_2] = 1/7 \cdot \Pr[S_2]$.

Game 4: As in Game 3, but $\mathcal{B}$ obtains the instance (BG, aP) with $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$ to the DLP in $\mathbb{G}_1$ and generates the public parameters pp based on BG. Furthermore, $\mathcal{B}$ simulates the oracles as in a real game, except for the oracle $\mathcal{O}^{U_V}$, which is simulated as follows:

$\mathcal{O}^{U_V}(\mathsf{opk}, \mathsf{nym}, \mathbb{A}', \mathbb{R}_V)$**:** $\mathcal{B}$ runs this oracle as in a real game, with the difference that $\mathcal{B}$ computes a credential $\mathsf{cred} \leftarrow (M, \sigma)$ with $M \leftarrow (\rho \cdot \mathtt{USK}[i_{\mathsf{nym}}][0] \cdot \mathsf{enc}(\mathtt{ATTR}[\mathsf{nym}])P, \rho \cdot \mathtt{USK}[i_{\mathsf{nym}}][1] \cdot (\lambda - \mathsf{nym})P, \rho \cdot \mathtt{USK}[i_{\mathsf{nym}}][1] \cdot q \cdot aP, \rho \cdot P)$, with $\rho \xleftarrow{R} \mathbb{Z}_p$ and $\sigma \leftarrow \mathsf{Sign}_{\mathcal{R}}(M, \mathsf{osk})$. The showing is then performed with respect to cred.

Transition 4 - Game 3 → Game 4: We have to show that the adversary cannot detect that the showings in $\mathcal{O}^{U_V}$ are performed with respect to a different

credential component C_3. To do so, we define the following two distributions containing the exchanged value C_3 and all other values containing discrete logarithms related to ρ and q (since we will base our indistinguishability proof on these values). Thereby, $\mathcal{D}_1$ resembles the distribution as in a real oracle simulation, whereas $\mathcal{D}_2$ resembles the distribution after exchanging the C_3 component of the credential. Both distributions are defined with respect to the adversary's view V on the system.

$$\begin{aligned}\mathcal{D}_1(V) :\equiv \Big[&\big(\mathsf{enc}(\mathbb{A})(\alpha)r_i \cdot \rho P, (\lambda - \mathsf{nym})u_i \cdot \rho P, u_i \cdot \rho q P, \rho P, qP, u_i P,\\ &\mathsf{enc}(\overline{\mathbb{A}'})(\alpha)r_i \cdot \rho P, \big(\nu u_i \textstyle\prod_{\mathsf{nym}\in\mathtt{RN}}(\lambda - \mathsf{nym})\big) \cdot \rho P, e\big((\nu u_i d) \cdot \rho P, \hat{P}\big)\big)\Big] \approx\\ \mathcal{D}_2(V) :\equiv \Big[&\big(\mathsf{enc}(\mathbb{A})(\alpha)r_i \cdot \rho P, (\lambda - \mathsf{nym})u_i \cdot \rho P, u_i \cdot \rho q a P, \rho P, qP, u_i P,\\ &\mathsf{enc}(\overline{\mathbb{A}'})(\alpha)r_i \cdot \rho P, \big(\nu u_i \textstyle\prod_{\mathsf{nym}\in\mathtt{RN}}(\lambda - \mathsf{nym})\big) \cdot \rho P, e\big((\nu u_i d) \cdot \rho P, \hat{P}\big)\big)\Big].\end{aligned}$$

To show that the distributions above are indistinguishable under DDH, we introduce the following intermediate distribution:

$$\begin{aligned}\mathcal{D}_3(V) :\equiv \Big[&\mathsf{b} \xleftarrow{R} \mathbb{Z}_p, \big(\mathsf{enc}(\mathbb{A})(\alpha)r_i \cdot \rho P, (\lambda - \mathsf{nym})u_i \cdot \rho P, \mathsf{b}P, \rho P, qP, u_i P,\\ &\mathsf{enc}(\overline{\mathbb{A}'})(\alpha)r_i \cdot \rho P, \big(\nu u_i \textstyle\prod_{\mathsf{nym}\in\mathtt{RN}}(\lambda - \mathsf{nym})\big) \cdot \rho P, e\big((\nu u_i d) \cdot \rho P, \hat{P}\big)\big)\Big]\end{aligned}$$

Now, assume a DDH instance $(P, \rho P, qP, \mathsf{r}P)$ and observe that this instance can easily be padded to the distributions $\mathcal{D}_1$ and $\mathcal{D}_3$, as we know all required remaining discrete logarithms. Then, we obtain a distribution identical to $\mathcal{D}_1$ if $\mathsf{r} = \rho q$, whereas we obtain a distribution identical to $\mathcal{D}_3$ if r is random.

Furthermore, observe that the distributions $\mathcal{D}_3$ and $\mathcal{D}_2$ are identical, since a is only contained in $u_i \cdot \rho q a P$ and (BG, aP) is a random DLP instance. All in all, we have $|\Pr[S_3] - \Pr[S_4]| \leq \epsilon_{\mathrm{DDH}}(\kappa)$.

Game 5: As Game 4, but $\mathcal{B}$ additionally obtains $\rho^* u_{i^*_{\mathsf{nym}^*}} a$ by the knowledge extractor of PoK (for the C_3 part of the credential).

Transition 5 - Game 4 → Game 5: Since this change is only conceptual, we have that $\Pr[S_4] = \Pr[S_5]$.

In Game 5, $\mathcal{B}$ can obtain $u_{i^*_{\mathsf{nym}^*}} \leftarrow \mathtt{USK}[i^*_{\mathsf{nym}^*}][1]$ and compute $a \leftarrow \frac{\rho^* u_{i^*_{\mathsf{nym}^*}} a}{\rho^* u_{i^*_{\mathsf{nym}^*}}}$ as a solution to the given DLP instance in $\mathbb{G}_1$, i.e., $\Pr[S_5] \leq \epsilon_{\mathrm{DL}}(\kappa)$. We have $\Pr[S_0] = \Pr[S_1] = \Pr[S_2] = \frac{\Pr[S_3]}{\Pr[\neg F]}$. Furthermore, we have that $|\Pr[S_3] - \Pr[S_4]| \leq \epsilon_{\mathrm{DDH}}(\kappa)$, yielding $\Pr[S_3] \leq \Pr[S_4] + \epsilon_{\mathrm{DDH}}(\kappa)$ and $\Pr[S_4] = \Pr[S_5] \leq \epsilon_{\mathrm{DL}}(\kappa)$. Taking all together, we have $\Pr[\neg F] \cdot \Pr[S_0] = \Pr[S_3] \leq \epsilon_{\mathrm{DDH}}(\kappa) + \epsilon_{\mathrm{DL}}(\kappa)$ and thus $\Pr[S_0] \leq 1/\Pr[\neg F] \cdot (\epsilon_{\mathrm{DDH}}(\kappa) + \epsilon_{\mathrm{DL}}(\kappa)) = 7 \cdot (\epsilon_{\mathrm{DDH}}(\kappa) + \epsilon_{\mathrm{DL}}(\kappa))$.

Type 4: 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 collision-freeness game of the accumulator scheme Acc. Subsequently, we describe how $\mathcal{S}$ simulates the environment for $\mathcal{A}$ and interacts with the challenger $\mathcal{C}$.

$\mathcal{B}$ obtains the accumulator public key $\mathsf{pk}_{\Pi} \leftarrow (\mathsf{BG}, (\lambda^i P)_{i=0}^{t'}, (\lambda^i \hat{P})_{i=0}^{t'})$ from $\mathcal{C}$. Then, $\mathcal{S}$ completes the setup and initializes rpk and the organization key pair $(\mathsf{osk}, \mathsf{opk})$ in a way compatible with pk_{Π}. $\mathcal{S}$ runs $\mathcal{A}(\mathsf{pp}, \mathsf{opk}, \mathsf{rpk})$ and simulates all oracles as in a 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^* and revocation information Π'^*, then $\mathcal{S}$ rewinds $\mathcal{A}$ to the step after sending the commitments $(K_Q, K_{C_3}, K_{C_4}, \mathbf{k}_{\mathbf{d}'}, K_{\Pi'})$ in PoK and restarts $\mathcal{A}$ with a new challenge $c' \neq c$. Rewinding yields the discrete logarithms ρu_i, ρ, d^* and π of C_3, C_4, $\mathbf{d}'$ and Π', respectively. $\mathcal{S}$ now computes $\mathsf{cred}_0^* \leftarrow \rho^{-1} \cdot \mathsf{cred}'^*[0]$ on the message part of the credential. Let $\mathsf{cred}' \in \mathtt{CRED}$ be such that $\mathsf{cred}'[0] = \mathsf{cred}_0^*$. If there is no such cred', then $\mathcal{S}$ and, in further consequence, $\mathcal{B}$ aborts. Otherwise, let nym^* be such that $\mathsf{cred}' = \mathtt{CRED}[\mathsf{nym}^*]$. Now, by the verification relation we know that $e(\pi \cdot \Pi, \hat{P}) = e(\rho u_i(\lambda - \mathsf{nym}^*)P, \hat{W}'^*) \cdot e(d^* P, \hat{P})$. By rearranging, we see that $e(\Pi, \hat{P}) = e((\lambda - \mathsf{nym})P, \frac{\rho u_i}{\pi}\hat{W}'^*) \cdot e(\frac{d^*}{\pi}P, \hat{P})$, which means that we can output $((\frac{\rho u_i}{\pi}\hat{W}'^*, \frac{d^*}{\pi}), \mathsf{nym}^*, \mathtt{RNYM})$ as a non-membership witness for an accumulated value giving a collision for the accumulator. □

B.2 Proof of Proposition 1

Proof. Let $\mathcal{A}$ be a generic PPT adversary and let $\sigma : \mathbb{G}_1 \to \{0,1\}^{m_1}$, $\hat{\sigma} : \mathbb{G}_2 \to \{0,1\}^{m_2}$ and $\tau : \mathbb{G}_T \to \{0,1\}^{m_T}$ be random encoding functions with w.l.o.g. $m_1 < m_2 < m_T$. $\mathcal{A}$ cannot work directly with group elements, but is forced to work with their image under $\sigma, \hat{\sigma}$ and τ. Furthermore, $\mathcal{A}$ is given oracle access to perform generic bilinear group operations (operations in $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ and pairings). As $\mathcal{A}$ is given access to the group element encodings, it can perform equality checks on its own through string equality tests. At last, we require that $\mathcal{A}$ can only submit already queried encodings to the group oracles (We can enforce this by choosing m_1, m_2 and m_T large enough making the probability of guessing bitstrings in the image of $\sigma, \hat{\sigma}$ and τ, respectively, negligible).

Now, let $\mathcal{B}$ be an algorithm interacting with $\mathcal{A}$ as follows. $\mathcal{B}$ picks a random bit b, picks $\sigma_0, \ldots, \sigma_4 \xleftarrow{R} \{0,1\}^{m_1}$ as encodings of $\mathbb{G}_1$ elements and assigns polynomials $1, R, S, T, STU$ to these values. Likewise, $\mathcal{B}$ picks $\hat{\sigma}_0, \hat{\sigma}_1, \hat{\sigma}_2, \hat{\sigma}_5 \xleftarrow{R} \{0,1\}^{m_2}$ as encodings of $\mathbb{G}_2$ elements and assigns polynomials $1, R, S, (1-b)V + b \cdot RU$ to these values. $\mathcal{B}$ stores $(1, \sigma_0), (R, \sigma_1), (S, \sigma_2), (T, \sigma_3), (STU, \sigma_4)$ in a list L_1 and $(1, \hat{\sigma}_0), (R, \hat{\sigma}_1), (S, \hat{\sigma}_2), ((1-b)V + b \cdot RU, \hat{\sigma}_5)$ in a list L_2 and gives the respective encodings to $\mathcal{A}$. Then, $\mathcal{B}$ initializes a list L_T to manage elements of $\mathbb{G}_T$ and simulates the group oracles as follows.

Group action in $\mathbb{G}_1$: Given two bitstrings σ_0, σ_1 representing elements in $\mathbb{G}_1$, $\mathcal{B}$ recovers the corresponding polynomials $f_0, f_1 \in \mathbb{Z}_p[R, S, T, U]$ and computes $f_0 + f_1$. If L_1 contains $f_0 + f_1$, $\mathcal{B}$ returns its associated bitstring. Else, $\mathcal{B}$ picks $\sigma \xleftarrow{R} \{0,1\}^{m_1}$, returns σ and stores $(f_0 + f_1, \sigma)$ in L_1.

Inversion in $\mathbb{G}_1$: Given a bitstring σ representing an element in $\mathbb{G}_1$, $\mathcal{B}$ recovers the corresponding values $f \in \mathbb{Z}_p[R, S, T, U]$ and computes $-f$. In case that L_1 already contains $-f$, $\mathcal{B}$ returns its associated bitstring. Otherwise, $\mathcal{B}$ chooses $\sigma' \xleftarrow{R} \{0,1\}^{m_1}$, returns σ' and stores $(-f, \sigma')$ in L_1.

Pairing: Given two bitstrings $\sigma, \hat{\sigma}$ representing elements in $\mathbb{G}_1$ and $\mathbb{G}_2$, $\mathcal{B}$ recovers the corresponding values $f \in \mathbb{Z}_p[R,S,T,U]$ from L_1 and $\hat{f} \in \mathbb{Z}_p[R,S,U,V]$ from L_2. In case that L_T already contains $f \cdot \hat{f} \in \mathbb{Z}_p[R,S,T,U,V]$, $\mathcal{B}$ returns its associated bitstring τ. Otherwise, $\mathcal{B}$ chooses $\tau \xleftarrow{R} \{0,1\}^{m_T}$, returns τ and stores $(f \cdot \hat{f}, \tau)$ in L_T.

The group action and inversion oracle for $\mathbb{G}_2$ and $\mathbb{G}_T$ are simulated analogously to those for $\mathbb{G}_1$. The ones for $\mathbb{G}_2$ consider polynomials from $\mathbb{Z}_p[R,S,U,V]$ stored in list L_2 and the ones for $\mathbb{G}_T$ consider polynomials from $\mathbb{Z}_p[R,S,T,U,V]$ stored in list L_T.

When $\mathcal{A}$ has finished querying the group oracles, $\mathcal{A}$ outputs a bit b^*. Then, $\mathcal{B}$ chooses $r,s,t,u,v \xleftarrow{R} \mathbb{Z}_p$ and sets $R \leftarrow r, S \leftarrow s, T \leftarrow t, U \leftarrow u, V \leftarrow v$.

Now, if the simulation was consistent, no information about b got revealed and hence $\mathcal{A}$ can only guess b with probability $1/2$. Nevertheless, the simulation can be inconsistent, if two distinct polynomials in L_1, L_2 or L_T evaluate to the same value after choosing concrete values for R,S,T,U,V.

We need to prove that such a collision in L_1, L_2 and L_T cannot be caused by $\mathcal{A}$ itself. All monomials except for STU and RU are independent. Since $\mathcal{A}$ is not given direct access to the encoding of U besides being unable to increase the degrees of the polynomials in L_1 and L_2, $\mathcal{A}$ is unable to produce collisions in L_1 and L_2. Observe that the polynomials contained in L_T have total degree at most 5, as they arise from the multiplication of polynomials in L_1 and L_2 and since the group action and the inversion in $\mathbb{G}_T$ do not increase the degree of polynomials in L_T. In particular, L_T can only contain polynomials consisting of total-degree-0 monomials, the total-degree-1 monomials arising from degree-0 polynomials in L_1 and total-degree 1 polynomials in L_2 or vice-versa and the monomials R^2, RS, S^2, RT, ST, $RSTU$, S^2TU, $((1-b)V + b \cdot RU)R$, $((1-b)V + b \cdot RU)S$, $((1-b)V + b \cdot RU)T$ and $((1-b)V + b \cdot RU)STU$. Thus, also in this case $\mathcal{A}$ is not able to generate a collision on purpose.

It remains to be shown that the probability of a collision, where two distinct polynomials in L_1, L_2 or L_T evaluate to the same value after the substitution is negligible (or alternatively that their difference polynomial evaluates to 0). Suppose that $\mathcal{A}$ has issued q queries to the group oracles. Let $|L_1| = O(q)$, $|L_2| = O(q)$ and $|L_T| = O(q)$, then there are $O(\binom{q}{2})$ possibilities of colliding polynomials. By the Schwartz-Zippel lemma and the collision argument, the probability of such an error in the simulation of the generic bilinear group is $O(\frac{q^2}{p})$ and is, thus negligible in the security parameter. □

B.3 Proof of Theorem 4

Proof. We prove the anonymity of Scheme 2 using a sequence of games, where each game is indistinguishable from the previous one. Henceforth, we denote the event that an adversary wins Game i by S_i. In all games, the setup is as in the original game, with the following differences. Upon generation of the public parameter pp—instead of choosing Q at random—one chooses $q \xleftarrow{R} \mathbb{Z}_p^*$ and sets $Q \leftarrow qP$. Then, the environment stores q as well as the trapdoors α and λ used

for generating the tuples $(\alpha^i P)_{i=0}^t, (\alpha^i \hat{P})_{i=0}^t$ and $(\lambda^i P)_{i=0}^t, (\lambda^i \hat{P})_{i=0}^t$ contained in pp.

Game 0: The original anonymity game with $b = 0$.

Game 1: As Game 0, but the PoK in all showings is conducted by proving knowledge of q and simulating the proof part for the remainder. Furthermore, all calls to $\mathsf{ChgRep}_{\mathcal{R}}(M, \sigma, \rho, \mathsf{pk})$ are replaced by $(\rho M, \mathsf{Sign}_{\mathcal{R}}(\rho M, \mathsf{sk}))$.

Transition 1 - Game 0 → Game 1: Since the witness indistinguishability of the OR proof is unconditional and Scheme 3 perfectly adapts signatures, we have that $\Pr[S_1] = \Pr[S_0]$.

Game 2: As Game 1, except for the oracle $\mathcal{O}^{LoR}$, which is simulated as follows:

$\mathcal{O}^{LoR}(\mathsf{osk}, \mathsf{opk}, \mathsf{rsk}, \mathsf{rpk}, b, \mathsf{nym}_0, \mathsf{nym}_1, \mathbb{A}', \mathbb{R}_V)$**:** As in a real game, but the showing is simulated independently of bit b as follows. $\mathcal{S}$ chooses a message $(M_1, M_2, M_3, M_4) \xleftarrow{R} (\mathbb{G}_1^*)^4$, sets the shown credential $\mathsf{cred} \leftarrow ((M_1, M_2, M_3, M_4), \sigma)$, with $\sigma \leftarrow \mathsf{Sign}_{\mathcal{R}}((M_1, M_2, M_3, M_4), \mathsf{osk})$. Furthermore, $\mathcal{S}$ computes $\mathcal{C}_{\overline{\mathbb{A}'}} \leftarrow \frac{1}{\mathsf{enc}(\mathbb{A}')(\alpha)} M_1$ using trapdoor α. Finally, $\mathcal{S}$ picks $\Pi' \xleftarrow{R} \mathbb{G}_1$, $\hat{W}' \xleftarrow{R} \mathbb{G}_2$ and computes $\mathbf{d}' \leftarrow e(\Pi', \hat{P})/e(M_2, \hat{W}')$.

Transition 2 - Game 1 → Game 2: To show that the games are indistinguishable, we have to show that the adversary will not detect that the showings in $\mathcal{O}^{LoR}$ are performed with respect to a random credential. To do so, we consider a subset V of the adversary's view on the system. In particular, in V we consider all values containing discrete logarithms contained in values which are exchanged in the transition between Game 2 and Game 3 (all other values are independent, and, thus, do not give an advantage in the decision).

$$\begin{aligned} V = \big(&P, \hat{P}, Q, \mathsf{enc}(\mathbb{A})(\alpha)P, \mathsf{enc}(\mathbb{A})(\alpha)\hat{P}, \mathsf{enc}(\overline{\mathbb{A}'})(\alpha)P, \mathsf{enc}(\overline{\mathbb{A}'})(\alpha)\hat{P}, \\ &(\lambda - \mathsf{nym})P, (\lambda - \mathsf{nym})\hat{P}, W, \hat{W}, \Pi, \hat{\Pi}, C_4, U_i, R_i, u_i Q, r_i \mathsf{enc}(\mathbb{A})P, \\ &u_i(\lambda - \mathsf{nym})P, \hat{W}', C_3, C_2, C_1, \mathcal{C}_{\overline{\mathbb{A}'}}, \Pi', \mathbf{g}^{\rho\nu u_i}\big). \end{aligned}$$

Note that $\mathbf{d}' = \mathbf{g}^{\rho\nu u_i d} = e(\rho\nu u_i \Pi, \hat{P})/e(C_2, \nu\hat{W})$ is implicitly contained in V. As the adversary knows all potential values for d, and, thus, can obtain candidate values for $\mathbf{g}^{\rho\nu u_i}$, we additionally include $\mathbf{g}^{\rho\nu u_i}$ for the correct guess and show that it is indistinguishable from random. For our illustrations, we further make the discrete logarithms to the bases $P, \hat{P}$ and $\mathbf{g}$ explicit:

$$\begin{aligned} V = \big(&P, \hat{P}, qP, eP, e\hat{P}, \overline{e}P, \overline{e}\hat{P}, \\ &nP, n\hat{P}, wP, w\hat{P}, aP, a\hat{P}, \rho P, u_i P, r_i P, u_i qP, r_i eP, \\ &u_i nP, \nu w\hat{P}, \rho u_i qP, \rho u_i nP, \rho r_i eP, \rho r_i \overline{e}P, \rho\nu u_i aP, \mathbf{g}^{\rho\nu u_i}\big). \end{aligned}$$

Now, to distinguish between Game 2 and Game 3, the adversary is required to distinguish the distributions $\mathcal{D}_1(V)$ and $\mathcal{D}_2(V)$, which are defined as follows:

$$\mathcal{D}_1(V) :\equiv \Big[\big(P, \hat{P}, qP, eP, e\hat{P}, \overline{e}P, \overline{e}\hat{P}, \\ nP, n\hat{P}, wP, w\hat{P}, aP, a\hat{P}, \rho P, u_iP, r_iP, u_iqP, r_ieP, \\ u_inP, \nu w\hat{P}, \rho u_iqP, \rho u_inP, \rho r_ieP, \rho r_i\overline{e}P, \rho\nu u_iaP, \mathbf{g}^{\rho\nu u_i}\big)\Big],$$

$$\mathcal{D}_2(V) :\equiv \Big[\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}, \mathsf{e}, \mathsf{f} \xleftarrow{R} \mathbb{Z}_p, \big(P, \hat{P}, qP, eP, e\hat{P}, \overline{e}P, \overline{e}\hat{P}, \\ nP, n\hat{P}, wP, w\hat{P}, aP, a\hat{P}, \rho P, u_iP, r_iP, u_iqP, r_ieP, \\ u_inP, \mathsf{a}\hat{P}, \mathsf{b}P, \mathsf{c}P, \mathsf{d}P, \mathsf{d}e^{-1}\overline{e}P, \mathsf{e}P, \mathbf{g}^{\mathsf{f}}\big)\Big].$$

As a first step, we can simplify the distributions: The values $e, \overline{e}, n$ are computationally hidden via the DDH instances spanned by u_i, ρ and r_i, ρ, respectively (the simplified distributions can then simply be padded to $\mathcal{D}_1(V)$ and $\mathcal{D}_2(V)$). In particular, we have

$$\mathcal{D}'_1(V) :\equiv \Big[\big(P, \hat{P}, qP, wP, w\hat{P}, aP, a\hat{P}, \rho P, u_iP, r_iP, qu_iP, \\ \nu w\hat{P}, \rho u_iqP, \rho u_iP, \rho r_iP, \rho\nu u_iaP, \mathbf{g}^{\rho\nu u_i}\big)\Big],$$

$$\mathcal{D}'_2(V) :\equiv \Big[\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}, \mathsf{e}, \mathsf{f} \xleftarrow{R} \mathbb{Z}_p, \big(P, \hat{P}, qP, wP, w\hat{P}, aP, a\hat{P}, \rho P, u_iP, r_iP, qu_iP, \\ \mathsf{a}\hat{P}, \mathsf{b}P, \mathsf{c}P, \mathsf{d}P, \mathsf{e}P, \mathbf{g}^{\mathsf{f}}\big)\Big].$$

In addition, we claim that the accumulator function can be interpreted as an algorithm to randomly sample an element from $\mathbb{G}_1$ or $\mathbb{G}_2$, respectively. That is, the values a and w, i.e., the polynomial evaluations of the accumulator and the witnesses at a random λ contained in the t-SDH tuples in pp, are negligibly close to uniform.

Claim 1. *Let t-co-SDH$_i^*$ hold and $\mathcal{I} = (\mathsf{BG}, (\lambda^k P_1)_{k\in[t]}, (\lambda^k P_2)_{k\in[t]})$ be a t-co-SDH$_i^*$ instance (i.e., $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$, $\lambda \xleftarrow{R} \mathbb{Z}_p$ and $t = \mathsf{poly}(\kappa)$). Let the map $g_{\mathcal{I},i} : \mathbb{Z}_p^t \to \mathbb{G}_i$ be defined as $(x_j)_{j\in[t]} \mapsto \prod(\lambda - x_j)_{j\in[t]}P_i$, where $P_1 = P$, $P_2 = \hat{P}$, and $i \in \{1, 2\}$. Then, for every PPT distinguisher the probability to distinguish the distribution ensemble $\{g_{\mathcal{I},i}(X)\}_{X\in\mathbb{Z}_p^t}$ from the uniform distribution $\{U_n\}_{U_n\in\mathbb{G}_1}$ by a polynomial number $s(\kappa)$ of samples is a negligible function in the security parameter κ. That is, the distributions are computationally indistinguishable by multiple samples [Gol08].*

Proof (of Claim 1). To see that Claim 1 holds, assume that $\{g_{\mathcal{I},i}(X)\}_{X\in\mathbb{Z}_p^t}$ is not negligibly close to uniform. In other words, this means that one can sample

X, X' such that $\prod_{x_\ell \in X}(\lambda - x_\ell) = \prod_{x_m \in X'}(\lambda - x_m)$ in PPT with non-negligible probability. Solving the equation with respect to λ allows to output arbitrary t-co-SDH$_i^*$ solutions with non-negligible probability. □

Now, we can further simplify $\mathcal{D}'_1(V)$ and $\mathcal{D}'_2(V)$ to $\mathcal{D}'_1$ and $\mathcal{D}'_2$:

$$\mathcal{D}'_1 :\equiv \Big[a, b, c, d, e, f, g \xleftarrow{R} \mathbb{Z}_p,$$
$$\big(P, \hat{P}, aP, bP, b\hat{P}, cP, c\hat{P}, dP, eP, fP, aeP, bg\hat{P}, aedP, edP, dfP, dgecP, \mathbf{g}^{dge}\big)\Big],$$
$$\mathcal{D}'_2 :\equiv \Big[a, b, c, d, e, f, \mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}, \mathsf{e}, \mathsf{f} \xleftarrow{R} \mathbb{Z}_p,$$
$$\big(P, \hat{P}, aP, bP, b\hat{P}, cP, c\hat{P}, dP, eP, fP, aeP, \mathsf{a}\hat{P}, \mathsf{b}P, \mathsf{c}P, \mathsf{d}P, \mathsf{e}P, \mathbf{g}^{\mathsf{f}}\big)\Big].$$

What remains is to prove that $\mathcal{D}'_1$ and $\mathcal{D}'_2$ are indistinguishable:

Lemma 1. *If the DDH assumption holds in $\mathbb{G}_1$ and the assumptions in Definition 11 and 12 hold, then for every PPT adversary $\mathcal{A}$, there is a negligible function $\epsilon_{\mathcal{D}}(\cdot)$ such that the probability to distinguish $\mathcal{D}'_1$ from $\mathcal{D}'_2$ is bounded by $\epsilon_{\mathcal{D}'}(\kappa)$.*

Proof (of Lemma 1). We show that—under the assumptions in Definition 11 and 12 and under the DDH assumption in $\mathbb{G}_1$—the probability to distinguish $\mathcal{D}'_1$ and $\mathcal{D}'_2$ is bounded by a negligible function in the security parameter κ, i.e., $\epsilon_{\mathcal{D}}(\kappa)$. We do so by a sequence of intermediate distributions. We start by introducing $\mathcal{D}'_3$:

$$\mathcal{D}'_3 :\equiv \Big[a, b, c, d, e, f, g, \mathsf{f} \xleftarrow{R} \mathbb{Z}_p,$$
$$\big(P, \hat{P}, aP, bP, b\hat{P}, cP, c\hat{P}, dP, eP, fP, aeP, bg\hat{P}, aedP, edP, dfP, dgecP, \mathbf{g}^{\mathsf{f}}\big)\Big].$$

$\mathcal{D}'_1$ and $\mathcal{D}'_3$ are indistinguishable under the assumption in Definition 11. Assume an instance $(P, \hat{P}, rP, r\hat{P}, sP, s\hat{P}, tP, ru\hat{P}, stuP, \mathbf{g}^{\mathsf{r}}) = (P, \hat{P}, bP, b\hat{P}, cP, c\hat{P}, dP, bg\hat{P}, gdcP, \mathbf{g}^{\mathsf{r}})$. We can choose $a, e, f \xleftarrow{R} \mathbb{Z}_p$ and obtain $(P, \hat{P}, aP, bP, b\hat{P}, cP, c\hat{P}, dP, eP, fP, aeP, bg\hat{P}, aedP, edP, dfP, dgecP, \mathbf{g}^{\mathsf{r}e})$. Then, if $\mathsf{r} = dg$, we obtain a distribution identical to $\mathcal{D}'_1$, whereas we obtain distribution identical to $\mathcal{D}'_3$ if r is random. We introduce $\mathcal{D}'_4$:

$$\mathcal{D}'_4 :\equiv \Big[a, b, c, d, e, f, g, \mathsf{a}, \mathsf{f} \xleftarrow{R} \mathbb{Z}_p,$$
$$\big(P, \hat{P}, aP, bP, b\hat{P}, cP, c\hat{P}, dP, eP, fP, aeP, \mathsf{a}\hat{P}, aedP, edP, dfP, dgecP, \mathbf{g}^{\mathsf{f}}\big)\Big].$$

$\mathcal{D}'_3$ is indistinguishable from $\mathcal{D}'_4$ under the assumption in Definition 12. Assume a corresponding instance $(P, \hat{P}, rP, r\hat{P}, sP, s\hat{P}, tP, stuP, \mathsf{r}\hat{P}) = (P, \hat{P}, bP, b\hat{P}, cP, c\hat{P}, eP, gecP, \mathsf{r}\hat{P})$. We can choose $a, d, f, \mathsf{f} \xleftarrow{R} \mathbb{Z}_p$ and obtain $(P, \hat{P}, aP, bP, b\hat{P}, cP,$

$c\hat{P}, dP, eP, fP, aeP, \mathsf{r}\hat{P}, aedP, edP, dfP, dgecP, \mathbf{g}^{\mathsf{f}}$). If $\mathsf{r} = bg$ we obtain a distribution identical to $\mathcal{D}'_3$, whereas we obtain a distribution identical to $\mathcal{D}'_4$ if r is random. We introduce $\mathcal{D}'_5$:

$$\mathcal{D}'_5 :\equiv \Big[a, b, c, d, e, f, g, \mathsf{a}, \mathsf{e}, \mathsf{f} \xleftarrow{R} \mathbb{Z}_p, \\ \big(P, \hat{P}, aP, bP, b\hat{P}, cP, c\hat{P}, dP, eP, fP, aeP, \mathsf{a}\hat{P}, aedP, edP, dfP, \mathsf{e}P, \mathbf{g}^{\mathsf{f}}\big)\Big].$$

It is easy to see that $\mathcal{D}'_4$ and $\mathcal{D}'_5$ are identically distributed, since g only occurs in $dgecP$. This means that $dgecP$ already looks random in $\mathcal{D}'_4$. Hence, in $\mathcal{D}'_5$ we can substitute it by $\mathsf{e} \xleftarrow{R} \mathbb{Z}_p$. We introduce $\mathcal{D}'_6$:

$$\mathcal{D}'_6 :\equiv \Big[a, b, c, d, e, f, g, \mathsf{a}, \mathsf{e}, \mathsf{f}, \mathsf{g} \xleftarrow{R} \mathbb{Z}_p, \\ \big(P, \hat{P}, aP, bP, b\hat{P}, cP, c\hat{P}, dP, eP, fP, aeP, \mathsf{a}\hat{P}, a\mathsf{g}P, \mathsf{g}P, dfP, \mathsf{e}P, \mathbf{g}^{\mathsf{f}}\big)\Big].$$

Under DDH, $\mathcal{D}_5$ is indistinguishable from $\mathcal{D}_6$. Assume a DDH instance $(P, rP, sP, \mathsf{r}P) = (P, dP, eP, \mathsf{r}P)$. We can choose $a, b, c, f, g, \mathsf{a}, \mathsf{e}, \mathsf{f} \xleftarrow{R} \mathbb{Z}_p$ and obtain $(P, \hat{P}, aP, bP, b\hat{P}, cP, c\hat{P}, dP, eP, fP, aeP, \mathsf{a}\hat{P}, a\mathsf{r}P, \mathsf{r}P, dfP, \mathsf{e}P, \mathbf{g}^{\mathsf{f}})$. Then, if $\mathsf{r} = de$, we have a distribution as in $\mathcal{D}_5$, whereas we have a distribution as in $\mathcal{D}_6$ if r is random. Now, we introduce $\mathcal{D}_7$:

$$\mathcal{D}'_7 :\equiv \Big[a, b, c, d, e, f, g, \mathsf{a}, \mathsf{b}, \mathsf{e}, \mathsf{f}, \mathsf{g} \xleftarrow{R} \mathbb{Z}_p, \\ \big(P, \hat{P}, aP, bP, b\hat{P}, cP, c\hat{P}, dP, eP, fP, aeP, \mathsf{a}\hat{P}, \mathsf{b}P, \mathsf{g}P, dfP, \mathsf{e}P, \mathbf{g}^{\mathsf{f}}\big)\Big].$$

Under DDH, $\mathcal{D}'_6$ is indistinguishable from $\mathcal{D}'_7$. Assume a DDH instance $(P, rP, sP, \mathsf{r}P) = (P, \mathsf{g}P, aP, \mathsf{r}P)$. We can choose $b, c, d, e, f, g, \mathsf{a}, \mathsf{e}, \mathsf{f}, \mathsf{g} \xleftarrow{R} \mathbb{Z}_p$ and obtain $(P, \hat{P}, aP, bP, b\hat{P}, cP, c\hat{P}, dP, eP, fP, aeP, \mathsf{a}\hat{P}, \mathsf{r}P, \mathsf{g}P, dfP, \mathsf{e}P, \mathbf{g}^{\mathsf{f}})$. Then, if $\mathsf{r} = a\mathsf{g}$ we have a distribution as in $\mathcal{D}'_6$, whereas we have a distribution as in $\mathcal{D}'_7$ if r is random. Below, we introduce $\mathcal{D}'_8$:

$$\mathcal{D}'_8 :\equiv \Big[a, b, c, d, e, f, g, \mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{e}, \mathsf{f} \xleftarrow{R} \mathbb{Z}_p, \\ \big(P, \hat{P}, aP, bP, b\hat{P}, cP, c\hat{P}, dP, eP, fP, aeP, \mathsf{a}\hat{P}, \mathsf{b}P, \mathsf{c}P, dfP, \mathsf{e}P, \mathbf{g}^{\mathsf{f}}\big)\Big].$$

It is immediate that $\mathcal{D}'_7$ and $\mathcal{D}'_8$ are identically distributed. Finally, under DDH in $\mathbb{G}_1$, $\mathcal{D}'_8$ is indistinguishable from $\mathcal{D}'_2$. Assume a DDH instance $(P, rP, sP, \mathsf{r}P) = (P, dP, fP, \mathsf{r}P)$. We can choose $a, b, c, e, \mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{e}, \mathsf{f} \xleftarrow{R} \mathbb{Z}_p$ and obtain $(P, \hat{P}, aP, bP, b\hat{P}, cP, c\hat{P}, dP, eP, fP, aeP, \mathsf{a}\hat{P}, \mathsf{b}P, \mathsf{c}P, \mathsf{r}P, \mathsf{e}P, \mathbf{g}^{\mathsf{f}})$. Then, if $\mathsf{r} = df$ we have a distribution as in $\mathcal{D}'_8$, whereas we have a distribution as in $\mathcal{D}'_2$ if r is random.

Taking all together, the advantage $\epsilon_{\mathcal{D}}(\kappa)$ of a distinguisher between $\mathcal{D}_1$ and $\mathcal{D}_2$ is bounded by $\epsilon_{\mathcal{D}}(\kappa) \leq 3 \cdot \epsilon_{DDH}(\kappa) + \epsilon_{\text{Def. 11}}(\kappa) + \epsilon_{\text{Def. 12}}(\kappa)$ and a distinguisher between $\mathcal{D}'_1$ and $\mathcal{D}'_2$ implies a distinguisher for one of the intermediate distributions, which proves Lemma 1. □

By Claim 1 and Lemma 1, we know that under t-co-SDH$_i^*$, the probability to distinguish Game 1 and Game 2 is bounded by $|\Pr[S_2] - \Pr[S_3]| \leq \epsilon_{\mathcal{D}}(\kappa)$.

Game 3: As Game 2, but we set $b = 1$.

Transition - Game 2 → Game 3: In Game 2, all values are independent of b, meaning that flipping b does not influence the distributions, i.e., $\Pr[S_2] = \Pr[S_3]$.

Game 4: As Game 3, but we simulate the $\mathcal{O}^{LoR}$ oracle as in the real game.

Transition - Game 3 → Game 4: Under the same argumentation as in Transition 2, we know that under t-co-SDH$_i^*$ the probability to distinguish Game 3 and Game 4 is bounded by $|\Pr[S_3] - \Pr[S_4]| \leq \epsilon_{\mathcal{D}}(\kappa)$.

Game 5: As Game 4, but we honestly compute the OR proof PoK with respect to $C_3, C_4, \mathbf{d}', \Pi'$ and replace all calls to $(\rho M, \mathsf{Sign}_{\mathcal{R}}(\rho M, \mathsf{sk}))$ by $\mathsf{ChgRep}_{\mathcal{R}}(M, \sigma, \rho, \mathsf{pk})$.

Transition - Game 4 - Game 5: Under the same argumentation as in Transition 1, we know that $\Pr[S_4] = \Pr[S_5]$.

Game 0 represents the anonymity game with $b = 0$, whereas Game 5 represents the anonymity game with $b = 1$; both games are computationally indistinguishable. □

C RABC Based on U-Prove's Revocation Approach

The traditional paradigm for accumulator-based credential revocation requires a typically rather complex ZKPK. Thereby, one firstly has to show that an identifier nym encoded in the credential is identical to the value contained in a non-membership witness of a universal accumulator scheme without revealing the respective witness. This requires a ZKPK of a non-membership witness to some nym and a proof of equality of this nym with the one contained in the shown credential. Secondly, it requires to prove in zero-knowledge that this witness satisfies the accumulator verification relation to demonstrate that nym has not been revoked.

Now, to incorporate such a revocation approach into the ABC system of [HS], the following modifications are required (see Scheme 4 for the resulting RABC system). During $(\mathsf{Obtain}, \mathsf{Issue})$, the obtainer of the credential additionally has to provide the issuer with a Pedersen commitment $C_2 \leftarrow \mathsf{nym} \cdot P + \upsilon Q$ (with $\upsilon \xleftarrow{R} \mathbb{Z}_p^*$) to nym to the issuer, and to conduct a ZKPK that the claimed nym is equal to the nym contained in C_2. On successful completion, the value C_2 and a random element $T \in \mathbb{G}_1$ are included into the credential. Furthermore, the public parameters are augmented with two independent generators $M, N \in \mathbb{G}_1$. Then, upon $(\mathsf{Show}, \mathsf{Verify})$, the following equality proof, which relates the nym contained in both C_2 (of the randomized credential) and an auxiliary commitment $D \leftarrow \mathsf{nym} \cdot M + t_0 N$ (with $t_0 \xleftarrow{R} \mathbb{Z}_p^*$), has to be conducted. Due to the randomization that happens during each showing, proving this equality essentially requires a proof of the following quadratic relation (cf. [BS02]) and as in our first construction we additionally require a credential component C_3 where

we need to prove knowledge of $\log_T C_3$ for technical reasons:

$$\mathsf{PoK}\left\{(\alpha,\beta,\gamma,\delta,\epsilon,\zeta,\tau): \begin{array}{rcrcr} C_3 = \tau T & \wedge & C_4 = \gamma P & \wedge \\ D = \alpha M + \beta N & \wedge \\ \gamma D = \delta M + \epsilon N & \wedge & C_2 = \delta P + \zeta Q \end{array}\right\}. \quad (1)$$

It remains to describe the second part. Up to now, the most efficient universal accumulator—suited for our setting—is an optimized version of Acc in Scheme 1, which was used in [ACN13, NP14] as a revocation mechanism for U-Prove credentials. Here, also the W-part of the non-membership witness is computed in $\mathbb{G}_1$ and a non-membership witness ω_{nym} with respect to an accumulator $\Pi_{\mathtt{RNYM}}$ for some $\mathsf{nym} \in \mathtt{NYM}$ is extended by a third element $V \leftarrow \Pi_{\mathtt{RNYM}} + \mathsf{nym} \cdot W - dP$. This value is chosen such that $\lambda W = V$ holds (observe that $(\lambda - \mathsf{nym})W = \Pi_{\mathtt{RNYM}} - dP$ and, hence, $\lambda W = \Pi_{\mathtt{RNYM}} + \mathsf{nym} \cdot W - dP = V$ holds). Since this value can be computed using only public information, this modification does not influence the collision freeness of the accumulator scheme. Then, verifying non-membership claims amounts to checking whether the following two equations hold:

$$V - \Pi_{\mathtt{RNYM}} = \mathsf{nym} \cdot W - dP \quad (2) \qquad \lambda W = V. \quad (3)$$

Based on this modification, a ZKPK of non-membership for some $\mathsf{nym} \in \mathtt{NYM}$ looks as follows. For our illustrations, we assume that the proof in Equation (1) was successful and that D is used as input for the proof in Equation (4). Furthermore, we assume that a commitment to the accumulator secret key λN is contained in the public key of the revocation authority. Then, the original values of W and V are blinded by choosing $t_1 \xleftarrow{R} \mathbb{Z}_p$ and computing $\mathcal{W} \leftarrow W + t_1 N$ and $\mathcal{V} \leftarrow V + t_1(\lambda N)$. Finally, with $t_1, t_2, t_3 \xleftarrow{R} \mathbb{Z}_p$, the auxiliary commitments $R \leftarrow t_1 M + t_2 N$ and $S \leftarrow d^{-1} M + t_3 N$ are computed and the following proof is performed:

$$\mathsf{PoK}\left\{\begin{pmatrix}\alpha,\beta,\theta,\iota,\kappa,\\ \lambda,\mu,\nu,\xi,\pi\end{pmatrix}: \begin{array}{rcrcr} D = \alpha M + \beta N & \wedge & R = \theta M + \iota N & \wedge \\ \alpha R = \kappa M + \lambda N & \wedge & S = \mu M + \nu N & \wedge \\ & & \pi S = M + \xi N & \wedge \\ \mathcal{V} - \Pi_X = -\alpha \mathcal{W} + \kappa N + \theta(\lambda N) - \pi P \end{array}\right\}. \quad (4)$$

Thereby, the last AND clause corresponds to a blinded version of Equation (2), whereas the remaining clauses are required for proving that the nym encoded in D corresponds to the non-membership witness used in the proof. It also ensures that the non-membership witness is formed correctly, i.e., $d \neq 0$. In addition to the PoK, Equation (3) must be checked to hold by means of checking $e(\mathcal{W}, \lambda \hat{P}) \stackrel{?}{=} e(\mathcal{V}, \hat{P})$ (as done in [ACN13]).

Finally, we note that the importance of the authenticity of the revocation information in certain variants of U-Prove's revocation approach was quite recently pointed out in [HKK15]. There it is shown how to trace U-Prove tokens via non-authentic revocation information, if the revocation information can not be publicly verified and the revocation authority is dishonest. We emphasize that these results do not apply to our approach, since every revocation-related

Setup: Given $(1^\kappa, \mathsf{aux})$, parse $\mathsf{aux} \leftarrow (t, t')$, run $\mathsf{pp}' = (\mathsf{BG}, (\alpha^i P)_{i\in[t]}, (\alpha^i \hat{P})_{i\in[t]}) \leftarrow \mathsf{Setup}_{\mathsf{PC}}(1^\kappa, t)$. Then, let $H_s : \{0,1\}^* \to \mathbb{Z}_p^*$ be a collision-resistant keyed hash function used inside $\mathsf{enc}(\cdot)$, which is drawn uniformly at random from a family of collision-resistant keyed hash functions $\{(H_s, s)\}_{s\in S}$. Finally, choose $Q, T \xleftarrow{R} \mathbb{G}_1$ and output $\mathsf{pp} \leftarrow (H_s, \mathsf{enc}, Q, T, \mathsf{pp}', t')$.

RAKeyGen: Given pp, choose $\lambda \xleftarrow{R} \mathbb{Z}_p^*$, compute $\mathsf{pk}_\Pi = ((\lambda^i P)_{i\in[t']}, (\lambda^i \hat{P})_{i\in[t']})$ and choose $M, N \xleftarrow{R} \mathbb{G}_1$. Set $(\mathsf{rsk}, \mathsf{rpk}) \leftarrow (\lambda, (\mathsf{pk}_\Pi, M, N, \lambda N))$ and return $(\mathsf{rsk}, \mathsf{rpk})$.

OrgKeyGen: Given pp, return $(\mathsf{osk}, \mathsf{opk}) \leftarrow \mathsf{KeyGen}_{\mathcal{R}}(\mathsf{BG}, \ell = 4)$.

UserKeyGen: Given pp, pick $r \xleftarrow{R} \mathbb{Z}_p^*$ and return $(\mathsf{usk}, \mathsf{upk}) \leftarrow (r, R)$ with $R \leftarrow rP$.

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

$\mathsf{Issue}(\mathsf{pp}, \mathsf{rpk}, \mathsf{upk}_i, \mathsf{osk}_j, \mathsf{nym}, \mathbb{A})$		$\mathsf{Obtain}(\mathsf{pp}, \mathsf{usk}_i, \mathsf{opk}_j, \mathsf{nym}, \mathbb{A})$
$e(C_1, \hat{P}) \stackrel{?}{=} e(R_i, \mathsf{enc}(\mathbb{A})(\alpha)\hat{P})$		$\upsilon \xleftarrow{R} \mathbb{Z}_p^*,\ (C_1, C_2) \leftarrow$
	$\xleftarrow{C_1, C_2}$	$(r_i \mathsf{enc}(\mathbb{A})(\alpha)P, \mathsf{nym} \cdot P + \upsilon Q)$
	$\xleftrightarrow{\mathsf{PoK}}$	
$\sigma \leftarrow \mathsf{Sign}_{\mathcal{R}}((C_1, C_2, T, P), \mathsf{osk}_j)$	$\xrightarrow{\sigma}$	$\mathsf{Verify}_{\mathcal{R}}((C_1, C_2, T, P), \sigma, \mathsf{opk}_j) \stackrel{?}{=} 1$
		$\mathsf{cred}_{\mathsf{nym}} \leftarrow ((C_1, C_2, T, P), \sigma)$

where PoK is: $\mathsf{PoK}\{(\alpha) : C_2 = \mathsf{nym} \cdot P + \alpha Q\}$.

(Show, Verify): Show and Verify interact in the following way, where $\mathbb{R}_V = \Pi \leftarrow \mathbb{R}[1]$ and $\mathbb{R}_S^{\mathsf{nym}} = (\Pi, (W, V, d)) \leftarrow (\mathbb{R}[1], \mathbb{R}[2][\mathsf{nym}])$:

$\mathsf{Verify}(\mathsf{pp}, \mathsf{rpk}, \mathsf{opk}_j, \mathbb{A}', \mathbb{R}_V)$		$\mathsf{Show}(\mathsf{pp}, \mathsf{rpk}, \mathsf{usk}_i, \mathsf{opk}_j, \mathsf{cred}_{\mathsf{nym}}, \mathbb{A}, \mathbb{A}', \mathbb{R}_S^{\mathsf{nym}})$
		$\mathsf{cred} \leftarrow \mathsf{ChgRep}_{\mathcal{R}}(\mathsf{cred}_{\mathsf{nym}}, \rho, \mathsf{opk}_j)$
		$\mathcal{C}_{\overline{\mathbb{A}'}} \leftarrow (\rho \cdot r_i) \cdot \mathsf{enc}(\overline{\mathbb{A}'})(\alpha)P$
		$t_0, t_1, t_2, t_3 \xleftarrow{R} \mathbb{Z}_p,\ \mathcal{V} \leftarrow V + t_1(\lambda N),$
		$\mathcal{W} \leftarrow W + t_1 N,\ D \leftarrow \mathsf{nym} \cdot M + r^* N$
		$R \leftarrow t_1 M + t_2 N,$
$\Big[\mathsf{Verify}_{\mathcal{R}}(\mathsf{cred}, \mathsf{opk}_j)\ \wedge$	$\xleftarrow{\mathsf{cred}, \mathcal{C}_{\overline{\mathbb{A}'}}, \mathcal{W}, \mathcal{V}, D, R, S}$	$S \leftarrow d^{-1} M + t_3 N$
$\mathsf{VerifyFactor}_{\mathsf{PC}}(\mathsf{pp}', C_1, \mathsf{enc}(\mathbb{A}'), \mathcal{C}_{\overline{\mathbb{A}'}})$		
$\wedge\ e(\mathcal{W}, \lambda\hat{P}) \stackrel{?}{=} e(\mathcal{V}, \hat{P})\Big] \stackrel{?}{=} 1$	$\xleftrightarrow{\mathsf{PoK}}$	

where $\mathsf{cred} = ((C_1, C_2, C_3, C_4), \sigma)$ and PoK is:

$$\mathsf{PoK}\left\{ \begin{pmatrix} \alpha, \beta, \gamma, \delta, \epsilon, \\ \zeta, \eta, \theta, \iota, \kappa, \lambda, \\ \mu, \nu, \xi, \pi, \tau \end{pmatrix} : \begin{array}{r} Q = \eta P \ \vee\ (C_3 = \tau T \ \wedge\ C_4 = \gamma P \ \wedge \\ D = \alpha M + \beta N \ \wedge\ \gamma D = \delta M + \epsilon N \ \wedge \\ C_2 = \delta P + \zeta Q \ \wedge\ R = \theta M + \iota N \ \wedge \\ \alpha R = \kappa M + \lambda N \ \wedge\ S = \mu M + \nu N \ \wedge \\ \pi S = M + \xi N \ \wedge \\ \mathcal{V} - \Pi_X = -\alpha\mathcal{W} + \kappa N + \theta(\lambda N) - \pi P) \end{array} \right\}.$$

Revoke: Given pp, $(\mathsf{rsk}, \mathsf{rpk})$, $\mathtt{NYM}$ and $\mathtt{RNYM}$, this algorithm computes $\Pi \leftarrow \mathsf{Eval}_{\mathsf{Acc}}(\mathtt{RNYM}, (\mathsf{sk}_\Pi, \mathsf{pk}_\Pi))$. For all $\mathsf{nym} \in \mathtt{NYM}$ it computes $(W_{\mathsf{nym}}, d_{\mathsf{nym}}) \leftarrow \mathsf{WitCreate}_{\mathsf{Acc}}(\Pi, \mathtt{RNYM}, \mathsf{nym}, (\mathsf{sk}_\Pi, \mathsf{pk}_\Pi))$, $V_{\mathsf{nym}} \leftarrow \Pi + \mathsf{nym} \cdot W - dP$ and sets $\mathtt{WIT}[\mathsf{nym}] \leftarrow (W_{\mathsf{nym}}, V_{\mathsf{nym}}, d_{\mathsf{nym}})$. Note that W_{nym} is evaluated in $\mathbb{G}_1$ here. Finally, it returns $\mathbb{R} \leftarrow (\Pi, \mathtt{WIT})$.

Scheme 4: Multi-Show RABC System Adapting U-Prove Revocation

computation can be publicly verified and our model requires honestly generated revocation keys. Another recent work [HKK14] describes an attack on the instantiation of the revocation-related ZKPKs for U-Prove proposed in [ACN13]. But we note that this attack builds upon a wrong interpretation due to inconsistencies in the notation of [ACN13] and thus has no practical relevance.

C.1 Security of the RABC System

Theorem 5. *The RABC system in Scheme 4 is correct.*

The correctness of Scheme 4 follows from inspection.

Theorem 6. *If* PolyCommitFO *is factor-sound,* $\{(H_s, s)\}_{s\in S}$ *is a collision-resistant hash function family, the underlying SPS-EQ-*$\mathcal{R}$ *is* EUF-CMA *secure and perfectly adapts signatures,* Acc *is collision-free and the DDH assumption holds in* $\mathbb{G}_1$*, then Scheme 4 is unforgeable.*

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 $\mathsf{nym}^* = \perp$ 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 credential of user i^* under nym^* with respect to $\mathbb{A}'^*$ such that $\mathbb{A}'^* \not\sqsubseteq \mathtt{ATTR}[\mathsf{nym}^*]$ holds. Then, we construct an adversary $\mathcal{B}$ that uses $\mathcal{A}$ to break

- **Type 2A:** collision-resistance of the hash function used in the encoding $\mathsf{enc}(\cdot)$ of attributes.
- **Type 2B:** the factor soundness of PolyCommitFO.

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

- **Type 3A:** we construct an adversary $\mathcal{B}$ that uses $\mathcal{A}$ to break the DLP in $\mathbb{G}_1$ (with respect to Q).
- **Type 3B:** we show that the success probability of $\mathcal{A}$ is bounded by $\epsilon_{\mathrm{DDH}}(\kappa) + \epsilon_{\mathrm{DL}}(\kappa)$, where $\epsilon_{\mathrm{DDH}}(\kappa)$ and $\epsilon_{\mathrm{DL}}(\kappa)$ are the success probabilities for DDH and DLP in $\mathbb{G}_1$.

Type 4: Adversary $\mathcal{A}$ manages to conduct a showing protocol accepted by the verifier using some credential corresponding to a revoked pseudonym $\mathsf{nym}^* \in \mathtt{RN}$. Then, we construct an adversary $\mathcal{B}$ that uses $\mathcal{A}$ to break the collision-freeness of the accumulator scheme Acc.

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. For the PoK, we assume that the reduction always aborts if the respective discrete logarithm cannot be extracted because the wrong part of the OR statement was honestly computed. In Type 3B we make the abort probability explicit, whereas it is omitted in the other cases.

Type 1: Analogously to the Type 1 unforgeability proof of Scheme 2.

Type 2: Analogously to the Type 2 unforgeability proofs of Scheme 2.

Type 3A: Analogously to the Type 3A unforgeability proof of Scheme 2.

Type 3B: In the following, we will show that the success probability of a Type-3B adversary is bounded by $\epsilon_{\mathrm{DDH}}(\kappa) + \epsilon_{\mathrm{DL}}(\kappa)$. In all games, the setup is as in the original game, with the following differences. Upon generation of the public parameter pp—instead of choosing Q at random—one chooses $q, t \xleftarrow{R} \mathbb{Z}_p^*$ and sets $Q \leftarrow qP, T \leftarrow tP$. Then, the environment stores q, t as well as the trapdoors α and λ used for generating the tuples $(\alpha^i P)_{i=0}^t, (\alpha^i \hat{P})_{i=0}^t$ and $(\lambda^i P)_{i=0}^t, (\lambda^i \hat{P})_{i=0}^t$ contained in pp.

Game 0: The original unforgeability game.

Game 1: As Game 0, but the PoK in all showings is conducted by honestly proving knowledge of q and simulating the proof part for the remainder.

Transition 1 - Game 0 → Game 1: Since the witness indistinguishability of the OR proof is unconditional, we have that $\Pr[S_1] = \Pr[S_0]$.

Game 2: As Game 1, except that all calls to $\mathsf{ChgRep}_{\mathcal{R}}(M, \sigma, \rho, \mathsf{pk})$ are replaced by $(\rho M, \mathsf{Sign}_{\mathcal{R}}(\rho M, \mathsf{sk}))$.

Transition 2 - Game 1 → Game 2: Since Scheme 3 perfectly adapts signatures, we have $\Pr[S_2] = \Pr[S_1]$.

Game 3: As Game 2, but if $\mathcal{A}$ delivers a valid showing using a credential cred^*, then $\mathcal{B}$ rewinds $\mathcal{A}$ to the step after sending the commitments in PoK and restarts $\mathcal{A}$ with a new challenge $c' \neq c$. Then, by the knowledge extractor of PoK (for the C_4-part of the proof), $\mathcal{B}$ obtains $\rho^* \in \mathbb{Z}_p^*$ and can find out the pseudonym nym^* of cred'^* via computing $\mathsf{cred}_0^* \leftarrow \rho^{-1}\mathsf{cred}'^*[0]$ on the message part of the credential. Let F denote the event that there is no $\mathsf{cred}' \in \mathtt{CRED}$ such that $\mathsf{cred}'[0] = \mathsf{cred}_0^*$ or if $i^*_{\mathsf{nym}^*} \notin \mathtt{HU} \setminus \mathtt{KU}$. If F occurs, then $\mathcal{B}$ aborts.

Transition 3 - Game 2 → Game 3: Game 3 is equivalent to Game 2, unless abort event F happens. Event F occurs if and only if $\mathcal{A}$ is no Type-3B adversary, i.e., $\Pr[F] = 6/7$. Thus, $\Pr[S_3] = \Pr[\neg F] \cdot \Pr[S_2] = (1 - \Pr[F]) \cdot \Pr[S_2] = 1/7 \cdot \Pr[S_2]$.

Game 4: As in Game 3, but $\mathcal{B}$ obtains the instance (BG, aP) with $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$ to the DLP in $\mathbb{G}_1$ and generates the public parameters pp based on BG. Furthermore, $\mathcal{B}$ simulates the oracles as in a real game, except for the oracle $\mathcal{O}^{U_V}$, which is simulated as follows:

$\mathcal{O}^{U_V}(\mathsf{opk}, \mathsf{nym}, \mathbb{A}', \mathbb{R}_V)$**:** $\mathcal{B}$ runs this oracle as in a real game, with the difference that $\mathcal{B}$ computes a credential $\mathsf{cred} \leftarrow (M, \sigma)$ with $M \leftarrow (\rho \cdot \mathtt{USK}[i_{\mathsf{nym}}][0] \cdot \mathsf{enc}(\mathtt{ATTR}[\mathsf{nym}])P, \mathsf{nym} \cdot P + \upsilon Q, \rho t \cdot aP, \rho \cdot P)$, with $\rho \xleftarrow{R} \mathbb{Z}_p$ and $\sigma \leftarrow \mathsf{Sign}_{\mathcal{R}}(M, \mathsf{osk})$. The showing is then performed with respect to cred.

Transition 4 - Game 3 → Game 4: We have to show that the adversary cannot detect that the showings in $\mathcal{O}^{U_\mathsf{V}}$ are performed with respect to a different credential component C_3. To do so, we define the following two distributions containing the exchanged value C_3 and all other values containing discrete logarithms related to ρ and q (since we will base our indistinguishability proof on these values). Thereby, $\mathcal{D}_1$ resembles the distribution as in a real oracle simulation, whereas $\mathcal{D}_2$ resembles the distribution after exchanging the C_3 component of the credential. Both distributions are defined with respect to the adversary's view V on the system.

$$\mathcal{D}_1(V) :\equiv \Big[\big(\mathsf{enc}(\mathbb{A})r_i \cdot \rho P, \mathsf{nym} \cdot \rho P + \upsilon q \cdot \rho P, \rho t P, \rho P, tP, \mathsf{enc}(\overline{\mathbb{A}'})(\alpha) r_i \cdot \rho P\big)\Big] \approx$$

$$\mathcal{D}_2(V) :\equiv \Big[\big(\mathsf{enc}(\mathbb{A})r_i \cdot \rho P, \mathsf{nym} \cdot \rho P + \upsilon q \cdot \rho P, a \cdot \rho t P, \rho P, tP, \mathsf{enc}(\overline{\mathbb{A}'})(\alpha) r_i \cdot \rho P\big)\Big].$$

To see that $\mathcal{D}_1(V)$ and $\mathcal{D}_2(V)$ are indistinguishable, we introduce an intermediate distribution $\mathcal{D}_3$:

$$\mathcal{D}_3(V) :\equiv \Big[\mathsf{b} \xleftarrow{R} \mathbb{Z}_p, \big(\mathsf{enc}(\mathbb{A})r_i \cdot \rho P, \mathsf{nym} \cdot \rho P + \upsilon q \cdot \rho P, \mathsf{b}P, \rho P, tP, \mathsf{enc}(\overline{\mathbb{A}'})(\alpha) r_i \cdot \rho P\big)\Big].$$

Now, assume a DDH instance $(P, \rho P, tP, \mathsf{r}P)$ and observe that this instance can be padded to the distributions in $\mathcal{D}_1$ and $\mathcal{D}_3$ since we know all required discrete logarithms. Then, we have a distribution as in $\mathcal{D}_1(V)$ if $\mathsf{r} = \rho t$, whereas we have a distribution as in $\mathcal{D}_3(V)$ if r is random.

Furthermore, $\mathcal{D}_3$ and $\mathcal{D}_2$ are identically distributed, since a is only contained in $a \cdot \rho t P$ and (BG, aP) is a random DLP instance. All in all, we have $|\Pr[S_3] - \Pr[S_4]| \leq \epsilon_{\mathrm{DDH}}(\kappa)$.

Game 5: As Game 4, but $\mathcal{B}$ additionally obtains $\rho^* a$ by the knowledge extractor of PoK (for the C_3 part of the credential).

Transition 5 - Game 4 → Game 5: Since this change is only conceptual, we have that $\Pr[S_4] = \Pr[S_5]$.

In Game 5, $\mathcal{B}$ can compute $a \leftarrow \frac{\rho^* a}{\rho^*}$ as a solution to the given DLP instance in $\mathbb{G}_1$, i.e., $\Pr[S_5] = \epsilon_{\mathrm{DL}}(\kappa)$. We have $\Pr[S_0] = \Pr[S_1] = \Pr[S_2] = \frac{\Pr[S_3]}{\Pr[\neg F]}$. Furthermore, we have that $|\Pr[S_3] - \Pr[S_4]| \leq \epsilon_{\mathrm{DDH}}(\kappa)$, yielding $\Pr[S_3] \leq \Pr[S_4] + \epsilon_{\mathrm{DDH}}(\kappa)$ and $\Pr[S_4] = \Pr[S_5] = \epsilon_{\mathrm{DL}}(\kappa)$. Taking all together, we have $\Pr[\neg F] \cdot \Pr[S_0] = \Pr[S_3] \leq \epsilon_{\mathrm{DDH}}(\kappa) + \epsilon_{\mathrm{DL}}(\kappa)$ and thus $\Pr[S_0] \leq {}^1\!/_{\Pr[\neg F]} \cdot (\epsilon_{\mathrm{DDH}}(\kappa) + \epsilon_{\mathrm{DL}}(\kappa)) = 7 \cdot (\epsilon_{\mathrm{DDH}}(\kappa) + \epsilon_{\mathrm{DL}}(\kappa))$.

Type 4: 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 collision-freeness game of the accumulator scheme Acc. Subsequently, we describe how $\mathcal{S}$ simulates the environment for $\mathcal{A}$ and interacts with the challenger $\mathcal{C}$.

$\mathcal{B}$ obtains the accumulator public key $\mathsf{pk}_{\Pi} \leftarrow (\mathsf{BG}, (\lambda^i P)_{i=0}^{t'}, (\lambda^i \hat{P})_{i=0}^{t'})$ from $\mathcal{C}$. Then, $\mathcal{S}$ completes the setup and initializes rpk and the organization key pair $(\mathsf{osk}, \mathsf{opk})$ in a way compatible with pk_{Π}. $\mathcal{S}$ runs $\mathcal{A}(\mathsf{pp}, \mathsf{opk}, \mathsf{rpk})$ and simulates all oracles as in a 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 in PoK and restarts $\mathcal{A}$ with a new challenge $c' \neq c$. Rewinding allows to obtain the used non-membership witness (W^*, V^*, d^*) and the used credential randomizer ρ, by the extraction of the corresponding discrete logarithms. $\mathcal{S}$ now computes $\mathsf{cred}_0^* \leftarrow \rho^{-1} \cdot \mathsf{cred}'^*[0]$ on the message part of the credential. Let $\mathsf{cred}' \in \mathtt{CRED}$ be such that $\mathsf{cred}'[0] = \mathsf{cred}_0^*$. If there is no such cred, then $\mathcal{S}$ and, in further consequence, $\mathcal{B}$ aborts. Otherwise, let nym^* be such that $\mathsf{cred}' = \mathtt{CRED}[\mathsf{nym}^*]$. If $\mathsf{nym}^* \notin \mathtt{RN}$, then $\mathcal{S}$ aborts. Otherwise, we know that the extracted witness (W^*, V^*, d^*) yields a correct evaluation of the verification relation, even though the corresponding nym has been revoked. Thus, $\mathcal{B}$ outputs $((W^*, V^*, d^*), \mathsf{nym}^*, \mathtt{RNYM})$ as a non-membership witness for an accumulated value, giving a collision for the accumulator. □

Theorem 7. *If the underlying SPS-EQ-$\mathcal{R}$ perfectly adapts signatures and the DDH assumption in $\mathbb{G}_1$ holds, then Scheme 4 is anonymous.*

Proof. Analogous to the anonymity proof of Scheme 2, we prove anonymity using a sequence of games, where each game is indistinguishable from the previous one. Henceforth, we denote the event that an adversary wins Game i by S_i. In all games, the setup is as in the original game, with the following differences. Upon generation of the public parameter pp—instead of choosing Q at random—one chooses $q, t \xleftarrow{R} \mathbb{Z}_p^*$ and sets $Q \leftarrow qP, T \leftarrow tP$. Then, the environment stores q, t as well as the trapdoors α and λ used for generating the tuples $(\alpha^i P)_{i=0}^t, (\alpha^i \hat{P})_{i=0}^t$ and $(\lambda^i P)_{i=0}^t, (\lambda^i \hat{P})_{i=0}^t$ contained in pp.

Game 0: The original anonymity game with $b = 0$.

Game 1: As Game 0, but the PoK in all showings is conducted by proving knowledge of q and simulating the proof part for the remainder. Furthermore, all calls to $\mathsf{ChgRep}_{\mathcal{R}}(M, \sigma, \rho, \mathsf{pk})$ are replaced by $(\rho M, \mathsf{Sign}_{\mathcal{R}}(\rho M, \mathsf{sk}))$.

Transition 1 - Game 0 → Game 1: Since the witness indistinguishability of the OR proof is unconditional and Scheme 3 perfectly adapts signatures, we have that $\Pr[S_1] = \Pr[S_0]$.

Game 2: As Game 1, except for the oracle $\mathcal{O}^{LoR}$, which is simulated as follows:

$\mathcal{O}^{LoR}(\mathsf{osk}, \mathsf{opk}, \mathsf{rsk}, \mathsf{rpk}, b, \mathsf{nym}_0, \mathsf{nym}_1, \mathbb{A}', \mathbb{R}_V)$**:** As in a real game, but the showing is simulated independently of bit b as follows. $\mathcal{S}$ chooses a message $(M_1, M_2, M_3, M_4) \xleftarrow{R} (\mathbb{G}_1^*)^4$, sets the shown credential $\mathsf{cred} \leftarrow ((M_1, M_2, M_3, M_4), \sigma)$, with $\sigma \leftarrow \mathsf{Sign}_{\mathcal{R}}((M_1, M_2, M_3, M_4), \mathsf{osk})$. Furthermore, $\mathcal{S}$ computes $\mathcal{C}_{\overline{\mathbb{A}'}} \leftarrow \frac{1}{\mathsf{enc}(\mathbb{A}')(\alpha)} C_1$ using trapdoor α. It chooses $\xi \xleftarrow{R} \mathbb{Z}_p^*$, computes $\mathcal{V} \leftarrow \xi\lambda P$, $\mathcal{W} \leftarrow \xi P$ and chooses $R \xleftarrow{R} \mathbb{G}_1$, $S \xleftarrow{R} \mathbb{G}_1$, $D \xleftarrow{R} \mathbb{G}_1$.

Transition Game 1 → Game 2: To show that the game change is indistinguishable, we have to show that the adversary will not detect that the showings in $\mathcal{O}^{RoR}$ are performed with respect to a random credential. To do so, we define the adversary's view V on the system. In particular, in V we only consider those values, which are crucial to proving anonymity with respect to one particular

showing.

$$V = \big(P, \hat{P}, Q, T, M, N, \lambda P, \mathsf{enc}(\mathbb{A})(\alpha)P, \mathsf{enc}(\mathbb{A})(\alpha)\hat{P},$$
$$\mathsf{enc}(\overline{\mathbb{A}'})(\alpha)P, \mathsf{enc}(\overline{\mathbb{A}'})(\alpha)\hat{P}, W, \hat{W}, \mathit{\Pi}, \hat{\mathit{\Pi}}, R_i, r_i\mathsf{enc}(\mathbb{A})P, \mathsf{nym}P + \upsilon Q, \mathcal{W}, \mathcal{V},$$
$$R, S, D, C_4, C_3, C_2, C_1, \mathcal{C}_{\overline{\mathbb{A}'}}\big).$$

Subsequently, we consider the two summands of $\mathsf{nym}P+\upsilon Q$ as separate elements, since the adversary can obtain them upon issuing. For our further illustrations, we make the discrete logarithms to the bases $P, \hat{P}, Q, M, N, T$ explicit.

$$V = \big(P, \hat{P}, Q, T, M, N, \lambda P, eP, e\hat{P},$$
$$\overline{e}P, \overline{e}\hat{P}, wP, w\hat{P}, aP, a\hat{P}, r_iP, r_ieP, nP, \upsilon Q, wP + t_1N, \lambda(wP + t_1N),$$
$$t_1M + t_2N, d^{-1}M + t_3N, nM + t_0N, \rho P, \rho T, \rho(nP + \upsilon Q), \rho r_ieP, \rho r_i\overline{e}P\big).$$

Now, to distinguish between Game 2 and Game 3, the adversary is required to distinguish the distributions $\mathcal{D}_1(V)$ and $\mathcal{D}_2(V)$, which are defined as follows:

$$\mathcal{D}_1(V) :\equiv \Big[\big(P, \hat{P}, Q, T, M, N, \lambda P, eP, e\hat{P},$$
$$\overline{e}P, \overline{e}\hat{P}, wP, w\hat{P}, aP, a\hat{P}, r_iP, r_ieP, nP, \upsilon Q, wP + t_1N, \lambda(wP + t_1N),$$
$$t_1M + t_2N, d^{-1}M + t_3N, nM + t_0N, \rho P, \rho T, \rho(nP + \upsilon Q), \rho r_ieP, \rho r_i\overline{e}P\big)\Big].$$

$$\mathcal{D}_2(V) :\equiv \Big[\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}, \mathsf{e}, \mathsf{f} \xleftarrow{R} \mathbb{Z}_p, \big(P, \hat{P}, Q, T, M, N, \lambda P, eP, e\hat{P},$$
$$\overline{e}P, \overline{e}\hat{P}, wP, w\hat{P}, aP, a\hat{P}, r_iP, r_ieP, nP, \upsilon Q, \mathsf{a}P, \lambda\mathsf{a}P,$$
$$\mathsf{b}P, \mathsf{c}P, \mathsf{d}P, \rho P, \rho T, \mathsf{e}P, \mathsf{f}P, \mathsf{f}e^{-1}\overline{e}P\big)\Big].$$

Observe that both, the left and the right distribution in Game 2, are distributed as $\mathcal{D}_1(V)$, whereas $\mathcal{D}_2(V)$ is distributed as the output of $\mathcal{O}_{LoR}$ in Game 3. Note that in this case all relevant values involved in the distribution changes are already drawn uniformly at random in the original game. Thus, we can directly prove Lemma 2 with respect to the distributions parametrized by V.

Lemma 2. *If the DDH assumption holds in $\mathbb{G}_1$, then for every PPT adversary $\mathcal{A}$ the probability to distinguish $\mathcal{D}_1(V)$ from $\mathcal{D}_2(V)$ is bounded by $2 \cdot \epsilon_{DDH}(\kappa)$.*

Proof (of Lemma 2). We show that—under the DDH assumption in $\mathbb{G}_1$—the probability to distinguish $\mathcal{D}_1(V)$ from $\mathcal{D}_2(V)$ is bounded by a negligible function in the security parameter κ. We do so, by a sequence of intermediate distributions. We start by defining $\mathcal{D}_3$:

$$\mathcal{D}_3(V) :\equiv \Big[\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d} \xleftarrow{R} \mathbb{Z}_p, \big(P, \hat{P}, Q, T, M, N, \lambda P, eP, e\hat{P},$$
$$\overline{e}P, \overline{e}\hat{P}, wP, w\hat{P}, aP, a\hat{P}, r_iP, r_ieP, nP, \upsilon Q, \mathsf{a}P, \lambda\mathsf{a}P,$$
$$\mathsf{b}P, \mathsf{c}P, \mathsf{d}P, \rho P, \rho T, \rho(nP + \upsilon Q), \rho r_ieP, \rho r_i\overline{e}P\big)\Big].$$

It is easy to see that $\mathcal{D}_1(V)$ and $\mathcal{D}_3(V)$ are indistinguishable, since the respective values in $\mathcal{D}_1(V)$ represent unconditionally hiding Pedersen commitments. Subsequently, we introduce $\mathcal{D}_4$:

$$\mathcal{D}_4(V) :\equiv \Big[\mathsf{a},\mathsf{b},\mathsf{c},\mathsf{d},\mathsf{e} \xleftarrow{R} \mathbb{Z}_p, \big(P,\hat{P},Q,T,M,N,\lambda P,eP,e\hat{P},\\ \overline{e}P,\overline{e}\hat{P},wP,w\hat{P},aP,a\hat{P},r_iP,r_ieP,nP,\upsilon Q,\\ \mathsf{a}P,\lambda\mathsf{a}P,\mathsf{b}P,\mathsf{c}P,\mathsf{d}P,\rho P,\rho T,\mathsf{e}P,\rho r_ieP,\rho r_i\overline{e}P\big)\Big].$$

To see that $\mathcal{D}_3(V)$ and $\mathcal{D}_4(V)$ are indistinguishable, assume a DDH instance $(P,\rho P,\upsilon P,\mathsf{r}P)$. Now, we can pad this instance to $(P,\hat{P},qP,tP,M,N,\lambda P,eP,e\hat{P},\overline{e}P,\overline{e}\hat{P},wP,w\hat{P},aP,a\hat{P},r_iP,r_ieP,nP,q\cdot\upsilon P,\mathsf{a}P,\lambda\mathsf{a}P,\mathsf{b}P,\mathsf{c}P,\mathsf{d}P,\rho P,t\cdot\rho P,n\cdot\rho P+q\cdot\mathsf{r}P,r_ie\cdot\rho P,r_i\overline{e}\cdot\rho P)$, using $e,\overline{e},w,a,r_i,n,q,\mathsf{a},\lambda,\mathsf{b},\mathsf{c},\mathsf{d},t$. Then, we have a distribution as in $\mathcal{D}_3(V)$ if $(P,\rho P,\upsilon P,\mathsf{r}P)$ is a valid DDH instance and a distribution as in $\mathcal{D}_4(V)$ if r is random. Below, we introduce $\mathcal{D}_5$:

$$\mathcal{D}_5(V) :\equiv \Big[\mathsf{a},\mathsf{b},\mathsf{c},\mathsf{d},\mathsf{e},\mathsf{f} \xleftarrow{R} \mathbb{Z}_p, \big(P,\hat{P},Q,T,M,N,\lambda P,eP,e\hat{P},\\ \overline{e}P,\overline{e}\hat{P},wP,w\hat{P},aP,a\hat{P},r_iP,r_ieP,nP,rQ,\mathsf{a}P,\lambda\mathsf{a}P,\\ \mathsf{b}P,\mathsf{c}P,\mathsf{d}P,\rho P,\rho T,\mathsf{e}P,\mathsf{f}eP,\mathsf{f}\overline{e}P\big)\Big].$$

To see that $\mathcal{D}_4(V)$ and $\mathcal{D}_5(V)$ are indistinguishable, assume a DDH instance $(P,\rho P,r_iP,\mathsf{r}P)$. Now, we can pad this instance to $(P,\hat{P},qP,tP,M,N,\lambda P,eP,e\hat{P},\overline{e}P,\overline{e}\hat{P},wP,w\hat{P},aP,a\hat{P},r_iP,e\cdot r_iP,nP,\upsilon Q,\mathsf{a}P,\lambda\mathsf{a}P,\mathsf{b}P,\mathsf{c}P,\mathsf{d}P,\rho P,t\cdot\rho P,\mathsf{e}P,e\cdot\mathsf{r}P,\overline{e}\cdot\mathsf{r}P)$, using $e,\overline{e},w,a,n,q,\mathsf{a},\lambda,\mathsf{b},\mathsf{c},\mathsf{d},\mathsf{e},t$. Then, we have a distribution as in $\mathcal{D}_4(V)$ if $(P,\rho P,r_iP,\mathsf{r}P)$ is a valid DDH instance and a distribution as in $\mathcal{D}_5(V)$ if r is random.

Finally, it is easy to see that $\mathcal{D}_5$ and $\mathcal{D}_2$ are identically distributed. A multiplication of $\mathsf{f}eP$ and $\mathsf{f}\overline{e}P$ by e^{-1} does not change the distribution: f is random and not contained in other elements and, hence, unconditionally hides e^{-1}.

Taking all together, the advantage of any PPT adversary to distinguish $\mathcal{D}_1(V)$ from $\mathcal{D}_2(V)$ is negligible and bounded by $\epsilon_{\mathcal{D}}(\kappa) \leq 2\cdot\epsilon_{DDH}(\kappa)$, which proves Lemma 2. □

By Lemma 2 we know that $|\Pr[S_2]-\Pr[S_3]| \leq \epsilon_{\mathcal{D}}(\kappa)$.

Game 3: As Game 2, but we set $b=1$.

Transition - Game 2 → Game 3: In Game 2, all values are independent of b, meaning that flipping b does not influence the distributions, i.e., $\Pr[S_2]=\Pr[S_3]$.

Game 4: As Game 3, but we simulate the $\mathcal{O}^{LoR}$ oracle as in the real game.

Transition - Game 3 → Game 4: Under the same argumentation as in Transition 2, we know that the probability to distinguish Game 3 and Game 4 is bounded by $|\Pr[S_3]-\Pr[S_4]| \leq \epsilon_{\mathcal{D}}(\kappa)$.

Game 5: As Game 4, but we honestly compute the OR proof PoK with respect to $C_3, C_4, \mathbf{d}', \Pi'$ and replace all calls to $(\rho M, \mathsf{Sign}_{\mathcal{R}}(\rho M,\mathsf{sk}))$ by $\mathsf{ChgRep}_{\mathcal{R}}(M,\sigma,\rho,$

pk).

Transition - Game 4 - Game 5: Under the same argumentation as in Transition 1, we know that $\Pr[S_4] = \Pr[S_5]$.

Game 0 represents the anonymity game with $b = 0$, whereas Game 5 represents the anonymity game with $b = 1$; both games are computationally indistinguishable. □