Security, Privacy & Incentive Provision for Mobile Crowd Sensing Systems

Stylianos Gisdakis, Thanassis Giannetsos, Panos Papadimitratos
Networked Systems Security Group, KTH Royal Institute of Technology, Stockholm, Sweden
{*gisdakis*, *athgia*, *papadim*}@kth.se

***Abstract*—Recent advances in sensing, computing, and networking have paved the way for the emerging paradigm of Mobile Crowd Sensing (MCS). The openness of such systems and the richness of data MCS users are expected to contribute to them raise significant concerns for their security, privacy-preservation and resilience. Prior works addressed different aspects of the problem. But in order to reap the benefits of this new sensing paradigm, we need a holistic solution. That is, a secure and accountable MCS system that preserves user privacy, and enables the provision of incentives to the participants. At the same time, we are after a MCS architecture that is resilient to abusive users and guarantees privacy protection even against multiple misbehaving and intelligent MCS entities (servers). In this work, we meet these challenges and propose a comprehensive security and privacy-preserving architecture. With a full blown implementation, on real mobile devices, and experimental evaluation we demonstrate our system's efficiency, practicality, and scalability. Last but not least, we formally assess the achieved security and privacy properties. Overall, our system offers strong security and privacy-preservation guarantees, thus, facilitating the deployment of trustworthy MCS applications.**

***Index Terms*—Mobile Crowd Sensing, Security, Privacy, Incentive Mechanisms**

I. Introduction

Mobile Crowdsensing [1] (MCS) has emerged as a novel paradigm for data collection and collective knowledge formation practically about *anything*, from *anywhere* and at *anytime*. This new trend leverages the proliferation of modern sensing-capable devices in order to offer a better understanding of people's activities and surroundings. Emerging applications range from environmental monitoring [2, 3] to intelligent transportation [4, 5, 6] and assistive healthcare [7].

MCS users are expected to contribute sensed data tagged with spatio-temporal information which, if misused, could reveal sensitive user-specific information such as their whereabouts and their health condition [8, 9]. Even worse, data contributions are strongly correlated with the current user context (e.g., whether they are at home or at work, walking or driving, etc.); there is a significant risk of *indirectly* inferring daily routines or habits of users participating in MCS applications. By inferring user context, one can obtain deeper insights into individual behavior, thus, enabling accurate user profiling [10, 11]. As recent experience shows, assuming that users can simply trust the MCS system they contribute sensitive data to, is no longer a viable option. Therefore, it becomes imperative to ensure user *privacy* in mobile crowdsensing scenarios.

Furthermore, although privacy protection will facilitate user participation it cannot, per-se, ensure it. This is critical since if users do not engage in great numbers, thus, providing a sufficient influx of contributions, MCS systems will not succeed. In the absence of intrinsic motivation, providing *incentives* becomes vital [12]. Indeed, the research community has identified various forms of incentives based on monetary rewards [13], social or gaming-related mechanisms [14] along with methods for incorporating them in MCS systems [15, 16, 17, 18]. In particular, micro-payments have been shown effective in encouraging user participation and increasing their productivity [19].

However, the common challenge is providing incentives in a privacy-preserving manner; users should be gratified without associating themselves with the data they contribute. One possible solution the literature has proposed is the use of *reverse* auctions, among anonymous data providers and requesters [13, 20]. Such schemes necessitate user participation throughout the whole duration of a task. However, MCS users may join and leave sensing campaigns at any time, thus, making the implementation of such auction-based mechanisms impractical [16]. Moreover, the employed incentive provision methods must be fair: (selfish) users should not be able to exploit them and gain inordinate, to their contributions, utilities.

At the same time, aiming for the participation of any user possessing a sensing-capable device is a double-edged sword: participants can be adversarial seeking to manipulate (or even dictate) the MCS system output by *polluting* the data collection process. Even worse, detecting offending users and sifting their malicious contributions is hindered by the desired (for privacy-protection) user anonymity. What we need is mechanisms that can hold offending users accountable, but without necessarily disclosing their identity.

Motivation & Contributions: To reap the benefits of this new community sensing paradigm we must work towards three directions; *incentivizing user participation*, *protecting the users from the system* (i.e., ensuring their privacy) and, at the same time, *protecting the system from malicious users* (i.e., holding them accountable of possible system-offending actions). Despite the plethora of existing research efforts, the state-of-the-art in the area of secure and privacy-preserving MCS systems still lacks comprehensive solutions; most works either focus solely on user privacy without considering accountability or they facilitate incentive provision in a non-privacy-preserving manner (i.e., by linking users to their contributions). Therefore, the design of secure and privacy-preserving MCS systems, capable of insentivizing large-scale user participation, is the main challenge ahead.

To meet this challenge, we extend SPPEAR [21], the state-of-the-art security and privacy architecture for MCS systems to address: (*i*) security, (*ii*) privacy, (*iii*) accountability and (*iv*) incentive provision. More specifically, although SPPEAR offers broadened security and privacy protection under weak trust assumptions (where even system entities might try to harm user privacy), it does not capture the complete landscape of all possible privacy repercussions that such attacks entail. We also extend SPPEAR's simplistic receipt-based rewarding mechanism into a solution that fairly remunerates participating users while supporting different incentive mechanisms including, but not limited to, micro-payements. Overall, the suggested architecture provides high user-privacy assurance, while facilitating the ample participation of extrinsically motivated users.

We provide an implementation of our system on real mobile devices and extensively assess its efficiency and practicality. Furthermore, we present a formal analysis of the achieved security and privacy properties in the presence of strong adversaries. To better examine the privacy implications of such a broadened adversarial model, we also provide the first, to the best of our knowledge, instantiation of inference attacks (in the domain of MCS) that "honest-but-curious" system entities can launch against user privacy. More specifically, we show how such entities can extract sensitive user information (i.e., whereabouts, activities) by leveraging machine learning techniques and we discuss possible mitigation strategies.

The rest of this paper is organized as follows: Sec. II presents the related work in the area of secure and privacy-preserving MCS systems. We, then, describe the system and adversarial models for our scheme (Sec. III) and discuss the envisioned MCS security and privacy requirements (Sec. IV). In Sec. V, we provide an overview of the system and the services it offers followed by a detailed presentation of all implemented components and protocols (Sec. VI). Sec. VII presents a rigorous formal assessment of the achieved properties. The experimental setup, used to evaluate our system, along with the performance results are presented in Sec. VIII, before we conclude the paper in Sec. IX.

II. Related Work

The security and the privacy of MCS have attracted the attention of the research community [22, 8, 23]. Several works try to protect user privacy by *anonymizing* user contributed data [24, 25, 26] and *obfuscating* location information [27, 28, 29]. Additionally, other research efforts employ generalization [30] or perturbation [31, 32] of user contributions; i.e., deliberately reducing the quality and the quantity of the information users submit to the MCS system. Nevertheless, although such techniques can enhance user privacy they do not capture the full scope of privacy-protection; knowing that a user participates in sensing campaigns monitoring, for example, noise pollution during early morning hours already reveals sensitive information such as the coarse-grained location of her home [33]. Moreover, strong privacy-protection must hold even in the case that MCS system entities cannot be trusted: i.e., they are *curious* to learn and infer private user information.

AnonySense [24] is a general-purpose framework for secure and privacy-preserving tasking and reporting. Reports are submitted through wireless access points, while leveraging Mix Networks to de-associate the submitted data from their sources. However, the way it employs group signatures (i.e., [34]), for the cryptographic protection of submitted reports, renders it vulnerable to Sybil attacks (Sec. VII). Although AnonySense can evict malicious users, filtering out their faulty contributions requires the *de-anonymization* of benign reports[1]; besides being costly, this process violates the anonymity of legitimate participants. Misbehavior detection may occur even at the end of the sensing task when all contributions are available. On the contrary, our system shuns out offending users and sifts their malicious input through an efficient revocation mechanism (Sec. VI-D) that does not erode the privacy of benign users.

Group signature schemes can prevent anonymity abuse by limiting the rate of user authentications (and, thus, of the samples they submit), to a predefined threshold (k) for a given time interval [35]. Exceeding this threshold is considered misbehavior and results in de-anonymization and revocation. Nonetheless, this technique cannot capture other types of misbehavior, i.e., when malicious users pollute the collected data by submitting $(k-1)$ faulty samples within a time interval. In contrast, our scheme is *misbehavior-agnostic* and *prevents* such anonymity abuse by leveraging authorization tokens and pseudonyms with non-overlapping validity periods (Sec. VII).

PEPSI [25] prevents unauthorized entities from querying the results of sensing tasks with provable security. It leverages a centralized solution that focuses on the privacy of data queriers; i.e., entities interested in sensing information without considering accountability and privacy-preserving incentive mechanisms. PEPPeR [26] protects the privacy of the information querying nodes (and, thus, not of the information contributing nodes), by decoupling the process of node discovery from the access control mechanisms used to query these nodes. PRISM [36] focuses on the secure deployment of sensing applications and does not consider privacy.

In PoolView [31] mobile clients perturb private measurements before sharing them. To thwart inference attacks, leveraging the correlation of user data, the authors propose an obfuscation model. The novelty of this scheme is based on the fact that although private user data cannot be obtained, statistics over them can be accurately computed. PoolView considers only privacy of data streams and, thus, does not consider on accountability for misbehaving users.

In [37] the authors propose a privacy-preserving data reporting mechanism for MCS applications. The intuition behind this work is that user privacy is protected by breaking the link between the data and the participants. Nonetheless, opposite to our work, the proposed scheme solely focuses on privacy and, thus, does not consider incentive mechanisms and accountability for misbehaving users.

Addressing aspects beyond the scope of this work, in [38, 39] the authors propose a reputation-based mechanism for assessing the data-trustworthiness of user contributed data. Similarly, SHIELD [40] leverages machine learning techniques to detect and sift faulty data originating from adversarial users seeking to pollute the data collection process. In this work, we

[1] Submitted by users that belong to the same cryptographic group as the revoked ones.

assume the existence of such a scheme capable of assessing the overall contributions made by anonymous (for privacy-protection users).

A significant body of work in the area of MCS focuses on the provision of *incentives* to stimulate user participation [15, 41, 42, 20, 43, 19]. These works leverage mechanisms such as auctions, dynamic pricing, monetary coupons, service quotas and reputation accuracy. However, they do not consider user privacy and, thus, can leak sensitive information by linking the identity of users with the data they contribute. The approach presented in [44] user privacy by remunerating users according to their privacy exposure: as the privacy exposure of users increases, better services (e.g., QoS-wise) and rewards are offered to them as compensation.

III. System & Threat Model

System Model: We consider generic MCS systems comprising the following entities:

Task Initiators (TI), (Information Consumers): Organizations or individuals initiating data collection campaigns by recruiting users and distributing sensing tasks to them. The TI initiates sensing tasks and campaigns. Each task is essentially a specification of the sensors users must employ, the area of interest, and the lifetime of the task. The area of interest is the locality within which participating users must contribute data and it can be defined either explicitly (e.g., coordinates forming polygons on maps) or implicitly (through annotated geographic areas, e.g., Stockholm). In any case, it is divided into *regions* that can correspond to, for example, smaller administrative areas (e.g., municipalities) comprising the area of interest.

Users (Information Producers): Operators of sensing-capable mobile devices (e.g., smart-phones, tablets), and navigation modules (e.g., GPS). Devices possess transceivers allowing them to communicate over wireless local area (i.e., 802.11a/b/g/n) and (or) cellular networks (3G and LTE).

Back-end Infrastructure: System entities responsible for supporting the life-cycle of sensing tasks: they register and authenticate users, collect and aggregate user-contributed reports and, finally, disseminate the results (in various forms) to all interested stake-holders.

Threat Model: MCS can be abused both by *external* and *internal* adversaries. The former are entities without any established association with the system; thus, their disruptive capabilities are limited. They can eavesdrop communications in an attempt to gather information on user activities. They might also manipulate the data collection process by contributing unauthorized samples or replaying the ones of benign users. Nonetheless, such attacks can be easily mitigated by employing simple encryption and access control mechanisms. External adversaries may also target the availability of the system by launching, for example, jamming and (D)DoS attacks. However, such clogging attacks are beyond the scope of this work and, therefore, we rely on the network operators (e.g., Internet Service Providers (ISPs)) for their mitigation.

Internal adversaries are legitimate participants of the system that exhibit malicious behavior. We do not refer only to human operators with malevolent intentions but, more generally, to compromised devices (clients), e.g., running a rogue version of the MCS application. Such adversaries, can submit faulty, yet authenticated, reports during the data collection process. Their aim is to *distort* the system's perception of the sensed phenomenon, and thus, degrade the usefulness of the sensing task. For instance, in the context of traffic monitoring campaigns [4], malicious users might contribute false information (e.g., low velocities) to impose a false perception of the congestion levels of the road network. Such data pollution attacks can have far graver implications if malicious users impersonate other entities or pose with multiple identities (i.e., acting as a Sybil entity).

Internal adversaries may also have a strong motive to manipulate the incentive provision mechanism. For instance, leveraging their (for privacy protection) anonymity, they could try to increase their utility (e.g., coupons, receipts) without offering the required contributions.

At the same time, internal attacks can target user privacy, i.e., seek to identify, trace and profile users, notably through MCS-specific actions[2]. This is especially so in the case of *honest-but-curious* and *information-sharing* infrastructure components; i.e, entities (Sec. V) that execute the protocols correctly but are curious to infer private user data by (possibly) colluding with other entities in the system (Sec. VII-B).

IV. Security & Privacy Requirements

In this work, we aim for accountable yet privacy-preserving MCS architectures that can integrate advanced incentive mechanisms. Definitions of the expected security and privacy requirements follow:

- ***R1: Privacy Preserving Participation***: Privacy preservation in the context of MCS mandates that user participation is *anonymous* and *unobservable*. More specifically, users should contribute to sensing tasks without revealing their identity. Identities are both *user* (e.g., name, email address) and *device*-specific; e.g., device identifiers such as the International Mobile Subscriber Identity (IMSI) and the International Mobile Station Equipment Identity (IMEI).

Furthermore, external (e.g., cellular providers) or internal (i.e., MCS infrastructure entities or users) observers should not be able to infer that anonymous users have (or will) contribute to specific sensing tasks.

User-contributed data should be *unlinkable*: no entity having access to user reports (i.e., information users contribute to the MCS system) should be able to link reports to the users from which they originated or to infer whether two or more reports were contributed by the same user.

- ***R2: Privacy-Preserving & Fair Incentive Mechanisms***: Users should be rewarded for their participation without associating themselves to the data they contribute. Furthermore, incentive mechanisms must be *resilient*; misbehaving or selfish users should not be able to exploit them for increasing their utility without making the necessary contributions.

- ***R3: Communication Integrity, Confidentiality and Authentication***: All system entities should be authenticated and their communications should be protected from any alteration by and disclosure to unauthorized parties.

[2] For instance, user de-anonymization by examining the content of the reports they submit [24]

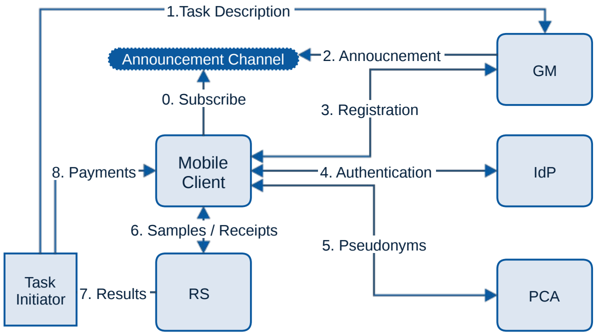


Fig. 1: System Overview

Notation	Meaning
TI	Task Initiator
GM	Group Manager
IdP	Identity Provider
PCA	Pseudonymous Certification Authority
RS	Reporting Service
RA	Resolution Authority
PK_x	Public key of authority X
PR_x	Private key of authority X
tr	Sensing task request
gsk_i	Group signing key
gpk	Group public key
PS	Pseudonym
t	Authorization token
$transient$	Transient SAML identifier
r	Report receipt
σ_X	Signature of authority X
ϕ_i	Shapley value of user i

TABLE I: Abbreviations & Notations

- ***R4: Authorization and Access Control***: Participating users should act according to the policies specified by the sensing task. To enforce such policies, *access control* and *authorization* mechanisms must be in place.
- ***R5: Accountability***: Offending users should be held accountable for any disruptive or system-harming actions.
- ***R6: Data Verification***: MCS systems must provide the necessary means to identify and sift faulty data originating from, potentially, misbehaving users.

V. System Entities

In this section, we begin with an overview of the system entities (Fig. 1) comprising our architecture and we, then, move on explaining how *trust relations* are established amongst them:

- **Mobile Client:** Users download a mobile client on their devices. This application collects and delivers sensed information by interacting with the rest of the infrastructure.
- **Group Manager (GM):** It is responsible for registering user devices to sensing tasks, issuing them anonymous credentials. The GM authorizes the participation of devices (in tasks) in an *oblivious manner*, using *authorization tokens*.
- **Identity Provider (IdP):** This entity authenticates user devices and mediates their participation to sensing tasks.
- **Pseudonym Certification Authority (PCA):** It provides anonymized ephemeral credentials (digital certificates), termed *pseudonyms*, to the users (mobile clients). Pseudonyms (i.e., the corresponding private/public keys) can cryptographically protect (i.e., ensure the integrity and the authenticity) information that clients submit. For *unlinkability* purposes, devices can obtain multiple pseudonyms from the PCA.

Reporting Service (RS): Mobile clients submit samples to this entity responsible for storing and processing the collected data. Although privacy-preserving data processing [45, 46]) could be employed, we neither assume nor require such mechanisms; this is orthogonal to our work and largely depends on the task/application. The RS issues *receipts* to participants later used for redeeming rewards.

- **Resolution Authority (RA):** This entity is responsible for revoking the anonymity of offending devices (e.g., devices that disrupt the system or pollute the data collection process).

Our goal is to separate functions across different entities, according to the *separation-of-duties* principle [47]: each entity is given the minimum information required to execute the desired task. This is to meet the requirements (Sec. IV) under weakened assumptions on system trustworthiness; in particular we achieve strong privacy protection even in the case of "*honest-but-curious*" infrastructure. Sec. VII further discusses these aspects.

Trust Establishment: To establish trust between system entities (Fig. 1), we leverage Security Assertion Markup Language (SAML) assertions that represent authentication and authorization claims, produced by one entity for another. To establish trust between the IdP and the PCA, a Web Service (WS)-Metadata exchange takes place. Metadata are XML-based entity descriptors containing information including authentication requirements, entity URIs, protocol bindings and digital certificates. The metadata published by the IdP contain the *X.509* certificates the PCA must use to verify the signatures of the assertions produced by the IdP. The PCA publishes metadata that contain its digital identifier and certificates.

To verify authorization tokens (Sec. VI-A), the IdP possesses the digital certificate of the GM. The pseudonyms issued to user devices are signed with the PCA private key. New tasks are signed by the TIs and verified by the GM. Finally, the RS possess the digital certificate of the PCA.

The confidentiality and the integrity of the communication is guaranteed by end-to-end authenticated Transport Layer Security (TLS) channels established between the devices and the MCS entities (i.e., IdP, PCA, RS). Furthermore, to prevent de-anonymyzation on the basis of network identifiers, mobile clients can interact with system entities via the TOR anonymization network [48].

VI. Preliminaries & System Protocols

As depicted in Fig. 1, the TI creates and signs task requests (tr) with a private key (PR_{TI}) of an ECDSA key-pair and sends them to the GM. The public key (PK_{TI}) is certified and known to the GM.

Upon reception of a tr, the GM *challenges* the TI with a random nonce to verify that it is actually the holder of the corresponding PR_{TI}. Then, the GM instantiates a *group signature* scheme that allows each participant (P_i) to anonymously authenticate herself with a private group signing (gsk_i). The GM pushes the group public key (gpk) to the IdP that is responsible for authenticating users.

Group signatures fall into two categories: static (fixed number of group members) and dynamic (dynamic addition of group

Algorithm 1 *Authorization Token* Acquisition

```
Initialization Phase(GM)
Data: $N$ generated authentication tokens

Begin
  1. GM $\twoheadrightarrow S : [\sqrt{N}, \sqrt{N}]$
  2. GM $\twoheadrightarrow 2\sqrt{N}$ random keys
     $(R_1, ..., R_{\sqrt{N}}), (C_1, ..., C_{\sqrt{N}})$,
     for each Row & Column
  3. for every $X_{i,j}$ in $S$ do
       GM $\twoheadrightarrow \{K_{i,j}, Y_{i,j}\}$, where
       $K_{i,j} = g^{R_i C_j}$, where
       $\{G_g, g\} \xrightarrow{\text{DDH}} \{Grp, Genr\}$
       $Y_{i,j} = commit_{K_{i,j}}(X_{i,j})$

     end
  3. GM sends to the device

       $Y_{1,1}, ..., Y\sqrt{N}, \sqrt{N}$

End
```

```
Transfer Phase(GM & DV)
Data: Computed token commitments $Y_{i,j}$

Begin
  1. GM $\twoheadrightarrow \{r_R, r_C\}$
  2. Randomize row & column keys:
       $(R_1 \cdot r_R, , ..., R_{\sqrt{N}} \cdot r_R)$
       $(C_1 \cdot r_C, ..., C_{\sqrt{N}} \cdot r_C)$
  3. If device wishes $X_{i,j}$
     then
       $OT_1^{\sqrt{N}}[GM, DV] \xrightarrow{\text{Pick}} R_i \cdot r_R$
       $OT_1^{\sqrt{N}}[GM, DV] \xrightarrow{\text{Pick}} C_j \cdot r_C$
     end

  4. GM sends $g^{\frac{1}{r_R r_C}}$
  5. Device reconstructs
       $K_{i,j} = g^{(\frac{1}{r_R r_C} R_i) \cdot r_R C_j \cdot r_C}$
  6. Obtain $X_{i,j}$ by opening $Y_{i,j}$ with $K_{i,j}$
End
```

participants). Selecting the appropriate scheme depends on the sensing task. For instance, sensing campaigns requiring the participation of only "premium" users can be accommodated by static group signature schemes since the number of participants is known. Otherwise, dynamic group signatures are necessary. Our system supports, but is not limited to, two schemes; *Short Group Signatures* [34] (static) and the *Camenisch-Groth* scheme [49] (dynamic).

Clients receive task descriptions (tr) through a Publish/Subscribe *announcement channel*. They can automatically connect (i.e., subscribe) and receive all task descriptors, tr, immediately after they are published by the GM. Each client can employ task filtering based on the device's current location so that users are presented with only those tasks for which they can accommodate the specified area of interest. If a user is willing to participate in a task, she authorizes her device to obtain the group credentials (i.e., gsk_i) and an *authorization token* from the GM (Sec. VI-A). Then, the device initiates the authentication protocol with the IdP and obtains pseudonyms from the PCA (Sec. VI-B). With these pseudonyms the device can (anonymously) authenticate the samples it submits to the task channel and receive the corresponding payment receipts (Sec. VI-C).

A. Registration & Authorization Token Acquisition

To participate in a sensing task, the mobile client registers with Group Manager (GM) to obtain the private group key key gsk_i by initiating an interactive $JOIN$ protocol with the GM.[3] This protocol guarantees *exculpability*: no entity can forge signatures besides the intended holder of the key (gsk_i) [50].

Subsequently, the GM generates an *authorization token dispenser*, D_{auth}. Each token of the dispenser binds the client identity with the identifier of each active task. This binding is done with secure and salted cryptographic hashes. Tokens are also signed by the GM to ensure their authenticity. More specifically, the dispenser is a vector of tokens, $D_{auth} = [t_1, t_2, ..., t_N]$, where each token, t_i, has the form:

$$t_i = \{t_{id}, h(user_{id} \,||\, task_i \,||\, n), task_i\}_{\sigma_{GM}}$$

where N is the number of currently active sensing tasks, n is a nonce, and t_{id} is the token identifier.

To participate in a task, the device must pick the corresponding token. Nevertheless, merely requesting a token would compromise users' privacy; besides knowing real user identity, the GM would learn the task she wishes to contribute to. For instance, knowing a user participates in a sensing task measuring noise pollution during night hours within an area "A", can help the GM deduce the user home location [51].

To mitigate this, we leverage *Private Information Retrieval (PIR)* techniques. Currently, our system supports the "*Oblivious Transfer with Adaptive Queries*" protocol [52]. The scheme has two phases (see Alg. 1): the initialization phase, performed by the GM, and the token acquisition phase involving both the device and the GM. For the former, the GM generates and arranges the N authorization tokens in a two-dimensional array, S, with $\sqrt{N}$ rows and $\sqrt{N}$ columns. Then, it computes $2\sqrt{N}$ random keys, $(R_1, R_2, ..., R_{\sqrt{N}}), (C_1, C_2, ..., C_{\sqrt{N}})$, and a commitment, $Y_{i,j}$, for each element of the array. These commitments are sent to the device.

During the token acquisition phase, the GM randomizes the $2\sqrt{N}$ keys with two elements r_R and r_C. Then, the device initiates two Oblivious Transfer sessions to obtain the desired token, $X_{i,j}$; one for the row key, $R_i \cdot r_R$, and another for the column key, $C_j \cdot r_C$. After receiving $g^{\frac{1}{r_R r_C}}$, from the GM, and with the acquired keys, the device can now obtain $X_{i,j}$ by opening the already received commitment, $Y_{i,j}$.

The security of this scheme relies on the Decisions Diffie-Helman assumption [52]. As the token acquisition protocol leverages oblivious transfer, the GM does not know which token was obtained and, thus, cannot deduce the task the user wishes to contribute to. In Sec. VIII we present a detailed performance analysis of the PIR scheme.

B. Device Authentication

Having the signing key, gsk_i, and the authorization token, t_i, the device can now authenticate itself to the IdP and receive pseudonyms from the PCA. Pseudonyms are X.509 certificates binding anonymous identities to public keys. Fig. 2 illustrates the protocol phases:

Phase 1: The mobile client generates the desired amount of key-pairs and creates the same number of Certificate Signing Requests (CSRs) (Step 1).

Phase 2: The client then submits the generated CSRs to the PCA to obtain pseudonyms (Step 2). Since the device is not yet authenticated, the PCA issues a SAML authentication request (Step 3) to the IdP, signed with its private key and encrypted with the public key of the IdP. SAML requires that requests contain a random *transient identifier* ($transient_{id}$) for managing the session during further execution of the protocol. The request is then relayed by the device to the IdP (Step 4), according to the protocol bindings agreed between the PCA and the IdP during the metadata exchange (Sec. V).

Phase 3: The IdP decodes and decrypts the authentication request, verifies the XML signature of the PCA and initiates the authentication process. As aforementioned, our authentication

[3] Due to space limitations, we refer the reader to [34, 49]

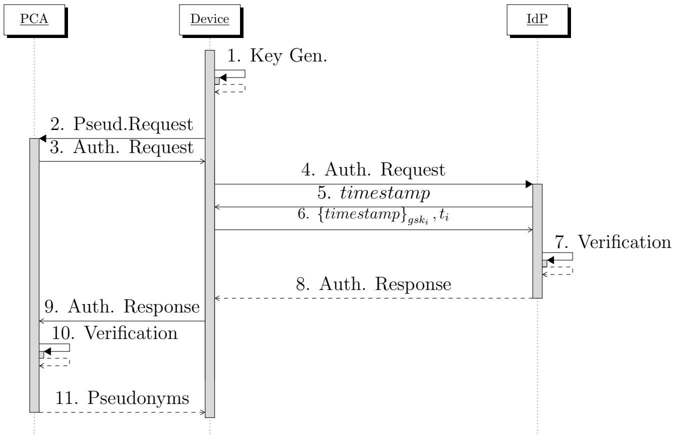


Fig. 2: Authentication Protocol

is based on group signatures. In particular, the IdP sends a challenge (in the form of a timestamp/nonce) to the device (Step 5). The device, then, produces a group signature on the challenge with its signing key gsk_i. It also submits the token, t_i, obtained by the GM (Step 6). The IdP verifies the challenge with the use of the gpk (obtained from the GM). Upon successful authentication (Step 7), the IdP generates a SAML authentication response signed with its private key and encrypted with the public key of the PCA. The response contains the $transient_{id}$ and an authentication statement (i.e., assertion): this asserts that the device was successfully authenticated (anonymously) through a group signature scheme and it includes the authorization token and the access rights of the device. Finally, the SAML response is encoded and sent back to the device (Step 8).

Phase 4: The device delivers the SAML assertion to the PCA (Step 9), which decrypts it and verifies its signature and fields (Step 10). Once the transaction is completed, the device is authenticated and it receives valid pseudonyms (Step 11).

Each pseudonym has a time validity that specifies the period (i.e., the pseudonym life time) for which it can be used. The PCA issues pseudonyms with non-overlapping life times (i.e., pseudonyms are not valid during the same time interval). Otherwise, malicious users could expose multiple identities simultaneously, i.e., launch *Sybil attacks*.

C. Sample Submission and Incentives Support

With the acquired pseudonyms, the device can now participate in the sensing task by signing the samples it contributes and attaching the corresponding pseudonym. More specifically, each sample, s_i, is:

$$s_i = \{v \,||\, t \,||\, (loc) \,||\, \sigma_{PrvKey} \,||\, C_i\}$$

where v is the value of the sensed phenomenon, t is a time-stamp and σ_{PrvKey} is the digital signature, over all the sample fields, generated with the private key whose public key is included in the pseudonym C_i. The loc field contains the current location coordinates of the device. In Sec. VII-C, we analyze the privacy implications due to device location in samples. Upon reception of a sample, the RS verifies its signature and time-stamp, against the time validity of the pseudonym. If the sample is deemed authentic, the RS prepares a receipt, r_i, for the device:

$$r_i = \{receipt_{id} \,||\, region_i \,||\, task_{id} \,||\, time \,||\, \sigma_{RS}\}$$

σ_{RS} is the digital signature of the RS. $region_i$ is the region (Sec. III) including the loc specified in the submission s_i. The device stores all receipts until the end of the task.

D. Pseudonym Revocation

If required, our system provides efficient means for shunning out offending users. Assume a device whose (anonymously) submitted samples significantly deviate from the rest. This could be an indication of misbehavior; e.g., an effort to *pollute* the results of the task. We refrain from discussing the details of such a misbehavior detection mechanism and we refer the reader to SHIELD [40], the state-of-the-art data verification framework for MCS systems. Misbehaving devices should be prevented from further contributing to the task. On the other hand, it could also be the case that the devices equipped with problematic sensors must be removed from the sensing task. To address the above scenarios, we design two grained revocation protocols, suitable for different levels of escalating misbehavior:

Total Revocation: The RA coordinates this protocol based on a (set of) pseudonym(s) PS_i (Fig. 3). Upon completion, the device owning the pseudonym is evicted from the system:

Phase 1: The RA provides the PCA with the PS_i (Step 1). The PCA, then, responds with the authorization token, t_i, included in the SAML assertion that authorized the generation of pseudonym PS_i (Step 2). This token is then passed by the RA to the GM (Step 3).

Phase 2: Based on the received t_i, the GM retrieves the whole token dispenser, D_{auth}, that included t_i. This dispenser is sent to the IdP (Step 4) that blacklists all its tokens and sends back a confirmation to the GM (Steps 5, 6). From this point on, the device can no longer get authenticated because all of its tokens were invalidated.

Phase 3: To revoke the already issued pseudonyms, the GM sends the dispenser, D_{auth}, to the PCA that determines which of these tokens it has issued pseudonyms for. It, then, updates its Certificate Revocation List (CRL) with all the not yet expired pseudonyms of the device (Steps 7, 8), forbidding it essentially from (further) submitting any samples to the RS.

Partial Revocation: This protocol evicts a device from a specific sensing task. The RA sends the pseudonym, PS_i, to the PCA, which retrieves the token, t_i, from the SAML assertion that authorized the issuance of PS_i. Consequently, the PCA revokes all the pseudonyms that were issued for t_i. As a device is issued only one token per task, and this is now revoked, the device can no longer participate in this specific task. The partial revocation protocol does not involve the GM and, thus, it does not revoke anonymity of devices.

E. Task Finalization & User Remuneration

Upon completion of the sensing task, our system remunerates users for their contribution. In case the remuneration mechanism mandates, for example, micro-payments, each task description (i.e., the corresponding tr) specifies the amount of remuneration, B, that users will share.

This process is initiated when the completion of the task is announced to the publish/subscribe channel (Sec. VI). Upon reception of this finalization message, participants provide the

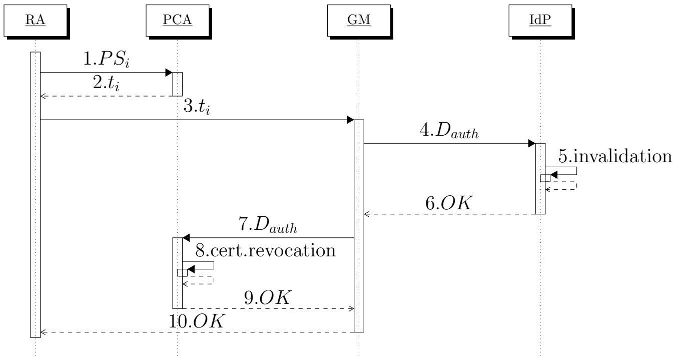


Fig. 3: Pseudonym Revocation

TI with all the receipts they collected for their data submissions (Sec. VI-C). The TI must then decide on a fair allocation of the tasks' remuneration amount (to the participating users) based on the level of contribution (i.e., number of submitted data samples) that each individual user had. To do this, we use Shapley value [53], an intuitive concept from coalitional game theory that characterizes fair credit sharing among involved players (i.e., users). This metric allows us to fairly quantify the remuneration of each user. Each user will be remunerated with an amount equal to $\phi_i \cdot B$. To compute ϕ_i the TI works as follows:

1) Shapley Value: Let N be the total number of participating users. For each subset of users (coalition) $S \subset N$, let $v(S)$ be a value describing the importance of the subset of users S.

For a value function v the Shapley value is a unique vector $\phi = [\phi_1(v), \phi_2(v), ..., \phi_N(v)]$ computed as follows:

$$\phi_i(v) = \frac{1}{|N|!} \sum_{\Pi} [v(P_i^{\Pi} \cup i) - v(P_i^{\Pi})] \tag{1}$$

where the sum is computed over all $|N|!$ possible orders (i.e., permutations) of users and P_i^{Π} is the set of users preceding user i in the order Π. Simply put, the Shapley value of each user is the average of her marginal contributions.

Computing the Shapley value for tasks with a large number of participants is computationally inefficient due to the combinatorial nature of the calculation. Nonetheless, an unbiased estimator of the Shapley value is the following [53]:

$$\hat{\phi_i}(v) = \frac{1}{k} \sum_{\Pi} [v(P_i^{\Pi} \cup i) - v(P_i^{\Pi})] \tag{2}$$

where k is the number of randomly selected user subsets (coalitions) to be considered; it essentially determines the error between the real value and its estimate.

2) Defining the value function v: Our goal is to remunerate users based not only on the number of their data submissions but also on the spatial dispersion of their contributions. Intuitively, this mechanism should favor reports submitted for regions where the system perception of the sensed phenomenon is low (i.e., less received data samples). On the other hand, the value accredited to similar, or possibly replayed (i.e., the same measurement for the same region), samples should be diminished.

To achieve this, we devise the value function, v, as follows: Let $R = [R_1, R_2, ..., R_N]$ be the number of receipts the TI receives from each user. The value $v(S)$ of a coalition S is computed as:

$$v(S) = H(R_S) \cdot \sum_{i \in S} R_i \tag{3}$$

R_S is the vector defining the number of samples this coalition has contributed for each region. For instance, let us assume a task for which the area of interest is divided into four regions $[reg_\alpha, reg_\beta, reg_\gamma, reg_\delta]$. Moreover, let S_2 be a coalition of two users each of which has submitted one sample to each of the regions. In this case, $R_S = [2, 2, 2, 2]$. $H(R_S)$ is Shannon's entropy:

$$H(R_S) = -\sum p_i \cdot log(p_i) \tag{4}$$

where p_i is the proportion of samples, conditional on coalition S, in region i. $H(R_S)$ is equal to 1 when all regions have received the same number of samples. In this case, the value of a coalition, $v(S)$, is the sum of samples that participating users contributed to the task. If a coalition is heavily biased towards some regions, then H tends to 0 and, thus, $v(S)$ will be equal to some (small) fraction of the sum of samples.

The above described remuneration protocol must be executed on top of a data verification mechanism, such as [40], that can detect and sift untrustworthy user contributions and, in combination with the revocation protocol (Sec. VI-D), evict malicious users without gratifying them.

VII. Security and Privacy Analysis

We begin with a discussion of the security and privacy of our system with respect to the requirements defined in Section IV. We then proceed with a formal security and privacy analysis.

Communications take place over secure channels (TLS). This ensures communication *confidentiality* and *integrity*. Furthermore, each system entity possesses an authenticating digital certificate (R_3).

In our scheme, the GM is the *Policy Decision Point*, which issues authorization decisions with respect to the eligibility of a device for a specific sensing task. The IdP is the *Policy Enforcement Point* which authorizes the participation of a device on the basis of authorization tokens (R_4).

Malicious devices can inject faulty reports to *pollute* the data collection process. For instance, consider a traffic monitoring task in which real-time traffic maps (of road networks) are built based on user submitted location and velocity reports. By abusing their anonymity or, if possible, by launching a Sybil attack, misbehaving users can impose a false perception over the congestion levels of the road network. Schemes (e.g., [24]) relying on group signatures for authenticating user reports are vulnerable to abuse: detecting if two reports were generated by the same device mandates the opening of the signatures of all reports, irrespectively of the device that generated them. Besides being costly[4], this approach violates the privacy of legitimate users.

We overcome this challenge with the use of authorization tokens: they indicate that the device was authenticated, for a given task, and that it received pseudonyms with non-overlapping lifetimes. This way, the PCA can corroborate the time validity of the previously issued pseudonyms and,

[4] Due to space limitations we refer the reader to [34]

Datum	Entity	Secrecy	Strong Secrecy/ Unlinakbility
Dev. id (id)	GM	✓	✓
Auth. Token (t)	IdP, PCA	✓	✓
Subm. sample. (s)	RS	✓	✓
Device pseud. (PS)	RS, PCA	✓	✓
Receipt (r)	RS	✓	✓

TABLE II: Secrecy Analysis for Dolev-Yao Adversaries

if requested by the device, provide it with new pseudonyms that do not overlap the previously issued ones. Thus, adversarial devices cannot exhibit Sybil behavior since they cannot use multiple pseudonyms simultaneously. Nevertheless, re-using pseudonyms for cryptographically protecting multiple reports, trades-off privacy (linkability) for overhead (Sec. VII-C).

The employed *Private Information Retrieval* scheme prevents a curious GM from deducing which task a user wishes to participate in. Moreover, devices get authenticated to the IdP without revealing their identity (i.e., group-signatures). Finally, pseudonyms allow devices to anonymously, and without being linked, prove the authenticity of the samples they submit. By using multiple pseudonyms (ideally one per report) and by interacting with the RS via TOR, devices can achieve enhanced report unlinkability. Furthermore, TOR prevents system entities and cellular ISPs from de-anonymizing devices based on network identifiers (R_1). Essentially, with end-to end encryption and TOR, our system prevents ISPs from gaining any additional information from the participation to a sensing task.

The first two columns of Table II present the information each system entity possesses. Our approach, based on the separation of duties principle, prevents single infrastructure entities from accessing all user-sensitive pieces of information (colluding system entities are discussed in Sec. VII-B).

The employed cryptographic primitives ensure that offending users cannot deny their actions. More specifically, the interactive protocols, executed during the registration phase (Sec. VI-A), guarantees that gsk_i is known only to the device and as a result, *exculpability* is ensured [34]. Furthermore, digital signatures are generated with keys known only to the device and thus, non-repudiation is achieved.

Our system can shun out offending devices (Sec. VI-D) without, necessarily, disclosing their identity (R_1, R_5). To achieve permanent eviction of misbehaving mobile clients the registration phase can be enhanced with authentication methods that entail network operators (e.g., GBA [4]). However, we leave this as a future direction.

We consider operation in semi-trusted environments. In particular, a PCA can be compromised and issue certificates for devices not authenticated by the IdP. If so, the PCA does not possess any SAML assertion for the issued pseudonyms, and thus, it can be held culpable for misbehavior. Moreover, the IdP cannot falsely authenticate non-registered devices: it cannot forge the authorization tokens included in the SAML assertions (Sec. VI-B). As a result, the PCA will refuse issuing pseudonyms and, thus, the IdP will be held accountable. Moreover, SAML authentication responses (Sec. VI-B) are digitally signed by the IdP and thus cannot be forged or tampered by malicious devices. Overall, in our system, one entity can serve as a witness of the actions performed by another; this way we establish a strong *chain-of-custody* (R_5).

A special case of misbehavior is when a malicious RS seeks to exploit the total revocation protocol (Sec. VI-D) to de-anonymize users. To mitigate this, we mandate that strong indications of misbehavior are presented to the RA before the resolution and revocation protocols is initiated. Nonetheless, such aspects are beyond the scope of this work.

Malicious users cannot forge receipts since they are signed by the RS. Furthermore, they are bound to specific tasks and thus they cannot be used to earn rewards from other tasks. Colluding malicious users might exchange receipts. Nevertheless, all receipts are invalidated, by the TI, upon submission and, thus, they cannot be “double-spent” (R_2).

Receipts, generated by the RS, are validated by the TI, neither of which knows the long-term identity of the user. As a result, the incentive mechanism protects user anonymity.

Finally, although our system does not assess the trustworthiness of user contributed data (i.e., R_6) it can seamlessly integrate data verification schemes, such as [40].

For the correctness of the employed cryptographic primitives (i.e., group signature, PIR schemes) we refer to [34, 49, 52]. In what follows, we focus on the secrecy and strong-secrecy properties of our system in the presence of external adversaries and infomration-sharing honest-but-curious system entities.

A. Secrecy against Dolev-Yao adversaries

We use ProVerif [54] to model our system in π-Calculus. System entities and clients are modeled as processes and protocols (i.e., authentication, Sec. VI-B, sample submission, Sec. VI-C, revocation, Sec. VI-D) are parallel composition of multiple copies of processes. ProVerif requires sets of *names* and *variables* along with a finite *signature*, Σ, comprising all the function symbols accompanied by their *arity*. The basic cryptographic primitives are modeled as symbolic operations over bit-strings representing messages encoded with the use of *constructors* and *destructors*. Constructors generate messages whereas destructors decode messages.

ProVerif verifies protocols in the presence of Dolev-Yao adversaries [55]: they can eavesdrop, modify and forge messages according to the cryptographic keys they possess. To protect communications, every emulated MCS entity in the analysis maintains its own private keys/credentials. This model cannot capture the case of curious and information-sharing MCS system entities (discussed in Sec. VII-B).

In ProVerif, the attacker's knowledge on a piece of information i, is queried with the use of the predicate $attacker(i)$. This initiates a resolution algorithm whose input is a set of Horn clauses that describe the protocol. If i can be obtained by the attacker, the algorithm outputs $true$ (along with a counter-example) or $false$ otherwise. ProVerif can also prove *strong-secrecy* properties; adversaries cannot infer changes of secret values. To examine if strong-secrecy properties hold for a datum i, the predicate *noninterf* is used. We evaluate the properties of all specific to our system data. Table II summarizes our findings: our system guarantees not only the secrecy but also the strong-secrecy of all critical pieces of information and, thus, it preserves user privacy.

Since Dolev-Yao adversaries cannot infer changes over the aforementioned data. For instance, adversaries cannot relate two

Honest-but-curious (colluding) entities	Information Linked	Privacy Implications
GM	-	No sensitive information can be inferred.
IdP	t	The IdP can simply infer that an anonymous user wishes to participate in a task.
PCA	PS, t	The PCA will infer that an anonymous user wishes to receive pseudonyms for a given task.
RS	s, PS, r	The RS knows that a given report was submitted for a specific sensing task.
GM, IdP	t, id	The GM and the IdP can infer that a user with a known identity wishes to participate to a specific task.
GM, PCA	t, id, PS	The GM and the PCA can infer that a user with a user with a known identity wishes to participate to a specific task and has received pseudonyms.
GM, RS	s, PS, r	When the GM and the RS collude they can infer that a report was submitted by a pseudonymous user.
IdP, PCA	t, PS	These authorities can infer that an anonymous user received pseudonyms for a specific task.
PCA, RS	t, PS, s, r	The PCA and the RS can infer that an anonymous user received pseudonyms for a specific task and has submitted a report.
GM, PCA, RS	all	Full de-anonymization of the user, the task she participates in and the reports she has submitted.

TABLE III: Honest-but-curious entities with ProVerif.

tokens, t_1 and t_2, belonging to the same user; the same holds for the other protocol-specific data (e.g., samples, receipts).

B. Honest-but-curious System Entities

We consider the case of colluding (i.e., information-sharing) honest-but-curious system entities aiming to infer private user information. We model such behavior in ProVerif by using a *spy channel*, accessible by the adversary, where a curious authority publishes its state and private keys. To emulate colluding infrastructure entities, we assume multiple spy channels for each of them. We set the adversary to $passive$: she can only read messages from accessible channels but not inject any message. For this analysis we additionally define the following functions in ProVerif:

$$MAP(x, y) = MAP(y, x)$$
$$LINK(MAP(x, a), MAP(a, y)) = MAP(x, y)$$

The first is a constructor stating that the function MAP is symmetric. The second is a destructor stating that MAP is transitive. For example, whenever the device submits an authorization token to the IdP it holds that $MAP(ANON\_USER_{\alpha}, token_x)$ (i.e., an anonymous user, α, wants to authenticate for task x). Of course, the GM (and, thus, the adversary listening to the spy channel in case the GM is honest-but-curious) also knows $MAP(token_x, USER_{\alpha})$. In case these two entities collude, querying $MAP(ANON\_USER_{\alpha}, USER_{\alpha})$ yields true; these colluding entities know that a user, with a known identity, participates in a task. Similarly we can issue other queries (e.g., $(MAP(USER_{\alpha}, PSEUDONYM_y)$, $MAP(USER_{\alpha}, REPORT_y))$. Table III presents the pieces of information that is known or can be inferred (along with their semantics) for various combinations of honest-but-curious colluding entities.

Single system entities cannot de-anonymize users as they have limited access to user information (Table II). Furthermore, our system is privacy-preserving even when two authorities collude. To completely de-anonymize users and their actions, it is required that the GM, the PCA and the RS collaborate. Of course, if these entities are deployed within different administrative domains, their collusion is rather improbable. Nonetheless, if they are within the same administrative domain, the separation-of-duties requirement no longer holds; thus, user privacy cannot be guaranteed.[5]

C. Pseudonyms and Protection

To evaluate the unlinkability achieved by pseudonyms, we consider the following MCS application: drivers, with the use of their smart-phones, report their current location and velocity to the RS. We assume that the RS is not trusted: it performs no aggregation or obfuscation of the submitted data but rather tries to create detailed location profiles for each vehicle, by linking successive location samples submitted under the same or different pseudonyms. Various techniques leveraging location information and mobility can simulate such attacks. Here we emulate such adversarial behavior with a Kalman filter tracker. We consider 250 vehicles and a geographic area of 105 urban road links in the city of Stockholm. We generate mobility traces with the SUMO [4] microscopic road traffic simulator. Our aim is to understand the privacy implications of varying pseudonym utilization policies. In Fig. 4 (a), we plot the fraction of vehicles that our tracker tracked for more than 50% of their trip, as a function of the report submission frequency (from $10\,s$ to $5\,min$ period interval) for different pseudonym (re)usage policies, i.e., the number of reports signed under the same pseudonym.

The tracker tracks 37% of the vehicles[6] for a reporting frequency of $10\,s$ and a use of 1 pseudonym per report (maximum unlinkability). Nonetheless, its success decreases for more realistic reporting frequencies: the tracker receives less corrections and, thus, produces worse predictions. On the other hand, using the same pseudonym for multiple samples trades-off privacy for overhead (but not significantly). For a sampling frequency of $1\ report/min$, we observe that approximately 5% of the vehicles are tracked for more than 50% of their trips. Similarly, by reusing the same pseudonym for 5 samples, 27% of the vehicles are tracked for more than 50% of their trips. Overall, the effect of pseudonym reuse weakens as the sampling frequency decreases to frequencies relevant to the MCS context, i.e., $1\ report/30s$.

In Fig. 4 (b), we show that as the number of users increases, so does the overall privacy offered by pseudonyms. For instance, for 100 simulated vehicles, with a sampling rate of $10\,s$, and changing pseudonyms every 10 samples, we see that almost 100% of all vehicles can be tracked for more than 50% of their trips. Nonetheless, as the population of participating vehicles grows, the tracker's accuracy deteriorates because the RS receives more location samples and, thus, the probability

[5] Please note that any distributed architecture would fail to preserve privacy in this scenario.

[6] Please note that the regularity of vehicular movement works in favor of the tracker.

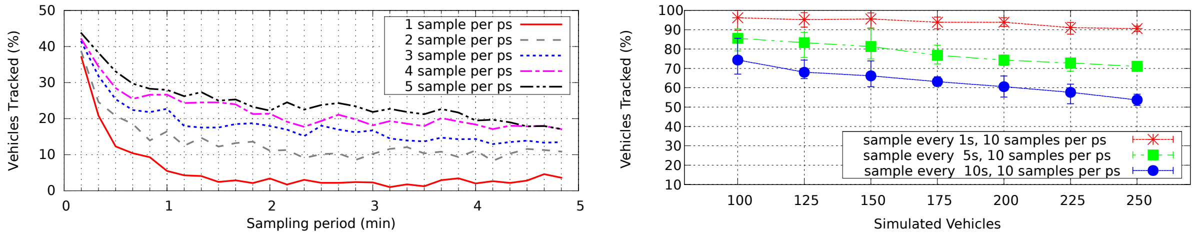
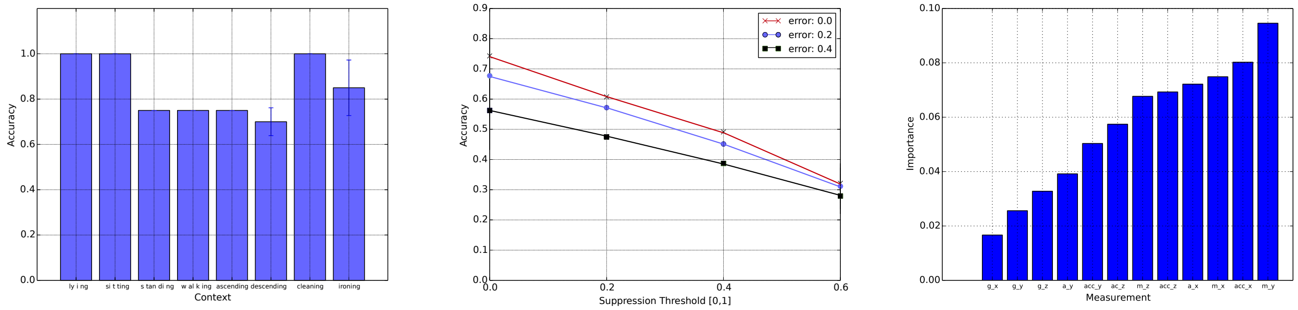


Fig. 4: Privacy Evaluation for Mobility: Impact of sampling rate (a) and population (b).



(a) (b) (c)

Fig. 5: Inferring User Context: (a,b) Classification Accuracy, (c) Sensor Evaluation.

of erroneously linking two successive samples also increases. Simply put, *users can better hide inside large crowds*.

D. Inferring User Context from Sensor Readings

For this analysis we assume the worst-case scenario in terms of privacy: we assume that user samples are linked and this linking is facilitated by the limited user mobility (e.g., being at home) and by the fact that they submit multiple samples under the same pseudonym. The honest-but-curious RS might attempt to infer the user context (i.e., activities: walking, driving, sleeping) from those linked sensor readings [10, 11]. The rest of this section discusses instantiations of such privacy attacks and evaluates the effectiveness of different mitigation strategies.

1) Adversarial Instantiation: We leverage machine learning mechanisms for predicting the user context. More specifically, we assume that an honest-but-curious RS has a statistical model of possible sensor values characterizing different user contexts. Such knowledge can be obtained by, e.g., user(s) cooperating with the RS. What the RS essentially needs is labeled training sets: values from various sensors (e.g., accelerometer) mapped to specific contexts or activities.

After obtaining training sets, the honest-but-curious RS instantiates an *ensemble* of classifiers to predict the context of the participating users. For the purpose of this investigation, we use *Random Forests*: collections of decision trees, each trained over a different bootstrap sample. A decision tree is a classification model created during the exploration of the training set. The interior nodes of the tree correspond to possible values of the input data. For instance, an interior node could describe the values of a sensor s_1. Nodes can have other nodes as children, thus, creating decision paths (e.g., $s_1 > \alpha$ *and* $s_2 < \beta$). Tree leafs mark decisions (i.e., classifications) of all training data described by the path from the root to the leaf. For example, samples for which sensors $s1$ and $s2$ take the values $s_1 > \alpha$, $s_2 < \beta$ describe the *walking* activity. After training, the RS can classify user contexts based on the sensor values sent by their mobile clients.

2) Attack Evaluation and Mitigation Strategies: For the analysis, we employ the PAMAP[7] dataset which contains sensor readings (i.e., accelerometer, gyroscope, magnetometer) from 17 subjects performing 14 different activities (e.g., walking, cycling, laying, ironing, computer work). We consider only a subset of the included sensor types focusing on those that already available in current smart-phones: temperature (Samsung Galaxy S4 has a dedicated temperature sensor), accelerometer, gyroscope and magnetometer. For each evaluation scenario, we select one subject (at random) for training the classifier ensemble and, then, examine its accuracy for the rest of the dataset subjects. We additionally consider two of the most well-know mitigation strategies against such inference attacks, and assess their effectiveness: (*i*) suppressing sensor readings (i.e., contributing samples according to some probability) and (*ii*) introducing noise to the submitted measurements.

As shown in Fig. 5 (a), the overall ensemble classification accuracy (for different user contexts) is above 50%. This serves as an indication that an honest-but-curious RS can effectively target user contextual privacy. Fig. 5 (b) illustrates the classification accuracy when one of the previously described mitigation strategies is employed. In particular, we assume that users can either introduce some kind of error to their submitted measurements or decide, according to some probability (i.e., suppression threshold), whether to submit a sample or not. What we see is that when the suppression probability increases, the accuracy of the classifier decreases. This is to be expected because the classifier receives less samples and, thus, produces worse predictions. Moreover, as the figure shows, introducing noise in the data samples can also improve user privacy.

[7] http://www.pamap.org/demo.html

Not submitting enough samples results in the accumulation of fewer receipts by the client (Sec. VI-C): simply put, this strategy trades-off rewards and credits for better privacy protection. At the same time, anomaly detection mechanisms can *flag* samples to which an error has been deliberately introduced. Overall, although orthogonal to this work, the fine-tuning of these (or similar) mitigation strategies merits further investigation.

Fig. 5 (c) presents the informativeness of the employed sensor types with respect to user contexts. We express "sensor importance" as the (normalized) total reduction of uncertainty brought by that feature (i.e., the Gini importance [56]). As it can be seen, magnetometers and gyroscopes are the most intrusive sensors as they reveal the most about a user's context. By leveraging such knowledge, participating users can have an estimation on their (possible) privacy exposure prior to their participation in a sensing task; simply by examining the types of sensors the task requires.

VIII. System Performance Evaluation

A. System Setup

The IdP, GM, PCA, and RA are deployed, for testing purposes, on separate Virtual Machines (VMs) with dual-core 2.0 GHz CPUs. We distribute the services provided by PCA over two VMs for our dependability evaluation (the same can be applied to the other entities, but we omit the discussion due to space limitations). We use the OpenSSL library for the cryptographic operations, i.e., the Elliptic Curve Digital Signature Algorithm (ECDSA) and TLS, and the JPBC library for implementing the group signature schemes. We deployed the sensing application on different Android smart-phones with 4-Cores/1 GB RAM and 2-Cores/1 GB RAM. For the evaluation of Sec. VIII-C we employ Jmeter to emulate multiple devices accessing the infrastructure concurrently.

For sample submission and verification, we employ the ECDSA with keys computed over 224 bit prime fields (*secp224k1* curve), thus, achieving a 112 bit security [57].

B. User-Side Evaluation

Figure 6 illustrates the performance of the authentication and pseudonym acquisition protocol (Sec. VI-B) on the two mobile devices. For this evaluation, devices request one authorization token from a set of 10 (i.e., 10 active tasks). We present the time needed to execute the different steps of the algorithm (i.e., pseudonym generation, acquisition time and authentication at the IdP), averaged over 50 observations. For the dual-core phone, the time needed to get authenticated and obtain 10 pseudonyms is around $8\,s$. This increases linearly as the device requests more pseudonyms: for 50 pseudonyms, the authentication protocol is executed in $22\,s$. On the IdP site, authentication (based on group signatures) requires $4\,s$. For the quad-core device, the protocol requires significantly less time ($11\,s$ for 50 pseudonyms). When using TOR, we experience additional network latency. Due to space limitations, we present here the results only for the quad-core device. TOR introduces a latency of approximately $10\,s$, thus raising the authentication time to 23 s for 50 pseudonyms. Even for substantial reporting (task) periods such a number of pseudonyms provides adequate privacy (Sec. VII-C).

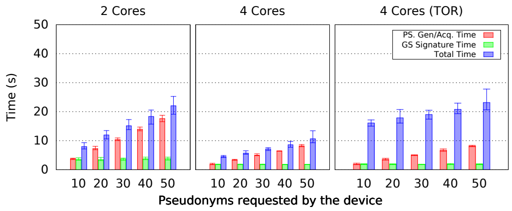


Fig. 6: Authentication Protocol

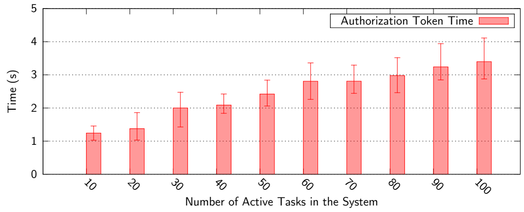


Fig. 7: Token Acquisition Time

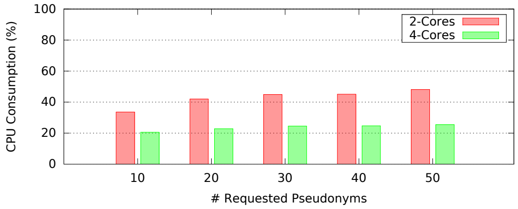


Fig. 8: CPU Consumption

We also compare the efficiency of EC-based digital signatures with group-signature schemes. This comparison yields that ECDSA with SHA512 is approximately 10 times faster (on the quad-core device) compared to group signature schemes (i.e., BBS scheme [34]) with the same security level. This serves as a comparison of our system with AnonySense [24] that relies on group signatures: as devices are expected to submit a considerable amount of digitally signed samples, it is critical, from the energy consumption point of view, that the process is as efficient as possible.

Figure 7 evaluates the implemented PIR scheme: we show the time needed to obtain an authorization token (for one task) on the quad-core device, as a function of the number of active tasks. This delay increases mildly with the number of active tasks in the system. Even for a set of 100 active tasks, the PIR protocol is executed in approximately $3.5\,s$.

We measure CPU utilization for the authentication protocol on the two mobile devices (Figure 8). For the dual-core device, the amount of CPU consumed ranges from 36%, for 10 pseudonyms, to approximately 50% for 50 pseudonyms. For the quad-core phone, the CPU consumption significantly drops, ranging from 20%, for 10 pseudonyms, to 23% for 50 pseudonyms. For comparison purposes, we measured the CPU consumption of the Facebook application on the quad-core device. On average the Facebook client consumes 18% of the CPU, which is close to the CPU consumption of our client on the same device (for 50 pseudonyms).

C. Infrastructure-Side Evaluation

We assess the performance of our infrastructure under stressful, yet realistic, scenarios assuming a traffic monitoring

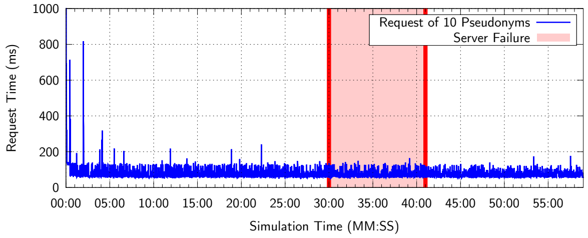


Fig. 9: System Reliability in Real-World Scenario

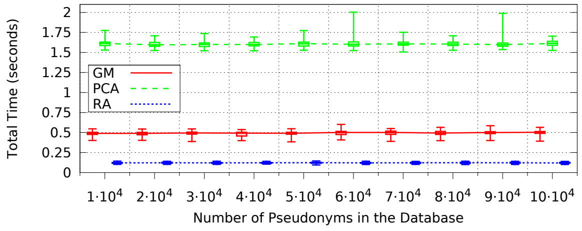


Fig. 10: Device Revocation

sensing task: the mobile devices of drivers get authenticated to our system and receive pseudonyms to protect and submit data with respect to the encountered traffic conditions. This is a demanding case of MCS since it entails strict location-privacy protection requirements: users request many pseudonyms to protect their privacy while submitting frequent location samples.

To emulate this task, we use the "TAPAS" data set [58] that contains synthetic traffic traces from the city of Cologne (Germany) during a whole day. We assume a request policy of 10 pseudonyms every 10 minutes, i.e. pseudonym lifetime of 1 minute each. By combining this policy with $5\,000$ randomly chosen vehicular traces from the data set, we create threads for Jmeter. Each thread is scheduled according to the TAPAS mobility traces, with journeys specified by start and end timestamps. Figure 9 shows that our system is efficient in this high-stress scenario: it serves each request, approximately, in less than $200\,ms$. Furthermore, during the 1 $hour$ execution of this test, we simulate an outage of one of the two PCAs lasting 11 minutes. As shown in the shaded area of Figure 9, the request latency does not increase and the system recovers transparently from the outage.

Figure 10 shows the time required for a single device revocation, as a function of the number of pseudonyms in the database. The RA queries the PCA for the authorization token that the device used to request the pseudonym PS. After retrieving the token, the RA asks the GM to translate it to the device long term identifier. Then, the GM invalidates all the dispenser corresponding to the token and informs the IdP (Section VI-D). Accordingly, the PCA revokes all device pseudonyms. These two processing delays are accounted for as the time spent on PCA (t_{PCA}) and GM (t_{GM}), respectively. The total time spent on RA is $t_{\text{TOT}} = t_{\text{RA}} + t_{\text{PCA}} + t_{\text{GM}}$, where t_{TOT} is the total execution time of revocation protocol.

The pseudonym set is generated by assuming the same request policy for all devices. This maximizes the entropy of the database set. Each assumed device obtained 10 tokens for requesting a set of 10 pseudonyms per token, thus giving the overall ratio 1 device : 10 tokens : 100 pseudonyms. The box-plots in Figure 10 depict the results averaged over 100 runs, with the pseudonym set increasing from $10\,000$ to $100\,000$ items linearly (i.e., we assume more devices). The performance of the system is not significantly affected by the size of the set. On average, revocation of a device requires $2.3\,s$.

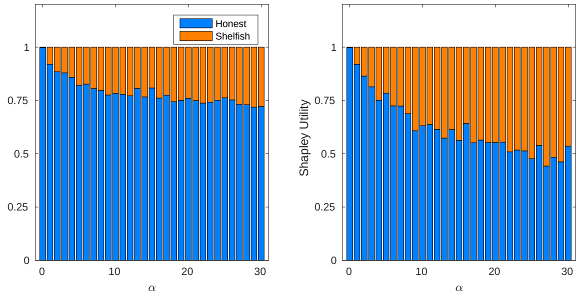


Fig. 11: Shapley Utility for different fraction of selfish users (10% and 30%)

D. Remuneration Evaluation

We evaluate the proposed remuneration mechanism for the unbiased estimator of Eq. 2. We start by assessing the mechanism fairness assuming sensing tasks with two user types: honest users monitoring the sensed phenomenon and submitting samples as they move along different regions and, selfish users that obtain a single measurement, for a single region, and massively replay it to the RS. This way, selfish users try to gain inordinate, to their efforts, rewards: although they do not spend resources for sensing the monitored phenomenon and their location, they submit samples to the RS. To emulate such a greedy deviant behavior, we synthetize a dataset of 40 users participating in a sensing task. Fig. 11 presents our findings: the left plot corresponds to a scenario where the fraction of selfish users is 10%. The parameter α of the $x - axis$ is the replaying frequency of selfish users: for $\alpha = 20$ malicious users submit (i.e., replay) 20 times more messages than an honest user to the RS. As the figure shows, even for the extreme case that $\alpha = 30$, selfish users receive, on average, 25% of the total value; this is a result of the employed value function (Sec. VI-E2): coalitions in which selfish users participate are unbalanced; they contain many reports for one region, and are, thus, evaluated lower ($H \to 0$) compared to more balanced coalitions. Increasing the amount of selfish users to 30% yields higher utility for them but, still, disproportional to the number of reports they replay to the system.

Indeed, selfish users could become malicious and spoof their device location, thus, submitting reports for regions they are not physically present. Mitigating such behavior is orthogonal to this investigation since it necessitates a data trustworthiness and verification framework such as [40] or position verification and location attestation schemes. Furthermore, shelfish users can also share measurements: a user in region A might receive measurements from another user, for a region B, and submit it to the system as hers (and vice versa). This behavior can be easily mitigated due to the our sybil-proof scheme: simply examining the distance between samples submitted under the same pseudonym serves as an indication of such attacks (i.e., when the corresponding distances are implausible).

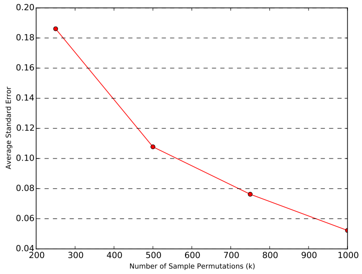


Fig. 12: Standard error of the estimator

Fig. 12 assesses the accuracy of the Shapley Estimator as a function of the of number of sample permutations (variable k of Eq. 2) for a sensing tasks with 40 participating users. As we do not have any ground-truth we plot on the $y-axis$ the standard statistical error of the values the estimator assigns to the different users. We observe that the relative standard error is small. Moreover, as we trade-off efficiency for accuracy (i.e., increasing k), the error significantly diminishes.

IX. Conclusions

Technological advances in sensing, microelectronics and their integration in everyday consumer devices laid the groundwork for the rise of people-centric sensing. However, its success requires effective protocols that guarantee security and privacy for MCS systems and their users. To meet this challenge, we presented a novel secure and accountable MCS architecture that can safeguard user privacy while supporting user incentive mechanisms. Our architecture achieves security, privacy and resilience in the presence of strong adversaries. Moreover, it enables the provision of incentives in a privacy-preserving manner; a catalyst for user participation. We formally evaluated the achieved security and privacy properties and provided a full-blown implementation of our system on actual devices.

References

[1] B. Guo et al. "From Participatory Sensing to Mobile Crowd Sensing". In: *Proceedings of the IEEE PERCOM Workshops*. March 2014, pp. 593–598.

[2] M. V. Kaenel, P. Sommer, and R. Wattenhofer. "Ikarus: Large-scale Participatory Sensing at High Altitudes". In: *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications*. Phoenix, USA, 2011.

[3] D. Mendez et al. "P-Sense: A Participatory Sensing system for air pollution monitoring & control". In: *IEEE International Conference on Pervasive Computing and Communications (PerCom)*. Seattle, 2011.

[4] S. Gisdakis et al. "Secure & Privacy-Preserving Smartphone-Based Traffic Information Systems". In: *IEEE Transactions on ITS* (2015), pp. 1428–1438.

[5] A. Thiagarajan et al. "VTrack: Accurate, Energy-aware Road Traffic Delay Estimation Using Mobile Phones". In: *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*. Berkeley, USA, 2009.

[6] S. Gisdakis et al. "SEROSA: SERvice oriented security architecture for Vehicular Communications". In: *IEEE Vehicular Netw Conf.* Boston, USA, 2013, pp. 111–118.

[7] T. Giannetsos, T. Dimitriou, and N. R. Prasad. "People-centric sensing in assistive healthcare: Privacy challenges and directions". In: *Security and Communications Network* 4.11 (Nov. 2011), pp. 1295–1307.

[8] D. Christin et al. "A Survey on Privacy in Mobile Participatory Sensing Applications". In: *J. Syst. Softw.* 84.11 (2011), pp. 1928–1946.

[9] Ioannis Krontiris, Felix Freiling, and Tassos Dimitriou. "Location privacy in urban sensing networks: research challenges and directions". In: *IEEE Wireless Communications* 17.5 (Oct. 2010), pp. 30–35.

[10] Michaela Götz, Suman Nath, and Johannes Gehrke. "MaskIt: Privately Releasing User Context Streams for Personalized Mobile Applications". In: *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. Scottsdale, Arizona, USA, 2012.

[11] Abhinav Parate et al. "Leveraging Graphical Models to Improve Accuracy and Reduce Privacy Risks of Mobile Sensing". In: *11th ACM Int. Conference on Mobile Systems, Applications, and Services*. Taipei, Taiwan, 2013, pp. 83–96.

[12] L.G. Jaimes, I.J. Vergara-Laurens, and A. Raij. "A Survey of Incentive Techniques for Mobile Crowd Sensing". In: *Internet of Things Journal, IEEE* 2.5 (2015), pp. 370–380.

[13] Dejun Yang et al. "Crowdsourcing to Smartphones: Incentive Mechanism Design for Mobile Phone Sensing". In: *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*. Mobicom '12. Istanbul, Turkey: ACM, 2012, pp. 173–184.

[14] Boya Di et al. "Incentive mechanism for collaborative smartphone sensing using overlapping coalition formation games". In: *2013 IEEE Global Communications Conference, GLOBECOM 2013, Atlanta, GA, USA, December 9-13, 2013*. 2013, pp. 1705–1710.

[15] B. Kantarci and H.T. Mouftah. "Trustworthy Sensing for Public Safety in Cloud-Centric Internet of Things". In: *Internet of Things Journal, IEEE* 1.4 (2014), pp. 360–368.

[16] Xinglin Zhang et al. "Free Market of Crowdsourcing: Incentive Mechanism Design for Mobile Sensing". In: *IEEE Trans. Parallel Distrib.* 25.12 (2014), pp. 3190–3200.

[17] I. Koutsopoulos. "Optimal incentive-driven design of participatory sensing systems". In: *INFOCOM, 2013 Proceedings IEEE*. 2013, pp. 1402–1410.

[18] Long Pham et al. "Multi-attribute online reverse auctions: Recent research trends". In: *European Journal of Operational Research* 242.1 (2015), pp. 1–9.

[19] S. Reddy et al. "Examining Micro-payments for Participatory Sensing Data Collections". In: *ACM International Conference on Ubiquitous Computing*. UbiComp. Copenhagen, Denmark, 2010.

[20] I. Krontiris and A. Albers. "Monetary incentives in participatory sensing using multi-attributive auctions". In: *International Journal on Parallel Emerging Distributed Systems* 27.4 (2012), pp. 317–336.

[21] S. Gisdakis, T. Giannetsos, and P. 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. Oxford, United Kingdom, 2014.

[22] Delphine Christin. "Privacy in mobile participatory sensing: Current trends and future challenges". In: *Journal of Systems and Software* (2015).

[23] A. Kapadia, D. Kotz, and N. Triandopoulos. "Opportunistic Sensing: Security Challenges for the New Paradigm". In: *Proceedings of the International Conference on COMmunication Systems And NETworks*. Bangalore, India, 2009.

[24] M. Shin et al. "AnonySense: A system for anonymous opportunistic sensing." In: *Pervasive and Mobile Computing* 7.1 (2011), pp. 16–30.

[25] E. De Cristofaro and C. Soriente. "Extended Capabilities for a Privacy-Enhanced Participatory Sensing Infrastructure (PEPSI)". In: *IEEE Transactions on Information Forensics and Security* 8.12 (2013), pp. 2021–2033.

[26] T. Dimitriou, I. Krontiris, and A. Sabouri. “PEPPeR: A querier’s Privacy Enhancing Protocol for PaRticipatory sensing”. In: *Security and Privacy in Mobile Information and Communication Systems*. Springer, 2012, pp. 93–106.

[27] I. Boutsis and V. Kalogeraki. “Privacy Preservation for Participatory Sensing Data”. In: *IEEE Conference on Pervasive Computing and Communications (PerCom)*. 2013.

[28] C.Y. Chow, M. Mokbel, and X. Liu. “Spatial cloaking for anonymous location-based services in mobile P2P environments.” In: *GeoInformatica* 15.2 (2011), pp. 351–380.

[29] S. Gao et al. “TrPF: A Trajectory Privacy-Preserving Framework for Participatory Sensing”. In: *IEEE Transactions on Information Forensics & Security* 8.6 (2013), pp. 874–887.

[30] L. Kazemi and C. Shahabi. “TAPAS: Trustworthy privacy-aware participatory sensing”. In: *Knowledge and Information Systems* 37.1 (2013), pp. 105–128.

[31] Raghu K. Ganti et al. “PoolView: Stream Privacy for Grassroots Participatory Sensing”. In: *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*. SenSys ’08. Raleigh, NC, USA, 2008.

[32] Shunsuke Aoki et al. “Perturbation with General Tendency for Mobile Community Sensing”. In: *Proceedings of the 2013 IEEE Second International Conference on Mobile Services*. IEEE Computer Society. 2013, pp. 23–30.

[33] T. Giannetsos, S. Gisdakis, and P. Papadimitratos. “Trustworthy People-Centric Sensing: Privacy, Security and User Incentives Road-map”. In: *IEEE 13th Mediterranean Ad Hoc Networking Workshop (Med-hoc-Net)*. Piran, Slovenia, 2014.

[34] D. Boneh, X. Boyen, and H. Shacham. “Short group signatures”. In: *Int. Cryptology Conference (CRYPTO)*. 2004.

[35] J. Camenisch et al. “How to win the clonewars: efficient periodic n-times anonymous authentication”. In: *ACM 13th conference on Computer and Communications Security*. New York, USA, 2006.

[36] T. Das et al. “PRISM: platform for remote sensing using smartphones”. In: *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*. San Francisco, USA, 2010.

[37] Yonglei Yao, L.T. Yang, and N.N. Xiong. “Anonymity-Based Privacy-Preserving Data Reporting for Participatory Sensing”. In: *Internet of Things Journal, IEEE* 2.5 (2015), pp. 381–390.

[38] D. Christin et al. “IncogniSense: An anonymity preserving reputation framework for participatory sensing.” In: *Pervasive and Mobile Comp.* 9.3 (2013), pp. 353–371.

[39] Xinlei Oscar Wang et al. “ARTSense: Anonymous reputation and trust in participatory sensing.” In: *32nd Int. Conference on Computer Communications*. Turin, Italy, 2013.

[40] S. Gisdakis, T. Giannetsos, and P. Papadimitratos. “SHIELD: A Data Verification Framework for Participatory Sensing Systems”. In: *ACM Conference on Security & Privacy in Wireless and Mobile Networks*. New York, 2015.

[41] E. Androulaki et al. “Reputation systems for anonymous networks”. In: *Privacy Enhancing Technologies*. 2008.

[42] T. Luo and C. K. Tham. “Fairness and social welfare in incentivizing participatory sensing.” In: *IEEE Conf. on Sensor, Mesh & Ad Hoc Com. and Networks*. Seoul, 2012.

[43] J. S. Lee and B. Hoh. “Dynamic pricing incentive for participatory sensing.” In: *Journal of Pervasive and Mobile Computing* 6.6 (2010), pp. 693–708.

[44] Buster O. Holzbauer, Boleslaw K. Szymanski, and Eyuphan Bulut. “Socially-aware Market Mechanism for Participatory Sensing”. In: *Proceedings of the First ACM International Workshop on Mission-oriented Wireless Sensor Networking*. MiSeNet. Istanbul, Turkey, 2012.

[45] L. Sweeney. “k-anonymity: A Model for Protecting Privacy”. In: *International Journal of Uncertainty, Fuzziness and Knowledge Based Systems.* 10.5 (Oct. 2002), pp. 557–570.

[46] A. Machanavajjhala et al. “L-diversity: Privacy beyond k-anonymity”. In: *ACM Transactions on Knowledge Discovery Data* 1.1 (Mar. 2007), pp. 1–47.

[47] J. H. Saltzer and M. D. Schroeder. “The protection of information in computer systems”. In: *Proceedings of the IEEE* 63.9 (1975), pp. 1278–1308.

[48] R. Dingledine, N. Mathewson, and P. Syverson. “Tor: the second-generation onion router”. In: *Proc. of the 13th Conf. on USENIX Security Symp.* San Diego, USA, 2004.

[49] J. Camenisch and J. Groth. “Group signatures: Better efficiency and new theoretical aspects”. In: *Security in Communication Networks*. Springer, 2005, pp. 120–133.

[50] G. Ateniese et al. “A practical and provably secure coalition-resistant group signature scheme”. In: *Advances in Cryptology*. 2000.

[51] A. Santos et al. “Context Inference for Mobile Applications in the UPCASE Project”. In: *MobileWireless Middleware, Operating Systems, and Applications*. Vol. 7. Springer Berlin Heidelberg, 2009, pp. 352–365.

[52] M. Naor and B. Pinkas. “Oblivious Transfer with Adaptive Queries”. In: *Proceedings of the 19th Conference on Advances in Cryptology (CRYPTO)*. London, 1999.

[53] Rade Stanojevic, Nikolaos Laoutaris, and Pablo Rodriguez. “On economic heavy hitters: shapley value analysis of 95th-percentile pricing.” In: *Internet Measurement Conf.* 2010.

[54] B. Blanchet. “Automatic proof of strong secrecy for security protocols”. In: *IEEE Symp. on Security, Privacy*. 2004.

[55] D. Dolev and A. C. Yao. *On the security of public key protocols*. Tech. rep. Stanford University, 1981.

[56] Leo Breiman. “Random Forests”. In: *Machine Learning* 45.1 (2001), pp. 5–32.

[57] D. R. L. Brown. *Recommended Elliptic Curve Domain Parameters*. Tech. rep. Certicom Research, Jan. 2010.

[58] S. Uppoor and M. Fiore. “Large-scale urban vehicular mobility for networking research”. In: *Proc. of the 3rd IEEE Vehicular Networking Conf. (VNC)*. Amsterdam, Nov. 2011.

Stylianos Gisdakis received his Diploma in Computer Science from Athens University of Economics and Business (AUEB) in 2008. He received his MSc degree in 2011 on Information and Communication Systems Security from the Royal Institute of Technology (KTH), Sweden, where he is currently working towards the Ph.D. degree in Networked Systems Security.

Thanassis Giannetsos earned his Ph.D. degree from University of Aalborg, Denmark in 2012. Currently he is a Senior Researcher at the Networked Systems Security group with KTH Royal Institute of Technology, Stockholm, Sweden. His research interests span from the theoretical foundations of cryptography to the design and implementation of efficient and secure communication protocols.

Panagiotis (Panos) Papadimitratos earned his Ph.D. degree from Cornell University, Ithaca, NY, in 2005. He then held positions at Virginia Tech, EPFL and Politecnico of Torino. Panos is currently an Associate Professor at KTH, where he leads the Networked Systems Security Group. His research agenda includes a gamut of security and privacy problems, with emphasis on wireless networks. Web page: www.ee.kth.se/nss