Breaking and Fixing Anonymous Credentials for the Cloud*

Ulrich Haböck[1][0000−0003−0467−9260] and Stephan Krenn[2][0000−0003−2835−9093]

[1] University of Applied Sciences FH Campus Wien, Vienna, Austria
ulrich.haboeck@fh-campuswien.ac.at
[2] AIT Austrian Institute of Technology GmbH, Vienna, Austria
stephan.krenn@ait.ac.at

Abstract. In an attribute-based credential (ABC) system, users obtain a digital certificate on their personal attributes, and can later prove possession of such a certificate in an unlinkable way, thereby selectively disclosing chosen attributes to the service provider. Recently, the concept of encrypted ABCs (EABCs) was introduced by Krenn et al. at CANS 2017, where virtually all computation is outsourced to a semi-trusted cloud-provider called wallet, thereby overcoming existing efficiency limitations on the user's side, and for the first time enabling "privacy-preserving identity management as a service".
While their approach is highly relevant for bringing ABCs into the real world, we present a simple attack allowing the wallet to learn a user's attributes when colluding with another user – a scenario which is not covered by their modeling but which needs to be considered in practice. We then revise the model and construction of Krenn et al. in various ways, such that the above attack is no longer possible. Furthermore, we also remove existing non-collusion assumptions between wallet and service provider or issuer from their construction. Our protocols are still highly efficient in the sense that the computational effort on the end user side consists of a single exponentiation only, and otherwise efficiency is comparable to the original work of Krenn et al.

Keywords: Attribute-based credentials ⋄ privacy-preserving authentication ⋄ strong authentication

1 Introduction

Anonymous attribute-based credential systems (ABCs) – first envisioned by Chaum [15,16] and extended in a large body of work [8,9,11,12,13,14,20,26,27,29] – are a cryptographic primitive enabling user-centric identity management. In ABC systems, a *user* receives a certificate on his personal data such as name, nationality, or date of birth from an *issuer*. Later, the user can *present* this certificate to *service providers* (or *relying parties*), thereby deciding which attributes to reveal or to keep private, in a way that makes different authentication processes unlinkable to each other. While the service provider receives strong authenticity guarantees on the received attributes, the user's privacy is maintained, even against colluding issuers and service providers.

However, despite of their obvious benefits, ABC systems have not yet found their way into relevant real-world applications. One main reason for this are computational costs, which make them unsuitable for resource-constraint devices.

* This article is based on the version published by Springer-Verlag available at https://doi.org/10.1007/978-3-030-31578-8_14.

This drawback was recently addressed by Krenn et al. [24], who proposed a scheme dubbed EABC, where virtually all computations can be outsourced to a semi-trusted *wallet*. The underlying idea was that users get signatures on their attributes, encrypted under some proxy re-encryption [6] scheme, from the issuer, and upload signature and ciphertexts to the wallet, together with a re-encryption key from their own public key to the intended service provider's public key. For presentation, the wallet re-encrypts the ciphertexts of the revealed attributes for the service provider, randomizes the remaining ciphertexts, and attaches a zero-knowledge proof of knowledge of a signature on the underlying ciphertexts. By the privacy property of the proxy re-encryption scheme, the wallet can translate encryptions from users to service providers, without ever learning any information about the underlying plaintexts. However, while solving the efficiency drawbacks of previous ABC systems, the attacker model underlying [24] is unrealistic, as they make very strong non-collusion assumptions between the wallet on the one hand, and service providers or issuers on the other hand. Even worse, we point out a trivial attack which allows the wallet to recover a user's personal attributes when colluding with another user. While this collusion is not reflected in their security model (i.e., not considered an attack there), we believe that this modeling is unrealistic, and user-wallet collusions need to be considered in any practical protocol in order to capture, e.g., the case of malicious administrators.[3]

The attack on Krenn et al.[24]. The fundamental problem of [24] is that for efficiency reasons their construction makes use of bi-directional multi-hop proxy re-encryption schemes (in order to not having to use generic approaches to zero-knowledge). That is, having a re-encryption key $rk_{A\to B}$ that allows a proxy to translates a ciphertext c_A encrypted under pk_A to a ciphertext c_B under pk_B without learning the plaintext, and a re-encryption key $rk_{B\to C}$, the proxy can also translate c_A to c_C under pk_C (multi-hop); furthermore, $rk_{A\to B}$ can efficiently be turned into $rk_{B\to A}$ (bi-directionality).

Assume now that Alice A wants to authenticate herself towards some service provider SP and thus stores $rk_{A\to SP}$ and encryptions c_A of her personal attributes on the wallet. Let the malicious administrator M also sign up for SP and compute $rk_{M\to SP}$. Using the bi-directionality of the proxy re-encryption scheme, this directly gives $rk_{SP\to M}$, and using the multi-hop functionality, M can now translate all of A's ciphertexts for herself, thereby fully breaking Alice's privacy. Even more, because of the concrete choice of the deployed re-encryption scheme, the attacker could even recover Alice's secret key as $sk_A = rk_{A\to SP}^{-1} \cdot sk_{M\to SP} \cdot sk_M^{-1}$ without having to assume a corrupt service provider. Actually, also the secret key of the service provider can be recovered as $sk_{SP} = sk_{M\to SP} \cdot sk_M^{-1}$.

Note that this attack is not specific to the deployed scheme of Blaze et al. [6], but arises in any multi-hop proxy re-encryption scheme that is used for outsourced data sharing application using long-term keys for the relying parties.

Mitigation strategies. A straightforward solution to this problem might be to replace the deployed proxy re-encryption scheme by a single-hop and/or uni-directional encryption scheme. However, it turns out that the algebraic structures of existing signature and encryption schemes (with the required properties) would then no longer allow for efficient zero-knowledge proofs or knowledge, and the benefits of [24] would dissolve. Very informally speaking, the reason for this is that all such potential schemes would "consume" the one available pairing in the system. Further-

[3] Note that in previous versions of this paper including [23] the formulation suggested that the attack was within the modeling presented by Krenn et al. [24]; however, we want to make explicit that the attack is not possible in their model, but should be considered in practice.

more, the other limitations of [24] (i.e., non-collusion assumptions) would not be addressed by such a modification.

Our contribution. The main contribution of this paper is to overcome the security limitations of [24] without harming the efficiency of the scheme. That is, we provide an instantiation of an EABC system that does not require any artificial non-collusion assumptions, at the cost of only a single exponentiation on the user's side. Furthermore, in contrast to [24], our system also gives metadata-privacy guarantees in the sense that the wallet only learns the policy for which it is computing the presentation tokens (i.e., which attributes are revealed and which remain undisclosed), but does no longer learn for which service provider it is computing the presentation, such that reliably tracking users becomes virtually impossible. Hiding the presentation policy within a set of policies could be achieved by the techniques of Krenn et al. [24, § 6.1] for a linear overhead.

In a bit more detail, our contribution is multifold.

- Firstly, we replace the static long-term keys used by the service providers in [24] by ephemeral keys which are only used for a single authentication. This is achieved through an interactive key agreement protocol between the two parties, which guarantees freshness of the agreed keys. By this, a malicious administrator can no longer run the attack described above, as the $rk_{A\to SP}$ and $rk_{M\to SP}$ will no longer be bound to the same key of the service provider.
- Next, by using independent keys for the individual user attributes, even a collusion of service provider and wallet may only reveal the information that the user was willing to share with the service provider in any case.
- Thirdly, by replacing the signature scheme deployed in the issuance phase by a blinded version of the same scheme, our construction achieves high unlinkability guarantees even in the case of wallet-issuer collusions. Our blinded version of the structure-preserving signature scheme of Abe et al. [2] may be also of independent interest beyond the scope of this paper.
- Finally, by having a separate identity key that is not stored on the wallet but locally on the user's device, the service provider is guaranteed that the user is actively participating in the protocol. While Krenn et al. [24] considered it undesirable that users need to carry secret key material with them, we believe that having no information stored locally results in unrealistic trust assumptions as there the wallet could impersonate a user towards any service provider that the user ever signed up for.

Related work. Since [8,11] several improvements have been done to reduce the computational costs on the user side when presenting credentials, e.g., [3,4,10,14,27]. Most notably, [20] introduce a novel approach based on a structure-preserving signature scheme on equivalence classes (SPS-EQ) and set commitments, in which the user's effort is decreased to five, plus the number of undisclosed attributes, exponentiations on an elliptic curve, and a two exponent zero-knowledge proof. Our work outsources the largest part of computations to the cloud-based identity provider (the wallet), leaving the user with a single exponent zero-knowledge proof, independent of the size of the credential. A similar idea is followed in Direct Anonymous Attestation with Attributes (DAA-A) [17], which also externalizes most of the TPM's effort to a helping environment (the platform's "host" therein). However, DAA-A differs significantly from EABC by the trust assumptions on the host, which are neither realistic nor desireable in the case of cloud-based identity providers.

Outline. This paper is organized as follows. In Section 2 we discuss the building blocks of EABC systems, and in particular the schemes needed for our concrete instantiation. Then, in Section 3 we give a high-level description of EABC systems, its revised adversary model and security notions.

Finally, Section 4 presents the concrete EABC instantiation, including security statements, the proofs of which are postponed to Appendix A.

2 Preliminaries

In the following we introduce the necessary background needed in the rest of the paper. In particular, we recap the notions of proxy re-encryption and structure-preserving signatures. We then present a transformation of the AGHO signature scheme [2] into a blinded version, which combines both features, blindness and structure-preservation, needed to efficiently instantiate EABC systems.

2.1 Notation

We denote the security parameter by λ. All probabilistic, polynomial time (PPT) algorithms are denoted by sans-serif letters ($\mathsf{A}, \mathsf{B}, \ldots$), and their combination in two-party or three party protocols by $\langle \mathsf{A}, \mathsf{B} \rangle$ and $\langle \mathsf{A}, \mathsf{B}, \mathsf{C} \rangle$, respectively. Whenever we sample a random element m uniformly from a finite set M, we denote this by $m \leftarrow\!\!\$\, M$. We write $\mathbb{Z}_q$ for the integers modulo a prime number q, $\mathbb{Z}_q^*$ for its multiplicative group, and $1/e$ for the modular inverses. We shall make extensive use of non-interactive zero-knowledge proofs of knowledge, where we use the Camenisch-Stadler notation to specify the proof goal. For example,

$$\mathsf{NIZK}\left[(\alpha, \beta, \Gamma) : y_1 = g^\alpha \wedge y_2 = g^\alpha \cdot h^\beta \wedge R = e(\Gamma, H)\right]$$

denotes a non-interactive zero-knowledge proof of knowledge proving knowledge of values α, β, Γ such that the expression on the right-hand side is satisfied. In most situations, extractability of zero-knowledge proofs will be sufficient. However, in a single case we will require simulation-sound extractablility [22].

2.2 Anonymous Re-Randomizable Proxy Re-Encryption

A proxy re-encryption (PRE) scheme is an asymmetric encryption scheme which allows a third party (the *proxy*) to transform ciphertexts encrypted for one party into ciphertexts encrypted for another one, without learning the underlying plaintext. As in [24], we instantiate our EABC system by using the scheme by Blaze et al. [6] (BBS); the associated issues in [24] are mitigated by a different use of the scheme. It possesses all the security properties needed for proving our system secure, yet it yields algebraic simple relations for encryption, re-encryption and re-randomization, altogether allowing for efficient zero-knowledge proofs of statements which involve these operations.

The BBS scheme consists of six PPT algorithms,

$$\mathsf{PRE}_{BBS} = (\mathsf{Par}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{ReKey}, \mathsf{ReEnc}),$$

where $\mathsf{Par}(\lambda)$ outputs the system parameters $pp = (\mathbb{G}, q, g)$, where $\langle g \rangle = \mathbb{G}$ is a group of prime order q. $\mathsf{Gen}(pp)$ generates a key pair (sk, pk) by $sk \leftarrow\!\!\$\, \mathbb{Z}_q$ and $pk = (pp, g^{sk})$. Encryption and decryption works as for ElGamal [21], i.e.,

$$\mathsf{Enc}(pk, m) = c = (c_1, c_2) = (g^r, pk^r \cdot m),$$

where $m \in \mathbb{G}$ is the message, $r \leftarrow\!\!\$\, \mathbb{Z}_q$, and $\mathsf{Dec}(sk, c) = c_1^{-sk} \cdot c_2$.

Given two key pairs (pk_1, sk_1), (pk_2, sk_2), their re-encryption key $rk = rk_{pk_1 \to pk_2}$ is derived by $\mathsf{ReKey}(sk_1, pk_1, sk_2, pk_2) = sk_1 \cdot sk_2^{-1}$, and

$$\mathsf{ReEnc}(rk, c) = (c_1^{rk}, c_2)$$

transforms a ciphertext $c = (c_1, c_2)$ for pk_1 to one with respect to pk_2.

The relevant properties of the BBS scheme are summarized next.

Proposition 1 ([6]). *Under the DDH assumption in the message space $\mathbb{G}$, the BBS scheme is PRE-IND-CPA secure. That is,it is IND-CPA secure even under knowledge of (polynomially many) re-encryption keys that do not allow the adversary to trivially decrypt the challenge ciphertext.*

Proposition 2 ([24]). *The BBS PRE scheme with re-randomization function $\mathsf{ReRand}(pk, c) = \mathsf{Enc}(pk, 1) \cdot \mathsf{Enc}(pk, c) = (g^r \cdot c_1, pk^r \cdot c_2)$, $r \leftarrow_\$ \mathbb{Z}_q$, has the* ciphertext re-randomization property. *That is, given pk, a message m and its ciphertext c, then the output distribution of $\mathsf{ReRand}(pk, c)$ is computationally indistinguishable from that of $\mathsf{Enc}(pk, m)$.*

Proposition 3 ([24]). *Under the DDH assumption in $\mathbb{G}$, the BBS proxy-re-encryption scheme is* anonymous. *That is, for any* PPT *adversary* Adv *there exists a negligible function ν such that*

$$\left| \Pr \begin{bmatrix} pp \leftarrow \mathsf{Par}(\lambda); (sk_i, pk_i) \leftarrow \mathsf{Gen}(pp), i \in \{0,1\}; \\ (m, st) \leftarrow \mathsf{Adv}(pp, pk_1, pk_2); & : b^* = b \\ b \leftarrow_\$ \{0,1\}; b^* \leftarrow \mathsf{Adv}(st, \mathsf{Enc}(pk_b, m)) \end{bmatrix} - \frac{1}{2} \right| \leq \nu(\lambda)$$

2.3 Structure-Preserving Blind Signatures

The structure-preserving signature scheme of Abe et al. [1,2] is based on asymmetric bilinear groups $(\mathbb{G}, \mathbb{H}, \mathbb{T}, e)$ with the feature that messages, signatures and verification keys consist of elements from $\mathbb{G}$ and/or $\mathbb{H}$, and verification is realized by pairing-product equations over the key, the message and the signature. This allows for efficient zero-knowledge proofs of claims involving the message and the signature, which is why they apply to various cryptographic protocols, e.g., [1,19,20,24]. Similarly, our construction relies on the scheme in [2] (AGHO), since it allows to sign vectors of group elements. The AGHO scheme

$$\mathsf{SIG}_{AGHO} = (\mathsf{Par}, \mathsf{Gen}, \mathsf{Sig}, \mathsf{Vf})$$

consists of four PPT algorithms. The setup algorithm Par generates the scheme's parameters $pp = (\mathbb{G}, \mathbb{H}, \mathbb{T}, q, e, G, H)$ which are comprised of groups $\mathbb{G}$, $\mathbb{H}$, $\mathbb{T}$ of prime order q, a bilinear mapping $e : \mathbb{G} \times \mathbb{H} \longrightarrow \mathbb{T}$, and their respective generators G, H, $e(G, H)$. $\mathsf{Gen}(pp)$ produces a private-public key pair (sk, vk),

$$sk = \left(v, (w_i)_{i=1}^l, z\right) \quad \text{and} \quad vk = \left(V, (W_i)_{i=1}^l, Z\right) = \left(H^v, (H^{w_i}), H^z\right),$$

where all the secret components v, z, and w_i are randomly sampled from $\mathbb{Z}_q$. Given $m = (g_i)_{i=1}^l$ from $\mathbb{G}^l$, we have that $\sigma = \mathsf{Sig}(sk, m) = \sigma = (R, S, T) \in \mathbb{G} \times \mathbb{G} \times \mathbb{H}$, where

$$R = G^r, \qquad S = G^z \cdot R^{-\cdot v} \cdot \prod_{i=1}^{l} g_i^{-w_i}, \qquad T = H^{1/r},$$

for $r \leftarrow_\$ \mathbb{Z}_q^*$. The verification condition of $\sigma = (R, S, T)$ is given by the two bilinear equations $e(S, H) \cdot e(R, V) \cdot \prod_i e(g_i, W_i)$ and $e(R, T) = e(G, H)$.

Theorem 1 ([2]). *In the generic group model, the AGHO signature scheme* $\mathsf{SIG} = (\mathsf{Par}, \mathsf{Gen}, \mathsf{Sig}, \mathsf{Vf})$ *is strongly existentially unforgeable under adaptive chosen message attacks (sEUF-CMA). That is, for every* PPT *adversary* Adv *there exists a negligible function* ν *such that*

$$\Pr\left[\begin{array}{ll} pp \leftarrow \mathsf{Par}(\lambda); (vk, sk) \leftarrow \mathsf{Gen}(pp) & \\ (m^*, \sigma^*) \leftarrow \mathsf{Adv}^{\mathsf{Sig}(pp, sk, \,.\,)} & : \begin{array}{l} \mathsf{Vf}(vk, (m^*, \sigma^*)) = 1 \wedge \\ (m^*, \sigma^*) \notin Q \end{array} \end{array}\right] \leq \nu(\lambda)$$

where Adv *has access to a signing oracle* $\mathsf{Sig}(pp, sk\,.\,)$*, which on input* m *computes a valid signature* σ*, adds* (m, σ) *to the initially empty list* Q*, and returns* σ*.*

Blind signatures allow a user to obtain signatures in a way such that both the message as well as the resulting signature remain hidden from the signer. *Restrictive* blind schemes additionally allow the signer to encode information into the message, while still preserving the unlinkability of the resulting message-signature pair to the issuance session. The notion of restrictiveness goes back to Brands [7], and various adaptions have been made since then, e.g., [8,18,25]. In the context of anonymous credentials, and for the first time done in [8], such restricted message is typically a commitment on a value defined by the issuer. As such, we consider a restrictive blind signature scheme

$$\mathsf{BSIG} = (\mathsf{Par}, \mathsf{Gen}, \mathsf{User}, \mathsf{Signer}, \mathsf{Vf})$$

being based on a blind signature scheme and a commitment scheme

$$\mathsf{COM} = (\mathsf{Par}_{COM}, \mathsf{Comm}_{COM}, \mathsf{Vf}_{COM})$$

for values x such that its output is in the message space of the signature. Par, on input the security parameter λ, sets up the scheme's parameters pp, including a compliant setting of COM, and $\mathsf{Gen}(pp)$ generates a private-public key pair (sk, vk). The interactive algorithms User and Signer define the issuance protocol

$$\big\langle \mathsf{User}(vk, x), \mathsf{Signer}(sk, x) \big\rangle$$

between a user and a signer with private-public key pair (sk, pk), which on input a commonly agreed value $x \in X$ outputs to the user a certificate (com, σ), which consists of a commitment com of x and a valid signature σ on com, together with an opening w . The verification $\mathsf{Vf}(vk, x, w, (com, \sigma)) = 1$ is a separate validity check of the commitment com on (x, w) and the signature σ on com.

The notions of unforgeability and blindness adapted to our setting of restrictive blind signatures are as follows.

Definition 1. *A restrictive blind signature scheme* $\mathsf{BSIG} = (\mathsf{Par}, \mathsf{Gen}, \mathsf{User}, \mathsf{Signer}, \mathsf{Vf})$ *is* strongly unforgeable *if for any* PPT *adversary* Adv *there exists a negligible function* ν *such that*

$$\Pr\left[\begin{array}{ll} pp \leftarrow \mathsf{Par}(\lambda); & \\ (vk, sk) \leftarrow \mathsf{Gen}(pp); Q \leftarrow \emptyset & \\ (x^*, (w_i^*, com_i^*, \sigma_i^*)_{i=1}^q) \leftarrow \mathsf{Adv}^{\langle\,.\,, \mathsf{Signer}(sk, \,.\,)\rangle} & : \begin{array}{l} q > mult(x^*) \wedge \\ \textit{for } 1 \leq i \leq q \\ \mathsf{Vf}(vk, x^*, w_i^*, (com_i^*, \sigma_i^*)) = 1 \wedge \\ \bigwedge\limits_{j \neq i} (com_j^*, \sigma_j^*) \neq (com_i^*, \sigma_i^*) \end{array} \end{array}\right]$$

is $\leq \nu(\lambda)$*, where* Adv *has access to a signing oracle* $\langle\,.\,, \mathsf{Signer}(sk, \,.\,)\rangle$ *which logs every successful query* x *in an initially empty list* Q*, and* $mult(x^*)$ *denotes the multiplicity of successful queries with* $x = x^*$*.*

Definition 2. *A restrictive blind signature scheme* $\mathsf{BSIG} = (\mathsf{Par}, \mathsf{Gen}, \mathsf{User}, \mathsf{Signer}, \mathsf{Vf})$ *satisfies* blindness*, if for any* PPT *adversary* Adv *there exists a negligible function* ν *such that*

$$\left| \Pr \left[\begin{array}{l} pp \leftarrow \mathsf{Par}(\lambda); (vk^*, x_0^*, x_1^*, st) \leftarrow \mathsf{Adv}(pp); \\ ((w_i, com_i, \sigma_i), st) \leftarrow \left\langle \mathsf{User}(vk, x_i^*), \mathsf{Adv}(st) \right\rangle, \ i \in \{0,1\} \\ \textbf{if } \sigma_0 = \bot \vee \sigma_1 = \bot \textbf{ then } \quad (\sigma_0, \sigma_1) = (\bot, \bot) \\ b \leftarrow_\$ \{0,1\}; \ b^* \leftarrow \mathsf{Adv}(st, (com_b, \sigma_b), (com_{1-b}, \sigma_{1-b})) \end{array} : b^* = b \right] - \frac{1}{2} \right| \leq \nu(\lambda) \,.$$

Blind AGHO Scheme. Our structure-preserving restrictive blind signature scheme $\mathsf{BSIG}_{AGHO} = (\mathsf{Par}, \mathsf{Gen}, \mathsf{User}, \mathsf{Signer}, \mathsf{Vf})$ is based on the AGHO scheme SIG_{AGHO}, a compatible commitment scheme COM, and two non-interactive extrable zero-knowledge proof systems, to both of which we refer to as NIZK without causing confusion. Par, Gen, and Vf are the corresponding algorithms from SIG_{AGHO}, besides that Par also queries Par_{COM} so that the commitments are elements of the messages space $\mathbb{G}^l$ of the AGHO scheme, and furthermore generates a common reference string for the zero-knowledge proof systems. We stress that our scheme is not merely based on a structure-preserving signature (as the ones from, e.g., [1,19,20]) but is structure-preserving by itself, which is an essential feature for our EABC instantiation.

The underlying idea in Defintion 3 is as follows. The user computes a commitment $m = com$ on the value x, and blinds m additively by a random pad P. It then obfuscates the pad by a simple exponentiation, and proves to the signer the wellformedness of the blinded commitment $\overline{m}$ and the obfuscated pad $\overline{P}$ in zero-knowledge, hiding m and P from the signer. On a valid wellformedness proof, the signer produces 'signatures' on $\overline{m}$ and $\overline{P}$, both of which are not valid signatures by themselves but can be combined to a valid one – whereas its randomness is determined by a Diffie-Hellman key establishment between the user and the signer.

Definition 3 (Blind AGHO signature on committed values). *The issuance protocol* $\left\langle \mathsf{User}(vk, x), \mathsf{Signer}(sk, x) \right\rangle$ *runs between a signer* S *with AGHO signing keys* $sk = (v, (w_i)_{i=1}^l, z)$*,* vk*, and a user* U *who wishes to receive a certificate* (w, com, σ) *on the commonly agreed value* x *from the signer.*

1. *U computes com on x with opening w by using Comm_{COM}. By our assumption on COM, $m = com$ is from the message space of the signature, i.e. $m = (m_i)_{i=1}^l \in \mathbb{G}^l$.*
2. *U blinds m using a random pad $P = (P_i)_{i=1}^l \leftarrow_\$ \mathbb{G}^l$, and obtains $\overline{m} = (\overline{m}_i)_{i=1}^l = \left(m_i \cdot P_i^{-1}\right)_{i=1}^l$. It further chooses $e, f \leftarrow_\$ \mathbb{Z}_q^*$, a random decomposition $f = f_1 + f_2$ of f, and sets $\overline{P} = (P_i^e)_{i=1}^l$, $(G_1, G_2, G_3) = (G^e, G^{f_1}, G^{e \cdot f_2})$. U then sends $\overline{m}$, $\overline{P}$, (G_1, G_2, G_3) to S and gives a zero knowledge of wellformedness*

$$\pi_U = \mathsf{NIZK}\Big[(\eta, \varphi_1, \varphi_2, \omega) : G_1^\eta = G \wedge G^{\varphi_1} = G_2 \wedge G_1^{\varphi_2} = G_3 \wedge \mathsf{Vf}_{COM}(\overline{m} \cdot \overline{P}^\eta, x, \omega) = 1\Big],$$

   using the witnesses $(\eta, \varphi_1, \varphi_2, \omega) = (1/e, f_1, f_2, w)$.
3. *S verifies π_U, and returns $\bot$ if not valid. Otherwise it generates a random decomposition $z = z_1 + z_2$ of its signing key's z, and computes the 'signatures' $\overline{\sigma} = \left(\overline{R}, \overline{S}_1, \overline{S}_2, \overline{T}\right)$, with $\overline{R} = G^r$, $\overline{T} = H^{1/r}$, $\overline{S}_1 = G^{z_1} \cdot G_2^{-r \cdot v} \cdot \prod_{i=1}^l \overline{m}_i^{-w_i}$, $\overline{S}_2 = G_1^{z_2} \cdot G_3^{-r \cdot v} \cdot \prod_{i=1}^l \overline{P}_i^{-w_i}$, where $r \leftarrow_\$ \mathbb{Z}_q^*$. It then*

returns $\overline{\sigma}$ to U supplemented by a proof of wellformedness

$$\pi_S = \mathsf{NIZK}\Big[(\rho,\tau,(\omega_i)_i,\zeta_1,\zeta_2) : \bigwedge_i H^{\omega_i} = W_i \ \wedge\ H^{\zeta_1}\cdot H^{\zeta_2} = Z \ \wedge \\ G^{\rho} = \overline{R} \ \wedge\ \overline{T}^{\rho} = H \ \wedge\ V^{\rho}\cdot H^{-\tau} = 1 \ \wedge \\ G^{\zeta_1}\cdot G_2^{-\tau}\cdot\prod_{i=1}^{l}\overline{m}_i^{-\omega_i} = \overline{S}_1 \ \wedge\ G_1^{\zeta_2}\cdot G_3^{-\tau}\cdot\prod_{i=1}^{l}\overline{P}_i^{-\omega_i} = \overline{S}_2\Big],$$

by using the witnesses $(\rho,\tau,(\omega_i)_i,\zeta_1,\zeta_2) = (r, r\cdot v, (w_i)_i, z_1, z_2)$.

4. *U checks if π_S is valid. If so, she outputs $m = com$, w, and $\sigma = (R,S,T)$, where $R = \overline{R}^f$, $S = \overline{S}_1\cdot\overline{S}_2^{1/e}$, $T = \overline{T}^{1/f}$. (Otherwise she outputs $\perp$).*

Corrrectness of the blind AGHO scheme follows from $R = \overline{R}^f = G^{r\cdot f}$, $T = \overline{T}^{1/f} = H^{\frac{1}{r\cdot f}}$, and

$$S = \overline{S}_1\cdot\overline{S}_2^{1/e} = G^{z_1}\cdot G_2^{-r\cdot v}\cdot\prod_{i=1}^{l}\overline{m}_i^{-w_i}\cdot\Big(G_1^{z_2}\cdot G_3^{-r\cdot v}\cdot\prod_{i=1}^{l}\overline{P}_i^{-w_i}\Big)^{1/e} = \\ = G^{z_1+z_2}\cdot G^{-r\cdot(f_1+f_2)\cdot v}\cdot\prod_{i=1}^{l}(\overline{m}_i\cdot P_i)^{-w_i} = G^{z}\cdot G^{-r\cdot f\cdot v}\cdot\prod_{i=1}^{l}m_i^{-w_i},$$

alltogether representing an AGHO signature of $m = (m_i)$ with respect to $sk = (v,(w_i),z)$ using randomness $r\cdot f$.

Theorem 2. *Suppose that both* NIZK *in Definition 3 are extractable. If the commitment scheme* COM *is computationally hiding, then under the DDH assumption in $\mathbb{G}$ the restrictive blind signature scheme BSIG_{AGHO} satisfies blindness. Furthermore, if* COM *is computationally binding, BSIG_{AGHO} is strongly unforgeable in the generic group model.*

The proof of Theorem 2 is postponed to Appendix A.1

3 EABC: High-Level Description

An *encrypted* attribute-based credential (EABC) system, introduced in [24], allows the delegation of selective disclosure to a third party in a privacy-preserving manner by means of proxy re-encryption and redactable signatures. There are four types of players in an EABC system, as depicted in Figure 1: *issuers*, *users*, *services*, and the central *wallet*. Each user U holds an *identity key* sk_U proving her identity and which is securely stored on her trusted device (e.g., a smart card or TPM). U engages in an *issuance protocol* with an issuer I to receive an encrypted credential C on certain attributes only readable to her such that C is bound to her identity key sk_U. U further owns an account managed by the wallet W, a typically cloud-based identity provider, to which she uploads all of her encrypted credentials C while not providing it the encryption keys $k(C)$. At any time later, when U wants to access a service S she is asked to attest some of her attributes. To convince S of the requested attributes without revealing any further testified information, U chooses one (or several) of her credentials from her account, selects a subset of attributes contained therein, and

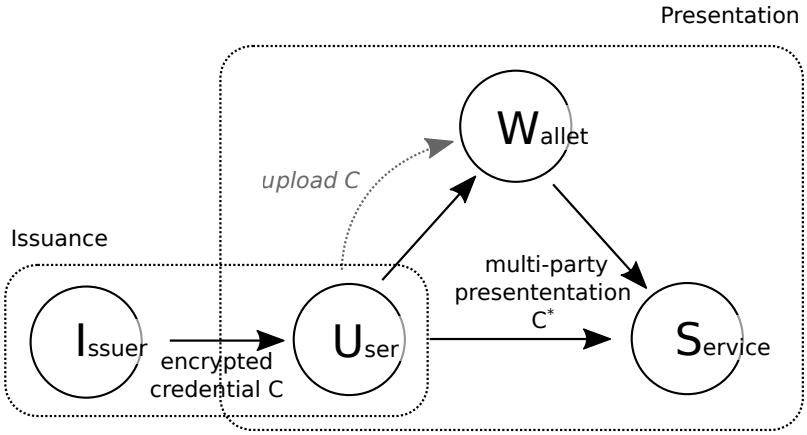

Fig. 1. Overview of an EABC system and its two main protocols.

instructs W to engage in a *presentation protocol* with S, which serves the latter re-encryptions of the requested attributes together with a proof of validity. In this protocol, the wallet W undertakes (almost) all costly operations while reducing U's effort to a possible minimum, requiring her only to supply the re-encryption keys for the selected attributes, and a proof of her consent (via her sk_U) to the presentation process. This last proof is also how the overall model of EABCs differs from that in [24], where no computation is required on the user's side at all; however, as discussed earlier, we believe this is needed for a realistic attacker scenario, as otherwise the wallet could arbitrarily impersonate the user towards any service provider that the user ever signed up for.

3.1 Formal Definition

An EABC system with attribute space $\mathbb{A}$ is built on a structure-preserving blind signature scheme BSIG in the sense of Section 2.3, an anonymous re-randomizable proxy re-encryption scheme PRE (cf. Section 2.2) which acts on the message space of BSIG, and two zero-knowledge proof systems to which we both refer as ZKP without causing confusion. Formally, an EABC system

$$\mathsf{EABC} = (\mathsf{Par}, \mathsf{Gen}_I, \mathsf{Gen}_U, \mathsf{Iss}, \mathsf{User}_I, \mathsf{User}_P, \mathsf{Wall}, \mathsf{Serv})$$

consists of the (PPT) algorithms Par, Gen_I, and Gen_U for setup and key generation, and the interactive (PPT) algorithms Iss, User_I, User_P, Wall, Serv which are the components of the issuance and presentation protocol described below. Given the security parameter λ, $\mathsf{Par}(\lambda)$ generates the system parameters sp. These are comprised of the parameters for BSIG, PRE, and a common reference string for ZKP. Every user U holds a PRE secret-public key pair (sk_U, pk_U) generated by $\mathsf{Gen}_U(sp) = \mathsf{Gen}_{PRE}(sp)$, her *identity key*, which is used to generate her certificate pseudonyms. On demand, a user repeatedly queries $\mathsf{Gen}_U(sp)$ to generate the encryption keys $k(C)$ of her credentials. Each issuer I is holder of a key pair for the blind signature scheme $(sk_I, vk_I) \leftarrow \mathsf{Gen}_I(sp)$, $\mathsf{Gen}_I = \mathsf{Gen}_{BSIG}$, where sk_I denotes the secret signing key and vk_I its public verification key. Note that, unlike in [24] a service has no permanent key material for the proxy re-encryption scheme. Its PRE key will be an ephemeral one-time key, one for each presentation. Both the issuance and

the presentation protocol run over server-side authenticated, integrity protected and confidential connections (associated with some random session identifier sid) and are as follows.

Issuance. The issuance protocol

$$\left\langle \mathsf{User}_I\left(sid, sk_U, A, vk_I\right), \mathsf{Iss}\left(sid, A, sk_I[, pk_U]\right) \right\rangle$$

is performed between a user U with identity keys (sk_U, pk_U) and an issuer I with signature key pair (sk_I, vk_I) who is the supplier of the random session identifier sid. Both user and issuer agreed on the unencrypted content, the attributes $A = (a_i)_{i=1}^{l} \in \mathbb{A}^l$ beforehand. Depending on the type of issuance, it might be mandatory that U authenticates with its identity key, hence we leave it optional whether pk_U is supplied to I or not, denoted by $[, pk_U]$. If successful, the protocol outputs to U an encrypted attribute-based credential (C, vk_I) together with it's (secret) key material $sk = sk(C)$, the latter of which U keeps on her device (and never provides it to a third party). In all other cases, the user receives $\perp$.

Presentation. The presentation protocol is a three party protocol

$$\left\langle \mathsf{User}_P\left(sid, sk_U, sk(C), D\right), \mathsf{Wall}\left(sid, C, D, vk_I\right), \mathsf{Serv}\left(sid, D, vk_I\right) \right\rangle$$

and involves a user U with identity key sk_U, the wallet W which hosts U's credential (C, vk_I), and a service S, who provides the random session identifier sid. As before, $sk(C)$ is the user's secret key material for C. The user decides the attributes in C to be disclosed to S beforehand, associated with some index subset $D \subseteq \{1, \ldots, l\}$. At the end of the protocol the service receives a *presentation* C^* of C, which is comprised of the requested attributes, re-encrypted to a random one-time key sk' of S, together with a proof of validity. The service verifies the proof by help of the issuer's public key vk_I. If valid, the service accepts and decrypts the attributes using sk'. Otherwise it rejects and outputs $\perp$ to both U and W.

3.2 EABC Security Notions

We widen the adversary model from [24] to a setting which does not impose any trust assumption on the wallet. An attacker who controls several players of the EABC system, i.e. the central wallet, some of its users, service providers and issuers, should not be able to compromise the system in a more than obvious manner. That is, the adversary should not be able to

1. efficiently generate valid presentations which do not match any of the adversary's credentials (*unforgeability*),
2. alter the statement of a presentation successfully without knowledge of the user (*non-deceivability*),
3. learn anything from the encrypted credentials besides the information disclosed under full control of the owner (*privacy*), and
4. distinguish presentations with the same disclosed content when the underlying encrypted credentials are not known to the adversary (*unlinkability*).

We note that unforgeability of EABC covers impersonation attacks against honest users, but not a malicious wallet trying to manipulate of the outcome of an honest user's presentation session (within the set of her attributes), i.e. non-decevability. Furthermore, since now the adversary takes the position of the wallet as well as a service, we are confronted with two notions of unlinkability: the untraceability of encrypted credentials back to it's issuance session (covered by privacy), and the indistinguishability of presentations with the same disclosed content (unlinkability).

All security notions are given in a game-based manner, and we assume server-authenticated, confidential and integrity protected connections between the protocol participants.

Unforgeability for EABC The *unforgeability experiment* paraphrases a malicous wallet and (adaptively many) malicious users, who altogether try to trick an honest service into accepting a presentation which does not match any of the adversary's queries to an honest issuer. The experiment manages a list L which records the key material of all honest system participants during the entire lifetime of the system, i.e. the honest user's identity keys (pk_U, sk_U) and honest issuer keys (vk_I, sk_I). The list L is also used to log all honest user's credentials $(C, vk_I, sk_U, sk(C))$ under a unique handle h.

At any time the adversary Adv *is given access to all public information contained in* L*, i.e. the public keys* pk_U*,* vk_I *and the handles* h*, and as wallet* W^* *it may retrieve the encrypted credentials* (C, vk_I) *of every handle* h *contained in* L*.*

Besides L, the experiment maintains another list Q_{Adv} used for logging all adversaries queries to honest issuers of the system. At first, the experiment initializes the system by running $sp \leftarrow \mathsf{Par}(\lambda)$, setting $L = \emptyset$, $Q_{\mathsf{Adv}} = \emptyset$, and returns sp to the adversary Adv. The adversary may then generate and control adaptively many (malicious) players, and interact with the honest ones by use of the following oracles:

Issuer oracle $\mathsf{I}(vk_I, A\ [, pk_U^*])$**.** This oracle, on input an issuer's vk_I, attributes $A = (a_i)_i$, and optionally a public identity key pk_U^*, provides the adversary a (stateful) interface to an honest issuer's $\mathsf{Iss}(A, sk_I\ [, pk_U^*])$ in the issuance protocol, provided that vk_I is listed in L. If not, then the oracle generates a fresh pair of issuer keys $(sk_I, vk_I) \leftarrow \mathsf{Gen}_I(sp)$, adds it to L, and returns vk_I to the caller Adv. Whenever the protocol execution is successful from the issuer's point of view, the oracle adds $(vk_I, (a_i)_i\ [, pk_U^*])$ to Q_{Adv}.

User-issuance oracle $\mathsf{U}_I(pk_U, A, vk_I^*)$**.** This oracle provides the interface to an honest user's $\mathsf{User}_I(sk_U, A, vk_I^*)$ in an adversarily triggered issuance session. If vk_I^* belongs to an honest issuer (being listed in L) the oracle aborts. As above, if pk_U is not in L, the oracle adds fresh $(sk_U, pk_U) \leftarrow \mathsf{Gen}_U(pp)$ to L, and informs adversary about the new pk_U. Whenever the session yields a valid credential C for the user, the oracle adds $(C, vk_I^*, sk_U, sk(C))$ together with a fresh handle h to L, and outputs (h, C, vk_I^*) to the adversary.

Issuance oracle $\mathsf{UI}(pk_U, A, vk_I)$**.** This oracle perfoms a full issuance session between an honest user pk_U and an honest issuer vk_I on the attributes A, logs the resulting credential $(C, vk_I, sk_U, sk(C))$ in L and outputs its handle h and the protocol transcript to the caller. Again, if either pk_U or vk_I are not in L the oracle generates the required identities, adds them to L and returns their new public keys before running the issuance.

User-presentation oracle $\mathsf{U}_P(h, D)$**.** The user-presentation oracle initiates a presentation session for an existing handle h, and provides both interfaces of $\mathsf{User}_P(sk_U, sk(C), D)$,where $(sk_U, sk(C))$ belong to h, to the caller. If the handle h is not listed in L, the oracle aborts.

Eventually the adversary Adv runs a presentation session claiming credentials of some honest, but adversarily chosen vk_I. The experiment is successful if Adv manages to make $\mathsf{Serv}[vk_I]$ accept the presentation but the disclosed attributes $o_S^* = (a_i^*)_{i \in D^*}$ do not correspond to any of the adversary's credentials issued by vk_I, which we denote by $o_S^* \notin Q_{\mathsf{Adv}}|_{D^*}$.

Definition 4. *An EABC system* EABC *is* unforgeable*, if for any* PPT *adversary* Adv *the success probability in the following experiment is bounded by a negligible function in* λ*.*

Unforgeability Experiment $\mathsf{Exp}^{\text{forg}}_{\mathsf{Adv}}(\lambda)$

$pp \leftarrow \mathsf{Par}(\lambda); L = \emptyset; Q_{\mathsf{Adv}} = \emptyset;$
$(vk_I, st) \leftarrow \mathsf{Adv}(pp),$ *with* vk_I *listed in* L
$\langle o^*_U, o^*_S \rangle \leftarrow \langle \mathsf{Adv}(st), \mathsf{Serv}(vk_I, D) \rangle$
if $o^*_U \neq \perp \wedge o^*_S \notin Q_{\mathsf{Adv}}|_{D^*}$ **return** *success* **else return** *failed*

In this experiment, $\mathsf{Adv} = \mathsf{Adv}^{\mathsf{I},\mathsf{U}_I,\mathsf{UI},\mathsf{U}_P}$ *has access to the above defined (interactive) oracles,* o^*_U *denotes the serivce's verdict (ouput to the user-side), and* o^*_S *are the disclosed attributes* $(a^*_i)_{i \in D^*}$ *(output on the service-side).*

Non-Deceivability of Honest Users *Non-deceivability* (of honest users) is the infeasability of successfully altering the presentation goal without being exposed to the honest user. Note that this property is not automatically covered by Definition 4, since such a change of goal might be just between two vk_I-credentials of one and the same user. We formulate this property by means of the *non-deceivability experiment*, which is almost identical to the unforgeability experiment, except that in the last step the adversary Adv opens a presentation session on behalf of an *honest* user for a credential C and index set D chosen by the adversary.

Definition 5. *An EABC system* EABC *is* non-deceivable towards a user*, if for any* PPT *adversary* Adv *the success probability in the following experiment is bounded by a negligible function in* λ*.*

Non-Deceivability Experiment $\mathsf{Exp}^{\text{decv}}_{\mathsf{Adv}}(\lambda)$

$pp \leftarrow \mathsf{Par}(\lambda); L = \emptyset;$
$(h, D, st) \leftarrow \mathsf{Adv}(pp),$ *such that* h *is listed in* L
Let C*,* vk_I*,* sk_U*,* $sk(C)$*, and* $(a_i)_i$ *belong to* h*;*
$\langle o^*_U, o^*_S \rangle \leftarrow \langle \mathsf{User}_P(sk_U, sk(C), D), \mathsf{Adv}(st), \mathsf{Serv}(D, vk_I) \rangle$
if $o^*_U \neq \perp \wedge o^*_S \neq (a_i)_{i \in D}$ **return** *success* **else return** *failed*

As in Defintion 4, $\mathsf{Adv} = \mathsf{Adv}^{\mathsf{I},\mathsf{U}_I,\mathsf{UI},\mathsf{U}_P}$ *has access to the oracles described in Section 3.2,* o^*_U *denotes the serivce's verdict (ouput to the user-side), and* o^*_S *are the disclosed attributes* $(a^*_i)_{i \in D^*}$ *(output on the service-side).*

Privacy for EABC Privacy for EABC systems, as understood by [24] is the indistinguishability of encrypted credentials (C, vk_I) under chosen message attack. Allowing wallet-service collusions under which the attacker may get access to disclosed attributes, privacy needs to embrace *untraceability*, i.e. the infeasability of linking a presentation of an encrypted credential with its issuance session. We capture both properties by a single indistinguishability experiment $\mathsf{Exp}^{\text{priv}}_{\mathsf{Adv}}$, which states that nothing more can be learned from a credential (C, vk_I) (not even its owner pk_U) than what has been disclosed.

The adversary's environment in $\mathsf{Exp}^{\text{priv}}_{\mathsf{Adv}}$ is as in the unforgeability experiment from Section 3.2. That is, the experiment maintains a list L for the public and secret data of all honest participants, and the adversary is given access to the same honest participant oracles I, U_I, UI, U_P. First, the experiment generates the system parameters pp and a (random) subset $D \subseteq \{1, \ldots, n\}$, and lets the

adversary choose one of its issuance keys vk_I^*, two honest (not necessarily different) user identities pk_{U_1}, pk_{U_2} and their queries A_0, A_0 being compliant on D, i.e. $A_1|_D = A_2|_D = (a_i)_{i\in D}$. Then the experiment performs issuance sessions with vk_I^* on A_0 and A_1 (but does not log the resulting credentials C_0 and C_1 in the list L). It chooses a random bit b, tells the adversary C_b and lets Adv play the wallet and the service in a final presentation session for C_b, from which it tries to guess the random bit b.

Definition 6. *An EABC system* EABC *satisfies* privacy*, if for any* PPT *adversary* Adv *the advantage* $\left|\Pr\left[\mathsf{Exp}^{\mathrm{priv}}_{\mathsf{Adv}}(\lambda) = \mathit{success}\right] - \frac{1}{2}\right|$ *in the following experiment is bounded by a negligible function in* λ.

Indistinguishability Experiment $\mathsf{Exp}^{\mathrm{priv}}_{\mathsf{Adv}}(\lambda)$

$pp \leftarrow \mathsf{Par}(\lambda); L = \emptyset; D \leftarrow_{\$} 2^{\{1,\ldots,l_{max}\}};$
$(st, vk_I^*, (pk_{U_0}, A_0), (pk_{U_1}, A_1)) \leftarrow \mathsf{Adv}(pp),$
  with $A_0|_D = A_1|_D$ *and* pk_{U_0}, pk_{U_1} *listed in* L
$\langle (C_i, sk(C_i)), st\rangle \leftarrow \langle \mathsf{User}_I(sk_{U_i}, A_i, vk_I^*), \mathsf{Adv}(st)\rangle, i \in \{0,1\}$
$b \leftarrow_{\$} \{0,1\},$
$b^* \leftarrow \langle \mathsf{User}_P(sk_{U_b}, sk(C_b), D, vk_I^*), \mathsf{Adv}(st, C_b)\rangle$
if $b^* = b$ **return** *success* **else return** *failed*

Again, the adversary $\mathsf{Adv} = \mathsf{Adv}^{\mathsf{I},\mathsf{U}_I,\mathsf{UI},\mathsf{U}_P}$ *is given access to the oracles as described in Section 3.2.*

Unlinkability of Presentations Unlinkability of presentations is the infeasability for a malicious service to link any two presentation sessions with the user or the credentials hidden behind the presentation. Here, the service may collude with issuers (in practice both can be even one and the same entity), but in contrast to the above experiments, the wallet W is assumed to be honest. We express this property by means of the *unlinkability experiment* which essentially is as $\mathsf{Exp}^{\mathrm{priv}}_{\mathsf{Adv}}$ from Section 3.2, but the adversary is not given access to the U_P oracle, and it is forbidden to retrieve any credential C from L. In return it is given access to the following honest wallet oracles:

Wallet oracle $\mathsf{W}[h, D]$ This oracle provides the interfaces to an honest wallet's $\mathsf{Wall}[C, D, vk_I]$, where C and vk_I belong to the handle h listed in L. If the handle does not exist, the oracle aborts.

User-wallet oracle $\mathsf{UW}[h, D]$ This oracle, on input the handle h and index subset D, looks up the corresponding credential C and key material sk_U, $sk = sk(C)$ in L and provides the caller the interfaces to the presentation session $\langle \mathsf{User}_P[sk_U, sk(C), vk_I], \mathsf{Wall}[C, D, vk_I],\ .\ \rangle$. As above, if the handle does not exist the oracle aborts.

Definition 7. *An EABC system* EABC *is* unlinkable*, if for any* PPT *adversary* Adv *the advantage* $\left|\Pr\left[\mathsf{Exp}^{\mathrm{link}}_{\mathsf{Adv}}(\lambda) = \mathit{success}\right] - \frac{1}{2}\right|$ *in the following experiment is bounded by a negligible function in* λ.

Unlinbkability Experiment $\mathsf{Exp}^{\text{link}}_{\mathsf{Adv}}(\lambda)$

$pp \leftarrow \mathsf{Par}(\lambda); L = \emptyset;$
$(st, vk_I^*, (pk_{U_0}, A_0), (pk_{U_1}, A_1), D) \leftarrow \mathsf{Adv}(pp),$
with $A_0|_D = A_1|_D$ *and* pk_{U_0}, pk_{U_1} *listed in* L
$\langle (C_i, sk(C_i)), st\rangle \leftarrow \langle \mathsf{User}_I(sk_{U_i}, A_i, vk_I^*), \mathsf{Adv}(st)\rangle, i \in \{0,1\}$
$b \leftarrow_{\$} \{0,1\};$
$b^* \leftarrow \langle \mathsf{User}_I(sk_{U_i}, sk(C_b), vk_I^*), \mathsf{Wall}(C_b, D, vk_I^*), \mathsf{Adv}(st)\rangle$
if $b^* = b$ **return** *success* **else return** *failed*

In the experiment the adversary $\mathsf{Adv} = \mathsf{Adv}^{\mathsf{I},\mathsf{U}_1,\mathsf{UI},\mathsf{W},\mathsf{UW}}$ *is given access to all the honest-participant oracles from Section 3.2, and the above defined honest wallet oracle* W *and* UW.

4 Instantiating EABCs

We instantiate $\mathsf{EABC} = (\mathsf{Par}, \mathsf{Gen}_I, \mathsf{Gen}_U, \mathsf{User}_I, \mathsf{Iss}, \mathsf{User}_P, \mathsf{Wall}, \mathsf{Serv})$ using the structure-preserving blind signature scheme $\mathsf{BSIG} = \mathsf{BSIG}_{AGHO}$ from Section 2.3, the anonymous re-randomizable proxy re-encryption scheme $\mathsf{PRE} = \mathsf{PRE}_{BBS}$ by Blaze, Bleumer, and Strauss (BBS, cf. Section 2.2), and two non-interactive zero-knowledge proof systems (cf. Section 2.1), one of which is simulation extractable. For notational convenience, we shall refer to both as NIZK without causing confusion.

4.1 System Parameters and Key Generation

Given the security parameter λ, a trusted[4] third party generates the system parameters by $sp \leftarrow \mathsf{Par}(\lambda)$, which internally queries Par_{BSIG} and Par_{PRE} in such a way that the message space for BSIG and the ciphertext space of PRE is the same group $\mathbb{G}$ of prime order q. Furthermore, it uses Gen_{ZKP} to set up a common reference string for the NIZK. Specifically, the system parameters are

$$sp = (L, \mathbb{G}, \mathbb{H}, \mathbb{T}, q, e, G, H, g, crs)\,,$$

whereas L is the maximum number of attributes allowed in a credential, $\mathbb{G}$, $\mathbb{H}$, $\mathbb{T}$ are the AGHO pairing groups of prime order q, with bilinear mapping $e : \mathbb{G} \times \mathbb{H} \longrightarrow \mathbb{T}$ and respective generators G, H, and $e(G,H)$, $g \in \mathbb{G}$ is the generator for the BBS encryption scheme, and crs is the common reference string for the NIZK proof systems. We further assume that all attributes $a \in \mathbb{A}$ are represented by group elements from $\mathbb{G}$.

An issuer I's signing key generated by $\mathsf{Gen}_I(sp) = \mathsf{Gen}_{BSIG}(sp)$ consists of the AGHO keys $sk_I = (v, (w_i)_{i=1}^l, z)$ and $vk_I = (V, (W_i)_{i=1}^l, Z)$, where $l \leq L$, and a user's identity key consists of the BBS keys (sk_U, pk_U) generated by $\mathsf{Gen}_U(sp) = \mathsf{Gen}_{PRE}(sp)$.

4.2 Issuance

The issuance protocol is the restrictive blind AGHO signature (Definition 3) based on the commitment

$$\mathsf{Comm}(sk_U, (a_i)_i) = (c_0, (pk_i, c_i)_i),$$

[4] In practice, the generation of the system parameters can be realized using multi-party techniques.

which embodies the encrypted certificate to be signed, being comprised by U's *certificate pseudonym* $c_0 = \mathsf{Enc}_{BBS}(pk_U, 1)$, and the *attribute encryptions* $c_i = \mathsf{Enc}_{BBS}(pk_i, a_i)$ with respect to fresh proxy re-encryption keys $(pk_i)_i$. Although authentication of U is outside the scope of the blind AGHO scheme, we nevertheless integrate it into the issuance protocol by extending the wellformedness proof of the restrictive scheme by a proof of knowing the secret key belonging to pk_U. The protocol runs over a server-authenticated, confidential and integrity protected channel.

Definition 8 (Issuance Protocol). $\langle \mathsf{User}_I(sid, sk_U, (a_i)_{i=1}^l, vk_I), \mathsf{Iss}(sid, pk_U, (a_i)_{i=1}^l, sk_I) \rangle$ *is a protocol between a user* U *with identity key* (sk_U, pk_U) *and an issuer* I *with AGHO keys* (sk_I, vk_I). *Both user and issuer agreed on the unencrypted content, the attributes* $A = (a_i)_{i=1}^l \in \mathbb{A}^l$, *beforehand.*

1. U *computes* $com = \left(c_0, (pk_i, c_i)_{i=1}^l\right)$ *by generating a fresh pseudonym* $c_0 = (c_{0,1}, c_{0,2}) = \mathsf{Enc}_{BBS}(pk_U, 1)$ *and attribute encryptions* $c_i = (c_{i,1}, c_{i,2}) = \mathsf{Enc}_{BBS}(pk_i, a_i)$ *using a fresh set of attribute keys* $(sk_i, pk_i) \leftarrow \mathsf{Gen}_{BBS}(sp)$, $1 \le i \le l$. *For notational convenience we write* $m = (m_{i,j}) = (pk_i, c_{i,1}, c_{i,2})_{i=0}^l$ *for com, where we set* $(sk_0, pk_0) = (sk_U, 1)$ *and* $a_0 = 1$.
2. *With the above described commitment scheme,* U *engages the restrictive blind signing session with* I *(Definition 3) to receive an* encrypted credential

$$C = \left\{ \left(c_0, (pk_i, c_i)_{i=1}^l \right), \sigma = (R, S, T) \right\},$$

with σ *being a valid AGHO signature by* I *on com, and with* $(sk_i)_{i=0}^l$ *as the opening of com. Demanding additional authentication of the user* U *by means of her identity key* sk_U, *the zero-knowledge proof* π_U *as described in Definition 3, is explicitly described by*

$$\begin{aligned} \pi_U = \mathsf{NIZK}\Big[\big(\eta, \varphi_1, \varphi_2, (\kappa_i, \lambda_i)_{i=0}^l\big) : g^{\kappa_0} = pk_U \\ \wedge\ G_1^{\eta} = G\ \wedge\ G^{\varphi_1} = G_2\ \wedge\ G_1^{\varphi_2} = G_3 \\ \bigwedge_{i=0}^{l} G_1^{\lambda_i} = G^{\kappa_i} \wedge \overline{m}_{i,0} \cdot \overline{P}_{i,0}^{\eta} = g^{\kappa_i} \wedge \overline{m}_{i,2} \cdot \overline{P}_{i,2}^{\eta} = \overline{m}_{i,1}^{\kappa_i} \cdot \overline{P}_{i,1}^{\lambda_i} \cdot a_i \Big], \end{aligned}$$

which is bound to the unique session identifier sid. Here, the user U *chooses* $(\eta, \varphi_1, \varphi_2) = (1/e, f_1, f_2)$, *and* $(k_i, \lambda_i) = (sk_i, {}^{sk_i}/_e)$, $0 \le i \le l$, *as witnesses.*

Remark 1. In some situations a user might be allowed to stay anonymous towards the issuer. In such case the user's public identity pk_U is not part of the commited x and hence not provided to I, the term $g^{\kappa_0} = pk_U$ in π_U is omitted. In another setting similar to [8,11] a user might be known to I under a pseudonym $P_U = \mathsf{Enc}_{BBS}(pk_U, 1)$ of her. Here, U proves to I that the same secret key sk_U is used in both $P_U = (P_{U,1}, P_{U,2})$ and the certificate pseudonym, replacing $g^{\kappa_0} = pk_U$ by $c_{0,1}^{\kappa_0} = c_{0,2}\ \wedge\ P_{U,1}^{\kappa_0} = P_{U,2}$.

4.3 Presentation

The presentation of a credential C, as described in full detail by Definition 10, is essentially based on re-encrypting a re-randomization of C into a selective readable version $\overline{C}$ for the service S, supplemented by two linked zero-knowledge proofs: the computationally costly *presentation proof* π_P, which is performed by the wallet W and which relates the transformed $\overline{C}$ to the original C (the

latter, including its signature is hidden from S), and the *ownership proof* π_O on the pseudonym of $\overline{C}$, proving knowledge of the secret identity key belonging to the pseudonym. The first proof is efficiently instantiated by help of the structure-preservation property of the blind AGHO scheme, the ownership proof supplied by the user is a simple proof of knowledge of a single exponent.

For the sake of readability, we gather the establishment of the service's session keys and its corresponding transformation information in a separate subprotocol, the ReKey protocol. Both Protocols from Defintion 9 and 10 run over a server-authenticated, confidential and integrity protected channel, and are associated with the same random session identifier *sid* supplied by the service. Furthermore, the non-interactive proofs (π_O and π_S below) are bound to the context of the presentation, in particular the common public parameters *sid*, vk_I and D.

Definition 9 (ReKey protocol). *This protocol between the user U and the service S is a subprotocol of the presentation protocol from Defintion 10.*

1. *U chooses a random one-time key $sk' \leftarrow_\$ \mathbb{Z}_q$, and forwards sk' and D to S.*
2. *U re-randomizes[5] her pseudonym c_0 by $e \leftarrow_\$ \mathbb{Z}_q$, $\overline{c}_0 = (\overline{c}_{0,1}, \overline{c}_{0,2}) = \left(c_{0,1}^e, c_{0,2}^e\right)$ and proves to S that she is in posession of its secret key, by supplying a simulation extractable zero-knowledge proof*
$$\pi_O = \mathsf{NIZK}\left[\kappa : \overline{c}_{0,1}^{\kappa} = \overline{c}_{0,2}\right],$$
in which she uses $\kappa = sk_U$ as witness.
3. *S verifies π_O, and if valid it keeps $\overline{c}_0$ and sk'.Otherwise, S aborts the protocol. On the user side, U takes the secret attribute keys $(sk_i)_i$ belonging to C and determines $rk'_0 = {}^1/_e$ and the re-encryption keys $rk'_i = {}^{sk_i}/_{sk'}$, $i \in D$.*

Definition 10 (Presentation Protocol). *The presentation protocol of encrypted ABCs $\left\langle \mathsf{User}_P(sid, sk_U, (sk_i)_{i\in D}) \right.$ is between a user U with identity key sk_U who owns the credential C issued by I, the wallet W, and the service S (the supplier of the random session identifier sid). Here, $D \subseteq \{1, \ldots, l\}$ denotes the index set of the attributes to be disclosed, and $(sk_i)_{i\in D}$ are U's corresponding attribute keys.*

1. *U performs Protocol from Definition 9 with S, and if successful it sends the re-randomized one-time pseudonym $\overline{c}_0$ together with the re-encryption keys rk'_0, $(rk'_i)_{i\in D}$ and D to W.*

   From now on we proceed similar to [24]:
2. *(Randomization and re-encryption) For $i \in D$, the wallet W re-randomizes the ciphertexts c_i to $\overline{c}_i = \left(c_{i,1} \cdot g^{f_i}, c_{i,2} \cdot pk_i^{f_i}\right)$, with $f_i \leftarrow_\$ \mathbb{Z}_q$. All other attributes are randomized inconsistently, by choosing $v_{i,0}$, $v_{i,1}$, $v_{i,2} \leftarrow_\$ \mathbb{Z}_q$ and setting $\overline{pk}_i = pk_i \cdot g^{v_{i,0}}$, $\overline{c}_i = (c_{i,1} \cdot g^{v_{i,1}}, c_{i,2} \cdot g^{v_{i,3}})$ for all $i \notin D$. Using the re-encryption keys $(rk'_i)_{i\in D}$ the wallet translates the attributes belonging to $i \in D$ by $d_i = \mathsf{ReEnc}_{BBS}(rk'_i, \overline{c}_i) = \left(\overline{c}_{i,1}^{rk'_i}, \overline{c}_{i,2}\right)$.*
3. *(Presentation) W randomizes T by $\overline{T} = T^x$, $x \leftarrow_\$ \mathbb{Z}_q$, forwards $(d_i)_{i\in D}$, $\left(\overline{pk}_i, \overline{c}_i\right)_{i\notin D}$, and $\overline{T}$ to S, and provides a wellformedness proof of these elements via*
$$\pi_P = \mathsf{NIZK}\Big[(P, \Sigma, \xi), \eta, (\kappa_i, \gamma_i)_{i\in D}, (\nu_{i,0}, \nu_{i,1}, \nu_{i,2})_{i\notin D} : (1) \wedge (2)\Big],$$
which is defined by the relations (1) and (2) below.

[5] For the sake of efficiency, U might outsource the re-randomization of its pseudonym to the wallet.

4. *S verifies π_P using the verified one-time pseudonym received in Protocol 9. If valid, it decrypts the attributes $(d_i)_{i\in D}$ with its one-time key sk'. (Otherwise S outputs $\perp$).*

Remark 2. Showing more than one credential is efficiently implemented by merging their ownership proofs to a single NIZK which simultaneously proves knowledge of sk_U on all used pseudonyms, $\mathsf{NIZK}\left[(\kappa) : \bigwedge_C \overline{c}_{0,1}(C)^{\kappa} = \overline{c}_{0,2}(C)\right]$.

Equation (1) and (2) mentioned in Protocol 10, state that $(P, \Sigma, \overline{T}^{1/\xi})$ is a valid signature for a quadratic derivative of the above group elements $(\overline{c}_{0,1}, \overline{c}_{0,2})$, $(d_{i,1}, d_{i,2})_{i\in D}$ and $(\overline{pk}_i, \overline{c}_{i,1}, \overline{c}_{i,2})_{i\notin D}$, i.e.

$$\begin{aligned} e(\Sigma, H) \cdot e(\rho, V) \cdot e(\overline{c}_{0,1}, W_{0,1})^{\eta} \cdot e(\overline{c}_{0,2}, W_{0,2})^{\eta} \cdot \\ \cdot \prod_{i\in D} e\left(pk', W_{i,0}\right)^{\kappa_i} \cdot e(g, W_{i,1})^{-\gamma_i \cdot \kappa_i} \cdot e(pk', W_{i,2})^{-\gamma_i} \cdot \\ \cdot \prod_{i\notin D} e\left(g, W_{i,0}\right)^{-\nu_{i,0}} \cdot e(g, W_{i,1})^{-\nu_{i,1}} \cdot e(g, W_{i,2})^{-\nu_{i,2}} \; = \\ = e(G, Z) \cdot \prod_{i\in D} e(d_{i,1}, W_{i,1})^{-1} \cdot e(d_{i,2}, W_{i,2})^{-1} \cdot \\ \prod_{i\notin D} e(\overline{pk}_i, W_{i,0})^{-1} \cdot e(\overline{c}_{i,1}, W_{i,1})^{-1} \cdot e(\overline{c}_{i,2}, W_{i,2})^{-1}, \quad (1) \end{aligned}$$

where V, Z, and $(W_{i,0}, W_{i,1}, W_{i,2})_{i=0}^{l}$ are the components of the issuers verification key, and

$$e(P, \overline{T}) \cdot e(G, H)^{-\xi} = 1. \quad (2)$$

Linearization of the quadratic terms in (1) is accomplished by standard techniques and given in Appendix A.3. An honest prover chooses $(P, \Sigma, \xi) = (R, S, x)$, and uses the parameters from step 2 of Protocol 10, i.e. $\eta = rk'_0$, $(\kappa_i, \gamma_i) = (1/rk'_i, rk'_i \cdot f_i)$ for $i \in D$, and $(\nu_{i,0}, \nu_{i,1}, \nu_{i,2}) = (v_{i,0}, v_{i,1}, v_{i,2})$ for all $i \notin D$.

Theorem 3. *Suppose that the AGHO signature scheme is EUF-CMA secure, and that the* NIZK *from Definition 9 is simulation extractable. Then, under the DDH-assumption in $\mathbb{G}$,*

1. *the proxy re-encryption scheme PRE_{BBS} is PRE-IND-CPA secure, anonymous, and has the ciphertext re-randomization property,*
2. *the structure-preserving blind signature scheme BSIG_{AGHO} is unforgeable and has the blinding property,*

hence our EABC system satisfies unforgeability*,* non-deceivability*,* privacy *and* unlinkability *in the sense of Section 3.2.*

The proof of Theorem 3 is given in Appendix A.2.

5 Conclusions

In this paper, we pointed out a problem in Krenn et al.'s modeling of cloud-based attribute-based credential system [24] by presenting a simple and efficient attack allowing the wallet to recover

a user's personal attributes in the real world. We then provided a revised model and a provably secure construction which not only solves this issue, but also reduces the trust assumptions stated in [24] with regards to collusions between the central wallet and other entities in the system. As a building block of potentially independent interest we presented a blind variant of the Abe et al. structure-preserving signature scheme [2].

While we did not provide a concrete implementation of our construction, we expect only very minor performance drawbacks with respect to [24], while correcting all the deficiencies in their work. There, for a security parameter of $\lambda = 112$, all computations on all parties' sides were between 50ms and 440ms when presenting 12 out of 25 attributes. By inspecting the computational efforts needed in our protocol and theirs, one can see only negligible differences, except for the proof of knowledge of a single exponent which is required on the user's side in our construction. However, such computations are efficiently doable, and thus our protocol still provides a significant performance improvement compared to fully locally hosted "conventional" attribute-based credential systems. Finally, we leave a full-fledged implementation, not only of the cryptographic algorithms but of the full system, as open work to demonstrate the real-world applicability of EABCs in general and our construction in particular, and to help ABC systems to finally pave their way into the real world. For this, several approaches can be envisioned, in particular for the presentation protocol, where the optimal choice may depend on external constraints as well as requirements of the specific application domain. Firstly, using the wallet also as a communication proxy, would not require further network anonymisation layers, yet leak metadata to the wallet. Alternatively, by merely outsourcing the computational effort to the wallet and routing the traffic through the user, one could reach the same privacy guarantees as in conventional systems, at the cost of increased bandwidth requirements compared to the first approach; furthermore, the responsibility of transport layer anonymity would be with the user. Finally, an approach close to OpenID Connect could be achieved by combining these two approaches.

Acknowledgements. The first author was partly supported by the "Embedded Lab Vienna for IoT & Security" (ELVIS), funded by the City of Vienna. The second author has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 830929 ("CyberSec4Europe").

References

1. Abe, M., Fuchsbauer, G., Groth, J., Haralambiev, K., Ohkubo, M.: Structure-preserving signatures and commitments to group elements. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 209–236. Springer (2010)
2. Abe, M., Groth, J., Haralambiev, K., Ohkubo, M.: Optimal structure-preserving signatures in asymmetric bilinear groups. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 649–666. Springer (2011)
3. Au, M.H., Susilo, W., Mu, Y.: Constant-size dynamic k-times anonymous authentication. In: De Prisco, R., Yung, M. (eds.) Security and Cryptography for Networks. pp. 111–125. Springer (2006)
4. Barki, A., Brunet, S., Desmoulins, N., Traoré, J.: Improved algebraic MACs and practical keyed-verification anonymous credentials. In: Avanzi, R., Heys, H. (eds.) Selected Areas in Cryptography – SAC 2016. pp. 360–380. Springer (2017)
5. Bellare, M., Boldyreva, A., Micali, S.: Public-key encryption in a multi-user setting: Security proofs and improvements. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 259–274. Springer (2000)

6. Blaze, M., Bleumer, G., Strauss, M.: Divertible protocols and atomic proxy cryptography. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 127–144. Springer (1998)
7. Brands, S.: Untraceable off-line cash in wallets with observers (extended abstract). In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 302–318. Springer (1993)
8. Brands, S.: Rethinking Public Key Infrastructure and Digital Certificates – Building in Privacy. Ph.D. thesis, Eindhoven Institute of Technology (1999)
9. Camenisch, J., Dubovitskaya, M., Haralambiev, K., Kohlweiss, M.: Composable and modular anonymous credentials: Definitions and practical constructions. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT, Part II. LNCS, vol. 9453, pp. 262–288. Springer (2015)
10. Camenisch, J., Groß, T.: Efficient attributes for anonymous credentials. In: CCS '08. pp. 345–356. ACM (2008)
11. Camenisch, J., Herreweghen, E.V.: Design and implementation of the *idemix* anonymous credential system. In: Atluri, V. (ed.) ACM CCS 2002. pp. 21–30. ACM (2002)
12. Camenisch, J., Lysyanskaya, A.: An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 93–118. Springer (2001)
13. Camenisch, J., Lysyanskaya, A.: A signature scheme with efficient protocols. In: Cimato, S., Galdi, C., Persiano, G. (eds.) SCN 2002. LNCS, vol. 2576, pp. 268–289. Springer (2002)
14. Camenisch, J., Lysyanskaya, A.: Signature schemes and anonymous credentials from bilinear maps. In: Franklin, M.K. (ed.) CRYPTO. LNCS, vol. 3152, pp. 56–72. Springer (2004)
15. Chaum, D.: Untraceable electronic mail, return addresses, and digital pseudonyms. Commun. ACM **24**, 84–88 (1981)
16. Chaum, D.: Security without identification: Transaction systems to make big brother obsolete. Commun. ACM **28**, 1030–1044 (1985)
17. Chen, L., Urian, R.: Daa-a: Direct anonymous attestation with attributes. In: Conti, M., Schunter, M., Askoxylakis, I. (eds.) Trust and Trustworthy Computing. pp. 228–245. Springer (2015)
18. Chen, X., Zhang, F., Mu, Y., Susilo, W.: Efficient provably secure restrictive partially blind signatures from bilinear pairings. In: Crescenzo, G.D., Rubin, A.D. (eds.) FC 2006. LNCS, vol. 4107, pp. 251–265. Springer (2006)
19. Fuchsbauer, G., Hanser, C., Slamanig, D.: Practical round-optimal blind signatures in the standard model. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9216, pp. 233–253. Springer (1993)
20. Fuchsbauer, G., Hanser, C., Slamanig, D.: Structure-preserving signatures on equivalence classes and constant-size anonymous credentials. J. Cryptology (2018)
21. Gamal, T.E.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: CRYPTO. Lecture Notes in Computer Science, vol. 196, pp. 10–18. Springer (1984)
22. Groth, J.: Simulation-sound nizk proofs for a practical language and constant size group signatures. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 444–459. Springer (2006)
23. Haböck, U., Krenn, S.: Breaking and Fixing Anonymous Credentials for the Cloud. In: Mu, Y., Deng, R.H., Huang, X. (eds.) CANS '19. pp. 249–269. Springer (2019)
24. Krenn, S., Lorünser, T., Salzer, A., Striecks, C.: Towards attribute-based credentials in the cloud. In: Capkun, S., Chow, S.S.M. (eds.) CANS 2017. LNCS, vol. 11261, pp. 179–202. Springer (2017)
25. Maitland, G., Boyd, C.: A provably secure restrictive partially blind signature scheme. In: Naccache, D., Paillier, P. (eds.) PKC 2002. LNCS, vol. 2274, pp. 99–114. Springer (2002)
26. Paquin, C., Zaverucha, G.: U-prove cryptographic specification v1.1 (revision 2). Tech. rep., Microsoft Corporation (April 2013)
27. Ringers, S., Verheul, E.R., Hoepman, J.: An efficient self-blindable attribute-based credential scheme. In: Kiayias, A. (ed.) FC 2017. LNCS, vol. 10322, pp. 3–20. Springer (2017)
28. Stadler, M.: Publicly verifiable secret sharing. In: Maurer, U. (ed.) EUROCRYPT '96. LNCS, vol. 1070, pp. 190–199. Springer (1996)
29. Yang, R., Au, M.H., Xu, Q., Yu, Z.: Decentralized blacklistable anonymous credentials with reputation. In: Susilo, W., Yang, G. (eds.) ACISP 2018. LNCS, vol. 10946, pp. 720–738. Springer (2018)

A Supplementary Material

A.1 Proof of Theorem 2

In the proof of the blindness property of our restrictive blind AGHO scheme, we shall make use of the following well-known self-reducibility of the DDH problem, which goes back to [28] and has been extended in [5] to justify secure randomness re-use in mutli-recipient ElGamal encryption.

Lemma 1 ([5]). *Let $\{G_\lambda\}_{\lambda\in\mathbb{N}}$ be a family of groups for which the DDH assumption hold, and $l = l(\lambda) \in \mathbb{N}$ some polynomial in λ. Then the same assumption holds for the decisional problem of 'joint' DH triplets $(X_i, Y_i, Z_i)_{i=1}^l$ with $X_1 = \ldots = X_l$. That is, for any* PPT *algorithm* Adv*, the advantage*

$$\Big| P\left[\mathsf{Adv}\left(g, (g^a, g^{b_i}, g^{a\cdot b_i})_{i=1}^l\right) = \mathit{true}\right] \quad - \quad P\left[\mathsf{Adv}\left(g, (g^a, g^{b_i}, g^{c_i})_{i=1}^l\right) = \mathit{true}\right] \Big|$$

is bounded by some negligible function in λ, where the probabilities are taken over all random choices $g \leftarrow_\$ G$, $a \leftarrow_\$ \mathbb{Z}_q$, $b_i, c_i \leftarrow_\$ \mathbb{Z}_q$, and all random coins of Adv*.*

For the sake of completeness we give a proof of Lemma 1.

Proof. We construct a randomized reduction from the single DDH problem to that of 'joint' triplets, using a standard re-randomization technique.

For any single triplet $(X, Y, Z) = (g^a, g^b, g^c) \in G_\lambda^3$, consider its polynomially many re-randomizations

$$\begin{aligned}(X', Y_i', Z_i')_{i=1}^l &= (X, Y^{e_i} \cdot g^{u_i}, Z^{e_i} \cdot X^{u_i})_{i=1}^l = \\ &= \left(g^a, g^{b\cdot e_i + u_i}, g^{c\cdot e_i + a\cdot u_i}\right)_{i=1}^l,\end{aligned}$$

by sampling $e_i, u_i \leftarrow_\$ \mathbb{Z}_q$, $1 \le i \le l(\lambda)$. If $a \cdot b = c$, then each linear mapping Λ_i which sends (e_i, u_i) to $(b \cdot e_i + u_i, c \cdot e_i + a \cdot u_i)$, is q–to–1. Hence our re-randomization results in a uniform selection of l-tuples of joint DH triplets with $X = g^a$. On the other hand, if $a \cdot b \neq c$, then Λ_i is bijective and therefore induces a uniform distribution on the set of arbitrary triplets $(X, Y_i, Z_i)_{i=1}^l$ with $X = g^a$.

Now any PPT distinguisher applied to the randomized l-tuples yields a distinguisher for the single triplet DDH in G. By the DDH assumption on G the advantage of Adv is negligible.

Blindness. To show blindess of the restrictive blind AGHO scheme, suppose that the DDH assumption holds in $\mathbb{G}$. Starting with the blindness experiment implicitly described by Definition 2, and gradually changing the actions on the user side of the signing protocol (Definition 3), we give a sequence of indistinguishable games for the adversary Adv, which eventually ends up at a game characterizing the hiding property of COM.

Game 1 is the experiment implicitly defined by Definition 2 in which Adv is successful, whenever its guess b^* is correct.

Game 2 is as Game 1, except that we use the extractability property of the second NIZK to obtain $sk^* = (v, (w_i), z)$ belonging to vk^* from a valid π_S, skip the user's deblinding operations in Step 3 of Definition 3 and directly output $\sigma = (\overline{R}^f, \overline{R}^{vf} \cdot G^z \cdot \prod_{j=1}^l m_j^{w_j}, \overline{T}^{\frac{1}{f}})$ instead.

Game 3: Based on Game 2 we first perform the following syntactical change. Instead of generating (G_1, G_2, G_3), $\overline{m}$, $\overline{P}$ as in Step 1 of Protocol 3, we query an external random source of $(l+1)$-tuples $(X, Y_i, Z_i)_{i=0}^{l}$ of joint DH triplets with $X \neq 1_{\mathbb{G}}$, choose $f \leftarrow_{\$} \mathbb{Z}_q^*$ and set

$$\begin{aligned}(G_1, G_2, G_3) &= (X, G^f \cdot Y_0^{-1}, Z_0),\\ \overline{m} &= m \cdot (Y_i^{-1})_{i=1}^{l},\\ \overline{P} &= (Z_i)_{i=1}^{l}.\end{aligned}$$

Then, as we do not know the exponents of G_1, G_2, G_3, we use the zero-knowledge property of the first NIZK to simulate a valid proof π_U on input (G_1, G_2, G_3), $\overline{m}$ and $\overline{P}$.

Game 4 is as Game 3, but we replace the external random source of joint DH triplets (satisfying $X \neq 1$) by one as stated by Lemma 1 (i.e., with no restriction on X). Since the distributions the random sources are statistically close, this change is computationally indistinguishable for the adversary.

In *Game 5* we finally substitute the source of DH triplets in Game 4 by one of arbitrary random triplets (sharing the same X-coordinate). By the DH assumption on $\mathbb{G}$ and Lemma 1, the adversary's view in Game 5 is computationally indistinguishable from that of Game 4.

Note that in Game 5 the output σ is either $\perp$ or a valid signature of the queried message m with signature randomness $\overline{R}^f$ entirely independent from the adversary's view (note that a valid signature implies that $\overline{R} \neq 1_{\mathbb{G}}$). Moreover, since (G_1, G_2, G_3), $\overline{m}$, $\overline{P}$ hides the queried message perfectly, together with the witness indistinguishability of π_U, we reason that the success probability of guessing b in Game 5 is equal to that of distinguishing two commitments com_b, com_{1-b} without their openings. By the hiding property of COM, the latter probability is negligible.

Altogether, we conclude that the success probability of Adv in all other games (and in particular in Game 1) is negligible, too, proving blindness.

Unforgeability. First of all, note that in Definition 3 the exponents (e, f_1, f_2) are uniquely determined by (G_1, G_2, G_3), and $\overline{\sigma} = (\overline{R}, \overline{S}_1, \overline{S}_2, \overline{T})$ as returned by an honest signer is uniformly distributed on the relation

$$\Big\{(\overline{R}, \overline{S}_1, \overline{S}_2, \overline{T}) \in \mathbb{G}^* \times \mathbb{G} \times \mathbb{G} \times \mathbb{H}^* : \\ \mathsf{Vf}\left(vk, \overline{m} \cdot \overline{P}^{1/e}, \left(\overline{R}^f, \overline{S}_1 \cdot \overline{S}_2^{1/e}, \overline{T}^{1/f}\right)\right) = 1\Big\}.$$

This fact follows immediately from the signer's choice of r and the random decomposition of $z = z_1 + z_2$. Hence, once given (e, f_1, f_2), we may (perfectly) simulate $\overline{\sigma}$ by calling an ordinary AGHO signing oracle to obtain a valid signature $\sigma = (R, S, T)$ on $m = \overline{m} \cdot \overline{P}^{1/e}$, and compute $\overline{\sigma} = \left(R^{1/f}, S_1, \left(S \cdot S_1^{-1}\right)^e, T^f\right)$ using $f = f_1 + f_2$ and a randomly sampled $S_1 \leftarrow_{\$} \mathbb{G}$.

Now, to prove unforgeability, we consider the following sequence of indistinguishable games a PPT adversary Adv:

Game 1 is the experiment implicitly described by Definition 1, in which Adv is successful, whenever it produces more valid signatures than queried under the adversarily chosen restriction y^*.

Game 2 is as Game 1, except that using the witnesses to produce π_S in Step 3 of Definition 3, we use the zero-knowledge property of the second NIZK to simulate a valid π_S on input $\overline{\sigma} = (\overline{R}, \overline{S}_1, \overline{S}_2, \overline{T})$ instead.

Game 3 is based on Game 2, but we replace the generation of $\overline{\sigma}$ in Step 3 of Definition 3 as follows: we use the extractability of the first NIZK to obtain the witnesses (e, f_1, f_2) and (m, w) with $\mathsf{Vf}_{COM}(m, (x, w)) = 1$ from a valid π_U, determine $\overline{\sigma}$ using an ordinary AGHO signing oracle as described above, and attach w, (m, σ) to the entry x in Q.

Game 3 is a PPT algorithm which queries an ordinary AGHO signing oracle on a commitment m on the value x of the blind signing session. By strong unforgeability of the AGHO signature (Theorem 1) we have that, with overwhelming probability every adversarily derived (m_i^*, σ_i^*) matches (exactly[6]) one of the entries in Q. Moreover, by the binding property of COM, the probability that x^* does not equal x of its matching entry in Q is negligible. In other words, the success probability in Game 3 is negligible in λ.

Since all changes between the above games are indistinguishable for Adv, we have proved strong unforgeability for the restrictive blind AGHO scheme in the generic group model.

A.2 Proof of Theorem 3

The security properties of the building blocks of the EABC system follow from Section 2: Statement 1 summarizes Proposition 1, 2 and 3, and statement 2 follows from Theorem 3, as commitment by BBS encryption is perfectly binding and computationally hiding under the DDH assumption in $\mathbb{G}$. Based on those properties we show unforgeability, non-deceivability, privacy and unlinkability of the EABC system.

As in Appendix A.1, we assume that the reader is familiar with the standard arguments involving zero-knowledge proofs, hence we omit the technical details whenever using NIZK properties such as zero-knowledgeness, soundness, or (simulation-sound) extractrability.

Unforgeability. Since our adversary model allows arbitrary wallet-service collusions, an attacker is trivially able to retrieve the secret keys of the attributes once disclosed to him, which in the worst case is all attribute keys of all credentials of a user. Therefore, unforgeability of presentations is tied to the unforgeability of their ownership proofs, which involve the only secret assumedly unknown to the attacker. To prove unforgeability, we gradually turn a PPT presentation forger Adv from $\mathsf{Exp}^{\mathrm{forg}}_{\mathsf{Adv}}$ into a solver for the discrete logarithm problem in $\mathbb{G}$, which succeeds on one out of polynomially many random instances.

Game 1 is $\mathsf{Exp}^{\mathrm{forg}}_{\mathsf{Adv}}$ from Definition 4, in which Adv is successful whenever it is able to serve an honest service a valid presentation $\pi^* = (\overline{c}_0^*, (d_i^*)_{i\in D^*}, (\overline{pk}_i^*, \overline{c}_i^*), \overline{T}^*, \pi_P^*, \pi_O^*)$ together with adversarily chosen sk^*, and vk_I from the set of honestly generated issuer keys, for which $(\mathsf{Dec}_{BBS}(sk^*, d_i^*))_{i\in D^*} = (a_i^*)_{i\in D^*}$ does not match any of the queries listed in Q.

Game 2 is based on Game 1, but we replace the generation of a new honest user's identity keys by some arbitrary element $pk_U = g_U \in \mathbb{G}$ supplied by an external, uniformly distributed random source $\mathcal{U}_{\mathbb{G}}$. Not knowing the corresponding secret keys, i.e. the exponent of g_U with respect to g, we use the zero-knowledge property of the ownership NIZK to simulate valid ownership proofs

[6] By the probabilistic nature of the AGHO scheme, the signatures returned by the signing oracle are overwhelmingly all different.

inside the user-presentation oracle U_P (as well as π_U in U_I and UI whenever authentication of the user is demanded). As such change induces a computationally indistinguishable distribution of ownership proofs, the success probability of Game 2 differs only negligibly from that in Game 1.

Game 3 is as Game 2, whereas we use the (simulation) extractability of the two linked NIZKs to retrieve from π^* an encrypted credential C from vk_I which decrypts to the attributes $(a_i^*)_{i\in D^*}$, together with the secret key sk for the pseudonym used in C. Again, the success probabilities of Game 3 and Game 2 differ only by a negligible amount.

By the unforgeability of the restrictive blind AGHO scheme, if $(a_i^*)_{i\in D^*}$ does not match any of the adversary's queries, then C must belong to one of the honest user's pk_U. In other words, the derived sk is the discrete logarithm of one of the polynomially many $pk_U = g_U$ queried from $\mathcal{U}_\mathbb{G}$ during the experiment. By the DDH assumption on $\mathbb{G}$ the success probability in Game 3 is negligible. Since the changes between the above games are compuationally indistinguishable, the same follows for Game 1, proving unforgeability of the EABC system.

Non-deceivability. The intuition here is as follows: Since Adv may get to know all attribute keys of a user (as discussed above), non-deceivability is tied to the infeasability of mapping an honest user's one-time pseudonym onto another of her credentials. Technically, we instrument a successful Adv in $\mathsf{Exp}^{\text{decv}}_{\mathsf{Adv}}$ to obtain a solver for the following type of discrete logarithm problem:
Given polynomially many uniformly chosen random elements $(g_i)_{i=1}^{p(\lambda)}$ *from* $\mathbb{G}$*, find an exponent* e *such that* $g_i^e = g_j$ *for some* $i \neq j$*.*
By the DDH assumption on $\mathbb{G}$, the success probability of such a solver is negligible.

Game 1 is $\mathsf{Exp}^{\text{decv}}_{\mathsf{Adv}}$ from Definition 5, where Adv is successful whenever it is able to change the intended attributes $(a_i)_{i\in D}$ in the interaction with an honest user's $\mathsf{U}_P(h, D)$ and an honest service's $\mathsf{Serv}(D, vk_I)$. That is, it is able to generate $\pi^* = (\bar{c}_0^*, (d_i^*)_{i\in D}, (\overline{pk}_i^*, \bar{c}_i^*)_{i\notin D}, \overline{T}^*)$ and a valid π_P^* for it, which is compliant with the user's one-time-pseudonym $\bar{c}_0$ for $C = h(C)$, but $(a_i^*)_{i\in D} = (\mathsf{Dec}_{BBS}(sk', d_i^*))_{i\in D} \neq (a_i)_{i\in D}$.

Game 2 is as Game 1, but with the following syntactical change in the issuance oracles U_I and UI: We generate $c_0 = (g_C, g_C^{\mathsf{sk}_U})$ by quering an external source $\mathcal{U}_\mathbb{G}$ of uniformly distributed random elements g_C from $\mathbb{G}$.

Game 3 is as Game 2, but whenever Adv yields a succesful presentation we use the extractability of the NIZK obtain from π^* a valid credential $C' = \{(c_0', (pk_i', c_i')_i), \sigma'\} \neq C$ which incorporates the attribute encryptions of $(a_i^*)_{i\in D}$, and an exponent e' which relates the honest user's $\bar{c}_0$ to the one in C' by $c_0' = \bar{c}_0^{e'}$. With rk_0' supplied by the user, the pseudonyms of the C and C' are related by $c_0' = c_0^e$ with $e = {}^{e'}/_{rk_0'}$.

By unforgeability of the blind AGHO scheme (Theorem 2) we conclude that with overwhelming probability C' as derived from the presentation in Game 3 is one of the other regularily issued credentials, and hence it incorporates another pseudonym $(g_{C'}, g_{C'}^{sk_U})$ from U. Thus Game 3 is a PPT algorithm which queries $\mathcal{U}_\mathbb{G}$ polynomially often and eventually produces an exponent e which solves the discrete logarithm problem between two of the random elements from $\mathcal{U}_\mathbb{G}$. As stated above, the success probability of such solver is negligible, and so is that of Game 3.

Since all changes between the above games are compuationally indistinguishable, the success probability in Game 1 is negligible, too, proving non-deceivability.

Privacy Note that $\mathsf{Exp}^{\mathrm{priv}}_{\mathsf{Adv}}$ from Definition 6 paraphrases the blindness experiment from Definition 2 with $x_0 = (sk_{U_0}, A_0)$ and $x_1 = (sk_{U_1}, A_1)$, under the strengthening that Adv (again, by collusion) might retrieve the attribute keys $(sk_i)_{i\in D}$ of the subset D for which $A_0|_D = A_1|_D$. Nevertheless, it can be seen directly from the proof given in Appendix A.1 that blindness holds even if the adversary is given access to 'partial openings' of the commmitment, i.e. $(sk_i)_{i\in D}$. With this observation, privacy follows from blindness of the restrictive blind AGHO scheme.

Unlinkability Recall that besides π_P and π_O, a presentation of C is comprised of the one-time pseudonym $\bar{c}_0$, the re-randomized re-encryptions $(d_i)_{i\in D}$ of the disclosed attributes $(a_i)_{i\in D}$, and the randomized elements $(\overline{p}k_i, \bar{c}_j)_{i\notin D}$, and $\overline{T}$.

Game 1 is $\mathsf{Exp}^{\mathrm{link}}_{\mathsf{Adv}}(\lambda)$ from Definition 7 which yields success if the adversary Adv is able to guess (pk_{U_b}, C_b) behind the presentation session.

Game 2 and *Game 3* are based on Game 1, where we use the zero-knowledge property of both NIZK to gradually replace π_O and π_S by simulations on $\bar{c}_0$, and $\bar{c}_0, (d_i)_{i\in D}, (\overline{p}k_i, \bar{c}_i)_{i\notin D}, \overline{T}$.

Game 4 is as Game 3, but now we finally replace $\bar{c}_0$, $(d_i)_{i\in D}$ by fresh attribute encryptions $\mathsf{Enc}_{BBS}(pk_{U_b}, 1)$, $(\mathsf{Enc}_{BBS}(sk', a_i))_{i\in D}$, and $(\overline{p}k_i, \bar{c}_j)_{i\notin D}$, $\overline{T}$ by arbitrary random elements. By the re-randomization property of the BBS scheme together with the witness hiding property of both NIZK, the resulting presentation is computationally indistinguishable from that in Game 3.

Note that Game 4 paraphrases the experiment from Proposition 3, in which the adversary tries to guess the random bit b from $\mathsf{Enc}_{BBS}(pk_{U_b}, 1)$, $b = 0, 1$. By anonymity of the BBS scheme, which holds under the DDH assumption on $\mathbb{G}$, the success probability in Game 4 is negligible.

As all changes between the above games are computationally indistinguishable, the success probabyility in Game 1 is negligible, too. This shows unlinkability, and the proof of Theorem 3 is complete.

A.3 Linearization of the Presentation Proof

Linearization of (1) is accomplished by supplying S with Pedersen commitments[7] $k_i = g_p^{1/r k_i} \cdot h_p^{r_i}$, $r_i \leftarrow\!\!\$\, \mathbb{Z}_q$, on the witnesses for $\kappa_i = 1/r k_i$, $i \in D$, introducing extra variables φ_i for the quadratic terms $\kappa_i \cdot \gamma_i$, $i \in D$, and encoding their relation $\varphi_i = \kappa_i \cdot \gamma_i$ by

$$g_p^{\kappa_i} \cdot h_p^{\rho_i} = k_i, \tag{3}$$

$$k_i^{\gamma_i} \cdot g_p^{-\varphi_i} \cdot h_p^{-\delta_i} = 1, \tag{4}$$

with ρ_i and δ_i as auxiliary variables. Since Pedersen commitments are perfectly hiding and computationally binding under the DDH assumption in $\mathbb{G}$, we may replace π_P by

$$\pi'_P = \mathsf{NIZK}\big[(P, \Sigma, \xi), \eta, (\kappa_i, \gamma_i, \varphi_i, \rho_i, \delta_i)_{i\in D}, (\nu_{i,1}, \nu_{i,2}, \nu_{i,3})_{i\notin D} : (5) \wedge (2) \bigwedge_{i\in D} ((3) \wedge (4))\big],$$

[7] The setup of the commitment scheme needs to be integrated in the generation of the system parameters.

where

$$\begin{aligned} e(\sigma, H) \cdot e(\rho, V) \cdot e(\overline{c}_{0,1}, W_{0,1})^{\eta} \cdot e(\overline{c}_{0,2}, W_{0,2})^{\eta} \cdot \\ \cdot \prod_{i \in D} e\left(\mathsf{pk}', W_{i,0}\right)^{\kappa_i} \cdot e(g, W_{i,1})^{-\varphi_i} \cdot e(\mathsf{pk}', W_{i,2})^{-\gamma_i} \cdot \\ \cdot \prod_{i \notin D} e\left(g, W_{i,0}\right)^{-\nu_{i,0}} \cdot e(g, W_{i,1})^{-\nu_{i,1}} \cdot e(g, W_{i,2})^{-\nu_{i,2}} \; = \\ = e(G, Z) \cdot \prod_{i \in D} e(d_{i,1}, W_{i,1})^{-1} \cdot e(d_{i,2}, W_{i,2})^{-1} \cdot \\ \cdot \prod_{i \notin D} e(\overline{\mathsf{pk}}_i, W_{i,0})^{-1} \cdot e(\overline{c}_{i,1}, W_{i,1})^{-1} \cdot e(\overline{c}_{i,2}, W_{i,2})^{-1}. \quad (5) \end{aligned}$$

An honest prover chooses as witnesses $(P, \Sigma, \xi) = (R, S, x)$, $\eta = rk'_0$, $(\kappa_i, \gamma_i) = (1/rk'_i, rk'_i \cdot f_i)$ and $(\nu_{i,1}, \nu_{i,2}, \nu_{i,3}) = (v_{i,1}, v_{i,2}, v_{i,3})$ as before, and $(\varphi_i, \rho_i, \delta_i) = (f_i, r_i, rk'_i \cdot f_i \cdot r_i)$ for the new variables $i \in D$.