A Signature Scheme with Unlinkable-yet-Acountable Pseudonymity for Privacy-Preserving Crowdsensing

Victor Sucasas, IEEE Member; Georgios Mantas, IEEE Member;
Joaquim Bastos; Francisco Damião; Jonathan Rodriguez, IEEE Senior Member

***Abstract*—Crowdsensing requires scalable privacy-preserving authentication that allows users to send anonymously sensing reports, while enabling eventual anonymity revocation in case of user misbehavior. Previous research efforts already provide efficient mechanisms that enable conditional privacy through pseudonym systems, either based on Public Key Infrastructure (PKI) or Group Signature (GS) schemes. However, previous schemes do not enable users to self-generate an unlimited number of pseudonyms per user to enable users to participate in diverse sensing tasks simultaneously, while preventing the users from participating in the same task under different pseudonyms, which is referred to as sybil attack. This paper addresses this issue by providing a scalable privacy-preserving authentication solution for crowdsensing, based on a novel pseudonym-based signature scheme that enables unlinkable-yet-accountable pseudonymity. The paper provides a detailed description of the proposed scheme, the security analysis, the performance evaluation, and details of how it is implemented and integrated into a real crowdsensing platform.**[1]

I. Introduction

Context awareness and environmental information gathering is a cornerstone component in Smart Cities, since it facilitates the intelligent management of urban subsystems, responsible for energy, water, public lighting, transportation, or environmental control [1],[2],[3]. In a standard deployment, this environmental data is obtained through pre-installed data collection stations provisioned with dedicated sensing systems. However, mobile crowdsensing shifts this anchored data collection paradigm towards a mobile user-centric distributed sensing network [4],[5]. Information is gathered and transmitted directly by users' handsets, hence allowing the management systems in the Smart City to use measurements taken at the ideal locations, which is where the citizens actually are. On the citizens' side, user participation can be fostered through incentive programmes, where users are rewarded for their contribution to the crowdsensing tasks [6].

Crowdsensing caters for outstanding benefits in terms of increased data accuracy and decreased deployment cost, but it also brings a security challenge in terms of trust between the Smart City and crowdsensing participants. From the Smart City perspective the security concern is twofold. Firstly, sensors are distributed and carried be citizens who could be malicious users sensing fake data or honest users with defective sensors. Secondly, citizens are rewarded for contributing to the sensing tasks, and hence selfish users could submit their reports several times to increase their profits. From the citizens' perspective the main concern is the location privacy, since crowdsensing participants submit sensing data together with geolocation information [7]. Thus the Smart City can identify and locate citizens, which can discourage citizens from participating in the sensing tasks [8],[9],[10].

Previous works have addressed these challenges by proposing conditional privacy-preserving authentication mechanisms based on pseudonym systems. Citizens submit authenticated sensing reports using pseudonyms, which preserves citizens' privacy, while enabling the Smart City to verify the sender's legitimacy to participate in the sensing task. The Smart City can trigger the revocation of the users' privacy privileges in case of misuse, due to sensor malfunctioning or user misconduct, and block the participation of dishonest users in future sensing tasks. In cases of severe misbehaviour, the Smart City may also demand user de-anonymization and prosecution. Also, selfish users can be avoided by controlling the pseudonym generation process, to ensure that only one pseudonym per user is allowed in each sensing task. On the citizens side, pseudonyms from the same user are unlinkable to each other, and renewed periodically or on a sensing task basis, and hence they cannot be linked to the user's identity through statistical analysis [11]. However, current pseudonym systems proposed for crowdsensing applications, mainly based on Group Signature (GS) schemes [12],[13] or Public Key Infrastructure (PKI) [14],[6], do not provide a scalable mechanism to generate several unlinkable pseudonyms on the user side while preventing selfish users from participating in the same task several times.

More concretely, PKI-based systems do not enable pseudonym self-generation on the user side. Citizens must contact a Certification Authority (CA) periodically to obtain and renew their pseudonyms [14]. Also, to prevent selfish users a new pseudonym should be issued for each sensing task [6], which might cause scalability problems in large deployments. GS-based systems enable pseudonym self-generation on the user side with a single credential issued from the CA, which is more scalable than PKI-based solutions. However, GS-based systems are limited to one pseudonym per user that is renewed periodically in a time-slot basis [13],[15]. Thus, citizens cannot hold several valid pseudonyms simultaneously. Although it is possible to provide several credentials per user to enable users to hold several unlinkable pseudonyms simultaneously [16], it

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

F. Damiaão is with BeyondVision LDA, Portugal (email: francisco.damiao@pdmfc.com).

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

J. Rodriguez is with the Mobile and Satellite Communications Research Group, School of Engineering, Faculty of computing, Engineering and Science, University of South Wales, (email: jonathan.rodriguez@southwales.ac.uk).

would not be possible to detect a selfish user participating in a single sensing task with several pseudonyms. Other pseudonym systems, based on Identity Based Cryptography (IBC), enable the self-generation of several pseudonyms on the user side with a single credential [17], but they also do not prevent selfish users. To the best of the authors' knowledge, there are no mechanisms in the state of the art that enable the unlimited self-generation of unlinkable pseudonyms on the user side while preventing selfish users from using more than one pseudonym in the same task. It is worth mentioning that previous privacy-preserving systems, providing anonymity revocation, could link pseudonyms to the same credential and detect selfish users. However, this would require to revoke pseudonyms of all users, including honest users. This paper provides an extensive review of current pseudonym systems and crowdsensing privacy-preserving frameworks, in sections II and IV, and describes the shortcomings for their application in crowdsensing scenarios.

To address this security challenge, we introduce the concept of unlinkable-yet-accountable pseudonymity. According to this concept, users should be able to generate several pseudonyms to participate, simultaneously, in several sensing tasks. Different pseudonyms generated from one credential cannot be linked to the same credential, and different pseudonyms generated from one credential cannot be used for the same task. This means that, from the system perspective, given a set of pseudonyms it is not possible to know which pseudonyms belong to the same credential, i.e. user. However, given a set of pseudonyms used in a specific sensing task, it is certain that each pseudonym belongs to a different credential, i.e. user. Therefore, the unlinkable-yet-accountable feature enables users to participate in several tasks simultaneously without these tasks being linked to each other. Also it enables the Smart City to be sure that users will not be able to participate in the same sensing task with two or more different pseudonyms.

This paper proposes a novel pseudonym-based signature scheme, based on IBC, with unlinkable-yet-accountable pseudonymity. The proposed scheme is resilient to attack scenarios where selfish and dishonest users are present. The proposed scheme is resilient to the following attack scenarios: i) honest users with malfunctioning sensors transmitting faulty data; ii) selfish users sending the same sensing report multiple times to increase their rewards (which is known as sybil attack [12],[18]); and iii) malicious users transmitting fake data to mislead the sensing process (this attack can be leveraged with a sybil attack). In the proposed scheme, the unlinkable-yet-accountable feature is not achieved by means of de-anonymizing or revoking users pseudonyms, hence it preserves the privacy privileges of honest users. Additionally, the proposed scheme also provides an efficient implementation of an anonymity revocation mechanism, which is an essential component in crowdsensing applications, and one of the current bottlenecks for the deployment of pseudonym-based systems in real crowdsensing scenarios. The paper provides: i) an extensive review on the state of the art; ii) detailed construction of the proposed scheme; iii) the proof of security of the proposed scheme; iv) complexity comparison between the proposed scheme and previous schemes; v) performance evaluation; and vi) shows how it is integrated into a real crowdsensing system and implemented in a real Smart City pilot in the city of Lisbon, Portugal.

II. Related work on pseudonym systems

The notion of identity confidentiality was described by Pfitzmann and Hansen [19] in their definition of anonymity. In such definition, an entity belonging to 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 a short, medium or even long term. Moreover, fully anonymous systems can be jeopardized by misusers that abuse their anonymous state to break the service terms and conditions, hence the users' privacy rights should be conditional to the users' behavior.

Pseudonyms provide a suitable means to keep the users' sessions, since they can serve as identifiers for entities while still preserving the anonymous state. Entities can hold many pseudonyms representing the entity, its roles or functions, or the different relations of the entity with different organizations [20],[21]. 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. Also, a user can use many pseudonyms and perform a cognitive procedure to drive the pseudonym switching mechanism, which prevents pseudonym tracking through statistical analysis [11]. 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 [16],[22]. However, pseudonym systems can be implemented in multiple manners and provide a diverse set of features:

A. Pseudonymity through digital anonymous credentials

Digital anonymous credential systems [23],[24] were proposed to enable anonymous systems. In such systems, users are granted anonymous credentials, which are composed of a blind signature from a Certification Authority (CA) on a committed value and an efficient mechanism for the credential holder to prove the knowledge of such signature. The unlinkability feature of the proof of knowledge of such signature (also called signature of knowledge) allows credential holders to use the credential multiple times towards a verifier without being tracked and without revealing any other information than the possession of such credential. There are mechanisms to create pseudonyms between users and organizations and to link such pseudonyms to the credential issued by the CA [21]. Also, revocation mechanisms can be enabled in digital anonymous credential systems [25]. However, in these works, pseudonyms represent relations between organizations and users, hence the creation of a pseudonym is a two party protocol that involves linking a previous pseudonym with another organization to the new one (also called transferring the credential). It is worth mentioning the works [26],[15], which provide a construction

where the user proves for possession of an anonymous credential and links this credential to a valid token that can be seen as a pseudonym. The scheme provides a time-based mechanism where the tokens are renewed periodically. Although users cannot choose to maintain the tokens, i.e. they are forced to switch tokens periodically, a new token can be linked to a previous one, hence it allows users to maintain anonymous user sessions active. The downside of this approach for its application in the crowdsensing scenario is that it does not provide a revocation mechanism to support conditional privacy, and it does not enable more than one pseudonym per user and time-slot.

B. Pseudonymity through group signatures

Group signature (GS) schemes, similar to digital anonymous credentials, allow users to remain unidentifiable under the anonymity set formed by the group members. Users in possession of a group signing key can sign messages on behalf of a group [27],[23], and these messages can be verified with a group public key. GS schemes enable conditional privacy since group members can include their encrypted unique public keys or identities along with the signatures, which can be later opened by a revocation manager. Also, GS-based systems may integrate pseudonyms in the form of revocation tokens [28],[13]. The main drawback of group signature schemes is their complexity, mainly in the anonymity revocation process which is the main bottleneck of this technology [29]. It is worth mentioning the work in [13], where the authors propose a GS scheme that includes embedded pseudonyms as revocation tokens for a crowdsensing application. This work also caters for a revocation mechanism whose complexity is sublinear with respect to the number of revoked users, hence more efficient than other GS-based counterparts. This work however, like other GS-based schemes, only enables users to hold one pseudonym at a time, and all signatures produced with the same pseudonym are inherently linkable. Intuitively, it would be possible to grant several credentials to each citizen [16] to generate several unlinkable pseudonyms simultaneously, but a user holding those credentials would be able to behave like different users, hence the unlinkable-yet-accountable property would not be provided.

C. Pseudonymity through public keys and symmetric keys

A Pubic Key Infrastructure (PKI) solution is simpler in terms of revocation efficiency, and more convenient than previously referenced solutions in some scenarios such as in Vehicular Networks (VANETs). Some research works have proposed that users can send signed messages using public key cryptography, with the public keys representing their pseudonyms [14],[30]. Some previous works on crowdsensing follow this approach [6]. However, this approach requires a certification authority to generate and distribute, and periodically renew, a large number of public key certificates. These certificates have also to be transmitted together with the signed messages. A more lightweight solution is proposed in [31],[32], which suggests a pseudonym-based authentication scheme based on one-time passwords, where pseudonyms are generated with a hash chain. Nonetheless, this approach requires a two party protocol for pseudonym generation in both sides (the citizen and the Smart City in our scenario), which must hold the same secret key to derive the pseudonym set in both sides. It also requires a periodic renewal of the pseudonym set.

D. Pseudonymity through identity-based cryptography

Identity based cryptography (IBC) [33] considers the user's identity as the public key, hence users can sign messages that can be verified with their identities. Note that, as proposed by [34], if the identity of the user is replaced by a pseudonym, then the user can sign messages that can be verified with that pseudonym. In this scenario, conditional privacy (i.e. anonymity revocation in case of misbehavior) is still possible when the CA issues both the secret keys and the pseudonyms [35], or at least generates the secret key associated to the pseudonyms [17]. It is also possible to generate both, the secret key and pseudonyms in the user side, but such feature must go together with an efficient manner to control the users' pseudonym generation capabilities and to enable the revocation in case of misuse. A possibility to control the pseudonym self-generation is to require the pseudonyms to be signed by a trusted third party [36], e.g. the CA. This signing process by a CA can also be done blindly [37], which enables users' privacy with respect to the CA. However, the requirement for the interaction with a trusted third party to generate or certify the pseudonyms can limit the scalability and the efficiency of the system. Group signatures or ring signatures can be used to address this scalability issue [38],[39], since the users belonging to a group of legitimate users can generate their own pseudonyms and corresponding private keys and sign these pseudonyms with the group signing key. However, this approach does not provide the desired property of unlinkable-yet-accountabile pseudonymity. In general, the main limitation for the application of current pseudonym systems based on IBC in our crowdsensing scenario is that the schemes enabling unlinkable pseudonym self-generation, e.g. [17],[38],[39], do not prevent users from using several pseudonyms in the same sensing task.

III. System requirements for privacy-preserving authentication

We have exposed several technologies that can be used to implement a pseudonym-based signature scheme to enable privacy-preserving authentication in crowdsensing. However, to be applied in a crowdsensing scenario, the adopted pseudonym-based signature scheme should enable the following features:

- Pseudonym self-generation: Users should be able to generate an unlimited number of pseudonyms on the user side with a single credential issued by the CA. This feature limits the number of connections to the CA which makes the crowdsensing system scalable.
- Efficient revocation: Given a pseudonym, the CA should be able to compute the user pseudonym set efficiently and to retrieve her identity. The system should also rely on a revocation list that grows linearly with the number of revoked users, or implement a revocation mechanism that is tolerant to possible delays in the revocation list distribution [40],[13].
- Backward unlinkability: The revocation should provide backward unlinkability (also referred as forward

unlinkability in some works [35]), i.e. a revoked pseudonym should not be linkable to previous non-revoked pseudonyms of the same user.

- All-or-nothing transferability: A legitimate user should not be able to transfer a pseudonym to another non-legitimate user and enable the recipient of the pseudonym to generate valid signatures without enabling her to generate more pseudonyms and signatures under the legitimate users' behalf.
- Unlinkability-yet-accountability: All pseudonyms generated with the same credential and used for different sensing tasks should not be linkable to each other. However, all pseudonyms used for a specific sensing task should be generated from different credentials. It is worth highlighting that pseudonym revocation should not be required, i.e. the CA should not be involved to provide unlikability-yet-accountability in the Smart City side.

IV. Related work on privacy-preserving crowdsensing frameworks

The first proposed framework for privacy-preserving crowdsensing, to the best of the authors knowledge, was Anonysense [12] which already emphasizes the need for privacy preservation in both the network and application layers. For the network layer the authors propose to obfuscate the MAC and IP addresses with mechanisms such as [41],[42]. In the application layer, they adopt a privacy-preserving authentication mechanism that relies on a GS scheme, concretely the scheme in [43]. However, this scheme does not provide any concrete anonymity revocation mechanism and should only be used in systems where unconditional privacy is desired. Also, it does not integrate pseudonyms in the signatures, which yields signatures from users unlinkable, hence it does not prevent selfish users. As a result, this system is only effective in a scenario that considers honest users.

Sppear [6] provides a system model that splits the crowdsensing architecture in several entities, and achieves privacy even when several *honest-but-curious* entities within this architecture collude against the users' privacy. Privacy-preserving authentication is achieved through a GS scheme and PKI together. First a Group Manager (GM) provides a group signing key (gsk) to legitimate users. Then users can use this gsk to authenticate themselves towards an Identity Provider (IdP), which enables a Pseudonym Certification Authority (PCA) to provide pseudonyms to users in the form of public keys, with their corresponding certificate and secret key. However, this solution requires users to authenticate themselves towards the IdP, using the gsk, to obtain new pseudonyms from the PCA to perform unlinkable sensing tasks. Hence, pseudonym self-generation is not provided. All-or-nothing transferability is also not accomplished, since users can demand pseudonyms to the PCA and transfer them to not legitimate users.

Groupsense [13] proposes a novel GS scheme, called SRBE, that enables pseudonym self-generation in the user side with a single credential from the CA. The scheme proposes a slotted time-division where users can self-generate a unique and unlinkable pseudonym in each slot. SRBE also allows revocation with a complexity that is sublinear with respect to the number of revoked pseudonyms. This system provides all mentioned features in sec. III, except for the last one, unlinkable-yet-accountable pseudonymity. SRBE does not enable users to hold several unlinkable pseudonyms per time-slot, which is actually an open problem in GS schemes [13]. Although it is possible to grant users several credentials, hence several pseudonyms per time-slot, then a selfish user could participate in the same sensing task with several pseudonyms.

TABLE I. Privacy-Preserving Frameworks

Framework / scheme	Anonyense [6]	Sppear [6]	GroupSense [13]	PS [17]	Our Scheme
self-generation	NA	-	√ 1 pseu per credential	√ unlimited	√ unlimited
anonymity / pseudonymity revocation	-	√	√	√	√
backward unlinkability	NA	√	√	√	√
all-or-nothing transferability	√	-	√	√	√
unlinkable-yet-accountable	-	√	-	-	√

The work in [17] also provides a pseudonym-based signature scheme, based on IBC, for service oriented architectures that is applicable to crowdsensing. It provides a mechanism for users to self-generate an unlimited number of pseudonyms from a single credential issued by the CA. The CA can still track the self-generated pseudonyms and link them to the stored credential of misbehaving users. However, this scheme does not provide any control or limitation on the pseudonym generation, which would enable sybil attacks from selfish users. Also, revocation requires an on-demand search on the users' credentials with three pairing operations per stored credential, which increases the time complexity of the revocation mechanism. In the same work [17], authors propose another signature scheme with pseudonym controlled variation, which limits the pseudonym self-generation to one pseudonym per time-slot. This scheme enables the pre-computation of the users' pseudonyms in the CA, which enables a fast revocation mechanism. However, this second scheme does not provide pseudonym unlinkability.

It is also worth mentioning some valuable proposals based on homomorphic encryption for privacy-preservation in crowdsensing such as [44],[45],[46],[47],[48]. These works provide elegant and efficient manners to aggregate data and calculate reliability degrees from users while preserving users' privacy. Also the work in [49], which provides a scheme that enables complex arithmetic operations for data aggregation and processing of encrypted sensing data. However, contrary to our proposed scenario, these works adopt the *honest-but-curious* user threat model, in which users may try to infer other users' information but do not have any motivation to send faulty data to the data processing services, i.e. users do not disrupt the proper functioning of the Smart City data processing mechanism. Although it is a reasonable assumption, the consideration of rogue users or malfunctioning sensing devices should not be left unconsidered due to the relevance that such data may have in the management of some Smart City urban subsystems. In this scenario, when user misbehavior (intentionally or not) is considered, the digital signing of sensed data is key [44],[50].

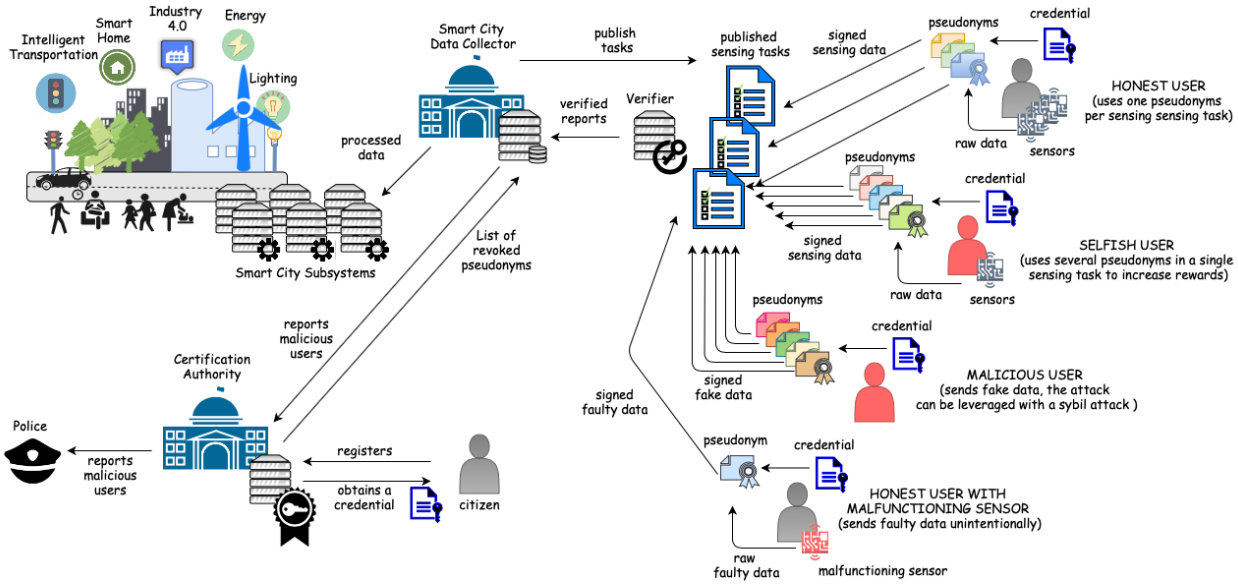


Fig. 1. System model for the proposed pseudonym-based signature scheme.

V. System Model

The proposed crowdsensing scenario considers the following entities, also detailed in Fig. 1:

1) **The Certification Authority (CA)**: is in charge of: i) issuing credentials to users; ii) revoke the user privacy rights, i.e. it can retrieve the real identity of a user given her pseudonym; iii) distributing a revocation list with all revoked pseudonyms.
2) **The Verifier**: is an Authentication Server (AS) in the Smart City side. It is responsible for: i) authenticating the sensing reports sent by users; ii) delivering the authenticated sensing data to a data processing service; and iii) issuing receipts to the users to acknowledge their participation in the crowdsensing system.
3) **The Data Collector**: also referred to as data processing service that receives authenticated sensing data from the Verifier. It is responsible for: i) identifying outliers or misleading sensing data and report it to the CA to eventually trigger the revocation process; and ii) planning the sensing tasks, i.e. it publishes the required environmental data and the incentive program to foster user participation. For each sensing task, the data collector enables a specific task index, hence users can participate with their unique pseudonym of such index value.
4) **The users**: are citizens provided with a valid credential issued by the CA and they can: i) self-generate pseudonyms; and ii) sign the sensing data. The user can self-generate a new pseudonym per each sensing task by using the index value of such sensing task. Users can participate in several sensing tasks simultaneously, i.e. providing data from different sensors.

It is worth commenting that we leave the incentive program implementation out of the scope of the paper, and focus on the privacy-preserving authentication aspect of the sensing data delivery. Previous works such as [6] detail how this process can be performed securely.

VI. Threat Model

The threat model, also depicted in Fig. 1, considers three different entities: i) the CA; ii) The Smart City including the data collector and the verifier; and iii) the users (i.e. citizens). The CA is an *honest* entity and independent from other Smart City entities. Thus, the CA is legitimate to revoke users privacy privileges in case of misuse, either by pseudonym revocation or de-anonimization. The verifier and the data collector are considered *honest-but-curious* entities, which means that they may use the sensing reports and pseudonyms to locate and profile specific users by linking the transmission of several reports and GPS information to infer users' activities. Finally, users are considered *dishonest* and can perform several misbehaving actions: i) send faulty data unintentionally due to a sensor malfunctioning; ii) send fake data intentionally to disrupt service; or iii) send the same valid report several times to maximize the benefit from the incentive program, i.e. using two or more different pseudonyms for the same task (which is considered a sybil attack).

VII. Main Contribution of Proposed Work

The main contribution of this work is a pseudonym-based signature scheme that enables users to self-generate unlimited unlinkable-yet-accountable pseudonyms. The users can generate a number of unlinkable pseudonyms to participate simultaneously in several sensing tasks. Different pseudonyms generated from a user cannot be linked to the same user, but different pseudonyms from the same user cannot be used for the same sensing task. This feature permits the Smart City to control the number of participants in a sensing task and protects the incentive system from selfish users. We achieve this feature by embedding an index value in the pseudonym generation and signing algorithms of the signature scheme, which enables users to generate only one valid pseudonym per index value. The correct construction of the pseudonym and the signature can be validated by the verifier in the Smart City side without any interaction with the CA.

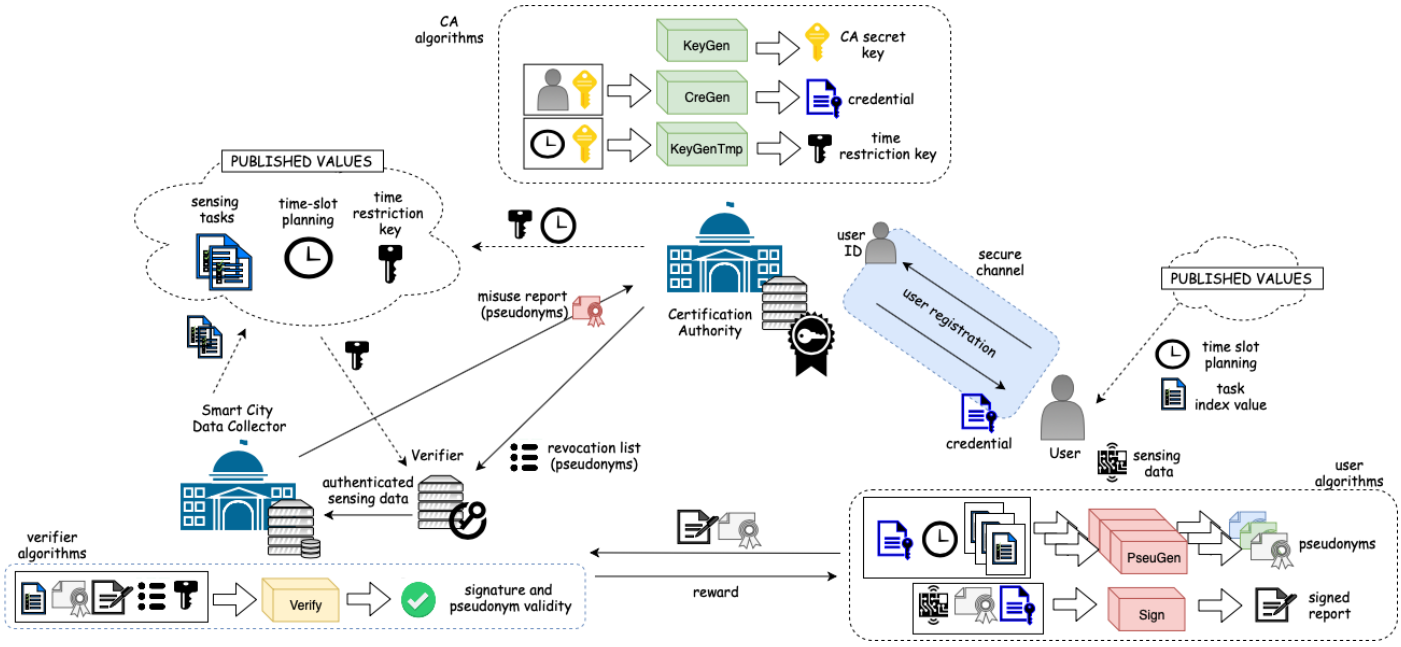


Fig. 2. Algorithms of the proposed pseudonym-based signature scheme.

The security of the proposed signature scheme relies on the $k - CAA$ problem (sec. VIII-C3) proposed by [51], that has also been followed by other signature schemes such as [52],[53],[54],[55],[56],[57],[17]. The latter, [17], also provides a pseudonym-based signature scheme with the flexibility to generate unlimited pseudonyms on the user side, as explained in sec. IV. In that construction, each pseudonym is generated using a credential issued by the CA and a secret key randomly chosen by the user.

In our proposed scheme however, the generated key on the user side must fulfil a linear relation with the secret key obtained from the CA and a random factor obtained through a hash-chain. Such a hash-chain uses the index value of each sensing task, which binds the pseudonym generation process to each sensing task. The proposed signature scheme is constructed from a zero-knowledge proof using the Fiat-Shamir heuristic VIII-C, and the linear relationship between secret keys is included in the proof of knowledge by means of auxiliary keys (sec. VIII-D).

Additionally, the proposed scheme provides an efficient mechanism to revoke pseudonyms that is based on pseudonym pre-computation. This mechanism allows the CA to pre-compute users pseudonyms (which does not require pairing computations) to obtain a pseudonym list, and then perform revocation with a binary search tree. It is worth highlighting that pseudonym pre-computation would not be possible if users were able to generate pseudonyms using randomly generated keys [17]. Moreover, the paper provides a batch verification mechanism, which is convenient in a scenario where the signature verification process is centralised. The paper caters for a detailed performance evaluation section, a formal security analysis and complexity analysis that shows a lower number of pairing computations when compared with previous pseudonym-based schemes proposed for crowdsensing.

VIII. Proposed Signature Scheme

This section describes the operations of the proposed pseudonym-based signature scheme, provides the mathematical preliminaries to understand the construction and the proofs of security, and then provides the security definition and security analysis.

A. Signature scheme algorithms

The algorithms are performed by three different kinds of entities: i) Certification Authority (CA); ii) Verifier; and iii) User. Also, the time is divided in time-slots, and the switching between time-slots is synchronized in all entities. The signature scheme is composed of the following algorithms, also depicted in Fig. 2:

CA algorithms:

- **KeyGen**: The CA runs this algorithm taking as input a security parameter K. The output is a set of *public* parameters PP and *secret* key s.
- **KeyGenTmp**: The CA runs this algorithm taking as input a time-slot value i. The output is a *public* time-restriction key W_i.
- **CreGen**: The CA runs this algorithm to generate a credential $cred$ for a user with identity ID. The CA stores the tuple $(ID, cred)$ in a registry REG. The user stores the credential secretly.

User algorithms:

- **PseuGen**: The user runs this algorithm, taking as input the credential $cred$, the index of the pseudonym x, and the time-slot value i. The index is provided by the data collector when publishing a new sensing task, and it is a public value. The output is a pseudonym $pseu_{x,i}$ and

a secret value μ'. The CA can also run this algorithm since it keeps the credentials in the registry REG

- **Sign**: The user runs this algorithm to sign an arbitrary message M, with a pseudonym $pseu_{x,i}$, for the index x and the credential $cred$ in a time-slot i. The output is the signature σ.

Verifier algorithms:

- **Verify**: The Verifier runs this algorithm. It performs the following two sub-algorithms, and it outputs "valid" if and only if both sub-algorithms output "valid":
  - **RevCheck**: The algorithms returns "invalid" if RL constains the pseudonym $pseu$. It outputs "valid" otherwise.
  - **SignCheck**: Checks whether the signature σ is correct for the message M, with respect to the pseudonym $pseu_{x,i}$ for the time-slot i and index x.

B. Specific CA operations for conditional privacy

Additionally, the CA can perform the following operations to enable conditional privacy, i.e. to be able to revoke users' privacy privileges and track misbehaving users, which consists of including some or all pseudonyms of the revoked user in the Revocation List (RL). Such RL is transmitted to the Verifier:

- **GenPseuList**(REG, i,$\{x_1, \ldots, x_n\}$): The algorithm is executed before the activation of the time-slot i. It generates the pseudonym list $PseuL$ with all valid pseudonyms of registered users for time-slot i and index values $\{x_1, \ldots, x_n\}$, where n is the maximum number of pseudonyms per user, i.e. the maximum number of planned sensing tasks.
- **Open**($pseu$, i): If the pseudonym $pseu$ is part of a valid signature, then the algorithm returns the ID of the user owning the pseudonym $pseu$.
- **Revoke**($pseu, i, \{x_i, \ldots, x_j\}, RL$): It revokes the user owning the pseudonym $pseu$ by including in the RL the user's pseudonyms with indexes $\{x_i, \ldots, x_j\}$ where $i, \ldots, j \in [1, n]$. The set of indexes must be composed of at least one index, and a maximum of n.

C. 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 K be a security parameter that defines the number of bits of p. Then e is a bilinear map [33], in the groups (G_1, G_T), $e : G_1 \times G_1 \rightarrow G_T$[2], 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 $Q = nP$.

3) (k,n)-CAA Problem: Let G_1, G_T be two cyclic groups of prime order and e be a bilinear map defined as above. Let $P, P_1, \ldots, P_n \in G_1$, and let $x, a_1, \ldots, a_k \in Z_p$ (where k and n are integers;). As defined in [17], the Collusion Attack Algorithm problem with k traitors and n examples, $((k, n) - CAA)$, is defined as:

Given the set $\{xP, xP_j | 1 \leq j \leq n\}$, there is not any polynomial time algorithm for obtaining $\frac{1}{(x+a)}P$ for some $aP_j \notin \{a_i P_j | 1 \leq i \leq k; 1 \leq j \leq n\}$ with no negligible probability [51].

D. Pseudonym-based signature scheme construction

This section details the construction of the proposed pseudonym-based signature scheme with unlinkable-yet-accountable pseudonymity by extending the signature scheme in [17]. The table II describes the different elements and the algorithms from which they are obtained. In general, capital letters are used for elements in the group G_1 and low case letters are used for elements in Z_p^* and G_T. Also, elements of the form $f()$ denote public functions. The term PP denotes the set of *Public Parameters*. Bold values are secret values while the rest are public values.

TABLE II. PUBLIC AND PRIVATE ELEMENTS

Element	Derived from	Element	Derived from
PP (P)	KeyGen	PP ($e()$)	KeyGen
PP (W)	KeyGen	PP ($H_1()$)	KeyGen
CA secret key (s)	KeyGen	PP ($H_2()$)	KeyGen
time-rest key W_i	KeyGenTmp	PP ($H_3()$)	KeyGen
G_1, G_T, Z_p^*	KeyGen	PP ($T()$)	KeyGen
pseudonym idx (x)	user choice	time-variant Q_i	$H_1(T(time))$
cred secret (Su)	CreGen	signature (c)	Sign
cred secret (μ)	CreGen	signature ($s1$)	Sign
pseu secret (μ')	PseuGen	signature ($s2$)	Sign
pseudonym (Pu)	PseuGen	signature ($s3$)	Sign
pseudonym ($\tilde{P}u$)	PseuGen	signature ($s4$)	Sign
signature ($y1$)	Sign	signature ($s5$)	Sign
signature ($y2$)	Sign	time-slot (i)	$T(time)$

KeyGen(1^K) For a given security parameter K, the algorithm selects a prime number p of K bits. Then select two cyclic groups of order p, G_1 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 two cryptographic hash functions $H_1, H_2 : \{0,1\}^* \rightarrow G_1$[3] and a hash function $H_3 : \{0,1\}^* \rightarrow G_T$. The algorithm selects a function $T()$ whose input is a $time$ value obtained from the system clock and its output is the corresponding time-slot value i. The algorithm runs at the CA side.

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

[3] As it is explained in [33], it is sufficient with a function $H : \{0,1\}^* \rightarrow A$, for a given set A, and an admissible encoding function $L : A \rightarrow G_1$

The algorithm performs the following operations to generate the public parameters $PP = \{G_1, G_T, Z_p, P, W, g, h, e(), H_1(), H_2(), H_3(), T()\}$ and the secret key s.

1) selects to generators g, $h \xleftarrow{R} G_T$ such that the discrete logarithm with respect to each other is not known.
2) selects a secret $s \xleftarrow{R} Z_p^*$
3) selects a generator $P \xleftarrow{R} G_1$
4) computes a public key $W = sP$

KeyGenTmp(i,PP) The algorithm runs at the CA and gets as input a time-slot value i to generate a public time restriction key W_i, which is transmitted from the CA to the Verifier before the activation of each time-slot. The algorithm performs the following steps:

1) computes the time variant parameter $Q_i = H_1(i)$
2) computes $W_i = sQ_i$

CreGen(s,PP,ID) The algorithm runs at the CA and uses the secret key s and public parameters PP to generate a credential for a user. The CA runs this algorithm and sends the credential $cred = (\mu, Su)$ to the user over a secure channel. The user must authenticate first, and provide its real identity ID. Although the value ID is not used in the credential generation, the CA stores the tuple $(ID, cred)$ in an internal registry REG. The user can verify the correctness of the credential by checking if $e(\mu P + W, Su) = e(P, P)$ holds.

1) selects $\mu \xleftarrow{R} Z_p^*$
2) uses the secret value s and computes $Su = P\frac{1}{(s+\mu)}$
3) the credential is the tuple $cred = (\mu, Su)$

PseuGen($cred$,x,i,PP): The algorithm runs at the user side and requires a valid credential. It generates a pseudonym for the index value x and time-slot i. The algorithm performs the following steps:

1) computes the time-variant parameter $Q_i = H_1(i)$
2) sets $d = H_3^x(T(time))$, where $H^x()$ results on applying $H()$ recursively x times, i.e. a hash-chain.
3) computes $\mu' = (d - \mu)/2 \ (mod\ p)$
4) computes $Pu = (\mu + \mu')Q_i$
5) computes $\hat{P}u = \mu' S_u$
6) the pseudonym is the tuple $pseu_{x,i} = (Pu, \hat{P}u)$
7) the secret key associated with this pseudonym is μ'

Sign($cred$,μ',x,i,$pseu_{x,i}$,m,PP) the algorithm runs at the user side and requires a valid pseudonym (Pu,$\hat{P}u$) with an index value x, time-slot i, and the secret values Su, μ and μ'. It outputs a signature for a given message m of arbitrary length by performing the following steps:

1) computes the time-variant parameter $Q_i = H_1(i)$
2) selects random factors r_1, r_2, r_3, r_4, r_5, γ, $\delta \xleftarrow{R} Z_p^*$.
3) computes $T_{G_1} = r_1 Q_i$
4) computes $t_2 = [e(Q_i, P + \hat{P}u)]^{r_2}$.
5) computes auxiliary public keys $\tilde{y}_1 = h^{\gamma} g^{\mu+\mu'}$ and $\tilde{y}_2 = h^{\delta} g^{\mu'}$
6) computes $t_3 = h^{r_3} g^{r_1}$, $t_4 = h^{r_4} g^{r_2}$ and $t_5 = h^{r_5}$
7) computes the challenge[4] $c = H_2(m||x||\tilde{y}_1||\tilde{y}_2||T_{G_1}||t_2||t_3||t_4||t_5||Pu||\hat{P}u||Q_i)$
8) computes responses $s_1 = c(\mu+\mu')+r_1$, $s_2 = c\mu'+r_2$, $s_3 = -c\gamma + r_3$, $s_4 = -c\delta + r_4$, $s_5 = -c(\delta+\gamma)+r_5$.
9) The signature is composed by the auxiliary keys, the challenge and the responses, i.e the tuple $\sigma = (c, s_1, s_2, s_3, s_4, s_5, \tilde{y}_1, \tilde{y}_2)$

SignCheck(σ,$pseu_{x,i}$,i,x,W_i,m,PP): The algorithm runs at the verifier side and checks the validity of a signature of a given message m, for a given pseudonym $pseu_{x,i}$ of index x, in a time-slot i. It requires the public parameters PP and the time restriction key W_i. It performs the following operations:

1) computes the time variant parameters $Q_i = H_1(T(time))$
2) computes $d = H_3^x(i)$
3) computes $\bar{T}_{G_1} = s_1 Q_i - cPu$
4) computes $\bar{t}_2 = [e(Q_i, P + \hat{P}u)]^{s_2} / e(Pu + W_i, \hat{P}u)^c$.
5) computes $\bar{t}_3 = h^{s_3} g^{-s_1} \tilde{y}_1^c$
6) computes $\bar{t}_4 = h^{s_4} g^{-s_2} \tilde{y}_2^c$
7) computes $\bar{t}_5 = h^{s_5} (\frac{\tilde{y}_1 \tilde{y}_2}{g^d})^c$
8) checks whether the equality $c' = c$ holds, where: $c' = H_2(m||x||\tilde{y}_1||\tilde{y}_2||\bar{T}_{G_1}||\bar{t}_2||\bar{t}_3||\bar{t}_4||\bar{t}_5||Pu||\hat{P}u||Q_i)$
9) the algorithm outputs "valid" if $c = c'$ and "invalid" otherwise.

E. Specific CA operations for conditional privacy

Independent from the signature scheme algorithms, the proposed system requires the following operations to maintain the datasets described in table III, which enable an efficient anonymity revocation mechanism. Although the proposed signature scheme could provide anonymity revocation without implementing the following operations, this would imply an on-demand computation of 3 pairing operations in the CA side, per revoked pseudonym and per stored credential, which would be inefficient for the proposed crowdsensing application. The efficiency of these operations is detailed in sec. 3.

TABLE III. CONDITIONAL PRIVACY ENABLING DATASETS

DataSet	Updated	Entries	Holding Entity
REG	CreGen	$\{ID, cred\}$ per user	CA
PseuL	$T(time)$	$\{ID, pseu_{1,i}, \dots\}$ per user	CA
RL	Revoke	$pseu_{x_j,i}$ per revoked pseu	CA, Verifier

GenPseuList(REG, i,$\{x_1, \dots, x_n\}$): The operation is executed by the CA before activation of each time-slot i. It takes the registry REG and uses the algorithm *PseuGen(cred,x,i,PP)* for each entry in REG for index values in the set $\{x_1, \dots, x_n\}$. It creates the pseudonym list $PseuL$ with one entry per user of the form $\{ID, pseu_{i,x_1}, \dots, pseu_{i,x_n}\}$. Note that for this operation the CA uses the same *PseuGen* algorithm as the user.

Open($pseu$, i): It is performed by the CA. If the pseudonyms $pseu$ was part of a valid signature for time-slot i, then this operation uses a binary search in the $PseuL$ of time-slot i to find the user identity ID.

Revoke($pseu, i, \{x_i, \dots, x_j\}, RL$): It is performed by the CA. It uses the *Open(pseu,i)* algorith to find the user ID, then

[4]where the operator || represents concatenation.

it retrieves the pseudonyms of the user from the $PseuL$ and updates the RL including the revoked user's pseudonyms for the specific indexes, i.e. $\{pseu_{x_i,i}, \ldots, pseu_{x_j,i}\}$. The RL is sent to the Verifier.

F. Correctness of signature scheme

The signature scheme is correct, if and only if, for all signatures σ of a message m, generated by a *Sign* algorithm, with valid pseudonym and valid credential, the output of Verify algorithm is always "valid", except fi *RevCheck* outputs "invalid".

Note that, given a valid signature $\sigma = (c, s_1, s_2, s_3, s_4, s_5, \tilde{y}_1, \tilde{y}_2)$ of a message m, with a pseudonym $(Pu, \hat{Pu})$ with index x, time-slot i, obtained from a credential (μ, Su), and with a set of public parameters $(G_1, G_T, P, g, h, W, W_i, Q_i)$, then $c = c'$ holds, i.e. the following relations must hold: i) $T'_{G_1} = T_{G_1}$; ii) $t_2 = t'_2$; iii) $t_3 = t'_3$; iv) $t_4 = t'_4$; and v) $t_5 = t'_5$;

Note that:

$$\bar{T}_{G_1} = s_1 Q_i - cPu = (c(\mu + \mu') + r_1)Q_i - c(\mu + \mu')Q_i = rQ_i = T_{G_1}$$

$$\bar{t}_2 = \frac{[e(Q_i, P + \hat{Pu})]^{s_2}}{e(Pu + W_i, \hat{Pu})^c} = \frac{[e(Q_i, P)e(Q_i, \hat{Pu})]^{s_2}}{e((\mu + \mu' + s)Q_i, \frac{\mu'}{\mu+s}P)^c} = \frac{[e(Q_i, P)e(Q_i, \hat{Pu})]^{s_2}}{[e((\mu + s)Q_i, \frac{\mu'}{\mu+s}P)e(\mu' Q_i, \frac{\mu'}{\mu+s}P)]^c} = \frac{[e(Q_i, P)e(Q_i, \hat{Pu})]^{s_2}}{[e(Q_i, P)e(Q_i, \hat{Pu})]^{\mu' c}} = t_2$$

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

$$\bar{t}_4 = h^{s_4} g^{s_2} \tilde{y}_2^c = h^{-c\delta + r_4} g^{-c(\mu') + r_2} (h^\delta g^{(\mu')})^c = h^{r_4} g_{r_2} = t_4$$

$$\bar{t}_5 = h^{s_5} [\frac{\tilde{y}_1 \tilde{y}_2}{g^d}]^c = h^{-c(\delta+\gamma) + r_5} [\frac{h^\gamma g^{(\mu+\mu')} h^\delta g^{(\mu')}}{g^d}]^c = h^{r_5} g^{(\mu + 2\mu' - d)} = h^{r_5} = t_5$$

It is worth commenting that this signature scheme, is converted from a zero-knowledge proof using the Fiat-Shamir heuristic. Concretely, in [17] the proposed ZK-proof is:

$$PoK\{(\mu, \mu + \mu') : Pu = (\mu + \mu')Q_i \wedge v = v'^{\mu'}\}$$

where $v = e(Pu + W_i, \hat{Pu})$

and $v' = [e(Q_i, P)e(Q_i, \hat{Pu})]^{\mu'}$

However, the scheme proposed in this paper extends this PoK to include a linear relation between the values μ and μ', i.e. $2\mu' + \mu = d$.

$$PoK\{(\mu, \mu + \mu') : Pu = (\mu + \mu')Q_i \wedge v = v'^{\mu'} \wedge 2\mu' + \mu = d\} \quad (1)$$

As described in [58], the linear relation of secret keys with different bases can be proved using auxiliary keys, hence the proposed PoK is transformed for practical reasons into:

$$PoK\{(\mu, \mu + \mu', \gamma, \delta) : Pu = (\mu + \mu')Q_i \wedge v = v'^{\mu'} \wedge \tilde{y}_1 = h^\gamma g^{(\mu+\mu')} \wedge \tilde{y}_2 = h^\gamma g^{(\mu')} \wedge \tilde{y}_{1,2} = h^{\gamma+\delta}\}$$

G. Security analysis

This section provides the security analysis of the proposed pseudonym-based signature scheme by defining the unforgeability and the unlinkability-yet-accountability properties of the scheme. To give some intuition: i) unforgeability ensures that if a user outputs a signature that is flagged as valid, then the user must have a valid credential; ii) unlinkability-yet-accountability ensures that a user given a valid credential cannot output two valid signatures with two different pseudonyms for the same index value and same time-slot. It also ensures that it is not possible to distinguish if signatures associated to two different pseudonyms with different index values and/or time-slot values were obtained from the same credential or different credentials.

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 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 to C, then C uses the CreGen algorithm and returns a credential (μ, Su) to A.*
   - *pseudonym queries: A presents a credential to C, an index value x, a time-slot value i, and queries a pseudonym to C. C runs PseuGen and returns a pseuondym $(Pu, \hat{Pu})$ and its secret μ'.*
   - *signature queries: A sends a message m, a pseudonym $(Pu, \hat{Pu})$, and the secret values (μ', μ, Su) 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 $(Pu, \hat{Pu})$ that was never queried and which corresponding secret values, i.e. (μ', μ, Su) 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,n)-CAA assumption with an advantage $\epsilon(k/q_{key})^{q_{key}}(n/q_{H_1})^{q_{H_1}}(1-(q_{H_3}/p))$ considering a polynomial number of queries q_{key}, q_{H_1} and q_{H_3}.

proof: This proof follows similar steps as the proof in [17], but adapted to the proposed scheme. First, let's consider an instance of the (k,n)-CAA problem as defined in sec. VIII-C3, i.e. $P, P_1, \ldots, P_n \in G_1$, and let $x, a_1, \ldots, a_k \in Z_p$ (where k and n are integers). Then C adopts the role of the challenger in the *GAME 1* and uses the adversary A as subalgorithm as follows:

The challenger C computes the system public parameters and sends the parameters to the adversary A. These parameters include the public key $W = xP$ and the time restriction key $W_i = xQ_i$, where $Q_i = \lambda_i P_i$is obtained from the oracle H_1 on input a time-slot value i.

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, C picks a secret key randomly within the predefined set, if such pick falls within the set of $\{a_1, \ldots, a_k\}$ then C generates a credential $cre = (a_i, A_i) = (a_i, 1/(x + a_i)P)$ and returns this credential to A. 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 $cred = (a_i, A_i)$ to C, then C obtains a random d from the H_3 oracle and computes $a_i' = (d - a_i)/2$ and computes $P_u = (a_i + a_i')\lambda_i P_i$ and $\hat{P}_u = a_i' A_i$.
3) **Signature Queries**: A sends a message m and a pseudonym to C who returns the signature $(c, s_1, s_2, s_3, s_4, s_5, \hat{y}_1, \hat{y}_2)$ to A if the presented pseudonym was previously queried. To compute these values C uses the $SignCheck$ algorithm and sets the response of the oracle H_2 to c, if this causes a collision with a previous query to the oracle H_2 then C aborts. The probability of not aborting at this step, taking into account that A makes q_{H_2} queries to H_2 oracle, and that the response of this oracle is random in Z_p^*, is $1 - (q_{H_2}/p)$.

The challenger also defines the random oracles H_1, H_2 and H_3 as follows:

- H_1 oracle Queries: C prepares a polynomial number of responses for $\{q_1, \ldots, q_{H_1}\}$ and the sets $\{P_1, \ldots, P_n\}$ is distributed randomly among them. Then, when a value i is submitted to H_1 the challenger C chooses randomly among the q_{H_1} prepared responses and replies to A. If the answer is in the set $\{P_1, \ldots, P_n\}$ then C picks randomly $\lambda_i \xleftarrow{R} Z_p^*$ and outputs $\lambda_i P_i$. If the answer is not within the set $\{P_1, \ldots, P_n\}$ then the C halts and aborts this game. Hence the probability of not aborting is $(n/q_{H_1})^{q_{H_1}}$.
- H_2 oracle Queries: On any input of the form $(m||x||\tilde{y}_1||\tilde{y}_2||\bar{T}_{G_1}||\bar{t}_2||\bar{t}_3||\bar{t}_4||\bar{t}_5||Pu||\hat{P}u||a_i)$ the challenger C picks $c \xleftarrow{\text{R}} Z_p^*$ and responds.
- H_3 oracle Queries: On any input of the form $i = T(time)$ the challenger C picks $d \xleftarrow{\text{R}} Z_p^*$ and responds.

Let's assume that after performing a polynomial number of queries, A presents a signature $(c, s_1, s_2, s_3, s_4, s_5, \tilde{y}_1, \tilde{y}_2)$ that is valid for a pseudonym $(Pu, \hat{P}u)$. The challenger C can verify if the credential of the presented pseudonym was obtained during the credential queries by checking whether $e(Pu+W_i, Su) = e(Q_i, P+\hat{P}u)$ holds for any of the queried credentials. If the credential was not queried, and the signature is valid, for the Forking Lemma[5] [59] the adversary will be able to output another valid signature with the same inputs but different challenge. This is, A will be able to present after polynomial time another signature σ' with the same auxiliary keys $(\tilde{y}_1, \tilde{y}_2)$ and commitments $T_{G_1} = T'_{G_1}$, $t_2 = t'_2$, $t_3 = t'_3$, $t_4 = t'_4$, $t_5 = t'_5$ but different challenge and responses, i.e. $(c', s_1', s_2', s_3', s_4', s_5'))$ such that $c \neq c'$, $s_1 \neq s_1'$, $s_2 \neq s_2'$, $s_3 \neq s_3'$, $s_4 \neq s_4'$, $s_5 \neq s_5'$.

Then A can solve the (k,n)-CAA problem. Since:

$$\begin{array}{l} T_{G_1} = T'_{G_1} \Rightarrow s_1 Q_i - cPu = s_1' Q_i - c' Pu \Rightarrow \\ (z_1 - z_1')Q_i = (c - c')Pu \Rightarrow Pu = \frac{(z_1 - z_1')}{(c - c')} Q_i \end{array} \quad (2)$$

Also:

$$\begin{array}{l} t_2 = t_2' \Rightarrow \frac{\left[e(Q_i,P)e(Q_i,\hat{P}u)\right]^{s_2}}{e(Pu+W_i,\hat{P}u)^{c}} = \frac{\left[e(Q_i,P)e(Q_i,\hat{P}u)\right]^{s_2'}}{e(Pu+W_i,\hat{P}u)^{c'}} \Rightarrow \\ e(Q_i,P)^{(s_2-s_2')/(c-c')} = e\left(Pu - \frac{s_2-s_2'}{c-c'}Q_i + W_i, \hat{P}u\right) \Rightarrow \\ e\left(\frac{(s_1-s_1')-(s_2-s_2')}{c-c'}P_i + xP_i, \frac{c-c'}{s_2-s_2}\hat{P}u\right) = e(P_i, P) \end{array} \quad (3)$$

Hence, the adversary A can find a solution (aP_i, S) where

$$S = \frac{(c - c')}{(z_2 - z_2')}\hat{P}u \quad (4)$$

and

$$aP_i = \frac{(z_1 - z_1') - (z_2 - z_2')}{(c - c')}P_i \quad (5)$$

[5]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.

Note that, since the probability of not aborting in this game is $(k/q_{key})^{q_{key}}(n/q_{H_1})^{q}_{H_1}(1-(q_{H_3}/p))$ the advantage of C in solving the $(k,n)-CAA$ problem is not negligible.

Unlinkability-yet-Accountability: A pseudonym-based signature scheme is unlinkable-yet-accountable if:

- given two pseudonym-based signatures with different index value x and/or time-slot value i, there is no algorithm deciding whether these two signatures were obtained from the same credential or two different credentials that is better than a random guess. This feature in this kind of construction is straightforward to prove.

  Following the same reasoning as [58], all elements in the signature are randomized, i.e. the commitments $(s_1, s_2, s_3, s_4, s_5)$ are randomized with random factors $(r_1, r_2, r_3, r_4, r_5)$, and the auxiliary keys $(\hat{y}_1, \hat{y}_2)$ are randomized with random factors (γ, δ). Hence it is not possible to distinguish these elements from random values obtained from groups Z_p^* and G_T respectively. Also the challenge is obtained from a random oracle. Similarly, as explained in [17], the pseudonym values $(P_u, \hat{P}_u)$, constructed as $((\mu'+\mu)Q_i, \mu' Su)$, are randomized with the factor μ' and look like random values in G_1. Hence, only messages signed by the same pseudonym can be linked to the same credential, i.e. pseudonyms from the same user with the same index value and time-slot.

- 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 to C, then C uses the CreGen algorithm and returns a credential (μ, Su) to A.*
   - *pseudonym queries: A presents a credential to C, and an index value x, and a time-slot value i, and queries a pseudonym to C. C runs PseuGen and returns a pseudonym $(Pu, \hat{P}u)$ and its secret μ' which is computed as $\mu' = (d-\mu)/2$.*
   - *signature queries: A sends a message m, a pseudonym $(Pu, \hat{P}u)$, and the secret values (μ', μ, Su) to C. C runs the Sign algorithm and returns σ to A.*
3) *After a polynomial number of queries, A outputs a signature σ, and pseudonym $(P_u, \hat{P}_u)$ on the message m for a credential $cred = (\mu, Su)$ that was previously obtained in the credential queries, and for a pseudonym index x and time-slot i that was previously queried in a pseudonym query. A wins the game if the signature is flagged as valid by the Verify algorithm and the pseudonym is different than the one returned by C in the pseudonym query.*

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 the steps given in [58]. Let us assume that C is given an instance two random elements g and $h \in G_T$, for which the discrete logarithm a is not know $g = h^a$. Then the challenger C can use these values as generators of G_T in the Setup phase and use A as sub-algorithm to compute a. This is, after performing the adversary queries as defined above, A presents a valid signature $\sigma = (c, s_1, s_2, s_3, s_4, s_5, \tilde{y}_1, \tilde{y}_2)$ and a pseudonym $(P_u, \hat{P}_u)$, for a index value x and time-slot i to win the above game. The challenger C can verify that the presented pseudonym was obtained with a credential obtained in a credential query by checking whether $e(Pu + W_i, Su) = e(Q_i, P)e(Q_i, \hat{P}u)$ holds for any of the queried credentials. Then C can verify that the pseudonym for time-slot i and index value x was queried in the pseudonym queries for such credential. Finally, C verifies that the pseudonym returned in the pseudonym query does not match the pseudonym presented by A. Hence, A has obtained a pseudonym which secret value μ' does not follow the relation $\mu' = (d-\mu)/2$. Then, for the Forking lemma, A can obtain in polynomial time another valid signature with the same commitments and auxiliary keys but with different challenge and responses, i.e. $\sigma' = (c', s_1', s_2', s_3', s_4', s_5')$. I such case we have that:

$$\begin{array}{l} t_3 = t_3' \Rightarrow h^{s_3} g^{s_1} \tilde{y}_1^c = h^{s_3'} g^{s_1'} \tilde{y}_1^{c'} \Rightarrow \\ \tilde{y}_1^{c-c'} = h^{s_3'-s_3} g^{s_1'-s_1} \Rightarrow \\ \tilde{y}_1 = h^{\frac{s_3'-s_3}{c-c'}} g^{\frac{s_1'-s_1}{c-c'}} \end{array} \quad (6)$$

similarly

$$\begin{array}{l} t_4 = t_4' \Rightarrow h^{s_4} g^{s_2} \tilde{y}_2^c = h^{s_4'} g^{s_2'} \tilde{y}_2^{c'} \Rightarrow \\ \tilde{y}_2 = h^{\frac{s_4'-s_4}{c-c'}} g^{\frac{s_2'-s_2}{c-c'}} \end{array} \quad (7)$$

$$\begin{array}{l} t_5 = t_5' \Rightarrow h^{s_5} \tilde{y}_{1,2}^c = h^{s_5'} \tilde{y}_{1,2}^{c'} \Rightarrow \\ \tilde{y}_{1,2} = h^{\frac{s_5'-s_5}{c-c'}} \end{array} \quad (8)$$

Since $\tilde{y}_{1,2} = \frac{\tilde{y}_1 \tilde{y}_2}{g^d}$, we have that:

$$h^{\frac{(s_3'-s_3)+(s_4'-s_4)}{c-c'}} g^{\frac{(s_1'-s_1)+(s_2'-s_2)}{c-c'}} g^{-d} = h^{\frac{s_5'-s_5}{c-c'}} \quad (9)$$

Since $s_1 = c(\mu+\mu') + r_1$, $s_1' = c'(\mu+\mu') + r_1$, $s_2 = c\mu' + r_2$ and $s_2' = c'\mu' + r_2$, then we have that if the equation $2\mu' + \mu = d$ does not hold, the g-part of the above equation does not vanish, and then it is possible to compute the discrete logarithm of h with respect to g as follows:

$$log_g(h) = \left[\frac{(s_1'-s_1)+(s_2'-s_2)}{c-c'} - d\right]\left[\frac{(s_3'-s_3)+(s_4'-s_4)}{s_5'-s_5}\right] \quad (10)$$

IX. Batch Verification

In a crowdsensing application the verification process of the signed sensing data is centralized in the AS that adopts the role of Verifier, i.e. the signed messages are not transmitted from users to other users, but to the AS in the Smart City data processing service. Hence, it is convenient to provide a mechanism where a large group of signed messages, from different users, can be verified efficiently. Batch verification provides such method. It consists of verifying a group of signatures at once instead of verifying one by one sequentially. The main advantage is that it reduces the computational cost, the disadvantage is that if one only signature in the batch is not valid, then the verification of the whole batch fails.

The proposed pseudonym-based signature scheme can be slightly modified to enable batch verification. Such modification was proposed in [60] for short identity-based signatures, but it can be applied to our proposed pseudonym-based signature scheme with some mathematical manipulation using the pairing properties. This modification consists of increasing the signature size by including the commitment that requires higher computational costs to avoid its computation during verification. The validity of this commitment is also verified, but such verification is merged with that of other signatures to reduce the number of pairing operations. Note that, in the previously described *SignCheck* algorithm, the commitment with higher complexity (i.e. involving pairing operations) is:

$$\bar{t}_2 = [e(Q_i, P + \hat{P}u)]^{s_2} / e(Pu + W_i, \hat{P}u)^c \qquad (11)$$

This value, computed in the *Sign* as t_2, is included in the modified signature, i.e. $\sigma = (c, t_2, s_1, s_2, s_3, s_4, s_5, \tilde{y}_{1,j}, \tilde{y}_{2,j})$. The batch verification algorithm for a group of N signatures $\sigma_j = (c_j, t_{2,j}, s_{1,j}, s_{2,j}, s_{3,j}, s_{4,j}, s_{5,j}, \tilde{y}_{1,j}, \tilde{y}_{2,j})$, with corresponding pseudonym $(Pu_j,\ \hat{P}u_j)$ of index x_j, where $j = 1 \dots N$, is performed as follows:

For each received signature, the AS:

1) Computes the time variant parameters $Q_i = H_1(T(time))$ and $d = H_3^{x_j}(T(time))$.
2) Computes $\bar{T}_{G_1,j} = s_{1,j} Q_i - c_j P u_j$
3) Computes $\bar{t}_{3,j} = h^{s_{3,j}} g^{-s_{1,j}} \tilde{y}_{1,j}^{c_j}$
4) Computes $\bar{t}_{4,j} = h^{s_{4,j}} g^{-s_{2,j}} \tilde{y}_{2,j}^{c_j}$
5) Computes $\bar{t}_{5,j} = h^{s_{5,j}} (\frac{\tilde{y}_{1,j}\tilde{y}_{2,j}}{g^d})^{c_j}$
6) Validates the signature if the equality $c'_j = c_j$ holds, where:
   $c'_j = H_2(m||x_j||\tilde{y}_{1,j}||\tilde{y}_{2,j}||\bar{T}_{G_1}|| \dots$
   $\dots t_{2,j}||\bar{t}_{3,j}||\bar{t}_{4,j}||\bar{t}_{5,j}||Pu_j||\hat{P}u_j||Q_i)$

Note that the only difference between the above algorithm and the *SignCheck* algorithm is the absence of computation of the commitment $\bar{t}_{2,j}$, which is already included in the signature. The validity of this value has to be checked as well, but this verification can be done for all the signatures at once by using the Small Exponents Test [61]. For that, the Verifier generates N random values $\delta_j \in Z_p$ (for $j = 1 \dots N$) and checks whether the following equation holds:

$$\prod_{j=1}^{N} (t_{2,j})^{\delta_j} = \frac{e\left(Q_i, \sum_{j=1}^{N} (\delta_j s_{2,j})[\hat{P}u_j + P]\right)}{\prod_{j=1}^{N} e\left(Pu_j + W_i, (\delta_j c_j)\hat{P}u_j\right)} \qquad (12)$$

A. Batch verification correctness

It is trivial to see that this batch verification algorithm performs the same validity tests as the *SignCheck* algorithm, which has been proven secure in section VIII-G. It remains to be shown that eq. 12 is equivalent to performing eq. 11 for the N signatures in the batch. Note that, with some mathematical manipulation:

$$\begin{aligned}
\prod_{j=1}^{N} (t_{2,j})^{\delta_j} &= \frac{e\left(Q_i, \sum_{j=1}^{N} (\delta_j s_{2,j})[\hat{P}u_j + P]\right)}{\prod_{j=1}^{N} e\left(Pu_j + W_i, (\delta_j c_j)\hat{P}u_j\right)} = \\
&= \frac{e(Q_i, (\sum_{j=1}^{N} (\delta_j s_{2,j}))P) e(Q_i, \sum_{j=1}^{N} (\delta_j s_{2,j} \hat{P}u_j))}{\prod_{j=1}^{N} e(Pu_j + W_i, \hat{P}u_j)^{\delta_j c_j}} = \\
&= \frac{\prod_{j=1}^{N} e(Q_i, P)^{\delta_j s_{2,j}} e(Q_i, \hat{P}u_j)^{\delta_j s_{2,j}}}{\prod_{j=1}^{N} e(Pu_j + W_i, \hat{P}u_j)^{\delta_j c_j}} = \\
&= \prod_{j=1}^{N} \left[\frac{[e(Q_i, P + \hat{P}u_j)]^{s_{2,j}}}{e(Pu_j + W_i, \hat{P}u_j)^{c_j}} \right]^{\delta_j}
\end{aligned} \qquad (13)$$

Removing the random values δ_j in both sides then we have that eq. 12 is equivalent to performing eq. 11 for the N signatures.

It is also worth commenting that in an optimal design of a batch verification mechanism, the number of pairing operations should be constant and not dependant of the number of signatures in the batch [60][62]. In such a case, batch verification can be cost-effective even in the presence of invalid signatures [60], since techniques such as *divide-and-conquer* [63] can be applied efficiently. However, in our proposed batch verification mechanism the number of pairing operations is not constant. The number of pairing operations for batch verification is N+1 (eq. 12), where N is the number of signatures. The proposed mechanism reduces the number of pairing operations with respect to the sequential performance of individual signature verifications, which is 2N, but it is not resilient to invalid signatures and it should not be used for large batch sizes. Also, signatures of messages which sensing data fall out of reasonable boundaries should be verified individually.

X. Complexity analysis

This section caters for a complexity analysis that compares the proposed pseudonym-based signature scheme with

TABLE IV. COMPUTATIONAL COMPLEXITY

Scheme	*algorithm*	e()	Exp. G_1	Exp. G_T	O()
Our Scheme	Sign	1	1	10	O(1)
	SignCheck	2	2	11	O(1)
	RevCheck	0	0	0	O($log_2 R$)
	Revoke	0	0	0	O($log_2 N$)
PS [17]	Sign	2	1	1	O(1)
	SignCheck	3	2	2	O(1)
	RevCheck	-	-	-	-
	Open	3	0	0	O(N)
SRBE [13]	Sign	3	5	3	O(1)
	SignCheck	4	3	4	O(1)
	RevCheck	0	0	0	O($log_2 R$)
	Revoke	0	0	0	O($log_2 R$)
CLHZ [64]	Sign	5	5	5	O(1)
	SignCheck	7	6	7	O(1)
	RevCheck	0	0	0	O(R)
	Revoke	0	0	0	O(1)

[13], which is the most recent GS scheme proposed for crowdsensing. We also include in our comparison the scheme [64], which data is obtained from the complexity analysis in [13], and the pseudonym-based signature scheme from [17]. Table IV shows: i) the number of pairing operations $e()$; ii) exponentiations in G_1[6]; and iii) exponentiations in G_T. These operations are more complex than the rest and dominate the computational effort. Among these three kind of operations, bilinear pairing operations $e()$ are the most costly, hence these are the ones that we aimed to minimize in our design. We also express the complexity $O()$ in terms of the number of registered users N and revoked pseudonyms R.

The revocation complexity of O($log_2 R$), with no exponentiations or bilinear operations in both [13] and in our proposed scheme, is achieved due to a revocation process based on a binary tree search. A given pseudonym is looked up in a pseudonym list which is binary sorted. Such list is precomputed by the entity with revocation rights, i.e. the GS in [13] or the CA in our proposed scheme. Hence, the low complexity of the revocation process comes at the cost of increasing the complexity in the GS or CA for pseudonym list computation. Fortunately, pseudonym computation does not require bilinear operations (the performance evaluation in sec. XI details this process). Other works such as [17] perform an on demand search on the list of issued credentials, which requires 3 pairing operations per credential. In this work authors do not formalize a revoke algorithm, but detail the operations required to track a user from a valid signature by iterating over the registered credentials, which we include in table IV with the name of *Open* algorithm. The efficiency of our proposed scheme, which is shown in next section, is explained by a lower dependency on bilinear pairing operations.

XI. PERFORMANCE EVALUATION

The java library in [65] was used for the implementation of the proposed pseudonym-based signature scheme and its integration into our crowdsensing platform[7]. Namely, a type A curve ($y^2 = x^3 + ax$) over the field F_q with the recommended settings in [66], 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 V. It is worth clarifying that the verification of a signature requires the pseudonym values together with the signature, which sums up to 7068 bits. The pseudonym index is an integer value of a small bit number, and it can be implicit of the specific sensing task.

TABLE V. SIGNATURE SCHEME ELEMENTS' BIT LENGTH

Emement	bits	Emement	bits
credential (Su)	1024	signature challenge (c)	160
credential (u)	160	signature response ($s1$)	160
pseudonym (Pu)	1024	signature response ($s2$)	160
pseudonym ($\tilde{P}u$)	1024	signature response ($s3$)	160
pseudonym idx (x)	8	signature response ($s4$)	160
signature aux key ($y1$)	1024	signature response ($s5$)	160
signature aux key ($y2$)	1024	Message Overhead	7068

The computation time for the proposed signature scheme was obtained by running the different steps of the cryptosystem on a Intel Core i7 through 10000 iterations. The time performance for pseudonym generation and message signing operations are also tested in a Samsung Galaxy S6. The results are presented in Fig. 3. The comparison between the time performance for batch verification and individual signature verification for the Intel Core i7 is given in Fig. 4. Batch verification has been tested in the absence of invalid signatures.

It is worth commenting that the users generate pseudonyms on-demand, by using the index value published by the data collector entity for each planned sensing task. Similarly, the message signing algorithm on the user side is only performed when data is to be transmitted. Effective strategies out of the scope of this paper are deployed to reduce the periodicity of the data transmitted. Regarding the message overhead, it is worth mentioning previous work that suggests the inclusion of crowdsensing overhead piggybacked in phone calls [67].

The CA must be efficient in the revocation process, i.e. identifying and excluding users in case of misbehavior should require negligible time. Fast revocation requires and efficient user identification and revocation list computation. In our proposed system, the algorithm $Revoke$ requires the execution of the *Open* algorithm, which consists of a binary search on $PseuL$, i.e. the list of pseudonyms. This binary search takes a time in the order of milliseconds, which is negligible. The $PseuL$ list is pre-computed by the CA before the activation of each time-slot. The pseudonym list is updated on-demand when new users register in the system and recomputed before every time-slot starts. Figure 5 shows the time required for the pseudonym list computation for different configurations and number of users.

The revocation list RL is obtained by indexing the revoked pseudonyms in the list of all users' pseudonyms, hence computation and update times for the revocation list are negligible, our tests yield values in the order milliseconds. This high efficiency in the user identification and revocation process is achieved at the cost of increasing the complexity of the CA, since the pseudonym list computation for all registered users in the CA is a time consuming operation, which imposes a lower bound in the duration of the time-slots. The duration of

[6]Note that we use additive notation for operations in G_1 throughout this paper, hence exponentiation in G_1 are depicted as multiplications between Elliptic curve points and elements in Z_p^*

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

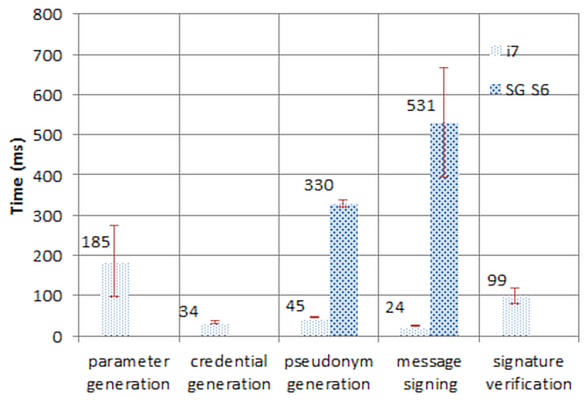


Fig. 3. Time performance in milliseconds in a Intel Core i7 and Samsung Galaxy S6.

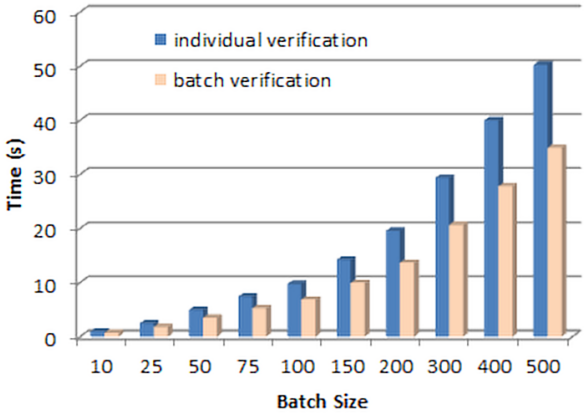


Fig. 4. Time performance for individual signature verification and batch verification.

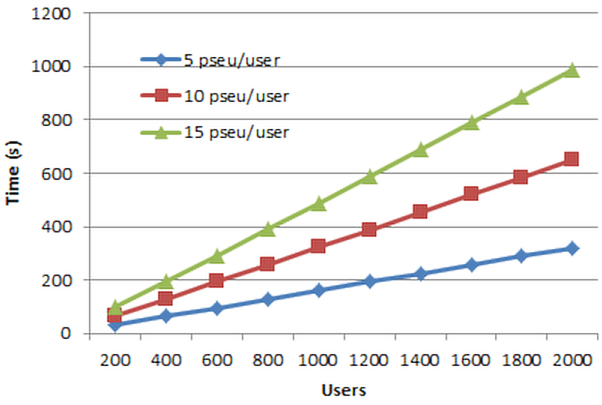


Fig. 5. Pseudonym list computation time in seconds in a Intel Core i7.

a time slot should be significantly higher than the pseudonym list computation time.

It is also worth mentioning that to speed up the $PseuL$ computation in the CA, only the Pu value is computed in each pseudonym, i.e. the $\hat{Pu}$ value is not obtained. The Pu value is unique for every pseudonym, hence it is sufficient to identify a user in case of revocation of her privacy rights. Similarly, when the RL is transmitted from the CA to the Verifier only the Pu value per revoked pseudonym is included in the list.

These tests have been performed with and Android app and

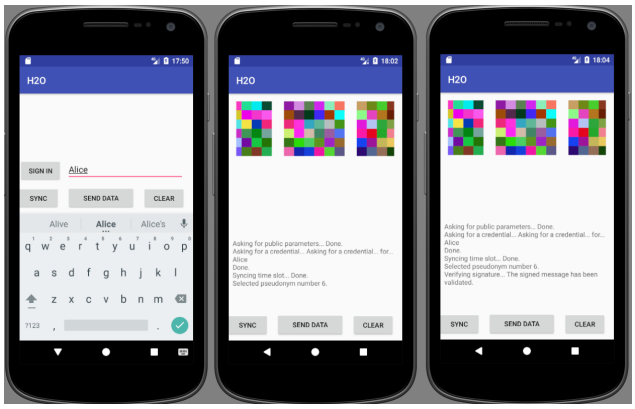


Fig. 6. Snapshots of the Android app used for testing. From left to right: i) the user is asked to sign in and obtains a credential; ii) the app generates a number of pseudonyms; iii) the user selects a pseudonym and sends a signed message to the Authentication server.

prototypes for both the CA server and the authentication server. A snapshot of the Android application is presented in Fig. 6. Note that the pseudonyms are represented with chromatic maps, i.e. a matrix of colored squares that codify the 1024 bits of the Pu value of a pseudonym. This app was only used for the testing of the pseudonym system, which is integrated into the pilot described in sec. XII.

XII. Pilot Implementation

The European H2O R&I project, under the EUREKA-CATRENE programme, seeks to develop human-centric secure architectures to support the rapidly emerging wearable computing for Smart City application domains. In this context, several use cases have been defined focusing on key practical application aspects, and crowdsensing forms part of the envisioned use cases. A pilot based on this crowdsensing use has been deployed in the city of Lisbon, Portugal, and it is active since end-2018. This pilot counts with a working prototype, which includes: i) a CA issuing credentials; ii) users that register in the CA to obtain a credential to self-generate pseudonyms and participate in the sensing tasks; iii) a verifier that validates the signed messages from users; and iv) a data collector that processes the data. The data collector also publishes the different sensing tasks in which citizens can participate. For that, the data collector publishes the sensing data required (i.e. temperature, humidity, CO2 levels, etc.) and the specific pseudonym index that citizens can use for participating in that task. Although an incentive program should also be implemented such as [6], this has been left for future work.

In this prototype the time-slot is set to 24 hours. Hence, the pseudonym list $PseuL$ and Revocation List RL are updated by the CA on a daily basis or when a new user is registered or revoked, respectively. The RL is transmitted from the CA to the verifier when it is updated. The $PseuL$ is generated in the previous time-slot before its activation, which prevents from synchronization delays that could enable revoked users to submit data during the RL updates.

In this pilot a smartphone crowdsening application has been developed and it integrates the proposed pseudonym-based signature scheme. This application polls periodically the

wearable sensors for data, available on a wearable bracelet that has a Bluetooth Low Energy (BLE) interface for communication. The cloud back-end service stores the received data in a specialized NoSQL time series database, InfluxDB, and Grafana for data visualization. The authentication server and CA are implemented into the same cloud server, although located in separated entities. Figure 7 shows the current implementation diagram of the current cloud server implementation. The crowdsensing mobile application, which is used as gateway between the wearable sensors and the cloud server, is developed using Google Android tools, namely Android Studio and Android SDK. This application adopts similar interface as the sensor manufacturer app [68], but including a pseudonym selection panel. Figure 8.1 shows a snapshot of the crowdsensing android application. The wearable device also displays the sensing data or the current active pseudonym, as it is shown in Fig. 8.2. Although in the current implementation all the user side operations of the pseudonym-based signature scheme are performed in the gateway, i.e. the smartphone, in the future the operations involving the credential's secret key will be performed in a secure element already embedded in the wearable device.

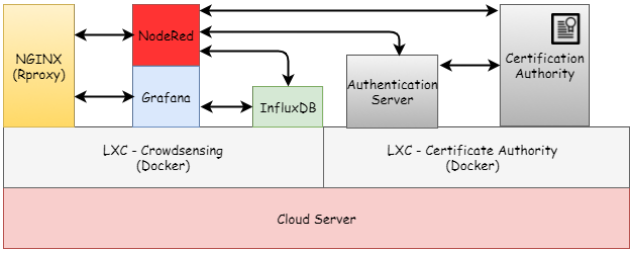


Fig. 7. Cloud server system architecture of the crowdsensing pilot.

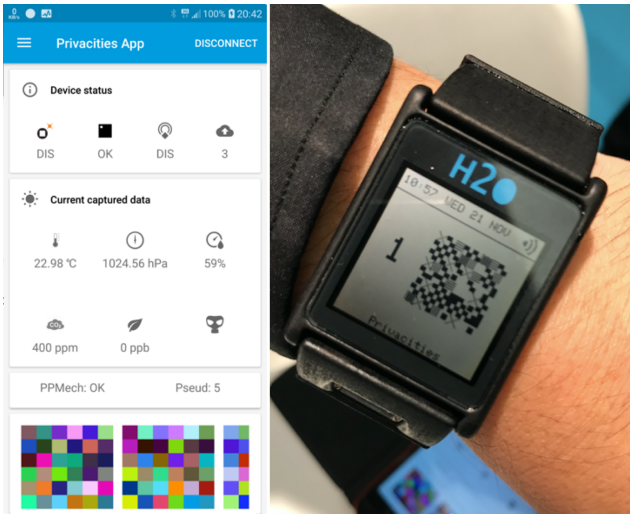


Fig. 8. From left to right: i) crowdsensing mobile app user interface, it displays the pseudonym selection pannel (bottom part) and the sensing data (upper part); ii) wearable sensor device controller integrated into smartwatch form factor, it displays the selected pseudonym.

The cloud server implements a graphical user interface where the different measurements obtained from users are located as landmarks. Figure 9 shows the the data submitted by different users from a specific point of interest located in Lisbon, Portugal.

XIII. Conclusion

This paper addresses the privacy issue of crowdsensing in Smart Cities by proposing a pseudonym-based signature scheme. The proposed scheme is scalable, since users self-generate pseudonyms, hence it does not require users to trigger periodic pseudonym or key renewal from the CA. Similar to previous works this scheme also enables a time-slot division where users pseudonyms are renewed. However, unlike previous schemes the proposed pseudonym-based signature scheme enables users to self-generate an unlimited number of unlinkable pseudonyms while preventing selfish users from using these pseudonyms for the same task. This feature is convenient for crowdsensing applications since it enables users to participate in several sensing tasks simultaneously, while preventing the Smart City from linking different task participations to a single user. Also, it prevents users from participating in a sensing task with more than one pseudonyms, which could compromise the incentive program and could also have a deleterious effect in the sensing data collection process.

Additionally, the proposed scheme requires less bilinear pairing computations than previous schemes proposed in crowdsensing applications. In fact, only one pairing operation is required on the user side for the *Sign* algorithm. Hence, the proposed scheme can be efficiently implemented as shown in the performance evaluation section. Also, the proposed batch verification reduces the time required for a centralized verification up to 34% with respect to the proposed individual signature verification. In addition, the pseudonym revocation list is computed and renewed efficiently by indexing the list entries into the full pseudonym list, which is pre-computed before each time slot. Finally, the paper also shows how the proposed mechanism can be implemented and integrated into a real crowdsensing platform.

XIV. Acknowledgements

This work is supported by the European Regional Development Fund (FEDER), through the Regional Operational Programme of Centre (CENTRO 2020) of the Portugal 2020 framework [Project PRIVACITIES with Nr. 017785 (CENTRO-01-0247-FEDER-017785)], and has also been carried out in the scope of the project labelled as CAT209-H2O (Human to Objects - Easy Interactions in the Smart City) by the European EUREKA-CATRENE programme.

References

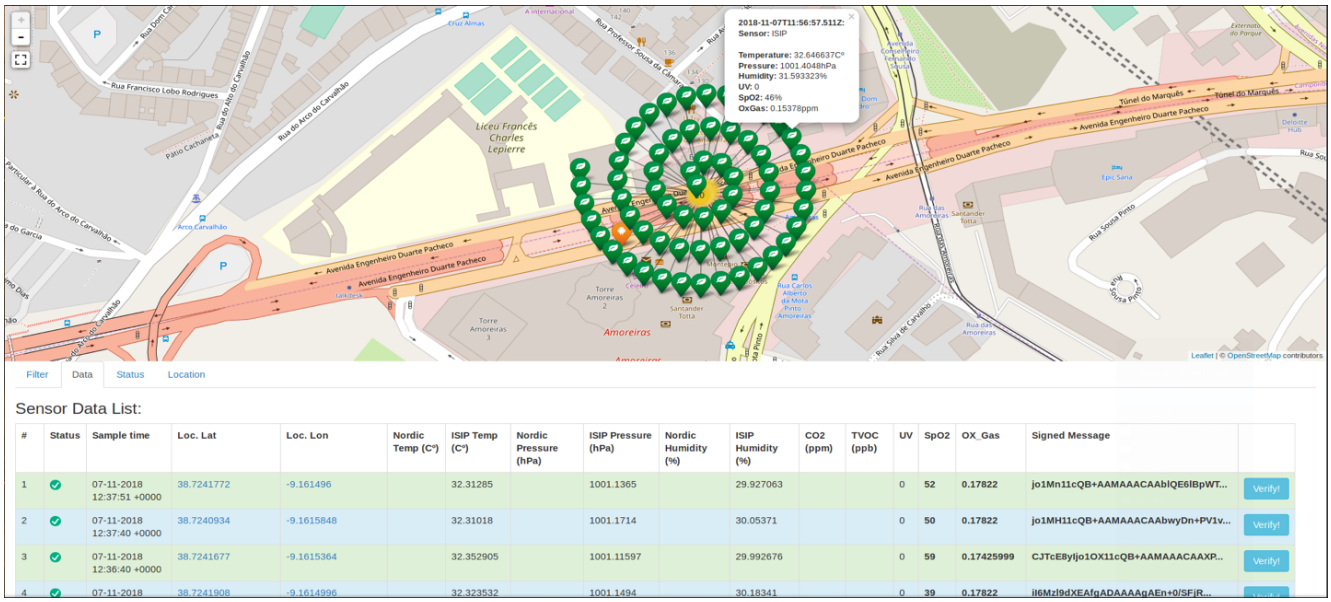
[1] P. Kulkarni and T. Farnham. Smart city wireless connectivity considerations and cost analysis: Lessons learnt from smart water case studies. *IEEE Access*, 4:660–672, 2016.

[2] F. G. Brundu, E. Patti, A. Osello, M. D. Giudice, N. Rapetti, A. Krylovskiy, M. Jahn, V. Verda, E. Guelpa, L. Rietto, and A. Acquaviva. Iot software infrastructure for energy management and simulation in smart cities. *IEEE Transactions on Industrial Informatics*, 13(2):832–840, April 2017.

[3] X. Li, Z. Lv, I. H. Hijazi, H. Jiao, L. Li, and K. Li. Assessment of urban fabric for smart cities. *IEEE Access*, 4:373–382, 2016.

[4] M. Pouryazdan and B. Kantarci. The smart citizen factor in trustworthy smart city crowdsensing. *IT Professional*, 18(4):26–33, July 2016.

[5] G. Cardone, L. Foschini, P. Bellavista, A. Corradi, C. Borcea, M. Talasila, and R. Curtmola. Fostering participaction in smart cities: a geo-social crowdsensing platform. *IEEE Communications Magazine*, 51(6):112–119, June 2013.



#	Status	Sample time	Loc. Lat	Loc. Lon	Nordic Temp (C°)	ISIP Temp (C°)	Nordic Pressure (hPa)	ISIP Pressure (hPa)	Nordic Humidity (%)	ISIP Humidity (%)	CO2 (ppm)	TVOC (ppb)	UV	SpO2	OX_Gas	Signed Message	
1	✓	07-11-2018 12:37:51 +0000	38.7241772	-9.161496		32.31285		1001.1365		29.927063			0	52	0.17822	jo1Mn11cQB+AAMAAACAAblQE6lBpWT...	Verify!
2	✓	07-11-2018 12:37:40 +0000	38.7240934	-9.1615848		32.31018		1001.1714		30.05371			0	50	0.17822	jo1MH11cQB+AAMAAACAAbwyDn+PV1v...	Verify!
3	✓	07-11-2018 12:36:40 +0000	38.7241677	-9.1615364		32.352905		1001.11597		29.992676			0	59	0.17425999	CJTcEByIjo1OX11cQB+AAMAAACAAXP..	Verify!
4	✓	07-11-2018	38.7241908	-9.1614996		32.323532		1001.1494		30.18341			0	39	0.17822	il6Mzl9dXEAfgADAAAAgAEn+0/SFjR...	

Fig. 9. Web interface of the crowdsensing pilot in Lisbon, the screenshot displays data obtained by different users in a specific point of interest.

[6] Stylianos Gisdakis, Thanassis Giannetsos, and Panos Papadimitratos. Sppear: Security & privacy-preserving architecture for participatory-sensing applications. In *Proceedings of the 2014 ACM Conference on Security and Privacy in Wireless & Mobile Networks*, WiSec '14, pages 39–50, New York, NY, USA, 2014. ACM.

[7] P. Sotres, J. R. Santana, L. S nchez, J. Lanza, and L. Mu oz. Practical lessons from the deployment and management of a smart city internet-of-things infrastructure: The smartsantander testbed case. *IEEE Access*, 5:14309–14322, 2017.

[8] K. Zhang, J. Ni, K. Yang, X. Liang, J. Ren, and X. S. Shen. Security and privacy in smart city applications: Challenges and solutions. *IEEE Communications Magazine*, 55(1):122–129, January 2017.

[9] R. Khatoun and S. Zeadally. Cybersecurity and privacy solutions in smart cities. *IEEE Communications Magazine*, 55(3):51–59, March 2017.

[10] A. Martinez-Balleste, P. A. Perez-martinez, and A. Solanas. The pursuit of citizens' privacy: a privacy-aware smart city is possible. *IEEE Communications Magazine*, 51(6):136–141, June 2013.

[11] K. Emara. Safety-aware location privacy in vanet: Evaluation and comparison. *IEEE Transactions on Vehicular Technology*, PP(99):1–1, 2017.

[12] Cory Cornelius, Apu Kapadia, David Kotz, Daniel Peebles, Minho Shin, and Nikos Triandopoulos. Anonysense: privacy-aware people-centric sensing. In *MobiSys*, 2008.

[13] Rahaman Sazzadur, Cheng Long, Yao Danfeng Daphne, Li He, and Park Jung-Min Jerry. Provably secure anonymous-yet-accountable crowdsensing with scalable sublinear revocation. In *Proceedings on Privacy Enhancing Technologies*, PETS-2017, pages 384–403, 2017.

[14] Maxim Raya and Jean-Pierre Hubaux. The security of vehicular ad hoc networks. In *Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks*, SASN '05, pages 11–21, New York, NY, USA, 2005. ACM.

[15] 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.

[16] 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.

[17] 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.

[18] J. Lin, M. Li, D. Yang, G. Xue, and J. Tang. Sybil-proof incentive mechanisms for crowdsensing. In *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pages 1–9, May 2017.

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

[20] 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.

[21] 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.

[22] 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.

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

[24] Foteini Baldimtsi and Anna Lysyanskaya. Anonymous credentials light. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, CCS '13, pages 1087–1098, New York, NY, USA, 2013. ACM.

[25] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques: Advances in Cryptology*, EUROCRYPT '01, pages 93–118, London, UK, UK, 2001. Springer-Verlag.

[26] Jan Camenisch, Susan Hohenberger, Markulf Kohlweiss, Anna Lysyanskaya, and Mira Meyerovich. How to win the clonewars: Efficient periodic n-times anonymous authentication. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*, CCS '06, pages 201–210, New York, NY, USA, 2006. ACM.

[27] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings*, volume 2656 of *Lecture Notes in Computer Science*, pages 614–629. Springer, 2003.

[28] Keita Emura and Takuya Hayashi. A light-weight group signature scheme with time-token dependent linking. In Tim Güneysu, Gregor Leander, and Amir Moradi, editors, *Lightweight Cryptography for*

Security and Privacy, pages 37–57, Cham, 2016. Springer International Publishing.

[29] Mark Manulis, Nils Fleischhacker, Felix Gunther, Franziskus Kiefer, and Bertram Poettering. Group signatures: Authentication with privacy. Bundesamt fur Sicherheit in der Informationstechnik. Tech. Rep., 2012.

[30] 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.

[31] Yun Huang, Zheng Huang, Haoran Zhao, and Xuejia Lai. A new one-time password method. *IERI Procedia*, 4:32 – 37, 2013. 2013 International Conference on Electronic Engineering and Computer Science (EECS 2013).

[32] H. A. Nugroho, Y. Priyana, A. S. Prihatmanto, and K. H. Rhee. Pseudonym-based privacy protection for steppy application. In *2016 6th International Annual Engineering Seminar (InAES)*, pages 138–143, Aug 2016.

[33] 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.

[34] 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.

[35] Rongxing Lu, Xiaodong Lin, Zhiguo Shi, and X.S. Shen. A lightweight conditional privacy-preservation protocol for vehicular traffic-monitoring systems. *Intelligent Systems, IEEE*, 28(3):62–65, May 2013.

[36] Yipin Sun et al. An efficient pseudonymous authentication scheme with strong privacy preservation for vehicular communications. *Vehicular Technology, IEEE Transactions on*, 59(7):3589–3603, Sept 2010.

[37] Dijiang Huang. Pseudonym-based cryptography for anonymous communications in mobile ad hoc networks. *Int. J. Secur. Netw.*, 2(3/4):272–283, April 2007.

[38] L. L. Wang, G. Z. Liu, L. j. Sun, and Y. W. Lin. An effective pseudonym generating scheme for privacy and anonymity in vanets. In *2016 International Conference on Information System and Artificial Intelligence (ISAI)*, pages 267–270, June 2016.

[39] Giorgio Calandriello, Panos Papadimitratos, Jean-Pierre Hubaux, and Antonio Lioy. Efficient and robust pseudonymous authentication in vanet. In *Proceedings of the Fourth ACM International Workshop on Vehicular Ad Hoc Networks*, VANET '07, pages 19–28, New York, NY, USA, 2007. ACM.

[40] U. Rajput, F. Abbas, H. Eun, and H. Oh. A hybrid approach for efficient privacy-preserving authentication in vanet. *IEEE Access*, 5:12014–12030, 2017.

[41] Tao Jiang, Helen J. Wang, and Yih-Chun Hu. Preserving location privacy in wireless lans. In *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services*, MobiSys '07, pages 246–257, New York, NY, USA, 2007. ACM.

[42] Marco Gruteser and Dirk Grunwald. Enhancing location privacy in wireless lan through disposable interface identifiers: A quantitative analysis. *Mob. Netw. Appl.*, 10(3):315–325, June 2005.

[43] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matt Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, pages 41–55, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[44] Y. Zheng, H. Duan, X. Yuan, and C. Wang. Privacy-aware and efficient mobile crowdsensing with truth discovery. *IEEE Transactions on Dependable and Secure Computing*, PP(99):1–1, 2017.

[45] Chenglin Miao, Wenjun Jiang, Lu Su, Yaliang Li, Suxin Guo, Zhan Qin, Houping Xiao, Jing Gao, and Kui Ren. Cloud-enabled privacy-preserving truth discovery in crowd sensing systems. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, SenSys '15, pages 183–196, New York, NY, USA, 2015. ACM.

[46] Y. Zhang, Q. Chen, and S. Zhong. Efficient and privacy-preserving min and k th min computations in mobile sensing systems. *IEEE Transactions on Dependable and Secure Computing*, 14(1):9–21, Jan 2017.

[47] C. Miao, L. Su, W. Jiang, Y. Li, and M. Tian. A lightweight privacy-preserving truth discovery framework for mobile crowd sensing systems. In *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pages 1–9, May 2017.

[48] Y. Zheng, H. Duan, and C. Wang. Learning the truth privately and confidently: Encrypted confidence-aware truth discovery in mobile crowdsensing. *IEEE Transactions on Information Forensics and Security*, 13(10):2475–2489, Oct 2018.

[49] Y. Zhang, Q. Chen, and S. Zhong. Privacy-preserving data aggregation in mobile phone sensing. *IEEE Transactions on Information Forensics and Security*, 11(5):980–992, May 2016.

[50] Klaus Kursawe, George Danezis, and Markulf Kohlweiss. Privacy-friendly aggregation for the smart-grid. In *Proceedings of the 11th International Conference on Privacy Enhancing Technologies*, PETS'11, pages 175–191, Berlin, Heidelberg, 2011. Springer-Verlag.

[51] Shigeo Mitsunari, R Sakai, and M Kasahara. A new traitor tracing. E85-A, 02 2002.

[52] T. Hyla and J. Peja . Demonstrably secure signature scheme resistant to$k$-traitor collusion attack. *IEEE Access*, 6:50154–50168, 2018.

[53] Girraj Kumar and B.B. Singh. Efficient id-based blind message recovery signature scheme from pairings. *IET Information Security*, 12, 12 2017.

[54] Lu Yang and Jiguo Li. Efficient certificate-based signcryption secure against public key replacement attacks and insider attacks. *TheScientificWorldJournal*, 2014:295419, 05 2014.

[55] Raylin Tso, Xun Yi, and Xinyi Huang. Efficient and short certificateless signature secure against realistic adversaries. *The Journal of Supercomputing*, 55:173–191, 02 2011.

[56] Kyu Choi, Jong Park, Jung Hwang, and Dong Hoon Lee. *Efficient Certificateless Signature Schemes*, pages 443–458. 06 2007.

[57] Jin Zhou, YaJuan Zhang, and Yuefei Zhu. Security arguments for a class of id-based signatures. *IACR Cryptology ePrint Archive*, 2007:49, 01 2007.

[58] J. Camenisch. Group signature schemes and payment systems based on the discrete logarithm problem. 1998.

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

[60] Anna Lisa Ferrara, Matthew Green, Susan Hohenberger, and Michael Østergaard Pedersen. Practical short signature batch verification. Event Dates: April 20-24, 2009, April 2009.

[61] Mihir Bellare, Juan A. Garay, and Tal Rabin. *Fast batch verification for modular exponentiation and digital signatures*, pages 236–250. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

[62] Jan Camenisch, Susan Hohenberger, and Michael Østergaard Pedersen. *Batch Verification of Short Signatures*, pages 246–263. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[63] Jaroslaw Pastuszak, Dariusz Michałek, Josef Pieprzyk, and Jennifer Seberry. *Identification of Bad Signatures in Batches*, pages 28–45. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.

[64] Cheng-Kang Chu, Joseph K. Liu, Xinyi Huang, and Jianying Zhou. Verifier-local revocation group signatures with time-bound keys. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '12, pages 26–27, New York, NY, USA, 2012. ACM.

[65] Angelo De Caro and Vincenzo Iovino. jpbc: Java pairing based cryptography. In *Proceedings of the 16th IEEE Symposium on Computers and Communications, ISCC 2011*, pages 850–855, Kerkyra, Corfu, Greece, June 28 - July 1, 2011. IEEE.

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

[67] Haoyi Xiong, Daqing Zhang, Leye Wang, J. Paul Gibson, and Jie Zhu. Eemc: Enabling energy-efficient mobile crowdsensing with anonymous participants. *ACM Trans. Intell. Syst. Technol.*, 6(3):39:1–39:26, April 2015.

[68] Nordic Semiconductor. https://www.nordicsemi.com/eng/products/nordic-thingy-52.

XV. BIOGRAPHIES

Victor Sucasas obtained his Ph.D. on Electronic Engineering at University of Surrey (UK) in 2016. He has extensive research experience as a researcher at Instituto de Telecomunicações - Aveiro, Portugal and as a PhD student at University of Surrey, Guildford, UK, where he worked on European projects FP7-GREENET, ECSEL-SWARMs and CATRENE-H2O. Over the past 6 years, he has been an active researcher in several fields such as wireless cooperative communications, network security and privacy preserving systems.

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.

Joaquim Bastos received his Bachelors and M.Sc. in Electronics and Telecommunications engineering from the University of Aveiro (Portugal), in 1997 and 2006 respectively. From 1998 to 2002, he was a development manager at Philips Portugal and afterwards at Comverse Network Systems France. In 2003, he became a researcher at the Instituto de Telecomunicações, and participated in international research projects, such as FP6-IST's MATRICE, 4MORE, ORACLE, and as WP leader in FP7-ICT's WHERE and WHERE2, in Celtic's MOBILIA, in CATRENE's NewP@ss, and in ECSEL's SWARMs. He is author of several conference and journal publications, and his main research interests include: wireless communication systems, cognitive radio, IoT and network security.

Francisco Damião obtained his BSC in Electronics and Telecommunications Engineering in 1992 at Instituto Politecnico de Lisboa (ISEL)(PT) and his MSC in Information Technologies in 1994 at Universidade Nova de Lisboa FCT (PT). He has extensive industry expertise in designing, implementing, integrating and supporting large scale deployments of Information Technology projects covering several technology stacks. During the past years he has participated in European projects, under the frameworks ECSEL and CATRENE, as an expert on M2M/IoT, Low Power WAN Networks and Cloud Services XaaS fields.

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 Chartered Engineer and IET Fellow, and since 2017, he has been a Professor in Mobile Communities at the University of South Wales, Pontypridd, U.K.