

Hide Now, Trace Later: Retrospective Attribution in Issuer-Hiding Credentials

Stephan Krenn^[0000–0003–2835–9093], Doryan Lesaignoux^[0009–0009–3854–1498], Omid Mir^[0000–0003–1691–5291], and Gabriele Spini^[0000–0002–8578–3707]

AIT Austrian Institute of Technology, Vienna, Austria
`{firstname.lastname}@ait.ac.at`

Abstract. Attribute-based credentials (ABCs) enable privacy-preserving authentication by allowing users to prove statements about certified attributes without revealing unnecessary information. However, while ABCs hide undisclosed attributes, the credential verification process inherently reveals the issuer’s identity, which can leak contextual information such as a user’s nationality or residency. Issuer-hiding ABCs address this by concealing the issuer during verification, but introduce a critical accountability problem: if an issuer is compromised or acts maliciously, previously accepted presentations originating from that issuer cannot be singled out without significant privacy violation and high burden on the central authority performing the inspection.

We propose an issuer-inspection mechanism that enables retrospective attribution while preserving privacy. An inspection authority can release keys allowing verifiers to locally identify presentations originating from a specific compromised issuer and a defined time period, without affecting other issuers or earlier time periods. Concretely, we formalize the syntax and security requirements of issuer inspection and provide a generic construction from standard cryptographic building blocks; we further present a concrete pairing-based instantiation using Groth’s randomizable structure-preserving signatures, Schnorr-style proofs and an adaptation of Gentry’s anonymous identity-based encryption. Our analysis of computational and communication costs indicates that the proposed functionality remains practical for realistic parameter choices.

Keywords: Attribute-based credentials · Anonymous credentials · Issuer hiding · Traceability · Accountability

1 Introduction

Attribute-based credentials (ABCs) allow users to obtain certified attributes from issuers and later prove statements about these attributes to verifiers without revealing unnecessary information. ABC systems support selective disclosure and unlinkable presentations, and therefore provide a strong basis for privacy-preserving authentication. That is, instead of revealing their full identity, users can prove only the properties that are required for a specific service. First envisioned by Chaum [17] and first realized by Camenisch et al. [13,15,16], modern systems come with rich functionality, including delegation [3,20,26,43], cloud-based [38,28], and revocation [13,1,45]. Driven by large-scale digital-identity initiatives such as eIDAS 2.0, there is increasing interest in deploying such privacy-preserving credentials in practice, accompanied by standardization efforts by the IETF [41], ISO/IEC [30], and ETSI [21].

However, the disclosed attributes are not the only information that can affect a user’s privacy: if a presentation only proves that a user is eligible for a certain service, knowing which organization issued the underlying credential may reveal additional information about the user. Issuer-hiding credentials [8,42,34,48] address this problem by hiding the concrete issuer behind a presentation. In the approach of Bobolz et al. [8], a verifier can define a policy containing a set of acceptable issuers, while the user proves that the credential was issued by one of them without revealing which one. This allows the verifier to express its trust requirements while preserving privacy with respect to the concrete issuer.

Hiding the issuer, however, creates a natural accountability problem. ABC systems normally assume that an issuer only issues credentials for attributes that it is authorized to certify. If an issuer is compromised or acts maliciously, this assumption no longer holds. Such an issuer may create arbitrary credentials containing false attributes, and these credentials are cryptographically indistinguishable from honestly issued ones. In an issuer-hiding system, a verifier additionally cannot determine which accepted presentations originate from this corrupted issuer: it only learns that the credential was issued by one of the issuers allowed by its policy.

Once the compromise becomes known, the affected issuer can simply be removed from future issuer policies [8]. This prevents credentials from this issuer from being used in future presentations. However, it does not solve the problem of presentations that were already accepted before the compromise was detected. Consider, for example, a service provider that offers reduced-price tickets to users who can prove that they are eligible for a senior discount. If one of several accepted issuers is compromised and issues fraudulent credentials, the service provider may accept a large number of invalid presentations before the compromise becomes known. Afterwards, the issuer can be excluded from the policy, but the service provider still does not know which of its previous presentations were based on credentials from the malicious issuer. Consequently, it is difficult to determine the damage caused by the issuer and to hold it accountable.

A natural way to address this problem is to use the classical concept of inspection, also called anonymity revocation, from anonymous credential systems [13,14,6]. In such systems, information that is hidden during normal verification can be recovered by a designated inspector under defined conditions. One could therefore make the issuer identity inspectable and ask the inspector to identify presentations that originate from a compromised issuer.

This approach, however, has several drawbacks for issuer-hiding credentials:

- First, the inspector would have to process individual presentation tokens. In a large system, there may be thousands of verifiers that accept credentials from hundreds of issuers and store large numbers of presentation tokens. If an issuer is later found to be compromised, potentially all relevant tokens would have to be sent to the inspector and checked individually. This makes the inspector a central scalability bottleneck exactly when a potentially large number of past presentations has to be analysed.
- Second, the inspector becomes a privacy bottleneck. To determine whether a presentation originates from one particular malicious issuer, the inspector may have to recover the issuer identity of every inspected token. As a result, the inspector learns information about honest presentations that is intentionally hidden from the verifier. In a system designed to hide issuer identities, giving a central party access to the issuer behind every inspected presentation is undesirable.
- Third, unrestricted inspection may reveal substantially more information than required. A compromised issuer does not necessarily imply that all credentials ever issued by it are invalid. For example, if the issuer is known to have been compromised only from a certain point in time, presentations before this point can still be considered legitimate. There is therefore no need to reveal the issuer of such older presentations. The same holds for all presentations issued by honest issuers.

What is needed is therefore a more selective and scalable mechanism for retrospective attribution. Assume that an issuer I^* is found to be compromised from epoch t^* onwards. An inspection authority should then be able to release issuer- and epoch-specific key material that allows verifiers to identify presentations originating from I^* during the affected epochs. Ideally, this key material can be made public. Each verifier could then locally check its own stored presentation tokens without sending them to the inspection authority. At the same time, presentations originating from other issuers should remain unaffected. Likewise, presentations involving I^* before the compromise epoch t^* should retain their issuer privacy. Only presentations related to the compromised issuer and the affected epochs should become recognizable.

Such a mechanism would allow the expensive retrospective analysis to be performed locally by the verifiers that actually hold the presentation tokens. At the same time, it would avoid turning the

inspector into a central privacy and scalability bottleneck. This would make it possible to preserve issuer privacy during normal operation while still enabling accountability when an issuer is later found to be malicious or compromised.

1.1 Our Contributions

Our contributions are as follows.

Retrospective local issuer tracing. We introduce, to the best of our knowledge, the first mechanism for retrospectively tracing the hidden issuer of previously accepted attribute-based credential presentations. Instead of requiring a central inspector to process stored presentations, an authority releases issuer- and epoch-specific tracing keys that allow verifiers to inspect their own records locally. This turns issuer tracing into a targeted and scalable operation, while preserving issuer privacy for all unrelated issuers and epochs.

Security foundations. We develop a security model for this new setting that captures the natural requirements of privacy, unforgeability, and accountability. Importantly, moving the actual tracing operation from the opening authority to the verifier changes the trust model and creates a new class of framing attacks: malicious presentations *together with manipulated opening keys* could otherwise be used to falsely blame an honest issuer. We identify the additional security requirement needed to rule out such attacks and formalize it for the underlying predicate-encryption layer.

Generic construction and instantiation. We show that retrospective local issuer tracing can be realized generically from standard cryptographic building blocks: digital signatures, non-interactive zero-knowledge proofs, and predicate encryption. We prove the resulting construction secure under the corresponding assumptions. We then give a concrete pairing-based realization using Groth signatures [27], Schnorr-style proofs [49] made non-interactive via Fiat–Shamir [22,5], and Gentry’s anonymous identity-based encryption [25,46] to realize the required equality predicate. We further show that the resulting encryption layer satisfies the additional security requirement needed for secure tracing.

Efficiency. Our evaluation shows that the additional tracing functionality remains practical even for large issuer sets. For a policy covering 3000 issuers and credentials with 10 attributes, a presentation takes approximately 126.45ms, verification approximately 54.23ms, and local opening approximately 6.34ms. These results indicate that fine-grained retrospective tracing can be added while preserving the practical usability of issuer-hiding credentials.

1.2 Related Work

Issuer hiding in anonymous credentials was introduced by Bobolz et al. [8] and has since been studied in several variants and under different assumptions [9,48,35]. These works aim at hiding the concrete issuer while still allowing a verifier to establish that the issuer satisfies an acceptance policy. Recently, Begum and Nakanishi [2] considered issuer revocation in issuer-hiding ABCs and showed how revoked issuers can be excluded efficiently using an accumulator. Such revocation mechanisms prevent credentials of a revoked issuer from being used in future presentations, but do not allow verifiers to retrospectively identify already accepted presentations originating from an issuer that is later found to be compromised. Yang et al. [50] study “privately accountable” multi-signatures for issuer-hiding credentials, where accountability is tied to identifying *the signers* involved in generating a credential, whereas our work targets retrospective identification of the hidden issuer of already accepted credential presentations by means of issuer- and epoch-specific tracing keys.

Inspection is a well-established mechanism for adding accountability to otherwise anonymous authentication. Anonymous credential systems already support optional anonymity revocation [13], and inspection was later included as a general feature of the Privacy-ABC architecture developed in ABC4Trust [6], with more recent solutions including, e.g., [29,31].

In all these approaches, however, tracing concerns the anonymous *user* or credential holder, rather than the hidden credential issuer. A closely related mechanism appears in group signatures, where signatures hide the identity of the group member but can be opened by a designated authority [18,4,39]. Traceable signatures refine this idea by allowing tracing information for a particular member to be released, such that signatures of this member can be recognized without opening signatures of other members [36]. Traceable ring signatures similarly restrict anonymity in settings without a group manager and allow repeated signing by the same member under a common tag to be detected and, under defined conditions, traced [24]. These mechanisms all provide controlled ways to reduce anonymity for accountability purposes. In contrast, our setting concerns the issuer hidden within an anonymous credential presentation rather than the user or signer, and requires inspection to be restricted jointly to a specific issuer and epoch while leaving all unrelated presentations unaffected.

2 Preliminaries

We denote the security parameter by $\lambda \in \mathbb{N}$ and its unary representation by 1^λ . Deterministic assignment is written ‘ $\leftarrow$ ’, while ‘ $\leftarrow_\$$ ’ denotes randomized output or uniform sampling. Vectors are denoted by $\vec{a} = (a_1, \dots, a_L)$, and $\perp$ denotes failure. Unless stated otherwise, all algorithms and adversaries are probabilistic polynomial-time (PPT). We write $[a = b]$ for the indicator that evaluates to 1 if $a = b$ and to 0 otherwise. We write $\varepsilon(\lambda)$ for a negligible function and denote the neutral element of a group $\mathbb{G}$ by $1_{\mathbb{G}}$.

We briefly recall the syntax and security properties of the building blocks underlying our generic construction, referring to the original literature for details.

Digital signatures. A digital signature scheme $\Sigma = (\text{Setup}, \text{KeyGen}, \text{Sign}, \text{Verify})$ consists of PPT algorithms for parameter generation, key generation, signing, and verification. Informally, EUF-CMA security requires that no PPT adversary, even with access to a signing oracle, can produce a valid signature on a message that it has not previously queried.

Non-interactive zero-knowledge proofs of knowledge. Let $\mathcal{L}$ be an NP language with corresponding relation $\mathcal{R}$. A non-interactive zero-knowledge proof system $\Pi = (\text{Setup}, \text{Prove}, \text{Verify})$ for $\mathcal{L}$ consists of three PPT algorithms, where **Setup** generates public parameters, **Prove** generates a proof π for a statement $x \in \mathcal{L}$ using a witness w such that $\mathcal{R}(x, w) = 1$, and **Verify** checks the validity of π . Informally, zero knowledge requires that proofs can be simulated without knowledge of a witness and are indistinguishable from honestly generated proofs. Simulation-sound extractability requires that, even after seeing simulated proofs, no PPT adversary can produce a fresh accepting proof for which an extractor cannot recover a witness w satisfying $\mathcal{R}(x, w) = 1$ [7].

Predicate encryption. Predicate encryption allows ciphertexts to be associated with attributes and decryption keys with predicates, such that decryption succeeds only if the predicate evaluates to true on the encrypted attribute [33].

Definition 1 (Predicate Encryption). Let $\mathcal{F} = \{f : \mathcal{X} \rightarrow \{0, 1\}\}$ be a family of predicates. A predicate encryption scheme consists of the following algorithms:

Setup $(1^\lambda) \rightarrow pp$: On input the security parameter λ , the algorithm generates the necessary parameters.

KeyGen $(pp) \rightarrow (mpk, msk)$: On input the parameter pp , the algorithm generates a master public and secret key pair.

Derive $(msk, f) \rightarrow sk_f$: On input the master secret key msk and a predicate $f \in \mathcal{F}$, the algorithm derives a predicate secret key.

Enc $(mpk, x, msg) \rightarrow ct$: On input the master public key mpk , an attribute $x \in \mathcal{X}$, and a message $msg \in \mathcal{M}$, the algorithm generates a ciphertext.

$\text{Dec}(\text{sk}_f, \text{ct}) \rightarrow \text{msg}/\perp$: On input a predicate secret key sk_f and a ciphertext ct , the algorithm returns the message if $f(x) = 1$.

Informally, adaptive attribute hiding requires that an adversary cannot distinguish which of two admissible attribute-message pairs was encrypted, even if it can obtain predicate keys before and after seeing the challenge ciphertext.

Definition 2 (Adaptive Attribute Hiding). Let $\Lambda = (\text{Setup}, \text{KeyGen}, \text{Derive}, \text{Enc}, \text{Dec})$ be a predicate encryption scheme for a predicate family $\mathcal{F}$ over an attribute space $\mathcal{X}$. We say that Λ satisfies adaptive attribute hiding if, for every PPT adversary $\mathcal{A}$, there exists a negligible function ε such that:

$$\Pr \left[w = 1 : \begin{array}{l} pp \leftarrow_{\$} \text{Setup}(1^\lambda), \\ (mpk, msk) \leftarrow_{\$} \text{KeyGen}(pp), Q_{\text{Derive}} \leftarrow \emptyset, \\ ((m_0, x_0), (m_1, x_1), \text{state}) \leftarrow_{\$} \mathcal{A}^{\mathcal{O}_{\text{Derive}}}(mpk), \\ b \leftarrow_{\$} \{0, 1\}, \text{ct}^* \leftarrow_{\$} \text{Enc}(mpk, x_b, m_b), \\ b' \leftarrow_{\$} \mathcal{A}^{\mathcal{O}_{\text{Derive}}}(\text{ct}^*, \text{state}), w \leftarrow_{\$} \{0, 1\}, \\ \text{if } \text{Adm}(Q_{\text{Derive}}, x_0, x_1, m_0, m_1) = 1 \text{ then } w \leftarrow [b = b'] \end{array} \right] - \frac{1}{2} \leq \varepsilon(\lambda).$$

Here, the oracle $\mathcal{O}_{\text{Derive}}$, on input $f \in \mathcal{F}$, computes $\text{sk}_f \leftarrow_{\$} \text{Derive}(msk, f)$, adds f to Q_{Derive} , and returns sk_f . Furthermore, $\text{Adm}(Q_{\text{Derive}}, x_0, x_1, m_0, m_1) = 1$ if, for all $f \in Q_{\text{Derive}}$, it holds that $f(x_0) = f(x_1)$ and, if $m_0 \neq m_1$, additionally $f(x_0) = f(x_1) = 0$.

Opening Soundness. In addition, we require a tailored property capturing that, for an honestly generated master public key, no ambiguous opening key can be constructed even by an adversary that obtains the master secret key. More precisely, a maliciously generated key that successfully opens a ciphertext for an attribute satisfying a predicate must not also open a ciphertext for an attribute not satisfying this predicate. This must hold even if the adversary chooses both attributes and fully controls the randomness used to generate one of the ciphertexts. While being non-standard, the property is naturally satisfied by many schemes, as we will argue in Section 5.

Definition 3 (Opening Soundness). Let $\Lambda = (\text{Setup}, \text{KeyGen}, \text{Derive}, \text{Enc}, \text{Dec})$ be a predicate-encryption scheme for a predicate family $\mathcal{F}$ over an attribute space $\mathcal{X}$ and message space $\mathcal{M}$. We say that Λ satisfies malicious-key opening soundness with respect to a fixed public message $1_{\mathcal{M}} \in \mathcal{M}$ if, for every PPT adversary $\mathcal{A}$, there exists a negligible function ϵ such that

$$\Pr \left[\begin{array}{l} f(x_0) = 1 \wedge f(x_1) = 0, \\ \text{Dec}(\text{sk}^*, c_0) = 1_{\mathcal{M}}, \\ \text{Dec}(\text{sk}^*, c_1) = 1_{\mathcal{M}} \end{array} : \begin{array}{l} pp \leftarrow_{\$} \text{Setup}(1^\lambda), \\ (mpk, msk) \leftarrow_{\$} \text{KeyGen}(pp), \\ (f, x_0, x_1, \text{sk}^*, r_1) \leftarrow_{\$} \mathcal{A}(mpk, msk), \\ c_0 \leftarrow_{\$} \text{Enc}(mpk, x_0, 1_{\mathcal{M}}), \\ c_1 \leftarrow_{\$} \text{Enc}(mpk, x_1, 1_{\mathcal{M}}; r_1) \end{array} \right] \leq \epsilon(\lambda).$$

3 Issuer-Hiding Credentials with Opening

We define the syntax of issuer-hiding ABCs with opening by extending the framework of Bobolz et al. [8] with opening functionality. Extensions of the interfaces are highlighted in Definition 4.

At a high level, issuers and openers first generate their respective key pairs using IKGen and OKGen . Issuers create credentials with Issue , which are checked by VfCred . Verifiers define and validate acceptable-issuer policies using PresPolicy and VfPolicy . Users then generate presentations with Present , which are checked by Verify . If presentations from a particular issuer ipk and epoch t must later be traced, the opener derives the corresponding opening key with OKDerive , and this key can then be used with Open to identify exactly those presentation tokens that match the targeted issuer and epoch.

Definition 4 (Issuer-Hiding ABCs with Opening). Let $\mathcal{X}$ be an attribute space. An issuer-hiding AC scheme with opening consists of the following PPT algorithms:

$\text{Setup}(1^\lambda) \rightarrow pp$: On input the security parameter λ , the public-parameter generation algorithm outputs the public parameters.

$\text{IKGen}(pp) \rightarrow (isk, ipk)$: On input the public parameters pp , the algorithm generates an issuer secret and public key pair.

$\text{OKGen}(pp) \rightarrow (osk, opk)$: On input the public parameters pp , the algorithm generates an opener secret and public key pair.

$\text{OKDerive}(osk, ipk, t) \rightarrow osk_{ipk,t}$: On input the opener secret key osk , an issuer public key ipk , and an epoch t , the algorithm derives an opening key for issuer ipk and epoch t .

$\text{Issue}(isk, \vec{a}) \rightarrow cred$: On input an issuer secret key isk and attributes $\vec{a} \in \mathcal{X}$, the algorithm issues a credential.

$\text{VfCred}(cred, \vec{a}, ipk) \rightarrow b$: On input a credential $cred$, attributes $\vec{a}$, and an issuer public key ipk , the algorithm outputs a bit indicating the validity of the credential.

$\text{PresPolicy}(ipk_i) \rightarrow pol$: On input a set of issuer public keys ipk_i , the algorithm generates a presentation policy specifying the issuers accepted for a presentation session.

$\text{VfPolicy}(pol, \{ipk_i\}) \rightarrow b$: On input a presentation policy pol and a set of issuer public keys $\{ipk_i\}$, the algorithm outputs a bit indicating the validity of the policy for the provided public keys.

$\text{Present}(opk, ipk, cred, \phi, \vec{a}, t, pol, ctx) \rightarrow pt$: On input the opener public key opk , an issuer public key ipk , a credential $cred$, a predicate ϕ , attributes $\vec{a}$, an epoch t , a presentation policy pol , and a context ctx , the algorithm generates a presentation token.

$\text{Verify}(pt, \phi, pol, t, opk, ctx) \rightarrow b$: On input a presentation token pt , a predicate ϕ , a presentation policy pol , an epoch t , the opener public key opk , and a context ctx , the algorithm computes a bit indicating the validity of the presentation.

Here, $\phi : \mathbb{A}^L \rightarrow \{0,1\}$ is a predicate over the user's attributes that must be satisfied for verification to succeed, i.e., successful verification requires $\phi(\vec{a}) = 1$.

$\text{Open}(pt, opk, osk_{ipk,t}, ipk, t, pol, \phi) \rightarrow b$: On input a presentation token pt , the opener public key opk , an opening key $osk_{ipk,t}$, an issuer public key ipk , an epoch t , a presentation policy pol , and a predicate ϕ , the algorithm outputs a bit indicating whether the token originates from issuer ipk in epoch t .

3.1 Security framework

We now present our security properties, inspired by [8] and adapted to our setting. Our main security properties, formalized in Definitions 5 to 7, are based on the oracles defined in Figure 1. Informally, $\mathcal{O}_{\text{Issue}}$ generates a fresh credential for given attributes and issuer, and returns a handle of the credential to $\mathcal{A}$. The adversary can now use $\mathcal{O}_{\text{Present}}$ to obtain presentations of these credentials, thereby simulating the behavior of honest users. Alternatively, $\mathcal{A}$ may decide to corrupt credentials, in which case they are fully disclosed, using $\mathcal{O}_{\text{Reveal}}$. Finally, $\mathcal{O}_{\text{Open}}$ allows the adversary to obtain opening keys for given issuers and epochs.

Correctness. We omit a formal definition, as correctness is standard: honestly issued credentials verify, honestly generated presentations are accepted, and opening correctly identifies the underlying issuer.

Unlinkability. We require that an adversary cannot determine which of two valid credentials was used to generate a presentation. The adversary may fully control the issuers and choose two credentials, possibly with different attributes and issuers, together with the policy pol , predicate ϕ , epoch t , and context ctx . Given a presentation generated from one of the two credentials, it should not identify the underlying credential or issuer, provided both credentials are valid, satisfy

Oracle $\mathcal{O}_{\text{Open}}(ipk, t, opk)$

- 1: Find $(opk, osk) \in Q_{\text{opener}}$
- 2: $osk_{ipk,t} \leftarrow_{\$} \text{OKDerive}(osk, ipk, t)$
- 3: $Q_{\text{Open}} \leftarrow Q_{\text{Open}} \cup \{(ipk, t, opk)\}$
- 4: **return** $osk_{ipk,t}$

Oracle $\mathcal{O}_{\text{Present}}(j, \text{pol}, \phi, t, \text{ctx}, opk)$

- 1: Find $(\text{cred}, ipk, \vec{a}, j) \in Q_{\text{Issue}}$
- 2: $\text{pt} \leftarrow_{\$} \text{Present}(opk, ipk, \text{cred}, \phi, \vec{a}, t, \text{pol}, \text{ctx})$
- 3: $Q_{\text{Present}} \leftarrow Q_{\text{Present}} \cup \{(\text{pt}, j, \text{pol}, \phi, t, opk, \text{ctx})\}$
- 4: **return** pt

Oracle $\mathcal{O}_{\text{Issue}}(ipk, \vec{a})$

- 1: Find $(isk, ipk) \in Q_{\text{ipk}}$
- 2: $j \leftarrow j + 1$
- 3: $\text{cred} \leftarrow \text{Issue}(isk, \vec{a})$
- 4: $Q_{\text{Issue}} \leftarrow Q_{\text{Issue}} \cup \{(\text{cred}, ipk, \vec{a}, j)\}$
- 5: **return** j

Oracle $\mathcal{O}_{\text{Reveal}}(j)$

- 1: Find $(\text{cred}, ipk, \vec{a}, j) \in Q_{\text{Issue}}$
- 2: $Q_{\text{Reveal}} \leftarrow Q_{\text{Reveal}} \cup \{j\}$
- 3: **return** cred

Fig. 1: Oracles used in the experiments. All oracles return $\perp$ if a lookup fails.

ϕ , and their issuers are accepted by pol . This must hold even if the adversary obtains opening keys for all issuer–epoch pairs except the two challenge pairs.

Since the challenge issuers are selected through indices into the issuer set validated by pol , both are necessarily accepted by the presentation policy.

Definition 5 (Unlinkability). *An issuer-hiding attribute-based credential system with opening satisfies unlinkability if, for every PPT adversary $\mathcal{A}$, there exists a negligible function ε such that $\left| \Pr[\text{Exp}_{\mathcal{A}}^{\text{Unlinkability}}(\lambda) = 1] - \frac{1}{2} \right| \leq \varepsilon(\lambda)$, where the experiment is defined in Experiment 1.*

Experiment 1 – $\text{Exp}_{\mathcal{A}}^{\text{Unlinkability}}(\lambda)$

- 1: $b \leftarrow_{\$} \{0, 1\}$
- 2: $Q_{\text{Open}} \leftarrow \emptyset$
- 3: $pp \leftarrow_{\$} \text{Setup}(1^\lambda)$
- 4: $(osk, opk) \leftarrow_{\$} \text{OKGen}(pp)$
- 5: $Q_{\text{opener}} \leftarrow \{(opk, osk)\}$
- 6: $(\{\text{cred}_l, \vec{a}_l, i_l\}_{l=0}^1, \text{pol}, \phi, \{ipk_i\}, t, \text{ctx}, \text{state}) \leftarrow_{\$} \mathcal{A}^{\mathcal{O}_{\text{Open}}}(pp, opk)$
- 7: $\text{pt}^* \leftarrow_{\$} \text{Present}(opk, ipk_{i_b}, \text{cred}_b, \phi, \vec{a}_b, t, \text{pol}, \text{ctx})$
- 8: $b^* \leftarrow_{\$} \mathcal{A}^{\mathcal{O}_{\text{Open}}}(\text{pt}^*, \text{state})$
- 9: **if** :
 - 10: (a) $\text{VfCred}(\text{cred}_l, \vec{a}_l, ipk_{i_l}) = 1$ for $l \in \{0, 1\}$,
 - 11: (b) $\phi(\vec{a}_l) = 1$ for $l \in \{0, 1\}$,
 - 12: (c) $\text{VfPolicy}(\text{pol}, \{ipk_i\}) = 1$ and
 - 13: (d) $(ipk_{i_l}, t, opk) \notin Q_{\text{Open}}$ for $l \in \{0, 1\}$
 - 14: **return** $[b = b^*]$
- 15: **else**
- 16: $b' \leftarrow_{\$} \{0, 1\}$
- 17: **return** b'

Unforgeability. Unforgeability requires that it is infeasible for an adversary to generate a valid presentation token unless it has previously obtained either a credential from an accepted issuer on attributes satisfying ϕ , or a presentation token for the same tuple $(\text{pol}, \phi, t, \text{ctx})$. Note that unforgeability only makes sense if the challenge policy pol contains only honestly generated issuer public keys. However, through the presentation oracle $\mathcal{O}_{\text{Present}}$, the adversary may request presentation tokens with respect to arbitrary sets of issuer public keys, involving adversarial issuers.

Definition 6 (Unforgeability). *An issuer-hiding attribute-based credential system with opening satisfies unforgeability if, for every PPT adversary $\mathcal{A}$, there exists a negligible function ε such that $\Pr[\text{Exp}_{\mathcal{A}}^{\text{Unforgeability}}(\lambda) = 1] \leq \varepsilon(\lambda)$, where the experiment is defined in Experiment 2.*

Experiment 2 $\text{Exp}_{\mathcal{A}}^{\text{Unforgeability}}(\lambda, n, m)$

```

1:  $pp \leftarrow_{\$} \text{Setup}(1^\lambda)$ 
2:  $j \leftarrow 0$ 
3:  $Q_{\text{Issue}} \leftarrow \emptyset, Q_{\text{Present}} \leftarrow \emptyset, Q_{\text{Reveal}} \leftarrow \emptyset$ 
4:  $(osk_i, opk_i) \leftarrow_{\$} \text{OKGen}(pp)$  for  $i = 1, \dots, n$ 
5:  $(isk_i, ipk_i) \leftarrow_{\$} \text{IKGen}(pp)$  for  $i = 1, \dots, m$ 
6:  $Q_{\text{ipk}} \leftarrow \{(isk_i, ipk_i)\}_{i=1}^m$ 
7:  $(I^*, \text{state}) \leftarrow_{\$} \mathcal{A}^{\mathcal{O}_{\text{Issue}}, \mathcal{O}_{\text{Present}}, \mathcal{O}_{\text{Reveal}}}(pp, \{ipk_i\}_{i=1}^m, \{(opk_i, osk_i)\}_{i=1}^n)$ 
8:  $\text{pol}^* \leftarrow_{\$} \text{PresPolicy}(I^*)$ 
9:  $(\text{pt}^*, \phi^*, t^*, \text{ctx}^*, \text{opk}^*) \leftarrow_{\$} \mathcal{A}^{\mathcal{O}_{\text{Issue}}, \mathcal{O}_{\text{Present}}, \mathcal{O}_{\text{Reveal}}}(\text{state}, \text{pol}^*)$ 
10: if :
11:   (a)  $I^* \subseteq \{ipk_i\}_{i=1}^m$  and  $\text{opk}^* \in \{opk_i\}_{i=1}^n$ ,
12:   (b)  $\text{Verify}(\text{pt}^*, \phi^*, \text{pol}^*, t^*, \text{opk}^*, \text{ctx}^*) = 1$ ,
13:   (c)  $\nexists(\cdot, \cdot, \text{pol}^*, \phi^*, t^*, \text{opk}^*, \text{ctx}^*) \in Q_{\text{Present}}$ , and
14:   (d)  $\forall(\text{cred}, ipk_i, \vec{a}, j) \in Q_{\text{Issue}}, \nexists j \in Q_{\text{Reveal}}$  such that  $\phi^*(\vec{a}) = 1$  and  $ipk_i \in I^*$ 
15:   return 1
16: return 0

```

Issuer non-frameability. Issuer non-frameability requires that an adversary cannot produce a fresh valid presentation that opens to an honest issuer without having obtained a suitable credential from that issuer, even if all other issuer keys are adversarially chosen and the adversary eventually gains full control over the opener. The adversary may access the issuance, presentation, and reveal oracles, but wins only if the resulting token verifies, opens to the honest issuer, was not obtained from the presentation oracle for the same $(\text{pol}, \phi, t, \text{ctx})$ tuple, and is not based on a revealed credential satisfying the challenge predicate. In practice, ctx is chosen freshly for each presentation and thereby provides session binding, so this condition captures freshness while excluding trivial replay of an oracle-generated presentation.

Definition 7 (Issuer non-frameability). *An issuer-hiding credential system with opening satisfies issuer non-frameability if, for every PPT adversary $\mathcal{A}$, there exists a negligible function ε s.t., $\Pr[\text{Exp}_{\mathcal{A}}^{\text{Issuer-Non-Frameability}}(\lambda) = 1] \leq \varepsilon(\lambda)$, where the experiment is defined in Experiment 3.*

Experiment 3 $\text{Exp}_{\mathcal{A}}^{\text{Issuer-Non-Frameability}}(\lambda)$

```

1:  $pp \leftarrow_{\$} \text{Setup}(1^\lambda)$ 
2:  $j \leftarrow 0$ 
3:  $Q_{\text{Issue}} \leftarrow \emptyset, Q_{\text{Present}} \leftarrow \emptyset, Q_{\text{Reveal}} \leftarrow \emptyset$ 
4:  $(isk, ipk) \leftarrow_{\$} \text{IKGen}(pp)$ 
5:  $Q_{\text{ipk}} \leftarrow \{(isk, ipk)\}$ 
6:  $(osk, opk) \leftarrow_{\$} \text{OKGen}(pp)$ 
7:  $(\text{pt}^*, \{ipk_i^*\}, osk_{ipk, t^*}^*, t^*, \text{pol}^*, \phi^*, \text{ctx}^*) \leftarrow_{\$} \mathcal{A}^{\mathcal{O}_{\text{Issue}}, \mathcal{O}_{\text{Present}}, \mathcal{O}_{\text{Reveal}}}(ipk, opk, osk)$ 
8: if :
9:   (a)  $\text{Open}(\text{pt}^*, opk, osk_{ipk, t^*}^*, ipk, t^*, \text{pol}^*, \phi^*) = 1$ ,
10:   (b)  $\text{Verify}(\text{pt}^*, \phi^*, \text{pol}^*, t^*, opk, \text{ctx}^*) = 1$ ,
11:   (c)  $\text{VfPolicy}(\text{pol}^*, \{ipk_i^*\}) = 1$ ,
12:   (d)  $ipk \in \{ipk_i^*\}$ ,
13:   (e)  $\nexists(\cdot, \cdot, \text{pol}^*, \phi^*, t^*, opk, \text{ctx}^*) \in Q_{\text{Present}}$ ,
14:   (f)  $\nexists j \in Q_{\text{Reveal}}$  such that  $(\text{cred}, ipk, \vec{a}, j) \in Q_{\text{Issue}} \wedge \phi^*(\vec{a}) = 1$ 
15:   return 1
16: return 0

```

4 A Generic Construction

In this section, we provide a generic black-box construction of our scheme that satisfies the security definitions presented in Section 3, together with a formal security analysis.

4.1 Our Construction

In Construction 1, we now present a generic black-box construction from standard building blocks, thereby extending the construction of Bobolz et al. [8]. The construction combines two digital-signature schemes Σ_V and Σ_I , a non-interactive zero-knowledge proof system Π , and a predicate-encryption scheme Λ . The signature scheme Σ_I is used by issuers to certify user attributes, while Σ_V is used to authenticate the issuer public keys contained in a presentation policy. Predicate encryption is used to achieve our tracing capabilities, and the NIZK proof ties all components of a presentation together without revealing sensitive information. For the predicate-encryption scheme, we assume that the family $\mathcal{F}$ of predicates is parametrized by pairs of issuer public key and epochs, and that each function checks equality with a given pair; formally, $\mathcal{F}$ is defined as follows:

$$f_{ipk,t} : \{0,1\}^{\lambda_1} \times \{0,1\}^{\lambda_2} \rightarrow \{0,1\} : (ipk', t') \mapsto [(ipk, t) = (ipk', t')].$$

On a high level, every presentation pt is bound to a pair consisting of the hidden issuer ipk and the current epoch t . To do so, the user encrypts a fixed message $1_{\mathcal{M}}$ under the predicate-encryption attribute (ipk, t) . The presentation then proves in ZK that the user possesses a valid credential under ipk certifying attributes satisfying the predicate ϕ , that ipk is accepted by the presentation policy, and that the ciphertext was correctly formed for the same issuer-epoch pair.

While the ciphertext does not reveal the issuer during normal verification, this can be used for later opening. For a target issuer ipk and epoch t , the opener derives a predicate key for the equality predicate $f_{ipk,t}$, which accepts exactly the attribute (ipk, t) . Such a key therefore allows a verifier to test whether a stored presentation pt was created for this particular issuer and epoch, while ciphertexts associated with other issuer-epoch pairs remain unaffected. Since opening keys may be publicly released, the **Open** algorithm has to check that a given opening key is consistent with the claimed issuer-epoch pair with an encryption-decryption test: first encrypting a fixed value $1_{\mathcal{M}}$ under the claimed issuer-epoch pair (ipk, t) , and then decrypting it with the claimed derived key $osk_{ipk',t'}$ and checking that the value is indeed $1_{\mathcal{M}}$. Technically, this is the point where the opening-soundness property introduced in Definition 3 is required to ensure non-frameability.

4.2 Security Analysis

Theorem 1. *Let Λ be an attribute-hiding predicate encryption and Π is zero-knowledge proof system. Then, Construction 1 achieves unlinkability.*

Proof (Proof sketch). The full proof is given in Section A.1. The proof proceeds by a sequence of hybrids. First, we replace the NIZK proof in the challenge presentation by a simulated proof; this is indistinguishable by the zero-knowledge property of Π . Next, we replace the challenge ciphertext, which encrypts under the attribute (ipk_{ib}, t) , by an encryption under a fixed challenge attribute. The attribute-hiding security of Λ guarantees that this change is indistinguishable, since the adversary is not allowed to obtain opening keys for either challenge issuer at the challenge epoch. Thus, both the ciphertext and the simulated proof are independent of the challenge bit b , and hence the adversary can guess b with probability at most $1/2$. $\square$

Theorem 2. *Let Σ_I and Σ_V be EUF-CMA-secure signature schemes, and let Π be zero-knowledge and a simulation-sound extractable non-interactive proof system. Then Construction 1 achieves unforgeability.*

Proof (Proof sketch). The full proof is given in Section A.2. The proof is similar to the one in [8]. A valid presentation simultaneously proves that the underlying issuer is authorized by the policy and

Construction 1 Generic construction of issuer-hiding ABCs with opening

Setup(1^λ): Run $pp_I \leftarrow_{\$} \Sigma_I.\text{Setup}(1^\lambda)$, $pp_V \leftarrow_{\$} \Sigma_V.\text{Setup}(1^\lambda)$, $pp_\Lambda \leftarrow_{\$} \Lambda.\text{Setup}(1^\lambda)$, $pp_\Pi \leftarrow_{\$} \Pi.\text{Setup}(1^\lambda)$.
return $pp := (pp_I, pp_V, pp_\Lambda, pp_\Pi)$.

IKGen(pp): **return** $(isk, ipk) \leftarrow_{\$} \Sigma_I.\text{KeyGen}(pp_I)$

OKGen(pp): **return** $(osk, opk) \leftarrow_{\$} \Lambda.\text{KeyGen}(pp_\Lambda)$

OKDerive(osk, ipk, t): **return** $osk_{ipk,t} \leftarrow_{\$} \Lambda.\text{Derive}(osk, f_{ipk,t})$

Issue($isk, \vec{a}$): **return** $\text{cred} \leftarrow_{\$} \Sigma_I.\text{Sign}(isk, \vec{a})$.

VfCred($\text{cred}, \vec{a}, ipk$): **return** $b \leftarrow \Sigma_I.\text{Verify}(ipk, \text{cred}, \vec{a})$.

PresPolicy($\{ipk_i\}$): Run $(vsk, vpk) \leftarrow \Sigma_V.\text{KeyGen}(pp_V)$ and $\sigma_i \leftarrow_{\$} \Sigma_V.\text{Sign}(vsk, ipk_i)$ for all i .
return $\text{pol} = (vpk, \{(ipk_i, \sigma_i)\})$.

VfPolicy($\text{pol}, \{ipk'_j\}$): Parse pol as $(vpk, \{(ipk_i, \sigma_i)\})$.
 1: **if** $ipk'_j \in \{ipk_i\} \forall j$ and $\Sigma_V.\text{Verify}(vpk, \sigma_i, ipk_i) = 1 \forall i$ **then return** 1.
 2: **else: return** 0.

Present($opk, ipk, \text{cred}, \phi, \vec{a}, t, \text{pol}, \text{ctx}$):
 1: $r \leftarrow_{\$} \mathcal{R}$, $\text{ct} \leftarrow \Lambda.\text{Enc}(opk, (ipk, t), 1_{\mathcal{M}}; r)$
 2: Parse pol as $(vpk, \{(ipk_i, \sigma_i)\})$. Let σ be the element such that $(ipk, \sigma) \in \{(ipk_i, \sigma_i)\}$.
 3: Compute a proof of knowledge π as follows:

$$\pi \leftarrow_{\$} \Pi.\text{Prove}[(\sigma, ipk, \text{cred}, \vec{a}, r) : \Sigma_I.\text{Verify}(ipk, \text{cred}, \vec{a}) = 1 \wedge \phi(\vec{a}) = 1 \wedge \Sigma_V.\text{Verify}(vpk, \sigma, ipk) = 1 \wedge \text{ct} = \Lambda.\text{Enc}(opk, (ipk, t), 1_{\mathcal{M}}; r)](\text{ctx}'),$$
 where $\text{ctx}' = (\text{ctx}, \text{pol}, \phi, t, opk, \text{ct})$ contains all public elements
 4: **return** presentation token $\text{pt} := (\text{ct}, \pi)$

Verify($\text{pt}, \phi, \text{pol}, t, opk, \text{ctx}$):
 1: Parse pt as (ct, π)
 2: **return** $b \leftarrow \Pi.\text{Verify}((\text{pol}, \phi, t, opk), \text{ctx}', \pi)$ for ctx' as in **Present**

Open($\text{pt}, opk, osk_{ipk',t'}, ipk, t, \text{pol}, \phi$) :
 1: Parse pt as (ct, π)
 2: $\text{ct}^* \leftarrow_{\$} \Lambda.\text{Enc}(opk, (ipk, t), 1_{\mathcal{M}})$
 3: **if** $\Lambda.\text{Dec}(osk_{ipk',t'}, \text{ct}^*) \neq 1_{\mathcal{M}}$ **then return** error message $\perp$
 4: **else if** $\Lambda.\text{Dec}(osk_{ipk',t'}, \text{ct}) = 1_{\mathcal{M}}$ **then return** 1
 5: **else: return** 0

that the underlying credential is valid. Extractability lets us recover these hidden signatures from a forged presentation. Therefore, a fresh successful forgery must either introduce an unauthorized issuer, which breaks Σ_V , or contain a new valid credential for an honest issuer, which breaks Σ_I . $\square$

Theorem 3. *Let Λ be a predicate-encryption scheme satisfying correctness and opening soundness, let Π be a zero-knowledge and simulation-sound extractable non-interactive proof system, and let Σ_I be an EUF-CMA-secure signature scheme. Then Construction 1 achieves issuer non-frameability.*

Proof (Proof sketch). The full proof is given in Section A.3. By simulation-sound extractability of the NIZK, we can extract from any fresh valid forgery the underlying issuer, credential, attributes, and encryption randomness. Opening soundness then guarantees that a presentation opening to the targeted honest issuer must indeed correspond to that issuer. Hence, the extracted credential must verify under the honest issuer's public key. $\mathcal{A}$ can therefore succeed only by knowing the issuer's secret key, forging a valid credential under Σ_I , or using a credential previously obtained

through the reveal oracle. The first case is excluded because the issuer is honest, the second contradicts the EUF-CMA security of Σ_I , and the third is ruled out by the winning conditions of the experiment. $\square$

5 Instantiation

In the following we now provide a concrete instantiation. Section 5.1 recaps the concrete building blocks we are using, while the resulting construction is given in Section 5.2. Finally, we provide an efficiency analysis in Section 5.3.

5.1 Building blocks

Groth signatures. In the following we briefly recap Groth's signature scheme [27] which is secure in the generic group model. While the scheme is able to simultaneously sign both $\mathbb{G}_1$ and $\mathbb{G}_2$ elements, we only present a variant for $\mathbb{G}_1$ elements, in line with the needs of our construction.

- **Groth₁.Setup**($1^\lambda, m$): Generate $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$ with $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. Sample $y \leftarrow_{\$} \mathbb{Z}_q$, set $Y \leftarrow g_1^y$, and return $pp = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2, m, Y)$.
- **Groth₁.KeyGen**(pp): Sample $\vec{u} = (u_1, \dots, u_{m-1}) \leftarrow_{\$} \mathbb{Z}_q^{m-1}$ and $v \leftarrow_{\$} \mathbb{Z}_q$. Set $U_i \leftarrow g_2^{u_i}$ for $i \in [m-1]$ and $V \leftarrow g_2^v$. Return $\text{sk} = (\vec{u}, v)$ and $\text{pk} = (\vec{U}, V)$.
- **Groth₁.Sign**(sk, msg): Parse $\text{msg} = (M_1, \dots, M_m) \in \mathbb{G}_1^m$. Sample $z \leftarrow_{\$} \mathbb{Z}_q^*$ and set $r = z^{-1}$. Compute $R \leftarrow g_2^r, S \leftarrow (Y g_1^v)^z, T \leftarrow \left(\prod_{i=1}^{m-1} M_i^{u_i} M_m Y^v \right)^z$ and return $\sigma = (R, S, T)$.
- **Groth₁.Verify**($\text{pk}, \text{msg}, \sigma$): Parse $\text{pk} = (\vec{U}, V)$ and $\sigma = (R, S, T)$. Return 1 iff $e(S, R) \stackrel{?}{=} e(Y, g_2)e(g_1, V)$ and $e(T, R) \stackrel{?}{=} \left(\prod_{i=1}^{m-1} e(M_i, U_i) \right) e(M_m, g_2)e(Y, V)$.
- **Groth₁.Rand**(σ): Sample $\beta \leftarrow_{\$} \mathbb{Z}_q^*$ and return $\sigma' = (R^{1/\beta}, S^\beta, T^\beta)$. This re-randomization yields unlinkable signatures on the same message.

We use Groth_2 to denote the analogous Groth signature for messages in $\mathbb{G}_2$.

Gentry Identity-Based Encryption. We further rely on a Type-3 instantiation [46] of Gentry's anonymous identity-based encryption (IBE) [25]. The scheme achieves ANON-IND-ID-CPA security, meaning that a ciphertext hides both the encrypted message and the identity under which it was created, even if the adversary obtains secret keys for arbitrary identities other than the challenge identities. This property is particularly useful in our setting, where the identity encodes the issuer-epoch pair and must remain hidden during presentation. At the same time, identity-specific secret keys provide exactly the selective decryption functionality required for opening. As detailed later, we exploit this structure to emulate the equality predicate needed by our generic predicate-encryption-based construction. For completeness, we briefly recall the algorithms of the scheme:

- **Gentry.Setup**(1^λ): Generate $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$. Let $H : \{0, 1\}^* \rightarrow \mathbb{Z}_q$ be hash function. Return $pp = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2, H)$.
- **Gentry.KeyGen**(pp): Sample $\alpha \leftarrow_{\$} \mathbb{Z}_q^*$, $h_1, h_2, h_3 \leftarrow_{\$} \mathbb{G}_2$, and set $g'_1 = g_1^\alpha$. Output $\text{mpk} = (g'_1, h_1, h_2, h_3)$ and $\text{msk} = \alpha$.
- **Gentry.Derive**($\text{mpk}, \text{msk}, \text{id}$): Given an identity $\text{id} \in \mathbb{Z}_q$, return $\perp$ if $\text{id} = \alpha$. Otherwise, for each $i \in \{1, 2, 3\}$, sample $r_{\text{id}, i} \leftarrow_{\$} \mathbb{Z}_q$ and compute $h_{\text{id}, i} = \left(h_i g_2^{-r_{\text{id}, i}} \right)^{1/(\alpha - \text{id})}$. Return $\text{sk}_{\text{id}} = \{(r_{\text{id}, i}, h_{\text{id}, i})\}_{i=1}^3$.
- **Gentry.Enc**($pp, \text{mpk}, \text{id}, \text{msg}$): Choose $s \leftarrow_{\$} \mathbb{Z}_q^*$ and compute $C_1 = g'_1{}^s g_1^{-s \text{id}}$, $C_2 = e(g_1, g_2)^s$, $C_3 = \text{msg } e(g_1, h_1)^{-s}$, $\xi = H(C_1, C_2, C_3)$, and $C_4 = e(g_1, h_2)^s e(g_1, h_3)^{s \xi}$. Return $C = (C_1, C_2, C_3, C_4)$.

Construction 2 Concrete instantiation: System initialization

Setup(1^λ) :

- 1: Generate $pp = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, y_1, y_2, g_1, g_2)$ with prime order q , where $g_1, y_1 \in \mathbb{G}_1$ and $g_2, y_2 \in \mathbb{G}_2$.
- 2: Define a hash and sign function $H : \mathbb{Z}_q^L \rightarrow \mathbb{G}_1$ such that $H(\vec{a}) = \prod_{i=1}^L H_i^{a_i}$, where $H_i \leftarrow_{\$} \mathbb{G}_1$ for all i .
- 3: Select a cryptographic hash function $H' : \{0, 1\}^* \mapsto \{0, 1\}^{\lambda_1}$
- 4: Select a cryptographic hash function $H'' : \{0, 1\}^* \rightarrow \mathbb{Z}_q$
- 5: Define $f : \{0, 1\}^{\lambda_1} \times \{0, 1\}^{\lambda_2} \rightarrow \mathbb{Z}_q : (a, b) \mapsto 2^{\lambda_2}a + b$, where $2^{\lambda_1 + \lambda_2} < q$.

IKGen(pp) :

- 1: **return** $(isk_{\text{Groth}_1}, ipk_{\text{Groth}_1}) = (v, V_2) \leftarrow_{\$} \text{Groth}_1.\text{KeyGen}(pp)$

OKGen(pp) :

- 1: **return** $(osk_{\text{Gentry}}, opk_{\text{Gentry}}) = (w, (g'_1, h_1, h_2, h_3)) \leftarrow \text{Gentry}.\text{KeyGen}(pp)$
-

- $\text{Gentry}.\text{Dec}(pp, sk_{id}, C)$: Parse $C = (C_1, C_2, C_3, C_4)$ and $sk_{id} = \{(r_{id,i}, h_{id,i})\}_{i=1}^3$. Compute $\xi = H(C_1, C_2, C_3)$. If $C_4 \neq e(C_1, h_{id,2} h_{id,3}^\xi) C_2^{r_{id,2} + \xi r_{id,3}}$ or $C_2 = 1_{\mathbb{G}_T}$ return $\perp$. Otherwise, return $\text{msg} = C_3 e(C_1, h_{id,1}) C_2^{r_{id,1}}$.

As required by our application, we show that the Type-3 instantiation of Gentry's IBE scheme satisfies the opening-soundness property of Definition 3.

Theorem 4 (Opening-Soundness of the Gentry Encryption). *Assume that the discrete-logarithm problem is hard in $\mathbb{G}_2$. Then the Gentry-based equality-predicate encryption satisfies the opening soundness according to Definition 3 with respect to the message $1_{\mathbb{G}_T}$.*

We provide proof in Section A.4.

5.2 Concrete Construction

We instantiate the generic construction from Construction 1 using the building blocks introduced above. The credential and policy signatures are instantiated using Groth's signatures, while the equality predicate required for selective opening is emulated using Gentry's anonymous IBE scheme.

To bind opening to both an issuer and an epoch, each issuer is assigned the identifier $ipk_{id} = H'(ipk)$, and the pair (ipk_{id}, t) is mapped to the IBE identity $id = f(ipk_{id}, t) = 2^{\lambda_2} ipk_{id} + t$, where 2^{λ_2} bounds the number of supported epochs. For example, $\lambda_2 = 16$ supports approximately 180 years with one epoch per day. This encoding allows Gentry's IBE to realize the equality predicate required by the generic construction. Except with negligible probability of a collision in H' , distinct issuer-epoch pairs map to distinct IBE identities. By Theorem 4, the test $\text{Gentry}.\text{Dec}(osk_{id}, c) = 1_{\mathbb{G}_T}$ therefore realizes equality on the corresponding issuer-epoch pairs.

Parameter generation. Construction 2 defines the bilinear-group parameters, the algebraic hash $H : \mathbb{Z}_q^L \rightarrow \mathbb{G}_1$, and the mapping f from issuer-epoch pairs to IBE identities. The opener runs Gentry's key generation to obtain the master key used to derive issuer- and epoch-specific opening keys.

Issuance and policy generation. Construction 3 now instantiates credential issuance and presentation policies. A credential is a Groth signature on the encoded attribute vector $H(\vec{a})$. With independently sampled bases in the public parameters, this encoding is computationally binding under the discrete-logarithm assumption and allows the entire attribute vector to be represented by a single group element for signing. A presentation policy contains Groth signatures authenticating the public keys of all accepted issuers. Specifically, each policy signature authenticates both ipk (written as ipk_{Groth_1} in the instantiation) and its hash $H'(ipk)$, thus binding the issuer public

Construction 3 Concrete instantiation: Issuance and presentation policies

```

Issue( $isk_{\text{Groth}_1}, \vec{a}$ ):
  1: return  $\text{cred} = (R_2, S_1, T_1) \leftarrow_{\$} \text{Groth}_1.\text{Sign}(isk_{\text{Groth}_1}, H(\vec{a}))$ 
VfCred( $\text{cred}, \vec{a}, ipk_{\text{Groth}_1}$ ):
  1: return  $\text{Groth}_1.\text{Verify}(\text{cred}, H(\vec{a}), ipk_{\text{Groth}_1})$ 
PresPolicy( $\{ipk_{\text{Groth}_1, i}\}$ ):
  1:  $(vsk_{\text{Groth}_2}, vpk_{\text{Groth}_2}) = ((u', v'), (U'_1, V'_1)) \leftarrow \text{Groth}_2.\text{KeyGen}(pp)$ 
  2:  $\sigma_i = (R_{1,i}, S_{2,i}, T_{2,i}) \leftarrow_{\$} \text{Groth}_2.\text{Sign}(vsk_{\text{Groth}_2}, (g_2^{H'(ipk_{\text{Groth}_1, i)})}, ipk_{\text{Groth}_1, i}))$ 
  3: return  $\text{pol} = (vpk_{\text{Groth}_2}, \{(\sigma_i, ipk_{\text{Groth}_1, i})\})$ 
VfPolicy( $\text{pol}, \{ipk_{\text{Groth}_1, i}\}$ ):
  1: Parse  $\text{pol}$  as  $(vpk_{\text{Groth}_2}, \{(\sigma_i, ipk_{\text{Groth}_1, i})\})$ .
  2: if  $\text{Groth}_2.\text{Verify}(\sigma_i, (g_2^{H'(ipk_{\text{Groth}_1, i)})}, ipk_{\text{Groth}_1, i}), vpk_{\text{Groth}_2}) = 1$  for all  $i$ : return 1
  3: else return 0

```

key used for credential verification to the value used for opening, which live in different algebraic domains ($\mathbb{G}_2$ and $\mathbb{Z}_q$, respectively).

Policy generation and presentation. During presentation, depicted in Construction 4, the user randomizes the credential and the corresponding policy signature and computes the IBE-identity $\text{id} = 2^{\lambda_2} H'(ipk) + t$. The user then encrypts the neutral element $1_{\mathbb{G}_T}$ under the Gentry IBE identity id . The accompanying proof shows that the credential and policy signature refer to the same hidden issuer, that id is correctly derived from this issuer and the current epoch, that the ciphertext is a correct encryption of $1_{\mathbb{G}_T}$ under id , and that the certified attributes satisfy the required predicate. Specifically, we realize the NIZK using Schnorr-style proofs [49] made non-interactive via Fiat–Shamir [22, 5]. For admissible linear relations, such proofs provide strong simulation extractability with straight-line extraction in the algebraic-group and random-oracle models [47], where the multiplication proof arising in our relation can be linearized using standard techniques. Alternatively, simulation-sound extractability can be obtained via encryption-to-the-sky [44, 12] or related paradigms [23]. More complex predicates ϕ can be realized using standard techniques [19, 11, 37, 40, 10]. Note that the range of ipk_{id} need not be proven separately, since the presentation-policy signature binds it to $H'(ipk) \in \{0, 1\}^{\lambda_1}$. Furthermore, as the epoch t is public, the verifier directly checks that $t \in [0, 2^{\lambda_2})$.

Opening and issuer tracing. Construction 5 finally realizes the selective opening functionalities. Given a target issuer ipk and epoch t , **OKDerive** computes the corresponding identity $\text{id} = f(H'(ipk), t)$ and derives the associated Gentry secret key osk_{id} . Before applying this key to a presentation token, **Open** first verifies that osk_{id} indeed corresponds to the claimed issuer-epoch pair by generating a fresh encryption c^* of $1_{\mathbb{G}_T}$ under id and checking that it decrypts correctly. This prevents an incorrectly labeled opening key, for example a key for another issuer, from causing presentations to be falsely attributed to ipk . The algorithm then decrypts the ciphertext contained in the presentation token and returns 1 exactly if it decrypts to $1_{\mathbb{G}_T}$. By the equality property discussed above, this happens exactly when the presentation was created for the targeted issuer-epoch pair, except with negligible probability. Hence, a released opening key allows verifiers to locally identify matching presentations without affecting the privacy of presentations from other issuers or epochs.

Putting all these observations together, we now obtain the following:

Corollary 1. *Assume that the Groth signature schemes are EUF-CMA secure, that the Gentry IBE scheme is ANON-IND-ID-CPA secure, that H' is collision resistant, and that the discrete logarithm problem is hard in $\mathbb{G}_1$ and $\mathbb{G}_2$. Then, Constructions 2 to 5 satisfy unlinkability, unforgeability, and issuer non-frameability according to Definitions 5 to 7, in the generic group model and random oracle model.*

Construction 4 Concrete instantiation: Presentation and verification

Present($opk_{\text{Gentry}}, ipk_{\text{Groth}_1}, cred, \phi, \vec{a}, t, pol, ctx$):

- 1: Let $ipk_{id} \leftarrow H'(ipk_{\text{Groth}_1})$
- 2: $id \leftarrow f(ipk_{id}, t)$
- 3: $c = (C_1, C_2, C_3, C_4) \leftarrow \text{Gentry.Enc}(\text{msg} = 1_{\mathbb{G}_T}, opk_{\text{Gentry}}, id; r_{\text{Enc}})$, $r_{\text{Enc}} \leftarrow \mathbb{Z}_q^*$
- 4: Parse pol as $(vpk, \{(\sigma_i, ipk_i)\})$. Let k be the index such that $\{(\sigma_k, ipk_{\text{Groth}_1})\} \in pol$. Parse vpk as (U'_1, V'_1) .
- 5: $cred' \leftarrow \text{Groth}_1.\text{Rand}(cred) = (\hat{R}_2, \hat{S}_1, \hat{T}_1)$ // Randomization
- 6: $\sigma' \leftarrow \text{Groth}_2.\text{Rand}(\sigma) = (\hat{R}_{1,k}, \hat{S}_{2,k}, \hat{T}_{2,k})$
- 7: Pick random values $\alpha, \beta, \gamma, \delta_1, \delta_2 \leftarrow \mathbb{Z}_q^*$ // Blinding
- 8: $(\bar{R}_2, \bar{S}_1, \bar{T}_1) \leftarrow (\hat{R}_2, \hat{S}_1^{1/\alpha}, \hat{T}_1^{1/\beta})$
- 9: $(\bar{R}_{1,k}, \bar{S}_{2,k}, \bar{T}_{2,k}) \leftarrow (\hat{R}_{1,k}, \hat{S}_{2,k}^{1/\gamma}, \hat{T}_{2,k}^{1/\gamma})$
- 10: $\bar{ipk} = (\bar{ipk}_{\text{Groth}_1}, \bar{ipk}_{id}) = (ipk_{\text{Groth}_1}^{1/\delta_1}, g_2^{ipk_{id}/\delta_2})$
- 11: $\pi \leftarrow (II.\text{Prove}[(\alpha, \beta, \gamma, \delta_1, \delta_2, ipk_{id}, \vec{a}, id, r_{\text{Enc}}) :$

$$(1) \quad e(\bar{S}_1, \bar{R}_2)^\alpha = e(y_1, g_2) \cdot e(g_1, \bar{ipk}_{\text{Groth}_1})^{\delta_1} \wedge$$

$$e(\bar{T}_1, \bar{R}_2)^\beta = e(y_1, \bar{ipk}_{\text{Groth}_1})^{\delta_1} \cdot e(H(\vec{a}), g_2) \wedge$$

$$(2) \quad e(\bar{R}_{1,k}, \bar{S}_{2,k}) = e(g_1, y_2) \cdot e(V'_1, g_2) \wedge$$

$$e(\bar{R}_{1,k}, \bar{T}_{2,k})^\gamma = e(V'_1, y_2) \cdot e(g_1, \bar{ipk}_{\text{Groth}_1})^{\delta_1} \cdot e(U'_1, \bar{ipk}_{id})^{\delta_2} \wedge$$

$$(3) \quad id = f(ipk_{id}, t) \wedge$$

$$C_1 = (g_1 g_1^{-id})^{r_{\text{Enc}}} \wedge C_2 = e(g_1, g_2)^{r_{\text{Enc}}} \wedge C_3 = \text{msg} \cdot e(g_1, h_1)^{-r_{\text{Enc}}} \wedge$$

$$C_4 = e(g_1, h_2)^{r_{\text{Enc}}} e(g_1, h_3)^{\xi r_{\text{Enc}}}$$

$$(4) \quad \phi(\vec{a} = 1)(ctx')$$

where ctx' is as in Construction 1 and is included in the Fiat-Shamir challenge.

//(1):Credential Check (2):Policy Check (3):IBE&Check (4):Attributes Check

- 12: **return** $pt = (c, (\bar{R}_2, \bar{S}_1, \bar{T}_1, \bar{R}_{1,k}, \bar{S}_{2,k}, \bar{T}_{2,k}, \bar{ipk}, \pi))$

Verify($pt, \phi, pol, t, opk, ctx$) :

- 1: Parse pt as $((C_1, C_2, C_3, C_4), (\bar{R}_2, \bar{S}_1, \bar{T}_1, \bar{R}_{1,k}, \bar{S}_{2,k}, \bar{T}_{2,k}, \bar{ipk}, \pi))$
- 2: **return** 1 if the proof verification on π succeeds, $C_2 \neq 1_{\mathbb{G}_T}$, and $t \in [0, 2^{\lambda_2})$, and 0 otherwise.

Construction 5 Concrete instantiation: Issuer re-identification

OKDerive($osk_{\text{Gentry}}, t, ipk_{\text{Groth}_1}$):

- 1: $id = f(H'(ipk_{\text{Groth}_1}), t) \in \mathbb{Z}_q$
- 2: $osk_{id} \leftarrow \text{Gentry.Derive}(opk_{\text{Gentry}}, osk_{\text{Gentry}}, id)$
- 3: **return** osk_{id}

Open($pt, opk_{\text{Gentry}}, osk_{id}, ipk_{\text{Groth}_1}, t$) :

- 1: Parse pt as $(c, (\bar{R}_2, \bar{S}_1, \bar{T}_1, \bar{R}_{1,k}, \bar{S}_{2,k}, \bar{T}_{2,k}, \bar{ipk}, \pi))$
- 2: $id = f(H'(ipk_{\text{Groth}_1}), t) \in \mathbb{Z}_q$
- 3: $c^* \leftarrow \text{Gentry.Enc}(1_{\mathbb{G}_T}, opk_{\text{Gentry}}, id)$
- 4: **Abort** if $\text{Gentry.Dec}(osk_{id}, c^*) \neq 1_{\mathbb{G}_T}$.
- 5: **if** $\text{Gentry.Dec}(osk_{id}, c) = 1_{\mathbb{G}_T}$ **return** 1
- 6: **else return** 0

Proof. The corollary is a direct consequence of Theorems 1 to 3. Theorem 4 establishes the required opening soundness of the Gentry-based equality-predicate encryption, while the algebraic attribute encoding H is computationally binding under the discrete-logarithm assumption in $\mathbb{G}_1$. Finally, the required zero-knowledge property follows from the standard simulation of Schnorr/Fiat-Shamir proofs in the random-oracle model, while simulation-sound extractability follows from the strong simulation extractability of Fiat-Shamir for admissible linear relations in the algebraic-group and random-oracle models [47], using the standard linearization described above. Since generic-group

adversaries (as required for Groth signatures) form a subclass of algebraic adversaries, this in particular yields the required property in the generic-group model. $\square$

5.3 Evaluation

Table 1 reports the computational cost and estimated runtime of the main algorithms. Runtimes are derived from the number of operations in $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ and pairings, using average timings measured with `arkworks` 0.5¹ over BLS12-381 (offering a security level of almost 128 bits) in release mode on an Intel(R) Core(TM) i7-1370P CPU. Exponentiation in $\mathbb{G}_1$ and $\mathbb{G}_2$, and $\mathbb{G}_T$, and pairing take on average 0.0693 ms, 0.3518 ms, 0.4769 ms, and 0.7707 ms, respectively.

Computational Cost. We consider a setting with universities as issuers and students as users. Credentials contain 10 attributes, none disclosed during presentation (the most expensive case), and the policy covers 3000 issuers, roughly the number of higher education institutions in the European Union. We set $\lambda_1 = 192$ and $\lambda_2 = 16$ as required by our instantiation. Policy generation and verification scale linearly with the number of issuers and take approximately 4430 ms and 9250 ms, respectively, but are performed only once. The presentation takes about 126.45 ms and verification 54.23 ms while they are independent of the number of issuers. Finally, even for large parameter sizes, the runtimes remain reasonable, indicating the scalability of our scheme.

Table 1: Number of group exponentiations and pairings.

	Runtime	$\mathbb{G}_1$	$\mathbb{G}_2$	$\mathbb{G}_T$	Pairing
PresPolicy (3000 issuers)	4430 ms	3002	12000	-	-
VfPolicy (3000 issuers)	9250 ms	-	-	-	12000
Issue (10 attributes)	1.47 ms	11	2	-	-
VfCred	2.31 ms	-	-	-	3
Present (10 attributes)	126.45 ms	1586	6	19	7
Verify	54.23 ms	556	-	20	8
Open	6.34 ms	4	2	8	2

Communication Cost. Table 2 reports the storage cost. Keys and credentials have constant size, while the presentation policy grows linearly with the number I of issuers. The presentation token contains a constant number of group elements and grows with the number L of attributes only through its $\mathbb{Z}_q$ elements. For BLS12-381, we use compressed representations for the source groups. Elements of $\mathbb{G}_1$ and $\mathbb{G}_2$ require 48 and 96 bytes, respectively, while scalars in $\mathbb{Z}_q$ require 32 bytes. Elements of the target group $\mathbb{G}_T$ are represented in the full extension field and require 576 bytes in our implementation.

6 Conclusion, Future Work and Extensions

We introduced issuer-hiding attribute-based credentials with fine-grained opening, enabling issuer- and epoch-specific retrospective tracing while preserving the privacy of unrelated presentations. Our scheme is modular, and the issuer-inspection component is largely orthogonal to other components and standard extensions of credential systems. Features such as credential revocation or device binding can be incorporated using existing techniques. Moreover, the selective-opening mechanism may also be useful beyond issuer tracing. In particular, we believe that the same approach could be adapted to traditional user-tracing mechanisms, providing similarly fine-grained and locally verifiable tracing while limiting the information revealed by the inspection authority.

Future work includes adapting the approach to standardized or widely deployed credential schemes, such as BBS-based credentials [35,32], whose treatment of verifier policies and issuer authorization

¹ <https://github.com/arkworks-rs>

Table 2: Number of group elements and storage size of our algorithms.

	$\mathbb{G}_1$	$\mathbb{G}_2$	$\mathbb{G}_T$	$\mathbb{Z}_q$	Size (bytes)
Issuer secret key (<i>isk</i>)	-	-	-	1	32
Issuer public key (<i>ipk</i>)	-	1	-	-	96
Opener secret key (<i>osk</i>)	-	-	-	1	32
Opener public key (<i>opk</i>)	1	3	-	-	336
Credential (<i>cred</i>)	2	1	-	-	192
Presentation policy (<i>pol</i>)	$I + 2$	$3I$	-	-	$336I + 96$
Presentation token (<i>pt</i>)	4	5	3	$L + 10$	$2720 + 32L$

differs from our setting. Adapting our issuer-hiding and opening mechanism to this setting may therefore require additional techniques. Another direction is the design of efficient post-quantum issuer-hiding credentials with comparable fine-grained opening functionality.

Acknowledgments. This work was in part supported by the European Union’s Horizon Europe projects LICORICE and POSEIDON (grant agreement no. 101168311 and no. 101225797) and the CHIST-ERA project REMINDER through the Austrian Science Fund (FWF) project number I 6650-N. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the funding agencies. Neither European Union nor the granting authorities can be held responsible for them.

References

1. Acar, T., Nguyen, L.: Revocation for delegatable anonymous credentials. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 423–440. Springer, Berlin, Heidelberg (Mar 2011). https://doi.org/10.1007/978-3-642-19379-8_26
2. Begum, N., Nakanishi, T.: Issuer-revocable issuer-hiding attribute-based credentials using an accumulator. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **108**(9), 1265–1274 (2025). <https://doi.org/10.1587/TRANSFUN.2024EAP1136>, <https://doi.org/10.1587/transfun.2024eap1136>
3. Belenkiy, M., Camenisch, J., Chase, M., Kohlweiss, M., Lysyanskaya, A., Shacham, H.: Randomizable proofs and delegatable anonymous credentials. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 108–125. Springer, Berlin, Heidelberg (Aug 2009). https://doi.org/10.1007/978-3-642-03356-8_7
4. Bellare, M., Micciancio, D., Warinschi, B.: Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 614–629. Springer, Berlin, Heidelberg (May 2003). https://doi.org/10.1007/3-540-39200-9_38
5. 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. LNCS, vol. 7658, pp. 626–643. Springer, Berlin, Heidelberg (Dec 2012). https://doi.org/10.1007/978-3-642-34961-4_38
6. Bichsel, P., Camenisch, J., Dubovitskaya, M., Enderlein, R.R., Krenn, S., Krontiris, I., Lehmann, A., Neven, G., Paquin, C., Preiss, F., Rannenberg, K., Sabouri, A.: An architecture for privacy-abcs. In: Rannenberg, K., Camenisch, J., Sabouri, A. (eds.) Attribute-based Credentials for Trust: Identity in the Information Society, pp. 11–78. Springer (2015). https://doi.org/10.1007/978-3-319-14439-9_2, https://doi.org/10.1007/978-3-319-14439-9_2
7. Blum, M., Feldman, P., Micali, S.: Non-interactive zero-knowledge and its applications (extended abstract). In: 20th ACM STOC. pp. 103–112. ACM Press (May 1988). <https://doi.org/10.1145/62212.62222>
8. Bobolz, J., Eidens, F., Krenn, S., Ramacher, S., Samelin, K.: Issuer-hiding attribute-based credentials. In: Conti, M., Stevens, M., Krenn, S. (eds.) CANS 21. LNCS, vol. 13099, pp. 158–178. Springer, Cham (Dec 2021). https://doi.org/10.1007/978-3-030-92548-2_9
9. Bosk, D., Frey, D., Gestin, M., Piolle, G.: Hidden issuer anonymous credential. PoPETs **2022**(4), 571–607 (Oct 2022). <https://doi.org/10.56553/popets-2022-0123>
10. Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G.: Bulletproofs: Short proofs for confidential transactions and more. In: 2018 IEEE symposium on security and privacy (SP). pp. 315–334. IEEE (2018)

11. Camenisch, J., Chaabouni, R., shelat, a.: Efficient protocols for set membership and range proofs. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 234–252. Springer, Berlin, Heidelberg (Dec 2008). https://doi.org/10.1007/978-3-540-89255-7_15
12. 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. LNCS, vol. 9566, pp. 3–24. Springer, Cham (Aug 2016). https://doi.org/10.1007/978-3-319-31301-6_1
13. Camenisch, J., Lysyanskaya, A.: An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 93–118. Springer, Berlin, Heidelberg (May 2001). https://doi.org/10.1007/3-540-44987-6_7
14. Camenisch, J., Lysyanskaya, A.: An identity escrow scheme with appointed verifiers. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 388–407. Springer, Berlin, Heidelberg (Aug 2001). https://doi.org/10.1007/3-540-44647-8_23
15. Camenisch, J., Lysyanskaya, A.: A signature scheme with efficient protocols. In: Cimato, S., Galdi, C., Persiano, G. (eds.) SCN 02. LNCS, vol. 2576, pp. 268–289. Springer, Berlin, Heidelberg (Sep 2003). https://doi.org/10.1007/3-540-36413-7_20
16. Camenisch, J., Lysyanskaya, A.: Signature schemes and anonymous credentials from bilinear maps. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 56–72. Springer, Berlin, Heidelberg (Aug 2004). https://doi.org/10.1007/978-3-540-28628-8_4
17. Chaum, D.: Blind signatures for untraceable payments. In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds.) CRYPTO’82. pp. 199–203. Plenum Press, New York, USA (1982). https://doi.org/10.1007/978-1-4757-0602-4_18
18. Chaum, D., van Heyst, E.: Group signatures. In: Davies, D.W. (ed.) Advances in Cryptology - EUROCRYPT ’91, Workshop on the Theory and Application of Cryptographic Techniques, Brighton, UK, April 8–11, 1991, Proceedings. Lecture Notes in Computer Science, vol. 547, pp. 257–265. Springer (1991). https://doi.org/10.1007/3-540-46416-6_22, https://doi.org/10.1007/3-540-46416-6_22
19. Cramer, R., Damgård, I., Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols. In: Desmedt, Y. (ed.) CRYPTO’94. LNCS, vol. 839, pp. 174–187. Springer, Berlin, Heidelberg (Aug 1994). https://doi.org/10.1007/3-540-48658-5_19
20. Crites, E.C., Lysyanskaya, A.: Delegatable anonymous credentials from mercurial signatures. In: Matsui, M. (ed.) CT-RSA 2019. LNCS, vol. 11405, pp. 535–555. Springer, Cham (Mar 2019). https://doi.org/10.1007/978-3-030-12612-4_27
21. ETSI: Electronic Signatures and Trust Infrastructures (ESI); Selective Disclosure and Zero-Knowledge Proofs Applied to Electronic Attestation of Attributes; Part 2: Implementation in EUDI Wallet. Draft Technical Specification DTS/ESI-0019476-2, European Telecommunications Standards Institute (2026), draft TS 119 476-2, Version 0.0.4
22. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO’86. LNCS, vol. 263, pp. 186–194. Springer, Berlin, Heidelberg (Aug 1987). https://doi.org/10.1007/3-540-47721-7_12
23. Fischlin, M.: Communication-efficient non-interactive proofs of knowledge with online extractors. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 152–168. Springer, Berlin, Heidelberg (Aug 2005). https://doi.org/10.1007/11535218_10
24. Fujisaki, E., Suzuki, K.: Traceable ring signature. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 181–200. Springer, Berlin, Heidelberg (Apr 2007). https://doi.org/10.1007/978-3-540-71677-8_13
25. Gentry, C.: Practical identity-based encryption without random oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Berlin, Heidelberg (May / Jun 2006). https://doi.org/10.1007/11761679_27
26. Griffy, S., Lysyanskaya, A., Mir, O., Perez-Kempner, O., Slamanig, D.: Delegatable anonymous credentials from mercurial signatures with stronger privacy. In: Chung, K.M., Sasaki, Y. (eds.) ASIACRYPT 2024, Part II. LNCS, vol. 15485, pp. 296–325. Springer, Singapore (Dec 2024). https://doi.org/10.1007/978-981-96-0888-1_10
27. Groth, J.: Efficient fully structure-preserving signatures for large messages. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015, Part I. LNCS, vol. 9452, pp. 239–259. Springer, Berlin, Heidelberg (Nov / Dec 2015). https://doi.org/10.1007/978-3-662-48797-6_11
28. Haböck, U., Krenn, S.: Breaking and fixing anonymous credentials for the cloud. In: Mu, Y., Deng, R.H., Huang, X. (eds.) CANS 19. LNCS, vol. 11829, pp. 249–269. Springer, Cham (Oct 2019). https://doi.org/10.1007/978-3-030-31578-8_14
29. Héban, C., Pointcheval, D.: Traceable constant-size multi-authority credentials. In: Galdi, C., Jarecki, S. (eds.) SCN 22. LNCS, vol. 13409, pp. 411–434. Springer, Cham (Sep 2022). https://doi.org/10.1007/978-3-031-14791-3_18

30. ISO/IEC: ISO/IEC CD 24843: Information security — Attribute-based credentials (2026), <https://www.iso.org/standard/91042.html>, committee Draft, under development
31. Jia, M., Chen, J., Wang, Y., He, K., Shi, M., Du, R.: Multi-authority anonymous credentials with efficient and decentralized supervision. *IEEE Trans. Inf. Forensics Secur.* **20**, 8154–8166 (2025). <https://doi.org/10.1109/TIFS.2025.3592554>, <https://doi.org/10.1109/TIFS.2025.3592554>
32. Kaaniche, N., Kane, S., Laurent, M., Traoré, J.: Improved issuer hiding for bbs-based anonymous credentials. *Cryptology ePrint Archive* (2026)
33. Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: Smart, N.P. (ed.) *EUROCRYPT 2008*. LNCS, vol. 4965, pp. 146–162. Springer, Berlin, Heidelberg (Apr 2008). https://doi.org/10.1007/978-3-540-78967-3_9
34. Katz, J., Sefranek, M.: Issuer hiding for BBS-based anonymous credentials. *Cryptology ePrint Archive, Report 2025/2080* (2025), <https://eprint.iacr.org/2025/2080>
35. Katz, J., Sefranek, M.: Issuer hiding for BBS-based anonymous credentials. pp. 437–465. LNCS (2026). https://doi.org/10.1007/978-3-032-26737-5_14
36. Kiayias, A., Tsiounis, Y., Yung, M.: Traceable signatures. In: Cachin, C., Camenisch, J. (eds.) *EUROCRYPT 2004*. LNCS, vol. 3027, pp. 571–589. Springer, Berlin, Heidelberg (May 2004). https://doi.org/10.1007/978-3-540-24676-3_34
37. Krenn, S.: Bringing zero-knowledge proofs of knowledge to practice. Ph.D. thesis, University of Fribourg (2012), <https://d-nb.info/1025361660>
38. Krenn, S., Lorünser, T., Salzer, A., Striecks, C.: Towards attribute-based credentials in the cloud. In: Capkun, S., Chow, S.S.M. (eds.) *CANS 17*. LNCS, vol. 11261, pp. 179–202. Springer, Cham (Nov / Dec 2017). https://doi.org/10.1007/978-3-030-02641-7_9
39. Krenn, S., Samelin, K., Slamanig, D.: Foundations of dynamic group signatures: The case of malicious openers and issuers. *IACR Cryptol. ePrint Arch.* **2025**, 1752 (2025), <https://eprint.iacr.org/2025/1752>
40. Lipmaa, H.: On diophantine complexity and statistical zero-knowledge arguments. In: Lai, C.S. (ed.) *ASIACRYPT 2003*. LNCS, vol. 2894, pp. 398–415. Springer, Berlin, Heidelberg (Nov / Dec 2003). https://doi.org/10.1007/978-3-540-40061-5_26
41. Looker, T., Kalos, V., Whitehead, A., Lodder, M.: The bbs signature scheme. Internet-Draft draft-irtf-cfrg-bbs-signatures-07 (Sep 2024), <https://datatracker.ietf.org/doc/draft-irtf-cfrg-bbs-signatures/07/>, work in progress
42. Mir, O., Bauer, B., Griffy, S., Lysyanskaya, A., Slamanig, D.: Aggregate signatures with versatile randomization and issuer-hiding multi-authority anonymous credentials. In: Meng, W., Jensen, C.D., Cremers, C., Kirda, E. (eds.) *ACM CCS 2023*. pp. 30–44. ACM Press (Nov 2023). <https://doi.org/10.1145/3576915.3623203>
43. Mir, O., Slamanig, D., Bauer, B., Mayrhofer, R.: Practical delegatable anonymous credentials from equivalence class signatures. *PopETs* **2023**(3), 488–513 (Jul 2023). <https://doi.org/10.56553/popets-2023-0093>
44. Mir, O., Slamanig, D., Bauer, B., Mayrhofer, R.: Practical delegatable anonymous credentials from equivalence class signatures. *Proc. Priv. Enhancing Technol.* **2023**(3), 488–513 (2023). <https://doi.org/10.56553/POPETS-2023-0093>, <https://doi.org/10.56553/popets-2023-0093>
45. Nguyen, L.: Accumulators from bilinear pairings and applications. In: Menezes, A. (ed.) *CT-RSA 2005*. LNCS, vol. 3376, pp. 275–292. Springer, Berlin, Heidelberg (Feb 2005). https://doi.org/10.1007/978-3-540-30574-3_19
46. Okano, H., Emura, K., Ishibashi, T., Ohigashi, T., Suzuki, T.: Implementation of a strongly robust identity-based encryption scheme over type-3 pairings. *Int. J. Netw. Comput.* **10**(2), 174–188 (2020), <http://www.ijnc.org/index.php/ijnc/article/view/228>
47. Orrù, M.: Revisiting keyed-verification anonymous credentials. In: *ACM CCS 2025*. pp. 1188–1199. ACM Press (Nov 2025). <https://doi.org/10.1145/3719027.3765163>
48. Sanders, O., Traoré, J.: Compact issuer-hiding authentication, application to anonymous credential. *PopETs* **2024**(3), 645–658 (Jul 2024). <https://doi.org/10.56553/popets-2024-0097>
49. Schnorr, C.P.: Efficient identification and signatures for smart cards. In: Brassard, G. (ed.) *CRYPTO’89*. LNCS, vol. 435, pp. 239–252. Springer, New York (Aug 1990). https://doi.org/10.1007/0-387-34805-0_22
50. Yang, H., Yu, Y., Zhang, Q., Xin, K., Li, H., Yu, D.: Privately accountable multi-signature for issuer-hiding credentials. *IEEE Trans. Dependable Secur. Comput.* **23**(5), 10586–10598 (2026). <https://doi.org/10.1109/TDSC.2026.3704353>

A Deferred Proofs

A.1 Proof of the Theorem 1

Proof. We define the following set of games to show that the advantage of an adversary $\mathcal{A}$ to break unlinkability is negligible. Let

$$x_0 = (ipk_{i_0}, t) \quad \text{and} \quad x_1 = (ipk_{i_1}, t)$$

denote the two challenge attributes of the predicate-encryption scheme. We show through a sequence of hybrids that a real presentation generated using the first challenge credential is computationally indistinguishable from one generated using the second challenge credential.

Game 0. This is the real challenge experiment with the challenge bit fixed to $b = 0$. In particular, the challenge presentation is generated honestly as $\mathbf{pt}^* = (c_0^*, \pi_0^*)$, where $c_0^* \leftarrow_{\$} \Lambda.\text{Enc}(opk, x_0, 1_{\mathcal{M}}; r)$, and π_0^* is an honestly generated NIZK proof using the witness corresponding to $(ipk_{i_0}, \text{cred}_0, \vec{a}_0, r)$.

Game 1. Game 1 is identical to Game 0, except that the NIZK proof contained in the challenge presentation is replaced by a simulated proof.

During setup, the reduction generates $(pp_{\Pi}, \tau) \leftarrow \mathcal{S}_1(1^\lambda)$, where τ is kept private by the reduction. The challenge ciphertext remains unchanged:

$$c_0^* \leftarrow_{\$} \Lambda.\text{Enc}(opk, x_0, 1_{\mathcal{M}}; r),$$

but the proof is generated as

$$\pi_{\text{sim}}^* \leftarrow \mathcal{S}_2(pp_{\Pi}, \tau, \text{stmt}_0^*).$$

Hence, $\mathbf{pt}^* = (c_0^*, \pi_{\text{sim}}^*)$.

By the zero-knowledge property of Π , there exists a PPT adversary $\mathcal{B}_1$ such that

$$|\Pr[\mathcal{A}(G_0) = 1] - \Pr[\mathcal{A}(G_1) = 1]| \leq \text{Adv}_{\Pi, \mathcal{B}_1}^{\text{ZK}}(\lambda).$$

Game 2. Game 2 is identical to Game 1, except that we replace the encryption under x_0 by an encryption under x_1 . Namely, we replace

$$c_0^* \leftarrow_{\$} \Lambda.\text{Enc}(opk, x_0, 1_{\mathcal{M}}; r)$$

with

$$c_1^* \leftarrow_{\$} \Lambda.\text{Enc}(opk, x_1, 1_{\mathcal{M}}; r).$$

The proof remains simulated, now with respect to the corresponding public statement:

$$\pi_{\text{sim}}^* \leftarrow \mathcal{S}_2(pp_{\Pi}, \tau, \text{stmt}_1^*),$$

and the challenge presentation is $\mathbf{pt}^* = (c_1^*, \pi_{\text{sim}}^*)$.

The transition from Game 1 to Game 2 follows from the attribute-hiding security of Λ . An opening query for an issuer/epoch pair (ipk', t') corresponds to a derived-key query for the equality predicate

$$f_{ipk', t'}(ipk, t) = \begin{cases} 1 & \text{if } ipk = ipk' \text{ and } t = t', \\ 0 & \text{otherwise.} \end{cases}$$

By the winning conditions of the unlinkability experiment, the adversary cannot obtain opening keys for either challenge pair

$$x_0 = (ipk_{i_0}, t) \quad \text{or} \quad x_1 = (ipk_{i_1}, t).$$

Therefore, for every opening query (ipk', t') made by the adversary, we have

$$f_{ipk', t'}(x_0) = f_{ipk', t'}(x_1) = 0.$$

Moreover, the encrypted message is the same in both challenge ciphertexts, namely $1_{\mathcal{M}}$. Hence, all derived-key queries satisfy the winning condition of the attribute-hiding experiment.

The reduction $\mathcal{B}_2$ uses the challenge ciphertext received from the attribute-hiding experiment as the ciphertext contained in $\mathbf{pt}^*$. Since the NIZK proof is simulated, $\mathcal{B}_2$ does not need to know the challenge bit or the encryption randomness in order to generate the proof.

Thus, there exists a PPT adversary $\mathcal{B}_2$ such that

$$|\Pr[\mathcal{A}(G_1) = 1] - \Pr[\mathcal{A}(G_2) = 1]| \leq 2\text{Adv}_{\mathcal{A}, \mathcal{B}_2}^{\text{AH}}(\lambda).$$

The factor 2 appears because $\text{Adv}_{\mathcal{A}, \mathcal{B}_2}^{\text{AH}}$ is defined as the advantage over random guessing in the attribute-hiding experiment.

Game 3. Game 3 is identical to Game 2, except that the simulated NIZK component is changed back to the real NIZK distribution. In particular, the NIZK public parameters are generated honestly and the proof is honestly generated using the witness corresponding to the second challenge credential. Thus, $\mathbf{pt}^* = (c_1^*, \pi_1^*)$ is distributed exactly as a real presentation generated with challenge bit $b = 1$.

Again, by the zero-knowledge property of Π , there exists a PPT adversary $\mathcal{B}_3$ such that

$$|\Pr[\mathcal{A}(G_2) = 1] - \Pr[\mathcal{A}(G_3) = 1]| \leq \text{Adv}_{\Pi, \mathcal{B}_3}^{\text{ZK}}(\lambda).$$

Combining the three transitions, we obtain

$$\begin{aligned} & |\Pr[\mathcal{A}(G_0) = 1] - \Pr[\mathcal{A}(G_3) = 1]| \\ & \leq \text{Adv}_{\Pi, \mathcal{B}_1}^{\text{ZK}}(\lambda) + 2\text{Adv}_{\mathcal{A}, \mathcal{B}_2}^{\text{AH}}(\lambda) + \text{Adv}_{\Pi, \mathcal{B}_3}^{\text{ZK}}(\lambda). \end{aligned}$$

Since G_0 is the real presentation distribution for $b = 0$ and G_3 is the real presentation distribution for $b = 1$, let

$$p_0 = \Pr[\mathcal{A}(G_0) = 1] \quad \text{and} \quad p_1 = \Pr[\mathcal{A}(G_3) = 1].$$

Therefore,

$$\begin{aligned} \text{Adv}_{\mathcal{A}}^{\text{Unlinkability}}(\lambda) & \leq \frac{1}{2}\text{Adv}_{\Pi, \mathcal{B}_1}^{\text{ZK}}(\lambda) + \text{Adv}_{\mathcal{A}, \mathcal{B}_2}^{\text{AH}}(\lambda) \\ & \quad + \frac{1}{2}\text{Adv}_{\Pi, \mathcal{B}_3}^{\text{ZK}}(\lambda), \end{aligned}$$

which is negligible. Hence, Construction 1 satisfies unlinkability.

A.2 Proof of Theorem 2

Proof. Let $\mathcal{A}$ be a PPT adversary against the unforgeability of Construction 1. We show that a successful forgery by $\mathcal{A}$ either contradicts the simulation-sound extractability of Π , the EUF-CMA security of Σ_V , or the EUF-CMA security of Σ_I .

We first modify the experiment by replacing the NIZK proofs generated by $\mathcal{O}_{\text{Present}}$ with simulated proofs. During setup, the reduction generates the NIZK public parameters together with the corresponding simulation and extraction trapdoors. For every presentation query, the ciphertext is generated honestly, while the NIZK proof is replaced by a simulated proof for the corresponding public statement. By the zero-knowledge property of Π , this modification changes the adversary's success probability only negligibly.

Assume now that $\mathcal{A}$ outputs $(\mathbf{pt}^*, \phi^*, t^*, \text{ctx}^*, \text{opk}^*)$ satisfying the winning conditions of the unforgeability experiment, where $\mathbf{pt}^* = (c^*, \pi^*)$. Since π^* verifies and, by the freshness condition of the unforgeability experiment, it is fresh with respect to the simulated presentation proofs, we apply the extractor of Π to π^* .

Let **ExtFail** denote the event that $\mathcal{A}$ wins but the extractor fails to output a valid witness. By simulation-sound extractability of Π ,

$$\Pr[\text{ExtFail}] \leq \text{Adv}_{\Pi, \mathcal{B}_0}^{\text{SSE}}(\lambda)$$

for some PPT adversary $\mathcal{B}_0$, and is therefore negligible.

Whenever extraction succeeds, we obtain a witness

$$(\sigma, ipk, \text{cred}, \vec{a}, r)$$

such that

$$\Sigma_V.\text{Verify}(vpk, \sigma, ipk) = 1, \wedge \Sigma_I.\text{Verify}(ipk, \text{cred}, \vec{a}) = 1, \wedge \phi^*(\vec{a}) = 1,$$

and

$$c^* = \Lambda.\text{Enc}(opk^*, (ipk, t^*), 1_{\mathcal{M}}; r).$$

We distinguish the following two events:

$$\begin{aligned} \text{PolForge} &:= \{\mathcal{A} \text{ wins}, \neg \text{ExtFail}, ipk \notin I^*\}, \\ \text{CredForge} &:= \{\mathcal{A} \text{ wins}, \neg \text{ExtFail}, ipk \in I^*\}. \end{aligned}$$

Hence,

$$\Pr[\mathcal{A} \text{ wins}] \leq \Pr[\text{ExtFail}] + \Pr[\text{PolForge}] + \Pr[\text{CredForge}] + \varepsilon(\lambda),$$

where the additional negligible term accounts for replacing the honest presentation proofs by simulated proofs.

We now show that both **PolForge** and **CredForge** occur only with negligible probability.

Scenario 1: Policy forgery. Suppose that $\Pr[\text{PolForge}]$ is non-negligible. We construct a PPT adversary $\mathcal{B}_1$ against the EUF-CMA security of Σ_V .

The adversary $\mathcal{B}_1$ receives a verification key vpk and access to a signing oracle $\mathcal{O}_{\text{Sign}}$ for Σ_V . It generates the issuer key pairs

$$(isk_i, ipk_i) \leftarrow_{\$} \text{IKGen}(pp), \quad i = 1, \dots, m,$$

and the opener key pairs

$$(osk_i, opk_i) \leftarrow_{\$} \text{OKGen}(pp), \quad i = 1, \dots, n.$$

It then runs $\mathcal{A}$ and answers $\mathcal{O}_{\text{Issue}}$, $\mathcal{O}_{\text{Present}}$, and $\mathcal{O}_{\text{Reveal}}$ queries consistently with the modified experiment.

When $\mathcal{A}$ outputs I^* , the reduction queries its signing oracle on every $ipk_i \in I^*$ and obtains signatures

$$\sigma_i \leftarrow_{\$} \mathcal{O}_{\text{Sign}}(ipk_i).$$

It then defines

$$\text{pol}^* = (vpk, \{(\sigma_i, ipk_i)\}_{ipk_i \in I^*})$$

and continues the simulation.

If $\mathcal{A}$ wins and the event **PolForge** occurs, extraction gives a witness containing (σ, ipk) such that $\Sigma_V.\text{Verify}(vpk, \sigma, ipk) = 1$ but $ipk \notin I^*$. The reduction queried its signing oracle only on the issuer public keys contained in I^* . Hence ipk was never submitted to $\mathcal{O}_{\text{Sign}}$.

Therefore, (ipk, σ) is a valid EUF-CMA forgery for Σ_V . Consequently,

$$\Pr[\text{PolForge}] \leq \text{Adv}_{\Sigma_V, \mathcal{B}_1}^{\text{EUF-CMA}}(\lambda),$$

which is negligible.

Scenario 2: Credential forgery. Suppose now that $\Pr[\text{CredForge}]$ is non-negligible. We construct a PPT adversary $\mathcal{B}_2$ against the EUF-CMA security of Σ_I .

The adversary $\mathcal{B}_2$ receives a verification key pk and access to a signing oracle $\mathcal{O}_{\text{Sign}}$ for Σ_I . It chooses $k \leftarrow_{\$} \{1, \dots, m\}$ and embeds its challenge verification key as $\text{ipk}_k = \text{pk}$. For every $i \neq k$, it honestly generates

$$(\text{isk}_i, \text{ipk}_i) \leftarrow_{\$} \text{IKGen}(pp).$$

It also generates all opener keys honestly.

The reduction then runs $\mathcal{A}$. Oracle queries are answered as follows.

- For a query

$$\mathcal{O}_{\text{Issue}}(\text{ipk}_i, \vec{a}),$$

the reduction creates the corresponding entry in Q_{Issue} and returns its handle j . If $i \neq k$, the credential can be generated honestly using isk_i . If $i = k$, the reduction does not yet query its signing oracle; the credential is sampled lazily because it is not revealed to $\mathcal{A}$ by $\mathcal{O}_{\text{Issue}}$.

- For a query

$$\mathcal{O}_{\text{Reveal}}(j),$$

let $(\cdot, \text{ipk}_i, \vec{a}, j)$ be the corresponding entry in Q_{Issue} . If $i = k$ and the credential has not previously been generated, the reduction queries

$$\text{cred} \leftarrow_{\$} \mathcal{O}_{\text{Sign}}(\vec{a}),$$

stores the resulting credential, and returns it. For $i \neq k$, the credential is generated using isk_i . Repeated reveal queries are answered consistently with the same credential.

- For

$$\mathcal{O}_{\text{Present}}(j, \text{pol}, \phi, t, \text{ctx}),$$

the reduction retrieves the corresponding issuer and attributes from Q_{Issue} . If the query does not satisfy the validity conditions required by **Present**, the reduction returns $\perp$. Otherwise, it generates the ciphertext honestly and replaces the NIZK proof by a simulated proof for the corresponding valid public statement. In particular, when the credential belongs to the embedded issuer $\text{ipk}_k = \text{pk}$, the reduction can simulate the proof without knowing the underlying credential.

After $\mathcal{A}$ outputs I^* , the reduction generates

$$\text{pol}^* \leftarrow_{\$} \text{PresPolicy}(I^*)$$

honestly and continues answering oracle queries as above.

Suppose that $\mathcal{A}$ wins and **CredForge** occurs. The extractor outputs $(\sigma, \text{ipk}, \text{cred}, \vec{a}, r)$ with

$$\text{ipk} \in I^* \wedge \Sigma_I.\text{Verify}(\text{ipk}, \text{cred}, \vec{a}) = 1 \wedge \phi^*(\vec{a}) = 1.$$

Consider the event that the extracted issuer is the issuer in which the EUF-CMA challenge key was embedded, namely $\text{ipk} = \text{ipk}_k = \text{pk}$. In this case,

$$\Sigma_I.\text{Verify}(\text{pk}, \text{cred}, \vec{a}) = 1.$$

Moreover, the message $\vec{a}$ cannot previously have been submitted by $\mathcal{B}_2$ to its signing oracle. Indeed, the reduction queries $\mathcal{O}_{\text{Sign}}(\vec{a})$ only when the corresponding credential is revealed through $\mathcal{O}_{\text{Reveal}}$. But since

$$\text{ipk} = \text{pk} \in I^* \quad \text{and} \quad \phi^*(\vec{a}) = 1,$$

such a reveal query would violate the winning condition of the unforgeability experiment. Thus $\vec{a}$ was never queried to the signing oracle, and $(\vec{a}, \text{cred})$ is a valid EUF-CMA forgery for Σ_I .

Since k is chosen uniformly from $\{1, \dots, m\}$,

$$\Pr[\mathcal{B}_2 \text{ wins}] \geq \frac{1}{m} \Pr[\text{CredForge}].$$

Hence,

$$\Pr[\text{CredForge}] \leq m \cdot \text{Adv}_{\Sigma_I, \mathcal{B}_2}^{\text{EUF-CMA}}(\lambda).$$

Since m is polynomially bounded and Σ_I is EUF-CMA secure, $\Pr[\text{CredForge}]$ is negligible. Combining the three cases, we obtain

$$\begin{aligned} \Pr[\mathbf{Exp}_{\mathcal{A}}^{\text{Unforgeability}}(\lambda) = 1] &\leq \varepsilon(\lambda) + \text{Adv}_{\Pi, \mathcal{B}_0}^{\text{SSE}}(\lambda) \\ &\quad + \text{Adv}_{\Sigma_V, \mathcal{B}_1}^{\text{EUF-CMA}}(\lambda) + m \cdot \text{Adv}_{\Sigma_I, \mathcal{B}_2}^{\text{EUF-CMA}}(\lambda). \end{aligned}$$

All terms on the right-hand side are negligible. Therefore,

$$\Pr[\mathbf{Exp}_{\mathcal{A}}^{\text{Unforgeability}}(\lambda) = 1] \leq \varepsilon(\lambda),$$

and Construction 1 satisfies unforgeability.

A.3 Proof of Theorem 3

Proof. An adversary $\mathcal{A}$ wins the issuer-non-frameability experiment if it produces a fresh valid presentation pt^* that opens to the targeted honest issuer ipk , without having obtained through $\mathcal{O}_{\text{Reveal}}$ a credential of ipk on attributes satisfying the challenge predicate.

Game 0. This is the original issuer-non-frameability experiment.

Game 1. Game 1 is identical to Game 0, except that the NIZK proofs generated by $\mathcal{O}_{\text{Present}}$ are replaced by simulated proofs. During setup, the reduction generates the NIZK public parameters together with the corresponding simulation trapdoor and uses the simulator to answer presentation queries.

The ciphertexts contained in the presentations are generated honestly; only the corresponding NIZK proofs are simulated.

By the zero-knowledge property of Π , there exists a PPT adversary $\mathcal{B}_1$ such that

$$|\Pr[G_0 = 1] - \Pr[G_1 = 1]| \leq \text{Adv}_{\Pi, \mathcal{B}_1}^{\text{ZK}}(\lambda).$$

Game 2. Game 2 is identical to Game 1, except that, if $\mathcal{A}$ outputs a valid challenge presentation $\text{pt}^* = (c^*, \pi^*)$, we apply the extractor of Π to π^* .

Since the winning condition requires the challenge presentation to be fresh with respect to the presentation queries, simulation-sound extractability guarantees extraction even though $\mathcal{A}$ has previously seen simulated proofs. The extractor outputs a witness $(\sigma^*, ipk^*, \text{cred}^*, \vec{a}^*, r^*)$ satisfying

$$\begin{aligned} \Sigma_V.\text{Verify}(\sigma^*, ipk^*, vpk^*) &= 1, \wedge \Sigma_I.\text{Verify}(\text{cred}^*, \vec{a}^*, ipk^*) = 1, \wedge \phi^*(\vec{a}^*) = 1, \\ \wedge c^* &= A.\text{Enc}(opk, (ipk^*, t^*), 1_{\mathcal{M}}; r^*). \end{aligned}$$

Game 2 aborts if extraction fails or if the extracted witness does not satisfy the above relation.

By simulation-sound extractability of Π , there exists a PPT adversary $\mathcal{B}_2$ such that

$$|\Pr[G_1 = 1] - \Pr[G_2 = 1]| \leq \text{Adv}_{\Pi, \mathcal{B}_2}^{\text{SIM-Ext}}(\lambda).$$

Game 3. Game 3 is identical to Game 2, except that we abort if the challenge presentation opens to the targeted issuer (ipk, t^*) , while the issuer extracted from the NIZK proof is different from ipk , i.e., $ipk^* \neq ipk$. By the equality predicate of Λ and opening soundness Definition 3, a ciphertext that is bound by the extracted witness to (ipk^*, t^*) cannot simultaneously be accepted by **Open** as a ciphertext for (ipk, t^*) with $ipk^* \neq ipk$, except with negligible probability.

Suppose that this event occurs. We construct an adversary $\mathcal{B}_3$ against the opening soundness of Λ .

The adversary $\mathcal{B}_3$ defines

$$f = f_{ipk, t^*}, \quad x_0 = (ipk, t^*), \quad x_1 = (ipk^*, t^*),$$

and sets $sk^* = osk_{ipk, t^*}^*$. Since f_{ipk, t^*} is the equality predicate, we have $f(x_0) = 1$. Moreover, because $ipk^* \neq ipk$, we have $f(x_1) = 0$.

During the execution of **Open**, a fresh validation ciphertext

$$c_0 \leftarrow_{\$} \Lambda.\text{Enc}(opk, x_0, 1_{\mathcal{M}})$$

is generated. Since **Open** does not abort, we have

$$\Lambda.\text{Dec}(sk^*, c_0) = 1_{\mathcal{M}}.$$

Furthermore, since **Open** returns 1 on the challenge presentation,

$$\Lambda.\text{Dec}(sk^*, c^*) = 1_{\mathcal{M}}.$$

By extraction in Game 2, we also know that

$$c^* = \Lambda.\text{Enc}(opk, x_1, 1_{\mathcal{M}}; r^*).$$

Thus, the adversarially chosen key sk^* successfully decrypts both a ciphertext associated with an attribute x_0 satisfying $f(x_0) = 1$ and a ciphertext associated with an attribute x_1 satisfying $f(x_1) = 0$. This is exactly the bad event excluded by the malicious-key opening soundness of Λ .

Hence, there exists a PPT adversary $\mathcal{B}_3$ such that

$$|\Pr[G_2 = 1] - \Pr[G_3 = 1]| \leq \text{Adv}_{\Lambda, \mathcal{B}_3}^{\text{OS}}(\lambda).$$

Consequently, whenever $\mathcal{A}$ wins in Game 3, we have $ipk^* = ipk$.

Together with the extracted witness relation, this implies

$$\Sigma_I.\text{Verify}(\text{cred}^*, \vec{a}^*, ipk) = 1.$$

Thus, cred^* is a valid credential under the targeted honest issuer ipk on attributes $\vec{a}^*$ satisfying $\phi^*(\vec{a}^*) = 1$. We now construct a PPT adversary $\mathcal{B}_4$ against the EUF-CMA security of Σ_I .

Reduction to the underline signature. The adversary $\mathcal{B}_4$ receives a verification key pk and access to a signing oracle $\mathcal{O}_{\text{Sign}}$. It embeds the EUF-CMA challenge key as the honest issuer public key, $ipk = \text{pk}$, and runs $\mathcal{A}$.

The oracles are simulated as follows.

- For a query $\mathcal{O}_{\text{Issue}}(ipk, \vec{a})$, $\mathcal{B}_4$ creates a fresh handle j and stores $(\perp, ipk, \vec{a}, j)$ in Q_{Issue} , but does not yet query its signing oracle. This is possible because $\mathcal{O}_{\text{Issue}}$ returns only the handle j , and not the credential itself.
- For a query $\mathcal{O}_{\text{Present}}(j, \text{pol}, \phi, t, \text{ctx}, opk)$, $\mathcal{B}_4$ retrieves the corresponding attributes $\vec{a}$, generates the ciphertext honestly, and generates the NIZK proof using the simulator, as in Game 1. Hence, it does not need to know the corresponding credential and does not query $\mathcal{O}_{\text{Sign}}$.

- For a query $\mathcal{O}_{\text{Reveal}}(j)$, $\mathcal{B}_4$ retrieves the corresponding entry $(\perp, ipk, \vec{a}, j)$. If a credential has not previously been generated for this entry, it queries $\text{cred} \leftarrow_{\$} \mathcal{O}_{\text{Sign}}(\vec{a})$, replaces $(\perp, ipk, \vec{a}, j)$ in Q_{Issue} by $(\text{cred}, ipk, \vec{a}, j)$, and stores j in Q_{Reveal} . It then returns cred . Repeated reveal queries are answered consistently using the stored credential $(\text{cred}, ipk, \vec{a}, j)$.

After simulating the oracles, suppose that $\mathcal{A}$ wins Game 3. Then, by extraction, we obtain $(\sigma^*, ipk, \text{cred}^*, \vec{a}^*, r^*)$ with

$$\Sigma_I.\text{Verify}(\text{cred}^*, \vec{a}^*, ipk) = 1 \wedge \phi^*(\vec{a}^*) = 1.$$

We claim that $\mathcal{B}_4$ never queried its signing oracle on $\vec{a}^*$. Indeed, $\mathcal{B}_4$ queries $\mathcal{O}_{\text{Sign}}$ only when answering an $\mathcal{O}_{\text{Reveal}}$ query. If a credential corresponding to $\vec{a}^*$ had been revealed, then there would exist a handle j such that

$$(\text{cred}, ipk, \vec{a}^*, j) \in Q_{\text{Issue}}, \quad \wedge j \in Q_{\text{Reveal}} \quad \wedge \phi^*(\vec{a}^*) = 1$$

This would violate winning condition (f) of the issuer-non-frameability experiment.

Therefore, $\vec{a}^*$ was never submitted to the signing oracle, while cred^* is a valid signature on $\vec{a}^*$ under $ipk = \text{pk}$. Hence, $(\vec{a}^*, \text{cred}^*)$ is a valid EUF-CMA forgery against Σ_I .

It follows that $\Pr[G_3 = 1] \leq \text{Adv}_{\Sigma_I, \mathcal{B}_4}^{\text{EUF-CMA}}(\lambda)$. Combining the game transitions, we obtain

$$\begin{aligned} \Pr[\text{Exp}_{\mathcal{A}}^{\text{Issuer-Non-Frameability}}(\lambda) = 1] &\leq \text{Adv}_{\Pi, \mathcal{B}_1}^{\text{ZK}}(\lambda) + \text{Adv}_{\Pi, \mathcal{B}_2}^{\text{SIM-Ext}}(\lambda) \\ &\quad + \text{Adv}_{A, \mathcal{B}_3}^{\text{OS}}(\lambda) + \text{Adv}_{\Sigma_I, \mathcal{B}_4}^{\text{EUF-CMA}}(\lambda). \end{aligned}$$

All terms on the right-hand side are negligible. Therefore, Construction 1 satisfies issuer non-frameability.

A.4 Proof of Theorem 4

We show that any PPT adversary $\mathcal{A}$ that wins the opening-soundness experiment of Definition 3 with non-negligible probability can be used to construct a PPT algorithm $\mathcal{B}$ that solves the discrete-logarithm problem in $\mathbb{G}_2$ with non-negligible probability.

Simulation of the public parameters and mpk . Let $\mathcal{B}$ receive a discrete-logarithm challenge $(g_2, h_1 = g_2^x)$, where $x \in \mathbb{Z}_q$ is unknown. The reduction samples $\alpha \leftarrow_{\$} \mathbb{Z}_q^*$ and $h_2, h_3 \leftarrow_{\$} \mathbb{G}_2$, sets $g'_1 = g_1^\alpha$, and defines $mpk = (g'_1, h_1, h_2, h_3)$, and $msk = \alpha$. It then gives (mpk, msk) to $\mathcal{A}$. Since h_1 is distributed uniformly in $\mathbb{G}_2$, this is distributed identically to an honestly generated Gentry master key pair.

Reduction. Let the adversary output an arbitrary, potentially malicious, decryption key $\text{sk}^* = \{(\rho_i, K_i)\}_{i=1}^3$, where $\rho_i \in \mathbb{Z}_q$ and $K_i \in \mathbb{G}_2$. Importantly, we do not require sk^* to have been generated by Gentry.Derive . Together with (id_0, id_1, s_1) that are the values output by $\mathcal{A}$ in a winning execution of the opening-soundness experiment, where $id_0 \neq id_1$ and $s_1 \in \mathbb{Z}_q^*$. The reduction samples $s_0 \leftarrow_{\$} \mathbb{Z}_q^*$ and computes the honestly randomized ciphertext

$$c_0 = \text{Enc}(pp, mpk, id_0, 1_{\mathbb{G}_T}; s_0).$$

It also computes

$$c_1 = \text{Enc}(pp, mpk, id_1, 1_{\mathbb{G}_T}; s_1),$$

using the randomness s_1 chosen by the adversary. For each $j \in 0, 1$, write

$$c_j = (C_{1,j}, C_{2,j}, C_{3,j}, C_{4,j}).$$

The relevant ciphertext components are

$$C_{1,j} = (g'_1 g_1^{-id_j})^{s_j} = g_1^{s_j(\alpha - id_j)} \wedge C_{2,j} = e(g_1, g_2)^{s_j} \wedge C_{3,j} = e(g_1, h_1)^{-s_j}.$$

Now as in the winning condition, suppose that the malicious key sk^* successfully decrypts both ciphertexts to $1_{\mathbb{G}_T}$. Since successful decryption passes the ciphertext-validation checks and returns

$$C_{3,j} e(C_{1,j}, K_1) C_{2,j}^{\rho_1},$$

we obtain, for each $j \in \{0, 1\}$,

$$1_{\mathbb{G}_T} = e(g_1, h_1)^{-s_j} e\left(g_1^{s_j(\alpha - id_j)}, K_1\right) e(g_1, g_2)^{s_j \rho_1}.$$

By bilinearity,

$$1_{\mathbb{G}_T} = e\left(g_1, h_1^{-1} K_1^{\alpha - id_j} g_2^{\rho_1}\right)^{s_j}.$$

Moreover in the decryption we have $C_{2,j} \neq 1_{\mathbb{G}_T}$. where $C_{2,j} = e(g_1, g_2)^{s_j}$, this implies $s_j \neq 0$.

Since the pairing is non-degenerate and $s_j \neq 0$, exponentiation by s_j we have

$$h_1 = K_1^{\alpha - id_j} g_2^{\rho_1}. \quad (1)$$

Applying (1) for id_0 and id_1 gives

$$K_1^{\alpha - id_0} g_2^{\rho_1} = K_1^{\alpha - id_1} g_2^{\rho_1},$$

and therefore

$$K_1^{id_1 - id_0} = 1_{\mathbb{G}_2}.$$

Since $id_0 \neq id_1 \pmod{q}$ (the equality predicate, any winning execution necessarily has this fact), we have $d := id_1 - id_0$ is nonzero in $\mathbb{Z}_q$. As q is prime, d is invertible modulo q . From $K_1^d = 1_{\mathbb{G}_2}$, raising both sides to the power d^{-1} yields

$$K_1 = K_1^{dd^{-1}} = 1_{\mathbb{G}_2} \Rightarrow K_1 = 1_{\mathbb{G}_2}.$$

Substituting this into (1) yields

$$h_1 = g_2^{\rho_1}.$$

This immediately gives a reduction to the discrete-logarithm problem in $\mathbb{G}_2$. Recall that the reduction is given a challenge

$$(g_2, h_1 = g_2^x),$$

where x is unknown. If the adversary wins the malicious-key opening-soundness experiment, the argument above gives

$$h_1 = g_2^{\rho_1}.$$

Since the discrete-logarithm challenge satisfies $h_1 = g_2^x$, it follows that $\rho_1 = x \pmod{q}$. Hence, the reduction outputs ρ_1 and solves the discrete-logarithm instance. Therefore, any PPT adversary that violates opening soundness with non-negligible probability yields a PPT algorithm that solves the discrete-logarithm problem in $\mathbb{G}_2$ with non-negligible probability, contradicting the discrete-logarithm assumption.