Attribute-based pseudonymity for privacy-preserving authentication in cloud services

Victor Sucasas, Member IEEE, Georgios Mantas, Member IEEE,
Maria Papaioannou, Jonathan Rodriguez, Senior Member IEEE

***Abstract*—Attribute-based authentication is considered a cornerstone component to achieve scalable fine-grained access control in the fast growing market of cloud-based services. Unfortunately, it also poses a privacy concern. User's attributes should not be linked to the users' identity and spread across different organizations. To tackle this issue, several solutions have been proposed such as Privacy Attribute-based Credentials (Privacy-ABCs), which support pseudonym-based authentication with embedded attributes. Privacy-ABCs allow users to establish anonymous accounts with service providers while hiding the identity of the user under a pseudonym. However, Privacy-ABCs require the selective disclosure of the attribute values towards service providers. Other schemes such as Attribute-based Signatures (ABS) and mesh signatures do not require the disclosure of attributes; unfortunately, these schemes do not cater for pseudonym generation in their construction, and hence cannot be used to establish anonymous accounts. In this paper, we propose a pseudonym-based signature scheme that enables unlinkable pseudonym self-generation with embedded attributes, similarly to Privacy-ABCs, and integrates a secret sharing scheme in a similar fashion to ABS and mesh signature schemes for attribute verification. Our proposed scheme also provides verifiable delegation, enabling users to share attributes according to the service providers' policies.**[1]

I. INTRODUCTION

Conventionally, user authentication and authorization have been based on identifying the user and establishing long-lasting access rights to specific objects (e.g. data files, online accounts, etc.) or actions (e.g. modifying data or obtaining a specific service). The assignment of static relationships between users and service providers involves defining roles and linking users to one or more roles. This conventional approach however is not scalable in scenarios with a large number of users, or in systems where fine-grained access control policies must be applied. In this framework, Attribute-based Access Control (ABAC) has been suggested as a more flexible solution. ABAC provides a logical model where users are assigned a large attribute set, and the access to objects or services is controlled through the evaluation of a set of complex rules against the user's attribute set. Although ABAC systems were initially suggested for specific environments, e.g enterprise deployment NIST SP 800-162 [1], a significant number research works have adopted attribute-based authentication in diverse scenarios [2],[3],[4],[5].

This approach is promising for scenarios with massive user and object deployment such as cloud-based services [2][6], massive IoT (MIoT) [7][8], or eHealth [9]. However, in such large scenarios the application of fine-grained user authentication through attributes poses a threat in terms of privacy. Users reveal not only their identity, but also a large attribute set that can be used to trace and profile them. This issue has been addressed by some works that propose privacy-preserving authentication systems that embed attributes [10],[11],[12],[13][14]. These systems, mainly based on privacy-preserving attribute based credentials (Privacy-ABCs), enable users to generate pseudonyms with embedded attributes and authenticate towards a service provider while hiding the users' identity and limiting the information disclosed about the attribute set. There are efficient implementations, such as Idemix (IBM) and U-Prove (Microsoft), providing a rich set of functionalities. However, Privacy-ABCs require the selective disclosure of users attributes, which can narrow down the user identification process and jeopardize privacy. As an example, a small set of information consisting of age, gender and postal code (area of residence) is sufficient to fully identify 87% of U.S. citizens [15]. This issue has already been pointed out by other works such as [16],[17], that propose pseudonym-based authentication with attributes that limits the information revealed about the attribute set.

Ideally, attribute-based authentication should not disclose any information about the users' attribute set beyond the fact that a user qualifies for accessing a specific service. In this sense, attribute based signature (ABS) schemes provide such feature [18],[19],[20],[21]. With ABS schemes, attributes can be evaluated with respect to complex access structures without revealing the attributes. Unfortunately, ABS schemes do not cater for efficient pseudonym construction linked to the user's attributes. Note that pseudonymization is required to keep active sessions and provide bidirectional interactions between users and service providers. Pseudonymity enables anonymous accounts [22],[23]. Another approach worth mentioning is mesh signatures [24],[25], which follow a similar construction to ABS schemes. However, mesh signatures are designed to allow collusion between users, i.e. users can construct signatures with a collective attribute set which may trigger an undesirable privilege escalation in some scenarios.

In this paper we provide a signature scheme with attribute-based pseudonymity that, although not as modular as Privacy-ABCs, allows users to generate an unlimited number of pseudonyms with embedded attributes while keeping the attribute set private. Service providers can attest the compliancy

[1]V. Sucasas, G. Mantas and M. Papaioanu are with the Instituto de Telecomunicações, Aveiro, Portugal (email: vsucasas@av.it.pt, gimantas@av.it.pt, m.papaioannou@av.it.pt).

V. Sucasas is also with the Technology Innovation Institute, Abu Dhabi, UAE (email: victor.sucasas@tii.ae)

G. Mantas is also with the Faculty of Engineering and Science, University of Greenwich, UK (email: G.Mantas@greenwich.ac.uk)).

J. Rodriguez is with the Faculty of Computing, Engineering and Science, University of South Wales, (email: jonathan.rodriguez@southwales.ac.uk).

of the embedded attributes with respect to an access tree structure without disclosing the exact attributes embedded in the pseudonyms. The proposed scheme also enables conditional delegation, i.e. similar to mesh signatures users can share specific attributes to construct signatures with a wider attribute set. The problem of privilege escalation is avoided by allowing the service provider to limit the users' delegation capabilities, i.e. selecting the attributes that can be shared between users. Hence, the proposed pseudonym-based signature scheme mixes the features of: i) Privacy-ABCs, since it provides unlimited self-generation of unlinkable pseudonyms; ii) ABS schemes, since it only reveals the compliancy of the attribute set with respect to an expressive access policy; iii) and mesh signatures, since it allows attribute delegation between users, but limited by service providers. Previous approaches do not provide these features simultaneously.

The paper is structured as follows: i) section II describes the related work, and clarifies the innovation of the proposed scheme; ii) section III presents the system model; iii) section IV provides the preliminaries for understanding the scheme construction; iv) sections V and VI detail the construction of the signature scheme and the secret sharing scheme required in the signature construction; v) section VII provides the analysis of the correctness of the signature scheme; vi) VIII details the security analysis; vii) section IX shows the implementation details; viii) section X provides exemplary use-cases including a performance evaluation of those use-cases; ix) section XI provides extensive performance evaluation; x) section XII provides a detailed comparison between the proposed scheme and previous works; and xi) section XIII concludes this paper.

II. RELATED WORK

The notion of identity confidentiality was described by Pfitzmann and Hansen [26] in their definition of anonymity. In such definition, an entity belonging to a set of entities is considered to be in an anonymous state if it is not identifiable within that set, i.e. there is no technique, better than a random guess, to evaluate the identity of the entity within the set. Note that, according to such definition, subjects within the anonymous set can interact freely with other entities, and the only information that the other interacting party will get is that the subject belongs to the set. However, in such a system the lack of users' identifiers makes its application challenging in service oriented architectures, where user accounts and sessions must be kept in the short, medium or even long term.

Pseudonym systems enable users/entities to keep sessions active with service providers while still preserving an anonymous state. Entities can hold many pseudonyms representing the entity, its roles or functions, or the different relationships of the entity with different organizations [27][28]. Ideally, different pseudonyms of the same entity are not linkable between each other and do not leak any information about the entity's real identity. Hence, an entity hiding behind a pseudonym does not reveal any other information than the fact that it belongs to the group of entities that are entitled to use those pseudonyms. Pseudonym systems can also provide conditional privacy since they can enable the revocation of the privacy rights of misusers and permit a trusted authority to retrieve their real identities [29][30]. However, pseudonyms can be implemented in multiple manners and provide a diverse set of features. Several technologies have been proposed, such as anonymous credential systems [31],[32],[33],[34], group signatures [35],[31],[36], Public Key certificates [37],[38], or identity based cryptography (IBC) [39]. However, only some works have provided efficient constructions to embed attributes in the pseudonyms.

Privacy-ABCs such as Idemix [11] or U-Prove [12] are specific implementations of anonymous credential systems that enable users to authenticate privately. These systems integrate attributes in the form of tuples $(attribute_x, value)$. Idemix and U-Prove allow users to obtain attributes from multiple issuing authorities while preventing authorities from learning the users' attribute set, or linking the issuance process between the user and the issuer authority to the verification process between the user and the service provider. Privacy-ABCs provide constructions to enable pseudonym generation and credential revocation (allows to revoke the credential validity and inspect the users' identity in case of misbehaviour). The main limitation of these schemes is that the attribute verification process involves either the explicit disclosure of some attribute values (the ones required by the service provider) or the verification of predicates about the attribute values limited to equality and inequality [40],[41], e.g. it is possible to verify privately if two attributes hold the same value or whether such value belongs to a range of values.

ABS schemes [18],[19] may not provide the same level of anonymity against issuing authorities as Privacy-ABCs, since colluding authorities can obtain users' attribute set. Nevertheless, the attribute set is the only information leaked to issuing authorities, and it is never disclosed to verifiers (i.e. service providers). Also, schemes like the ones proposed in [20],[21] can prevent collusion if only one authority is honest. In ABS schemes, attributes are embedded into the signing keys in the form of descriptive elements, e.g. $attribute_x$. Hence, attributes only denote that the user holds that specific attribute but without a specific value associated to it. The main advantage of ABS schemes is that it enables users to prove that the embedded attribute set satisfies a predefined boolean function in the form of an access structure. This is normally implemented as a span program or an access tree [42]. No further information is disclosed about the attribute set. Mesh signature schemes [24] follow a similar approach since users can prove the possession of attribute sets with respect to a specific access tree structure[2]. However, mesh signatures enable collusion between users, i.e. a set of users can jointly sign a message using a collective set of attributes, which may be an undesired feature in some scenarios and should be limited by service providers.

In terms of privacy, ABS schemes and mesh signatures achieve higher guarantees than Privacy-ABCs, since the attributes are not disclosed. But the limitation is that ABS schemes and mesh signatures do not support pseudonymity. It is worth mentioning that an ABS scheme can be used to establish an anonymous communication by letting the user sign a session key and encrypt it with the public key of the service provider. The service provider, after verifying that the set of attributes in the signature complies with the predefined access policy, uses the session key to establish a secure channel with

[2]any access tree structure can actually be converted into a monotone span program with Lewko-Waters algorithm [43],[44]

the user. However, the user's session ends when the secure channel terminates. Also, the users could generate several anonymous sessions with the same provider, which may trigger sybil attacks [45]. Pseudonyms provide a stronger notion of private sessions since they allow users to maintain the private sessions and create accounts by reusing the same pseudonym. Moreover, organizations can limit the number of pseudonyms that a specific user can generate to access a specific service, hence reducing the attack surface for sybil attacks.

In this paper we propose a pseudonym-based signature scheme with embedded attributes that, similarly to ABS, can be verified according to a predefined access tree structure without disclosing the attributes. Also, similarly to Privacy-ABCs users can generate an unlimited number of pseudonyms with embedded attributes and sign messages with those pseudonyms. The proposed scheme provides attribute delegation, and enables the verifier (the service provider) to choose which attributes are shareable between users. The proposed scheme also enables anonymity revocation in case of misuse, and caters for an efficient implementation as shown in sec. IX.

III. System model

The proposed system considers the following entities, also detailed in Fig. 1, which perform the following functionalities:

1) **The Certification Authority (CA)**: is in charge of i) generating the public values and verification keys ii) issuing credentials to users; iii) revoking misusers' privacy rights, i.e. it can retrieve the real identity of a user given her pseudonym;
2) **The Verifier**: is an Authentication Server (AS) in the service provider side. It is responsible for: i) verifying signatures sent by users according to a predefined access tree structure.
3) **The users**: are entities provided with a valid credential issued by the CA with some embedded attributes, and they can: i) self-generate pseudonyms; and ii) sign messages with those pseudonyms according to a predefined access tree structure over the universe of attributes; iii) share attributes with other users.

The verifier is considered honest-but-curious, hence it tries to identify and trace users. The CA is considered honest, and it is assumed to be able to link pseudonyms to users (by means of using the revoke algorithm) and knows the attribute set of all users (which is provided in the CreGen algorithm). Finally, the users are considered dishonest, and may try to collude with other users and forge signatures with attributes that were not issued by the CA.

IV. Preliminaries

This section provides the mathematical background for the understanding of the proposed pseudonym-based signature scheme.

1) Bilinear Maps: Let G_1 and G_T be two cyclic groups of prime order p, where the discrete logarithm problem is hard. Let κ be a security parameter that defines the number of bits of p. Then e is a bilinear map [46], in the groups (G_1, G_T), $e : G_1 \times G_1 \rightarrow G_T$[3], if it satisfies:

- Bilinearity: $\forall \alpha, \beta \in Z_p^*$ and $P, Q, R \in G_1$, it holds that $e(\alpha P + \beta Q, R) = e(P, R)^{\alpha} e(Q, R)^{\beta}$ and $e(R, \alpha P + \beta Q) = e(R, P)^{\alpha} e(R, Q)^{\beta}$.
- No-degeneracy: There is at least one element $Q \in G_1$ such that $e(Q, Q) \neq 1_{G_T}$.
- Complexity: It is possible to compute efficiently the bilinear map e.

2) ECDLP: Let $G = \langle P \rangle$ be a cyclic group of prime order p. Given a point $Q \in G$, then the Elliptic Curve Discrete Logarithm Problem (ECDLP) states that it is computationally intractable, in polynomial time, to obtain an integer $n \in [1, p-1]$ such that $Q = nP$.

A. k-CAA Problem

Let G_1, G_T be two cyclic groups of prime order and e be a bilinear map defined as above, and let $x, a_1, \ldots, a_k \in Z_p$ (where k is an integer). The k-CAA problem, collusion attack with k traitors [47], is defined as: given the value $W = xP$ and the set $\left(Sa_1 = \frac{1}{x+a_1}P, \ldots, Sa_k = \frac{1}{x+a_k}P\right)$ compute a value $Sa_u = \frac{1}{x+a_u}P$ different from the previous set of values such that $e(P, P) = e(aP + W, Sa_u)$. It is proven [47] that the k-CAA is only solvable if the k-weak Diffie-Hellman Algorithm (k-wDHA) exists. The k-wDHA is an algorithm that is able to compute $\frac{1}{x}P$ from k+1 values of the form $P, xP, x^2P, \ldots, x^kP$.

B. Monotone access structures

Given a set $P = \{P_1, \ldots, P_n\}$, a collection of subsets of P, i.e. $\mathbb{A} \subseteq 2^{\{P_1, \ldots, P_n\}}$, is monotone if $\forall$ B,C: if $C \in \mathbb{A}$ and $C \subseteq B$ then $B \in \mathbb{A}$. In other words, any set in the collection extended with additional elements belongs as well to the collection. A monotone access structure Γ is a monotone collection of non-empty sets $\mathbb{A} \backslash \{0\}$. The sets in Γ are called qualified sets. Note that, since the access structure is monotone, it is possible to define subset of sets in $\mathbb{A}$ such that if one only element is removed from a set, the resulting set is non-qualified, these sets are called minimal qualified sets. An access structure can be defined by the set of minimal qualified sets.

1) Dual access structures: Given an access structure Γ containing subsets of a set $P = \{P_1, \ldots, P_n\}$, the dual access structure Γ^* contains all the subsets A such that $A^c \notin \Gamma$, where A^c is the complement of A in P, i.e. $A^c = \{P_i | P_i \in P \text{ and } P_i \notin A\}$. A dual access structure of a monotone access structure is also monotone and satisfies that $(\Gamma^*)^* = \Gamma$, hence it can also be defined by minimal qualified sets. The interesting property of dual access structures is that a set A is qualified in Γ if and only if it has non empty intersection with all qualified sets in Γ^* [48].

[3]we provide the definition of a symmetric pairing since our implementation uses a symmetric pairing configuration.

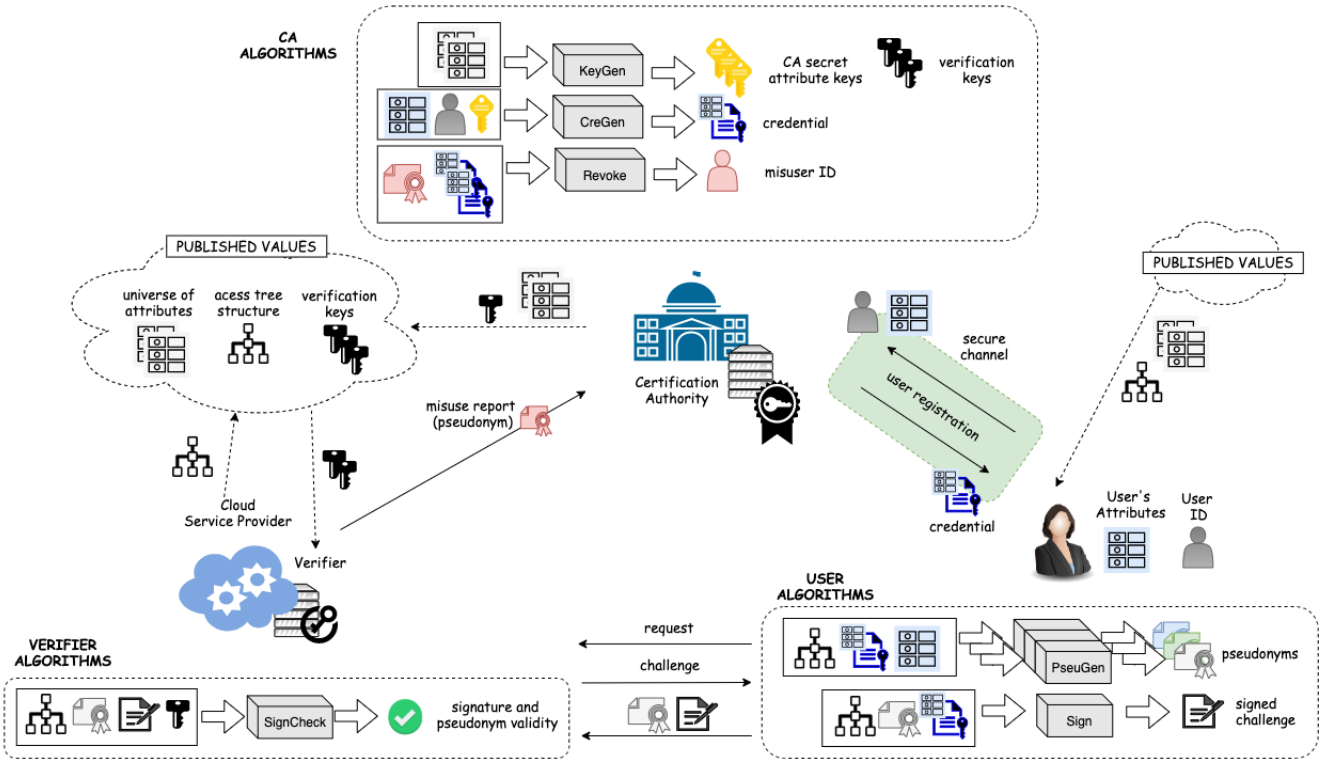

Fig. 1. System model for the proposed signature scheme.

C. Secret sharing

A secret sharing scheme § is a method to distribute shares of a secret value s among a set of n participants $P = \{P_1, \ldots, P_n\}$ such that only the qualified subsets can reconstruct the secret. The collection of qualified subsets conforms to the *access structure* of the secret sharing scheme [49],[50],[51]. Note that, for secret sharing schemes the access structures must be monotone, since a qualified subset of parties cannot become non-qualified by simply adding extra secret shares, since these new shares can be discarded. Secret sharing schemes are said to be perfect if no information about the secret is obtained by a non-qualified set of parties, regardless the computational power of the parties. If the length of the shares is equal to the length of the secret then the scheme is also ideal [52].

In the proposed signature scheme we require a perfect secret sharing scheme, and we require the scheme to be ideal for a more efficient implementation. We also require the secret sharing scheme to be semi-smooth, i.e. given a security parameter κ it should satisfy the following properties [53]:

1) the length of the shares are polynomial in κ
2) the reconstruction of the secret from a set of qualified shares is time polynomial in κ
3) testing consistency of a full set of shares, i.e. checking that all qualified sets can reconstruct the secret, can be done in polynomial time in κ
4) any non-qualified subset of shares can be extended to a consistent full set of shares in polynomial time in κ

1) Shamir secret-sharing scheme: Shamir scheme $\S_{k,n}$ is an example of a perfect, smooth and ideal secret sharing scheme. Let s be a secret value in a finite field Z_p, and $N = \{n_1, \ldots, n_n\}$ be a set of participants to be given a secret share of s also in Z_p, any subset of k participants (k being a threshold) can combine their shares to reconstruct the secret value. Shamir secret sharing scheme is a threshold secret sharing scheme, since the only condition for a subset to be a qualified set is having at least k members out of n. To construct the scheme, first, define a polynomial $f(x) = a_0 + a_1 x + \cdots + a_{k-1}x^{k-1} \in Z_p$ where $a_0 = s$ and the rest of coefficients $\{a_1, \ldots, a_{k-1}\}$ are chosen randomly in Z_p. Assign a non-zero point in the polynomial to each participant as its secret share $n_i = (x_i, f(x_i))$. Note that, given any subset of k secret shares $\{(x_1, f(x_1)), \ldots, (x_k, f(x_k))\}$, using polynomial interpolation it is possible to retrieve the polynomial coefficients, thus obtaining the secret a_0:

$$f(x) = \sum_{i=1}^{k} \Delta_{x_i}(x) f(x_i) \tag{1}$$

where $\Delta_{x_i}(x)$ is the Lagrange coefficient for x_i in $\{x_1, \ldots, x_k\}$.

V. Signature Scheme Construction

The proposed signature scheme is composed of algorithms **KeyGen**, **CreGen**, **Revoke**, **PseuGen**, **Sign** and **SignCheck**, executed by the CA, user and verifier as described in Fig. 1. Table I shows the values derived from each algorithm, secret values are coloured in blue.

KeyGen(1^κ) The algorithm runs on the CA side. For a given security parameter κ, the algorithm selects a prime number p of κ bits. Then selects two cyclic groups of order p, G_1

TABLE I. NOTATION

Element	Derived from	Element	Derived from
generator (P)	KeyGen	**CA keys** (s_i)	KeyGen
generator (g)	KeyGen	**credential** (Sa_i)	CreGen
generator (h)	KeyGen	**credential** (μ)	CreGen
attribute keys (W_i)	KeyGen	**pseu key** (μ')	PseuGen
pseu index (z)	PseuGen	signature (s_3)	Sign
pseudonym (Pu)	PseuGen	signature (s_5)	Sign
pseudonym (Pa_i)	PseuGen	signature ($s_{2,i}$)	Sign
auxiliary key ($\tilde{y}_1$)	Sign	signature ($s_{4,i}$)	Sign
auxiliary key ($\tilde{y}_2$)	Sign	signature (s_1)	Sign
signature (c)	Sign	signature (c_i)	Sign

and G_T, such that it exists a bilinear map $e : G_1 \times G_1 \rightarrow G_T$, and such that the ECDLP is hard in G_1 and the Discrete Logarithm problem (DLP) is hard in G_T. Also, it picks a cryptographic hash function $H() : \{0,1\}^* \rightarrow Z_p^*$. The algorithm also publishes a set of attributes $A = \{att_1, \ldots, att_n\}$, and performs the following operations to generate the public parameters $PP = \{G_1, G_T, Z_p, P, g, h, e(), H()\}$, one public key per attribute $\{W_1, \ldots, W_n\}$, and one master secret key per attribute $\{s_1, \ldots, s_n\}$:

1) selects a public generator $P \xleftarrow{R} G_1$
2) selects public generators h and $g \xleftarrow{R} G_T$
3) selects a secret value per attribute $s_i \xleftarrow{R} Z_p^* \ \forall i \in \{1, \ldots, n\}$
4) computs a public key per attribute $W_i = s_i P \ \forall i \in \{1, \ldots, n\}$

CreGen(s_i,PP,ID,S) The algorithm runs on the CA side. It uses the secret values s_i for $i \in \{1, \ldots, n\}$ and public parameters PP to generate a credential for a user with identity ID. The user requests a credential for the set $S \subseteq A$ of k attributes $S = \{att_{i_1}, \ldots, att_{i_k}\}$ where the set of indices $I_S = \{i_1, \ldots, i_k\}$ is a subset of the indices of A, i.e. $I_A = \{1, \ldots, n\}$. The CA runs this algorithm and sends the credential $cred = (\mu, Sa)$ where $Sa = \{Sa_{i_1}, \ldots, Sa_{i_k}\}$ to the user, over a secure channel. The user must authenticate first and provide its real identity ID[4] and the set of attributes S so the CA can verify that the user's claim on the attribute set is legitimate, and eventually deny the issuing of the credential if the user is not entitled to get a credential with the requested attribute set. Although the value ID is not used in the credential generation, the CA stores the tuple $(ID, cred, S)$ in an internal registry REG and performs the following operations:

1) randomly selects a secret value $\mu \xleftarrow{R} Z_p^*$
2) computes secret $Sa_i = P\frac{1}{(s_i+\mu)}; \forall\ i \in \{i_1, \ldots, i_k\}$
3) the credential is the tuple $cred = (\mu, Sa)$ where $Sa = \{Sa_{i_1}, \ldots, Sa_{i_k}\}$

The CA sends the credential to the user over a secure channel. The user can verify the correctness of the credential by checking wether $e(\mu P + W_i, Sa_i) = e(P, P)\ \forall\ i \in \{i_1, \ldots, i_k\}$ holds.

PseuGen($cred$,PP,S'): The algorithm runs on the user side and requires a valid credential for an attribute set S, and can optionally include delegated attributes from other users including credential values for a set S_d, i.e (μ_j, Sa_j) for $j \in I_{S_d}$. Given the set of attributes embedded in the credential S and a set S_d of delegated attributes the PseuGen algorithm generates a pseudonym for the attributes $S' = S \cup S_d$ where $I_{S'} = \{i'_1, \ldots, i'_{k'}\}$ by performing:

1) picks a random public value $z \xleftarrow{R} Z_p^*$
2) computes $d = H(z)$ ($H()$ being a hash function as per CA)
3) computes secret value $\mu' = (d - \mu)/2$
4) computes $Pu = (\mu + \mu')P$
5) computes pseudonym components for the attributes included in the user's credential :$Pa_i = \mu' Sa_i\ \forall\ i \in I_{S'}$ if $i \in I_S$
6) computes pseudonym components for the delegated attributes $Pa_j = (\mu_j - (\mu + \mu'))Sa_j\ \forall\ j \in I_{S'}$ if $j \in I_{S_d}$.
7) computes random simulated pseudonym components for the rest of the attributes $Pa_i = \alpha_i P$ where $\alpha_i \xleftarrow{R} Z_p^*\ \forall\ i \in I_{S'}^c$

The pseudonym for the subset S' is the tuple of $1 + n$ elements in G_1 and 1 element in Z_p^*, $pseu_{S'} = (Pu, Pa_1 \ldots, Pa_n, z)$. The secret value associated with this pseudonym is μ'. Note that the user can run this algorithm multiple times for the same subset S', obtaining new unlinkable pseudonyms with a different associated secret value.

Sign(μ,μ',$\{\mu_j\}$,$pseu_{S'}$,M,PP, AT, S_{ad}) the algorithm runs on the user side and requires a valid pseudonym $pseu_{S'}$, the secret value μ of the user's credential, the secret value μ_j for any delegated attribute, i.e. $j \in I_{S_d}$, and the pseudonym associated secret value μ'. The algorithm also requires the access tree structure AT and the set of attributes that admit delegation S_{ad}, both values are published by the verifier and hence are of public knowledge. The algorithms outputs a signature for a message M of arbitrary length by performing the following steps:

1) selects random factors r_1, r_3, $r_5 \xleftarrow{R} Z_p^*$
2) computes commitment[5] $T_{G_1} = r_1 P \in G_1$
3) computes commitments $t_3 = h^{r_3} g^{-r_1}$ and $t_5 = h^{r_5}$
4) selects random factors $r_{2,i} \xleftarrow{R} Z_p^*$ for $i \in I_{S'}$
5) computes commitments $t_{2,i} = [e(P, P + Pa_i)]^{r_{2,i}}$ for $i \in I_{S'}$
6) for attributes that do not admit delegation, i.e. $i \in S_{ad}$
   a) selects random factors $r_{4,i} \xleftarrow{R} Z_p^*$ for $i \notin I_{S'}$
   b) computes commitments $t_{4,i} = h^{r_{4,i}} g^{-r_{2,i}}$ for $i \in I_{S'}$
7) selects random factors δ and γ
8) computes auxiliary public keys $\tilde{y}_1 = h^{\gamma} g^{\mu + \mu'}$ and $\tilde{y}_2 = h^{\delta} g^{\mu'}$
9) obtains the challenges c_i for $i \in I_{S'^c}$ by filling the non-satisfying leaf nodes in the access tree by performing the phase 1 of the access tree construction as described in sec. VI-B.
10) for the attributes for which the user does not have a credential value or delegated value, i.e. $i \notin I_{S'}$:

[4] ID is a unique identifier to which several datasets are correlated, mainly consisting of user data

[5] a commitment is a probabilistic algorithm over a random value $r \in Z_p$

a) selects random responses $s_{2,i} \xleftarrow{R} Z_p^*$
b) computes simulated commitments:

$$t_{2,i} = \frac{e(P, P + Pa_i)^{s_{2,i}}}{e(Pu + W_i, Pa_i)^{c_i}} \quad (2)$$

c) if the attribute does not admit delegation, i.e. $i \notin S_{ad}$, then it also generates random responses $s_{4,i} \xleftarrow{R} Z_p^*$ and computes simulated commitments:

$$t_{4,i} = h^{s_{4,i}} g^{-s_{2,i}} \tilde{y}_2^{c_i} \quad (3)$$

11) obtains the challenge c:

$$c = H(Pu||Pa_1||\dots||Pa_n||T_{G_1}||t_{2,1}||\dots||t_{2,n}|| t_3||\{t_{4,i}\}||t_5||\tilde{y}_1||\tilde{y}_2||M||z) \quad (4)$$

where $\{t_{4,i}\}$ is the set of commitments for attributes that do not admit delegation.

12) obtains the set of $\{c_i\}$ for $i \in S'$ by completing the access tree for the satisfying leaf nodes performing phase 2 of access tree construction as described in sec. VI-B,
13) computes responses $s_1 = r_1 + (\mu + \mu')c$
14) computes responses $s_{2,i} = r_{2,i} + \mu' c_i$ for $i \in I_{S'}$
15) computes responses $s_3 = -c\gamma + r_3$, $s_5 = -c(\delta + \gamma) + r_5$
16) computes responses $s_{4,i} = -c_i\delta + r_{4,i}\ \forall\ i \in I_{S'^c}$ if the attribute does not admit delegation, i.e. $i \notin S_{ad}$

The signature is the tuple $\sigma = (c, c_1, \dots, c_n, s_1, s_{2,1}, \dots, s_{2,n}, s_3, \{s_{4,i}\}, s_5)$ where the set $\{c_1, \dots, c_n\} = \{c_i\} \cup \{c_j\}$ and $\{s_{2,1}, \dots, s_{2,n}\} = \{s_{2,i}\} \cup \{s_{2,j}\}$ for $i \in I_{S'}$ and $j \in I_{S'^c}$. Note that the values $\{s_{4,i}\}$ are only included for attributes that do not admit delegation.

SignCheck(σ,$pseu_{S'}$,M,PP,AT, S_{ad}): The algorithm runs on the verifier side and checks the validity of a signature of a message M, for a given pseudonym $pseu_{S'}$ (although included in this notation, the attribute set S' is hidden from the verifier thus preserving privacy). It also checks whether the pseudonym complies with the required access structure AT and the specified set of not delegatable attributes S_{ad}. The verifier performs the following operations:

1) computes $d = H(z)$
2) computes $\bar{T}_{G_1} = s_1 P - cPu$
3) for $i \in I_A$ computes

$$\bar{t}_{2,i} = \frac{e(P, P + Pa_i)^{s_{2,i}}}{e(Pu + W_i, Pa_i)^{c_i}} \quad (5)$$

4) computes $\bar{t}_3 = h^{s_3} g^{-s_1} \tilde{y}_1^c$
5) computes $\bar{t}_{4,i} = h^{s_{4,i}} g^{-s_{2,i}} \tilde{y}_2^c$ for all attributes that do not admit delegation, i.e. $i \notin S_{ad}$.
6) computes $\bar{t}_5 = h^{s_5} (\frac{\tilde{y}_1 \tilde{y}_2}{g^d})^c$
7) checks correctness of the challenge c:

$$c \stackrel{?}{=} H(Pu||Pa_1||\dots||Pa_n||T_{G_1}||t_{2,1}||\dots||t_{2,n}|| t_3||\{t_{4,i}\}||t_5||\tilde{y}_1||\tilde{y}_2||M||z)$$

The SignCheck algorithm flags the signature as valid if the set of shares $\{c_1, \dots, c_n\}$ is consistent with the secret c, as described in sec. VI-C.

Revoke($pseu_{S'}$): The algorithm runs at the CA side to find the owner of a valid pseudonym when that pseudonym is reported as misuser by a service provider. The CA has stored tuples of the form $(ID, cred, S)$ produced during credential generation, i.e. *CreGen* algorithm. For each stored tuple the CA performs the following operations:

1) checks whether S contains any attribute used in $pseu_{S'}$, otherwise the CA discards this tuple and continues to the next stored tuple.
2) $\forall\ Pa_i$ in $pseu_{S'}$ such that $i \in I_S$, gets Sa_i from $cred$ and checks whether the following equation holds:

$$e(Pu + W_i, Sa_i) \stackrel{?}{=} e(P, P + Pa_i) \quad (6)$$

if it holds for at least one Pa_i returns ID as the identity of the misuser, otherwise continues to next stored tuple.

A. Protocol Variants

The described protocol enables a credential holder (i.e. user) to generate an unlimited number of unlinkable pseudonyms. However, each user can only generate one specific pseudonym per value z. In the described construction of the PseuGen algorithm this value is obtained randomly. Since this value is public, it can also be chosen by the verifier beforehand. This is useful in scenarios where the service provider wants to limit the number of pseudonyms per user. For example, the service provider may want that users create anonymous accounts with only one pseudonym, hence it enables one specific service index from which users should generate their pseudonym [23].

It is also worth commenting that the credential generation described in the CreGen algorithm involves the CA learning the credential. This implies that the CA must be considered honest. Although this is a frequently adopted model, it is possible to relax this requirement if the credential is issued privately. As an example, the work in [22] provides an equivalent credential generation as the one described in CreGen algorithm (since it involves the same pseudo-random function), but without the CA learning the credential. This is achieved with homomorphic encryption. Although the pseudonym construction in [22] is drastically different to the one proposed in this paper, it could be possible to adopt a similar approach for credential construction to prevent the CA from learning credentials.

VI. Access Tree structure

The access structure of the associated secret sharing scheme $\S()$ can be represented with an access tree in a similar fashion as [54],[55], which follows the transformation proposed in [56]. However on our proposed scheme we require an access tree construction process that is divided in two phases, sec. VI-B. Let's consider a threshold secret sharing scheme $\S_{(k,num)}$ such as the one described in sec. IV-C1. Each non-leaf node x is a threshold gate $1 \leq k_x \leq num_x$ where num_x denotes the number of children of the node x, and k_x

the number of children that have to be satisfied to reconstruct the secret value of the node x, which we denote as $secret(x)$. The set of children nodes of x is denoted as $child(x)$ and are numbered with indexes from 1 to num_x. The parent node of a node x is denoted as $parent(x)$, and the index value of a node x with respect to its parent is denoted as $ind(x)$. Each leaf node in the tree corresponds to a different attribute in $A = \{att_1, \ldots, att_n\}$.

A. Dual access structure

The convenience of using an access tree structure is that the dual structure can be obtained by simply replacing threshold gates of the form $\S_{(k,num)}$ by $\S_{(num-k+1,num)}$. Note that if a node x has $k_x = 1$ or $k_x = num_x$ then the node is an OR gate or an AND gate respectively, and in the dual access structure the OR gate becomes an AND gate and vice versa.

Algorithm 1 BOTTOM_UP_FILLING(node, share)

START
SET $secret(node) = share$
GET $p = parent(node)$
GET $C_p = \{x | x \in child(p)\ and\ secret(x) \neq 0\}$ and let $|C_p|$ be its size
GET $C'_p = \{x | x \in child(p)\ and\ secret(x) = 0\}$
GET threshold gate parameters k_p and num_p
IF $|C_p| = k_p$
  SET $\S_{(k_p,num_p)}$ for k_p points $(ind(x), secret(x))$ for all $x \in C_p$
  FOR all $x \in C'_p$
    GET point $(ind(x), share_x)$ by interpolation from $\S_{(k_p,num_p)}$
    CALL algorithm $TOP\_DOWN\_FILLING(x, share_x)$
  ENDFOR
  GET point $(0, share_p)$ by interpolation from $\S_{(k_p,num_p)}$
  CALL algorithm $BOTTOM\_UP\_FILLING(p, share_p)$
ENDIF
END

B. Access tree construction

The access tree construction is required in the Sign algorithm, to obtain the challenges c_i for $i = 1, \ldots, n$, and it is performed in two phases: i) phase 1, it obtains the challenges for the simulated commitments, i.e. to commit the pseudonym components of S'^c corresponding to the attributes for which the user does not have the corresponding secret keys; ii) phase 2, it obtains the challenges for the commitments of the pseudonym components for which the user has the corresponding secret keys.

- phase 1: Iteratively select a random share $share_x \xleftarrow{R} Z_p^*$ and call $BOTTOM\_UP\_FILLING(x, share_x)$ for all leaf nodes $x \in S'^c$ that have not been assigned a secret share in a previous iteration of phase 1.
- phase 2: Call $TOP\_DOWN\_FILLING(r, c)$ where r is the root node of the access tree structure and c the challenge obtained in Sign algorithm (sec. V).

Algorithm 2 TOP_DOWN_FILLING(node, share)

START
SET $secret(node) = share$
GET $C_{node} = \{x | x \in child(node)\ and\ secret(x) \neq 0\}$ and let $|C_{node}|$ be its size
GET $C'_{node} = \{x | x \in child(node)\ and\ secret(x) = 0\}$
IF $|C_{node}| = 0$ **THEN** return;
ENDIF
GET threshold gate parameters k_{node} and num_{node}
FOR all $x \in child(node)$
  IF $x \in C_{node}$ **THEN** continue;
  IF $|C_{node}| >= k_{node}$
    SET $\S_{(k_p,num_p)}$ for k_p points $(ind(y), secret(y))$ for all $y \in C_{node}$
    GET point $(ind(x), share_x)$ by interpolation from $\S_{(k_p,num_p)}$
    CALL algorithm $TOP\_DOWN\_FILLING(x, share_x)$
  ELSE
    GET point $(ind(x), share_x)$ where $share_x \xleftarrow{R} Z_p^*$
    CALL algorithm $TOP\_DOWN\_FILLING(x, share_x)$
    SET $C_{node} = \{C_{node} \cup x\}$
  ENDIF
ENDFOR
END

Both algorithms, $BOTTOM\_UP\_FILLING$ and $TOP\_DOWN\_FILLING$, are described for the general case where nodes in the tree are described as threshold gates, i.e. $\S_{(k,n)}$, however in the special cases where $k = 1$ and $k = n$ the threshold gates are OR and AND gates respectively, which admits a simpler construction than Shamir secret sharing scheme [56]: i) for an OR gate x with $secret(x) = s$, all the $child(x)$ nodes receive s as secret share; and ii) for an AND gate x with $secret(x) = s$ each child node $i \in child(x)$ is assigned a $secret(i) = s_i$ such that $\sum_{i=1}^{num_x} s_i = s$.

C. Access tree paths satisfiability

The consistency of a full set of shares $S = \{c_1, \ldots, c_n\}$ for a secret c and an access tree structure Γ is evaluated by verifying the satisfiability of all minimal qualified sets $S' \in \Gamma$. We denote the subtree of a node x in Γ as Γ_x, where the node x is a threshold gate of the form $\S_{(k_x,num_x)}$. The secret share of x, i.e. $secret(x)$, can be obtained if the subtree Γ_x is satisfied for the set S', i.e. $\Gamma_x(S') = 1$, and can be evaluated by performing recursively:

- if x is a leaf-node then $\Gamma_x(S') = 1$ if $x \in S'$, otherwise $\Gamma_x(S') = 0$. If satisfied then $secret(x)$ is already included in the set of shares to be evaluated.
- if x is a non-leaf node then evaluate $\Gamma_{x'}(S')$ for all $x' \in child(x)$, if at least k_x child nodes return $\Gamma_{x'}(S') = 1$ then $\Gamma_x(S') = 1$, otherwise $\Gamma_x(S') = 0$. If x is satisfied then $secret(x)$ can be obtained by polynomial interpolation.

The full set of shares S is consistent for c with respect to an access tree structure Γ if all minimal qualified sets S' are satisfied for the root node r, i.e. $\Gamma_r(S') = 1$ and the share of the root node matches the secret c, i.e. $secret(r) = c$.

VII. CORRECTNESS

The signature scheme is correct, if and only if, for all signatures σ of a message M generated by a *Sign* algorithm with valid pseudonyms and valid credentials, the output of the *SignChek* algorithm is always "valid".

Given a valid signature $\sigma = (c, c_1, \ldots, c_n, s_1, s_{2,1}, \ldots, s_{2,n}, s_3, \{s_{4,i}\}, s_5)$, with $i \in S^c_{ad}$ (attributes that do not admit delegation), of a message M, with a pseudonym $pseu_{S'} = (Pu, Pa_1, \ldots, Pa_n, z)$ for a set S' of k' attributes, and a secret value μ' obtained from a credential $(\mu, Sa_{i_1}, \ldots, Sa_{i_k})$ and delegated attributes (mu_d, Sa_j) where $j \in S_{ad}$. The SignCheck algorithm flags a signature as valid if $\bar{T}_{G_1} = T_{G_1}$, $\bar{t}_{2,i} = t_{2,i}$, $\bar{t}_3 = t_3$, $\bar{t}_{4,i} = t_{4,i}$ and $\bar{t}_5 = t_5$. Also the set of shares $\{c_1, c_n\}$ should be consistent with the challenge c. These equalities hold in the proposed scheme since:

$$\begin{aligned}\bar{T}_{G_1} = s_1 P - cPu = \\ (c(\mu+\mu') + r_1)P - c(\mu+\mu')P = r_1 P = T_{G_1}\end{aligned} \tag{7}$$

Regarding the simulated commitments it is straightforward to see that the SignCheck algorithm always obtains the correct $\bar{t}_{2,i}$ value. For the rest of commitments for which the user possesses the secret values Sa_i, we have that:

$$\begin{aligned}\bar{t}_{2,i} = \frac{e(P, P+Pa_i)^{s_{2,i}}}{e(Pu+W_i, Pa_i)^{c_i}} = \frac{[e(P,P)e(P,Pa_i)]^{s_{2,i}}}{e((\mu+\mu'+s_i)P, \frac{\mu'}{\mu+s_i}P)^{c_i}} = \\ \frac{[e(P,P)e(P,Pa_i)]^{s_{2,i}}}{[e((\mu+s_i)P, \frac{\mu'}{\mu+s_i}P)e(\mu' P, \frac{\mu'}{\mu+s_i}P)]^{c_i}} = \\ \frac{[e(P,P)e(P,Pa_i)]^{s_{2,i}}}{[e(P,P)e(P,Pa_i)]^{\mu' c_i}} = \\ \frac{[e(P,P+Pa_i)]^{r_{2,i}+\mu' c_i}}{[e(P,P+Pa_i)]^{\mu' c_i}} = t_{2,i}\end{aligned} \tag{8}$$

Regarding the commitments for the attributes that do not admit delegation we also have that:

$$\begin{aligned}\bar{t}_3 = h^{s_3} g^{s_1} \tilde{y}_1^c = h^{-c\gamma + r_3} g^{-c(\mu+\mu')+r_1} (h^\gamma g^{(\mu+\mu')})^c = \\ h^{r_3} g^{-r_1} = t_3\end{aligned} \tag{9}$$

$$\begin{aligned}\bar{t}_{4,i} = h^{s_{4,i}} g^{s_{2,i}} \tilde{y}_2^{c_i} = h^{-c_i\delta + r_{4,i}} g^{-c_i(\mu')+r_{2,i}} (h^\delta g^{(\mu')})^{c_i} = \\ h^{r_{4,i}} g^{-r_{2,i}} = t_{4,i}\end{aligned} \tag{10}$$

$$\begin{aligned}\bar{t}_5 = h^{s_5} \left(\frac{\tilde{y}_1 \tilde{y}_2}{g^d}\right)^c = h^{-c(\delta+\gamma)+r_5} \left(\frac{h^\gamma g^{(\mu+\mu')} h^\delta g^{(\mu')}}{g^d}\right)^c = \\ h^{r_5} g^{(\mu+2\mu'-d)} = h^{r_5} = t_5\end{aligned} \tag{11}$$

To show the correctness of the signature scheme for the delegated attributes, we prove that for a user with a credential with secret value μ and pseudonym secret value μ' and delegated credential value of the i-th attribute $(\mu_d, Sa_i) = (\mu_d, \frac{1}{s_i+\mu_d}P)$, the following equation holds if the pseudonym for the delegated attribute is constructed as $Pa_i = \mu'_d Sa_i$ where $\mu'_d = (\mu+\mu') - \mu_d$:

$$\begin{aligned}e(Pu + W_i, Sa_i) = e(P, P + Pa_i) \Rightarrow \\ e((\mu+\mu')P + s_i P, Sa_i) = e(P, P + \mu'_d \frac{1}{s_i+\mu_d} P) \Rightarrow \\ e(P,P)^{\frac{\mu+s_i}{\mu_d+s_i}} e(P, Sa_i)^{\mu'} = e(P,P) e(P, Sa_i)^{\mu'_d} \Rightarrow \\ e(P,P)^{\frac{\mu-\mu_d}{s_i+\mu_d}} = e(P, Sa_i)^{\mu'_d - \mu'} \Rightarrow \\ e(P,P) = e(P,P)^{\frac{\mu'_d-\mu'}{\mu-\mu_d}} \Rightarrow \mu'_d = (\mu+\mu') - \mu_d\end{aligned} \tag{12}$$

Hence, we have that for delegated attributes:

$$\begin{aligned}\bar{t}_{2,i} = \frac{e(P, P+Pa_i)^{s_{2,i}}}{e(Pu+W_i, Pa_i)^{c_i}} = \frac{e(P, P+Pa_i)^{s_{2,i}}}{e(Pu+W_i, Sa_i)^{\mu'_d c_i}} = \\ \frac{e(P, P+Pa_i)^{s_{2,i}}}{e(P, P+Pa_i)^{\mu'_d c_i}} = t_{2,i}\end{aligned} \tag{13}$$

The SignCheck algorithm also requires consistency of the set of shares $\{c_1, \ldots, c_n\}$ for the secret c with respect to the access tree structure Γ^*, which requires satisfiability of all minimal qualified sets in Γ^* as described in VI-C. It is straightforward to see that the top-down access tree construction algorithm 2 yields such a consistent set of shares. Note that by definition (see sec. IV-B1) any qualified set S' in Γ has non empty intersection with all minimal qualified sets in Γ^*; therefore, if the signature is constructed with secret keys forming a qualified set in Γ, the phase 2 of the access tree construction (sec. VI-B) selects at least one of the shares in all qualified sets in Γ^* and assigns the secret c to the root node.

VIII. SECURITY ANALYSIS

This section provides the security analysis of the proposed scheme which focuses on the unforgeability of the signature scheme and the resiliency to collusion attacks. To give some insight: i) unforgeability ensures that if a user outputs a valid signature for a valid pseudonym, then the user has got a valid credential containing the attributes that satisfy the access structure; ii) resiliency to collusion attacks imply that if two colluding users output a valid signature with attributes that do not allow collusion, then one of the users has a credential including those attributes.

Unforgeability The pseudonym-based signature scheme is said to be strongly existentially unforgeable under the adaptive-chosen message attack if no probabilistic polynomial time (PPT) adversary A has a non-negligible advantage, making a maximum number of signature queries, in the following game between the adversary A and a challenger C:

GAME 1:

1) *Setup: C runs the KeyGen algorithm and obtains the public parameters. C sends the parameters to A*

2) *Adversary Queries: A makes queries to C:*
   - *credential queries: A queries a credential on a set of k attributes S to C, then C uses the CreGen algorithm and returns a credential $(\mu, Sa_{i_1}, \ldots, Sa_{i_k})$ to A.*
   - *pseudonym queries: A presents a credential to C, and a subset of attributes S and queries a pseudonym to C. C runs PseuGen and returns a pseudonym $(Pu, Pa_1, \ldots, Pa_n, z)$ and its secret μ'.*
   - *signature queries: A sends a message M, a pseudonym $(Pu, Pa_1, \ldots, Pa_n)$, and the secret values $(\mu, \mu', Sa_{i_1}, \ldots, Sa_{i_k})$ to C. C runs the Sign algorithm and returns σ to A*
3) *After a polynomial number of queries, A outputs a signature σ on a chosen message m for a pseudonym that was never queried and which corresponding secret values were never obtained during the pseudonym and credential queries. A wins the game if the SignCheck algorithm flags the signature σ as "valid".*

Theorem 1: In the random oracle model, and under the adaptive chosen message attack, if a PPT algorithm has a non negligible probability ϵ of breaking the unforgeability property then there exists an algorithm C that breaks the k-CAA assumption with an advantage $(\kappa/q_{key})^{q_{key}}(1-(q_H/p))$ for a polynomial number of q_{key} credential queries and q_H oracle queries.

proof: This proof follows analog steps to proof in [57] and [23]. First, let's consider n instances of the k-CAA problem as defined in sec. IV, i.e. $x_i, a_1, \ldots, a_k \in Z_p$ for $i \in \{1, \ldots, n\}$ (where k is an integer). Then C adopts the role of the challenger in the *GAME 1* and uses the adversary A as sub-algorithm as follows:

Challenger C computes the system public parameters and sends the parameters to the adversary A. These parameters include the public keys $W_i = x_i P$. Then the challenger lets the adversary A perform a series of queries:

Adversary Queries: The adversary A makes a polynomial number of queries to C:

1) **Credential Queries**: The challenger C prepares a polynomial number of responses for the credential query $\{w_1, w_2, \ldots, w_{q_{key}}\}$, and the set $\{a_1, \ldots, a_k\}$ is randomly distributed among these responses. When A queries a credential for k' attributes, C picks randomly within the predefined set, if such pick falls within the set of $\{a_1, \ldots, a_k\}$ then C generates the credential with values $(a_i, A_i) = (a_i, 1/(x_1 + a_i)P, \ldots, 1/(x_n + a_i)P)$. C returns to A the credential. Otherwise, if the random pick yields a value out of the set $\{a_1, \ldots, a_k\}$, the challenger aborts. The probability of not aborting is $(k/q_{key})^{q_{key}}$.
2) **Pseudonym Queries** : A presents a credential $cred = (a_i, Sa_{i,1}, \ldots, Sa_{i,k})$ that was previously queried to C, then C obtains a random a'_i and returns $P_u = (a_i + a'_i)P$, and $Pa_{i,1} \ldots Pa_{i,k}$ using the PseuGen algorithm.
3) **Signature Queries**: A sends a message M and a pseudonym to C who returns the signature $(c, c_1, \ldots, c_n, s_1, \ldots, s_5)$ to A if the presented pseudonym was previously queried. To compute these values C first obtains c randomly and sets the oracle to c for the received input, if this causes a collision with a previous query to the oracle $H()$ then C aborts. Then C uses the phase 2 of the tree construction (sec. VI) setting the secret value of the root node to c to obtain the challenges $c_1, \ldots, c_n$ and uses the SignCheck algorithm to obtain the responses $s_1, \ldots, s_5$. The probability of not aborting at this step, taking into account that A makes q_H queries to the H oracle, and that the response of this oracle is random in Z_p^*, is $1 - (q_H/p)$.

The challenger also defines the random oracle H as follows:

- $H()$ oracle Queries: On any input of the form $(Pu||Pa_1|| \ldots ||Pa_n||T_{G_1}||t_{2,1}|| \ldots ||t_{2,n}||M||z)$ the challenger C picks $c \xleftarrow{R} Z_p^*$ and responds. C sets the answer of the oracle as c for that input.

Let's assume that after performing a polynomial number of queries, A presents a signature containing $(c, c_1, \ldots, c_n, s_1, s_{2,1}, \ldots, s_{2,n}, s_3, \{s_{4,i}\}, \ldots, s_5)$ that is valid for a pseudonym $(Pu, Pa_1, \ldots, Pa_n)$. The challenger C can verify if the credential of the presented pseudonym was obtained during the credential queries by checking whether $e(Pu + W_i, A_i) = e(P, P + Pa_i)$ holds for any of the queried credentials. If the credential was not queried, and the signature is valid, for the Forking Lemma[6] [58] the adversary will be able to output another valid signature with the same inputs but different challenge c'. This is, A will be able to present after polynomial time another signature σ' with the same commitments $T_{G_1} = T'_{G_1}$, and $t_{2,i} = t'_{2,i}$, but different challenge and responses, i.e. $(c', c'_1, \ldots, c'_n, s'_1, s'_{2,1} \ldots, s'_{2,n})$ such that $c \neq c'$, which means that $s_1 \neq s'_1$ and at least one $c_i \neq c'_i$ and one $s_{2,i} \neq s'_{2,i}$ in every qualified set in Γ^*.

Then A can solve the $i-th$ instance of the k-CAA problem. Since:

$$\begin{aligned} &T_{G_1} = T'_{G_1} \Rightarrow s_1 P - cPu = s'_1 P - c' Pu \Rightarrow \\ &(s_1 - s'_1)P = (c - c')Pu \Rightarrow Pu = \tfrac{(s_1 - s'_1)}{(c - c')} P \end{aligned} \tag{14}$$

Also:

$$\begin{aligned} &t_{2,i} = t'_{2,i} \Rightarrow \tfrac{[e(P,P)e(P,Pa_i)]^{s_{2,i}}}{e(Pu+W,Pa_i)^{c_i}} = \tfrac{[e(P,P)e(P,Pa_i)]^{s'_{2,i}}}{e(Pu+W,Pa_i)^{c'_i}} \Rightarrow \\ &e(P,P)^{\frac{s_{2,i}-s'_{2,i}}{c_i-c'_i}} e(\tfrac{s_{2,i}-s'_{2,i}}{c_i-c'_i} P, Pa_i) = e(Pu + W_i, Pa_i) \Rightarrow \\ &e(P,P)^{(s_{2,i}-s'_{2,i})/(c_i-c'_i)} = e\left(Pu - \tfrac{s_{2,i}-s'_{2,i}}{c_i-c'_i} P + W, Pa_i\right) \Rightarrow \\ &e\left(\tfrac{(s_1-s'_1)(c_i-c'_i)-(s_{2,i}-s'_{2,i})(c-c')}{(c-c')(c_i-c'_i)} P + x_i P, \tfrac{c_i-c'_i}{s_{2,i}-s_{2,i}} Pa_i\right) \\ &= e(P,P) \end{aligned} \tag{15}$$

[6]According to the Forking lemma, if an algorithm can yield an output, from some inputs obtained from a given distribution, and this output has some property with non-negligible probability, then the adversary has a non-negligible probability of producing another output with the same property provided that the inputs are chosen from the same distribution.

Hence, the adversary A can find a solution to the i-th instance of the k-CAA problem $(a, S = \frac{1}{a+x_i}P)$ where

$$S = \frac{c_i - c'_i}{s_{2,i} - s_{2,i}} Pa_i \tag{16}$$

and

$$a = \frac{(s_1 - s'_1)(c_i - c'_i) - (s_{2,i} - s'_{2,i})(c - c')}{(c - c')(c_i - c'_i)} \tag{17}$$

Since the probability of not aborting in this game is $(\kappa/q_{key})^{q_{key}}(1 - (q_H/p))$ the advantage of C in solving the k-CAA problem is not negligible. Note that, solving the i-th instance of the K-CAA problem is equivalent to forging the corresponding secret key of the i-th attribute. Also, following the same reasoning as in [53], since the challenges $c \neq c'$ for the root node in the access tree Γ^*, for every minimal qualified set S in Γ^*, there is at least one $i \in I_S$ for which $c_i \neq c'_i$. This means that the adversary A can obtain with no negligible probability a secret key for at least one attribute in every qualified set in Γ^*. Hence, following the definition in IV-B1, the adversary A can obtain secret keys for a qualified set in Γ.

Collusion resiliency: The signature scheme is collusion resistant if any PPT adversary A has a non-negligible advantage in the GAME 2 between the adversary A and a challenger C:

GAME 2:

1) *Setup: C runs the KeyGen algorithm and obtains the public parameters. C sends the parameters to A*
2) *Adversary Queries: A makes to C:*
   - *credential queries: A queries a credential on a set of k attributes S to C, then C uses the CreGen algorithm and returns a credential $(\mu, Sa_{i_1}, \ldots, Sa_{i_k})$ to A.*
   - *pseudonym queries: A presents a credential to C, and a subset of attributes S and queries a pseudonym to C for a specific value z. C runs PseuGen and returns a pseudonym $(Pu, Pa_1, \ldots, Pa_n)$ and its secret μ'.*
   - *signature queries: A sends a message m, a pseudonym $(Pu, Pa_1, \ldots, Pa_n)$ for a specific value z, and the secret values $(\mu, \mu', Sa_{i_1}, \ldots, Sa_{i_k})$ to C. C runs the Sign algorithm and returns σ to A*
3) *After a polynomial number of queries, A outputs a valid signature σ, and pseudonym $(P_u, Pa_1, \ldots, Pa_n)$ for a specific value z, on the message m for an access tree structure which qualified sets contain attributes that do not admit collusion and that has not been previously queried by a single credential.*

Theorem 2 If a PPT adversary has a non negligible probability ϵ of winning the GAME 2, then it exists an algorithm C that is able to solve the discrete logarithm problem in G_T with no negligible advantage ϵ.

proof This proof follows similar reasoning as in [59]. Let us assume that C is given as instance two random elements g and $h \in G_T$, for which the discrete logarithm a is not know $g^a = h$. Then the challenger C can use these values as generators of G_T in the Setup phase and use A as subalgorithm to compute a. This is, after performing the adversary queries as defined above, A presents a valid signature $\sigma = (c, s_1, \{s_{2,i}\}, s_3, \{s_{4,i}\}, s_5, \tilde{y}_1, \tilde{y}_2)$ and a pseudonym $(P_u, Pa_1, \ldots, Pa_n)$ that is successful in the GAME 2. For the Forking lemma, A can obtain in polynomial time another valid signature with the same commitments and auxiliary keys but with different challenge $c \neq c'$ and responses, $s_1 \neq s'_1$, $s_3 \neq s'_3$ and $s_5 \neq s'_5$ and for at least one $c_i \neq c'_i$ and challenges $s_{2,i} \neq s'_{2,i}$, $s_{4,i} \neq s'_{4,i}$ in each qualified set in Γ^*. Then, we have that:

$$\begin{aligned} &t_3 = t'_3 \Rightarrow h^{s_3} g^{-s_1} \tilde{y}_1^c = h^{s'_3} g^{-s'_1} \tilde{y}_1^{c'} \Rightarrow \\ &\tilde{y}_1^{c-c'} = h^{s'_3 - s_3} g^{s_1 - s'_1} \Rightarrow \\ &\tilde{y}_1 = h^{\frac{s'_3 - s_3}{c - c'}} g^{\frac{s_1 - s'_1}{c - c'}} \end{aligned} \tag{18}$$

similarly

$$\begin{aligned} &t_{4,i} = t'_{4,i} \Rightarrow h^{s_{4,i}} g^{-s_{2,i}} \tilde{y}_2^{c_i} = h^{s'_{4,i}} g^{-s'_{2,i}} \tilde{y}_2^{c'_i} \Rightarrow \\ &\tilde{y}_2 = h^{\frac{s'_{4,i} - s_{4,i}}{c_i - c'_i}} g^{\frac{s_{2,i} - s'_{2,i}}{c_i - c'_i}} \end{aligned} \tag{19}$$

$$\begin{aligned} &t_5 = t'_5 \Rightarrow h^{s_5} \tilde{y}_{1,2}^c = h^{s'_5} \tilde{y}_{1,2}^{c'} \Rightarrow \\ &\tilde{y}_{1,2} = h^{\frac{s'_5 - s_5}{c - c'}} \end{aligned} \tag{20}$$

Since $\tilde{y}_{1,2} = \frac{\tilde{y}_1 \tilde{y}_2}{g^d}$, from eqs. (18), (19) and (20) we have that:

$$h^{\frac{s'_3 - s_3}{c - c'} + \frac{s'_{4,i} - s_{4,i}}{c_i - c'_i}} g^{\frac{s_1 - s'_1}{c - c'} + \frac{s_{2,i} - s'_{2,i}}{c_i - c'_i}} g^{-d} = h^{\frac{s'_5 - s_5}{c - c'}} \tag{21}$$

In the above equation the exponent of g must be zero

$$\frac{s_1 - s'_1}{c - c'} + \frac{s_{2,i} - s'_{2,i}}{c_i - c'_i} - d = 0 \tag{22}$$

otherwise it is possible to obtain the discrete logarithm of h with respect to g as follows:

$$log_g(h) = \frac{(s_1 - s'_1)(c_i - c'_i) + (s_{2,i} - s'_{2,i})(c - c') - d(c - c')(c_i - c'_i)}{(s'_5 - s_5)(c_i - c'_i) - (s'_3 - s_3)(c_i - c'_i) - (s_{4,i} - s'_{4,i})(c - c')} \tag{23}$$

Let's assume that the i-th attribute was obtained with the first credential $(\mu_1, \frac{1}{s_i + \mu_1}P)$, then $s_1 = c(\mu_1 + \mu'_1) + r_1$, $s'_1 = c'(\mu_1 + \mu'_1) + r_1$, $s_{2,i} = c_i \mu'_1 + r_{2,i}$ and $s'_{2,i} = c'_i \mu'_1 + r_{2,i}$. We have that to nullify the exponent of g in the equation 21 the linear relation $2\mu'_1 + \mu_1 = d$ must hold, which is actually accomplished if the PseuGen algorithm is performed correctly (sec. V), note that in PseuGen we have that $\mu'_1 = (d - \mu_1)/2$.

Now let's consider that the j-th attribute was delegated, i.e. obtained from another credential $(\mu_2, \frac{1}{s_j + \mu_2}P)$. In such case

$s_1 = c(\mu_1 + \mu'_1) + r_1$ and $s'_1 = c'(\mu_1 + \mu'_1) + r_1$ remain unchanged, but on the other hand we have that $s_{2,j} = c_j\mu'_2 + r_{2,j}$ and $s'_{2,j} = c'_j\mu'_2 + r_{2,j}$. We also have that $\mu'_2 = [(\mu_1 + \mu'_1) - \mu_2]$, otherwise correctness does not hold (eq. 12). Following the same reasoning as above we have that:

$$\begin{aligned} \mu'_2 = (d - \mu_1)/2 \Rightarrow 2\mu'_2 + \mu_1 = d \Rightarrow \\ 2[(\mu_1 + \mu'_1) - \mu_2] + \mu_1 = d \end{aligned} \tag{24}$$

From the i-th attribute the equation $2\mu'_1 + \mu_1 = d$ must hold, hence we have that $d + \mu_1 - 2\mu_2 + \mu_1 = d \Rightarrow \mu_1 = \mu_2$. Therefore both credentials must be the same credential.

IX. Implementation

The java library in [60] was used for the implementation of the proposed signature scheme[7]. Namely, a type A curve ($y^2 = x^3 + ax$) over the field F_q with the recommended settings in [61], i.e. the security parameters q and r are set to 512 bits and 160 bits respectively. In this curve, G_1 and G_T are cyclic groups of order a prime number of 160 bits where elements are represented with 1024 bits, and elements in Z_p are of 160 bits. The sizes of the different elements in the credential, pseudonyms and signatures are shown in Table II. It is worth clarifying that the verification of a signature, i.e. the SignCheck algorithm, requires the pseudonym values together with the signature, and the number of pseudonym values depends linearly on the size of the access structure. All tests have been performed in an Intel Core i9 with 16GB RAM.

TABLE II. Signature scheme elements' bit length

Element	bits	Element	bits
credential (Sa_i)	1024	signature challenge (c)	160
credential (μ)	1024	signature challenge (c_i)	160
secret value (μ')	160	signature response (s_1)	160
pseudonym (Pu)	1024	signature response ($s_{2,i}$)	160
pseudonym (Pa_i)	1024	signature response (s_3)	160
pseudonym (z)	160	signature response ($s_{4,i}$)	160
		signature response (s_5)	160

X. Use case scenarios

This section provides four exemplary use cases in the context of a Smart City. This section also provides the performance evaluation for the proposed scenarios, although more exhaustive performance results are given in the following section (sec. IX). Figure 2 depicts the scenarios with the corresponding access structures and the attribute list. Although not included in the figure, in this example the citizens can obtain attributes from a diverse set of entities: i) governmental institutions; ii) qualified medical centres; iii) academic institutions; iv) retailers. These attributes are issued through a trusted certification authority that grants a credential to registered citizens. The issuing process requires the certification authority to authenticate the citizen, and verify that the citizen is authorized to receive the corresponding attributes. Once the citizen holds a credential, he/she can generate unlimited number of unlinkable pseudonyms with embedded attributes, and authenticate towards diverse Smart City cloud services without disclosing the attributes. We include the following examples:

- Public transportation: several public lines connect residential areas to an industrial park. Any citizen living in one of those areas, and working in one of the corporations located in the industrial park can get online tickets at discounted price. Additionally, citizens with a large family, with a mobility impairment or authority members can benefit from the discount. Fig. 3 shows the performance for an access tree with 20 residential areas (postal codes) and 20 different corporations. The results were obtained for one citizen authenticating with the two attributes: i) a postal code; and ii) a corporation.
- Leisure events: the Smart City has a cloud service to manage leisure events. The events can be sponsored by several retailers, and any citizen holding at least two membership cars from the listed retailers, and living in the city, can obtain a free ticket. Citizens must be above 18 years old. Additionally, any person with a low salary range or jobless can obtain a free ticket. Fig. 3 shows the performance for an access tree with 20 postal codes, 20 different retailers, and 5 salary ranges. The results show one citizen authenticating with three attributes: i) a postal code; and ii) two retailer membership cards.
- Public parking service: the Smart City allows free parking in a specific public area to all city residents that have payed the corresponding subscription. Additionally, the citizens with mobility impairments are also entitled to use the parking slots. In this example the public parking area is close to a medical facility offering treatments for several chronic health conditions. Citizens with one of the listed health conditions are entitled to park in the public area to ease the access of vulnerable citizens to medical care. Fig. 3 shows the performance for an access tree structure with 20 postal codes and 10 different health conditions. The results show one citizen authenticated with one attribute: a health condition.
- Job program: the Smart City has a cloud platform offering online courses to improve the technical skills in key sectors for jobless citizens. The access policy requires that the citizen has the jobless status, and at least one of the qualifying degrees. Additionally, several companies have enrolled in the platform and allow some staff members with specific roles (e.g. engineers) to have access to the courses. Fig. 3 shows the performance for an access tree with 10 degrees and 20 corporations. The results show one citizen authenticating with two attributes: i) an academic degree; and ii) the jobless status.

In the proposed scenario, the four cloud services could be centralized through a single cloud platform. This platform can observe all pseudonyms from citizens accessing the diverse Smart City cloud services. However, since the attributes are never disclosed it is not possible for the cloud platform to link different pseudonyms to the same credential holder. Also, since the attributes are hidden, it is not possible to narrow down the user identification process. It is also worth clarifying that the cloud services can limit the number of pseudonyms that a

[7]The experimental implementation (JAVA source code) is available for interested researchers under request via email.

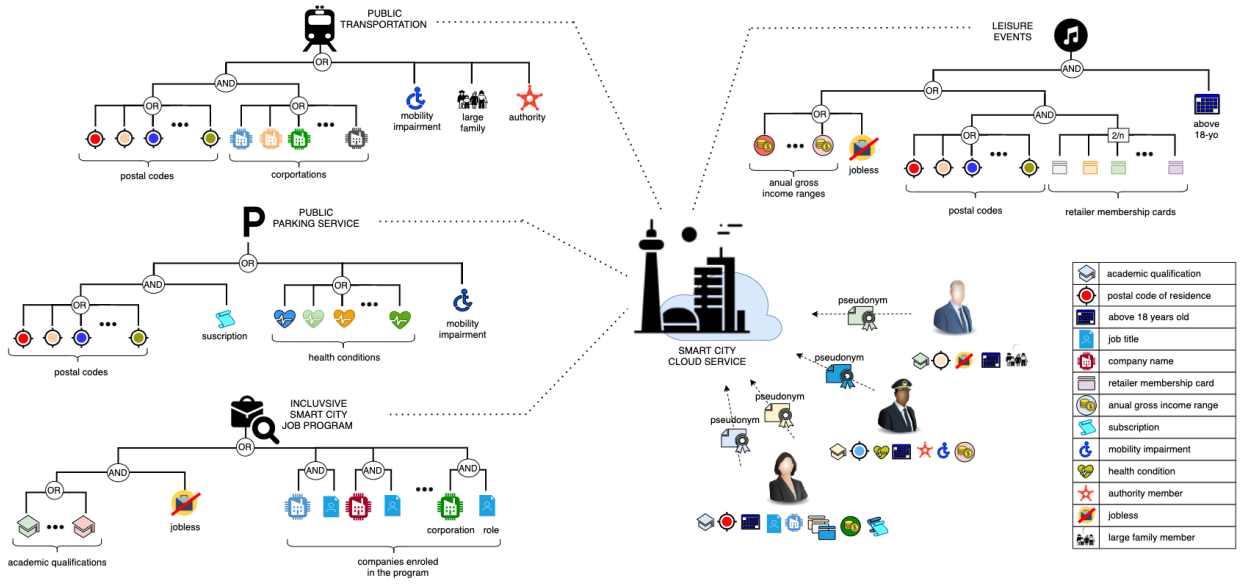

Fig. 2. Exemplary use-cases in the context of Smart City cloud services: i) management of public parking areas; public transportation; management of leisure events; inclusive job program.

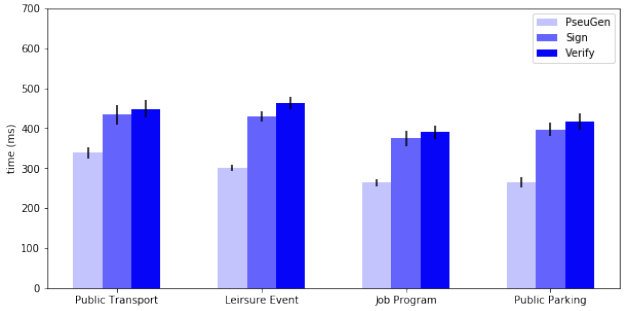

Fig. 3. Algorithms performance (ms) for the exemplary use-cases.

specific user can generate for each service to one pseudonym only, see sec. V-A, hence preventing citizens from using the credential multiple times to obtain benefits. Additionally, some of the attributes can be delegated by other users if the cloud service provider allows this option, e.g. for the leisure events scenario the retailer membership cards could be passed from one user to another, or the above 18 years old attribute could be delegated from an adult to a minor.

XI. Performance Evaluation

In this section we provide a more exhaustive performance evaluation using ring and bus structures. In a ring access structure there is only one OR gate with many attributes, thus the user can authenticate with any of the listed attributes. On the other hand, in the bus access structure the gate is an AND, hence the user must hold all the listed attributes. We have tested both structures for different sizes of the attribute set (between 10 and 100 attributes under the AND/OR gate), and considering both options: i) shareable (delegatable) attributes; and ii) exclusive (non-delegable) attributes. Performance results are shown in Figs. 5 and 4.

It is clear that the time complexity increases linearly with the size of the attribute set in the access structure. It is worth noting that in the ring access structure the user only needs one attribute in his/her credential, whereas in the bus access structure the user holds all the attributes. This difference does not affect the performance of the pseudonym generation (PseuGen) and the signature verification (SignCheck) algorithms. However, the signature generation (Sign) algorithm is 45% more efficient in the bus access structure than in the ring access structure. This is explained by the requirement of simulating all the missing attributes (see sec V), i.e. regardless which attribute the user holds, the user always presents a pseudonym containing pseudonym values for all attributes in the access structure. When those attributes are not included in the user's credential, the user computes random pseudonym values, see V. This does not affect the time complexity of PseuGen algorithm since the arithmetic operations are the same, but affects the Sign algorithm since the commitments for simulated pseudonym values require one more pairing. This must be taken into account in real deployments to avoid timing attacks [62][63]. It is also worth commenting that signature verification is 15% lighter when the attributes are shareable (delegatable), this is because the verifier skips one check during SignCheck algorithm (sec. V). This is not a vulnerability in terms of timing attacks since the list of shareable attributes in the access structure is public.

In terms of memory, the sizes of the signature and pseudonym grow linearly with the number of the attributes in the access structure. The size of the certification authority's public key grows linearly with the number of attributes in the universe of attributes, specifically one key per attribute is required. However, signature verification does not require the full set of public keys, only the subset of public keys for the attributes included in the access structure is required. Specifically, the number of pairings in our proposed signature scheme is $2n - n'$ for signing and $2n$ for verifying, where n is the number of attributes in the access structure and n' is the number of attributes included in the user's credential. In terms of exponentiations of elliptic curve points (group G_1), the proposed signature scheme requires a constant number of 1 for signing and 2 for verifying, whereas the number of expo-

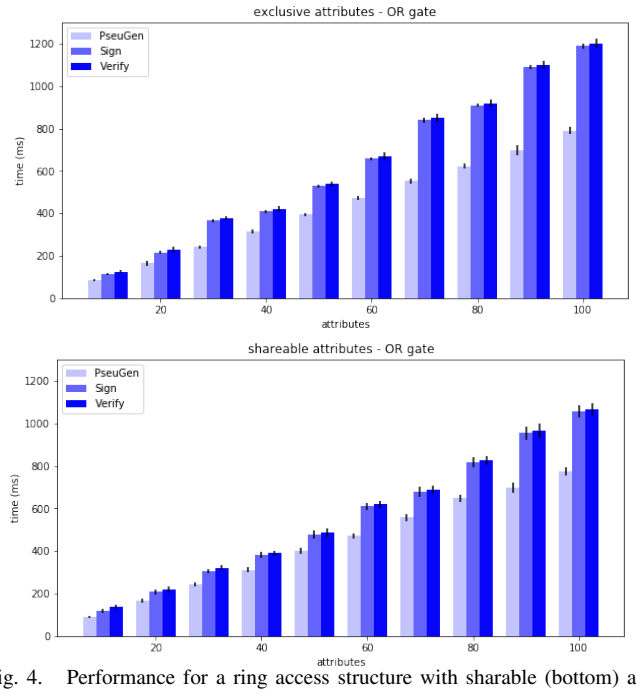

Fig. 4. Performance for a ring access structure with sharable (bottom) and exclusive (top) attributes.

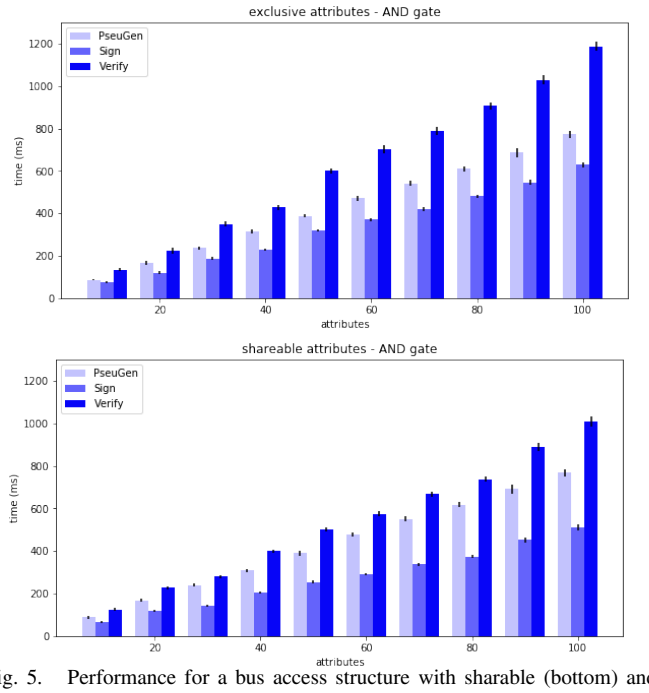

Fig. 5. Performance for a bus access structure with sharable (bottom) and exclusive (top) attributes.

nentiations in the cyclic group G_T is $7n+n'+2(n-n')+3\bar{d}$ for signing and $6+2n+3\bar{d}$ for verifying, where $\bar{d}$ is the number of attributes that do not admit delegation.

XII. Comparison with previous works

A. comparison with ABS schemes

ABS schemes, such as [18], are the most similar schemes in terms of functionality to the scheme proposed in this paper and also have similar complexity. In [18] (implementation 3, which is the most practical) the size of the signature and the complexity of the signature generation and verification algorithms grow linearly with the size of the matrix representing the monotone span program, which depends on the number of attributes and the depth of the access structure. For the ABS scheme in [18], signing does not require pairings but verifying requires $(l+1)t$ pairings, where l and t are the number of rows and columns of the matrix in the monotone span program respectively. The number of exponentiations of elliptic curve points is $6+2lt$ for signing and $(2l+1)t$ for verifying. It is worth clarifying that such matrix can be obtained from any access tree structure of g threshold gates of the form (k_i, n_i) $i \in [1, g]$, which produces a matrix of size $l = \sum_i (k_i - 1) + 1$ and $t = \sum_i (n_i - 1) + 1$. Hence, when applied to a specific access tree structure, the complexity of the ABS scheme depends not only on the size of the attribute set but also on the depth of the access tree structure. The advantage of the ABS scheme is that the public key is constant in size. In terms of functionality, the main similarity between the proposed signature scheme and the ABS scheme is that both hide the user's attributes and only attest the compliancy of the attribute set with respect to the policy defined in the access structure. However, the ABS scheme does not provide a construction to generate unlinkable pseudonyms and revoke misuser's credentials. Therefore, ABS schemes cannot be directly applied to create anonymous accounts.

We have provided scheme complexity details for [18] since this is the most acknowledged work on ABS schemes. However, it is worth mentioning that there are other valuable ABS schemes, such as the work in [64], which caters for a threshold ABS scheme (i.e. the access structure is composed by one threshold gate) where the signature is constant in size regardless the number of attributes. Also, the work in [65] was the first providing a revocation mechanism for the signature scheme, based on short-lived keys, i.e. the users' private keys have a short time of validity after which the users should obtain new private keys from the authority granting the attributes. Subsequent works, such as [20],[21], provide multi-authority attribute issuance with resiliency to colluded malicious authorities if at least one is honest. The work in [66] proposes a signature scheme for a distributed data ledger based on blockchain for medical records. The scheme does not hide the attributes, but it enables both multi-authority attribute issuance and revocation. Some works also propose the adaptation of ABS schemes to IoT and hardware constrained devices by means of server-aided computation [67][68]. It is clear that, in comparison with our proposed pseudonym-based signature scheme, previous ABS schemes can also provide attribute privacy and revocation, and additionally some schemes enable multi-authority for attribute issuance. However, none of the schemes provides a construction to integrate pseudonymity.

B. comparison with Privacy-ABCs

Privacy-ABCs [14],[11],[12], can also be used to authenticate a user privately, and create anonymous accounts by means of pseudonymity. Their complexity also grows linearly with the number of disclosed attributes in terms of proof size and number of operations for proof generation and verification (note that when privacy-ABCs are used as a signature scheme the proof can be seen as the signature). The public key also grows linearly with the number of attributes. Their advantage

is that, unlike ABS schemes, these systems can integrate pseudonym generation in their construction. In the case of [14] including a pseudonym is mandatory, and the same pseudonym is used for each specific credential, hence different authentications with the same credential are linkable. This is called single-show credential, which is convenient in some specific scenarios like electronic cash or e-tickets. The works [11],[14] provide multi-show credentials where the credential holder can generate multiple pseudonyms or even authenticate anonymously (without any pseudonym). Unlike our proposed system, the credential issuance is also anonymous, thus the certification authority does not learn the attribute set of the user. However, the big limitation of privacy-ABCs for private attribute-based authentication is the requirement to disclose attributes during authentication. The cloud service acting as verifier publishes a list of required attributes and the user must disclose those attributes, then prove in zero-knowledge that it holds a credential embedding the disclosed attributes together with some other undisclosed attributes.

Privacy-ABCs have been implemented in diverse scenarios such as road traffic services [69], smart-health [70], device-centric user authentication [71] (where user authentication is performed by means of an anonymous credential stored in a tamper-proof device), or online ticketing systems [72] (which allow users to authenticate anonymously and obtain digital single-use tickets). These systems enable multi-authorities, and also pseudonymity, however they require the disclosure of the attribute set. Some works like [72] implement range proofs [73], which allow credential holders to prove that a given attribute value falls within some range, hence providing a higher level of privacy. However, Privacy-ABCs do not cater for constructions where an expressive access policy on the attribute set can be attested privately.

C. Other works

Conceptually, the work in [16] follows a similar approach to Privacy-ABCs although with a different construction. In this work the verifier publishes a list of attributes, and during authentication the user shows a pseudonym and proves possession of those attributes by means of a private set intersection algorithm [74],[75], which requires homomorphic encryption. The authors provide three different constructions with increasing privacy levels. The highest level hides the user's attributes and shows only the size of the intersection between the user's attributes and the verifier's attribute list. Although this approach effectively hides the user's attributes, it does not provide the verifier with the flexibility to establish complex policies like in ABS schemes or our proposed scheme. It only lets the verifier set a threshold on the minimum number of attributes that the user should posses (thus it is equivalent to a threshold ABS scheme).

Other works propose ABAC systems based on Attribute Based Encryption (ABE). These schemes are not designed to enable anonymous user authentication, but rather provide anonymous access to data objects by means of encryption. Data is encrypted with embedded attributes, i.e. key-policy ABE (KP-ABE) [76], or encrypted with the access policy, i.e. ciphertext-policy ABE (CP-ABE) [77],[78]. The user can access (decrypt) the data if it has a decryption key with the corresponding attributes (CP-ABE) or the corresponding policy (KP-ABE). Generally, in these schemes either the attributes (KP-ABE) or the policy (CP-ABE) of the encrypted object are public, but there are constructions that support a hidden policy KP-ABE such as [79],[80],[81]. Hidden policy schemes require an additional algorithm called decryption test, which enable users to verify that they hold a policy-compliant set of attributes before performing a full decryption attempt. Some works combine ABE with ABS, such as the signcryption with multi-authority presented in [2], that supports MSPs for verification.

In the context of anonymous user authentication, ABE schemes have been adopted in some scenarios, such as the group key agreement protocol in [82],[83], which constructions are based on hashing and polynomial interpolation respectively. These systems enable two entities provided with a set of attributes to authenticate each other and establish a session key if and only if both entities hold the same attribute set. The attribute set of both entities is kept private when the attribute sets do not match (this is a similar concept to affiliation-hiding group signatures [84]). It is also worth mentioning the work in [85], that provides a Privacy-ABC system for fog computing that is not based on zero-knowledge proofs but on hashing. These works could be used for a scenario where users are authenticated anonymously, however these schemes do not support pseudonymity and in the majority of the cases the verification algorithm does not support an expressive attribute policy.

XIII. Conclusion

This paper provides a novel solution for attribute-based privacy-preserving authentication. The proposed scheme enables service providers to authenticate users and verify that their specific attributes comply with the service requirements without learning the users' identity or attribute set. It enables pseudonym self-generation in the user's side, and although it does not provide the same level of anonymity in the attribute issuance process as previous works based on Privacy-ABCs, the proposed scheme does not require the partial disclosure of the attributes in the verification process. The proposed scheme integrates a secret sharing scheme in the signature construction to enable attribute verification, with the advantage that the attributes are integrated into unlinkable self-generated pseudonyms. Also, the proposed scheme supports verifiable delegation of attributes. Service providers can specify which attributes are sharable between users and verify that non-shareable attributes are not delegated. The paper also caters for a comprehensive security analysis and the implementation and performance evaluation of the proposed scheme. Evaluation results show that signature generation and verification can be performed efficiently even with a considerable number of attributes, and that the computation time does not depend on the complexity of the service providers' access structures. As an open challenge, we leave the construction of a pseudonym-based signature scheme with the same properties but compatible with multi-authorities for attribute issuance. Also, the attribute delegation solution provided in this paper is permanent, in a future work we intend to provide a time-based attribute delegation feature.

XIV. Acknowledgment

This work was supported by the H2020-ECSEL SECREDAS project, which has received funding from the ECSEL-JU under grant agreement nr.783119. This Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation programme, and from Austria, Belgium, Czech Republic, Germany, Finland, Hungary, Italy, the Netherlands, Poland, Romania, Sweden and Tunisia.

References

[1] Vincent C. Hu, David Ferraiolo, Rick Kuhn, Adam Schnitzer, Kenneth Sandlin, Robert Miller, and Karen Scarfone. Guide to attribute based access control (abac) definition and considerations (draft). nist special publication 800-162, 2014.

[2] Q. Xu, C. Tan, Z. Fan, W. Zhu, Y. Xiao, and F. Cheng. Secure multi-authority data access control scheme in cloud storage system based on attribute-based signcryption. *IEEE Access*, 6:34051–34074, 2018.

[3] Q. Zhang, S. Wang, D. Zhang, J. Wang, and Y. Zhang. Time and attribute based dual access control and data integrity verifiable scheme in cloud computing applications. *IEEE Access*, 7:137594–137607, 2019.

[4] L. Yeh, Y. Chen, and J. Huang. Abacs: An attribute-based access control system for emergency services over vehicular ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 29(3):630–643, 2011.

[5] Y. Zhu, R. Yu, D. Ma, and W. Cheng-Chung Chu. Cryptographic attribute-based access control (abac) for secure decision making of dynamic policy with multiauthority attribute tokens. *IEEE Transactions on Reliability*, 68(4):1330–1346, 2019.

[6] N. Deng, S. Deng, C. Hu, and K. Lei. An efficient revocable attribute-based signcryption scheme with outsourced unsigncryption in cloud computing. *IEEE Access*, 8:42805–42815, 2020.

[7] A. Thakare, E. Lee, A. Kumar, V. B. Nikam, and Y. Kim. Parbac: Priority-attribute-based rbac model for azure iot cloud. *IEEE Internet of Things Journal*, 7(4):2890–2900, 2020.

[8] S. Ding, J. Cao, C. Li, K. Fan, and H. Li. A novel attribute-based access control scheme using blockchain for iot. *IEEE Access*, 7:38431–38441, 2019.

[9] V. C. Hu, D. R. Kuhn, D. F. Ferraiolo, and J. Voas. Attribute-based access control. *Computer*, 48(2):85–88, 2015.

[10] A. Lehmann, G. Neven, and J. Camenisch. Electronic identities need private credentials. *IEEE Security & Privacy*, 10(01):80–83, jan 2012.

[11] Jan Camenisch and Els Van Herreweghen. Design and implementation of the idemix anonymous credential system. CCS '02, pages 21–30, New York, NY, USA, 2002. ACM.

[12] Christian Paquin and Greg Zaverucha. U-prove cryptographic specification v1.1 (revision 3). Microsoft, December 2013.

[13] Jan Hajny and Lukas Malina. Unlinkable attribute-based credentials with practical revocation on smart-cards. In Stefan Mangard, editor, *Smart Card Research and Advanced Applications*, pages 62–76, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[14] Mira Belenkiy, Melissa Chase, Markulf Kohlweiss, and Anna Lysyanskaya. P-signatures and noninteractive anonymous credentials. In Ran Canetti, editor, *Theory of Cryptography*, pages 356–374, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[15] Latanya Sweeney. K-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, October 2002.

[16] L. Guo, C. Zhang, J. Sun, and Y. Fang. A privacy-preserving attribute-based authentication system for mobile health networks. *IEEE Transactions on Mobile Computing*, 13(01):1, jul 5555.

[17] J. Sun, C. Zhang, L. Guo, and Y. Fang. Paas: A privacy-preserving attribute-based authentication system for ehealth networks. In *2013 IEEE 33rd International Conference on Distributed Computing Systems*, pages 224–233, Los Alamitos, CA, USA, jun 2012. IEEE Computer Society.

[18] Hemanta K. Maji, Manoj Prabhakaran, and Mike Rosulek. Attribute-based signatures. In Aggelos Kiayias, editor, *Topics in Cryptology – CT-RSA 2011*, pages 376–392, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[19] Jin Li, Man Ho Au, Willy Susilo, Dongqing Xie, and Kui Ren. Attribute-based signature and its applications. In *AsiaCCS*, 2010.

[20] R. Guo, H. Shi, Q. Zhao, and D. Zheng. Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems. *IEEE Access*, 6:11676–11686, 2018.

[21] Y. Sun, R. Zhang, X. Wang, K. Gao, and L. Liu. A decentralizing attribute-based signature for healthcare blockchain. In *2018 ICCCN*, pages 1–9, 2018.

[22] J. Camenisch and A. Lehmann. Privacy-preserving user-auditable pseudonym systems. In *2017 IEEE European Symposium on Security and Privacy (EuroS P)*, pages 269–284, 2017.

[23] V. Sucasas, G. Mantas, J. Bastos, F. Damiao, and J. Rodriguez. A signature scheme with unlinkable-yet-acountable pseudonymity for privacy-preserving crowdsensing. *IEEE Transactions on Mobile Computing*, pages 1–1, 2019.

[24] Xavier Boyen. Mesh signatures. In Moni Naor, editor, *EUROCRYPT 2007*, pages 210–227, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[25] Xavier Boyen. Unconditionally anonymous ring and mesh signatures. *J. Cryptology*, 29(4):729–774, October 2016.

[26] Andreas Pfitzmann and Marit Hansen. Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management - a consolidated proposal for terminology. pages 1–83, February 2008.

[27] Andreas Pfitzmann and Marit Hansen. A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management, December 2009. v0.32.

[28] Anna Lysyanskaya, Ronald L. Rivest, Amit Sahai, and Stefan Wolf. Pseudonym systems. In *Proceedings of the 6th Annual International Workshop on Selected Areas in Cryptography*, SAC '99, pages 184–199, London, UK, UK, 2000. Springer-Verlag.

[29] Victor Sucasas, Georgios Mantas, Firooz B. Saghezchi, Ayman Radwan, and Jonathan Rodriguez. An autonomous privacy-preserving authentication scheme for intelligent transportation systems. *Computers & Security*, 60(Supplement C):193 – 205, 2016.

[30] V. Sucasas, G. Mantas, A. Radwan, and J. Rodriguez. An oauth2-based protocol with strong user privacy preservation for smart city mobile e-health apps. In *2016 IEEE International Conference on Communications (ICC)*, pages 1–6, May 2016.

[31] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *CRYPTO*, 2004.

[32] Foteini Baldimtsi and Anna Lysyanskaya. Anonymous credentials light. In *2013 ACM SIGSAC*, CCS '13, pages 1087–1098, New York, NY, USA, 2013. ACM.

[33] Jan Camenisch, Susan Hohenberger, Markulf Kohlweiss, Anna Lysyanskaya, and Mira Meyerovich. How to win the clonewars: Efficient periodic n-times anonymous authentication. CCS '06, pages 201–210, New York, NY, USA, 2006. ACM.

[34] M. Z. Lee, A. M. Dunn, J. Katz, B. Waters, and E. Witchel. Anon-pass: Practical anonymous subscriptions. *IEEE Security Privacy*, 12(3):20–27, May 2014.

[35] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. volume 2656 of *EUROCRYPT '03*, pages 614–629. Springer, 2003.

[36] Rahaman Sazzadur, Cheng Long, Yao Danfeng Daphne, Li He, and Park Jung-Min Jerry. Provably secure anonymous-yet-accountable crowdsensing with scalable sublinear revocation. PETS-2017, pages 384–403, 2017.

[37] Maxim Raya and Jean-Pierre Hubaux. The security of vehicular ad hoc networks. SASN '05, pages 11–21, New York, NY, USA, 2005. ACM.

[38] X. Liu, Z. Fang, and L. Shi. Securing vehicular ad hoc networks. In *2007 2nd International Conference on Pervasive Computing and Applications*, pages 424–429, July 2007.

[39] Yanchao Zhang, Wei Liu, and Wenjing Lou. Anonymous communications in mobile ad hoc networks. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 3, pages 1940–1951 vol. 3, March 2005.

[40] Pim Vullers and Gergely Alpár. Efficient selective disclosure on smart cards using idemix. In Simone Fischer-Hübner, Elisabeth de Leeuw, and Chris Mitchell, editors, *Policies and Research in Identity Management*, pages 53–67, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[41] Kai Rannenberg, Jan Camenisch, and Ahmad Sabouri. *Attribute-based Credentials for Trust: Identity in the Information Society*. Springer Publishing Company, Incorporated, 2014.

[42] M. Karchmer and A. Wigderson. On span programs. In *[1993] Proceedings of the Eigth Annual Structure in Complexity Theory Conference*, pages 102–111, May 1993.

[43] Allison Lewko and Brent Waters. Decentralizing attribute-based encryption. EUROCRYPT'11, pages 568–588, Berlin, Heidelberg, 2011. Springer-Verlag.

[44] Zhen Liu, Zhenfu Cao, and Duncan S. Wong. Efficient generation of linear secret sharing scheme matrices from threshold access trees. Cryptology ePrint Archive, Report 2010/374, 2010.

[45] A. K. Mishra, A. K. Tripathy, D. Puthal, and L. T. Yang. Analytical model for sybil attack phases in internet of things. *IEEE Internet of Things Journal*, 6(1):379–387, 2019.

[46] Dan Boneh and Matt Franklin. Identity-based encryption from the Weil pairing. *SIAM J. of Computing*, 32(3):586–615, 2003. extended abstract in Crypto'01.

[47] Shigeo Mitsunari, R Sakai, and M Kasahara. A new traitor tracing. volume E85-A, pages pp. 481–484. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 02 2002.

[48] Keith M. Martin, G.J. Simmons, and W.-A. Jackson. The geometry of shared secret schemes. *Bulletin of the ICA*, 1:71–88, 1991.

[49] Amos Beimel. Secret-sharing schemes: A survey. IWCC'11, pages 11–46, Berlin, Heidelberg, 2011. Springer-Verlag.

[50] D. R. Stinson. An explication of secret sharing schemes. *Des. Codes Cryptography*, 2(4):357–390, December 1992.

[51] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, November 1979.

[52] Jaume Martí-Farré and Carles Padró. Secret sharing schemes on access structures with intersection number equal to one. *Discrete Appl. Math.*, 154(3):552–563, March 2006.

[53] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. CRYPTO '94, pages 174–187, London, UK, UK, 1994. Springer-Verlag.

[54] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *2007 IEEE Symposium on Security and Privacy (SP '07)*, pages 321–334, May 2007.

[55] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. CCS '06, pages 89–98, New York, NY, USA, 2006. ACM.

[56] J. Benaloh and J. Leichter. Generalized secret sharing and monotone functions. CRYPTO '88, pages 27–35, Berlin, Heidelberg, 1990. Springer-Verlag.

[57] Yong Zhang and Jun-Liang Chen. A delegation solution for universal identity management in soa. *Services Computing, IEEE Transactions on*, 4(1):70–81, Jan 2011.

[58] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *JOURNAL OF CRYPTOLOGY*, 13:361–396, 2000.

[59] J. Camenisch. Group signature schemes and payment systems based on the discrete logarithm problem. ETH series in information security and cryptography. Hartung-Gorre-Verlag, 1998.

[60] Angelo De Caro and Vincenzo Iovino. jpbc: Java pairing based cryptography. In *ISCC 2011*, pages 850–855, Kerkyra, Corfu, Greece, June 28 - July 1, 2011. IEEE.

[61] Ben Lynn. *On the Implementation of pairing based cryptosystems*. PhD Thesis, Stanford University, 2007.

[62] A. Hayes. Network service authentication timing attacks. *IEEE Security Privacy*, 11(2):80–82, 2013.

[63] M. Cagalj, T. Perkovic, and M. Bugaric. Timing attacks on cognitive authentication schemes. *IEEE Transactions on Information Forensics and Security*, 10(3):584–596, 2015.

[64] Javier Herranz, Fabien Laguillaumie, Benoît Libert, and Carla Ràfols. Short attribute-based signatures for threshold predicates. In Orr Dunkelman, editor, *Topics in Cryptology - CT-RSA 2012 - The Cryptographers' Track at the RSA Conference 2012, San Francisco, CA, USA, February 27 - March 2, 2012. Proceedings*, volume 7178 of *Lecture Notes in Computer Science*, pages 51–67. Springer, 2012.

[65] Y. Lian, L. Xu, and X. Huang. Attribute-based signatures with efficient revocation. In *2013 5th International Conference on Intelligent Networking and Collaborative Systems*, pages 573–577, 2013.

[66] Qianqian Su, Rui Zhang, Rui Xue, and Pengchao Li. Revocable attribute-based signature for blockchain-based healthcare system. *IEEE Access*, 8:127884–127896, 2020.

[67] H. Xiong, Y. Bao, X. Nie, and Y. I. Asoor. Server-aided attribute-based signature supporting expressive access structures for industrial internet of things. *IEEE Transactions on Industrial Informatics*, 16(2):1013–1023, 2020.

[68] J. Sun, Y. Su, J. Qin, J. Hu, and J. Ma. Outsourced decentralized multi-authority attribute based signature and its application in iot. *IEEE Transactions on Cloud Computing*, pages 1–1, 2019.

[69] J. M. De Fuentes, L. González-Manzano, J. Serna-Olvera, and F. Veseli. Assessment of attribute-based credentials for privacy-preserving road traffic services in smart cities. *Personal Ubiquitous Comput.*, 21(5):869–891, oct 2017.

[70] J. Maria de Fuentes, L. Gonzalez-Manzano, A. Solanas, and F. Veseli. Attribute-based credentials for privacy-aware smart health services in iot-based smart cities. *Computer*, 51(7):44–53, July 2018.

[71] K. Papadamou, S. Zannettou, B. Chifor, S. Teican, G. Gugulea, A. Caponi, A. Recupero, C. Pisa, G. Bianchi, S. Gevers, C. Xenakis, and M. Sirivianos. Killing the password and preserving privacy with device-centric and attribute-based authentication. *IEEE Transactions on Information Forensics and Security*, 15:2183–2193, December 2020.

[72] J. Han, L. Chen, S. Schneider, H. Treharne, and S. Wesemeyer. Privacy-preserving electronic ticket scheme with attribute-based credentials. *IEEE Transactions on Dependable and Secure Computing*, pages 1–1, 2019.

[73] Jan Camenisch, Rafik Chaabouni, and abhi shelat. Efficient protocols for set membership and range proofs. In Josef Pieprzyk, editor, *Advances in Cryptology - ASIACRYPT 2008*, pages 234–252, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[74] Lea Kissner and Down Ssong. Private and threshold set-intersection. *In Advances in Cryptology - CRYPTO 2005*.

[75] Hao Chen, Kim Laine, and Peter Rindal. Fast private set intersection from homomorphic encryption. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 1243–1255. ACM, 2017.

[76] Nuttapong Attrapadung, Benoît Libert, and Elie de Panafieu. Expressive key-policy attribute-based encryption with constant-size ciphertexts. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *Public Key Cryptography – PKC 2011*, pages 90–108, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[77] Jin Li, Qiong Huang, Xiaofeng Chen, Sherman S. M. Chow, Duncan S. Wong, and Dongqing Xie. Multi-authority ciphertext-policy attribute-based encryption with accountability. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '11, pages 386–390, New York, NY, USA, 2011. Association for Computing Machinery.

[78] K. Liang, L. Fang, W. Susilo, and D. S. Wong. A ciphertext-policy attribute-based proxy re-encryption with chosen-ciphertext security. In *2013 5th International Conference on Intelligent Networking and Collaborative Systems*, pages 552–559, 2013.

[79] Y. Zhang, D. Zheng, and R. H. Deng. Security and privacy in smart health: Efficient policy-hiding attribute-based access control. *IEEE Internet of Things Journal*, 5(3):2130–2145, 2018.

[80] Yinghui Zhang, Xiaofeng Chen, Jin Li, Duncan S. Wong, and Hui Li.

Anonymous attribute-based encryption supporting efficient decryption test. *ASIA CCS '13*, 2013.

[81] Junzuo Lai, Robert H. Deng, and Yingjiu Li. Expressive cp-abe with partially hidden access structures. *ASIA CCS '13*, 2012.

[82] Qikun Zhang, Yong Gan, Lu Liu, Xianmin Wang, Xiangyang Luo, and Yuanzhang Li. An authenticated asymmetric group key agreement based on attribute encryption. *Journal of Network and Computer Applications*, 123:1–10, 2018.

[83] Z. Qikun, L. Yongjiao, G. Yong, Z. Chuanyang, L. Xiangyang, and Z. Jun. Group key agreement protocol based on privacy protection and attribute authentication. *IEEE Access*, 7:87085–87096, 2019.

[84] C. Xu, H. Guo, Z. Li, and Y. Mu. Affiliation-hiding authenticated asymmetric group key agreement based on short signature. *The Computer Journal*, 57(10):1580–1590, 2014.

[85] X. Yao, H. Kong, H. Liu, T. Qiu, and H. Ning. An attribute credential based public key scheme for fog computing in digital manufacturing. *IEEE Transactions on Industrial Informatics*, 15(4):2297–2307, 2019.

XV. Biographies

Victor Sucasas obtained his Ph.D. on Electrical and Electronic Engineering at University of Surrey (UK) in 2016. He has extensive research experience as a researcher at Instituto de Telecomunicações - Aveiro, Portugal and at University of Surrey, Guildford, UK, where he worked on European projects FP7-GREENET, ECSEL-SWARMs, CATRENE-H2O and ECSEL-SECREDAS. In 2016, he became a senior researcher at University of Aveiro in network security and privacy preserving systems. His research interests cover privacy-preserving authentication mechanisms, pseudonymization and anonymization. Currently he is a senior researcher at the crypto centre in the Technology Innovation Institute (TII), Abu Dhabi, EAU. He is a IEEE member and EAI Fellow.

Georgios Mantas received the Ph.D. degree in Electrical and Computer Engineering from the University of Patras, Greece, in 2012 and the M.Sc. degree in Information Networking from Carnegie Mellon University in 2008. In 2014, he became a post-doctoral researcher at the Instituto de Telecomunicações - Aveiro, Portugal, where he has been involved in research projects such as ECSEL-SemI40, CATRENE-MobiTrust, CATRENE-NewP@ss, ARTEMIS-ACCUS, FP7-CODELANCE, and FP7-SEC-SALUS. Since 2018, he has been a Lecturer at the University of Greenwich, UK. His research interests include network and system security, authentication mechanisms, privacy-preserving mechanisms, intrusion detection systems, and secure network coding.

Maria Papaioannou received the degree in electrical and computer engineering in 2016, and the Postgraduate Specialization in biomedical engineering from the University of Patras in 2018. Currently, she is pursuing the PhD degree in electrical and computer engineering with the University of Patras. Since 2018, she has been a member of the 4TELL Research Group, Instituto de Telecomunicações, Aveiro, Portugal, and she has been working in the European research project POCI-01-0247-FEDER-024539 5G Mobilizador, focused on privacy-preserving user authentication protocol for Smart City applications. Her research interests include cryptography, authentication and access control mechanisms, and network and system security.

Jonathan Rodriguez received the M.Sc. degree in electronic and electrical engineering and the Ph.D. degree both from the University of Surrey, U.K., in 1998 and 2004, respectively. He is author of more than 450 scientific works, including eight book titles. In 2005, he became a Researcher at the Instituto de Telecomunicações, Aveiro, Portugal, and in 2008, a Senior Researcher establishing the 4TELL Research Group. He has served as a Project Coordinator for major international research projects, including Eureka-LOOP,FP7-C2POWER, and H2020-SECRET and as a Technical Manager for FP7-COGEU and FP7-SALUS. He is a IEEE Senior member, Chartered Engineer and IET Fellow, and since 2017, he has been a Professor in Mobile Communications at the University of South Wales, Pontypridd, U.K.