Issuer-Hiding Attribute-Based Credentials

Jan Bobolz[1], Fabian Eidens[1],
Stephan Krenn[2], Sebastian Ramacher[2], and Kai Samelin[3]

[1] Paderborn University, Paderborn, Germany
{jan.bobolz,fabian.eidens}@uni-paderborn.de
[2] AIT Austrian Institute of Technology, Vienna, Austria
{stephan.krenn,sebastian.ramacher}@ait.ac.at
[3] Independent, Germany
kaispapers@gmail.com

Abstract. Attribute-based credential systems enable users to authenticate in a privacy-preserving manner. However, in such schemes verifying a user's credential requires knowledge of the issuer's public key, which by itself might already reveal private information about the user.
In this paper, we tackle this problem by introducing the notion of *issuer-hiding* attribute-based credential systems. In such a system, the verifier can define a set of acceptable issuers in an ad-hoc manner, and the user can then prove that her credential was issued by one of the accepted issuers – without revealing which one. We then provide a generic construction, as well as a concrete instantiation based on Groth's structure preserving signature scheme (ASIACRYPT'15) and simulation-sound extractable NIZK, for which we also provide concrete benchmarks in order to prove its practicability.
The online complexity of all constructions is independent of the number of acceptable verifiers, which makes it also suitable for highly federated scenarios.

Keywords: Cryptographic protocols · issuer-hiding · privacy-preserving · anonymous credentials · authentication

1 Introduction

Anonymous credential systems and their attribute-based extensions (ABCs) allow *users* to receive digital certificates (*credentials*) certifying certain pieces of personal information (*attributes*) from *issuers*. A user can then present her credential to a *verifier* in a way that respects the user's privacy while giving high authenticity guarantees to the verifier. That is, the user can decide, on a fine-granular basis, which information about her attributes she wants to disclose to the verifier, while no further information, including metadata, is revealed. In particular, different actions of the same user can only be linked through the disclosed information. In the most general case, the verifier can publish arbitrary predicates (Boolean formulas) over attribute values that users need to satisfy

for authentication (e.g., a user is older than 21, comes from a specific country, or has a certain name), and receives cryptographic evidence that such attribute values were certified by the given issuer. Anonymous credential systems were first envisioned by Chaum [24, 25]. Besides well-known systems like Microsoft's U-Prove [11, 43] and IBM's Identity Mixer [18–20, 22], a large body of work with different optimizations and functionalities can be found in the literature, e.g. [7, 8, 15–17, 27, 37, 44].

All of the aforementioned ABC systems have in common that the privacy guarantees only hold with respect to a single issuer key: whilst not being able to link actions of a single user, a verifier learns the public key of the issuer of the underlying credential. Even though this seems to be a natural property at first glance, it turns out that this approach leads to a tradeoff between scalability and user privacy. As an example, consider a state-wide electronic identity system with millions of users. In order to give users the highest level of privacy, all citizens' personal credentials need to be issued under the same public key. In case of a compromise of the secret key, all previously issued keys need to be invalidated, potentially requiring millions of certificates to be re-issued under a new key. Alternatively, different keys could be used for groups of citizens, e.g., randomly, per time period, or per administrative region. However, as the issuer's public key is revealed upon presentation, this approach dramatically reduces the size of the anonymity set of a specific user.

Furthermore, many scenarios would benefit from a dynamic and ad-hoc definition of a set of issuer keys accepted by a verifier. For instance, universities may issue electronic student IDs to their students. Depending on the concrete scenario, students may need to prove that they are enrolled at a specific university (e.g., to enter the campus), or that they are enrolled at any university without needing to reveal the university (e.g., to be granted an online student discount). Similarly, citizens may receive electronic identities from their nation state, which they can then use to prove that they are eligible, e.g., for participation in opinion polls in their country. However, they might want to use the same credential to also prove that they are living in any country of a specific region (e.g., the European Union) for cross-country citizen engagement processes which do not require to reveal the specific citizenship.

In vehicular ad-hoc networks (VANETs) [32] or vehicle-to-infrastructure networks (V2I), such a solution allows each car manufacturer to use their own secret keys (e.g., per model), while avoiding to reveal unnecessary information (e.g., the model) when authenticating towards other parties.

Finally, Cloudflare recently announced a replacement of CAPTCHAs by cryptographic attestation of personhood using, e.g., FIDO tokens.[4] The idea is that instead of solving a puzzle, users click a physical button on an accepted hardware token, which responds to a cryptographic challenge. However, as pointed out by Cloudflare, a user's key "looks like all other keys in the batch", meaning that the anonymity set of a user shrinks to the number of devices in a batch. It

[4] `https://blog.cloudflare.com/introducing-cryptographic-attestation-of-personhood/`

would thus be desirable to dynamically add additional batches to this anonymity set, without users needing to obtain new credentials for their existing devices.

Related work. Different mitigation strategies for these challenges exists. For instance, approaches for decentralizing the issuer have been proposed, e.g., by fully avoiding the need for a trusted issuer leveraging public append-only ledgers [33, 48], or by deploying threshold cryptography to require multiple issuers to contribute to the credential issuance process [14, 41, 47]. While such approaches reduce the risk of compromised issuer keys, they do not directly allow to dynamically adjust the set of issuers among which a user should remain private.

Delegatable credentials [4, 7, 13, 23, 27] offer an alternative solution, where users can issue credentials derived from their own credentials to other users. All credentials eventually trace back to a root authority's public key, yet the verifier does not learn the precise issuer within the delegation "tree". While delegatable credentials are a valuable tool, e.g., for managing privileges within organizations, they do not solve the issues addressed in this paper as they assume a single root authority, which will typically not exist in the federated scenarios sketched above. They also do not allow for ad-hoc definitions of accepted issuers. Furthermore, revocation of sub-issuers within the delegation tree is computationally expensive, while it can be achieved for free in our construction.

Closely related to anonymous credentials, also in self-sovereign identity (SSI) systems multiple issuers will participate. In such systems, e.g, [2,3,38], users can join by presenting some form of credential to one or multiple verification nodes. In eSSIF[5], which is the European Union's attempt to build a large scale SSI system, these credentials are issued by national authorities run by each member state. If the credential is accepted by the nodes, they record their decision on a distributed ledger. Even if these systems are not built from ABCs, they can be designed to mimic some of their functionalities. Indeed, whenever the user wants to present attributes included in their credential to a service provider, a presentation of some of the attributes can be computed with respect to the information stored on the ledger. Due to the trust put into the distributed ledger and the verification nodes, it is thereby not necessary to show the issuer public key to the verifier. Hence, this additional layer, i.e. the ledger and verification nodes, provides some level of mitigation against identification attempts based on the issuer. Yet, the issuer is known to the verification nodes responsible for the initial joining of the system. Especially when the system is built from a public ledger, a service provider could also run such a node and therefore information on the issuers could potentially be gathered. Also, the authenticity guarantees are no longer end-to-end, but partially rely on the verification nodes and the consensus mechanism employed for the distributed ledger.

Our Contributions. In this paper we address the discussed challenges by presenting an *issuer-hiding* attribute-based credential system. That is, our system allows a user to hide the issuer of her credential among a set of issuers. More

[5] `https://decentralized-id.com/government/europe/eSSIF/`

specifically, the verifier may issue a *policy* defining the issuers he is willing to accept for a certain presentation session, and the user may then prove that she indeed owns a credential issued by one of those issuers.

Firstly, this approach allows a user to use her credential in various contexts, as described in the examples above. Secondly, the revocation of issuers becomes efficient in the sense that credentials issued by a specific issuer can be revoked by simply no longer including this issuer in the policy. Finally, additional issuers can be added in a seamless fashion by adding them to the policy.

Overview of Our Approach. To explain the technicalities of our construction let us first solve the hiding of public keys during authentication straightforwardly. As already mentioned, a user's credential on attributes $m = [\texttt{age, name, state, reputation}]$ by issuer I_j is a signature *cred* on the message vector m valid under the issuer's public key ipk_j. To authenticate at a verifier V_k the user U proves validity of *cred* under the public key ipk_j. More formally, U sends a non-interactive proof $\text{NIZK}[(x = ipk_j, w = \{m, cred\}) \colon \mathsf{Verify}(ipk_j, cred, m)]$. Public common input to the NIZK is ipk_j. The witness, hence private input by the user are *cred* and m. The NIZK deals with the privacy of the witness, but ipk_j is publicly known. As a feature this lets verifiers interpret attributes and credentials with respect to the issuer, e.g. `reputation` has potentially more weight if ipk_j belongs to a government agency. In other cases, this is a detriment to user privacy, e.g. the attribute `state` certified in *cred* is never revealed by the user, nonetheless the verifier may learn `state` implicitly by looking at ipk_j.

An idea to hide ipk_j in the above NIZK is to build a structure reminiscent of ring signatures. For authentication, the user collects an appropriate set of issuer public keys $PK \coloneqq \{ipk_1, \ldots, ipk_j, \ldots, ipk_n\}$. Then we change the NIZK statement to $\text{NIZK}[(x = PK, w = \{ipk_j, m, cred\}) \colon \vee_{i=1}^{n} \mathsf{Verify}(ipk_i, cred, m)]$. We solved our problem, the or-statement in the NIZK hides under which *ipk* the user's credential is valid.

The downside is that naively the proof size and verification cost is now linear in $n \coloneqq |PK|$ which limits the practicability of this approach. Hence, the next essential step is to avoid the or-statement in the NIZK.

This can be achieved by letting the verifier sign the public keys of the accepted issuers, by computing $\sigma_j \xleftarrow{\$} \mathsf{Sign}(vsk, ipk_j)$ for all $ipk_j \in PK$, where (vsk, vpk) is the verifier's key pair. Instead of performing an or-proof, the user can now show that she knows a signature, issued by the verifier, on the public key of the issuer that issued the user's credential. That is, the user can now send $\text{NIZK}[(x = vpk, w = (\sigma, ipk, cred, m) \colon \mathsf{Verify}(ipk, cred, m) \wedge \mathsf{Verify}(vpk, \sigma, ipk)]$, which is independent of the number of accepted issuers, i.e., $|PK|$.

A remaining technicality is now that the same verifier may accept different issuers for different scenarios, which is why every σ_j needs to be bound to the specific scenario. Using ephemeral signature keys (vpk, vsk) in each presentation session would require linear computation for computing and verifying the signatures; alternatively, a unique key pair per verifier could be used, and $\sigma_j \xleftarrow{\$} (vsk, (ipk_j, \mathsf{domain}))$ could be bound to a specific application domain. We finally opted for a combination, where the verifier is still key-less, yet signatures

$\mathbf{Exp}^{\mathsf{EUF-CMA}}_{\mathsf{Adv}}(\lambda)$
$pp \xleftarrow{\$} \Sigma.\mathsf{ParGen}(1^\lambda)$
$(sk, pk) \xleftarrow{\$} \Sigma.\mathsf{KGen}(pp)$
$\mathcal{Q} \leftarrow \emptyset$
$(m^*, \sigma^*) \xleftarrow{\$} \mathsf{Adv}^{\mathcal{O}_{\mathtt{sign}}}(pk)$
return 0, if $m^* \in \mathcal{Q}$
return 1, if $\Sigma.\mathsf{Verify}(pp, pk, \sigma^*, m^*) = 1$
return 0

where:
$\mathcal{O}_{\mathtt{sign}}(m)$:
$\sigma \xleftarrow{\$} \Sigma.\mathsf{Sign}(pp, sk, m)$
$\mathcal{Q} \leftarrow \mathcal{Q} \cup \{m\}$
return σ

Exp. 1: EUF-CMA experiment for digital signatures.

on public keys can be reused. This is done by letting the verifier define *policies* where a policy consists of signatures on all *ipk*'s for a specific domain, but different signing keys are used for different domains, and thus the respective signing keys can be discarded after publishing a policy.

We formalize the above intuition through a generic construction, for which we provide formal security proofs. We then give a concrete instantiation based on Groth's structure preserving signature scheme [36]. To ease readability, our basic construction focuses on the core functionality of anonymous credential systems; however, we finally also discuss how to achieve advances functionalities including non-frameability, revocation of credentials, and fine-granular linkability.

Outline. We briefly introduce notation and necessary background in Section 2. Then, in Section 3 we present the syntax and security requirements for issuer-hiding ABCs, before giving our generic construction and security proofs in Section 4. A concrete instantiation is presented in Section 5 and numerous possible extensions are discussed in Section 6. We finally conclude in Section 7.

2 Preliminaries

We denote the main security parameter by λ. We write $a \xleftarrow{\$} \mathsf{A}$ to denote that a is the output of a potentially randomized algorithm A and $v \xleftarrow{\$} \mathcal{S}$ to denote that v is uniformly sampled at random from a set $\mathcal{S}$. If not explicitly stated otherwise, all algorithms are assumed to be polynomial-time (PPT).

2.1 Digital Signatures

A digital signature scheme consists of four algorithms:

- $pp \xleftarrow{\$} \Sigma.\mathsf{ParGen}(1^\lambda)$ generates public parameters pp.
- $(sk, pk) \xleftarrow{\$} \Sigma.\mathsf{KGen}(pp)$ generates a secret key sk and a public key pk.
- $\sigma \xleftarrow{\$} \Sigma.\mathsf{Sign}(pp, sk, m)$ creates a signature σ on message m.
- $b \leftarrow \Sigma.\mathsf{Verify}(pp, pk, \sigma, m)$ verifies the signature.

Following Goldwasser et al [34], we require a digital signature scheme to be existentially unforgeable, meaning that no adversary can efficiently come up with a valid signature on a new message of the adversary's choice, even if it is given

access to a signing oracle that may sign an arbitrary number of messages chosen by the adversary:

Definition 1. *A digital signature scheme is* EUF-CMA secure *if and only if for every PPT adversary* Adv *there exists a negligible function* negl *such that:*

$$\Pr\left[\mathbf{Exp}^{\mathsf{EUF-CMA}}_{\mathsf{Adv}}(\lambda) = 1\right] \leq \mathsf{negl}(\lambda)\,,$$

where the experiment is as defined in Experiment 1.

Structure-preserving signatures. We recall the randomizable structure-preserving signature scheme by Groth [36]. While the scheme is able to sign matrices of group elements, we only require it to sign a single group element. Similar to Camenisch et al. [13], we consider the scheme in two variants: Groth_1 signs elements of $\mathbb{G}_1$ (and its public keys live in $\mathbb{G}_2$), and $\mathsf{Groth2}$, which signs elements of $\mathbb{G}_2$ (with public keys in $\mathbb{G}_1$). We describe Groth_1. The other scheme, Groth_2, can be obtained easily by switching the roles of $\mathbb{G}_1$ and $\mathbb{G}_2$.

- $\mathsf{Groth}_1.\mathsf{ParGen}(1^\lambda)$ generates public parameters pp consisting of a bilinear group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\mathrm{T}, e, p, G, \tilde{G})$ of prime order p with generators $G \in \mathbb{G}_1, \tilde{G} \in \mathbb{G}_2$, and a random element $Y \xleftarrow{\$} \mathbb{G}_1$.
- $\mathsf{Groth}_1.\mathsf{KGen}(pp)$ generates a secret key $sk_{\mathrm{sps}} = v \xleftarrow{\$} \mathbb{Z}_p^*$ and the corresponding public key $pk_{\mathrm{sps}} = \tilde{V} = \tilde{G}^v$.
- $\mathsf{Groth}_1.\mathsf{Sign}(pp, sk_{\mathrm{sps}}, M)$ chooses $r \xleftarrow{\$} \mathbb{Z}_p^*$ and outputs the signature $\sigma = (\tilde{R}, S, T) = (\tilde{G}^r, (Y \cdot G^v)^{1/r}, (Y^v \cdot M)^{1/r})$.
- $\mathsf{Groth}_1.\mathsf{Rand}(pp, \sigma)$ chooses $r' \xleftarrow{\$} \mathbb{Z}_p^*$ and outputs $\sigma' = (\tilde{R}', S', T') = (\tilde{R}^r, S^{1/r}, T^{1/r})$.
- $\mathsf{Groth}_1.\mathsf{Verify}(pp, pk_{\mathrm{sps}}, \sigma, M)$ checks that $e(S, \tilde{R}) = e(Y, \tilde{G}) \cdot e(G, \tilde{V})$ and $e(T, \tilde{R}) = e(Y, \tilde{V}) \cdot e(M, \tilde{G})$.

This construction is EUF-CMA secure in the generic group model [36].

2.2 Zero-Knowledge Proofs

A non-interactive zero-knowledge proof of knowledge (NIZK) allows a prover to generate a cryptographic proof that he knows a secret witness w such that $(x, w) \in \mathcal{R}$ for some binary relation $\mathcal{R}$ and a public statement x, without revealing any additional information about w than what is already revealed by x. We denote the language associated with $\mathcal{R}$ by L.

Formally, a NIZK consists of three algorithms:

- $pp \xleftarrow{\$} \Pi.\mathsf{ParGen}(1^\lambda)$ generates the public parameters pp.
- $\pi \xleftarrow{\$} \Pi.\mathsf{Prove}(pp, x, w, \mathsf{ctx})$ generates a non-interactive zero-knowledge proof of knowledge π of w such that $(x, w) \in \mathcal{R}$ bound to ctx.
- $b \leftarrow \Pi.\mathsf{Verify}(pp, x, \mathsf{ctx}, \pi)$ verifies a proof π.

$\mathbf{Exp}^{\mathsf{zero-knowledge}}_{\mathsf{Adv}}(\lambda)$
$(pp, \tau) \xleftarrow{\$} \mathsf{Sim}_1(1^\lambda)$
$b \xleftarrow{\$} \{0,1\}$
$b^* \xleftarrow{\$} \mathsf{Adv}^{\mathcal{O}_b}(pp)$
return 1, if $b = b^*$
return 0

where:
$\mathcal{O}_0(x, w, \mathsf{ctx})$:
return $\pi \xleftarrow{\$} \Pi.\mathsf{Prove}(pp, x, w, \mathsf{ctx})$, if $(x, w) \in \mathcal{R}$
return $\perp$
$\mathcal{O}_1(x, w, \mathsf{ctx})$:
return $\pi \xleftarrow{\$} \mathsf{Sim}_2(pp, \tau, x, \mathsf{ctx})$, if $(x, w) \in \mathcal{R}$
return $\perp$

Exp. 2: Zero-knowledge experiment for NIZKs.

$\mathbf{Exp}^{\mathsf{SimSoundExt}}_{\mathsf{Adv}}(\lambda)$
$(pp, \tau, \zeta) \xleftarrow{\$} \mathsf{Ext}_1(1^\lambda)$
$Q \leftarrow \emptyset$
$(x^*, \mathsf{ctx}^*, \pi^*) \xleftarrow{\$} \mathsf{Adv}^{\mathcal{O}_{\mathtt{SIM}}}(pp)$
$w^* \xleftarrow{\$} \mathsf{Ext}_2(pp, \zeta, x^*, \mathsf{ctx}^*, \pi^*)$
return 1, if:
$\Pi.\mathsf{Verify}(pp, x^*, \mathsf{ctx}^*, \pi^*) = 1$,
$(x^*, w^*) \notin \mathcal{R}$, and
$(x^*, \mathsf{ctx}^*) \notin Q$
return 0

where:
$\mathcal{O}_{\mathtt{SIM}}(x, \mathsf{ctx})$:
$\pi \xleftarrow{\$} \mathsf{Sim}_2(pp, \tau, x, \mathsf{ctx})$
$Q \leftarrow Q \cup \{(x, \mathsf{ctx})\}$
return π

Exp. 3: Simulation-sound extractability experiment for NIZKs.

Besides correctness, we will require zero-knowledge and simulation-sound extractability from all NIZKs.

Informally, the zero-knowledge property ensures that the receiver of a NIZK does not learn anything beyond what is already revealed by the statement itself.

Definition 2. *A non-interactive proof system Π satisfies* zero-knowledge *for a relation $\mathcal{R}$, if and only if for every PPT adversary* Adv *there exists a PPT simulator* $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$ *such that there exists negligible functions* negl_1 *and* negl_2 *such that:*

$$\left| \Pr\left[\mathsf{Adv}(pp) = 1 : pp \xleftarrow{\$} \Pi.\mathsf{ParGen}(1^\lambda)\right] - \Pr\left[\mathsf{Adv}(pp) = 1 : (pp, \tau) \xleftarrow{\$} \mathsf{Sim}_1(1^\lambda)\right] \right| \leq \mathsf{negl}_1(\lambda)\,,$$

and

$$\left| \Pr\left[\mathbf{Exp}^{\mathsf{zero-knowledge}}_{\mathsf{Adv}}(\lambda) = 1\right] - \frac{1}{2} \right| \leq \mathsf{negl}_2(\lambda)\,,$$

where the experiment is as defined in Experiment 2.

Intuitively, simulation-sound extractability requires that any adversary that can generate a valid proof must also know a valid witness for this statement, even if it has previously seen arbitrarily many (simulated) proofs of potentially false

statements. Note that the original definition of Groth [35], combining simulation-soundness [45] and proofs of knowledge [28], is stronger than ours in the sense that the adversary also gets access to the extraction trapdoor; however, similar to previous work [1, 29, 30] the following slightly weaker definition is sufficient for our purposes. Furthermore, the inclusion of a context ctx essentially makes the NIZK a *signature* of knowledge [23].

Definition 3. *A zero-knowledge non-interactive proof system Π satisfies* simulation-sound extractability *for a relation $\mathcal{R}$, if and only if for every PPT adversary Adv there exists a PPT extractor $\mathsf{Ext} = (\mathsf{Ext}_1, \mathsf{Ext}_2)$ such that there exists a negligible function negl such that:*

$$\Big|\Pr\left[\mathsf{Adv}(pp,\tau) = 1 : (pp,\tau) \xleftarrow{\$} \mathsf{Sim}_1(1^\lambda)\right] - \Pr\left[\mathsf{Adv}(pp,\tau) = 1 : (pp,\tau,\zeta) \xleftarrow{\$} \mathsf{Ext}_1(1^\lambda)\right]\Big| = 0\,,$$

and

$$\Pr\left[\mathbf{Exp}^{\mathsf{SimSoundExt}}_{\mathsf{Adv}}(\lambda) = 1\right] \leq \mathsf{negl}(\lambda)\,,$$

where the experiment is as defined in Experiment 3 and $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$ is as in Definition 2.

For notational convenience, we use the following notation for NIZKs, initially introduced by Camenisch and Stadler [21]. In this notation, a statement like

$$\mathrm{NIZK}\left[(\alpha,\beta,\gamma) : y_1 = g_1^\alpha g_2^\beta \wedge y_2 = g_1^\alpha g_3^\gamma \wedge \alpha \geq \gamma\right](\mathsf{ctx})$$

denotes a non-interactive zero-knowledge proof of knowledge, bound to the context ctx, of values α, β, γ such that the relation on the right hand side is satisfied. We also omit the public proof parameters pp.

3 Framework for Issuer-Hiding ABCs

We next define the syntax for issuer-hiding attribute-based credential systems, and then formalize the security properties required from such a system.

3.1 Syntax

An issuer-hiding ABC system is specified by eight algorithms. Initially, the parameters are generated by ParGen. Having generated a key pair using IKGen, an issuer can then issue credentials on attributes to a user by means of Issue; users can verify the received credential locally by VfCred in order to detect malformed credentials. To define the set of accepted issuers, a verifier generates a policy using $\mathsf{PresPolicy}$, which can be checked for well-formedness using $\mathsf{VfPolicy}$ by everyone. Finally, holding a credential from an issuer and a policy from the verifier, a user uses $\mathsf{Present}$ to derive a presentation token, which is verified by Verify. The inputs and outputs of the algorithms are introduced in the following:

Parameter generation. The public parameters are generated as:

$$pp \xleftarrow{\$} \mathsf{ParGen}(1^\lambda).$$

The public parameters are assumed to be implicit input to all algorithms presented in the following. We assume pp in particular specifies the number L of attributes that may be certified per credential, as well as the attribute space $\mathbb{A}$.

Key generation. Issuers compute their respective private and public keys as:

$$(isk, ipk) \xleftarrow{\$} \mathsf{IKGen}(pp)\,.$$

Issuance. The issuer creates a credential $cred$ on attributes $\vec{a}$ as follows:

$$cred \xleftarrow{\$} \mathsf{Issue}(isk, \vec{a})\,.$$

For the sake of simplicity, this process is modeled as a non-interactive algorithm as opposed an interactive protocol between the issuer and the user.We assume that both entities have previously set up their respective keys, and that they agreed on attributes $\vec{a} = (\vec{a}_1, \ldots, \vec{a}_L)$ to be certified.

Credential verification. The validity of a credential can be checked as follows:

$$b \xleftarrow{\$} \mathsf{VfCred}(cred, \vec{a}, ipk)\,.$$

Presentation policies. Verifiers can define presentation policies defining sets of issuers they are willing to accept for certain presentation sessions:

$$pol \xleftarrow{\$} \mathsf{PresPolicy}(\{ipk_i\})\,.$$

Note that pol only defines the sets of issuers accepted by a verifier, but not, e.g., which attributes a verifier needs to disclose. By this, pol can be reused for multiple contexts, reducing the number of policies.

Policy verification. Presentation policies can be checked for validity as follows:

$$b \leftarrow \mathsf{VfPolicy}(pol, \{ipk_i\})\,.$$

Presentation. For practical reasons, we only focus on non-interactive presentation protocols. Having agreed on a presentation policy which has been verified by the user, she computes a presentation token:

$$pt \xleftarrow{\$} \mathsf{Present}(ipk, cred, \phi, \vec{a}, pol, \mathsf{ctx})\,.$$

The verifier then validates the token as:

$$b \leftarrow \mathsf{Verify}(pt, \phi, pol, \mathsf{ctx})\,.$$

Here, $\phi : \mathbb{A}^L \to \{0,1\}$ is a predicate over the user's attributes that needs to be satisfied in order to pass verification, i.e., verification only passes if $\phi(\vec{a}) = 1$. For instance, ϕ might require that some a_i equals some previously agreed value, corresponding to the disclosure of this attribute, or that $a_i \in [l, r]$ for some bounds l and r. Finally, the purpose of ctx is to define a context in which the presentation token is accepted, e.g., a session identifier or a random nonce to avoid replay attacks or similar.

Policies will typically be long-lived, and it thus not necessary for a user to verify the policy every time before computing a presentation token. We thus do not consider these computational costs as part of the verification algorithm.

3.2 Security Definitions

We next define necessary security properties for an issuer-hiding ABC system.

Correctness. We omit a formal definition here, as the requirements are what one would expect: if all parties follow the protocol specifications during all phases, any presentation token computed by the user will be accepted by the verifier.

Unforgeability. Unforgeability requires that it is infeasible for an adversary to generate a valid presentation token, if it has not previously received a credential on attributes satisfying ϕ from one of the accepted issuers, or a presentation token for the same (ctx, ϕ, *pol*).

In the following definition, note that while the challenge policy pol^* may only include honest issuers' keys, the adversary may request presentation tokens for arbitrary sets of *ipk*'s from the presentation oracle $\mathcal{O}_{\texttt{present}}$, covering the case of adversarial issuers.

Definition 4. *An issuer-hiding ABC system satisfies* unforgeability*, if and only if for every PPT adversary* Adv *and every number of issuers*[6] n_I *there exists a negligible function* negl *such that:*

$$\Pr\left[\mathbf{Exp}^{\mathsf{Unforgeability}}_{\mathsf{Adv}}(\lambda, n_\mathsf{I}) = 1\right] \leq \mathsf{negl}(\lambda)\,,$$

where the experiment is as defined in Experiment 4.

Unlinkability. Unlinkability requires that no two user actions can be linked by an adversary. This even needs to hold if the adversary has full control over verifiers, issuers, and the user's credential. In the experiment (cf. Experiment 5), we thus let the adversary output two sets of credentials, attributes, and respective issuers, as well as a presentation policy *pol*, a predicate ϕ, and the issuers' public keys. Upon receiving a presentation token derived from one of the two credentials, the adversary must not be able to decide which underlying credential was used, as long as both credentials are valid and consistent with ϕ.

[6] Parametrizing Experiment 4 by n_I is for notational convenience only. Modifying the experiment and the proofs in the remainder of this paper to support an arbitrary polynomial number of issuers is straightforward.

$\mathbf{Exp}^{\mathsf{Unforgeability}}_{\mathsf{Adv}}(\lambda, n_{\mathsf{I}})$
$pp \xleftarrow{\$} \mathsf{ParGen}(1^\lambda)$
$Q_{\mathtt{issue}} \leftarrow \emptyset$, $Q_{\mathtt{present}} \leftarrow \emptyset$, $Q_{\mathtt{reveal}} \leftarrow \emptyset$
$(isk_i, ipk_i) \xleftarrow{\$} \mathsf{IKGen}(pp)$ for $i = 1, \ldots, n_{\mathsf{I}}$
$(I^*, st) \xleftarrow{\$} \mathsf{Adv}^{\mathcal{O}_{\mathtt{issue}}, \mathcal{O}_{\mathtt{present}}, \mathcal{O}_{\mathtt{reveal}}}(pp, \{ipk_i\}_{i=1}^{n_{\mathsf{I}}})$
$pol^* \xleftarrow{\$} \mathsf{PresPolicy}(I^*)$
$(pt^*, \phi^*, \mathsf{ctx}^*) \xleftarrow{\$} \mathsf{Adv}^{\mathcal{O}_{\mathtt{issue}}, \mathcal{O}_{\mathtt{present}}, \mathcal{O}_{\mathtt{reveal}}}(st, pol^*)$
where the oracles are defined as follows:
$\mathcal{O}_{\mathtt{issue}}(i_j, \vec{a}_j)$
$cred_j \xleftarrow{\$} \mathsf{Issue}(isk_{i_j}, \vec{a}_j)$
add $(i_j, \vec{a}_j)$ to $Q_{\mathtt{issue}}$
$\mathcal{O}_{\mathtt{present}}(j, pol, \phi, \mathsf{ctx})$
add $(pol, \phi, \mathsf{ctx})$ to $Q_{\mathtt{present}}$
return $pt \xleftarrow{\$} \mathsf{Present}(cred_j, ipk_{i_j}, \vec{a}_j, \phi, pol, \mathsf{ctx})$
$\mathcal{O}_{\mathtt{reveal}}(j)$
add $(i_j, \vec{a}_j)$ to $Q_{\mathtt{reveal}}$
return $cred_j$
return 1 if:
$I^* \subseteq \{ipk_i\}_{i=1}^{n_{\mathsf{I}}}$
$\mathsf{Verify}(pt^*, pol^*, \phi^*, \mathsf{ctx}^*) = 1$
$(pol^*, \phi^*, \mathsf{ctx}^*) \notin Q_{\mathtt{present}}$
$\nexists (i_j, \vec{a}_j) \in Q_{\mathtt{reveal}}$ such that $\phi^*(\vec{a}_j) = 1$ and $ipk_{i_j} \in I^*$
return 0

Exp. 4: Unforgeability experiment

$\mathbf{Exp}^{\mathsf{Unlinkability}}_{\mathsf{Adv}}(\lambda)$
$b \xleftarrow{\$} \{0, 1\}$
$pp \xleftarrow{\$} \mathsf{ParGen}(1^\lambda)$
$(\{cred_l, \vec{a}_l, i_l\}_{l=0}^{1}, pol, \phi, \{ipk_i\}, \mathsf{ctx}, st) \xleftarrow{\$} \mathsf{Adv}(pp)$
$pt^* \xleftarrow{\$} \mathsf{Present}(\{ipk_i\}, cred_b, \phi, \vec{a}_b, pol, \mathsf{ctx})$
$b^* \xleftarrow{\$} \mathsf{Adv}(pt^*, st)$
return 1 if and only if:
$b = b^*$,
$\mathsf{VfCred}(cred_l, \vec{a}_l, ipk_{i_l}) = 1$ for $l \in \{0, 1\}$,
$\phi(\vec{a}_l) = 1$ for $l \in \{0, 1\}$, and
$\mathsf{VfPolicy}(pol, \{ipk_i\}) = 1$
return $b' \xleftarrow{\$} \{0, 1\}$

Exp. 5: Unlinkability experiment

Definition 5. *An issuer-hiding ABC system satisfies* unlinkability*, if and only if for every PPT adversary* Adv *there exists a negligible function* negl *such that:*

$$\left| \Pr\left[\mathbf{Exp}^{\mathsf{Unlinkability}}_{\mathsf{Adv}}(\lambda) = 1 \right] - \frac{1}{2} \right| \leq \mathsf{negl}(\lambda)\,,$$

where the experiment is as defined in Experiment 5.

$\underline{\mathsf{ParGen}(1^\lambda)}$. Return $pp \xleftarrow{\$} \Sigma.\mathsf{ParGen}(1^\lambda)$.

$\underline{\mathsf{IKGen}(pp)}$. Return $(isk, ipk) \xleftarrow{\$} \Sigma_\mathsf{I}.\mathsf{KGen}(pp)$.

$\underline{\mathsf{Issue}(isk, \vec{a})}$. Return $cred \xleftarrow{\$} \Sigma_\mathsf{I}.\mathsf{Sign}(pp, isk, \vec{a})$.

$\underline{\mathsf{VfCred}(cred, \vec{a}, ipk)}$. Return 1 if $\Sigma_\mathsf{I}.\mathsf{Verify}(pp, ipk, cred, \vec{a}) = 1$. Otherwise, return 0.

$\underline{\mathsf{PresPolicy}(\{ipk_i\})}$. Generate a signature key pair $(vsk, vpk) \xleftarrow{\$} \Sigma_\mathsf{V}.\mathsf{KGen}(pp)$ and compute the signature $\sigma_i \xleftarrow{\$} \Sigma_\mathsf{V}.\mathsf{Sign}(pp, vsk, ipk_i)$. Return

$$pol = (vpk, \{(ipk_i, \sigma_i)\}) .$$

$\underline{\mathsf{VfPolicy}(pol, \{ipk_i\})}$. Parse pol as $(vpk, \{(ipk_i, \sigma_i)\})$. Return 1 if and only if

$$\Sigma_\mathsf{V}.\mathsf{Verify}(pp, vpk, \sigma_i, ipk_i) = 1 \text{ for all } i .$$

Otherwise, return 0.

$\underline{\mathsf{Present}(cred, ipk, \vec{a}, \phi, pol, \mathsf{ctx})}$. Parse pol as $(vpk, \{(ipk_i, \sigma_i)\})$. Set j such that $ipk_j = ipk$. Return a presentation token pt as follows:

$$pt \xleftarrow{\$} \mathrm{NIZK}[(\sigma_j, ipk_j, cred, \vec{a}) : \Sigma_\mathsf{V}.\mathsf{Verify}(pp, vpk, \sigma_j, ipk_j) = 1 \ \wedge \ \Sigma_\mathsf{I}.\mathsf{Verify}(pp, ipk_j, cred, \vec{a}) = 1 \ \wedge \ \phi(\vec{a}) = 1](pol, \phi, \mathsf{ctx}) \quad (1)$$

$\underline{\mathsf{Verify}(pt, pol, \phi)}$. Return 1 if and only if pt verifies correctly. Otherwise, return 0.

Construction 1: Generic construction of issuer-hiding ABCs.

4 A Generic Construction

The following section describes a generic construction of issuer-hiding attribute-based credentials, and gives a formal security analysis of its security.

4.1 Construction

Let $\Sigma_\mathsf{I} = (\Sigma.\mathsf{ParGen}, \Sigma_\mathsf{I}.\mathsf{KGen}, \Sigma_\mathsf{I}.\mathsf{Sign}, \Sigma_\mathsf{I}.\mathsf{Verify})$ and $\Sigma_\mathsf{V} = (\Sigma.\mathsf{ParGen}, \Sigma_\mathsf{V}.\mathsf{KGen}, \Sigma_\mathsf{V}.\mathsf{Sign}, \Sigma_\mathsf{V}.\mathsf{Verify})$ be digital signature schemes (cf. Section 2.1) with a common parameter generation algorithm.

Our generic construction is now depicted in Construction 1. We refer to Section 1 for a detailed description of the intuition underlying this construction.

4.2 Security Analysis

Theorem 1. *If Σ_I and Σ_V are EUF-CMA secure and the* NIZK *is zero-knowledge and simulation-sound extractable, then Construction 1 is unforgeable.*

Intuitively, the adversary has two potential ways of breaking unforgeability: (1) he can forge a Σ_V signature on his own public key ipk' (that is not part of the

challenge policy pol^*), or (2) he can forge a credential by forging a Σ_{I} signature w.r.t. some honest issuer's public key ipk_i.

Proof. Let Adv be a PPT adversary. We first modify the unforgeability game by simulating all NIZK pt output by $\mathcal{O}_{\mathtt{present}}$. Because the NIZK is zero-knowledge, this increases the winning probability by at most a negligible amount. In the following, we argue that Adv's winning probability in the modified game is negligible.

In the (modified) unforgeability game, Adv outputs pol^* and (pt^*, ϕ^*). If Adv wins, we can apply the NIZK extractor to pt^* to extract a witness $(\sigma, ipk, cred, \vec{a})$. Let extractfail be the event that Adv wins but the extractor fails to output a valid witness. Let polforge be the event that Adv wins, the extractor does not fail, and the extracted ipk is not any honest issuer's public key, i.e. $ipk \notin \{ipk_i\}$. Let credforge be the event that Adv wins, the extractor does not fail, and the extracted ipk *is* one of the honest issuer's public keys, i.e. $ipk \in \{ipk_i\}$.

It holds that $\Pr[\mathsf{Adv}\text{ wins}] \leq \Pr[\mathsf{Adv}\text{ wins} \wedge \neg\text{extractfail}] + \Pr[\text{extractfail}] = \Pr[\text{polforge}] + \Pr[\text{credforge}] + \Pr[\text{extractfail}]$. Because the NIZK is simulation-sound extractable, $\Pr[\text{extractfail}]$ is negligible. We now show that both $\Pr[\text{polforge}]$ and $\Pr[\text{credforge}]$ are negligible, which will conclude this proof.

Type 1 adversaries. We construct an adversary B against Σ_{V}'s EUF-CMA security.

- B gets pp, pk as input and access to a signing oracle $\mathcal{O}_{\mathtt{sign}}$.
- B generates $(isk_i, ipk_i) \xleftarrow{\$} \mathsf{IKGen}(pp)$ for $i = 1, \ldots, n_{\mathsf{I}}$ and runs $(I^*, st) \xleftarrow{\$} \mathsf{Adv}^{\mathcal{O}_{\mathtt{issue}}, \mathcal{O}_{\mathtt{present}}, \mathcal{O}_{\mathtt{reveal}}}(pp, \{ipk_i\}_{i=1}^{n_{\mathsf{I}}})$. It answers oracle queries honestly using $\{isk_i\}$.
- B queries $\mathcal{O}_{\mathtt{sign}}$ for signatures σ_i on $ipk_i \in I^*$. It sets $pol^* = (pk, \{(ipk_i, \sigma_i)\})$ and runs $(pt^*, \phi^*, \mathsf{ctx}^*) \xleftarrow{\$} \mathsf{Adv}^{\mathcal{O}_{\mathtt{issue}}, \mathcal{O}_{\mathtt{present}}, \mathcal{O}_{\mathtt{reveal}}}(st, pol^*)$.
- If Adv's winning condition is not fulfilled, B aborts.
- Otherwise, B extracts a witness $(\sigma, ipk, cred, \vec{a})$ from pt^*.
- If $ipk \notin \{ipk_i\}$, B, it outputs (ipk, σ) as a forgery.

By construction, $\Pr[\mathsf{B}\text{ wins}] = \Pr[\text{polforge}]$ (note that if polforge occurs, B has not queried for a signature on ipk). Because Σ_{V} is EUF-CMA secure, $\Pr[\text{polforge}]$ is negligible.

Type 2 adversaries. We construct an adversary B against Σ_{I}'s EUF-CMA security.

- B gets pp, pk as input and access to a signing oracle $\mathcal{O}_{\mathtt{sign}}$.
- B chooses a random $k \xleftarrow{\$} \{1, \ldots, n_{\mathsf{I}}\}$.
- B sets $ipk_k = pk$ and generates $(isk_i, ipk_i) \xleftarrow{\$} \mathsf{IKGen}(pp)$ for $i \in \{1, \ldots, n_{\mathsf{I}}\} \setminus \{k\}$. It runs $(I^*, st) \xleftarrow{\$} \mathsf{Adv}^{\mathcal{O}_{\mathtt{issue}}, \mathcal{O}_{\mathtt{present}}, \mathcal{O}_{\mathtt{reveal}}}(pp, \{ipk_i\}_{i=1}^{n_{\mathsf{I}}})$.
  - It answers $\mathcal{O}_{\mathtt{issue}}(i_j, \vec{a}_j)$ by adding $(i_j, \vec{a}_j)$ to $Q_{\mathtt{issue}}$ (but not generating a credential).
  - It answers $\mathcal{O}_{\mathtt{present}}(j, pol, \phi, \mathsf{ctx})$ by creating a simulated NIZK pt (unless $\phi(\vec{a}_j) = 0$).

  - It answers $\mathcal{O}_{\texttt{reveal}}(j)$ with $cred_j$ as follows: if $i_j = k$, it queries $cred_j \xleftarrow{\$} \mathcal{O}_{\texttt{sign}}(\vec{a}_j)$. Otherwise, it computes $cred_j \xleftarrow{\$} \Sigma_{\mathsf{I}}.\mathsf{Sign}(pp, isk_{i_j}, \vec{a}_j)$. Repeat reveal queries for j are answered with the same value $cred_j$ every time.
- B generates $pol^* \xleftarrow{\$} \mathsf{PresPolicy}(I^*)$. Afterwards, it runs $(pt^*, \phi^*, \mathsf{ctx}^*) \xleftarrow{\$} \mathsf{Adv}^{\mathcal{O}_{\text{issue}}, \mathcal{O}_{\text{present}}, \mathcal{O}_{\text{reveal}}}(st, pol^*)$.
  - B answers oracle queries as above.
- If Adv's winning condition is not fulfilled, B aborts.
- Otherwise, B extracts a witness $(\sigma, ipk, cred, \vec{a})$ from pt^*.
- If $ipk = pk$, B outputs $(\vec{a}, \sigma)$ as a forgery.

Note that the way B answers oracle queries is consistent with the way the (modified) unforgeability experiment does so.

If credforge $\wedge$ $ipk = pk$, then $(\vec{a}, \sigma)$ is a valid forgery. σ is a valid signature on $\vec{a}$ by guarantee of the NIZK extractor. Furthermore, B has not queried for a signature on $\vec{a}$ (because $\phi^*(\vec{a}) = 1$ by guarantee of the extractor, but the winning condition guarantees that $\phi^*(\vec{a}') = 0$ for all signatures revealed through $\mathcal{O}_{\texttt{reveal}}$). Hence $\Pr[\mathsf{B} \text{ wins}] \geq \Pr[\text{credforge} \wedge ipk = pk] = \frac{1}{n_{\mathsf{I}}} \cdot \Pr[\text{credforge}]$. Because Σ_{I} is EUF-CMA secure, $\Pr[\text{credforge}]$ is negligible. □

Theorem 2. *If* NIZK *is zero-knowledge, then Construction 1 is unlinkable.*

Proof. Because the property follows almost immediately from the zero-knowledge property, we omit a full proof. Note that the unlinkability experiment ensures that the witnesses used when computing pt^* are valid for both $b = 0$ and $b = 1$ (for cases where the adversary does not output valid values, the experiment ends up outputting a random bit, not providing any advantage to the adversary). This means that the unlinkability experiment is computationally indistinguishable from one where pt^* is created by the NIZK simulator. In the latter, the view of Adv is independent of b. □

5 Concrete Instantiation

One possible instantiation of Construction 1 in Section 4 is with Groth's structure-preserving signatures (Section 2.1). This instantiation is inspired by delegatable credentials [13] where the issue of proving knowledge of a hidden signature on the hidden public key of another hidden signature also comes up (though in a different scenario).

Concretely, in this instantiation, the issuer signs attributes using hash-then-sign with the Pedersen hash $H(\vec{a}) = \prod_{i=1}^{L} H_i^{a_i}$ and the structure-preserving signature scheme Groth_1. The verifier signs valid issuer public keys using Groth_2. A presentation token is a Schnorr-style proof of knowledge [46] turned non-interactive using the Fiat-Shamir heuristic [31] which gives us a simulation-sound extractable NIZK. We assume that the statement and public values are included in the computation of the challenge in order to avoid malleability issues [6].

With these choices, the scheme works as specified in Construction 2.

$\underline{\mathsf{ParGen}(1^\lambda)}$. For $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_\mathrm{T}$ as in $\mathsf{Groth}_1.\mathsf{ParGen}$, choose $Y, H_i \xleftarrow{\$} \mathbb{G}_1$ and $\tilde{Y} \xleftarrow{\$} \mathbb{G}_2$. Define the hash function $H : \mathbb{Z}_p^L \to \mathbb{G}_1$, $H(\vec{a}) = \prod_{i=1}^L H_i^{a_i}$. Return $pp = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\mathrm{T}, e, p, G, \tilde{G}, Y, \tilde{Y}, (H_i)_{i=1}^L)$.

$\underline{\mathsf{IKGen}(pp)}$. Generate a Groth_1 key pair $(isk, ipk) = (v, \tilde{V}) \xleftarrow{\$} \mathsf{Groth}_1.\mathsf{KGen}(pp)$.

$\underline{\mathsf{Issue}(isk, \vec{a})}$. Return $cred = (\tilde{R}, S, T) \xleftarrow{\$} \mathsf{Groth}_1.\mathsf{Sign}(pp, v, H(\vec{a}))$.

$\underline{\mathsf{VfCred}(cred, \vec{a}, ipk)}$. Return whatever $\mathsf{Groth}_1.\mathsf{Verify}(pp, \tilde{V}, cred, H(\vec{a}))$ returns.

$\underline{\mathsf{PresPolicy}(\{ipk_i\})}$. Generate $(vsk, vpk) = (u, U) \xleftarrow{\$} \mathsf{Groth}_2.\mathsf{KGen}(pp)$ and $\sigma_i = (R_i, \tilde{S}_i, \tilde{T}_i) \xleftarrow{\$} \mathsf{Groth}_2.\mathsf{Sign}(pp, u, ipk_i)$. Return $pol = (U, \{(ipk_i, \sigma_i)\})$.

$\underline{\mathsf{VfPolicy}(pol, \{ipk_i\})}$. Parse pol as $(vpk, \{(ipk_i, \sigma_i)\})$. Return 1 if $\mathsf{Groth}_2.\mathsf{Verify}(pp, vpk, \sigma_i, ipk_i) = 1$ for all i. Otherwise, return 0.

$\underline{\mathsf{Present}(cred, ipk, \vec{a}, \phi, pol, \mathsf{ctx})}$. Parse pol as $(vpk, \{(ipk_i, \sigma_i)\})$. Let j be the index of the credential's issuer's public key, i.e., $ipk_j = ipk$. Randomize $cred$ and σ_j as

$$(\tilde{R}, S, T) \xleftarrow{\$} \mathsf{Groth}_1.\mathsf{Rand}(pp, cred) \quad \text{and} \quad (R_j, \tilde{S}_j, \tilde{T}_j) \xleftarrow{\$} \mathsf{Groth}_2.\mathsf{Rand}(pp, \sigma_j)\,.$$

Choose random blinding values $\alpha, \beta, \gamma, \delta \xleftarrow{\$} \mathbb{Z}_p^*$ and compute

- the blinded credential $(\tilde{R}, S', T') := (\tilde{R}, S^{1/\alpha}, T^{1/\beta})$ on $H(\vec{a})$ under the issuer's key ipk_j
- the issuer's blinded key $ipk'_j := ipk_j^{1/\gamma}$
- the blinded policy signature $(R_j, \tilde{S}_j, \tilde{T}'_j) := (R_j, \tilde{S}_j, \tilde{T}_j^{1/\delta})$ on V_j under the verifier's public key vpk

Compute a Schnorr-style proof π:

$$\begin{aligned}
\pi \xleftarrow{\$} \mathrm{NIZK}[(\alpha, \beta, \gamma, \delta, \vec{a}) :& \\
\mathsf{Groth}_1 \text{ credential check: } & e(S', \tilde{R})^\alpha = e(Y, \tilde{G}) \cdot e(G, ipk'_j)^\gamma \ \wedge \\
\mathsf{Groth}_1 \text{ credential check: } & e(T', \tilde{R})^\beta = e(Y, ipk'_j)^\gamma \cdot e(\prod_{i=1}^L H_i^{a_i}, \tilde{G}) \ \wedge \\
\mathsf{Groth}_2 \text{ policy check: } & e(R_j, \tilde{S}_j) = e(G, \tilde{Y}) \cdot e(vpk, \tilde{G}) \ \wedge \\
\mathsf{Groth}_2 \text{ policy check: } & e(R_j, \tilde{T}'_j)^\delta = e(vpk, \tilde{Y}) \cdot e(G, ipk'_j)^\gamma \ \wedge \\
\text{Attribute check: } & \phi(\vec{a}) = 1](pol, \phi, \mathsf{ctx})
\end{aligned}$$

Finally, return $pt = ((\tilde{R}, S', T'), ipk'_j, (R_j, \tilde{S}_j, \tilde{T}'_j), \pi)$.

$\underline{\mathsf{Verify}(pt, pol, \phi)}$. Return 1 if and only if pt verifies correctly. Otherwise, return 0.

Construction 2: Concrete instantiation of our generic construction.

5.1 Security Analysis

Theorem 3. *If* Groth_1 *and* Groth_2 *are EUF-CMA secure and the* NIZK *is zero-knowledge and simulation-sound extractable, then the concrete instantiation in Construction 2 is unforgeable and unlinkable.*

	Runtime	$\mathbb{G}_1$	$\mathbb{G}_2$	$\mathbb{G}_T$	Pairings
PresPolicy (10 issuers)	14 ms	3 027	11 448	0	100
PresPolicy (100 issuers)	115 ms	27 666	115 430	0	0
VfPolicy (10 issuers)	3 ms	0	0	20	60
VfPolicy (100 issuers)	27 ms	0	0	200	600
Issue	2 ms	1 684	278	0	0
Present	4 ms	3 327	1 206	4	7
Verify	3 ms	2 398	0	901	12

Table 1: Performance of Construction 2 on a Macbook Pro (i9-9980HK) with BN254 as the bilinear group. Other columns show the (device-independent) number of group operations (multiply and square operations, including those happening within exponentiations) and pairings performed.

Unforgeability and unlinkability follow from the generic construction's unforgeability and unlinkability, which we're instantiating.

If Groth_1 is EUF-CMA secure, then also the hash-then-sign version of it (which is what we are effectively using in the concrete construction) is EUF-CMA secure (under the discrete logarithm assumption, which is implied by security of Groth_1, the hash function H is collision-resistant). It remains to argue that Present is a good instantiation of the NIZK specified in the generic construction.

For the zero-knowledge property, $pt = ((\tilde{R}, S', T'), ipk'_j, (R_j, \tilde{S}_j, \tilde{T}'_j), \pi)$ can be simulated by choosing random $S', T', ipk'_j \xleftarrow{\$} \mathbb{G}_1$ and $\tilde{R}, \tilde{T}'_j \xleftarrow{\$} \mathbb{G}_2$, setting $(R_j, \tilde{S}_j) = (R_1^r, \tilde{S}_1^{1/r})$ for random $r \xleftarrow{\$} \mathbb{Z}_p^*$ (where $(R_1, \tilde{S}_1, \cdot) = ipk_1$), and then simulating a corresponding π using the NIZK simulator.

For the proof of knowledge property, note that from π one can extract $\alpha, \beta, \gamma, \delta, \vec{a}$ with properties that guarantee that $cred := (\tilde{R}, (S')^\alpha, (T')^\beta)$ is a valid Groth_1 signature on $H(\vec{a})$ under $ipk_j := (ipk'_j)^\gamma$ and that $\sigma_j := (R_j, \tilde{S}_j, (\tilde{T}'_j)^\delta)$ is a valid Groth_2 signature on ipk_j under vpk, and that $\phi(\vec{a}) = 1$. This means that we can extract a valid witness $\sigma_j, ipk_j, cred, \vec{a}$ from pt, as required.

5.2 Performance Evaluation

To practically evaluate our construction, we have implemented[7] a benchmarking prototype of Construction 2 using the Cryptimeleon library [9]. The results are shown in Table 1. For credentials, we are considering $L = 10$ random attributes, none of which are disclosed during presentation (which is the most expensive case of partial attribute disclosure). The policy consists of 10 or 100 valid issuers. This does not affect token verification, which is independent of policy size. Overall, performance is practical, especially given that VfPolicy only has to be run once for each new policy and can be delegated to a trusted party.

The sizes of all keys and tokens can be found in Table 2.

[7] Code available at `https://github.com/cryptimeleon/issuer-hiding-cred`.

	$\mathbb{G}_1$	$\mathbb{G}_2$	$\mathbb{G}_T$	$\mathbb{Z}_p$
Issuer secret key (isk)	-	-	-	1
Issuer public key (ipk)	-	1	-	-
Credential ($cred$)	2	1	-	-
Presentation policy (pol)	$I+1$	$3I$	-	-
Presentation token (pt)	3	4	-	$L+5$

Table 2: Number of group elements for the different keys and tokens in Construction 2, where I is the number of issuer keys accepted by a verifier, and L is the number of attributes certified in the credential.

6 Extensions

To simplify presentation, our main construction only focuses on the key functionality of an issuer-hiding ABC system. In the following, we discuss in detail how to achieve non-frameability and anonymity revocation, controlled linkability and pseudonyms, and credential revocation. Further extensions like updating of credentials [8] or advanced predicates over attributes [5,40] are omitted here because of space limitations.

Anonymity revocation and non-frameability. While ABCs are important to protect the users' privacy and related fundamental rights, the high anonymity guarantees may lead to misuse of credentials and associated rights. In order to prevent such misuse, anonymity revocation (or inspection) is an important advanced functionality. Anonymity revocation allows a predefined opener (or inspector) to identify the originator of a given presentation token pt [12,18]. Closely related to this is the notion of non-frameability, which guarantees that even if issuers and the opener collude, it is infeasible for them to convince any third party that a specific user computed a given pt unless she indeed did so. This notion is closely related to non-frameability for group signatures [10,26,39].

This feature is achieved by letting the user generate a secret private key $(upk, usk) \xleftarrow{\$} \mathsf{UKGen}(pp)$, and the opener a key pair $(opk, osk) \xleftarrow{\$} \Gamma.\mathsf{KGen}(pp)$ for an encryption scheme Γ. Upon issuance, the user would now embed upk as an additional attribute which is signed by the issuer. When computing a presentation token, the user would treat upk as an undisclosed attribute, yet still prove that she knows the corresponding usk. Furthermore, she would encrypt upk under the opener's key as $enc \xleftarrow{\$} \Gamma.\mathsf{Enc}(upk, opk)$, and prove that enc contains the same upk which is also certified in the credential.

More formally, the basic presentation token would then be extended as follows (changes to Eq. (1) are highlighted):

$$pt \xleftarrow{\$} \mathrm{NIZK}[(\sigma_j, ipk_j, cred, \vec{a}, \boxed{upk, usk, r}) : \Sigma_{\mathsf{V}}.\mathsf{Verify}(pp, vpk, \sigma_j, ipk_j) = 1 \ \wedge$$
$$\Sigma_{\mathsf{I}}.\mathsf{Verify}(pp, ipk_j, cred, (\vec{a}, \boxed{upk})) = 1 \ \wedge \ \boxed{\mathsf{VfUKey}(upk, usk) = 1} \ \wedge$$
$$\boxed{enc = \Gamma.\mathsf{Enc}(upk, opk; r)} \ \wedge \ \phi(\vec{a}) = 1](pol, \phi, \mathsf{ctx})$$

where $\mathsf{VfUKey}(upk, usk) = 1$ if and only if usk is the corresponding secret key for upk, and r denotes the random coins used for encryption. To inspect, the opener can now decrypt enc and prove that the decryption was performed correctly.

As a concrete instantiation compatible with our generic construction, the user's key pair could be $(upk, usk) \coloneqq (g^x, x)$, and the corresponding encryption scheme could be ElGamal encryption [**?**].

Controlled linkability. In the case of stateful applications, for instance, users may wish to be recognized yet not identified by a verifier. This can be achieved by using scope-exclusive pseudonyms [16], in which pseudonyms can be linked within a given scope, but not across different scopes. For a pseudonym system Ψ with user secret key usk, the users' public key would be $upk \leftarrow \Psi.\mathsf{Gen}(usk, \varepsilon)$, which similar to before is embedded into the credential. For a given scope sc, the user now shares $nym = \Psi.\mathsf{Gen}(usk, \mathsf{sc})$ with the verifier, and proves that this is consistent with the (undisclosed) upk.

The presentation token would now be computed as follows:

$$pt \xleftarrow{\$} \mathrm{NIZK}[(\sigma_j, ipk_j, cred, \vec{a}, \boxed{upk, usk}) : \Sigma_{\mathsf{V}}.\mathsf{Verify}(pp, vpk, \sigma_j, ipk_j) = 1 \ \wedge$$
$$\Sigma_{\mathsf{I}}.\mathsf{Verify}(pp, ipk_j, cred, (\vec{a}, \boxed{upk})) = 1 \ \wedge \ \boxed{upk = \Psi.\mathsf{Gen}(usk, \varepsilon)} \ \wedge$$
$$\boxed{nym = \Psi.\mathsf{Gen}(usk, \mathsf{sc})} \ \wedge \ \phi(\vec{a}) = 1](pol, \phi, \mathsf{ctx}, \boxed{\mathsf{sc}, nym})$$

A possible instantiation compatible with our construction is based on the pseudonym system by Camenisch et al. [16], where a pseudonym for a scope is computed as $nym \coloneqq H(\mathsf{sc})^{usk}$ for a random oracle H.

Revocation. In case of misuse, loss of privileges, or compromise of a credential, it may become necessary to invalidate an issued credential. Many approaches for revocation of anonymous credentials can be found in the literature. In the following we show how to incorporate an approach along the lines of Nakanishi et al. [42] into our solution. Their work follows a black-listing approach, where each credential contains a revocation handle rh which is never disclosed upon presentation. The revocation authority, holding a signature key pair (rpk, rsk), issues intervals $[a_i, b_i]$ of currently still valid revocation handles, together with signatures $\alpha_i \xleftarrow{\$} \Sigma_{\mathsf{R}}.\mathsf{Sign}(pp, rsk, a_i)$ and $\beta_i \xleftarrow{\$} \Sigma_{\mathsf{R}}.\mathsf{Sign}(pp, rsk, b_i)$. When computing a presentation token, the user now proves that the revocation handle embedded in her credential lies in some interval $[a_i, b_i]$ for which she knows corresponding signatures α_i and β_i. However, it may be the case that multiple revocation authorities, e.g., one per issuer, exist in our setting. We thus not only embed the rh but also the revocation authority's public key rpk as attributes in our credentials, and the user shows that the known signatures on the interval boundaries verify under this (undisclosed) signature key.

More formally, a presentation token would now be derived as follows:

$$
\begin{aligned}
pt \xleftarrow{\$} \mathrm{NIZK}[(\sigma_j, ipk_j, cred, \vec{a}, \ rpk, rh, \alpha, \beta, a, b\ ) : \ & \Sigma_{\mathsf{V}}.\mathsf{Verify}(pp, vpk, \sigma_j, ipk_j) = 1 \ \wedge \\
& \Sigma_{\mathsf{I}}.\mathsf{Verify}(pp, ipk_j, cred, (\vec{a}, \ rh, rpk\ )) = 1 \ \wedge \\
& \Sigma_{\mathsf{R}}.\mathsf{Verify}(pp, rpk, \alpha, a) = 1 \ \wedge \ \Sigma_{\mathsf{R}}.\mathsf{Verify}(pp, rpk, \beta, b) = 1 \ \wedge \\
& rh \in [a, b] \ \wedge \ \phi(\vec{a}) = 1](pol, \phi, \mathsf{ctx})
\end{aligned}
$$

7 Conclusion & Future Work

We introduced the notion of issuer-hiding anonymous credential system, which allows for a dynamic and ad-hoc definition of sets of issuers among whose credentials a user may stay anonymous during presentation—a feature with various application domains, ranging from student identities over national eIDs to remote attestation. We provided a generic construction where the communication and computational complexity during presentation is independent of the number of issuers, as well as an efficient instantiation.

Nevertheless, we identified some open research questions. While our construction requires a minor joint setup across different issuers to define some group generators and the number of attributes L, in real applications, e.g., different nation states may wish to include different numbers of attributes in their credentials, vary the order of attributes, or use alternative generators for security reasons. It would be interesting to see whether this can be achieved more efficiently than via the generic or-composition discussed in Section 1. Also, the size of verifier policies is currently linear in the number of accepted issuers. One approach to overcome this limitation could be to add the accepted public keys to an accumulator, for which users, knowing all accumulated values, could compute the witnesses themselves, resulting in constant-size policies if the *ipk*'s are assumed to be known. Instead of proving knowledge of a signature from the verifier, users would now prove that they know a witness for the public key that issued the credential. However, we are not aware of any accumulator and compatible signature scheme allowing for an efficient instantiation.

Acknowledgments. This work was in parts supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 871473 (KRAKEN) and 830929 (CyberSec4Europe), and by the German Research Foundation (DFG) within the Collaborative Research Centre On-The-Fly Computing (GZ: SFB 901/3) under the project number 160364472.

References

1. Abe, M., David, B., Kohlweiss, M., Nishimaki, R., Ohkubo, M.: Tagged one-time signatures: Tight security and optimal tag size. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. Springer, Heidelberg (Feb / Mar 2013)

2. Abraham, A., Hörandner, F., Omolola, O., Ramacher, S.: Privacy-preserving eID derivation for self-sovereign identity systems. In: Zhou, J., Luo, X., Shen, Q., Xu, Z. (eds.) ICICS 19. Springer, Heidelberg (Dec 2019)
3. Abraham, A., Theuermann, K., Kirchengast, E.: Qualified eid derivation into a distributed ledger based idm system. In: TrustCom/BigDataSE. IEEE (2018)
4. Belenkiy, M., Camenisch, J., Chase, M., Kohlweiss, M., Lysyanskaya, A., Shacham, H.: Randomizable proofs and delegatable anonymous credentials. In: Halevi, S. (ed.) CRYPTO 2009. Springer, Heidelberg (Aug 2009)
5. Bemmann, K., Blömer, J., Bobolz, J., Bröcher, H., Diemert, D., Eidens, F., Eilers, L., Haltermann, J., Juhnke, J., Otour, B., Porzenheim, L., Pukrop, S., Schilling, E., Schlichtig, M., Stienemeier, M.: Fully-featured anonymous credentials with reputation system. In: ARES. ACM (2018)
6. Bernhard, D., Pereira, O., Warinschi, B.: How not to prove yourself: Pitfalls of the Fiat-Shamir heuristic and applications to Helios. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. Springer, Heidelberg (Dec 2012)
7. Blömer, J., Bobolz, J.: Delegatable attribute-based anonymous credentials from dynamically malleable signatures. In: Preneel, B., Vercauteren, F. (eds.) ACNS 18. Springer, Heidelberg (Jul 2018)
8. Blömer, J., Bobolz, J., Diemert, D., Eidens, F.: Updatable anonymous credentials and applications to incentive systems. In: Cavallaro, L., Kinder, J., Wang, X., Katz, J. (eds.) ACM CCS 2019. ACM Press (Nov 2019)
9. Bobolz, J., Eidens, F., Heitjohann, R., Fell, J.: Cryptimeleon: A library for fast prototyping of privacy-preserving cryptographic schemes. IACR ePrint (2021)
10. Bootle, J., Cerulli, A., Chaidos, P., Ghadafi, E., Groth, J.: Foundations of fully dynamic group signatures. In: Manulis, M., Sadeghi, A.R., Schneider, S. (eds.) ACNS 16. Springer, Heidelberg (Jun 2016)
11. Brands, S.: Rethinking Public Key Infrastructure and Digital Certificates – Buildingin Privacy. Ph.D. thesis, Eindhoven Institute of Technology (1999)
12. Camenisch, J.: Concepts around privacy-preserving attribute-based credentials - making authentication with anonymous credentials practical. In: Privacy and Identity Management. Springer (2013)
13. Camenisch, J., Drijvers, M., Dubovitskaya, M.: Practical UC-secure delegatable credentials with attributes and their application to blockchain. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) ACM CCS 2017. ACM Press (Oct / Nov 2017)
14. Camenisch, J., Drijvers, M., Lehmann, A., Neven, G., Towa, P.: Short threshold dynamic group signatures. In: Galdi, C., Kolesnikov, V. (eds.) SCN 20. Springer, Heidelberg (Sep 2020)
15. 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 2015, Part II. Springer, Heidelberg (Nov / Dec 2015)
16. Camenisch, J., Krenn, S., Lehmann, A., Mikkelsen, G.L., Neven, G., Pedersen, M.Ø.: Formal treatment of privacy-enhancing credential systems. In: Dunkelman, O., Keliher, L. (eds.) SAC 2015. Springer, Heidelberg (Aug 2016)
17. Camenisch, J., Lehmann, A., Neven, G., Rial, A.: Privacy-preserving auditing for attribute-based credentials. In: Kutylowski, M., Vaidya, J. (eds.) ESORICS 2014, Part II. Springer, Heidelberg (Sep 2014)
18. Camenisch, J., Lysyanskaya, A.: An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. Springer, Heidelberg (May 2001)

19. Camenisch, J., Lysyanskaya, A.: A signature scheme with efficient protocols. In: Cimato, S., Galdi, C., Persiano, G. (eds.) SCN 02. Springer, Heidelberg (Sep 2003)
20. Camenisch, J., Lysyanskaya, A.: Signature schemes and anonymous credentials from bilinear maps. In: Franklin, M. (ed.) CRYPTO 2004. Springer, Heidelberg (Aug 2004)
21. Camenisch, J., Stadler, M.: Efficient group signature schemes for large groups (extended abstract). In: Kaliski Jr., B.S. (ed.) CRYPTO'97. Springer, Heidelberg (Aug 1997)
22. Camenisch, J., Van Herreweghen, E.: Design and implementation of the idemix anonymous credential system. In: Atluri, V. (ed.) ACM CCS 2002. ACM Press (Nov 2002)
23. Chase, M., Lysyanskaya, A.: On signatures of knowledge. In: Dwork, C. (ed.) CRYPTO 2006. Springer, Heidelberg (Aug 2006)
24. Chaum, D.: Untraceable electronic mail, return addresses, and digital pseudonyms. Commun. ACM **24**(2), 84–88 (1981)
25. Chaum, D.: Security without identification: Transaction systems to make big brother obsolete. Commun. ACM **28**(10), 1030–1044 (1985)
26. Chen, L., Pedersen, T.P.: New group signature schemes (extended abstract). In: Santis, A.D. (ed.) EUROCRYPT'94. Springer, Heidelberg (May 1995)
27. Crites, E.C., Lysyanskaya, A.: Delegatable anonymous credentials from mercurial signatures. In: Matsui, M. (ed.) CT-RSA 2019. Springer, Heidelberg (Mar 2019)
28. De Santis, A., Persiano, G.: Zero-knowledge proofs of knowledge without interaction (extended abstract). In: 33rd FOCS. IEEE Computer Society Press (Oct 1992)
29. Derler, D., Krenn, S., Samelin, K., Slamanig, D.: Fully collision-resistant chameleon-hashes from simpler and post-quantum assumptions. In: Galdi, C., Kolesnikov, V. (eds.) SCN 20. Springer, Heidelberg (Sep 2020)
30. Dodis, Y., Haralambiev, K., López-Alt, A., Wichs, D.: Efficient public-key cryptography in the presence of key leakage. In: Abe, M. (ed.) ASIACRYPT 2010. Springer, Heidelberg (Dec 2010)
31. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory **31**, 469–472 (1985)
32. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO'86. Springer, Heidelberg (Aug 1987)
33. de Fuentes, J.M., González-Manzano, L., Serna-Olvera, J., Veseli, F.: Assessment of attribute-based credentials for privacy-preserving road traffic services in smart cities. Pers. Ubiquitous Comput. **21**(5), 869–891 (2017)
34. Garman, C., Green, M., Miers, I.: Decentralized anonymous credentials. In: NDSS 2014. The Internet Society (Feb 2014)
35. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM Journal on Computing **17**(2), 281–308 (Apr 1988)
36. Groth, J.: Simulation-sound NIZK proofs for a practical language and constant size group signatures. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. Springer, Heidelberg (Dec 2006)
37. Groth, J.: Efficient fully structure-preserving signatures for large messages. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015, Part I. Springer, Heidelberg (Nov / Dec 2015)

38. Haböck, U., Krenn, S.: Breaking and fixing anonymous credentials for the cloud. In: Mu, Y., Deng, R.H., Huang, X. (eds.) CANS 19. Springer, Heidelberg (Oct 2019)
39. Khovratovich, D., Law, J.: Sovrin: digitial signatures in the blockchain area (2016), https://sovrin.org/wp-content/uploads/AnonCred-RWC.pdf
40. Krenn, S., Samelin, K., Striecks, C.: Practical group-signatures with privacy-friendly openings. In: ARES. ACM (2019)
41. Lipmaa, H.: On diophantine complexity and statistical zero-knowledge arguments. In: Laih, C.S. (ed.) ASIACRYPT 2003. Springer, Heidelberg (Nov / Dec 2003)
42. Moreno, R.T., Bernabé, J.B., Rodríguez, J.G., Frederiksen, T.K., Stausholm, M., Martínez, N., Sakkopoulos, E., Ponte, N., Skarmeta, A.F.: The OLYMPUS architecture - oblivious identity management for private user-friendly services. Sensors **20**(3), 945 (2020)
43. Nakanishi, T., Fujii, H., Hira, Y., Funabiki, N.: Revocable group signature schemes with constant costs for signing and verifying. In: Jarecki, S., Tsudik, G. (eds.) PKC 2009. Springer, Heidelberg (Mar 2009)
44. Paquin, C., Zaverucha, G.: U-prove cryptographic specification v1.1 (revision2). Technical report, Microsoft Corporation (Apr 2013)
45. Ringers, S., Verheul, E.R., Hoepman, J.H.: An efficient self-blindable attribute-based credential scheme. In: Kiayias, A. (ed.) FC 2017. Springer, Heidelberg (Apr 2017)
46. Sahai, A.: Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In: 40th FOCS. IEEE Computer Society Press (Oct 1999)
47. Schnorr, C.P.: Efficient signature generation by smart cards. Journal of Cryptology **4**(3), 161–174 (Jan 1991)
48. Sonnino, A., Al-Bassam, M., Bano, S., Meiklejohn, S., Danezis, G.: Coconut: Threshold issuance selective disclosure credentials with applications to distributed ledgers. In: NDSS 2019. The Internet Society (Feb 2019)
49. Yang, R., Au, M.H., Xu, Q., Yu, Z.: Decentralized blacklistable anonymous credentials with reputation. In: Susilo, W., Yang, G. (eds.) ACISP 18. Springer, Heidelberg (Jul 2018)