An efficient self-blindable attribute-based credential scheme

Sietse Ringers, Eric Verheul, and Jaap-Henk Hoepman

Radboud University, Nijmegen, The Netherlands
{sringers,e.verheul,jhh}@cs.ru.nl

Abstract. An attribute-based credential scheme allows a user, given a set of attributes, to prove ownership of these attributes to a verifier, voluntarily disclosing some of them while keeping the others secret. A number of such schemes exist, of which some additionally provide unlinkability: that is, when the same attributes were disclosed in two transactions, it is not possible to tell if one and the same or two different credentials were involved. Recently full-fledged implementations of such schemes on smart cards have emerged; however, these need to compromise the security level to achieve reasonable transaction speeds. In this paper we present a new unlinkable attribute-based credential scheme with a full security proof, using a known hardness assumption in the standard model. Defined on elliptic curves, the scheme involves bilinear pairings but only on the verifier's side, making it very efficient both in terms of speed and size on the user's side.

Keywords: attribute-based credentials, unlinkable, self-blindable, elliptic curves, bilinear pairings

1 Introduction

An attribute-based credential (ABC) scheme allows a user, given a set of attributes $k_1, \ldots, k_n$, to prove ownership of these attributes to a verifier, voluntarily disclosing some of them while keeping the others secret. A number of such credential schemes exist, of which some additionally provide *unlinkability*: that is, when reusing a credential the verifier cannot tell whether two transactions did or did not originate from the same user (assuming the same attributes with the same values were disclosed in both transactions). This allows for very flexible identity management schemes, that are simultaneously very secure and privacy-friendly.

Two well-known ABC schemes are Idemix [12,23] and U-Prove [10,28]. However, to date there is no provably secure scheme that is sufficiently efficient to allow truly secure implementations on smart cards, while also providing unlinkability of transactions. For example, since Idemix is based on the strong RSA-problem, one would want the keysize to be at least 2048 bits and preferably

even 4096 bits; the IRMA project[1] has implemented Idemix on smart cards using 1024 bits. On the other hand, U-Prove is more efficient but does not provide unlinkability; in addition, its security is not fully proven.

In this paper, we provide a new provably secure, efficient and unlinkable attribute-based credential scheme, that is based on the concept of *self-blindability* [32]: before showing the credential, it is randomly modified into a new one (containing the same attributes) that is still valid. This results in a showing protocol in which the verifier learns nothing at all about the credential besides the attributes that are disclosed (and the fact that the credential is valid). In fact, the showing protocol is a zero-knowledge proof of knowledge. The scheme does not rely on the random oracle model (although usage of this model can lead to a performance increase through the Fiat-Shamir heuristic [17]), and it uses elliptic curves and bilinear pairings, allowing the same security level as RSA-type groups at much smaller key sizes. Although computing a pairing is a much more expensive operation than performing exponentiations on an elliptic curve, all pairings occur on the verifier's side. In addition, the kinds of pairing that we use (Type 3) involves two distinct groups of which one is more expensive to do computations on. However, the user only needs to perform computations on the cheaper of the two. These two facts ensure that the amount of work that the user has to perform is minimal.

The unforgeability of our credential scheme will be implied by the LRSW assumption [13,26,25] introduced by Lysyanskaya et al., and used in many subsequent works (for example, [13,35,34,11,1]). Actually, for our purposes a weaker (in particular, non-interactive and thus falsifiable [27]) version of this assumption called the whLRSW assumption [35] will suffice. After having defined attribute-based credential schemes as well as unforgeability and unlinkability in the next section, we will discuss these assumptions in Section 3. In the same section we will introduce a signature scheme on the space of attributes, that will serve as the basis for our credential scheme. In Section 4 we turn to our credential scheme, defining issuing and showing protocols, and proving that these provide unlinkability and unforgeability for our scheme. This in turn implies the unforgeability of the signature scheme. In Section 5 we will discuss the performance of our scheme, by counting the amount of exponentiations that the user has to perform and by showing average runtimes of an implementation of our scheme. First, we briefly review and compare a number of other attribute-based credential schemes, in terms of features, efficiency and speed, and security.

1.1 Related work

The Idemix credential scheme [12,23] by Camenisch and Lysyanskaya is probably the most well-known unlinkable attribute-based credential scheme, relying on the difficulty of the strong RSA problem in the group of integers modulo an RSA modulus $n = pq$, of recommended size at least 2048 bits. Although this credential scheme has a lot of desirable properties (it is provably unlinkable and

[1] `https://www.irmacard.org`

unforgeable, and the length of the signatures does not depend on the amount of attributes), the large size of the modulus means that, when implementing the user on smart cards, it is difficult to get acceptable running times for the protocols. For example, in [33] the Idemix showing protocol has been implemented with 4 attributes and n around 1024 bits (while n should really be at least 2048 bits); there the running time for the ShowCredential protocol ranged from 1 to 1.3 seconds, depending on the amount of disclosed attributes.

Another well-known credential scheme is U-Prove [10,28] by Brands. Based on the difficulty of the discrete logarithm problem in a cyclic group, it can be implemented using elliptic curves, and additionally the showing protocol is much less complicated than that of Idemix, also resulting in more efficiency. However, in U-Prove two transactions executed with the same credential are always linkable, and the showing protocol is only honest-verifier zero-knowledge (i.e., there is no proof that dishonest verifiers cannot extract or learn information about the undisclosed attributes). Moreover, there is no unforgeability proof for U-Prove credentials, and it even seems that no such proof exists under standard intractability assumptions [4].

We also mention the "Anonymous Credentials Light" construction from [3], which can also be implemented on elliptic curves, but the credentials are not unlinkable; and [20], which runs in RSA groups like Idemix.

The credential scheme from [13], also by Camenisch and Lysyanskaya, is much closer to the scheme presented here: it is unlinkable, uses the (interactive) LRSW assumption, as well as elliptic curves and bilinear pairings (of the less efficient Type 1). In addition, how the signature scheme is used to obtain a credential scheme with a zero-knowledge disclosure protocol is similar to this work. The signature scheme that is used in [13] is, however, rather more complicated than ours: for example, when showing a credential the user has to compute an amount of pairings that is linear in the amount of disclosed attributes.

In [2] the BBS signature scheme [9] is modified into an unlinkable attribute-based credential scheme that, like the scheme from [13], requires the user to compute a number of (Type 2) pairings. However, the signatures in this scheme are short, and (like in Idemix but unlike our own scheme) its length does not depend on the amount of attributes.

More recently Fuchsbauer et al. [18] proposed a novel attribute-based credential scheme using structure-preserving signatures and a new commitment scheme, in which the undisclosed attributes are not hidden by knowledge proofs but rather by a partial opening to a commitment. As a result, like in Idemix the signature length does not depend on the amount of attributes. The scheme does, however, rely on a new variant of the strong Diffie-Hellman assumption that was newly introduced in the same paper.

In [5] an unlinkable scheme based on proofs of knowledge of Boneh-Boyen-like signature was proposed, achieving an efficient scheme with short signatures like Idemix and Fuchsbauer et al., and involving pairings only on the verifier's side.

In [22] we have examined a number of broken self-blindable credential schemes, and we posed a criterion which can indicate if a self-blindable credential scheme

is linkable or forgeable. The scheme that we introduce in this paper is however not susceptible to this criterion, as it only holds for deterministic signature schemes while ours is non-deterministic.

Finally, a blindable version of U-Prove was recently proposed in [21]. Although an unlinkable credential scheme is aimed at, the paper contains no unlinkability proof. Moreover, we have found that the scheme is forgeable: if sufficiently many users collide then they can create new credentials containing any set of attributes of their choice, without any involvement of the issuer [31].

2 Attribute-based credential schemes

First we fix some notation. We denote algorithms with calligraphic letters such as $\mathcal{A}$ and $\mathcal{B}$. By $y \leftarrow \mathcal{A}(x)$ we denote that y was obtained by running $\mathcal{A}$ on input x. If $\mathcal{A}$ is a deterministic algorithm then y is unique; if $\mathcal{A}$ is probabilistic then y is a random variable. We write $\mathcal{A}^O$ when algorithm $\mathcal{A}$ can make queries to oracle O. That is, A has an additional tape (read/write-once) on which it writes its queries; once it writes a special delimiter oracle O is invoked, and its answer appears on the query tape adjacent to the delimiter.

If $\mathcal{A}$ and $\mathcal{B}$ are interactive algorithms, we write $a \leftarrow \mathcal{A}(\cdot) \leftrightarrow \mathcal{B}(\cdot) \rightarrow b$ when $\mathcal{A}$ and $\mathcal{B}$ interact and afterwards output a and b, respectively. By $\mathcal{A}^{\blacksquare}\!\!\rightarrow\!\mathcal{B}$ we denote that algorithm $\mathcal{A}$ has black-box access to an interactive algorithm $\mathcal{B}$ – that is, $\mathcal{A}$ has oracle access to the next-message function function $\mathcal{B}_{x,y,r}(m)$ which, on input x that is common to $\mathcal{A}$ and $\mathcal{B}$, auxiliary input y and random tape r, specifies the message that $\mathcal{B}$ would send after receiving messages m. Finally, $|x|$ denotes the length of x in bits. For example, if x is an integer then $|x| = \lceil \log_2 x \rceil$.

For zero-knowledge proofs we will use the Camenisch-Stadler notation [14]. For example, if K, P_1, P_2 are elements of some (multiplicatively written) group then

$$\mathrm{PK}\big\{(k_1, k_2) \colon K = P_1^{k_1} P_2^{k_2}\big\}$$

denotes a zero-knowledge proof of knowledge of the numbers k_1, k_2 that satisfy the relation $K = P_1^{k_1} P_2^{k_2}$. (Unlike Camenisch and Stadler, we do not use Greek letters for the unknowns; instead we will consistently write them on the right-hand side of the equation.) Such proofs are based on standard techniques and occur in many areas of cryptography. In our case the protocol from [15] could for example be used.

For the full definitions of bilinear pairings, zero-knowledge proofs, and the unforgeability game of signature schemes, we refer to the full version of this paper [29].

Definition 1. An attribute-based credential scheme consists of the following protocols. (We assume a single issuer, but this can easily be generalized to multiple issuers.)

KeyGen$(1^\ell, n)$ This algorithm takes as input a security parameter ℓ and the number of attributes n that the credentials will contain, and outputs the

issuer's private key s and public key σ, which must contain the number n, and a description of the attribute space M.

Issue An interactive protocol between an issuer $\mathcal{I}$ and user $\mathcal{P}$ that results in a credential c:

$$\mathcal{I}(\sigma, s, (k_1, \ldots, k_n)) \leftrightarrow \mathcal{P}(\sigma, k_0, (k_1, \ldots, k_n)) \rightarrow c.$$

Here k_0 is the user's private key, that is to be chosen from the attribute space M by the user; the Issue protocol should prevent the issuer from learning it. We assume that before execution of this protocol, the issuer and user have reached agreement on the values of the attributes $k_1, \ldots, k_n$. The secret key and attributes $k_0, k_1, \ldots, k_n$ are contained in the credential c.

ShowCredential An interactive protocol between a user $\mathcal{P}$ and verifier $\mathcal{V}$ which is such that, if c is a credential[2] issued using the Issue protocol over attributes $(k_1, \ldots, k_n)$ using private signing key s corresponding to public key σ, then for any disclosure set $\mathcal{D} \subset \{1, \ldots, n\}$ the user can make the verifier accept:

$$\mathcal{P}(\sigma, c, \mathcal{D}) \leftrightarrow \mathcal{V}(\sigma, \mathcal{D}, (k_i)_{i \in \mathcal{D}}) \rightarrow 1.$$

Thus, the user will have to notify the verifier in advance of the disclosure set $\mathcal{D}$ and disclosed attributes $(k_i)_{i \in \mathcal{D}}$.

We expect our attribute-based credential scheme to satisfy the following properties.

- *Unforgeability* (see Definition 14): no user can prove possession of attributes that were not issued to it by the issuer.
- *Multi-show unlinkability* (see Definition 15): If a verifier $\mathcal{V}$ participates in the ShowCredential protocol twice, in which the same credential was involved, it should be impossible for it to tell whether both executions originated from the same credential or from two different ones.
- *Issuer unlinkability*: If in a run of the ShowCredential protocol certain attributes were disclosed, then of all credentials that the issuer issued with those attributes, the issuer cannot tell which one was used.
- *Offline issuer*: The issuer is not involved in the verification of credentials.
- *Selective disclosure*: Any subset of attributes contained in a credential can be disclosed.

The unforgeability and both kinds of unlinkability of an attribute-based credential scheme are defined in terms of two games. We have included these games in Appendix A.

The notion of unlinkability captures the idea that it is impossible for the verifier to distinguish two credentials from each other in two executions of the ShowCredential protocol, as long as they disclosed the same attributes with the

[2] As in Idemix and U-Prove, our ShowCredential protocol can easily be extended to simultaneously show multiple credentials that have the same secret key, and to proving that the hidden attributes satisfy arbitrary linear combinations [10].

same values. We will achieve this for our scheme by proving that our ShowCredential protocol is *black-box zero-knowledge*, which essentially means that the verifier learns nothing at all besides the statement that the user proves. Since the verifier learns nothing that it can use to link transactions, unlinkability follows from this (see Theorem 12).

3 Preliminaries

If $e\colon G_1 \times G_2 \to G_T$ is a bilinear pairing [19], we will always use uppercase letters for elements of G_1 or G_2, while lowercase letters (including Greek letters) will be numbers, i.e., elements of $\mathbb{Z}_p$. We will always use the index i for attributes, and in the unforgeability proofs below we will use the index j for multiple users or multiple credentials. For example, the number $k_{i,j}$ will refer to the i-th attribute of the credential of user j. If a, b are two natural numbers with $a < b$, then we will sometimes for brevity write $[a, b]$ for the set $\{a, \ldots, b\}$.

We write $\nu(\ell) < \mathrm{negl}(\ell)$ when the function $\nu\colon \mathbb{N} \to \mathbb{R}_{\geq 0}$ is negligible; that is, for any polynomial p there exists an ℓ' such that $\nu(\ell) < 1/p(\ell)$ for all $\ell > \ell'$.

3.1 Intractability assumptions

The unforgeability of the credential and signature schemes defined in the paper will depend on the *whLRSW assumption* [35], which as we will show below, is implied by the LRSW assumption [25,26] introduced by Lysyanskaya, Rivest, Sahai, and Wolf. The latter assumption has been proven to hold in the generic group model [30], and has been used in a variety of schemes (for example, [13,35,34,11,1]). Although this assumption suffices to prove unforgeability of our scheme, it is stronger than we need. In particular, the LRSW assumption is an interactive assumption, in the sense that the adversary is given access to an oracle which it can use as it sees fit. We prefer to use the weaker whLRSW assumption, which is implied by the LRSW assumption but does not use such oracles. Consequentially, unlike the LRSW assumption itself, and like conventional hardness assumptions such as factoring and DDH, this assumption is falsifiable [27]. We describe both assumptions below; then we prove that the LRSW assumption implies the whLRSW assumption. After this we will exclusively use the latter assumption.

Let $e\colon G_1 \times G_2 \to G_T$ be a Type 3 pairing, where the order p of the three groups is ℓ bits, and let $a, z \in_R \mathbb{Z}_p^*$. If $(\kappa, K, S, T) \in \mathbb{Z}_p \times G_1^3$ is such that $K \neq 1$, $S = K^a$ and $T = K^{z+\kappa az}$, then we call (κ, K, S, T) an *LRSW-instance*.

Definition 2 (LRSW assumption). Let e be as above, and let $O_{a,z}$ be an oracle that, when it gets $\kappa_j \in \mathbb{Z}_p$ as input on the j-th query, chooses a random $K_j \in_R G_1 \setminus \{1\}$ and outputs the LRSW-instance $(\kappa_j, K_j, K_j^a, K_j^{z+\kappa_j az})$. The *LRSW problem* is, when given $(p, e, G_1, G_2, G_T, Q, Q^a, Q^z)$ where $Q \in_R G_2 \setminus \{1\}$, along with oracle access to $O_{a,z}$, to output a new LRSW-instance $(\kappa, K, K^a, K^{z+\kappa az})$ where κ has never been queried to $O_{a,z}$. The *LRSW assumption* is that no probabilistic polynomial-time algorithm can solve the LRSW

problem with non-negligible probability in ℓ. That is, for every probabilistic polynomial-time algorithm $\mathcal{A}$ we have

$$\Pr\Big[a, z \in_R \mathbb{Z}_p^*;\ Q \in_R G_2 \setminus \{1\};\\ \sigma \leftarrow (p, e, G_1, G_2, G_T, Q, Q^a, Q^z);\quad (\kappa, K, S, T) \leftarrow \mathcal{A}^{O_{a,z}}(\sigma) :\\ K \in G_1 \setminus \{1\} \ \wedge\ \kappa \notin L \ \wedge\ S = K^a \ \wedge\ T = K^{z+\kappa az}\Big] < \mathrm{negl}(\ell),$$

where L is the list of oracle queries sent to $O_{a,z}$, and where the probability is over the choice of a, z, Q, and the randomness used by $\mathcal{A}$ and the oracle $O_{a,z}$.

Definition 3 (q-whLRSW assumption [35]). Let e be as above, and let $\{(\kappa_j, K_j, K_j^a, K_j^{z+\kappa_j az})\}_{j=1,\ldots,q}$ be a list of q LRSW-instances, where the κ_j and K_j are randomly distributed in $\mathbb{Z}_p$ and $G_1 \setminus \{1\}$, respectively. The *q-whLRSW problem* (for q-wholesale LRSW [35]) is, when given this list along with $(p, e, G_1, G_2, G_T, Q, Q^a, Q^z)$, to output a new LRSW-instance $(\kappa, K, K^a, K^{z+\kappa az})$ where $\kappa \notin \{\kappa_1, \ldots, \kappa_q\}$. The *$q$-whLRSW assumption* is that no probabilistic polynomial-time algorithm can solve the q-whLRSW problem with non-negligible probability in ℓ. That is, for every probabilistic polynomial-time algorithm $\mathcal{A}$ we have

$$\Pr\Big[a, z \in_R \mathbb{Z}_p^*;\ \kappa_1, \ldots, \kappa_q \in_R \mathbb{Z}_p;\ K_1, \ldots, K_q \in_R G_1 \setminus \{1\};\\ Q \in_R G_2 \setminus \{1\};\ \sigma \leftarrow (p, e, G_1, G_2, G_T, Q, Q^a, Q^z);\\ (\kappa, K, S, T) \leftarrow \mathcal{A}(\sigma, \{\kappa_j, K_j, K_j^a, K_j^{z+\kappa_j az}\}_{j\in[1,q]}) :\\ K \in G_1 \setminus \{1\} \ \wedge\ \kappa \notin \{\kappa_1, \ldots, \kappa_q\}\\ \wedge\ S = K^a \ \wedge\ T = K^{z+\kappa az}\Big] < \mathrm{negl}(\ell), \qquad (1)$$

where the probability is over the choice of $a, z,\ \kappa_1, \ldots, \kappa_q,\ K_1, \ldots, K_q, Q$, and the randomness used by $\mathcal{A}$.

Finally we define an unparameterized version of the assumption above by allowing q to be polynomial in ℓ, in the following standard way (e.g., [8]). Intuitively, the reason that this unparameterized assumption is implied by the LRSW assumption is simple: if there is no adversary that can create LRSW-instances when it can (using the oracle) control the κ's of the LRSW-instances that it gets as input, then an adversary that can create them *without* having control over the κ's also cannot exist.

Definition 4. Let e, p and $\ell = |p|$ be as above. The *whLRSW assumption* states that for all polynomials $q\colon \mathbb{N} \to \mathbb{N}$, the $q(\ell)$-whLRSW assumption holds.

Proposition 5. *The LRSW assumption implies the whLRSW assumption.*

We prove this in the full version of this paper [29]. Thus if we prove that our scheme is safe under the whLRSW asssumption, then it is also safe under the LRSW assumption. Additionally, we have found that the whLRSW assumption

can be proven by taking an extension [7] of the Known Exponent Assumption [16], so that unforgeability of our scheme can also be proven by using this assumption. However, because of space restrictions this proof could not be included here.

3.2 A signature scheme on the space of attributes

In this section we introduce a signature scheme on the space of attributes. This signature scheme will be the basis for our credential scheme, in the following sense: the Issue protocol that we present in Section 4 will enable issuing such signatures over a set of attributes to users, while the ShowCredential protocol allows the user to prove that it has a signature over any subset of its signed attributes.

Definition 6 (Signature scheme on attribute space). The signature scheme is as follows.

KeyGen$(1^\ell, n)$ The issuer generates a Type 3 pairing $e\colon G_1 \times G_2 \to G_T$, such that $|p| = \ell$ where p is the prime order of the three groups. Next it takes a generator $Q \in_R G_2$, and numbers $a, a_0, \ldots, a_n, z \in_R \mathbb{Z}_p^*$ and sets $A = Q^a, A_0 = Q^{a_0}, \ldots, A_n = Q^{a_n}$, and $Z = Q^z$. The public key is the tuple $\sigma = (p, e, Q, A, A_0, \ldots, A_n, Z)$ and the private key is the tuple $(a, a_0, \ldots, a_n, z)$.

Sign$(k_0, \ldots, k_n)$ The issuer chooses $\kappa \in_R \mathbb{Z}_p^*$ and $K \in_R G_1$, and sets $S = K^a, S_0 = K^{a_0}, \ldots, S_n = K^{a_n}$, and $T = (KS^\kappa \prod_{i=0}^n S_i^{k_i})^z$. The signature is $(\kappa, K, S, S_0, \ldots, S_n, T)$.

Verify$((k_0, \ldots, k_n), (\kappa, K, S, S_0, \ldots, S_n, T), \sigma)$ The signature is checked by setting $C = KS^\kappa \prod_{i=0}^n S_i^{k_i}$ and verifying that $K, C \neq 1$, as well as

$$\begin{aligned} &e(T, Q) \stackrel{?}{=} e(C, Z), \qquad e(S, Q) \stackrel{?}{=} e(K, A), \\ &e(S_i, Q) \stackrel{?}{=} e(K, A_i) \qquad \text{for each } i = 0, \ldots, n. \end{aligned} \tag{2}$$

The numbers $k_n \in \mathbb{Z}_p$ are the attributes. Although p may vary each time the KeyGen$(1^\ell, n)$ algorithm is invoked on a fixed security parameter ℓ, the attribute space $\mathbb{Z}_p$ will always contain $\{0, \ldots, 2^{\ell-1}\}$. In our credential scheme in section 4, the zeroth attribute k_0 will serve as the user's secret key, but at this point it does not yet have a special role.

Notice that contrary to Idemix and the BBS+ scheme from [2], but like the scheme from [13], the length of a signature is not constant in the amount n of attributes, but $O(n)$.

Although the element $C = KS^\kappa \prod_{i=0}^n S_i^{k_i}$ is, strictly speaking, not part of the signature and therefore also not part of the credential (since it may be calculated from κ, the attributes $(k_0, \ldots, k_n)$ and the elements $(K, S, S_0, \ldots, S_n)$), we will often think of it as if it is. Finally, we call a message-signature pair, i.e., a tuple of the form $((k_0, \ldots, k_n), (\kappa, K, S, S_0, \ldots, S_n, T))$ where $(\kappa, K, S, S_0, \ldots, S_n, T)$ is a valid signature over $(k_0, \ldots, k_n)$, a *credential*.

Notice that if $(k_0, \ldots, k_n), (\kappa, K, S, S_0, \ldots, S_n, T)$ is a valid credential, then for any $\alpha \in \mathbb{Z}_p^*$,

$$(k_0, \ldots, k_n), (\kappa, K^\alpha, S^\alpha, S_0^\alpha, \ldots, S_n^\alpha, T^\alpha) \tag{3}$$

is another valid credential having the same attributes. That is, in the terminology of Verheul [32] our credentials are *self-blindable*. This self-blindability is what makes this signature scheme suitable for the purpose of creating an unlinkable ShowCredential protocol.

The number κ will play a critical role in the unforgeability proof of our signature and credential schemes (Theorem 10).[3]

Theorem 7. *Our credentials are existentially unforgeable under adaptively chosen message attacks, under the whLRSW assumption.*

This is proven in the full version of this paper [29].

4 The credential scheme

In this section we present our credential scheme. The strategy is as follows: having defined an unforgeable signature scheme on the set of attributes $\mathbb{Z}_p^n$ (Definition 6), we provide an issuing protocol, in which the issuer grants a credential to a user, and a showing protocol, which allows a user to give a zero-knowledge proof to a verifier that he possesses a credential, revealing some of the attributes contained in the credential while keeping the others secret. The Issue protocol is shown in Figure 1, and the ShowCredential protocol is shown in Figure 2. Here and in the remainder of the paper, we will write $\mathcal{D} \subset \{1, \ldots, n\}$ for the index set of the disclosed attributes, and

$$\mathcal{C} = \{1, \ldots, n\} \setminus \mathcal{D}$$

for the index set of the undisclosed attributes. We do not consider the index 0 of the secret key k_0 to be part of this set, as it is always kept secret.

The Issue protocol is such that both parties contribute to κ and K with neither party being able to choose the outcome in advance (unlike the signing algorithm of the signature scheme from the previous section, where the signer chooses κ and K on its own). This ensures that these elements are randomly distributed even if one of the parties is dishonest. Additionally, the issuer is prevented from learning the values of κ and the secret key k_0.

As noted earlier, we assume that the user and issuer have agreed on the attributes $k_1, \ldots, k_n$ to be contained in the credential before executing this protocol. Similarly, we assume that the user sends the dislosure set $\mathcal{D}$ and disclosed attributes $(k_i)_{i \in \mathcal{D}}$ to the verifier prior to executing the ShowCredential protocol.

[3] We could have eased the notation somewhat by denoting the number κ as an extra attribute k_{n+1}, but because it plays a rather different role than the other attributes (it is part of the signature), we believe this would create more confusion than ease.

Common information: Attributes $k_1, \ldots, k_n$, issuer's public key $\sigma = (p, e, Q, A, A_0, \ldots, A_n, Z)$

User knows secret key k_0		**Issuer** knows $a, a_0, \ldots, a_n$, z
		choose $\bar{K} \in_R G_1$
	$\longleftarrow$	send $\bar{S} = \bar{K}^a, \bar{S}_0 = \bar{K}^{a_0}$
choose $\alpha, \kappa' \in_R \mathbb{Z}_p^*$		
set $S = \bar{S}^\alpha, S_0 = \bar{S}_0^\alpha$		
send $S, S_0, R = S^{\kappa'} S_0^{k_0}$	$\longrightarrow$	
$\mathrm{PK}\{(\kappa', k_0) \colon R = S^{\kappa'} S_0^{k_0}\}$	$\longleftrightarrow$	
		set $K = S^{1/a}$
		verify $S \neq \bar{S}, K = S_0^{1/a_0}$
		choose $\kappa'' \in_R \mathbb{Z}_p$
		set $S_i = K^{a_i}\ \forall i \in [1, n]$
		set $T = \left(K S^{\kappa''} R \prod_{i=1}^n S_i^{k_i}\right)^z$
	$\longleftarrow$	send $\kappa'', K, S_1, \ldots, S_n, T$
set $\kappa = \kappa' + \kappa''$		
return $(k_0, \ldots, k_n), (\kappa, K, S, S_0, \ldots, S_n, T)$		

Fig. 1. The Issue protocol. In the protocol, the issuers sends two elements $\bar{S}, \bar{S}_0$ (having the appropriate relative discrete log) to the user, who blinds them using a random number, and sends the blinded versions to the issuer. With respect to these blinded elements, the user then proves that it knows its secret key k_0 and its contribution κ' to the number κ. If the verifier is convinced, it chooses its own contribution κ'' to κ, and it computes the remaining elements $K, S_1, \ldots, S_n, T$ such that $(\kappa' + \kappa'', K, S, S_0, \ldots, S_n, T)$ is a valid signature over the attributes. These elements are sent to the user who finally constructs the credential.

If the user wants to be sure at the end of the Issue protocol that the new credential is valid, he will need to compute the pairings from equation (2). Even if the user is implemented on resource-constrained devices such as smart cards this is not necessarily a problem; generally in ABC's the issue protocol is performed much less often than the disclosure protocol so that longer running times may be more acceptable. Alternatively, the user could perform the ShowCredential protocol in which it discloses none of its attributes with the issuer, or perhaps another party; if the credential was invalid then this will fail.

The ShowCredential credential can be seen to consist of two separate phases: first, the user blinds the elements K, S, S_i, C and T with the number α as in equation (3), resulting in a new signature over his attributes. Second, the user uses the blinded elements to prove possession of this fresh signature over his attributes. The elements $\bar{S}$ and $\bar{S}_i$ can be used for this proof of knowledge only if they have all been correctly blinded using the same number α, which the verifier checks using the pairings at the end of the protocol. Thus, since α is only used to create a new blinded signature in advance of the proof of knowledge of this new signature, the value of α need not be known to the verifier, which

Common information: Issuer's public key $\sigma = (p, e, Q, A, A_0, \ldots, A_n, Z)$; disclosure set $\mathcal{D}$, undisclosed set $\mathcal{C} = \{1, \ldots, n\} \setminus \mathcal{D}$; disclosed attributes $(k_i)_{i \in \mathcal{D}}$

	User		**Verifier**
knows $K, S, S_0, \ldots, S_n, \kappa, (k_i)_{i \in \mathcal{C}}, C, T$			
	choose $\alpha, \beta \in_R \mathbb{Z}_p^*$		
	set $\bar{K} = K^\alpha$, $\bar{S} = S^\alpha$, $\bar{S}_i = S_i^\alpha \ \forall i \in [0, n]$		
	set $\tilde{C} = C^{-\alpha/\beta}$, $\tilde{T} = T^{-\alpha/\beta}$		
	send $\bar{K}$, $\bar{S}$, $(\bar{S}_i)_{i=0,\ldots,n}$, $\tilde{C}$, $\tilde{T}$	$\longrightarrow$	
	set $D = \bar{K}^{-1} \prod_{i \in \mathcal{D}} \bar{S}_i^{-k_i}$		set $D = \bar{K}^{-1} \prod_{i \in \mathcal{D}} \bar{S}_i^{-k_i}$
$\mathrm{PK}\{(\beta, \kappa, k_0, k_i)_{i \in \mathcal{C}} \colon D = \tilde{C}^\beta \bar{S}^\kappa \bar{S}_0^{k_0} \prod_{i \in \mathcal{C}} \bar{S}_i^{k_i}\}$		$\longleftrightarrow$	
			verify $e(\bar{K}, A) \stackrel{?}{=} e(\bar{S}, Q)$
			and $e(\bar{K}, A_i) \stackrel{?}{=} e(\bar{S}_i, Q) \ \forall i \in [0, n]$
			and $e(\tilde{C}, Z) \stackrel{?}{=} e(\tilde{T}, Q)$

Fig. 2. The ShowCredential protocol. We assume that the user has the element $C = KS^\kappa S_0^{k_0} \cdots S_n^{k_n}$ stored so that it does not need to compute it every time the protocol is run (see Section 5 for more such optimizations). In the protocol, the user first blinds K, S and each S_i with a random number, and C and T with a different random number, resulting in new elements $\bar{K}, \bar{S}, \bar{S}_i$ and $\tilde{C}, \tilde{T}$. These are sent to the verifier. Then, the user proves that he knows the hidden attributes and the number κ, as well as a number β which is such that $\tilde{C}^\beta$ is of the required form $\tilde{C}^\beta = \bar{K}\bar{S}^\kappa \bar{S}_0^{k_0} \prod_{i=1}^n \bar{S}_i^{k_i}$. If the proof of knowledge is valid and the elements $\bar{K}$, $\bar{S}$ and $\bar{S}_i$ on the one hand and $\tilde{C}$, $\tilde{T}$ on the other hand have the appropriate relative discrete logarithms (which the verifier checks by calculating a number of pairings), then the verifier accepts.

is why the user does not need to prove knowledge of it. The same holds for the number α that is used during issuance; as long as it is correctly applied (which the issuer here checks by directly using his secret key instead of having to compute pairings), the user can prove knowledge of κ' and his secret key k_0 without the issuer needing to know α.

Mathematically, we can formalize what the ShowCredential protocol should do as follows. The common knowledge of the user and verifier when running the ShowCredential protocol consists of elements of the following formal language:

$$L = \left\{ \left(\sigma, \mathcal{D}, (k_i)_{i \in \mathcal{D}}\right) \mid \mathcal{D} \subset \{1, \ldots, n\},\ k_i \in \mathbb{Z}_p \, \forall \, i \in \mathcal{D} \right\} \tag{4}$$

where σ ranges over the set of public keys of the credential scheme, and where n is the amount of attributes of σ. In addition, let the relation R be such that $R(x, w) = 1$ only if $x = (\sigma, \mathcal{D}, (k_i)_{i \in \mathcal{D}}) \in L$, and $w = ((k'_0, \ldots, k'_n), s)$ is a valid credential with respect to σ, with $k'_i = k_i$ for $i \in \mathcal{D}$ (i.e., the disclosed attributes $(k_i)_{i \in \mathcal{D}}$ are contained in the credential w.) Thus the equation $R(x, w) = 1$ holds only if w is a valid credential having attributes $(k_i)_{i \in \mathcal{D}}$.

Theorem 8. *The showing protocol is complete with respect to the language L: if a user has a valid credential then it can make the verifier accept.*

Proof. If the user follows the ShowCredential protocol, then $e(\bar{K}, A) = e(K^\alpha, Q^a) = e(K^{\alpha a}, Q) = e(S^\alpha, Q) = e(\bar{S}, Q)$, so the first verification that the verifier does will pass. An almost identical calculation shows that the second and third verifications pass as well. As to the proof of knowledge, setting $\bar{C} = C^\alpha$ we have

$$\tilde{C}^\beta \bar{S}^\kappa \bar{S}_0^{k_0} \prod_{i \in \mathcal{C}} \bar{S}_i^{k_i} = \bar{C}^{-1} \bar{S}^\kappa \bar{S}_0^{k_0} \prod_{i \in \mathcal{C}} \bar{S}_i^{k_i} = \bar{K}^{-1} \prod_{i \in \mathcal{D}} \bar{S}_i^{-k_i} = D, \tag{5}$$

so the user can perform this proof without problem. □

4.1 Unforgeability and unlinkability

The proofs of the following theorems may be found in the full version of this paper [29].

Lemma 9. *With respect to the language L defined in* (4)*, the* ShowCredential *protocol is black-box extractable.*

In the proofs of the unforgeability and unlinkability theorems, we will need a tuple $(\hat{K}, \hat{S}, \hat{S}_0, \ldots, \hat{S}_n, \hat{C}, \hat{T}) \in G_1^{n+5}$ such that $\hat{S} = \hat{K}^a$ and $\hat{S}_i = \hat{K}^{a_i}$ for all i, as well as $\hat{T} = \hat{C}^z$. For that reason we will henceforth assume that such a tuple is included in the issuer's public key. Note that one can view these elements as an extra credential of which the numbers $(\kappa, k_0, \ldots, k_n)$ are not known. Therefore the credential scheme remains unforgeable (the adversary can in fact already easily obtain such a tuple by performing an Issue query in the unforgeability game).[4]

Theorem 10. *Our credential scheme is unforgeable under the whLRSW assumption.*

Theorem 11. *The* ShowCredential *protocol is a black-box zero-knowledge proof of knowledge with respect to the language L.*

Theorem 12. *Let* (KeyGen, Issue, ShowCredential) *be an attribute-based credential scheme whose* ShowCredential *protocol is black-box zero-knowledge. Then the scheme is unlinkable.*

Theorem 13. *Our credential scheme is unlinkable.*

5 Performance

5.1 Exponentiation count

Table 1 compares the amount of exponentiations in our scheme to those of [13], U-Prove and Idemix. However, note that exponentiations in RSA-like groups,

[4] Credential owners already have such a tuple; verifiers can obtain one simply by executing the ShowCredential protocol; and issuers can of course create such tuples by themselves. Therefore in practice, each party participating in the scheme will probably already have such a tuple, so that including it in the public key may not be necessary in implementations.

Table 1. Exponentiation and pairing count for the user of the ShowCredential protocol of several attribute-based credential schemes. The columns G_{EC}, G_T and G_{RSA} show the amount of exponentiations in elliptic curves, the target group of a bilinear pairing, and RSA groups respectively, while the column labeled e counts the amount of pairings the user has to compute. The number n denotes the amount of attributes, excluding the secret key, and the function $\mathrm{pk}(n)$ denotes the amount of exponentiations necessary in order to perform a zero-knowledge proof of knowledge of n numbers (in the case of the Fiat-Shamir heuristic applied to the Schnorr Σ-protocol, which Idemix also uses, we have $\mathrm{pk}(n) = n$).

	G_{EC}	G_T	e	G_{RSA}	unlinkable
Our scheme	$n + \mathrm{pk}(\lvert\mathcal{C}\rvert + 3) + 6$	0	0	0	yes
[13]	$2n + 3$	$\mathrm{pk}(\lvert\mathcal{C}\rvert + 2)$	$n + 3$	0	yes
[18]	$\lvert\mathcal{C}\rvert + \mathrm{pk}(2) + 5$	0	0	0	yes
[5]	$\mathrm{pk}(\lvert\mathcal{C}\rvert + 7) + 5$	0	0	0	yes
Idemix	0	0	0	$\lvert\mathcal{C}\rvert + 3$	yes
U-Prove	$\lvert\mathcal{C}\rvert + 1$	0	0	0	no

Table 2. A comparison of the running times of various actions in the implementation of our credential scheme and the IRMA Idemix implementation, both of them using the Fiat-Shamir heuristic. The columns labeled "computing proof" and "verifying proof" show how long it takes to compute and to verify a disclosure proof, respectively, while the column labeled "verifying credential" shows how long it takes to verify the signature of a credential. The left column shows the total number of attributes and, if applicable, the amount of disclosed attributes (this does not apply to the "verifying credential" column). The attributes were randomly chosen 253-bit integers, the same across all tests, and the computations were performed on a dual-core 2.7 GHz Intel Core i5. All running times are in milliseconds, and were obtained by computing the average running time of 1000 iterations.

# attributes	computing proof		verifying proof		verifying credential	
total (discl.)	This work	Idemix	This work	Idemix	This work	Idemix
6 (1)	2.9	11.7	5.7	11.2	5.1	6.5
7 (1)	2.9	12.6	6.5	12.2	5.8	6.9
8 (1)	3.2	13.4	7.1	13.2	6.6	7.4
9 (1)	3.4	14.3	8.0	14.0	7.2	7.7
10 (1)	3.7	15.2	8.7	14.9	7.8	8.3
11 (1)	3.9	16.5	9.4	15.8	8.6	8.7
12 (1)	4.2	17.1	10.2	16.9	9.0	8.9
6 (5)	2.1	7.6	5.9	9.2		
7 (6)	2.1	7.5	6.5	9.7		
8 (7)	2.3	7.5	7.2	10.1		
9 (8)	2.4	7.4	7.9	10.7		
10 (9)	2.6	7.4	8.5	10.9		
11 (10)	2.7	7.5	9.1	11.4		
12 (11)	2.8	7.5	9.9	12.0		

on which Idemix depends, are significantly more expensive than exponentiations in elliptic curves. The scheme from [18] is slightly cheaper than ours for the prover, but relies on a newly introduced hardness assumption. Also, the U-Prove showing protocol offers no unlinkability. As to the scheme from [13], Camenisch and Lysyanskaya did not include a showing protocol that allows attributes to be disclosed (that is, it is assumed that all attributes are kept secret), but it is not very difficult to keep track of how much less the user has to do if he voluntarily discloses some attributes. We see that the amount of exponentiations that the user has to perform in the ShowCredential protocol of [13] is roughly 1.5 times as large as in our scheme. Since, additionally, computing pairings is significantly more expensive than exponentiating, we expect our credential scheme to be at least twice as efficient.

5.2 Implementation

In order to further examine the efficiency of our credential scheme we have written a preliminary implementation, using the high-speed 254-bit BN-curve and pairing implementation from [6]. The latter is written in C++ and assembly but also offers a Java API, and it uses the GMP library from the GNU project[5] for large integer arithmetic. Table 2 shows the running times of our implementation along with those from the Idemix implementation from the IRMA project.[6] We have tried to make the comparison as honest as possible by writing our implementation in Java, like the IRMA Idemix implementation, which we have modified to also use the GMP library for its large integer arithmetic. In addition, like IRMA we have used the Fiat-Shamir heuristic. However, the comparison can still only go so far, because the elliptic curve group that [6] offers is heavily optimized for fast computations, from which our scheme profits because it allows multiple issuers to use the same group. Such optimizations are not possible in Idemix because each Idemix public key necessarily involves its own group. Moreover, the IRMA Idemix implementation is 1024-bits, which according to [24] corresponds to a 144 bit curve (see also `www.keylength.com`), so that the two implementations do not offer the same level of security.

For these reasons we will go no further than draw qualitative conclusions from the data. Nevertheless, both remarks actually demonstrate the efficiency of our scheme: the first means that our scheme can be optimized further than Idemix could, and Table 2 shows that even though our implementation offers a much higher level of security, it is still significantly faster than the IRMA Idemix implementation. We believe therefore that the conclusion that our scheme is or can be more efficient than Idemix – at least for the user in the ShowCredential protocol – is justified.

[5] See `gmplib.org`.
[6] See `irmacard.org` and `github.com/credentials`.

6 Conclusion

In this paper we have defined a new self-blindable attribute-based credential scheme, and given a full security proof by showing that it is unforgeable and unlinkable. Our scheme is based on a standard hardness assumption and does not need the random oracle model. Based on the fact that it uses elliptic curves and bilinear pairings (but the latter only on the verifier's side), on a comparison of exponentiation counts, and on a comparison of run times with the IRMA Idemix implementation, we have shown it to be more efficient than comparable schemes such as Idemix and the scheme from [13], achieving the same security goals at less cost.

Acknowledgments

We are very grateful to the anonymous referees for their helpful and constructive feedback.

References

1. Ateniese, G., Camenisch, J., de Medeiros, B.: Untraceable rfid tags via insubvertible encryption. In: Proceedings of the 12th ACM Conference on Computer and Communications Security - CCS '05. pp. 92–101. ACM, New York, NY, USA (2005)
2. Au, M.H., Susilo, W., Mu, Y.: Constant-Size Dynamic k-TAA, pp. 111–125. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
3. Baldimtsi, F., Lysyanskaya, A.: Anonymous credentials light. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security. pp. 1087–1098. CCS '13, ACM, New York, NY, USA (2013)
4. Baldimtsi, F., Lysyanskaya, A.: On the security of one-witness blind signature schemes. In: Sako, K., Sarkar, P. (eds.) Advances in Cryptology - ASIACRYPT 2013. LNCS, vol. 8270, pp. 82–99. Springer Berlin Heidelberg (2013)
5. Barki, A., Brunet, S., Desmoulins, N., Traoré, J.: Improved algebraic MACs and practical keyed-verification anonymous credentials. In: Selected Areas in Cryptography - SAC 2016 (2016)
6. Beuchat, J., González-Díaz, J.E., Mitsunari, S., Okamoto, E., Rodríguez-Henríquez, F., Teruya, T.: High-speed software implementation of the optimal ate pairing over barreto-naehrig curves. In: Joye, M., Miyaji, A., Otsuka, A. (eds.) Pairing-Based Cryptography - Pairing 2010. Lecture Notes in Computer Science, vol. 6487, pp. 21–39. Springer (2010)
7. Bitansky, N., Canetti, R., Chiesa, A., Tromer, E.: From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference - ITCS '12. pp. 326–349. ACM, New York, NY, USA (2012)
8. Boneh, D., Boyen, X.: Short signatures without random oracles and the SDH assumption in bilinear groups. J. Cryptology 21(2), 149–177 (2008)
9. Boneh, D., Boyen, X., Shacham, H.: Short Group Signatures, pp. 41–55. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
10. Brands, S.: Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy. MIT Press (2000)

11. Camenisch, J., Hohenberger, S., Pedersen, M.Ø.: Batch verification of short signatures. In: Naor, M. (ed.) Advances in Cryptology - EUROCRYPT 2007. pp. 246–263. Springer Berlin Heidelberg, Berlin, Heidelberg (2007), https://eprint.iacr.org/2007/172.pdf
12. Camenisch, J., Lysyanskaya, A.: An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In: Pfitzmann, B. (ed.) Advances in Cryptology — EUROCRYPT 2001. LNCS, vol. 2045, pp. 93–118. Springer Berlin Heidelberg (2001)
13. Camenisch, J., Lysyanskaya, A.: Signature schemes and anonymous credentials from bilinear maps. In: Franklin, M. (ed.) Advances in Cryptology – CRYPTO 2004. LNCS, vol. 3152, pp. 56–72. Springer Berlin Heidelberg (2004)
14. Camenisch, J., Stadler, M.: Efficient group signature schemes for large groups. In: Kaliski, B.S.J. (ed.) Advances in Cryptology — CRYPTO '97. LNCS, vol. 1294, pp. 410–424. Springer Berlin Heidelberg (1997)
15. Cramer, R., Damgård, I., MacKenzie, P.: Efficient zero-knowledge proofs of knowledge without intractability assumptions. In: Imai, H., Zheng, Y. (eds.) Public Key Cryptography. LNCS, vol. 1751, pp. 354–372. Springer Berlin Heidelberg (2000)
16. Damgård, I.: Towards practical public key systems secure against chosen ciphertext attacks. In: Feigenbaum, J. (ed.) Advances in Cryptology - CRYPTO '91. LNCS, vol. 576, pp. 445–456. Springer (1991)
17. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) Advances in Cryptology – CRYPTO' 86. LNCS, vol. 263, pp. 186–194. Springer Berlin Heidelberg (1987)
18. Fuchsbauer, G., Hanser, C., Slamanig, D.: Structure-preserving signatures on equivalence classes and constant-size anonymous credentials. Cryptology ePrint Archive, Report 2014/944 (2014), https://eprint.iacr.org/2014/944
19. Galbraith, S.D., Paterson, K.G., Smart, N.P.: Pairings for cryptographers. Discrete Applied Mathematics 156(16), 3113–3121 (2008)
20. Hajny, J., Malina, L.: Unlinkable attribute-based credentials with practical revocation on smart-cards. In: Mangard, S. (ed.) Smart Card Research and Advanced Applications. LNCS, vol. 7771, pp. 62–76. Springer Berlin Heidelberg (2013)
21. Hanzlik, L., Kluczniak, K.: A short paper on how to improve U-Prove using self-blindable certificates. In: Christin, N., Safavi-Naini, R. (eds.) Financial Cryptography and Data Security: 18th International Conference, FC 2014. pp. 273–282. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)
22. Hoepman, J.H., Lueks, W., Ringers, S.: On linkability and malleability in self-blindable credentials. In: 9th WISTP International Conference on Information Security and Practice - WISTP'2015 (2015)
23. IBM Research Zürich Security Team: Specification of the Identity Mixer cryptographic library, version 2.3.0. Tech. rep., IBM Research, Zürich (feb 2012), https://tinyurl.com/idemix-spec
24. Lenstra, A.K., Verheul, E.R.: Selecting cryptographic key sizes. J. Cryptology 14(4), 255–293 (2001)
25. Lysyanskaya, A.: Pseudonym Systems. Master's thesis, Massachusetts Institude of Technology (1999), https://groups.csail.mit.edu/cis/theses/anna-sm.pdf
26. Lysyanskaya, A., Rivest, R.L., Sahai, A., Wolf, S.: Pseudonym systems. In: Heys, H., Adams, C. (eds.) Selected Areas in Cryptography: 6th Annual International Workshop, SAC'99. pp. 184–199. Springer Berlin Heidelberg, Berlin, Heidelberg (2000)

27. Naor, M.: On cryptographic assumptions and challenges. In: Boneh, D. (ed.) Advances in Cryptology - CRYPTO 2003. pp. 96–109. Springer Berlin Heidelberg, Berlin, Heidelberg (2003)
28. Paquin, C., Zaverucha, G.: U-prove cryptographic specification v1.1 (revision 3) (December 2013), http://research.microsoft.com/apps/pubs/default.aspx?id=166969, released under the Open Specification Promise
29. Ringers, S., Verheul, E., Hoepman, J.H.: An efficient self-blindable attribute-based credential scheme. Cryptology ePrint Archive, Report 2017/115 (2017), https://eprint.iacr.org/2017/115
30. Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Fumy, W. (ed.) Advances in Cryptology — EUROCRYPT '97. LNCS, vol. 1233, pp. 256–266. Springer Berlin Heidelberg (1997)
31. Verheul, E., Ringers, S., Hoepman, J.H.: The self-blindable U-Prove scheme from FC'14 is forgeable. Financial Cryptography and Data Security – FC'16 (2016), https://eprint.iacr.org/2015/725
32. Verheul, E.R.: Self-blindable credential certificates from the weil pairing. In: Boyd, C. (ed.) Advances in Cryptology - ASIACRYPT. LNCS, vol. 2248, pp. 533–551. Springer (2001)
33. Vullers, P., Alpár, G.: Efficient selective disclosure on smart cards using idemix. In: Fischer-Hübner, S., de Leeuw, E., Mitchell, C. (eds.) Policies and Research in Identity Management. IFIP Advances in Information and Communication Technology, vol. 396, pp. 53–67. Springer Berlin Heidelberg (2013)
34. Wachsmann, C., Chen, L., Dietrich, K., Löhr, H., Sadeghi, A.R., Winter, J.: Lightweight anonymous authentication with tls and daa for embedded mobile devices. In: Burmester, M., Tsudik, G., Magliveras, S., Ilić, I. (eds.) Information Security: 13th International Conference, ISC 2010. pp. 84–98. Springer Berlin Heidelberg, Berlin, Heidelberg (2011), https://eprint.iacr.org/2011/101.pdf
35. Wei, V.K., Yuen, T.H.: More short signatures without random oracles. IACR Cryptology ePrint Archive 2005, 463 (2005), http://eprint.iacr.org/2005/463

A Unforgeability and unlinkability games

Unforgeability of a credential scheme is defined using the following game (resembling the signature scheme unforgeability game).

Definition 14 (unforgeability game). The unforgeability game of an attribute-based credential scheme between a challenger and an adversary $\mathcal{A}$ is defined as follows.

Setup For a given security parameter ℓ, the adversary decides on the number of attributes $n \geq 1$ that each credential will have, and sends n to the challenger. The challenger then runs the $\mathsf{KeyGen}(1^\ell, n)$ algorithm from the credential scheme and sends the resulting public key to the adversary.

Queries The adversary $\mathcal{A}$ can make the following queries to the challenger.

- $\mathsf{Issue}(k_{1,j}, \ldots, k_{n,j})$ The challenger and adversary engage in the Issue protocol, with the adversary acting as the user and the challenger acting as the issuer, over the attributes $(k_{1,j}, \ldots, k_{n,j})$. It may choose these adaptively.

ShowCredential$(\mathcal{D}, k_1, \ldots, k_n)$ The challenger creates a credential with the specified attributes $k_1, \ldots, k_n$, and engages in the ShowCredential protocol with the adversary, acting as the user and taking $\mathcal{D}$ as disclosure set, while the adversary acts as the verifier.

Challenge The challenger, now acting as the verifier, and the adversary, acting as the user, engage in the ShowCredential protocol. If the adversary manages to make the verifier accept a credential with disclosed attributes $(k_i)_{i\in\mathcal{D}}$ (where $\mathcal{D} \neq \emptyset$), and there is no j such that $k_i = k_{i,j}$ for all $i \in \mathcal{D}$ (i.e., there is no single credential from one of the Issue queries containing all of the disclosed attributes $(k_i)_{i\in\mathcal{D}}$), then the adversary wins.

We say that the credential scheme is *unforgeable* if no probabilistic polynomial-time algorithm can win this game with non-negligible probability in the security parameter ℓ.

Next we turn to the unlinkability game.

Definition 15 (unlinkability game). The unlinkability game of an attribute-based credential scheme between a challenger and an adversary $\mathcal{A}$ is defined as follows.

Setup For a given security parameter ℓ, the adversary decides on the number of attributes $n \geq 1$ that each credential will have, and sends n to the challenger. The adversary then runs the KeyGen$(1^\ell, n)$ algorithm from the credential scheme and sends the resulting public key to the challenger.

Queries The adversary $\mathcal{A}$ can make the following queries to the challenger.

Issue$(k_{1,j}, \ldots, k_{n,j})$ The adversary chooses a set of attributes $(k_{1,j}, \ldots, k_{n,j})$, and sends these to the challenger. Then, acting as the issuer, the adversary engages in the Issue protocol with the challenger, issuing a credential j to the challenger having attributes $(k_{1,j}, \ldots, k_{n,j})$.

ShowCredential$(j, \mathcal{D})$ The adversary and challenger engage in the showing protocol on credential j, the challenger acting as the user and the adversary as the verifier. Each time the adversary may choose the disclosure set $\mathcal{D}$.

Corrupt(j) The challenger sends the entire internal state, including the secret key k_0, of credential j to the adversary.

Challenge The adversary chooses two uncorrupted credentials j_0, j_1 and a disclosure set $\mathcal{D} \subset \{1, \ldots, n\}$. These have to be such that the disclosed attributes from credential j_0 coincide with the ones from credential j_1, i.e., $k_{i,j_0} = k_{i,j_1}$ for each $i \in \mathcal{D}$. It sends the indices j_0, j_1 and $\mathcal{D}$ to the challenger, who checks that this holds; if it does not then the adversary loses.

Next, the challenger flips a bit $b \in_R \{0, 1\}$, and acting as the user, it engages in the ShowCredential with the adversary on credential j_b. All attributes whose index is in $\mathcal{D}$ are disclosed.

Output The adversary outputs a bit b' and wins if $b = b'$.

We define the advantage of the adversary $\mathcal{A}$ as $\mathsf{Adv}_\mathcal{A} := |\Pr[b = b'] - 1/2|$. When no probabilistic polynomial-time algorithm can win this game with non-negligible advantage in the security parameter ℓ, then we say that the credential scheme is *unlinkable*.