Attribute-Based Signatures with User-Controlled Linkability without Random Oracles

Ali El Kaafarani and Essam Ghadafi

University of Oxford, UK,
University of the West of England, Bristol, UK.

Abstract. Attribute-Based Signatures (ABS) are a versatile cryptographic primitive and have many applications. They are a generalization of many widely-used signature-related notions such as group, ring and mesh signatures. Attribute-Based Signatures with User-Controlled Linkability (ABS-UCL) combine properties of ABS and Direct Anonymous Attestation (DAA) thus allowing a user to anonymously and at will maintain a session with a verifier. Such a notion also has many applications. In this work, we provide the first constructions of ABS-UCL dispensing with heuristic assumptions such as random oracles. We start by providing a generic construction which avoids some of the inefficiency pitfalls of existing constructions. We then provide efficient instantiations supporting expressive signing policies. We also give a concrete construction for threshold policies yielding constant-size signatures. Some of the building blocks we construct might be of independent interest.

Keywords. Attribute-Based Signatures, User-Controlled Linkability, Standard Model.

1 Introduction

Attribute-Based Signatures (ABS), introduced by Maji et al. [28], are a promising, versatile primitive that allows signers to authenticate messages while enjoying fine-grained control over identifying information. In ABS, users sign messages w.r.t. policies satisfied by a set of attributes they possess. The verifier of a signature is convinced that a signer with a set of attributes satisfying the policy in question signed the message but learns neither the identity of the signer nor the exact attributes used. Attribute-based signatures are a generalization of many prominent notions such as group [11], ring [31] and mesh [8] signatures. They have many applications including trust negotiation, e.g. [18], attribute-based messaging, e.g. [5], and leaking secrets.

ABS schemes can be categorized according to how expressive the policies they support are. In threshold ABS (tABS), proposed by Shahandashti and Safavi-Naini [33], the signing policy is restricted to proving possession of at least t out of n attributes. Constructions of tABS schemes include [27, 19, 24]. ABS schemes supporting more expressive policies, i.e. monotonic access structures, were first given by Maji et al. [28]. Okamoto and Takashima [29, 30] gave constructions supporting non-monotonic access structures. Note that any scheme supporting monotonic access structures could be extended to support non-monotonic access structures simply by doubling the universe of attributes. ABS schemes for circuits were given by Tang et al. [34] and Sakai et al. [32].

Practical features such as decentralization [30], user-controlled linkability [15], traceability [17, 16, 21], and controllable-linkability [35] have been added to ABS schemes resulting in various ABS notions.

El Kaafarani et al. [15] introduced the notion of Attribute-Based Signatures with User-Controlled Linkability (ABS-UCL), which analogously to the Direct Anonymous Attestation (DAA) protocol [9], which is a standardized protocol that is deployed in practice, it adds the user-controlled linkability feature to standard ABS schemes. More precisely, the user can at her discretion choose to make some of her signatures aimed at a particular verifier linkable without sacrificing her anonymity. Thus, the user-controlled linkability feature allows anonymous users to establish and maintain sessions with particular verifiers. For instance, consider a potential buyer of age-restricted products who wishes to convince the seller that she indeed satisfies the policy in place for buying such products but without revealing her identity. In this scenario the user may also wish to link her current transaction to some of her earlier anonymous ones, e.g. to benefit from discounts. Another useful application of the controlled-linkability feature is resuming interrupted or lost authentication sessions between communicating parties. The more recent notion of DAA with attributes (DAA-A) [12] can also be viewed as a variant of ABS-UCL where the signer is split into a trusted (computationally-constrained) TPM and a more powerful but not necessarily trusted host. Note that unlike in DAA, where the signature attests to the fact that the user belongs to a particular group, and thanks to the expressiveness of the policies with which signatures in ABS schemes are associated, ABS-UCL allows the user to attest to much broader statements than merely proving she belongs to a particular group. Also note that ABS-UCL is very different from the notion of Attribute-Based Signatures with Controllable Linkability [35] in which a designated authority (equipped with a secret key) is able to check if two signatures originated from the same signer.

Traceable ABS (TABS) [17] and Decentralized Traceable ABS (DTABS) schemes [16, 21] add the traceability feature to ABS schemes by granting a designated opener a special tracing key using which she can revoke anonymity when the need arises.

To the best of our knowledge, currently there exist no constructions of anonymous signature schemes offering the user-controlled linkability feature and supporting attributes which do not rely on random oracles. The constructions in [15] as well as the more recent variants in [12] all rely on heuristic assumptions for their security.

Our Contribution. We give a generic construction of ABS-UCL supporting expressive policies, i.e. monotone access structures (and hence non-monotone access structures) and 3 efficient concrete constructions. The first two are instantiations of the generic construction whereas the third construction is a concrete one supporting threshold policies and yielding constant-size signatures. Our generic construction is tailored towards avoiding some of the inefficiency pitfalls of existing ones. Our instantiations are efficient and constitute the first constructions not relying on random oracles [2] and compare favourably to existing related constructions offering different features. As a special case of our constructions, i.e. when the user-controlled linkability requirement is dropped, we obtain efficient instantiations of standard ABS schemes which compare favourably to existing ones. As a building block for some of our constructions, we construct a new efficient partially structure-preserving signature scheme [22], which might be of independent interest.

Paper Organization. In Section 2, we give some preliminaries. In Section 3, we present the building blocks we use. We define ABS-UCL in Section 4. In Sections 5 and 6, we present our constructions of ABS-UCL.

2 Preliminaries

In this section we present some preliminaries.

Bilinear Groups. A bilinear group is a tuple $\mathcal{P} := (\mathbb{G}, \mathbb{H}, \mathbb{G}_T, p, G, \tilde{H}, e)$ where $\mathbb{G} := \langle G \rangle$, $\mathbb{H} := \langle \tilde{H} \rangle$ and $\mathbb{G}_T$ are groups of a prime order p. The function e is a non-degenerate bilinear map $\mathbb{G} \times \mathbb{H} \longrightarrow \mathbb{G}_T$. We focus on Type-III bilinear groups [20], where $\mathbb{G} \neq \mathbb{H}$ and no efficient isomorphisms between $\mathbb{G}$ and $\mathbb{H}$ are known in either direction, since it is well-known it is more efficient than the other settings.

Intractability Assumptions. We will use the following existing intractability assumptions:

DDH. Given $(G, G^a, G^b, G^c) \in \mathbb{G}^4$ for $a, b, c \leftarrow \mathbb{Z}_p$, where $\mathbb{G} = \langle G \rangle$ is of a prime order p, it is hard to decide whether or not $c = ab \pmod{p}$.

SXDH. This requires that the DDH assumption holds in both groups $\mathbb{G}$ and $\mathbb{H}$.

q-SDH [7]**.** Given $(G, G^x, \ldots, G^{x^q})$ for $x \leftarrow \mathbb{Z}_p$, where $\mathbb{G} = \langle G \rangle$ is of a prime order p, it is hard to output a pair $(c, G^{\frac{1}{x+c}})$, where $c \in \mathbb{Z}_p \backslash \{-x\}$.

q-DDHI [6]**.** Given $(G, G^x, \ldots, G^{x^q})$ for $x \leftarrow \mathbb{Z}_p$, where $\mathbb{G} = \langle G \rangle$ is of a prime order p, it is hard to distinguish $G^{\frac{1}{x}}$ from a random element of $\mathbb{G}$.

(ℓ, m, t)**-aMSE-CDH [25].** Given $\mathcal{P} := (\mathbb{G}, \mathbb{H}, \mathbb{G}_T, p, G, \tilde{H}, e)$, let $\vec{x} = (x_1, \cdots, x_{\ell+m}) \leftarrow (\mathbb{Z}_p^{\times})^{\ell+m}$, $g_1(X) = \prod_{i=1}^{\ell}(X + x_i)$ and $g_2(X) = \prod_{i=\ell+1}^{\ell+m}(X + x_i)$ for some $\ell, m, t \in \mathbb{N}$.

The assumption states that it is infeasible to compute $e(G, \tilde{H})^{\kappa g_1(\gamma)}$ given:

$$G, G^{\gamma}, \ldots, G^{\gamma^{\ell+t-2}}, G^{\kappa\gamma g_1(\gamma)} \quad (1) \qquad G^{\omega\gamma}, G^{\omega\gamma^2}, \ldots, G^{\omega\gamma^{\ell+t-2}} \quad (2)$$

$$G^{\alpha}, G^{\alpha\gamma}, \ldots, G^{\alpha\gamma^{\ell+t}} \quad (3) \qquad \tilde{H}, \tilde{H}^{\gamma}, \ldots, G^{\gamma^{m-2}}, \tilde{H}^{\kappa g_2(\gamma)} \quad (4)$$

$$\tilde{H}^{\omega}, \tilde{H}^{\omega\gamma}, \ldots, G^{\omega\gamma^{m-1}} \quad (5) \qquad \tilde{H}^{\alpha}, \tilde{H}^{\alpha\gamma}, \ldots, G^{\alpha\gamma^{2(m-t)+3}} \quad (6)$$

3 Building Blocks

In this section, we present the building blocks we use; this includes a new partially structure-preserving signature scheme that we construct.

3.1 Partially Structure-Preserving Signature Schemes

Ghadafi [22] defined Partially Structure-Preserving Signature (PSPS) schemes as a variant of structure-preserving signature schemes [1] where the only deviation from the definition of the latter is that some part of the message might be field elements rather than group elements. In this work we will use 2 PSPS schemes the first of which is new.

A New PSPS Scheme for the Message Space $\mathbb{G}^n \times \mathbb{Z}_p^{n'}$. We give here a PSPS scheme for the message space $\mathbb{G}^n \times \mathbb{Z}_p^{n'}$. It is an extension of the single-message structure-preserving scheme of Chatterjee and Menezes [10]. The new scheme is as follows:

- $\mathsf{KeyGen}(\mathcal{P})$: Choose $x_1, \ldots, x_n, y_1, \ldots, y_{n'}, z \leftarrow \mathbb{Z}_p$, $\tilde{X}_i := \tilde{H}^{x_i}, \tilde{Y}_i := \tilde{H}^{y_i}, \tilde{Z} := \tilde{H}^{z}$. Returns $\big(\mathsf{sk} := (x_1, \ldots, x_n, y_1, \ldots, y_{n'}, z), \mathsf{vk} := (\tilde{X}_1, \ldots, \tilde{X}_n, \tilde{Y}_1, \ldots, \tilde{Y}_{n'}, \tilde{Z})\big)$.
- $\mathsf{Sign}\Big(\mathsf{sk}, \big(\vec{U} = (U_1, \ldots, U_n), \vec{m} = (m_1, \ldots, m_{n'})\big) \in \mathbb{G}^n \times \mathbb{Z}_p^{n'}\Big)$: Choose $r \leftarrow \mathbb{Z}_p$. Set $R := G^r$, $\tilde{R} := \tilde{H}^r$, and $S := \prod_{i=1}^{n} U_i^{x_i} \cdot G^{\sum_{i=1}^{n'} m_i y_i + r^2 + z}$. Return $\sigma := \Big(\tilde{R}, R, S\Big) \in \mathbb{H} \times \mathbb{G}^2$.
- $\mathsf{Verify}\Big(\mathsf{vk}, \big(\vec{U} = (U_1, \ldots, U_n), \vec{m} = (m_1, \ldots, m_{n'})\big), \sigma = (\tilde{R}, R, S)\Big)$: Return 1 iff $\tilde{R} \in \mathbb{H}$, $R, S \in \mathbb{G}$, $U_i \in \mathbb{G}$, and the following two equations hold:

  $$e(R, \tilde{H}) = e(G, \tilde{R}) \qquad e(S, \tilde{H}) = \prod_{i=1}^{n} e(U_i, \tilde{X}_i) \prod_{i=1}^{n'} e(G^{m_i}, \tilde{Y}_i) e(R, \tilde{R}) e(G, \tilde{Z})$$

- $\mathsf{Randomize}\Big(\mathsf{vk}, \big(\vec{U} = (U_1, \ldots, U_n), \vec{m} = (m_1, \ldots, m_{n'})\big), \sigma = (\tilde{R}, R, S)\Big)$: Choose $r' \leftarrow \mathbb{Z}_p$, Set $R' := R \cdot G^{r'}$, $\tilde{R}' := \tilde{R} \cdot \tilde{H}^{r'}$, and $S' := S \cdot R^{2r'} \cdot G^{r'^2}$. Return $\sigma' := (\tilde{R}', R', S')$.

Correctness of the scheme is easy to verify. The signatures are perfectly randomizable as the distribution of randomized signatures is identical to that of fresh signatures on the same message. We prove the following theorem in Appendix A.

Theorem 1. *The signature scheme is existentially unforgeable against a chosen-message attack in the generic group model.*

PSPS Scheme for a Diffie-Hellman Pair [22]. We also use the recent efficient PSPS scheme by Ghadafi [22] which signs a Diffie-Hellman pair and a vector from $\mathbb{Z}_p^n$. It suffices for our case to have $n = 1$. The scheme from [22] is as follows, where as in [22], we let $\widehat{\mathbb{GH}}$ denote the set of Diffie-Hellman pairs, i.e. $\widehat{\mathbb{GH}} = \{(U, \tilde{V}) | (U, \tilde{V}) \in \mathbb{G} \times \mathbb{H}, e(U, \tilde{H}) = e(G, \tilde{V})\}$.

- $\mathsf{KeyGen}(\mathcal{P})$: Select $x, y_1, \ldots, y_n, z \leftarrow \mathbb{Z}_p^\times$. Set $\tilde{X} := \tilde{H}^x$, $\tilde{Y}_i := \tilde{H}^{y_i}$ for all $i \in [n]$, $\tilde{Z} := \tilde{H}^z$. Set $\mathsf{sk} := (x, y_1, \ldots, y_n, z)$ and $\mathsf{vk} := (\tilde{X}, \tilde{Y}_1, \ldots, \tilde{Y}_n, \tilde{Z})$.
- $\mathsf{Sign}\Big(\mathsf{sk}, \big((U, \tilde{V}), \vec{m} = (m_1, \ldots, m_n)\big)\Big)$: To sign $(U, \tilde{V}) \in \widehat{\mathbb{GH}}$ and a vector $(m_1, \ldots, m_n) \in \mathbb{Z}_p^n$, select $r \leftarrow \mathbb{Z}_p^\times$, and set $R := G^r$, $S := \left(U^r \cdot G^{r(x + \sum_{i=1}^n m_i y_i)}\right)^{\frac{1}{z}}$. Return $\sigma := (R, S) \in \mathbb{G}^2$.
- $\mathsf{Verify}\Big(\mathsf{vk}, \big((U, \tilde{V}), \vec{m}\big), \sigma = (R, S)\Big)$: Return 1 iff $R \in \mathbb{G}^\times$, $(U, \tilde{V}) \in \widehat{\mathbb{GH}}$, and $e(S, \tilde{Z}) = e(R, \tilde{V}) e(R, \tilde{X}) \prod_{i=1}^{n} e(R, \tilde{Y}_i^{m_i})$.
- $\mathsf{Randomize}\Big(\mathsf{vk}, \big((U, \tilde{V}), \vec{m}\big), \sigma = (R, S)\Big)$: Select $r' \leftarrow \mathbb{Z}_p^\times$, and set $R' := R^{r'}$, $S' := S^{r'}$. Return $\sigma' := (R', S')$.

3.2 Groth-Sahai Proofs

Groth-Sahai (GS) proofs [23] are non-interactive proofs in the CRS model. We will use GS proofs that are secure under the SXDH assumption.

For clarity, when describing the statements to be proven, we will we underline the variables which are part of the witness. The language for these proofs is of the form $\mathcal{L} := \{\mathsf{statement} \mid \exists\, \mathsf{witness} : E(\mathsf{statement}, \mathsf{witness}) \text{ holds } \}$, where $E(\mathsf{statement}, \cdot)$ is one of the four types as described in [23].

The system consists of the algorithms ($\mathsf{GSSetup}$, $\mathsf{GSProve}$, $\mathsf{GSVerify}$, $\mathsf{GSExtract}$, $\mathsf{GSSimSetup}$, $\mathsf{GSSimProve}$).

$\mathsf{GSSetup}$ takes as input the description of a bilinear group $\mathcal{P}$ and outputs a *binding* reference string crs and an extraction key xk. $\mathsf{GSProve}$ takes as input the string crs, a set of equations $\mathsf{statement}$ and a witness, and outputs a proof Ω for the satisfiability of the equations. $\mathsf{GSVerify}$ takes as input a set of equations, a string crs and a proof Ω and outputs 1 if the proof is valid, and 0 otherwise. $\mathsf{GSExtract}$ takes as input a binding crs, the extraction key xk and a valid proof Ω, and outputs the witness used for the proof. $\mathsf{GSSimSetup}$, on input a bilinear group $\mathcal{P}$, outputs a *hiding* string $\mathsf{crs}_{\mathrm{Sim}}$ and a trapdoor key tr that allows to simulate proofs. $\mathsf{GSSimProve}$ takes as input $\mathsf{crs}_{\mathrm{Sim}}$, a statement and the trapdoor tr and produces a simulated proof Ω_{Sim} without a witness.

The system can either be instantiated using a binding CRS crs (produced by $\mathsf{GSSetup}$) or a hiding CRS $\mathsf{crs}_{\mathrm{Sim}}$ (produced by $\mathsf{GSSimSetup}$). The distributions of strings crs and $\mathsf{crs}_{\mathrm{Sim}}$ are computationally indistinguishable and simulated proofs are indistinguishable from real proofs. The proof system has perfect completeness, (perfect) soundness, composable witness-indistinguishability/composable zero-knowledge. Formal definitions are given in Appendix C.

3.3 Linkable Indistinguishable Tag

A Linkable Indistinguishable Tag (LIT) scheme [4, 3] is defined w.r.t. a one-way function PK such that a tag created with a secret key sk can be verified using $\mathsf{PK}(\mathsf{sk})$. LIT consists of the algorithms $(\mathsf{KeyGen}, \mathsf{Tag}, \mathsf{Verify})$. $\mathsf{KeyGen}(1^\lambda)$ produces a secret key sk; $\mathsf{Tag}(\mathsf{sk}, m)$ outputs a tag τ on the message m; $\mathsf{Verify}(\mathsf{PK}(\mathsf{sk}), m, \tau)$ verifies the tag τ on m returning $0/1$ accordingly.

Besides correctness, the security of LIT [4, 3] requires linkability and f-indistinguishability. Informally, the former requires that an adversary who is allowed to control both the secret key and the message cannot produce identical tags unless they are on the same message/key pair. Indistinguishability, which is defined w.r.t. a one-way function f of the secret key, requires that an adversary who gets $f(\mathsf{sk})$ and access to a tag oracle, cannot determine whether or not a new tag on a message of its choice was produced using the same key used by the tag oracle. Formal definitions are given in Appendix **??**.

As in [3], we instantiate LIT in the standard model with the function underlying the weak Boneh-Boyen signature scheme [7]. The instantiation is secure under the q-DDHI assumption.

4 Definition and Security of ABS-UCL

In this section, we define the syntax and security of Attribute-Based Signatures with User-Controlled Linkability (ABS-UCL) [15].

An ABS-UCL scheme consists of the following algorithms [15], where pp output by Setup is an implicit input to the rest of the algorithms:

$\mathsf{Setup}(1^\lambda)$ on input a security parameter, it returns public parameters pp.

$\mathsf{AASetup}(\mathsf{aid})$ is run by attribute authority $\mathsf{AA}_{\mathsf{aid}}$ to generate her public/secret key pair $(\mathsf{vk}_{\mathsf{AA}}, \mathsf{sk}_{\mathsf{AA}})$.

$\mathsf{UKeyGen}(\mathsf{id})$ is run by user id to generate her personal secret key $\mathsf{usk}_{\mathsf{id}}$.

$\mathsf{AttKeyGen}(\mathsf{sk}_{\mathsf{AA}}, \mathsf{id}, f(\mathsf{usk}_{\mathsf{id}}), a)$ is run by attribute authority AA (managing attribute a), where f is an injective one-way function, it gives user id the secret key $\mathsf{sk}_{\mathsf{id},a}$.

$\mathsf{Sign}(\mathsf{usk}_{\mathsf{id}}, \{\mathsf{sk}_{\mathsf{id},a}\}_{a\in\mathcal{A}}, m, \mathbb{P}, \mathsf{recip})$ user id with attributes $\mathcal{A}$ s.t. $\mathbb{P}(\mathcal{A}) = 1$ uses this algorithm to sign a message m w.r.t. signing policy $\mathbb{P}$ and the recipient tag recip. The algorithm returns a signature σ.

$\mathsf{Verify}(\sigma, \{\mathsf{vk}_{\mathsf{AA}_i}\}_i, \mathbb{P}, m, \mathsf{recip})$ checks if the signature σ is valid on the message m w.r.t. (the possibly empty) recipient tag recip and the policy $\mathbb{P}$ returning $1/0$.

$\mathsf{Link}(\sigma_0, m_0, \{\mathsf{vk}_{\mathsf{AA}_i}\}_i, \mathbb{P}_0, \sigma_1, m_1, \{\mathsf{vk}_{\mathsf{AA}_j}\}_j, \mathbb{P}_1, \mathsf{recip})$ checks if the two signatures on their respective messages and w.r.t. $\mathsf{recip} \neq \perp$ and their respective signing policies were produced by the same user, outputting $0/1$ accordingly.

$\mathsf{Identify}(\mathsf{sk}, \sigma, m, \mathsf{recip}, \{\mathsf{vk}_{\mathsf{AA}_i}\}_i, \mathbb{P})$ is only used in the security model for capturing linkability. It checks whether the valid signature σ (w.r.t. the signing policy $\mathbb{P}$) on the message m and $\mathsf{recip} \neq \perp$ was produced by the secret key sk, outputting $0/1$ accordingly.

Security Requirements. Besides correctness, the security of ABS-UCL [15] requires unforgeability, linkability and anonymity which we define below.

UNFORGEABILITY. This property guarantees that users cannot output signatures on (message, recipient tag) pairs w.r.t a signing policy that is not satisfied by their set of attributes, even if they collude, ensuring collusion-resistance. It also ensures that an adversary cannot produce a signature which links to a signature by an honest user even if everyone else in the system is corrupt.

Definition 1 (Unforgeability). *An* ABS-UCL *scheme is unforgeable if for all security parameters $\lambda \in \mathbb{N}$, for all PPT adversaries the advantage in winning the following game is negligible:*

Setup: *The challenger runs* Setup *and gives* pp *to the adversary.*
Play: *The adversary can ask for attribute authorities to be created and get hold of their secret keys. She can also ask for honest users to be created and get hold of their personal secret keys. Moreover, the adversary can ask for keys for attributes for users and signatures on tuples $(m, \mathbb{P}, \mathsf{recip})$ of her choice on behalf of honest users.*
Output: *The adversary outputs either of the following:*

- ⋆ *A valid signature σ on m and* recip *w.r.t. $\mathbb{P}$, where $(m, \mathsf{recip}, \mathbb{P})$ was not queried to the signing oracle, and there exists no subset of attributes $\mathcal{A}^*$ whose keys have been revealed to the adversary or managed by corrupt attribute authorities such that $\mathbb{P}(\mathcal{A}^*) = 1$.*
- ⋆ *A tuple $(m_0, \sigma_0, \{\mathsf{vk}_{\mathsf{AA}_i}\}_i, \mathbb{P}_0, m_1, \sigma_1, \{\mathsf{vk}_{\mathsf{AA}_j}\}_j, \mathbb{P}_1, \mathsf{recip} \neq \perp, \mathsf{id})$, where σ_i is valid on m_i and* recip *w.r.t. $\mathbb{P}_i$, user* id *is honest,* $\mathsf{Link}(\sigma_0, m_0, \{\mathsf{vk}_{\mathsf{AA}_i}\}_i, \mathbb{P}_0, \sigma_1, m_1, \{\mathsf{vk}_{\mathsf{AA}_j}\}_j, \mathbb{P}_1, \mathsf{recip}) = 1$ *and either* $(\mathsf{id}, m_0, \mathsf{recip}, \mathbb{P}_0)$ *or* $(\mathsf{id}, m_1, \mathsf{recip}, \mathbb{P}_1)$ *was not queried to the signing oracle.*

LINKABILITY. This property ensures that only valid signatures directed at the same recipient and which were produced by the same user link.

Definition 2 (Linkability). *An* ABS-UCL *scheme is linkable if for all security parameters $\lambda \in \mathbb{N}$, for all PPT adversaries the advantage in winning the following game is negligible:*

Setup: *The challenger runs* Setup *and gives* pp *to the adversary.*
Play: *The adversary can choose all the secret keys of all users and attribute authorities.*
Output: *The adversary outputs* $(\sigma_1, \mathsf{recip}_1, m_1, \{\mathsf{vk}_{\mathsf{AA}_i}\}_i, \mathbb{P}_1, \mathsf{sk}_1)$ *and* $(\sigma_2, \mathsf{recip}_2, m_2, \{\mathsf{vk}_{\mathsf{AA}_j}\}_j, \mathbb{P}_2, \mathsf{sk}_2)$. *She wins if σ_i is valid (w.r.t. $\mathbb{P}_i$) on m_i and recip_i, for $i = 1, 2$ and either of the following holds:*

- ⋆ *σ_1 was produced by sk_1 and σ_2 was produced by sk_2 where $\mathsf{sk}_1 = \mathsf{sk}_2$ and $\mathsf{recip} = \mathsf{recip}_1 = \mathsf{recip}_2 \neq \perp$ but* $\mathsf{Link}(\sigma_1, m_1, \{\mathsf{vk}_{\mathsf{AA}_i}\}_i, \mathbb{P}_1, \sigma_2, m_2, \{\mathsf{vk}_{\mathsf{AA}_j}\}_j, \mathbb{P}_2, \mathsf{recip}) = 0$.

- ⋆ σ_1 *was produced by* sk_1 *and* σ_2 *was produced by* sk_2 *where* $\mathsf{sk}_1 = \mathsf{sk}_2$ *and* $\mathsf{Link}(\sigma_1, m_1, \{\mathsf{vk}_{\mathsf{AA}_i}\}_i, \mathbb{P}_1, \sigma_2, m_2, \{\mathsf{vk}_{\mathsf{AA}_j}\}_j, \mathbb{P}_2, \mathsf{recip}_k) = 1$ *for* $k \in \{1, 2\}$ *and either* $\mathsf{recip}_k = \perp$ *or* $\mathsf{recip}_1 \neq \mathsf{recip}_2$.
- ⋆ σ_1 *was produced by* sk_1 *and* σ_2 *was produced by* sk_2 *where* $\mathsf{sk}_1 \neq \mathsf{sk}_2$ *and* $\mathsf{recip} = \mathsf{recip}_1 = \mathsf{recip}_2 \neq \perp$ *and* $\mathsf{Link}(\sigma_1, m_1, \{\mathsf{vk}_{\mathsf{AA}_i}\}_i, \mathbb{P}_1, \sigma_2, m_2, \{\mathsf{vk}_{\mathsf{AA}_j}\}_j, \mathbb{P}_2, \mathsf{recip}) = 1$.

ANONYMITY. This ensures that neither the identity of the signer, nor the attributes used in the signing are revealed by the signature.

Definition 3 (Anonymity). *An* ABS-UCL *scheme is anonymous if for all security parameters* $\lambda \in \mathbb{N}$*, for all PPT adversaries the advantage in winning the following game is negligibly close to* $\frac{1}{2}$*:*

Setup: *The challenger runs* Setup *and gives* pp *to the adversary.*

Play I: *The adversary has full control over all attribute authorities. She can also get hold of the secret keys of signers of her choice; those signers automatically become corrupt users. Moreover, the adversary can get hold of the secret key of any attribute and signatures on tuples* $(m, \mathbb{P}, \mathsf{recip})$ *of her choice on behalf of honest users.*

Challenge: *The adversary outputs* $(m, \mathsf{id}_0, \mathcal{A}_0, \mathsf{id}_1, \mathcal{A}_1, \mathbb{P}, \mathsf{recip})$ *where* $\mathbb{P}(\mathcal{A}_i) = 1$ *for* $i = 0, 1$*. If* $\mathsf{recip} \neq \perp$*, we require that both* id_0 *and* id_1 *are honest users. The challenger sends back a signature* σ_b *generated using* $(\mathsf{id}_b, \mathcal{A}_b)$*, for* $b \leftarrow \{0, 1\}$.

Play II: *Same as in play I with the additional condition that if* $\mathsf{recip} \neq \perp$*, the adversary can corrupt neither* id_0 *nor* id_1.

Output: *The adversary outputs her guess* b^* *and wins if* $b^* = b$.

5 Efficient ABS-UCL Constructions for Expressive Policies

Here we give efficient constructions for monotone access policies. By doubling the attribute space, we also cover non-monotone access policies. Our construction is a modified and improved variant of the generic construction in [15]. We describe the idea of our construction generically and then give specific efficient instantiations. When producing the zero-knowledge proofs part of the signature, we use a span program [26] to represent the signing policy. Refer to [26] for more information about span programs.

The personal secret key of user id is a secret key $\mathsf{usk}_{\mathsf{id}} \in \mathcal{SK}$ for a linkable indistinguishable tag scheme LIT. Let $f : \mathcal{SK} \to \mathcal{F}$ and $\mathsf{PK} : \mathcal{SK} \to \mathcal{PK}$ be two one-way injective functions. In our instantiations, f and PK are exponentiations of $\mathsf{usk}_{\mathsf{id}}$ in groups $\mathbb{G}$ and $\mathbb{H}$, respectively. We make $f(\mathsf{usk}_{\mathsf{id}})$ public whereas $\mathsf{PK}(\mathsf{usk}_{\mathsf{id}})$ is only known to the user.

Since our main goal is to design efficient schemes while dispensing with heuristic assumptions, unlike the generic construction of [15] and other similar constructions of variants of standard-model attribute-based signature schemes, e.g. [16, 21], which do not offer the user-controlled linkability feature, we use an existentially unforgeable randomizable partially structure-preserving signature scheme (RPSPS) [22] to issue attribute credentials to users. Unlike e.g. [16, 21], we have weakened the requirement we need from the signature scheme from being structure-preserving [1] to being partially

structure-preserving [22]. This serves to improve the efficiency of the construction. We also require a second existentially unforgeable signature scheme DS whose public verification key is part of the public parameters of the system whereas its secret key is not known to any party.

The credential $\mathsf{sk}_{\mathsf{id},a}$ for attribute a to user id is a RPSPS signature on $(f(\mathsf{usk}_{\mathsf{id}}), a)$ using $\mathsf{sk}_{\mathsf{AA}_{\mathsf{aid}}}$ which is the secret key of the authority managing a.

To sign a message m w.r.t. a signing policy $\mathbb{P}$ where the signer possesses credentials $\{\mathsf{sk}'_{\mathsf{id},a}\}_{a\in\mathcal{A}}$ for a set of attributes $\mathcal{A}$ satisfying $\mathbb{P}(\mathcal{A}) = 1$, the signer first re-randomizes $\mathsf{sk}'_{\mathsf{id},a}$ into $\mathsf{sk}_{\mathsf{id},a}$ for all $a \in \mathcal{A}$ so that the new randomized credentials are unlinkable to the original credentials $\mathsf{sk}'_{\mathsf{id},a}$. Another deviation from the construction in [15] is that we only need to hide the components of the user's attribute credentials $\mathsf{sk}_{\mathsf{id},a}$ which depend on the user's secret key $\mathsf{usk}_{\mathsf{id}}$ and publicly release the remaining components. When signing a message m w.r.t. a signing policy $\mathbb{P}$ and a non-empty recipient tag recip, the user produces a proof of knowledge π to prove that she either has enough credentials for a set of attributes $\mathcal{A}$ such that $\mathbb{P}(\mathcal{A}) = 1$ or has a DS signature on $\mathcal{H}(m, \mathbb{P}, \mathsf{recip})$ for some collision-resistant hash function $\mathcal{H} : \{0,1\}^* \to \mathcal{M}_{\mathsf{DS}}$. Following the literature, we refer to the latter as a pseudo-attribute. Since in this case the signature is linkable, the NIZK proof π additionally proves that the tag τ on recip verifies w.r.t. the same user secret key $\mathsf{usk}_{\mathsf{id}}$ with which the attribute credentials are associated. To prove the latter, let IsConsistent : $\mathcal{F} \times \mathcal{PK} \to \{0,1\}$ be a predicate where $\mathcal{F}$ and $\mathcal{PK}$ are the ranges of the functions f and PK respectively. In our instantiations, IsConsistent requires checking that the pair is a Diffie-Hellman pair. For all remaining attributes in the policy $\mathbb{P}$ that the signer is not going to use, i.e. the set of attributes $a \in \mathbb{P} \setminus \mathcal{A}$, the user chooses random dummy credentials $\boxed{\mathsf{sk}_{\mathsf{id},a}}$. For all attributes $a \in \mathcal{A}$, we parse $\mathsf{sk}_{\mathsf{id},a}$ as $(\widehat{\mathsf{sk}}_{\mathsf{id},a}, \check{\mathsf{sk}}_{\mathsf{id},a})$, where $\widehat{\mathsf{sk}}_{\mathsf{id},a}$ are independent of $\mathsf{usk}_{\mathsf{id}}$. Similarly, for all attributes $a \in \mathbb{P} \setminus \mathcal{A}$, parse $\boxed{\mathsf{sk}_{\mathsf{id},a}}$ as $\left(\boxed{\widehat{\mathsf{sk}}_{\mathsf{id},a}}, \boxed{\check{\mathsf{sk}}_{\mathsf{id},a}}\right)$. Let $\boxed{\mathsf{sk}_{\mathsf{id},a_{\mathrm{psdo}}}}$ be the credential for the pseudo-attribute, i.e. the DS signature on $\mathcal{H}(m, \mathbb{P}, \mathsf{recip})$. We parse $\boxed{\mathsf{sk}_{\mathsf{id},a_{\mathrm{psdo}}}}$ as $\left(\boxed{\widehat{\mathsf{sk}}_{\mathsf{id},a_{\mathrm{psdo}}}}, \boxed{\check{\mathsf{sk}}_{\mathsf{id},a_{\mathrm{psdo}}}}\right)$, where $\boxed{\widehat{\mathsf{sk}}_{\mathsf{id},a_{\mathrm{psdo}}}}$ is independent of the message of DS. Since the signing key of DS is only known to the challenger in the security reduction, we can safely reveal the component $\boxed{\widehat{\mathsf{sk}}_{\mathsf{id},a_{\mathrm{psdo}}}}$ in the clear since it does not reveal whether $\mathsf{sk}_{\mathsf{id},a_{\mathrm{psdo}}}$ is a valid signature on $\mathcal{H}(m, \mathbb{P}, \mathsf{recip})$ or a fake dummy signature. If $\mathsf{recip} = \perp$, then we set $\tau = \perp$ and π excludes the latter part concerning proving correctness of the tag τ. The signature Σ is then $\left(\pi, \tau, \{\widehat{\mathsf{sk}}_{\mathsf{id},a}\}_{a\in\mathcal{A}} \cup \{\boxed{\widehat{\mathsf{sk}}_{\mathsf{id},a}}\}_{a\in\mathbb{P}\setminus\mathcal{A}} \cup \boxed{\widehat{\mathsf{sk}}_{\mathsf{id},a_{\mathrm{psdo}}}}\right)$.

To verify the signature, it suffices to just verify the proof π. To link two signatures, one checks that tag components of the two valid signatures are identical.

Details of the construction are given in Fig 1, where $\vec{z}$ and $\mathbf{M}$ are the secret vector and the public span matrix used in the span program, respectively. The languages associated

$\underline{\mathsf{Setup}(1^\lambda)}$

- $(\mathsf{crs}, \mathsf{xk}) \leftarrow \mathsf{NIZK.Setup}(1^\lambda)$. $(\mathsf{svk}, \mathsf{ssk}) \leftarrow \mathsf{DS.KeyGen}(1^\lambda)$.
- Choose a collision-resistant hash function $\mathcal{H} : \{0,1\}^* \to \mathcal{M}_{\mathsf{DS}}$. Return $\mathsf{pp} := (1^\lambda, \mathsf{crs}, \mathcal{H}, \mathsf{svk})$.

$\underline{\mathsf{AASetup}(\mathsf{pp}, \mathsf{aid})}$

- $(\mathsf{vk}_{\mathsf{AAaid}}, \mathsf{sk}_{\mathsf{AAaid}}) \leftarrow \mathsf{RPSPS.KeyGen}(1^\lambda)$. Return $(\mathsf{vk}_{\mathsf{AAaid}}, \mathsf{sk}_{\mathsf{AAaid}})$.

$\underline{\mathsf{UKeyGen}(\mathsf{pp})}$

- $\mathsf{usk}_{\mathsf{id}} \leftarrow \mathsf{LIT.KeyGen}(1^\lambda)$. Return $\mathsf{usk}_{\mathsf{id}}$.

$\underline{\mathsf{AttKeyGen}(\mathsf{id}, f(\mathsf{usk}_{\mathsf{id}}), a, \mathsf{sk}_{\mathsf{AAaid}(a)})}$

- $\mathsf{sk}'_{\mathsf{id},a} \leftarrow \mathsf{RPSPS.Sign}\left(\mathsf{sk}_{\mathsf{AAaid}(a)}, \left(f(\mathsf{usk}_{\mathsf{id}}), \cdot\right), a\right)$. Return $\mathsf{sk}'_{\mathsf{id},a}$.

$\underline{\mathsf{Sign}(m, \mathbb{P}, \mathsf{usk}_{\mathsf{id}}, \{\mathsf{sk}'_{\mathsf{id},a}\}_{a\in\mathcal{A}}, \mathsf{recip})}$

- Return $\perp$ if $\mathbb{P}(\mathcal{A}) = 0$.
- Let $a_{\mathrm{psdo}} := \mathcal{H}(m, \mathbb{P}, \mathsf{recip})$ and $\hat{\mathbb{P}} := \mathbb{P} \vee a_{\mathrm{psdo}}$.
- For each $a \in \mathcal{A}$, compute $\mathsf{sk}_{\mathsf{id},a} \leftarrow \mathsf{RPSPS.Randomize}\left(\mathsf{vk}_{\mathsf{AAaid}(a)}, \left(\left(f(\mathsf{usk}_{\mathsf{id}}), \mathsf{PK}(\mathsf{usk}_{\mathsf{id}})\right), a\right), \mathsf{sk}'_{\mathsf{id},a}\right)$.
- For each $a \in \mathbb{P} \setminus \mathcal{A}$, choose a random dummy credential $\boxed{\mathsf{sk}_{\mathsf{id},a}}$.
- Choose a random dummy credential $\boxed{\mathsf{sk}_{\mathsf{id},a_{\mathrm{psdo}}}}$ for the pseudo-attribute a_{psdo}.
- Parse $\mathsf{sk}_{\mathsf{id},\alpha}$ as $\left(\widehat{\mathsf{sk}}_{\mathsf{id},a}, \check{\mathsf{sk}}_{\mathsf{id},a}\right)$, where $\widehat{\mathsf{sk}}_{\mathsf{id},a}$ are independent of $\mathsf{usk}_{\mathsf{id}}$.
- Similarly, parse $\boxed{\mathsf{sk}_{\mathsf{id},a}}$ as $\left(\boxed{\widehat{\mathsf{sk}}_{\mathsf{id},a}}, \boxed{\check{\mathsf{sk}}_{\mathsf{id},a}}\right)$, and $\boxed{\mathsf{sk}_{\mathsf{id},a_{\mathrm{psdo}}}}$ as $\left(\boxed{\widehat{\mathsf{sk}}_{\mathsf{id},a_{\mathrm{psdo}}}}, \boxed{\check{\mathsf{sk}}_{\mathsf{id},a_{\mathrm{psdo}}}}\right)$.
- If $\mathsf{recip} = \perp$ Then
  - Set $\tau := \perp$, $\Omega := (\{\mathsf{vk}_{\mathsf{AAaid}(a)}\}_{a\in\mathbb{P}} \cup \mathsf{svk}, \{\widehat{\mathsf{sk}}_{\mathsf{id},a}\}_{a\in\mathcal{A}} \cup \{\boxed{\widehat{\mathsf{sk}}_{\mathsf{id},a}}\}_{a\in\hat{\mathbb{P}}\setminus\mathcal{A}})$.
  - $\pi \leftarrow \mathsf{NIZK.Prove}\left(\mathsf{crs}, \left((f(\mathsf{usk}_{\mathsf{id}}), \mathsf{PK}(\mathsf{usk}_{\mathsf{id}})), \vec{z}, \{\check{\mathsf{sk}}_{\mathsf{id},a}\}_{a\in\mathcal{A}} \cup \{\boxed{\check{\mathsf{sk}}_{\mathsf{id},a}}\}_{a\in\hat{\mathbb{P}}\setminus\mathcal{A}}\right) : \Omega \in \mathcal{L}'\right)$.
- Else
  - $\tau \leftarrow \mathsf{LIT.Tag}(\mathsf{usk}_{\mathsf{id}}, \mathsf{recip})$.
  - $\Omega := (\tau, \mathsf{recip}, \{\mathsf{vk}_{\mathsf{AAaid}(a)}\}_{a\in\mathbb{P}} \cup \mathsf{svk}, \{\widehat{\mathsf{sk}}_{\mathsf{id},a}\}_{a\in\mathcal{A}} \cup \{\boxed{\widehat{\mathsf{sk}}_{\mathsf{id},a}}\}_{a\in\hat{\mathbb{P}}\setminus\mathcal{A}})$.
  - $\pi \leftarrow \mathsf{NIZK.Prove}(\mathsf{crs}, \left((f(\mathsf{usk}_{\mathsf{id}}), \mathsf{PK}(\mathsf{usk}_{\mathsf{id}})), \vec{z}, \{\check{\mathsf{sk}}_{\mathsf{id},a}\}_{a\in\mathcal{A}} \cup \{\boxed{\check{\mathsf{sk}}_{\mathsf{id},\alpha}}\}_{a\in\hat{\mathbb{P}}\setminus\mathcal{A}}\right) : \Omega \in \mathcal{L})$.
- Return $\Sigma := \left(\pi, \tau, \{\widehat{\mathsf{sk}}_{\mathsf{id},a}\}_{a\in\mathcal{A}} \cup \{\boxed{\widehat{\mathsf{sk}}_{\mathsf{id},a}}\}_{a\in\hat{\mathbb{P}}\setminus\mathcal{A}}\right)$.

$\underline{\mathsf{Verify}(\Sigma, \{\mathsf{vk}_{\mathsf{AAaid}(a)}\}_{a\in\mathbb{P}}, \mathbb{P}, m, \mathsf{recip})}$

- Parse Σ as $\left(\pi, \tau, \{\widehat{\mathsf{sk}}_{\mathsf{id},a}\}_{a\in\hat{\mathbb{P}}}\right)$ and pp as $(1^\lambda, \mathsf{crs}, \mathsf{svk}, \mathcal{H})$. Return $\mathsf{NIZK.Verify}(\mathsf{crs}, \pi)$.

$\underline{\mathsf{Link}\left(\mathsf{recip}, (m_i, \{\mathsf{vk}_{\mathsf{AAaid}(a)}\}_{a\in\mathbb{P}_i}, \mathbb{P}_i, \Sigma_i)_{i=1,2}\right)}$

- Parse Σ_i as $\left(\pi_i, \tau_i, \{\widehat{\mathsf{sk}}_{\mathsf{id},a}\}_{a\in\hat{\mathbb{P}}_i}\right)$ and pp as $(1^\lambda, \mathsf{crs}, \mathsf{svk}, \mathcal{H})$.
- Return 0 if $\mathsf{recip} = \perp$ or $\exists i \in \{1,2\}$ s.t. $\mathsf{Verify}(\Sigma_i, \{\mathsf{vk}_{\mathsf{AAaid}(a)}\}_{a\in\mathbb{P}_i}, \mathbb{P}_i, m_i, \mathsf{recip}) = 0$.
- If $\tau_1 = \tau_2 \neq \perp$ Then Return 1 Else Return 0.

$\underline{\mathsf{Identify}(\mathsf{sk}, \Sigma, m, \mathsf{recip}, \{\mathsf{vk}_{\mathsf{AAaid}(a)}\}_{a\in\mathbb{P}}, \mathbb{P})}$

- Parse Σ as $\left(\pi, \tau, \{\widehat{\mathsf{sk}}_{\mathsf{id},a}\}_{a\in\hat{\mathbb{P}}}\right)$ and pp as $(1^\lambda, \mathsf{crs}, \mathsf{svk}, \mathcal{H})$.
- If $\mathsf{recip} = \perp$ or $\mathsf{Verify}(\Sigma, \{\mathsf{vk}_{\mathsf{AAaid}(a)}\}_{a\in\mathbb{P}}, \mathbb{P}, m, \mathsf{recip}) = 0$ Then Return 0.
- If $\mathsf{LIT.Tag}(\mathsf{sk}, \mathsf{recip}) = \tau$ Then Return 1 Else Return 0.

Fig. 1. Our generic construction of ABS with User-Controlled Linkability

with the NIZK system are as follows. For clarity we underline the witnesses:

$$\mathcal{L}: \left\{ \begin{array}{l} \left(\left(\tau, \mathsf{recip}, \{\mathsf{vk}_{\mathsf{AAaid}(a)}\}_{a\in\mathbb{P}} \cup \mathsf{svk}, \{\widehat{\mathsf{sk}}_{\mathsf{id},a}\}_{a\in\hat{\mathbb{P}}} \right), \left((f(\mathsf{usk}_{\mathsf{id}}), \mathsf{PK}(\mathsf{usk}_{\mathsf{id}})), \vec{z}, \{\check{\mathsf{sk}}_{\mathsf{id},a}\}_{a\in\hat{\mathbb{P}}} \right) \right) \\ : \underline{\vec{z}}\mathbf{M} = [1,0,\ldots,0] \ \wedge \ \mathsf{LIT.Verify}(\underline{\mathsf{PK}(\mathsf{usk}_{\mathsf{id}})}, \mathsf{recip}, \tau) = 1 \ \wedge \ \mathsf{IsConsistent}(\underline{f(\mathsf{usk}_{\mathsf{id}})}, \underline{\mathsf{PK}(\mathsf{usk}_{\mathsf{id}})}) = 1 \\ \bigwedge_{i=1}^{|\mathbb{P}|} \text{ if } \underline{z_i} \neq 0 \Rightarrow \mathsf{RPSPS.Verify}\left(\mathsf{vk}_{\mathsf{AAaid}(a_i)}, \left((\underline{f(\mathsf{usk}_{\mathsf{id}})}, \underline{\mathsf{PK}(\mathsf{usk}_{\mathsf{id}})}), a_i \right), (\widehat{\mathsf{sk}}_{\mathsf{id},a_i}, \underline{\check{\mathsf{sk}}_{\mathsf{id},a_i}}) \right) = 1 \\ \wedge \ \text{ if } \underline{z_{|\mathbb{P}|+1}} \neq 0 \Rightarrow \mathsf{DS.Verify}\left(\mathsf{svk}, \mathcal{H}(m, \mathbb{P}, \mathsf{recip}), \left(\widehat{\mathsf{sk}}_{\mathsf{id},a_{\mathrm{psdo}}}, \underline{\check{\mathsf{sk}}_{\mathsf{id},a_{\mathrm{psdo}}}} \right) \right) = 1 \end{array} \right\}.$$

$\mathcal{L}'$ is similar to $\mathcal{L}$ but without $\mathsf{LIT.Verify}(\underline{\mathsf{PK}(\mathsf{usk}_{\mathsf{id}})}, \mathsf{recip}, \tau) = 1$ and $\mathsf{IsConsistent}(\underline{f(\mathsf{usk}_{\mathsf{id}})}, \underline{\mathsf{PK}(\mathsf{usk}_{\mathsf{id}})}) = 1$.

Theorem 2. *The construction in Fig. 1 is a secure ABS-UCL scheme.*

Proof. Correctness is straightforward and is easy to verify. Also, linkability follows from that of the LIT scheme and is easy to verify.

Lemma 1. *The construction satisfies anonymity if* NIZK *is zero-knowledge and* LIT *is* f*-indistinguishable.*

Proof. Note that all public parts of the attribute credentials including that for the pseudo-attribute are independent of the witness of the NIZK proof.

We proceed by defining a sequence of games such that the last game is independent of the bit b used in the anonymity game. We prove that an adversary against anonymity behaves differently in any two consecutive games only with a negligible probability.

Let $\eta(\lambda)$ be a polynomial representing an upper bound on the number of users the adversary is allowed to create in the game. We let Game 0 be the original anonymity game but where we randomly guess the challenge user used by randomly choosing $\bar{\mathsf{id}}$ from the set $\{1,,\ldots\eta(\lambda)]$ and aborting if the challenge user chosen is different from $\bar{\mathsf{id}}$. We have a probability of $\dfrac{1}{\eta(\lambda)}$ of guessing the challenge user correctly. If the advantage of the adversary against the original anonymity game is non-negligible then so is her advantage against Game 0 since $\eta(\lambda)$ is polynomial in λ.

Let Game 1 be the same as Game 0 but now the CRS crs used for the NIZK is chosen as a hiding string. By the security of the NIZK system, the difference between games Game 0 and Game 1 is negligible.

Let Game 2 be the same as Game 1 but now proof π part of the signature is a simulated proof rather than a real proof. By the zero-knowledge property of NIZK, the difference between games Game 1 and Game 2 is negligible.

Let Game 3 be the same as Game 2 but now every time the game answers a signature on behalf of user $\bar{\mathsf{id}}$ (via a signing or a challenge query) for a recipient tag $\mathsf{recip} \neq \perp$, if such a recipient tag has already been queried on behalf of the same user, we return the same tag τ; otherwise, we choose a random user key $\mathsf{usk}_{\mathsf{id}} \in \mathcal{SK}$ and use it to produce a tag τ on recip. We can use a hybrid argument and a reduction to the f-indistinguishability of the LIT scheme to argue that the difference between games Game 3 and Game 2 is negligible. Let $\gamma(\lambda)$ be a polynomial representing an upper bound on the total number of signing and challenge queries involving user $\bar{\mathsf{id}}$ on non-empty recipient names $\mathsf{recip} \neq \perp$.

We define a sequence of games $\mathsf{GM}_{i=0}^{\gamma(\lambda)}$ where in game GM_j we answer the first j queries on non-empty recipient tags using the key for user $\bar{\mathsf{id}}$ and from the $(j+1)$-th query onwards we use tags under random keys. If the adversary behaves differently in any two subsequent games with a non-negligible probability, we can use her to break the f-indistinguishability of the LIT scheme. We have GM_0 = Game 3 and $\mathsf{GM}_{\gamma(\lambda)}$ = Game 2. By the f-indistinguishability of LIT we have that the difference between games Game 3 and Game 2 is negligible.

Now note that Game 3 is independent of the bit b used in the anonymity game and therefore the adversary has a negligible advantage against anonymity. □

Lemma 2. *The construction satisfies unforgeability if* NIZK *is sound,* RPSPS *and* DS *are existentially unforgeable, and the hash function $\mathcal{H}$ is collision-resistant.*

Proof. By the collision-resistance of the hash function $\mathcal{H}$ used in the pseudo-attribute a_{psdo}, the adversary has a negligible probability in finding two different tuples $(m, \mathbb{P}, \mathsf{recip}) \neq (m', \mathbb{P}', \mathsf{recip}')$ where $\mathcal{H}(m, \mathbb{P}, \mathsf{recip}) = \mathcal{H}(m', \mathbb{P}', \mathsf{recip}')$.

By choosing the CRS crs for NIZK as a binding one, the proofs are perfectly sound and we are guaranteed to be able to extract a valid witness from the proof π^* part of the forged signature Σ^*. The underlying witness would be either a valid signature on a new pseudo-attribute $(m, \mathbb{P}, \mathsf{recip})$ which was not queried to the sign oracle or a set of credentials on a one-way function of a user secret key $\mathsf{usk}_{\mathsf{id}}$ where some of the valid credentials were not obtained from the AttKeyGen oracle. In the former case, we can reduce unforgeability to the existential unforgeability of signature scheme DS used for the pseudo-attribute, whereas in the latter case, by guessing which attribute authority (with probability $\frac{1}{\eta(\lambda)}$, where $\eta(\lambda)$ is a polynomial representing an upper bound on the number of attribute authorities the adversary can create in the unforgeability game) managing a forged credential, we can reduce unforgeability to the existential unforgeability of signature scheme RPSPS used in issuing attribute credentials to users. □

5.1 Instantiations

In both instantiations below, we instantiate NIZK using the Groth-Sahai system (secure under SXDH) and instantiate the signature scheme DS using the full Boneh-Boyen signature scheme [7] (secure under q-SDH). We instantiate the LIT scheme LIT using the weak Boneh-Boyen signature [7] as in [3] (secure under q-DDHI). The only difference between the two instantiations lies in how we instantiate the signature scheme RPSPS. In the following, we let $F = f(\mathsf{usk}_{\mathsf{id}}) = G^{\mathsf{usk}_{\mathsf{id}}}$ and $\tilde{F} = \mathsf{PK}(\mathsf{usk}_{\mathsf{id}}) = \tilde{H}^{\mathsf{usk}_{\mathsf{id}}}$.

Instantiation I. Here we instantiate RPSPS using the efficient partially structure-preserving signature scheme from [22] as shown in Section 3.1.

Below we detail the Groth-Sahai NIZK proofs required for the instantiation.

Let $\mathbf{M} \in \mathbb{Z}_p^{|\hat{\mathbb{P}}|,\beta}$ be the span program for $\hat{\mathbb{P}} := \mathbb{P} \vee a_{\text{psdo}}$. The proof π part of the signature is a proof for the following:

- To prove $\vec{z}\mathbf{M} = [1, 0, \ldots, 0]$, the signer proves:

$$\sum_{i=1}^{|\hat{\mathbb{P}}|} (\underline{\tilde{z}_i} M_{i,1}) = 1 \qquad \sum_{i=1}^{|\hat{\mathbb{P}}|} (\underline{\tilde{z}_i} M_{i,j}) = 0, \text{ for } j = 2, \ldots, \beta$$

- For each $a_i \in \{1, \ldots, |\mathbb{P}|\}$, we have $\mathsf{sk}'_{\mathsf{id},a_i} = (R'_i, S'_i) \in \mathbb{G}^2$ where $\widehat{\mathsf{sk}'}_{\mathsf{id},a_i} = R'_i$ and $\check{\mathsf{sk}}'_{\mathsf{id},a_i} = S'_i$. The signer re-randomizes $\mathsf{sk}'_{\mathsf{id},a_i}$ by choosing $r' \leftarrow \mathbb{Z}_p^\times$ and computing $\mathsf{sk}_{\mathsf{id},a_i} := (R_i, S_i) = (R_i'^{r'}, S_i'^{r'})$.
  To prove if $\underline{\tilde{z}_i} \neq 0 \Rightarrow \mathsf{RPSPS.Verify}\left(\mathsf{vk}_{\mathsf{AAaid}(a_i)}, \left((F, \tilde{F}), a_i\right), (\widehat{\mathsf{sk}}_{\mathsf{id},a_i}, \underline{\check{\mathsf{sk}}_{\mathsf{id},a_i}})\right) = 1$, where $\mathsf{vk}_{\mathsf{AAaid}(a_i)} = (\tilde{X}_i, \tilde{Y}_i, \tilde{Z}_i) \in \mathbb{H}^3$. Note that R_i is independent of R'_i and $(F, \tilde{F})$. The signer then proves the following:
  $$\underline{\check{S}_i} = \underline{S_i}^{\tilde{z}_i} \qquad \underline{\check{R}_i} = \underline{R_i}^{\tilde{z}_i} \qquad e(\underline{\check{S}_i}, \tilde{Z}_i) = e(\underline{\check{R}_i}, \underline{\tilde{F}})e(\underline{\check{R}_i}, \tilde{X}_i)e(\underline{\check{R}_i}^{\alpha_i}, \tilde{Y}_i)$$
  Note that since R_i is public, the verifier can verify that $R_i \neq 1_{\mathbb{G}}$. The verifier can on her own compute a Groth-Sahai commitment to the value $\check{R}_i^{a_i}$ by computing $C_{\check{R}_i}^{a_i}$, where $C_{\check{R}_i}$ is the Groth-Sahai commitment (which is ElGamal ciphertext) to $\check{R}_i$. Such an observation improves the efficiency. Also, we only need to commit to the elements of the vector $\vec{z}$ in $\mathbb{H}$, which further improves the efficiency.
- For the pseudo-attribute a_{psdo}, we have $\boxed{\widehat{\mathsf{sk}}_{\mathsf{id},a_{\text{psdo}}}} = r_{\mathsf{FBB}}$, $\boxed{\check{\mathsf{sk}}_{\mathsf{id},a_{\text{psdo}}}} = \sigma_{\mathsf{FBB}}$ and $\mathsf{svk} = (\tilde{X}_{\mathsf{FBB}}, \tilde{Y}_{\mathsf{FBB}})$, the signer proves that
  $$\underline{\sigma_{\mathsf{FBB}}} = \underline{\sigma_{\mathsf{FBB}}}^{\tilde{z}_{|\mathbb{P}|+1}} \quad \underline{\check{G}} = \underline{G}^{\tilde{z}_{|\mathbb{P}|+1}} \quad e(\underline{\sigma_{\mathsf{FBB}}}, \tilde{X}_{\mathsf{FBB}} \cdot \tilde{Y}_{\mathsf{FBB}}^{r_{\mathsf{FBB}}} \cdot \tilde{G}^{a_{\text{psdo}}})e(\underline{\check{G}}, \tilde{H}) = 1$$

The signature size of this instantiation is $(15|\mathbb{P}|+15)\cdot|\mathbb{G}|+(14|\mathbb{P}|+22)\cdot|\mathbb{H}|+(\beta+3)\cdot|p|$ which is much more efficient than the traceable constructions in [16, 21].

Instantiation II. Here we instantiate RPSPS with our new partially structure-preserving signature scheme from Section 3.1. The only difference from the details of the NIZK proof π from that of instantiation I is in the part that proves possession of credentials for attributes used in the signing which we detail below. The rest of the details of the proof π and hence the signature are identical to those of instantiation I.

For each $a_i \in \{1, \ldots, |\mathbb{P}|\}$, we have $\mathsf{sk}'_{\mathsf{id},a_i} = (R'_i, S'_i, \tilde{R}'_i) \in \mathbb{G}^2 \times \mathbb{H}$ where $\widehat{\mathsf{sk}'}_{\mathsf{id},a_i} = (R'_i, \tilde{R}'_i)$ and $\check{\mathsf{sk}}'_{\mathsf{id},a_i} = S'_i$. The signer re-randomizes $\mathsf{sk}'_{\mathsf{id},a_i}$ by choosing $r' \leftarrow \mathbb{Z}_p^\times$ and computing $\mathsf{sk}_{\mathsf{id},a_i} := (R_i, \tilde{R}_i, S_i) = (R'_i \cdot G^{r'}, \tilde{R}'_i \cdot \tilde{H}^{r'}, S'_i \cdot R_i'^{2r'} \cdot G^{r'^2})$. To prove if $\underline{\tilde{z}_i} \neq 0 \Rightarrow \mathsf{RPSPS.Verify}\left(\mathsf{vk}_{\mathsf{AAaid}(a_i)}, \left(F, a_i\right), (\widehat{\mathsf{sk}}_{\mathsf{id},a_i}, \underline{\check{\mathsf{sk}}_{\mathsf{id},a_i}})\right) = 1$, where $\mathsf{vk}_{\mathsf{AAaid}(a_i)} = (\tilde{X}_i, \tilde{Y}_i, \tilde{Z}_i) \in \mathbb{H}^3$. The signer proves the following:

$$\underline{\check{S}_i} = \underline{S_i}^{\tilde{z}_i} \qquad \underline{\check{F}_i} = \underline{F}^{\tilde{z}_i} \qquad \underline{\check{R}_i} = \underline{R_i}^{\tilde{z}_i} \qquad \underline{\check{G}_i} = \underline{G}^{\tilde{z}_i}$$
$$e(\underline{\check{S}_i}, \tilde{H}) = e(\underline{F}, \tilde{X}_i)e(\underline{\check{G}_i}^{\alpha_i}, \tilde{Y}_i)e(\underline{\check{R}_i}, \tilde{R}_i)e(\underline{\check{G}_i}, \tilde{Z}_i)$$

Note that since $(R_i, \tilde{R}_i)$ are independent of $\mathsf{vk}_{\mathsf{AAaid}(a_i)}$ and thus when choosing dummy credentials for attributes in $\mathbb{P} \setminus \mathcal{A}$, any one can choose such a pair satisfying $e(R_i, \tilde{H}) = e(G, \tilde{R}_i)$.

The verifier can on her own compute a Groth-Sahai commitment to the value $\check{G}_i^{a_i}$ by computing $\mathcal{C}_{\check{G}_i}^{a_i}$, where $\mathcal{C}_{\check{G}_i}$ is the Groth-Sahai commitment (which is ElGamal ciphertext) to $\check{G}_i$. Such an observation improves the efficiency. In addition, we only need to commit to the elements of the vector $\vec{z}$ in $\mathbb{H}$, which further improves the efficiency. When verifying the signature, one additionally checks that $e(R_i, \tilde{H}) = e(G, \tilde{R}_i)$.

The signature size of this instantiation is $(19|\mathbb{P}| + 15) \cdot |\mathbb{G}| + (21|\mathbb{P}| + 22) \cdot |\mathbb{H}| + (\beta + 3) \cdot |p|$ which is again more efficient than the traceable constructions in [16, 21].

6 Construction of ABS-UCL for Threshold Policies

We give here an ABS-UCL construction supporting threshold policies and a single authority. The scheme is based on an improved variant of the ABS scheme in [24].

- $\mathsf{Setup}(1^\lambda, n)$: Generate a bilinear group $\mathcal{P} := (\mathbb{G}, \mathbb{H}, \mathbb{T}, G, \tilde{H}, p, e)$ and choose a collision-resistant hash function $\mathcal{H} : \{0,1\}^* \to \{0,1\}^k$, and a coding map $\zeta : \mathcal{A} \to \mathbb{Z}_p^*$. Define a set of pairwise different elements of $\mathbb{Z}_p^*$, $D = \{d_1, \cdots, d_{n-1}\}$ where D_i represents the first i elements of D, i.e. $D_i = \{d_1, \cdots, d_i\}$. These values correspond to $n-1$ dummy attributes that should be different from all attributes appearing in $\mathcal{A}$. Generate a CRS crs for the Groth-Sahai NIZK system and the secret/verification keys $(\mathsf{ssk}, \mathsf{svk})$ of a signature scheme (Full Boneh-Boyen) which will be used to sign a special attribute called the pseudo-attribute which is needed in the security proofs to simulate signing queries and to bind a signature to the message.
  The public parameters is $\mathsf{pp} = (\mathcal{A}, n, \lambda, \mathcal{P}, G', \tilde{H}', \mathsf{crs}, \mathcal{H}, D, \mathsf{svk})$, where $e(G, \tilde{H}') = e(G', \tilde{H})$.
- $\mathsf{AASetup}(\mathsf{pp})$: this is run by the attribute authority. It randomly chooses $\alpha, \gamma \in \mathbb{Z}_p^*$, and sets $U = G^{\alpha\gamma}$, and $V = e(G, \tilde{H})^\alpha$. The master secret key is $\mathsf{gmsk} = (\alpha, \gamma)$ whereas the master public key is $\mathsf{gmpk} = (U, V, G^\alpha, \{\tilde{H}^{\alpha\gamma^i}\}_{i=0,\cdots,2n-1})$.
- $\mathsf{AttKeyGen}(\mathsf{pp}, \mathsf{gmpk}, I, F, \tilde{F})$: given a set of attributes $I \subset \mathcal{A}$ and $F = (G')^{\mathsf{usk}_{\mathsf{id}}}$, and $\tilde{F} = (\tilde{H}')^{\mathsf{usk}_{\mathsf{id}}}$. It picks $r \leftarrow \mathbb{Z}_p^*$ at random and returns

$$\mathsf{sk}_{\mathsf{id},I} = \Big(\{(G \cdot F)^{\frac{r}{\gamma+\zeta(a)}}\}_{a\in I}, \{(\tilde{H} \cdot \tilde{F})^{r\gamma^i}\}_{i=0,\cdots,n-2}, (\tilde{H} \cdot \tilde{F})^{\frac{r-1}{\gamma}}\Big)$$

.

- $\mathsf{Sign}(\mathsf{pp}, \mathsf{gmpk}, \mathsf{sk}_{\mathsf{id},I}, \mathsf{usk}_{\mathsf{id}}, M, \mathbb{P}, \mathsf{recip})$: Given a message $M \in \{0,1\}^*$, a policy $\mathbb{P}(t, S)$, for which $t \le |S| = s \le n$. If $|S \cap I| < t$, return $\perp$. Otherwise, fix the signing set of attributes as $I_S \subseteq I$ where $|I_S| = t$ and compute the following:
  - Let $C_1 = (G \cdot F)^{z_1} \leftarrow \mathsf{Aggregate}(\{(G \cdot F)^{\frac{r}{\gamma+\zeta(a)}}\}_{a\in I_S}, I_S)$, where[1]

$$z_1 = \frac{r}{\prod_{a\in I_S}(\gamma + \zeta(a))}$$

  - Define the sets $W = S \cup D_{n+t-1-s}$ and $R = W \setminus I_S$ and compute $z_2 = \prod_{a\in R}(\zeta(a))$. Then compute $T_1 = C_1^{1/z_2}$.
  - Define the following polynomial in γ where $\deg(P(\gamma)) = n-2$

$$P(\gamma) = \frac{1}{\gamma}\Big(\prod_{a\in R}(\gamma + \zeta(a)) - \prod_{a\in R}(\zeta(a))\Big).$$

[1] Aggregate was originally defined in [13], and is based on the fact that a product of inverses of coprime polynomials can be written as a sum of inverses of affine polynomials, as described and used in [14]. More details on Aggregate can be found in [13].

Using the second part of the secret key as an input to a Lagrange interpolation algorithm, one can efficiently compute $\tilde{H}_1 = (\tilde{H} \cdot \tilde{F})^{rP(\gamma)}$. We actually need to compute the following value;

$$\tilde{T}_2 = (\tilde{H} \cdot \tilde{F})^{\frac{r-1}{\gamma}} \cdot \tilde{H}_1^{\frac{1}{\prod_{a \in R(\zeta(a))}}}$$

- If $\mathsf{recip} \neq \perp$, set $\tau = \chi^{\frac{1}{\mathsf{recip}+\mathsf{usk}_{\mathsf{id}}}}$, where $\chi \in \mathbb{G}$. Otherwise, $\tau = \perp$.
- Let $\tilde{H}_2 = \tilde{H}^{\alpha \cdot \prod_{a \in W}(\gamma+\zeta(a))}$, which can be computed from gmpk.
- When $\mathsf{recip} \neq \perp$, the signer proves that either she has enough attributes to satisfy the policy $\mathbb{P}$ or she has a signature on the pseudo-attribute $a_{\text{psdo}} = \mathcal{H}(M, \mathbb{P}, \mathsf{recip})$. This is realized by a GS NIZK proof for the following language:

$$\mathcal{L} : \left\{ \begin{array}{l} \left((\tau, \mathsf{recip}, \mathsf{pp}), \left((f(\mathsf{usk}_{\mathsf{id}}), \mathsf{PK}(\mathsf{usk}_{\mathsf{id}})), \tilde{T}_2, T_1, G'', \tilde{H}''\right)\right) : \\ e(U^{-1}, \underline{\tilde{T}_2}) \cdot e(T_1, \tilde{H}_2) = e(G^{\alpha}, \underline{\tilde{H}''}) \cdot e(G^{\alpha}, \underline{\tilde{F}}) \\ \bigwedge \underline{F} = (G')^{\underline{\mathsf{usk}_{\mathsf{id}}}} \bigwedge e(\underline{\sigma_{\mathsf{FBB}}^{\sim}}, \tilde{X}_{\mathsf{FBB}} \cdot \tilde{Y}_{\mathsf{FBB}}^{r_{\mathsf{FBB}}} \cdot \tilde{G}^{a_{\text{psdo}}}) = e(\underline{G''}, \tilde{H}) \\ \bigwedge e(\underline{G''} \cdot G^{-1}, \underline{\tilde{H}''} \cdot \tilde{H}^{-1}) = 1 \bigwedge e(\tau, \underline{\tilde{H}^{\mathsf{usk}_{\mathsf{id}}}}) \cdot e(\tau, \tilde{H}^{\mathsf{recip}}) = e(\underline{\chi}, \tilde{H}) \\ \bigwedge e(G, \underline{\tilde{F}}) = e(G', \underline{\tilde{H}^{\mathsf{usk}_{\mathsf{id}}}}) \end{array} \right\},$$

- When $\mathsf{recip} = \perp$, everything stays the same except that we remove the last two equations from the NIZK language and change the pseudo-attribute to $a_{\text{psdo}} = \mathcal{H}(M, \mathbb{P})$.

The signature is $\sigma = (\pi_{\mathsf{GS}}, \tau)$ which is of size $27 \cdot |\mathbb{G}| + 28 \cdot |\mathbb{H}|$.

- $\mathsf{Verify}(\mathsf{pp}, \mathsf{gmpk}, M, \mathbb{P}, \sigma, \mathsf{recip})$: Compute $H_2 = \tilde{H}^{\alpha \cdot \prod_{a \in W}(\gamma+\zeta(a))}$ and verify the GS proof π_{GS}.
- $\mathsf{Link}(\mathsf{pp}, \mathsf{gmpk}, \sigma_1, \sigma_2, M_1, M_2, \mathbb{P}_1, \mathbb{P}_2, \mathsf{recip})$: If σ_1 or σ_2 are invalid, return 0. Otherwise, parse σ_i as $(\pi_{\mathsf{GS}_i}, \tau_i)$ and return 1 if both τ_1 and τ_2 are non-trivial and $\tau_1 = \tau_2 \neq \perp$, return 0 otherwise.

Efficiency comparison with Herranz et al. results in [24]:. The two schemes that are presented in [24] employ the less efficient pairing setting, i.e. symmetric pairing. The sizes of the schemes' signatures are 15 and 3 group elements. The latter comes at the cost of having longer secret keys. Our scheme, even after adding the user-controlled linkability feature, is still comparable to the first scheme in terms of the size of the secret key and the size of the signature (as the group elements in our scheme are much smaller). However, the public parameters in our scheme are shorter (by k group elements, where k is the bit length of the output of the hash function used in the sheme). We get this improvment by using a diffetent technique, i.e. by using the pseudo-attribute idea, to bind the message to the signature. Moreover, using our technique to bind the message, and if we were to drop the user-linkability property from our scheme, we would get a standard ABS for threshold policies that has the same signature size as [24], but with much shorter public keys.

The proof of the following theorem is in Appendix B.

Theorem 3. *The construction above is a secure tABS-UCL.*

References

1. M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, and M. Ohkubo. Structure-preserving signatures and commitments to group elements. In *CRYPTO*, pages 209–236. Springer, 2010.
2. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS*, pages 62–73. ACM, 1993.
3. D. Bernhard, G. Fuchsbauer, and E. Ghadafi. Efficient signatures of knowledge and daa in the standard model. In *ACNS*, pages 518–533. Springer, 2013.
4. D. Bernhard, G. Fuchsbauer, E. Ghadafi, N. P. Smart, and B. Warinschi. Anonymous attestation with user-controlled linkability. *Int. J. Inf. Secur.*, 12(3):219–249, 2013.
5. R. Bobba, O. Fatemieh, F. Khan, C. A. Gunter, and H. Khurana. Using attribute-based access control to enable attribute-based messaging. In *ACSAC. IEEE CS*, pages 403–413, 2006.
6. D. Boneh and X. Boyen. *EUROCRYPT*, volume 3027, chapter Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles, pages 223–238. Springer, 2004.
7. D. Boneh and X. Boyen. Short signatures without random oracles and the SDH assumption in bilinear groups. *Journal of Cryptology*, 21(2):149–177, 2008.
8. X. Boyen. *Mesh Signatures*, volume 4515, pages 210–227. Springer, 2007.
9. E. Brickell, J. Camenisch, and L. Chen. Direct anonymous attestation. In *ACM CCS*, pages 132–145. ACM, 2004.
10. S. Chatterjee and A. Menezes. *ASIACRYPT*, volume 9452, chapter Type 2 Structure-Preserving Signature Schemes Revisited, pages 286–310. Springer, 2015.
11. D. Chaum and E. Van Heyst. Group signatures. EUROCRYPT, pages 257–265. Springer-Verlag, 1991.
12. L. Chen and R. Urian. *DAA-A: Direct Anonymous Attestation with Attributes*, volume 9229, pages 228–245. Springer, 2015.
13. C. Delerablée, P. Paillier, and D. Pointcheval. Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys. *Pairing-Based Cryptography–Pairing 2007*, pages 39–59, 2007.
14. C. Delerablée and D. Pointcheval. Dynamic threshold public-key encryption. In *CRYPTO*, volume 5157, pages 317–334. Springer, 2008.
15. A. El Kaafarani, L. Chen, E. Ghadafi, and J. Davenport. Attribute-based signatures with user-controlled linkability. In *CANS*, pages 256–269. Springer, 2014.
16. A. El Kaafarani, E. Ghadafi, and D. Khader. Decentralized traceable attribute-based signatures. In *CT-RSA*, pages 327–348. Springer, 2014.
17. A. Escala, J. Herranz, and P. Morillo. Revocable attribute-based signatures with adaptive security in the standard model. In *AFRICACRYPT*, pages 224–241. Springer, 2011.
18. K. B. Frikken, J. Li, and M. J. Atallah. Trust negotiation with hidden credentials, hidden policies, and policy cycles. In *NDSS*, pages 157–172, 2006.
19. M. Gagné, S. Narayan, and R. Safavi-Naini. Short pairing-efficient threshold-attribute-based signature. In *Pairing*, volume 7708, pages 295–313. Springer, 2013.
20. S. D. Galbraith, K. G. Paterson, and N. P. Smart. Pairings for cryptographers. *Discrete Appl. Math.*, 156(16):3113–3121, 2008.
21. E. Ghadafi. Stronger security notions for decentralized traceable attribute-based signatures and more efficient constructions. In K. Nyberg, editor, *CT-RSA*, pages 391–409. Springer, 2015.
22. E. Ghadafi. More efficient structure-preserving signatures - or: Bypassing the type-III lower bounds. Cryptology ePrint Archive, Report 2016/255, 2016. `http://eprint.iacr.org/`.
23. J. Groth and A. Sahai. Efficient noninteractive proof systems for bilinear groups. *SIAM J. Comput.*, 41(5):1193–1232, 2012.

24. J. Herranz, F. Laguillaumie, B. Libert, and C. Ràfols. Short attribute-based signatures for threshold predicates. In *CT-RSA*, volume 7178, pages 51–67. Springer, 2012.
25. J. Herranz, F. Laguillaumie, and C. Ràfols. Constant size ciphertexts in threshold attribute-based encryption. In *PKC*, pages 19–34. Springer, 2010.
26. M. Karchmer and A. Wigderson. On span programs. In *IEEE Structure in Complexity Theory*, pages 102–111, 1993.
27. J. Li, M. H. Au, W. Susilo, D. Xie, and K. Ren. Attribute-based signature and its applications. In *ASIACCS*, pages 60–69. ACM, 2010.
28. H. K. Maji, M. Prabhakaran, and M. Rosulek. Attribute-based signatures. In *CT-RSA*, pages 376–392. Springer, 2011.
29. T. Okamoto and K. Takashima. Efficient attribute-based signatures for non-monotone predicates in the standard model. In *PKC*, pages 35–52. Springer, 2011.
30. T. Okamoto and K. Takashima. Decentralized attribute-based signatures. In *PKC*, pages 125–142. Springer, 2013.
31. R. L. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *ASIACRYPT*, pages 552–565. Springer, 2001.
32. Y. Sakai, N. Attrapadung, and G. Hanaoka. *Attribute-Based Signatures for Circuits from Bilinear Map*, volume 9614, pages 283–300. Springer, 2016.
33. S. F. Shahandashti and R. Safavi-Naini. Threshold attribute-based signatures and their application to anonymous credential systems. In *AFRICACRYPT*, pages 198–216. Springer, 2009.
34. F. Tang, H. Li, and B. Liang. *Attribute-Based Signatures for Circuits from Multilinear Maps*, volume 8783, pages 54–71. Springer, 2014.
35. M. Urquidi, D. Khader, J. Lancrenon, and L. Chen. Attribute-based signatures with controllable linkability. In *INTRUST*, volume 9565, pages 114–129. Springer, 2015.

A Proof of Theorem 1

Proof. We prove that no linear combinations of the elements available to the adversary produce Laurent polynomials corresponding to a forgery on a message that was not queried to the sign oracle.

Public elements in $\mathbb{H}$ are $\tilde{H}$, $\tilde{X}_1$, …, $\tilde{X}_n$, $\tilde{Y}_1$, …, $\tilde{Y}_{n'}$, $\tilde{Z}$ which correspond to the discrete logarithms $1, x_1, \ldots, x_n, y_1, \ldots, y_{n'}, z$, respectively. Thus, this means that at the it-h sign query on $(\vec{U}_i, \vec{m}_i)$, $U_{i,j}$ (for $j = 1, \ldots, n$) can only be a linear combination of $G, \{R_k\}_{k=1}^{i-1}, \{S_k\}_{k=1}^{i-1}$. Thus, we have

$$u_{i,j} = a_{u_{i,j}} + \sum_{k=1}^{i-1} b_{u_{i,j,k}} r_k + \sum_{k=1}^{i-1} c_{u_{i,j,k}} (\sum_{l=1}^{n} u_{k,l} x_l + \sum_{l=1}^{n'} m_{k,l} y_l + r_k^2 + z),$$

After q signing queries, $\vec{u^*}$, which is the discrete logarithm of the forged vector $\vec{U}^*$ must be of the form

$$u_i^* = a_{u_i} + \sum_{k=1}^{q} b_{u_{i,k}} r_k + \sum_{k=1}^{q} c_{u_{i,k}} (\sum_{l=1}^{n} u_{k,l} x_l + \sum_{l=1}^{n'} m_{k,l} y_l + r_k^2 + z), \text{ for } i = 1, \ldots, n$$

Similarly, the (R^*, S^*) components part of the forgery can only be a linear combination of the group elements from $\mathbb{G}$, i.e. a linear combination of G, $\{R_i\}_{i=1}^{q}$ and $\{S_i\}_{i=1}^{q}$ and

therefore we have

$$r^* = a_r + \sum_{k=1}^{q} b_{r_k} r_k + \sum_{k=1}^{q} c_{r_k} (\sum_{l=1}^{n} u_{k,l} x_l + \sum_{l=1}^{n'} m_{k,l} y_l + r_k^2 + z)$$
$$s^* = a_s + \sum_{k=1}^{q} b_{s_k} r_k + \sum_{k=1}^{q} c_{s_k} (\sum_{l=1}^{n} u_{k,l} x_l + \sum_{l=1}^{n'} m_{k,l} y_l + r_k^2 + z)$$

Analogously, the $\tilde{R}^*$ part of the forgery can only be a linear combination of the elements from $\mathbb{H}$. Therefore, we have

$$\tilde{r}^* = a_{\tilde{r}} + \sum_{k=1}^{q} b_{\tilde{r}_k} \tilde{r}_k + \sum_{i=1}^{n} c_{\tilde{r}_i} x_i + \sum_{i=1}^{n'} d_{\tilde{r}_i} y_i + e_{\tilde{r}} z$$

For the forgery to be a valid signature, r^*, $\tilde{r}^*$ and s^* must satisfy

$$r^* = \tilde{r}^* \tag{7}$$
$$s^* = \sum_{l=1}^{n} u_l^* x_l + \sum_{l=1}^{n'} m_l^* y_l + r^{*2} + z \tag{8}$$

By (7), we must have $e_{\tilde{r}} = 0$ and $c_{\tilde{r}_i} = d_{\tilde{r}_i} = 0$ for all $i \in [n]$. Also, we must have $a_{\tilde{r}} = a_r$, $b_{\tilde{r}_k} = b_{r_k}$ for all k, and $c_{r_k} = 0$ for all $k \in [q]$. Therefore, we have

$$r^* = \tilde{r}^* = a_r + \sum_{k=1}^{q} b_{r_k} r_k$$

By (8), we must have

$$a_s + \sum_{k=1}^{q} b_{s_k} r_k + \sum_{k=1}^{q} c_{s_k} (\sum_{l=1}^{n} u_{k,l} x_l + \sum_{l=1}^{n'} m_{k,l} y_l + r_k^2 + z) = \sum_{l=1}^{n} u_l^* x_l + \sum_{l=1}^{n'} m_l^* y_l + r^{*2} + z$$

Thus, we must have

$$a_s + \sum_{k=1}^{q} b_{s_k} r_k + \sum_{k=1}^{q} c_{s_k} (\sum_{l=1}^{n} u_{k,l} x_l + \sum_{l=1}^{n'} m_{k,l} y_l + r_k^2 + z) = \sum_{l=1}^{n} u_l^* x_l + \sum_{l=1}^{n'} m_l^* y_l + \big(a_r + \sum_{k=1}^{q} b_{r_k} r_k\big)^2 + z$$

Note that on the left-hand side there is no term in $r_j r_k$ for all $k \neq j$. This means that on the right-hand side we must have $b_{r_j} b_{r_k} = 0$ for all $k \neq j$. This implies that there is only

one value of j such that $b_{r_j} \neq 0$, whereas $b_{r_k} = 0$ for all $k \neq j$. Thus, we have

$$a_s + \sum_{k=1}^{q} b_{s_k} r_k + \sum_{k=1}^{q} c_{s_k} (\sum_{l=1}^{n} u_{k,l} x_l + \sum_{l=1}^{n'} m_{k,l} y_l + r_k^2 + z) \\ = \sum_{l=1}^{n} u_l^* x_l + \sum_{l=1}^{n'} m_l^* y_l + (a_r + b_{r_j} r_j)^2 + z$$

Thus, we have

$$a_s + \sum_{k=1}^{q} b_{s_k} r_k + \sum_{k=1}^{q} c_{s_k} (\sum_{l=1}^{n} u_{k,l} x_l + \sum_{l=1}^{n'} m_{k,l} y_l + r_k^2 + z) \\ = \sum_{l=1}^{n} u_l^* x_l + \sum_{l=1}^{n'} m_l^* y_l + a_r^2 + 2 a_r b_{r_j} r_j + b_{r_j}^2 r_j^2 + z$$

Re-writing the left-hand side we have

$$a_s + b_{s_j} r_j + c_{s_j} \sum_{l=1}^{n} u_{j,l} x_l + c_{s_j} \sum_{l=1}^{n'} m_{j,l} y_l + c_{s_j} r_j^2 + c_{s_j} z \\ = \sum_{l=1}^{n} u_l^* x_l + \sum_{l=1}^{n'} m_l^* y_l + a_r^2 + 2 a_r b_{r_j} r_j + b_{r_j}^2 r_j^2 + z$$

The monomial z implies $c_{s,j} = 1$. Therefore, we have

$$a_s + b_{s_j} r_j + \sum_{l=1}^{n} u_{j,l} x_l + \sum_{l=1}^{n'} m_{j,l} y_l + r_j^2 + z \\ = \sum_{l=1}^{n} u_l^* x_l + \sum_{l=1}^{n'} m_l^* y_l + a_r^2 + 2 a_r b_{r_j} r_j + b_{r_j}^2 r_j^2 + z$$

By the monomial x_l, it is clear that we must have $u_{j,j} = u_l^*$ for all $l \in [n]$ and some $j \in [q]$. Similarly, the monomial y_l implies we must have $m_{j,l} = m_l^*$ for all $l \in [n']$ and some $j \in [q]$. This means the forgery is on vectors which have been queried to the sign oracle and therefore the adversary does not win.

This concludes the proof. □

B Single authority tABS-UCL: Security Proofs

Theorem 4 (Anonymity). *If the NIZK proof system* NIZK *is zero-knowledge, the linkable indistinguishable tag scheme* LIT *is indistinguishable, and the hash function $\mathcal{H}$ is collision-resistance then the tABS-UCL is anonymous.*

Proof. We will prove the anonymity of tABS-UCL by showing that a sequence of games are only negligibly indistinguishable from one another. The technique is similar to [3]. The challenger will beforehand guess the challenge user that will be chosen by the adversary $\mathcal{F}$ during the game, and will abort if they are not the same. If the adversary has

advantage ϵ in winning the unchanged game, than $\mathcal{F}$ will have an advantage $\epsilon/\eta(\lambda)$ in winning the second game, where $\eta(\lambda)$, a polynomial in λ, is the upper bounds of number of users that an adversary can create. The sequence of games will start with a game where the challenger guesses correctly the challenger user and end with a game that is independent of challenger user. If we can prove that $\mathcal{F}$'s behaviour changes only with negligible probability, then our tABS-UCL is anonymous.

We start with the first game Game 0 where the challenger guesses a particular user $\mathsf{id}^* \leftarrow [1, \eta(\lambda)]$, and aborts the game if in the challenge call we have $\mathsf{id}_b \neq \mathsf{id}^*$. In *Game-1*, we replace the crs of the NIZK proof by the hiding crs, i.e. we run $\mathsf{GSSimSetup}$ instead of $\mathsf{GSSetup}$.

By the Zero-knowledge proporty of the NIZK, we have that Game 0 and Game 1 differ only negligibly from each other.

In Game 2, we don't use $\mathsf{usk_{id}}^*$ while replying to KeyGen oracle queries. This game is only negligibly different from *Game-1* as the generated keys are always randomized with a fresh uniform random r and therefore are indistinguishable from random group elements.

In Game 3, for any Sign or Challenge queries that involve id^*, if the recip is new, we use independent uniformly random keys to produce the tags, i.e. σ_{UCL}, otherwise, we use the same tag that has been used before.

From Game 2 to Game 3, we will have a series of sub-games, where each two consecutive ones are indistinguishable by the indistinguishability of the LIT, i.e. they differ from each other by a single construction of a LIT tag. We start by answering all queries related to the selected signer id^* using its secret key $\mathsf{usk_{id}}^*$, then we move from a sub-game to another by answering one of these queries using a random key sk. We end up in the last sub-game where we answer all those queries using a key sk chosen uniformly at random. We have $\mu(\lambda)$ sub-games, where each two consecutive ones differ from each other by a negligible value, i.e. the advantage against the indistinguishability of the LIT. One can easily see that the last game is now independent of the challenge user used in answering challenge query, and hence the anonymity of tABS-UCL. □

Theorem 5 (Unforgeability). *if (ℓ, m, t)-aMSE-CDH holds, $\mathcal{H}$ is collision-resistance and the NIZK system is sound then our tABS-UCL is unforgeable.*

Proof. First, by the collision-resistance of $\mathcal{H}$ the adversary has a negligible probability in finding two different tuples $(m, \mathbb{P}, \mathsf{recip}) \neq (m', \mathbb{P}', \mathsf{recip}')$ where $\mathcal{H}(m, \mathbb{P}, \mathsf{recip}) = \mathcal{H}(m', \mathbb{P}', \mathsf{recip}')$.

Now, we take as input a problem instance of (ℓ, m, t)-aMSE-CDH. We denote the generators used in the given instance by $G_0, \tilde{H}_0$. Given an attacker $\mathcal{F}_1$ that can break the unforgeability of our tABS-UCL scheme, we will build an attacker $\mathcal{F}_2$ that can use $\mathcal{F}_1$ as a subroutine to solve (ℓ, m, t)-aMSE-CDH. The attacker gives the policy that he wants to be challenged on $\mathbb{P}^*(t^*, S^*)$ where $|S^*| = s^*$. We can set $n - s^* = \ell, n + t^* - 1 = m$ and $t^* + 1 = t$. The adversary $\mathcal{F}_2$ will then simulate the different algorithms of tABS-UCL as follows:

Setup.

- Define the attribute encoding ζ as follows:

$$\zeta(a) = \begin{cases} -x_i, \text{where } g_1(x_i) = 0 & \text{if } a \in \mathcal{P} \backslash S^* \\ -x_j, \text{where } g_2(x_j) = 0 & \text{if } a \in S^* \text{or } a \in D_1 = D_{n+t^*-1-s^*} \\ d, d \leftarrow Z_p & \text{if } a \in D \backslash D_1 \end{cases}$$

- Use the elements of line (1) to compute $G_0^{g_1(\gamma)}$, set $G := G_0^{g_1(\gamma)}$ and $\tilde{H} := \tilde{H}_0$.
- Sample $x, y \leftarrow Z_p^*$ and set $F_1 = G^x$ and $\tilde{F}_2 = \tilde{H}^y$.
- Use line (3) to compute $u = G^{\alpha\gamma} = G_0^{\alpha\gamma g_1(\gamma)}$ and $v = e(G, \tilde{H})^\alpha = e(G_0^{g_1(\gamma)\alpha}, \tilde{H}_0)$.
- Use line (6) to get $\{h^{\alpha\gamma^i}\}_{i=0,\ldots,2n-1}$. Then generate $(\mathsf{svk}, \mathsf{ssk})$ for the pseudo-attribute and crs for GS proofs in the soundness setting.

Key Generation. On input $\mathbb{P}(\Omega, t)$, if $|\Omega_S = \Omega \cap S^*| \geq t^*$, return $\perp$, otherwise it generates the key corresponding to S as follows: The first step is to compute

$$\left(\left\{ G^{\frac{r}{\gamma+\zeta(a)}} \right\}_{a\in\Omega}, \left\{ \tilde{H}^{r\gamma^i} \right\}_{i=0,\ldots,n-2}, \tilde{H}^{\frac{r-1}{\gamma}} \right)$$

and the second step is to compute:

$$\mathsf{sk}_\Omega = \left(\left\{ \left(G^{\frac{r}{\gamma+\zeta(a)}} \right)^{x\mathsf{sk}_{sid}+1} \right\}_{a\in\Omega}, \left\{ \left(\tilde{H}^{r\gamma^i} \right)^{y\mathsf{sk}_{sid}+1} \right\}_{i=0,\ldots,n-2}, \left(\tilde{H}^{\frac{r-1}{\gamma}} \right)^{y\mathsf{sk}_{sid}+1} \right)$$

The second step can be easily done as the challenger knows of $x, y, \mathsf{sk}_{\mathsf{id}}$. For the first step, here are the details (similar to [24]):

- Let where $\lambda_\Omega = \left(\prod_{a\in\Omega_S} \zeta(a) \right)^{-1}$, $r = (\omega y_\Omega \gamma + 1) Q_\Omega(\gamma)$. Define

$$Q_\Omega(X) = \begin{cases} 1 & \text{if } X = \gamma \text{ and } |\Omega_S| = 0 \\ \lambda_\Omega \cdot \prod_{a\in\Omega_S} (X + \zeta(a)) & \text{otherwise} \end{cases}$$

- Define

$$L_a(X) = \begin{cases} \dfrac{Q_\Omega(X)}{X + \zeta(a)} & \text{if } a \in \Omega_S \\ \dfrac{g_1(X)}{X + \zeta(a)} & \text{if } a \in \Omega \backslash \Omega_S \end{cases}$$

- For $a \in \Omega_S$, use the line (1) and (2) to compute $G^{\frac{r}{\gamma+\zeta(a)}} = G_0^{g_1(\gamma)\omega y_\Omega \gamma L_a(\gamma)} \cdot G_0^{g_1(\gamma) L_a(\gamma)}$. For $a \in \Omega \backslash \Omega_S$, compute $G^{\frac{r}{\gamma+\zeta(a)}} = G_0^{L_a(\gamma)\omega y_\Omega \gamma Q_\Omega(\gamma)} \cdot G_0^{Q_\Omega(\gamma) L_a(\gamma)}$.
- Use the lines (4) and (5) to compute $\left\{ \tilde{H}^{r\gamma^i} \right\}_{i=0,\cdots,n-2}$.
- Use the line (4) to compute $\tilde{H}^{\frac{Q_\Omega(\gamma)-1}{\gamma}}$ and (5) to compute $\tilde{H}^{Q_\Omega(\gamma)\omega y_\omega}$. Their product will give $\tilde{H}^{\frac{r-1}{\gamma}}$.

Signing Queries. Use the pseudo-attribute secret key ssk to sign any message of the attacker's choice.

Forgery. The adversary now outputs a valid signature. By the extractability of the GS proofs, $\mathcal{F}_2$ can either extract T_1 and T_2 or a valid signature on the pseudo-attribute. In the first case, we note that $\mathcal{F}_2$ knows of x, y and $\mathsf{sk}_{\mathsf{id}}$, and therefore can compute $T_1' = T_1^{\frac{1}{x\mathsf{sk}_{\mathsf{id}}+1}}, T_2' = T_2^{\frac{1}{y\mathsf{sk}_{\mathsf{id}}+1}}$. Using the lines (4) and (5), $\mathcal{F}_2$ can finally compute

$$e(T_1', \tilde{H}_0^{\kappa g_2(\gamma)}) \cdot e(G^{-\kappa\gamma g_1(\gamma)}, T_2') = e(G_0, \tilde{H}_0)^{\kappa g_1(\gamma)}$$

and therefore solve (ℓ, m, t)-aMSE-CDH. In the second case, the forgery on tABS-UCL will directly give a forgery on the underlying digital signature used to sign the pseudo-attributes (i.e. the Full Boneh-Boyen signature). □

Theorem 6 (User-controlled linkability). *The threshold attribute based signatures is User controlled linkable if the Linkable Indistinguishable tag scheme* LIT *is linkable.*

Proof. We will first deal with the case in which an adversary produces two supposedly linkable signatures, but when testing them with Link, it says they are not. Given that an adversary $\mathcal{C}$ has full control over the secret keys so he can generate secret keys to any user that he wants to be challenged on, say $\mathsf{id}_{\mathsf{Link}}$. He should also pick the verifier's name recip as a part of the challenge. At the end, he needs to produce two signatures, σ_1 and σ_2 on behalf of the user $\mathsf{id}_{\mathsf{Link}}$, for which $\sigma_1 = (\sigma_{\mathsf{ABS}_1}, \sigma_{\mathsf{UCL}_1})$ and $\sigma_2 = (\sigma_{\mathsf{ABS}_2}, \sigma_{\mathsf{UCL}_2})$. He wins if both signatures σ_1 and σ_2 verify correctly and $\mathsf{Link}(\sigma_1, \sigma_2, \mathsf{recip}) = 0$. The contradiction is straight forward here, non-linkable signatures would lead to $\sigma_{\mathsf{UCL}1} \neq \sigma_{\mathsf{UCL}2}$, where the fact that both signatures verify correctly against the same recipient name recip, would lead to $\sigma_{\mathsf{UCL}1} = \sigma_{\mathsf{UCL}2}$. In the second case, the adversary aims to break the soundness of the linking algorithm Link by producing supposedly non-linkable signatures (σ_1, σ_2) and yet Link tells that they are linkable. This case can be easily reduced to breaking the linkability property of the LIT scheme, as this can only be done by having $(\mathsf{sk}_1, \mathsf{recip}_1) \neq (\mathsf{sk}_2, \mathsf{recip}_2)$. □

C Non-Interactive Zero-Knowledge Proofs

The properties we require from a non-interctive zero-knowledge proof system are:

- **Completeness:** $\forall\lambda \in \mathbb{N}$, $\forall(x, y) \in \mathsf{R}$, we have

  $$\Pr\left[(\mathsf{crs}, \mathsf{xk}) \leftarrow \mathsf{GSSetup}(1^\lambda); \pi \leftarrow \mathsf{GSProve}(\mathsf{crs}, x, y) : \mathsf{GSVerify}(\mathsf{crs}, y, \pi) = 1\right] = 1$$

- **Soundness:** $\forall\lambda \in \mathbb{N}$, $\forall y \notin \mathcal{L}_{\mathsf{R}}$, we have for all adversaries $\mathcal{F}$

  $$\Pr\left[(\mathsf{crs}, \mathsf{xk}) \leftarrow \mathsf{GSSetup}(1^\lambda); \pi \leftarrow \mathcal{F}(\mathsf{crs}, y) : \mathsf{GSVerify}(\mathsf{crs}, y, \pi) = 1\right] \leq 2^{-\lambda}$$

  If the above probability is 0, we say the system has *perfect soundness*.

- **Knowledge Extraction:** A proof system is a *Proof of Knowledge* if there exists an efficient extractor algorithm $\mathsf{GSExtract}$ which can extract the witness from any proof the adversary outputs. Note that if a proof system is a proof of knowledge then it is sound. More formally, for all adversaries $\mathcal{F}$, we have

$$\Pr\Big[(\mathsf{crs}, \mathsf{xk}) \leftarrow \mathsf{GSSetup}(1^\lambda); (y, \pi) \leftarrow \mathcal{F}(\mathsf{crs}); x \leftarrow \mathsf{GSExtract}(\mathsf{crs}, \mathsf{xk}, y, \pi) \\ : \mathsf{GSVerify}(\mathsf{crs}, y, \pi) = 0 \text{ OR } (x, y) \in \mathsf{R}\Big] \leq 1 - \nu(\lambda)$$

  If the above probability is 1, we say the system has *perfect knowledge extraction*.
- **Zero-Knowledge:** The system is *zero-knowledge* if $\forall (x, y) \in \mathsf{R}$, we have for all PPT adversaries $\mathcal{F}$

$$\Pr\left[(\mathsf{crs}_{\mathrm{sim}}, \mathsf{tr}) \leftarrow \mathsf{GSSimSetup}(1^\lambda) : \mathcal{F}^{\mathsf{GSSim}(\mathsf{crs}_{\mathrm{sim}}, \mathsf{tr}, \cdot, \cdot)}(\mathsf{crs}_{\mathrm{sim}}) = 1\right] \\ \approx \Pr\left[(\mathsf{crs}, \mathsf{xk}) \leftarrow \mathsf{GSSetup}(1^\lambda) : \mathcal{F}^{\mathsf{GSProve}(\mathsf{crs}, \cdot, \cdot)}(\mathsf{crs}) = 1\right],$$

  where $\mathsf{GSSim}(\mathsf{crs}_{\mathrm{sim}}, \mathsf{tr}, x, y)$ outputs $\mathsf{GSSimProve}(\mathsf{crs}_{\mathrm{sim}}, \mathsf{tr}, y)$ if $(x, y) \in \mathsf{R}$ or $\perp$ otherwise.