Adversaries and Countermeasures in Privacy-Enhanced Urban Sensing Systems

Emiliano De Cristofaro[1] and Roberto Di Pietro[2]

[1] Palo Alto Research Center

[2] Università di Roma Tre, Italy

***Abstract*—Today's digital society increasingly relies on the interconnection of heterogenous components, encompassing assorted actors, entities, systems, and a variety of (often mobile) computing devices. Revolutionary computing paradigms, such as *People-Centric Urban Sensing*, have focused on the seamless collection of meaningful data from a large number of devices. The increasing complexity of deployed urban systems and related infrastructures, along with the growing amount of information collected, prompts a number of challenging security and privacy concerns. In this paper, we explore a number of scenarios where nodes of a Urban Sensing system are subject to individual queries. In this setting, multiple users and organizations (e.g. infrastructure operators) co-exist, but they may not trust each other to the full extent. As a result, we address the problem of protecting (i) secrecy of reported data and (ii) confidentiality of query interests from the prying eyes of malicious entities.**

We introduce a realistic network model and study different adversarial models and strategies, distinguishing between *resident* and *non-resident* adversaries, considering both *randomly distributed* and *local* attackers. For each of them, we propose a distributed privacy-preserving technique and evaluate its effectiveness via analysis and simulation. Our techniques are tunable, trading off the level of privacy assurance with a small overhead increase, and independent from the complexity of the underlying infrastructures. We additionally provide a relevant improvement of data reliability and availability, while only relying on standard symmetric cryptography. The practicality of our proposals is demonstrated both analytically and experimentally.

***Index Terms*—Urban Sensing, Querying, Privacy, Security, Complex Systems, Adversarial Models, Wireless Communications.**

I. Introduction

Today's computing ecosystem increasingly relies on the interconnection of heterogenous components, encompassing assorted actors, entities, systems, as well as a variety of computing devices. At the same time, advances in communication technologies have put within reach the ubiquitous and seamless collection of meaningful data from a large number of devices. In this context, innovative computing paradigms, such as *People-Centric Urban Sensing*, have been progressively explored by the research community. Starting with seminal papers by Campbell, et al. [1], and by Burke, et al. [2], Urban Sensing targets the integration of sensors into everyday personal devices and the emerging value recognition of people-centric sensing in urban environments [1].

There are many recent efforts that can be somehow categorized under the Urban Sensing paradigm, including [3–16]. By embedding sensor into wireless-enabled devices (e.g., smartphones), Urban Sensing targets dynamic information about environmental trends, e.g., ambient air quality [4], automobile events [5], urban traffic patterns [6], sound events [9], earthquakes [11], parking availabilities [12], sharing consumer pricing information in offline market [14], etc.

Seeking to realize the full potential of Urban Sensing, researchers are also proposing platforms for application developers [17] and devising business models based on incentive mechanisms for the capitalization on sensed data [18, 19].

However, the increasing complexity of deployed systems, along with the growing amount of meaningful information collected, prompts a number of challenging security and privacy concerns. As information collected by sensing devices is made accessible, and routed through, third-party entities, not only user anonymity, but also data confidentiality as well as privacy of queriers ought to be protected.

Observe that, while in many traditional sensing scenarios the network operator and sensor owners may be the same entity, in Urban Sensing this assumption is quite unrealistic. Multiple users and organizations (e.g. infrastructure operators) often collaborate, yet they may not trust each other to the full extent. These unique features do increase the level of complexity of such systems. Within this scenario, we focus on a specific, fundamental activity, that is a pre-condition for the provisioning of many other services and application: querying individual sensors. In particular, we address two privacy issues: (1) queriers might not be willing to disclose their interests, and (2) sensed data should be protected against unauthorized access.

A. Motivation & Challenges

At a first glance, it may appear that Urban Sensing somehow springs as an evolution of *Wireless Sensor Networks* (WSNs). However, these two computing models present several differences[1], as highlighted by [1]. In Urban Sensing, sensors are often relatively powerful devices (e.g., smartphones), and their resources may be larger than WSN motes. For instance, smartphones' batteries can be easily recharged and bandwidth constraints are not as tight as in WSNs (e.g., minimizing communication to preserve battery life is not as relevant).

[1]Nonetheless, we borrow some WSN terminology for our Urban Sensing context—e.g., we use *sensors* or, interchangeably, *nodes*, to denote sensing devices.

Smartphones are extremely *mobile*, as they leverage the ambulation of their carriers, whereas, WSN sensors are relatively static. Most importantly, in traditional WSNs, the network operator is always assumed to manage and own the motes. On the contrary, this assumption does not fit in complex Urban Sensing scenarios, where sensors participate into gathering and sharing local knowledge. As a result, different entities co-exist in a Urban Sensing system and might not trust each other.

Our work is motivated by a number of privacy-sensitive applications. Consider, for instance, smartphone users measuring environmental data, in the spirit of *participatory urbanism* [4] or pushed by some incentives, such as discounts on their phone bills [18]. Members of the program and external entities may access sensed data and query specific sensors, on demand. However, queriers do not want to reveal which sensor is being interrogated, as this reveals their interests and/or exposes sensitive information to other users, network operator, or eavesdroppers. Even though the identity of queried sensors is hidden, queriers should be prevented from unconditionally querying all sensors, whereas, they should only interrogate those sensors they are entitled to, e.g., based on program's policies.

In order to guarantee query privacy, one could collect all readings at some (external) server and privately query the database server using, for instance, Private Information Retrieval (PIR) techniques [20]. However, it is well-known that the state-of-the-art in PIR has not reached the point of being practical (see [21]). Also, it is not clear how to adapt PIR to our scenario: in PIR, server's database is public and *all* information can be retrieved, whereas, in Urban Sensing the querier should only get data from the sensors she interrogates (and she is entitled to).

B. Contributions

In this paper, we define, and focus on, *query privacy* in Urban Sensing systems. We introduce and analyze different adversarial models and strategies. Specifically, we distinguish between *resident* and *non-resident* adversaries: in the former model, the attacker controls a fraction of the sensors *at all times*, while, in the latter, she is assumed to corrupt sensors *after* they have sensed data. Next, we consider two possible adversarial strategies, depending on whether the adversary is *randomly distributed* or focuses on a specific region of the network—we call the latter *local*.

For each of the introduced models, we propose a corresponding distributed privacy-preserving technique. Our approach employs replication of sensed data, combined with probabilistic algorithms. As a positive effect, replication enhances data reliability and tackles possible disconnections and sensor failures, which are likely to happen in such a mobile environment. Next, our techniques rely on very efficient cryptographic tools and minimize the query preparation overhead. We do not require any specific assumption on the network topology and we minimize bandwidth requirements on the link between network gateway and querier. Indeed, the gateway receives just a single reply for the issued query from the network – this reply is further relayed to the querier. Thorough analysis and simulation results support our claims.

Paper Organization. In next section, we overview required preliminaries, while we describe our techniques to achieve query privacy in the presence of a non-resident adversary in Section III. Next, we turn to the resident adversary model: in Section IV, we present our approach against randomly distributed adversaries, while, in Section V, we detail how to cope with local adversaries. Section VI reports on performed experiments and discusses proposed techniques. After reviewing related work in Section VII, we conclude the paper in Section VIII.

II. Preliminaries

In this section, we overview network model and assumptions, and define privacy properties. We also present an overview of our system as well as strategies that can be adopted by the adversary.

A. Network Model and Assumptions

While there is no universally accepted model for Urban Sensing systems, we use reasonable assumptions leveraged from related work. Specifically, we assume a large-scale system composed of a gateway (as in [1]) and a multitude of nodes resembling a *mesh* network (similar to [22]). Specifically, we consider a Urban Sensing system consisting of n sensors, $\mathcal{S} = \{S_1, \cdots, S_n\}$, and a gateway, $\mathcal{GW}$. In the rest of the paper, we use the terms sensors/nodes, as well as network/system, interchangeably. The network operator is denoted with $\mathcal{OPR}$, while the owner of a sensor S_i – with $\mathcal{OWN}_i$. Finally, S_t denotes the target of a query (issued by $\mathcal{OWN}_t$). In the rest of the paper, without loss of generality, we refer to $\mathcal{OWN}_t$ as just $\mathcal{OWN}$.

Remark that we consider complex scenarios where $\mathcal{OWN}_i$'s and $\mathcal{OPR}$ are distinct entities.

We assume that each sensor S_i is securely initialized by its owner, i.e., $\mathcal{OWN}_i$. Each S_i is preloaded with a pairwise key k_i, shared with $\mathcal{OWN}_i$ only. (As it will become clear in the rest of the paper, only $\mathcal{OWN}_i$ should be able to successfully query sensor S_i). Further, note that the only other information a sensor has about the other nodes in network is the number of them (n).

Moreover, we *optionally* assume that sensors can securely establish pairwise key with other sensors in the network. We denote with $\mathcal{K}(S_i, S_l)$ a symmetric key shared between S_i and S_l. We leverage the state-of-the-art for for key establishment and management in mobile networks, such as [23–25].

Finally, in this paper we assume routing to be just an available primitive. This way, we leave the freedom to select the best appropriate primitive among the number of solutions already present in the literature. For instance, to avoid sending replica message to a specific node (something used in the following just to ease exposition), we could assume that the routing delivers a message sent to a network location to the node closest to this location [26], [27]. We further assume that that message forwarding is not affected by dropping or wormhole attacks (for these kinds of attacks a few solutions can be found in [28], [29]).

Experimental setting. Throughout the paper, all analytical findings are confirmed by experimental simulations. To test different combinations of network parameters and adversarial strategies, we used a simple Matlab simulator. To validate each event, reported simulations results were averaged over 1,000 repetitions.

B. Privacy Objectives

Query Privacy. Let S_t be the target of a query to the Urban Sensing system. We say that *query privacy* is guaranteed if any malicious adversary has just a negligible advantage over a random guess of the *identity* of queried sensor S_t. In other words, only $\mathcal{OWN}_t$ (or any entity authorized by $\mathcal{OWN}_t$) can successfully identify S_t as the target of a query issued to the Urban Sensing system.

We do not consider anonymity of the querier, i.e., hiding the fact that she is querying the network, as standard techniques, such as Tor [30], can be used to build an anonymous channel between $\mathcal{GW}$ and the querier.

Data Privacy. Let S_t be the target of a query to the Urban Sensing system, and $\mathcal{OWN}_t$ the owner of the corresponding sensor. We say that *data privacy* is guaranteed if any malicious adversary has a negligible advantage over a random guess of the *reading* reported by S_t. In other words, only $\mathcal{OWN}_t$ can access S_t's reading. Therefore, data privacy ensures that: (1) any eavesdropper in the network has a negligible probability of learning any reading provided by any sensor, and (2) although the identity of queried sensor is hidden by query privacy guarantees, external parties cannot query arbitrary sensors, but only those they "own", or they are entitled to.

C. Adversarial Capabilities

In the rest of the paper, we denote the adversary with $\mathcal{ADV}$. Our adversarial models will be introduced in Section II-F. Nonetheless, we now discuss our assumptions on $\mathcal{ADV}$'s capabilities, which are common to our models.

$\mathcal{ADV}$ controls a coalition of up to z sensors, with $0 < z < n$. Also, $\mathcal{ADV}$ is assumed to be *honest-but-curious*, i.e., it can: (1) eavesdrop on packets in physical proximity of corrupted sensors, and (2) compromise all information stored on the sensor, i.e., readings and secret keys. $\mathcal{ADV}$ does not interfere with sensors' behavior (i.e., she remains stealthy), while learning compromised sensors' secrets in order to compromise query and data privacy. Indeed, if $\mathcal{ADV}$ limits its activity to reading sensor memory contents, there might be no way to tell if a given sensor has ever been corrupted—should sensor code compromising occur, this action could be detected, for instance, using code attestation techniques, e.g., in [31, 32].

Observe that, in the context of Urban Sensing, sensors are essentially users' smartphones, thus, users themselves can be trying to violate privacy properties discussed above and even collude with each other. The above adversarial model (controlling up to z sensors) characterizes the class of adversaries considered in our applications. Finally, note that we aim at hiding the query target even to the queried sensor, as it might be under the control of the adversary. Indeed, the fact that a given sensor receives a query may leak sensitive information: since not all sensors are queried (which is usually the case to reduce energy and bandwidth congestion), but only the ones that are likely to match the query, the fact that a sensor is being queried reveals that the sensor being close to the event to be sensed is relevant.

We only focus on query and data privacy. Additional well-known issues, such as pairwise key establishment, key management, data integrity, and routing can be addressed by standard solutions. Also, Denial-of-Service (DoS), side-channel, and timing attacks are out of the scope of this paper.

As it will become clear throughout the rest of the paper, note that data privacy is straightforwardly guaranteed by our techniques, since a reading is always encrypted using a pairwise key shared between sensor and its owner, an adversary can violate data privacy only if it compromises the sensor originating the reading (and in that case, there is no way to protect data privacy).

Algorithm 1: Sensing

Executed by Sensor S_i at round j

1 $Senses\ d_i^j$;
2 $E_i^j = Enc_{k_i}(d_i^j)$;
3 $T_i^j = Enc_{k_i}(i, j)$;

Algorithm 2: Query

Assuming the query target is S_t at round j

1 [$\mathcal{OWN}$] Compute $T^* = Enc_{k_t}(t, j)$;
2 [$\mathcal{OWN}$] Send T^* to $\mathcal{GW}$;
3 [$\mathcal{GW}$, **on receiving** T^*]
  Randomly select α sensors, s.t. $\mathcal{Q} = \{q_1, \cdots, q_\alpha\}$;
4 [$\mathcal{GW}$] Forward T^* to sensors in $\mathcal{Q}$;
5 [Each $q_l \in \mathcal{Q}$, **on receiving** T^*]
  If (q_l *has* $T_i^j = T^*$) **Then** Send (E_i^j, T_i^j) to $\mathcal{GW}$;
6 [$\mathcal{GW}$, **on receiving** (E_i^j, T_i^j)]
  Forward (E_i^j) to $\mathcal{OWN}_t$

D. System Overview

We now discuss the three main operations—sensing, dissemination, and query—performed in our Urban Sensing system:

1) ***Sensing:*** At each round, any sensor S_i produces a reading. The reading sensed at round j is denoted with d_i^j. Subsequently, the sensor encrypts the data reading (obtaining E_i^j) and produces a data tag (T_i^j), as presented in Algorithm 1. We assume a secure, low-overhead, encryption scheme, $(Enc_{(\cdot)}(\cdot), Dec_{(\cdot)}(\cdot))$ such as AES [33].
2) ***Dissemination:*** At each round j, each sensor S_i disseminates encrypted reading and data tag produced in Algorithm 1 to β other sensors. We call them ***replica***

sensors and we denote them with the corresponding set $\mathcal{R}_i^j = \{r_{i_1}^j, \cdots, r_{i_\beta}^j\}$. The data dissemination strategy *depends on the adversarial model* and it is detailed in Section III, Section IV, and Section V. At the end of dissemination, each sensor in $\mathcal{R}_i^j$ will store (E_i^j, T_i^j). Optionally, each sensor after dissemination deletes both the reading and the data tag. We call such option the *disseminate-and-delete* mode.

3) ***Query:*** We consider the case where $\mathcal{OWN}_t$ wants to query the sensor S_t as for the reading at round j. Algorithm 2 presents the sequence of steps that are performed. Text in square brackets indicates the entity performing the corresponding operation. First, $\mathcal{OWN}_t$ sends to $\mathcal{GW}$ an encrypted tag corresponding to the sensor (and round) of interest. Then, $\mathcal{GW}$ forwards it to a *random set of* α *sensors* in the network. Finally, each of the α sensors verifies if any of the tags that it stores matches the incoming encrypted tag. In this case, the corresponding encrypted reading is forwarded back to $\mathcal{GW}$ that further relays it to $\mathcal{OWN}_t$.

Query Success. Given the probabilistic query phase (where the α sensors to interrogate are randomly selected), the query could also *fail*, i.e., none of the α sensors queried by $\mathcal{GW}$ actually stores a tag matching T^*. The success probability of the query relies on the values α and β, that introduce a tension in the protocol. On the one hand, the values α and β should be set as high as possible, in order to decrease the probability for a query to fail. On the other hand, high values for α and β would increase the message overhead.

We choose, as an acceptable trade-off, $\alpha = c_1\sqrt{n}$ and $\beta = c_2\sqrt{n}$, where c_1 and c_2 are small constants. Assuming $c_1 = c_2 = c$, the probability to have a successfully query is:

$$Pr(\text{Query Success}) \geq 1 - \frac{1}{e^c} \tag{1}$$

This analytical lower bound has also been validated experimentally through the methodology we introduce in Section II-E.

Corruption. As mentioned in Section II-C, recall that $\mathcal{ADV}$ corrupts z sensors, i.e., eavesdrops all the traffic in their physical proximity, and compromises all their secrets.

E. Notation

In the rest of the paper, we select $c_1 = c_2 = 1$. This choice of parameters leads to at least a 63% success probability for the query. Should the query fail, $\mathcal{GW}$ can issue another query, selecting another random set of sensors. Query failing could be checked associating a timeout to the issued query. In the following, however, we will assume that queries do not fail. This assumption does not affect, in any way, query or data privacy properties, while having just a small effect on performance (on the average, issuing the query two times, each time with a fresh randomly selected set of α sensors, it is enough for the query to succeed).

Table I summarizes terminology and notation used in the paper. Note that when it is clear from the context, we will not use the index specifying the round (j).

Symbol	Meaning
$\mathcal{OPR}$	Network operator
$\mathcal{ADV}$	Adversary attacking privacy goal
n	Total number of sensors
$\mathcal{S}$	Set of n sensors
$S_i \in \mathcal{S}$	Generic sensor
S_t	Target sensor (queried by $\mathcal{OWN}_t$)
$\mathcal{OWN}_i$	Owner of sensor S_i
z	Number of sensors controlled by $\mathcal{ADV}$
$\mathcal{Z}$	Set of z sensors controlled by $\mathcal{ADV}$
k_i	Pairwise symmetric key between $\mathcal{OWN}_i$ and S_i
$\mathcal{K}(S_i, S_l)$	Pairwise symmetric key between S_i and S_l
d_i^j	Data reading of sensor S_i at round j
E_i^j	Encryption of d_i with key k_i
T_i^j	Data tag produced by sensor S_i
T^*	Data tag produced by $\mathcal{OWN}_t$ to query
$\mathcal{Q}$	Set of sensors receiving T^*
$q_l \in \mathcal{Q}$	Generic sensor in $\mathcal{Q}$
α	Size of set $\mathcal{Q}$, i.e. $c_1\sqrt{n}$
$\mathcal{R}_i^j$	Set of replica sensors for sensor S_i at round j
$r_{i_l} \in \mathcal{R}_i^j$	Generic sensor in $\mathcal{R}_i$
β	Size of replica sensor sets, i.e., $c_2\sqrt{n}$

TABLE I: Notation.

F. Adversarial Models and Strategies

Throughout the rest of the paper, we distinguish among the following adversarial models:

- ***Non-Resident Adversary:*** In this adversarial model, we assume that $\mathcal{ADV}$ is not always present in the network but it corrupts sensors after both the sensing phase and the dissemination phase have been completed. Moreover, before the next round of sensing starts, it releases the corrupted sensors in order to maximize its chances to go undetected. Further, note that sensor release implies that $\mathcal{ADV}$ restores the original code run by the sensor. Indeed, if the code executed by a sensor is left altered by $\mathcal{ADV}$ this would be eventually detected [31]. To summarize, the operating cycle can be decomposed into the following sequence: (1) Sensing, (2) Dissemination, (3) Corruption, (4) Query, and (5) Sensor Release. We anticipate that the degree of privacy achieved against this type of $\mathcal{ADV}$ is higher than the one achieved against the resident one. Indeed, the fact that $\mathcal{ADV}$ is not present during the dissemination phase prevents $\mathcal{ADV}$ from acquiring valuable information.
- ***Resident Adversary:*** We consider the case where $\mathcal{ADV}$ is always on the network, i.e., it controls z sensors at all times. The operating cycle of the system is: (1) Sensing, (2) Dissemination, and (3) Query. Compared to the non-resident $\mathcal{ADV}$, two steps are "missing": corruption, since sensors are corrupted just once, and sensor release, as compromised sensors are not released. While privacy is harder to achieve, we show how to trade-off a reduction on the privacy level with a corresponding overhead decrease during dissemination. An efficient *dissemination* strategy is presented in Section V.

Algorithm 3: Dissemination (Non-resident Adversary)

Executed by Sensor S_i

1 Randomly select $\mathcal{R}_i = \{r_{i_1}, \cdots, r_{i_\beta}\}$;
2 $MST_i = MinimumSpanningTree(\mathcal{R}_i)$;
3 Send (E_i, T_i) to each sensor in $\mathcal{R}_i$ according to MST_i;
4 (***Optional***) Delete (d_i, E_i, T_i) ;

Next, we consider the two following strategies, with respect to the way $\mathcal{ADV}$ selects the sensors to compromise.

- ***Randomly Distributed Adversary.*** In this adversarial model, we assume $\mathcal{ADV}$ is randomly distributed over the network, i.e., it controls z randomly selected sensors.
- ***Local Adversary.*** This adversarial model captures an adversary focusing on a specific region of the network. For instance, consider the case where the adversary is physically unable to reach parts of the network. To ease presentation, we will consider a network structured as a grid and will assume $\mathcal{ADV}$ to be located only in one square-shaped sector.

III. Non-Resident Adversaries

We now present our techniques to achieve query privacy, as per definitions in Section II-B. We start by considering *non-resident* adversaries (introduced in Section II-F). We anticipate that our proposed protocol is independent of whether the non-resident adversary is *randomly distributed* or *local*.

A. Dissemination Strategy

In the presence of a *non-resident* adversary, our solution exploits the fact that $\mathcal{ADV}$ is not present in the network during the dissemination phase. If replica sensors are selected at random, and the adversary does not observe the sequence of dissemination messages, than the smartest choice for sensor S_i is to build a Minimum Spanning Tree [34] in order to reach all replica sensors in set $\mathcal{R}_i$. This minimizes the total number of messages needed for the dissemination. In Algorithm 3 we detail the sequence of steps run by sensors during dissemination in the presence of a *non-resident* adversary (either randomly distributed or local).

B. Privacy Analysis

We now analyze the degree of query privacy provided by our technique in the presence of non-resident adversary. Specifically, we quantify the probability that $\mathcal{ADV}$ successfully guesses the sensor that originated the data (i.e., the target sensor). In the rest of the paper, we informally call this event "$\mathcal{ADV}$ breaks query privacy" or "$\mathcal{ADV}$ wins", interchangeably. Note that we do not analyze achieved degree of data confidentiality, since it is assured by the encryption method algorithm adopted, that we assume to be secure. Indeed, if techniques such as encryption may be—and are—employed to protect data privacy (i.e. the content of a message), countermeasures to deceive context privacy (e.g. the sensor that originated the data) are much less straightforward while being equally needed [35].

In the *non-resident* model, after sensor compromise $\mathcal{ADV}$ can partition the whole set of sensors in two subsets. The first set is composed of the sensors she controls and that store T^*. We denote this set with $\mathcal{T}$. Note that $\mathcal{T} = \mathcal{Q} \cap \mathcal{Z}$. We denote with τ the size of $\mathcal{T}$ (i.e., $\tau = |\mathcal{T}|$).

Disseminate-and-delete mode. We now consider the privacy degree assuming the tags and the encrypted reading are immediately deleted by the sensors right after the dissemination phase has taken place, but before compromising. In this model it is interesting to note that if a sensor controlled by $\mathcal{ADV}$ holds the value T^*, then this sensor cannot be the target sensor, due to the deletion strategy. However, apart from this information, $\mathcal{ADV}$ cannot do anything but randomly chose the target sensor among all sensors but those in $\mathcal{T}$. Therefore:

$$Pr(\mathcal{ADV} \text{ wins} \mid |\mathcal{T}| = \tau) = 1/(n - \tau) \tag{2}$$

Hence, leveraging Eq. 2 and recalling that $|\mathcal{R}_t| = \beta$:

$$Pr(\mathcal{ADV} \text{ wins}) = \sum_{i=0}^{\beta} Pr(\mathcal{ADV} \text{ wins } |\tau = i) \cdot Pr(\tau = i) = \\ = \sum_{i=0}^{\beta} \frac{Pr(\tau = i)}{n - i} = \sum_{i=0}^{\beta} \frac{1}{n - i} \cdot \left[\frac{\binom{\beta}{i}\binom{n-\beta}{z-i}}{\binom{n}{z}} \right] \tag{3}$$

We point out that the analysis of the above probability has been validated via simulation, according to the methodology described in Section II-E. Simulation results are plotted in Fig. 1, with respect to different values for n (ranging from 512 to 2048) and z (ranging from 0.5% to 5% of the sensors). We remark that the probability in Eq. 3 only differs from our experimental evaluation by an error of at most 10^{-4} (we do not overlap the plot of Eq. 3 with the plotting of the experimental results to ease figure reading). As the size of the network increases, information available to $\mathcal{ADV}$ to make an educated guess of the query target sensor decreases (ratio of controlled sensors being equal). Thus, we conclude that the degree of privacy provided by our technique scales with the number of sensors in the network. Finally, note that the way $\mathcal{ADV}$ selects the sensors to compromise (i.e., whether $\mathcal{ADV}$ is *randomly distributed* or *local*) does not influence its capability of guessing the query target.

Disseminate-and-delete mode *not* enforced. We now relax the assumption introduced by the *disseminate-and-delete* mode and study the case where sensors do not delete disseminated data.

The $\mathcal{ADV}$ has two possible strategies to guess the target sensor, depending on the size of $\mathcal{T}$.

If $(\tau = 0)$, then the best choice $\mathcal{ADV}$ can perform (to guess the target sensor) is to select randomly the target from the sensors in the set $\mathcal{S} \backslash \mathcal{Z}$, thus: $Pr(\mathcal{ADV} \text{ wins}) = 1/(n - z)$.

Whereas, if $(\tau > 0)$, $\mathcal{ADV}$ has two options: (1) she could select the target sensor from $\mathcal{T}$, thus: $Pr(\mathcal{ADV} \text{ wins}) = 1/\beta$, or (2) she could select the target sensor selecting at random from the set $\mathcal{S} \backslash \mathcal{T}$, thus: $Pr(\mathcal{ADV} \text{ wins}) = 1/(n - z)$.

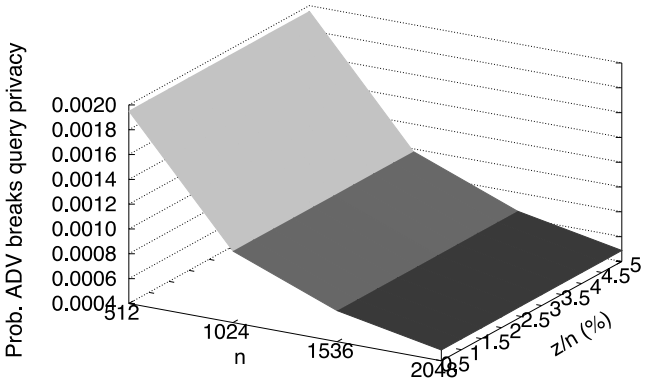

Fig. 1: Probability non-resident $\mathcal{ADV}$ wins in disseminate-and-delete mode.

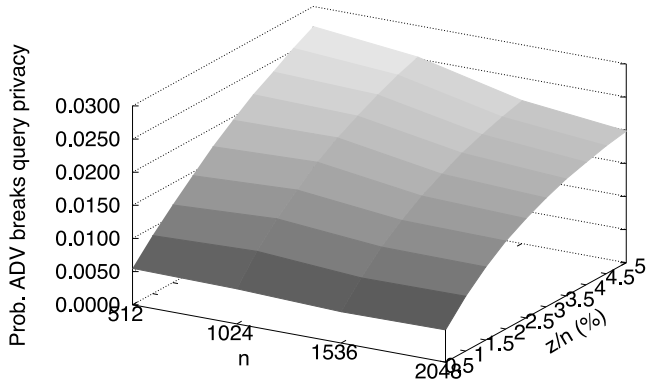

Fig. 2: Probability non-resident $\mathcal{ADV}$ wins when disseminate-and-delete mode is *not* enforced.

Combining the above results, under the rationale hypothesis that $\mathcal{ADV}$ wishes to maximize her winning probability, (assuming $1/(n-z) < 1/\beta$), we obtain:

$$\begin{aligned} Pr(\mathcal{ADV} \text{ wins}) &= Pr(\mathcal{ADV} \text{ wins}|\tau > 0) \cdot Pr(\tau > 0) + \\ &+ Pr(\mathcal{ADV} \text{ wins}|\tau = 0) \cdot Pr(\tau = 0) = \\ &= \frac{1}{\beta} \cdot (1 - Pr(\tau = 0)) + \frac{1}{n-z} \cdot Pr(\tau = 0) = \\ &= \frac{1}{\beta}\left[1 - \frac{\binom{n-\beta}{z}}{\binom{n}{z}}\right] + \frac{1}{n-z}\left[\frac{\binom{n-\beta}{z}}{\binom{n}{z}}\right] \end{aligned} \tag{4}$$

Once again, we have validated the analytical model through simulation. The results of our experimental evaluation are plotted in Fig. 2, with respect to different values of n (ranging from 512 to 2048) and z (ranging from 0.5% to 5% of the sensors). The probability in Eq. 4 only differs from our experimental evaluation by an error of at most 10^{-3}.

Remark. Analyzing the different privacy degrees offered by the two proposed techniques for non-resident adversary, (see Figures 1 and 2), we conclude that the enforcement of the *disseminate-and-delete* mode remarkably reduces the ability of the adversary in correctly guessing the query target. Note that its success probability degrades rapidly for increasing networks (x-axis), while it is only slightly affected by an increasing number of compromised nodes (z-axis). Thus, we recommend its use at all times. In any case, we conclude that the probability of $\mathcal{ADV}$ violating query privacy in both modes is low enough for most realistic applications.

C. Overhead Analysis

We now analyze the overhead (in terms of exchanged messages) resulting from our protocol for the *non-resident* adversarial model.

Sensing. In this phase (Algorithm 1), each sensor acquires a reading dependent on the environment and the physical phenomena of interest. Our protocol only requires sensor to perform two symmetric key encryptions, but the resulting overhead can be considered negligible. Indeed, in our experiments performed on a Nokia N900 smartphone (equipped with a 600MHz CPU), it only took less than $0.1ms$ to encrypt 8-byte (floating-point) numbers using AES with a 128-bit key.

Dissemination. During this phase, a number of dissemination messages are exchanged among sensors. Note that not privacy-preserving querying mechanism do not incur this overhead. Each sensor replicates its sensed data to β other sensors reached through a Minimum Spanning Tree. In order to estimate the resulting message overhead, we consider the following lower and upper bounds. The lower bound is reached if all replica sensors are neighboring sensors, hence the number of messages exchanged is (β). Whereas, as an upper-bound, we consider that all replica sensors are chosen at random and reached in a random order (i.e., *without* constructing the MST). Since on average the distance between two random sensors in a n-sized network is $O(\sqrt{n})$, and $\beta = \sqrt{n}$, the upper bound is $O(\beta \cdot \sqrt{n}) = O(n)$. As a result, the dissemination message overhead for the dissemination presented in Algorithm 3 is bounded as:

$$\beta < \text{Message overhead from Algorithm 3} < O(n)$$

In order to provide a more accurate measure, we decided to simulate the behavior of the dissemination using a Minimum Spanning Tree. Fig. 3 shows that the resulting message overhead can be experimentally upper-bounded by $O(\sqrt{n\sqrt{n}})$. Note that we simulated networks whose sizes range from 512 to 2048. We randomly selected β sensors and count the number of hops needed to reach them all following the Minimum Spanning Tree. To ease simulation, we assumed the network to be represented as a $\sqrt{n} \times \sqrt{n}$ grid, thus making routing algorithm straightforward.

Observe that dissemination imposes a limited computational overhead to each sensor, specifically, related to the computation of the Minimum Spanning Tree. This incurs $O(\beta \log \beta)$, using, for instance, Kruskal's algorithm [34].

Query. During this phase, $\mathcal{GW}$ forwards the encrypted query to α random sensors. These sensors can be reached with a minimum number of messages—again, through a Minimum Spanning Tree. Thus, as $\alpha = \sqrt{n}$, the number of messages exchanged in this phase can be upper bounded by $O(\sqrt{n\sqrt{n}})$. Since the query algorithm (Algorithm 2) is independent of the adversarial model, for all the following protocols presented in the rest of the paper, we will estimate the message overhead due to such algorithm as $O(\sqrt{n\sqrt{n}})$ messages.

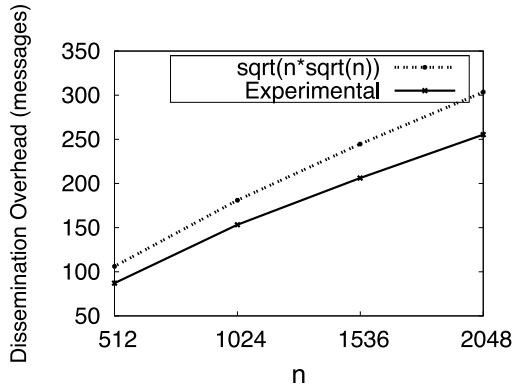

Fig. 3: Message overhead of dissemination in Algorithm 4.

Furthermore, the computational overhead imposed by the query algorithm (Algorithm 2) to involved parties can be considered negligible, as it only involves: (i) forwarding and comparisons ($\mathcal{GW}$, sensors in $\mathcal{Q}$), (ii) symmetric encryption/decryption operations ($\mathcal{OWN}_t$)—recall that it takes less than $1ms$ to encrypt 1 kilobyte via AES on a smartphone.

Total Overhead. In conclusion, we observe that our protocol for non-resident adversaries introduces a total average overhead of $O(\sqrt{n\sqrt{n}})$ messages per sensor. Whereas, the computational overhead incurred by our techniques is negligible, when compared to any non privacy-preserving mechanism.

IV. Randomly Distributed Resident Adversaries

We now consider the more complex case where the adversary is *resident*. As opposed to the setting discussed in Section III, we propose two different solutions depending on whether $\mathcal{ADV}$ is *randomly distributed* or *localized*. We start with the former, i.e., a resident $\mathcal{ADV}$ that selects the sensors to compromise in a random fashion, while we defer to Section V discussion of (resident) local adversaries.

A. Dissemination

For this adversary, we propose the following dissemination technique. Each sensor S_i initializes a counter to β and selects as a destination a random sensor in the network. When this receives the message, it decreases the counter and chooses another random sensor to further disseminate the reading. The dissemination continues until the counter reaches zero. Algorithm 4 presents the details of the dissemination algorithm for the *randomly distributed resident* adversary.

B. Privacy Analysis

$\mathcal{ADV}$ can violate the privacy of the target sensor if it controls the target sensor S_t or at least one of the sensors that route the message to the first sensor in the set of S_t's replica sensors, i.e., $\mathcal{R}_t$. In fact, in the latter case the sensor r_{t_1} learns the values (E_t, T_t, C_t). Now if and only if $C_t = \beta$, then $\mathcal{ADV}$ learns that the tag T_t (matching querier's query T^*) is being originated from the sensor S_t.

Probability that $\mathcal{ADV}$ controls r_{t_1} depends on the type of routing adopted. A higher number of sensors the message is routed by in its first step (on the physical path from S_t to r_{t_1}) results into a higher probability of $\mathcal{ADV}$ intercepting it.

Algorithm 4: Dissemination (Resident Randomly Distributed Adversary)

Executed by Sensor S_i

1 Randomly select sensor r_{i_1};
2 $C_i = \beta$;
3 $k = \mathcal{K}(S_i, r_{i_1})$;
4 $Ct = Enc_k(E_i, T_i, C_i)$;
5 Send Ct to r_{i_1};
6 (***Optional***) Delete (d_i^j, E_i, T_i) ;

Executed by Sensor r_{i_l} receiving Ct from $r_{i_{l-1}}$ (for $(l = 1$ to $\beta)$

6 $k = \mathcal{K}(r_{i_l}, r_{i_{l-1}})$;
7 $(E_i, T_i, C_i) = Dec_k(Ct)$;
8 Store (E_i, T_i);
9 $C_i = C_i - 1$;
10 **If** $(C_i > 0)$ **then** ;
$k = \mathcal{K}(r_{i_l}, r_{i_{l-1}})$;
$Ct = Enc_k(E_i, T_i, C_i)$;
Send Ct to $r_{i_{l+1}}$;

Recalling that β denotes the number of replica sensors, and assuming, on a conservative stance, that $\mathcal{ADV}$ wins if she is able to compromise one of those sensors, we obtain:

$$Pr(\mathcal{ADV}\ wins) = 1 - Pr(\mathcal{ADV}\ \neg wins) = 1 - \left(1 - \frac{\beta}{n}\right)^z \quad (5)$$

The results of our experimental evaluations are plotted in Fig. 4, with respect to different values of n (ranging from 512 to 2048) and z (ranging from 0.5% to 5% of the sensors). They confirm the probability in Eq. 5 with an error of at most 10^{-3}. We observe that, unfortunately, the probability of violating privacy in this scenario grows faster (with increasing number of compromised nodes) than in the non-resident adversary setting. Finding more effective techniques in such an adversarial setting remains an interesting open problem.

C. Overhead Analysis

Sensing. The overhead due to sensing is same as for the *non-resident* adversary, i.e., negligible.

Dissemination. During this phase, each sensor replicates its reading to β random sensors. Being the sensors randomly distributed we point out that, on average, each dissemination hop takes $O(\sqrt{n})$ messages. Hence, the total number of messages needed for the dissemination by each sensor is $O(\beta \cdot \sqrt{n}) = O(n)$. It is an interesting open problem to investigate whether the messages exchanged from the different sensors can be aggregated in order to reduce the overhead.

Query. The overhead for this action is same as for the *non-resident* adversary, i.e., $O(\sqrt{n\sqrt{n}})$.

D. Optimization

The solution proposed in this section incurs a relatively high overhead, i.e., $O(n)$ messages, due to the number of messages

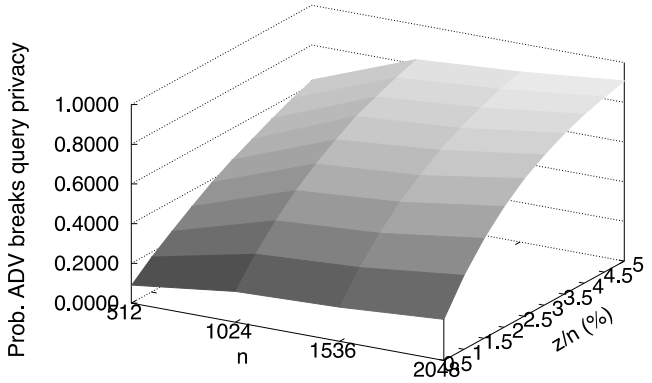

Fig. 4: Probability resident randomly distributed $\mathcal{ADV}$ wins

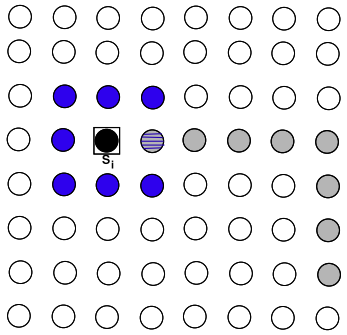

Fig. 5: Dissemination against resident local adversary in a 8x8 network: local chain (gray) and local (blue) diffusion.

exchanged in the dissemination phase, and achieves a lower privacy degree compared to the scenario with a non-resident $\mathcal{ADV}$. Note that the overhead results from the way replica sensors are chosen. As a results, we consider other strategies to select them. In particular, observe that the lower bound for the dissemination phase is $\Omega(\beta)$, i.e., disseminating messages just among the closest β neighboring sensors. This choice is viable and justified by the fact that, when $\mathcal{ADV}$ is resident and randomly distributed, the dissemination strategy does not influence the result of Eq. 5. Hence, choosing a dissemination strategy that minimizes the message overhead would not affect the level of privacy guarantees, while it would remarkably reduce message overhead.

V. Local Resident Adversary

We now consider the case when $\mathcal{ADV}$ is *resident* and *local* to a specific area of the network.

A. Dissemination

To ease exposition, we now consider the network to be deployed as a grid. Sensors are then placed at the intersections of an $\sqrt{n} \times \sqrt{n}$ matrix. We recall that $\mathcal{ADV}$ in this model occupies a squared-shape region of the network. We consider such an adversary to be deployed on a $\sqrt{z} \times \sqrt{z}$ square. Next, we assume that n and z are perfect square. We stress that this topology is assumed only to *ease presentation*, while our techniques are not restricted to it. We consider two different dissemination strategies in order to show the feasibility of the proposed approach.

Local Chain. In this approach, each sensor at dissemination time initializes a counter with β and selects the neighbor at its right. When the latter receives the replica to retain, it decreases the counter and forwards the message to another neighbor at its right. The dissemination stops when the counter reaches zero. Two special cases may occur: (1) when one of the sensors at the right border of the network is reached; we cope with this situation assuming that this sensor continues the dissemination on its bottom neighbor. Thus, it could be possible that a sensor at the bottom-right border is reached; then the dissemination continues to its left neighbor; and, (2) if a sensor at the bottom border of the network is reached, it performs the dissemination on the upper neighbors. This last case accounts for the fact that messages from the bottom border could reach the upper border.

Local diffusion. In this approach, each sensor, S_i, populates the set of replica nodes, $\mathcal{R}_i$, with neighbors, by selecting an inner $\sqrt{z} \times \sqrt{z}$ square in the grid network.

In Fig. 5, we depict the two dissemination strategies for the resident adversary. We depict a small network composed by $8 * 8$ sensors, and each sensor relies on 8 ($\beta = \sqrt{8 * 8} = 8$) replica sensors. The sensors in lighter color are the replica sensors selected with the local chain approach, whereas the darker ones are those selected with the local diffusion approach. (Note that the sensor immediately right of S_i is selected with both approaches). We choose a limited number of sensors only to improve figure's clarity; as shown later, our dissemination techniques also apply, efficiently, to significantly larger networks.

B. Privacy Analysis

We now analyze the privacy provided by the above approaches. We consider the adversary to break the query privacy whenever she controls a sensor storing a tag matching T^* (i.e., $\mathcal{OWN}_t$'s query). Indeed, due to the deterministic routing adopted, $\mathcal{ADV}$ could easily compute the sensor originating the message (i.e., S_t). Informally, $\mathcal{ADV}$ wins if $\mathcal{R}_t \cap \mathcal{Z}$ is not empty. The probability of such an event is different in the two different approaches presented above.

Local Chain. If the dissemination is performed always choosing the right neighbor (we will not consider the special case of a message reaching the border of the deployment area) the compromising probability can be upper bounded as follows. Since $\mathcal{ADV}$ occupies a square area, it is enough to consider where to place the center of such a square for $\mathcal{ADV}$ to win—such a center can be abstracted as a sensor (c_A). Indeed, once c_A is placed, the distribution of other sensors will follow. In particular, let us denote with d the discrete distance of c_A from its border (that is, $d = \sqrt{z}/2$). In the same way, we can consider just the placement of the target sensor (S_t), to derive the distribution of all the other sensors in $\mathcal{R}_t$. Now, since the distribution of sensors in $\mathcal{R}_t$ is a chain, for $\mathcal{ADV}$ to compromise at least a sensor in that set, it is sufficient that the sensor c_A is placed at a distance less than d from any of the sensors in $\mathcal{R}$. This area, is equal to a rectangle of length

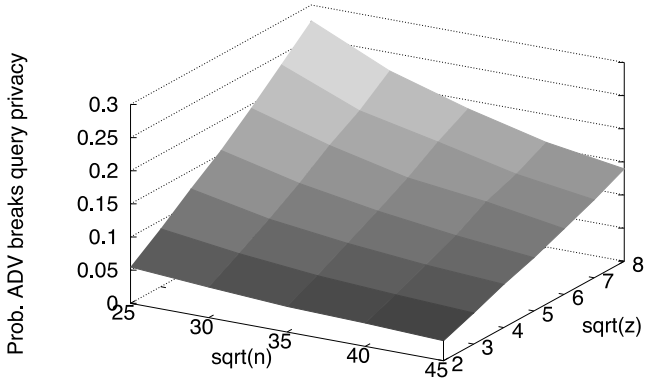

Fig. 6: Probability resident $\mathcal{ADV}$ compromises query privacy for local chain dissemination.

$\beta + 2d$ and height $2d$. Hence, the probability for $\mathcal{ADV}$ to compromise a sensor in $\mathcal{R}_t$ is given by:

$$Pr(\mathcal{ADV} \text{ wins}) < \frac{(\beta + 2d)2d}{n} = \frac{2d\beta + z}{n} \tag{6}$$

The upper bound of the above equation has been validated by experiments as described in Section II-E. Fig. 6 plots the probability of $\mathcal{ADV}$ guessing the target in our experimental results, in terms of the number of sensors in the network (side of the network grid) and sensors controlled by the adversary (side of the adversarial inner square). To ease presentation, we do not plot the analytical values as they tightly upper-bound the empirical probabilities, with a maximum error of 10^{-2}.

Local diffusion. Similarly to the previous technique, we say that $\mathcal{ADV}$ breaks query privacy if she controls a sensor in the square formed by sensors in $\mathcal{R}_t$. With a reasoning similar to the one adopted for chain deployment, the above event happens if the following sufficient condition is verified: $\mathcal{ADV}$ places c_A within a square of side-length $(\sqrt{\beta}+2d)$ that has S_t at the center. The probability of such an event can be upper bounded as follows:

$$Pr(\mathcal{ADV} \text{ wins}) < \frac{(\sqrt{\beta} + 2d)^2}{n} = \frac{\beta + z + 2\sqrt{\beta z}}{n} \tag{7}$$

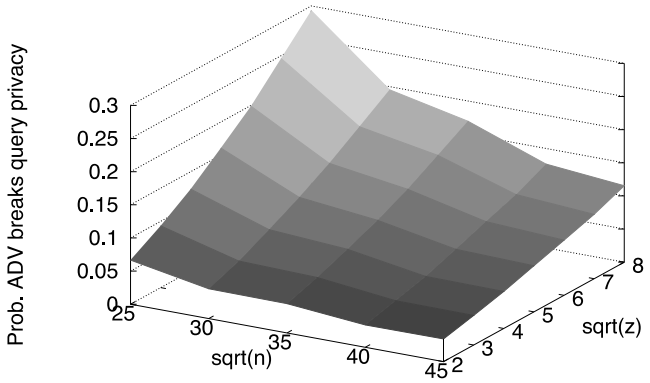

Fig. 7: Probability resident $\mathcal{ADV}$ compromises query privacy for local diffusion dissemination.

The upper bound of the above equation has been validated empirically as described in Section II-E. Fig. 7 plots the probability for $\mathcal{ADV}$ to compromise query privacy in our experimental results, in terms of sensors in the network (side of the network grid) and sensors controlled by the adversary (side of the adversarial inner square). To ease presentation, we do not plot the analytical values as the difference between analytical and experimental results reports, on any plotted point, an absolute error of at most 10^{-2}.

Remark: Note that, since every node has the same probability to be queried, the placement in different squares does not change the dissemination or query technique. If we relax the assumption of contiguous node compromise (done on a conservative stance, as this maximizes the probability of adversary violating query privacy with our dissemination strategies), then it would be interesting to analyze how adversary can increase the probability of violating query privacy.

C. Overhead Analysis

Sensing. Communication and computational overhead due to sensing is same as for the *non-resident* adversary, thus, we do not repeat here its analysis.

Dissemination. During this phase each sensor replicates its reading to β replica sensors. Since these sensors are always selected among neighboring sensors (in both the local chain and local diffusion modes), the total number of messages needed for the dissemination accounts to $O(\beta) = O(\sqrt{n})$ for each sensor. Computational overhead is negligible as it only involves forwarding.

Query. The overhead for this action (Algorithm 2) is same as for the *non-resident* adversary, hence, $O(\sqrt{n\sqrt{n}})$ total messages. Again, computational overhead is negligible, as it only involves forwarding and symmetric key operations.

Total Overhead. We conclude that our protocol for *local* resident adversaries (following both the local chain and the local diffusion approaches) introduces a total average overhead of $O(\sqrt{n})$ messages per sensor, which is smaller than that incurred by our technique for non-resident adversaries. Again, the computational overhead incurred by our techniques is negligible, when compared to any non privacy-preserving mechanism.

VI. Experimental Analysis

In order to assess the feasibility of our techniques, we conducted an experimental analysis on a simulated mesh network (similar to [22]). We measure the number of messages exchanged by each sensor, and the related traffic, for increasing network sizes (averaged over 1000 single queries). As reported in Table II, the induced overall overhead is limited enough for real-world deployment. Even for relatively large urban sensing systems (e.g., involving more than 2000 mobile phones), the traffic introduced by our privacy-enhancing techniques is limited to a few kilobytes – a reasonable overhead in the smartphone setting.

Observe that the degree of privacy provided by techniques against a *non-resident* $\mathcal{ADV}$ is higher than those against a

	Non-Resident Adv.		Resident Rand.Distr. Adv.		Resident Local Adv.	
Network Size	***Messages***	***Traffic***	***Messages***	***Traffic***	***Messages***	***Traffic***
512	194	$6.84KB$	619	$21.78KB$	130	$4.57KB$
1024	434	$11.74KB$	1205	$42.36KB$	213	$7.48KB$
1536	451	$15.86KB$	1781	$62.62KB$	284	$10.00KB$
2048	559	$19.66KB$	2352	$87.70KB$	349	$12.29KB$

TABLE II: Experimental analysis of query message overhead (per sensor).

resident $\mathcal{ADV}$, but they also incur higher message overhead. Also note that, in the *non-resident* adversary model, the way the $\mathcal{ADV}$ is distributed does not influence attained privacy degree. Interestingly, privacy guarantees increase for larger networks, even with the same percentage of corrupted sensors. In the case of *resident* adversaries, we have assumed that $\mathcal{ADV}$ is locally distributed in a specific area of the network.

Note that communication overhead is evenly distributed among sensors, while the communication between the gateway and the querier is kept to the bare minimum – just one message is routed to the $\mathcal{GW}$. Also, to the best of our knowledge, our work is the first to: (1) formalize the problem of protecting query and data privacy in Urban Sensing systems; and, (2) to analyze the aforementioned adversarial models. Finally, remark that all proposed techniques rely on generating replica of the sensed data. These replica, beyond influencing the achieved degree of privacy, have the positive side effect of enhancing data reliability and fault tolerance – sensitive parameters in such a highly mobile environment.

VII. Related Work

Urban Sensing It builds upon the integration of sensors into everyday personal devices and the emerging recognition of the value of people-centric sensing in urban environments [1]. Prominent examples of Urban Sensing projects include [3–16].

A. Security and Privacy in Urban and Participatory Sensing

As it often happens, however, the increasing amount of entailed information prompts a number of security and privacy concerns. Unfortunately, while urban sensing prototypes have multiplied, relatively little attention has been dedicated to security and privacy. Prior work has already highlighted concerns and challenges, but not many practical solutions have been proposed.

In [36], Shilton discusses privacy concerns related to the ubiquitous data collection of urban sensing applications. Next, Kapadia, et al. [37] as well as Christin, Hollick, and Manulis [38] discuss security arising challenges.

Next, recent techniques proposed by [39] and [17, 40] attempt to address privacy-related issues. Their main intuition is to protect anonymity of users, by using Mix Network techniques [41] and achieving k-anonymity and l-diversity [42]. Unfortunately, the degree of confidentiality is relatively limited: reports are encrypted under the public key of so-called *Report Service* (RS) – a trusted party collecting reports and distributing them to queriers. (In other words, the RS may learn both sensors' reports and queriers' interests.)

Authors in [43] argue that privacy can be protected if each user has access to a private server and uses it as a proxy between her and the application targeting user data. The work in [22] focuses on privacy-preserving data aggregation, i.e., computation of sum, average, variance, etc. Similarly, [44] presents a mechanism to compute community statistics of time-series data, while protecting anonymity (using data perturbation in a closed community with a known empirical data distribution). Other proposals, such as [45] and [46], aim at guaranteeing integrity and authenticity of user-generated contents, by employing TPMs.

One common feature of these techniques is that the underlying infrastructure used to collect and deliver reports consists of almost ubiquitous 802.11x access points. Specifically, the work in [39] uses standard MAC-IP address recycling techniques to guarantee user unlinkability between reports (with respect to the access point). However, the assumption of WiFi infrastructures imposes severe limitations on applications' scope in Urban Sensing. Indeed, an ubiquitous presence of open WiFi networks is not realistic today or anticipated in the next future. Another limitation of prior work, such as [39], concerns the use of Mix Networks [41]. Mix Networks serve as anonymizing channels between sensors and report servers: they de-link reports submitted by sensors before they reach the applications. In other words, Mix Networks act as proxies to forward user reports only when some system-defined criteria are met.

The recent PEPSI project [47] introduces a cryptographic model and a provably-secure infrastructure for privacy protection in *participatory sensing*, without relying on Mix Networks.

Finally, note that this paper extends our *preliminary results* on query privacy in Urban Sensing systems, appeared in [48]. Compared to our previous publication, this work provides additional results, in particular, concentrating on newly introduced randomly distributed and resident adversaries. Also, it provides an experimental analysis that measures traffic overhead incurred by each of our proposed techniques.

B. Privacy in Wireless Sensor Network

Somehow related to Urban and Participatory Sensing paradigms are Wireless Sensor Networks (WSNs). In WSNs, spatially distributed autonomous sensors monitor physical or environmental conditions (e.g., temperature, sound, vibration, pressure, motion or pollutants), and cooperatively pass their data through the network to a main location [49]. (For more details on WSNs, refer to [50]).

Besides security issues, a few techniques have explicitly focused on privacy challenges in WSNs. De Cristofaro, Ding, and Tsudik [51] target the problem of privately querying a sensor network in a *sensing-as-service* model (i.e., network operators can offer on-demand access to sensor readings). Their solution combines Onion Routing [52] and k-anonymity techniques [53] to minimize the information leaked to an eavesdropper regarding the identity of the queried sensor. Although it addresses a relatively similar problem, this technique cannot be adapted to the Urban Sensing context, since the network topology is assumed to be static and known in advance to the querier. Also, the querier can interrogate any sensor, at will, i.e., sensor data is publicly available. Finally, note that an overhead of $O(k)$ messages is always generated between the sensor network gateway and the external querier.

Other protocols in [54] and [55] also look at query privacy in WSNs. The former scheme assumes non-collusion of two servers (otherwise privacy is completely lost), whereas, the latter requires the presence of trusted storage nodes.

VIII. Conclusion

In this paper, we have explored query privacy guarantees in Urban Sensing systems. The complexity of these systems is inherited by the presence of multiple actors, such as infrastructure operators, device owners, queriers, and so on. This demands an array of techniques which mask such a complexity while addressing related security and privacy threats. To this end, we have proposed two adversarial models – resident and non-resident – as well as two novel strategies available to an adversary attacking query privacy, depending on whether she is randomly distributed or local to a specific region of the network. For each of these settings, we have presented a probabilistic distributed technique that trades off achieved privacy level with a potential communication overhead.

Also, data confidentiality is guaranteed using inexpensive standard symmetric cryptography. Our techniques have been studied both analytically and experimentally, and—when compared to the threat—they resulted efficient and effective in complex Urban Sensing systems. Finally, simulation results have supported the practicality of our solutions.

As part of future work, we plan to further investigate: (1) whether the gateway, observing many requests/responses, can try to intersect the observed data to infer information about sensors and queriers; (2) more effective and efficient techniques for randomly distributed random adversaries; and, (3) conducting deployment and testing of a real-world privacy-enhanced urban sensing application.

Acknowledgments

We would like to thank the anonymous reviewers for their comments that helped improving the paper, as well as Claudio Soriente, Daniele Perito, and Gene Tsudik for their valuable feedback.

Simulations have been made possible by a Standard HPC 2012 grant from CASPUR. This work was partially supported by the EU FP7 project "Network of Excellence on Engineering Secure Future Internet Software Services and Systems" (NESSOS), FP7-ICT-2009-5 n. 256980. Roberto Di Pietro was partially supported by a Chair of Excellence from University Carlos III, Madrid, Spain.

*Bibliography

[1] A. Campbell, S. Eisenman, N. Lane, E. Miluzzo, and R. Peterson, "People-centric urban sensing," in *WiCon*, 2006.

[2] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. Srivastava, "Participatory Sensing," in *World Sensor Web Workshop*, 2006.

[3] A. Santanche, S. Nath, J. Liu, B. Priyantha, and F. Zhao, "Senseweb: Browsing the physical world in real time," *IPSN*, 2006.

[4] E. Paulos, R. Honicky, and E. Goodman, "Sensing Atmosphere," in *SenSys Workshops*, 2007.

[5] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden, "CarTel: A Distributed Mobile Sensor Computing System," in *SenSys*, 2006.

[6] P. Mohan, V. Padmanabhan, and R. Ramjee, "Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones," in *SenSys*, 2008.

[7] D. Kim, J. Hightower, R. Govindan, and D. Estrin, "Discovering semantically meaningful places from pervasive RF-beacons," in *UbiComp*, 2009.

[8] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda, "Peir, the personal environmental impact report, as a platform for participatory sensing systems research," in *MobiSys*, 2009.

[9] H. Lu, W. Pan, N. Lane, T. Choudhury, and A. Campbell, "SoundSense: Scalable Sound Sensing for People-Centric Applications on Mobile Phones," in *MobiSys*, 2009.

[10] E. Miluzzo, N. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. Eisenman, X. Zheng, and A. Campbell, "Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application," in *SenSys*, 2008.

[11] Stanford University, "Quake-Catcher Network," http://qcn.stanford.edu, 2010.

[12] 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 *MobiSys*, 2010.

[13] B. Longstaff, S. Reddy, and D. Estrin, "Improving activity classification for health applications on mobile devices using active and semi-supervised learning," in *PervasiveHealth*, 2010.

[14] Y. Dong, S. Kanhere, C. Chou, and N. Bulusu, "Automatic collection of fuel prices from a network of mobile cameras," *DCOSS*, 2008.

[15] H. Lu, N. Lane, S. Eisenman, and A. Campbell, "Bubble-sensing: Binding sensing tasks to the physical world," *PMC*, vol. 6, no. 1, 2010.

[16] E. Miluzzo, C. Cornelius, A. Ramaswamy, T. Choudhury, Z. Liu, and A. Campbell, "Darwin phones: the evolution of sensing and inference on mobile phones," in *MobiSys*, 2010.

[17] T. Das, P. Mohan, V. Padmanabhan, R. Ramjee, and A. Sharma, "PRISM: Platform for Remote Sensing using Smartphones," in *MobiSys*, 2010.

[18] J. Lee and B. Hoh, "Sell Your Experiences: A Market Mechanism based Incentive for Participatory Sensing," in *PerCom*, 2010.

[19] S. Reddy, D. Estrin, and M. Srivastava, "Recruitment Framework for Participatory Sensing Data Collections," *Pervasive Computing*, 2010.

[20] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, "Private information retrieval," *Journal of ACM*, vol. 45, no. 6, 1998.

[21] R. Sion and B. Carbunar, "On the Computational Practicality of Private Information Retrieval," in *NDSS*, 2007.

[22] J. Shi, R. Zhang, Y. Liu, and Y. Zhang, "PriSense: Privacy-Preserving Data Aggregation in People-Centric Urban Sensing Systems," in *INFOCOM*, 2010.

[23] D. Liu, P. Ning, and W. Du, "Group-based key predistribution for wireless sensor networks," *ACM Trans. Sensor Networks (TOSN)*, 2008.
[24] R. Anderson, H. Chan, and A. Perrig, "Key Infection: Smart Trust for Smart Dust," in *ICNP*, 2004.
[25] A. Gupta, J. Kuri, and P. Nuggehalli, "A new scheme for establishing pairwise keys for wireless sensor networks," in *ICDCN*, 2006.
[26] A. Caruso, S. Chessa, S. De, and A. Urpi, "GPS free coordinate assignment and routing in wireless sensor networks," in *INFOCOM*, 2005.
[27] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *MobiCom*, 2000.
[28] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-resilient, energy-efficient multipath routing in wireless sensor networks," in *MobiHoc*, 2001.
[29] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," *Ad Hoc Networks*, vol. 1, no. 2-3, 2003.
[30] R. Dingledine, N. Mathewson, and P. Syverson, "TOR: The second-generation onion router," in *USENIX Security*, 2004.
[31] K. Chang and K. G. Shin, "Distributed authentication of program integrity verification in wireless sensor networks," *ACM TISSEC*, vol. 11, 2008.
[32] D. Perito and G. Tsudik, "Secure Code Update for Embedded Devices via Proofs of Secure Erasure," in *ESORICS*, 2010.
[33] J. Daeman and V. Rijmen, "AES proposal: Rijndael," 1999.
[34] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*. MIT Press, Cambridge, MA, 2001.
[35] S. Ortolani, M. Conti, B. Crispo, and R. Di Pietro, "Event handoff unobservability in WSN," in *iNetSec*, 2010.
[36] K. Shilton, "Four Billion Little Brothers? Privacy, Mobile Phones, and Ubiquitous Data Collection," *Communications of the ACM*, vol. 52, no. 11, 2009.
[37] A. Kapadia, D. Kotz, and N. Triandopoulos, "Opportunistic Sensing: Security Challenges for the New Paradigm," in *COMNETS*, 2009.
[38] D. Christin, M. Hollick, and M. Manulis, "Security and Privacy Objectives for Sensing Applications in Wireless Community Networks," in *ICCCN*, 2010.
[39] C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos, "AnonySense: Privacy-aware people-centric sensing," in *MobiSys*, 2008.
[40] K. Huang, S. Kanhere, and W. Hu, "Preserving Privacy in Participatory Sensing Systems," *Computer Communications*, vol. 33, no. 11, 2010.
[41] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of ACM*, vol. 24, no. 2, 1981.
[42] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," *ACM TKDD*, vol. 1, no. 1, 2007.
[43] R. Cáceres, L. P. Cox, H. Lim, A. Shakimov, and A. Varshavsky, "Virtual individual servers as privacy-preserving proxies for mobile devices," in *MobiHeld Workshop*, 2009.
[44] R. Ganti, N. Pham, Y. Tsai, and T. Abdelzaher, "PoolView: Stream Privacy for Grassroots Participatory Sensing," in *SenSys*, 2008.
[45] A. Dua, N. Bulusu, W. Feng, and W. Hu, "Towards Trustworthy Participatory Sensing," in *HotSec*, 2009.
[46] P. Gilbert, L. Cox, J. Jung, and D. Wetherall, "Toward trustworthy Mobile Sensing," in *HotMobile*, 2010.
[47] E. De Cristofaro and C. Soriente, "PEPSI: Privacy-Enhanced Participatory Sensing Infrastructure," in *WiSec*, 2011.
[48] E. De Cristofaro and R. Di Pietro, "Preserving Query Privacy in Urban Sensing Systems," in *ICDCN*, 2012.
[49] A. Bagula and M. Zennaro, "Wireless Sensors for All—Low cost, open solutions for environmental monitoring," http://www.ws4all.org, 2011.
[50] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, 2002.
[51] E. De Cristofaro, X. Ding, and G. Tsudik, "Privacy-preserving querying in wireless sensor networks," in *ICCCN*, 2009.
[52] M. G. Reed, P. F. Syverson, and D. M. Goldschlag, "Anonymous connections and onion routing," *IEEE JSAC*, vol. 16, no. 4, 1998.
[53] L. Sweeney, "k-Anonymity: A model for Protecting Privacy," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, 2002.
[54] B. Carbunar, Y. Yu, W. Shi, M. Pearce, and V. Vasudevan, "Query privacy in wireless sensor networks," *ACM TOSN*, vol. 6, no. 2, 2010.
[55] F. Chen and A. Liu, "SafeQ: Secure and Efficient Query Processing in Sensor Networks," in *INFOCOM*, 2010.

Emiliano De Cristofaro s a Research Scientist at the Palo Alto Research Center (PARC), where he works in the Security and Privacy group. He got a B.Sc. in Computer Science from the University of Salerno (Italy) in 2005 and a PhD in Networked Systems from the University of California, Irvine. His research interests include security, privacy, and applied cryptography. More information at: http://www.emilianodc.com.

Roberto Di Pietro Roberto Di Pietro is currently an Assistant Professor at the Department of Mathematics of Universit di Roma Tre -UniRoma3- Roma, Italy. He graduated in Pisa (1994) and achieved his PhD at University of Roma "La Sapienza" (2004). He is the Principal Investigator of the SPRINGeR research group at UniRoma3. His main research interests include: security and privacy for distributed systems (e.g. mobile, ad-hoc, and wireless networks; Cloud); applied cryptography; and, computer forensics. More information at: http://ricerca.mat.uniroma3.it/users/dipietro/.