Privacy-Enhanced Participatory Sensing with Collusion-Resistance and Data Aggregation

Felix Günther[1,⋆], Mark Manulis[2], and Andreas Peter[3,⋆⋆]

[1] Cryptography and Complexity Theory Group, Technische Universität Darmstadt, Germany
guenther@cs.tu-darmstadt.de
[2] Department of Computing, University of Surrey, United Kingdom
mark@manulis.eu
[3] Distributed and Embedded Security Group, University of Twente, The Netherlands
a.peter@utwente.nl

Abstract. Participatory sensing enables new paradigms and markets for information collection based on the ubiquitous availability of smartphones, but also introduces *privacy* challenges for participating users and their data. In this work, we review existing security models for privacy-preserving participatory sensing and propose several improvements that are both of theoretical and practical significance. We first address an important drawback of prior work, namely the lack of consideration of *collusion attacks* that are highly relevant for such multi-user settings. We explain why existing security models are insufficient and why previous protocols become insecure in the presence of colluding parties. We remedy this problem by providing new security and privacy definitions that guarantee meaningful forms of collusion resistance. We propose new collusion-resistant participatory sensing protocols satisfying our definitions: a generic construction that uses anonymous identity-based encryption (IBE) and its practical instantiation based on the Boneh-Franklin IBE scheme.

We then extend the functionality of participatory sensing by adding the ability to perform *aggregation* on the data submitted by the users, without sacrificing their privacy. We realize this through an *additively-homomorphic IBE* scheme which in turn is constructed by slightly modifying the Boneh-Franklin IBE scheme. From a practical point of view, the resulting scheme is suitable for calculations with small sensor readings/values such as temperature measurements, noise levels, or prices, which is sufficient for many applications of participatory sensing.

1 Introduction

Participatory sensing is a novel paradigm for data collection using smartphones or other mobile devices with applications ranging from sensing of environmental conditions like traffic [29], urban air [37] and noise level [39,18], or earth quakes [11] to environmental impact measurements [34], sport activities [21], market aspects like fuel prices [20], or concerns of personal health like diets [40]. They all leverage the high and increasing distribution of mobile phones, whose number of subscriptions surpassed 6 billion including a high share of smartphones with sufficient computation power for a variety of sensing tasks.

The employment of people's mobile phones as sensors however also introduces privacy risks. These sensors—now carried around by their owners—reveal sensitive location and behavioral information. In many settings, the sensed data itself is highly privacy-sensitive and requires appropriate protection when published or reported to a central data pool. Participatory sensing hence introduces the challenging task to handle the sensed data in a secure and privacy-preserving manner while offering maximal benefit from the obtained data to its users.

The utility for the above mentioned applications of participatory sensing increases with a growing number of participants. Providing people with an incentive to participate is therefore of crucial importance. From a business point of view, it is reasonable to assume that such an incentive is given by a privacy-preserving

[⋆] Felix Günther is supported by the German Federal Ministry of Education and Research (BMBF) within EC SPRIDE.

[⋆⋆] Andreas Peter is supported by the THeCS project as part of the Dutch national program COMMIT.

version of participatory sensing which may ultimately attract more people to participate. This argument becomes even more striking when the sensors are supposed to read very sensitive data such as data related to the personal health of participants. For instance, in the European Union (cf. European Data Protection Directive [23]), the data collector must prove sound security and stewardship of such sensitive data, which can be done through the use of provably secure cryptographic techniques.

The PEPSI Model. The only provably secure cryptographic treatment of participatory sensing so far is due to De Cristofaro and Soriente [15,17,16], who came up with a clear and concise infrastructural model and formally specified desirable privacy goals. Their model, called *PEPSI*, involves *mobile nodes* that sense and report data such as temperature, noise level, etc., forming the user basis for participatory sensing, *queriers* that represent entities (individuals or organizations) that consume sensed data such as "noise level on Time Square, New York", and an intermediate *service provider* that stores data reports received from mobile nodes and forwards the data to subscribed queriers. The service provider is an indispensable part of the infrastructure, needed to provide adequate efficiency and enable asynchronous communication between (resource-constrained) mobile nodes and queriers. However, its intermediary position, receiving both sensing data reports as well as interest subscriptions of queriers, induces additional privacy challenges, treated in PEPSI's corresponding privacy requirements.

Our Contribution. We show that although PEPSI contains formal definitions of privacy for participatory sensing, it leaves aside a very important security aspect, namely *collusion attacks* across different parties. In an application environment with many interacting mobile nodes and queriers, the possibility that some of them collude (potentially also with the service provider) in order to gain insight into the interests of others constitutes a realistic threat with devastating consequences on privacy. For instance, consider a scenario where a mobile node and a querier, who should be restricted to upload data (or obtain data, in the case of the querier) for registered interests only, follow the protocol honestly, but collude by exchanging their obtained keys. In PEPSI, these colluding parties (even if registered only for a single, identical interest), are able to obtain and decrypt sensor readings for any interest of their choice due to the lack of collusion resistance, thus completely breaching the privacy of all other mobile nodes in the system. Note that this form of collusion is already given when a user is registered as both a mobile node and a querier. This simple example illustrates the high importance of collusion resistance in participatory sensing for the protection of all participants' privacy.

We therefore revisit the PEPSI model and protocols from the perspective of collusion resistance and give more precise definitions for its three main privacy goals, namely *node privacy* that protects the content and nature of data reports, *query privacy* that hides the information for which queriers subscribe, and *report unlinkability* that guarantees untraceability of the reports submitted by mobile nodes. In order to distinguish both models, we refer to our model for a *privacy-enhanced participatory sensing infrastructure with collusion resistance* as PEPSICo. Subsequent to defining our extended security model, we give a generic and provably secure PEPSICo construction using identity-based encryption (IBE) and a concrete instantiation based on the Boneh-Franklin IBE scheme [3]. Our construction offers collusion resistance and enjoys particularly low computation, communication, and storage overhead.

Beyond this, in our model we additionally enable support for *data aggregation* at the service provider that, besides functional benefits for participatory sensing, helps to further reduce the communication overhead and to increase the privacy of individual reports. By sending only one aggregated report (with the size of a single one) instead of several single reports, aggregation reduces the amount of transferred data. Moreover, aggregated values hide the contained accumulated individual values, thus increasing the privacy of individual users.

For the purpose of data aggregation, we construct and analyze an *additively homomorphic* IBE scheme as a variant of the Boneh-Franklin IBE scheme and prove its security under the Decisional Bilinear Diffie-Hellman assumption in the random oracle model. This IBE scheme can be directly used within our generic collusion-resistant participatory sensing protocols to achieve data aggregation. We note that our additively homomorphic IBE scheme is only suitable for calculations with small sensor readings, which however is sufficient for most of the above mentioned applications of participatory sensing. For all our constructions, we analyze the performance and offer comparisons to prior work.

2 Related Work

Privacy challenges in participatory sensing were pointed out by many different researchers in the past, emphasizing their importance [43,30] or even suggesting the design of privacy-preserving data aggregation schemes [10], however without providing concrete solutions. One of the first privacy-aware architectures is *AnonySense* [12], aiming at k-anonymity [44] and using mix networks [8], however without providing confidentiality of reports or queries against the service provider. Later, an extension by Huang et al. [28] targets ℓ-diversity [33] but still relies on multiple non-colluding parties. Dimitriou et al. [19], with *PEPPeR*, aimed at the protection of querier privacy using blind signatures [9], however focused on querier privacy only and required direct communication between those and mobile nodes.

So far, the only framework that aims at cryptographically provable privacy is *PEPSI* by De Cristofaro and Soriente [15,17,16]. It is based on a simple but versatile architecture that involves a trusted setup for the key generation phase and an *untrusted* service provider for all later phases (see Section 3 for more details). In contrast to our work, PEPSI does not allow for data aggregation and, more importantly, does not protect against collusion attacks which has destructive implications on privacy as we show in Section 3.1.

In the context of secure data aggregation, a lot of work has been done in wireless sensor networks (see, e.g., [36]), though often focused on external adversaries. Castelluccia et al. [7], e.g., employ symmetric homomorphic encryption for data aggregation on the path to the service provider, receiving results in the clear. *PoolView* by Ganti et al. [25] is an aggregation approach based on perturbation, however designed for closed communities with a known user set and data distribution. Shi et al. [42] proposed with *PriSense* another aggregation approach for participatory sensing where fragmented data is reported via different paths using so-called cover nodes. For mobile nodes forming a communication ring a private aggregation scheme was introduced by Li et al. [31].

A different approach to match data reports with queries would be to incorporate encryption with keyword search, introduced in the public key setting by Boneh et al. [2] and later for the identity-based setting by Boyen et al. [6]. Keyword search however inherently allows an owner of a detection trapdoor for a keyword to identify this keyword in a given set, rendering anonymity against the service provider impossible in our setting.

3 The PEPSI Model

In this section we briefly recall the PEPSI model as introduced by De Cristofaro and Soriente [15,17,16]. Their infrastructure considers the following parties: *mobile nodes (MNs)* are the devices that sense and report data, *queriers* are end-users interested in receiving sensor reports, the *network operator (NO)* is the provider of cellular network access for MNs, the *service provider (SP)* is the intermediary party between MNs and queriers that relays matching reports to subscribed queriers, and the *registration authority (RA)* is the trusted party performing system setup and node registration. The following PEPSI construction was proposed in [15,17] using an encryption approach derived from the Boneh-Franklin IBE scheme [3]. It uses groups $\mathbb{G}$, $\mathbb{G}_T$ of prime order q with a generator $g \in \mathbb{G}$ and an efficient bilinear map $e : \mathbb{G} \times \mathbb{G} \mapsto \mathbb{G}_T$ such that $e(g^a, g^b) = e(g, g)^{ab}$ for all $a, b \in \mathbb{Z}_q$ and $e(g, h) \neq 1_{\mathbb{G}_T}$ whenever $g, h \neq 1_G$.

Setup: The RA generates the bilinear group parameters $(\mathbb{G} = \langle g \rangle, q, e \colon \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T)$, picks $s \in_R \mathbb{Z}_q^*$ as the master secret key msk and makes $Q := g^s$ public. Further, RA chooses a "nonce" $z \in_R \mathbb{Z}_q^*$, sets $R := g^z$, and fixes three cryptographic hash functions $H_1 \colon \{0,1\}^* \to \mathbb{G}$, $H_2, H_3 \colon \mathbb{G}_T \to \{0,1\}^n$.

MN Registration: A MN registers for the sensing of certain data at the RA and obtains the pair (z, id) where z is the "nonce" from Setup and id the identifier for the readings MN provides.

Query Registration: A querier registers at the RA for some query identifier id^* (e.g., "temperature in Berlin, Germany") and obtains the pair (sk_{id^*}, R) for $sk_{id^*} := H_1(id^*)^s$. It then subscribes at the SP to receive reports for id^* by sending $T^* := H_2(e(R, sk_{id^*}))$.

Data Report: In order to submit a data reading m, a MN sends the pair $(T, c) := (H_2(e(Q, H_1(id)^z)), \mathsf{Enc}_k(m))$ to the SP (via NO's infrastructure), with $k := H_3(e(Q, H_1(id)^z))$ being the key for some symmetric encryption operation Enc, e.g., AES. T is called a *tag*.

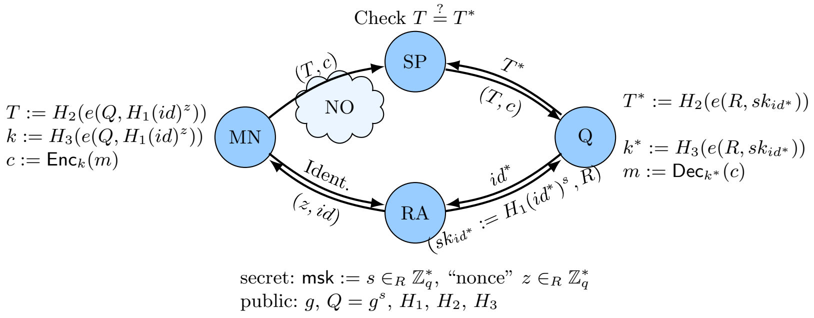

Check $T \overset{?}{=} T^*$
SP
(T, c)
NO
T^*
(T, c)
$T := H_2(e(Q, H_1(id)^z))$
$k := H_3(e(Q, H_1(id)^z))$
$c := \mathsf{Enc}_k(m)$
MN
Q
$T^* := H_2(e(R, sk_{id^*}))$
$k^* := H_3(e(R, sk_{id^*}))$
$m := \mathsf{Dec}_{k^*}(c)$
Ident.
(z, id)
RA
id^*
$(sk_{id^*} := H_1(id^*)^s, R)$
secret: $\mathsf{msk} := s \in_R \mathbb{Z}_q^*$, “nonce” $z \in_R \mathbb{Z}_q^*$
public: g, $Q = g^s$, H_1, H_2, H_3

Fig. 1. PEPSI scheme as proposed by De Cristofaro and Soriente [15,17,16].

Query Execution: The SP matches received reports with query subscriptions by comparing the tag T of a report with the stored subscriptions T^* and forwards matching reports (T, c) to the according queriers. The receiving querier computes $k^* := H_3(e(R, sk_{id^*}))$ and $m := \mathsf{Dec}_{k^*}(c)$.

Nonce Renewal: The RA periodically distributes a fresh z to the MNs and $R = g^z$ to the queriers in order to ban misbehaving MNs.

The PEPSI model identifies three *privacy goals*, stated here informally: *node privacy* requires that NO, SP, unauthorized queriers, and other MNs learn nothing about the data or nature (e.g., query id) of a report submitted by a MN; *query privacy* demands that NO, SP, MNs, and other queriers learn nothing about the query identifier a querier subscribes to; *report unlinkability* is achieved if no party can link multiple data reports as originating from the same MN.

3.1 Limitations of PEPSI

With PEPSI [15,17,16], De Cristofaro and Soriente proposed the first cryptographic framework for a formal analysis of security and privacy in participatory sensing. As mentioned earlier, their model however does not achieve the required *collusion resistance* necessary for a secure and privacy-protected participatory sensing infrastructure and cannot deal with *data aggregation* at the SP.

While PEPSI excludes some forms of collusions by trust assumptions (e.g., between the SP and queriers), two types of collusions remain unmentioned[4] which lead to serious privacy loopholes in their construction (full details can be found in Section 4.4):

- **Collusion of SP and MN.** All MNs possess the (same) “nonce” z allowing to compute key k and tag T for any identity. The colluding SP and MN can thus together decrypt all reports and determine the identity behind all query subscriptions the SP receives (breaking *node privacy* and *query privacy*).
- **Collusion of MN and Querier.** The colluding MN and querier can use the “nonce” z to subscribe for any identity (computing the resp. tag) and decrypt all received reports (computing the resp. key), thus breaking *node privacy.*[5]

Concerning data aggregation, De Cristofaro and Soriente acknowledge [16] that performing aggregation at the SP would be an expedient capability in the setting of participatory sensing; their constructions however only allow for single encrypted measurements.

[4] The original PEPSI paper [15, Section 4.2] only requires the SP not to collude with the RA or queriers, missing the collusion attacks mentioned here. In the later journal version [16, Section III.D], De Cristofaro and Soriente assume an honest-but-curious, non-colluding behavior of all participating parties, i.e., completely exclude the treatment of collusions in their model.

[5] We stress that if no (additional) identity management is implemented to authenticate queriers as such when interacting with the SP, this attack actually constitutes a *total privacy breach* as *every* mobile node can subscribe for *any* query identifier without registering as a querier and decrypt *all* received data reports. The collusion-resistant model we introduce eliminates this attack independent of identity management.

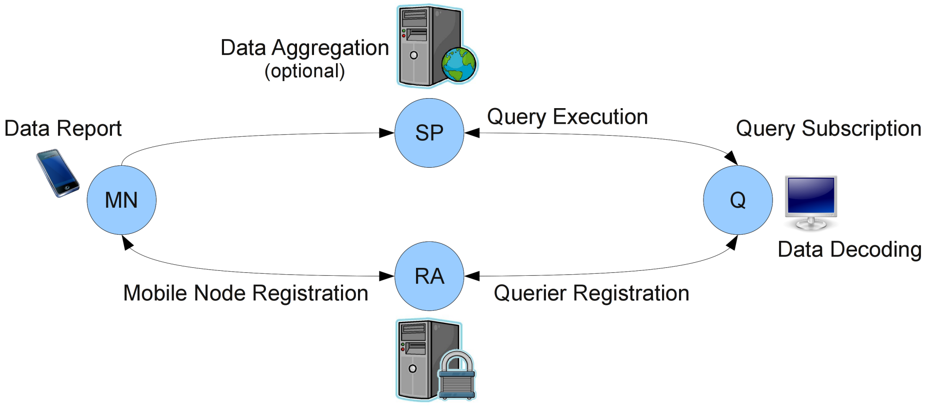


Fig. 2. The PEPSICo infrastructure. Mobile nodes (MNs) and queriers (Qs) register to the registration authority (RA). MNs report data to the service provider (SP), queriers subscribe for reports at the SP. The SP may aggregate multiple reports and sends reports matching with subscriptions to the according querier, which decodes them.

We argue that collusions, especially one person registering both as mobile node and querier but also—though to a lesser extent—between MNs and the SP, are a realistic threat in participatory sensing scenarios with devastating consequences on privacy within PEPSI. Therefore, meaningful forms of collusion resistance must also be reflected in the corresponding privacy definitions. Moreover, it would be desirable—both performance- and privacy-wise—to directly allow for aggregation of data reports in the underlying model. This motivates the following revision of the original PEPSI model.

4 PEPSICo: Revised Model for Participatory Sensing

In this section, we propose a revised model for a *privacy-enhanced participatory sensing infrastructure* which captures *collusion-resistance* and foresees optional *data aggregation*, denoted as PEPSICo.

4.1 The PEPSICo Model

The PEPSICo system model (cf. Figure 2) involves mobiles nodes (MNs), queriers, a service provider (SP), and a registration authority (RA) with identical roles as in the PEPSI model.

Mobile Nodes (MNs): Mobile nodes are devices carried by people or mobile entities that sense data and report it via, e.g., cellular networks to the service provider.

Queriers: Queriers are end-users that are interested in receiving sensor reports and register at the service provider for this purpose.

Service Provider (SP): The service provider is the connection party between mobile nodes and queriers that relays matching data reports to accordingly subscribed queriers.

Registration Authority (RA): The registration authority performs the system setup and handles the registration of participating parties.

We however drop the network operator, as its attack capabilities in our model are strictly weaker than those of the service provider. Thus, considering the latter only is sufficient.

Definition 1 (PEPSICo **Instantiation).** *An* instantiation of the privacy-enhanced participatory sensing infrastructure with collusion resistance *(*PEPSICo *instantiation)* PI *consists of the seven algorithms* Setup*,* RegisterMN, RegisterQ, ReportData, SubscribeQuery, ExecuteQuery*, and* DecodeData *and, potentially, the* optional AggregateData *algorithm defined as follows.*

$\texttt{Setup}(1^n)$**:** *The setup is executed by the RA to initialize* PI. *On input the security parameter $n \in \mathbb{N}$, this probabilistic algorithm outputs the RA's secret key* RAsk *and a master public key* RApk. RApk *contains a description of the query identity space $\mathcal{I}$ and the message space $\mathcal{M}$.*

$\texttt{RegisterMN}(\mathsf{RApk}, \mathsf{RAsk}, qid)$**:** *The MN registration is executed by the RA to register a new MN for a given query identity qid. On input* RApk, RAsk, *and a query identity $qid \in \mathcal{I}$, this probabilistic algorithm outputs a mobile node registration value* regMN_{qid} *for qid, which the RA sends to the MN.*

$\texttt{RegisterQ}(\mathsf{RApk}, \mathsf{RAsk}, qid)$**:** *The querier registration is executed by the RA to register a new querier for a given query identity $qid \in \mathcal{I}$. On input* RApk, RAsk, *and qid, this probabilistic algorithm outputs a querier registration value* regQ_{qid} *for qid, which the RA sends to the querier.*

$\texttt{ReportData}(\mathsf{RApk}, \mathsf{regMN}_{qid}, qid, m)$**:** *The data report algorithm is executed by the MN to report a message $m \in \mathcal{M}$ under some query identity $qid \in \mathcal{I}$. On input* RApk, *a MN registration value* regMN_{qid}, *qid, and m, this probabilistic algorithm outputs a data report c, which the MN sends to the SP.*

$\texttt{SubscribeQuery}(\mathsf{RApk}, \mathsf{regQ}_{qid}, qid)$**:** *The query subscription is executed by the querier to subscribe for a given query identity $qid \in \mathcal{I}$. On input* RApk, *a querier registration value* regQ_{qid}, *and qid, this probabilistic algorithm outputs a subscription token s, which the querier sends to the SP.*

$\texttt{ExecuteQuery}(\mathsf{RApk}, c, s)$**:** *The query execution is executed by the SP. On input the master public key* RApk, *a data report c, and a subscription token s, this deterministic algorithm outputs either c (indicating that c matches with s) or $\perp$ (indicating mismatch) to the querier who provided the token s.*

$\texttt{DecodeData}(\mathsf{RApk}, \mathsf{regQ}_{qid}, qid, c)$**:** *The data decoding is executed by a querier on a received data report c to obtain the contained message. On input* RApk, *a querier registration value* regQ_{qid}, *a query identity $qid \in \mathcal{I}$, and c, this deterministic algorithm outputs either a message m or $\perp$, indicating failure.*

$\texttt{AggregateData}(\mathsf{RApk}, \mathbf{c})$**:** *The* optional *data aggregation is executed by the SP on a vector of data reports $\mathbf{c} = (c_1, \ldots, c_k)$ and, if all match, outputs a single, aggregated data report. On input* RApk *and* $\mathbf{c}$, *this probabilistic algorithm outputs either a single data report c or $\perp$, indicating failure.*

If PI *provides the* $\mathsf{AggregateData}$ *operation, it is called a* $\mathsf{PEPSICo}$ instantiation with data aggregation*. To be* sound*, a* $\mathsf{PEPSICo}$ *instantiation* PI *has to satisfy the condition that data reports match with query subscriptions and are decodable using the querier registration value generated for the same query identity, even if they were previously aggregated by the service provider.*

4.2 Trust Assumptions and Adversary Model

In our model, we allow collusions between the SP, mobile nodes, and queriers against other mobile nodes or queriers. Particularly, we consider mobile nodes to be arbitrary, unauthenticated users. Since our model aims at the higher-level application of participatory sensing it is assumed that (uncorrupted) parties communicate over confidential yet not necessarily authenticated channels.

In order to define security and privacy of a $\mathsf{PEPSICo}$ instantiation PI, we consider a probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ interacting with PI. We allow for corruptions of MNs, queriers, the SP, and (in special cases) the RA. Let $\mathcal{CI}_{MN}$ resp. $\mathcal{CI}_Q$ denote the set of identities $\mathcal{A}$ learned registration values for by corrupting MNs resp. queriers and $\mathcal{CI} := \mathcal{CI}_{MN} \cup \mathcal{CI}_Q$. Corruption of the SP resp. RA is denoted by $\mathcal{C}_{SP} = 1$ resp. $\mathcal{C}_{RA} = 1$; initially both are 0. $\mathcal{A}$ has access to the following oracles:

$\mathsf{CorruptMN}(qid)$**:** On input a query id qid, compute $\mathsf{regMN}_{qid} \leftarrow \texttt{RegisterMN}(\mathsf{RApk}, \mathsf{RAsk}, qid)$, provide $\mathcal{A}$ with regMN_{qid}, and add qid to $\mathcal{CI}_{MN}$.

$\mathsf{CorruptQ}(qid)$**:** On input a query id qid, compute $\mathsf{regQ}_{qid} \leftarrow \texttt{RegisterQ}(\mathsf{RApk}, \mathsf{RAsk}, qid)$, provide $\mathcal{A}$ with regQ_{qid}, and add qid to $\mathcal{CI}_Q$.

$\mathsf{CorruptSP}()$**:** Set flag $\mathcal{C}_{SP} := 1$. (This influences subsequent $\mathsf{ReportData}$ queries.)

$\mathsf{CorruptRA}()$**:** Provide $\mathcal{A}$ with RAsk and set flag $\mathcal{C}_{RA} := 1$.

$\mathsf{ReportData}(qid, m, \mathbf{s})$**:** On input a query id qid, a message m, and a vector of subscription tokens $\mathbf{s} = (s_1, \ldots, s_k)$, let $\mathsf{regMN}_{qid} \leftarrow \texttt{RegisterMN}(\mathsf{RApk}, \mathsf{RAsk}, qid)$ and $c \leftarrow \texttt{ReportData}(\mathsf{RApk}, \mathsf{regMN}_{qid}, qid, m)$.

If $\mathcal{C}_{SP} = 1$, c is given to $\mathcal{A}$. Otherwise $\mathcal{A}$ receives $\mathbf{c} := (c_1, \ldots, c_k)$, where $c_i \leftarrow \texttt{ExecuteQuery}(\mathsf{RApk}, c, s_i)$ for $i \in \{1, \ldots, k\}$ (some of the c_i may be $\perp$).[6]

$\mathsf{SubscribeQuery}(qid)$**:** On input a query id qid, compute $\mathsf{regQ}_{qid} \leftarrow \texttt{RegisterQ}(\mathsf{RApk}, \mathsf{RAsk}, qid)$, $s \leftarrow \texttt{SubscribeQuery}(\mathsf{RApk}, \mathsf{regQ}_{qid}, qid)$, and give s to $\mathcal{A}$.

$\mathsf{DecodeData}(qid, c)$**:** On input a query id qid and a data report c, compute $\mathsf{regQ}_{qid} \leftarrow \texttt{RegisterQ}(\mathsf{RApk}, \mathsf{RAsk}, qid)$, $m \leftarrow \texttt{DecodeData}(\mathsf{RApk}, \mathsf{regQ}_{qid}, qid, c)$, and give m to $\mathcal{A}$.

4.3 Privacy and Security Definitions

We proceed by strengthening the definitions of the three central privacy goals for participatory sensing identified in [15] with collusion-resistance.

Node Privacy. Our notion of *node privacy* formalizes confidentiality of data reports against the SP, unauthorized queriers, and other MNs. More precisely, node privacy hides both the message and the query identity of a report from these parties, even if all of them collude. We thus model node privacy as indistinguishability of data reports generated using two query identity-message pairs freely chosen by an adaptive adversary that can obtain data reports, subscribe to queries, and corrupt SP as well as MNs and queriers for other query identities. Similar to classical security notions for encryption, we distinguish between node privacy under chosen-ciphertext and under chosen-plaintext attacks, where in the first the adversary has additional access to the decoding oracle.

Definition 2 (Node Privacy). *Let* PI *be a* PEPSICo *instantiation and* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *a PPT adversary interacting with* PI *via the queries defined in Section 4.2 within the following game* $\mathrm{Game}^{\mathsf{NP\text{-}CCA}}_{\mathsf{PI},\mathcal{A}}(n)$*:*

Setup. $\texttt{Setup}(1^n)$ *is executed and outputs* $(\mathsf{RAsk}, \mathsf{RApk})$.

Phase I. $\mathcal{A}_1$ *receives* RApk *and has access to the oracles* CorruptMN*,* CorruptQ*,* CorruptSP*,* ReportData*,* SubscribeQuery*, and* DecodeData*. Eventually,* $\mathcal{A}_1$ *stops and outputs two challenge query identity-message pairs* (qid_0, m_0)*,* (qid_1, m_1) *and a vector of subscription tokens* $\mathbf{s} = (s_1, \ldots, s_k)$*.*

Challenge. *A bit* $b \in_R \{0,1\}$ *is chosen,* $\mathsf{regMN}_{qid_b} \leftarrow \texttt{RegisterMN}(\mathsf{RApk}, \mathsf{RAsk}, qid_b)$ *and* $c \leftarrow \texttt{ReportData}(\mathsf{RApk}, \mathsf{regMN}_{qid_b}, qid_b, m_b)$ *are executed. If* $\mathcal{C}_{SP} = 1$*, set* $\mathbf{R} := (c)$*, otherwise set* $\mathbf{R} := (c_1, \ldots, c_k)$*, where* $c_i \leftarrow \texttt{ExecuteQuery}(\mathsf{RApk}, c, s_i)$ *for* $i \in \{1, \ldots, k\}$*.*

Phase II. $\mathcal{A}_2$ *receives* RApk *and* $\mathbf{R}$ *and has access to the oracles from Phase I.*

Guess. *Eventually,* $\mathcal{A}_2$ *outputs a guess* $b' \in \{0,1\}$ *for* b*.*

Adversary $\mathcal{A}$ *wins the game, denoted by* $\mathrm{Game}^{\mathsf{NP\text{-}CCA}}_{\mathsf{PI},\mathcal{A}}(n) = 1$*, if* $b = b'$*,* $\{qid_0, qid_1\} \cap \mathcal{CI} = \emptyset$*, and all the following conditions hold:*

1. $\mathcal{A}$ *did not query* SubscribeQuery *with* qid_0 *or* qid_1*.*
2. *If* $\mathcal{C}_{SP} = 1$*, then* $\mathcal{A}$ *did not query* ReportData *with* qid_0 *or* qid_1*.*
3. *In Phase II* $\mathcal{A}$ *did not query* $\mathsf{DecodeData}(qid_0, \mathbf{R}[i])$ *or* $\mathsf{DecodeData}(qid_1, \mathbf{R}[i])$ *for any element* $\mathbf{R}[i]$ *of* $\mathbf{R}$*.*

We say PI *provides* node privacy under chosen-ciphertext attacks *(or* NP-CCA *security) if for all PPT adversaries* $\mathcal{A}$ *the following advantage function is negligible in* n*:*

$$\mathrm{Adv}^{\mathsf{NP\text{-}CCA}}_{\mathsf{PI},\mathcal{A}}(n) := \left|\Pr\left[\mathrm{Game}^{\mathsf{NP\text{-}CCA}}_{\mathsf{PI},\mathcal{A}}(n) = 1\right] - \frac{1}{2}\right|.$$

Consider the game $\mathrm{Game}^{\mathsf{NP\text{-}CPA}}_{\mathsf{PI},\mathcal{A}}(n)$*, which is identical to* $\mathrm{Game}^{\mathsf{NP\text{-}CCA}}_{\mathsf{PI},\mathcal{A}}(n)$*, except that* $\mathcal{A}$ *is not given access to the* DecodeData *oracle. We say* PI *provides* node privacy under chosen-plaintext attacks *(or* NP-CPA *security) if for all PPT adversaries* $\mathcal{A}$ *the analogously defined advantage* $\mathrm{Adv}^{\mathsf{NP\text{-}CPA}}_{\mathsf{PI},\mathcal{A}}(n)$ *is negligible in* n*.*

[6] The intuition of separating the cases $\mathcal{C}_{SP} = 1$ and $\mathcal{C}_{SP} = 0$ (i.e., SP is corrupted or not) is as follows: If SP is corrupted, $\mathcal{A}$ sees any data report sent to SP. Otherwise, $\mathcal{A}$ only learns reports for which he can provide a matching subscription token s_i.

Remark 1. $\mathsf{PEPSICo}$ schemes with data aggregation *never* provide $\mathsf{NP\text{-}CCA}$ security, as an adversary in $\mathrm{Game}^{\mathsf{NP\text{-}CCA}}_{\mathsf{PI},\mathcal{A}}(n)$ can apply the $\mathtt{AggregateData}$ algorithm on challenge c and a c' for known m' and decode the result using the $\mathtt{DecodeData}$ oracle. Therefore, the desirable privacy flavor in case of aggregation is $\mathsf{NP\text{-}CPA}$.

Query Privacy. By *query privacy* we formalize the privacy of queriers when subscribing for query identities. We require that the query identity of a subscription is hidden from the SP as well as MNs and other queriers, even if all of them collude. Query privacy is thus modeled as indistinguishability of subscription tokens for two query identities freely chosen by an adaptive adversary that can obtain data reports, subscribe to queries, and corrupt SP as well as MNs and queriers for other query identities.

Definition 3 (Query Privacy). *Let* PI *be a* $\mathsf{PEPSICo}$ *instantiation and* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *a PPT adversary interacting with* PI *via the queries defined in Section 4.2 within the following game* $\mathrm{Game}^{\mathsf{QP}}_{\mathsf{PI},\mathcal{A}}(n)$*:*

Setup. $\mathtt{Setup}(1^n)$ *is executed and outputs* $(\mathsf{RAsk}, \mathsf{RApk})$*; set* $\mathcal{C}_{SP} := 1$.
Phase I. $\mathcal{A}_1$ *receives* RApk *and has access to the oracles* $\mathsf{CorruptMN}$*,* $\mathsf{CorruptQ}$*,* $\mathsf{ReportData}$*,* $\mathsf{SubscribeQuery}$*, and* $\mathsf{DecodeData}$*. Eventually,* $\mathcal{A}_1$ *stops and outputs two challenge query identities* qid_0 *and* qid_1*.*
Challenge. *A bit* $b \in_R \{0,1\}$ *is chosen,* $\mathsf{regQ}_{qid_b} \leftarrow \mathtt{RegisterQ}(\mathsf{RApk}, \mathsf{RAsk}, qid_b)$ *and* $s \leftarrow \mathtt{SubscribeQuery}(\mathsf{RApk}, \mathsf{regQ}_{qid_b}, qid_b)$ *are executed.*
Phase II. $\mathcal{A}_2$ *receives* RApk *and* s *and has access to the oracles from Phase I.*
Guess. *Eventually,* $\mathcal{A}_2$ *outputs a guess* $b' \in \{0,1\}$ *for* b*.*

Adversary $\mathcal{A}$ *wins the game, denoted by* $\mathrm{Game}^{\mathsf{QP}}_{\mathsf{PI},\mathcal{A}}(n) = 1$*, if* $b = b'$*,* $\{qid_0, qid_1\} \cap \mathcal{CI} = \emptyset$*, and* $\mathcal{A}$ *did not query* $\mathsf{ReportData}$ *or* $\mathsf{SubscribeQuery}$ *with* qid_0 *or* qid_1*. We say* PI *provides* query privacy *if for all PPT adversaries* $\mathcal{A}$ *the following advantage function is negligible in* n*:*

$$\mathrm{Adv}^{\mathsf{QP}}_{\mathsf{PI},\mathcal{A}}(n) := \left| \Pr\left[\mathrm{Game}^{\mathsf{QP}}_{\mathsf{PI},\mathcal{A}}(n) = 1\right] - \frac{1}{2} \right| .$$

Report Unlinkability. *Report unlinkability* prevents the linkage of two reports as originating from the same MN by any other party, *including* the RA. As MNs (as well as queriers) are not distinguished by device identifiers or anything similar in our model, we tie the notion of report unlinkability to the MN registration value used to generate a data report. We model report unlinkability as indistinguishability of the MN registration value used to generate a data report for a query identity-message pair freely chosen by an adaptive adversary that can obtain data reports, subscribe to queries, and corrupt SP, any MN and querier as well as the RA (after setup).

Definition 4 (Report Unlinkability). *Let* PI *be a* $\mathsf{PEPSICo}$ *instantiation and* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *a PPT adversary interacting with* PI *via the queries defined in Section 4.2 within the following game* $\mathrm{Game}^{\mathsf{RU}}_{\mathsf{PI},\mathcal{A}}(n)$*:*

Setup. $\mathtt{Setup}(1^n)$ *is executed and outputs* $(\mathsf{RAsk}, \mathsf{RApk})$*.*
Phase I. $\mathcal{A}_1$ *receives* RApk *and has access to the oracles* $\mathsf{CorruptMN}$*,* $\mathsf{CorruptQ}$*,* $\mathsf{CorruptSP}$*,* $\mathsf{CorruptRA}$*,* $\mathsf{ReportData}$*,* $\mathsf{SubscribeQuery}$*, and* $\mathsf{DecodeData}$*. Eventually,* $\mathcal{A}_1$ *stops and outputs a challenge query identity-message pair* (qid, m)*.*
Challenge. $\mathtt{RegisterMN}(\mathsf{RApk}, \mathsf{RAsk}, qid)$ *is executed twice, resulting in registration values* regMN^0_{qid}*,* regMN^1_{qid}*. A bit* $b \in_R \{0,1\}$ *is chosen and* $c \leftarrow \mathtt{ReportData}(\mathsf{RApk}, \mathsf{regMN}^b_{qid}, qid, m)$ *is executed.*
Phase II. $\mathcal{A}_2$ *receives* RApk*,* regMN^0_{qid}*,* regMN^1_{qid}*, and* c *and has access to the oracles from Phase I.*
Guess. *Eventually,* $\mathcal{A}_2$ *outputs a guess* $b' \in \{0,1\}$ *for* b*.*

Adversary $\mathcal{A}$ *wins the game, denoted by* $\mathrm{Game}^{\mathsf{QP}}_{\mathsf{PI},\mathcal{A}}(n) = 1$*, if* $b = b'$*. We say* PI *provides* report unlinkability *if for all PPT adversaries* $\mathcal{A}$ *the following advantage function is negligible in* n*:*

$$\mathrm{Adv}^{\mathsf{RU}}_{\mathsf{PI},\mathcal{A}}(n) := \left| \Pr\left[\mathrm{Game}^{\mathsf{RU}}_{\mathsf{PI},\mathcal{A}}(n) = 1\right] - \frac{1}{2} \right| .$$

4.4 Collusion Attacks against PEPSI

We conclude the exposition of our model by showing formally that the PEPSI construction [15,17,16] does not satisfy our requirements on node and query privacy due to collusion attacks. More precisely, considering PEPSI as an instance of our model, we specify collusion attacks against the two privacy properties that exploit corrupted mobile nodes:

Collusion against Node Privacy: $\mathcal{A}_1$ calls $\mathsf{CorruptSP}$ and outputs arbitrary but distinct (qid_0, m_0), (qid_1, m_1) and empty $\mathbf{s} = ()$. $\mathcal{A}_2$ receives $c = (T, c')$, calls $\mathsf{CorruptMN}(qid')$ for $qid' \notin \{qid_0, qid_1\}$, and receives $\mathsf{regMN}_{qid'} = z$. $\mathcal{A}_2$ computes $T_0 := H_2(e(Q, H_1(id_0)^z))$ and, if $T_0 = T$, outputs 0, otherwise 1. $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ always wins.

Collusion against Query Privacy: $\mathcal{A}_1$ outputs arbitrary but distinct qid_0, qid_1. $\mathcal{A}_2$ receives $s = T$, calls $\mathsf{CorruptMN}(qid')$ for $qid' \notin \{qid_0, qid_1\}$, and receives $\mathsf{regMN}_{qid'} = z$. $\mathcal{A}_2$ computes $T_0 := H_2(e(Q, H_1(id_0)^z))$ and, if $T_0 = T$, outputs 0, otherwise 1. $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ always wins.

Therefore, a new approach is required to maintain privacy protection in a participatory sensing scenario facing collusion attacks, which we present now.

5 A Generic Solution using Identity-Based Encryption

Preliminaries. Before introducing our generic instantiation, we briefly recap the definitions of an IBE scheme and the according indistinguishability and anonymity notions due to Boneh and Franklin [3] and Abdalla et al. [1] we build upon.

Definition 5 (Identity-Based Encryption). *An* identity-based encryption scheme *(IBE scheme) $\mathcal{E}$ consists of four algorithms:*

- $\mathsf{Setup}(1^n)$ *generates the master public resp. secret key* mpk *and* msk.
- $\mathsf{Extract}(\mathsf{mpk}, \mathsf{msk}, id)$ *outputs the private key sk_{id} corresponding to an identity $id \in \{0,1\}^*$.*
- $\mathsf{Enc}(\mathsf{mpk}, id, m)$ *encrypts a message $m \in \mathcal{M}$ to a ciphertext $c \in \mathcal{C}$.*
- $\mathsf{Dec}(\mathsf{mpk}, sk_{id}, c)$ *decrypts a ciphertext $c \in \mathcal{C}$ to a message $m \in \mathcal{M}$.*

An IBE scheme $\mathcal{E}$ is correct *if for all $id \in \{0,1\}^*$, $m \in \mathcal{M}$: $\mathsf{Dec}(\mathsf{mpk}, \mathsf{Extract}(\mathsf{mpk}, \mathsf{msk}, id), \mathsf{Enc}(\mathsf{mpk}, id, m)) = m$.*

Homomorphic IBE: *An IBE scheme $\mathcal{E}$ is* homomorphic *if for all $id \in \{0,1\}^*$, all $m, m' \in \mathcal{M}$: $\mathsf{Dec}(\mathsf{mpk}, \mathsf{Extract}(\mathsf{mpk}, \mathsf{msk}, id), \mathsf{Enc}(\mathsf{mpk}, id, m) \circ \mathsf{Enc}(\mathsf{mpk}, id, m')) = m + m'$, where $\circ : \mathcal{C} \times \mathcal{C} \mapsto \mathcal{C}$ and $+ : \mathcal{M} \times \mathcal{M} \mapsto \mathcal{M}$.*

Definition 6 (Security Notions for IBE). *Let $\mathcal{E}$ be an IBE scheme and $\mathcal{A}$ a PPT adversary in the following* $\mathsf{ANO\text{-}IND\text{-}ID\text{-}CCA}$ *game:*

$$\begin{aligned}
\mathrm{Game}^{\mathsf{ANO\text{-}IND\text{-}ID\text{-}CCA}}_{\mathcal{E},\mathcal{A}}(n)\colon\ & (\mathsf{msk}, \mathsf{mpk}) \leftarrow \mathsf{Setup}(1^n) \\
& ((id_0, m_0), (id_1, m_1)) \leftarrow \mathcal{A}^{\mathsf{Extract}(\mathsf{mpk},\mathsf{msk},\cdot),\mathsf{Dec}(sk_{id},\cdot)}(\mathsf{mpk}) \\
& c \leftarrow \mathsf{Enc}(\mathsf{mpk}, id_b, m_b) \quad \text{for } b \in_R \{0,1\} \\
& b' \leftarrow \mathcal{A}^{\mathsf{Extract}(\mathsf{mpk},\mathsf{msk},\cdot),\mathsf{Dec}(sk_{id},\cdot)}(\mathsf{mpk}, c) \\
& \text{return } b = b'
\end{aligned}$$

where $\mathcal{A}$ must not query the $\mathsf{Extract}$ *oracle on id_0, id_1 nor the* Dec *oracle on sk_{id_0}, sk_{id_1} and the challenge c. The advantage of $\mathcal{A}$ in winning is defined as $\mathrm{Adv}^{\mathsf{ANO\text{-}IND\text{-}ID\text{-}CCA}}_{\mathcal{E},\mathcal{A}}(n) := \left|\Pr\left[\mathrm{Game}^{\mathsf{ANO\text{-}IND\text{-}ID\text{-}CCA}}_{\mathcal{E},\mathcal{A}}(n) = 1\right] - \frac{1}{2}\right|$. We say $\mathcal{E}$ provides* anonymity and indistinguishability under chosen-ciphertext attacks *(or* $\mathsf{ANO\text{-}IND\text{-}ID\text{-}CCA}$ *security) if for all $\mathcal{A}$ this advantage is negligible in n.*

There are further variants: If we require $id_0 = id_1$, the resulting game models only indistinguishability ($\mathsf{IND\text{-}ID\text{-}CCA}$ *security); for $m_0 = m_1$ only anonymity (*$\mathsf{ANO\text{-}ID\text{-}CCA}$ *security). Removing the* Dec *oracle results in the respective* chosen-plaintext *variants (*$\mathsf{ANO\text{-}IND\text{-}ID\text{-}CPA}$, $\mathsf{IND\text{-}ID\text{-}CPA}$, *resp.* $\mathsf{ANO\text{-}ID\text{-}CPA}$ *security).*

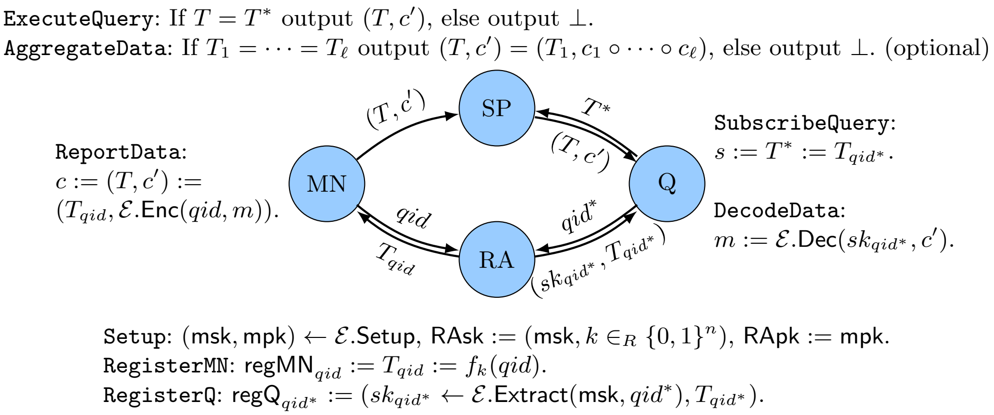

ExecuteQuery: If $T = T^*$ output (T, c'), else output $\perp$.
AggregateData: If $T_1 = \cdots = T_\ell$ output $(T, c') = (T_1, c_1 \circ \cdots \circ c_\ell)$, else output $\perp$. (optional)
SP
(T, c')
T^*
(T, c')
SubscribeQuery:
$s := T^* := T_{qid^*}$.
ReportData:
$c := (T, c') :=$
$(T_{qid}, \mathcal{E}.\mathsf{Enc}(qid, m))$.
MN
Q
qid
qid^*
DecodeData:
$m := \mathcal{E}.\mathsf{Dec}(sk_{qid^*}, c')$.
T_{qid}
RA
(sk_{qid^*}, T_{qid^*})
Setup: $(\mathsf{msk}, \mathsf{mpk}) \leftarrow \mathcal{E}.\mathsf{Setup}$, $\mathsf{RAsk} := (\mathsf{msk}, k \in_R \{0,1\}^n)$, $\mathsf{RApk} := \mathsf{mpk}$.
RegisterMN: $\mathsf{regMN}_{qid} := T_{qid} := f_k(qid)$.
RegisterQ: $\mathsf{regQ}_{qid^*} := (sk_{qid^*} \leftarrow \mathcal{E}.\mathsf{Extract}(\mathsf{msk}, qid^*), T_{qid^*})$.

Fig. 3. Generic PEPSICo instantiation $\mathsf{PI}_{\mathsf{IBE}}$ based on an IBE scheme $\mathcal{E}$ and a PRF f.

Lemma 1. *An IBE scheme is* ANO-ID-CCA- *and* IND-ID-CCA-*secure if and only if it is* ANO-IND-ID-CCA-*secure. The same holds for the chosen-plaintext case.*

5.1 Generic $\mathsf{PI}_{\mathsf{IBE}}$ Construction using Identity-Based Encryption

Our generic PEPSICo scheme, denoted $\mathsf{PI}_{\mathsf{IBE}}$ and specified in Definition 7, incorporates an IBE scheme $\mathcal{E}$ and a pseudorandom function (PRF) $f\colon \{0,1\}^n \times \{0,1\}^* \to \{0,1\}^n$. Figure 3 illustrates its mapping to the PEPSICo infrastructure.

Definition 7 **($\mathsf{PI}_{\mathsf{IBE}}$ Scheme).** *Let* $\mathcal{E} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ *be an identity-based encryption scheme and* $f\colon \{0,1\}^n \times \{0,1\}^* \to \{0,1\}^n$ *a pseudorandom function. The* $\mathsf{PI}_{\mathsf{IBE}}$ *scheme is defined as follows:*

Setup(1^n): *Let* $(\mathsf{msk}, \mathsf{mpk}) \leftarrow \mathsf{Setup}(1^n)$ *and* $k \in_R \{0,1\}^n$. *Output* $\mathsf{RAsk} := (\mathsf{msk}, k)$ *and* $\mathsf{RApk} := \mathsf{mpk}$. $\mathcal{M}$ *is the message space of* $\mathcal{E}$, $\mathcal{I} = \{0,1\}^*$.
RegisterMN$(\mathsf{RApk}, \mathsf{RAsk}, qid)$: *Let* $T_{qid} := f_k(qid)$, *output* $\mathsf{regMN}_{qid} := T_{qid}$.
RegisterQ$(\mathsf{RApk}, \mathsf{RAsk}, qid)$: *Let* $sk_{qid} \leftarrow \mathsf{Extract}(\mathsf{mpk}, \mathsf{msk}, qid)$ *and compute* $T_{qid} := f_k(qid)$. *Output* $\mathsf{regQ}_{qid} := (sk_{qid}, T_{qid})$.
ReportData$(\mathsf{RApk}, \mathsf{regMN}_{qid}, qid, m)$: *Output* $c := (T_{qid}, \mathsf{Enc}(\mathsf{mpk}, qid, m))$.
SubscribeQuery$(\mathsf{RApk}, \mathsf{regQ}_{qid}, qid)$: *Output* $s := T_{qid}$.
ExecuteQuery(RApk, c, s): *Parse* c *as* (T, c'). *If* $T = s$ *output* c, *else output* $\perp$.
DecodeData$(\mathsf{RApk}, \mathsf{regQ}_{qid}, qid, c)$: *Parse* c *as* (T, c'). *Output* $m := \mathsf{Dec}(\mathsf{mpk}, sk_{qid}, c')$.

If $\mathcal{E}$ *is homomorphic (cf. Definition 5) w.r.t. some operation* $\circ$, *then* $\mathsf{PI}_{\mathsf{IBE}}$ *supports* data aggregation *using the following generic algorithm:*

AggregateData$(\mathsf{RApk}, \mathbf{c})$: *Parse* $\mathbf{c}$ *as* $((T_1, c_1), \ldots, (T_\ell, c_\ell))$. *If* $T_1 = \cdots = T_\ell$, *compute* $c' = c_1 \circ c_2 \circ \cdots \circ c_\ell$ *and output* $c = (T_1, c')$, *otherwise output* $\perp$.

Soundness of $\mathsf{PI}_{\mathsf{IBE}}$ follows from the correctness (and for data aggregation also the additive homomorphism) of $\mathcal{E}$.

Remark 2. While $\mathsf{PI}_{\mathsf{IBE}}$ uses $T_{qid} := f_k(qid)$ to match reports with subscriptions, one could further accommodate time periods to indicate validity of reports/subscriptions and ensure unlinkability of tags across different time periods. This could for example be achieved by requiring the mobile node resp. querier to compute $H(T_{qid}, \mathsf{tp})$ for the current value T_{qid} (which still has to be kept secret from the adversary) and a time period tp using a collision-resistant hash function H and treat this values as part of the report c and the subscription token s.

5.2 Security Analysis

We obtain the following security result for $\mathsf{PI}_{\mathsf{IBE}}$.

Theorem 1 (Privacy and Security of $\mathsf{PI}_{\mathsf{IBE}}$). *If f is pseudorandom and $\mathcal{E}$ provides* ANO-IND-ID-CCA *(resp.* ANO-IND-ID-CPA*) security, then* $\mathsf{PI}_{\mathsf{IBE}}$ *provides node privacy under chosen-ciphertext (resp. chosen-plaintext) attacks, query privacy, and report unlinkability as defined in Definitions 2, 3, and 4.*

Proof of Node Privacy. We prove node privacy of $\mathsf{PI}_{\mathsf{IBE}}$ in two steps: First, we first replace the pseudorandom function f with a real random one and prove this to be indistinguishable. We then show how an adversary against the instantiation with a random function can be used to break the security of the underlying IBE scheme $\mathcal{E}$.

Assume we have an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against $\mathsf{PI}_{\mathsf{IBE}}$ with non-negligible advantage $\mathrm{Adv}^{\mathsf{NP\text{-}CCA}}_{\mathsf{PI}_{\mathsf{IBE}},\mathcal{A}}(n)$.[7] We first consider the game $\mathrm{Game}^{\mathsf{NP\text{-}CCA}^*}_{\mathsf{PI}_{\mathsf{IBE}},\mathcal{A}}(n)$, which is like $\mathrm{Game}^{\mathsf{NP\text{-}CCA}}_{\mathsf{PI}_{\mathsf{IBE}},\mathcal{A}}(n)$, except that instead of f a real random function $g\colon \{0,1\}^n \times \{0,1\}^* \to \{0,1\}^n$ is used to compute the tags T_{qid}. We argue that $\varepsilon(n) := \left|\mathrm{Adv}^{\mathsf{NP\text{-}CCA}}_{\mathsf{PI}_{\mathsf{IBE}},\mathcal{A}}(n) - \mathrm{Adv}^{\mathsf{NP\text{-}CCA}^*}_{\mathsf{PI}_{\mathsf{IBE}},\mathcal{A}}(n)\right|$ is negligible, otherwise $\mathcal{A}$ can be used to construct a distinguisher $\mathcal{D}$ between f and g by relaying evaluations of f in the game to its oracle. If $\mathcal{D}$ is given oracle access to f, then it acts like the challenger in $\mathrm{Game}^{\mathsf{NP\text{-}CCA}}_{\mathsf{PI}_{\mathsf{IBE}},\mathcal{A}}(n)$, otherwise like in $\mathrm{Game}^{\mathsf{NP\text{-}CCA}^*}_{\mathsf{PI}_{\mathsf{IBE}},\mathcal{A}}(n)$. $\mathcal{D}$ outputs the game result (i.e., $b = b'$) as its own guess and thus has advantage ε to distinguish f and g. As f by assumption is pseudorandom, ε is negligible.

Thus $\mathcal{A}$'s advantage in the modified game $\mathrm{Game}^{\mathsf{NP\text{-}CCA}^*}_{\mathsf{PI}_{\mathsf{IBE}},\mathcal{A}}(n)$ is non-negligible, too. We construct an adversary $\mathcal{B}$ with non-negligible advantage in breaking the ANO-IND-ID-CCA security of $\mathcal{E}$ which uses $\mathcal{A}$ as follows.

Setup. $\mathcal{B}$ receives the master public key mpk in the ANO-IND-ID-CCA game.

Phase I. $\mathcal{B}$ provides $\mathcal{A}_1$ with $\mathsf{RApk} = \mathsf{mpk}$ and answers the oracle queries as specified. It uses its Extract oracle to obtain secret keys sk_{qid} for CorruptQ queries, chooses tags $T_{qid} \in_R \{0,1\}^n$ at random on first request (reusing the value later), and relays DecodeData queries to its own Dec oracle.

$\mathcal{A}_1$ eventually outputs (qid_0, m_0), (qid_1, m_1), and $\mathbf{s} = (s_1, \ldots, s_k)$.

Challenge. $\mathcal{B}$ forwards $(qid_0, m_0), (qid_1, m_1)$ as its own challenge and receives c^*. $\mathcal{B}$ chooses $T \in_R \{0,1\}^n$ and sets $c := (T, c^*)$. If $\mathcal{C}_{SP} = 1$, $\mathcal{B}$ sets $\mathbf{R} := (c)$, else $\mathbf{R} := (c_1, \ldots, c_k)$ for $c_i \leftarrow \mathtt{ExecuteQuery}(\mathsf{RApk}, c, s_i)$.

Phase II. $\mathcal{B}$ provides $\mathcal{A}_2$ with RApk and $\mathbf{R}$ and answers queries as above.

Guess. $\mathcal{A}_2$ outputs a guess $b' \in \{0,1\}$, which $\mathcal{B}$ forwards as its own guess.

As $\mathcal{B}$ perfectly simulates $\mathrm{Game}^{\mathsf{NP\text{-}CCA}^*}_{\mathsf{PI}_{\mathsf{IBE}},\mathcal{A}}(n)$ for $\mathcal{A}$, we obtain $\mathrm{Adv}^{\mathsf{ANO\text{-}IND\text{-}ID\text{-}CCA}}_{\mathcal{E},\mathcal{B}}(n) = \mathrm{Adv}^{\mathsf{NP\text{-}CCA}^*}_{\mathsf{PI}_{\mathsf{IBE}},\mathcal{A}}(n)$, which is non-negligible. □

Note that any $\mathsf{PI}_{\mathsf{IBE}}$ construction *with* data aggregation can only provide node privacy under chosen-plaintext attacks (cf. Remark 1). Data aggregation however introduces additional privacy benefits: if queriers only receive aggregated values (e.g., a sum) then individual measurements submitted by mobile nodes remain to some extent hidden from potential queriers.

Proof of Query Privacy. Assume we have an adversary $\mathcal{A}$ against $\mathsf{PI}_{\mathsf{IBE}}$ with non-negligible advantage $\mathrm{Adv}^{\mathsf{QP}}_{\mathsf{PI}_{\mathsf{IBE}},\mathcal{A}}(n)$. Similar to the node privacy proof we consider $\mathrm{Game}^{\mathsf{QP}^*}_{\mathsf{PI}_{\mathsf{IBE}},\mathcal{A}}(n)$, which is identical to $\mathrm{Game}^{\mathsf{QP}}_{\mathsf{PI}_{\mathsf{IBE}},\mathcal{A}}(n)$, except that instead of f a real random function g is used to compute the tags T_{qid}. This is likewise indistinguishable for $\mathcal{A}$, i.e., $\left|\mathrm{Adv}^{\mathsf{QP}}_{\mathsf{PI}_{\mathsf{IBE}},\mathcal{A}}(n) - \mathrm{Adv}^{\mathsf{QP}^*}_{\mathsf{PI}_{\mathsf{IBE}},\mathcal{A}}(n)\right|$ is negligible.

In $\mathrm{Game}^{\mathsf{QP}^*}_{\mathsf{PI}_{\mathsf{IBE}},\mathcal{A}}(n)$, $\mathcal{A}$ now receives a challenge subscription token s chosen at random. As $\mathcal{A}$ is not allowed to corrupt MNs or queriers registered for qid_0 or qid_1 or query ReportData or SubscribeQuery on qid_0 or qid_1,

[7] We prove the NP-CCA/ANO-IND-ID-CCA case here, the NP-CPA/ANO-IND-ID-CPA case works identical by removing the DecodeData oracle queries.

he receives no further evaluation of g under qid_0 or qid_1. Thus, for $\mathcal{A}$, the probabilities $\Pr[g(qid_0) = s]$ and $\Pr[g(qid_1) = s]$ are equal for any value s. Hence $\mathcal{A}$ can guess b no better than with probability $\frac{1}{2}$, so $\mathrm{Adv}^{\mathsf{QP}^*}_{\mathsf{PI}_{\mathsf{IBE}},\mathcal{A}}(n) = 0$ and $\mathrm{Adv}^{\mathsf{QP}}_{\mathsf{PI}_{\mathsf{IBE}},\mathcal{A}}(n)$ is negligible. □

Proof of Report Unlinkability. As all regMN_{qid} values for the same qid are equal in $\mathsf{PI}_{\mathsf{IBE}}$ (namely $\mathsf{regMN}_{qid} = f_k(qid)$), $\mathcal{A}$ in $\mathrm{Game}^{\mathsf{RU}}_{\mathsf{PI}_{\mathsf{IBE}},\mathcal{A}}(n)$ receives in Phase II two identical values $\mathsf{regMN}^0_{qid} = \mathsf{regMN}^1_{qid} = f_k(qid)$. Thus, regMN^0_{qid}, regMN^1_{qid}, and c received by $\mathcal{A}$ are all independent of the bit b which $\mathcal{A}$ hence can guess no better than with probability $\frac{1}{2}$, i.e., $\mathrm{Adv}^{\mathsf{RU}}_{\mathsf{PI}_{\mathsf{IBE}},\mathcal{A}}(n) = 0$. □

6 Concrete PEPSICo Instantiations

We now show how our generic $\mathsf{PI}_{\mathsf{IBE}}$ construction (without and with data aggregation) can be instantiated in practice.

6.1 PEPSICo Schemes in the Random Oracle and Standard Model

The generic $\mathsf{PI}_{\mathsf{IBE}}$ construction can directly be instantiated with the IBE scheme proposed by Boneh and Franklin [3], which provides ANO-IND-ID-CPA security (under the Bilinear Diffie-Hellman (BDH) assumption [3] in the random oracle model). The resulting PEPSICo scheme, denoted $\mathsf{PI}_{\mathsf{BF}}$, thus by Theorem 1 provides node privacy under chosen-plaintext attacks, query privacy, and report unlinkability. As our comparison in Section 7 shows, $\mathsf{PI}_{\mathsf{BF}}$ offers the same high practical performance as the original PEPSI scheme.

Since our result in Theorem 1 holds in the standard model, we can easily obtain further PEPSICo schemes whose security does not require random oracles. For instance, the anonymous IBE schemes by Boyen and Waters [6] or Gentry [26] can likewise be used as appropriate building blocks to instantiate $\mathsf{PI}_{\mathsf{IBE}}$.

6.2 PEPSICo Schemes with Data Aggregation

We continue our presentation of practical PEPSICo instantiations with those allowing for data aggregation.

Additively Homomorphic IBE Scheme. For our PEPSICo scheme with data aggregation we first introduce and analyze an *additively homomorphic* IBE scheme AIBE that we developed as a modification of the Boneh-Franklin IBE scheme [3].

Definition 8 (AIBE Scheme). *The additively homomorphic IBE scheme* AIBE *is defined as follows.*

$\mathsf{Setup}(1^n)$. *Generate the bilinear group parameters* $(\mathbb{G} = \langle g\rangle, q, e\colon \mathbb{G}\times\mathbb{G} \to \mathbb{G}_T)$ *with* $\mathbb{G}_T = \langle \bar{g}\rangle$ *for* $\bar{g} = e(g, g)$. *Choose* $x \in_R \mathbb{Z}_q^*$, *set* $y := g^x$, *and fix a cryptographic hash function* $H\colon \{0,1\}^* \to \mathbb{G}^*$. *The message space is* $\mathcal{M} = \mathbb{Z}_M = \{0, \dots, M-1\} \subseteq \mathbb{Z}_q$ *with* $M = p(n) < q$ *for some polynomial* p, *the ciphertext space is* $\mathcal{C} = \mathbb{G}^* \times \mathbb{G}_T$. *Output* $\mathsf{mpk} = (q, \mathbb{G} = \langle g\rangle, \mathbb{G}_T = \langle\bar{g}\rangle, e, y, H)$ *and* $\mathsf{msk} = x$.

$\mathsf{Extract}(\mathsf{mpk}, \mathsf{msk}, id)$. *Compute and output* $sk_{id} := H(id)^x$.

$\mathsf{Enc}(\mathsf{mpk}, id, m)$. *Choose* $r \in_R \mathbb{Z}_q^*$ *and output* $c = (g^r, \bar{g}^m \cdot e(H(id), y)^r)$.

$\mathsf{Dec}(\mathsf{mpk}, sk_{id}, c)$. *Parse* c *as* (c_1, c_2). *Compute* $\overline{m} := c_2/e(sk_{id}, c_1)$ *and* $m = \log_{\bar{g}}(\overline{m})$ *as the discrete logarithm to the base* $\bar{g}$ *of* $\overline{m}$ *in* $\mathbb{G}_T$ *(which takes polynomial time in* n *as* $m < M$ *, cf. the performance discussion below).*

Correctness of AIBE follows from the fact that

$$\log_{\bar{g}}(\overline{m}) = \log_{\bar{g}}(c_2/e(sk_{id}, c_1)) = \log_{\bar{g}}(\bar{g}^m \cdot e(H(id), y)^r / e(H(id)^x, g^r)) = m.$$

Our AIBE scheme is *additively* homomorphic in the message space $\mathcal{M} = \mathbb{Z}_M$ by element-wise multiplication of ciphertexts: $c \cdot c' = (g^r \cdot g^{r'}, \bar{g}^m \cdot e(H(id), y)^r \cdot \bar{g}^{m'} \cdot e(H(id), y)^{r'}) = (g^{r+r'}, \bar{g}^{m+m'} \cdot e(H(id), y)^{r+r'}) = \mathsf{Enc}(\mathsf{mpk}, id, m + m' \mod q)$. The beneficial additive homomorphism of our AIBE scheme comes at the cost of two practical disadvantages: the limited (i.e., only polynomial-sized) messages space and the need to compute a discrete logarithm for decryption. We will see in Section 6.2 that—though theoretically notable—both constraints are acceptable in many practical scenarios.

Security Analysis. We now recall the well-known DBDH assumption that is used in Theorem 2 to prove the security of the AIBE scheme.

Definition 9 (DBDH Assumption). *The* Decisional Bilinear Diffie-Hellman (DBDH) assumption *with respect to a bilinear group generation algorithm $\mathcal{G}(1^n)$, $n \in \mathbb{N}$, states that for all PPT algorithms $\mathcal{A}$ the advantage function*

$$\mathrm{Adv}^{\mathsf{DBDH}}_{\mathcal{G},\mathcal{A}}(n) := \left|\Pr\left[\mathcal{A}(g, q, e, g^{x_1}, g^{x_2}, g^{x_3}, h_b) = b\right] - \tfrac{1}{2}\right|$$

where $(\mathbb{G} = \langle g \rangle, q, e\colon \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T) \leftarrow \mathcal{G}(1^n)$ and $w, x_1, x_2, x_3 \in_R \mathbb{Z}_q$, $h_0 = e(g, g)^{x_1 x_2 x_3}$, $h_1 = e(g, g)^w$, $b \in_R \{0, 1\}$ is negligible in n.

Theorem 2 (ANO-IND-ID-CPA Security of AIBE). AIBE *provides anonymity and indistinguishability under chosen-plaintext attacks under the DBDH assumption, in the random oracle model.*

We prove the ANO-IND-ID-CPA security of AIBE by proving its indistinguishability in Theorem 3 and its anonymity in Theorem 4, combining both results using Lemma 1.

Theorem 3 (IND-ID-CPA Security of AIBE). *If the DBDH assumption holds for $\mathcal{G}$ and H is a random oracle, then* AIBE *provides* IND-ID-CPA *security.*

Theorem 3 is proven similar to the Boneh-Franklin scheme (cf. [4, Theorem 4.1]). We first introduce the following public-key version APub of our AIBE scheme.

Definition 10 (APub). *Let $\mathcal{G}$ be a (symmetric) bilinear group generator. The additively homomorphic public-key encryption scheme* APub *is defined as follows.*

$\mathsf{KeyGen}(1^n)$. *Generate $(\mathbb{G} = \langle g \rangle, q, e\colon \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T) \leftarrow \mathcal{G}(1^n)$ with $\mathbb{G}_T = \langle \bar{g} \rangle$ for $\bar{g} = e(g, g)$. Choose $x \in_R \mathbb{Z}_q^*$, $h \in_R G^*$ and set $y := g^x$. Output $pk = (q, \mathbb{G} = \langle g \rangle, \mathbb{G}_T = \langle \bar{g} \rangle, e, y, h)$ and $sk = h^x$.*
$\mathsf{Enc}(pk, m)$. *Choose $r \in_R \mathbb{Z}_q^*$ and output $c = (g^r, \bar{g}^m \cdot e(h, y)^r)$.*
$\mathsf{Dec}(sk, c)$. *Parse c as (c_1, c_2). Compute $\overline{m} := c_2 / e(sk, c_1)$ and $m = \log_{\bar{g}}(\overline{m})$.*

Correctness of APub follows similarly as for AIBE by

$$\log_{\bar{g}}(\overline{m}) = \log_{\bar{g}}(c_2/e(sk, c_1)) = \log_{\bar{g}}(\bar{g}^m \cdot e(h, y)^r / e(h^x, g^r)) = \log_{\bar{g}}(\bar{g}^m \cdot e(h, g)^{rx} / e(h, g)^{rx}) = m.$$

First, we show that an attacker against the IND-ID-CPA security of AIBE can be used to break the IND-CPA security of APub. We then prove that the APub scheme is secure under the DBDH assumption.

Lemma 2. *Let H be a random oracle and let $\mathcal{A}$ be an adversary against the* IND-ID-CPA *security of* AIBE *with advantage $\mathrm{Adv}^{\mathsf{IND\text{-}ID\text{-}CPA}}_{\mathsf{AIBE},\mathcal{A}}(n)$ which issues at most q_E key extraction queries. Then there is an adversary $\mathcal{B}$ against the* IND-CPA *security of* APub *with advantage $\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{APub},\mathcal{B}}(n) \geq \frac{1}{e(q_E+1)} \cdot \mathrm{Adv}^{\mathsf{IND\text{-}ID\text{-}CPA}}_{\mathsf{AIBE},\mathcal{A}}(n)$, where $e \approx 2.72$ is the base of the natural logarithm.*

Proof. We construct adversary $\mathcal{B}$, which interacts with $\mathcal{A}$ in the IND-ID-CPA game and controls the random oracle H, as follows.[8]

$\mathcal{B}$ receives the public key $pk = (q, \mathbb{G} = \langle g \rangle, \mathbb{G}_T = \langle \bar{g} \rangle, e, y, h)$ in the IND-CPA game and provides $\mathcal{A}$ with the master public key $\mathsf{mpk} = (q, \mathbb{G} = \langle g \rangle, \mathbb{G}_T = \langle \bar{g} \rangle, e, y, H)$, where H is the random oracle controlled by $\mathcal{B}$, which handles queries to H and Extract queries by $\mathcal{A}$ as follows.

[8] Note that this proof works nearly identical to the corresponding proof of Lemma 4.2 in [4].

H-queries. $\mathcal{A}$ can query H at any time. In order to answer those queries consistently, $\mathcal{B}$ keeps a (initially empty) list H^{list} of tuples $\langle id_i, h_i, x_i, c_i\rangle$ and responds to queries of H with identity id_i as follows:

- If id_i appears in H^{list} in a tuple $\langle id_i, h_i, x_i, c_i\rangle$, $\mathcal{B}$ responds with $H(id_i) = h_i$.
- Otherwise, $\mathcal{B}$ chooses $c_i \in_R \{0,1\}$ with $\Pr[c_i = 0] = \delta$ (for some δ to be determined later) and $x_i \in_R \mathbb{Z}_q^*$ at random. If $c_i = 0$, it sets $h_i := g^{x_i}$, else it sets $h_i := h^{x_i}$. Finally, $\mathcal{B}$ adds the tuple $\langle id_i, h_i, x_i, c_i\rangle$ to H^{list} and outputs $H(id_i) = h_i$.

Note that, as $x_i \in_R \mathbb{Z}_q^*$ is chosen at random, the output h_i is uniformly distributed in $\mathbb{G}^*$.

$\mathsf{Extract}$-queries. $\mathcal{B}$ responds on queries of $\mathsf{Extract}$ with identity id_i by first computing $h_i \leftarrow H(id_i)$ as described above. Let $\langle id_i, h_i, x_i, c_i\rangle$ be the corresponding tuple in H^{list}. If $c_i = 1$, then $\mathcal{B}$ fails and aborts the attack. Otherwise, it outputs $sk_i := y^{x_i}$. Note that, as $c_i = 0$, $H(id_i) = h_i = g^{x_i}$ and thus $sk_i = y^{x_i} = h_i^x = H(id_i)^x$ as desired.

At some point in time, $\mathcal{A}$ outputs an identity id^* and two messages m_0 and m_1. $\mathcal{B}$ forwards the messages to its own challenger and receives the encryption $c = (c_1, c_2)$ of m_b for a random $b \in \{0,1\}$. Now $\mathcal{B}$ computes $h_i \leftarrow H(id_i)$ as described above. Let $\langle id_i, h_i, x_i, c_i\rangle$ be the corresponding tuple in H^{list}. If $c_i = 0$, $\mathcal{B}$ fails and aborts the attack. Otherwise, $\mathcal{B}$ responds to $\mathcal{A}$ with $c' = (c_1^{x_i^{-1}}, c_2)$, where x_i^{-1} is the inverse of x_i in $\mathbb{Z}_q^*$. Note that c' is a valid AIBE-encryption of m_b under identity id^* since $h_i = h^{x_i}$ and thus (for $r' := rx_i^{-1}$)

$$\begin{aligned} c' = (c_1^{x_i^{-1}}, c_2) &= (g^{rx_i^{-1}}, \bar{g}^{m_b} \cdot e(h,y)^r) = (g^{rx_i^{-1}}, \bar{g}^{m_b} \cdot e(h,y)^{rx_ix_i^{-1}}) \\ &= (g^{rx_i^{-1}}, \bar{g}^{m_b} \cdot e(h^{x_i},y)^{rx_i^{-1}}) = (g^{r'}, \bar{g}^{m_b} \cdot e(h_i,y)^{r'}). \end{aligned}$$

$\mathcal{A}$ continues and might issue H- or $\mathsf{Extract}$-queries, which $\mathcal{B}$ handles as described above. Finally, $\mathcal{A}$ outputs a guess b' which $\mathcal{B}$ forwards to its own challenger.

If $\mathcal{B}$ does not abort, it perfectly simulates the $\mathsf{IND\text{-}ID\text{-}CPA}$ game for $\mathcal{A}$, as the responses to H-queries are uniformly and independently distributed in $\mathbb{G}^*$, the $\mathsf{Extract}$-queries are answered correctly, and c' is a proper AIBE encryption of m_b. Thus in this case, $\mathrm{Adv}^{\mathsf{ANO\text{-}ID\text{-}CPA}}_{\mathsf{APub},\mathcal{B}}(n) \geq \mathrm{Adv}^{\mathsf{ANO\text{-}IND\text{-}ID\text{-}CPA}}_{\mathsf{AIBE},\mathcal{A}}(n)$. It remains to analyze the probability that $\mathcal{B}$ does not abort, which is δ^{q_E} for the q_E for the phases where $\mathcal{A}$ may issue $\mathsf{Extract}$-queries and $1-\delta$ for the challenge phase, i.e., the overall probability that $\mathcal{B}$ does not abort is $\delta^{q_E}(1-\delta)$. This value is maximized at $\delta_{opt} = 1 - \frac{1}{q_E+1}$, thus for δ_{opt}, $\mathcal{B}$ does not abort with probability at least $\frac{1}{e(q_E+1)}$. This results in the overall advantage $\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{APub},\mathcal{B}}(n) \geq \frac{1}{e(q_E+1)} \cdot \mathrm{Adv}^{\mathsf{IND\text{-}ID\text{-}CPA}}_{\mathsf{AIBE},\mathcal{A}}(n)$ for $\mathcal{B}$. □

Lemma 3. *Let $\mathcal{A}$ be an adversary against the $\mathsf{IND\text{-}CPA}$ security of APub with advantage $\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{APub},\mathcal{A}}(n)$. Then there is an algorithm $\mathcal{B}$ that breaks the DBDH assumption with $\mathrm{Adv}^{\mathsf{DBDH}}_{\mathcal{G},\mathcal{B}}(n) = \frac{1}{2} \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{APub},\mathcal{A}}(n)$.*

Proof. We construct adversary $\mathcal{B}$, which interacts with $\mathcal{A}$ in the $\mathsf{IND\text{-}CPA}$ game as follows. $\mathcal{B}$ receives elements $(g, q, e, g^{x_1}, g^{x_2}, g^{x_3}, h_b)$ (where $h_0 = e(g,g)^{x_1x_2x_3}$ and $h_1 = e(g,g)^w$ for $w \in_R \mathbb{Z}_q$). It provides $\mathcal{A}$ with $pk = (q, \mathbb{G} = \langle g\rangle, \mathbb{G}_T = \langle \bar{g}\rangle, e, y, h)$, where $\bar{g} := e(g,g)$, $y := g^{x_1}$ and $h := g^{x_2}$. $\mathcal{A}$ outputs two messages m_0 and m_1. $\mathcal{B}$ chooses $b' \in_R \{0,1\}$ and $r \in_R \mathbb{Z}_q^*$ at random, computes $c = (c_1, c_2) = (g^r, \bar{g}^{m_{b'}} \cdot h_b)$, and provides $\mathcal{A}$ with c. Finally, $\mathcal{A}$ outputs a guess b''. If $b' = b''$, $\mathcal{B}$ outputs 0, otherwise 1. Observe that if $b = 0$, c is a valid encryption of $m_{b'}$, whereas otherwise, h_b is completely random in $\mathbb{G}_T$, i.e., c_2 perfectly hides $m_{b'}$. This leads to

$$\begin{aligned} \Pr[\mathcal{B} \text{ outputs } b] &= \Pr[\mathcal{A} \text{ outputs } b' \mid b = 0] \cdot \Pr[b = 0] + \Pr[\mathcal{A} \text{ outputs } 1 - b' \mid b = 1] \cdot \Pr[b = 1] \\ &= \left(\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{APub},\mathcal{A}}(n) + \tfrac{1}{2}\right) \cdot \tfrac{1}{2} + \tfrac{1}{2} \cdot \tfrac{1}{2} = \tfrac{1}{2} \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{APub},\mathcal{A}}(n) + \tfrac{1}{2} \end{aligned}$$

and thus $\mathrm{Adv}^{\mathsf{DBDH}}_{\mathcal{G},\mathcal{B}}(n) = \frac{1}{2} \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{APub},\mathcal{A}}(n)$. □

Combining Lemma 2 and 3, if H is a random oracle, an adversary $\mathcal{A}$ with $\mathrm{Adv}^{\mathsf{IND\text{-}ID\text{-}CPA}}_{\mathsf{AIBE},\mathcal{A}}(n)$ can be used by an algorithm $\mathcal{B}$ to break the DBDH assumption with $\mathrm{Adv}^{\mathsf{DBDH}}_{\mathcal{G},\mathcal{B}}(n) \geq \frac{1}{2e(q_E+1)} \cdot \mathrm{Adv}^{\mathsf{IND\text{-}ID\text{-}CPA}}_{\mathsf{AIBE},\mathcal{A}}(n)$. This proves Theorem 3. □

Theorem 4 (ANO-ID-CPA **Security of** AIBE)**.** *If the DBDH assumption holds for $\mathcal{G}$ and H is a random oracle, then* AIBE *provides* ANO-ID-CPA *security.*

Proof. From Theorem 3 and the assumptions we know that AIBE is IND-ID-CPA-secure. Assume now we have an adversary $\mathcal{A}$ against the ANO-ID-CPA security of AIBE with advantage $\mathrm{Adv}^{\mathsf{ANO\text{-}ID\text{-}CPA}}_{\mathsf{AIBE},\mathcal{A}}(n)$. We construct an adversary $\mathcal{B}$ against the IND-ID-CPA security of AIBE as follows: $\mathcal{B}$ forwards the received mpk to $\mathcal{A}$ and relays $\mathsf{Extract}$-queries to its own oracle. When $\mathcal{A}$ outputs (id_0, id_1, m), $\mathcal{B}$ chooses $b' \in_R \{0,1\}$ and $R \in_R \mathbb{Z}_q$ at random, outputs $(id_{b'}, m, R)$ as its own challenge request, and receives a ciphertext $c = (c_1, c_2)$ (c is an encryption of m if $b = 0$, of R otherwise) which it outputs as its response to $\mathcal{A}$. Finally, $\mathcal{A}$ outputs a guess b''. If $b' = b''$, $\mathcal{B}$ outputs 0, otherwise 1. Observe that if $b = 0$, c is a valid encryption of m under $id_{b'}$ and thus $\mathcal{B}$ perfectly simulates the ANO-ID-CPA game for $\mathcal{A}$. If however $b = 1$, then g^R and thus also c_2 is uniformly distributed in $\mathbb{G}_T$ and hence independent of $id_{b'}$, resulting in $\mathcal{A}$ being not able to guess b' better than with probability $\frac{1}{2}$. Therefore

$$\begin{aligned}\Pr[\mathcal{B} \text{ outputs } b] &= \Pr[\mathcal{A} \text{ outputs } b' \mid b = 0] \cdot \Pr[b = 0] + \Pr[\mathcal{A} \text{ outputs } 1 - b' \mid b = 1] \cdot \Pr[b = 1] \\ &= \left(\mathrm{Adv}^{\mathsf{ANO\text{-}ID\text{-}CPA}}_{\mathsf{AIBE},\mathcal{A}}(n) + \tfrac{1}{2}\right) \cdot \tfrac{1}{2} + \tfrac{1}{2} \cdot \tfrac{1}{2} = \tfrac{1}{2} \cdot \mathrm{Adv}^{\mathsf{ANO\text{-}ID\text{-}CPA}}_{\mathsf{AIBE},\mathcal{A}}(n) + \tfrac{1}{2}\end{aligned}$$

and thus $\mathrm{Adv}^{\mathsf{IND\text{-}ID\text{-}CPA}}_{\mathsf{AIBE},\mathcal{B}}(n) = \frac{1}{2} \cdot \mathrm{Adv}^{\mathsf{ANO\text{-}ID\text{-}CPA}}_{\mathsf{APub},\mathcal{A}}(n)$. □

Combining Theorems 3 and 4, Lemma 1 implies Theorem 2, i.e., AIBE is ANO-IND-ID-CPA-secure. □

Performance Discussion and Analysis of AIBE. The additive homomorphism of AIBE has a limitation in that the computation of discrete logarithms in $\mathbb{G}_T$ is required to perform the decryption operation. Therefore, AIBE is suited only for messages from a short interval. By means of brute force it would take on average $M/2$ multiplications in $\mathbb{G}_T$ to check whether $\bar{m} = \bar{g}^i$ for each i in $[0, M-1]$. Using Pollard's kangaroo method [38] to compute discrete logarithms in the interval $[0, M-1]$ requires expected time $O(\sqrt{M})$. As a third option, constant decryption time can be achieved with a polynomial-size lookup table with stored powers of $\bar{g}$. The time required to compute such a table equals the time of a complete brute-force run in $[0, M-1]$.

We implemented both the brute-force and Pollard's kangaroo method using the Pairing-Based Cryptography (PBC) library [32] (in version 0.5.12) with a symmetric type-a pairing, which is defined over the elliptic curve $y^2 = x^3 + x$ with 160-bit group order and embedding degree 2. The measurements of computing the discrete logarithm $x \in_R [0, M]$ for a given $\bar{g}^x \in \mathbb{G}_T$ performed on a 2.10GHz Intel(R) Core(TM)2 Duo T8100 CPU with 2GB RAM running Kubuntu 10.04 resulted in, on average, $0.004 \cdot \frac{M}{2}$ ms for the brute-force and $0.185 \cdot \sqrt{M}$ ms (average coefficient) for Pollard's kangaroo method. The latter outperforms the brute-force approach for message spaces with over $1{,}000$ elements and remains feasible even for complete 32-bit integer values (9.084 sec).[9] Our AIBE scheme thus remains practical when aggregating small values in $\mathbb{Z}_M$, i.e., integers of up to 32-bit length.

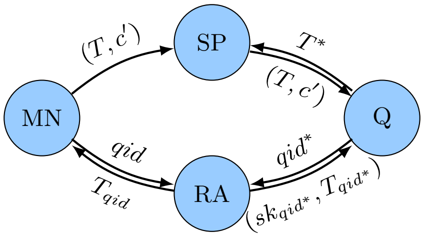
Note that the restriction of AIBE to a polynomial message space is typical for additively homomorphic encryption schemes based on the Decisional (Bilinear) Diffie-Hellman assumption where messages are encrypted in the exponents. Examples are the exponential ElGamal scheme (where, in contrast to the original version [22], messages are encrypted in the exponent as $\mathsf{Enc}(m) = (g^r, g^m \cdot h^r))$ used, e.g., in electronic voting schemes [14], the homomorphic scheme by Boneh, Goh, and Nissim [5], or the encryption scheme incorporating secret sharing proposed by Shi et al. [41].

PEPSICo Scheme with Data Aggregation. We now instantiate the generic $\mathsf{PI}_{\mathsf{IBE}}$ construction with the AIBE scheme and denote the resulting PEPSICo scheme with data aggregation as $\mathsf{PI}_{\mathsf{AIBE}}$, depicted in Figure 4. Combining Theorems 1 and 2, the resulting scheme provides node privacy under chosen-plaintext attacks,

[9] Note that decryption in our scenario will *not* be performed by mobile devices but by queriers with computing power comparable to our test machine.

`ExecuteQuery`: If $T = T^*$ output (T, c'), else output $\perp$.
`AggregateData`: If $T_1 = \cdots = T_\ell$ output $(T, c') = \Big(T_1, \Big(\prod_{i=1}^{\ell} c_{i,1}, \prod_{i=1}^{\ell} c_{i,2}\Big)\Big)$, else output $\perp$.

`ReportData`:
$r \in_R \mathbb{Z}_q^*$, $c := (T_{qid}, c')$,
$c' = (c_1, c_2) = (g^r, \bar{g}^m \cdot$
$e(H(qid), y)^r)$.



`SubscribeQuery`:
$s := T^* := T_{qid^*}$.

`DecodeData`: $m :=$
$\log_{\bar{g}}(c_2 / e(sk_{qid^*}, c_1))$.

`Setup`: $\mathsf{RAsk} := (x \in_R \mathbb{Z}_q^*, k \in_R \{0,1\}^n)$, $\mathsf{RApk} := (g, y := g^x, H)$.
`RegisterMN`: $\mathsf{regMN}_{qid} := T_{qid} := f_k(qid)$.
`RegisterQ`: $\mathsf{regQ}_{qid^*} := (sk_{qid^*} := H(qid^*)^x, T_{qid^*})$.

Fig. 4. PEPSICo instantiation with data aggregation $\mathsf{PI}_{\mathsf{AIBE}}$ based on the AIBE scheme and a pseudorandom function f.

query privacy, and report unlinkability (under the DBDH assumption in the random oracle model). We evaluate its practical performance in Section 7.

Besides the additional privacy achieved by report aggregation (cf. Section 5.2), our $\mathsf{PI}_{\mathsf{AIBE}}$ scheme allows for a second powerful privacy mechanism. Consider a participatory sensor network with *tree-based routing* where mobile nodes route their messages along a path of other nodes to the service provider. In such a scenario, the `AggregateData` operation of $\mathsf{PI}_{\mathsf{AIBE}}$ can already be executed by MNs on the path (remember that no secret key is needed), thus further increasing the privacy of MNs and data reports vis-à-vis the SP. This approach moreover is computationally cheap for MNs as the aggregation of two data reports in $\mathsf{PI}_{\mathsf{AIBE}}$ requires only two group multiplications. Though not representable in our PEPSICo model, the security proven for $\mathsf{PI}_{\mathsf{AIBE}}$ presumably carries over also to such tree-based routing setting with aggregation performed along the path to the SP.

Similar to Section 6.1, we obtain a secure PEPSICo instantiation with data aggregation in the standard model using an appropriate, ANO-IND-ID-CPA-secure *additively homomorphic* IBE scheme. While no scheme has been proposed as such, we can build one based on Gentry's scheme [26], leveraging its multiplicative homomorphism in $\mathbb{G}_T$ by usage of $g^m \in \mathbb{G}_T$ instead of $m \in \mathbb{G}_T$ (i.e., similar to AIBE). The resulting scheme is less efficient though than our AIBE scheme and can be proven secure only under the less common Decisional Augmented Bilinear Diffie-Hellman Exponent assumption (cf. [26]).

General Statistics. It is an interesting observation (whose details lie outside the scope of this paper) that due to the additively homomorphic property of the introduced IBE scheme and the fact that legitimate queriers will have the secret key to decrypt, any statistics can be computed in a secure two-party computation manner between the service provider and the querier. This can be done by using the standard "blinding" technique from Cramer et al. [13] to realize secure two-party computation of any function. With this approach, the querier only learns the requested statistics without learning anything at all.

6.3 Variants with Anonymous Registration

Depending on the participatory sensing scenario, it might be desirable to hide the interests of mobile nodes and queriers not only from the service provider, but also from the registration authority. Without going into great detail, we conclude the presentation of our PEPSICo instantiations with a short discussion on how to conceal the interests of mobile nodes and queriers at registration time.

For the *mobile node registration*, our generic $\mathsf{PI}_{\mathsf{IBE}}$ instantiation could make use of an *oblivious pseudorandom function* (OPRF) [35,24], that allows for a PRF evaluation $f_k(x)$ in a two-party protocol between a client (providing the input x) and a server (providing the key k), such that the client learns the PRF value $f_k(x)$, but the server learns nothing. In our setting, the mobile node could register at the RA by (interactive)

Table 1. Comparison of computation and communication overhead of PEPSI [15] and our $\mathsf{PI}_{\mathsf{BF}}$ and $\mathsf{PI}_{\mathsf{AIBE}}$ schemes (cf. Sections 6.1 and 6.2).

	PEPSI		$\mathsf{PI}_{\mathsf{BF}}$		$\mathsf{PI}_{\mathsf{AIBE}}$	
Algorithm	Comp.	Comm.	Comp.	Comm.	Comp.	Comm.
`Setup`	2E	–	1E	–	1E	–
`RegisterMN`	–	n	1f	n	1f	n
`RegisterQ`	1E	2G	1f+1E	1G+n	1f+1E	1G+n
`ReportData`	1E+1P+2H	2n	2E+1P+2H	1G+2n	3E+1P+1H	2G+n
`SubscribeQuery`	1P+1H	n	–	n	–	n
`ExecuteQuery`	–	2n	–	1G+2n	–	2G+n
`DecodeData`	1P+1H	–	1P+1H	–	1P+1DL	–
`AggregateData`	n/a	n/a	n/a	n/a	$\approx 0^*$	–

E — modular exponentiation in $\mathbb{G}$ or $\mathbb{G}_T$; P — pairing evaluation; H — hash function evaluation; f — PRF evaluation; DL — computation of discrete logarithm; G — group element in $\mathbb{G}$ or $\mathbb{G}_T$; n — message length, Hash/PRF output length

* The `AggregateData` algorithm of $\mathsf{PI}_{\mathsf{AIBE}}$ requires 2ℓ group multiplications to aggregate ℓ ciphertexts, negligible compared to the other units used.

evaluation of an OPRF on input *qid* (the query identity) and key k (provided by the RA). As in $\mathsf{PI}_{\mathsf{IBE}}$, the mobile node would obtain $T_{qid} = f_k(qid)$ as registration token, whereas the RA would not learn the query identity the mobile node is interested in.

In the *querier registration*, the tag T_{qid^*} for query identity qid^* could be computed in the same way as for the mobile node registration using an OPRF. Moreover, the querier could obtain its secret key sk_{qid^*} from the RA in an oblivious way by using a *blind* identity-based encryption scheme [27]. Combining both OPRF-based tags and blind identity-based encryption[10], our generic $\mathsf{PI}_{\mathsf{IBE}}$ instantiation thus allows for anonymous registration of mobile nodes and queriers wrt. the registration authority.

7 Performance Evaluation and Comparisons

We now evaluate the performance of our two concrete $\mathsf{PEPSICo}$ schemes: $\mathsf{PI}_{\mathsf{BF}}$ from Section 6.1 and $\mathsf{PI}_{\mathsf{AIBE}}$ from Section 6.2. In particular, we compare the induced computation, communication, and storage overhead of the two schemes with the original PEPSI scheme [15,17], though keeping in mind that it does not fulfill the requirements of node and query privacy in our model due to collusion attacks.

Table 1 shows the computation and communication overhead introduced by PEPSI, $\mathsf{PI}_{\mathsf{BF}}$, and $\mathsf{PI}_{\mathsf{AIBE}}$. PEPSI and $\mathsf{PI}_{\mathsf{BF}}$ perform similar in computation, except that $\mathsf{PI}_{\mathsf{BF}}$ uses a pseudorandom function for tag generation. Computation overhead of $\mathsf{PI}_{\mathsf{AIBE}}$ (the only scheme providing data aggregation) is significantly higher only for the `DecodeData` operation, which requires computation of a discrete logarithm. Note that `DecodeData` is *not* executed by the (resource-constrained) mobile nodes, but by queriers with a presumable computing power comparable to the machine running our test measurements. In return, $\mathsf{PI}_{\mathsf{AIBE}}$ saves decryption time if reports are aggregated, requiring only $2(\ell - 1)$ cheap group multiplications to aggregate ℓ reports. Indeed, based on our measurements for discrete logarithm and pairing computation[11], $\mathsf{PI}_{\mathsf{AIBE}}$ even *outperforms* $\mathsf{PI}_{\mathsf{BF}}$ wrt. the decryption overhead if single data messages are integers between 0 and about 1000—independently of how many messages are aggregated in an arbitrary large message space.

Concerning communication costs, the only practical difference is in the length of ciphertexts. While ciphertexts in PEPSI have the same length as messages, in $\mathsf{PI}_{\mathsf{BF}}$ and $\mathsf{PI}_{\mathsf{AIBE}}$ they additionally contain one group element of $\mathbb{G}$ ($\mathsf{PI}_{\mathsf{AIBE}}$ uses another group element of $\mathbb{G}$ instead of an n-bit string in the second component—a difference negligible in practice). Aggregation in $\mathsf{PI}_{\mathsf{AIBE}}$ however allows for huge savings (a factor ℓ for ℓ aggregated reports) in the communication between SP and queriers. More important, $\mathsf{PI}_{\mathsf{BF}}$ and $\mathsf{PI}_{\mathsf{AIBE}}$ do not

[10] Anonymous registration based on an OPRF in a symmetric setting resp. a blind IBE scheme in an asymmetric setting has been separately discussed also as an extended capability of the PEPSI scheme [16].

[11] Discrete logarithm in interval $[0, M]$: $0.18\sqrt{M}$ ms. Pairing: 5.99 ms (cf. Section 6.2).

Table 2. Comparison of space requirements of PEPSI [15] and our $\mathsf{PI}_{\mathsf{BF}}$ and $\mathsf{PI}_{\mathsf{AIBE}}$ schemes (cf. Sections 6.1 and 6.2).

Component	PEPSI	$\mathsf{PI}_{\mathsf{BF}}$	$\mathsf{PI}_{\mathsf{AIBE}}$
RA Public Key RApk	3G+n	3G+n	3G+n
RA Secret Key RAsk	1G+2n	2n	2n
MN Registration Value regMN_{qid}	n	n	n
Querier Registration Value regQ_{qid}	2G	1G+n	1G+n
Data Report c	2n	1G+2n	2G+n
Subscription Token s	n	n	n

G — group element in $\mathbb{G}$ or $\mathbb{G}_T$; n — message length, Hash/PRF output length

require any periodic update operations as opposed to the regular "nonce renewal" of PEPSI, saving further computation and communication resources.

Table 2 shows the (virtually identical) space requirements of all three schemes. The use of a pseudorandom function to generate tags in $\mathsf{PI}_{\mathsf{BF}}$ and $\mathsf{PI}_{\mathsf{AIBE}}$ saves one group element in RAsk, whereas data reports c in $\mathsf{PI}_{\mathsf{BF}}$ and $\mathsf{PI}_{\mathsf{AIBE}}$ contain one additional group element ($\mathsf{PI}_{\mathsf{AIBE}}$ also uses another group element that replaces the n-bit string in the second component of the ciphertext). Additionally, the aggregation of reports possible in $\mathsf{PI}_{\mathsf{AIBE}}$ further saves storage capacity of the SP and queriers.

In summary, $\mathsf{PI}_{\mathsf{BF}}$ performs similar to PEPSI wrt. computation overhead and key sizes and has only slightly higher communication overhead, while providing stronger node privacy, query privacy, and report unlinkability guarantees in the presence of colluding parties. For small messages, $\mathsf{PI}_{\mathsf{AIBE}}$ is almost as fast as the $\mathsf{PI}_{\mathsf{BF}}$ scheme while achieving the same level of security and enabling support for aggregation. The latter property allows for a significant reduction of the communication overhead between service provider and queriers and can offer more stringent privacy guarantees with respect to individual data reports.

8 Conclusion and Outlook

Participatory sensing allows for novel paradigms of information collection, but also introduces privacy challenges for data report and data retrieval. We presented $\mathsf{PEPSICo}$, a refined version of the PEPSI model [15] that protects data confidentiality and user privacy under collusion attacks and additionally allows for data aggregation. Our generic and concrete instantiations leveraging anonymous identity-based encryption (IBE) achieve full privacy as well as equally high practical performance as earlier approaches. For future work, constructing an efficient additively homomorphic IBE scheme with exponential-sized message space remains an open problem of independent interest in the setting of data aggregation.

References

1. M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi. Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions. In *CRYPTO 2005*, pages 205–222, 2005.
2. D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano. Public Key Encryption with Keyword Search. In *EUROCRYPT 2004*, pages 506–522, 2004.
3. D. Boneh and M. K. Franklin. Identity-Based Encryption from the Weil Pairing. In *CRYPTO 2001*, pages 213–229, 2001.
4. D. Boneh and M. K. Franklin. Identity-Based Encryption from the Weil Pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003.
5. D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-DNF Formulas on Ciphertexts. In *TCC 2005*, pages 325–341, 2005.
6. X. Boyen and B. Waters. Anonymous Hierarchical Identity-Based Encryption (Without Random Oracles). In *CRYPTO 2006*, pages 290–307, 2006.

7. C. Castelluccia, A. C.-F. Chan, E. Mykletun, and G. Tsudik. Efficient and Provably Secure Aggregation of Encrypted Data in Wireless Sensor Networks. *ACM Transactions on Sensor Networks*, 5(3), 2009.
8. D. Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Communications of the ACM*, 24(2):84–88, 1981.
9. D. Chaum. Blind Signatures for Untraceable Payments. In *CRYPTO 1982*, pages 199–203, 1982.
10. D. Christin, M. Hollick, and M. Manulis. Security and Privacy Objectives for Sensing Applications in Wireless Community Networks. In *ICCCN 2010*, pages 1–6, 2010.
11. E. S. Cochran, J. F. Lawrence, C. Christensen, and R. S. Jakka. The Quake-Catcher Network: Citizen Science Expanding Seismic Horizons. *Seismological Research Letters*, 80(1):26–30, 2009.
12. C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos. AnonySense: Privacy-Aware People-Centric Sensing. In *MobiSys 2008*, pages 211–224, 2008.
13. R. Cramer, I. Damgård, and J. B. Nielsen. Multiparty Computation from Threshold Homomorphic Encryption. In *EUROCRYPT 2001*, pages 280–299, 2001.
14. R. Cramer, R. Gennaro, and B. Schoenmakers. A Secure and Optimally Efficient Multi-Authority Election Scheme. In *EUROCRYPT 1997*, pages 103–118, 1997.
15. E. De Cristofaro and C. Soriente. Short Paper: PEPSI: Privacy-Enhanced Participatory Sensing Infrastructure. In *ACM WISEC 2011*, pages 23–28, 2011.
16. E. De Cristofaro and C. Soriente. Extended Capabilities for a Privacy-Enhanced Participatory Sensing Infrastructure (PEPSI). *IEEE Transactions on Information Forensics and Security*, 8(12):2021–2033, 2013.
17. E. De Cristofaro and C. Soriente. Participatory Privacy: Enabling Privacy in Participatory Sensing. *IEEE Network*, 27(1):32–36, 2013.
18. E. D'Hondt, M. Stevens, and A. Jacobs. Participatory noise mapping works! An evaluation of participatory sensing as an alternative to standard techniques for environmental monitoring. *Pervasive and Mobile Computing*, 9(5):681–694, 2013.
19. T. Dimitriou, I. Krontiris, and A. Sabouri. PEPPeR: A Querier's Privacy Enhancing Protocol for PaRticipatory Sensing. In *MobiSec 2012*, pages 93–106, 2012.
20. Y. Dong, S. S. Kanhere, C. T. Chou, and N. Bulusu. Automatic Collection of Fuel Prices from a Network of Mobile Cameras. In *DCOSS 2008*, pages 140–156, 2008.
21. S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell. The BikeNet mobile sensing system for cyclist experience mapping. In *SenSys 2007*, pages 87–101, 2007.
22. T. Elgamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In *CRYPTO 1984*, pages 10–18, 1984.
23. European Parliament and Council. EU Directive 95/46/EC, 1995. Available at `http://www.dataprotection.ie/viewdoc.asp?docid=89`.
24. M. J. Freedman, Y. Ishai, B. Pinkas, and O. Reingold. Keyword Search and Oblivious Pseudorandom Functions. In *TCC 2005*, pages 303–324, 2005.
25. R. K. Ganti, N. Pham, Y.-E. Tsai, and T. F. Abdelzaher. PoolView: Stream Privacy for Grassroots Participatory Sensing. In *SenSys 2008*, pages 281–294, 2008.
26. C. Gentry. Practical Identity-Based Encryption Without Random Oracles. In *EUROCRYPT 2006*, pages 445–464, 2006.
27. M. Green and S. Hohenberger. Blind Identity-Based Encryption and Simulatable Oblivious Transfer. In *ASIACRYPT 2007*, pages 265–282, 2007.
28. K. L. Huang, S. S. Kanhere, and W. Hu. Preserving privacy in participatory sensing systems. *Computer Communications*, 33(11):1266–1280, 2010.
29. B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden. CarTel: A Distributed Mobile Sensor Computing System. In *SenSys 2006*, pages 125–138, 2006.
30. A. Kapadia, D. Kotz, and N. Triandopoulos. Opportunistic Sensing: Security Challenges for the New Paradigm. In *COMSNETS 2009*, pages 1–10, 2009.
31. Q. Li and G. Cao. Efficient Privacy-Preserving Stream Aggregation in Mobile Sensing with Low Aggregation Error. In *PETS 2013*, pages 60–81, 2013.
32. B. Lynn. Pairing-Based Cryptography (PBC) library. Available at `http://crypto.stanford.edu/pbc/`.
33. A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. ℓ-Diversity: Privacy Beyond k-Anonymity. *ACM Transactions on Knowledge Discovery from Data*, 1(1), 2007.
34. M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. H. Hansen, E. Howard, R. West, and P. Boda. PEIR, the personal environmental impact report, as a platform for participatory sensing systems research. In *MobiSys 2009*, pages 55–68, 2009.

35. M. Naor and O. Reingold. Number-theoretic Constructions of Efficient Pseudo-random Functions. In *FOCS 1997*, pages 458–467, 1997.
36. S. Özdemir and Y. Xiao. Secure data aggregation in wireless sensor networks: A comprehensive overview. *Computer Networks*, 53(12):2022–2037, 2009.
37. E. Paulos, R. J. Honicky, and E. Goodman. Sensing Atmosphere. Technical Report 203, Human-Computer Interaction Institute, Carnegie Mellon University, 2007.
38. J. M. Pollard. Monte Carlo methods for index computation (mod p). *AMS Mathematics of Computation*, 32(143):918–924, 1978.
39. R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu. Ear-Phone: An End-to-End Participatory Urban Noise Mapping System. In *IPSN 2010*, pages 105–116, 2010.
40. S. Reddy, A. Parker, J. Hyman, J. Burke, D. Estrin, and M. H. Hansen. Image Browsing, Processing, and Clustering for Participatory Sensing: Lessons From a DietSense Prototype. In *EmNets 2007*, pages 13–17, 2007.
41. E. Shi, T.-H. H. Chan, E. G. Rieffel, R. Chow, and D. Song. Privacy-Preserving Aggregation of Time-Series Data. In *NDSS 2011*, 2011.
42. J. Shi, R. Zhang, Y. Liu, and Y. Zhang. PriSense: Privacy-Preserving Data Aggregation in People-Centric Urban Sensing Systems. In *IEEE INFOCOM 2010*, pages 758–766, 2010.
43. K. Shilton. Four Billion Little Brothers?: Privacy, mobile phones, and ubiquitous data collection. *Communications of the ACM*, 52(11):48–53, 2009.
44. L. Sweeney. k-Anonymity: A Model for Protecting Privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.