Issuer-Hiding Anonymous Tokens and Credentials from Key-Randomizable Signatures

Andrea Flamini[1], Karla Friedrichs[1], Jonathan Katz[2], Watson Ladd[3], Anja Lehmann[1], and Marek Sefranek[4]

[1] Hasso Plattner Institute, University of Potsdam, Germany
{andrea.flamini,karla.friedrichs,anja.lehmann}@hpi.de
[2] Google, USA
jkatz2@gmail.com
[3] Akamai, USA
watsonbladd@gmail.com
[4] TU Wien, Austria
marek.sefranek@tuwien.ac.at

Abstract. Anonymous authentication schemes allow a user to prove they have been authorized by some issuer, without revealing any additional information about the user's identity. Two variants are currently seeing strong real-world interest. *Anonymous tokens* (ATs) allow one-time proofs of authorization, e.g., for privacy-preserving rate limiting as currently being pursued in the MoLE effort at the IETF. *Anonymous credentials* (ACs) incorporate attributes that can be selectively disclosed in an unbounded number of proofs; these underpin digital-identity systems such as the European Digital Identity (EUDI) wallet.

Both anonymous tokens and credentials are envisioned as being deployed in systems with multiple issuers. In standard schemes, proofs reveal which issuer authorized a user, leaking more information than necessary and shrinking a user's anonymity set. To address this, there has been significant recent interest in *issuer-hiding* schemes that only reveal that the authorizing issuer lies in a specified policy set.

We present generic constructions of issuer-hiding ATs/ACs based on *key-randomizable signatures* that give rise to several efficient instantiations. Specifically, we show efficient issuer-hiding ATs based on the Tessaro–Zhu blind-signature scheme, and efficient issuer-hiding ACs (without policy keys) compatible with existing BBS-based schemes.

1 Introduction

Anonymous authentication schemes allow a user to prove they have been authorized by some issuer, without revealing any additional information. In such schemes, at a high level, a trusted issuer uses its secret key sk to generate some cryptographic information certifying an authorized user; that user can subsequently prove to a third-party verifier (who knows the issuer's public key pk) that it is indeed authorized. Such proofs (also called *presentations*) are unlinkable

to the user, and even a verifier who colludes with the issuer cannot distinguish proofs from two different users satisfying the same authorization policy.

Two classes of anonymous authentication schemes have recently received significant real-world attention. *Anonymous tokens* (ATs), aka blind signatures [14], are the simpler of the two. They allow a user and issuer to run an issuance protocol after which the user obtains a token (signature) σ on a nullifier (message) m; the user can subsequently send m, σ to a verifier in order to prove that the user is authorized for some task. (The nullifier is used to detect repeated usage of a token.) That presentation cannot be linked to the execution of the issuance protocol that generated that token, thus allowing the user to remain anonymous. By design, however, each token is intended to be used only once, and repeated presentations of a token can be linked to each other.

Anonymous credentials (ACs) [15,8,9,10] are a more powerful primitive. Here, an issuer generates a credential on a vector of attributes $\boldsymbol{m} = (m_i)_{i \in [n]}$; then, for any subset $\mathcal{D} \subseteq [n]$, the user can prove they hold a credential on disclosed attributes $(m_i)_{i \in \mathcal{D}}$ without revealing anything about the undisclosed attributes $(m_i)_{i \notin \mathcal{D}}$ or about the credential itself. In contrast to tokens, credentials can be used multiple times; not only is the presentation of a credential unlinkable to (the issuance of) the credential itself, but multiple shows of the same credential are also unlinkable to each other.

Both these primitives have seen a surge of real-world interest. Anonymous tokens have been deployed in the context of Privacy Pass [21] to distinguish bots from humans while preserving user privacy, as well as by Google and Apple [60]. Anonymous credentials are being proposed as part of the European effort to build a European Digital Identity (EUDI) wallet [25], rooted in the eIDAS2.0 regulation that mandates technologies enabling unlinkable authentications with minimal disclosure [51]. The nascent PACT/MoLE architecture being developed at the IETF [38,53,54] relies on both anonymous tokens and anonymous credentials. In that system, an *anchor* issues an anonymous token (called an *endorsement*) to a user; endorsements are intended to be "scarce" (and thus difficult for a bot to obtain in bulk) and could be based, e.g., on an email address, paid subscription, or verified phone number. *Moderators* accept endorsements and, in exchange, issue a credential that can be used for subsequent access to services.

In all the applications above, the goal is to allow authorized users to gain access *while revealing the minimal information possible*. Traditional analyses of anonymous authentication schemes assume a single issuer, but all the deployments mentioned above envision an ecosystem of multiple issuers. The presence of multiple issuers introduces a new form of information leakage, since (when using a traditional scheme) a user's presentation reveals the issuer from whom they obtained their token/credential. (Indeed, for standard schemes this is inherent, as the verifier must know which public key to use for verification.) This can be highly problematic. Consider a US-wide system where information from state driver's licenses is embedded in credentials and each state acts as an independent issuer. Although it should be possible for a user to prove, say, that they are a US resident without revealing anything about their address, using standard

anonymous credentials would force a user to reveal which state issued their credential (and hence what state they live in). For PACT/MoLE, learning which anchor issued an endorsement allows a verifier to learn that the user has an account with a specific site, and enables cross-site fingerprinting [38]. In both examples the verifier does not actually need to know which issuer authorized a user; it only wants assurance that the user was certified by *some* issuer in a specified set of issuers (either US states or issuers trusted by the verifier). Ideally, then, in all these applications the identity of the issuer should not be leaked.

Issuer-hiding schemes [3] prevent such leakage. At a high level, such schemes explicitly consider multiple issuers, with respective public keys $\mathsf{pk}_1, \ldots, \mathsf{pk}_\ell$, and users can obtain tokens/credentials from one or more of those issuers. A verifier can specify a *policy set* $\mathcal{P} \subseteq \{\mathsf{pk}_i\}_{i\in[\ell]}$ indicating which issuers they trust in a certain context. Now, a user who holds a valid token/credential with respect to *any* issuer whose key is in $\mathcal{P}$ can use that token/credential *without revealing the issuer from which it obtained that credential* (beyond the fact that its key is in $\mathcal{P}$). Issuer hiding has been identified as an explicit requirement in the context of the EUDI wallet [27] and PACT/MoLE [38], and has also been highlighted by the W3C Anti-Fraud Community Group [55].

We are not aware of prior work on issuer-hiding anonymous tokens in the sense we consider here, (the notion considered by Li et al. [45] hides the issuer from the user during issuance) and thus our work gives the first formal treatment and constructions of that primitive. Several works [2,3,5,17,24,16,47,52,42,31] have explored constructions of issuer-hiding anonymous-credential schemes. Many of these works give bespoke constructions that are not compatible with existing schemes. Others require *policy keys* which complicate deployment [31, Section 7]; we refer to Section 1.2 for further discussion.

Our contributions. We show an efficient and generic approach for incorporating issuer-hiding in existing constructions of anonymous tokens/credentials whose underlying signature scheme is *key-randomizable* [11], while avoiding policy keys. Roughly speaking, key-randomizable signatures are signature schemes with the property that given a valid signature on a message under some public key pk, it is possible to generate a valid signature on the same message under a random public key $\widetilde{\mathsf{pk}}$; see further details in Section 2.

We then show that our generic constructions can be instantiated with existing anonymous token/credential schemes. Specifically, we show in Section 3 how to obtain issuer-hiding ATs based on Tessaro–Zhu blind signatures [58]. Using the observation that BBS signatures are key-randomizable (as used in [31]), we construct an issuer-hiding, BBS-based AC scheme in Section 4, which in contrast to other constructions [42,31] avoids policy keys. Our formalization of key randomizability of BBS signatures may be of independent interest, and may have other applications. As an example, we show that it can be used to prove tight multi-signer security for BBS signatures and tight multi-issuer security for BBS-based ACs.

All our efficient instantiations rely on Σ-protocols for 1-out-of-ℓ disjunctive statements. In Section 5, we consider multiple approaches to instantiating those

protocols, implement each of them, and experimentally compare their tradeoffs in terms of proving time, verification time, and proof size. We find that all instantiations offer good performance for policy sets of up to 64 issuers.

1.1 Overview of Our Techniques

We begin with a brief description of key-randomizable signatures, and then describe our key ideas at a high level.

Key-randomizable signatures. Roughly, a key-randomizable signature scheme provides algorithms RandPK, $\mathsf{RandSig}$ that allow anyone to transform a valid public key/signature pair (pk, σ) for some message into another valid pair $(\widetilde{\mathsf{pk}}, \tilde{\sigma})$ on the same message. Key randomizability implies, at a minimum, that $\widetilde{\mathsf{pk}}$ has the same distribution as an honestly generated public key; moreover, without the secret key corresponding to pk it should be infeasible to forge a valid signature on a new message under a randomized key $\widetilde{\mathsf{pk}}$ generated from pk using RandPK. *Strong* key randomizability additionally requires $\tilde{\sigma}$ to be distributed like an honestly generated signature. We refer to Section 2.1 for formal definitions.

Our core idea. There is an obvious approach for constructing issuer-hiding anonymous tokens based on zero-knowledge (ZK) proofs. Suppose a user holding (m, σ), issued by the issuer whose public key is pk_1, wants to convince a verifier that they hold a valid token with respect to some public key in a policy set $\mathcal{P} = \{\mathsf{pk}_1, \ldots, \mathsf{pk}_\ell\}$. The user can send m along with a ZK proof that

$$\exists \sigma : \mathsf{Vf}(\mathsf{pk}_1, m, \sigma) = 1 \vee \cdots \vee \mathsf{Vf}(\mathsf{pk}_\ell, m, \sigma) = 1.$$

The complexity of such a proof depends, in part, on the complexity of proving each clause of the disjunction, so our goal is to make each clause as simple to prove as possible. Proving that a signature verifies may be complex, in general, and may involve hash functions that are difficult to handle inside a ZK proof.

We observe that if the underlying signature scheme is strongly key-randomizable, better efficiency is possible. The user holding (m, σ) relative to pk_1 chooses randomness r, uses $\mathsf{RandPK}, \mathsf{RandSig}$ with randomness r to generate $\widetilde{\mathsf{pk}}, \tilde{\sigma}$, and then sends m, the randomized values $\widetilde{\mathsf{pk}}, \tilde{\sigma}$, and a zero-knowledge proof that

$$\exists r : \widetilde{\mathsf{pk}} = \mathsf{RandPK}(\mathsf{pk}_1; r) \vee \cdots \vee \widetilde{\mathsf{pk}} = \mathsf{RandPK}(\mathsf{pk}_\ell; r).$$

The verifier checks that $\mathsf{Vf}(\widetilde{\mathsf{pk}}, m, \tilde{\sigma}) = 1$ and verifies the zero-knowledge proof. The key observation is that $\widetilde{\mathsf{pk}}$ obscures the true issuer; this allows $\widetilde{\mathsf{pk}}, \tilde{\sigma}$ to be sent in the clear and signature verification to be done *outside* the ZK proof. The ZK proof itself thus reduces to an OR proof of, in many cases, simpler statements.

We can apply a similar idea to anonymous credentials. At an abstract level, and assuming for simplicity all attributes are disclosed, anonymous credentials allow a user to prove possession of a credential (i.e., a signature) σ on the vector of attributes $\boldsymbol{m}$ with respect to a public key pk by giving a ZK proof that

$$\exists \sigma : \mathsf{Vf}(\mathsf{pk}, \boldsymbol{m}, \sigma) = 1.$$

To support issuer hiding relative to a policy set $\mathcal{P} = \{\mathsf{pk}_1, \ldots, \mathsf{pk}_\ell\}$ that contains pk, a natural approach is to instead give a ZK proof that

$$\exists \sigma : \mathsf{Vf}(\mathsf{pk}_1, \boldsymbol{m}, \sigma) = 1 \vee \cdots \vee \mathsf{Vf}(\mathsf{pk}_\ell, \boldsymbol{m}, \sigma) = 1.$$

As before, we observe that if the underlying signature scheme supports key randomization, then better efficiency is possible. Specifically, the user chooses randomness r, uses RandPK, $\mathsf{RandSig}$ with randomness r to generate $\widetilde{\mathsf{pk}}, \tilde{\sigma}$, and then sends $\widetilde{\mathsf{pk}}$ along with a ZK proof that

$$\exists \tilde{\sigma}, r : \ \mathsf{Vf}(\widetilde{\mathsf{pk}}, \boldsymbol{m}, \tilde{\sigma}) = 1 \ \wedge \left(\widetilde{\mathsf{pk}} = \mathsf{RandPK}(\mathsf{pk}_1; r) \vee \cdots \vee \widetilde{\mathsf{pk}} = \mathsf{RandPK}(\mathsf{pk}_\ell; r)\right).$$

(Here we do not need *strong* key randomizability since $\tilde{\sigma}$ is not sent.) The complexity of proving the first clause of the conjunction is identical to the complexity of the original scheme, while the second clause of the conjunction is an OR proof over simpler statements than in the naive approach above.

Concrete constructions. We prove that both our generic construction can be instantiated with any anonymous token or credential scheme that admit key randomization: issuer-hiding privacy follows from key randomizability together with the zero-knowledge property of the OR proof, and issuer-hiding unforgeability reduces to the standard security of the underlying scheme. For issuer-hiding anonymous tokens, we build on the Tessaro–Zhu (TZ) blind signature scheme [58]. We choose this scheme because it avoids pairings (which have not yet been standardized and are difficult to deploy on some platforms), is efficient, and can be proven secure in a concurrent setting with an exponential concrete-security bound; for those reasons, it is also currently under consideration by the MoLE working group [48]. Establishing the requirements for TZ required special care, as the two properties we need—straight-line extractability of the ZK proof and one-more unforgeability of the blind signature—have each been proven independently in the AGM(+ROM), but do not compose out of the box.

For anonymous credentials, we focus on incorporating issuer hiding in existing BBS-based schemes [4,10,1,7,59], which are a leading candidate for standardization [39,46,56]. For this we formalize key randomizability of BBS signatures. This yields an efficient issuer-hiding AC scheme that is fully compatible with existing BBS-based credentials, and does not require any policy keys; a point we discuss further when we compare to prior work.

On post-quantum security. Both our concrete constructions are perfectly issuer-hiding. This means that even though they are not quantum-secure (since they are not unforgeable against a quantum adversary), they protect against "store now, de-anonymize later" attacks, an important consideration for potential deployment. Our generic results could be used to add issuer hiding to quantum-secure anonymous tokens and credentials given suitable key-randomizable signatures.

1.2 Related Work

We are not aware of prior work on issuer-hiding anonymous tokens in the sense we consider here. (Li et al. [45] consider a different notion of issuer hiding

where the issuer is hidden from the user during issuance.) On the other hand, multiple works [2,3,5,17,24,16,47,52,42,31,57] have shown constructions of issuer-hiding anonymous-credential schemes. One advantage of our work is that it gives an instantiation that is fully compatible with existing BBS-based anonymous credentials [4,10,1,7,59], for which several standards are currently under development [39,46,56]. Other recent works achieving this [42,31,40] require public *policy keys* bound to a particular policy set $\mathcal{P}$. In the Katz–Sefranek scheme [42] such policy keys are generated by each verifier along with an associated verification key that must be kept secret; in the schemes by Flamini et al. [31] and Kaaniche et al. [40] the policy keys must be generated by some trusted authority. In either case, requiring such policy keys complicates deployment [31, Section 7]. Another disadvantage of the aforementioned schemes is that they rely on an idealized group model, whether the generic group model (GGM) or algebraic group model (AGM). On the other hand, they allow for constant-size presentations, while in the best case ours have length logarithmic in the size of the policy set.

Kaaniche et al. [40] consider issuer-hiding ACs with *accountability*, whereby a trusted party can determine the issuer of a credential in case of abuse. They point out that some prior schemes allow a malicious issuer to "delegate" its capabilities to a colluding user and thus avoid being held accountable. We do not consider accountability in this work, and in fact our schemes are perfectly issuer hiding so accountability is impossible. Thus, we do not consider the attacks identified in their work to be relevant to the schemes we construct.

Signatures with randomizable keys have been used in settings similar to ours: Fleischhacker et al. [32] used them for unlinkable sanitizable signatures, and Derler and Slamanig [23] for ring signatures. In the latter, a signer likewise publishes a fresh-looking key along with a proof that it is a shift of one of the keys in a ring. Derler and Slamanig [23] also show that key homomorphism yields tight multi-user security, related to our result for BBS (Appendix B.1). However, their reduction requires key prefixing, whereas ours applies to BBS as standardized. In contrast to ring signatures, privacy in our IHAT setting must hold even towards the issuer that produced the token, and unforgeability must be captured in a one-more sense, which turned out to be the most subtle aspects of our analysis.

Paper History. An earlier version of this work focused solely on issuer-hiding BBS credentials without policy keys. This version gives a generic construction from any key-randomizable signature scheme, of which BBS is one instance. It also provides the first formal treatment of issuer-hiding anonymous tokens, with a generic construction and an instantiation based on Tessaro–Zhu blind signatures.

2 Preliminaries

We use ":=" for deterministic assignment, and "$\leftarrow$" for the output of a randomized algorithm or uniform selection from a set. We work in a concrete-security setting and thus do not have an explicit security parameter; our implementations target roughly 128-bit computational security. For any group $\mathbb{G}$, we let $\mathbb{G}^* = \mathbb{G} \setminus \{1\}$.

Zero-knowledge proofs of knowledge. We rely on non-interactive zero-knowledge proofs of knowledge (ZKPoKs) in the random-oracle model. For simplicity we assume perfect completeness and perfect zero knowledge; neither is strictly necessary. We also rely on two different notions of extraction: *straight-line extractability* (cf. [30]), meaning that a knowledge extractor can extract a witness from a proof output by an adversary without rewinding the adversary, and (tag-based) *simulation extractability* [22,35,28], meaning that a knowledge extractor can extract a witness from an adversary who outputs a new valid proof, even if that adversary was previously given multiple simulated proofs.

2.1 Key-Randomizable Signatures

Roughly speaking, a signature scheme is *key-randomizable* if there are algorithms RandPK, $\mathsf{RandSig}$ that transform a signature σ on a message m with respect to a public key pk into a signature $\tilde{\sigma}$ on the same message m with respect to a random public key $\widetilde{\mathsf{pk}}$. Moreover, this transformation is reversible given the randomness used. We remark that while several prior works have defined related notions [11,17,23], none of them exactly match what we need.

For a signature scheme $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Vf})$, we say a public key pk is *valid* if it is in the support of KeyGen, and assume validity can be checked efficiently.

Definition 1 (Key-Randomizable Signatures). *Signature scheme* $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Vf})$ *is* key-randomizable *if there are efficient algorithms* RandPK, $\mathsf{RandSig}$ *such that the following hold:*

- *For any valid* pk*, the following distributions are identical:*

$$\left\{\widetilde{\mathsf{pk}} \leftarrow \mathsf{RandPK}(\mathsf{pk}) : \widetilde{\mathsf{pk}}\right\} \quad \textit{and} \quad \left\{(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen} : \mathsf{pk}\right\}.$$

- *For any* pk, m, σ*, and randomness* r*, if* $\mathsf{Vf}(\mathsf{pk}, m, \sigma) = 1$ *and we compute* $\widetilde{\mathsf{pk}} := \mathsf{RandPK}(\mathsf{pk}; r)$ *and* $\tilde{\sigma} := \mathsf{RandSig}(\sigma, m; r)$*, then* $\mathsf{Vf}(\widetilde{\mathsf{pk}}, m, \tilde{\sigma}) = 1$.
- *There is an efficient algorithm* $\mathsf{RandSig}^{-1}$ *such that for any* $\mathsf{pk}, \widetilde{\mathsf{pk}}, m, \tilde{\sigma}, r$ *with*

$$\mathsf{Vf}(\widetilde{\mathsf{pk}}, m, \tilde{\sigma}) = 1 \quad \textit{and} \quad \widetilde{\mathsf{pk}} = \mathsf{RandPK}(\mathsf{pk}; r),$$

  it holds that $\mathsf{Vf}(\mathsf{pk}, m, \mathsf{RandSig}^{-1}(\tilde{\sigma}, m, r)) = 1$.

We write $\mathsf{Rand}(\mathsf{pk}, \sigma, m; r)$ for $(\mathsf{RandPK}(\mathsf{pk}; r), \mathsf{RandSig}(\sigma, m; r))$, noting that the same randomness r is shared by both underlying algorithms. Moreover, we omit the message m from $\mathsf{RandSig}$, $\mathsf{RandSig}^{-1}$, and Rand whenever they do not depend on it, which is the case for both of our instantiations.

Definition 1 only requires the randomized public key to be indistinguishable from a fresh key. We define a stronger notion that requires this for the signature as well. For a signature scheme $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Vf})$, let SK denote an (all-powerful) algorithm that on input pk samples a secret key sk with the same conditional distribution as the output of KeyGen, i.e., the following are identical:

$$\left\{(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen} : (\mathsf{pk}, \mathsf{sk})\right\} \quad \text{and} \quad \left\{(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen} : (\mathsf{pk}, \mathsf{SK}(\mathsf{pk}))\right\}.$$

Definition 2 (Strongly Key-Randomizable Signatures). *Key-randomizable signature scheme* $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Vf}, \mathsf{RandPK}, \mathsf{RandSig})$ *is* strongly key-randomizable *if the following stronger version of the first condition of Definition 1 holds: For any valid* pk *and any* m*, the following distributions are identical*

$$\left\{\mathsf{sk} \leftarrow \mathsf{SK}(\mathsf{pk}); \sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m); (\widetilde{\mathsf{pk}}, \tilde{\sigma}) \leftarrow \mathsf{Rand}(\mathsf{pk}, \sigma, m) : (\widetilde{\mathsf{pk}}, \tilde{\sigma})\right\}$$
$$\textit{and} \quad \left\{(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}; \sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m) : (\mathsf{pk}, \sigma)\right\}.$$

Note that strong key-randomizability does *not* require the randomized signature $\tilde{\sigma}$ to be indistinguishable from σ, which is required by the *perfect adaptability* notion in [23]. Such a stronger guarantee is neither needed by our constructions (for tokens, σ is unknown to the issuer by blindness; for credentials, $\tilde{\sigma}$ is only used as a witness in a zero-knowledge proof), nor achieved by our instantiations (where parts of the signature remain unchanged).

In this paper, we show that the BBS signature scheme [4,59] satisfies the property of key-randomizability in Definition 1, while the signature underlying the Tessaro–Zhu blind signature scheme BS_3 [58] satisfies the stronger notion of Definition 2. However, it is easy to see that many other common signature schemes are key-randomizable, for instance, BLS, with the randomization algorithms defined in [23, Lemma 4] and Schnorr, with algorithms described in [23, Lemma 2], if they are extended with the natural $\mathsf{RandSig}^{-1}$ algorithms.

3 Issuer-Hiding Anonymous Tokens

Section 3.1 recalls definitions for standard anonymous tokens. We formalize *issuer-hiding* anonymous tokens for the first time in Section 3.2. In Section 3.3 we show a generic way to add issuer hiding to any AT scheme whose underlying signature scheme is strongly key-randomizable, and in Section 3.4 we discuss a concrete instantiation based on Tessaro–Zhu blind signatures.

3.1 Anonymous Tokens

An anonymous-token scheme allows a user to obtain a token bound to a message; that message-token pair can later be presented to a third party. Our definitions of anonymous tokens are identical to standard blind signatures; we use the term "anonymous tokens" to parallel our later results on anonymous credentials.

Definition 3 (Anonymous Tokens). *An anonymous-token (AT) scheme consists of the following algorithms:*

Setup *generates public parameters* params*, implicitly given to all algorithms below.*
KeyGen *allows an issuer to generate a public/private key pair* $(\mathsf{pk}, \mathsf{sk})$*.*
$\langle \mathsf{Issue}(\mathsf{sk}), \mathsf{Obtain}(\mathsf{pk}, m) \rangle$ *is an interactive protocol run between an issuer with input* sk *and a user with input the issuer's public key* pk *and a message* m*, at the end of which the user outputs a token* σ *or* $\perp$*. We assume that if the output is not* $\perp$*, then* $\mathsf{Vf}(\mathsf{pk}, m, \sigma) = 1$ *(i.e., the user rejects invalid tokens).*

Vf *verifies a token* σ *on message* m *under public key* pk.

Correctness requires that for all m*, if we run* $\mathsf{params} \leftarrow \mathsf{Setup}$*,* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}$*, and* $\sigma \leftarrow \langle \mathsf{Issue}(\mathsf{sk}), \mathsf{Obtain}(\mathsf{pk}, m) \rangle$*, then* $\mathsf{Vf}(\mathsf{pk}, m, \sigma) = 1$ *(which implies* $\sigma \neq \perp$*).*

Any AT scheme naturally defines an underlying signature scheme, where the key-generation algorithm and verification algorithm are inherited from the AT scheme directly, and the signing algorithm $\mathsf{Sign}(\mathsf{sk}, m)$ is defined to output signatures with the same distribution obtained by running the honest issuance protocol $\langle \mathsf{Issue}(\mathsf{sk}), \mathsf{Obtain}(\mathsf{pk}, m) \rangle$ (where pk is the public key associated with sk).

One-more unforgeability. Unforgeability means an adversary cannot produce more valid tokens (on distinct messages) than the number of interactions of the issuance protocol it ran.

Definition 4 (One-More Unforgeability). *Consider the following experiment involving a stateful adversary* $\mathcal{A}$ *and anonymous-token scheme* Π*:*

1. *Set* $\mathsf{cnt} := 0$*. Run* $\mathsf{params} \leftarrow \mathsf{Setup}$ *and* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}$*.*
2. $\mathcal{A}(\mathsf{params}, \mathsf{pk})$ *may repeatedly act as the user in (concurrent) executions of the issuance protocol with an honest issuer holding* sk*. For each completed execution, set* $\mathsf{cnt} := \mathsf{cnt} + 1$*.*
3. $\mathcal{A}$ *outputs* k *message-token pairs* $\{(m_i, \sigma_i)\}_{i \in [k]}$*.*

$\mathcal{A}$ $\mathsf{succeeds}$ *if the following three conditions hold:*

(1) The $\{m_i\}$ *are distinct.*
(2) All tokens are valid, i.e., $\forall i : \mathsf{Vf}(\mathsf{pk}, m_i, \sigma_i) = 1$*.*
(3) k *surpasses the number of granted tokens, i.e.,* $k > \mathsf{cnt}$*.*

Let $\mathsf{Adv}^{\mathsf{omuf}}_{\mathcal{A},\Pi} \stackrel{\text{def}}{=} \Pr[\mathcal{A} \text{ succeeds}]$.

Blindness. Blindness ensures that the issuer cannot link an issuance session with a revealed message-token pair.

Definition 5 (Blindness). *Consider the following experiment involving a stateful adversary* $\mathcal{A}$ *and anonymous-token scheme* Π*:*

1. *Run* $\mathsf{params} \leftarrow \mathsf{Setup}$*.*
2. $\mathcal{A}(\mathsf{params})$ *outputs* m_0*,* m_1*, and* pk*. Choose* $b \leftarrow \{0, 1\}$*. Then* $\mathcal{A}$ *acts as the issuer in two (concurrent) executions of the issuance protocol with an honest user, where in the first execution the user has input* (pk, m_b) *and outputs* σ_b*, and in the second the user has input* (pk, m_{1-b}) *and outputs* σ_{1-b}*. If* $\sigma_0 = \perp$ *or* $\sigma_1 = \perp$*, set* $(\sigma_0, \sigma_1) := (\perp, \perp)$*.*
3. *Give* (σ_0, σ_1) *to* $\mathcal{A}$*, which outputs* b'*.*

$\mathcal{A}$ $\mathsf{succeeds}$ *if* $b' = b$*. Let* $\mathsf{Adv}^{\mathsf{blind}}_{\mathcal{A},\Pi} \stackrel{\text{def}}{=} |\Pr[\mathcal{A} \text{ succeeds}] - 1/2|$.

We say Π is *perfectly blind* if $\mathsf{Adv}^{\mathsf{blind}}_{\mathcal{A},\Pi} = 0$ for all (even all-powerful) adversaries $\mathcal{A}$.

3.2 Issuer-Hiding Anonymous Tokens (IHAT)

An issuer-hiding anonymous-token (IHAT) scheme allows a verifier to specify a policy set $\mathcal{P}$ of issuers (identified by their public keys). A user can then convince the verifier that it has a token issued by one of the issuers in $\mathcal{P}$, without revealing which one. We are interested in adding issuer hiding to existing AT schemes, so in particular an IHAT scheme inherits the algorithms of Definition 3.

Definition 6 (Issuer-Hiding Anonymous Tokens). *An IHAT scheme consists of the algorithms from Definition 3, plus the following two algorithms:*

$\mathsf{Show}(\mathcal{P}, \mathsf{pk}, m, \sigma)$ *outputs a presentation* π *of a token* σ *for* m *with respect to public key* $\mathsf{pk} \in \mathcal{P}$*, where* $\mathcal{P}$ *is a set of public keys.*
$\mathsf{VfShow}(\mathcal{P}, m, \pi)$ *verifies a presentation* π *of a message* m *w.r.t. a policy set* $\mathcal{P}$*.*

In addition to the correctness condition from Definition 3, we require that for any $\mathcal{P}, \mathsf{pk}, m, \sigma$ *for which* $\mathsf{pk} \in \mathcal{P}$ *and* $\mathsf{Vf}(\mathsf{pk}, m, \sigma) = 1$*, if we compute* $\pi \leftarrow \mathsf{Show}(\mathcal{P}, \mathsf{pk}, m, \sigma)$ *then* $\mathsf{VfShow}(\mathcal{P}, m, \pi) = 1$*.*

In contrast to the case of anonymous credentials (cf. Definition 11), we do not include a presentation context ctx here since tokens are intended to be used once and the inclusion of the nullifier in the presentation prevents token re-use.

Issuer-hiding one-more unforgeability. Unforgeability in the issuer-hiding case is more subtle, since not only can an adversary $\mathcal{A}$ obtain tokens from multiple issuers, but the adversary's presentations now involve sets of issuers rather than a single issuer. In particular, it does not suffice to require that the adversary cannot generate more successful presentations than the number of executions of the issuance protocols it runs, as this would allow (for example) an adversary to run a single execution with the issuer with public key pk but then output a successful presentation for a policy set that does not contain pk.

Intuitively, we want to ensure that each time $\mathcal{A}$ executes the issuance protocol with an issuer with public key pk, it obtains the ability to generate exactly one presentation for a policy set containing pk. We formalize this by requiring that every successful presentation by $\mathcal{A}$ for a policy set $\mathcal{P}$ can be mapped *injectively* to an execution of the issuance protocol with some issuer in $\mathcal{P}$.

Definition 7 (Issuer-Hiding One-More Unforgeability). *Consider the following experiment involving a stateful adversary* $\mathcal{A}$ *and IHAT scheme* Π*:*

1. *Set* $\ell := \mathsf{cnt} := 0$ *and run* $\mathsf{params} \leftarrow \mathsf{Setup}$*.*
2. $\mathcal{A}(\mathsf{params})$ *may repeatedly do the following:*
   - *query oracle* $\mathcal{O}\mathsf{KeyGen}$*, which on the* i*th such query generates a key pair* $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}$*, increments* ℓ*, and returns* pk_i *to* $\mathcal{A}$*.*
   - *act as the user in (concurrent) executions of the issuance protocol with an honest issuer holding* sk_i*, where* $i \in [\ell]$ *is chosen by* $\mathcal{A}$ *in each execution. For each completed execution, set* $\mathsf{cnt} := \mathsf{cnt} + 1$ *and let* $i_{\mathsf{cnt}} = i$ *be the index of the chosen key.*
3. $\mathcal{A}$ *outputs* k *tuples* $\{(\mathcal{P}_i, m_i, \pi_i)\}_{i \in [k]}$ *with* $\bigcup_{i \in [k]} \mathcal{P}_i \subseteq \{\mathsf{pk}_i\}_{i \in [\ell]}$*.*

$\mathcal{A}$ succeeds *if the following three conditions hold:*

(1) The $\{m_i\}$ *are distinct.*
(2) All presentations are valid, i.e., $\forall i : \mathsf{VfShow}(\mathcal{P}_i, m_i, \pi_i) = 1$.
(3) There is no injective map $f\colon [k] \to [\mathsf{cnt}]$ *such that* $\forall j \in [k] : \mathsf{pk}_{i_{f(j)}} \in \mathcal{P}_j$*, i.e., such that policy set* $\mathcal{P}_j$ *contains the public key* $\mathsf{pk}_{i_{f(j)}}$ *used in (completed) issuance protocol execution* $f(j)$.[1]

Let $\mathsf{Adv}^{\mathsf{ih\text{-}omuf}}_{\mathcal{A},\Pi} \stackrel{\text{def}}{=} \Pr[\mathcal{A} \text{ succeeds}]$.

Our definition does not consider it an attack if an adversary replays a valid presentation in a different context (including a changed policy set). One could incorporate an explicit context into the definition to prevent this, if desired.

Issuer-hiding blindness. Our notion of issuer-hiding blindness is patterned on the definition of blindness (Definition 5): we let an adversary choose two messages m_0, m_1 and then run two executions of the issuance protocol with an honest user, where in one execution (chosen at random) the user has input m_0 and in the other the user has input m_1. The adversary may use different public keys $\mathsf{pk}, \mathsf{pk}'$ (representing different issuers) in the two executions. The adversary can also specify two different policy sets $\mathcal{P}_0, \mathcal{P}_1$, so long as they each contain both pk and pk'; the user then presents the token it received on m_0 for $\mathcal{P}_0$, and the token it received on m_1 to $\mathcal{P}_1$. As in standard blindness, the adversary should be unable to determine which message was signed in which execution of the issuance protocol. Note that since the two executions may use different keys, this implies that the adversary cannot tell from a presentation the issuer from which a user obtained its token.

Our blindness definition is inspired by the privacy notion for *threshold* blind signatures [19,41,44]. There, a subset of signers jointly issues a blind signature that verifies under the group's common public key, and blindness requires that an adversary cannot determine from a signature which subset was involved in issuing the signature. Our experiment takes the same form as theirs, with the two public keys in place of the two signer sets.

Definition 8 (Issuer-Hiding Blindness). *Consider the following experiment involving a stateful adversary* $\mathcal{A}$ *and issuer-hiding anonymous-token scheme* Π*:*

1. *Run* $\mathsf{params} \leftarrow \mathsf{Setup}$.
2. $\mathcal{A}(\mathsf{params})$ *outputs* m_0, m_1 *and valid keys* $\mathsf{pk}, \mathsf{pk}'$*. Choose* $b \leftarrow \{0,1\}$*. Then* $\mathcal{A}$ *acts as the issuer in two (concurrent) executions of the issuance protocol with an honest user, where in the first execution the user has input* (pk, m_b) *and outputs* σ_b*, and in the second the user has input* (pk', m_{1-b}) *and outputs* σ_{1-b}*.*
3. $\mathcal{A}$ *outputs* $\mathcal{P}_0, \mathcal{P}_1$ *with* $\{\mathsf{pk}, \mathsf{pk}'\} \subseteq \mathcal{P}_0 \cap \mathcal{P}_1$*.*
4. *If* $\sigma_0 = \perp$ *or* $\sigma_1 = \perp$*, then give* $(\perp, \perp)$ *to* $\mathcal{A}$*. Otherwise, compute proofs* $\pi_b \leftarrow \mathsf{Show}(\mathcal{P}_b, \mathsf{pk}, m_b, \sigma_b)$ *and* $\pi_{1-b} \leftarrow \mathsf{Show}(\mathcal{P}_{1-b}, \mathsf{pk}', m_{1-b}, \sigma_{1-b})$*, and give* (π_0, π_1) *to* $\mathcal{A}$*.*

[1] We remark that this condition can be checked in polynomial time, e.g., by modeling it as a network-flow problem (maximum bipartite matching).

5. $\mathcal{A}$ *outputs* b'.

$\mathcal{A}$ succeeds *if* $b' = b$. *Let* $\mathsf{Adv}^{\text{ih-blind}}_{\mathcal{A},\Pi} \stackrel{\text{def}}{=} |\Pr[\mathcal{A} \text{ succeeds}] - 1/2|$.

We say Π is *perfectly issuer-hiding blind* if $\mathsf{Adv}^{\text{ih-blind}}_{\mathcal{A},\Pi} = 0$ for all adversaries $\mathcal{A}$.

We remark that issuer-hiding blindness does not imply blindness. Since our goal is to add issuer-hiding blindness to existing schemes (that already achieve blindness), we do not view this as a drawback of the definition.

3.3 A Generic IHAT Construction

We now show how to add issuer hiding to any AT scheme whose underlying signature scheme is strongly key-randomizable. Issuance is the same; only presentation of a token changes. As described in the Introduction, instead of giving a ZK proof that σ verifies under one of the keys $\mathsf{pk} \in \mathcal{P}$, the user randomizes (pk, σ) into $(\widetilde{\mathsf{pk}}, \tilde{\sigma})$ and sends them in the clear; this is safe since they are distributed like a fresh key/token pair. $(\widetilde{\mathsf{pk}}, \tilde{\sigma})$ can be checked with the unmodified Vf. The only ZK part is an OR proof that $\widetilde{\mathsf{pk}}$ is a randomization of some key in $\mathcal{P}$.

Definition 9 (Generic IHAT Scheme Π_{IHAT}). *Let* Π *be an AT scheme whose underlying signature scheme is strongly key randomizable via algorithms* RandPK, $\mathsf{RandSig}$ *(cf. Definition 2). IHAT scheme* Π_{IHAT} *adds the following algorithms:*

$\mathsf{Show}(\mathcal{P}, \mathsf{pk}, m, \sigma)$**:** *Parse* $\mathcal{P} = \{\mathsf{pk}_1, \ldots, \mathsf{pk}_\ell\}$. *Choose randomness* r *and compute* $(\widetilde{\mathsf{pk}}, \tilde{\sigma}) := \mathsf{Rand}(\mathsf{pk}, \sigma; r)$. *Compute a ZKPoK* π *for the statement*

$$\exists r : \widetilde{\mathsf{pk}} = \mathsf{RandPK}(\mathsf{pk}_1; r) \vee \cdots \vee \widetilde{\mathsf{pk}} = \mathsf{RandPK}(\mathsf{pk}_\ell; r),$$

and output the presentation $(\widetilde{\mathsf{pk}}, \tilde{\sigma}, \pi)$.

$\mathsf{VfShow}(\mathcal{P}, m, (\widetilde{\mathsf{pk}}, \tilde{\sigma}, \pi))$**:** *Output 1 iff* π *is valid and* $\mathsf{Vf}(\widetilde{\mathsf{pk}}, m, \tilde{\sigma}) = 1$.

Correctness is immediate. Next, we show that Π_{IHAT} is both issuer-hiding one-more unforgeable and blind.

Theorem 1 (Issuer-Hiding One-More Unforgeability of Π_{IHAT}). *Let* $\mathcal{A}_{\mathsf{IHAT}}$ *be an adversary making at most* ℓ *key-generation queries and participating in at most* I *executions of the issuance protocol. Then there is an adversary* $\mathcal{A}$ *running in roughly the same time as* $\mathcal{A}_{\mathsf{IHAT}}$ *and participating in at most* I *executions of the issuance protocol such that*

$$\mathsf{Adv}^{\text{ih-omuf}}_{\mathcal{A}_{\mathsf{IHAT}},\Pi_{\mathsf{IHAT}}} \leq \ell \cdot \mathsf{Adv}^{\text{omuf}}_{\mathcal{A},\Pi},$$

unless $\mathcal{A}$ *violates straight-line extractability of the ZKPoK.*

Proof. We construct an adversary $\mathcal{A}$ against AT scheme Π:

1. $\mathcal{A}$ is given params and pk. It chooses a uniform index $i^* \leftarrow \{1, \ldots, \ell\}$ and simulates the oracles for $\mathcal{A}_{\mathsf{IHAT}}(\mathsf{params})$ as follows:

- $\mathcal{O}\mathsf{KeyGen}$: On the ith query, if $i = i^*$, then $\mathcal{A}$ gives $\mathsf{pk}_{i^*} := \mathsf{pk}$ to $\mathcal{A}_{\mathsf{IHAT}}$. Otherwise, it runs $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}$ and gives pk_i to $\mathcal{A}_{\mathsf{IHAT}}$.
- When $\mathcal{A}_{\mathsf{IHAT}}$ runs the issuance protocol for the ith key pk_i, if $i = i^*$, then $\mathcal{A}$ forwards messages between $\mathcal{A}_{\mathsf{IHAT}}$ and its own issuance oracle. Otherwise, it runs the issuance protocol honestly using the key sk_i that it knows.

2. At the end of its execution, $\mathcal{A}_{\mathsf{IHAT}}$ outputs $\{(\mathcal{P}_j, m_j, (\widetilde{\mathsf{pk}}_j, \tilde{\sigma}_j, \pi_j))\}_{j\in[k]}$. If $\mathcal{A}_{\mathsf{IHAT}}$ succeeded, then do the following:
   - For $j = 1, \ldots, k$, extract[2] from π_j values r_j, i_j such that
   $$\widetilde{\mathsf{pk}}_j = \mathsf{RandPK}(\mathsf{pk}_{i_j}; r_j).$$
   If any extraction fails, abort. Call $\mathsf{pk}_{i_1}, \ldots, \mathsf{pk}_{i_k}$ the *used public keys*.
   - It follows from the fact that $\mathcal{A}_{\mathsf{IHAT}}$ succeeded that there must be at least one $i \in \{1, \ldots, \ell\}$ such that pk_i appears more often in the list of used public keys than the number of times $\mathcal{A}_{\mathsf{IHAT}}$ ran the issuance protocol using pk_i. If that property holds for i^*, continue; otherwise abort.
   - Output $\{(m_j, \mathsf{RandSig}^{-1}(\tilde{\sigma}_j, r_j))\}_{j\in[k]:\, i_j = i^*}$, where $\mathsf{RandSig}^{-1}$ is the algorithm guaranteed to exist by Definition 2.

$\mathcal{A}$ perfectly simulates the issuer-hiding unforgeability experiment for $\mathcal{A}_{\mathsf{IHAT}}$, and if $\mathcal{A}_{\mathsf{IHAT}}$ succeeds then with probability at least $1/\ell$ it holds that $\mathcal{A}$ does not abort (unless $\mathcal{A}_{\mathsf{IHAT}}$ violates knowledge soundness). The theorem follows. □

Theorem 2 (Issuer-Hiding Blindness of Π_{IHAT}). *Let Π be an AT scheme whose underlying signature scheme is strongly key-randomizable, and assume the ZKPoK used in the construction above satisfies perfect zero knowledge. If Π is perfectly blind, then Π_{IHAT} satisfies perfect issuer-hiding blindness.*

Proof. Fix an adversary $\mathcal{A}_{\mathsf{IHAT}}$ attacking issuer-hiding blindness of Π_{IHAT}. We consider a sequence of experiments $\mathbf{Expt}_0, \ldots, \mathbf{Expt}_5$ where, in each of them, we let $m_0, m_1, \mathsf{pk}, \mathsf{pk}'$ be the initial output of $\mathcal{A}_{\mathsf{IHAT}}(\mathsf{params})$ and choose $b \leftarrow \{0, 1\}$. We let p_i be the probability that $\mathcal{A}_{\mathsf{IHAT}}$ correctly guesses bit b in $\mathbf{Expt}_i$.

$\mathbf{Expt}_0$. Here, $\mathcal{A}_{\mathsf{IHAT}}$ acts as the issuer in two executions of the issuance protocol with an honest user, where in the first execution the user has input (pk, m_b) and outputs σ_b, and in the second the user has input (pk', m_{1-b}) and outputs σ_{1-b}. Following those executions, $\mathcal{A}_{\mathsf{IHAT}}$ outputs $\mathcal{P}_0, \mathcal{P}_1$. If $\perp \in \{\sigma_0, \sigma_1\}$ then give $(\perp, \perp)$ to $\mathcal{A}_{\mathsf{IHAT}}$; otherwise, compute

$$(\widetilde{\mathsf{pk}}_b, \tilde{\sigma}_b) \leftarrow \mathsf{Rand}(\mathsf{pk}, \sigma_b) \quad \text{and} \quad (\widetilde{\mathsf{pk}}_{1-b}, \tilde{\sigma}_{1-b}) \leftarrow \mathsf{Rand}(\mathsf{pk}', \sigma_{1-b}),$$

and give $\widetilde{\mathsf{pk}}_0, \tilde{\sigma}_0, \pi_0$ and $\widetilde{\mathsf{pk}}_1, \tilde{\sigma}_1, \pi_1$ to $\mathcal{A}_{\mathsf{IHAT}}$ where π_0, π_1 are ZKPoKs of the appropriate statements, namely, that $\widetilde{\mathsf{pk}}_0$ (resp., $\widetilde{\mathsf{pk}}_1$) is a randomization of a public key in $\mathcal{P}_0$ (resp., $\mathcal{P}_1$).

[2] Note that here we rely on straight-line extractability since we require extraction from multiple proofs.

Note that p_0 is exactly the probability that $\mathcal{A}_{\mathsf{IHAT}}$ succeeds in the issuer-hiding blindness experiment.

$\mathbf{Expt}_1$. In this experiment we replace σ_b with an honestly generated signature σ'_b. In detail: as before, $\mathcal{A}_{\mathsf{IHAT}}$ acts as the issuer in two executions of the issuance protocol, where in the first execution the user has input (pk, m_b) and outputs σ_b, and in the second the user has input (pk', m_{1-b}) and outputs σ_{1-b}. Following those executions, $\mathcal{A}_{\mathsf{IHAT}}$ outputs $\mathcal{P}_0, \mathcal{P}_1$. If $\perp \in \{\sigma_0, \sigma_1\}$ then give $(\perp, \perp)$ to $\mathcal{A}_{\mathsf{IHAT}}$; otherwise, compute $\mathsf{sk} \leftarrow \mathsf{SK}(\mathsf{pk})$ followed by $\sigma'_b \leftarrow \mathsf{Sign}(\mathsf{sk}, m_b)$, then compute

$$(\widetilde{\mathsf{pk}}_b, \tilde{\sigma}_b) \leftarrow \mathsf{Rand}(\mathsf{pk}, \sigma'_b) \quad \text{and} \quad (\widetilde{\mathsf{pk}}_{1-b}, \tilde{\sigma}_{1-b}) \leftarrow \mathsf{Rand}(\mathsf{pk}', \sigma_{1-b}),$$

and give $\widetilde{\mathsf{pk}}_0, \tilde{\sigma}_0, \pi_0$ and $\widetilde{\mathsf{pk}}_1, \tilde{\sigma}_1, \pi_1$ to $\mathcal{A}_{\mathsf{IHAT}}$, where π_0, π_1 are computed as before.

We show that $p_0 = p_1$ by relying on perfect blindness of Π. Define an adversary $\mathcal{A}$ as follows: $\mathcal{A}$ is given params and runs $\mathcal{A}_{\mathsf{IHAT}}(\mathsf{params})$ to obtain $m_0, m_1, \mathsf{pk}, \mathsf{pk}'$. It chooses $b \leftarrow \{0,1\}$ and outputs $\bar{m}_0 \coloneqq m_b, \bar{m}_1 \coloneqq m_b$, and pk in its own experiment. $\mathcal{A}$ then interacts with the honest user in two executions of the issuance protocol, where the user has input (pk, m_b) in both. In the first execution with the honest user, $\mathcal{A}$ forwards messages back-and-forth to the first execution of $\mathcal{A}_{\mathsf{IHAT}}$. In the second execution with the honest user, $\mathcal{A}$ samples $\mathsf{sk} \leftarrow \mathsf{SK}(\mathsf{pk})$ and behaves as an honest issuer using sk. For the second execution with $\mathcal{A}_{\mathsf{IHAT}}$, we simply have $\mathcal{A}$ behave as an honest user with input (pk', m_{1-b}); let σ_{1-b} be the output of that execution. $\mathcal{A}$ is then given signatures $(\bar{\sigma}_0, \bar{\sigma}_1)$ in its own experiment. If $(\bar{\sigma}_0, \bar{\sigma}_1) = (\perp, \perp)$ or if $\sigma_{1-b} = \perp$, then give $(\perp, \perp)$ to $\mathcal{A}_{\mathsf{IHAT}}$. Otherwise, compute

$$(\widetilde{\mathsf{pk}}_b, \tilde{\sigma}_b) \leftarrow \mathsf{Rand}(\mathsf{pk}, \bar{\sigma}_0) \quad \text{and} \quad (\widetilde{\mathsf{pk}}_{1-b}, \tilde{\sigma}_{1-b}) \leftarrow \mathsf{Rand}(\mathsf{pk}', \sigma_{1-b}),$$

and give $\widetilde{\mathsf{pk}}_0, \tilde{\sigma}_0, \pi_0$ and $\widetilde{\mathsf{pk}}_1, \tilde{\sigma}_1, \pi_1$ to $\mathcal{A}_{\mathsf{IHAT}}$ (where π_0, π_1 are computed as before). Finally, $\mathcal{A}$ outputs 1 iff $b' = b$, where b' is the output of $\mathcal{A}_{\mathsf{IHAT}}$.

Let $\bar{b}$ be the random bit chosen in $\mathcal{A}$'s experiment that $\mathcal{A}$ is trying to guess. The signatures $(\bar{\sigma}_0, \bar{\sigma}_1)$ given to $\mathcal{A}$ are both signatures on m_b under pk (assuming $(\bar{\sigma}_0, \bar{\sigma}_1) \neq (\perp, \perp)$), where one was computed by interacting with $\mathcal{A}_{\mathsf{IHAT}}$ and the other was computed honestly. Thus, when $\bar{b} = 0$ the view of $\mathcal{A}_{\mathsf{IHAT}}$ is identical to its view in $\mathbf{Expt}_0$, whereas if $\bar{b} = 1$ the view of $\mathcal{A}_{\mathsf{IHAT}}$ is identical to its view in $\mathbf{Expt}_1$. It follows that $|p_1 - p_0| \leq \mathsf{Adv}^{\mathsf{blind}}_{\mathcal{A},\Pi} = 0$, since Π is perfectly blind.

$\mathbf{Expt}_2$. Here, we modify the experiment in an analogous way by replacing σ_{1-b} with an honestly generated signature σ'_{1-b}. As above, we have $p_2 = p_1$.

$\mathbf{Expt}_3$. This experiment is identical to the previous one except that the ZK proofs π_0, π_1 are both simulated. Assuming perfect zero knowledge, we have $p_3 = p_2$.

$\mathbf{Expt}_4$. In this experiment we replace each invocation of Rand by an honestly generated public key and signature. In detail: as in the previous experiment, $\mathcal{A}_{\mathsf{IHAT}}$ acts as the issuer in two executions of the issuance protocol, where in the first execution the user has input (pk, m_b) and outputs σ_b, and in the second

the user has input (pk', m_{1-b}) and outputs σ_{1-b}. Then $\mathcal{A}_{\mathsf{IHAT}}$ outputs $\mathcal{P}_0, \mathcal{P}_1$. If $\perp \in \{\sigma_0, \sigma_1\}$ then $\mathcal{A}_{\mathsf{IHAT}}$ is given $(\perp, \perp)$. Otherwise, for $i \in \{0,1\}$, we compute

$$(\widetilde{\mathsf{pk}}_i, \widetilde{\mathsf{sk}}_i) \leftarrow \mathsf{KeyGen} \quad \text{and} \quad \tilde{\sigma}_i \leftarrow \mathsf{Sign}(\widetilde{\mathsf{sk}}_i, m_i),$$

and give $\widetilde{\mathsf{pk}}_0, \tilde{\sigma}_0, \pi_0$ and $\widetilde{\mathsf{pk}}_1, \tilde{\sigma}_1, \pi_1$ to $\mathcal{A}_{\mathsf{IHAT}}$ (where π_0, π_1 are simulated). Strong key randomizability of the underlying signature scheme implies $p_4 = p_3$.

$\mathbf{Expt}_5$. We now change the message used in one of the executions with $\mathcal{A}_{\mathsf{IHAT}}$ so that both use the same message. Specifically, $\mathcal{A}_{\mathsf{IHAT}}$ runs two executions of the issuance protocol, where in the first execution the user has input (pk, m_0) and outputs σ_b, and in the second the user has input (pk', m_0) and outputs σ_{1-b}. Then $\mathcal{A}_{\mathsf{IHAT}}$ outputs $\mathcal{P}_0, \mathcal{P}_1$. If $\perp \in \{\sigma_0, \sigma_1\}$ then $\mathcal{A}_{\mathsf{IHAT}}$ is given $(\perp, \perp)$. Otherwise, for $i \in \{0,1\}$ we compute $(\widetilde{\mathsf{pk}}_i, \widetilde{\mathsf{sk}}_i) \leftarrow \mathsf{KeyGen}$ followed by $\tilde{\sigma}_i \leftarrow \mathsf{Sign}(\widetilde{\mathsf{sk}}_i, m_i)$, and give $\widetilde{\mathsf{pk}}_0, \tilde{\sigma}_0, \pi_0$ and $\widetilde{\mathsf{pk}}_1, \tilde{\sigma}_1, \pi_1$ to $\mathcal{A}_{\mathsf{IHAT}}$ (where π_0, π_1 are simulated). Perfect blindness of Π immediately implies that $p_5 = p_4$.

Finally, note that the view of $\mathcal{A}_{\mathsf{IHAT}}$ in $\mathbf{Expt}_5$ is independent of b and so $p_5 = 1/2$. This completes the proof. □

3.4 A Concrete Instantiation: IHAT-TZ

Our generic construction can be instantiated with any blind signature scheme whose underlying signature scheme is strongly key randomizable. Here we describe an instantiation using the Tessaro–Zhu blind signature scheme BS_3 [58]. We choose this scheme because it is currently under consideration by the MoLE working group [54]; it is efficient, avoids pairings, and can be proven concurrently secure with an exponential concrete-security bound based on the discrete-logarithm assumption in the algebraic-group and random-oracle models. It is also perfectly blind. We show that the underlying signature scheme is strongly key-randomizable.

The BS_3 scheme. Since our goal here is to show that the underlying signature scheme of BS_3 is strongly key-randomizable, we describe the underlying scheme only and refer to the work of Tessaro–Zhu for a description of the blind-issuance protocol. Let $\mathbb{G}$ be a group of prime order p with generator g, and let $H\colon \mathbb{G} \times \mathbb{G} \times \{0,1\}^* \to \mathbb{Z}_p^*$ be a hash function modeled as a random oracle.

Definition 10 (Signature Scheme Underlying BS_3). *The signature scheme underlying BS_3 consists of the following algorithms:*

Setup**:** *Choose $Z \leftarrow \mathbb{G}^*$ and output $\mathsf{params} := Z$.*[3]
KeyGen**:** *Choose $x \leftarrow \mathbb{Z}_p$ and output $\mathsf{sk} := x$ and $\mathsf{pk} := X = g^x$.*
$\mathsf{Sign}(\mathsf{sk}, m)$**:** *Choose $a, t \leftarrow \mathbb{Z}_p$ and $y \leftarrow \mathbb{Z}_p^*$. Set $c := H(g^a, g^t Z^y, m)$ and $s := a + cyx$, and output $\sigma := (c, s, y, t)$.*
$\mathsf{Vf}(\mathsf{pk}, m, (c, s, y, t))$**:** *Output 1 iff $y \neq 0$ and $H(g^s X^{-cy}, g^t Z^y, m) = c$.*

[3] Tessaro and Zhu let each signer choose Z and include it in its public key. It is easy to see that all signers can share the same Z.

We show that this signature scheme is strongly key-randomizable by defining appropriate algorithms RandPK, $\mathsf{RandSig}$, and $\mathsf{RandSig}^{-1}$.

Key randomization of $\mathbf{BS}_3$. A public key is randomized by an additive shift of the secret key. Because the challenge c does not depend on the public key, only the response s has to be adapted:

$\mathsf{RandPK}(X; r)$**:** Output $\tilde{X} := X \cdot g^r$.
$\mathsf{RandSig}((c, s, y, t); r)$**:** Output $(c, \tilde{s}, y, t)$ with $\tilde{s} := s + rcy$.
$\mathsf{RandSig}^{-1}((c, \tilde{s}, y, t), r)$**:** Output (c, s, y, t) with $s := \tilde{s} - rcy$.

Lemma 1. *The signature scheme underlying BS_3 is strongly key randomizable via* RandPK*,* $\mathsf{RandSig}$*, and* $\mathsf{RandSig}^{-1}$ *defined above.*

Proof. We verify the required conditions of Definitions 1 and 2:

Strong key randomizability (Definition 2). Fix some $\mathsf{pk} = X$ and message m. It is immediate that the randomized public key $\widetilde{\mathsf{pk}} = \tilde{X} := X \cdot g^r$ is distributed exactly like a fresh public key, with corresponding secret key $\tilde{x} = \log_g \tilde{X} = r + \log_g X$. Moreover, the randomization $(c, \tilde{s}, y, t)$ of a fresh signature (c, s, y, t) can be viewed as being generated by choosing uniform $a, t \in \mathbb{Z}_p$ and $y \in \mathbb{Z}_p^*$, computing $c := H(g^a, g^t Z^y, m)$, and setting $\tilde{s} = a + cy\tilde{x}$, and thus has the same distribution as a fresh signature on m using secret key $\tilde{x}$.

Second and third condition of Definition 1. Fix any $\mathsf{pk} = X$, m, $\sigma = (c, s, y, t)$, for which $\mathsf{Vf}(\mathsf{pk}, m, \sigma) = 1$. It is easy to check that for any r and corresponding $\widetilde{\mathsf{pk}} = X \cdot g^r$ and $\tilde{\sigma} = (c, s + rcy, y, t)$ we have $\mathsf{Vf}(\widetilde{\mathsf{pk}}, m, \tilde{\sigma}) = 1$.

In an analogous way, fix any r and corresponding $\widetilde{\mathsf{pk}} = X \cdot g^r$, and let $\tilde{\sigma} = (c, \tilde{s}, y, t)$ be any token with $\mathsf{Vf}(\widetilde{\mathsf{pk}}, m, \tilde{\sigma}) = 1$. Since $\mathsf{RandSig}^{-1}(\tilde{\sigma}, r) = (c, \tilde{s} - rcy, y, t)$ and $g^{\tilde{s}-rcy} \cdot X^{-cy} = g^{\tilde{s}} \cdot (Xg^r)^{-cy} = g^{\tilde{s}} \cdot \widetilde{\mathsf{pk}}^{-cy}$, the verification equation is the same in both cases, and hence $\mathsf{Vf}(\mathsf{pk}, m, \mathsf{RandSig}^{-1}(\tilde{\sigma}, r)) = 1$. □

Description of the IHAT scheme. For completeness, we describe the Show and VfShow algorithms we obtain from the above that give an IHAT scheme based on BS_3, which we call IHAT-TZ:

$\mathsf{Show}(\mathcal{P}, \mathsf{pk}, m, \sigma)$**:** Let $\mathcal{P} = \{\mathsf{pk}_1, \ldots, \mathsf{pk}_\ell\}$ and $\sigma = (c, s, y, t)$. Choose $r \leftarrow \mathbb{Z}_p$ and set $\widetilde{\mathsf{pk}} := \mathsf{pk} \cdot g^r$ and $\tilde{\sigma} := (c, s + rcy, y, t)$. The presentation π includes $\widetilde{\mathsf{pk}}$, $\tilde{\sigma}$, and a ZKPoK that

$$\exists r : g^r = \widetilde{\mathsf{pk}}/\mathsf{pk}_1 \vee \cdots \vee g^r = \widetilde{\mathsf{pk}}/\mathsf{pk}_\ell.$$

$\mathsf{VfShow}(\mathcal{P}, m, \pi)$**:** Parse π as $\widetilde{\mathsf{pk}}$, $\tilde{\sigma} = (c, s, y, t)$, and a ZKPoK of the statement above. Accept iff $y \neq 0$, $H\big(g^s \widetilde{\mathsf{pk}}^{-cy}, g^t Z^y, m\big) = c$ (where Z is from the public parameters), and the ZKPoK verifies.

There are various options for the ZKPoK; we defer discussion to Section 5.1. In fact, for the classical OR proofs considered there, proving straight-line extractability turned out to be non-trivial and had to be adapted to our setting. While straight-line extractability of the basic Schnorr ZKPoK was shown in the AGM+ROM by Fuchsbauer and Orrù [33], and subsequently extended to Σ-protocols for conjunctions of linear relations over groups by Orrù [49], neither result covers disjunctions. Moreover, these results do not directly compose with the one-more unforgeability of BS_3 [58] (also proven in the AGM+ROM). Since security proofs in the AGM are relative to the group elements an adversary sees, we show in Appendix C that the OR ZKPoK is straight-line extractable in the context of our IHAT-TZ scheme.

4 Issuer-Hiding Anonymous Credentials

We show that key randomization can also be useful for anonymous credentials, where it yields a generic way to add issuer hiding that is more efficient than a naive OR proof over full credential presentations. In Section 4.1, we recall anonymous credentials and define their issuer-hiding variant. In Section 4.2 we show a generic way to add issuer hiding to any AC scheme whose underlying signature scheme is key randomizable, and in Section 4.3 we give a concrete instantiation using BBS-based anonymous credentials.

4.1 Definitions for (Issuer Hiding) Anonymous Credentials

An anonymous-credential scheme enables a user to obtain a credential σ on a vector $\boldsymbol{m}$ of n attributes; the user can later reveal a subset $(m_i)_{i\in\mathcal{D}}$ (for $\mathcal{D}\subseteq[n]$) of those attributes to a verifier and prove that it possesses a valid credential on a vector including those attributes. Those proofs can incorporate a *context* ctx that, among other things, can be used to prevent replay attacks.

Definition 11 (Anonymous Credentials). *An anonymous-credential (AC) scheme (for n attributes) consists of the following algorithms:*

Setup *generates public parameters* params*, implicitly given to all algorithms below.*
KeyGen *allows an issuer to generate a public/private key pair* $(\mathsf{pk},\mathsf{sk})$*.*
$\mathsf{Issue}(\mathsf{sk},\boldsymbol{m})$ *creates a credential σ for vector of attributes $\boldsymbol{m}$ using secret key* sk*.*
$\mathsf{VfCred}(\mathsf{pk},\boldsymbol{m},\sigma)$ *verifies credential σ on $\boldsymbol{m}$ under* pk*.*
$\mathsf{Show}(\mathsf{pk},\mathcal{D},\boldsymbol{m},\mathsf{ctx},\sigma)$ *generates a proof π for credential σ on attributes $\boldsymbol{m}$ in context* ctx*, disclosing only $(m_i)_{i\in\mathcal{D}}$.*
$\mathsf{VfShow}(\mathsf{pk},\mathcal{D},(m_i)_{i\in\mathcal{D}},\mathsf{ctx},\pi)$ *verifies that π is a valid presentation of $(m_i)_{i\in\mathcal{D}}$ with respect to* pk *in context* ctx*.*

Correctness requires that for all $\boldsymbol{m}=(m_i)_{i\in[n]}$, $\mathcal{D}\subseteq[n]$, and $\mathsf{ctx}\in\{0,1\}^*$, if we run $\mathsf{params}\leftarrow\mathsf{Setup}$*,* $(\mathsf{pk},\mathsf{sk})\leftarrow\mathsf{KeyGen}$*, and* $\sigma\leftarrow\mathsf{Issue}(\mathsf{sk},\boldsymbol{m})$*, followed by* $\pi\leftarrow\mathsf{Show}(\mathsf{pk},\mathcal{D},\boldsymbol{m},\mathsf{ctx},\sigma)$ *then*

$$\mathsf{VfCred}(\mathsf{pk},\boldsymbol{m},\sigma)=1 \quad \textit{and} \quad \mathsf{VfShow}(\mathsf{pk},\mathcal{D},(m_i)_{i\in\mathcal{D}},\mathsf{ctx},\pi)=1.$$

AC schemes must be *unforgeable*, i.e., users cannot present attributes for which they hold no credential, and *anonymous*, i.e., presentations are unlinkable and reveal only the disclosed attributes (and the issuer), even to colluding issuers and verifiers. We refer to Appendix A for formal definitions.

Issuer hiding. Standard anonymity of anonymous credentials does not hide the issuer, as VfShow takes the issuer's public key pk as input. An issuer-hiding anonymous-credential (IHAC) scheme additionally hides this information. It allows a verifier to specify a policy set $\mathcal{P}$ of issuers; a user can then prove possession of an appropriate credential issued by one of the issuers in $\mathcal{P}$, without revealing which one. Our definitions are taken almost verbatim from the work of Katz and Sefranek [42], except that we do not have policy keys and can therefore simplify the syntax. They also coincide with the definitions of Flamini et al. [31] for issuer hiding without policy keys (called Type-0 IHAC there).

Definition 12 (Issuer-Hiding Anonymous Credentials). *An issuer-hiding anonymous-credential scheme consists of algorithms* Setup, KeyGen, Issue, *and* VfCred *as in Definition 11, with* Show *and* VfShow *extended as follows:*[4]

$\mathsf{Show}(\mathcal{P}, \mathsf{pk}, \mathcal{D}, \boldsymbol{m}, \mathsf{ctx}, \sigma)$ *generates a proof* π *for credential* σ *on attributes* $\boldsymbol{m}$ *under* $\mathsf{pk} \in \mathcal{P}$ *in context* ctx*, disclosing only* $(m_i)_{i\in\mathcal{D}}$ *and the policy set* $\mathcal{P}$*.*

$\mathsf{VfShow}(\mathcal{P}, \mathcal{D}, (m_i)_{i\in\mathcal{D}}, \mathsf{ctx}, \pi)$ *verifies that* π *is a valid presentation of* $(m_i)_{i\in\mathcal{D}}$ *with respect to some key in* $\mathcal{P}$ *in context* ctx*.*

In addition to the correctness condition from Definition 11, we require that for all $\boldsymbol{m} = (m_i)_{i\in[n]}$*, all* params *output by* Setup*, any* $\{(\mathsf{pk}_i, \mathsf{sk}_i)\}_{i\in[\ell]}$ *output by* KeyGen*, any* $\mathcal{D} \subseteq [n]$*, any* ctx*, any* $j \in [\ell]$*, any* σ *with* $\mathsf{VfCred}(\mathsf{pk}_j, \boldsymbol{m}, \sigma) = 1$*, and all* π *output by* $\mathsf{Show}(\{\mathsf{pk}_i\}_{i\in[\ell]}, \mathsf{pk}_j, \mathcal{D}, \boldsymbol{m}, \mathsf{ctx}, \sigma)$*, we have*

$$\mathsf{VfShow}(\{\mathsf{pk}_i\}_{i\in[\ell]}, \mathcal{D}, (m_i)_{i\in\mathcal{D}}, \mathsf{ctx}, \pi) = 1.$$

Issuer-hiding unforgeability. Issuer-hiding unforgeability ensures that no adversary can generate a valid proof for disclosed attributes $(m_i)_{i\in\mathcal{D}}$ under policy set $\mathcal{P}$ in context ctx, unless it has obtained a credential on matching attributes from an issuer in $\mathcal{P}$, or replays a previous proof for the same policy set, disclosed attributes, and context. In the issuer-hiding setting, presentations are made with respect to a policy set, so the game involves multiple issuers. The adversary gets a KeyGen oracle to obtain honestly generated keys, and must produce a forgery under a policy set that exclusively contains such keys.

Definition 13 (Issuer-Hiding Unforgeability). *Consider the following experiment involving a stateful adversary* $\mathcal{A}$ *and scheme* Π*:*

1. *Run* $\mathsf{params} \leftarrow \mathsf{Setup}$*.*
2. $\mathcal{A}(\mathsf{params})$ *may repeatedly query the following oracles:*
   - $\mathcal{O}\mathsf{KeyGen}$*, which on the* i*th such query computes* $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}$ *and returns* pk_i *to* $\mathcal{A}$*.*

[4] We assume $|\mathcal{P}| > 1$; when $|\mathcal{P}| = 1$ the syntax from Definition 11 can be used.

- $\mathcal{O}\mathsf{Issue}$, *which on input* $(j, \boldsymbol{m})$ *returns* $\mathsf{Issue}(\mathsf{sk}_j, \boldsymbol{m})$, *where* sk_j *is the secret key generated in the* j*th query to* $\mathcal{O}\mathsf{KeyGen}$.
- $\mathcal{O}\mathsf{Issue}'$, *which on the* i*th such query* $(j, \boldsymbol{m}_i)$ *computes* $\sigma_i \leftarrow \mathsf{Issue}(\mathsf{sk}_j, \boldsymbol{m}_i)$, *but returns nothing to* $\mathcal{A}$. *We say* σ_i *is* associated with pk_j *in this case.*
- $\mathcal{O}\mathsf{Show}$, *which on input* $(i, \mathcal{P}, \mathcal{D}, \mathsf{ctx})$, *where* σ_i *is the credential generated in the* i*th query to* $\mathcal{O}\mathsf{Issue}'$ *and* σ_i *is associated with some* $\mathsf{pk}_j \in \mathcal{P}$, *returns* $\mathsf{Show}(\mathcal{P}, \mathsf{pk}_j, \mathcal{D}, \boldsymbol{m}_i, \mathsf{ctx}, \sigma_i)$.

3. $\mathcal{A}$ *outputs* $\mathcal{P}^* \subseteq \{\mathsf{pk}_i\}$, $\mathcal{D}^* \subseteq [n]$, $(m_i^*)_{i \in \mathcal{D}^*}$, ctx^*, *and* π.

$\mathcal{A}$ succeeds *if*

(1) $\mathsf{VfShow}(\mathcal{P}^*, \mathcal{D}^*, (m_i^*)_{i \in \mathcal{D}^*}, \mathsf{ctx}^*, \pi) = 1$,
(2) $\mathcal{A}$ *never made a query* $\mathcal{O}\mathsf{Issue}(j, \boldsymbol{m}')$ *with* $\mathsf{pk}_j \in \mathcal{P}^*$ *and* $(m_i')_{i \in \mathcal{D}^*} = (m_i^*)_{i \in \mathcal{D}^*}$.
(3) $\mathcal{A}$ *never made a query* $\mathcal{O}\mathsf{Show}(i, \mathcal{P}^*, \mathcal{D}^*, \mathsf{ctx}^*)$ *where* σ_i *is associated with a key in* $\mathcal{P}^*$, *and* $(m_{i,j})_{j \in \mathcal{D}^*} = (m_j^*)_{j \in \mathcal{D}^*}$.

Let $\mathsf{Adv}^{\mathsf{ih\text{-}ac\text{-}uf}}_{\mathcal{A},\Pi} \stackrel{\text{def}}{=} \Pr[\mathcal{A} \text{ succeeds}]$.

Issuer-hiding anonymity. Issuer-hiding anonymity requires that proofs reveal the minimum information necessary. Specifically, a proof only reveals the disclosed attributes, is unlinkable to the credential used to generate it or to other proofs generated using that credential (beyond whatever linkability is implied by the disclosed attributes), and hides which issuer in $\mathcal{P}$ issued that credential. This holds even if verifiers and issuers are malicious and colluding.

Definition 14 (Issuer-Hiding Anonymity). *Consider the following experiment involving a stateful adversary* $\mathcal{A}$ *and scheme* Π*:*

1. *Run* $\mathsf{params} \leftarrow \mathsf{Setup}$.
2. $\mathcal{A}(\mathsf{params})$ *outputs* $(\mathcal{P}, \mathcal{D}, \mathsf{ctx}, \boldsymbol{m}_0, \sigma_0, \mathsf{pk}_0, \boldsymbol{m}_1, \sigma_1, \mathsf{pk}_1)$ *where:*
   - $\mathsf{pk}_0, \mathsf{pk}_1 \in \mathcal{P}$,
   - $\mathsf{VfCred}(\mathsf{pk}_0, \boldsymbol{m}_0, \sigma_0) = \mathsf{VfCred}(\mathsf{pk}_1, \boldsymbol{m}_1, \sigma_1) = 1$, *and*
   - $m_{0,i} = m_{1,i}$ *for all* $i \in \mathcal{D}$.
3. *Choose* $b \leftarrow \{0,1\}$, *and set* $\pi \leftarrow \mathsf{Show}(\mathcal{P}, \mathsf{pk}_b, \mathcal{D}, \boldsymbol{m}_b, \mathsf{ctx}, \sigma_b)$.
4. $\mathcal{A}(\pi)$ *outputs a bit* b'.

$\mathcal{A}$ succeeds *if* $b' = b$. *Let* $\mathsf{Adv}^{\mathsf{ih\text{-}anon}}_{\mathcal{A},\Pi} \stackrel{\text{def}}{=} |\Pr[\mathcal{A} \text{ succeeds}] - 1/2|$.

Π is *perfectly issuer-hiding anonymous* if $\mathsf{Adv}^{\mathsf{ih\text{-}anon}}_{\mathcal{A},\Pi} = 0$ for all adversaries $\mathcal{A}$.

The sign-then-prove paradigm. Most anonymous-credential schemes follow the sign-then-prove paradigm introduced by Camenisch and Lysyanskaya [9,10]. That is, a credential is simply a signature on the attribute vector (i.e., $\mathsf{Issue}(\mathsf{sk}, \boldsymbol{m})$ outputs a signature σ on $\boldsymbol{m}$), and a presentation $\mathsf{Show}(\mathsf{pk}, \mathcal{D}, \boldsymbol{m}, \mathsf{ctx}, \sigma)$ is a simulation-extractable ZKPoK of such a signature:

$$\exists (m_i)_{i \notin \mathcal{D}}, \sigma : \mathsf{Vf}(\mathsf{pk}, \boldsymbol{m}, \sigma) = 1.$$

(The disclosed attributes $(m_i)_{i\in\mathcal{D}}$ are part of the statement.) The ZKPoK incorporates a tag that includes ctx, to ensure that the proof cannot be used in a different context. Examples of anonymous-credential schemes following this paradigm include [9,50,4,10,59,1,7]. Throughout this paper, we only consider AC schemes that follow the sign-then-prove paradigm.

We let $\mathsf{Adv}^{\mathsf{uf}}_{\mathcal{A},\Sigma}$ denote the advantage of an adversary $\mathcal{A}$ attacking the underlying signature scheme Σ in the standard sense of existential unforgeability under chosen-message attacks. Unforgeability of Π using the sign-then-prove paradigm follows from the unforgeability of Σ and the simulation-extractability of the ZKPoK.

4.2 A Generic IHAC Construction

We now show how to add issuer hiding to any AC scheme that follows the sign-then-prove paradigm and whose underlying signature scheme is key-randomizable. The approach is the same as for our IHAT construction, i.e. the user randomizes (pk,σ) into $(\widetilde{\mathsf{pk}},\tilde{\sigma})$, and proves in an OR proof that $\widetilde{\mathsf{pk}}$ is a randomization of some key in $\mathcal{P}$. The only difference is that $\tilde{\sigma}$ is not sent in the clear, but used as the witness to a ZKPoK-based presentation with respect to $\widetilde{\mathsf{pk}}$. Since $\tilde{\sigma}$ is never revealed, here plain (rather than strong) key randomizability suffices.

Our scheme improves upon the naive approach [3], which requires an OR proof over ℓ full presentations, one for each key in $\mathcal{P}$. Instead, we need only a single presentation of the randomized signature, together with an OR proof over key randomizations. This leads to a more efficient instantiation whenever proving that $\widetilde{\mathsf{pk}}$ is a randomization of pk_i is cheaper than proving possession of a signature under pk_i. We show in Section 4.3 that this is the case for BBS.

Definition 15 (Generic IHAC Scheme Π_{IHAC}). *Let $\Sigma = (\mathsf{Setup},\mathsf{KeyGen},\mathsf{Sign},\mathsf{Vf})$ be a signature scheme that is key-randomizable via algorithms RandPK, $\mathsf{RandSig}$. IHAC scheme Π_{IHAC} uses the Setup and KeyGen algorithms of Σ, sets $\mathsf{Issue}(\mathsf{sk},\boldsymbol{m}) \coloneqq \mathsf{Sign}(\mathsf{sk},\boldsymbol{m})$ and $\mathsf{VfCred}(\mathsf{pk},\boldsymbol{m},\sigma) \coloneqq \mathsf{Vf}(\mathsf{pk},\boldsymbol{m},\sigma)$, and defines:*

$\mathsf{Show}(\mathcal{P},\mathsf{pk},\mathcal{D},\boldsymbol{m},\mathsf{ctx},\sigma)$**:** *Parse $\mathcal{P} = \{\mathsf{pk}_1,\ldots,\mathsf{pk}_\ell\}$. Choose randomness r and compute $(\widetilde{\mathsf{pk}},\tilde{\sigma}) \coloneqq \mathsf{Rand}(\mathsf{pk},\sigma;r)$. Compute a ZKPoK π (with context ctx) for the statement*

$$\exists (m_i)_{i\notin\mathcal{D}},\tilde{\sigma},r : \mathsf{Vf}(\widetilde{\mathsf{pk}},\boldsymbol{m},\tilde{\sigma}) = 1 \\ \wedge \left(\widetilde{\mathsf{pk}} = \mathsf{RandPK}(\mathsf{pk}_1;r) \vee \cdots \vee \widetilde{\mathsf{pk}} = \mathsf{RandPK}(\mathsf{pk}_\ell;r)\right), \quad (1)$$

and output the presentation $(\widetilde{\mathsf{pk}},\pi)$.

$\mathsf{VfShow}(\mathcal{P},\mathcal{D},(m_i)_{i\in\mathcal{D}},\mathsf{ctx},(\widetilde{\mathsf{pk}},\pi))$**:** *Output 1 iff π is a valid proof (with context ctx) for statement (1) with respect to $\mathcal{P}$, $\mathcal{D}$, $(m_i)_{i\in\mathcal{D}}$, and $\widetilde{\mathsf{pk}}$.*

Correctness is immediate. Note that Π_{IHAC} is exactly the sign-then-prove AC scheme for Σ, except that the ZKPoK is computed with respect to the randomized key $\widetilde{\mathsf{pk}}$ and extended by an OR clause over $\mathcal{P}$. In particular, the first clause can be proven with the same (efficient) protocol as in the standard AC scheme.

Theorem 3 (Issuer-Hiding Unforgeability of Π_{IHAC}). *Let $\mathcal{A}_{\mathsf{IHAC}}$ be an adversary making at most ℓ key-generation queries. Then there is an adversary $\mathcal{A}$ running in roughly the same time as $\mathcal{A}_{\mathsf{IHAC}}$ such that*

$$\mathsf{Adv}^{\mathsf{ih\text{-}ac\text{-}uf}}_{\mathcal{A}_{\mathsf{IHAC}},\Pi_{\mathsf{IHAC}}} \leq \ell \cdot \mathsf{Adv}^{\mathsf{uf}}_{\mathcal{A},\Sigma}\,,$$

unless $\mathcal{A}$ violates simulation extractability of the ZKPoK.

Proof. Let $\mathcal{A}_{\mathsf{IHAC}}$ be an adversary against the issuer-hiding scheme, and let ℓ be an upper bound on the number of $\mathcal{O}\mathsf{KeyGen}$ queries it makes. We construct the following adversary $\mathcal{A}$ against the unforgeability of Σ:

1. $\mathcal{A}$ is given params and pk. It chooses a uniform index $i^* \leftarrow \{1,\dots,\ell\}$ and simulates the oracles for $\mathcal{A}_{\mathsf{IHAC}}(\mathsf{params})$ as follows:
   - $\mathcal{O}\mathsf{KeyGen}$: On the ith query, if $i = i^*$, then $\mathcal{A}$ gives $\mathsf{pk}_{i^*} := \mathsf{pk}$ to $\mathcal{A}_{\mathsf{IHAC}}$. Otherwise, it runs $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}$ and gives pk_i to $\mathcal{A}_{\mathsf{IHAC}}$.
   - $\mathcal{O}\mathsf{Issue}(j, \boldsymbol{m})$: If $j = i^*$, then $\mathcal{A}$ queries its own signing oracle to obtain a signature σ on $\boldsymbol{m}$, and returns σ to $\mathcal{A}_{\mathsf{IHAC}}$. Otherwise, it returns the honestly computed $\sigma \leftarrow \mathsf{Issue}(\mathsf{sk}_j, \boldsymbol{m})$ to $\mathcal{A}_{\mathsf{IHAC}}$.
   - $\mathcal{O}\mathsf{Issue}'(j, \boldsymbol{m}_i)$: On the ith query, if $j = i^*$, then $\mathcal{A}$ does nothing. Otherwise, it honestly computes $\sigma_i \leftarrow \mathsf{Issue}(\mathsf{sk}_j, \boldsymbol{m}_i)$).
   - $\mathcal{O}\mathsf{Show}(i, \mathcal{P}, \mathcal{D}, \mathsf{ctx})$: Let j be the index of the associated public key. If $j = i^*$, then $\mathcal{A}$ returns $\widetilde{\mathsf{pk}} \leftarrow \mathsf{RandPK}(\mathsf{pk})$ and a simulated proof (with context ctx) of statement (1) to $\mathcal{A}_{\mathsf{IHAC}}$. Otherwise, it returns an honestly computed proof $\mathsf{Show}(\mathcal{P}, \mathsf{pk}_j, \mathcal{D}, \boldsymbol{m}_i, \mathsf{ctx}, \sigma_i)$ to $\mathcal{A}_{\mathsf{IHAC}}$.
2. At the end of its execution, $\mathcal{A}_{\mathsf{IHAC}}$ outputs $\mathcal{P}^*, \mathcal{D}^*, (m_i^*)_{i\in\mathcal{D}^*}, \mathsf{ctx}^*, \widetilde{\mathsf{pk}}^*$, and a proof (with context ctx^*) of statement (1). If $\mathcal{A}_{\mathsf{IHAC}}$ succeeded, extract values $(m_i^*)_{i\notin\mathcal{D}^*}, \tilde{\sigma}^*, r$, and i' such that $\mathsf{Vf}(\widetilde{\mathsf{pk}}^*, \boldsymbol{m}^*, \tilde{\sigma}^*) = 1$ for $\boldsymbol{m}^* = (m_i^*)_{i\in[n]}$, and $\widetilde{\mathsf{pk}}^* = \mathsf{RandPK}(\mathsf{pk}_{i'}; r)$ with $\mathsf{pk}_{i'} \in \mathcal{P}^*$. If extraction fails or $i' \neq i^*$, abort. Otherwise, compute $\sigma^* := \mathsf{RandSig}^{-1}(\tilde{\sigma}^*, r)$ and output $(\boldsymbol{m}^*, \sigma^*)$.

Note that $\mathcal{A}$ provides a perfect simulation of the issuer-hiding unforgeability experiment for $\mathcal{A}_{\mathsf{IHAC}}$, and if $\mathcal{A}_{\mathsf{IHAC}}$ succeeds then with probability $1/\ell$ it holds that $\mathcal{A}$ does not abort (unless $\mathcal{A}_{\mathsf{IHAC}}$ violates simulation extractability). Moreover, since $\mathcal{A}_{\mathsf{IHAC}}$ never made a query $\mathcal{O}\mathsf{Issue}(j, \boldsymbol{m}')$ with $\mathsf{pk}_j \in \mathcal{P}^*$ and $(m_i')_{i\in\mathcal{D}^*} = (m_i^*)_{i\in\mathcal{D}^*}$, and in particular never made such a query with $j = i^*$, it follows that $\mathcal{A}$ never queried $\boldsymbol{m}^*$ to its own signing oracle, and thus $\mathcal{A}$ succeeds in forging a signature on a previously unsigned message. The theorem follows. □

Remark 1. It is possible to obtain a tighter reduction by reducing to *multi-signer* unforgeability of the underlying signature scheme Σ. In Appendix B.2 we prove this for the BBS-based instantiation we consider in the next section.

Theorem 4 (Issuer-Hiding Anonymity of Π_{IHAC}). *If the signature scheme Σ is key randomizable and the ZKPoK for statement (1) is perfect zero knowledge, Π_{IHAC} is perfectly issuer-hiding anonymous.*

Proof. Fix an adversary $\mathcal{A}_{\mathsf{IHAC}}$ attacking issuer-hiding anonymity. We consider a sequence of experiments where, in each, we let $(\mathcal{P}, \mathcal{D}, \mathsf{ctx}, \boldsymbol{m}_0, \sigma_0, \mathsf{pk}_0, \boldsymbol{m}_1, \sigma_1, \mathsf{pk}_1)$ be the initial output of $\mathcal{A}_{\mathsf{IHAC}}(\mathsf{params})$ and choose $b \leftarrow \{0,1\}$. Let p_i be the probability that $\mathcal{A}_{\mathsf{IHAC}}$ correctly guesses bit b in $\mathbf{Expt}_i$.

$\mathbf{Expt}_0$. Here, we compute $(\widetilde{\mathsf{pk}}, \tilde{\sigma}) \leftarrow \mathsf{Rand}(\mathsf{pk}_b, \sigma_b)$ and give $\widetilde{\mathsf{pk}}$ and an honestly generated proof of statement (1) to $\mathcal{A}_{\mathsf{IHAC}}$. Note that p_0 is exactly the probability that $\mathcal{A}_{\mathsf{IHAC}}$ succeeds in the issuer-hiding anonymity experiment.

$\mathbf{Expt}_1$. In this experiment we replace the honestly generated proof with a simulated proof. It follows immediately from perfect zero knowledge that $p_1 = p_0$.

$\mathbf{Expt}_2$. Here we compute $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}$ and give pk and a simulated proof to $\mathcal{A}_{\mathsf{IHAC}}$. Key randomizability of Σ implies that $p_2 = p_1$. Since the view of $\mathcal{A}_{\mathsf{IHAC}}$ in $\mathbf{Expt}_2$ is independent of b, we have $p_2 = 1/2$. This completes the proof. □

4.3 A Concrete Instantiation: IHAC-BBS

Our generic construction can be instantiated with any AC scheme that follows the sign-then-prove paradigm and builds on a key-randomizable signature scheme. We describe an instantiation using BBS-based anonymous credentials, which are undergoing standardization [46,56] and have been identified as the most promising candidate for a ZKP-based EUDI Wallet solution [26]. Several recent works [42,31,40] show how to incorporate issuer hiding into existing BBS-based credentials; as discussed in Section 1.2, however, those schemes all require policy keys, which our construction does not require. To apply our generic construction, we formalize that BBS signatures are key-randomizable.

BBS-based ACs. BBS-based anonymous credentials [10,7,59] follow the sign-then-prove paradigm using the BBS signature scheme [4], which we now describe. Let $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ be groups of prime order p with a bilinear map $\mathsf{e}\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$.

Definition 16 (BBS Signatures [4]). *The BBS signature scheme for n attributes $\boldsymbol{m} = (m_i)_{i\in[n]} \in \mathbb{Z}_p^n$ consists of the following algorithms:*

Setup: *Choose $g_1 \leftarrow \mathbb{G}_1^*$, $g_2 \leftarrow \mathbb{G}_2^*$, $h_1, \ldots, h_n \leftarrow \mathbb{G}_1^*$, and output* $\mathsf{params} := (g_1, g_2, h_1, \ldots, h_n)$.
KeyGen: *Choose $x \leftarrow \mathbb{Z}_p$ and output $\mathsf{sk} := x$ and $\mathsf{pk} := g_2^x$ along with*[5] $\mathsf{pk}' := g_1^x$.
Sign$(\mathsf{sk}, \boldsymbol{m})$: *Choose $e \leftarrow \mathbb{Z}_p \setminus \{-x\}$ and output $\sigma := (A, e)$ with $A := C(\boldsymbol{m})^{1/(x+e)}$ for $C(\boldsymbol{m}) \stackrel{\text{def}}{=} g_1 \cdot \prod_{i\in[n]} h_i^{m_i}$.*
Vf$(\mathsf{pk}, \boldsymbol{m}, (A, e))$: *Output 1 iff $\mathsf{e}(A, \mathsf{pk}) = \mathsf{e}(A^{-e} \cdot C(\boldsymbol{m}), g_2)$.*

[5] We include pk' for technical reasons. It is not needed for signature verification, but is used for simulation. It can also be used to improve efficiency as we show in Section 5.2. Including pk' in the public key does not affect security (existing security proofs can be easily adapted to take it into account [59,13,12]), and correctness of pk' can be easily verified by checking that $\mathsf{e}(g_1, \mathsf{pk}) \stackrel{?}{=} \mathsf{e}(\mathsf{pk}', g_2)$.

Key randomizability has been observed informally by [31]. We formalize this insight by defining appropriate algorithms RandPK, $\mathsf{RandSig}$, and $\mathsf{RandSig}^{-1}$. As for BS_3, a public key is randomized by an additive shift of the secret key. Since a signature only depends on $x+e$, it suffices to shift e in the opposite direction, while A remains unchanged:

$\mathsf{RandPK}((\mathsf{pk}, \mathsf{pk}'); r)$**:** Output $\widetilde{\mathsf{pk}} \coloneqq g_2^r \cdot \mathsf{pk}$ along with $\widetilde{\mathsf{pk}}' \coloneqq g_1^r \cdot \mathsf{pk}'$.
$\mathsf{RandSig}((A, e); r)$**:** Output $(A, \tilde{e})$ with $\tilde{e} \coloneqq e - r$.
$\mathsf{RandSig}^{-1}((A, \tilde{e}), r)$**:** Output (A, e) with $e \coloneqq \tilde{e} + r$.

Lemma 2. *The BBS signature scheme is key randomizable via* RandPK, $\mathsf{RandSig}$, *and* $\mathsf{RandSig}^{-1}$ *defined above.*

Proof. We verify the required conditions of Definition 1:

First condition. Fix some $\mathsf{pk} = g_2^x$ (with $\mathsf{pk}' = g_1^x$). It is immediate that $(\widetilde{\mathsf{pk}}, \widetilde{\mathsf{pk}}') = (g_2^r \cdot \mathsf{pk}, g_1^r \cdot \mathsf{pk}')$, where r is uniform, is distributed exactly like a fresh public key, with corresponding secret key $\tilde{x} = x + r$.

Second condition. Fix any $\mathsf{pk} = g_2^x$, $\boldsymbol{m}$, and $\sigma = (A, e)$ for which $\mathsf{Vf}(\mathsf{pk}, \boldsymbol{m}, \sigma) = 1$, i.e., $A^{x+e} = C(\boldsymbol{m})$. It is easy to check that for any r and corresponding $\widetilde{\mathsf{pk}} = g_2^r \cdot \mathsf{pk}$ and $\tilde{\sigma} = (A, e - r)$, we have $\mathsf{Vf}(\widetilde{\mathsf{pk}}, \boldsymbol{m}, \tilde{\sigma}) = 1$, since $(x + r) + (e - r) = x + e$.

Third condition. Fix any $\mathsf{pk} = g_2^x$, any r and corresponding $\widetilde{\mathsf{pk}} = g_2^r \cdot \mathsf{pk} = g_2^{x+r}$, any $\boldsymbol{m}$, and *any* $\tilde{\sigma} = (A, \tilde{e})$ with $\mathsf{Vf}(\widetilde{\mathsf{pk}}, \boldsymbol{m}, \tilde{\sigma}) = 1$, i.e., $A^{(x+r)+\tilde{e}} = C(\boldsymbol{m})$. Since $\mathsf{RandSig}^{-1}(\tilde{\sigma}, r) = (A, \tilde{e} + r)$ and $x + (\tilde{e} + r) = (x + r) + \tilde{e}$, it follows that $\mathsf{Vf}(\mathsf{pk}, \boldsymbol{m}, \mathsf{RandSig}^{-1}(\tilde{\sigma}, r)) = 1$. □

Description of the IHAC scheme. For completeness, we describe the Show and VfShow algorithms we obtain from the above that give an IHAC scheme for BBS-based anonymous credentials, which we call IHAC-BBS:

$\mathsf{Show}(\mathcal{P}, \mathsf{pk}, \mathcal{D}, \boldsymbol{m}, \mathsf{ctx}, (A, e))$**:** Let $\mathcal{P} = \{\mathsf{pk}_i\}_{i \in [\ell]}$. Choose $r \leftarrow \mathbb{Z}_p$, and set $\widetilde{\mathsf{pk}} \coloneqq g_2^r \cdot \mathsf{pk}$ and $\tilde{e} \coloneqq e - r$. Then choose $t \leftarrow \mathbb{Z}_p^*$ and compute $A' \coloneqq A^t$ and $\bar{A} \coloneqq (A')^{-\tilde{e}} \cdot C(\boldsymbol{m})^t$. The proof consists of $\widetilde{\mathsf{pk}}$, A', and $\bar{A}$, along with a ZKPoK (with context ctx) that

$$\exists (m_i)_{i \notin \mathcal{D}}, \tilde{e}, t, r : \bar{A} = (A')^{-\tilde{e}} \cdot C(\boldsymbol{m})^t \\ \wedge \left(g_2^r = \widetilde{\mathsf{pk}}/\mathsf{pk}_1 \vee \cdots \vee g_2^r = \widetilde{\mathsf{pk}}/\mathsf{pk}_\ell \right). \tag{2}$$

$\mathsf{VfShow}(\mathcal{P}, \mathcal{D}, (m_i)_{i \in \mathcal{D}}, \mathsf{ctx}, \pi)$**:** Parse π as $\widetilde{\mathsf{pk}}, A', \bar{A}$, and a ZKPoK of the statement above. Accept iff (1) $A' \neq 1$, (2) the ZKPoK (with context ctx) is valid, and (3) $\mathsf{e}\big(A', \widetilde{\mathsf{pk}}\big) = \mathsf{e}(\bar{A}, g_2)$.

Note that while Show reveals values derived from the BBS signature (A' and $\bar{A}$), this does not conflict with our result that key randomizability suffices: both are part of the ZKPoK of the BBS signature and are thus covered by its perfect zero-knowledge property.

There are various options for both the first clause of the ZKPoK (see the work of Tessaro and Zhu [59]) as well as for the disjunctive clause (see Section 5.1).

We can prove tighter concrete-security bounds for issuer-hiding unforgeability of our BBS-based IHAC scheme than what is implied by Theorem 3, and in particular can avoid the factor of ℓ loss. We provide the details in Appendix B.

Comparison to a naive construction. For completeness, we compare IHAC-BBS with the naive way of adding issuer hiding to BBS-based credentials, as sketched in Section 1.1.

Recall that a standard (issuer-revealing) BBS presentation consists of A', $\bar{A}$, and a ZKPoK of the first clause of statement (2) (with e in place of $\tilde{e}$), and is verified by checking that $\mathsf{e}(A', \mathsf{pk}) = \mathsf{e}(\bar{A}, g_2)$. Since this check reveals pk, the naive approach to issuer hiding [3] must include such a pair $(A'_i, \bar{A}_i)$ for every $\mathsf{pk}_i \in \mathcal{P}$. For the public key pk_{i^*} for which (A, e) is a valid signature, the user can generate $(A'_{i^*}, \bar{A}_{i^*})$ as above; for $i \neq i^*$ the user can simulate the required elements by choosing uniform s_i and computing $A'_i := g_1^{s_i}$ and $\bar{A}_i := (\mathsf{pk}'_i)^{s_i}$. The proof consists of $\{(A'_i, \bar{A}_i)\}_{i\in[\ell]}$ and a ZKPoK that

$$\exists (m_i)_{i\notin\mathcal{D}}, e, t : \bar{A}_1 = (A'_1)^{-e} \cdot C(\boldsymbol{m})^t \ \vee \ \cdots \ \vee \ \bar{A}_\ell = (A'_\ell)^{-e} \cdot C(\boldsymbol{m})^t,$$

which the verifier checks along all ℓ pairing equations $\mathsf{e}(A'_i, \mathsf{pk}_i) \stackrel{?}{=} \mathsf{e}(\bar{A}_i, g_2)$.

Hence, every clause of the naive disjunction is a full BBS presentation. In contrast, IHAC-BBS reveals the randomized key $\widetilde{\mathsf{pk}}$, makes a single BBS presentation under $\widetilde{\mathsf{pk}}$, and adds an OR proof of knowledge of a discrete logarithm with respect to one of the keys in $\mathcal{P}$. Compared to a standard BBS presentation, the overhead of IHAC-BBS is therefore one element of $\mathbb{G}_2$ and roughly 2ℓ elements of $\mathbb{Z}_p$, independent of the number of hidden attributes. The naive approach instead requires $2(\ell-1)$ additional elements of $\mathbb{G}_1$ and $(\ell-1)(n-|\mathcal{D}|+3)$ additional elements of $\mathbb{Z}_p$. For $\ell = 27$ issuers (e.g., all EU member states) and $n - |\mathcal{D}| = 20$ hidden attributes, this is 1.9 kB versus 24.1 kB on BLS12-381 (without point compression). Moreover, since the 2ℓ elements $(A'_i, \bar{A}_i)$ must be sent in the clear, the naive scheme remains linear in ℓ even with logarithmic-size OR proofs, whereas IHAC-BBS can directly benefit from such proofs. The size gap above widens with the number of hidden attributes, as only the naive scheme's overhead grows with $n - |\mathcal{D}|$; the concrete performance comparison in Section 5.3 uses fewer hidden attributes for simplicity and is thus conservative.

5 Implementation and Experimental Results

We implemented various versions of our IHAT-TZ (cf. Section 3.4) and IHAC-BBS (cf. Section 4.3) schemes. Both schemes rely on 1-out-of-ℓ disjunctive ZK proofs of knowledge of a discrete logarithm, and we discuss different instantiations of that proof in Section 5.1. In Section 5.2 we discuss implementation details and optimizations. Our experimental results are presented in Section 5.3.

5.1 Instantiating the 1-out-of-ℓ Disjunctive Proof

Fix group elements $g, y_1, \ldots, y_\ell \in \mathbb{G}$. Both our schemes rely on a ZKPoK that

$$\exists r : \bigvee_{i=1}^{\ell} (g^r = y_i). \tag{3}$$

While a proof of a single clause can be handled using the standard Schnorr Σ-protocol, there are different ways to handle the proof of the 1-out-of-ℓ disjunction. Below, we describe the variants we implemented. All the variants can be shown to be simulation-extractable using the techniques of Faust et al. [28]. Some of the variants can also be shown to be straight-line extractable (we point these out below); the others give only a heuristic instantiation.

Standard. The most straightforward way of constructing a proof for statement (3) is to use the classical method for handling disjunctions [18] based on Σ-protocols, along with the Fiat–Shamir transform to make the proof non-interactive. The resulting proof is of size $\Theta(\ell)$.

We consider two well-known variants of the naive approach. Recall that, abstractly, the Fiat–Shamir transform has the prover generate a first message A, hash[6] that message to obtain a challenge $c := H(A)$, and then generate a corresponding response z. In our context, A is a list of $\Theta(\ell)$ group elements, c is a single scalar, and z is a list of $\Theta(\ell)$ scalars. The first variant of the naive approach, which we call **Std-s(hort)**, includes c and z; the second variant, which we call **Std-f(ast)** includes A and z. Verification of Std-s proofs requires $\Theta(\ell)$ exponentiations, whereas randomized verification of Std-f proofs can be done with just a single multi-exponentiation.

In Appendix C we show that Std-f and Std-s are straight-line extractable when applied to IHAT-TZ.

Stacking Sigmas. The stacking Sigmas (**SΣ**) approach [34] allows proving a disjunction of ℓ statements with *logarithmic* communication. We refer to the original paper for a description of the approach, and only note that it relies on a 1-of-ℓ commitment scheme (we use the variant from [34, Appendix G]) that can be based on any group $\mathbb{G}$ in which the discrete-logarithm problem is hard as well as a public random permutation on $\mathbb{G}$. We defer further discussion to Section 5.2.

Groth–Kohlweiss (GK). Groth and Kohlweiss [36] show a technique that can be used to give logarithmic-length ZKPoKs for statement (3). Their work is presented as a proof that one-out-of-n commitments is a commitment to 0, but since g^r can be viewed as a Pedersen commitment to 0 (with respect to an arbitrary other base), we can adapt their proof to our context. We again refer to the original paper for details.

5.2 Implementation Details and Optimizations

All implementations were in Go with the native heap implementation and bignum arithmetic, and share the same code for group operations including fast multi-exponentiation algorithms. We use SHA-256 for the Fiat–Shamir transform.

[6] We omit for simplicity additional information included in the hash.

For IHAT-TZ we use Ristretto-255 [37] as the underlying group. Our implementation of IHAC-BBS uses the BLS12-381 curve [6] without point compression, as implemented in CIRCL [29]. For the first clause in statement (2) we use the optimized ZKPoK of Tessaro and Zhu [59, Section 5.2].

We always implement the 1-of-ℓ commitment scheme of the SΣ approach using the Ristretto-255 group (even for IHAC-BBS).[7] We implement a public random permutation P on the Ristretto-255 curve using an 8-round Feistel network based on SHA-256, which is indifferentiable from a random permutation if SHA-256 is modeled as a random oracle [20]. The time to evaluate P was negligible compared to group operations, so does not significantly affect our experimental results.

Optimizations for IHAC-BBS. As described, IHAC-BBS requires a proof involving elements of $\mathbb{G}_2$ (cf. statement (2)). We show that it is possible to shift the statement being proved from $\mathbb{G}_2$ to $\mathbb{G}_1$. This significantly improves both proof size and proving/verification time since elements of $\mathbb{G}_1$ have shorter representations and group operations in $\mathbb{G}_1$ are faster.

Recall that issuers include $\mathsf{pk}' = g_1^x$ as part of their public keys. We can thus modify presentations as follows: users compute $\widetilde{\mathsf{pk}}, A'$, and $\bar{A}$ as described in Section 4.3, and now additionally compute $\widetilde{\mathsf{pk}}' = g_1^r \cdot \mathsf{pk}'$. They then generate a ZKPoK (with context ctx) that

$$\exists (m_i)_{i \notin \mathcal{D}}, \tilde{e}, t, r : \bar{A} = (A')^{-\tilde{e}} \cdot C(\boldsymbol{m})^t \wedge \left(\bigvee_i g_1^r = \widetilde{\mathsf{pk}}' / \mathsf{pk}'_i \right).$$

Verification is done as before, with the additional check that $\mathsf{e}(\widetilde{\mathsf{pk}}', g_2) \stackrel{?}{=} \mathsf{e}(g_1, \widetilde{\mathsf{pk}})$. In fact, by using randomization we can combine this last check with the existing check that $\mathsf{e}(A', \widetilde{\mathsf{pk}}) \stackrel{?}{=} \mathsf{e}(\bar{A}, g_2)$ so as to avoid computing additional pairings.

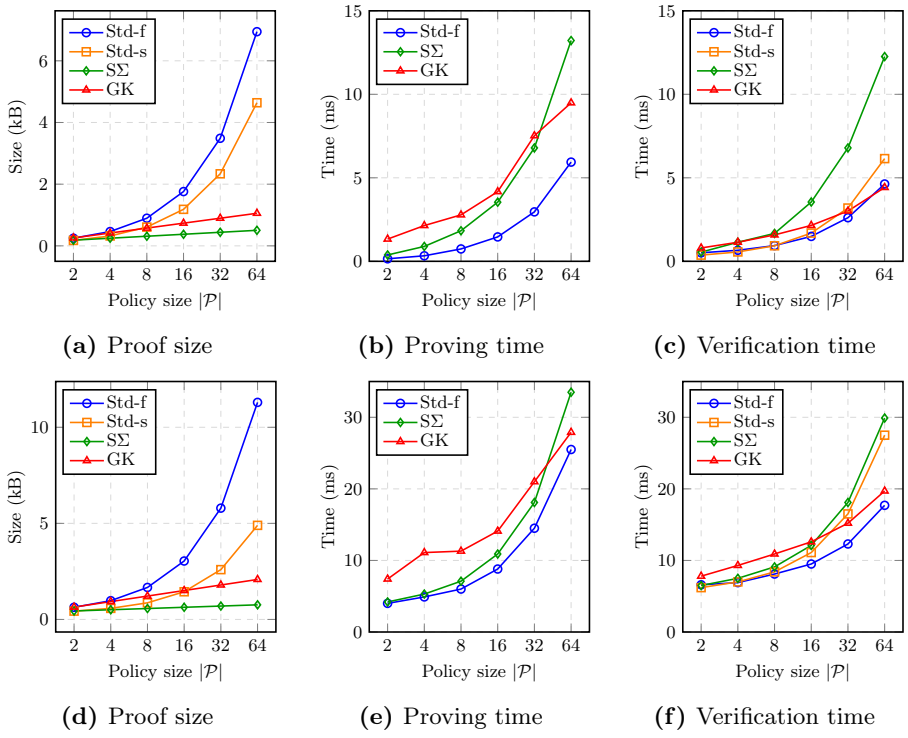
The cost of this shift is an additional $\mathbb{G}_1$ element in proofs, one additional exponentiation during proving, and two during verification (needed for the randomized check mentioned above). Even for moderate ℓ the tradeoff is worthwhile.

5.3 Experimental Results

All experiments were done on an Apple M1 via the Go benchmarking system. Standard deviations were below 10% for all reported values. For each of IHAT-TZ and IHAC-BBS we implement variants with each of the four ZKPoKs described in Section 5.1; we also implement the naive IHAC scheme described in Section 4.3 using both the Std-s and SΣ proof variants. (It is not compatible with the GK variant.) All IHAC experiments use a credential with $n = 3$ attributes where presentations disclose one of them.

Our results for IHAT-TZ and IHAC-BBS are given in Figure 1. (The proving times of Std-f and Std-s are almost identical in both cases, so we only include the proving times of Std-f.) We make the following observations:

[7] Group elements of the Ristretto-255 curve are much shorter (32 bytes) than $\mathbb{G}_1$ elements of BLS12-381 (96 bytes without point compression).



(a) Proof size **(b)** Proving time **(c)** Verification time

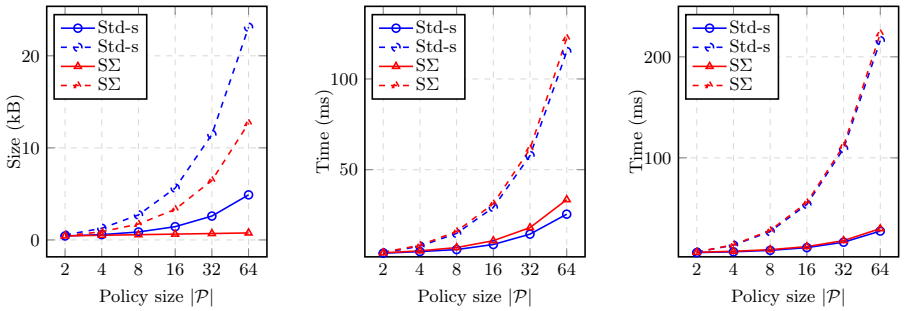
(d) Proof size **(e)** Proving time **(f)** Verification time

Fig. 1. Experimental results for IHAT-TZ (top row) and IHAC-BBS (bottom row). The x-axis in all cases is on a logarithmic scale.

- Our schemes have good performance for policy sets of up to $\ell = 64$ issuers. With the exception of Std-f, all variants have presentations under 5 kB. All variants of IHAT-TZ have proving time/verification times under 15 ms; variants of IHAC-BBS have proving/verification times under 35 ms.
- It is immediately apparent that the proof sizes of Std-f and Std-s grow linearly in ℓ (exponentially in $\log \ell$), whereas the proof sizes of SΣ and GK grow logarithmically in ℓ (linearly in $\log \ell$). Thus, for large ℓ the proof overheads of SΣ and GK are much better than for the Std schemes. Somewhat surprising, however, is that even for $\ell = 4$ the length of SΣ is better than Std-s.
- The prover time for the Std schemes is better than for either SΣ or GK, although the gap between them shrinks as ℓ gets larger. SΣ, on the other hand, becomes less competitive as ℓ grows.
- The Std-f variant consistently offers the fastest verification time, but GK is not much worse. Again, SΣ becomes less competitive as ℓ grows.

For applications, one may choose which metric to optimize. SΣ offers the shortest proofs with a computational cost that is comparable to—even if slightly worse than—the other schemes as long as $\ell \leq 16$. On the other hand, Std-f offers the fastest proving and verification times for $\ell \leq 64$.

Overall, GK appears to be the best choice for a wide range of applications. It offers short proofs (even if not the shortest), has verification time comparable to—even if slightly worse than—the best schemes, and proving time that is reasonable for small ℓ while scaling gracefully to large ℓ.



(a) Proof size **(b)** Proving time **(c)** Verification time

Fig. 2. Comparing IHAC-BBS (solid lines) with the naive scheme (dashed lines), where the disjunctive proof is instantiated with either the Std-s (blue) or SΣ (red) approach. The x-axis is on a logarithmic scale.

Comparison to a naive scheme. Figure 2 compares IHAC-BBS to the naive IHAC scheme described in Section 4.3. Even for $\ell = 16$ issuers, we improve proof size by over 5× (for the SΣ variant), reduce proving time by more than 2.8×, and reduce verification time by over 4.6×. The difference becomes even more dramatic as the size of the policy set increases.

Acknowledgments

Work of A.F. and A.L. was partially funded by SPRIND (the German Federal Agency for Breakthrough Innovation) and Common Codes. The authors are solely responsible for the content of this publication; the positions presented here do not reflect the views of SPRIND or Common Codes.

Work of M.S. was funded by the Vienna Science and Technology Fund (WWTF) [10.47379/VRG18002] and [10.47379/ICT25-075], and by the Austrian Science Fund (FWF) [10.55776/F8515-N].

References

1. Au, M.H., Susilo, W., Mu, Y.: Constant-size dynamic k-TAA. In: De Prisco, R., Yung, M. (eds.) SCN 2006. LNCS, vol. 4116, pp. 111–125. Springer (2006). https://doi.org/10.1007/11832072_8
2. Bicer, O., Küpçü, A.: Versatile ABS: Usage limited, revocable, threshold traceable, authority hiding, decentralized attribute based signatures. Cryptology ePrint Archive, Report 2019/203 (2019), https://eprint.iacr.org/2019/203

3. Bobolz, J., Eidens, F., Krenn, S., Ramacher, S., Samelin, K.: Issuer-hiding attribute-based credentials. In: CANS 2021. LNCS, vol. 13099, pp. 158–178. Springer (Dec 2021). https://doi.org/10.1007/978-3-030-92548-2_9
4. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Franklin, M. (ed.) Crypto 2004. LNCS, vol. 3152, pp. 41–55. Springer (Aug 2004). https://doi.org/10.1007/978-3-540-28628-8_3
5. 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
6. Bowe, S.: BLS12-381: New zk-SNARK elliptic curve construction (2017), https://electriccoin.co/blog/new-snark-curve/
7. Camenisch, J., Drijvers, M., Lehmann, A.: Anonymous attestation using the strong Diffie Hellman assumption revisited. In: Franz, M., Papadimitratos, P. (eds.) TRUST 2016. LNCS, vol. 9824, pp. 1–20. Springer, Cham (Aug 2016). https://doi.org/10.1007/978-3-319-45572-3_1
8. 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 (May 2001). https://doi.org/10.1007/3-540-44987-6_7
9. 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 (Sep 2003). https://doi.org/10.1007/3-540-36413-7_20
10. 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 (Aug 2004). https://doi.org/10.1007/978-3-540-28628-8_4
11. Celi, S., Griffy, S., Hanzlik, L., Perez-Kempner, O., Slamanig, D.: SoK: Signatures with randomizable keys. In: Clark, J., Shi, E. (eds.) FC 2024, Part II. LNCS, vol. 14745, pp. 160–187. Springer (Mar 2024). https://doi.org/10.1007/978-3-031-78679-2_9
12. Chairattana-Apirom, R., Hofheinz, D., Tessaro, S.: Tight security for BBS signatures. In: Daemen, J., Thomé, E. (eds.) Eurocrypt 2026, Part I. LNCS, vol. 16541, pp. 246–276. Springer (2026). https://doi.org/10.1007/978-3-032-25291-3_9
13. Chairattana-Apirom, R., Tessaro, S.: On the concrete security of BBS/BBS+ signatures. In: Hanaoka, G., Yang, B.Y. (eds.) Asiacrypt 2025, Part VI. LNCS, vol. 16250, pp. 402–434. Springer (Dec 2025). https://doi.org/10.1007/978-981-95-5119-4_13
14. 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
15. Chaum, D.: Security without identification: transaction systems to make big brother obsolete. Commun. ACM **28**(10), 1030–1044 (Oct 1985). https://doi.org/10.1145/4372.4373, https://doi.org/10.1145/4372.4373
16. Connolly, A., Deschamps, J., Lafourcade, P., Perez-Kempner, O.: Protego: Efficient, revocable and auditable anonymous credentials with applications to hyperledger fabric. In: Isobe, T., Sarkar, S. (eds.) Indocrypt 2022. LNCS, vol. 13774, pp. 249–271. Springer (Dec 2022). https://doi.org/10.1007/978-3-031-22912-1_11
17. Connolly, A., Lafourcade, P., Perez-Kempner, O.: Improved constructions of anonymous credentials from structure-preserving signatures on equivalence classes. In: PKC 2022, Part I. LNCS, vol. 13177, pp. 409–438. Springer (2022). https://doi.org/10.1007/978-3-030-97121-2_15

18. 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 (Aug 1994). https://doi.org/10.1007/3-540-48658-5_19
19. Crites, E.C., Komlo, C., Maller, M., Tessaro, S., Zhu, C.: Snowblind: A threshold blind signature in pairing-free groups. In: Handschuh, H., Lysyanskaya, A. (eds.) Crypto 2023, Part I. LNCS, vol. 14081, pp. 710–742. Springer (Aug 2023). https://doi.org/10.1007/978-3-031-38557-5_23
20. Dai, Y., Steinberger, J.P.: Indifferentiability of 8-round Feistel networks. In: Robshaw, M., Katz, J. (eds.) Crypto 2016, Part I. LNCS, vol. 9814, pp. 95–120. Springer (Aug 2016). https://doi.org/10.1007/978-3-662-53018-4_4
21. Davidson, A., Goldberg, I., Sullivan, N., Tankersley, G., Valsorda, F.: Privacy pass: Bypassing internet challenges anonymously. PoPETs **2018**(3), 164–180 (Jul 2018). https://doi.org/10.1515/popets-2018-0026
22. De Santis, A., Di Crescenzo, G., Ostrovsky, R., Persiano, G., Sahai, A.: Robust non-interactive zero knowledge. In: Kilian, J. (ed.) Crypto 2001. LNCS, vol. 2139, pp. 566–598. Springer (Aug 2001). https://doi.org/10.1007/3-540-44647-8_33
23. Derler, D., Slamanig, D.: Key-homomorphic signatures: definitions and applications to multiparty signatures and non-interactive zero-knowledge. DCC **87**(6), 1373–1413 (2019). https://doi.org/10.1007/s10623-018-0535-9
24. Elkhiyaoui, K., De Caro, A., Androulaki, E.: Multi-issuer anonymous credentials without a root authority. Cryptology ePrint Archive, Report 2021/1669 (2021), https://eprint.iacr.org/2021/1669
25. European Commission: The European Digital Identity Wallet Architecture and Reference Framework (2023), https://digital-strategy.ec.europa.eu/en/library/european-digital-identity-wallet-architecture-and-reference-framework, see also https://github.com/eu-digital-identity-wallet/eudi-doc-architecture-and-reference-framework/issues/200
26. European Commission: Specification for the implementation of Zero-Knowledge Proofs based on multi-message signatures in the EUDI Wallet. v1.0 (2026), https://github.com/eu-digital-identity-wallet/eudi-doc-standards-and-technical-specifications/blob/main/docs/technical-specifications/ts14-zkps-from-mms.md
27. European Digital Identity Cooperation Group: European digital identity wallet, architecture and reference framework (2026), https://github.com/eu-digital-identity-wallet/eudi-doc-architecture-and-reference-framework/blob/main/docs/annexes/annex-2/annex-2.02-high-level-requirements-by-topic.md#a2331-topic-53-zero-knowledge-proofs
28. Faust, S., Kohlweiss, M., Marson, G.A., Venturi, D.: On the non-malleability of the Fiat-Shamir transform. In: Galbraith, S.D., Nandi, M. (eds.) Indocrypt 2012. LNCS, vol. 7668, pp. 60–79. Springer (Dec 2012). https://doi.org/10.1007/978-3-642-34931-7_5
29. Faz-Hernandez, A., Kwiatkowski, K.: Introducing CIRCL: An advanced cryptographic library, v1.6.3 (Jun 2019), available at https://github.com/cloudflare/circl
30. 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 (Aug 2005). https://doi.org/10.1007/11535218_10
31. Flamini, A., Friedrichs, K., Lehmann, A.: Constant-size issuer hiding for BBS credentials via randomizable keys. Cryptology ePrint Archive, Paper 2026/369 (2026), https://eprint.iacr.org/2026/369

32. Fleischhacker, N., Krupp, J., Malavolta, G., Schneider, J., Schröder, D., Simkin, M.: Efficient unlinkable sanitizable signatures from signatures with re-randomizable keys. In: Cheng, C.M., Chung, K.M., Persiano, G., Yang, B.Y. (eds.) PKC 2016, Part I. LNCS, vol. 9614, pp. 301–330. Springer (Mar 2016). https://doi.org/10.1007/978-3-662-49384-7_12
33. Fuchsbauer, G., Orrù, M.: Non-interactive mimblewimble transactions, revisited. In: Agrawal, S., Lin, D. (eds.) Asiacrypt 2022, Part I. LNCS, vol. 13791, pp. 713–744. Springer (Dec 2022). https://doi.org/10.1007/978-3-031-22963-3_24
34. Goel, A., Green, M., Hall-Andersen, M., Kaptchuk, G.: Stacking sigmas: A framework to compose Σ-protocols for disjunctions. In: Dunkelman, O., Dziembowski, S. (eds.) Eurocrypt 2022, Part II. LNCS, vol. 13276, pp. 458–487. Springer (May / Jun 2022). https://doi.org/10.1007/978-3-031-07085-3_16
35. Groth, J.: Simulation-sound NIZK proofs for a practical language and constant size group signatures. In: Lai, X., Chen, K. (eds.) Asiacrypt 2006. LNCS, vol. 4284, pp. 444–459. Springer (Dec 2006). https://doi.org/10.1007/11935230_29
36. Groth, J., Kohlweiss, M.: One-out-of-many proofs: Or how to leak a secret and spend a coin. In: Oswald, E., Fischlin, M. (eds.) Eurocrypt 2015, Part II. LNCS, vol. 9057, pp. 253–280. Springer (Apr 2015). https://doi.org/10.1007/978-3-662-46803-6_9
37. Hamburg, M., de Valence, H., Lovecruft, I., Arcieri, T.: The Ristretto group, https://ristretto.group/
38. Hendrickson, S., Iyengar, J., Pauly, T., Valdez, S., Wood, C.A.: Private access tokens (2021), https://www.ietf.org/archive/id/draft-private-access-tokens-01.html, Internet Draft
39. ISO/IEC: Information technology—security techniques—anonymous digital signatures, part 2: Mechanisms using a group public key (2013), https://www.iso.org/standard/56916.html, ISO/IEC 20008-2:2013
40. Kaaniche, N., Kane, S., Laurent, M., Traoré, J.: Improved issuer hiding for BBS-based anonymous credentials. Cryptology ePrint Archive, Paper 2026/555 (2026), https://eprint.iacr.org/2026/555
41. Karantaidou, I., Renawi, O., Baldimtsi, F., Kamarinakis, N., Katz, J., Loss, J.: Blind multisignatures for anonymous tokens with decentralized issuance. In: Luo, B., Liao, X., Xu, J., Kirda, E., Lie, D. (eds.) ACM CCS 2024. pp. 1508–1522. ACM Press (Oct 2024). https://doi.org/10.1145/3658644.3690364
42. Katz, J., Sefranek, M.: Issuer hiding for BBS-based anonymous credentials. In: Bai, S., Persichetti, E. (eds.) PKC 2026, Part III. LNCS, vol. 16553, pp. 437–465. Springer (2026). https://doi.org/10.1007/978-3-032-26737-5_14
43. Kiltz, E., Masny, D., Pan, J.: Optimal security proofs for signatures from identification schemes. In: Robshaw, M., Katz, J. (eds.) Crypto 2016, Part II. LNCS, vol. 9815, pp. 33–61. Springer (Aug 2016). https://doi.org/10.1007/978-3-662-53008-5_2
44. Lehmann, A., Nazarian, P., Özbay, C.: Stronger security for threshold blind signatures. In: Fehr, S., Fouque, P.A. (eds.) Eurocrypt 2025, Part II. LNCS, vol. 15602, pp. 335–364. Springer (May 2025). https://doi.org/10.1007/978-3-031-91124-8_12
45. Li, Z., Su, X., Wei, D., Liao, G., Wang, M.: iToken: One-time-use anonymous token with issuance hiding. Cryptology ePrint Archive, Paper 2026/570 (2026), https://eprint.iacr.org/2026/570
46. Looker, T., Kalos, V., Whitehead, A., Lodder, M., Bernstein, G.M.: The BBS signature scheme. Internet-Draft draft-irtf-cfrg-bbs-signatures-10, Internet Engineering

Task Force (Jan 2026), https://datatracker.ietf.org/doc/draft-irtf-cfrg-bbs-signatures/12/
47. 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
48. Mohan, D.I.: MoLE cryptography (Sep 2026), https://moderation-of-unlinkable-endorsements.github.io/internet-drafts/draft-authors-mole-crypto.html
49. Orrù, M.: Revisiting keyed-verification anonymous credentials. In: Huang, C.Y., Chen, J.C., Shieh, S.P., Lie, D., Cortier, V. (eds.) ACM CCS 2025. pp. 1188–1199. ACM Press (Oct 2025). https://doi.org/10.1145/3719027.3765163
50. Pointcheval, D., Sanders, O.: Short randomizable signatures. In: Sako, K. (ed.) CT-RSA 2016. LNCS, vol. 9610, pp. 111–126. Springer (Feb / Mar 2016). https://doi.org/10.1007/978-3-319-29485-8_7
51. Regulation (EU) 2024/1183 of the European parliament and of the council of 11 April 2024 amending regulation (EU) no 910/2014 as regards establishing the European Digital Identity Framework. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:32024R1183 (2024)
52. 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
53. Schlesinger, S., Jackson, D., Meunier, T.: Moderation of unlinkable endorsements (MoLE) architecture (2026), https://www.ietf.org/archive/id/draft-jms-mole-architecture-00.html
54. Schlesinger, S., Jackson, D., Meunier, T.: MoLE protocols (2026), https://www.ietf.org/archive/id/draft-jms-mole-protocols-00.html
55. Schlesinger, S.: Design proposal for PACT via ACTs with aggregating issuers (2026), https://github.com/antifraudcg/pact/issues/6
56. Sporny, M., Bernstein, G.: Data integrity BBS cryptosuites v1.0 (2026), https://www.w3.org/TR/vc-di-bbs, W3C Candidate Recommendation Draft
57. Ta, T.A., Chau, S.: Post-quantum issuer-hiding anonymous credential scheme. In: Proceedings of the 2025 1st Workshop on Quantum-Resistant Cryptography and Security. pp. 48–57. QRSEC '25, ACM (2026). https://doi.org/10.1145/3733820.3764678
58. Tessaro, S., Zhu, C.: Short pairing-free blind signatures with exponential security. In: Dunkelman, O., Dziembowski, S. (eds.) Eurocrypt 2022, Part II. LNCS, vol. 13276, pp. 782–811. Springer (May / Jun 2022). https://doi.org/10.1007/978-3-031-07085-3_27
59. Tessaro, S., Zhu, C.: Revisiting BBS signatures. In: Hazay, C., Stam, M. (eds.) Eurocrypt 2023, Part V. LNCS, vol. 14008, pp. 691–721. Springer (Apr 2023). https://doi.org/10.1007/978-3-031-30589-4_24
60. Yun, C., Wood, C.A., Raykova, M., Schlesinger, S.: Anonymous tokens with hidden metadata (2025), https://datatracker.ietf.org/doc/html/draft-yun-cfrg-athm-00

A Definitions of Anonymous Credentials

For completeness, we recall the standard definitions of unforgeability and anonymity for anonymous credentials, taken almost verbatim from Katz and Sefranek [42].

None of our results depend on these definitions directly. Instead, they serve as a reference point for the issuer-hiding notions in Section 4.1, which extend them to multiple issuers and additionally hide which issuer issued a presented credential.

Unforgeability. Unforgeability ensures that no adversary can generate a valid proof for disclosed attributes $(m_i)_{i\in\mathcal{D}}$ in context ctx unless it has obtained a credential on an attribute vector whose attributes match the disclosed attributes, or replays a previous proof for the same disclosed attributes and context.

Definition 17 (Unforgeability). *Consider the following experiment involving a stateful adversary $\mathcal{A}$ and scheme Π:*

1. *Run* $\mathsf{params} \leftarrow \mathsf{Setup}$ *and* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}$.
2. $\mathcal{A}(\mathsf{params}, \mathsf{pk})$ *may repeatedly query the following oracles:*
   - $\mathcal{O}\mathsf{Issue}$*, which on input* $\boldsymbol{m}$ *returns* $\mathsf{Issue}(\mathsf{sk}, \boldsymbol{m})$.
   - $\mathcal{O}\mathsf{Issue}'$*, which on the* i*th query with input* $\boldsymbol{m}_i$ *generates a credential* $\sigma_i \leftarrow \mathsf{Issue}(\mathsf{sk}, \boldsymbol{m}_i)$*, but returns nothing to* $\mathcal{A}$.
   - $\mathcal{O}\mathsf{Show}$*, which on input* $(i, \mathcal{D}, \mathsf{ctx})$ *returns proof* $\mathsf{Show}(\mathsf{pk}, \mathcal{D}, \boldsymbol{m}_i, \mathsf{ctx}, \sigma_i)$.
3. $\mathcal{A}$ *outputs* $\mathcal{D}^* \subseteq [n]$, $(m_i^*)_{i\in\mathcal{D}^*}$, ctx^*, *and* π.

$\mathcal{A}$ $\mathsf{succeeds}$ *if*

1. $\mathsf{VfShow}(\mathsf{pk}, \mathcal{D}^*, (m_i^*)_{i\in\mathcal{D}^*}, \mathsf{ctx}^*, \pi) = 1$,
2. $\mathcal{A}$ *never queried* $\mathcal{O}\mathsf{Issue}(\boldsymbol{m})$ *with* $(m_i)_{i\in\mathcal{D}^*} = (m_i^*)_{i\in\mathcal{D}^*}$,
3. $\mathcal{A}$ *never queried* $\mathcal{O}\mathsf{Show}(i, \mathcal{D}^*, \mathsf{ctx}^*)$ *where* $(m_{i,j})_{j\in\mathcal{D}^*} = (m_j^*)_{j\in\mathcal{D}^*}$.

Let $\mathsf{Adv}^{\mathsf{ac\text{-}uf}}_{\mathcal{A},\Pi} \stackrel{\text{def}}{=} \Pr[\mathcal{A}\ \mathsf{succeeds}]$.

Anonymity. Anonymity requires that proofs reveal the minimum information necessary. Specifically, a proof only reveals disclosed attributes, and is unlinkable to the credential used to generate it or to other proofs generated using that credential (beyond whatever linkability is implied by the disclosed attributes). This holds even if verifiers and issuers are malicious and colluding.

Definition 18 (Anonymity). *Consider the following experiment involving a stateful adversary $\mathcal{A}$ and scheme Π:*

1. *Run* $\mathsf{params} \leftarrow \mathsf{Setup}$.
2. $\mathcal{A}(\mathsf{params})$ *outputs* $(\mathsf{pk}, \mathcal{D}, \mathsf{ctx}, \boldsymbol{m}_0, \sigma_0, \boldsymbol{m}_1, \sigma_1)$ *where:*
   - $\mathsf{VfCred}(\mathsf{pk}, \boldsymbol{m}_0, \sigma_0) = \mathsf{VfCred}(\mathsf{pk}, \boldsymbol{m}_1, \sigma_1) = 1$ *and*
   - $m_{0,i} = m_{1,i}$ *for all* $i \in \mathcal{D}$.
3. *Choose* $b \leftarrow \{0, 1\}$*, and set* $\pi \leftarrow \mathsf{Show}(\mathsf{pk}, \mathcal{D}, \boldsymbol{m}_b, \mathsf{ctx}, \sigma_b)$.
4. $\mathcal{A}(\pi)$ *outputs a bit* b'.

$\mathcal{A}$ $\mathsf{succeeds}$ *if* $b' = b$. *Let* $\mathsf{Adv}^{\mathsf{anon}}_{\mathcal{A},\Pi} \stackrel{\text{def}}{=} |\Pr[\mathcal{A}\ \mathsf{succeeds}] - 1/2|$.

We say Π is *perfectly anonymous* if $\mathsf{Adv}^{\mathsf{anon}}_{\mathcal{A},\Pi} = 0$ for all adversaries $\mathcal{A}$.

B Tighter Security for IHAC-BBS

We show that it is possible to prove tighter security for the BBS-based IHAC scheme described in Section 4.3 than what is implied by Theorem 3. Specifically, we show in Appendix B.1 that it is possible to prove tight multi-signer security for BBS signatures; interestingly, our proof relies on key randomizability. We then show in Appendix B.2 how to modify the proof of Theorem 3 so as to reduce to multi-signer security of BBS signatures.

B.1 Tight Multi-Signer Security of BBS Signatures

We recall the notions of strong (single-signer) *existential unforgeability under adaptive chosen-message attacks* ("unforgeability" for short) and strong multi-signer security[8] of a signature scheme $\Sigma = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Vf})$.

Definition 19 (Strong Unforgeability). *Consider the following experiment involving a stateful adversary $\mathcal{A}$ and signature scheme Σ:*

1. *Run* $\mathsf{params} \leftarrow \mathsf{Setup}$ *and* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}$.
2. $\mathcal{A}(\mathsf{params}, \mathsf{pk})$ *may repeatedly query the oracle* $\mathcal{O}\mathsf{Sign}$*, which on input m returns* $\mathsf{Sign}(\mathsf{sk}, m)$.
3. $\mathcal{A}$ *outputs m^* and σ.*

$\mathcal{A}$ succeeds *if (1)* $\mathsf{Vf}(\mathsf{pk}, m^*, \sigma) = 1$ *and (2) $\mathcal{A}$ never made a query* $\mathcal{O}\mathsf{Sign}(m^*)$ *with output σ. Let* $\mathsf{Adv}^{\mathsf{suf}}_{\mathcal{A},\Sigma} \stackrel{\text{def}}{=} \Pr[\mathcal{A} \text{ succeeds}]$.

Definition 20 (Strong Multi-Signer Security). *Consider the following experiment involving a stateful adversary $\mathcal{A}$ and signature scheme Σ:*

1. *Run* $\mathsf{params} \leftarrow \mathsf{Setup}$.
2. $\mathcal{A}(\mathsf{params})$ *may repeatedly query the following oracles:*
   - $\mathcal{O}\mathsf{KeyGen}$*, which on the ith such query computes* $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}$ *and returns* pk_i *to $\mathcal{A}$.*
   - $\mathcal{O}\mathsf{Sign}$*, which on input (j, m) returns* $\mathsf{Sign}(\mathsf{sk}_j, m)$.
3. $\mathcal{A}$ *outputs i^*, m^*, and σ.*

$\mathcal{A}$ succeeds *if (1)* $\mathsf{Vf}(\mathsf{pk}_{i^*}, m^*, \sigma) = 1$ *and (2) $\mathcal{A}$ never made a query* $\mathcal{O}\mathsf{Sign}(i^*, m^*)$ *with output σ. Let* $\mathsf{Adv}^{\mathsf{mu\text{-}suf}}_{\mathcal{A},\Sigma} \stackrel{\text{def}}{=} \Pr[\mathcal{A} \text{ succeeds}]$.

We let $\mathsf{BBS} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Vf})$ denote the BBS signature scheme (cf. Definition 16), and show that its strong multi-signer security *tightly* reduces to strong single-signer security. (Note that existing security results for BBS signatures [59,12] prove strong unforgeability.)

[8] This notion is also known as "multi-user" security in previous works [43].

Theorem 5 (Tight Multi-Signer Security of BBS). *Let $\mathcal{A}$ be an adversary against strong multi-signer security of BBS signatures making at most $q_S < p - 1$ signing queries. Then there is an adversary $\mathcal{B}$ against strong unforgeability of BBS signatures running in roughly the same time as $\mathcal{A}$ and making at most q_S signing queries such that*

$$\mathsf{Adv}^{\mathsf{mu\text{-}suf}}_{\mathcal{A},\mathsf{BBS}} \leq 2 \cdot \mathsf{Adv}^{\mathsf{suf}}_{\mathcal{B},\mathsf{BBS}}.$$

Proof. Fix an adversary $\mathcal{A}$ in the experiment from Definition 20 for BBS signatures. Consider the following adversary $\mathcal{B}$ against strong (single-signer) unforgeability of BBS signatures with access to the oracle $\mathcal{O}\mathsf{Sign}'$:

1. On input $g_1, g_2, \boldsymbol{h} = (h_i)_{i\in[n]}$, and $\mathsf{pk} = g_2^x$, run $\mathcal{A}(g_1, g_2, \boldsymbol{h})$ answering its queries as follows:
   - $\mathcal{O}\mathsf{KeyGen}$: On the ith query, sample $b_i \leftarrow \{0,1\}$ and $r_i \leftarrow \mathbb{Z}_p$. Return $\mathsf{pk}_i := \mathsf{pk}^{b_i} \cdot g_2^{r_i}$ to $\mathcal{A}$.
   - $\mathcal{O}\mathsf{Sign}(i_j, \boldsymbol{m}_j)$: On the jth query, if $b_{i_j} = 1$, query $(A_j, e'_j) \leftarrow \mathcal{O}\mathsf{Sign}'(\boldsymbol{m}_j)$, set $e_j := e'_j - r_{i_j}$, and return (A_j, e_j) to $\mathcal{A}$. Otherwise, return the honestly computed signature $(A_j, e_j) \leftarrow \mathsf{Sign}(r_{i_j}, \boldsymbol{m}_j)$.
2. When $\mathcal{A}$ outputs $(i^*, \boldsymbol{m}, (A, e))$:
   (a) If $\exists j^* \in [q_S]$, $i_{j^*} \neq i^*$, such that $g_2^{e-e_{j^*}} = \mathsf{pk}_{i_{j^*}}/\mathsf{pk}_{i^*}$ and $b_{i_{j^*}} \neq b_{i^*}$, extract $x = \log_{g_2} \mathsf{pk}$ as

$$x := (b_{i_{j^*}} - b_{i^*}) \cdot (e - e_{j^*} + r_{i^*} - r_{i_{j^*}}). \tag{4}$$

   Then choose an arbitrary message $\boldsymbol{m}' \in \mathbb{Z}_p^n$ with $C(\boldsymbol{m}') \neq 1$ on which $\mathcal{O}\mathsf{Sign}'$ has not been queried (since $q_S < p - 1$, such a message always exists), honestly compute $\sigma \leftarrow \mathsf{Sign}(x, \boldsymbol{m}')$, and output the forgery $(\boldsymbol{m}', \sigma)$.
   (b) Otherwise, output $(\boldsymbol{m}, (A, e + r_{i^*}))$.

It is not hard to see that the simulation of the multi-signer security experiment for $\mathcal{A}$ is perfect. The public keys are distributed independently and uniformly in $\mathbb{G}_2$. Signing queries are answered correctly: if $b_{i_j} = 0$, this is clear; otherwise, (A_j, e_j) satisfies the verification equation and $e_j = e'_j - r_{i_j}$ is uniform in $\mathbb{Z}_p \setminus \{-x - r_{i_j}\}$ since e'_j is uniform in $\mathbb{Z}_p \setminus \{-x\}$. Moreover, $\mathcal{B}$ makes at most q_S queries to $\mathcal{O}\mathsf{Sign}'$ as claimed.

Define the event

$$\mathsf{Coll} \stackrel{\text{def}}{=} \exists j^* \in \{j \in [q_S] \mid i_j \neq i^*\} : g_2^{e-e_{j^*}} = \mathsf{pk}_{i_{j^*}}/\mathsf{pk}_{i^*}.$$

Since the choice of each b_i is independent of the view of $\mathcal{A}$ and $i_{j^*} \neq i^*$, we get

$$\Pr\left[\mathcal{A} \text{ succeeds} \wedge \mathsf{Coll} \wedge b_{i_{j^*}} \neq b_{i^*}\right] = \frac{1}{2} \cdot \Pr\left[\mathcal{A} \text{ succeeds} \wedge \mathsf{Coll}\right]$$

and

$$\Pr\left[\mathcal{A} \text{ succeeds} \wedge \overline{\mathsf{Coll}} \wedge b_{i^*} = 1\right] = \frac{1}{2} \cdot \Pr\left[\mathcal{A} \text{ succeeds} \wedge \overline{\mathsf{Coll}}\right].$$

We will show that

$$\begin{aligned}\Pr[\mathcal{B}\ \mathsf{succeeds}] \geq \Pr\big[\mathcal{A}\ \mathsf{succeeds} \wedge \mathsf{Coll} \wedge b_{i_{j^*}} \neq b_{i^*}\big] \\ + \Pr\big[\mathcal{A}\ \mathsf{succeeds} \wedge \overline{\mathsf{Coll}} \wedge b_{i^*} = 1\big],\end{aligned}$$

which, together with the previous two equations, gives the bound in the theorem. To that end, we show that $\mathcal{B}$ succeeds if either of the two cases above occurs.

Case 1: $\mathcal{A}$ succeeds, Coll occurs, and $b_{i_{j^*}} \neq b_{i^*}$. In this case, $\mathcal{B}$ outputs $(\boldsymbol{m}', \sigma)$, where $\boldsymbol{m}'$ is an arbitrary message with $C(\boldsymbol{m}') \neq 1$ on which $\mathcal{O}\mathsf{Sign}'$ has not been queried and $\sigma \leftarrow \mathsf{Sign}(x, \boldsymbol{m}')$ with x computed as in Equation (4). Clearly, the forgery is non-trivial. It remains to show that the extracted x is the discrete logarithm of the given public key pk. Using $b_{i_{j^*}} - b_{i^*} = \pm 1$, we get

$$\begin{aligned} g_2^{e-e_{j^*}} = \mathsf{pk}_{i_{j^*}}/\mathsf{pk}_{i^*} &\iff e - e_{j^*} = b_{i_{j^*}} x + r_{i_{j^*}} - (b_{i^*} x + r_{i^*}) \\ &\iff e - e_{j^*} + r_{i^*} - r_{i_{j^*}} = (b_{i_{j^*}} - b_{i^*}) x \\ &\iff x = (b_{i_{j^*}} - b_{i^*}) \cdot (e - e_{j^*} + r_{i^*} - r_{i_{j^*}}). \end{aligned}$$

Case 2: $\mathcal{A}$ succeeds, Coll does not occur, and $b_{i^*} = 1$. In this case, $\mathcal{B}$ receives $(i^*, \boldsymbol{m}, (A, e))$ from $\mathcal{A}$ and outputs the forgery $(\boldsymbol{m}, (A, e + r_{i^*}))$. Recall that if $\mathcal{A}$ succeeds, then (1) $\mathsf{Vf}(\mathsf{pk}_{i^*}, \boldsymbol{m}, (A, e)) = 1$, i.e.,

$$\mathsf{e}(A, \mathsf{pk}_{i^*}) = \mathsf{e}(A^{-e} C(\boldsymbol{m}), g_2) \tag{5}$$

with $C(\boldsymbol{m}) \neq 1$, and (2) $\mathcal{A}$ never made a query $\mathcal{O}\mathsf{Sign}(i^*, \boldsymbol{m})$ with output (A, e). We need to show that $\mathsf{Vf}(\mathsf{pk}, \boldsymbol{m}, (A, e + r_{i^*})) = 1$ and that $\mathcal{B}$ never made a query $\mathcal{O}\mathsf{Sign}'(\boldsymbol{m})$ with output $(A, e + r_{i^*})$. First, since $b_{i^*} = 1$, we have $\mathsf{pk}_{i^*} = \mathsf{pk} \cdot g_2^{r_{i^*}}$. Thus, using Equation (5), we get

$$\begin{aligned} \mathsf{e}(A, \mathsf{pk}) &= \mathsf{e}(A, \mathsf{pk} \cdot g_2^{r_{i^*}} \cdot g_2^{-r_{i^*}}) \\ &= \mathsf{e}(A, \mathsf{pk}_{i^*}) \cdot \mathsf{e}(A, g_2^{-r_{i^*}}) \\ &= \mathsf{e}(A^{-e} C(\boldsymbol{m}), g_2) \cdot \mathsf{e}(A^{-r_{i^*}}, g_2) \\ &= \mathsf{e}(A^{-(e+r_{i^*})} C(\boldsymbol{m}), g_2), \end{aligned}$$

and so $(A, e + r_{i^*})$ is a valid BBS signature on $\boldsymbol{m}$ under pk. Second, suppose toward a contradiction that $\mathcal{B}$ made a query $\mathcal{O}\mathsf{Sign}'(\boldsymbol{m})$ with output $(A, e + r_{i^*})$ when answering the jth signing query of $\mathcal{A}$. This means that $b_{i_j} = 1$ and that $\mathcal{A}$ received $(A_j, e_j) \coloneqq (A, e + r_{i^*} - r_{i_j})$. Moreover, we must have $i_j \neq i^*$; otherwise $(A_j, e_j) = (A, e)$. But this contradicts $\overline{\mathsf{Coll}}$ since

$$g_2^{e-e_j} = g_2^{e-(e+r_{i^*}-r_{i_j})} = g_2^{r_{i_j}-r_{i^*}} = \mathsf{pk}_{i_j}/\mathsf{pk}_{i^*}.$$

Therefore, $(\boldsymbol{m}, (A, e + r_{i^*}))$ is a non-trivial forgery. □

B.2 Tighter Issuer-Hiding Unforgeability for IHAC(-BBS)

The following is a tighter version of Theorem 3. Although written for our BBS-based instantiation IHAC-BBS, the proof can be adapted easily for our generic construction. We write Π_{IHAC} for IHAC-BBS.

Theorem 6 (Tighter Issuer-Hiding Unforgeability of IHAC-BBS). *Assume the ZKPoK used in Π_{IHAC} has perfect zero knowledge. Let $\mathcal{A}$ be an adversary against issuer-hiding unforgeability of Π_{IHAC} making at most q_K key-generation queries and q_I issuance queries with $q_K+q_I < p-1$. Then there is an adversary $\mathcal{B}$ against strong multi-signer unforgeability (cf. Definition 20) of BBS signatures running in roughly the same time as $\mathcal{A}$ and making at most q_K key-generation queries and $q_K + q_I$ signing queries such that*

$$\mathsf{Adv}^{\mathsf{ih\text{-}ac\text{-}uf}}_{\mathcal{A},\Pi_{\mathsf{IHAC}}} \leq \mathsf{Adv}^{\mathsf{mu\text{-}suf}}_{\mathcal{B},\mathsf{BBS}} + \frac{1}{(p-1)p^{n-1}}\,,$$

unless $\mathcal{A}$ violates simulation extractability of the ZKPoK.

Proof. We begin by describing the high-level intuition for our reduction. In the multi-signer unforgeability experiment for BBS signatures the adversary has access to a key-generation oracle and a signing oracle, which can be used directly to simulate $\mathcal{O}\mathsf{KeyGen}$ and $\mathcal{O}\mathsf{Issue}$ queries, respectively. For $\mathcal{O}\mathsf{Show}$ queries, we rely on the facts that (a) any valid signature for a key $\mathsf{pk} = g_2^x$ allows computation of a pair (A, A^x) for some $A \in \mathbb{G}_1^*$, and (b) given such a pair, $\mathsf{Show}(\mathcal{P}, \mathsf{pk}, \mathcal{D}, \boldsymbol{m}, \mathsf{ctx}, \sigma)$ with $\mathsf{pk} = g_2^x$ can be perfectly simulated. The first fact follows easily: given a valid signature (A, e) on $\boldsymbol{m}$ with respect to pk, we have $(A, A^{-e} \cdot C(\boldsymbol{m})) = (A, A^x)$. As for the second claim, note that the output of Show can be simulated as follows:

- Sample $r \leftarrow \mathbb{Z}_p$ and blind pk as $\widetilde{\mathsf{pk}} := g_2^r \cdot \mathsf{pk}$.
- Sample $t \leftarrow \mathbb{Z}_p^*$, and set $A' := A^t$ and $\bar{A} := (A^x \cdot A^r)^t$.
- Simulate the ZKPoK of statement (2).

It is not hard to verify that this simulation is perfect.

We now give the details of the proof. Consider the following adversary $\mathcal{B}$ against strong multi-signer unforgeability of BBS signatures with access to oracles $\mathcal{O}\mathsf{KeyGen}'$ and $\mathcal{O}\mathsf{Sign}$:

1. On input $g_1, g_2, \boldsymbol{h}$, run $\mathcal{A}(g_1, g_2, \boldsymbol{h})$ answering its queries as follows:
   - $\mathcal{O}\mathsf{KeyGen}$: On the ith query, query $\mathsf{pk}_i \leftarrow \mathcal{O}\mathsf{KeyGen}'$ and return pk_i to $\mathcal{A}$. Also sample $\boldsymbol{m}_i \leftarrow \mathbb{Z}_p^n$ with $C(\boldsymbol{m}_i) \neq 1$ and query $(A_i,e_i) \leftarrow \mathcal{O}\mathsf{Sign}(i, \boldsymbol{m}_i)$. Set $\bar{A}_i := A_i^{-e_i} \cdot C(\boldsymbol{m}_i)$ and save $A_i, \bar{A}_i$ for later.
   - $\mathcal{O}\mathsf{Issue}(j, \boldsymbol{m})$: Query $\sigma \leftarrow \mathcal{O}\mathsf{Sign}(j, \boldsymbol{m})$ and return σ to $\mathcal{A}$.
   - $\mathcal{O}\mathsf{Issue}'(i_j, \boldsymbol{m}'_j)$: Do nothing.
   - $\mathcal{O}\mathsf{Show}(j, \mathcal{P}, \mathcal{D}, \mathsf{ctx})$: Sample $r \leftarrow \mathbb{Z}_p$ and $t \leftarrow \mathbb{Z}_p^*$. Return $\widetilde{\mathsf{pk}} := g_2^r \cdot \mathsf{pk}_{i_j}$, $A' := A_{i_j}^t$, $\bar{A} := (\bar{A}_{i_j} \cdot A_{i_j}^r)^t$, and a simulated ZKPoK of statement (2).

2. When $\mathcal{A}$ outputs $\mathcal{P}^* = \{\mathsf{pk}'_j\} \subseteq \{\mathsf{pk}_i\}$, $\mathcal{D}^*$, $(m_i^*)_{i\in\mathcal{D}^*}$, ctx^*, $(\widetilde{\mathsf{pk}}, A', \bar{A})$, and a ZKPoK of statement (2), abort if $\mathcal{A}$ does not succeed. Otherwise, try to extract $(m_i^*)_{i\notin\mathcal{D}}, \tilde{e}, t, j^*, r$ from the ZKPoK and abort if extraction fails.
3. Let $i^* \in [q_K]$ s.t. $\mathsf{pk}_{i^*} = \mathsf{pk}'_{j^*}$. Set $\boldsymbol{m}^* \coloneqq (m_i^*)_{i\in[n]}$. If $C(\boldsymbol{m}^*) \neq 1$ and $t \neq 0$, output the forgery $(i^*, \boldsymbol{m}^*, (A, e))$, where $A \coloneqq (A')^{1/t}$ and $e \coloneqq \tilde{e} + r$. Otherwise, choose an arbitrary message $\boldsymbol{m}' \in \mathbb{Z}_p^n$ with $C(\boldsymbol{m}') \neq 1$ on which $\mathcal{O}\mathsf{Sign}$ has not been queried (since $q_I < p - 1$, such a message always exists), honestly compute $\sigma \leftarrow \mathsf{Sign}(-(\tilde{e}+r), \boldsymbol{m}')$, and output the forgery $(i^*, \boldsymbol{m}', \sigma)$.

The simulation of the issuer-hiding unforgeability experiment for $\mathcal{A}$ is perfect: $(g_1, g_2, \boldsymbol{h})$, $\mathcal{O}\mathsf{KeyGen}$, and $\mathcal{O}\mathsf{Issue}$ clearly have the same distribution, and by the arguments laid out at the beginning of the proof, $\mathcal{O}\mathsf{Show}$ also has the correct distribution.

Next, we show that $\mathcal{B}$ succeeds if $\mathcal{A}$ succeeds unless $\mathcal{A}$ violates simulation extractability. First note that if the output of $\mathcal{A}$ satisfies the conditions of Definition 13, then extraction fails only if $\mathcal{A}$ violates simulation extractability of the ZKPoK for statement (2): the ZKPoK returned by $\mathcal{A}$ verifies and $\mathcal{A}$ has not obtained another proof on the same pair of statement $(\mathcal{P}^*, \mathcal{D}^*, (m_i^*)_{i\in\mathcal{D}^*}, \widetilde{\mathsf{pk}}, A', \bar{A})$ and context ctx^* since it never made a query $\mathcal{O}\mathsf{Show}(j, \mathcal{P}^*, \mathcal{D}^*, \mathsf{ctx}^*)$ with $\mathsf{pk}_{i_j} \in \mathcal{P}^*$ and $m'_{j,i} = m_i^*$ for all $i \in \mathcal{D}^*$. If extraction succeeds, then $\mathcal{B}$ has values $(m_i^*)_{i\notin\mathcal{D}}, \tilde{e}, t, i^*, r$ for which

$$\bar{A} = (A')^{-\tilde{e}} \cdot C(\boldsymbol{m}^*)^t \ \wedge\ \widetilde{\mathsf{pk}} = g_2^r \cdot \mathsf{pk}_{i^*}.$$

Moreover, since $\mathsf{VfShow}(\mathcal{P}^*, \mathcal{D}^*, (m_i^*)_{i\in\mathcal{D}^*}, \mathsf{ctx}^*, \pi) = 1$, we have

$$\mathsf{e}\big(A', \widetilde{\mathsf{pk}}\big) = \mathsf{e}(\bar{A}, g_2) = \mathsf{e}\big((A')^{-\tilde{e}} \cdot C(\boldsymbol{m}^*)^t, g_2\big). \tag{6}$$

If $C(\boldsymbol{m}^*) = 1$ or $t = 0$, then the equation above simplifies to $\mathsf{e}\big(A', \widetilde{\mathsf{pk}}\big) = \mathsf{e}\big((A')^{-\tilde{e}}, g_2\big)$. Thus, $\widetilde{\mathsf{pk}} = g_2^{-\tilde{e}}$ and $\mathsf{pk}_{i^*} = g_2^{-(\tilde{e}+r)}$, and $\sigma \leftarrow \mathsf{Sign}(-(\tilde{e}+r), \boldsymbol{m}')$ output by $\mathcal{B}$ in this case is a valid BBS signature on $\boldsymbol{m}'$ under pk_{i^*}.

Otherwise, we have $C(\boldsymbol{m}^*) \neq 1$ and $t \neq 0$. Letting $A = (A')^{1/t}$, we get $\bar{A} = (A^{-\tilde{e}} \cdot C(\boldsymbol{m}^*))^t$, which together with Equation (6) implies

$$\mathsf{e}\big(A, \widetilde{\mathsf{pk}}\big) = \mathsf{e}(A^{-\tilde{e}} \cdot C(\boldsymbol{m}^*), g_2).$$

Thus,

$$\begin{aligned}
\mathsf{e}(A, \mathsf{pk}_{i^*}) &= \mathsf{e}\big(A, \widetilde{\mathsf{pk}} \cdot g_2^{-r}\big) \\
&= \mathsf{e}\big(A, \widetilde{\mathsf{pk}}\big) \cdot \mathsf{e}(A, g_2^{-r}) \\
&= \mathsf{e}(A^{-\tilde{e}} \cdot C(\boldsymbol{m}^*), g_2) \cdot \mathsf{e}(A^{-r}, g_2) \\
&= \mathsf{e}(A^{-(\tilde{e}+r)} \cdot C(\boldsymbol{m}^*), g_2) = \mathsf{e}(A^{-e} \cdot C(\boldsymbol{m}^*), g_2),
\end{aligned}$$

and (A, e) output by $\mathcal{B}$ is a valid BBS signature on $\boldsymbol{m}^*$ under pk_{i^*}.

It remains to argue that $\mathcal{B}$ has not queried its own signing oracle on $\boldsymbol{m}^*$. Since $\mathcal{A}$ never queried $\mathcal{O}\mathsf{Issue}(j, \boldsymbol{m})$ with $\mathsf{pk}_j \in \mathcal{P}^*$ and $m_i = m_i^*$ for all $i \in \mathcal{D}^*$, the

only way this could have happened is if $\boldsymbol{m}^* = \boldsymbol{m}_{i^*}$. But $\boldsymbol{m}_{i^*}$ is chosen uniformly (subject to the constraint that $C(\boldsymbol{m}_{i^*}) \neq 1$), and the view of $\mathcal{A}$ is independent of $\boldsymbol{m}_{i^*}$. Thus, the probability that $\boldsymbol{m}^* = \boldsymbol{m}_{i^*}$ is at most $\frac{1}{(p-1)p^{n-1}}$.

Overall, we get

$$\mathsf{Adv}^{\mathsf{ih\text{-}ac\text{-}uf}}_{\mathcal{A},\Pi_{\mathsf{IHAC}}} \leq \mathsf{Adv}^{\mathsf{mu\text{-}suf}}_{\mathcal{B},\mathsf{BBS}} + \frac{1}{(p-1)p^{n-1}},$$

where $\mathcal{B}$ makes at most $q_K + q_I < p - 1$ signing queries. □

Combining Theorem 6 with our tight multi-signer to single-signer reduction for BBS signatures (Theorem 5), we obtain the following corollary.

Corollary 1. *Assume the ZKPoK used in Π_{IHAC} has perfect zero knowledge. Let $\mathcal{A}$ be an adversary against issuer-hiding unforgeability of Π_{IHAC} making at most q_K key-generation queries and q_I issuance queries with $q_K + q_I < p - 1$. Then there is an adversary $\mathcal{B}$ against strong unforgeability (cf. Definition 19) of BBS signatures running in roughly the same time as $\mathcal{A}$ and making at most $q_K + q_I$ signing queries such that*

$$\mathsf{Adv}^{\mathsf{ih\text{-}ac\text{-}uf}}_{\mathcal{A},\Pi_{\mathsf{IHAC}}} \leq 2 \cdot \mathsf{Adv}^{\mathsf{suf}}_{\mathcal{B},\mathsf{BBS}} + \frac{1}{(p-1)p^{n-1}},$$

unless $\mathcal{A}$ violates simulation extractability of the ZKPoK.

C Straight-Line Extractability of Classical OR Proofs

We show that the two variants of the classical OR proof [18] from Section 5.1—Std-f and Std-s—are *straight-line* extractable when used in our IHAT-TZ instantiation (Section 3.4). Rather than proving standard knowledge soundness in which the adversary picks the statement at will, we show how to extract inside the issuer-hiding one-more unforgeability experiment (Definition 7), which is exactly what the proof of Theorem 1 requires.

Both ingredients we combine here reside in the AGM: the original security proof of BS_3 by Tessaro and Zhu [58] is in the algebraic-group and random-oracle models (AGM+ROM), and straight-line extractability of the Fiat–Shamir transform of Σ-protocols for linear relations was recently shown by Orrù [49], under an independence condition on the group elements of the relation. These results do not compose out of the box, since AGM reductions are always made relative to a basis, and inside a larger protocol the adversary sees further group elements that it may build its proof from. Here it also gets Z, the honest keys $\mathsf{pk}_1, \ldots, \mathsf{pk}_\ell$, and a pair (A_k, C_k) per issuance session. Ruling these out turned out to be non-trivial, and is shown via the reductions $\mathcal{B}_1$ and $\mathcal{B}_2$ to the discrete-logarithm problem below. Note that we thereby stay within the assumptions already used for BS_3, i.e., our issuer-hiding version IHAT-TZ needs nothing beyond hardness of the discrete-logarithm problem in the AGM+ROM.

Recall that the main difference between these two approaches is that an Std-f proof contains the first prover message (a list of group elements), while Std-s instead has the verifier recompute these group elements from the challenge that is now included in the proof. In the AGM+ROM, these two variants are equivalent because the adversary must also provide AGM representations for any group elements it queries to the random oracle. Therefore, we only treat Std-f in the remainder of this section.

Let $H' \colon \{0,1\}^* \to \mathbb{Z}_p$ be a hash function modeled as a random oracle. Let $\mathcal{P} = \{\mathsf{pk}_i\}_{i \in [\ell]}$ be a set of public keys $\mathsf{pk}_i \in \mathbb{G}$ and let $\widetilde{\mathsf{pk}} \in \mathbb{G}$ be a claimed randomization of one of those keys. We first provide the details of the ZKPoK for proving that

$$\exists r \in \mathbb{Z}_p, i^* \in [\ell] : g^r = \widetilde{\mathsf{pk}}/\mathsf{pk}_{i^*},$$

using the Std-f approach:

- The prover, on input $\mathcal{P}$, $\widetilde{\mathsf{pk}}$, r, i^*, chooses $r_{i^*} \leftarrow \mathbb{Z}_p$ and sets $R_{i^*} \coloneqq g^{r_{i^*}}$. It then chooses $c_i, z_i \leftarrow \mathbb{Z}_p$ and sets $R_i \coloneqq g^{z_i} \cdot (\widetilde{\mathsf{pk}}/\mathsf{pk}_i)^{-c_i}$ for each $i \neq i^*$. Finally, it computes $c \coloneqq H'\big(\widetilde{\mathsf{pk}}, (\mathsf{pk}_i, R_i)_{i \in [\ell]}\big)$, $c_{i^*} \coloneqq c - \sum_{i \neq i^*} c_i$, and $z_{i^*} \coloneqq r_{i^*} + r c_{i^*}$, and returns the proof $\pi \coloneqq (R_i, c_i, z_i)_{i \in [\ell]}$.
- The verifier, on input $\mathcal{P}$, $\widetilde{\mathsf{pk}}$, π, checks that $H'\big(\widetilde{\mathsf{pk}}, (\mathsf{pk}_i, R_i)_{i \in [\ell]}\big) = \sum_{i \in [\ell]} c_i$ and that $g^{z_i} = R_i \cdot (\widetilde{\mathsf{pk}}/\mathsf{pk}_i)^{c_i}$ for all $i \in [\ell]$.

Next, we describe the issuer-hiding one-more unforgeability experiment from Definition 7 for the IHAT-TZ scheme instantiated with the above ZKPoK in the AGM+ROM. To simplify the presentation, we focus on the case where the adversary outputs a single forgery, i.e., $k = 1$. This case directly extends to straight-line extraction from multiple forgeries while preserving the extraction error. The experiment proceeds as follows:

1. Algebraic adversary $\mathcal{A}$ is given $g, Z \in \mathbb{G}$ and access to the following oracles:
   - $\mathcal{O}\mathsf{KeyGen}$, which on the ith query chooses $\mathsf{sk}_i \leftarrow \mathbb{Z}_p$ and returns $\mathsf{pk}_i \coloneqq g^{\mathsf{sk}_i}$ to $\mathcal{A}$.
   - Concurrent executions of the issuance protocol, where the kth execution of the issuance protocol with issuer i_k proceeds as follows:
     (a) The issuer samples $a_k, t_k \leftarrow \mathbb{Z}_p$, $y_k \leftarrow \mathbb{Z}_p^*$, and returns the first message $(A_k, C_k) \coloneqq (g^{a_k}, g^{t_k} \cdot Z^{y_k})$ to $\mathcal{A}$.
     (b) $\mathcal{A}$ responds with an arbitrary challenge $c'_k \in \mathbb{Z}_p^*$.[9]
     (c) The issuer sets $s_k \coloneqq a_k + c'_k \cdot y_k \cdot \mathsf{sk}_{i_k}$ and returns the second message (s_k, y_k, t_k) to $\mathcal{A}$.
   - The random oracle for the hash function H'. Importantly, $\mathcal{A}$ has to provide AGM representations of any group elements in its queries.

[9] In the original protocol description [58], the challenge can be an arbitrary $\mathbb{Z}_p$ element and the issuer aborts if it is zero. W.l.o.g., we assume that the challenge space is $\mathbb{Z}_p^*$.

2. $\mathcal{A}$ outputs $\mathcal{P} = \{\mathsf{pk}_i\}_{i\in I_\mathcal{P}}, m, \big(\widetilde{\mathsf{pk}}, \tilde{\sigma}, (R_i, c_i, z_i)_{i\in I_\mathcal{P}}\big)$, where $I_\mathcal{P} \subseteq [\ell]$ is an index set, along with AGM representations $\boldsymbol{\alpha}, (\boldsymbol{\beta}_i)_{i\in I_\mathcal{P}}$ such that

$$\widetilde{\mathsf{pk}} = g^{\alpha_1} \cdot Z^{\alpha_2} \cdot \prod_{i\in[\ell]} \mathsf{pk}_i^{\alpha_{2+i}} \cdot \prod_{k\in[I]} \big(A_k^{\alpha_{2+\ell+k}} \cdot C_k^{\alpha_{2+\ell+I+k}}\big),$$

$$\forall j \in I_\mathcal{P} : R_j = g^{\beta_{j,1}} \cdot Z^{\beta_{j,2}} \cdot \prod_{i\in[\ell]} \mathsf{pk}_i^{\beta_{j,2+i}} \cdot \prod_{k\in[I]} \big(A_k^{\beta_{j,2+\ell+k}} \cdot C_k^{\beta_{j,2+\ell+I+k}}\big),$$

where ℓ is the number of key-generation queries made by $\mathcal{A}$ and I is the number of times $\mathcal{A}$ ran the issuance protocol. If $\mathcal{A}$ succeeds, then in particular:

$$H'\big(\widetilde{\mathsf{pk}}, (\mathsf{pk}_i, R_i)_{i\in I_\mathcal{P}}\big) = \sum_{i\in I_\mathcal{P}} c_i, \tag{7}$$

$$\forall j \in I_\mathcal{P} : g^{z_j} = R_j \cdot (\widetilde{\mathsf{pk}}/\mathsf{pk}_j)^{c_j}. \tag{8}$$

Without loss of generality, we make the following simplifying assumptions:

(a) $\mathcal{A}$ makes every query to H' exactly once.

(b) $\mathcal{A}$ queries H' on $\big(\widetilde{\mathsf{pk}}, (\mathsf{pk}_i, R_i)_{i\in I_\mathcal{P}}\big)$ before returning its final output, and the AGM representations accompanying this query are consistent with $\boldsymbol{\alpha}, (\boldsymbol{\beta}_i)_{i\in I_\mathcal{P}}$, i.e., they match. This is w.l.o.g. because if $\mathcal{A}$ outputs an inconsistent representation $\boldsymbol{\alpha}' \neq \boldsymbol{\alpha}$ for $\widetilde{\mathsf{pk}}$ (or analogously for any R_j), both representations must evaluate to the same group element. This yields a non-trivial discrete-logarithm relation among the basis elements, which can be turned into a formal reduction to the discrete-logarithm problem.

(c) $\mathcal{A}$ completes all executions of the issuance protocol, i.e., it sends a response $c'_k \in \mathbb{Z}_p^*$ for each $k \in [I]$. Let $(A_k, C_k, c'_k, s_k, y_k, t_k)_{k\in[I]}$ be the resulting protocol transcripts. This is w.l.o.g. since for any adversary $\mathcal{A}'$ that does not finish all of its issuance protocol executions there is an $\mathcal{A}$ running in roughly the same time such that $\mathsf{Adv}^{\mathsf{ih\text{-}ac\text{-}uf}}_{\mathcal{A},\Pi} = \mathsf{Adv}^{\mathsf{ih\text{-}ac\text{-}uf}}_{\mathcal{A}',\Pi}$: $\mathcal{A}$ behaves exactly as $\mathcal{A}'$, except that before returning the output of $\mathcal{A}'$, it completes all unfinished issuance protocol executions by running the honest user algorithm on arbitrary distinct messages that are different from the messages output by $\mathcal{A}'$. This way, it obtains honestly generated tokens that it can use together with the output of $\mathcal{A}'$ to succeed in its experiment if and only if $\mathcal{A}'$ succeeds.

(d) $\mathcal{A}$ ignores the basis elements $(A_k, C_k)_{k\in[I]}$ in its representations. That is, for all $k \in [I]$, the components $2+\ell+k$ and $2+\ell+I+k$ of $\boldsymbol{\alpha}$ and each $\boldsymbol{\beta}_i$ are zero. For example, the representation of $\widetilde{\mathsf{pk}}$ simplifies to $g^{\alpha_1} \cdot Z^{\alpha_2} \cdot \prod_{i\in[\ell]} \mathsf{pk}_i^{\alpha_{2+i}}$. This is w.l.o.g. since, by the previous assumption, $\mathcal{A}$ knows elements (c'_k, s_k, y_k, t_k) such that $A_k = g^{s_k} \cdot \mathsf{pk}_{i_k}^{-c'_k y_k}$ and $C_k = g^{t_k} \cdot Z^{y_k}$ for all $k \in [I]$, i.e., it can represent them in terms of the other basis elements.

Let Π denote the IHAT-TZ scheme where the ZKPoK is instantiated with Std-f. Let $\mathsf{Adv}^{\mathsf{dl}}_{\mathcal{A},\mathbb{G}}$ denote the probability that adversary $\mathcal{A}$ solves the discrete-logarithm problem in $\mathbb{G}$. We prove the following theorem, establishing that the classical OR proof for our IHAT-TZ scheme is straight-line extractable in the AGM+ROM under the discrete-logarithm assumption.

Theorem 7 (Straight-Line Extractability of Std-f). *Model the hash function H' as a random oracle. Let $\mathcal{A}$ be an algebraic adversary against issuer-hiding one-more unforgeability of Π that makes at most $q_{H'}$ queries to H'. Then there exists a straight-line extractor $\mathcal{E}$ and an adversary $\mathcal{B}$ against the discrete-logarithm problem in $\mathbb{G}$ running in roughly the same time as $\mathcal{A}$ such that*

$$\Pr\big[\mathcal{A} \text{ succeeds } \wedge\ \mathcal{E} \text{ fails to extract}\big] \leq 2 \cdot \mathsf{Adv}^{\mathsf{dl}}_{\mathcal{B},\mathbb{G}} + \frac{2q_{H'}+1}{p}.$$

Proof. Fix an algebraic adversary $\mathcal{A}$ against issuer-hiding one-more unforgeability of the IHAT-TZ scheme. We define the following extractor $\mathcal{E}$:

1. Given $\mathcal{A}$'s output $\mathcal{P} = \{\mathsf{pk}_i\}_{i\in I_\mathcal{P}}, m, (\widetilde{\mathsf{pk}}, \tilde{\sigma}, (R_i, c_i, z_i)_{i\in I_\mathcal{P}})$ and its AGM representations $\boldsymbol{\alpha}, (\boldsymbol{\beta}_i)_{i\in I_\mathcal{P}}$, find the first index $i^* \in I_\mathcal{P}$ such that
$$\widetilde{\mathsf{pk}} = g^{\alpha_1} \cdot \mathsf{pk}_{i^*}.$$
2. If no such index exists, abort. Otherwise, output the witness (α_1, i^*).

Note that $\mathcal{E}$ is straight-line since it only needs the input and output from a single run of the adversary $\mathcal{A}$. If $\mathcal{E}$ does not abort, then it always extracts a valid witness. In particular, this means that $\mathcal{A}$'s AGM representation satisfies $\alpha_2 = 0$, $\alpha_{2+i^*} = 1$, and $\alpha_{2+i} = 0$ for all $i \in [\ell] \setminus \{i^*\}$.

We consider a sequence of experiments $\mathbf{Expt}_0, \ldots, \mathbf{Expt}_4$, and let p_i be the probability that $\mathcal{A}$ succeeds in $\mathbf{Expt}_i$ while the extractor $\mathcal{E}$ fails to produce a valid witness and aborts, i.e., $p_i \stackrel{\text{def}}{=} \Pr\big[\mathcal{A} \text{ succeeds in } \mathbf{Expt}_i \wedge \mathcal{E} \text{ fails to extract}\big]$.

$\mathbf{Expt}_0$. This is the original issuer-hiding one-more unforgeability experiment (cf. Definition 7) with the hash function H' modeled as a random oracle. At the end of its execution, $\mathcal{A}$ outputs $\mathcal{P} = \{\mathsf{pk}_i\}_{i\in I_\mathcal{P}}, m, \big(\widetilde{\mathsf{pk}}, \tilde{\sigma}, (R_i, c_i, z_i)_{i\in I_\mathcal{P}}\big)$ along with AGM representations $\boldsymbol{\alpha}, (\boldsymbol{\beta}_i)_{i\in I_\mathcal{P}}$ such that

$$\widetilde{\mathsf{pk}} = g^{\alpha_1} \cdot Z^{\alpha_2} \cdot \prod_{i\in[\ell]} \mathsf{pk}_i^{\alpha_{2+i}},$$
$$\forall j \in I_\mathcal{P} : R_j = g^{\beta_{j,1}} \cdot Z^{\beta_{j,2}} \cdot \prod_{i\in[\ell]} \mathsf{pk}_i^{\beta_{j,2+i}}.$$

If $\mathcal{A}$ succeeds, then Equations (7) and (8) hold. Substituting the above terms, Equation (8) can be rewritten as

$$\forall j \in I_\mathcal{P}:$$
$$g^{z_j} = g^{\beta_{j,1}+c_j\alpha_1} \cdot Z^{\beta_{j,2}+c_j\alpha_2} \cdot \mathsf{pk}_j^{\beta_{j,2+j}+c_j(\alpha_{2+j}-1)} \cdot \prod_{\substack{i\in[\ell]\\ i\neq j}} \mathsf{pk}_i^{\beta_{j,2+i}+c_j\alpha_{2+i}}. \quad (9)$$

$\mathbf{Expt}_1$. This experiment is identical to $\mathbf{Expt}_0$, except that we further restrict the adversary's winning condition: $\mathcal{A}$ only succeeds if

$$\forall j \in I_\mathcal{P} : \beta_{j,2} + c_j\alpha_2 = 0.$$

We have

$$\begin{aligned} p_0 &\le p_1 + \Pr\big[\mathcal{A} \text{ succeeds in } \mathbf{Expt}_0 \ \wedge\ \mathcal{E} \text{ fails to extract } \wedge\ \exists j^* : \beta_{j^*,2} + c_{j^*}\alpha_2 \neq 0\big] \\ &\le p_1 + \Pr\big[\mathcal{A} \text{ succeeds in } \mathbf{Expt}_0 \ \wedge\ \exists j^* : \beta_{j^*,2} + c_{j^*}\alpha_2 \neq 0\big]. \end{aligned}$$

Next, we construct an adversary $\mathcal{B}_1$ against the discrete-logarithm problem in $\mathbb{G}$ such that $\Pr\big[\mathcal{A} \text{ succeeds in } \mathbf{Expt}_0 \ \wedge\ \exists j^* : \beta_{j^*,2} + c_{j^*}\alpha_2 \neq 0\big] \le \mathsf{Adv}^{\mathsf{dl}}_{\mathcal{B}_1,\mathbb{G}}$, establishing

$$p_0 \le p_1 + \mathsf{Adv}^{\mathsf{dl}}_{\mathcal{B}_1,\mathbb{G}}. \tag{10}$$

On input $(g, Z) \in \mathbb{G}^2$, $\mathcal{B}_1$ simulates the oracles for $\mathcal{A}(g, Z)$ as follows:

- It honestly simulates the random oracle $H' \colon \{0,1\}^* \to \mathbb{Z}_p$.
- $\mathcal{O}\mathsf{KeyGen}$: On the ith query, $\mathcal{B}_1$ runs $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{KeyGen}$ and returns pk_i to $\mathcal{A}$.
- When $\mathcal{A}$ runs the ith execution of the issuance protocol with the jth issuer, $\mathcal{B}_1$ acts as an honest issuer. In particular, the first message it returns to $\mathcal{A}$ is $(A_i, C_i) \coloneqq (g^{a_i}, g^{t_i} Z^{y_i})$ for $a_i, t_i \leftarrow \mathbb{Z}_p$ and $y_i \leftarrow \mathbb{Z}_p^*$.

Finally, when $\mathcal{A}$ outputs $\mathcal{P}, m, \big(\widetilde{\mathsf{pk}}, \tilde{\sigma}, (R_i, c_i, z_i)_{i \in I_{\mathcal{P}}}\big)$ and AGM representations $\boldsymbol{\alpha}, (\boldsymbol{\beta}_i)_{i \in I_{\mathcal{P}}}$, $\mathcal{B}_1$ aborts if $\beta_{j,2} + c_j\alpha_2 = 0$ for all $j \in I_{\mathcal{P}}$. Otherwise, let j^* be such that $\beta_{j^*,2} + c_{j^*}\alpha_2 \neq 0$. $\mathcal{B}_1$ outputs

$$\log_g Z \coloneqq \frac{\begin{pmatrix} z_{j^*} - \beta_{j^*,1} - c_{j^*}\alpha_1 - (\beta_{j^*,2+j^*} + c_{j^*}(\alpha_{2+j^*} - 1)) \cdot \mathsf{sk}_{j^*} \\ - \sum_{i \in [\ell], i \neq j^*} (\beta_{j^*,2+i} + c_{j^*}\alpha_{2+i}) \cdot \mathsf{sk}_i \end{pmatrix}}{\beta_{j^*,2} + c_{j^*}\alpha_2}.$$

Clearly, $\mathcal{B}_1$ perfectly simulates $\mathbf{Expt}_0$ for $\mathcal{A}$. Next, we argue that if $\mathcal{A}$ succeeds and $\beta_{j^*,2} + c_{j^*}\alpha_2 \neq 0$, then $\mathcal{B}_1$ indeed computes the discrete logarithm of its challenge Z. $\mathcal{A}$ succeeding means that Equation (9) holds; in particular:

$$g^{z_{j^*}} = g^{\beta_{j^*,1} + c_{j^*}\alpha_1} \cdot Z^{\beta_{j^*,2} + c_{j^*}\alpha_2} \cdot \mathsf{pk}_{j^*}^{\beta_{j^*,2+j^*} + c_{j^*}(\alpha_{2+j^*} - 1)} \cdot \prod_{\substack{i \in [\ell] \\ i \neq j^*}} \mathsf{pk}_i^{\beta_{j^*,2+i} + c_{j^*}\alpha_{2+i}}.$$

From this, we obtain

$$\begin{aligned} &(\beta_{j^*,2} + c_{j^*}\alpha_2) \cdot \log_g(Z) \\ &= z_{j^*} - \beta_{j^*,1} - c_{j^*}\alpha_1 - (\beta_{j^*,2+j^*} + c_{j^*}(\alpha_{2+j^*} - 1)) \cdot \mathsf{sk}_{j^*} \\ &\quad - \sum_{i \in [\ell], i \neq j^*} (\beta_{j^*,2+i} + c_{j^*}\alpha_{2+i}) \cdot \mathsf{sk}_i, \end{aligned}$$

which shows that $\mathcal{B}_1$'s output is correct if $\beta_{j^*,2} + c_{j^*}\alpha_2 \neq 0$.

$\mathbf{Expt}_2$. This experiment is identical to $\mathbf{Expt}_1$, except that the adversary additionally needs to satisfy $\alpha_2 = 0$ to succeed. Thus, we have

$$\begin{aligned} p_1 &\le p_2 + \Pr\big[\mathcal{A} \text{ succeeds in } \mathbf{Expt}_1 \ \wedge\ \mathcal{E} \text{ fails to extract } \wedge\ \alpha_2 \neq 0\big] \\ &\le p_2 + \Pr\big[\forall j \in I_{\mathcal{P}} : \beta_{j,2} + c_j\alpha_2 = 0 \ \wedge\ \alpha_2 \neq 0\big] \\ &\le p_2 + \Pr\big[\textstyle\sum_{j \in I_{\mathcal{P}}} (\beta_{j,2} + c_j\alpha_2) = 0 \ \wedge\ \alpha_2 \neq 0\big] \\ &= p_2 + \Pr\big[\textstyle\sum_{j \in I_{\mathcal{P}}} \beta_{j,2} + c\alpha_2 = 0 \ \wedge\ \alpha_2 \neq 0\big], \end{aligned}$$

where $c = H'\big(\widetilde{\mathsf{pk}}, (\mathsf{pk}_i, R_i)_{i\in I_{\mathcal{P}}}\big)$ and we used that $c = \sum_{i\in I_{\mathcal{P}}} c_i$ in the final step. We will argue that $\Pr\big[\sum_{j\in I_{\mathcal{P}}} \beta_{j,2} + c\alpha_2 = 0 \ \wedge\ \alpha_2 \neq 0\big] \leq \frac{q_{H'}}{p}$, showing that

$$p_1 \leq p_2 + \frac{q_{H'}}{p}. \tag{11}$$

Note that $\mathcal{A}$ must provide the AGM representations $\alpha_2, (\beta_{j,2})_{j\in I_{\mathcal{P}}}$ when making the query $H'\big(\widetilde{\mathsf{pk}}, (\mathsf{pk}_i, R_i)_{i\in I_{\mathcal{P}}}\big)$. Therefore, $c \in \mathbb{Z}_p$ was chosen uniformly at random *after* the values $\alpha_2, (\beta_{j,2})_{j\in I_{\mathcal{P}}}$ are fixed, and the probability that $\sum_{j\in I_{\mathcal{P}}} \beta_{j,2} + c\alpha_2 = 0$ for $\alpha_2 \neq 0$ is at most $\frac{q_{H'}}{p}$ (over at most $q_{H'}$ queries to H').

$\mathbf{Expt}_3$. This experiment is identical to $\mathbf{Expt}_2$, except that for $\mathcal{A}$ to succeed, the following additional condition must hold:

$$\begin{aligned}&\forall j \in I_{\mathcal{P}}\colon\\ &\beta_{j,2+j} + c_j(\alpha_{2+j} - 1) = 0 \ \wedge\ \forall i \in [\ell], i \neq j : \beta_{j,2+i} + c_j\alpha_{2+i} = 0.\end{aligned}$$

We denote the event that this condition holds by Cancel. It holds that

$$\begin{aligned}p_2 &\leq p_3 + \Pr\big[\mathcal{A} \text{ succeeds in } \mathbf{Expt}_2 \ \wedge\ \mathcal{E} \text{ fails to extract} \ \wedge\ \overline{\mathsf{Cancel}}\big]\\ &\leq p_3 + \Pr\big[\mathcal{A} \text{ succeeds in } \mathbf{Expt}_2 \ \wedge\ \overline{\mathsf{Cancel}}\big].\end{aligned}$$

We proceed by constructing a discrete-logarithm adversary $\mathcal{B}_2$ such that $\Pr\big[\mathcal{A} \text{ succeeds in } \mathbf{Expt}_2 \ \wedge\ \overline{\mathsf{Cancel}}\big] \leq \mathsf{Adv}^{\mathsf{dl}}_{\mathcal{B}_2,\mathbb{G}} + \frac{1}{p}$, showing that

$$p_2 \leq p_3 + \mathsf{Adv}^{\mathsf{dl}}_{\mathcal{B}_2,\mathbb{G}} + \frac{1}{p}. \tag{12}$$

On input $(g, X) \in \mathbb{G}^2$, $\mathcal{B}_2$ chooses $\zeta \leftarrow \mathbb{Z}_p$, sets $Z \coloneqq g^{\zeta}$, and runs $\mathcal{A}(g, Z)$ simulating its oracles as follows:

- It honestly simulates the random oracle $H'\colon \{0,1\}^* \to \mathbb{Z}_p$.
- $\mathcal{O}\mathsf{KeyGen}$: On the ith query, $\mathcal{B}_2$ samples $\gamma_i, \delta_i \leftarrow \mathbb{Z}_p$ and returns $\mathsf{pk}_i \coloneqq X^{\gamma_i} \cdot g^{\delta_i}$ to $\mathcal{A}$.
- When $\mathcal{A}$ runs the ith execution of the issuance protocol with the kth issuer:
  1. Sample $s_i, t'_i \leftarrow \mathbb{Z}_p$, $y'_i \leftarrow \mathbb{Z}_p^*$, and return the first issuer message $(A_i, C_i) \coloneqq (g^{s_i} \cdot \mathsf{pk}_k^{-y'_i}, g^{t'_i})$ to $\mathcal{A}$.
  2. Upon receiving the first user message $c_i \in \mathbb{Z}_p^*$ from $\mathcal{A}$, set $y_i \coloneqq y'_i/c_i$, $t_i \coloneqq t'_i - y_i \cdot \zeta$, and return the second issuer message (s_i, y_i, t_i) to $\mathcal{A}$.

Finally, when $\mathcal{A}$ outputs $\mathcal{P}, m, \big(\widetilde{\mathsf{pk}}, \tilde{\sigma}, (R_i, c_i, z_i)_{i\in I_{\mathcal{P}}}\big)$ and AGM representations $\boldsymbol{\alpha}, (\boldsymbol{\beta}_i)_{i\in I_{\mathcal{P}}}$, $\mathcal{B}_2$ aborts if the event Cancel occurs. Otherwise, let j^* be such that $\beta_{j^*,2+j^*} + c_{j^*}(\alpha_{2+j^*} - 1) \neq 0$ or $\beta_{j^*,2+i} + c_{j^*}\alpha_{2+i} \neq 0$ for some $i \in [\ell] \setminus \{j^*\}$. If the following denominator is zero, $\mathcal{B}_2$ aborts; else it outputs

$$\log_g X \coloneqq \frac{\begin{pmatrix} z_{j^*} - \beta_{j^*,1} - c_{j^*}\alpha_1 - \delta_{j^*}(\beta_{j^*,2+j^*} + c_{j^*}(\alpha_{2+j^*} - 1))\\ - \sum_{i\in[\ell], i\neq j^*} \delta_i(\beta_{j^*,2+i} + c_{j^*}\alpha_{2+i})\end{pmatrix}}{\gamma_{j^*}(\beta_{j^*,2+j^*} + c_{j^*}(\alpha_{2+j^*} - 1)) + \sum_{i\in[\ell], i\neq j^*} \gamma_i(\beta_{j^*,2+i} + c_{j^*}\alpha_{2+i})}.$$

Note that $\mathcal{B}_2$ perfectly simulates $\mathbf{Expt}_2$ for $\mathcal{A}$. The honest distribution of issuer messages w.r.t. a public key pk and user message $c \in \mathbb{Z}_p^*$ is the same as sampling $A, C \leftarrow \mathbb{G}$, $y \leftarrow \mathbb{Z}_p^*$, and then setting $s := \log_g(A) + cy \cdot \log_g(\mathsf{pk})$ and $t := \log_g(C) - y \cdot \log_g(Z)$. This is equivalent to choosing $C \leftarrow \mathbb{G}$, $s \leftarrow \mathbb{Z}_p$, $y' \leftarrow \mathbb{Z}_p^*$, and then setting $y := y'/c$, $A := g^s \cdot \mathsf{pk}^{-cy}$, and $t := \log_g(C) - y \cdot \log_g(Z)$, as done by $\mathcal{B}_2$.

Next, we argue that if $\mathcal{A}$ succeeds and Cancel does not occur, then $\mathcal{B}_2$ indeed computes the discrete logarithm of its challenge X. If $\mathcal{A}$ succeeds in $\mathbf{Expt}_2$, Equation (9) holds and $\beta_{j^*,2} + c_{j^*}\alpha_2 = 0$. Substituting $\mathsf{pk}_i = X^{\gamma_i} \cdot g^{\delta_i}$, we have:

$$\begin{aligned}
g^{z_{j^*}} &= g^{\beta_{j^*,1} + c_{j^*}\alpha_1} \cdot \mathsf{pk}_{j^*}^{\beta_{j^*,2+j^*} + c_{j^*}(\alpha_{2+j^*} - 1)} \cdot \prod_{\substack{i \in [\ell] \\ i \neq j^*}} \mathsf{pk}_i^{\beta_{j^*,2+i} + c_{j^*}\alpha_{2+i}} \\
&= g^{\beta_{j^*,1} + c_{j^*}\alpha_1} \cdot (X^{\gamma_{j^*}} g^{\delta_{j^*}})^{\beta_{j^*,2+j^*} + c_{j^*}(\alpha_{2+j^*} - 1)} \cdot \prod_{\substack{i \in [\ell] \\ i \neq j^*}} (X^{\gamma_i} g^{\delta_i})^{\beta_{j^*,2+i} + c_{j^*}\alpha_{2+i}} \\
&= g^{\beta_{j^*,1} + c_{j^*}\alpha_1 + \delta_{j^*}(\beta_{j^*,2+j^*} + c_{j^*}(\alpha_{2+j^*} - 1)) + \sum_{i \in [\ell], i \neq j^*} \delta_i(\beta_{j^*,2+i} + c_{j^*}\alpha_{2+i})} \\
&\quad \cdot X^{\gamma_{j^*}(\beta_{j^*,2+j^*} + c_{j^*}(\alpha_{2+j^*} - 1)) + \sum_{i \in [\ell], i \neq j^*} \gamma_i(\beta_{j^*,2+i} + c_{j^*}\alpha_{2+i})}.
\end{aligned}$$

From this, we obtain

$$\begin{aligned}
&\left(\gamma_{j^*}(\beta_{j^*,2+j^*} + c_{j^*}(\alpha_{2+j^*} - 1)) + \sum_{i \in [\ell], i \neq j^*} \gamma_i(\beta_{j^*,2+i} + c_{j^*}\alpha_{2+i}) \right) \cdot \log_g(X) \\
&= z_{j^*} - \beta_{j^*,1} - c_{j^*}\alpha_1 - \delta_{j^*}(\beta_{j^*,2+j^*} + c_{j^*}(\alpha_{2+j^*} - 1)) \\
&\quad - \sum_{i \in [\ell], i \neq j^*} \delta_i(\beta_{j^*,2+i} + c_{j^*}\alpha_{2+i}).
\end{aligned}$$

Since the blinding factors g^{δ_i} perfectly hide all the γ_i from $\mathcal{A}$, the probability that $\gamma_{j^*}(\beta_{j^*,2+j^*} + c_{j^*}(\alpha_{2+j^*} - 1)) + \sum_{i \in [\ell], i \neq j^*} \gamma_i(\beta_{j^*,2+i} + c_{j^*}\alpha_{2+i}) = 0$ is at most $1/p$ (note that $\overline{\mathsf{Cancel}}$ means that the other terms are not all zero). Thus, $\mathcal{B}_2$'s output is correct in this case, except with probability $1/p$.

$\mathbf{Expt}_4$. This experiment is identical to $\mathbf{Expt}_3$, except that the adversary only succeeds if there is an index $i^* \in I_{\mathcal{P}}$ such that

$$\alpha_{2+i^*} = 1 \ \wedge \ \forall i \in [\ell], i \neq i^* : \alpha_{2+i} = 0.$$

Denoting this event as Zero, we have

$$\begin{aligned}
p_3 &\leq p_4 + \Pr\left[\mathcal{A} \text{ succeeds in } \mathbf{Expt}_3 \ \wedge \ \mathcal{E} \text{ fails to extract } \ \wedge \ \overline{\mathsf{Zero}}\right] \\
&\leq p_4 + \Pr\left[\mathsf{Cancel} \ \wedge \ \overline{\mathsf{Zero}}\right],
\end{aligned}$$

where Cancel is the event that, for all $j \in I_{\mathcal{P}}$, we have $\beta_{j,2+j} + c_j(\alpha_{2+j} - 1) = 0$ and $\beta_{j,2+i} + c_j\alpha_{2+i} = 0$ for every $i \in [\ell] \setminus \{j\}$. A similar argument as in $\mathbf{Expt}_2$

shows that $\Pr\big[\mathsf{Cancel} \wedge \overline{\mathsf{Zero}}\big] \leq \frac{q_{H'}}{p}$, which means

$$p_3 \leq p_4 + \frac{q_{H'}}{p}. \tag{13}$$

In summary, if $\mathcal{A}$ succeeds in $\mathbf{Expt}_4$ then there exists an index $i^* \in I_{\mathcal{P}}$ such that $\alpha_2 = 0$, $\alpha_{2+i^*} = 1$, and $\alpha_{2+i} = 0$ for all $i \in [\ell] \setminus \{i^*\}$. Thus, $\widetilde{\mathsf{pk}} = g^{\alpha_1} \cdot \mathsf{pk}_{i^*}$, meaning $\mathcal{E}$ does not abort and successfully extracts a valid witness (α_1, i^*). This implies that $p_4 = \Pr\big[\mathcal{A} \text{ succeeds in } \mathbf{Expt}_4 \ \wedge \ \mathcal{E} \text{ fails to extract}\big] = 0$, which combined with Equations (10) to (13) yields

$$p_0 \leq \mathsf{Adv}^{\mathsf{dl}}_{\mathcal{B}_1,\mathbb{G}} + \mathsf{Adv}^{\mathsf{dl}}_{\mathcal{B}_2,\mathbb{G}} + \frac{2q_{H'}+1}{p}.$$

Finally, consider the discrete-logarithm adversary $\mathcal{B}$ that runs either $\mathcal{B}_1$ or $\mathcal{B}_2$ with probability $1/2$ each. For this adversary, it holds that

$$\mathsf{Adv}^{\mathsf{dl}}_{\mathcal{B},\mathbb{G}} = \frac{1}{2} \cdot \mathsf{Adv}^{\mathsf{dl}}_{\mathcal{B}_1,\mathbb{G}} + \frac{1}{2} \cdot \mathsf{Adv}^{\mathsf{dl}}_{\mathcal{B}_2,\mathbb{G}},$$

completing the proof. □