Short Attribute-Based Signatures for Threshold Predicates

Javier Herranz[1], Fabien Laguillaume[2], Benoît Libert[3], and Carla Ràfols[4]

[1] Universitat Politècnica de Catalunya, Dept. Matemàtica Aplicada IV (Spain)
[2] Université de Caen Basse Normandie and CNRS/ENSL/INRIA/UCBL LIP, Lyon (France)
[3] Université Catholique de Louvain, ICTEAM Institute – Crypto Group (Belgium)
[4] Universitat Rovira i Virgili, UNESCO Chair in Data Privacy, Tarragona (Catalonia)

Abstract. Attribute-based cryptography is a natural solution for fine-grained access control with respect to security policies. In the case of attribute-based signatures (ABS), users obtain from an authority their secret keys as a function of the attributes they hold, with which they can later sign messages for any predicate satisfied by their attributes. A verifier will be convinced of the fact that the signer's attributes satisfy the signing predicate while remaining completely ignorant of the identity of the signer. In many scenarios where authentication and anonymity are required, like distributed access control mechanisms in ad hoc networks, the bandwidth is a crucial and sensitive concern. The signatures' size of all previous ABS schemes grows linearly in the number of attributes involved in the signing predicate. We propose the first two attribute-based signature schemes with constant size signatures. Their security is proven in the selective-predicate and adaptive-message setting, in the standard model, under chosen message attacks, with respect to some algorithmic assumptions related to bilinear groups. The described schemes are for the case of threshold predicates, but they can be extended to admit some other (more expressive) kinds of monotone predicates.

1 Introduction

Attribute-based cryptography offers a real alternative to public-key cryptography when the systems to be protected also require anonymity among users following a security policy. In this setting, users obtain their secret keys from an authority as a function of their attributes. The operation involving the secret key proves somehow that the user holds a certain subset of attributes, without leaking information on his identity or on his total set of attributes.

One of the major issues in attribute-based cryptography is to save bandwidth, and in particular to get ciphertexts or signatures of constant size, *i.e.,* not depending on the number of involved attributes. Other important issues are the construction of systems achieving security in the strongest possible model and being as expressive as possible, *i.e.,* admitting a wide variety of policies.

The goal of this work is to address the first question in the context of signature design.

Attribute-based cryptography first appeared in [15] with an attribute-based encryption scheme, as an extension of fuzzy identity-based cryptosystems [29]. Since then, the notion of attribute-based encryption (ABE for short, conjugated into *key policy* or *ciphertext policy*) has received a lot of attention (see, e.g., [2, 17, 20]), notably with attempts to compress ciphertexts (see [13, 17, 1]).

Attribute-based signatures (ABS) have been explicitly introduced more recently in [24] (see also [30, 21, 22]), although the idea was implicitly considered before (for instance, in [10]). They are related to the notion of (threshold) ring signatures [28, 9] or mesh signatures [8], but offer much more flexibility and versatility to design secure complex systems, since the signatures are linked not to the users themselves, but to their attributes. As a consequence, these signatures have a wide range of applications, like private access control, anonymous credentials, trust negotiations, distributed access control mechanisms for ad hoc networks or attribute-based messaging (see [24] for detailed descriptions of applications). In terms of security, ABS must first satisfy unforgeability, which guarantees that a signature cannot be computed by a user who does not have the right attributes, even if he colludes with other users by pooling together their secret keys. The other security requirement is the privacy of user's attributes, in the sense that a signature should not leak any information about the actual attributes that have been employed to produce it.

Related work. The schemes proposed by Maji, Prabhakaran, Rosulek in [24] support very expressive signing predicates, but their most practical one is only proven secure in the generic group model. The scheme of [27] is claimed to be "almost optimally efficient", although its signatures' length grows linearly in the size of the span program (which is greater than the number of involved attributes in the signing predicate). Our result shows that specific families of predicates (e.g., threshold predicates) allow for more compact signatures. Other instantiations in [24] are secure in the standard model, but are substantially less inefficient (*i.e.,* signatures consist of a linear number of group elements in the security parameter) as they use Groth-Sahai proofs for relations between the bits of elements in the group. In the standard model, Okamoto and Takashima designed [27] a *fully* secure ABS which supports general non-monotone predicates. The scheme is built upon dual pairing vector spaces [26] and uses proof techniques from functional encryption [20]. Escala, Herranz and Morillo also proposed in [14] a fully secure ABS in the standard model, with the additional property of revocability, meaning that a third party can extract the identity of a signer in case of dispute (thanks to a secret that can be computed by the master entity). As it turns out, *none* of the previous schemes achieves constant-size signatures.

Our contribution. This paper describes the first two threshold ABS schemes featuring constant-size signatures and proves them secure in the selective-predicate setting (*i.e.,* as opposed to the *full* security setting) in the standard model. We hope our results will inspire ideas leading to the design of fully secure ABS

schemes with constant-size signatures and supporting more expressive predicates than in this paper. The new schemes are built (non-generically) on two different constant-size attribute-based encryption schemes. In both schemes, n denotes the maximum size of the admitted signing predicates.

- Our first scheme supports (weighted) threshold predicates for small[1] universes of attributes. Its design is inspired by the constant-size ciphertext-policy ABE scheme from [17] by Herranz, Laguillaumie and Ràfols, in the sense that the signer implicitly proves his ability to decrypt a ciphertext by using the Groth-Sahai proof systems [16], and by binding the signed message (and the corresponding predicate) to the signature using a technique suggested by Malkin, Teranishi, Vahlis and Yung [23]. The signature consists of 15 group elements, and the secret key of a user holding a set Ω of attributes has $|\Omega|+n$ elements. Our scheme is selective-predicate and adaptive-message unforgeable under chosen message attacks if the augmented multi-sequence of exponents computational Diffie-Hellman assumption [17] and the Decision Linear assumption [5] hold. The privacy of the attributes used to sign is proved in the computational sense under the Decision Linear assumption [5].
- The second scheme supports threshold predicates (as well as compartmented and hierarchical predicates) for *large* universes of attributes, which can be obtained by hashing arbitrary strings. It is built upon a key-policy ABE scheme proposed by Attrapadung, Libert and de Panafieu [1] and has signatures consisting of *only* 3 group elements. The secret keys are longer than in the first scheme, as they include $(2n+2) \times (|\Omega|+n)$ group elements. On the other hand, its selective-predicate and adaptive-message unforgeability relies on the more classical n-Diffie-Hellman exponent assumption. Moreover, the scheme protects the privacy of the involved attributes unconditionally.

Organization. Section 2 gives the algorithmic setting and defines the syntax and the security properties of attribute-based signatures. In Sections 3 and 4 we describe our two constructions for threshold predicates. Section 5 discusses extensions of both schemes to more general predicates.

2 Background

We will treat a vector as a column vector. For any $\vec{\alpha} = (\alpha_1, \ldots, \alpha_n)^\top \in \mathbb{Z}_p^n$, and any element g of a group $\mathbb{G}$, $g^{\vec{\alpha}}$ stands for $(g^{\alpha_1}, \ldots, g^{\alpha_n})^\top \in \mathbb{G}^n$. The inner product of $\vec{a}, \vec{z} \in \mathbb{Z}_p^n$ is denoted as $\langle \vec{a}, \vec{z} \rangle = \vec{a}^\top \vec{z}$. Given $g^{\vec{a}}$ and $\vec{z}$, $(g^{\vec{a}})^{\vec{z}} := g^{\langle \vec{a}, \vec{z} \rangle}$ is computable without knowing $\vec{a}$. For equal-dimension vectors $\vec{A}$ and $\vec{B}$ of exponents or group elements, $\vec{A} \cdot \vec{B}$ stands for their component-wise product. We denote by I_n the identity matrix of size n. For any set U, we define $2^U = \{S \mid S \subseteq U\}$. Given a set $S \subset \mathbb{Z}_p$, and some $i \in S$, the i-th Lagrange basis polynomial is $\Delta_i^S(X) = \prod_{j \in S \setminus \{i\}} (X-j)/(i-j)$.

[1] *i.e.* polynomial in the security parameter, which is sufficient for many applications.

2.1 Complexity Assumptions

Our two schemes work in the setting of bilinear groups. That is, we use a pair of multiplicative groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order p with an efficiently computable mapping $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ s.t. $e(g^a, h^b) = e(g, h)^{ab}$ for any $(g, h) \in \mathbb{G} \times \mathbb{G}$, $a, b \in \mathbb{Z}$ and $e(g, h) \neq 1_{\mathbb{G}_T}$ whenever $g, h \neq 1_{\mathbb{G}}$.

The security of our first scheme is partially based on the hardness of the computational version of a problem appeared in [17] under the name of *augmented multi-sequence of exponents decisional Diffie-Hellman problem*. Its decisional version was proven to be hard in generic groups.

Definition 1 ($(\tilde{\ell}, \tilde{m}, \tilde{t})$-aMSE-CDH - [17]). *The* $(\tilde{\ell}, \tilde{m}, \tilde{t})$-augmented multi-sequence of exponents computational Diffie-Hellman *($(\tilde{\ell}, \tilde{m}, \tilde{t})$-aMSE-CDH) problem related to the group pair* $(\mathbb{G}, \mathbb{G}_T)$ *is to compute* $T = e(g_0, h_0)^{\kappa \cdot f(\gamma)}$ *when* $\kappa, \alpha, \gamma, \omega$ *are unknown random elements of* $\mathbb{Z}_p$ *and* g_0 *and* h_0 *are generators of* $\mathbb{G}$ *on input the vector* $\vec{x}_{\tilde{\ell}+\tilde{m}} = (x_1, \ldots, x_{\tilde{\ell}+\tilde{m}})^\top$*, whose components are pairwise distinct elements of* $\mathbb{Z}_p$*, and the values*

$$\begin{cases} g_0, g_0^{\gamma}, \ldots, g_0^{\gamma^{\tilde{\ell}+\tilde{t}-2}}, & g_0^{\kappa \cdot \gamma \cdot f(\gamma)}, & \text{(l.1)} \\ g_0^{\omega\gamma}, \ldots, g_0^{\omega\gamma^{\tilde{\ell}+\tilde{t}-2}}, & & \text{(l.2)} \\ g_0^{\alpha}, g_0^{\alpha\gamma}, \ldots, g_0^{\alpha\gamma^{\tilde{\ell}+\tilde{t}}}, & & \text{(l.3)} \\ h_0, h_0^{\gamma}, \ldots, h_0^{\gamma^{\tilde{m}-2}}, & h_0^{\kappa \cdot g(\gamma)} & \text{(l.4)} \\ h_0^{\omega}, h_0^{\omega\gamma}, \ldots, h_0^{\omega\gamma^{\tilde{m}-1}}, & & \text{(l.5)} \\ h_0^{\alpha}, h_0^{\alpha\gamma}, \ldots, h_0^{\alpha\gamma^{2(\tilde{m}-\tilde{t})+3}} & & \text{(l.6)}, \end{cases}$$

where $f(X) = \prod_{i=1}^{\tilde{\ell}} (X + x_i)$ *and* $g(X) = \prod_{i=\tilde{\ell}+1}^{\tilde{\ell}+\tilde{m}} (X + x_i)$.

The security analysis of our first scheme also relies on the Decision Linear assumption.

Definition 2 ([5]). *In a group* $\mathbb{G}$ *of order* p*, the* Decision Linear Problem *(DLIN) is to distinguish the distributions* $(g, g^a, g^b, g^{a \cdot \delta_1}, g^{b \cdot \delta_2}, g^{\delta_1 + \delta_2})$ *and* $(g, g^a, g^b, g^{a \cdot \delta_1}, g^{b \cdot \delta_2}, g^{\delta_3})$*, with* $a, b, \delta_1, \delta_2, \delta_3 \xleftarrow{R} \mathbb{Z}_p$.

This problem is to decide if vectors $\vec{g_1} = (g^a, 1, g)^\top$, $\vec{g_2} = (1, g^b, g)^\top$ and $\vec{g_3} = (g^{a\delta_1}, g^{b\delta_2}, g^{\delta_3})^\top$ are linearly dependent in the $\mathbb{Z}_p$-module $\mathbb{G}^3$ formed by entry-wise multiplication.

The security of our second scheme is based on a non-interactive and falsifiable [25] assumption, the hardness of n-Diffie-Hellman Exponent problem, proven to hold in generic groups in [4].

Definition 3 ([6]). *In a group* $\mathbb{G}$ *of prime order* p*, the* n-Diffie-Hellman Exponent *(n-DHE) problem is, given a tuple* $(g, g^{\gamma}, g^{\gamma^2}, \ldots, g^{\gamma^n}, g^{\gamma^{n+2}}, \ldots, g^{\gamma^{2n}})$ *where* $\gamma \xleftarrow{R} \mathbb{Z}_p$*,* $g \xleftarrow{R} \mathbb{G}$*, to compute* $g^{\gamma^{n+1}}$.

2.2 Groth-Sahai Proof Systems

To simplify the description, our first scheme uses Groth-Sahai proofs based on the DLIN assumption and symmetric pairings, although instantiations based on the symmetric external Diffie-Hellman assumption are also possible. In the DLIN setting, the Groth-Sahai proof systems [16] use a common reference string comprising vectors $\vec{g_1}, \vec{g_2}, \vec{g_3} \in \mathbb{G}^3$, where $\vec{g_1} = (g_1, 1, g)^\top$, $\vec{g_2} = (1, g_2, g)^\top$ for some $g_1, g_2, g \in \mathbb{G}$. To commit to $X \in \mathbb{G}$, one sets $\vec{C} = (1, 1, X)^\top \cdot \vec{g_1}^r \cdot \vec{g_2}^s \cdot \vec{g_3}^t$ with $r, s, t \xleftarrow{R} \mathbb{Z}_p$. In the soundness setting (*i.e.*, when proofs should be perfectly sound), $\vec{g_3}$ is set as $\vec{g_3} = \vec{g_1}^{\xi_1} \cdot \vec{g_2}^{\xi_2}$ with $\xi_1, \xi_2 \xleftarrow{R} \mathbb{Z}_p^*$. Commitments $\vec{C} = (g_1^{r+\xi_1 t}, g_2^{s+\xi_2 t}, X \cdot g^{r+s+t(\xi_1+\xi_2)})^\top$ are then Boneh-Boyen-Shacham (BBS) ciphertexts [5] that can be decrypted using $a = \log_g(g_1)$, $b = \log_g(g_2)$.

In contrast, defining $\vec{g_3} = \vec{g_1}^{\xi_1} \cdot \vec{g_2}^{\xi_2} \cdot (1, 1, g^{-1})^\top$ gives linearly independent $\{\vec{g_1}, \vec{g_2}, \vec{g_3}\}$ and $\vec{C}$ is a perfectly hiding commitment. Moreover, proofs are perfectly witness indistinguishable (WI) in that two proofs generated using any two distinct witnesses are perfectly indistinguishable. Under the DLIN assumption, the WI and the soundness setting are computationally indistinguishable.

To prove that committed group elements satisfy certain relations, the Groth-Sahai techniques require one commitment per variable and one proof element (made of a constant number of group elements) per relation. Such proofs are available for pairing-product relations, which are of the type

$$\prod_{i=1}^{n} e(\mathcal{A}_i, \mathcal{X}_i) \cdot \prod_{i=1}^{n} \cdot \prod_{j=1}^{n} e(\mathcal{X}_i, \mathcal{X}_j)^{a_{ij}} = t_T, \tag{1}$$

for variables $\mathcal{X}_1, \ldots, \mathcal{X}_n \in \mathbb{G}$ and constants $t_T \in \mathbb{G}_T$, $\mathcal{A}_1, \ldots, \mathcal{A}_n \in \mathbb{G}$, $a_{ij} \in \mathbb{Z}_p$, for $i, j \in \{1, \ldots, n\}$.

At some additional cost (typically, auxiliary variables have to be introduced), pairing-product equations admit non-interactive zero-knowledge (NIZK) proofs (this is the case when the target element t_T has the special form $t_T = \prod_{i=1}^{t} e(S_i, T_i)$, for constants $\{(S_i, T_i)\}_{i=1}^{t}$ and some $t \in \mathbb{N}$): on a simulated common reference string (CRS), prepared for the WI setting, a trapdoor makes it possible to simulate proofs without knowing the witnesses. Linear pairing product equations (where $a_{ij} = 0$ for all i, j in (1)) consist of only 3 group elements and we only need linear equations here.

2.3 Syntax of Threshold Attribute-Based Signatures

We describe the syntax and security model of attribute-based signatures with respect to *threshold* signing predicates $\Gamma = (t, S)$, but the algorithms and security model for more general signing predicates can be described in a very similar way. In the threshold case, every message Msg is signed for a subset S of the universe of attributes and a threshold t such that $1 \leq t \leq |S|$ of the sender's choice.

An *attribute-based signature* ABS = (ABS.TSetup, ABS.MSetup, ABS.Keygen, ABS.Sign, ABS.Verify) consists of five probabilistic polynomial-time (PPT) algorithms:

- $\mathsf{TSetup}(\lambda, \mathcal{P}, n)$: is the randomized *trusted setup* algorithm taking as input a security parameter λ, an attribute universe $\mathcal{P}$ and an integer $n \in \mathsf{poly}(\lambda)$ which is an upper bound on the size of threshold policies. It outputs a set of public parameters pms (which contains λ, $\mathcal{P}$ and n). An execution of this algorithm is denoted as $\mathsf{pms} \leftarrow \mathsf{ABS.TSetup}(1^\lambda, \mathcal{P}, n)$.
- $\mathsf{MSetup}(\mathsf{pms})$: is the randomized *master setup* algorithm, that takes as input pms and outputs a master secret key msk and the corresponding master public key mpk.We write $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{ABS.MSetup}(\mathsf{pms})$ to denote an execution of this algorithm.
- $\mathsf{Keygen}(\mathsf{pms}, \mathsf{mpk}, \mathsf{msk}, \Omega)$: is a *key extraction* algorithm that takes in public parameters pms, the master keys mpk and msk, and an attribute set $\Omega \subset \mathcal{P}$. The output is a private key SK_Ω. To denote an execution of this algorithm, we write $SK_\Omega \leftarrow \mathsf{ABS.Keygen}(\mathsf{pms}, \mathsf{mpk}, \mathsf{msk}, \Omega)$.
- $\mathsf{Sign}(\mathsf{pms}, \mathsf{mpk}, SK_\Omega, \mathsf{Msg}, \Gamma)$: is a randomized *signing* algorithm which takes as input the public parameters pms, the master public key mpk, a secret key SK_Ω, a message Msg and a threshold signing policy $\Gamma = (t, S)$ where $S \subset \mathcal{P}$ and $1 \leq t \leq |S| \leq n$. It outputs a signature σ. We denote the action taken by the signing algorithm as $\sigma \leftarrow \mathsf{ABS.Sign}(\mathsf{pms}, \mathsf{mpk}, SK_\Omega, \mathsf{Msg}, \Gamma)$.
- $\mathsf{Verify}(\mathsf{pms}, \mathsf{mpk}, \mathsf{Msg}, \sigma, \Gamma)$: is a deterministic *verification* algorithm taking as input the public parameters pms, a master public key mpk, a message Msg, a signature σ and a threshold predicate $\Gamma = (t, S)$. It outputs 1 if the signature is deemed valid and 0 otherwise. To refer to an execution of the verification protocol we write $b \leftarrow \mathsf{ABS.Verify}(\mathsf{pms}, \mathsf{mpk}, \mathsf{Msg}, \sigma, \Gamma)$.

For correctness, for any $\lambda \in \mathbb{N}$, any integer $n \in \mathsf{poly}(\lambda)$, any universe $\mathcal{P}$, any set of public parameters $\mathsf{pms} \leftarrow \mathsf{ABS.TSetup}(1^\lambda, \mathcal{P}, n)$, any master key pair $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{ABS.MSetup}(\mathsf{pms})$, any subset $\Omega \subset \mathcal{P}$ and any threshold policy $\Gamma = (t, S)$ where $1 \leq t \leq |S|$, it is required that

$$\mathsf{ABS.Verify}\big(\mathsf{pms}, \mathsf{mpk}, \mathsf{Msg}, \mathsf{ABS.Sign}(\mathsf{pms}, \mathsf{mpk}, SK_\Omega, \mathsf{Msg}, \Gamma), \Gamma\big) \;=\; 1$$

whenever $SK_\Omega \leftarrow \mathsf{ABS.Keygen}(\mathsf{pms}, \mathsf{mpk}, \mathsf{msk}, \Omega)$ and $|\Omega \cap S| \geq t$.

2.4 Security of Threshold Attribute-Based Signatures

Unforgeability and privacy are the typical requirements for attribute-based signature schemes.

Unforgeability. An ABS scheme must satisfy the usual property of unforgeability, even against a group of colluding users that pool their secret keys. We consider a relaxed notion where the attacker *selects* the signing policy $\Gamma^\star = (t^\star, S^\star)$ that he wants to attack at the beginning of the game. However, the message $\mathsf{Msg}^\star$ whose signature is eventually forged is not selected in advance. The attacker can ask for valid signatures for messages and signing policies of his adaptive choice. The resulting property of *selective-predicate and adaptive-message unforgeability under chosen message attacks* ($\mathsf{sP\text{-}UF\text{-}CMA}$, for short) is defined by considering the following game.

Definition 4. *Let λ be an integer. Consider the following game between a probabilistic polynomial time (PPT) adversary $\mathcal{F}$ and its challenger.*

Initialization. *The challenger begins by specifying a universe of attributes $\mathcal{P}$ as well as an integer $n \in \mathsf{poly}(\lambda)$, which are sent to $\mathcal{F}$. Then, $\mathcal{F}$ selects a subset $S^\star \subset \mathcal{P}$ of attributes such that $|S^\star| \leq n$ and a threshold $t^\star \in \{1, \ldots, |S^\star|\}$. These define a threshold predicate $\Gamma^\star = (t^\star, S^\star)$.*

Setup. *The challenger runs $\mathsf{pms} \leftarrow \mathit{ABS.TSetup}(1^\lambda, \mathcal{P}, n)$ and $(\mathsf{mpk}, \mathit{msk}) \leftarrow \mathit{ABS.MSetup}(\mathsf{pms})$, and sends $\mathsf{pms}, \mathsf{mpk}$ to the forger $\mathcal{F}$.*

Queries. *$\mathcal{F}$ can interleave private key and signature queries.*

- ***Private key queries.*** *$\mathcal{F}$ adaptively chooses a subset of attributes $\Omega \subset \mathcal{P}$ under the restriction that $|\Omega \cap S^\star| < t^\star$ and must receive $SK_\Omega \leftarrow \mathit{ABS.Keygen}(\mathsf{pms}, \mathsf{mpk}, \mathsf{msk}, \Omega)$ as the answer.*
- ***Signature queries.*** *$\mathcal{F}$ adaptively chooses a pair (Msg, Γ) consisting of a message Msg and a threshold predicate $\Gamma = (t, S)$ such that $1 \leq t \leq |S| \leq n$. The challenger chooses an arbitrary attribute set $\Omega \subset \mathcal{P}$ such that $|\Omega \cap S| \geq t$, runs $SK_\Omega \leftarrow \mathit{ABS.Keygen}(\mathsf{pms}, \mathsf{mpk}, \mathsf{msk}, \Omega)$ and computes[2] a signature $\sigma \leftarrow \mathit{ABS.Sign}(\mathsf{pms}, \mathsf{mpk}, SK_\Omega, \mathsf{Msg}, \Gamma)$ which is returned to $\mathcal{F}$.*

Forgery. *At the end of the game, $\mathcal{F}$ outputs a pair $(\mathsf{Msg}^\star, \sigma^\star)$. We say that $\mathcal{F}$ is successful if:*

- *$\mathit{ABS.Verify}(\mathsf{pms}, \mathsf{mpk}, \mathsf{Msg}^\star, \sigma^\star, \Gamma^\star) = 1$, and*
- *$\mathcal{F}$ has not made any signature query for the pair $(\mathsf{Msg}^\star, \Gamma^\star)$.*

The forger's advantage in breaking the sP-UF-CMA security of the scheme is defined as $\mathsf{Succ}^{\mathit{sP\text{-}UF\text{-}CMA}}_{\mathcal{F},\mathit{ABS}}(\lambda) = \Pr[\mathcal{F}\ \mathit{wins}]$. A threshold attribute-based signature scheme ABS is selective-predicate adaptive-message unforgeable (or sP-UF-CMA unforgeable) *if, for any PPT adversary $\mathcal{F}$, $\mathsf{Succ}^{\mathit{sP\text{-}UF\text{-}CMA}}_{\mathcal{F},\mathit{ABS}}(\lambda)$ is a negligible function of λ.*

Privacy (of Involved Attributes). This property ensures that a signature leaks nothing about the attributes that have been used to produce it beyond the fact that they satisfy the signing predicate. Privacy must hold even against attackers that control the master entity and is defined *via* a game between an adversary $\mathcal{D}$ and its challenger. Depending on the resources allowed to $\mathcal{D}$ and on its success probability, we can define computational privacy and perfect (unconditional) privacy.

Definition 5. *Let $\lambda \in \mathbb{N}$ and consider this game between a distinguisher $\mathcal{D}$ and its challenger.*

Setup. *The adversary $\mathcal{D}$ specifies a universe of attributes $\mathcal{P}$ and an integer $n \in \mathsf{poly}(\lambda)$, that are sent to the challenger. The challenger runs $\mathsf{pms} \leftarrow$*

[2] Since a given attribute set Ω may have many valid private keys SK_Ω, a generalization of the definition could allow $\mathcal{F}$ to obtain many signatures from the same private key SK_Ω. However, due to the signer privacy requirement, which is formalized hereafter, this does not matter.

$\mathit{ABS.TSetup}(1^\lambda, \mathcal{P}, n)$ *and sends* pms *to* $\mathcal{D}$. *The adversary* $\mathcal{D}$ *runs* $(\mathsf{mpk}, \mathit{msk}) \leftarrow \mathit{ABS.MSetup}(\mathsf{pms})$ *and sends* $(\mathsf{mpk}, \mathit{msk})$ *to the challenger (who must verify consistency of this master key pair).*

Challenge. $\mathcal{D}$ *outputs a tuple* $(\Gamma, \Omega_0, \Omega_1, \mathsf{Msg})$, *where* $\Gamma = (t, S)$ *is a threshold predicate such that* $1 \leq t \leq |S| \leq n$ *and* Ω_0, Ω_1 *are attribute sets satisfying* $|\Omega_b \cap S| \geq t$ *for each* $b \in \{0,1\}$. *The challenger picks a random bit* $\beta \xleftarrow{R} \{0,1\}$, *runs* $SK_{\Omega_\beta} \leftarrow \mathit{ABS.Keygen}(\mathsf{pms}, \mathsf{mpk}, \mathsf{msk}, \Omega_\beta)$ *and computes* $\sigma^\star \leftarrow \mathit{ABS.Sign}(\mathsf{pms}, \mathsf{mpk}, SK_{\Omega_\beta}, \mathsf{Msg}, \Gamma)$, *which is sent as a challenge to* $\mathcal{A}$.

Guess. $\mathcal{D}$ *outputs a bit* $\beta' \in \{0,1\}$ *and wins if* $\beta' = \beta$.

The advantage of $\mathcal{D}$ *is measured in the usual way, as the distance* $\mathsf{Adv}^{\mathit{Priv}}_{\mathcal{D},\mathit{ABS}}(\lambda) := |\Pr[\beta' = \beta] - \frac{1}{2}|$.

A threshold attribute-based signature scheme ABS *is said* computationally private *if* $\mathsf{Adv}^{\mathit{Priv}}_{\mathcal{D},\mathit{ABS}}(\lambda)$ *is a negligible function of* λ *for any PPT distinguisher* $\mathcal{D}$ *and it is said* perfectly/unconditionally private *if* $\mathsf{Adv}^{\mathit{Priv}}_{\mathcal{D},\mathit{ABS}}(\lambda) = 0$ *for any (possibly computationally unbounded) distinguisher* $\mathcal{D}$.

3 A First Short Attribute-Based Signature Scheme for Threshold Predicates

We present here our first scheme to produce attribute-based signatures with constant size, for threshold predicates. The secret key sk_Ω for a user holding a set of attributes Ω contains $|\Omega| + n$ elements, where n is the maximum size of the attribute set for any signing policy. This construction is for "small" universes of attributes $\mathcal{P} = \{\mathsf{at}_1, \ldots, \mathsf{at}_\eta\}$, for some integer $\eta \in \mathbb{N}$, as public parameters have linear size in η; therefore, η must be polynomial in the security parameter of the scheme. Attributes $\{\mathsf{at}_i\}_{i=1}^{\eta}$ are arbitrary strings which some encoding function ς maps to $\mathbb{Z}_p^*$. Since the scheme is a small universe construction, we may set $n = \eta$ in the description hereafter.

The construction builds on the ABE scheme of Herranz *et al.* [17]. The intuition is to have the signer implicitly prove his ability to decrypt a ciphertext corresponding to that ABE scheme. This non-interactive proof is generated using the Groth-Sahai proof systems [16], by binding the signed message (and the corresponding predicate) to the non-interactive proof using a technique suggested by Malkin *et al.* [23]. In some sense, this technique can be seen as realizing signatures of knowledge in the standard model: it consists in embedding the message to be signed in the Groth-Sahai CRS by calculating part of the latter as a "hash value" of the message. As noted in [23], Waters' hash function [32] is well-suited to this purpose since, in the security proof, it makes it possible to answer signing queries using simulated NIZK proofs. At the same time, with non-negligible probability, adversarially-generated signatures are produced using a perfectly sound Groth-Sahai CRS and they thus constitute real proofs, from which witnesses can be extracted.

In [23], the above technique was applied to an instantiation of Groth-Sahai

proofs based on the Symmetric eXternal Diffie-Hellman assumption (and thus asymmetric pairings). In this section, we adapt this technique so as to get it to work with symmetric pairings and the linear assumption.

In the notations of the verification algorithm, when $\vec{C} = (C_1, C_2, C_3)^\top \in \mathbb{G}^3$ is a vector of group elements and if $g \in \mathbb{G}$, we denote by $E(g, \vec{C})$ the vector of pairing values $\big(e(g, C_1), e(g, C_2), e(g, C_3)\big)^\top$.

▶ $\mathsf{TSetup}(\lambda, \mathcal{P}, n)$: the trusted setup algorithm conducts the following steps.

1. Choose groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$ with an efficiently computable bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Select generators $g, h \xleftarrow{R} \mathbb{G}$ and also choose a collision-resistant hash function $H : \{0,1\}^* \to \{0,1\}^k$, for some $k \in \mathsf{poly}(\lambda)$.
2. Define a suitable injective encoding ς sending each one of the n attributes $\mathsf{at} \in \mathcal{P}$ onto an element $\varsigma(\mathsf{at}) = x \in \mathbb{Z}_p^\star$. Choose a set $\mathcal{D} = \{d_1, \ldots, d_{n-1}\}$ consisting of $n-1$ pairwise different elements of $\mathbb{Z}_p^*$, which must also be different from the encoding of any attribute in $\mathcal{P}$. For any integer i lower or equal to $n-1$, we denote as $\mathcal{D}_i$ the set $\{d_1, \ldots, d_i\}$.
3. Generate Groth-Sahai reference strings by choosing random generators $g_1, g_2 \xleftarrow{R} \mathbb{G}$ and defining vectors $\vec{g_1} = (g_1, 1, g)^\top \in \mathbb{G}^3$ and $\vec{g_2} = (1, g_2, g)^\top \in \mathbb{G}^3$. Then, for each $i \in \{0, \ldots, k\}$, pick $\xi_{i,1}, \xi_{i,2} \xleftarrow{R} \mathbb{Z}_p$ at random and define a vector $\vec{g_{3,i}} = \vec{g_1}^{\xi_{i,1}} \cdot \vec{g_2}^{\xi_{i,2}} = \big(g_1^{\xi_{i,1}}, g_2^{\xi_{i,2}}, g^{\xi_{i,1}+\xi_{i,2}}\big)^\top$. Exponents $\{(\xi_{i,1}, \xi_{i,2})\}_{i=0}^k$ can then be discarded as they are no longer needed.

The resulting public parameters are

$$\mathsf{pms} = \Big(\mathcal{P},\ n,\ \lambda,\ \mathbb{G},\ \mathbb{G}_T,\ g,\ h,\ \vec{g_1},\ \vec{g_2},\ \{\vec{g_{3,i}}\}_{i=0}^k,\ H,\ \varsigma,\ \mathcal{D}\Big).$$

▶ $\mathsf{MSetup}(\mathsf{pms})$: picks at random $\alpha, \gamma \xleftarrow{R} \mathbb{Z}_p^*$ and sets $u = g^{\alpha\gamma}$ and $v = e(g^\alpha, h)$. The master secret key is $\mathsf{msk} = (\alpha, \gamma)$ and the master public key consists of

$$\mathsf{mpk} = \Big(\ u,\ v,\ g^\alpha,\ \Big\{h^{\alpha\gamma^i}\Big\}_{i=0,\ldots,2n-1}\Big).$$

▶ $\mathsf{Keygen}(\mathsf{pms}, \mathsf{mpk}, \mathsf{msk}, \Omega)$: given an attribute set Ω and $\mathsf{msk} = (\alpha, \gamma)$, pick $r \xleftarrow{R} \mathbb{Z}_p^*$ and compute

$$SK_\Omega = \Big(\ \Big\{g^{\frac{r}{\gamma+\varsigma(\mathsf{at})}}\Big\}_{\mathsf{at}\in\Omega}, \Big\{h^{r\gamma^i}\Big\}_{i=0,\ldots,n-2}, h^{\frac{r-1}{\gamma}}\Big). \tag{2}$$

▶ $\mathsf{Sign}(\mathsf{pms}, \mathsf{mpk}, SK_\Omega, \mathsf{Msg}, \Gamma)$: to sign $\mathsf{Msg} \in \{0,1\}^*$ w.r.t. the policy $\Gamma = (t, S)$, where $S \subset \mathcal{P}$ is an attribute set of size $s = |S| \leq n$ and $1 \leq t \leq s \leq n$, the algorithm returns $\perp$ if $|\Omega \cap S| < t$. Otherwise, it first parses SK_Ω as in (2) and conducts the following steps.

1. Let Ω_S be any subset of $\Omega \cap S$ with $|\Omega_S| = t$. From all $\mathsf{at} \in \Omega_S$, using the algorithm $\mathsf{Aggregate}$ of [12], compute the value

$$A_1 = \mathsf{Aggregate}(\{g^{\frac{r}{\gamma+\varsigma(\mathsf{at})}}, \varsigma(\mathsf{at})\}_{\mathsf{at}\in\Omega_S}) = g^{\frac{r}{\prod_{\mathsf{at}\in\Omega_S}(\gamma+\varsigma(\mathsf{at}))}}.$$

From A_1, compute $T_1 = A_1^{\frac{1}{\prod_{\mathsf{at}\in(S\cup\mathcal{D}_{n+t-1-s})\setminus\Omega_S}\varsigma(\mathsf{at})}}$.

2. Define the value $P_{(\Omega_S,S)}(\gamma)$ as

$$P_{(\Omega_S,S)}(\gamma) = \frac{1}{\gamma}\left(\prod_{\mathsf{at}\in(S\cup\mathcal{D}_{n+t-1-s})\setminus\Omega_S}(\gamma+\varsigma(\mathsf{at})) - \prod_{\mathsf{at}\in(S\cup\mathcal{D}_{n+t-1-s})\setminus\Omega_S}\varsigma(\mathsf{at})\right).$$

Since $|\Omega_S| = t$, the degree of $P_{(\Omega_S,S)}(X)$ is $n-2$. Therefore, from the private key SK_Ω, one can compute $h^{r\cdot P_{(\Omega_S,S)}(\gamma)/(\prod_{\mathsf{at}\in(S\cup\mathcal{D}_{n+t-1-s})\setminus\Omega_S}\varsigma(\mathsf{at}))}$ and multiply it with the last element $h^{\frac{r-1}{\gamma}}$ of SK_Ω to obtain

$$T_2 = h^{\frac{r-1}{\gamma}} \cdot h^{r\frac{P_{(\Omega_S,S)}(\gamma)}{\prod_{\mathsf{at}\in(S\cup\mathcal{D}_{n+t-1-s})\setminus\Omega_S}\varsigma(\mathsf{at})}}.$$

Note that the obtained values $T_1, T_2 \in \mathbb{G}$ satisfy the equality

$$e(T_2, u^{-1}) \cdot e\Big(T_1, h^{\alpha\cdot\prod_{\mathsf{at}\in(S\cup\mathcal{D}_{n+t-1-s})}(\gamma+\varsigma(\mathsf{at}))}\Big) \;=\; e(g^\alpha, h) \tag{3}$$

and that, in the terms in the left-hand-side of equality (3), the second argument of each pairing is publicly computable using pms and mpk.

3. Compute $M = m_1 \ldots m_k = H(\mathsf{Msg}, \Gamma) \in \{0,1\}^k$ and use M to form a message-specific Groth-Sahai CRS $\mathbf{g}_M = (\vec{g_1}, \vec{g_2}, \vec{g}_{3,M})$. Namely, for $i = 0$ to k, parse $\vec{g_{3,i}}$ as $(g_{X,i}, g_{Y,i}, g_{Z,i})^\top \in \mathbb{G}^3$. Then, define the vector $\vec{g}_{3,M} = \big(g_{X,0}\cdot\prod_{i=1}^k g_{X,i}^{m_i},\ g_{Y,0}\cdot\prod_{i=1}^k g_{Y,i}^{m_i},\ g_{Z,0}\cdot\prod_{i=1}^k g_{Z,i}^{m_i}\big)^\top$.
4. Using the newly defined $\mathbf{g}_M = (\vec{g_1}, \vec{g_2}, \vec{g}_{3,M})$, generate Groth-Sahai commitments to T_1 and T_2. Namely, pick $r_1, s_1, t_1, r_2, s_2, t_2 \xleftarrow{R} \mathbb{Z}_p$ and compute $\vec{C}_{T_j} = (1, 1, T_j)^\top \cdot \vec{g_1}^{r_j} \cdot \vec{g_2}^{s_j} \cdot \vec{g}_{3,M}^{\,t_j}$ for $j \in \{1,2\}$. Then, generate a NIZK proof that committed variables (T_1, T_2) satisfy the pairing-product equation (3). To this end, we introduce an auxiliary variable $\Theta \in \mathbb{G}$ (with its own commitment $\vec{C}_\Theta = (1,1,\Theta)^\top \cdot \vec{g_1}^{r_\theta} \cdot \vec{g_2}^{s_\theta} \cdot \vec{g}_{3,M}^{\,t_\theta}$, for $r_\theta, s_\theta, t_\theta \xleftarrow{R} \mathbb{Z}_p$), which takes on the value $\Theta = h$, and actually prove that

$$e(T_1, H_S) = e(g^\alpha, \Theta) \cdot e(T_2, u) \tag{4}$$
$$e(g, \Theta) = e(g, h), \tag{5}$$

where $H_S = h^{\alpha\cdot\prod_{\mathsf{at}\in(S\cup\mathcal{D}_{n+t-1-s})}(\gamma+\varsigma(\mathsf{at}))}$. The proofs for relations (4) and (5) are called $\vec{\pi_1}$ and $\vec{\pi_2}$, respectively, and they are given by

$$\vec{\pi}_1 = \big(H_S^{r_1}\cdot(g^\alpha)^{-r_\theta}\cdot u^{-r_2},\ H_S^{s_1}\cdot(g^\alpha)^{-s_\theta}\cdot u^{-s_2},\ H_S^{t_1}\cdot(g^\alpha)^{-t_\theta}\cdot u^{-t_2}\big)^\top$$
$$\vec{\pi}_2 = \big(g^{r_\theta},\ g^{s_\theta},\ g^{t_\theta}\big)^\top.$$

Finally, output the signature $\sigma = \big(\vec{C}_{T_1}, \vec{C}_{T_2}, \vec{C}_\theta, \vec{\pi}_1, \vec{\pi_2}\big) \in \mathbb{G}^{15}$.

▶ $\mathsf{Verify}(\mathsf{pms}, \mathsf{mpk}, \mathsf{Msg}, \sigma, \Gamma)$: it first parses Γ as a pair (t, S) and σ as $\big(\vec{C}_{T_1}, \vec{C}_{T_2}, \vec{C}_\theta, \vec{\pi}_1, \vec{\pi}_2\big)$. It computes $M = m_1 \dots m_k = H(\mathsf{Msg}, \Gamma) \in \{0,1\}^k$ and forms the corresponding vector

$$\vec{g}_{3,M} = \Big(g_{X,0} \cdot \prod_{i=1}^{k} g_{X,i}^{m_i},\ g_{Y,0} \cdot \prod_{i=1}^{k} g_{Y,i}^{m_i},\ g_{Z,0} \cdot \prod_{i=1}^{k} g_{Z,i}^{m_i}\Big)^\top \in \mathbb{G}^3.$$

Then, parse the proofs $\vec{\pi_1}$ and $\vec{\pi_2}$ as vectors $(\pi_{1,1}, \pi_{1,2}, \pi_{1,3})^\top$ and $(\pi_{2,1}, \pi_{2,2}, \pi_{2,3})^\top$, respectively. Define $H_S = h^{\alpha \cdot \prod_{\mathsf{at} \in (S \cup \mathcal{D}_{n+t-1-s})} (\gamma + \varsigma(\mathsf{at}))}$ and return 1 if and only if these relations are both satisfied:

$$E(H_S, \vec{C_{T_1}}) = E(g^\alpha, \vec{C}_\theta) \cdot E(u, \vec{C_{T_2}}) \cdot E(\pi_{1,1}, \vec{g_1}) \cdot E(\pi_{1,2}, \vec{g_2}) \cdot E(\pi_{1,3}, \vec{g}_{3,M}) \quad (6)$$

$$E(g, \vec{C}_\theta) = E\big(g, (1,1,h)\big) \cdot E(\pi_{2,1}, \vec{g_1}) \cdot E(\pi_{2,2}, \vec{g_2}) \cdot E(\pi_{2,3}, \vec{g}_{3,M}). \quad (7)$$

Correctness. The correctness follows from that of Groth-Sahai proofs.

Security Analysis. The scheme is selective-predicate and adaptive-message unforgeable assuming the hardness of both the DLIN problem and the $(\tilde{\ell}, \tilde{m}, \tilde{t})$-aMSE-CDH problem. Computational privacy can be proven based on the hardness of the DLIN problem.

Theorem 1. *The scheme is selective-predicate and adaptive-message unforgeable under chosen-message attacks assuming that (1) H is a collision-resistant hash function; (2) the DLIN assumption holds in $\mathbb{G}$; (3) the $(\tilde{\ell}, \tilde{m}, \tilde{t})$-aMSE-CDH assumption holds in $(\mathbb{G}, \mathbb{G}_T)$.* (The proof can be found in [18]).

Theorem 2. *This scheme has computational privacy, assuming that DLIN holds in $\mathbb{G}$.*

Proof. (Sketch.) The proof consists in considering two games: Game_0 and Game_1. The first game, Game_0, is the real privacy game as described in Definition 5. In particular, when executing the trusted setup algorithm $\mathsf{ABS.TSetup}$, the challenger chooses the vectors $(\vec{g_1}, \vec{g_2}, \{\vec{g_{3,i}}\}_{i=0}^k)$ such that $\vec{g_{3,i}}$ is linearly dependent with $(\vec{g_1}, \vec{g_2})$, for all $i = 0, \dots, k$. The only difference between Game_1 and Game_0 is that, in Game_1, the vector $\vec{g_{3,i}}$ is chosen at random so that it is linearly independent with $(\vec{g_1}, \vec{g_2})$, for all $i = 0, \dots, k$. Groth-Sahai [16] proved that this change is indistinguishable, under the DLIN assumption. Finally, in Game_1, the only values that could leak any information about the subset of attributes held by the signer are $\vec{C}_{T_1}, \vec{C}_{T_2}, \vec{\pi}_1$. But in the setting of Game_1, these commitments and proofs are perfectly hiding: they do not reveal any information about the committed values T_1, T_2. Therefore, privacy of the attributes holds unconditionally in Game_1. □

4 A Second Short Attribute-Based Signature Scheme for Threshold Predicates

The main advantage of our second ABS scheme over the previous one is that signatures are much shorter, since they have only three group elements. This

comes at the cost of longer secret keys sk_Ω, containing $(2n+2) \times (|\Omega| + n)$ group elements. Another advantage is that the size of the considered universe of attributes may be much larger, even exponential in the security parameter λ; we only need that all attributes in the universe $\mathcal{P}$ are encoded as different elements of $\mathbb{Z}_p^*$.

▶ $\mathsf{TSetup}(\lambda, \mathcal{P}, n)$: chooses a collision-resistant hash function $H : \{0,1\}^* \to \{0,1\}^k$, for some integer $k \in \mathsf{poly}(\lambda)$, as well as bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$ with $g \xleftarrow{R} \mathbb{G}$. It also picks $u_0, u_1, \ldots, u_k \xleftarrow{R} \mathbb{G}$ and sets $\vec{U} = (u_0, u_1, \ldots, u_k)^\top$. It finally chooses a set $\mathcal{D} = \{d_1, \ldots, d_n\}$ of n distinct elements of $\mathbb{Z}_p$ that will serve as dummy attributes.
The resulting public parameters are $\mathsf{pms} = \Big(\mathcal{P},\ n,\ \lambda,\ \mathbb{G},\ \mathbb{G}_T, g,\ \vec{U},\ \mathcal{D},\ H\Big)$.

▶ $\mathsf{MSetup}(\mathsf{pms})$: randomly chooses $\alpha, \alpha_0 \xleftarrow{R} \mathbb{Z}_p$, $\vec{\alpha} = (\alpha_1, \ldots, \alpha_N)^\top \xleftarrow{R} \mathbb{Z}_p^N$, where $N = 2n + 1$. It then computes $e(g,g)^\alpha$, $h_0 = g^{\alpha_0}$, $\vec{H} = (h_1, \ldots, h_N)^\top = g^{\vec{\alpha}}$.
The master secret key is defined to be $\mathsf{msk} = g^\alpha$ and the master public key is $\mathsf{mpk} = \Big(e(g,g)^\alpha,\ h_0,\ \vec{H}\Big)$.

▶ $\mathsf{Keygen}(\mathsf{pms}, \mathsf{mpk}, \mathsf{msk}, \Omega)$: to generate a key for the attribute set Ω, the algorithm picks a polynomial $Q_\Omega[X] = \alpha + \beta_1 X + \cdots + \beta_{n-1} X^{n-1}$ where $\beta_1, \ldots, \beta_{n-1} \xleftarrow{R} \mathbb{Z}_p$. Then, it proceeds as follows.

1. For each attribute $\omega \in \Omega$, choose a random exponent $r_\omega \xleftarrow{R} \mathbb{Z}_p$ and generate a key component $\mathsf{SK}_\omega = (D_{\omega,1}, D_{\omega,2}, K_{\omega,1}, \ldots, K_{\omega,N-1})$ where
$$D_{\omega,1} = g^{Q_\Omega(\omega)} \cdot h_0^{r_\omega}, \quad D_{\omega,2} = g^{r_\omega}, \quad \Big\{K_{\omega,i} = \big(h_1^{-\omega^i} \cdot h_{i+1}\big)^{r_\omega}\Big\}_{i=1,\ldots,N-1}. \tag{8}$$
2. For each $d \in \mathcal{D}$, choose a fresh random value $r_d \in \mathbb{Z}_p$ and generate a private key component $\mathsf{SK}_d = (D_{d,1}, D_{d,2}, K_{d,1}, \ldots, K_{d,N-1})$ as in (8):
$$D_{d,1} = g^{Q_\Omega(d)} \cdot h_0^{r_d}, \quad D_{d,2} = g^{r_d}, \quad \Big\{K_{d,i} = \big(h_1^{-w^i} \cdot h_{i+1}\big)^{r_d}\Big\}_{i=1,\ldots,N-1}. \tag{9}$$

The private key finally consists of $SK_\Omega = \big(\{\mathsf{SK}_\omega\}_{\omega\in\Omega}, \{\mathsf{SK}_d\}_{d\in\mathcal{D}}\big)$.

▶ $\mathsf{Sign}(\mathsf{pms}, \mathsf{mpk}, SK_\Omega, \mathsf{Msg}, \Gamma)$: to sign $\mathsf{Msg} \in \{0,1\}^*$ w.r.t. the policy $\Gamma = (t, S)$, where S is an attribute set of size $s = |S| \leq n$ and $t \in \{1, \ldots, s\}$, the algorithm first computes $M = H(\mathsf{Msg}, \Gamma) \in \{0,1\}^k$ and parses the private key SK_Ω as $\big(\{\mathsf{SK}_\omega\}_{\omega\in\Omega}, \{\mathsf{SK}_d\}_{d\in\mathcal{D}}\big)$.

1. It considers the subset $\mathcal{D}_{n-t} \subset \mathcal{D}$ containing the $n-t$ first attributes of $\mathcal{D}$ (chosen in some pre-specified lexicographical order). It also chooses an arbitrary subset $S_t \subset \Omega \cap S$ such that $|S_t| = t$ and defines $\vec{Y} = (y_1, \ldots, y_N)^\top$ as the vector containing the coefficients of the polynomial
$$P_S(Z) = \sum_{i=1}^{n-t+s+1} y_i Z^{i-1} = \prod_{\omega\in S}(Z - \omega) \cdot \prod_{d\in\mathcal{D}_{n-t}} (Z - d). \tag{10}$$

Since $n - t + s + 1 \leq 2n + 1 = N$, the coordinates $y_{n-t+s+2}, \ldots, y_N$ are all set to 0.

2. For each $\omega \in S_t$, use $\mathsf{SK}_\omega = (D_{\omega,1}, D_{\omega,2}, \{K_{\omega,i}\}_{i=1}^{N-1})$ to compute

$$D'_{\omega,1} = D_{\omega,1} \cdot \prod_{i=1}^{N-1} K_{\omega,i}^{y_{i+1}} = g^{Q_\Omega(\omega)} \cdot \big(h_0 \cdot \prod_{i=1}^{N} h_i^{y_i}\big)^{r_\omega}. \tag{11}$$

The last equality comes from the fact that $P_S(\omega) = 0$ for all $\omega \in S$.

3. Likewise, for each dummy attribute $d \in \mathcal{D}_{n-t}$, use $\mathsf{SK}_d = (D_{d,1}, D_{d,2}, \{K_{d,i}\}_{i=1}^{N-1})$ to compute

$$D'_{d,1} = D_{d,1} \cdot \prod_{i=1}^{N-1} K_{d,i}^{y_{i+1}} = g^{Q_\Omega(d)} \cdot \big(h_0 \cdot \prod_{i=1}^{N} h_i^{y_i}\big)^{r_d}. \tag{12}$$

4. Using $\{D'_{\omega,1}\}_{\omega \in S_t}$ and $\{D'_{d,1}\}_{d \in \mathcal{D}_{n-t}}$ and the corresponding $D_{\omega,2} = g^{r_w}$, $D_{d,2} = g^{r_d}$, compute

$$D_1 = \prod_{\omega \in S_t} {D'_{\omega,1}}^{\Delta_\omega^{S_t \cup \mathcal{D}_{n-t}}(0)} \cdot \prod_{d \in \mathcal{D}_{n-t}} {D'_{d,1}}^{\Delta_d^{S_t \cup \mathcal{D}_{n-t}}(0)} = g^\alpha \cdot (h_0 \cdot \prod_{i=1}^{N} h_i^{y_i})^r, \tag{13}$$

$$D_2 = \prod_{\omega \in S_t} {D_{\omega,2}}^{\Delta_\omega^{S_t \cup \mathcal{D}_{n-t}}(0)} \cdot \prod_{d \in \mathcal{D}_{n-t}} {D_{d,2}}^{\Delta_d^{S_t \cup \mathcal{D}_{n-t}}(0)} = g^r, \tag{14}$$

where $r = \sum_{\omega \in S_t} \Delta_\omega^{S_t \cup \mathcal{D}_{n-t}}(0) \cdot r_\omega + \sum_{d \in \mathcal{D}_{n-t}} \Delta_d^{S_t \cup \mathcal{D}_{n-t}}(0) \cdot r_d$.

5. Parse $M \in \{0,1\}^k$ as a string $m_1 \ldots m_k$ where $m_j \in \{0,1\}$ for $j = 1, \ldots, k$. Then, choose $z, w \xleftarrow{R} \mathbb{Z}_p$ and compute

$$\sigma_1 = D_1 \cdot \big(h_0 \cdot \prod_{i=1}^{N} h_i^{y_i}\big)^w \cdot \big(u_0 \cdot \prod_{j=1}^{k} u_j^{m_j}\big)^z, \qquad \sigma_2 = D_2 \cdot g^w, \qquad \sigma_3 = g^z.$$

Return the signature $\sigma = (\sigma_1, \sigma_2, \sigma_3) \in \mathbb{G}^3$.

▶ $\mathsf{Verify}(\mathsf{pms}, \mathsf{mpk}, \mathsf{Msg}, \sigma, \Gamma)$: it parses Γ as a pair (t, S). It computes $M = H(\mathsf{Msg}, \Gamma) \in \{0,1\}^k$ and considers the subset $\mathcal{D}_{n-t} \subset \mathcal{D}$ containing the $n - t$ first dummy attributes of $\mathcal{D}$. Then, it defines the vector $\vec{Y} = (y_1, \ldots, y_N)^\top$ from the polynomial $P_S(Z)$ as per (10). The algorithm accepts the signature $\sigma = (\sigma_1, \sigma_2, \sigma_3)$ as valid and thus outputs 1 if and only if

$$e(g,g)^\alpha = e(\sigma_1, g) \cdot e\big(\sigma_2, h_0 \cdot \prod_{i=1}^{N} h_i^{y_i}\big)^{-1} \cdot e\big(\sigma_3, u_0 \cdot \prod_{j=1}^{k} u_j^{m_j}\big)^{-1}. \tag{15}$$

Correctness. The correctness of the scheme follows from the property that for each attribute $\omega \in S_t \subset S \cap \Omega$, the vector $\vec{X}_\omega^N = (1, \omega, \omega^2, \ldots, \omega^{N-1})$ is orthogonal to $\vec{Y}$, so that we have

$$D'_{\omega,1} = g^{Q_\Omega(\omega)} \cdot \Big(h_0 \cdot h_1^{-(\langle \vec{X}_\omega^N, \vec{Y} \rangle - y_1)} \prod_{i=2}^{N} h_i^{y_i}\Big)^{r_\omega} = g^{Q_\Omega(\omega)} \cdot \Big(h_0 \cdot \prod_{i=1}^{N} h_i^{y_i}\Big)^{r_\omega},$$

which explains the second equality of (11) and the same holds for (12). In addition, the values (D_1, D_2) obtained as per (13)-(14) satisfy $e(D_1, g) = e(g, g)^{\alpha} \cdot e(h_0 \cdot \prod_{i=1}^{N} h_i^{y_i}, D_2)$, which easily leads to the verification equation (15).

Security Analysis. This second scheme is selective-predicate and adaptive-message unforgeable by reduction to the hardness of the n-Diffie-Hellman Exponent (n-DHE) problem ([6]). This scheme also enjoys unconditional privacy, which is another advantage over our first scheme.

Theorem 3. *The scheme is selective-predicate and adaptive-message unforgeable under chosen-message attacks if H is collision-resistant and if the $(2n+1)$-DHE assumption holds in $\mathbb{G}$, where n is the maximal number of attributes in the set S.* (The proof can be found in [18].)

Theorem 4. *This second ABS scheme enjoys perfect privacy.*

Proof. A valid signature for the threshold policy (t, S) which was produced using the subset of attributes $S_t \subset S$, $|S_t| = t$ and with randomness w can also be produced for any other set $S'_t \subset S$, $|S'_t| = t$ with randomness w'. More specifically, if $r = \sum_{\omega \in S_t} \Delta_{\omega}^{S_t \cup \mathcal{D}_{n-t}}(0) \cdot r_{\omega} + \sum_{d \in \mathcal{D}_{n-t}} \Delta_{d}^{S_t \cup \mathcal{D}_{n-t}}(0) \cdot r_d$ and $r' = \sum_{\omega \in S'_t} \Delta_{\omega}^{S_t \cup \mathcal{D}_{n-t}}(0) \cdot r_{\omega} + \sum_{d \in \mathcal{D}_{n-t}} \Delta_{d}^{S_t \cup \mathcal{D}_{n-t}}(0) \cdot r_d$, any pair (w, w') satisfying $r + w = r' + w'$ will result in the same signature for S_t and S'_t. □

5 More General Signing Predicates

Our schemes admit some extensions to deal with more general monotone predicates. In general, a predicate is a pair (S, Γ), where $S = \{\mathsf{at}_1, \ldots, \mathsf{at}_s\}$ is a set of attributes and $\Gamma \subset 2^S$ is a monotone increasing family of subsets of S. An attribute-based signature for a pair (S, Γ) convinces the verifier that the signer holds some subset of attributes $A \in \Gamma$, without revealing any information on A.

5.1 Extensions for the First Scheme

Similarly to what is suggested in [12], our first signature scheme can be extended to admit weighted threshold predicates, that is, pairs (S, Γ) for which there exists a threshold t and an assignment of weights $\omega : S \to \mathbb{Z}^+$ such that $\Omega \in \Gamma \iff \sum_{\mathsf{at} \in \Omega} \omega(\mathsf{at}) \geq t$.

Furthermore, since the final form of the signatures in our first threshold scheme is that of a Groth-Sahai non-interactive proof, one could consider signing predicates which are described by a monotone formula (OR / AND gates) over threshold clauses. The Groth-Sahai proof would be then a proof of knowledge of some values that satisfy a monotone formula of equations. The size of such a proof (and therefore, the size of the resulting attribute-based signatures) would be linear in the number of threshold clauses in the formula. We stress that this is still better than having size linear in the number of involved attributes, as in all previous constructions.

5.2 Extensions for the Second Scheme

The idea of our second scheme is that a (threshold) attribute-based signature can be computed only if the signer holds t attributes in S which, combined with $n - t$ dummy attributes, lead to n attributes at such that $P_S(\mathsf{at}) = 0$. This makes it possible to interpolate a polynomial $Q_\Omega(X)$ with degree $n-1$, recover in some way the value g^α and produce a valid signature. To admit any possible value of the threshold t in $\{1, \ldots, n\}$, the number of dummy attributes must be n. We can use similar ideas for other families of predicates which are realized with a secret sharing scheme with properties which resemble those of Shamir's. The ideas underlying this extension are quite related to those in [11], where dummy attributes were used to design attribute-based encryption schemes for general decryption predicates. An illustrative example, considering hierarchical threshold predicates, is given in the full version of this paper [18].

Disclaimer and acknowledgments

This work was started while the second and third authors visited Universitat Politècnica de Catalunya. J. Herranz is supported by a *Ramón y Cajal* grant, partially funded by the European Social Fund (ESF) of the Spanish MICINN Ministry, F. Laguillaumie by the French ANR-07-TCOM-013-04 PACE Project and B. Libert by the F.RS.-F.N.RS. through a "Chargé de recherches" fellowship and by the BCRYPT Interuniversity Attraction Pole. Both J. Herranz and C. Ràfols are partially supported by the Spanish MICINN Ministry under project MTM2009-07694 and ARES – CONSOLIDER INGENIO 2010 CSD2007-00004. C. Ràfols is with the UNESCO Chair in Data Privacy, but the views expressed in this paper are her own and do not commit UNESCO.

References

1. N. Attrapadung, B. Libert, E. de Panafieu. Expressive key-policy attribute-based encryption with constant-size ciphertexts. In *PKC'11*, 90–108, 2011.
2. J. Bethencourt, A. Sahai, B. Waters. Ciphertext-policy attribute-based encryption. In *IEEE S&P'07*, IEEE Society Press, 321-334, 2007.
3. D. Boneh, X. Boyen. Efficient selective-ID secure identity-based encryption without random oracles. In *Eurocrypt'04*, 223–238, 2004.
4. D. Boneh, X. Boyen, E.-J. Goh. Hierarchical identity-based encryption with constant size ciphertext. In *Eurocrypt'05*, 440–456, 2005.
5. D. Boneh, X. Boyen, H. Shacham. Short group signatures. In *Crypto'04*, 41–55, 2004.
6. D. Boneh, C. Gentry, B. Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *Crypto'05*, 258–275, 2005.
7. D. Boneh, M. Hamburg. Generalized identity-based and broadcast encryption schemes. In *Asiacrypt'08*, 455–470, 2008.
8. X. Boyen. Mesh signatures. In *Eurocrypt'07*, 210–227, 2007.
9. E. Bresson, J. Stern, M. Szydlo. Threshold ring signatures and applications to ad-hoc groups. In *Crypto'02*, 465–480, 2002.

10. M. Chase, A. Lysyanskaya. On signatures of knowledge. In *Crypto'06*, 78–96, 2006.
11. V. Daza, J. Herranz, P. Morillo, C. Ràfols. Extended access structures and their cryptographic applications. In *Applicable Algebra in Engineering, Communication and Computing*, Volume 21, Issue 4, 257–284, 2010.
12. C. Delerablée, D. Pointcheval. Dynamic threshold public-key encryption. In *Crypto'08*, 317–334, 2008.
13. K. Emura, A. Miyaji, A. Nomura, K. Omote, M. Soshi. A ciphertext-policy attribute-based encryption scheme with constant ciphertext length. In *ISPEC '09*, 13–23, 2009.
14. A. Escala, J. Herranz, P. Morillo. Revocable attribute-based signatures with adaptive security in the standard model. In *Africacrypt'11*, 224–241, 2011.
15. V. Goyal, O. Pandey, A. Sahai, B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM CCS'06*, ACM Press, 89–98, 2006.
16. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Eurocrypt'08*, 415–432, 2008.
17. J. Herranz, F. Laguillaumie, C. Ràfols. Constant-size ciphertexts in threshold attribute-based encryption. In *PKC'10*, 19–34, 2010.
18. J. Herranz, B. Libert, F. Laguillaumie, C. Ràfols. Short attribute-based signatures for threshold predicates. Preprint available at: `http://hal.archives-ouvertes.fr/hal-00611651/fr/` , 2011.
19. D. Hofheinz, E. Kiltz. Programmable hash functions and their applications. In *Crypto'08*, 21–38, 2008.
20. A. Lewko, T. Okamoto, A. Sahai, K. Takashima, B. Waters. Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption. In *Eurocrypt'10*, 62–91, 2010.
21. J. Li, M.H. Au, W. Susilo, D. Xie, K. Ren. Attribute-based signature and its applications. In *ASIACCS'10*, ACM Press, 60–69, 2010.
22. J. Li, K. Kim. Hidden attribute-based signatures without anonymity revocation. In *Information Sciences*, 180 (9), 1681–1689, 2010.
23. T. Malkin, I. Teranishi, Y. Vahlis, M. Yung. Signatures resilient to continual leakage on memory and computation. In *TCC'11*, 89–106, 2011.
24. H.K. Maji, M. Prabhakaran, M. Rosulek. Attribute-based signatures. In *CT-RSA'11*, 376–392, 2011.
25. M. Naor. On cryptographic assumptions and challenges. In *Crypto'03*, 96–109, 2003.
26. T. Okamoto, K. Takashima. Homomorphic encryption and signatures from vector decomposition. In *Pairing'08*, 57–74, 2008.
27. T. Okamoto, K. Takashima. Efficient attribute-based signatures for non-monotone predicates in the standard model. In *PKC'11*, 35–52, 2011.
28. R. L. Rivest, A. Shamir, Y. Tauman. How to leak a secret. In *Asiacrypt'01*, 552-565, 2001.
29. A. Sahai, B. Waters. Fuzzy identity-based encryption In *Eurocrypt'05*, 457–473, 2005.
30. S.F. Shahandashti, R. Safavi-Naini. Threshold attribute-based signatures and their application to anonymous credential systems. In *Africacrypt'09*, 198–216, 2009.
31. T. Tassa. Hierarchical threshold secret sharing. In *Journal of Cryptology*, 20 (2), 237–264, 2007.
32. B. Waters. Efficient identity-based encryption without random oracles. In *Eurocrypt'05*, 114–127, 2005.