Privacy-Preserving Participatory Sensing with Data Aggregation

Schutz der Privatsphäre und Datenaggregation in partizipativen Sensornetzwerken

Master-Thesis by Felix Günther

March 2013

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Informatik
Cryptographic Protocols Group

Privacy-Preserving Participatory Sensing with Data Aggregation
Schutz der Privatsphäre und Datenaggregation in partizipativen Sensornetzwerken

Master-Thesis by Felix Günther

Advisor: Prof. Dr. Mark Manulis

Filing Date: March 19, 2013

Erklärung zur Master-Thesis

Hiermit versichere ich gemäß der Allgemeinen Prüfungsbestimmungen der Technischen Universität Darmstadt (APB) §23 (7), die vorliegende Masterarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 19. März 2013

(Felix Günther)

"Civilization is the progress toward a society of privacy."

Ayn Rand (1905 – 1982) in *The Fountainhead* (1943)

Abstract

Participatory sensing enables new paradigms for information collection based on the ubiquitous availability of smartphones, capable of sensing data and events of common or personal interest. Being permanent companions, the use of smartphones as sensors however introduces the challenge to protect the privacy of users and their collected data while preserving the informational benefits of participatory sensing. The few approaches to date that approach these challenges fail to achieve cryptographically provable privacy, rely on strong assumptions in the underlying infrastructure, or suffer from collusion attacks.

In this work, we present the first cryptographically sound model for *privacy-preserving participatory sensing infrastructures*, usable as an architectural building block for the interaction of data reporters and receivers. We formalize the notions of *node privacy*, *query privacy*, and *report unlinkability* as the three main privacy requirements for participatory sensing, aiming at the confidentiality of reported data, the privacy of data retrieving parties, and the anonymity of data reporters—even under the attack of multiple colluding parties. Incorporating identity-based encryption, we provide a generic and provably secure instantiation of our model and present a concrete and practical construction with equally high performance as previous approaches while achieving provable privacy.

In addition, we extend our model in a generic way to allow for seamless *data aggregation* in order to reduce the overall communication overhead and further increase the achieved privacy. Based on additively homomorphic identity-based encryption, we provide a generic instantiation that allows for private data aggregation in participatory sensing scenarios. We present a novel *additively homomorphic identity-based encryption scheme*, achieving indistinguishability and anonymity of ciphertexts under the decisional bilinear Diffie-Hellman assumption and practical performance in small message spaces. Incorporating this new scheme, we obtain a participatory sensing infrastructure with provable privacy and efficient data aggregation.

Zusammenfassung

Partizipative Sensornetzwerke eröffnen neue Möglichkeiten, Informationen von öffentlichem oder privatem Interesse durch allgegenwärtig verfügbare Smartphones mit entsprechenden Sensoren zu sammeln. Da Smartphones allerdings ständige Begleiter von Personen sind, führt diese Art der Datensammlung zu neuen, in dieser Form bisher nicht da gewesenen Herausforderungen an den Schutz der Privatsphäre der entsprechenden Nutzer/-innen, während gleichzeitig die Nützlichkeit dieser neuen Technik erhalten bleiben soll. Die wenigen Ansätze, die bislang versuchen diese beiden Ziele zu verbinden, erreichen keinen beweisbar sicheren Schutz der Privatsphäre, basieren auf starken Annahmen die zugrunde liegende Infrastruktur betreffend oder bieten keine Sicherheit gegen mehrere, sich verbündende Angreifer.

In dieser Arbeit präsentieren wir das erste kryptographisch fundierte Modell für eine *privatsphäre-schützende partizipative Sensornetz-Infrastruktur*, das als Baustein für Architekturen zur Kommunikation von datengenerierenden und -empfangenden Parteien genutzt werden kann. Wir formalisieren in diesem Zusammenhang mit *Node Privacy*, *Query Privacy* und *Report Unlinkability* die drei wesentlichen Schutzziele für Privatsphäre in partizipativen Sensornetzwerken, welche auf die Vertraulichkeit der gesammelten Daten, die Privatsphäre der empfangenden Parteien und die Anonymität der Sender von Daten – auch bei Angriffen von mehreren, sich verbündenden Parteien – abzielen. Mittels identitätsbasierter Verschlüsselung konstruieren wir eine generische und beweisbar sichere Instantiierung unseres Modells und präsentieren eine konkrete praktische Konstruktion mit der gleichen hohen Effizienz, die vorangegangene Ansätze erreicht haben, allerdings mit nun beweisbarem Schutz der Privatsphäre.

Darüber hinaus erweitern wir im Anschluss unser Modell in generischer Art und Weise, um zusätzlich die *Aggregation von gesammelten Daten* zu ermöglichen, welche den Kommunikationsaufwand weiter reduziert und zudem die erreichte Privatsphäre noch erhöht. Basierend auf additiv-homomorpher identitätsbasierter Verschlüsselung konstruieren wir eine generische Instantiierung unseres Modells, die Datenaggregation unter Schutz der Privatsphäre in partizipativen Sensornetzwerken ermöglicht. Wir präsentieren im Anschluss ein neues, *additiv-homomorphes identitätsbasiertes Verschlüsselungsverfahren*, das Ununterscheidbarkeit und Anonymität von Chiffraten unter der Decisional-Bilinear-Diffie-Hellman-Annahme bei praktikabler Performanz auf kleinen Nachrichtenräumen bietet. Durch Anwendung dieses neuen Verfahrens erhalten wir eine Infrastruktur für partizipative Sensornetzwerke, die beweisbaren Schutz der Privatsphäre mit effizienter Datenaggregation kombiniert.

Acknowledgements

I would like to thank my supervisor Mark Manulis very much for the possibility to work on this interesting topic of my own choice and his encouraging guidance while preserving great freedom for me to explore and integrate various related aspects and ideas. Moreover, I am extremely grateful to Andreas Peter, who stepped into one of our early discussions and since then accompanied my work with countless fruitful discussions, sincere interest, and kind advices that in turn will accompany me for a long time. Furthermore, I am thankful to my colleagues Nils Fleischhacker and Franziskus Kiefer for insightful discussions of ideas and to Franziskus and my brother Oliver Günther for early respectively final cross-reading of my work.

Last, not least, I thank my wife Juliane for her everyday support that made this work possible in the first place.

Contents

1 Introduction

Participatory sensing [10, 11] is novel paradigm to collect data from smartphones and other mobile devices carried by a heavily increasing number of people. Based on this paradigm (also known as opportunistic, people-centric, or urban sensing), a wide range of applications have been suggested that collect and process information on, for example, environmental conditions like traffic [35], urban air [44] and noise pollution [47], free parking slots [41], or earth quakes [18], on market aspects like fuel prices [25], or concerning personal health like diets [48]. All these applications leverage the high and increasing distribution and availability of mobile phones, whose number of subscriptions surpassed 5 billion with an exceeding share of smartphones with sufficient computation power for—at least small—sensing tasks.

In contrast to wireless sensor networks, where sensors are owned, deployed, and maintained by a single organization, individual users act as the owner of sensors in the setting of participatory sensing who service their mobile phones on their own and contribute to a common pool of data, usually stored by a central service provider. However, the usage of people's mobile phones as sensors also introduces new security and privacy aspects that have to be taken care of. Most prominently, sensors in a participatory sensing scenario are no longer stationary devices, but are instead carried around by their owners all the time, thus revealing sensitive data about their location or even image or sound captures if used for according tasks. Additionally, the sensed data is not a priori publicly obtainable and might thus be privacy-sensitive and require appropriate protection when published or reported to a central data pool, whereas data collected in wireless sensor networks in general is legitimately acquirable by the respective organization. Participatory sensing hence introduces the challenging task to handle the obtained data in a secure and privacy-preserving manner while achieving the greatest possible benefit from the sensed data.

In the last years, many approaches have been made to achieve privacy in participatory sensor networks (cf. our treatment of related work in Chapter 2 for an extensive overview). Despite this considerable corpus of work only a single recent work by De Cristofaro and Soriente [22] approached a formally precise definition of security and privacy in participatory sensing for their scheme called *PEPSI*, however excluded important aspects as, e.g., attacks by multiple colluding parties.

This work hence introduces the first comprehensive and cryptographically precise definition of privacy-preserving participatory sensing. To this extent, we take up the architectural model of De Cristofaro and Soriente based on the observation, that common infrastructures for participatory sensing involve the following minimal set of parties:

- *Sensing Devices:* Devices (e.g., smartphones) carried by people, vehicles, or other entities that sense and report data (e.g., temperature, noise level, etc.) using appropriate sensors, forming the basis for participatory sensing. We subsequently refer to those sensing devices as *mobile nodes*.

- *Queriers:* Individuals, institutions, or other entities interested in sensed data (e.g., "noise level on Time Square, New York") that subscribe for such information and receive corresponding sensor reports.

- *Network Operators:* Entities that provide the communication infrastructure of the participatory sensing application.

Additionally, most participatory sensing infrastructures include an intermediary *service provider*, storing data reports received by mobile nodes and processing the data for or relaying it to interested queriers. The service provider is in general an indispensable party in a participatory sensor networks, as mobile nodes are resource-constrained devices being not permanently connected to the network and thus incapable of providing all interested queriers with their data reports by themselves, especially not in a time-delayed manner. However, an intermediary service provider introduces yet further privacy challenges, as it not only receives all data reports but potentially also learns the information interests of all queriers in a participatory sensing application.

We therefore introduce a model of a privacy-preserving participatory sensing infrastructure (PPPSI) based on the described architecture and refine the privacy requirements suggested by De Cristofaro and Soriente in the new model in order to provide three main privacy objectives: *node privacy*, *query privacy*, and *report unlinkability*. Node privacy aims at the protection of both the content and purpose of a data report issued by a mobile node against unauthorized queriers, the service provider, and other mobile nodes, even if all of them collude. Query privacy formalizes the opposite privacy requirement that neither the service provider, nor other queriers or mobile nodes shall be able to determine to what sensing information a querier subscribes. Report unlinkability finally assures the indistinguishability of mobile nodes by requiring, that data reports cannot be traced back to the issuing mobile node by any party. We consider our privacy-preserving participatory sensing infrastructure PPPSI as an independent building block, abstracting from the underlying

network infrastructure. Our model thus addresses the open challenges to provide composeable privacy solution and cryptographically provable privacy for participatory sensing.[1]

Subsequent to the specification of our model, we introduce a generic and provably secure instantiation based on identity-based encryption (IBE) that ensures node privacy, query privacy, and report unlinkability. We moreover provide a practical instantiation of our generic scheme based on the IBE scheme proposed by Boneh and Franklin [6] which not only achieves full privacy in our model (in contrast to the PEPSI scheme by De Cristofaro and Soriente [22] that suffers from collusion attacks on node and query privacy) but performs equally well as the PEPSI scheme in terms of computation, communication, and storage overhead, providing good performance in practice. Our scheme thus solves the open problem to achieve privacy of mobile nodes and queriers under collusion attacks, posed by De Cristofaro and Soriente in their work [22, 23].

In the second part of this work, we take on another aspect of (participatory) sensor networks affecting not only privacy but also their performance, namely the *aggregation* of data reports in order to reduce the overall communication overhead and achieve further increased privacy by hiding single data reports in aggregated values. We therefore extend our privacy-preserving participatory sensing infrastructure to allow for such data aggregation by the service provider while at the same time preserving the confidentiality of data reports. Subsequently, we provide a generic instantiation of the extended model with data aggregation based on *additively homomorphic* identity-based encryption. In order to instantiate our generic construction, we present a novel, additively homomorphic IBE scheme which—to the best of our knowledge—is the first IBE scheme constructed to provide this property. We are able to show that the resulting construction is not only provably secure but also practical for common participatory sensing scenarios and even outperforms the construction based on the Boneh-Franklin IBE scheme for small message spaces.

1.1 Organization

The rest of this work is organized as follows. First, we discuss previous work on privacy-enhanced participatory sensing and other related approaches in Chapter 2. Chapter 3 summarizes important definitions and security notions used throughout this work. We introduce PEPSI [22], the single cryptographic model for participatory sensing introduced so far, in Chapter 4 and subsequently discuss its limitations in Chapter 5. These lead us to Chapter 6, where we introduce our new and comprehensive model for privacy-preserving participatory sensing infrastructures, defining security and privacy formally for such scenarios. In Chapter 7 we then provide a generic and provably secure instantiation of our model based on identity-based encryption, give a concrete example based on the IBE scheme of Boneh and Franklin [6], and discuss its security and efficiency in comparison with the PEPSI scheme. In order to even further improve the privacy and efficiency of our model, we introduce and formalize the use of data aggregation in Chapter 8 and subsequently instantiate the extended model based on our novel additively homomorphic identity-based encryption scheme in Chapter 9. After proving privacy and security of this instantiation with data aggregation, we complete our analysis in the same chapter by comparing the efficiency of our two schemes with and without data aggregation. Finally, we conclude and discuss open research questions and possible future work in Chapter 10.

[1] Cf. for example the survey of Christin et al. [17, Section 5, Challenges 2 and 4].

2 Related Work

The privacy challenges that come along with participatory sensing [10, 11] have been pointed out by many different researchers in the past who emphasized the importance to solve these challenges [52, 36], partly even suggesting the design of privacy-preserving data aggregation schemes [16], however without providing concrete solutions.

One of the first privacy-aware architectures for participatory sensing is *AnonySense* [19], which aims at the anonymity of mobile nodes using mix networks [14] and incorporates separated, non-colluding servers for task issuing and report handling. The proposed scheme achieves k-anonymity [53], but does not provide confidentiality of reports or queries wrt. the service tasking and reporting servers. Later extension of the AnonySense approach by Huang et al. [34] broadened the statistical privacy goals towards ℓ-diversity [40]. However, their scheme still relies on multiple trusted or non-colluding parties and achieves statistical, but no cryptographically provable privacy. Dimitriou et al. [24] introduced *PEPPeR*, a scheme that aims at protecting the privacy of queriers in participatory sensing scenarios and uses cryptographic tokens based on blind signatures [15]. Their work however focuses on querier privacy only and requires queriers to communicate directly with mobile nodes, introducing a high communication overhead and potentially also availability bottlenecks.

The first and—to the best of our knowledge—only framework that aims at cryptographically provable security and privacy is the PEPSI scheme proposed by De Cristofaro and Soriente [22, 23]. For the first time, they approached a cryptographic definition of privacy both of data reporting and querying parties, based on a conceptually simple but versatile architecture with a single trusted entity for key issuing. PEPSI moreover aims at providing not only anonymity of mobile nodes but also confidentiality of data reports against an untrusted service provider. In our work, we pick up the PEPSI model and refine it in order to overcome collusion attacks against the original scheme of De Cristofaro and Soriente. Moreover, we extend our enhanced model to allow for data aggregation, providing additional privacy and further increased efficiency.

Methods for secure data aggregation have already been extensively discussed in the setting of wireless sensor networks (see e.g. [43, 3, 38, 49]), however often focused on external adversaries or aiming primarily at efficiency. Castelluccia et al. [13, 12] for example employ additively homomorphic but still symmetric encryption for data aggregation on the path from mobile nodes to the service provider, achieving private aggregation though without confidentiality of the result wrt. the service provider. *PoolView*, proposed by Ganti et al. [29], constitutes a privacy-preserving aggregation approach for participatory sensor networks based on data perturbation, however designed only for closed communities with a known users set and data distribution. Shi et al. [51] introduced another aggregation scheme for participatory sensing aiming at node privacy, called *PriSense*, where data is fragmented and reported to the service provider on different paths via so-called cover nodes—thereby also introducing additional communication overhead between mobile nodes.

More distant are approaches based on Trusted Platform Modules that aim at the integrity and confidentiality [26] or privacy [31] of reported data.

3 Preliminaries

In this chapter, we introduce the basic encryption paradigms, hardness assumptions, and security notions that build the basis for our subsequent constructions.

3.1 Public-Key and Identity-Based Encryption

In this section we introduce the notions of public-key and identity-based encryption.

Definition 3.1 (Public-Key Encryption Scheme). A *public-key encryption scheme* Π consists of the three algorithms KeyGen, Enc, and Dec defined as follows.

$\mathsf{KeyGen}(1^n)$. On input the security parameter n, this probabilistic algorithm outputs a public key pk and a private key sk.[1]

$\mathsf{Enc}(pk, m)$. On input the public key pk and a message $m \in \mathcal{M}$, this probabilistic algorithm outputs a ciphertext $c \in \mathcal{C}$.

$\mathsf{Dec}(sk, c)$. On input the private key sk, and a ciphertext $c \in \mathcal{C}$, this deterministic algorithm outputs a message $m \in \mathcal{M}$.

To be correct, Π has to satisfy the following condition:

$$\forall\, m \in \mathcal{M} : \mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) = m. \qquad \blacksquare$$

Definition 3.2 (Identity-Based Encryption Scheme). An *identity-based encryption scheme* (IBE scheme) $\mathcal{E}$ consists of the four algorithms Setup, $\mathsf{Extract}$, Enc, and Dec defined as follows.

$\mathsf{Setup}(1^n)$. On input the security parameter n, this probabilistic algorithm outputs the master public key mpk (containing the public system parameters) and the master secret key msk for the scheme.

$\mathsf{Extract}(\mathsf{mpk}, \mathsf{msk}, id)$. On input the master public and secret key mpk and msk and an arbitrary identity $id \in \{0,1\}^*$ (interpreted as public key), this probabilistic algorithm outputs the corresponding private key sk_{id}.

$\mathsf{Enc}(\mathsf{mpk}, id, m)$. On input the master public key, an identity $id \in \{0,1\}^*$, and a message $m \in \mathcal{M}$, this probabilistic algorithm outputs a ciphertext $c \in \mathcal{C}$.

$\mathsf{Dec}(\mathsf{mpk}, sk_{id}, c)$. On input the master public key, a private key sk_{id}, and a ciphertext $c \in \mathcal{C}$, this deterministic algorithm outputs a message $m \in \mathcal{M}$.

To be correct, $\mathcal{E}$ has to satisfy the following condition:

$$\forall\, id \in \{0,1\}^*, \forall\, m \in \mathcal{M} : \mathsf{Dec}(\mathsf{mpk}, sk_{id}, \mathsf{Enc}(\mathsf{mpk}, id, m)) = m, \quad \text{where } sk_{id} \leftarrow \mathsf{Extract}(\mathsf{mpk}, \mathsf{msk}, id). \qquad \blacksquare$$

3.2 Pairings and Related Hardness Assumptions

As the identity-based encryption schemes we use in this work are based on pairings, we subsequently introduce the notion of bilinear groups and pairings as well as the related hardness assumptions used in later security proofs, namely the bilinear Diffie-Hellman assumption and its decisional variant.

Definition 3.3 (Bilinear Group Generator and Bilinear Maps). Let $\mathcal{G}$ be an algorithm that on input the security parameter 1^n outputs a prime number q with $|q| = n$, the description of two groups $\mathbb{G}_1 = \langle g_1 \rangle$, $\mathbb{G}_2 = \langle g_2 \rangle$ of order q, and the description of a map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Its output is denoted by $\mathcal{G}(1^n) = (\mathbb{G}_1 = \langle g_1 \rangle, \mathbb{G}_2 = \langle g_2 \rangle, q, e)$.

$\mathcal{G}$ is called a *bilinear group generator* and e a *bilinear map* (or *pairing*), if the following three properties hold for e:

[1] We assume that n can be determined from both pk and sk.

1. *Efficiency*: The bilinear map e is computable in polynomial time.

2. *Bilinearity*: For all $u \in \mathbb{G}_1$, $v \in \mathbb{G}_2$ and $a, b \in \mathbb{Z}_q$ it holds that $e(u^a, v^b) = e(u, v)^{ab}$.

3. *Non-Degeneracy*: It holds that $e(g_1, g_2) \neq 1$.

If $\mathbb{G}_1 = \mathbb{G}_2$, then we denote both groups by $\mathbb{G} = \langle g \rangle$ and call $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ a *symmetric* pairing. ■

Definition 3.4 (Bilinear Diffie-Hellman (BDH) Assumption). Let $\mathcal{G}$ be a bilinear group generator as defined in Definition 3.3, generating a symmetric pairing. The *bilinear Diffie-Hellman (BDH) assumption* states that for all PPT algorithms $\mathcal{A}$ the following advantage function is negligible in n:

$$\mathrm{Adv}^{\mathsf{BDH}}_{\mathcal{G},\mathcal{A}}(n) := \Pr\left[\mathcal{A}(g, q, e, g^{x_1}, g^{x_2}, g^{x_3}) = e(g,g)^{x_1 x_2 x_3} \;\middle|\; (\mathbb{G} = \langle g \rangle, q, e) \leftarrow \mathcal{G}(1^n), x_1, x_2, x_3 \in_R \mathbb{Z}_q\right]. \quad ■$$

Definition 3.5 (Decisional Bilinear Diffie-Hellman (DBDH) Assumption). Let $\mathcal{G}$ be a bilinear group generator as defined in Definition 3.3, generating a symmetric pairing. The *decisional bilinear Diffie-Hellman (DBDH) assumption* states that for all PPT algorithms $\mathcal{A}$ the following advantage function is negligible in n:

$$\mathrm{Adv}^{\mathsf{DBDH}}_{\mathcal{G},\mathcal{A}}(n) := \left|\Pr\left[\mathcal{A}(g, q, e, g^{x_1}, g^{x_2}, g^{x_3}, h_b) = b \;\middle|\; \begin{array}{c}(\mathbb{G} = \langle g \rangle, q, e) \leftarrow \mathcal{G}(1^n),\ w, x_1, x_2, x_3 \in_R \mathbb{Z}_q, \\ h_0 = e(g,g)^{x_1 x_2 x_3}, h_1 = e(g,g)^w, b \in_R \{0,1\}\end{array}\right] - \frac{1}{2}\right|. \quad ■$$

3.3 Pseudorandom Functions, Hash Functions, and the Random Oracle Model

In this section we shortly recap the definitions of pseudorandom and hash functions, which we employ later in our schemes, and give a short intuition of the random oracle model used in some of our proofs.

Definition 3.6 (Pseudorandom Function). Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be an efficient keyed function (note that we write $f_k(x)$ for $f(k,x)$). We say that f is a *pseudorandom function* (PRF) if for all PPT algorithms $\mathcal{D}$ the advantage function

$$\mathrm{Adv}^{\mathsf{PRF}}_{f,\mathcal{A}}(n) := \left|\Pr\left[\mathcal{D}^{f_k(\cdot)}(1^n) = 1\right] - \Pr\left[\mathcal{D}^{g(\cdot)}(1^n) = 1\right]\right|$$

is negligible, where $k \in_R \{0,1\}^n$ and g is chosen at random from the set of function $\{0,1\}^n \to \{0,1\}^n$. ■

Note that, although we above defined pseudorandom functions to be of fixed length (i.e., on input x of length n the output $f_k(x)$ has length n, too), we will later also make use of *variable-length* pseudorandom functions $f : \{0,1\}^n \times \{0,1\}^* \to \{0,1\}^n$, which can be build by first hashing the input value x using a collision resistant hash function $H : \{0,1\}^* \to \{0,1\}^n$ (see Definition 3.7 below) before applying a (fixed-length) pseudorandom function $f' : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$, i.e., $f_k(x) := f'_k(H(x))$. For the security analysis of this construction we refer to Goldreich [32].

Definition 3.7 (Collision-Resistant Hash Functions). Let H be a keyed hash function forming the function family $\{H_k : \{0,1\}^* \to \{0,1\}^n \mid k \leftarrow \mathsf{Gen}(1^n)\}$ (where Gen generates a random key for H) and $\mathcal{A}$ a PPT adversary in the following collision-resistance game:

$$\begin{array}{l}\mathrm{Game}^{\mathsf{Col\text{-}Res}}_{H,\mathcal{A}}(n): \\ \quad k \leftarrow \mathsf{Gen}(1^n) \\ \quad (m_0, m_1) \leftarrow \mathcal{A}(k) \\ \quad \text{return } H_k(m_0) = H_k(m_1)\end{array}$$

We say that H is *collision-resistant* if for all PPT adversaries $\mathcal{A}$ the following success probability is negligible in n:

$$\mathrm{Succ}^{\mathsf{Col\text{-}Res}}_{H,\mathcal{A}}(n) := \Pr\left[\mathrm{Game}^{\mathsf{Col\text{-}Res}}_{H,\mathcal{A}}(n) = 1\right]. \quad ■$$

For simplicity, we omit the key index when talking about hash function in the rest of this work, i.e., the denotation of a collision-resistant hash function $H\colon \{0,1\}^* \to \{0,1\}^n$ fixes a concrete function uniformly chosen from the family $\{H_k\colon \{0,1\}^* \to \{0,1\}^n \mid k \leftarrow \mathsf{Gen}(1^n)\}$.

In the **random oracle model** (ROM) introduced by Bellare and Rogaway [5], hash functions possess another property besides collision-resistance: A hash function H is modeled in security proofs in the ROM as truly random function, which all participating parties evaluate by querying a random oracle that on input x returns $H(x)$. As such an oracle (being an infinite object due to the domain $\{0,1\}^*$ of H) cannot exist in practice, when implementing a scheme proven secure in the ROM, one has to instantiate the random oracle with a real cryptographic hash function like SHA-1. Due to this gap between security proof and instantiation, proofs in the random oracle model do not imply security in the real world; they however provide a useful validation of cryptographic constructions. Security proofs that do not rely on such random oracles are called proofs in the *standard model*.

One may further distinguish non-programmable from programmable random oracles, where in the latter the output of the random oracle can be programmed by a cryptographic reduction. For a detailed analysis of this differentiation we refer to the work on programmability of random oracles by Fischlin et al. [28]. Throughout this work we however treat random oracles always as programmable ones.

3.4 Security and Privacy Definitions for Encryption

There are numerous definitions that formalize various aspects of security and privacy of encryption schemes. For later reference, we state here the main notions we are using in this work, namely indistinguishability and anonymity under chosen-ciphertext and chosen-plaintext attacks, in separate definitions as well as in the combined form.

We first recap the classical indistinguishability under chosen-ciphertext attacks[2] (due to Rackoff and Simon [46]) and chosen-plaintext attacks (originating to Goldwasser and Micali [33]) for public-key encryption schemes.

Definition 3.8 (Indistinguishability under Chosen-Ciphertext Attacks (IND-CCA)). Let Π be a public-key encryption scheme as defined in Definition 3.1 and $\mathcal{A}$ a PPT adversary in the following IND-CCA game:

$$
\begin{aligned}
&\mathrm{Game}^{\mathsf{IND\text{-}CCA}}_{\Pi,\mathcal{A}}(n):\\
&\quad (sk,pk) \leftarrow \mathsf{KeyGen}(1^n)\\
&\quad (m_0,m_1) \leftarrow \mathcal{A}^{\mathsf{Dec}(sk,\cdot)}(pk)\\
&\quad b \in_R \{0,1\}\\
&\quad c \leftarrow \mathsf{Enc}(pk,m_b)\\
&\quad b' \leftarrow \mathcal{A}^{\mathsf{Dec}(sk,\cdot)}(pk,c)\\
&\quad \text{return } b = b'
\end{aligned}
$$

where we require that $\mathcal{A}$ does not query the Dec oracle on the challenge ciphertext c. The advantage of $\mathcal{A}$ in winning the game $\mathrm{Game}^{\mathsf{IND\text{-}CCA}}_{\Pi,\mathcal{A}}(n)$ is defined as

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

We say that Π provides *indistinguishability under chosen-ciphertext attacks* (or IND-CCA security) if for all PPT adversaries $\mathcal{A}$ the advantage function $\mathrm{Adv}^{\mathsf{IND\text{-}CCA}}_{\Pi,\mathcal{A}}(n)$ is negligible in n. ■

Definition 3.9 (Indistinguishability under Chosen-Plaintext Attacks (IND-CPA)). Let Π be a public-key encryption scheme as defined in Definition 3.1 and $\mathcal{A}$ a PPT adversary in the game $\mathrm{Game}^{\mathsf{IND\text{-}CPA}}_{\Pi,\mathcal{A}}(n)$, which is identical to $\mathrm{Game}^{\mathsf{IND\text{-}CCA}}_{\Pi,\mathcal{A}}(n)$ from Definition 3.8, except that $\mathcal{A}$ is not given access to the Dec oracle. We say that Π provides *indistinguishability under chosen-plaintext attacks* (or IND-CPA security) if for all PPT adversaries $\mathcal{A}$ the following advantage function is negligible in n:

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

■

Boneh and Franklin [6], when proposing their identity-based encryption scheme, extended the public-key notions of indistinguishability to identity-based encryption in the following natural way.

[2] Note that in this work we consider chosen-ciphertext attacks always in the *adaptive* (CCA2) version, where the adversary has access to the decryption oracle in both stages of the attack.

Definition 3.10 (Indistinguishability under Chosen-Ciphertext Attacks for IBE (IND-ID-CCA)). Let $\mathcal{E}$ be an identity-based encryption scheme as defined in Definition 3.2 and $\mathcal{A}$ a PPT adversary in the following IND-ID-CCA game:

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

where we require that $\mathcal{A}$ does neither query the Extract oracle on the challenge identity id^* nor the Dec oracle on sk_{id^*} and the challenge ciphertext c. The advantage of $\mathcal{A}$ in winning the game $\mathrm{Game}^{\mathsf{IND\text{-}ID\text{-}CCA}}_{\mathcal{E},\mathcal{A}}(n)$ is defined as

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

We say that $\mathcal{E}$ provides *indistinguishability under chosen-ciphertext attacks* (or IND-ID-CCA security) if for all PPT adversaries $\mathcal{A}$ the advantage function $\mathrm{Adv}^{\mathsf{IND\text{-}ID\text{-}CCA}}_{\mathcal{E},\mathcal{A}}(n)$ is negligible in n. ■

Definition 3.11 (Indistinguishability under Chosen-Plaintext Attacks for IBE (IND-ID-CPA)). Let $\mathcal{E}$ be an identity-based encryption scheme as defined in Definition 3.2 and $\mathcal{A}$ a PPT adversary in the game $\mathrm{Game}^{\mathsf{IND\text{-}ID\text{-}CPA}}_{\mathcal{E},\mathcal{A}}(n)$, which is identical to $\mathrm{Game}^{\mathsf{IND\text{-}ID\text{-}CCA}}_{\mathcal{E},\mathcal{A}}(n)$ from Definition 3.10, except that $\mathcal{A}$ is not given access to the Dec oracle. We say that $\mathcal{E}$ provides *indistinguishability under chosen-plaintext attacks* (or IND-ID-CPA security) if for all PPT adversaries $\mathcal{A}$ the following advantage function is negligible in n:

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

■

In order to capture the privacy of identities (i.e., public keys) in identity-based ciphertexts, Abdalla et al. [1] adapted the notion of key privacy for public-key encryption (which we do not restate here, cf. Bellare et al. [4] for details) to identity-based encryption schemes as follows.

Definition 3.12 (Anonymity under Chosen-Ciphertext Attacks for IBE (ANO-ID-CCA)). Let $\mathcal{E}$ be an identity-based encryption scheme as defined in Definition 3.2 and $\mathcal{A}$ a PPT adversary in the following ANO-ID-CCA game:

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

where we require that $\mathcal{A}$ does neither query the Extract oracle on the challenge identities id_0 or id_1 nor the Dec oracle on sk_{id_0} or sk_{id_1} and the challenge ciphertext c. The advantage of $\mathcal{A}$ in winning the game $\mathrm{Game}^{\mathsf{ANO\text{-}ID\text{-}CCA}}_{\mathcal{E},\mathcal{A}}(n)$ is defined as

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

We say that $\mathcal{E}$ provides *anonymity under chosen-ciphertext attacks* (or ANO-ID-CCA security) if for all PPT adversaries $\mathcal{A}$ the advantage function $\mathrm{Adv}^{\mathsf{ANO\text{-}ID\text{-}CCA}}_{\mathcal{E},\mathcal{A}}(n)$ is negligible in n. ■

Definition 3.13 (Anonymity under Chosen-Plaintext Attacks for IBE (ANO-ID-CPA)). Let $\mathcal{E}$ be an identity-based encryption scheme as defined in Definition 3.2 and $\mathcal{A}$ a PPT adversary in the game $\text{Game}^{\text{ANO-ID-CPA}}_{\mathcal{E},\mathcal{A}}(n)$, which is identical to $\text{Game}^{\text{ANO-ID-CCA}}_{\mathcal{E},\mathcal{A}}(n)$ from Definition 3.12, except that $\mathcal{A}$ is not given access to the Dec oracle. We say that $\mathcal{E}$ provides *anonymity under chosen-plaintext attacks* (or ANO-ID-CPA security) if for all PPT adversaries $\mathcal{A}$ the following advantage function is negligible in n:

$$\text{Adv}^{\text{ANO-ID-CPA}}_{\mathcal{E},\mathcal{A}}(n) := \left| \Pr\left[\text{Game}^{\text{ANO-ID-CPA}}_{\mathcal{E},\mathcal{A}}(n) = 1\right] - \frac{1}{2} \right|.$$

■

Finally, the notions of anonymity and indistinguishability (for identity-based encryption) can be combined in an equivalent (cf. Lemma 3.16) single notion as follows.

Definition 3.14 (Anonymity and Indistinguishability under Chosen-Ciphertext Attacks for IBE (ANO-IND-ID-CCA)). Let $\mathcal{E}$ be an identity-based encryption scheme as defined in Definition 3.2 and $\mathcal{A}$ a PPT adversary in the following ANO-IND-ID-CCA game:

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

where we require that $\mathcal{A}$ does neither query the Extract oracle on the challenge identities id_0 or id_1 nor the Dec oracle on sk_{id_0} or sk_{id_1} and the challenge ciphertext c. The advantage of $\mathcal{A}$ in winning the game $\text{Game}^{\text{ANO-IND-ID-CCA}}_{\mathcal{E},\mathcal{A}}(n)$ is defined as

$$\text{Adv}^{\text{ANO-IND-ID-CCA}}_{\mathcal{E},\mathcal{A}}(n) := \left| \Pr\left[\text{Game}^{\text{ANO-IND-ID-CCA}}_{\mathcal{E},\mathcal{A}}(n) = 1\right] - \frac{1}{2} \right|.$$

We say that $\mathcal{E}$ provides *anonymity and indistinguishability under chosen-ciphertext attacks* (or ANO-IND-ID-CCA security) if for all PPT adversaries $\mathcal{A}$ the advantage function $\text{Adv}^{\text{ANO-IND-ID-CCA}}_{\mathcal{E},\mathcal{A}}(n)$ is negligible in n. ■

Definition 3.15 (Anonymity and Indistinguishability under Chosen-Plaintext Attacks for IBE (ANO-IND-ID-CPA)). Let $\mathcal{E}$ be an identity-based encryption scheme as defined in Definition 3.2 and $\mathcal{A}$ a PPT adversary in the game $\text{Game}^{\text{ANO-IND-ID-CPA}}_{\mathcal{E},\mathcal{A}}(n)$, which is identical to $\text{Game}^{\text{ANO-IND-ID-CCA}}_{\mathcal{E},\mathcal{A}}(n)$ from Definition 3.14, except that $\mathcal{A}$ is not given access to the Dec oracle. We say that $\mathcal{E}$ provides *anonymity and indistinguishability under chosen-plaintext attacks* (or ANO-IND-ID-CPA security) if for all PPT adversaries $\mathcal{A}$ the following advantage function is negligible in n:

$$\text{Adv}^{\text{ANO-IND-ID-CPA}}_{\mathcal{E},\mathcal{A}}(n) := \left| \Pr\left[\text{Game}^{\text{ANO-IND-ID-CPA}}_{\mathcal{E},\mathcal{A}}(n) = 1\right] - \frac{1}{2} \right|.$$

■

Lemma 3.16. *An identity-based encryption scheme $\mathcal{E}$ provides* ANO-ID-CCA *and* IND-ID-CCA *security if and only if it provides* ANO-IND-ID-CCA *security. The same holds for the* -CPA *variants.*

Proof (informal). The implication from left to right can be proven using two game hops: First, the challenger in the ANO-IND-ID-CCA game replaces the tuple (id_0, m_0) with (id_0, m_1) (which the adversary cannot distinguish due to the IND-ID-CCA security of $\mathcal{E}$), then the challenger exchanges the tuple (id_0, m_1) with (id_1, m_1) (indistinguishable by the adversary due to the ANO-ID-CCA security of $\mathcal{E}$). Now both challenge tuples are identical (namely (id_1, m_1)).

The implication from right to left holds trivially, as every successful adversary against the ANO-ID-CCA or IND-ID-CCA security of $\mathcal{E}$ is equally successful in the ANO-IND-ID-CCA game (by choosing $m_0 = m_1 = m$ resp. $id_0 = id_1 = id^*$). □

3.5 The Identity-Based Encryption Scheme by Boneh and Franklin

As some of our proposed schemes as well as the PEPSI scheme base on the ANO-IND-ID-CPA-secure IBE scheme introduced by Boneh and Franklin [6, 7] (labeled "BasicIdent" in their paper), we briefly recap this scheme here.

Definition 3.17 (Boneh-Franklin Scheme [6]). Let $\mathcal{G}$ be a bilinear group generator (for a symmetric pairing) as defined in Definition 3.3.

$\mathsf{Setup}(1^n)$. Run $\mathcal{G}(1^n)$ to obtain a prime q, two groups $\mathbb{G} = \langle g \rangle$ and $\mathbb{G}_T$ of order q, and a bilinear map $e\colon \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Choose $x \in_R \mathbb{Z}_q^*$ and set $y := g^x$. Choose two cryptographic hash functions $H_1\colon \{0,1\}^* \to \mathbb{G}^*$ and $H_2\colon \mathbb{G}_T \to \{0,1\}^\ell$ for some ℓ; both are modeled as random oracles in the security analysis.

The message space is $\mathcal{M} = \{0,1\}^\ell$, the ciphertext space is $\mathcal{C} = \mathbb{G}^* \times \{0,1\}^\ell$. Output the master public key $\mathsf{mpk} = (q, \mathbb{G} = \langle g \rangle, \mathbb{G}_T, e, \ell, y, H_1, H_2)$ and the master secret key $\mathsf{msk} = x$.

$\mathsf{Extract}(\mathsf{mpk}, \mathsf{msk}, id)$. Compute and output $sk_{id} := H_1(id)^x$.

$\mathsf{Enc}(\mathsf{mpk}, id, m)$. Choose $r \in_R \mathbb{Z}_q^*$ and output the ciphertext $c = (c_1, c_2) = (g^r, m \oplus H_2(e(H_1(id), y)^r))$.

$\mathsf{Dec}(\mathsf{mpk}, sk_{id}, c)$. Parse c as (c_1, c_2). Compute $c_2 \oplus H_2(e(sk_{id}, c_1)) = m$. ■

Boneh and Franklin [7, Theorem 4.1] proved their scheme to be IND-ID-CPA-secure under the BDH assumption (cf. Definition 3.4); Abdalla et al. [2, Theorem 4.4] have shown—also under the BDH assumption—its ANO-ID-CPA security (both in the random oracle model). The Boneh-Franklin scheme thus by Lemma 3.16 provides ANO-IND-ID-CPA security.

Theorem 3.18 (ANO-IND-ID-CPA Security of the Boneh-Franklin Scheme). *If the BDH assumption from Definition 3.4 holds for $\mathcal{G}$ and the hash functions H_1 and H_2 are random oracles, then the Boneh-Franklin identity-based encryption scheme defined in Definition 3.17 provides anonymity and indistinguishability under chosen-plaintext attacks (*ANO-IND-ID-CPA *security).*

4 PEPSI: Model and Instantiation

In 2011, De Cristofaro and Soriente proposed with PEPSI [22] (privacy-enhanced participatory sensing infrastructure) the first approach to cryptographically treat privacy in participatory sensing. In this chapter, we recall the PEPSI model together with its instantiation proposed by De Cristofaro and Soriente [22]. For details of the introduced schemes we stick to the extended version [21] of their paper.

4.1 Infrastructure and Operations

The infrastructure for participatory sensing is modeled in PEPSI by interaction of the following *parties*.

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.

Network Operator (NO): The network operator provides cellular network access for mobile nodes.

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.

The different parties interact with each other by the following *operations*.

Setup: In this operation, the registration authority generates all necessary parameters and cryptographic keys.

MN Registration: Users (i.e., mobile nodes) register for sensing (once) at the registration authority.

Query Registration: Queriers request authorization to query certain readings (e.g., "temperature in Berlin, Germany", later denoted as "query identifier") from the registration authority and accordingly subscribes for those readings at the service provider.

Data Report: Mobile nodes report their sensed data (via the network operator's network) to the service provider.

Query Execution: The service provider compares received data reports with registered query subscriptions in order to relay matching reports to the according queriers.

The complete infrastructure (together with the operations) introduced in PEPSI is illustrated in Figure 4.1.

4.2 Soundness and Privacy Requirements

In the following we give a brief intuition of the *soundness and privacy requirements* posed along with PEPSI. For a more detailed definition we refer to the extended PEPSI paper [21, Section 3.3 and Appendix B].

Soundness: As the transfer of data from mobile nodes to (authorized) queriers is the main goal of participatory sensing, a PEPSI instantiation is said to be sound, if an appropriately authorized querier receives the corresponding data reports.

Node Privacy: A PEPSI instantiation provides node privacy if neither the network operator, nor the service provider, nor any unauthorized querier or other mobile node learns anything about the data reported by a mobile node or its purpose (i.e., its query identifier).

Query Privacy: A PEPSI instantiation provides query privacy if neither the network operator, nor the service provider, nor any mobile node or other querier learns anything about the query identifier a querier subscribes to.

Report Unlinkability: A PEPSI instantiation provides report unlinkability if no party can link two (or more) data reports as originating from the same mobile node.

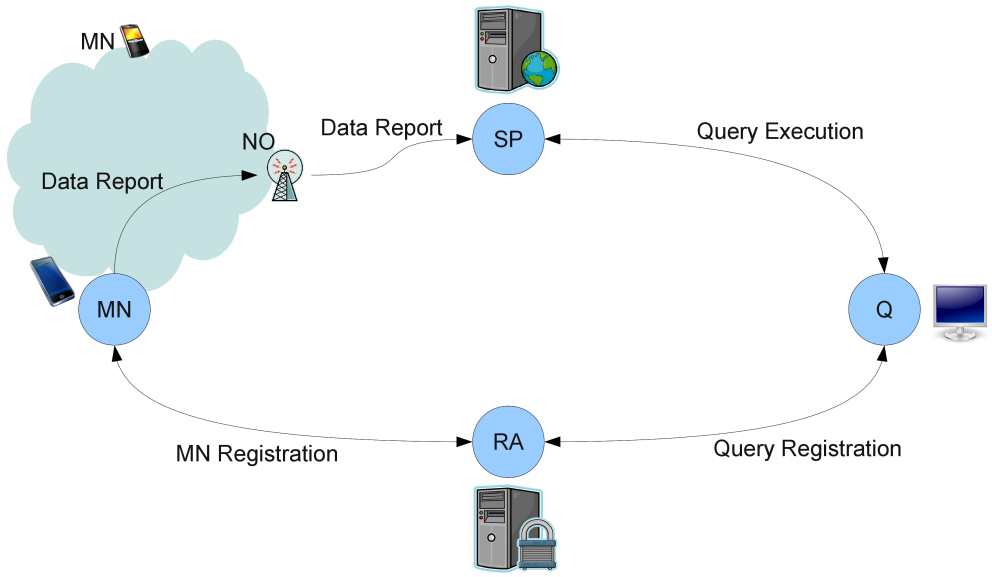

Figure 4.1: The PEPSI infrastructure. Mobile nodes (MNs) and queriers (Qs) register to the registration authority (RA). Mobile nodes report data via the network operator (NO) to the service provider (SP), which sends them to queriers with matching subscriptions.

4.3 Instantiation by De Cristofaro and Soriente

De Cristofaro and Soriente proposed an instantiation of the PEPSI model in their work [22, 21], which uses an encryption approach derived from the identity-based encryption scheme proposed by Boneh and Franklin [6, 7] (cf. Definition 3.17). We restate their instantiation of the operations in the PEPSI model in the following and illustrate it in Figure 4.2.

Setup: The registration authority (RA), given a security parameter n runs $\mathcal{G}(1^n)$ to obtain a prime q, two groups $\mathbb{G} = \langle g \rangle$ and $\mathbb{G}_T$ of order q, and a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. It chooses $s \in_R \mathbb{Z}_q^*$ and sets $Q := g^s$. g and Q are public parameters, s is RA's master secret key msk.

Then, RA chooses a "nonce" $z \in_R \mathbb{Z}_q^*$ and sets $R := g^z$. Finally, the RA chooses three cryptographic hash functions $H_1 : \{0,1\}^* \to \mathbb{G}$, $H_2 : \mathbb{G}_T \to \{0,1\}^n$, and $H_3 : \mathbb{G}_T \to \{0,1\}^n$.

MN Registration: The mobile nodes registers at the registration authority and obtains the pair (z, id) where z is the nonce generated during setup and id the identifier for the readings the mobile node provides

Query Registration: The querier registers at the registration authority for some query identifier id^* and obtains the pair (sk_{id^*}, R) where $sk_{id^*} := H_1(id^*)^s$ is computed by the RA using its master secret key msk.[1]

In the next step, the querier subscribes for his query identifier id^* by sending $T^* := H_2(e(R, sk_{id^*}))$ to the service provider (SP).

Data Report: The mobile node, in order to submit a data reading m, sends the service provider (using the infrastructure of the network operator (NO)) the pair $(T, c) := (H_2(e(Q, H_1(id)^z)), \mathsf{Enc}_k(m))$, where $k := H_3(e(Q, H_1(id)^z))$ is the key for some symmetric encryption operation Enc, e.g., AES. T is called a "tag", c is the ciphertext containing the data.

Query Execution: The service provider matches a tag T of a reading sent by some mobile node with the stored query subscriptions T^* and forwards the reading pairs (T, c) to the queriers with matching T^*.

The querier, on receiving (T, c), computes $k^* := H_3(e(R, sk_{id^*}))$ and obtains $m = \mathsf{Dec}_{k^*}(c)$.

As the set of mobile nodes is assumed to be dynamic, where new nodes can register and malicious ones can be excluded, De Cristofaro and Soriente propose a periodic "nonce renewal", where the RA chooses a fresh z and distribute z to the MNs (e.g., using broadcast encryption) and $R = g^z$ to the queriers.

[1] It is not explicitly stated that the querier obtains R in the original paper, but as the querier uses R in the next step of the registration, this is implicitly required.

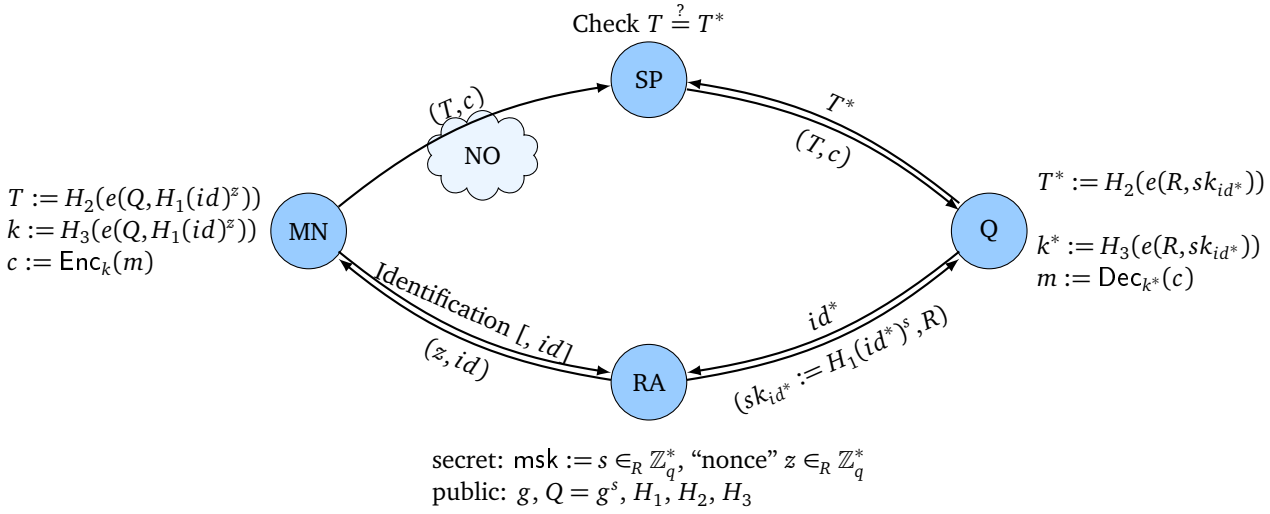

Figure 4.2: Instantiation of the PEPSI model as proposed by De Cristofaro and Soriente [22, 21].

5 Limitations of PEPSI

As already acknowledged by De Cristofaro and Soriente [22, 23], the proposed PEPSI model and instantiation provides only limited collusion resistance. Some of the possible collusions (e.g., between the service provider and queriers) are—by assumption—excluded in the PEPSI model, others remain unmentioned and thus constitute an actual breach of security. Participatory sensing is employed in a setting where especially the data providing mobile nodes, but potentially also the queriers, cannot be uniquely identified and should thus not be trusted. Moreover, a main goal of PEPSI was to provide privacy against an untrusted service provider. We thus argue that collusion resistance (against collusions of the service provider, mobile nodes, and preferably also queriers) is an essential requirement for a participatory sensing infrastructure that aims at preserving privacy.

In this chapter, we catch up on the discussion of this important security aspect by analyzing all possible collusions in the PEPSI model and their impact for the instantiation given by De Cristofaro and Soriente. Unfortunately, it turns out that the proposed instantiation is unable to provide collusion resistance in the own PEPSI model against two of the most considerable collusion attacks (namely, collusion of the service provider and a mobile node and collusion of a mobile node and a querier). In addition we argue that the PEPSI model itself does not take account of all collusions that should be considered (as, e.g., collusions of the service provider and a querier).

5.1 Possible Collusions and Their Impact

In the following we discuss all possible collusions of parties interacting in the PEPSI model and their impact on the proposed privacy requirements (cf. Section 4.2) for the instantiation given by De Cristofaro and Soriente (cf. Section 4.3). We therefore analyze all pairwise collusions of a mobile node (MN), a querier (Q), the service provider (SP), the network provider (NP), and the registration authority (RA). The results are described in the following and summarized in what we call the "collusion impact matrix" for the PEPSI instantiation in Table 5.1.

MN-Q: MN possesses the nonce z, which allows to compute the tag T for any identity id. With this tag, colluding MN and Q can register for any identity to receive (encrypted) readings. Using nonce z, they can also compute the according key k for any identity and thus decrypt the readings (breaking *node privacy*).

MN-SP: MN possesses the nonce z, which allows to compute the key k for any identity and thus decrypt all readings SP receives (breaking *node privacy*). Also, the tag T for any identity id can be computed from z, and thus colluding MN and SP can check to which identity a querier registers (breaking *query privacy*).

MN-NO: MN possesses the nonce z, which allows to compute the key k for any identity and thus decrypt arbitrary readings (which would break *node privacy*). However, the readings NO receives are by assumption encrypted under SP's public key, preventing this attack.

MN-RA: As the RA already knows z and neither MN nor RA receive any (encrypted) readings, this collusion has no impact.

Q-SP: Q possesses the private keys sk_{id} for all identities id he is registered for. This allows to compute the tag T for all those identities and thus colluding Q and SP could check to which identity other queriers register (which would *partially* break *query privacy*). However, by assumption SP does not collude with queriers.

Note that, although colluding Q and SP can decrypt all readings for identities id Q knows the private key sk_{id} for, this does not constitute a (partial) node privacy breach as Q is authorized to decrypt those readings.

Q-NO: As NO does not receive any query subscriptions, there cannot be a query privacy breach. Moreover, Q can only decrypt readings for identities for which he already has a secret key sk_{id}, so there also is no node privacy breach. Thus, this collusion has no impact.

Q-RA: As the RA is able to generate the tag and secret key for any identity, it is no surprise that, if acting as or colluding with a querier, the registration for arbitrary identities and decryption of all received readings is possible (breaking *node privacy*). We argue that this impact is clearly inevitable in this setting and thus the RA has to be trusted to not collude.

SP-NO: As neither the SP nor the NO possess any secret information, this collusion has no impact.

	Querier	Service Provider	Network Operator	Registration Authority
Mobile Node	NP	NP, QP	$[\text{NP}]_1$	$\emptyset$
Querier		$[\text{partial QP}]_2$	$\emptyset$	NP
Service Provider			$\emptyset$	$[\text{NP}, \text{QP}]_2$
Network Operator				$[\text{NP}]_1$

NP — node privacy break; QP — query privacy break; $[\dots]_n$ — mitigated by assumption n below
1: Readings are sent to SP encrypted under SP's public key.
2: SP does not collude with RA or queriers.

Table 5.1: Collusion impact matrix for the instantiation of the PEPSI model by De Cristofaro and Soriente [22, 21].

SP-RA: As the RA is able to generate the tag and secret key for any identity, it is no surprise that, if colluding with the SP, the decryption of all received readings and exposure of all query subscriptions is possible (breaking *node privacy* and *query privacy*). We argue that this impact is clearly inevitable in this setting and thus the RA has to be trusted to not collude.

NO-RA: As the RA is able to generate the secret key for any identity, it is no surprise that, if colluding with the NO, the decryption of all readings passing could be possible (which would break *node privacy*). However, the readings NO receives are by assumption encrypted under SP's public key, preventing this attack.

5.2 Security Breaches in the Model

In the analysis of possible collusions in the previous section, it turned out there are two important breaches of the privacy requirements defined in the PEPSI model, which are neither excluded in its definition nor mitigated by additional assumptions. For this reason, we discuss both—namely the collusion of the service provider and a mobile node and the collusion of a mobile node and a querier—in the following.

5.2.1 Collusion of the Service Provider and a Mobile Node

Not being excluded in the PEPSI model and thus possible, the collusion of the service provider and a mobile node leads to a full breach of node privacy and query privacy as follows.

All mobile nodes possess the "nonce" z (being system-wide the same value), which enables each of them to compute the key k and tag T for any identity id. Using this capability, the service provider and an arbitrary colluding mobile node can together decrypt all data reports the service provider receives, thus breaking *node privacy*. They are also able to compare the tag of any registering querier against a precomputed list of the tags of all identities, thus breaking *query privacy*. This impact is depicted in Figure 5.1.

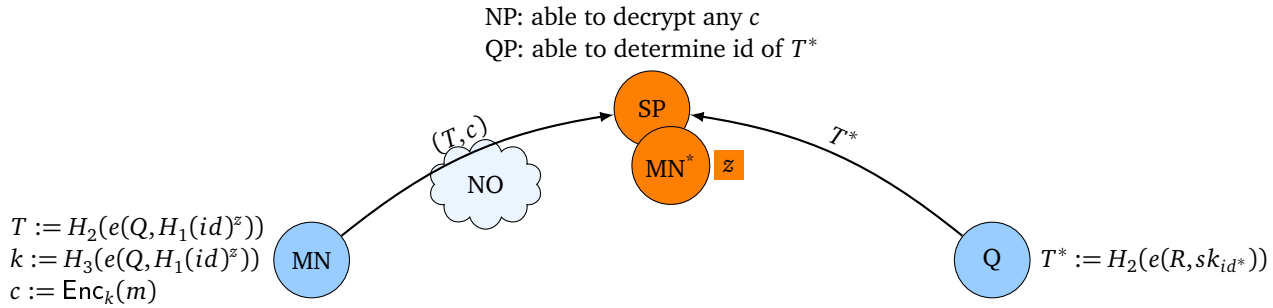

Figure 5.1: Impact of colluding service provider SP and mobile node MN* on node privacy (NP) and query privacy (QP).

5.2.2 Collusion of a Mobile Node and a Querier

The collusion of an arbitrary mobile node with an arbitrary querier also is not excluded in the PEPSI model and thus possible. It leads to a full breach of node privacy as discussed in the following.

Again, the colluding mobile node and querier can use the "nonce" z (possessed by the mobile node) to compute the key k and tag T for any identity id. Using this capability, they are able to register for any identity (by computing the appropriate tag) to receive the according (encrypted) readings. By computing the key k for this identity, they can also decrypt these readings, thus breaking *node privacy*. This impact is depicted in Figure 5.2.

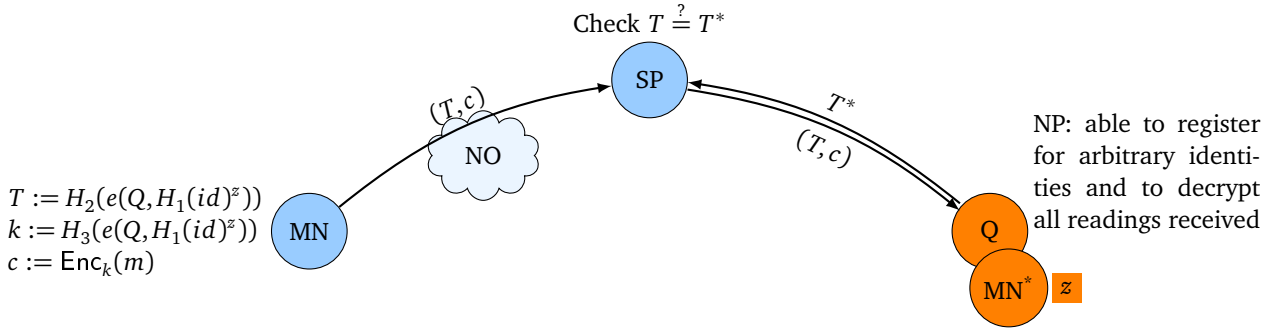

Figure 5.2: Impact of colluding mobile node MN* and querier Q on node privacy (NP).

5.3 Further Aspects of PEPSI's Privacy Definitions

Collusions of a mobile node and the service provider resp. a mobile node and a querier are allowed in the PEPSI model (and should be). However, the formal definitions of node and query privacy given by De Cristofaro and Soriente in the extended version of their paper [21, Appendix B] do not capture either of both, as the adversary in the respective games does not receive the "nonce" z, possessed by each mobile node.

Additionally, the node privacy game wrt. the service provider resembles the ANO-IND-ID-CPA game (cf. Definition 3.15), however without providing the adversary with oracle access for key extraction or even encryption[1]. In the definition of node privacy wrt. unauthorized queriers, extraction queries are possible, but encryption again is not. Finally, query privacy is modeled similar to the ANO-ID-CPA game, though again without oracle access for key extraction or encryption given to the adversary.

We argue that not only collusions, but also the possibility that an adversary sees encryptions of chosen plaintexts or obtains secret keys for some query identities should be reflected in the security model for a participatory sensing infrastructure aiming at privacy protection. Therefore, in the next chapter, we propose a new security model which captures all these aspects while preserving the participatory sensing scenario of PEPSI.

[1] Note that for encryption, the semi-public "nonce" z is used, which is known only to mobile nodes.

6 Security Model

As we have seen in the previous chapters, the system architecture proposed by De Cristofaro and Soriente [22, 21] suits the scenario of participatory sensing well. However, their privacy and security model unfortunately does not cover all aspects of this scenario and especially leaves important collusion attacks (e.g., of a mobile node and the service provider) unconsidered.

For this reason we introduce a new and comprehensive **Privacy-Preserving Participatory Sensing Infrastructure (PPPSI)**, which keeps the general participatory sensing architecture used in PEPSI and complements it with a well-founded model for strong privacy protection and security in this scenario. We conclude with short argument, why the allowed collusion attacks render the PEPSI scheme insecure in our model, before we introduce generic and concrete instantiations of our model providing full privacy in the next chapter.

6.1 The Privacy-Preserving Participatory Sensing Infrastructure PPPSI

We subsequently introduce our refined model, namely the privacy-preserving participatory sensing infrastructure (PPPSI), by first describing the involved parties and operations on a high level and then defining formally what an instantiation of this model is.

6.1.1 Parties

The privacy-preserving participatory sensing infrastructure PPPSI involves the following *parties*, taking the same 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.

Note that, in contrast to the PEPSI model, we do not model the network operator (providing the cellular network infrastructure for mobile nodes) as a distinct party, as it has less attack capabilities than the service provider and we will allow the adversary to corrupt the latter in our security games.

6.1.2 Operations

The different parties interact with each other by the following *operations*. We describe here the high-level intuition of the operations and give the formal treatment of the according algorithms in the next subsection.

Setup: The setup algorithm, denoted by `Setup`, is executed by the registration authority to initialize the PPPSI. It generates the registration authority's secret key and the general public key, containing the system parameters.

Mobile Node Registration: The mobile node registration algorithm, denoted by `RegisterMN`, is executed by the registration authority to register a new mobile node for a given query string (or: *query identity*, e.g., "temperature in Berlin, Germany") the mobile node wants to report data for. The registration authority sends the issued registration value to the mobile node.

Querier Registration: The querier registration algorithm, denoted by `RegisterQ`, is executed by the registration authority to register a new querier for a given query string (or: *query identity*) the querier wants to receive data reports for. The registration authority sends the issued registration value to the querier.

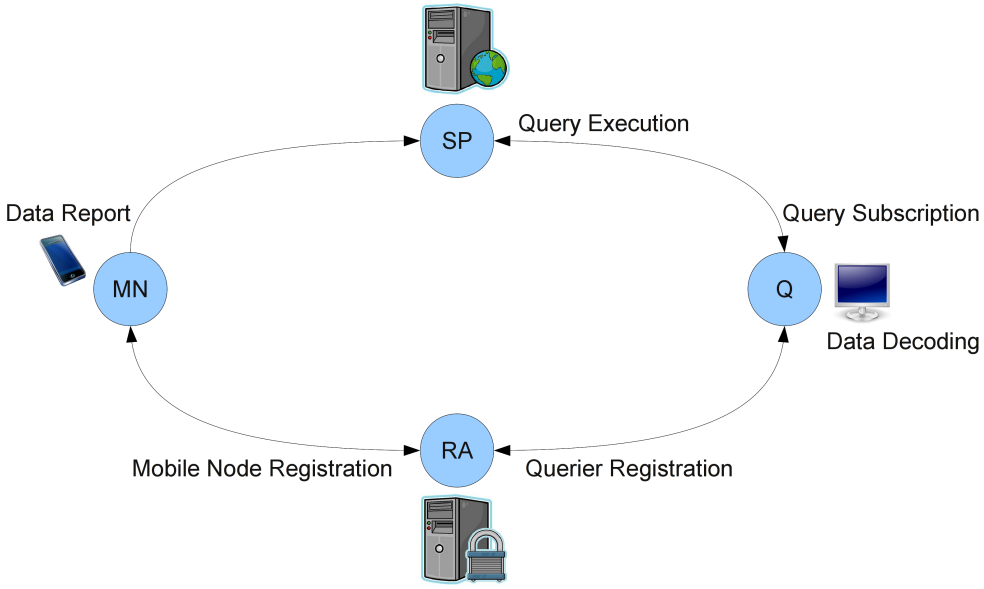

Figure 6.1: PPPSI infrastructure. Mobile nodes (MNs) and queriers (Qs) register to the registration authority (RA). Mobile nodes report data to the service provider (SP), queriers subscribe for reports at the service provider. The service provider sends reports matching with subscriptions to the according querier, which decodes them.

Data Report: The data report algorithm, denoted by `ReportData`, is executed by the mobile node to generate a report for a given query identity and message, which it sends to the service provider.

Query Subscription: The query subscription algorithm, denoted by `SubscribeQuery`, is executed by the querier to generate a subscription token for a given query identity, which it sends to the service provider in order to subscribe to reports for this query identity.

Query Execution: The query execution algorithm, denoted by `ExecuteQuery`, is executed by the service provider on a data report and a subscription token and, if both match, outputs the data report, which the service provider sends to the querier which provided the token.

Data Decoding: The data decoding algorithm, denoted by `DecodeData`, is executed by a querier on a received data report in order to decode it and obtain the encoded message.

The complete infrastructure together with the operations as defined for PPPSI is illustrated in Figure 6.1.

6.1.3 Instantiation

We now formally define what an instantiation of the privacy-preserving participatory sensing infrastructure (PPPSI) is.

Definition 6.1 (PPPSI Instantiation). An *instantiation of the privacy-preserving participatory sensing infrastructure* (PPPSI instantiation) PI consists of the seven algorithms `Setup`, `RegisterMN`, `RegisterQ`, `ReportData`, `SubscribeQuery`, `ExecuteQuery`, and `DecodeData` defined as follows.

`Setup`(1^n)**:** On input the security parameter n, this probabilistic algorithm outputs a secret key RAsk for the registration authority and a master public key RApk. RApk contains a description of the query identity space $\mathcal{I}$ and the message space $\mathcal{M}$.

`RegisterMN`$(\mathsf{RApk}, \mathsf{RAsk}, qid)$**:** On input the master public key RApk, the registration authority's secret key RAsk, and a query identity $qid \in \mathcal{I}$, this probabilistic algorithm outputs a mobile node registration value regMN_{qid} for qid.

`RegisterQ`$(\mathsf{RApk}, \mathsf{RAsk}, qid)$**:** On input the master public key RApk, the registration authority's secret key RAsk, and a query identity $qid \in \mathcal{I}$, this probabilistic algorithm outputs a querier registration value regQ_{qid} for qid.

`ReportData`$(\mathsf{RApk}, \mathsf{regMN}_{qid}, qid, m)$**:** On input the master public key RApk, a mobile node registration value regMN_{qid}, a query identity $qid \in \mathcal{I}$, and a message $m \in \mathcal{M}$, this probabilistic algorithm outputs a data report c.

$\texttt{SubscribeQuery}(\mathsf{RApk}, \mathsf{regQ}_{qid}, qid)$: On input the master public key RApk, a querier registration value regQ_{qid}, and a query identity $qid \in \mathcal{I}$, this probabilistic algorithm outputs a subscription token s.

$\texttt{ExecuteQuery}(\mathsf{RApk}, c, s)$: On input the master public key RApk, a data report c, and a subscription token s, this deterministic algorithm outputs either c or $\perp$, indicating failure.

$\texttt{DecodeData}(\mathsf{RApk}, \mathsf{regQ}_{qid}, qid, c)$: On input the master public key RApk, a querier registration value regQ_{qid}, a query identity $qid \in \mathcal{I}$, and a data report c, this deterministic algorithm outputs either a message m or $\perp$, indicating failure.

To be *sound*, a PPPSI 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, i.e.:

$$\begin{aligned}
&\forall\, n \in \mathbb{N}, \forall\, (\mathsf{RAsk}, \mathsf{RApk}) \leftarrow \texttt{Setup}(1^n),\\
&\forall\, qid \in \mathcal{I}, \forall\, \mathsf{regMN}_{qid} \leftarrow \texttt{RegisterMN}(\mathsf{RApk}, \mathsf{RAsk}, qid), \forall\, \mathsf{regQ}_{qid} \leftarrow \texttt{RegisterQ}(\mathsf{RApk}, \mathsf{RAsk}, qid),\\
&\forall\, m \in \mathcal{M}, \forall\, c \leftarrow \texttt{ReportData}(\mathsf{RApk}, \mathsf{regMN}_{qid}, qid, m), \forall\, s \leftarrow \texttt{SubscribeQuery}(\mathsf{RApk}, \mathsf{regQ}_{qid}, qid) :\\
&\quad \texttt{DecodeData}(\mathsf{RApk}, \mathsf{regQ}_{qid}, qid, \texttt{ExecuteQuery}(\mathsf{RApk}, c, s)) = m.
\end{aligned}$$

■

6.2 Trust Assumptions

In contrast to the PEPSI model, we drop essentially all trust assumptions concerning mobile nodes, queriers, and the service provider as well as their interaction. We argue that first, the service provider—being at the center of data exchange—should not be trusted at all, i.e., it should neither be provided with data reports in clear, nor should it be trusted to not collude with other parties in order to break users' privacy. Second, mobile nodes—i.e., arbitrary users that collect data in a participatory sensing scenario—in general will not be authenticated and should thus not be trusted, too. Third, although queriers receive decryption keys and will hence potentially be authenticated, they should not be trusted to not collude with mobile nodes or the service provider, as such behavior is most probably undetectable in practice.

Based on these restricted trust assumptions, we subsequently define our adversary model for the privacy-preserving participatory sensing infrastructure (PPPSI), where we allow the adversary to corrupt mobile nodes, queriers, the service provider, and—in some cases—even the registration authority (the last of course only *after* the whole scheme was set up). By this we model not only the distrust that should be placed into the service provider, mobiles nodes, and queriers, but also the possibility that some of them (or all) collude in order to attack the privacy of other mobile nodes or queriers.

As our PPPSI model focuses on the higher-level application of participatory sensing, we assume that the involved parties communicate over secure channels, which however do not need to be authenticated as we do not consider identification of mobile nodes or queriers in our work.[1]

6.3 Adversary Model

In order to define security and privacy of a PPPSI instantiation PI, we consider a probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ interacting with PI via the oracle functions defined below.

We allow for corruptions of mobile nodes, queriers, the service provider, and (in special cases) the registration authority. By $\mathcal{CI}_{MN}$ resp. $\mathcal{CI}_{Q}$ we denote the set of identities for which $\mathcal{A}$ learned registration values due to corruptions of mobile nodes resp. queriers and set $\mathcal{CI} := \mathcal{CI}_{MN} \cup \mathcal{CI}_{Q}$ to be the union of both sets. Corruption of the service provider resp. the registration authority is denoted by $\mathcal{C}_{SP} = 1$ resp. $\mathcal{C}_{RA} = 1$; initially both $\mathcal{C}_{SP} := 0$ and $\mathcal{C}_{RA} := 0$.

The oracles $\mathcal{A}$ is given access to are defined as follows.

$\mathsf{CorruptMN}(qid)$: On input a query identity 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 identity 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 $\mathcal{C}_{SP} := 1$. (As the service provider does not possess a secret key, this query reveals none to the adversary. However, the changed value of $\mathcal{C}_{SP}$ influences the results of subsequent ReportData queries (see below).)

[1] Note that in practice one might want to use authenticated channels for the registration of queriers (and potentially also of mobile nodes) in order to identify the registering party.

CorruptRA(): Provide $\mathcal{A}$ with RAsk and set $\mathcal{C}_{RA} := 1$. (Note that, although $\mathcal{A}$ can now compute all oracles on its own and potentially manipulate its RAsk copy, the challenge computation in later games will always be done using the original, unmodified RAsk as output by the initial Setup execution.)

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

If $\mathcal{C}_{SP} = 1$, c is given to $\mathcal{A}$. Otherwise the vector $\mathbf{c} := (c_1, \ldots, c_k)$ is given to $\mathcal{A}$, where c_i is computed as $c_i \leftarrow \mathtt{ExecuteQuery}(\mathsf{RApk}, c, s_i)$ for $i \in \{1, \ldots, k\}$. (Note that the value of some c_i may be $\perp$.)

SubscribeQuery(qid): On input a query identity qid, compute $\mathsf{regQ}_{qid} \leftarrow \mathtt{RegisterQ}(\mathsf{RApk}, \mathsf{RAsk}, qid)$ and $s \leftarrow \mathtt{SubscribeQuery}(\mathsf{RApk}, \mathsf{regQ}_{qid}, qid)$ and provide $\mathcal{A}$ with s.

DecodeData(qid, c): On input a query identity qid and a data report c, compute $\mathsf{regQ}_{qid} \leftarrow \mathtt{RegisterQ}(\mathsf{RApk}, \mathsf{RAsk}, qid)$ and $m \leftarrow \mathtt{DecodeData}(\mathsf{RApk}, \mathsf{regQ}_{qid}, qid, c)$ and provide $\mathcal{A}$ with m.

6.4 Privacy and Security Definitions

Like De Cristofaro and Soriente in the PEPSI model [22, 21], we are interested in three main security properties that we likewise call *node privacy*, *query privacy*, and *report unlinkability*. In the following, we provide the high-level intuition as well as an exact formal definition of all three notions.

6.4.1 Node Privacy

Our notion of *node privacy* formalizes the most obvious privacy and security requirement in a participatory sensing scenario, namely the confidentiality of data reports and their purpose with respect to the service provider, unauthorized queriers, and other mobile nodes. To be precise, we want to hide both the message within a data report as well as the query identity a report was generated for from these parties, even if all of them collude. We therefore model node privacy as indistinguishability of data reports generated from two query identity-message pairs freely chosen by the the adversary, while allowing the adversary to corrupt the service provider as well as mobile nodes and queriers for other query identities.

Definition 6.2 (Node Privacy under Chosen-Ciphertext Attacks). Let PI be a PPPSI instantiation and $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ a PPT adversary interacting with PI via the queries defined in Section 6.3 within in the following game $\mathrm{Game}^{\text{NP-CCA}}_{\mathsf{PI},\mathcal{A}}(n)$:

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

Phase I. $\mathcal{A}_1$ receives RApk and is given access to the oracles CorruptMN, CorruptQ, CorruptSP, ReportData, SubscribeQuery, and DecodeData.

Eventually, $\mathcal{A}_1$ stops and outputs a tuple $((qid_0, m_0), (qid_1, m_1), \mathbf{s})$ containing 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 uniformly at random. $\mathtt{RegisterMN}(\mathsf{RApk}, \mathsf{RAsk}, qid_b)$ is executed and outputs regMN_{qid_b}. Then $\mathtt{ReportData}(\mathsf{RApk}, \mathsf{regMN}_{qid_b}, qid_b, m_b)$ is executed and outputs a report c.

If $\mathcal{C}_{SP} = 1$, set $\mathbf{R} := (c)$. Otherwise set $\mathbf{R} := (c_1, \ldots, c_k)$, where c_i is computed as $c_i \leftarrow \mathtt{ExecuteQuery}(\mathsf{RApk}, c, s_i)$ for $i \in \{1, \ldots, k\}$.[2]

Phase II. $\mathcal{A}_2$ receives RApk and $\mathbf{R}$ and is again given access to the oracles CorruptMN, CorruptQ, CorruptSP, ReportData, SubscribeQuery, and DecodeData.

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

The adversary $\mathcal{A}$ wins the game, denoted by $\mathrm{Game}^{\text{NP-CCA}}_{\mathsf{PI},\mathcal{A}}(n) = 1$, if all the following conditions hold:

1. $b = b'$.
2. $\{qid_0, qid_1\} \cap \mathcal{CI} = \emptyset$.
3. $\mathcal{A}$ did not query SubscribeQuery with query identity qid_0 or qid_1.

[2] Note that the value of a c_i can be $\perp$.

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

We say that 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:

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

Note that in Definition 6.2 we provide the adversary with the capability to decrypt arbitrary data reports (though not the challenge report in order to exclude trivial attacks). It is for this reason that we call the security notion above node privacy *under chosen-ciphertext attacks*.

As one might expect, we will see later that in order to achieve this strong node privacy notion, a PPPSI instantiation has to employ an encryption scheme withstanding chosen-ciphertext attacks. Not all encryption schemes provide this security level and, in particular, *homomorphic* schemes (which we will later use to enable data aggregation) in principle cannot achieve chosen-ciphertext security. We thus also define a chosen-plaintext variant of node privacy, which is identical to the definition above, except that the adversary is not allowed to access the decryption oracle for data reports.

Definition 6.3 (Node Privacy under Chosen-Plaintext Attacks). Let PI be a PPPSI instantiation and $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ a PPT adversary interacting with PI via the queries defined in Section 6.3 within in the game $\mathrm{Game}^{\text{NP-CPA}}_{\mathsf{PI},\mathcal{A}}(n)$, which is identical to $\mathrm{Game}^{\text{NP-CCA}}_{\mathsf{PI},\mathcal{A}}(n)$ from Definition 6.2, except that $\mathcal{A}$ is not given access to the DecodeData oracle. We say that PI provides *node privacy under chosen-plaintext attacks* (or NP-CPA security) if for all PPT adversaries $\mathcal{A}$ the following advantage function is negligible in n:

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

6.4.2 Query Privacy

By the notion of *query privacy* we formalize the privacy of query strings (i.e., query identities) a querier registers for. We require that a PPPSI instantiation hides the query identity a subscription token was generated for from the service provider as well as mobile nodes and other queriers, even if all of them collude. Thus, we model query privacy as indistinguishability of subscription tokens generated from two query identities freely chosen by the the adversary, while allowing the adversary to corrupt the service provider as well as mobile nodes and queriers for other query identities.

Definition 6.4 (Query Privacy). Let PI be a PPPSI instantiation and $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ a PPT adversary interacting with PI via the queries defined in Section 6.3 within in the following game $\mathrm{Game}^{\text{QP}}_{\mathsf{PI},\mathcal{A}}(n)$:

Setup. $\mathtt{Setup}(1^n)$ is executed and outputs (RAsk, RApk). Furthermore $\mathcal{C}_{SP} := 1$ is set, i.e., the service provider is assumed to be corrupted from the start.

Phase I. $\mathcal{A}_1$ receives RApk and is given access to the oracles CorruptMN, CorruptQ, ReportData, SubscribeQuery, and DecodeData.

Eventually, $\mathcal{A}_1$ stops and outputs a tuple (qid_0, qid_1) containing two challenge query identities qid_0, qid_1.

Challenge. A bit $b \in_R \{0,1\}$ is chosen uniformly at random. $\mathtt{RegisterQ}(\mathsf{RApk}, \mathsf{RAsk}, qid_b)$ is executed and outputs regQ_{qid_b}. Then $\mathtt{SubscribeQuery}(\mathsf{RApk}, \mathsf{regQ}_{qid_b}, qid_b)$ is executed and outputs a subscription token s.

Phase II. $\mathcal{A}_2$ receives RApk and s and is again given access to the oracles CorruptMN, CorruptQ, ReportData, SubscribeQuery, and DecodeData.

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

The adversary $\mathcal{A}$ wins the game, denoted by $\mathrm{Game}^{\text{QP}}_{\mathsf{PI},\mathcal{A}}(n) = 1$, if all the following conditions hold:

1. $b = b'$.
2. $\{qid_0, qid_1\} \cap \mathcal{CI} = \emptyset$.
3. $\mathcal{A}$ did not query ReportData or SubscribeQuery with query identity qid_0 or qid_1.

We say that PI provides *query privacy* if for all PPT adversaries $\mathcal{A}$ the following advantage function is negligible in n:

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

6.4.3 Report Unlinkability

Report unlinkability captures our requirement on PPPSI instantiations to prevent the linkage of two data reports as originating from the same mobile node by any other party, *including* the registration authority. As mobile nodes (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 mobile node registration value used to generate a data report. Hence, we model report unlinkability as indistinguishability of the mobile node registration value used to generate a data report for a query identity-message pair freely chosen by the adversary, while allowing the adversary to corrupt the service provider, any mobile node and querier, as well as the registration authority (the last of course only after the setup phase).

Definition 6.5 (Report Unlinkability). Let PI be a PPPSI instantiation and $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ a PPT adversary interacting with PI via the queries defined in Section 6.3 within in the following game $\text{Game}^{\text{RU}}_{\text{PI},\mathcal{A}}(n)$:

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

Phase I. $\mathcal{A}_1$ receives RApk and is given access to the oracles CorruptMN, CorruptQ, CorruptSP, CorruptRA, ReportData, SubscribeQuery, and DecodeData.

Eventually, $\mathcal{A}_1$ stops and outputs a tuple (qid^*, m^*) containing a challenge query identity qid^* and a message m^*.

Challenge. $\texttt{RegisterMN}(\text{RApk}, \text{RAsk}, qid^*)$ is executed twice, resulting in two registration values $\text{regMN}^0_{qid^*}$ and $\text{regMN}^1_{qid^*}$. A bit $b \in_R \{0,1\}$ is chosen uniformly at random. Then $\texttt{ReportData}(\text{RApk}, \text{regMN}^b_{qid^*}, qid, m)$ is executed and outputs a report c.

Phase II. $\mathcal{A}_2$ receives RApk, $\text{regMN}^0_{qid^*}$, $\text{regMN}^1_{qid^*}$, and c and is again given access to the oracles CorruptMN, CorruptQ, CorruptSP, CorruptRA, ReportData, SubscribeQuery, and DecodeData.

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

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

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

6.5 Insecurity of PEPSI as PPPSI Instantiation

If we now consider the PEPSI scheme proposed by De Cristofaro and Soriente [22, 21] as an instantiation PI_{CS}[3] of PPPSI, it turns out that the collusions discussed in Chapter 5 result in PI_{CS} failing to provide node privacy and query privacy in our model. We sketch the according attacks—which both work pretty similar by corrupting a mobile node—in the following.

Node Privacy: $\mathcal{A}_1$ calls CorruptSP and outputs two arbitrary, but different query identities qid_0 and qid_1, two arbitrary messages m_0 and m_1, and $\mathbf{s} = ()$. $\mathcal{A}_2$ receives $c = (T, c')$, calls $\text{CorruptMN}(qid')$ for some arbitrary qid' with $qid' \notin \{qid_0, qid_1\}$, and receives $\text{regMN}_{qid'} = z$. $\mathcal{A}_2$ 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, i.e., $\text{Adv}^{\text{NP-CCA}}_{\text{PI}_{CS},\mathcal{A}}(n) = \frac{1}{2}$, which is not negligible. (The attack on NP-CPA works identically.)

This attack leverages the collusion of the service provider with some arbitrary mobile node. Note that the collusion of a querier with some arbitrary mobile node also suffices to break node privacy.

Query Privacy: $\mathcal{A}_1$ outputs two arbitrary, but different query identities qid_0 and qid_1. $\mathcal{A}_2$ receives $s = T$, calls $\text{CorruptMN}(qid')$ for some arbitrary qid' with $qid' \notin \{qid_0, qid_1\}$, and receives $\text{regMN}_{qid'} = z$. $\mathcal{A}_2$ 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, i.e., $\text{Adv}^{\text{QP}}_{\text{PI}_{CS},\mathcal{A}}(n) = \frac{1}{2}$ which is not negligible.

This result leads to the question, how our new PPPSI model can be instantiated in order to achieve node privacy, query privacy, and report unlinkability. We give a positive answer to this question in the next chapter.

[3] We do not provide the exact PI_{CS} instantiation here, as the transformation of the PEPSI instantiation by De Cristofaro and Soriente (cf. Section 4.3) into our PPPSI model is straightforward.

7 A Generic Solution

In this chapter we introduce a secure instantiation of our privacy-preserving participatory sensing infrastructure (PPPSI) defined in Chapter 6 that generically bases on an identity-based encryption scheme. We prove that our generic construction achieves node privacy, query privacy, and report unlinkability (as defined in Section 6.4), given that the underlying IBE scheme provides anonymity and indistinguishability of ciphertexts. Subsequently, we present a concrete instantiation of our construction based on the Boneh-Franklin scheme and show that it achieves full privacy in our model and mitigates collusion attacks while preserving an equally low computation, communication, and storage overhead as of the PEPSI scheme. Finally, we briefly discuss options for a secure instantiation of our generic solution in the standard model.

7.1 Generic IBE Instantiation of PPPSI

We introduce a generic PPPSI instantiation PI_{IBE} based on an (arbitrary) identity-based encryption scheme $\mathcal{E}$ and a pseudorandom function f that provides node privacy, query privacy, and report unlinkability, given that $\mathcal{E}$ has anonymous and indistinguishable ciphertexts. The flavor of node privacy depends on the security of $\mathcal{E}$: ANO-IND-ID-CCA security of $\mathcal{E}$ leads to NP-CCA security of PI_{IBE}, ANO-IND-ID-CPA security to NP-CPA security.

The generic PI_{IBE} instantiation is defined as follows.

Definition 7.1 (PI_{IBE} Instantiation). Let $\mathcal{E} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ be an identity-based encryption scheme and $f : \{0,1\}^n \times \{0,1\}^* \to \{0,1\}^n$ a pseudorandom function. The PI_{IBE} instantiation is defined as follows.

$\mathtt{Setup}(1^n)$: Let $(\mathsf{msk}, \mathsf{mpk}) \leftarrow \mathsf{Setup}(1^n)$ and choose $k \in_R \{0,1\}^n$. Output $\mathsf{RAsk} := (\mathsf{msk}, k)$ and $\mathsf{RApk} := \mathsf{mpk}$. The message space $\mathcal{M}$ is the message space of $\mathcal{E}$ and the identity space $\mathcal{I} = \{0,1\}^*$.

$\mathtt{RegisterMN}(\mathsf{RApk}, \mathsf{RAsk}, qid)$: Compute $T_{qid} := f_k(qid)$ and output $\mathsf{regMN}_{qid} := T_{qid}$.

$\mathtt{RegisterQ}(\mathsf{RApk}, \mathsf{RAsk}, qid)$: Let $sk_{qid} \leftarrow \mathsf{Extract}(\mathsf{mpk}, \mathsf{msk}, qid)$ and compute $T_{qid} := f_k(qid)$. Output $\mathsf{regQ}_{qid} := (sk_{qid}, T_{qid})$.

$\mathtt{ReportData}(\mathsf{RApk}, \mathsf{regMN}_{qid}, qid, m)$: Compute $c' := \mathsf{Enc}(\mathsf{mpk}, qid, m)$ and output $c := (T_{qid}, c')$.

$\mathtt{SubscribeQuery}(\mathsf{RApk}, \mathsf{regQ}_{qid}, qid)$: Output $s := T_{qid}$.

$\mathtt{ExecuteQuery}(\mathsf{RApk}, c, s)$: Parse c as (T, c'). If $T = s$ output c, else output $\perp$.

$\mathtt{DecodeData}(\mathsf{RApk}, \mathsf{regQ}_{qid}, qid, c)$: Parse c as (T, c'). Output $m := \mathsf{Dec}(\mathsf{mpk}, sk_{qid}, c')$. ■

The soundness of PI_{IBE} follows directly from the correctness of $\mathcal{E}$. Figure 7.1 depicts the interaction between the parties within the PI_{IBE} instantiation.

7.2 Security Analysis

Our generic PPPSI instantiation PI_{IBE} provides node privacy (where the flavor depends on whether the ciphertexts of the underlying IBE scheme $\mathcal{E}$ are anonymous and indistinguishable under chosen-ciphertext or chosen-plaintext attacks), query privacy, and report unlinkability as proven in the following three theorems.

Theorem 7.2 (Node Privacy of PI_{IBE}). *Let PI_{IBE} be the PPPSI instantiation defined in Definition 7.1 based on an identity-based encryption scheme $\mathcal{E}$ and a pseudorandom function f. If f is a pseudorandom function and $\mathcal{E}$ provides ANO-IND-ID-CCA (resp. ANO-IND-ID-CPA) security, then PI_{IBE} provides node privacy under chosen-ciphertext attacks (resp. node privacy under chosen-plaintext attacks) as defined in Definition 6.2 (resp. Definition 6.3).*

Proof. We prove the theorem in two steps: First, we replace the pseudorandom function f with a real random function and show that this cannot be distinguished by $\mathcal{A}$ if f is pseudorandom. Then we show that an adversary against the instantiation using a real random function can be used to break the ANO-IND-ID-CCA (resp. ANO-IND-ID-CPA) security of the underlying IBE scheme $\mathcal{E}$. We now provide both proof steps in detail for the NP-CCA / ANO-IND-ID-CCA case, the proof for the NP-CPA / ANO-IND-ID-CPA works identical by removing the handling for DecodeData oracle queries.

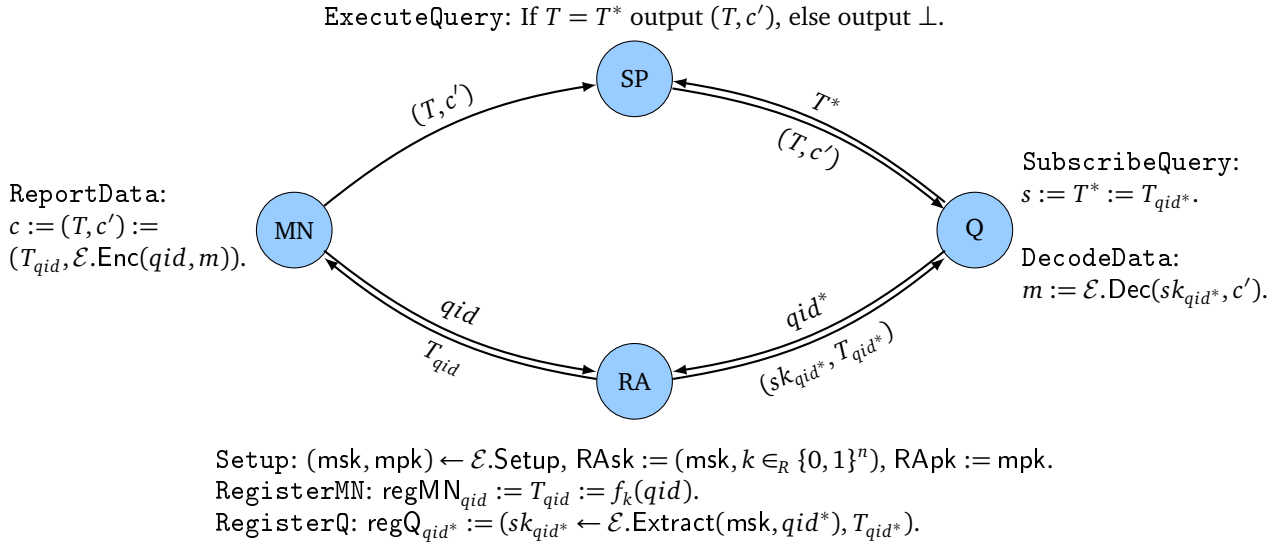

Figure 7.1: Generic PPPSI instantiation PI_{IBE} based on an IBE scheme $\mathcal{E}$ and a pseudorandom function f.

1. Assume we have an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against PI_{IBE} with non-negligible advantage $\mathrm{Adv}^{\mathsf{NP\text{-}CCA}}_{\mathsf{PI}_{IBE},\mathcal{A}}(n)$. Consider the game $\mathrm{Game}^{\mathsf{NP\text{-}CCA}^*}_{\mathsf{PI}_{IBE},\mathcal{A}}(n)$, which is like $\mathrm{Game}^{\mathsf{NP\text{-}CCA}}_{\mathsf{PI}_{IBE},\mathcal{A}}(n)$, except that instead of the pseudorandom function 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} for a query identity qid.

   We argue that $\left|\mathrm{Adv}^{\mathsf{NP\text{-}CCA}}_{\mathsf{PI}_{IBE},\mathcal{A}}(n) - \mathrm{Adv}^{\mathsf{NP\text{-}CCA}^*}_{\mathsf{PI}_{IBE},\mathcal{A}}(n)\right|$ is negligible, as otherwise $\mathcal{A}$ can be used to construct a distinguisher $\mathcal{D}$ between f and g as follows: $\mathcal{D}$ handles everything in $\mathrm{Game}^{\mathsf{NP\text{-}CCA}^*}_{\mathsf{PI}_{IBE},\mathcal{A}}(n)$ like the real challenger, except for evaluations of f, where $\mathcal{D}$ does not choose a key k for f, but instead always asks its own oracle function to compute a value T_{qid}. Observe that, if $\mathcal{D}$ is given oracle access to a pseudorandom function f, then $\mathcal{D}$ acts like the challenger in the game $\mathrm{Game}^{\mathsf{NP\text{-}CCA}}_{\mathsf{PI}_{IBE},\mathcal{A}}(n)$. If however $\mathcal{D}$ is given oracle access to a real random function g, $\mathcal{D}$ acts like the challenger in the game $\mathrm{Game}^{\mathsf{NP\text{-}CCA}^*}_{\mathsf{PI}_{IBE},\mathcal{A}}(n)$. $\mathcal{D}$ outputs the game result (i.e., $b = b'$) as its own guess and thus $\mathrm{Adv}^{\mathsf{PRF}}_{f,\mathcal{D}}(n) = \left|\mathrm{Adv}^{\mathsf{NP\text{-}CCA}}_{\mathsf{PI}_{IBE},\mathcal{A}}(n) - \mathrm{Adv}^{\mathsf{NP\text{-}CCA}^*}_{\mathsf{PI}_{IBE},\mathcal{A}}(n)\right|$. As f is by assumption a pseudorandom function, this is negligible.

2. As, by assumption, $\mathcal{A}$'s advantage $\mathrm{Adv}^{\mathsf{NP\text{-}CCA}}_{\mathsf{PI}_{IBE},\mathcal{A}}(n)$ is non-negligible and we proved $\left|\mathrm{Adv}^{\mathsf{NP\text{-}CCA}}_{\mathsf{PI}_{IBE},\mathcal{A}}(n) - \mathrm{Adv}^{\mathsf{NP\text{-}CCA}^*}_{\mathsf{PI}_{IBE},\mathcal{A}}(n)\right|$ to be negligible, it follows that $\mathcal{A}$'s advantage $\mathrm{Adv}^{\mathsf{NP\text{-}CCA}^*}_{\mathsf{PI}_{IBE},\mathcal{A}}(n)$ in the modified game $\mathrm{Game}^{\mathsf{NP\text{-}CCA}^*}_{\mathsf{PI}_{IBE},\mathcal{A}}(n)$ is non-negligible, too.

   We now construct an adversary $\mathcal{B}$ with non-negligible advantage $\mathrm{Adv}^{\mathsf{IND\text{-}ID\text{-}CCA}}_{\mathcal{E},\mathcal{B}}(n)$ against the ANO-IND-ID-CCA security of $\mathcal{E}$ which makes use of $\mathcal{A}$ in $\mathrm{Game}^{\mathsf{NP\text{-}CCA}^*}_{\mathsf{PI}_{IBE},\mathcal{A}}(n)$ and proceeds as follows.

   Setup. $\mathcal{B}$ receives the master public key mpk in the ANO-IND-ID-CCA game, sets $\mathbf{T} := ()$ to be an empty vector, and $\mathcal{C}_{SP} := 0$.

   Phase I. $\mathcal{B}$ provides $\mathcal{A}_1$ with $\mathsf{RApk} = \mathsf{mpk}$ and answers oracle queries as follows.

   $\mathsf{CorruptMN}(qid)$: If $\mathbf{T}[qid]$ is not set, $\mathcal{B}$ chooses $T_{qid} \in_R \{0,1\}^n$ and sets $\mathbf{T}[qid] := T_{qid}$. $\mathcal{B}$ outputs $\mathsf{regMN}_{qid} := \mathbf{T}[qid]$.

   $\mathsf{CorruptQ}(qid)$: $\mathcal{B}$ uses its $\mathsf{Extract}$ oracle on qid and obtains sk_{qid}. If $\mathbf{T}[qid]$ is not set, $\mathcal{B}$ chooses $T_{qid} \in_R \{0,1\}^n$ and sets $\mathbf{T}[qid] := T_{qid}$. $\mathcal{B}$ outputs $\mathsf{regQ}_{qid} := (sk_{qid}, \mathbf{T}[qid])$.

   $\mathsf{CorruptSP}()$: $\mathcal{B}$ sets $\mathcal{C}_{SP} := 1$.

   $\mathsf{ReportData}(qid, m, \mathbf{s})$: If $\mathbf{T}[qid]$ is not set, $\mathcal{B}$ chooses $T_{qid} \in_R \{0,1\}^n$ and sets $\mathbf{T}[qid] := T_{qid}$. $\mathcal{B}$ computes $c' \leftarrow \mathsf{Enc}(\mathsf{mpk}, qid, m)$ and sets $c := (\mathbf{T}[qid], c')$.

   If $\mathcal{C}_{SP} = 1$, $\mathcal{B}$ gives c to $\mathcal{A}$. Otherwise $\mathcal{B}$ gives $\mathbf{c} := (c_1, \ldots, c_k)$ to $\mathcal{A}$, where it computes c_i as $c_i \leftarrow \mathtt{ExecuteQuery}(\mathsf{RApk}, c, s_i)$ for $i \in \{1, \ldots, k\}$.

   $\mathsf{SubscribeQuery}(qid)$: If $\mathbf{T}[qid]$ is not set, $\mathcal{B}$ chooses $T_{qid} \in_R \{0,1\}^n$ and sets $\mathbf{T}[qid] := T_{qid}$. $\mathcal{B}$ outputs $s := \mathbf{T}[qid]$.

$\mathsf{DecodeData}(qid, c)$: $\mathcal{B}$ parses c as (T, c') and forwards (qid, c') to its Dec oracle. It receives its output m which it returns to $\mathcal{A}$.

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

Challenge. $\mathcal{B}$ forwards $(qid_0, m_0), (qid_1, m_1)$ as its own challenge and receives the challenge ciphertext $c^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, qid_b, m_b)$ for some unknown $b \in \{0,1\}$. $\mathcal{B}$ chooses $T^* \in_R \{0,1\}^n$ and sets $c^* := (T^*, c')$.

If $\mathcal{C}_{SP} = 1$, $\mathcal{B}$ sets $\mathbf{R} := (c)$. Otherwise it sets $\mathbf{R} := (c_1, \ldots, c_k)$, where c_i is computed as $c_i \leftarrow \mathtt{ExecuteQuery}(\mathsf{RApk}, c, s_i)$ for $i \in \{1, \ldots, k\}$.

Phase II. $\mathcal{B}$ provides $\mathcal{A}_2$ with RApk and $\mathbf{R}$ and answers oracle queries like in Phase I.

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

Note that choosing $T_{qid} \in_R \{0,1\}^n$ and remembering the value for a query identity qid is identical to computing $T_{qid} := g(qid)$ for a random function g. Thus $\mathcal{B}$ perfectly simulates the game $\mathsf{Game}^{\mathsf{NP\text{-}CCA}^*}_{\mathsf{PI}_{IBE},\mathcal{A}}(n)$ for $\mathcal{A}$, so $\mathrm{Adv}^{\mathsf{ANO\text{-}IND\text{-}ID\text{-}CCA}}_{\mathcal{E},\mathcal{B}}(n) = \mathrm{Adv}^{\mathsf{NP\text{-}CCA}^*}_{\mathsf{PI}_{IBE},\mathcal{A}}(n)$ which is non-negligible. □

Theorem 7.3 (Query Privacy of PI_{IBE}). *Let PI_{IBE} be the* PPPSI *instantiation defined in Definition 7.1 based on an identity-based encryption scheme $\mathcal{E}$ and a pseudorandom function f. If f is a pseudorandom function, then PI_{IBE} provides query privacy as defined in Definition 6.4.*

Proof. As in the proof of Theorem 7.2 above, we replace the pseudorandom function f with a real random function and show that this cannot be distinguished by $\mathcal{A}$ if f is a pseudorandom function.

Assume we have an adversary $\mathcal{A}$ against PI_{IBE} with non-negligible advantage $\mathrm{Adv}^{\mathsf{QP}}_{\mathsf{PI}_{IBE},\mathcal{A}}(n)$. Consider the game $\mathsf{Game}^{\mathsf{QP}^*}_{\mathsf{PI}_{IBE},\mathcal{A}}(n)$, which is like $\mathsf{Game}^{\mathsf{QP}}_{\mathsf{PI}_{IBE},\mathcal{A}}(n)$, except that instead of the pseudorandom function 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} for a query identity qid. We argue that $\left|\mathrm{Adv}^{\mathsf{QP}}_{\mathsf{PI}_{IBE},\mathcal{A}}(n) - \mathrm{Adv}^{\mathsf{QP}^*}_{\mathsf{PI}_{IBE},\mathcal{A}}(n)\right|$ is negligible, as otherwise $\mathcal{A}$ can be used to construct a distinguisher $\mathcal{D}$ between f and g as provided in the proof of Theorem 7.2.

Now $\mathcal{A}$ receives a challenge subscription token s which was chosen at random. As $\mathcal{A}$ is not allowed to corrupt mobile nodes or queriers registered for qid_0 or qid_1 or query ReportData or SubscribeQuery on qid_0 or qid_1, 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 $\frac{1}{2}$.

It follows that $\mathrm{Adv}^{\mathsf{QP}^*}_{\mathsf{PI}_{IBE},\mathcal{A}}(n) = 0$ and thus $\mathrm{Adv}^{\mathsf{QP}}_{\mathsf{PI}_{IBE},\mathcal{A}}(n)$ is negligible. □

Theorem 7.4 (Report Unlinkability of PI_{IBE}). *The* PPPSI *instantiation PI_{IBE} defined in Definition 7.1, based on an identity-based encryption scheme $\mathcal{E}$ and a pseudorandom function f, provides report unlinkability as defined in Definition 6.5.*

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

7.3 Instantiation Using the Boneh-Franklin IBE Scheme

We now instantiate the generic IBE construction PI_{IBE} with the IBE scheme proposed by Boneh and Franklin [6, 7] (cf. Definition 3.17), resulting in the following PI_{BF} instantiation.

Definition 7.5 (PI_{BF} Instantiation). Let $f : \{0,1\}^n \times \{0,1\}^* \to \{0,1\}^n$ be a pseudorandom function and $\mathcal{G}$ be a bilinear group generator (for a symmetric pairing) as defined in Definition 3.3. The PI_{BF} instantiation is defined as follows.

$\mathtt{Setup}(1^n)$: Run $\mathcal{G}(1^n)$ to obtain a prime q, two groups $\mathbb{G} = \langle g \rangle$ and $\mathbb{G}_T$ of order q, and a bilinear map $e\colon \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Choose $x \in_R \mathbb{Z}_q^*$ and set $y := g^x$. Choose two cryptographic hash functions $H_1\colon \{0,1\}^* \to \mathbb{G}^*$ and $H_2\colon \mathbb{G}_T \to \{0,1\}^\ell$ for some ℓ; both are modeled as random oracles in the security analysis. Set the master public key to be $\mathsf{mpk} = (q, \mathbb{G} = \langle g \rangle, \mathbb{G}_T, e, \ell, y, H_1, H_2)$ and the master secret key to be $\mathsf{msk} = x$.

Choose furthermore $k \in_R \{0,1\}^n$. Output $\mathsf{RAsk} := (\mathsf{msk}, k)$ and $\mathsf{RApk} := \mathsf{mpk}$. The message space is $\mathcal{M} = \{0,1\}^\ell$, the identity space $\mathcal{I} = \{0,1\}^*$.

`RegisterMN`$(\mathsf{RApk}, \mathsf{RAsk}, qid)$: Compute $T_{qid} := f_k(qid)$ and output $\mathsf{regMN}_{qid} := T_{qid}$.

`RegisterQ`$(\mathsf{RApk}, \mathsf{RAsk}, qid)$: Compute $sk_{qid} := H_1(qid)^x$ and $T_{qid} := f_k(qid)$. Output $\mathsf{regQ}_{qid} := (sk_{qid}, T_{qid})$.

`ReportData`$(\mathsf{RApk}, \mathsf{regMN}_{qid}, qid, m)$: Choose $r \in_R \mathbb{Z}_q^*$ and compute $c' = (c_1, c_2) = (g^r, m \oplus H_2(e(H_1(qid), y)^r))$. Output $c := (T_{qid}, c')$.

`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') and c' as (c_1, c_2). Output $m := c_2 \oplus H_2(e(sk_{qid}, c_1))$. ■

Figure 7.2 depicts the interaction between the parties within the PI_{BF} instantiation.

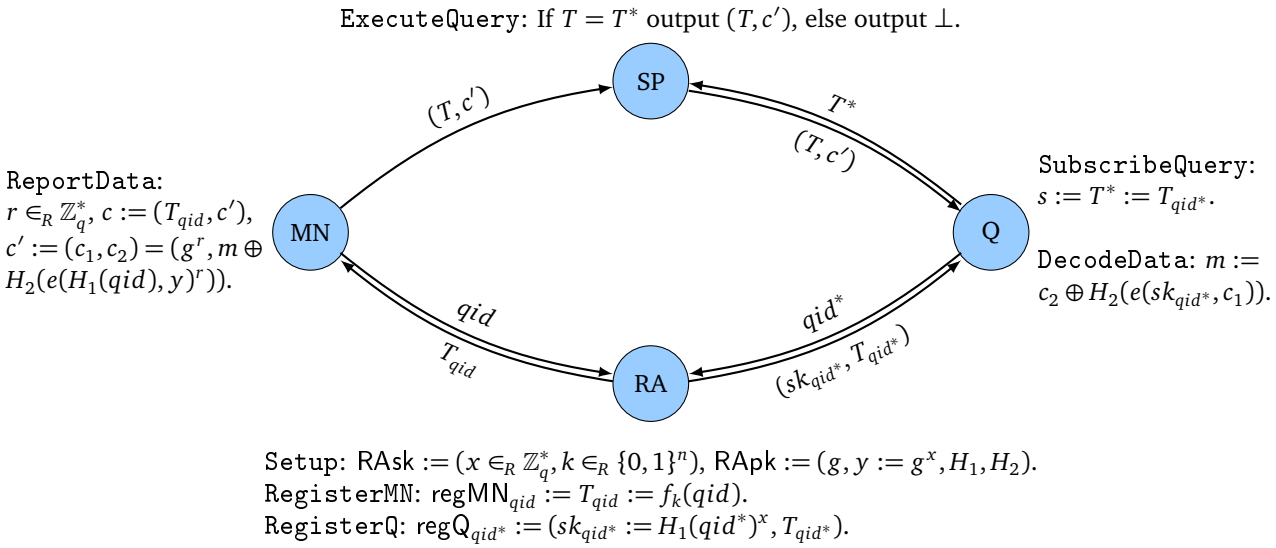

Figure 7.2: PPPSI instantiation based on the Boneh-Franklin IBE scheme and a pseudorandom function f.

7.3.1 Security Analysis

The PI_{BF} construction is a direct instantiation of PI_{IBE} from Definition 7.1 with the IBE scheme proposed by Boneh and Franklin (cf. Definition 3.17). As the latter provides ANO-IND-ID-CPA security (under the BDH assumption in the random oracle model), it directly follows from Theorems 7.2, 7.3, and 7.4, that PI_{BF} provides node privacy under chosen-plaintext attacks, query privacy, and report unlinkability (under the BDH assumption in the random oracle model and given that f is pseudorandom).

7.4 Comparison of PEPSI and the Boneh-Franklin Instantiation PI_{BF}

As already discussed in Section 6.5, PEPSI does not provide node and query privacy in our PPPSI security model due to possible collusions of mobile nodes with the service provider or queriers. Since our PI_{BF} instantiation satisfies those security notions and is—like PEPSI—based on the identity-based encryption scheme proposed by Boneh and Franklin, the obvious question to ask is how both approaches compare wrt. computation and communication overhead as well as key and message sizes. For simplification, we assume a message length of n bits which allows to replace the symmetric en- and decryption in PEPSI with a simple XOR (as in the Boneh-Franklin scheme) and thus eases comparison. Note that in practice the Boneh-Franklin IBE scheme in our PI_{BF} construction could of course be used as a key encapsulation scheme for hybrid encryption, which allows to encrypt messages of arbitrary length as in the original PEPSI scheme.

Table 7.1 shows the computation and communication overhead introduced by both schemes. We see that computationally both schemes are quite similar, the only difference is the need for evaluations of the pseudorandom function by the registration authority in PI_{BF} when registering mobile nodes or queriers (an overhead that can be traded in for additional space when storing the T_{qid} tags in a lookup table). Concerning communication costs, the only practical difference

is in the length of ciphertexts. While ciphertexts in PEPSI have the same length as messages, in PI_{BF} they additionally contain one group element from $\mathbb{G}$. Not mentioned in the table but nevertheless noteworthy is that the PI_{BF} construction does not require an operation like the periodic "nonce renewal", proposed for PEPSI in order to distribute fresh "nonce" values z to mobile nodes and g^z values to queriers, which saves a significant amount of computation and communication resources.

	PEPSI		PPPSI **instantiation** PI_{BF}	
Algorithm	Computation	Communication	Computation	Communication
`Setup`	2E	–	1E	–
`RegisterMN`	–	n	1f	n
`RegisterQ`	1E	2G	1f + 1E	1G + n
`ReportData`	1E + 1P + 2H	2n	2E + 1P + 2H	1G + 2n
`SubscribeQuery`	1P + 1H	n	–	n
`ExecuteQuery`	–	2n	–	1G + 2n
`DecodeData`	1P + 1H	–	1P + 1H	–

E — modular exponentiation in $\mathbb{G}$ or $\mathbb{G}_T$; P — pairing evaluation; H — hash function evaluation; f — PRF evaluation; G — group element in $\mathbb{G}$ or $\mathbb{G}_T$; n — message length and output length of hash and pseudorandom functions

Table 7.1: Comparison of computation and communication overhead between PEPSI and the PI_{BF} instantiation.

Table 7.2 provides a comparison of the space requirements of both schemes, which are again nearly identical in all cases except for the registration authority's secret key RAsk and the data reports c. The use of a pseudorandom function to generate tags in the PI_{BF} construction saves a group element in RAsk. In contrast—as already discussed above—data reports c in PI_{BF} contain an additional group element in the ciphertext part.

Component	**PEPSI**	PPPSI **instantiation** PI_{BF}
RA Public Key RApk	3G + n	3G + n
RA Secret Key RAsk	1G + 2n	2n
Mobile Node Registration Value regMN_{qid}	n	n
Querier Registration Value regQ_{qid}	2G	1G + n
Data Report c	2n	1G + 2n
Subscription Token s	n	n

G — group element in $\mathbb{G}$ or $\mathbb{G}_T$;
n — message length and output length of hash and pseudorandom functions

Table 7.2: Comparison of space requirements between PEPSI and the PI_{BF} instantiation.

In summary we can conclude that the PI_{BF} construction is a secure PPPSI instantiation, providing node privacy, query privacy, and report unlinkability, that performs virtually similar to the PEPSI scheme wrt. computation overhead and key sizes and has only slightly higher communication overhead.

7.4.1 Possible Collusions and Their Impact

Finally, we take a look at the impact of collusions between the involved parties for the PI_{BF} instantiation, presented—similar as for the PEPSI scheme in Section 5.1—in the collusion impact matrix in Table 7.3. We argue that the impact of collusions in the PI_{BF} instantiation is reduced to a minimum for the given scenario. The break of node and query privacy induced by a collusion of the registration authority with queriers or even the service provider is no surprise, as the registration authority—being the key-issuing party—can generate any tag or decryption key desired. The only other (partial) node and query privacy breaks arise when a mobile node or querier colludes with the service provider as the latter then knows the beforehand secret tag for the query identities the mobile node or querier was registered for and can thus easily match incoming data reports and query submissions.[1] We however argue that this partial loss of node resp. query privacy due to collusion of mobile nodes or queriers with the service provider is inevitable in our scenario where

[1] Note that these partial node and query privacy breaks do *not* apply to the formal security games, as a collusion with mobile nodes or queriers registered for the challenge query identity are forbidden in these games.

	Querier	Service Provider	Registration Authority
Mobile Node	∅	[partial NP, partial QP]$_1$	∅
Querier		[partial QP]$_1$	[NP]$_2$
Service Provider			[NP, QP]$_2$

NP — node privacy break; QP — query privacy break; $[\dots]_n$ — mitigated by comment n below
1: Partial loss of node resp. query privacy for colluding mobile node/querier and service provider is inevitable (cf. discussion in the text).
2: The registration authority is trusted and assumed to be non-colluding.

Table 7.3: Collusion impact matrix for the PI_{BF} instantiation.

the service provider shall be able to match data reports with query submissions, as this process—when the query identity of one side is known—necessarily reveals the (same) identity on the matching other side.

7.5 Secure PPPSI Instantiations in the Standard Model

The security of the PI_{BF} construction based on the IBE scheme by Boneh and Franklin introduced and discussed above relies on proofs in the random oracle model. Our proofs of the Theorems 7.2, 7.3, and 7.4 however work in the standard model. We thus can achieve PPPSI security in the standard model by instantiating our *generic* PI_{IBE} scheme with an identity-based encryption scheme providing ANO-IND-ID-CCA (or ANO-IND-ID-CPA) security in the standard model. Possible IBE schemes that allow for such a PPPSI instantiation in the standard model include, e.g., those proposed by Boyen and Waters [9] or Gentry [30]. In this work we however focus on the scheme by Boneh and Franklin [6] (and variants of it), as it is more efficient in practice and provably secure (though only in the random oracle model) under the well-established DBDH assumption (cf. Definition 3.5).

8 Adding Data Aggregation

In order to reduce the communication overhead between the service provider and queriers and increase the privacy of mobile nodes and their data reports, it would be interesting to leverage the (potentially high) computation capabilities of the service provider for an *aggregation* of received data reports (for the same query identity) before forwarding them to subscribed queriers. We implement this idea in this chapter by extending our privacy-preserving participatory sensing infrastructure PPPSI by a suitable aggregation operation for data reports. In the accompanying discussion, it turns out that the established security model can be retained even for the PPPSI model with data aggregation, as it already covers all new security-relevant aspects.

8.1 The PPPSI Model with Data Aggregation

For our extended privacy-preserving participatory sensing infrastructure (PPPSI) model with data aggregation we consider the same interacting parties and operations as in the PPPSI model without data aggregation (cf. Sections 6.1.1 and 6.1.2), however add a "data aggregation" operation, which may be used by the service provider to aggregate received data reports.

Data Aggregation: The data aggregation algorithm, denoted by `AggregateData`, is executed by the service provider on two (or more) data reports and, if both (resp. all) match, outputs a single, aggregated data report.

The infrastructure of PPPSI with data aggregation is illustrated in Figure 8.1.

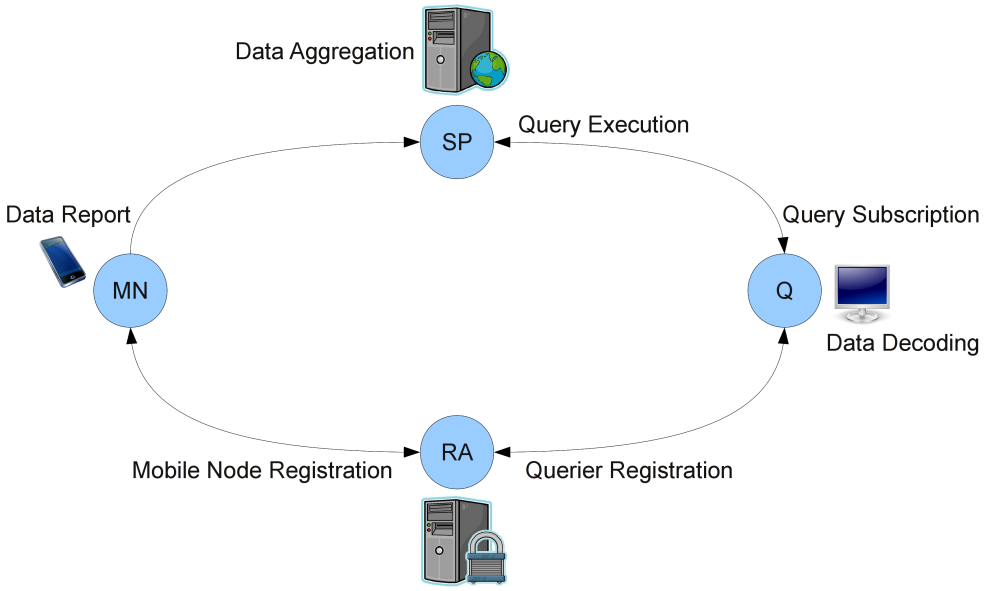

Figure 8.1: PPPSI infrastructure with data aggregation. Mobile nodes (MNs) and queriers (Qs) register to the registration authority (RA). Mobile nodes report data to the service provider (SP), queriers subscribe for reports at the service provider. The service provider may aggregate multiple reports and sends reports matching with subscriptions to the according querier, which decodes them.

We now provide the formal definition of an extended PPPSI instantiation with data aggregation.

Definition 8.1 (PPPSI instantiation with data aggregation). An *instantiation of the privacy-preserving participatory sensing infrastructure with data aggregation* (PPPSI instantiation with data aggregation) Π consists of the eight algorithms `Setup`, `RegisterMN`, `RegisterQ`, `ReportData`, `SubscribeQuery`, `ExecuteQuery`, `DecodeData`, and `AggregateData`, where the first seven are defined as in Definition 6.1 and `AggregateData` is defined as follows.

`AggregateData`$(\mathsf{RApk}, \mathbf{c})$: On input the master public key RApk and a vector of data reports $\mathbf{c} = (c_1, \ldots, c_k)$, this probabilistic algorithm outputs either a single data report c or $\perp$, indicating failure.

To be *sound*, a PPPSI 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. We formalize this soundness condition as follows.

$$
\begin{aligned}
&\forall\, n \in \mathbb{N}, \forall\, (\mathsf{RAsk}, \mathsf{RApk}) \leftarrow \mathtt{Setup}(1^n), \forall k \in \mathbb{N}, \forall\, qid \in \mathcal{I},\\
&\forall\, (\mathsf{regMN}^1_{qid}, \ldots, \mathsf{regMN}^k_{qid}) \leftarrow (\mathtt{RegisterMN}(\mathsf{RApk}, \mathsf{RAsk}, qid), \ldots, \mathtt{RegisterMN}(\mathsf{RApk}, \mathsf{RAsk}, qid)),\\
&\forall\, \mathsf{regQ}_{qid} \leftarrow \mathtt{RegisterQ}(\mathsf{RApk}, \mathsf{RAsk}, qid), \forall\, m_1, \ldots, m_k \in \mathcal{M},\\
&\forall\, (c_1, \ldots, c_k) \leftarrow (\mathtt{ReportData}(\mathsf{RApk}, \mathsf{regMN}^1_{qid}, qid, m_1), \ldots, \mathtt{ReportData}(\mathsf{RApk}, \mathsf{regMN}^k_{qid}, qid, m_k)),\\
&\forall\, s \leftarrow \mathtt{SubscribeQuery}(\mathsf{RApk}, \mathsf{regQ}_{qid}, qid), \forall\, c \leftarrow \mathtt{AggregateData}(\mathsf{RApk}, (c_1, \ldots, c_k)) :\\
&\sum_{i=1}^{k} m_i \in \mathcal{M} \Longrightarrow \mathtt{DecodeData}(\mathsf{RApk}, \mathsf{regQ}_{qid}, qid, \mathtt{ExecuteQuery}(\mathsf{RApk}, c, s)) = \sum_{i=1}^{k} m_i. \qquad \blacksquare
\end{aligned}
$$

Note that, although we fix here (formally in our definition of soundness) the aggregation operation to be the *sum*, i.e., aggregation of data reports should lead to the addition of the contained messages, any other operation could be used instead of the sum. We focus here though on the additive aggregation, as the sum operation is particularly useful in the scenario of participatory sensing. In the common case that the average of multiple sensed values is the desired information, one could think of data reports c containing a counter value, which can be incremented accordingly on aggregation and later be taken into account in the output computation after decoding.

8.2 Adversary Model and Security Definitions

The added `AggregateData` algorithm receives no secret keys as input, so we do not have to extend the original adversary model as defined in Section 6.3 with another oracle. Concerning the security definitions introduced in Section 6.4, one might at first glance think of an extension to the node privacy game from Definitions 6.2 and 6.3, where the adversary does not output two single challenge messages m_0, m_1 but two message vectors $\mathbf{m_0}$, $\mathbf{m_1}$ (of same length) and receives not only the according reports $c_1, \ldots, c_k$ for all messages in $\mathbf{m_b}$, but also the aggregation $\mathtt{AggregateData}(\mathsf{RApk}, (c_1, \ldots, c_k))$. However, as `AggregateData` can be executed by the adversary, this modification boils down to an indistinguishability game for multiple encryptions instead of single encryptions. This difference in turn has no impact on the indistinguishability of ciphertexts[1], so we can also keep our security definitions as introduced in Section 6.4 in the setting with data aggregation.

We however point out that for exactly the same reason—i.e., that `AggregateData` does not receive any secret keys—a PPPSI instantiation providing data aggregation can *never* provide node privacy under chosen-ciphertext attacks. This is because an adversary in the NP-CCA game (cf. Definition 6.2) can always apply the `AggregateData` algorithm on the challenge ciphertext c and the output of `ReportData` for some known m, decode the result to m^* using the `DecodeData` oracle (which is allowed as the aggregated report will differ from c), and restore the challenge message as $m_b = m^* - m$. Thus, node privacy under chosen-plaintext attacks is the strongest node privacy notion for PPPSI instantiations with data aggregation. We will see in the next chapter, that we can indeed achieve this privacy by employing a novel *additively homomorphic* identity-based encryption scheme in our generic PPPSI construction.

[1] Cf. for example the according analysis in Katz and Lindell [37, Section 10.2.2] for the case of public key encryption.

9 Data Aggregation using Additively Homomorphic Encryption

In this chapter we show that our extended privacy-preserving participatory sensing infrastructure (PPPSI) with data aggregation can be instantiated in an efficient and secure manner using *additively homomorphic identity-based encryption*. To this extent, we first extend our generic PPPSI instantiation PI_{IBE} from Chapter 7 to provide data aggregation based on an additively homomorphic IBE scheme and show that the resulting generic construction provides full privacy and security in our model. We then present (to the best of our knowledge) the first additively homomorphic IBE scheme, provably secure under the DBDH assumption, discuss its performance, and show its practical efficiency based on implementation measurements. Afterwards, we provide the PPPSI instantiation based on this scheme, examine its privacy advantages and compare the computation, communication, and storage overhead introduced by our two practical PPPSI instantiations (with and without data aggregation). Finally, we briefly discuss the possibilities for instantiations with data aggregation that are provably secure in the standard model.

9.1 Generic Additively Homomorphic IBE Instantiation of PPPSI with Data Aggregation

We naturally extend our generic instantiation PI_{IBE} of the basic PPPSI model to an instantiation of PPPSI with data aggregation by incorporating an *additively homomorphic* identity-based encryption scheme $\mathcal{E}$ (providing ANO-IND-ID-CPA security) and a pseudorandom function f. The extended instantiation (also denoted by PI_{IBE}) provides node privacy (under chosen-plaintext attacks), query privacy, and report unlinkability and is defined as follows.

Definition 9.1 (PI_{IBE} Instantiation with Data Aggregation). Let $\mathcal{E} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Enc}, \mathsf{Dec})$ be an additively homomorphic identity-based encryption scheme and $f : \{0,1\}^n \times \{0,1\}^* \to \{0,1\}^n$ a pseudorandom function. Let $\circ$ denote the homomorphic operation on two ciphertexts, which leads to an addition of the encrypted messages, i.e., for all identities id and all messages m, m'

$$\mathsf{Dec}(\mathsf{mpk}, sk_{id}, \mathsf{Enc}(\mathsf{mpk}, id, m) \circ \mathsf{Enc}(\mathsf{mpk}, id, m')) = m + m', \qquad \text{where } sk_{id} \leftarrow \mathsf{Extract}(\mathsf{mpk}, \mathsf{msk}, id).$$

We construct the PI_{IBE} instantiation with data aggregation as follows.

The algorithms `Setup`, `RegisterMN`, `RegisterQ`, `ReportData`, `SubscribeQuery`, `ExecuteQuery`, and `DecodeData` are the same as those in Definition 7.1. It remains to define the `AggregateData` algorithm.

`AggregateData`$(\mathsf{RApk}, \mathbf{c})$: Parse $\mathbf{c}$ as $((T_1, c_1), (T_2, c_2), \ldots, (T_\ell, c_\ell))$. If $T_1 = T_2 = \cdots = T_\ell$, then compute $c' = c_1 \circ c_2 \circ \cdots \circ c_\ell$ and output $c = (T_1, c')$, otherwise output $\perp$. ■

The soundness of PI_{IBE} with data aggregation follows directly from the correctness and the additive homomorphism of $\mathcal{E}$. Figure 9.1 depicts the interaction between the parties within the PI_{IBE} instantiation with data aggregation.

9.1.1 Security Analysis

The PI_{IBE} construction above only extends the construction from Definition 7.1 with a realization of the `AggregateData` algorithm. As discussed in Section 8.2, the security model for PPPSI instantiations without data aggregation also applies for instantiations with such. Thus, if the underlying IBE scheme $\mathcal{E}$ provides ANO-IND-ID-CPA security and f is pseudorandom, we can likewise deduce from Theorems 7.2, 7.3, and 7.4 that the extended PI_{IBE} instantiation from Definition 9.1 provides node privacy under chosen-plaintext attacks, query privacy, and report unlinkability.

There is an additional privacy benefit that inherently comes along with aggregation of data reports: as queriers in this case only receive aggregated values (here: sums), not only the identity of mobile nodes but also the exact values of their single measurements are—to some extent—hidden. The privacy provided by the aggregation of course depends on the number of aggregated reports, the kind of measured data, as well as its distribution, but will in any case be higher than without data aggregation.

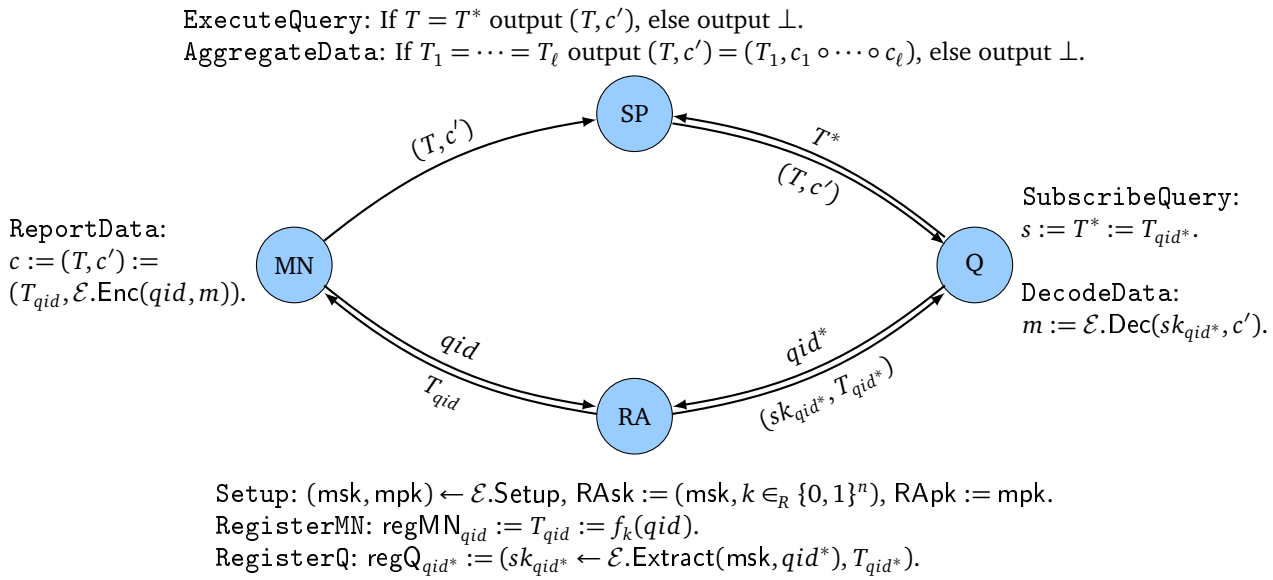

Figure 9.1: Generic PPPSI instantiation with data aggregation PI_{IBE} based on an additively homomorphic IBE scheme $\mathcal{E}$ and a pseudorandom function f.

9.2 The Additively Homomorphic Identity-based Encryption Scheme AIBE

In order to provide an instantiation of the generic PI_{IBE} instantiation with data aggregation we now introduce a novel additively homomorphic identity-based encryption scheme, denoted as AIBE, which we developed as a variation of the IBE scheme of Boneh and Franklin [6] (cf. Definition 3.17). It is—to the best of our knowledge—the first IBE scheme constructed to provide this property and achieves ANO-IND-ID-CPA security under the DBDH assumption in the random oracle model.

Definition 9.2 (Additively Homomorphic IBE Scheme AIBE). Let $\mathcal{G}$ be a bilinear group generator (for a symmetric pairing) as defined in Definition 3.3. The additively homomorphic identity-based encryption scheme AIBE is defined as follows.

$\mathsf{Setup}(1^n)$. Run $\mathcal{G}(1^n)$ to obtain a prime q, two groups $\mathbb{G} = \langle g \rangle$ and $\mathbb{G}_T = \langle \bar{g} \rangle$ of order q (with $\bar{g} = e(g,g)$), and a bilinear map $e\colon \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Choose $x \in_R \mathbb{Z}_q^*$ and set $y := g^x$. Choose a cryptographic hash function $H\colon \{0,1\}^* \to \mathbb{G}^*$ modeled as random oracles in the security analysis.

The message space is $\mathcal{M} = \mathbb{Z}_M = \{0, \ldots, M-1\} \subseteq \mathbb{Z}_q$ with $M = p(n) < q$ for some polynomial p, the ciphertext space is $\mathcal{C} = \mathbb{G}^* \times \mathbb{G}_T$. Output the master public key $\mathsf{mpk} = (q, \mathbb{G} = \langle g \rangle, \mathbb{G}_T = \langle \bar{g} \rangle, e, y, H)$ and the master secret key $\mathsf{msk} = x$.

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

$\mathsf{Enc}(\mathsf{mpk}, id, m)$. Choose $r \in_R \mathbb{Z}_q^*$ and output the ciphertext $c = (c_1, c_2) = (g^r, \bar{g}^m \cdot e(H(id), y)^r)$.

$\mathsf{Dec}(\mathsf{mpk}, sk_{id}, c)$. 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 as $m < M$ is polynomial in n, cf. the performance discussion and analysis in Section 9.2.2 below). ■

The correctness of AIBE follows from the fact that

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

Furthermore, observe that our AIBE scheme is *additively homomorphic* in the message space $\mathcal{M} = \mathbb{Z}_M$, as a pairwise multiplication of two encryptions of m and m' under the same identity id results in the encryption of $m + m' \mod q$:

$$\begin{aligned}
\mathsf{Enc}(\mathsf{mpk}, id, m) \cdot \mathsf{Enc}(\mathsf{mpk}, id, m') &= (g^r, \bar{g}^m \cdot e(H(id), y)^r) \cdot (g^{r'}, \bar{g}^{m'} \cdot e(H(id), y)^{r'}) \\
&= (g^r \cdot g^{r'}, \bar{g}^m \cdot e(H(id), y)^r \cdot \bar{g}^{m'} \cdot e(H(id), y)^{r'}) \\
&= (g^{r+r'}, \bar{g}^{m+m'} \cdot e(H(id), y)^{r+r'}) = \mathsf{Enc}(\mathsf{mpk}, id, m + m' \mod q).
\end{aligned}$$

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 9.2.2 that—though theoretically notable—both constraints are acceptable in many practical scenarios.

9.2.1 Security Analysis

Our AIBE scheme provides ANO-IND-ID-CPA security, which we prove separately for the two aspects of indistinguishability and anonymity in the following theorems.

Theorem 9.3 (IND-ID-CPA Security of AIBE). *Let* AIBE *be the identity-based encryption scheme defined in Definition 9.2. If the DBDH assumption from Definition 3.5 holds for $\mathcal{G}$ and the hash function H is a random oracle, then* AIBE *provides indistinguishability under chosen-plaintext attacks (*IND-ID-CPA *security).*

The proof of Theorem 9.3 works similar to the IND-ID-CPA proof for the Boneh-Franklin scheme (cf. [7, Theorem 4.1]). We similarly first introduce the following public-key version APub of our AIBE scheme.

Definition 9.4 (Additively Homomorphic Public-Key Encryption Scheme APub). Let $\mathcal{G}$ be a bilinear group generator (for a symmetric pairing) as defined in Definition 3.3. The additively homomorphic public-key encryption scheme APub is defined as follows.

$\mathsf{KeyGen}(1^n)$. Run $\mathcal{G}(1^n)$ to obtain a prime q, two groups $\mathbb{G} = \langle g \rangle$ and $\mathbb{G}_T = \langle \bar{g} \rangle$ of order q (with $\bar{g} = e(g,g)$), and a bilinear map $e\colon \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Choose $x \in_R \mathbb{Z}_q^*$ and set $y := g^x$. Pick a random element $h \in_R \mathbb{G}^*$.

The message space is $\mathcal{M} = \mathbb{Z}_M = \{0, \ldots, M-1\} \subseteq \mathbb{Z}_q$ with $M = p(n) < q$ for some polynomial p, the ciphertext space is $\mathcal{C} = \mathbb{G}^* \times \mathbb{G}_T$. Output the public key $pk = (q, \mathbb{G} = \langle g \rangle, \mathbb{G}_T = \langle \bar{g} \rangle, e, y, h)$ and the secret key $sk = h^x$.

$\mathsf{Enc}(pk, m)$. Choose $r \in_R \mathbb{Z}_q^*$ and output the ciphertext $c = (c_1, c_2) = (g^r, \bar{g}^m \cdot e(h,y)^r)$.

$\mathsf{Dec}(sk, c)$. Parse c as (c_1, c_2). Compute $\overline{m} := c_2 / e(sk, c_1)$ and $m = \log_{\bar{g}}(\overline{m})$ as the discrete logarithm to the base $\bar{g}$ of $\overline{m}$ in $\mathbb{G}_T$. ■

The correctness of APub follows similarly as for AIBE by

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

In the first step of the proof we show that an attacker against the IND-ID-CPA security of AIBE can be used to break the IND-CPA security of APub, which essentially means that the Extract oracle does not help an adversary in the IND-ID-CPA game. We then show that the APub scheme is secure under the DBDH assumption. The two steps are proven in the Lemmas 9.5 and 9.6 and finally combined to prove Theorem 9.3.

Lemma 9.5. *Let H be a random oracle and let $\mathcal{A}$ be an adversary with advantage $\mathrm{Adv}^{\text{IND-ID-CPA}}_{\text{AIBE},\mathcal{A}}(n)$ against the* IND-ID-CPA *security of* AIBE *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 $\mathrm{Adv}^{\text{IND-CPA}}_{\text{APub},\mathcal{B}}(n) \geq \frac{1}{e(q_E+1)} \cdot \mathrm{Adv}^{\text{IND-ID-CPA}}_{\text{AIBE},\mathcal{A}}(n)$, 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.[1]

$\mathcal{B}$ 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.

***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 $\mathbb{G}^*$.

[1] Note that this proof works nearly identical to the corresponding proof of Lemma 4.2 in [7].

Extract-**queries.** $\mathcal{B}$ responds on queries of 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 $r' := rx_i^{-1}$)

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

$\mathcal{A}$ continues and might issue H- or 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 IND-ID-CPA game for $\mathcal{A}$, as the responses to H-queries are uniformly and independently distributed in $\mathbb{G}^*$, the Extract-queries are answered correctly, and c' is a proper AIBE encryption of m_b. Thus in this case, $\text{Adv}^{\text{ANO-ID-CPA}}_{\text{APub},\mathcal{B}}(n) \geq \text{Adv}^{\text{ANO-IND-ID-CPA}}_{\text{AIBE},\mathcal{A}}(n)$. 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 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 $\text{Adv}^{\text{IND-CPA}}_{\text{APub},\mathcal{B}}(n) \geq \frac{1}{e(q_E+1)} \cdot \text{Adv}^{\text{IND-ID-CPA}}_{\text{AIBE},\mathcal{A}}(n)$ for $\mathcal{B}$. □

Lemma 9.6. *Let $\mathcal{A}$ be an adversary with advantage $\text{Adv}^{\text{IND-CPA}}_{\text{APub},\mathcal{A}}(n)$ against the IND-CPA security of APub. Then there is an algorithm $\mathcal{B}$ that breaks the DBDH assumption from Definition 3.5 with $\text{Adv}^{\text{DBDH}}_{\mathcal{G},\mathcal{B}}(n) = \frac{1}{2} \cdot \text{Adv}^{\text{IND-CPA}}_{\text{APub},\mathcal{A}}(n)$.*

Proof. We construct adversary $\mathcal{B}$, which interacts with $\mathcal{A}$ in the IND-CPA game as follows.

$\mathcal{B}$ receives elements $(g, q, e, g^{x_1}, g^{x_2}, g^{x_3}, h_b)$ (where $b \in_R \{0,1\}$, $h_0 = e(g,g)^{x_1 x_2 x_3}$ and $h_1 = e(g,g)^w$ for $w \in_R \mathbb{Z}_q$) and has to guess b. It provides $\mathcal{A}$ with the public key $pk = (q, \mathbb{G} = \langle g\rangle, \mathbb{G}_T = \langle \bar{g}\rangle, e, y, h)$, where $\bar{g} := e(g,g)$, $y := g^{x_1}$ and $h := g^{x_2}$. At some point in time, $\mathcal{A}$ 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

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

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

Proof of Theorem 9.3. Combining Lemma 9.5 and 9.6 we can conclude that, if H is a random oracle, an arbitrary adversary $\mathcal{A}$ with advantage $\text{Adv}^{\text{IND-ID-CPA}}_{\text{AIBE},\mathcal{A}}(n)$ can be transformed into an algorithm $\mathcal{B}$ breaking the DBDH assumption with advantage $\text{Adv}^{\text{DBDH}}_{\mathcal{G},\mathcal{B}}(n) \geq \frac{1}{2e(q_E+1)} \cdot \text{Adv}^{\text{IND-ID-CPA}}_{\text{AIBE},\mathcal{A}}(n)$. This proves Theorem 9.3. □

Theorem 9.7 (ANO-ID-CPA Security of AIBE). *Let AIBE be the identity-based encryption scheme defined in Definition 9.2. If the DBDH assumption from Definition 3.5 holds for $\mathcal{G}$ and the hash function H is a random oracle, then AIBE provides anonymity under chosen-plaintext attacks (ANO-ID-CPA security).*

Proof. From Theorem 9.3 we know that AIBE provides IND-ID-CPA security as by assumption the DBDH assumption holds for $\mathcal{G}$ and the hash function H is a random oracle.

Assume now we have an adversary $\mathcal{A}$ against the ANO-ID-CPA security of AIBE with advantage $\text{Adv}^{\text{ANO-ID-CPA}}_{\text{AIBE},\mathcal{A}}(n)$. We construct an adversary $\mathcal{B}$ against the IND-ID-CPA security of AIBE with advantage $\text{Adv}^{\text{IND-ID-CPA}}_{\text{AIBE},\mathcal{B}}(n) = \frac{1}{2} \cdot \text{Adv}^{\text{ANO-ID-CPA}}_{\text{AIBE},\mathcal{A}}(n)$ as follows: $\mathcal{B}$ forwards the received mpk to $\mathcal{A}$ and relays 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 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 the value c_2 in $c = (c_1, c_2)$ 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

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

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

Combining the results from Theorems 9.3 and 9.7, Lemma 3.16 implies that AIBE is ANO-IND-ID-CPA secure.

9.2.2 Performance Discussion and Analysis

While the setup, key extraction, and encryption operations as well as the additive homomorphic combination of ciphertexts base on common and efficient group operations like exponentiations and pairings, the decryption of an AIBE ciphertext takes time polynomial in the size M of the message space, as a discrete logarithm to the base $\bar{g}$ in $\mathbb{G}_T$ has to be computed. This can be done using the brute-force method (i.e., iterating i over $0, \ldots, M-1$ and checking whether $\bar{m} = \bar{g}^i$) which requires on average $M/2$ multiplications in $\mathbb{G}_T$. A second approach is to use Pollard's kangaroo method [45] (also known as Pollard's lambda method) to compute discrete logarithms in the interval $[0, M-1]$ in expected time $O(\sqrt{M})$. We implemented both approaches to provide a feeling for the time required to compute discrete logarithms for small exponents. The results are illustrated in detail in the following subsection. As a third option, one can even achieve constant decryption time if a polynomial-size lookup table with stored powers of $\bar{g}$ is used. The time required to compute such a lookup table for the interval $[0, M-1]$ equals the time a brute-force run over the same interval takes. We thus do not evaluate this variant further in this work.

In any case, the need of solving a discrete logarithm is the reason why AIBE is restricted to message spaces of size polynomial in n, in order to guarantee an (at most) polynomial decryption time. Note that this restriction of the message space is typical for additively homomorphic encryption schemes based on the Decisional Diffie-Hellman or the DBDH assumption, as messages are encrypted in the exponents in this setting. Examples for such schemes are the exponential ElGamal encryption scheme (where, in contrast to the original version [27], messages are encrypted in the exponent as $\mathsf{Enc}(m) = (g^r, g^m \cdot h^r)$) used, e.g., in electronic voting schemes [20], the homomorphic scheme by Boneh, Goh, and Nissim [8], or the encryption scheme incorporating secret sharing proposed by Shi et al. [50].

Computing Discrete Logarithms for Small Exponents

In order to obtain an impression on how much time is needed to compute a discrete logarithm in the group $\mathbb{G}_T$ of an element $h = g^x$ for small x, we implemented both the brute-force and Pollard's kangaroo method[2] to determine x for a given $h = g^x \in \mathbb{G}_T$. In the brute-force approach, all elements g^0, g^1, …, g^M in a range $[0, M]$ are computed and checked against h. As each element can be obtained by multiplying the previous one with g (starting with $1 \in \mathbb{G}_T$), this approach requires M group multiplications in $\mathbb{G}_T$ to scan the whole message space $\mathbb{Z}_M$ and on average $M/2$ to find the exponent for a random message. Pollard's kangaroo method [45] is a more sophisticated approach, which tries to let two sequences of group elements collide—one starting from an element with known exponent, the other starting from the element h—in order to compute the discrete logarithm of h. This probabilistic algorithm has an asymptotic runtime of $O(\sqrt{M})$ for the interval $[0, M]$, its success rate of approx. $1 - e^{-\theta}$ for $\theta = 1, 2, 3, \ldots$ depends on the (selectable) average distance between the group elements in the two sequences which defines θ.

We implemented both approaches using the Pairing-Based Cryptography (PBC) library [39] (in version 0.5.12) with a symmetric type-a pairing[3]. For Pollard's kangaroo method we chose an average hop distance of $\frac{1}{4}\sqrt{M}$, resulting in $\theta = 4$ and thus an estimated success rate of 98%. The measurements were performed on a 2.10GHz Intel(R) Core(TM)2 Duo T8100 CPU with 2GB RAM running Kubuntu 10.04.

Figure 9.2 shows the results of our measurements, namely

1. the average time Pollard's kangaroo method took in interval $[0, M]$,

[2] For simplicity, we implemented the original algorithm proposed by Pollard [45] as we were only interested in the general feasibility of our AIBE scheme. Note that there exists a more efficient *Distinguished Points* variant [54, 42] of the kangaroo method, which is possibly preferable for practical implementations.

[3] The type-a pairing is defined over the elliptic curve $y^2 = x^3 + x$ with 160-bit group order and embedding degree 2.

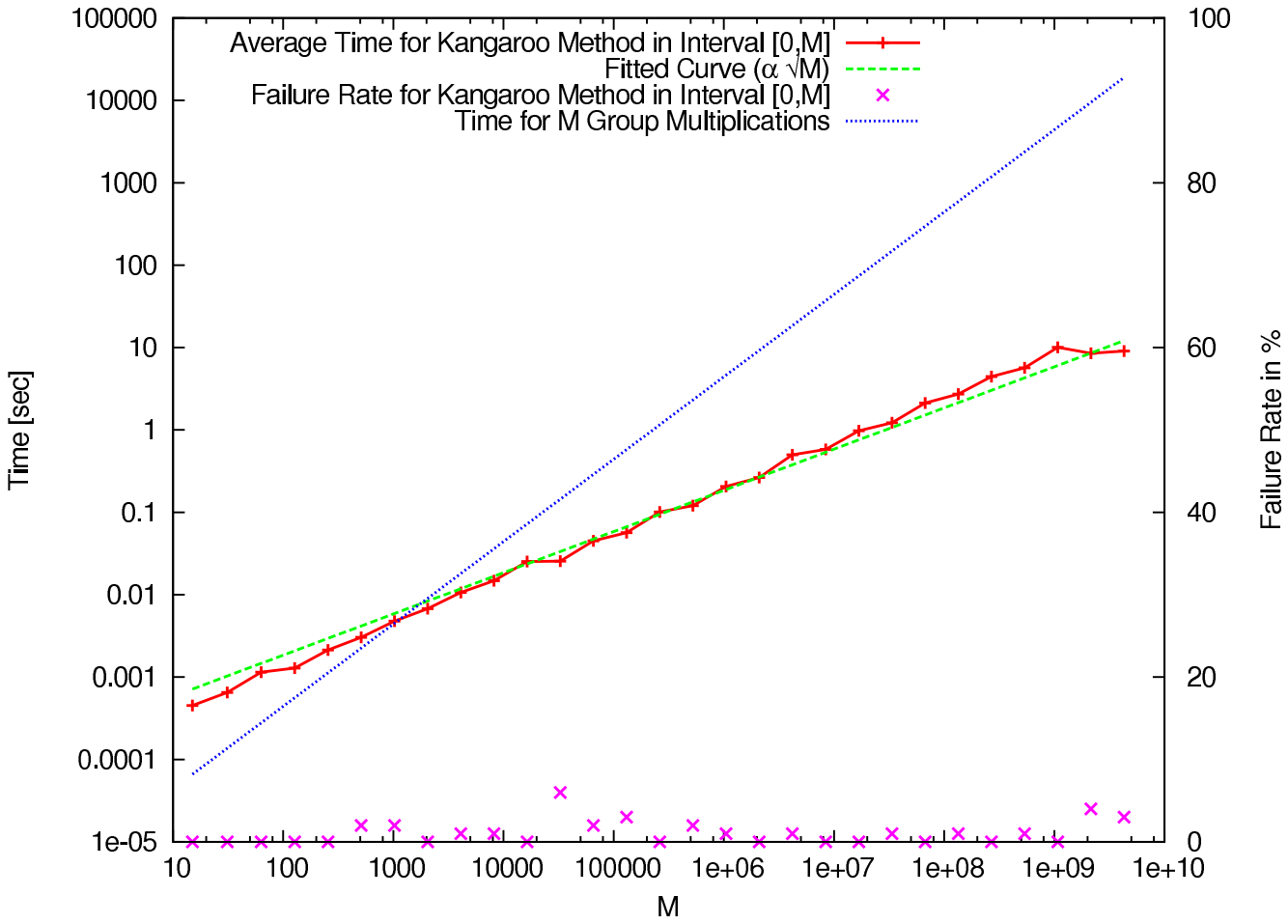

Figure 9.2: Plots (in log-log scale) of the average time Pollard's kangaroo method took in interval $[0, M]$, the fitted curve $\alpha \cdot \sqrt{M}$ for the average coefficient $\alpha = 0.184792$ ms, the failure rate of Pollard's kangaroo method in interval $[0, M]$, and the time for M group multiplications in $\mathbb{G}_T$.

2. the fitted curve $\alpha \cdot \sqrt{M}$ for the average coefficient $\alpha = 0.184792$ ms (remember that Pollard's kangaroo algorithms runs in time $O(\sqrt{M})$),
3. the failure rate of Pollard's kangaroo method in interval $[0, M]$,
4. and the time for M group multiplications in $\mathbb{G}_T$, i.e., the time required for the brute-force approach.

As expected, our implementation of Pollard's kangaroo method runs in time $O(\sqrt{M})$. It requires—as the fitted curve shows—on average about $0.185 \cdot \sqrt{M}$ ms to compute the discrete log for an element with random exponent from the interval $[0, M]$. In contrast, the brute-force approach always requires the same time (namely 4.41017 μs) for one group multiplication and comparison with the target element, which we extrapolated to about $0.004 \cdot M$ ms for M group multiplications in the figure. It is easy to see that Pollard's kangaroo method outperforms the brute-force approach as soon as the message space contains more than about 1,000 elements. Moreover, it remains feasible even for complete 32-bit integer values, i.e., the interval $[0, 2^{32} - 1]$, with on average about 9.084 sec for the computation of a discrete logarithm.

Based on the results obtained in our measurements, we can state that the proposed AIBE scheme is practical for aggregating small values in $\mathbb{Z}_M$ (with M polynomial in n) and even for computations on 32-bit integers.

9.3 PPPSI Instantiation Using the AIBE Scheme

We are now able to instantiate the generic IBE construction with data aggregation PI_{IBE} from Definition 9.1 with the AIBE scheme introduced above. We denote the resulting PPPSI instantiation with data aggregation as PI_{AIBE}.

Definition 9.8 (PI_{AIBE} Instantiation). Let $f : \{0,1\}^n \times \{0,1\}^* \to \{0,1\}^n$ be a pseudorandom function and $\mathcal{G}$ be a bilinear group generator (for a symmetric pairing) as defined in Definition 3.3. The PI_{AIBE} instantiation is defined as follows.

`Setup`(1^n): Run $\mathcal{G}(1^n)$ to obtain a prime q, two groups $\mathbb{G} = \langle g \rangle$ and $\mathbb{G}_T = \langle \bar{g} \rangle$ of order q (with $\bar{g} = e(g,g)$), and a bilinear map $e\colon \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Choose $x \in_R \mathbb{Z}_q^*$ and set $y := g^x$. Choose a cryptographic hash function $H\colon \{0,1\}^* \to \mathbb{G}^*$ modeled as random oracles in the security analysis. Set the master public key to be $\mathsf{mpk} = (q, \mathbb{G} = \langle g \rangle, \mathbb{G}_T = \langle \bar{g} \rangle, e, y, H)$ and the master secret key to be $\mathsf{msk} = x$.

Choose furthermore $k \in_R \{0,1\}^n$. Output $\mathsf{RAsk} := (\mathsf{msk}, k)$ and $\mathsf{RApk} := \mathsf{mpk}$. The message space is $\mathcal{M} = \mathbb{Z}_M = \{0, \ldots, M-1\} \subseteq \mathbb{Z}_q$ with $M = p(n) < q$ for some polynomial p, the identity space is $\mathcal{I} = \{0,1\}^*$.

`RegisterMN`$(\mathsf{RApk}, \mathsf{RAsk}, qid)$: Compute $T_{qid} := f_k(qid)$ and output $\mathsf{regMN}_{qid} := T_{qid}$.

`RegisterQ`$(\mathsf{RApk}, \mathsf{RAsk}, qid)$: Compute $sk_{qid} := H(qid)^x$ and $T_{qid} := f_k(qid)$. Output $\mathsf{regQ}_{qid} := (sk_{qid}, T_{qid})$.

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

`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') and c' as (c_1, c_2). Compute $\overline{m} := c_2 / e(sk_{qid}, c_1)$ and $m = \log_{\bar{g}}(\overline{m})$ as the discrete logarithm to the base $\bar{g}$ of $\overline{m}$ in $\mathbb{G}_T$. Output m.

`AggregateData`$(\mathsf{RApk}, \mathbf{c})$: Parse $\mathbf{c}$ as $((T_1, (c_{1,1}, c_{1,2})), (T_2, (c_{2,1}, c_{2,2})), \ldots, (T_\ell, (c_{\ell,1}, c_{\ell,2})))$. If $T_1 = T_2 = \cdots = T_\ell$, then compute $c' = (c_1, c_2) = \left(\prod_{i=1}^{\ell} c_{i,1}, \prod_{i=1}^{\ell} c_{i,2}\right)$ and output $c = (T_1, c')$, otherwise output $\perp$. ■

The soundness of PI_{AIBE} follows directly from the correctness of AIBE and the fact that encryptions of AIBE are additively homomorphic. Figure 9.3 depicts the interaction between the parties within the PI_{AIBE} instantiation.

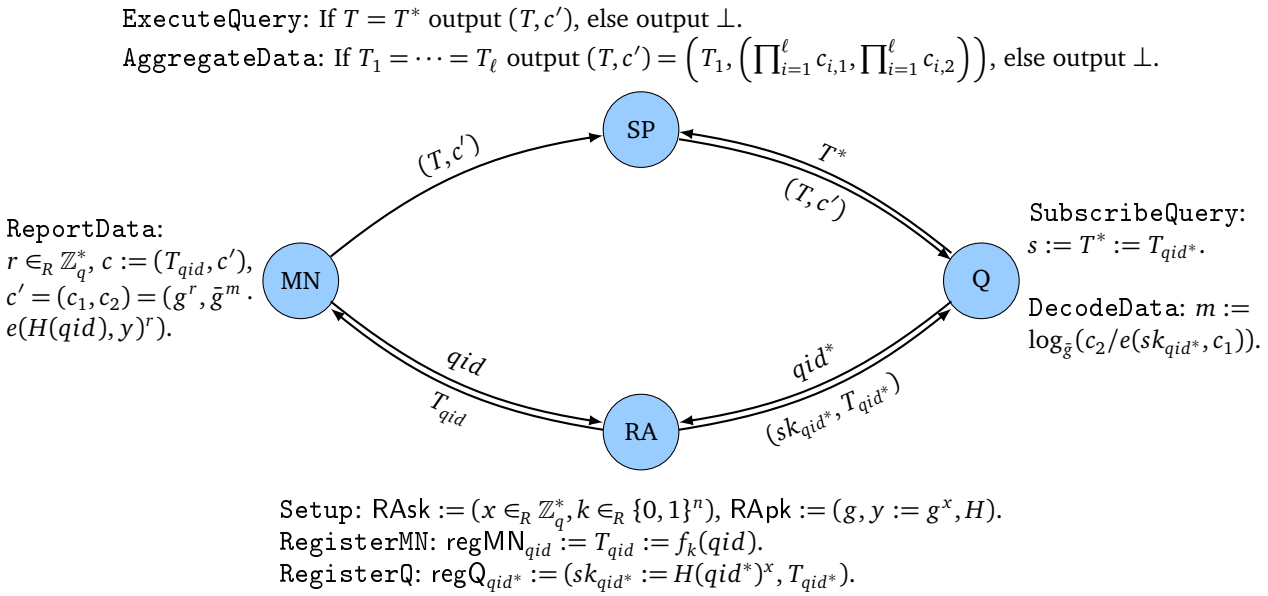

Figure 9.3: PPPSI instantiation with data aggregation PI_{AIBE} based on the AIBE scheme and a pseudorandom function f.

9.3.1 Security Analysis

The PI_{AIBE} construction is a direct instantiation of extended generic scheme PI_{IBE} from Definition 9.1 with our AIBE scheme. As the AIBE scheme provides ANO-IND-ID-CPA security (under the DBDH assumption in the random oracle model), it directly follows from Theorems 7.2, 7.3, and 7.4 that PI_{AIBE} provides node privacy under chosen-plaintext attacks, query privacy, and report unlinkability (in the random oracle model under the DBDH assumption and given that f is pseudorandom).

In addition to the privacy benefits achieved already by the aggregation of reports (as discussed in Section 9.1.1), our PI_{AIBE} instantiation allows for a second, even more powerful privacy mechanism. Consider a (participatory) sensor network with *tree-based routing*, i.e., where mobile nodes do not (all) directly communicate with the service provider, but route their messages along a path (of other mobiles nodes) to it. In such a scenario, common in wireless sensor networks, the `AggregateData` operation of PI_{AIBE} does not has to be restricted to the service provider, but can also be executed by mobile nodes on the path (remember that no secret key is needed for this purpose), thus increasing the privacy of mobiles nodes and data reports vis-à-vis the service provider. Note moreover that this approach is computationally very cheap for mobile nodes in our PI_{AIBE} instantiation, as the aggregation of two data reports (e.g., one own measurement and one received by a child node in the tree) requires only two group multiplications. Though not representable in our PPPSI model, we argue that the security proven in this model also carries over to the depicted scenario with tree-based routing and aggregation on the path to the service provider.

9.4 Comparison of the AIBE Instantiation PI_{AIBE} and the Boneh-Franklin Instantiation PI_{BF}

We now take a closer look on the computation, communication and storage overhead imposed by our PI_{AIBE} instantiation with data aggregation compared to the PI_{BF} instantiation (without data aggregation) based on the Boneh-Franklin scheme (cf. Definition 7.5).

Table 9.1 shows the computation and communication overhead introduced by both schemes. The computation overhead is virtually equal except for the `DecodeData` operation, which—as already discussed above—requires the computation of a discrete logarithm in $\mathbb{G}_T$ in the PI_{AIBE} instantiation and thus imposes a significantly higher computation overhead for this operation. In contrast however, computation power for the decryption of reports can be saved if data reports are aggregated by the service provider (which is cheap as it requires only $2(\ell-1)$ group multiplications to aggregate ℓ reports) before being transmitted to the querier. Indeed, we can provide an estimate for the condition, under which the PI_{AIBE} scheme is *more efficient* than the PI_{BF} instantiation. Remember from our measurements in Section 9.2.2 that the computation of a discrete logarithm requires time $\alpha \cdot \sqrt{M}$ for exponents in the interval $[0, M]$ and some coefficient α (on our system $\alpha = 0.184792$ ms). Denote by β the time required for a pairing evaluation (on the same system we measured $\beta = 5.988357$ ms). Then both schemes perform equally for the transmission of ℓ (aggregatable) data reports if $\alpha \cdot \sqrt{M} = \beta \cdot (\ell - 1)$, as the aggregation replaces $\ell - 1$ decryptions. Thus, on our system, if $M \leq (\beta/\alpha)^2 \cdot (\ell-1) \approx 1050.14 \cdot (\ell-1)$, then the decryption of ℓ aggregated reports with messages in the interval $[0, M]$ in PI_{AIBE} is faster than ℓ decryptions in the PI_{BF} scheme. In other words, if messages of an application are integers between 0 and about $1,000$, then the PI_{AIBE} scheme will *in any case* outperform the PI_{BF} instantiation.

Concerning the communication overhead, both schemes perform similar, except for the communication between service provider and querier, where aggregation allows for huge savings, namely a reduction of the communication overhead by the factor ℓ for each transmission of ℓ aggregated data reports.

	PI_{BF} **instantiation**		PI_{AIBE} **instantiation**	
Algorithm	Computation	Communication	Computation	Communication
`Setup`	1E	–	1E	–
`RegisterMN`	1f	n	1f	n
`RegisterQ`	1f + 1E	1G + n	1f + 1E	1G + n
`ReportData`	2E + 1P + 2H	1G + 2n	3E + 1P + 1H	2G + n
`SubscribeQuery`	–	n	–	n
`ExecuteQuery`	–	1G + 2n	–	2G + n
`DecodeData`	1P + 1H	–	1P + 1DL	–
`AggregateData`	n/a	n/a	≈ 0[4]	–

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 and output length of hash and pseudorandom functions

Table 9.1: Comparison of computation and communication overhead between the PI_{BF} and the PI_{AIBE} instantiation.

Table 9.2 provides a comparison of the space requirements of the PI_{BF} and the PI_{AIBE} instantiations. They only differ in the data report size with two group element and n bits for PI_{AIBE} compared to one group element and $2n$ bits for PI_{BF}, which is negligible in practice.

[4] The `AggregateData` algorithm of PI_{AIBE} requires 2ℓ group multiplications modulo some integer to aggregate ℓ ciphertexts, which is negligible in comparison to the other units used (e.g., group exponentiation, pairings, etc.).

Component	PI_{BF} **instantiation**	PI_{AIBE} **instantiation**
RA Public Key RApk	3G + n	3G + n
RA Secret Key RAsk	2n	2n
Mobile Node Registration Value regMN_{qid}	n	n
Querier Registration Value regQ_{qid}	1G + n	1G + n
Data Report c	1G + 2n	2G + n
Subscription Token s	n	n

G — group element in $\mathbb{G}$ or $\mathbb{G}_T$;
n — message length and output length of hash and pseudorandom functions

Table 9.2: Comparison of space requirements between the PI_{BF} and the PI_{AIBE} instantiation.

We can conclude that the PI_{AIBE} scheme performs faster or equally fast for small messages and provides the same security as the PI_{BF} instantiation. Furthermore, the possible data aggregation in PI_{AIBE} allows for a significant reduction of the communication overhead between service provider and querier and constitutes an additional privacy benefit, as queriers only receive aggregated messages.

9.5 Secure PPPSI Instantiations with Data Aggregation in the Standard Model

Similar to the situation of general PPPSI instantiations in the standard model discussed in Section 7.5, we can achieve a secure PPPSI instantiation with data aggregation using an *additively homomorphic* identity-based encryption scheme that provides ANO-IND-ID-CPA security[5] in the standard model. While no additively homomorphic IBE scheme secure in the standard model has been proposed as such, we can build one based on the scheme proposed by Gentry [30] which—though unnoted in his paper—is multiplicatively homomorphic in $\mathbb{G}_T$. We can leverage this homomorphism by applying a similar approach as for the AIBE scheme, namely to use $g^m \in \mathbb{G}_T$ (for $m \in \mathbb{Z}_M$ with M polynomial in n) instead of $m \in \mathbb{G}_T$. As Gentry's scheme is ANO-IND-ID-CPA-secure in the standard model, this results in an ANO-IND-ID-CPA-secure *additively* homomorphic IBE scheme that can be used to obtain a secure PPPSI instantiation with data aggregation in the standard model. Note however that this scheme is less efficient than our AIBE scheme and can be proven secure—instead under the well-established DBDH assumption—only under the lesser known "decisional augmented bilinear Diffie-Hellman exponent assumption" (cf. [30, Section 2.3]), though in the standard model.

[5] Note that homomorphic encryption scheme can never provide ANO-IND-ID-CCA security, as discussed in Section 8.2.

10 Conclusion and Outlook

Participatory sensing allows for a new paradigm of information collection, however also introduces new privacy challenges concerning the data reporting and retrieving parties involved. Previous approaches failed to provide privacy in a cryptographically provable manner, relied on specific network infrastructures, or suffered from collusion attacks.

In this work, we presented for the first time a rigorous security model for privacy-preserving participatory sensing infrastructures, formalizing the main privacy targets in order to protect the confidentiality of data reports and ensure the anonymity of both data reporters and receivers. We provided both generic and concrete instantiations for our model based on identity-based encryption that achieve full privacy protection and equally high practical performance compared with previous approaches.

Furthermore, we extended our model with a mechanism for private data aggregation and provided a generic instantiation based on additively homomorphic identity-based encryption. We presented a novel identity-based encryption scheme with additive homomorphism secure under the decisional bilinear Diffie-Hellman assumption that achieves practical performance for small message spaces. Applying this new scheme in our generic construction, we obtained a participatory sensing infrastructure that combines provable privacy with efficient data aggregation.

Interesting tasks for future work include the protection of privacy of mobile nodes and queriers with respect to the registration authority during the registration process. Additionally, the construction of an efficient additively homomorphic identity-based encryption scheme with exponential message space remains an open challenge.

Bibliography

[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 *International Cryptology Conference (CRYPTO 2005)*, pages 205–222, 2005.

[2] 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. Cryptology ePrint Archive, Report 2005/254, 2005. `http://eprint.iacr.org/`.

[3] H. Alzaid, E. Foo, and J. G. Nieto. Secure Data Aggregation in Wireless Sensor Network: a survey. In *Australasian Information Security Conference (AISC 2008)*, pages 93–105, 2008.

[4] M. Bellare, A. Boldyreva, A. Desai, and D. Pointcheval. Key-Privacy in Public-Key Encryption. In *International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2001)*, pages 566–582, 2001.

[5] M. Bellare and P. Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In *ACM Conference on Computer and Communications Security (CCS 1993)*, pages 62–73, 1993.

[6] D. Boneh and M. K. Franklin. Identity-Based Encryption from the Weil Pairing. In *International Cryptology Conference (CRYPTO 2001)*, pages 213–229, 2001.

[7] D. Boneh and M. K. Franklin. Identity-Based Encryption from the Weil Pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003.

[8] D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-DNF Formulas on Ciphertexts. In *Theory of Cryptography (TCC 2005)*, pages 325–341, 2005.

[9] X. Boyen and B. Waters. Anonymous Hierarchical Identity-Based Encryption (Without Random Oracles). In *International Cryptology Conference (CRYPTO 2006)*, pages 290–307, 2006.

[10] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava. Participatory Sensing. In *Workshop on World-Sensor-Web (WSW 2006)*, pages 117–134, 2006.

[11] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, and R. A. Peterson. People-Centric Urban Sensing. In *International Workshop on Wireless Internet (WICON 2006)*, 2006.

[12] C. Castelluccia, A. C.-F. Chan, E. Mykletun, and G. Tsudik. Efficient and Provably Secure Aggregation of Encrypted Ddata in Wireless Sensor Networks. *ACM Transactions on Sensor Networks*, 5(3), 2009.

[13] C. Castelluccia, E. Mykletun, and G. Tsudik. Efficient Aggregation of encrypted data in Wireless Sensor Networks. In *International Conference on Mobile and Ubiquitous Systems (MobiQuitous 2005)*, pages 109–117, 2005.

[14] D. Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Communications of the ACM*, 24(2):84–88, 1981.

[15] D. Chaum. Blind Signatures for Untraceable Payments. In *Advances in Cryptology (CRYPTO 1982)*, pages 199–203, 1982.

[16] D. Christin, M. Hollick, and M. Manulis. Security and Privacy Objectives for Sensing Applications in Wireless Community Networks. In *International Conference on Computer Communications and Networks (ICCCN 2010)*, pages 1–6, 2010.

[17] D. Christin, A. Reinhardt, S. S. Kanhere, and M. Hollick. A survey on privacy in mobile participatory sensing applications. *Journal of Systems and Software*, 84(11):1928–1946, 2011.

[18] 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.

[19] C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos. AnonySense: Privacy-Aware People-Centric Sensing. In *International Conference on Mobile Systems, Applications, and Services (MobiSys 2008)*, pages 211–224, 2008.

[20] R. Cramer, R. Gennaro, and B. Schoenmakers. A Secure and Optimally Efficient Multi-Authority Election Scheme. In *International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT 1997)*, pages 103–118, 1997.

[21] E. D. Cristofaro and C. Soriente. PEPSI: Privacy-Enhanced Participatory Sensing Infrastructure (Extended Version). Available online at `http://www.emilianodc.com/PEPSI/pepsi-ext.pdf` (accessed March 2013), 2011.

[22] E. D. Cristofaro and C. Soriente. Short Paper: PEPSI: Privacy-Enhanced Participatory Sensing Infrastructure. In *ACM Conference on Wireless Network Security (WISEC 2011)*, pages 23–28, 2011.

[23] E. D. Cristofaro and C. Soriente. Participatory Privacy: Enabling Privacy in Participatory Sensing. *IEEE Network*, 27(1):32–36, 2013.

[24] 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 (MobiSec 2012)*, pages 93–106, 2012.

[25] Y. Dong, S. S. Kanhere, C. T. Chou, and N. Bulusu. Automatic Collection of Fuel Prices from a Network of Mobile Cameras. In *Distributed Computing in Sensor Systems (DCOSS 2008)*, pages 140–156, 2008.

[26] A. Dua, N. Bulusu, W. chang Feng, and W. Hu. Towards Trustworthy Participatory Sensing. In *Usenix Workshop on Hot Topics in Security (HotSec 2009)*, 2009.

[27] T. Elgamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In *Advances in Cryptology (CRYPTO 1984)*, pages 10–18, 1984.

[28] M. Fischlin, A. Lehmann, T. Ristenpart, T. Shrimpton, M. Stam, and S. Tessaro. Random Oracles with(out) Programmability. In *International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2010)*, pages 303–320, 2010.

[29] R. K. Ganti, N. Pham, Y.-E. Tsai, and T. F. Abdelzaher. PoolView: Stream Privacy for Grassroots Participatory Sensing. In *International Conference on Embedded Networked Sensor Systems (SenSys 2008)*, pages 281–294, 2008.

[30] C. Gentry. Practical Identity-Based Encryption Without Random Oracles. In *International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2006)*, pages 445–464, 2006.

[31] P. Gilbert, L. P. Cox, J. Jung, and D. Wetherall. Toward Trustworthy Mobile Sensing. In *Workshop on Mobile Computing Systems and Applications (HotMobile '10)*, pages 31–36, 2010.

[32] O. Goldreich. *The Foundations of Cryptography - Volume 2, Basic Applications*. Cambridge University Press, 2004.

[33] S. Goldwasser and S. Micali. Probabilistic Encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[34] K. L. Huang, S. S. Kanhere, and W. Hu. Preserving privacy in participatory sensing systems. *Computer Communications*, 33(11):1266–1280, 2010.

[35] 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 *International Conference on Embedded Networked Sensor Systems (SenSys 2006)*, pages 125–138, 2006.

[36] A. Kapadia, D. Kotz, and N. Triandopoulos. Opportunistic Sensing: Security Challenges for the New Paradigm. In *Communication Systems and Networks and Workshops (COMSNETS 2009)*, pages 1–10, 2009.

[37] J. Katz and Y. Lindell. *Introduction to Modern Cryptography*. Chapman and Hall/CRC Press, 2007.

[38] V. Kumar and S. Madria. Secure Data Aggregation in Wireless Sensor Networks. In *Wireless Sensor Network Technologies for the Information Explosion Era*, pages 77–107. 2010.

[39] B. Lynn. Pairing-Based Cryptography (PBC) library. Available at `http://crypto.stanford.edu/pbc/`.

[40] 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.

[41] S. Mathur, T. Jin, N. Kasturirangan, J. Chandrasekaran, W. Xue, M. Gruteser, and W. Trappe. ParkNet: Drive-by Sensing of Road-Side Parking Statistics. In *International Conference on Mobile Systems, Applications, and Services (MobiSys 2010)*, pages 123–136, 2010.

[42] R. Montenegro and P. Tetali. How long does it take to catch a wild kangaroo? In *Symposium on Theory of Computing (STOC 2009)*, pages 553–560, 2009.

[43] S. Özdemir and Y. Xiao. Secure data aggregation in wireless sensor networks: A comprehensive overview. *Computer Networks*, 53(12):2022–2037, 2009.

[44] E. Paulos, R. J. Honicky, and E. Goodman. Sensing Atmosphere. Technical Report 203, Human-Computer Interaction Institute, Carnegie Mellon University, 2007.

[45] J. M. Pollard. Monte Carlo methods for index computation (mod p). *AMS Mathematics of Computation*, 32(143):918–924, 1978.

[46] C. Rackoff and D. R. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In *International Cryptology Conference (CRYPTO 1991)*, pages 433–444, 1991.

[47] 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 *International Conference on Information Processing in Sensor Networks (IPSN 2010)*, pages 105–116, 2010.

[48] 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 *Workshop on Embedded Networked Sensors (EmNets 2007)*, pages 13–17, 2007.

[49] Y. Sang, H. Shen, Y. Inoguchi, Y. Tan, and N. Xiong. Secure Data Aggregation in Wireless Sensor Networks: A Survey. In *International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT 2006)*, pages 315–320, 2006.

[50] E. Shi, T.-H. H. Chan, E. G. Rieffel, R. Chow, and D. Song. Privacy-Preserving Aggregation of Time-Series Data. In *Network and Distributed System Security Symposium (NDSS 2011)*, 2011.

[51] J. Shi, R. Zhang, Y. Liu, and Y. Zhang. PriSense: Privacy-Preserving Data Aggregation in People-Centric Urban Sensing Systems. In *IEEE International Conference on Computer Communications (INFOCOM 2010)*, pages 758–766, 2010.

[52] K. Shilton. Four Billion Little Brothers?: Privacy, mobile phones, and ubiquitous data collection. *Communications of the ACM*, 52(11):48–53, 2009.

[53] L. Sweeney. k-Anonymity: A Model for Protecting Privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

[54] P. C. van Oorschot and M. J. Wiener. Parallel collision search with cryptanalytic applications. *Journal of Cryptology*, 12(1):1–28, 1999.