# SPPEAR: Security & Privacy-Preserving Architecture for Participatory-Sensing Applications

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 participatory sensing (PS). The openness of such systems and the richness of user data they entail raise significant concerns for their security, privacy 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 comprehensive solution. That is, a secure and accountable PS system that preserves user privacy, and enables the provision of incentives to the participants. At the same time, we are after a PS system that is resilient to abusive users and guarantees privacy protection even against multiple misbehaving PS entities (servers). We address these seemingly contradicting requirements with our SPPEAR architecture. Our full blown implementation and experimental evaluation demonstrate that SPPEAR is efficient, practical, and scalable. Last but not least, we formally assess the achieved security and privacy properties. Overall, our system is a comprehensive solution that significantly extends the state-of-the-art and can catalyze the deployment of PS applications.

## Categories and Subject Descriptors

D.4.6 [**Operating Systems**]: Security and Protection; E.3 [**Data Encryption**]: Public Key Cryptosystems

## General Terms

Experimentation, Performance, Security

## Keywords

Participatory Sensing; Security; Privacy; Anonymity

## 1. INTRODUCTION

Mobile platforms with a broadening gamut of sensing capabilities are now available in increasing numbers. Unlike wireless sensor networks, which are deployed in a given area, smartphones (for example) are anyway carried around by numerous users. Leveraging their wide scale proliferation and sensing capabilities, one could collect valuable data of unprecedented quality and quantity, practically from everywhere. This new paradigm of *participatory* or *mobile crowd* sensing [1, 2] is brought forth by numerous research projects, ranging from environmental monitoring [3, 4] and urban sensing [5, 6] to intelligent transportation systems [7, 8], assistive health-care [9, 10] and public safety [11].

Participatory Sensing (PS) has the potential to offer a new understanding of our environment and lead to innovative applications that create added value for the contributing users. However, for this to materialize, users must embrace the initiatives *"from the people, for the people"* systems and participate in great numbers. The ubiquity of mobile devices renders mass participation feasible but, at the same time, users are increasingly concerned with the *security* and the *privacy* of their sensitive information; recent revelations of mass surveillance [12] aggravate such anxieties.

The more the users engage and are called upon by the PS system, the richer the data they contribute (or consume) and, thus, the more susceptible they are to privacy threats. Sensitive information, including daily routines, location and social relations, is given away [13]. The fine-grained nature of such personal data can lead to extensive user-profiling, unsolicited targeted advertisement or, even, personal attacks and stalking [14]. This is intensified when users belong to small groups that share similar characteristics (e.g., work/residence area, entertainment preferences [15]). However, as recent experience shows, assuming that users can simply trust the PS system they contribute sensitive data to, is no longer a viable option [16, 17]. Therefore, it is imperative to address privacy concerns because users perceive them to be significant; as a result, they may refuse to use or even oppose a service.

Even though protecting privacy is a necessary condition for user participation, it is not (by itself) a sufficient one. Indeed, the research community has identified the importance of *incentivizing* users so that they provide a continuous influx of contributions. The type of incentives and the way they materialize (i.e., reputation systems [18], service quotas [19], or monetary rewards [20]) largely depend on the stake-holder(s) that initiate the sensing tasks. However, it is necessary to provide such incentives in a privacy preserving manner. For example, users must be able to receive quotas for their contributions without associating themselves with the data or the task they participated in.

On the other hand, the desired openness of participatory sensing, i.e., anyone that *can* get involved *should* contribute


*WiSec'14,* July 23–25, 2014, Oxford, UK.

http://dx.doi.org/10.1145/2627393.2627402.

data, introduces a series of threats to the *trustworthiness* of the system as it does not preempt adversarial behavior and malicious (or erroneous) contributions. Attackers can interfere with the sensing process and, therefore, manipulate the results of the PS tasks. To this end, we need protocols that can hold offending users *accountable*, but without necessarily disclosing their identity.

To reap the benefits of this new community sensing paradigm, it is imperative to address all these issues while complying with the seemingly contradicting demands for privacy and accountability. This sets the challenge ahead: *How to build secure and accountable PS architectures that can safeguard user privacy while supporting various user incentive mechanisms?* Despite the plethora of research efforts, what remains is to consider the aforementioned problem as a whole. Existing works are concerned with parts of the problem at hand; they either ensure privacy and security without considering accountability [14, 21, 22, 23, 24]; or they differentiate between users that either contribute or consume information [25, 26]. At the same time, a separate body of research investigates incentive mechanisms for PS without considering user privacy [18, 19, 20].

*Contributions:* Our work meets this challenge, proposing SPPEAR; a comprehensive secure and privacy-preserving architecture for PS systems, which systematically addresses all key PS aspects, i.e., privacy, security, accountability and incentives provision. More specifically, SPPEAR ($i$) is scalable, dependable and applicable to any type of PS application, ($ii$) guarantees user non-identifiability and offers strong *privacy protection*, ($iii$) limits participation to legitimate users in a fully *accountable* manner, ($iv$) efficiently shuns out offending users without, necessarily, revealing their identity, ($v$) is resilient to compromised and colluding PS entities, and ($vi$) can support various incentive mechanisms in a privacy-preserving manner. We provide a full-blown 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.

The paper is organized as follows: first, we survey the state-of-the-art research efforts in the area (Section 2). We then describe the system and adversarial models (Section 3) and discuss the PS security and privacy requirements (Section 4). In Section 5, we provide an overview of SPPEAR and the services it offers followed by a detailed presentation of all implemented components and protocols (Section 6). We present a formal assessment of the achieved properties (Section 7) and a detailed performance evaluation (Section 8). Finally, in Section 9 we conclude this work.

## 2. RELATED WORK

Participatory sensing has attracted the attention of the research community, especially in the context of security and privacy [27, 28]. [29] introduces the concept of participatory privacy regulations which allow participants to control the information they disclose. [14, 24, 30, 31, 32] preserve location privacy through *obfuscation* (i.e., by generalizing or peturbing spatiotemporal information associated to participants) and *anonymization* (i.e., by removing user identities).

The integrity and the authenticity of user-generated content is guaranteed by leveraging Trusted Platform Modules in [33] and [34]. However, these schemes do not consider other security aspects of the PS environment; i.e., system abuse by malicious (or erroneous) contributions or accountability for misbehaving users. This renders them vulnerable to information distortion and data pollution: malicious users can attack the data collection process by submitting faulty samples, without being held culpable for their actions.

AnonySense [21] is a general-purpose framework for secure and privacy preserving tasking and reporting. Reports are submitted through wireless access points, while leveraging Mix Networks [35] to de-associate the submitted data from their origin. However, the way it employs the short group signatures scheme defined in [36], for the cryptographic protection of submitted reports, renders it vulnerable to sybil attacks (Section 7). Although AnonySense can evict malicious users, filtering out their past and faulty contributions requires the *de-anonymization* of benign reports[1]; besides being a costly operation, this process violates the anonymity of legitimate participants. Misbehavior detection is a lengthy process that may occur even at the end of the sensing task when all contributions are available (e.g., by detecting outliers). SPPEAR shuns out offending users and filters out their malicious input through an efficient revocation mechanism (Section 6.6) that does not erode the privacy of benign users.

Other 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 [37]. Exceeding this is considered misbehavior and results in user de-anonymization and revocation. Nonetheless, this technique cannot capture other types of misbehavior, i.e., when malicious users/devices pollute the data collection process by submitting $(k-1)$ faulty samples within a time interval. In contrast, SPPEAR is *misbehavior-agnostic* and *prevents* such anonymity abuse by leveraging authorization tokens and pseudonyms with non-overlapping validity periods (Section 7).

PEPSI [22] prevents unauthorized entities from querying the results of sensing tasks with provable security. It is based on a centralized solution that focuses on the privacy of data queriers; i.e., entities interested in sensing information. Unlike our work, PEPSI does not consider accountability and privacy-preserving incentive mechanisms and it does not ensure privacy against cellular Internet Service Providers (ISPs).

PEPPeR [26] protects the privacy of the parties querying mobile nodes (and not of the information contributing nodes), by decoupling the process of node discovery from the access control mechanisms used to query these nodes. PRISM [23] focuses on the secure deployment of sensing applications and does not consider privacy. It follows the *push model* for distributing tasks to nodes: service providers disseminate applications to mobile devices (according to criteria such as their location). This approach enables timely and scalable application deployment, but harms user privacy since service providers have knowledge of the device locations. On the contrary, our work provides comprehensive security and privacy protetion, in the presence of stronger adversaries, for all users irrespectively of their role (i.e., contributing or querying).

Complementary studies focus on the provision of *incentives* to stimulate user participation [18, 19, 20, 38, 39] by leveraging various incentive mechanisms such as auctions, dynamic pricing, monetary coupons, service quotas and reputation systems. However, as these mechanisms do not consider pri-

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

vacy, they can reveal sensitive information by linking users to the data they contribute to the system.

# 3. SYSTEM AND ADVERSARY MODEL

**System Model**: We consider a generic Participatory Sensing (PS) system that consists of [40]:

- **Task Service Providers**: they initiate data collection campaigns, defining the scope and the domain of the sensing tasks (e.g., estimation of traffic congestion from 8 to 10 AM).
- **Users**: they carry mobile devices (e.g., smart phones, tablets, smart vehicles) equipped with embedded sensors (e.g., cameras, microphones, light sensors, gyroscopes) and navigation modules (e.g., GPS). Mobile devices collect sensory data and report them to the PS infrastructure[2]. Additionally, involved participants can also query the results of a sensing task.
- **Back-end infrastructure**: it is responsible for supporting the life cycle of a sensing task; it registers and authenticates users, collects and aggregates user-contributed reports and, finally, disseminates the results (of the sensing task) to all interested stake-holders and to the task service provider that initiated the task.

**Adversary Model**: The openness of PS systems renders them vulnerable to abuse by both *external* and *internal* adversaries.

External adversaries are entities without an association to the PS system, and thus, they have limited disruptive capabilities. They can eavesdrop communications (to gather information on task participation and user activities). They might manipulate the data collection process by submitting unauthorized samples or replaying the ones of benign users. They can also target the availability of the system by launching jamming and D(D)oS attacks. The latter attacks are beyond our scope and, therefore, we rely on the network operators (e.g., ISPs) for their mitigation.

Internal adversaries can be users or PS system entities that exhibit malicious behavior. Users, or their compromised devices, might contribute faulty measurements or attempt to impersonate other entities and pose with multiple identities (i.e., acting as a Sybil entity). Moreover, adversarial users could try to exploit the incentive mechanisms in an attempt to increase their utility (e.g., coupons, rewards, quotas, receipts) either without offering the required contributions (i.e., not *complying* with the requirements of the task [41]) or by *double-spending* already redeemed quotas.

At the same time, internal attacks can target user privacy, i.e., seek to identify, trace and profile users, notably through PS-specific actions[3]. This is especially so in the case of misbehaving infrastructure components. More specifically, we consider: (*i*) *fully compromised* entities that exhibit arbitrary malicious behavior, (*ii*) "*honest-but-curious*" entities executing correctly the protocols but curious to learn private user data, and (*iii*) *colluding* entities, collectively trying to harm user privacy.

[2]Devices leverage any type of telecommunication networks (e.g., 3/4G, WiFi, WiMax).

[3]For instance, user de-anonymization by examining the content of the reports they submit [21]

# 4. SECURITY & PRIVACY REQUIREMENTS

Recent revelations of mass surveillance [12] make people increasingly concerned about the security and privacy of their personal information. PS systems will not succeed if they require users to perform a leap of faith and contribute their sensitive data. Users demand strong security and privacy guarantees. However, security and privacy protection alone cannot ensure that users will embrace PS applications. To reap the benefits of this emerging paradigm, we need a synthesis of the above with *incentive* mechanisms.

SPPEAR, our security and privacy-preserving architecture for PS systems, offers a broadened security and privacy protection under weakened trust assumptions. In particular, we address:

**R1. Communication integrity, confidentiality and authentication**: The communication among the PS entities should be authenticated and protected from any alteration and/or disclosure to unauthorized entities.

**R2. Authorization and Access Control**: The participating user device should act according to the policies specified by the sensing task, defined by the corresponding initiator. To enforce such policies, the PS architecture should provide *access control* and *authorization* services.

**R3. Non-Repudiation and Accountability of Actions**: Actions should be non-repudiable and all system entities (i.e., users and infrastructure components) should be held accountable for their actions.

**R4. Anonymity**: Users (their devices and their actions) should not be identifiable. Observers should not be able to infer private information and whether a user performed or will perform a specific action. Moreover, no observer should be able to link an action to the user or infer if two (or more) actions were performed by the same user (device). Anonymity is *conditional* in the sense that it can be revoked when users deliberately disrupt the operation of the system or contaminate the data collection process (i.e., by submitting faulty reports)[4].

**R5. Fairness**: Misbehaving users should not be able to exploit the incentive mechanisms (e.g., receipts) to increase their utility without making the requested contributions [28].

# 5. SPPEAR ARCHITECTURE

In this section we provide an overview of SPPEAR, its entities and protocols.

## 5.1 System Entities

**Users (Information Prosumers):** Users act both as *information producers* (i.e., submit data) and *information consumers* (i.e., request information from the system). User devices with sensing capabilities (e.g., mobile phones, vehicles), participate in tasks by submitting authenticated samples, or by querying for (collected) data.

**Task Service (TS):** This entity initiates sensing tasks and campaigns. It also, defines and provides the rewards participants shall receive for their contributions [40].

**Group Manager (GM):** It is responsible for the registration of user devices, issuing anonymous credentials to them. Furthermore, the Group Manager authorizes the par-

[4]The faulty behavior detection depends on the tasks, and it is orthogonal to this investigation.

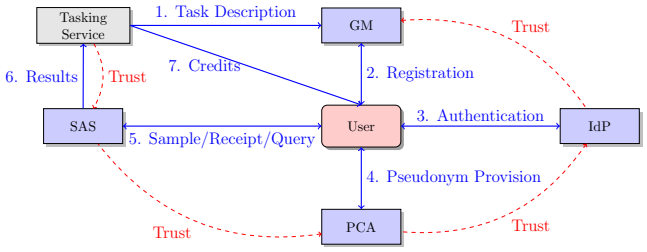


**Figure 1: SPPEAR Overview**

ticipation of devices in various tasks in an *oblivious manner*, using *authorization tokens*.

**Identity Provider (IdP):** It offers identity and credential management services (e.g., user authentication and access control, among others) to the PS system.

**Pseudonym Certification Authority (PCA):** It provides anonymized ephemeral credentials, termed *pseudonyms*, to the devices; they are used to cryptographically protect (i.e., ensure the integrity and the authenticity) the submitted samples, or to authenticate devices querying the results of the sensing task. To achieve *unlinkability*, devices obtain multiple pseudonyms from the PCA.

**Sample Aggregation Service (SAS):** User devices submit samples to this entity which is responsible for storing and processing the collected data. This is orthogonal to our work and it depends on the task/application. Although some privacy preserving data processing [42, 43, 44]) could be employed, we neither assume nor require such mechanisms. For each authentic submitted sample, the SAS issues a *receipt* to the device, which later submits it to claim credits for the sensing task. The SAS exposes interfaces that enable any *authenticated* and *authorized* user to query for the results of sensing tasks/campaigns.

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

SPPEAR separates processes and functions across entities, according to the *separation-of-duties* principle [45]: each entity is given the minimum information required to execute the desired task. This way, SPPEAR achieves its goals under weakened assumptions on the trustworthiness of the PS system. In particular, we ensure user privacy even in the case of "*honest-but-curious*" infrastructure and prevent a single PS entity from answering the user-information sensitive question: "*Which user (Who) submitted What sample, for which task (Where) and When*".

## 5.2 Trust Establishment

The aforementioned system entities need to establish trust relations. SPPEAR leverages Security Assertion Markup Language (SAML) assertions [46] 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 that contain information such as authentication requirements, entity URIs, protocol bindings and digital certificates. The metadata published by the IdP contain the *X.509* certificates the PCA uses to verify the signatures of the assertions produced by the IdP. Similarly, the PCA publishes metadata that contain its digital identifier and certificates.

To verify the authorization token, the IdP possesses the digital certificate of the GM. The pseudonyms issued to the user devices are signed with the PCA's private key. The SAS possesses the digital certificate of the PCA. An overview of our design and the trust relations of its components are illustrated in Figure 1.

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 PS entities (i.e., IdP, PCA, SAS). Furthermore, to prevent de-anonymyzation on the basis of network identifiers, we leverage the TOR anonymization network [47].

# 6. SPPEAR PROTOCOLS

In a nutshell, the Task Service (TS) generates sensing tasks and campaigns. Each task is associated with the number of credits, $C$, that users shall receive from the TS for their participation, as long as they submit at least $n$ reports to the Sample Aggregation Service (SAS). The ($C$, $n$) parameters are included in the task description. Once ready, the TS informs the Group Manager (GM) about the newly generated task. Then, the GM initializes a *group signature* scheme which allows each participant ($P_i$) to anonymously authenticate herself with a private key ($gsk_i$). The GM pushes the group public key to the Identity Provider (IdP) responsible for authenticating users (Section 6.1).

The GM publishes a list of active tasks that users regularly retrieve in order to select the ones they want to contribute to. The task description can be done with the use of task-specific languages similar to *AnonyTL* [21]. 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 (Section 6.2). Then, the device initiates the authentication protocol with the IdP and it obtains pseudonyms from the Pseudonym Certification Authority (PCA) (Section 6.3). With these pseudonyms the device can (anonymously) authenticate the samples it submits to the SAS (and receive a credit receipt for each of them) or get authenticated to query the task results (Section 6.4). Finally, the device presents $n$ receipts to the TS to receive the task credits (Section 6.5).

## 6.1 Task Initialization

The life cycle of a sensing task starts when the TS registers it to the GM which, in turn, examines its requirements and generates a task descriptor, in XML format. Then, the GM instantiates a group signature scheme by computing a group public key, $gpk$.

Group signatures fall into two categories, in terms of *group dynamicity*: static and dynamic. The former requires a fixed number of group members, whereas the latter allows dynamic addition of members to the group. The selection of the appropriate scheme is coupled to the context of the sensing task. To exemplify this, assume a sensing campaign that requires the participation of only "premium" users. In this case, the number of eligible users is known and thus static group signature schemes are applicable. Otherwise, dynamic group signatures are necessary. SPPEAR supports, but is not limited to, two group signature schemes; *Short Group Signatures* [36] (static) and the *Camenisch-Groth* scheme [48] (dynamic).

**Algorithm 1**: *Authorization Token* Acquisition

**Result**: Device obtains *authorization token* $X_{i,j}$

**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**
4. 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**

## 6.2 Device Registration and Authorization Token Acquisition

To participate in a sensing task, a user must register her device to the Group Manager (GM) and obtain the private key $gsk_i$. Towards this end, the device initiates an interactive $JOIN$ protocol with the GM.[5] This protocol guarantees *exculpability*: no entity can forge signatures besides the intended holder of the key [49].

The GM generates an *authorization token dispenser*, $D_{auth}$. Each token in it binds the identity of the registered user to the identifiers of the active tasks and the type of relevant access rights (submit samples or access the results of the sensing task). The binding is done with the use of 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 $t_i$ is:

$$\{t_{id}, h(user_{id} \,||\, task_i \,||\, n), task_i, submit/query\}_{\sigma_{GM}}$$

$N$ denotes the number of currently active sensing tasks, $n$ is a nonce and $t_{id}$ is the token identifier. In order to participate in a task, the device must pick the corresponding token the dispenser.

Nevertheless, merely requesting a token would compromise users' privacy; besides being aware of the real identity of the user, the GM would learn the task she wishes to contribute to and this could lead to a breach of her privacy. For example, participating in a task that measures noise pollution during night hours within an area "A" can help the GM deduce sensitive user information such as home location and personal activities (among others) [50, 51].

To prevent this, SPPEAR employs *Private Information Retrieval (PIR)* techniques. Currently, we support 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.

[5]Due to space limitations, we refer the reader to [36, 48]

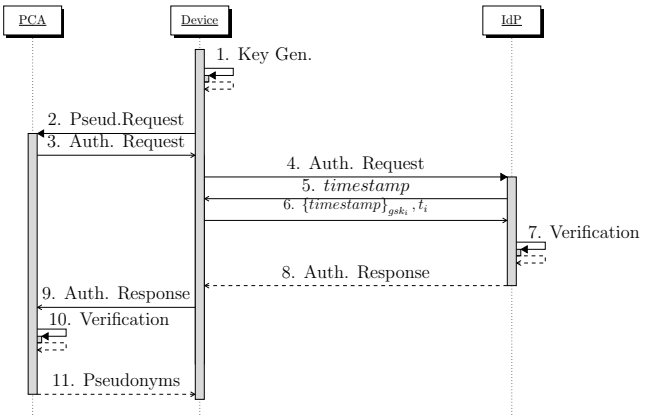


**Figure 2: Authentication Protocol**

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 Decisional Diffie-Helman (DDH) assumption [52]. As the token acquisition protocol leverages oblivious transfer, the GM does not know which token was obtained by the device and, thus, cannot deduce the task the user wishes to contribute to. In Sec. 8 we present the scheme complexity along with a quantitative analysis of its performance (Sec. 8.3).

## 6.3 Device Authentication

Having the signing key, $gsk_i$, and the authorization token, the device can now authenticate itself to the IdP and receive pseudonyms from PCA. A pseudonym is an X.509 certificate [53] that binds an anonymous identity with a public key. Figure 2 presents the authentication protocol, based on Web-Services, which is as follows:

**Phase 1:** The device generates the desired amount of keypairs and creates the same number of Certificate Signing Requests (CSRs) (Step 1).

**Phase 2:** The device requests pseudonyms from the PCA with the generated CSRs (Step 2). Since the device is not yet authenticated, the PCA issues a SAML authentication request [46] (Step 3) to the IdP, signed with its private key and encrypted with the public key of the IdP. According to SAML specifications, the request contains a random *transient identifier* ($tr_{id}$) for identifying and 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 phase (Sec. 5.2).

**Phase 3:** The IdP decodes and decrypts the authentication request, verifies the XML signature of the PCA and initiates the authentication process with the device. Our authentication is based on group signatures. More specifically, 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 for the desired sensing task (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 $tr_{id}$ and an authentication statement (i.e., assertion): this asserts that the device was successfully authenticated anonymously through a group signature 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 its fields (Step 10). Once the transaction is completed, the device is authenticated and it receives valid pseudonyms. The access rights of the device are included as attributes in these pseudonyms (Step 11).

Each pseudonym has a time validity that specifies the period (i.e., the pseudonym life time) for which the pseudonym can be used. If the obtained pseudonyms had overlapping life times, malicious users could expose multiple identities simultaneously, i.e., launch *sybil attacks*. To prevent this, we require that the PCA issue pseudonyms with non-overlapping life times (i.e., two pseudonyms are never valid during the same time interval).

## 6.4 Sample Submission and Incentives Support

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

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

$v$ is the value of the sensed phenomenon, $t$ is a time-stamp, $loc$ are the coordinates of the device; $\sigma_{PrvKey}$ 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 SAS verifies the signature and time-stamp, against the time validity of the pseudonym. If the sample is deemed authentic, the SAS prepares a receipt, $r_i$, for the device:

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

$\sigma_{SAS}$ denotes the digital signature of the SAS. The device stores each received receipt until the end of the task.

To query the results of the task, the device can authenticate itself to the SAS with a pseudonym (using two-way authentication over TLS). The use of different pseudonyms for interacting with the SAS provides unlinkability. To ensure device anonymity, communications are done over TOR.

## 6.5 Task Finalization

As they submit reports to the SAS, devices accumulate a number of receipts. To receive the credits assigned to a sensing task, a device has to collect at least $n$ receipts (where $n$ is specified by the tasking service). When a device has the required amount of receipts, it submits them to the Task Service (TS). The TS verifies their signatures and then invalidates them (i.e., stores them in a database and marks them as used), so that they cannot be re-used by any other device. If the number of submitted receipts satisfies the task requirements (i.e., $n$ receipts), the $C$ credits are given to the user; either in the form of reputation or in the form of a voucher. Again, to ensure the anonymity of the device, communications are done over TOR.

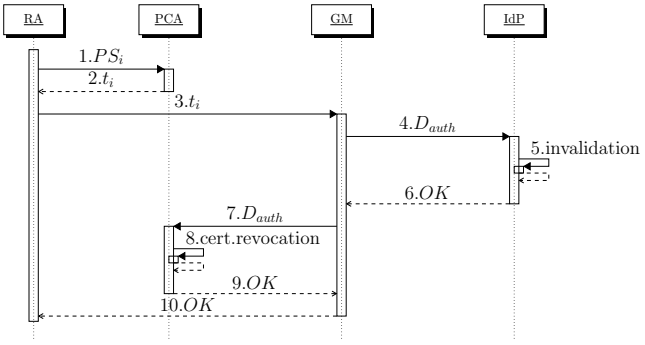


**Figure 3: Pseudonym Revocation**

## 6.6 Pseudonym Revocation

If required, SPPEAR provides efficient means for shunning out offending users. Assume one device whose (anonymously) submitted samples significantly deviate from the rest. This could serve as an indication of misbehavior (e.g., an effort to *pollute* the results of the sensing task). In that case, the device should be prohibited from further participating in the task. If not deliberately misbehaving, one of the device sensors malfunctions. In that case, it might be required that the device is revoked from the affected tasks (e.g., the ones that rely on the specific malfunctioning sensor). To address the above scenarios, we design fine-grained revocation protocols, suitable for different levels of escalating misbehavior:

**Total Revocation:** The RA coordinates this protocol based on a (set of) pseudonym(s) $PS_i$ (see Figure 3). Upon completion, the device for which the pseudonym was issued is evicted from the system:

**Phase 1:** The RA provides the PCA with the $PS_i$ (Step 1). The PCA 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 passed by the RA to the GM (Step 3).

**Phase 2:** Based on $t_i$, 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 its tokens were invalidated.

**Phase 3:** To revoke the already issued pseudonyms, the GM sends the dispenser, $D_{auth}$, to the PCA. The PCA determines the tokens in the dispenser it has issued pseudonyms for. Then, it updates its Certificate Revocation List (CRL) with all the not yet expired pseudonyms of the device (Steps 7, 8). At this point, the device can no longer submit samples to the SAS because its pseudonyms have been revoked; clearly, SAS has access to the PCA's CRL.

**Partial Revocation:** This protocol evicts a device from a specific sensing task. The RA sends the pseudonym, $PS_i$, that needs to be revoked 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 can be issued only one token

per task, and this is now revoked, the device can no longer participate in the task. The partial revocation protocol does not involve the GM and thus, it does not remove the device anonymity; this is important for preserving the anonymity of malfunctioning (but not malicious) devices.

# 7. SECURITY AND PRIVACY ANALYSIS

In this section, we first discuss SPPEAR with respect to the requirements defined in Section 3. We then provide a formal analysis of the achieved security and privacy properties.

Communications take place over secure channels (TLS). This ensures communication *confidentiality* and *integrity*. Furthermore, each system entity has a digital certificate for authentication (thus we get $R_1$).

In SPPEAR, 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 (thus we get $R_2$).

Malicious devices can leverage anonymity and inject faulty reports to "pollute" the collection process. As an example, let us 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 pseudo- or anonymity or, if possible, by launching a sybil-attack, misbehaving users can provoke a false perception over the congestion levels of a road network and thus, disrupt traffic. State-of-the-art schemes (e.g., [21]) that rely on group signatures to authenticate submitted reports are vulnerable to abuse. More specifically, it is impossible to detect if two reports were generated by the same device without opening the signatures of all reports, irrespectively of the device that generated them. Besides being a costly operation,[6] this approach would violate 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 time validities. This way, the PCA can corroborate the time validity of the previously issued pseudonyms and if requested by the device, provide it with new pseudonyms that do not overlap the previously issued ones. This prevents malicious devices from using multiple pseudonyms, simultaneously, and renders SPPEAR secure against Sybil attacks. Nevertheless, re-using the same pseudonym to cryptographically protect more than one reports trades off privacy (linkability) for overhead (Section 8.5 contains an extensive analysis on this).

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 thanks to 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 SAS via TOR, devices can achieve enhanced sample/report unlinkability. Furthermore, TOR prevents the IdP, the PCA, and the cellular ISPs from de-anonymizing devices based on network identifiers, such as MAC and IP addresses ($R_4$). SPPEAR "hides" (by leveraging end-to-end encryption and TOR) from the cellular ISP all fine-grained and sensitive information exchanged in the PS context. Essentially, the cellular ISPs gain no additional information from the participation of the device in the sensing task.

[6]Due to space limitations we refer the reader to [36].

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

**Table 1: Secrecy Analysis for Dolev-Yao Adversaries**

The first three columns of Table 1 present the information each SPPEAR entity possesses. Our approach (i.e., the separation of duties design principle) prevents a single infrastructure entity from accessing all user-sensitive pieces of information (we elaborate on colluding infrastructure entities in Section 7.1.2).

The cryptographic primitives employed by SPPEAR guarantee that no (offending) user can deny her actions. More specifically, due to the interactive protocols executed during the registration phase (Section 6.2), $gsk_i$ is known only to the user device and as a result, *exculpability* is ensured [36]. Furthermore, digital signatures are generated with keys known only to the device and thus, non-repudiation is achieved.

SPPEAR can shun out offending devices (see Section 6.6) without, necessarily, disclosing their identity ($R_3, R_4$). To achieve permanent eviction of misbehaving devices the registration phase can be enhanced with authentication methods that entail network operators (e.g., GBA [54]). However, we leave this as a future direction.

We consider operation in semi-trusted environments. More specifically, 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 accountable for misbehavior. Moreover, the IdP cannot falsely authenticate non-registered devices: it cannot forge the authorization tokens included in the SAML assertions (see Section 6.3). As a result, the PCA will refuse issuing pseudonyms and, thus, the IdP will be held accountable. Moreover, SAML authentication responses (Section 6.3) are digitally signed by the IdP and thus cannot be forged or tampered by malicious devices. Overall, in SPPEAR, one entity can serve as a witness of the actions performed by another; this way we establish a strong *chain-of-custody* ($R_3$).

A special case of misbehavior is a malicious SAS that exploits the total revocation protocol (Section 6.6) to de-anonymize users. To mitigate such behavior, we require that strong indications of misbehavior be presented to the RA before the resolution and revocation protocols are executed. Nevertheless, such aspects are beyond the scope of this work.

Malicious users cannot generate receipts because they cannot forge the signature of the SAS. Furthermore, each receipt is bound to a task and, thus, cannot be used to earn credits from another task. Colluding malicious users can exchange sample receipts among them. Nevertheless, receipts are invalidated upon submission and cannot be "double-spent" (thus we get requirement $R_5$).

Receipts are generated by the SAS and validated by the Tasking Service (Section 6.4), neither of which knows the long-term identity of the user. As a result, the presented incentive mechanism preserves user anonymity ($R_4$).

| Honest-but-curious (colluding) entities | Information Leaked | 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. |
| SAS | $s$, $PS$, $r$ | The SAS 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, SAS | $s$, $PS$, $r$ | When the GM and the SAS 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, SAS | $t$, $PS$, $s$, $r$ | The PCA and the SAS can infer that an anonymous user received pseudonyms for a specific task and has submitted a report. |
| GM, PCA, SAS | all | Full de-anonymization of the user, the task she participates in and the reports she has submitted. |

**Table 2: Honest-but-curious entities with ProVerif.**

## 7.1 Formal Analysis

For the correctness of the employed cryptographic primitives (i.e., the group signature and the PIR schemes) we refer to [36, 48] and [52]. Here we formalize SPPEAR security and privacy properties with respect to the introduced entities and functionalities.

We use ProVerif, an automated protocol verifier [55], to model SPPEAR in $\pi$-Calculus [55]. In ProVerif, entities (infrastructure components and users) are described as processes. Protocols (i.e., authentication, Section 6.3, sample submission, Section 6.4, and revocation, Section 6.6) are modeled as a parallel composition of multiple copies of these processes. ProVerif assumes sets of *names* and *variables* along with a finite *signature*, $\Sigma$, 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 retrieve parts of the messages they operate on.

Adversaries in ProVerif follow the Dolev-Yao model [56]: they can eavesdrop, modify and forge messages according to the cryptographic keys they possess. To protect communications, every emulated PS entity in the analysis maintains its own private keys/credentials. This adversarial model captures any type of misbehavior besides curious and colluding system entities (considered in Section 7.1.2), as it assumes that the adversary does not have any knowledge on their corresponding cryptographic keys.

### 7.1.1 Secrecy, Strong Secrecy and Unlinkability

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 verify *strong-secrecy* properties implying the adversary cannot infer changes over secret values. To examine strong-secrecy for datum $i$, the predicate *noninterf* is used. We evaluate the properties of SPPEAR-specific data. Table 1 summarizes our findings: SPPEAR ensures not only secrecy but also strong-secrecy for all the critical pieces of information. Thus, it guarantees system security and user privacy.

As adversaries cannot infer changes over the aforementioned data, unlinkability [57] (with respect to Dolev-Yao adversaries) is achieved [58]. More specifically, given two tokens, $t_1$ and $t_2$, belonging to the same user, it is impossible for the adversary to relate them. The same holds for the rest of SPPEAR-specific data (i.e., samples, pseudonyms, receipts).

### 7.1.2 Honest-but-curious System Entities

We additionally consider the case of honest-but-curious system entities that collude to breach user privacy. We model such behavior in ProVerif by using a *spy channel* that is accessible by the adversary, and where a curious authority publishes its state, keys and variables. Accordingly, to emulate colluding infrastructure entities, we assume multiple spy channels for each of them. Consequently, the Dolev-Yao adversary (by monitoring these channels) will have access to all the information handled by these entities. Table 2 presents the pieces of information that leak (along with their semantics) for various combinations of honest-but-curious colluding entities.

Single system entities cannot fully de-anonymize users as they have limited access to user information (Table 1). Furthermore, SPPEAR also prevents de-anonymization even when two authorities collude. In order to completely de-anonymize users and their actions, it is required that the GM, the PCA and the SAS collaborate. In case these components are deployed within different administrative domains, their collusion is rather improbable. Nevertheless, if they are within the same administrative domain, the separation-of-duties requirement (according to which SPPEAR is designed) may no longer hold; thus, user privacy would not be guaranteed.[7].

# 8. PERFORMANCE EVALUATION

In this section, we discuss the complexity of the employed cryptographic primitives and provide a thorough assessment of SPPEAR's efficiency and dependability (with respect to both the devices and the infrastructure). Furthermore, through realistic simulations, we evaluate the (privacy) effectiveness of the pseudonym usage against a location/sample linking attack.

## 8.1 Complexity Analysis

Table 3 provides an overview of the complexity of the cryptographic primitives employed by SPPEAR. For group signatures, we focus on the number of *modular exponentiations* ($ME$) and *pairing evaluations* ($PE$). For sample submission and verification, we employ the Elliptic Curve Digital Signature Algorithm (ECDSA) with keys computed over 224

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

| Function | Complexity | Entities |
|---|---|---|
| Authentication (BBS) | $12ME + 5PE$ [59] | IdP, User Device |
| Authentication (CG) | $10ME$ [59] | IdP, User Device |
| Sample Submission | $(6n+2)MM + MI + 5nSQ$ [60] | User Device |
| Sample Verification | $(12n+2)MM + MI + 10nSQ$ [60] | SAS |
| Receipt Generation | $(6n+2)MM + MI + 5nSQ$ [60] | SAS |

**Table 3: Complexity Analysis**

bit prime fields (*secp224k1* curve), thus, achieving a 112 bit security level [61]. We present the complexity of ECDSA with respect to *modular multiplications* ($MM$), *squaring* ($SQ$) and *modular inversions* ($MI$) for $n$-bit multiplication operands.

Recall from Section 6.2 that the PIR scheme requires $N$ exponentiations (the number of active tasks). Moreover, each of the two $OT$ transactions requires $O(\sqrt{N})$ steps.

## 8.2 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 [62] library for the cryptographic operations, i.e., the ECDSA and TLS and the JPBC [63] library for the group signature schemes. We have deployed our sensing application on Android smartphones with different specifications: 4-Cores/1 GB RAM and 2-Cores/1 GB RAM.

To emulate the real-world networks, we introduce an artificial network delay at the data link layer: the employed queuing discipline increases randomly the network latency following a normal distribution with a mean of 10 $ms$ and variance of 2.5 $s$.

For the infrastructure evaluation (see Section 8.4) we use Jmeter™ to emulate multiple devices that access the infrastructure concurrently. Additionally, we implement a *mobility tracker*, similar to the one presented in [64] based on Kalman Filters, to perform the privacy evaluation in Section 8.5.

## 8.3 User-Side Evaluation

Figure 4 illustrates the performance of the authentication and pseudonym acquisition protocol on the two Android devices. We assume a device requests an authorization token within a set of 10 tokens (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 (Section 6.3) 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 (around 11 $s$ for 50 pseudonyms). When executing the protocol over TOR, overhead is introduced in the form of network latency. Due to space limitations, we present here the results only for the quad-core device. A latency of 10 $s$ is introduced, thus raising the authentication time to 23 $s$ for 50 pseudonyms. Even for demanding

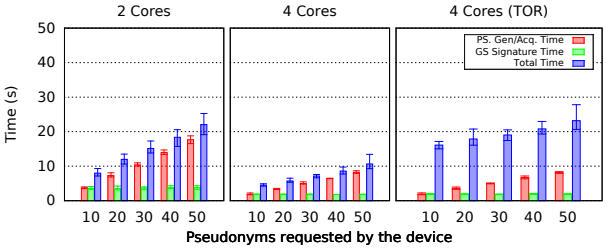


**Figure 4: Authentication Protocol**

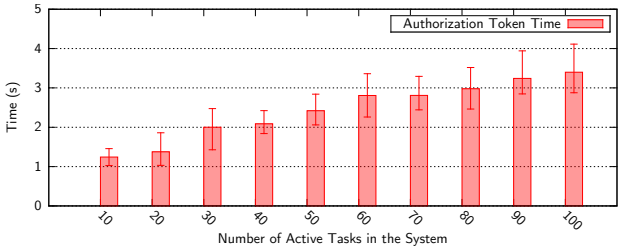


**Figure 5: Token Acquisition Time**

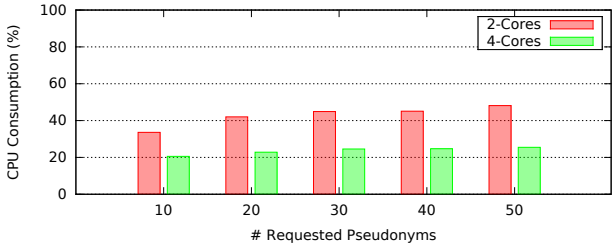


**Figure 6: CPU Consumption**

PS tasks, such a number of pseudonyms provides adequate privacy (Section 8.5).

We assess the efficiency of EC-based digital signatures compared to group-signature schemes on the quad-core device. We find that ECDSA with SHA512 is approximately 10 times faster compared to group signature schemes (BBS scheme [36] for the same security level). This is an important aspect of our design, also compared to AnonySense [21] 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.

For the implemented PIR scheme, Figure 5 shows the time needed to obtain an authorization token for one task on the quad-core device: it increases mildly with the number of active tasks in the system. Even for a set of 100 active tasks, the time needed to obtain one authorization token is approximately 3.5 $s$.

We assess $CPU$ utilization for the two mobile devices (Figure 6). For the dual-core device, the amount of CPU required 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, that is close to the CPU consumption of our client on the same device (for 50 pseudonyms).

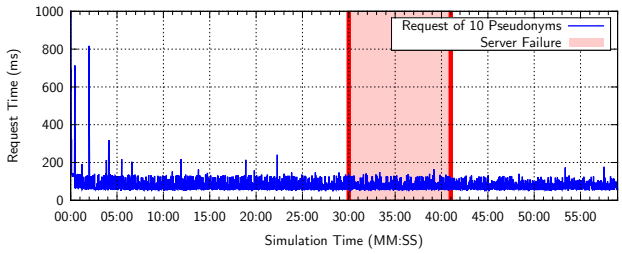


**Figure 7: System Reliability in Real-World Scenario**

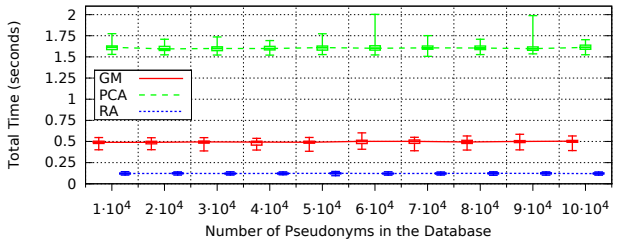


**Figure 8: Device Revocation**

## 8.4 Infrastructure-Side Evaluation

To assess the performance of the system under stressful, yet realistic, scenarios we assumed a privacy-demanding use case that includes *mobility*. The devices of car drivers or passengers get authenticated to our system and receive pseudonyms to submit data on encountered traffic conditions in a privacy-preserving manner. This scenario is a demanding case of participatory sensing, because it entails strict location privacy protection requirements.

To model such conditions, we use the "TAPAS" data set [65] 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 [66]. 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 7 shows that our system performs really well 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, disconnecting completely the VM from the network for 11 minutes. As shown in the grade area of Figure 7, the request latency does not increase and the system recovers transparently from the outage.

Figure 8 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 token $t_i$ that the device used to request the pseudonym $PS$. After retrieving $t_i$, the RA asks the GM to translate $t_i$ to the device long term identifier ($lt_{\text{id}}$). Then, the GM invalidates all device tokens for which the $t_i$ was issued and informs the IdP (Section 6.6). Accordingly, the PCA revokes all device pseudonyms. These two processing delays are accounted for as the time spent on PCA ($t_{\text{PCA}}$) and GM ($t_{\text{GM}}$), respectively. The total time spent on RA is $t_{\text{TOT}} = t_{\text{RA}} + t_{\text{PCA}} + t_{\text{GM}}$, where $t_{\text{TOT}}$ is the total execution time of the pseudonym resolution protocol.

The pseudonym set is generated by assuming the same request policy for all devices. This approach maximizes the entropy of the database set. Each assumed device obtained 10 tokens for requesting a set of 10 pseudonyms, thus giving the overall ratio 1 device : 10 tokens : 100 pseudonyms. The box-plots in Figure 8 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 affected by the size of the set. On average, revocation of a device requires 2.3 $s$.

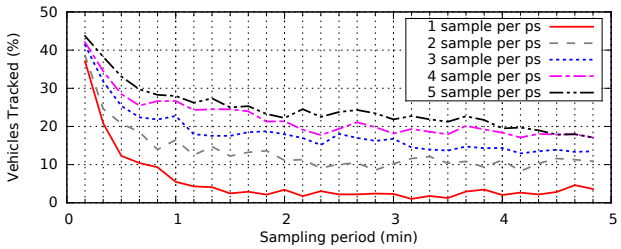


**Figure 9: Privacy Evaluation for Mobility**

## 8.5 Pseudonyms and Protection

To evaluate privacy, notably, the unlinkability achieved by pseudonyms, we consider the following sensing task: drivers, with the use of their smartphones, report their current location and velocity to the SAS. We assume here that the SAS 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 that leverage location information and mobility can be employed to track vehicles. In this context, we emulate such adversarial behavior with a mobility tracker (Section 8.2). 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 [67] microscopic road traffic simulator. Our objective is to understand the privacy implications of varying pseudonym utilization policies. In Figure 9, we present our findings: we plot the fraction of vehicles our tracker tracked for more than 50% of their trip, as a function of the report submission frequency (from 10 $s$ to 5 $min$) for different pseudonym reusage policies, i.e., the number of reports signed under the same pseudonym.

The tracker successfully tracks 37% of the vehicles[8] for a reporting frequency of 10 $s$ and a use of 1 pseudonym per report (maximum unlinkability). The tracking success significantly decreases as we move towards more realistic reporting frequencies: the Kalman Filter-based 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 only 5% of the vehicles are tracked for more than 50% of their trips. By reusing the same pseudonym for 5 reports, this fraction goes to 27%. One interesting observation is that the effect of pseudonym reuse weakens as the sampling frequency decreases to frequencies more relevant to the context of PS, i.e., 1 $report/30s$ [7].

As discussed in Section 8.3, the quad-core device needs approximately 10 $s$ to acquire 50 pseudonyms (Figure 4); which, based on the results of this section, can provide enhanced location privacy for mobility based participatory sensing tasks. Even when pseudonyms are reused, SPPEAR still

[8]Moreover, please note that the regularity of vehicular movement works in favor of the tracker.

offers strong location privacy. Nonetheless, the pseudonym usage policy can be tuned to the level of privacy users require.

# 9. 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 PS systems and their users. To meet this challenge, we presented SPPEAR; a novel secure and accountable PS architecture that can safeguard user privacy while supporting user incentive mechanisms. SPPEAR 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] J. Burke et al. "Participatory sensing". In: *Workshop on World-Sensor-Web: Mobile Device Centric Sensor Networks and Applications*. Boulder, USA, 2006.

[2] R. K. Ganti, F. Ye, and H. Lei. "Mobile crowdsensing: current state and future challenges." In: *IEEE Communications Magazine* 49.11 (2011), pp. 32–39.

[3] 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.

[4] 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.

[5] L. Deng and L. P. Cox. "LiveCompare: Grocery Bargain Hunting Through Participatory Sensing". In: *ACM 10th Workshop on Mobile Computing Systems and Applications (HotMobile)*. Santa Cruz, California, 2009.

[6] E. Miluzzo et al. "Tapping into the Vibe of the City Using VibN, a Continuous Sensing Application for Smartphones". In: *ACM 1st International Symposium From Digital Footprints to Social and Community Intelligence SCI*. Beijing, China, 2011.

[7] B. Hull et al. "CarTel: a distributed mobile sensor computing system". In: *Proceedings of the 4th International Conference on Embedded networked Sensor Systems*. Boulder, USA, 2006.

[8] 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.

[9] 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.

[10] N. Lane et al. "BeWell: A Smartphone Application to Monitor, Model and Promote Wellbeing". In: *5th International ICST Conference on Pervasive Computing Technologies for Healthcare*. Dublin, Apr. 2012.

[11] J. Ballesteros et al. "Safe cities. A participatory sensing approach". In: *IEEE 37th Conference on Local Computer Networks*. 2012.

[12] G. Greenwald. *"NSA Prism Program Taps in to User Data of Apple, Google and Others"*. June 2013. URL: `http://www.theguardian.com/world/2013/jun/06/us-tech-giants-nsa-data`.

[13] S. Cleveland. "In search of user privacy protection in ubiquitous computing." In: *IEEE 13th Conference on Information Reuse and Integration (IRI)*. 2012, pp. 694–699.

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

[15] A. Singla and A. Krause. "Incentives for Privacy Tradeoff in Community Sensing". In: *Proceedings of the 1st AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*. Palm Springs, 2013.

[16] E. Mills. *"Google sued over Android data location collection"*. 2011. URL: `http://news.cnet.com/8301-27080_3-20058493-245.html`.

[17] J. Lowensohn. *"Apple sued over location tracking in iOS"*. 2011. URL: `http://news.cnet.com/8301-27076_3-20057245-248.html`.

[18] E. Androulaki et al. "Reputation systems for anonymous networks". In: *Privacy Enhancing Technologies*. 2008.

[19] T. Luo and C. K. Tham. "Fairness and social welfare in incentivizing participatory sensing." In: *IEEE 9th Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*. Seoul, 2012.

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

[22] 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.

[23] 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.

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

[25] L. Kazemi and C. Shahabi. "Towards preserving privacy in participatory sensing". In: *IEEE Workshop on Pervasive Computing and Communications (PerCom)*. Seattle, 2011.

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

[28] 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.

[29] K. Shilton et al. "Participatory privacy in Urban Sensing." In: *International Workshop on Mobile Device and Urban Sensing (MODUS)*. St. Louis, USA, 2008.

[30] K. L. Huang, S. S. Kanhere, and W. Hu. "Towards privacy-sensitive participatory sensing". In: *IEEE Conference on Pervasive Computing and Communications*. Galveston, USA, Mar. 2009.

[31] 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.

[32] 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.

[33] A. Dua et al. "Towards Trustworthy Participatory Sensing". In: *Proceedings of the 4th USENIX Conference on Hot Topics in Security*. Montreal, Canada, 2009.

[34] P. Gilbert et al. "Toward Trustworthy Mobile Sensing". In: *Proceedings of the 11th Workshop on Mobile Computing Systems & Applications*. Annapolis, USA, 2010.

[35] D. L. Chaum. "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms". In: *ACM Communications* 24.2 (Feb. 1981), pp. 84–90.

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

[37] 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.

[38] 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.

[39] S. Reddy et al. "Examining Micro-payments for Participatory Sensing Data Collections". In: *ACM 12th International Conference on Ubiquitous Computing*. Copenhagen, Denmark, 2010.

[40] 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.

[41] J. Rula et al. "No "one-size fits all": Towards a principled approach for incentives in mobile crowdsourcing". In: *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications (HotMobile)*. Santa Barbara, CA, 2014.

[42] 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.

[43] N. Li and T. Li. "t-Closeness: Privacy Beyond k-Anonymity and l-Diversity". In: *IEEE 23rd International Conference on Data Engineering (ICDE)*. Istanbul, 2007.

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

[45] 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.

[46] S. Cantor et al. *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0*. Tech. rep. OASIS Standard, Mar. 2005.

[47] R. Dingledine, N. Mathewson, and P. Syverson. "Tor: the second-generation onion router". In: *Proceedings of the 13th Conference on USENIX Security Symposium*. San Diego, USA, 2004.

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

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

[50] L. Pournajaf et al. *A Survey on Privacy in Mobile Crowd Sensing Task Management*. Tech. rep. TR-2014-00. Department of Mathematics and Computer Science, Emory University, 2014.

[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] S. Santesson et al. *"Internet X.509 Public Key Infrastructure Qualified Certificates Profile"*. RFC 3039 (Proposed Standard). Internet Engineering Task Force, Jan. 2001.

[54] 3rd Generation Partnership Project. *Technical Specification Group Services and System Aspects; Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA)*. 2011.

[55] B. Blanchet. "Automatic proof of strong secrecy for security protocols". In: *IEEE Symposium on Security & Privacy*. 2004.

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

[57] A. Pfitzmann and M. Koehntopp. "Anonymity, Unobservability, and Pseudonymity - A Proposal for Terminology". In: *Designing Privacy Enhancing Technologies*. Lecture Notes in Computer Science. Springer, 2001, pp. 1–9.

[58] M. Christofi and A. Gouget. "Formal Verification of the mERA-Based eServices with Trusted Third Party Protocol". In: *Information Security and Privacy Research*. Vol. 376. IFIP Advances in Information and Communication Technology. Springer Berlin Heidelberg, 2012, pp. 299–314.

[59] M. Manulis et al. *Group Signatures: Authentication with Privacy*. Tech. rep. Bundesamt fur Sicherheit in der Informationstechnik, 2012.

[60] J. Petit. "Analysis of ECDSA Authentication Processing in VANETs". In: *3rd International Conference on New Technologies, Mobility and Security*. Cairo, 2009.

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

[62] *"OpenSSL Project"*. URL: `http://www.openssl.org/`.

[63] A. De Caro and V. Iovino. "jPBC: Java pairing based cryptography". In: *Proceedings of the 16th IEEE Symposium on Computers and Communications, ISCC 2011*. Kerkyra, Corfu, Greece, June 28 - July 1, 2011, pp. 850–855.

[64] B. Wiedersheim et al. "Privacy in Inter-Vehicular Networks: Why Simple Pseudonym Change Is Not Eenough". In: *International Conference on Wireless On-Demand Network Systems and Services*. Kranjska Gora, Slovenia, Feb. 2010.

[65] S. Uppoor and M. Fiore. "Large-scale urban vehicular mobility for networking research". In: *Proceedings of the 3rd IEEE Vehicular Networking Conference (VNC)*. Amsterdam, Nov. 2011.

[66] G. Calandriello et al. "On the Performance of Secure Vehicular Communication Systems". In: *Dependable and Secure Computing, IEEE Transactions on* 8.6 (2011), pp. 898–912.

[67] D. Krajzewicz et al. "Recent Development and Applications of SUMO - Simulation of Urban MObility". In: *International Journal On Advances in Systems and Measurements* 5.4 (Dec. 2012), pp. 128–138.