Privacy-Preserving Mechanisms for Crowdsensing: Survey and Research Challenges

Idalides J. Vergara-Laurens, *Member, IEEE*, Luis G. Jaimes, *Graduate Student Member, IEEE*,
and Miguel A. Labrador, *Member, IEEE*

***Abstract*—Crowdsensing (CS) is a new data collection paradigm based on the willingness of people to utilize their mobile devices to sense and transmit data of interest. Given the large amount of cellular users, mobile sensor networks will be able to collect enough data to address large-scale societal problems in a fast, easy, and cost-effective manner. One important issue in CS is that of privacy; without appropriate privacy-preserving mechanisms, many users will not be willing to participate in the data collection process. This paper presents the state-of-the-art in privacy-preserving mechanisms for CS systems. After a general description of CS systems and their main components, this paper addresses the most important issues to consider in the design, implementation, and evaluation of privacy-preserving mechanisms. Then, following a new taxonomy, the most important mechanisms available in the literature are described and qualitatively evaluated. Finally, this paper presents research challenges that should be addressed in order to improve the performance of future privacy-preserving mechanisms for CS systems.**

***Index Terms*—Anonymization, crowdsensing (CS), encryption, obfuscation, participatory sensing, privacy, ubiquitous sensing.**

I. Introduction

CROWDSENSING (CS) is a new sensing paradigm that relies on willingness of people to utilize their mobile devices sensors (e.g., microphone, camera, and accelerometer) to sense and report the data [1]–[4]. In this context, community-oriented CS applications are developed to address specific issues affecting a community. For example, the application described in [5] asks users to tag poisonous plants in a community park to alert other neighbors and develop a map with "safe" trails. Another example application is the one described in [6], which alerts users of traffic jams and accidents. A step further is taken by researchers of the autosense project, they propose a system to alert users about the most stressful routes [7]. Thereby, a user might prefer to take a longer but less stressful route, instead of the shorter stressful one. A more large-scale application could ask users around a state or even a country to take samples of air quality, sound pollution, visual contamination, or even measure pollen levels in the air in order to create thematic maps.

One of the most significant advantages of these CS applications is that given the large number of existing cellular users, they have the potential to collect data like never before and from places not economically feasible before, and in a fast, easy, and cost-effective manner. For example, CS-based traffic congestion applications have the potential to collect real-time data not only from main interstate roads but also from secondary and even tertiary roads, something that is very costly using current technologies. Deploying static sensors over all roads will be economically expensive in terms of capital, installation and maintenance costs. Similarly, a CS-based application to measure environmental variables in a city or state will allow the detection of abnormal levels that are nearly impossible to detect with the current static environmental sensing stations, which rely on estimation and interpolation techniques to provide a value of the variable of interest miles away from the station. In addition, since the sensing devices are in the hands of the users, CS applications incur in very small installation and maintenance costs, and do not need to provide energy and Internet connectivity (i.e., each participant uses her own) to report data.

While these are significant advantages, this new sensing paradigm also brings new problems. For example, since the sensing devices are in the hands of the users, CS applications are easier to hack in order to report "different" values. Similarly, since the sensing devices are usually cheaper than those utilized in current technologies, they are more prone to failures, reporting "invalid" data (e.g., due to sensors failure, samples taken in locations other than the specified in the request, etc.) Therefore, new mechanisms to secure and validate the reported data are needed. Another important issue is that of the willingness of participation. CS applications cannot assume that the users will be willing to participate and spend their cellular phones' battery and data plan to sense and report the sample data if they do not have a direct benefit [8], [9]. For some applications this benefit is intrinsic and the users will be more than willing to participate, like in the case of the poisonous flowers in the community park. However, for some other applications, like the pollution application, some sort of incentive (maybe monetary) mechanism is needed if we want to guarantee some level of user participation [10]–[12]. Finally, there is the issue of privacy. For most, if not all, applications, privacy-preserving mechanisms must be in place to guarantee the privacy of the participating users [13], [14]. This is a

Manuscript received January 31, 2016; revised May 10, 2016; accepted July 6, 2016. Date of publication July 27, 2016; date of current version August 9, 2017.

I. J. Vergara-Laurens is with the Department of Computer and Electrical Engineering, Universidad del Turabo, Gurabo, PR 00778-3030 USA.

L. G. Jaimes is with the Department of Computer Science and Engineering, Bethune-Cookman University, Daytona Beach, FL 32114 USA (e-mail: ljaimes@mail.usf.edu).

M. A. Labrador is with the Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620 USA.

Digital Object Identifier 10.1109/JIOT.2016.2594205

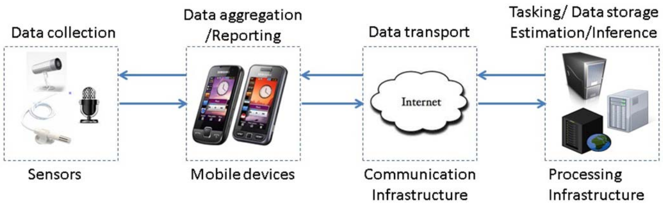
Data collection
Data aggregation /Reporting
Data transport
Tasking/ Data storage Estimation/Inference
Internet
Sensors
Mobile devices
Communication Infrastructure
Processing Infrastructure

Fig. 1. General hardware architecture for CS systems.

key issue in CS systems and the main focus of this survey paper.

This paper surveys the state-of-the-art in privacy-preserving mechanisms for CS systems. After a general description of CS systems and their main components, this paper analyzes the most important issues to consider in the design, implementation, and evaluation of privacy-preserving mechanisms. Then, following a new taxonomy, the most important mechanisms available in the literature are described and qualitatively evaluated according to these design issues. Finally, this paper presents open research challenges that should be addressed to improve the performance of future privacy-preserving mechanisms for CS systems.

II. CS Systems

This section includes a general architecture for CS systems and describes its main components.

A. System Hardware Architecture

CS systems may have different types of architecture. The simplest one utilize the cell phone built-in sensors such as accelerometers, gyroscopes. Some others architectures consist of a wireless sensor network (WSN) in which the sensors are connected to mobile phones. As any other WSN, a CS application gathers data, performs an initial aggregation and transmits such data to a central infrastructure where the data are processed in order to produce useful information. Another configuration, consists of a WSN that connects several wireless body area networks. A general hardware architecture for CS systems is shown in Fig. 1. As shown, the architecture consists of the following four main components [2], [15].

1) Sensors: These devices reside inside (e.g., accelerometers, gyroscope, and microphone) or outside (e.g., in-vehicle sensors and pollution sensors) of the mobile device, and are used to measure variables of interest. These measurements might include: user's location, images, sounds, videos, acceleration, and environmental measurements.

2) Mobile Device: Mobile devices are responsible for the data consolidation, report creation, and report transmission to the processing infrastructure using the communication infrastructure. Typical mobile devices include: smart phones, smart watches, smart glasses, tablets, and laptops.

3) Communication Infrastructure: This is the communication network (e.g., Wi-Fi and 4G networks) utilized to transport the data from the mobile devices to the processing infrastructure.

4) Processing Infrastructure: These servers are responsible for the tasking, data storage, and estimation-inference processes.

B. Protocols

In addition to the hardware components of the architecture, the functionality of CS systems is accomplished by several protocols that allow the transmission of data from the application server to the mobile devices and vice versa. The most important protocols are the following.

1) Authentication Protocols: These protocols are designed to authenticate the participants of the system and encrypt the data. The main objective is to avoid the collection of malicious data from unreliable sources and protect the integrity of the data. The current trend is for every participating device to be registered to the system [16]. In some cases, such as the system presented in [17], the system installs a participant certificate in the mobile device to authenticate the user. This certificate, which is renovated periodically by a *registration authority*, is then used to encrypt and transmit the collected data to the data broker.

2) Tasking Protocols: These are the protocols used during the tasking process to send the task requests to the mobile devices. Their main objective is to provide the mobile devices with the instructions for the data collection process. Since the system usually targets a specific region and data type, the application server generates a sensing model that includes the target region, target sensing data, sensing time window, reporting time window, and the sensors needed to collect the data [13], [14], [18], [19]. This model is distributed to the participating devices using the tasking protocols. This distribution is performed in some cases using a central tasking server [20], in a distributed fashion such as in the work presented in [21], or in a hybrid manner using an approach like the one presented in [22].

3) Reporting Protocols: Once the mobile devices receive the task message, they collect data during the reporting time window and report the data to the central servers, as defined by the application server [23].

4) Communication Protocols: CS system entities use standard Internet communication protocols, such as the TCP/IP, protocols defined by the Standard IEEE 802, and Bluetooth stack among others, to send and receive the data between mobile devices and, to and from the processing infrastructure. In some cases, the CS infrastructure have the form of a delay-tolerant network or disruption-tolerant network [24]. In these cases, the mobile nodes or data mules are the primary source of transitive communication between two non-neighboring nodes in the network.

III. Design Considerations for Privacy-Preserving Mechanisms

This section introduces those aspects that are important to consider when designing privacy-preserving mechanisms for CS systems.

A. Type of System: Centralized or Distributed

A centralized system assumes the existence of one or more trusted entities acting as proxies for providing privacy preservation. For example, some systems use an external entity that receives the actual value from participants, computes a generalized value, that represents a set of participants, which is transmitted to the participants; finally, the generalized value is reported to the application servers by participants [25]. Distributed systems, on the other hand, do not need a central entity to provide privacy. In general, centralized systems are easier to manage but inherit the disadvantages of centralized architectures such as a single point of failures and the fact that if the server is compromised all users can be compromised. On the other hand, distributed systems are harder to design, maintain, and implement but are usually more robust against attacks.

B. Threat Model or Types of Attacks

Privacy-preserving mechanisms can be designed to defend the CS system from the following types of attacks.

1) Snapshots and Historical Attacks: If the adversary is limited to capture a single report transmitted by a participant, the system may be subject to snapshot attacks. On the other hand, if the adversary has access to the collected data (historical data from the repository server) and is able to establish historical correlations between different reports, the attack is called historical [26].

2) Internal and External Attacks: Privacy-preserving mechanisms can be designed to defend the system against external (i.e., people that do not participate in the system) and/or internal adversaries (i.e., people that participate in the systems such as staff and participants).

3) Attacks Exploiting Knowledge of the Defense: It is a common assumption in security research that adversaries know the algorithm used for protecting the information since these algorithms are often released to the public [27]. Therefore, if the privacy-preserving mechanism is known to everyone and the adversary makes use of that knowledge to compromise the system, then the attack is called def-aware. On the contrary, the attack is called def-unaware [26], [28], [29].

C. Information Loss

According to [23] and [30], data accuracy becomes orthogonal to security/privacy in the case of CS applications. For instance, some privacy mechanisms generalize the actual location of the participant to a bigger area and a larger generalization guarantees a higher level of privacy. However, from the application view point, the reported data with the generalized location might not be good enough to provide the intended service, making the application useless. Therefore, the system requirements should define the maximum size of these generalized areas.

The accuracy of the systems is normally measured by the information loss (IL), a metric meant to indicate how different the reported data are from the real ones.

D. Reporting Frequency

The reporting frequency is the periodicity at which the mobile device must report the sensed data to the central servers [23]. Some systems require real-time reports while others allow for batch reporting [31], [32]. In this latter case, sensed data are stored in the mobile device and a batch of records is reported periodically. Privacy-preserving mechanisms for applications that require real-time reporting must be simple in terms of computational complexity, otherwise they will drain the battery of the mobile device very fast.

E. Computational Complexity

Given the memory, energy, CPU power, and communication constraints found in mobile devices, this design aspect is very important. This is the main reason why some complex privacy-preserving mechanisms are implemented in a central server, usually called the anonymizer. Simpler and less expensive mechanisms are run directly in the mobile device at the expense of its resources, as the privacy-preserving mechanism is designed with these constrained resources in mind [33].

F. Communication Overhead

In WSNs, data transmission is the most energy consuming process [34]. Therefore, it is important to reduce the amount of transmitted data in order to reduce energy consumption in mobile devices. Consequently, privacy-preserving mechanisms should aim at not increasing the amount of transmitted data significantly.

G. Energy Consumption

Mobile devices are usually severely constrained in terms of battery, memory, CPU power, and communication capabilities. Therefore, battery-hungry mechanisms are usually not very appealing, as users will not be willing to participate in the application [34], [35].

IV. Taxonomy

CS systems have associated two major processes in which the participant is involved: 1) tasking and 2) reporting processes [36], [37]. The tasking process is used to task mobile devices for a specific sensing campaign. In this process, the tasking protocols are used in order to provide the mobile devices with the instructions for the data collection process (e.g., target region, target sensing data, sensing time window, reporting time window, and the sensors needed to collect the data). The major privacy-related challenge for the tasking process is that the system has to task trusty and appropriate participants without learning participants' sensitive data.

In the reporting process, the participants send the sensed data to the processing infrastructure. Since the reported information contain data from the variable of interest (e.g., measurements, location, time, and physiological signals), an adversary may use these data to identify the user and learn her sensitive data (e.g., medical condition, home address, and travel route). Therefore, each process has different privacy-protection requirements and the privacy-preserving mechanisms must address each particular problem.

Fig. 2 presents the three-level taxonomy used in this section to classify privacy-preserving mechanisms. At the top level, privacy-preserving mechanisms are classified based on the target process: tasking or reporting process. This classification recognizes that these privacy-preserving mechanisms

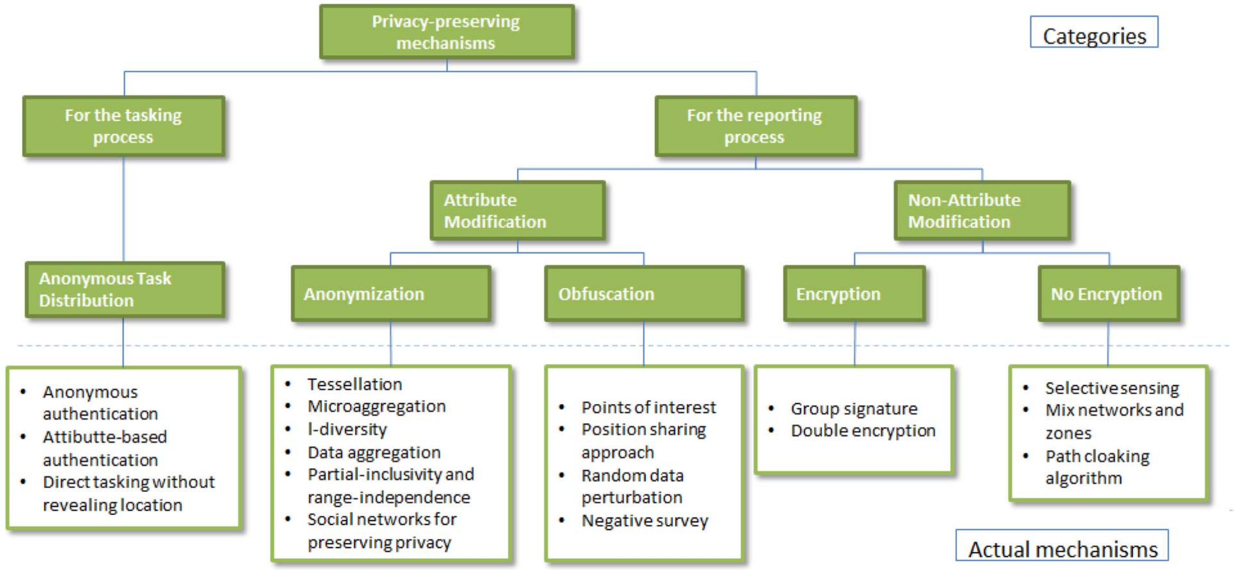

Fig. 2. Taxonomy of privacy-preserving mechanisms for CS systems.

need to be applied not only to the reporting process but to the tasking process as well, something that has not been acknowledged thus far. In the case of the tasking process, the privacy-preserving mechanisms available in the literature aim to avoid the association between the identity of the participant and her sensitive data through anonymous task distribution approaches [13].

For the reporting process, the privacy-preserving mechanisms are classified according to whether or not the mechanism modifies the attributes: attribute modification and nonattribute modification. Attribute modification mechanisms aim to avoid the association between a participant and her sensitive data modifying the actual data and reporting the modified data (e.g., the reported location is changed to a generalized value that represents a group of participants instead of one participant). These methods are categorized in two: 1) anonymization-based and 2) obfuscation-based. Anonymization-based mechanisms change the participant's actual data to a generalized data that represent a group of participants. Obfuscation-based mechanisms, on the other hand, change the actual data without considering other participants.

Nonattribute modification methods normally use cryptographic techniques in order to protect participants' sensitive data without modifying the actual value. The next sections explain the taxonomy in more detail, describe the most important privacy-preserving mechanisms available in the literature, and evaluate them in a qualitative manner according to the design issues presented on Section III.

V. Privacy-Preserving Mechanisms for the Tasking Process

During the tasking process, which may be repeated several times according to the user and the system, the privacy of the user may be in jeopardy if appropriate privacy-preserving mechanisms are not in place [36], [37]. For example, if the system asks certain mobile devices to participate in a campaign in a specific region, and the participants, currently located in that region, report their willingness to participate, then an adversary could identify the whereabouts of those participants. As a result, most mechanisms in this category aim to guarantee the privacy of the participants avoiding the association between their identities and their actual locations during the tasking process. Many of these mechanisms are based on the anonymous distribution approach, in which the system does not learn the actual location of the participants during the tasking process.

The anonymous distribution of the tasking process can be accomplished using anonymous authentication, attribute-based authentication, and tasking the users directly without revealing their locations.

A. Anonymous Authentication

In this approach, presented in [16], [17], and [38], users receive tasks through *beacons* without the need to reveal their identities to the system. For example, the anonysense system, presented in [17], periodically posts the tasking campaign and when the participants are in public locations, locations considered by the participant as nonsensitive ones, they download all available tasks from a tasking service. For each connection, the participant performs an anonymous authentication, based on a group signature defined by direct anonymous attestation [39], in order to prove to the system that it is a valid participant, but without revealing its identity. Therefore, the system only learns that some participants are in public locations but nothing else, which is the main advantage of this system. However, its main drawback is that, since the system only learns that some users are in public locations, the system cannot predict how many users are likely to visit a particular region and guarantee a good inference and data analysis.

Another interesting application corresponds to the anonymous authentication of visitors (AAV) protocol [16]. The main idea is to authenticate the information reported by a visitor

TABLE I
SUMMARY OF PRIVACY-PRESERVING MECHANISMS FOR THE TASKING PROCESS

Approach	Type of system	Knowledge of defense	Protection internal /external attacks	Computational complexity	Communication overhead	Energy consumption
A. Anonymous task distribution						
1. Anonymous authentication	Distributed	Def-aware	Internal /External	Low	Low	Low
2. Attribute-based authentication	Centralized	Def-aware	External	High	High	High
3. Direct tasking without revealing location	Distributed	Def-aware	Internal /External	Low	Low	Low

inside a thematic park without divulging the visitor's identity. Here, AAV protocol is executed in two phases: 1) "certified pseudonym issuing phase" and 2) "subsequent interaction phase." The visitors's mobile app generates a pseudonym *P* and utilizes a partially blind signature [40] to hide *P* in a blinded message *B*, the app then sends the ticket ID along with *B* to the app server. The server inputs an expiry date while digitally signing *B*. As a result the app server has no clue about visitor's pseudonym and cannot link the future communications from this pseudonym to the visitor.

Anonymous authentication schemes are distributed since there is no central entity that knows the actual locations of the users, and protect against external and internal attacks because participants do not reveal their locations to the system. In addition, they are classified as def-aware with low complexity and low energy consumption since they do not require additional processing or transmission.

B. Attribute-Based Authentication

In this approach, users authenticate to the system by revealing only their attributes, such as group affiliation or sensor availability, but without revealing their identities [13], [37], [41]. The main idea is to use cryptographic primitives to prove that they belong to a certain group without revealing their precise identity. For example, using a group certificate, a user can prove that she satisfies the system requirements while being anonymized to a group of participants. The main drawback of these mechanisms is associated with the size and characteristics of the groups. An adversary with additional knowledge could identify the mobile device and track the user to certain groups, allowing him to identify the precise participant.

Attribute-based systems are centralized systems. Since there is participant authentication using group certificates, an adversary could identify a group of users that share the same attribute characteristics. However, since the user does not reveal her identity to the system, the adversary cannot identify her from the group. Additionally, the attribute-based authentication technique avoids nonauthorized or unreliable participants, i.e., participants that do not meet the campaign requirements such as a specific sensor from participating in the campaign. Attribute-based schemes protect against external attacks because the system can learn the location of a user without revealing her identity. However, an internal adversary could identify the location of a group of participants and using external knowledge could identify them. Additionally, this approach is classified as a def-aware method since an adversary could know that this mechanism has been used, but he needs the encryption key in order to access the participants' sensitive data. Finally, since it requires additional processing and transmissions for authentication and encryption, this approach is considered as having high computational complexity because it is based on cryptographic schemes. Communication overhead and energy consumption are considered high as well.

C. Direct Tasking Without Revealing Location

This approach, applied in [42] and [43], tasks participants based on their identity, but they do not reveal their location. Reference [37] proposed to use anonymizing networks, such as Tor [44]. This system is a distributed overlay network designed to anonymize TCP-based applications where each node in the path knows its predecessor and successor, but no other nodes in the circuit, guaranteeing the privacy of the client.

Direct tasking schemes are classified as distributed because there is no a central entity knowing the locations of the participants. Additionally, these schemes avoid nonreliable sources because they only select the participants that meet the system requirements for the specific campaign. They protect against external and internal attacks because participants do not reveal their locations to the system. In addition, they are classified as def-aware methods because an adversary could know that this mechanism has been used but he is not able to discover the participants' sensitive data. Finally, since they do not require additional processing or transmissions, they are considered as having low complexity, overhead, and energy consumption because the mobile device is not required to perform additional process.

Table I summarizes the mechanisms presented in this section along with the qualitative evaluation.

VI. PRIVACY-PRESERVING MECHANISMS FOR THE REPORTING PROCESS: ATTRIBUTE MODIFICATION MECHANISMS

During the normal operation of the CS application, each participant transmits the sensed data to the central servers during the reporting process. Since these applications associate the data with the time and location of the user, the reporting process is the most privacy-sensible process in the system. An important aspect during the reporting process is the use of quasi-identifiers. According to [45], a quasi-identifier is defined as an attribute that can be linked to publicly available data to identify individuals (i.e., date of birth, gender, zip code, etc.). Therefore, attribute modification mechanisms aim to transform quasi-identifiers in such a way that they cannot be used to identify a specific participant in the system. These mechanisms are classified as anonymization- and obfuscation-based. For this and following sections, the location is considered as the sensitive data; however, the mechanisms are extensible to other sensitive data such as physiological signals, vehicle speed, etc.

A. Attribute Modification Mechanisms: Anonymization

These techniques aim to avoid the association between the individual and her sensitive data anonymizing the user's identity to a group of users. The key idea, initially developed for location-based services in [46], is to transform the participant's quasi-identifier value to a generalized value corresponding to a set of users [28]. Usually, the quasi-identifier corresponds to the participant's location. The following are the most important mechanisms in this category.

1) Tessellation: In this scheme each location point is enlarged to a region called a tile, containing k users [38]. The sensed data are reported using the tile identifiers as the sensing location. The implementation in [38] is used in Wi-Fi networks, in which each access point (AP) covers a region called a cell, and keeps track of the average number of connected devices k_t. If $k_t < k$ then the cells are combined into bigger tiles in order to guarantee k-anonymity. Once the tiles are defined, participants report their sense data tagged to the center of the tiles instead of their actual locations. However, due to the tile size, its main drawback is the lack of accuracy for applications that require fine-grained location information, such as an application for monitoring traffic in highways, as the system may not be able to identify a particular intersection. Additionally, given that APs keep track of connected users, this scheme is only applicable for external attacks; in the case of $k_t < k$ it is possible that an internal attack could identify a single user into a small region, violating the k-anonymity condition.

Tessellation is a centralized approach since it requires an anonymizer knowing the actual participants' locations in order to generate the anonymized location. This approach produces a high IL because the tiles can be very large in order to comply with the k-anonymity requirement. It is classified as a def-aware method since an adversary could know that this mechanism has been used, but he is not able to discover the participants' sensitive data. As a centralized approach, an internal adversary could get access to the anonymizer server and the participants' private data. The computational complexity is considered to be low since this approach does not require any additional computation in the mobile device. However, the communication overhead is considered as medium because the participant is required to send an additional message to the anonymizer in order to obtain the anonymized location. Finally, the energy consumption is considered as medium because of the additional message.

2) Microaggregation: Microaggregation is a k-anonymity approach that defines a set of equivalent classes of k users for protecting against an adversary [47], [48]. In this case, the sensitive attributes in the records of the k-users are changed to an average value according to an average function. However, optimally solving microaggregation on multivariate data sets is known to be NP-hard. Therefore, approximation algorithms are used in practice [49]. A variant of the maximum distance to average vector (VMDA) is proposed in [50]; this approach consists on defining tiles of different sizes according to the distance between the users in a particular time window, as shown in Fig. 3. This scheme could be implemented in a centralized fashion or in a distributed way using a peer-to-peer scheme where participants exchange location data before reporting them to the application server. The main drawback of this scheme lies in its complexity and the need to exchange actual location with another entity, an anonymization server (AS) in the centralized case, or a set of neighbors in the distributed case.

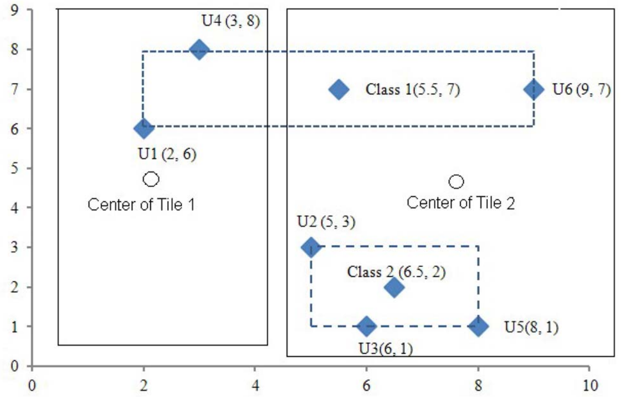

Fig. 3. Example comparing tessellation and microaggregation approaches.

Fig. 3 shows an example comparing microaggregation and tessellation approaches using six users $U1$–$U6$. In the case of tessellation, users $U4$ and $U1$ are anonymized to the center of tile 1; on the other hand, they are anonymized to *class1* using the microaggregation approach. Given the distance between actual and anonymized locations, the IL is higher in tessellation than in microaggregation [51]. As shown in Fig. 3, microaggregation produces a medium level of IL because the VMDA provides better accuracy than tessellation; however, it depends on the dispersion of the users in the area of interest.

This approach is classified as a def-aware method since an adversary could know that this mechanism is being used, but he is not able to discover the participants' sensitive data. In the case of a centralized system, an internal adversary could get access to the anonymizer server and obtain the participants' private data. Therefore, this approach is considered to protect against external adversaries. The computational complexity is considered low because this approach does not require any additional computation in the mobile device; however, the computational complexity is considered medium if done distributed. The communication overhead is considered medium since participants are required to send an additional message to the anonymizer in order to obtain the anonymized location. Finally, the energy consumption is also medium because of the additional communication overhead.

3) L-Diversity: This approach, presented in [52], is a two-stage anonymization scheme which applies the VMDA at first over the spatial dimension and then over the

TABLE II
FIRST STAGE OF *L*-DIVERSITY AFTER APPLYING VMDA OVER THE SPATIAL DIMENSION

User	Location	Time	Class Id	Anonymized location	Anonymized time
U1	(2,3)	4:20	1	(2.5,3)	**:**
U2	(5,6)	4:10	3	(7.5, 7.5)	**:**
U3	(8,2)	4:45	2	(7, 2.5)	**:**
U4	(10,9)	4:30	3	(7.5, 7.5)	**:**
U5	(4,5)	4:00	1	(2.5,3)	**:**
U6	(7,8)	3:50	3	(7.5, 7.5)	**:**
U7	(9,2)	12:30	2	(7, 2.5)	**:**
U8	(2,8)	1:50	4	(1.5, 7.5)	**:**
U9	(1,1)	3:15	1	(2.5,3)	**:**
U10	(5,3)	2:16	2	(7, 2.5)	**:**
U11	(2,6)	7:08	4	(1.5, 7.5)	**:**
U12	(1,9)	4:02	4	(1.5, 7.5)	**:**

TABLE III
SECOND STAGE OF *L*-DIVERSITY AFTER APPLYING VMDA OVER THE TEMPORAL DIMENSION

User	Location	Time	Class Id	Anonymized location	Anonymized time	Group
U1	(2,3)	4:20	1	(2.5,3)	4:17	1
U2	(5,6)	4:10	3	(7.5, 7.5)	4:17	1
U3	(8,2)	4:45	2	(7, 2.5)	4:17	1
U4	(10,9)	4:30	3	(7.5, 7.5)	4:17	1
U5	(4,5)	4:00	1	(2.5,3)	4:17	1
U6	(7,8)	3:50	3	(7.5, 7.5)	5:08	2
U7	(9,2)	12:30	2	(7, 2.5)	5:08	2
U8	(2,8)	1:50	4	(1.5, 7.5)	5:08	2
U9	(1,1)	3:15	1	(2.5,3)	5:08	2
U10	(5,3)	2:16	2	(7, 2.5)	5:08	2
U11	(2,6)	2:08	4	(1.5, 7.5)	5:08	2
U12	(1,9)	4:02	4	(1.5, 7.5)	4:17	1

temporal dimension. Previous approaches assumed that the temporal variable was anonymized using a simple technique such as adding a random value to the actual time or changing the value to an average time window. However, this approach anonymizes the participants to different classes in the spatial dimension and these classes are then anonymized to groups in the temporal dimension using VMDA in both cases. Therefore, the final report sent by each user to the application server contains the anonymized location and time and the group ID.

To define the size of each group, the authors use *Group size* $= k * l$, where k means the required k-anonymity level and l defines the level of diversity. Table II shows the first stage of l-diversity after applying VMDA over the spatial dimension using $k = 3$, which defines the size of each class. Table III shows the second stage of l-diversity after applying VMDA over the temporal dimension using $k = 3$ and $l = 2$ for the temporal dimension. Consequently, the size of each group is $3 * 2 = 6$. As it can be seen, the mechanism anonymizes the location, the time, and the group Id.

The main advantage of l-diversity is its capability for protecting against historical attacks. On the other hand, its main drawback is the need of a proxy aware of the actual location of each user to produce the anonymized location, time, class ID, and/or group ID. Thus, there is no guarantee that privacy will be preserved if any user or proxy is compromised.

L-diversity works in a centralized manner since it requires an anonymizer knowing the actual locations of the participants to generate the anonymized locations. This approach produces a medium level of IL since the VMDA produces a better accuracy than tessellation. *L*-diversity is classified as a def-aware method since an adversary could know that this mechanism is being used, but he is not able to discover participants' sensitive data. Since it is a centralized approach, an internal adversary accessing the anonymizer server could have access to participants' private data. The computational complexity is considered to be low because this approach does not require any additional computation in the mobile device; all the anonymization process occurs in the central anonymizer. The communication overhead is considered medium because the participants are required to send an additional message to the anonymizer in order to obtain the anonymized location. Finally, the energy consumption is considered medium because of the additional communication overhead. A comprehensive review about *L*-diversity, *k*-anonymity, and micro-agregation in the context of CS can be found in [53].

4) Data Aggregation: Data aggregation techniques are based on aggregation functions. According to [54], a data aggregation function is defined as $y(t) \equiv f(d_1(t), d_2(t), d_N(t))$, where $d_i(t)$ is the individual sensor reading at time t for user i. Typical functions of f include *sum, average, min, max, and count*. The model called cluster-based data aggregation presented in [54], divides the network into clusters and each user in a cluster reports the data to a leader or cover node, which reports the cluster's aggregation to the central server. For the transmission of the data, each node uses a pairwise shared key scheme to encrypt/decrypt data slices transmitted to their cover node, a node that it uses to preserve its privacy. According to [55], statistical functions (e.g., average, count, variance, standard deviation, and any other moment of the measured data) can be reduced to the additive aggregation function sum. Therefore, [54] focus on an additive aggregation function where $f(t) = \sum_{i=1}^{N} d_i(t)$. For instance, a cluster containing three members: A, B, and C, where a, b, and c represent the private data held by these nodes, respectively. Let A be the cluster leader of this cluster. Then, node A calculates

$$\begin{aligned} v_A^A &= a + r_1^A x + r_2^A x^2 \\ v_B^A &= a + r_1^A y + r_2^A y^2 \\ v_C^A &= a + r_1^A z + r_2^A z^2 \end{aligned} \tag{1}$$

where x, y, and z are numbers known to all nodes, r_1^A and r_2^A are two random numbers generated by node A and known only to it. Similarly, nodes B and C calculate v_A^B, v_B^B, v_C^B, and v_A^C, v_B^C, v_C^C independently.

Then each node i encrypts the corresponding v_j^i and sends it to each other node j. When node A receives v_A^B and v_A^C, it computes $F_A = v_A^A + v_A^B + v_A^C = a + b + c + r_1 x + r_2 x^2$, where $r_i = r_i^A + r_i^B + r_i^C$. Similarly, when node B receives v_B^A and v_B^C, it computes $F_B = v_B^A + v_B^B + v_B^C = a + b + c + r_1 y + r_2 y^2$, and node C computes $F_C = v_C^A + v_C^B + v_C^C = a + b + c + r_1 z + r_2 z^2$, when it receives v_C^A and v_C^B. Then, nodes B and C send F_B and F_C to node A. Therefore, node A knows

$$\begin{aligned} F_A &= v_A^A + v_A^B + v_A^C = a + b + c + r_1 x + r_2 x^2 \\ F_B &= v_B^A + v_B^B + v_B^C = a + b + c + r_1 y + r_2 y^2 \\ F_C &= v_C^A + v_C^B + v_C^C = a + b + c + r_1 z + r_2 z^2 \end{aligned} \tag{2}$$

which can be represented as

$$F = GU \tag{3}$$

where

$$G = \begin{bmatrix} 1 & x & x^2 \\ 1 & y & y^2 \\ 1 & z & z^2 \end{bmatrix} \tag{4}$$

$$U = \begin{bmatrix} a+b+c \\ r_1 \\ r_2 \end{bmatrix}. \tag{5}$$

Therefore, node A can compute $a+b+c$ from

$$U = G^{-1}F. \tag{6}$$

The main problem of this approach is the computational overhead: the larger the cluster size, the higher the computational overhead. However, a large cluster size improves privacy under node collusion attacks [54].

Other data aggregation mechanisms use a slicing technique, like the slice-mix-aggregate (SMART) mechanisms presented in [54]. The idea of SMART is to reduce the computational overhead at the expense of a slightly increase in the communication overhead. SMART is based on a three-stage procedure: 1) slicing; 2) mixing; and 3) aggregation, similar to the scheme implemented in [56]. The main idea, in both cases, is that each node i selects a set of n nodes called *cover nodes*, with $n \leq N-1$, where N the total number of nodes in the network. Then, node i slices its data into $n+1$ random slices and sends each of the n slices to a different *cover node* and keeps the last slice to itself. Note that each node i can be a *cover node* for other nodes in the set. Thus, each node can receives j different slices from j nodes in the network and mix them. When it is required, each node sends the mixed data to the central node A. Finally, node A makes the aggregation and consolidation of the data. These schemes are designed to protect against internal attacks and rely on an encrypted transmission for protecting the system from external attackers.

In both cases, slicing and nonslicing-based techniques, data aggregation works in a distributed manner since there is no a central entity that needs to know the participants' locations. The IL is low because data are not modified during the process. Data aggregation is considered def-aware because each node can select randomly its cover nodes. Therefore, it protects against internal and external adversaries. However, the computational complexity is considered to be high given the matrices operations. Additionally, the communication overhead is considered high since participants need to send data to each of the other n participants. Therefore, the energy consumption is also high.

5) Partial-Inclusivity and Range-Independence: This approach is based on the observation that range queries or reports sent by participants have significant overlaps [10]. Therefore, instead of having each participant to report a separate record, only a group of representative participant reports are sent to the server, as a consolidation of the sensed data in a particular region [57]. This scheme is based on the cloaking of participants' locations to a generalized region called *cloaked region*. For this, each participant defines two parameters k and A, which are the minimum number of participants in her cloaked region and the minimum area of the cloaked region, respectively. In order to define her cloaked region, the participant communicates with her $k-1$ closest neighbors and this cloaked region is reported to the central server. Additionally, the system defines a set of *DC-points*, i.e., locations from which data need to be collected. When needed, the server queries the participants containing a particular set of DC-points in their cloaked regions. Then, each participant computes her Voronoi cell and defines her influence radius r_u, which is the radius of the smallest circle that contains her Voronoi cell. This radius r_u is used to compute the representation score of each participant in the cloaked region. This score value is computed by the local peers in the cloaked region and the peer with the highest score is the one reporting the data to the server. In this way, the server receives only data from one participant in the cloaked region preserving the privacy of the participants in that region.

These schemes are distributed because they do not require of an anonymizer knowing the locations of the participants in order to generate the anonymized location. The IL is low because the system reports the semantically correct data, i.e., the reported data represent correctly the actual data. It is considered to be def-aware since an external adversary could know that this scheme is being used but he is not able to access the actual participants' data. This scheme has a high computational complexity because it requires additional processing for computing the Voronoi cells and the voting process. The communication overhead is high because each participant transmits data to each peer during the cloaked region formation as well as for the voting process during the selection of the representative participant of the cloaked region.

6) Social Networks for Preserving Privacy: Social networks have also been used to preserve privacy in CS systems [58]. These mechanisms assume that the third-party server which all mobile users transmit their data to may not be trusted or may be compromised by attackers [59]. Therefore, an adversary could make the association between the sensed data and specific participants. Reference [59] presented a mechanism called privacy assurance system in mobile sensing network in which users form a social network through social interaction based on direct personal contact or by recommendation based on common interests or location; in this latter case, the server returns a list of subscribers and the users choose some of them to send friendship requests through the server. Finally, the sensed data are passed to the mobile network multiple times before uploading them to the server, anonymizing the original owner of each piece of data. Therefore, the ownership of data and location of the participant privacy are preserved.

Approaches based on social networks are distributed since there is no a central entity that needs to know the participants' locations. These schemes only protect against external adversary attacks because the participants are required to send their actual locations to other participants in the system who might be adversaries. The IL is low since they do not change the actual locations of the users. These mechanisms require a significant amount of additional communication and computations since participants need to send the data to other participants in the system and need to process and forward incoming data from other participants as well. Therefore, the energy consumption is also high.

B. Attribute Modification Mechanisms: Obfuscation

Obfuscation-based techniques aim to protect the association between a participant and her private data by transforming the user's private information, usually the participant's location, without considering other users' data. The key idea is to modify the private information in such a way that an adversary could not infer the actual value. In this case, differently from the anonymization-based techniques, it is less important to know the data of other users in order to provide privacy protection [41]. As a result, these mechanisms are applied in the mobile devices without contacting another entity of the system.

1) Points of Interest: Privacy-preserving mechanisms based on points of interest (POI) define spatial locations where specific information need to be captured by the system. The work presented in [30] uses a centralized scheme in which an AS receives the actual data from the mobile node in which each record corresponds to data sensed in a POI. The AS runs a k-anonymity algorithm and reports the anonymized data to the application server. During the anonymization process, the very first actual POI is anonymized with any $k-1$ out of remaining $n-1$ POIs. The key of this approach is how to select, from the second iteration and onward, the $k-1$ POIs that can lead to significant revelation toward the actual data of the POIs. The solution proposed consists of a greedy algorithm that computes every combination of records that comply with the k requirement and selects the combination with the maximum match of deduced data against the actual one. The main problem of this approach is the high computational complexity; according to the authors, this is a $O(N^N)$ algorithm. To avoid the high cost, the authors proposed a division of the area of interest into subregions with $N \leq 7$, where N is the number of POI in each subregion.

The work presented in [23], also defines a set of POIs where the system needs data from. Each POI has an influence region where the variable of interest is considered to be constant. The location obfuscation mechanism of the system divides the area of interest into a set of regions called cells (i.e., Voronoi cell), each one centered on a POI according to the requirements of the application. Then, each participant receives the sensing map, consisting on the distribution of the POI. Finally, during the reporting process, each participant modifies each record changing the actual location to the closest POI and reports the modified record to the system. Fig. 4 shows an example of obfuscation using the POI approach. As shown in the figure, actual locations L_i are changed to the closest POI P_j; for instance, locations $L1$ and $L2$ are obfuscated to $P1$ while $L3$ and $L4$ are obfuscated to $P2$.

This latter approach is classified as distributed since there is no a central entity that knows the locations of the participants. The IL is medium since this technique modifies the locations of the participants according to the distribution of the POIs. This scheme protects against external and internal attacks because participants do not reveal their actual location to the system. The computational complexity is considered medium because the selection of the closest POI for each record is performed by a $O(N^2)$ algorithm. The communication overhead is low because this scheme does not require of additional transmissions from the mobile device. Thus, the energy consumption is also considered low.

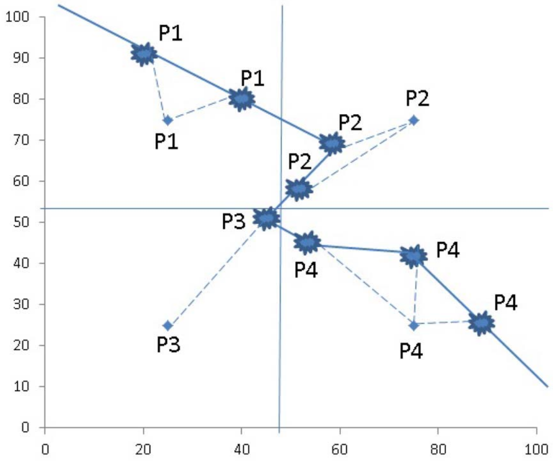

Fig. 4. Example of obfuscation using POI.

2) Position Sharing Approach: In this approach, presented in [60], users split up their precise position into position shares of limited precision. The obfuscated shares are distributed among a set of location servers where each share is a vector, such that the concatenation of all vectors is pointing to the exact user coordinates. Therefore, a subset of vectors gives an obfuscated position where every added refinement share should increase the precision by a well-defined value. This approach is based on the fact that a compromised location server will only reveal position information of strictly limited precision. In this case, the basic privacy guarantees depend on the size of the obfuscation area obtained after the fusion of k shares; then, if an attacker knows the applied share generation algorithm, he can derive additional information from the obtained k shares. In order to provide privacy guarantees, the probabilistic increase of precision must be pre-estimated for any k. Consequently, an attacker cannot derive information of higher precision that is intended by the corresponding share generation algorithm. However, its main drawback is the significantly additional transmission overhead because the mobile device needs to transmit each share to a different server.

Position sharing approaches are distributed since there is no a central entity knowing the actual locations of the participants. The IL is low because the schemes do not change the actual location of the participants. These schemes only protect against external attacks since an internal adversary accessing all the servers could identify a particular participant based on historical records. The computational complexity is low because this scheme does not require significant additional processing. However, its message complexity is significant since participants need to send several obfuscated shares to different servers. Therefore, the energy consumption of this scheme is considered high.

3) Random Data Perturbation: This technique, proposed in [61], modifies the original data set with random noise drawn from a known distribution. For example, a Gaussian distribution could be used to change the actual location of the user. In the central entity of the system, the perturbed data is used to reconstruct the original distribution using an iterative algorithm based on the *Bayes theorem* [62]. Ganti *et al.* [63] presented a novel data perturbation mechanism that generates a noise model with similar characteristics to the phenomenon being monitored by the system. Then, this model is distributed to all the participants so they can generate the noise locally. Moreover, the participants are requested to modify the configuration parameters regularly. The main drawback of this approach is that independent noise is insufficient to prevent adversaries from reconstructing the original data [64]. Further, mechanisms like the one presented in [63], which use global noise models, require an initial knowledge about the actual data, which is not always available.

Some implementations of random data perturbation are considered distributed, as there is no central entity knowing the locations of the participants and the generalization is performed locally in the mobile devices. However, other implementations use an anonymizer server, which receives the actual location from all the participants. In both cases, the IL depends on the distribution and the parameters used to perturb the data. On average, it is considered of a medium level because low perturbation means no privacy and high perturbation may render useless results at the application level. Random perturbation is classified as a def-aware method since an adversary could know that this mechanism has been used, but he is not able to discover the sensitive data of the participants. The computational complexity is medium because it requires additional processing in order to generate the perturbed locations. When the scheme is implemented in a distributed fashion, it does not incur in additional communication overhead; however, additional messages are required in the centralized approach. Therefore, the energy consumption of this scheme is considered medium.

4) Negative Survey: This centralized approach, proposed in [65], guarantees the privacy of the participants ensuring that mobile devices report fake data values, measurements that were not collected. Then, the central entity uses these negative samples to reconstruct a histogram of the actual values. In order to accomplish this, [65] proposed two protocols: 1) the node and 2) the base station protocols. The proposed approach divides the data range into categories, each category representing a data subrange [e.g., if the system is monitoring traffic speed, each category represents a speed range (0–9), (10–20), and so on]. Then, each node runs the node protocol, which chooses a negative data category from an array representing all the categories. This value is then transmitted to the base station. Once the base station receives the reported data, assuming it knows both the number of sensor nodes and the set of categories used by the nodes, it proceeds to restore the original frequency distribution. The main drawback of this approach is that participants need to reveal their actual locations; the scheme is rather used to anonymize environmental data such as the participants' speed. Therefore, an adversary could track a particular user in the system.

The negative survey mechanism is centralized since participants reveal their actual locations to the system. The IL is considered low because the variable is modified by the participant reporting a range instead of the actual value. However, the system should define the correct range size in order to guarantee a low IL, otherwise the IL could reach a high level [65]. Also, an internal adversary could infer the value of the variable for a particular user. Therefore, it is considered that protects only against external adversaries. The computational complexity is low because this scheme does not require additional processing in the mobile device. However, it requires of additional messages from the mobile device to the server, then the communication overhead and the energy consumption are considered medium.

VII. Privacy-Preserving Mechanisms for the Reporting Process: Nonattribute Modification Mechanisms

This section presents some of the most relevant privacy-preserving mechanisms that do not involve attribute transformation. These mechanisms rely on encryption or no-encryption techniques.

A. Encryption-Based Mechanisms

Mechanisms in this category rely on cryptographic methods for data transmission and/or storage [66]. An important aspect of these mechanisms is that although running encryption algorithms is possible in mobile devices, they are associated with high energy consumption and management issues related to key propagation strategies [67]. The following are the most important approaches for CS.

1) Group Signature: The objective of group signature is to guarantee the anonymity of the users and the integrity for the sensed data [17]. To achieve these goals, the system uses an encryption model based on group certificates. In this approach, a *registration authority* sends a certificate to registered users in the system. When the user sends the sensed data, he uses the certificate for encrypting and sending the data [38]. Then, the *report server* receives the encrypted data but it cannot identify the actual user because of the *group signature*. In order to guarantee the integrity of the data, the report service only accepts reports signed with the appropriated certificate; signatures do not reveal which mobile device produced them.

This is a distributed approach since there is no central entity knowing the location of the participants. The IL is considered low since this technique does not change the actual location of the users. Additionally, it is considered to be def-aware since an adversary needs the key in order to decrypt the messages and access the actual data of the participants. However, an internal adversary could have access to the participants' data, therefore it is classified as protecting against external attacks. The computational complexity of this scheme is high due to the encryption mechanism during the reporting process. Additionally, the communication overhead is considered high because the need to establish a secure channel between mobile devices and the report server. Thus, the energy consumption of this scheme is high.

2) Double Encryption: This approach, presented in [68], uses two servers: 1) an identification proxy and 2) an application server. The mobile application encrypts the sensed data and uses the public key of the identification proxy to encrypt its identification data. The ID proxy validates the signature of the sender to guarantee the integrity of the data; however, it does not have the capability to decrypt the sensed data because the application server is the one who has the private key for the data. Using this scheme, the system has a double protection layer against an adversary and therefore, the probability that an adversary could compromise both the ID proxy and the application server is very low. The tradeoff of this approach is the extra processing and transmission overhead associated with the encryption techniques, which is significant for mobile devices.

The double encryption mechanism is considered distributed because there is no central entity knowing the locations of the participants. The IL is considered low since this technique does not modify the actual locations of the users. Additionally, it is considered to be def-aware since an adversary needs two keys in order to decrypt the messages and obtain access to the actual participants' data. However, a system administrator could have access to the participants' data, so then it only protects against external attacks. The computational complexity of this scheme is high due to the encryption during the reporting process. Additionally, the communication overhead is considered high because the need to establish a secure channel between mobile devices and the report server. Thus, the energy consumption of this scheme is also high.

B. No Encryption-Based Mechanisms

This category contains privacy-preserving methods that do not change the attributes but do not make use of cryptographic mechanisms.

1) Selective Sensing: These distributed techniques are based on the willingness of the participants to activate/deactivate sensors [69]. According to [20], the basic implementation of this scheme is a binary one in which the sensor is completely activated or deactivated. Therefore, the participant decides to protect her sensitive data by not participating in the campaign when the sensor is completely deactivated. On the other hand, several schemes allow that participants decide selectively the sensor measurement granularity depending on several factors such as current location and time [70]. In this way, the privacy of the participant is protected, because each participant may select what, when, and where to sense and report the sensed data to the system. The participant can select which locations, moments, and data are considered sensitive and decide not to sense and report these data to the system, preserving the privacy. However, their main drawback is the lack of accuracy given the data suppression of the reports.

The IL of selective sensing is considered high because this techniques is based on suppression, thus some valuable data may be missed. An internal adversary could infer the missing values using the reported ones, then it only protects against external adversaries. The computational complexity is medium because it requires additional processing for selecting the data to be transmitted. Since this scheme does not require any extra communication from the mobile device, its communication overhead is considered low and the energy consumption is considered low too.

2) Mix Networks and Zones: A *mix network* is a store and forward network that offers anonymous communication facilities [71]. These networks contain message routing nodes alongside *mix nodes*. Reference [72] defined a *mix node* as a node that collects *n* equal size packets as input and reorders them before forwarding. A *mix zone* is defined as a connected spatial region of maximum size in which none of a group of users has registered any application callback. An *application zone* is defined as an area where a user has registered for a callback. This paper defines a middleware system for user anonymization based on mix zones. In this system, zones can be defined *a priori* or calculated for a group of users. Therefore, during the reporting process, users belonging to the same group send their messages to their application zone, then these messages are mixed across their mixed zone preserving the privacy of the participants. However, if a mix zone has a diameter much larger than the distance the user can cover during a location update period, the system cannot mix the users adequately [73], [74].

Mix networks and zones work in a distributed manner since there is no a central entity that knows the locations of the users. The IL is considered low because the locations of the participants are not modified during the process. The adversary may know that this scheme is being applied but she needs to intercept the transmissions of all mix nodes in order to identify a particular participant and associate her with her sensitive data. Therefore, this scheme protects against external attacks since a system administrator can gain access to all nodes in the mix network. The computational complexity is low because this scheme does not require of additional processing in the mobile device. The communication overhead and the energy consumption are both considered low for this scheme.

3) Path Cloaking Algorithm: This approach, presented in [75], suggests that the sampling frequency used by the participants to send position updates should be limited to large intervals. The algorithms based on this approach suppose that path suppression in high density areas increases the chance of confusing or mixing several different traces. According to [76], in traffic sensing applications, paths of individual vehicles can be reconstructed from a mix of anonymous samples belonging to several vehicles because consecutive location samples from a vehicle exhibit temporal and spatial correlation. Therefore, these algorithms lack of privacy in low density areas where an adversary could identify a particular participant. In order to solve that, [77] presents a scheme based on the time to confusion metric (i.e., the time between two sample points where an adversary could not determine the next sampling location with enough certainty from a set of records). The main idea is that each participant reveals her location only if the time from her previous report is less than the time to confusion value

TABLE IV
SUMMARY OF PRIVACY-PRESERVING MECHANISMS FOR THE REPORTING PROCESS

Approach	Type of system	Information loss	Knowledge of defense	Protection from internal /external attacks	Computational complexity	Communication overhead	Energy consumption
Attribute modification A. Anonymization							
1. Tessellation	Centralized	High	Def-aware	External	Low	Medium	Medium
2. Microaggregation	Centralized/Distributed	Medium	Def-aware	External	Low	Medium	Medium
3. l-diversity	Centralized	Medium	Def-aware	External	Low/Medium	Medium	Medium
4. Data aggregation	Distributed	Low	Def-aware	Internal/External	High	High	High
5. Partial-inclusivity and Range-independence	Distributed	Low	Def-aware	External	High	High	High
6. Social networks for preserving privacy	Distributed	Low	Def-aware	External	Medium	High	High
B. Obfuscation							
1. Points of interest	Centralized/Distributed	Medium	Def-aware	Internal/External	Medium	Low	Low
2. Position sharing approach	Distributed	Low	Def-aware	External	Medium	High	High
3. Random data perturbation	Centralized/Distributed	Medium	Def-aware	External	Medium	Medium	Medium
4. Negative survey	Centralized	Medium	Def-unaware	External	Low	Medium	Medium
No attribute modification A. Encryption							
1. Group signature	Distributed	Low	Def-aware	External	High	High	High
2. Double Encryption	Distributed	Low	Def-aware	External	High	High	High
B. No encryption							
1. Selective sensing	Distributed	High	Def-aware	External	Medium	Low	Low
2. Mix networks	Distributed	Low	Def-aware	External	Low	Low	Low
3. Path cloaking algorithm	Centralized	High	Def-aware	External	Low	Low	Low

defined by the system. For example, if a participant reports her location in a t_p, then an adversary could identify the records of that participant looking for records with a time variation of t_p. Using the time to confusion metric, the participant will report data in different time windows. Therefore, an adversary could not identify the records of a particular participant with certainty.

Additionally, [76] presents a modification to the previous approach that provides adequate privacy guarantees under the reacquisition tracking model. This reacquisition model defines that an adversary skips samples with high confusion in order to reacquire the correct trace even after a point of confusion. Therefore, the new algorithm computes the *confusionTimeout* from every prior released location in addition to the time from the last released position, then samples can only be released if all these confusion values are above the *confusion threshold*. Moreover, the algorithm ensures that every sample must maintain confusion to any samples that are released during the last γ minutes and before the *confusionTimeout* was reset.

The path cloaking algorithm is centralized because it requires an anonymizer to know the locations of the participants to determine the samples to be reported. The IL is considered high because this technique is based on data suppression. This scheme only protects against external attacks since an internal adversary, with access to the anonymizer, could identify a particular participant based on historical records. The computational complexity is low because all the processes occur in the anonymizer server and no privacy process is performed in the mobile device. The communication overhead is low as well because the mobile device reports their data to the anonymizer server and no additional transmissions are required. Therefore, the energy consumption of this privacy mechanism is considered low as well.

Table IV summarizes the mechanisms presented in this section along with their qualitative evaluation according to the design considerations described in Section III.

VIII. OPEN RESEARCH CHALLENGES

This section presents some of the most important open problems and challenges in privacy-preserving mechanisms for CS.

A. In Privacy-Preserving Mechanisms for the Tasking Process

The main purpose of privacy-preserving mechanisms for the tasking process is to guarantee that users can be tasked anonymously [37]. Although several models have been presented, as described in Section V, some methods such as the anonymous authentication and the direct tasking without revealing location lack the capability of predicting the number of users visiting a particular region. Others, such as the ones based on attribute-based authentication allow an adversary with additional knowledge to track the user to certain groups, allowing the identification of a particular participant in a group. Therefore, privacy mechanisms for the tasking process offer good privacy protection or guarantee good data quality but not both at the same time. The challenge therefore is how to protect the participants privacy while guaranteeing a good data quality during the tasking process.

B. In Privacy-Preserving Mechanisms for the Reporting Process

The main challenge associated with the reporting process is to guarantee a good data accuracy because most applications depend on the quality of the reported data. Since most privacy mechanisms modify the participants data in order to protect their privacy, they introduce certain noise to the system that affects the accuracy of the prediction/inference of the system. Based on Table IV, privacy-preserving mechanisms based on encryption techniques offer the best data accuracy since they do not modify the data at all. However, given the computational complexity of the encryption algorithms and the extra communication overhead that they introduce, the applicability of these mechanisms is constrained by the limited amount of resources available in mobile devices. The challenge is then how to guarantee a low IL and a low energy consumption at the same time. An example of a possible solution was presented in [23] where a probabilistic model is used to tradeoff accuracy and power consumption according to the application needs using POIs, to reduce power consumption, and double encryption, to increase data accuracy. However, this scheme does not consider the density of the population in the cells of the system. For instance, if the POI approach is used in low density cells, an adversary could easily identify a specific participant, jeopardizing her privacy. Therefore, balancing data accuracy and energy consumption is still an important open problem.

C. In the Selection of the Appropriate Privacy-Preserving Mechanism

There are different criteria, methods, and metrics that can be used to evaluate the performance of privacy-preserving mechanisms. Nonetheless, it would be good if we had a universal metric and methodology that we could use to compare and evaluate these systems and be able to choose the most appropriate one according to the application and users needs. This methodology should include benchmarks for the performance evaluation of privacy mechanisms in terms of energy consumption, running time, and communication overhead as well as probability that an adversary could identify a particular participant in the system.

IX. Conclusion

This paper presents the state-of-the-art of privacy-preserving mechanisms for CS systems. A three level taxonomy is presented to classify the mechanisms. In the first level, privacy-preserving mechanisms are divided into mechanisms for the tasking or the reporting process depending on which process they are designed to work on. The main characteristic of privacy mechanisms for the tasking process is the anonymous tasking or tasking the participants without revealing their location or identity. On the other hand, the mechanisms for the reporting process are divided into attribute modification and nonattribute modification mechanisms depending on whether they modify or not the attribute to protect. In the first case, privacy mechanisms modify the actual data via anonymization in which the participants' data are generalized to a group of users, or obfuscation, in which the participants' data are modified without considering the data from other participants. In nonattribute modification schemes, the participants privacy is preserved using encryption or no encryption techniques such as selective sensing, in which the participants decide what/where/when to report the data to the system.

In addition, this paper includes a qualitative evaluation of the mechanisms based on several metrics and important aspects, such as the threat model, IL, computational complexity, communication overhead, and energy consumption associated with each mechanism. The main conclusion is the orthogonal relationship between data quality, privacy, and resource consumption since privacy mechanisms with a good quality performance and low IL, such as encryption-based techniques, demand a high amount of resources, which are limited on mobile devices.

References

[1] J. Burke *et al.*, "Participatory sensing," in *Proc. Center Embedded Netw. Sens.*, Boulder, CO, USA, 2006, pp. 1–5.

[2] D. L. Estrin, "Participatory sensing: Applications and architecture," in *Proc. 8th Int. Conf. Mobile Syst. Appl. Services (MobiSys)*, San Francisco, CA, USA, Jun. 2010, pp. 3–4.

[3] B. Guo, F. Calabrese, E. Miluzzo, and M. Musolesi, "Mobile crowd sensing: Part 1," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 20–21, Aug. 2014.

[4] B. Guo, F. Calabrese, E. Miluzzo, and M. Musolesi, "Mobile crowd sensing: Part 2," *IEEE Commun. Mag.*, vol. 52, no. 10, pp. 76–77, Oct. 2014.

[5] Z. Tang, Y. Zhou, H. Yu, Y. Gu, T. Liu, "Developing an interactive mobile volunteered geographic information platform to integrate environmental big data and citizen science in urban management," in *Proc. Workshop Bit Data Urban Informat.*, 2014, pp. 33–48.

[6] Waze Ltd. *Waze Outsmarting Traffic Together*. [Online]. Available: http://www.waze.com

[7] S. Vhaduri, A. Ali, M. Sharmin, K. Hovsepian, and S. Kumar, "Estimating drivers' stress from GPS traces," in *Proc. 6th Int. Conf. Autom. User Interfaces Interact. Veh. Appl.*, Seattle, WA, USA, 2014, pp. 1–8.

[8] L. G. Jaimes, I. J. Vergara-Laurens, and A. Raij, "A survey of incentive techniques for mobile crowd sensing," *IEEE Internet Things J.*, vol. 2, no. 5, pp. 370–380, Oct. 2015.

[9] H. Gao *et al.*, "A survey of incentive mechanisms for participatory sensing," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 2, pp. 918–943, 2nd Quart. 2015.

[10] L. G. Jaimes, I. Vergara-Laurens, and M. A. Labrador, "A location-based incentive mechanism for participatory sensing systems with budget constraints," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PERCOM)*, Lugano, Switzerland, Mar. 2012, pp. 103–108.

[11] R. Kawajiri, M. Shimosaka, and H. Kashima, "Steered crowdsensing: Incentive design towards quality-oriented place-centric crowdsensing," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, Seattle, WA, USA, 2014, pp. 691–701.

[12] I. Koutsopoulos, "Optimal incentive-driven design of participatory sensing systems," in *Proc. Infocom*, Turin, Italy, 2013, pp. 1402–1410.

[13] D. Christin, "Privacy in mobile participatory sensing: Current trends and future challenges," *J. Syst. Softw.*, vol. 116, pp. 57–68, Jun. 2016.

[14] D. Christin, A. Reinhardt, S. S. Kanhere, and M. Hollick, "A survey on privacy in mobile participatory sensing applications," *J. Syst. Softw.*, vol. 84, no. 11, pp. 1928–1946, 2011.

[15] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw.*, Istanbul, Turkey, 2012, pp. 173–184.

[16] D. M. Konidala, R. H. Deng, Y. Li, H. C. Lau, and S. E. Fienberg, "Anonymous authentication of visitors for mobile crowd sensing at amusement parks," in *Information Security Practice and Experience*. Heidelberg, Germany: Springer, 2013, pp. 174–188.

[17] A. Kapadia, N. Triandopoulos, C. Cornelius, D. Peebles, and D. Kotz, "Anonysense: Opportunistic and privacy-preserving context collection," in *Proc. 6th Int. Conf. Mobile Syst. Appl. Services (MobiSys)*, Sydney, NSW, Australia, 2008, pp. 280–297.

[18] L. Pournajaf, L. Xiong, and V. Sunderam, "Dynamic data driven crowd sensing task assignment," *Proc. Comput. Sci.*, vol. 29, pp. 1314–1323, Dec. 2014.

[19] L. Pournajaf, L. Xiong, D. A. Garcia-Ulloa, and V. Sunderam, "A survey on privacy in mobile crowd sensing task management," Dept. Math. Comput. Sci., Emory Univ., Atlanta, GA, USA, Tech. Rep. TR-2014-002, 2014.

[20] T. Das, P. Mohan, V. N. Padmanabhan, R. Ramjee, and A. Sharma, "PRISM: Platform for remote sensing using smartphones," in *Proc. 8th Int. Conf. Mobile Syst. Appl. Services (MobiSys)*, San Francisco, CA, USA, Jun. 2010, pp. 63–76.

[21] S. B. Eisenman, N. D. Lane, and A. T. Campbell, "Techniques for improving opportunistic sensor networking performance," in *Distributed Computing in Sensor Systems* (LNCS 5067). Heidelberg, Germany: Springer, 2008, pp. 157–175.

[22] H. Lu, N. D. Lane, S. B. Eisenman, and A. T. Campbell, "Bubble-sensing: A new paradigm for binding a sensing task to the physical world using mobile phones," in *Proc. Int. Workshop Mobile Devices Urban Sens. (MODUS)*, St. Louis, MO, USA, Apr. 2008, pp. 1–8.

[23] I. J. Vergara-Laurens and M. A. Labrador, "Preserving privacy while reducing power consumption and information loss in LBS and participatory sensing applications," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM Workshops)*, Houston, TX, USA, Dec. 2011, pp. 1247–1252.

[24] H. Ji, L. Xie, C. Wang, Y. Yin, and S. Lu, "CrowdSensing: A crowd-sourcing based indoor navigation using RFID-based delay tolerant network," *J. Netw. Comput. Appl.*, vol. 52, pp. 79–89, Jun. 2015.

[25] 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*. Heidelberg, Germany: Springer, 2012, pp. 93–106.

[26] M. Wernke, P. Skvortsov, F. Dürr, and K. Rothermel, "A classification of location privacy attacks and approaches," *Pers. Ubiquitous Comput.*, vol. 18, no. 1, pp. 163–175, 2014.

[27] C. Bettini, S. Mascetti, X. Sean-Wang, and D. Freni, "Spatial generalization algorithms for LBS privacy preservation," *J. Location Based Services*, vol. 1, no. 3, pp. 179–207, 2007.

[28] C. Bettini, S. Mascetti, X. Sean-Wang, D. Freni, and S. Jajodia, "Anonymity and historical-anonymity in location-based services," in *Privacy in Location-Based Applications* (LNCS 5599). Heidelberg, Germany: Springer-Verlag, 2009, pp. 1–30.

[29] F. Bonchi and E. Ferrari, *Privacy-Aware Knowledge Discovery: Novel Applications and New Techniques*. Boca Raton, FL, USA: CRC Press, 2010.

[30] M. Murshed, T. Sabrina, A. Iqbal, and K. M. Alam, "A novel anonymization technique to trade off location privacy and data integrity in participatory sensing systems," in *Proc. 4th Int. Conf. Netw. Syst. Security (NSS)*, Melbourne, VIC, Australia, Sep. 2010, pp. 345–350.

[31] J. Xu, J. Xiang, and D. Yang, "Incentive mechanisms for time window dependent tasks in mobile crowdsensing," *IEEE Trans. Wireless Commun.*, vol. 14, no. 11, pp. 6353–6364, Nov. 2015.

[32] L. G. Jaimes, I. Vergara-Laurens, and A. Raij, "A crowd sensing incentive algorithm for data collection for consecutive time slot problems," in *Proc. IEEE Latin-America Conf. Commun. (LATINCOM)*, 2014, pp. 1–5.

[33] M.-R. Ra, B. Liu, T. F. La Porta, and R. Govindan, "Medusa: A programming framework for crowd-sensing applications," in *Proc. 10th Int. Conf. Mobile Syst. Appl. Services*, Ambleside, U.K., 2012, pp. 337–350.

[34] I. J. Vergara-Laurens, D. Mendez, and M. A. Labrador, "Privacy, quality of information, and energy consumption in participatory sensing systems," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PERCOM)*, Budapest, Hungary, Mar. 2014, pp. 199–207.

[35] H. Xiong, D. Zhang, L. Wang, and H. Chaouchi, "EMC3: Energy-efficient data transfer in mobile crowdsensing under full coverage constraint," *IEEE Trans. Mobile Comput.*, vol. 14, no. 7, pp. 1355–1368, Jul. 2015.

[36] I. Krontiris and T. Dimitriou, "Privacy-respecting discovery of data providers in crowd-sensing applications," in *Proc. IEEE Int. Conf. Distrib. Comput. Sensor Syst. (DCOSS)*, Cambridge, MA, USA, 2013, pp. 249–257.

[37] A. Kapadia, D. Kotz, and N. Triandopoulos, "Opportunistic sensing: Security challenges for the new paradigm," in *Proc. 1st Int. Workshop Commun. Syst. Netw. (COMSNETS)*, Bengaluru, India, Jan. 2009, pp. 1–10.

[38] C. Cornelius *et al.*, "Anonysense: Privacy-aware people-centric sensing," in *Proc. 6th Int. Conf. Mobile Syst. Appl. Services (MobiSys)*, Breckenridge, CO, USA, Jun. 2008, pp. 211–224.

[39] E. Brickell, J. Camenisch, and L. Chen, "Direct anonymous attestation," in *Proc. 11th ACM Conf. Comput. Commun. Security*, Washington, DC, USA, 2004, pp. 132–145.

[40] D. Chaum, "Blind signatures for untraceable payments," in *Advances in Cryptology*. Boston, MA, USA: Springer US, 1983, pp. 199–203.

[41] I. J. Vergara, "A hybrid privacy-preserving mechanism for participatory sensing systems," Ph.D. dissertation, Dept. Comput. Sci., Univ. South Florida, Tampa, FL, USA, 2014.

[42] J. Al-Muhtadi, R. Campbell, A. Kapadia, M. D. Mickunas, and S. Yi, "Routing through the mist: Privacy preserving communication in ubiquitous computing environments," in *Proc. 22nd Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Vienna, Austria, Jul. 2002, pp. 74–83.

[43] M. Gruteser and D. Grunwald, "Enhancing location privacy in wireless LAN through disposable interface identifiers: A quantitative analysis," *Mobile Netw. Appl.*, vol. 10, no. 3, pp. 315–325, Jun. 2005.

[44] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *Proc. 13th Conf. USENIX Security Symp.*, Berkeley, CA, USA, 2004, p. 21.

[45] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, "Preventing location-based identity inference in anonymous spatial queries," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 12, pp. 1719–1733, Dec. 2007.

[46] K. Wang, B. C. M. Fung, and P. S. Yu, "Handicapping attacker's confidence: An alternative to k-anonymization," *Knowl. Inf. Syst.*, vol. 11, no. 3, pp. 345–368, Apr. 2007.

[47] R. Mortazavi, S. Jalili, and H. Gohargazi, "Multivariate microaggregation by iterative optimization," *Appl. Intell.*, vol. 39, no. 3, pp. 529–544, 2013.

[48] D. Sánchez, J. Domingo-Ferrer, S. Martínez, and J. Soria-Comas, "Utility-preserving differentially private data releases via individual ranking microaggregation," *Inf. Fusion*, vol. 30, pp. 1–14, Jul. 2016.

[49] Q. Li and G. Cao, "Efficient privacy-preserving stream aggregation in mobile sensing with low aggregation error," in *Privacy Enhancing Technologies*. Heidelberg, Germany: Springer, 2013, pp. 60–81.

[50] J. Domingo-Ferrer and V. Torra, "Ordinal, continuous and heterogeneous k-anonymity through microaggregation," *Data Min. Knowl. Discov.*, vol. 11, no. 2, pp. 195–212, Aug. 2005.

[51] I. J. Vergara-Laurens, D. Mendez-Chaves, and M. A. Labrador, "On the interactions between privacy-preserving, incentive, and inference mechanisms in participatory sensing systems," in *Network and System Security*, (LNCS 7873). Heidelberg, Germany: Springer, 2013, pp. 614–620.

[52] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "L-diversity: Privacy beyond k-anonymity," *ACM Trans. Knowl. Discov. Data (TKDD)*, vol. 1, no. 1, Mar. 2007, Art. no. 3.

[53] K. L. Huang, S. S. Kanhere, and W. Hu, "Preserving privacy in participatory sensing systems," *Comput. Commun.*, vol. 33, no. 11, pp. 1266–1280, 2010.

[54] W. He, X. Liu, H. Nguyen, K. Nahrstedt, and T. Abdelzaher, "PDA: Privacy-preserving data aggregation in wireless sensor networks," in *Proc. 26th IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Anchorage, AK, USA, May 2007, pp. 2045–2053.

[55] C. Castelluccia, E. Mykletun, and G. Tsudik, "Efficient aggregation of encrypted data in wireless sensor networks," in *Proc. 2nd Annu. Int. Conf. Mobile Ubiquitous Syst. Netw. Services (MobiQuitous)*, San Diego, CA, USA, Jul. 2005, pp. 109–117.

[56] J. Shi, R. Zhang, Y. Liu, and Y. Zhang, "PriSense: Privacy-preserving data aggregation in people-centric urban sensing systems," in *Proc. 29th Conf. Comput. Commun. (INFOCOM)*, San Diego, CA, USA, Mar. 2010, pp. 1–9.

[57] L. Kazemi and C. Shahabi, "A privacy-aware framework for participatory sensing," *SIGKDD Explor. Newslett.*, vol. 13, no. 1, pp. 43–51, 2011.

[58] I. Krontiris and N. Maisonneuve, "Participatory sensing: The tension between social translucence and privacy," in *Trustworthy Internet*. Milan, Italy: Springer, 2011, pp. 159–170.

[59] L. Hu and C. Shahabi, "Privacy assurance in mobile sensing networks: Go beyond trusted servers," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PERCOM Workshops)*, Mannheim, Germany, 2010, pp. 613–619.

[60] F. Dürr, P. Skvortsov, and K. Rothermel, "Position sharing for location privacy in non-trusted systems," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PERCOM)*, Seattle, WA, USA, Mar. 2011, pp. 189–196.

[61] R. Agrawal and R. Srikant, "Privacy-preserving data mining," *ACM SIGMOD Rec.*, vol. 29, no. 2, pp. 439–450, 2000.

[62] Y. Zhu and R. Sivakumar, "Challenges: Communication through silence in wireless sensor networks," in *Proc. 11th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, Cologne, Germany, 2005, pp. 140–147.

[63] R. K. Ganti, N. Pham, Y. E. Tsai, and T. F. Abdelzaher, "Poolview: Stream privacy for grassroots participatory sensing," in *Proc. 6th ACM Conf. Embedded Netw. Sensor Syst.*, Raleigh, NC, USA, Nov. 2008, pp. 281–294.

[64] J. Krumm, "Inference attacks on location tracks," in *Pervasive Computing* (LNCS 4480). Heidelberg, Germany: Springer, 2007, pp. 127–143.

[65] J. Horey, M. M. Groat, S. Forrest, and F. Esponda, “Anonymous data collection in sensor networks,” in *Proc. 4th Annu. Int. Conf. Mobile Ubiquitous Syst. Netw. Services (MobiQuitous)*, Philadelphia, PA, USA, Aug. 2007, pp. 1–8.

[66] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, “Private queries in location based services: Anonymizers are not necessary,” in *Proc. ACM Int. Conf. Manag. Data (SIGMOD)*, Vancouver, BC, Canada, Jun. 2008, pp. 121–132.

[67] L. Eschenauer and V. D. Gligor, “A key-management scheme for distributed sensor networks,” in *Proc. 9th ACM Conf. Comput. Commun. Security (CCS)*, Washington, DC, USA, 2002, pp. 41–47.

[68] B. Hoh *et al.*, “Virtual trip lines for distributed privacy-preserving traffic monitoring,” in *Proc. Int. Conf. Mobile Syst. Appl. Services (MobiSys)*, Breckenridge, CO, USA, Jun. 2008, pp. 15–28.

[69] K. Shilton, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, “Participatory privacy in urban sensing,” in *Proc. Int. Workshop Mobile Devices Urban Sensing (MODUS)*, St. Louis, MO, USA, 2008, pp. 1–7.

[70] B. N. Schilit *et al.*, “Challenge: Ubiquitous location-aware computing and the ‘Place Lab’ initiative,” in *Proc. 1st ACM Int. Workshop Wireless Mobile Appl. Services WLAN Hotspots (WMASH)*, San Diego, CA, USA, Sep. 2003, pp. 29–35.

[71] D. L. Chaum, “Untraceable electronic mail, return addresses, and digital pseudonyms,” *Commun. ACM*, vol. 24, no. 2, pp. 84–90, Feb. 1981.

[72] A. R. Beresford and F. Stajano, “Location privacy in pervasive computing,” *IEEE Pervasive Comput.*, vol. 2, no. 1, pp. 46–55, Jan./Mar. 2003.

[73] B. Agir, T. G. Papaioannou, R. Narendula, K. Aberer, and J.-P. Hubaux, “User-side adaptive protection of location privacy in participatory sensing,” *GeoInformatica*, vol. 18, no. 1, pp. 165–191, 2014.

[74] X. Chen, X. Wu, X.-Y. Li, Y. He, and Y. Liu, “Privacy-preserving high-quality map generation with participatory sensing,” in *Proc. IEEE INFOCOM*, Toronto, ON, Canada, 2014, pp. 2310–2318.

[75] B. Zan, P. Hao, M. Gruteser, and X. Ban, “VTL zone-aware path cloaking algorithm,” in *Proc. 14th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Washington, DC, USA, 2011, pp. 1525–1530.

[76] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady, “Preserving privacy in GPS traces via uncertainty-aware path cloaking,” in *Proc. 14th ACM Conf. Comput. Commun. Security (CCS)*, Alexandria, VA, USA, 2007, pp. 161–171.

[77] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady, “Achieving guaranteed anonymity in GPS traces via uncertainty-aware path cloaking,” *IEEE Trans. Mobile Comput.*, vol. 9, no. 8, pp. 1089–1107, Aug. 2010.

Idalides J. Vergara-Laurens (M’09) received the B.S. degree in computer system engineering from the Universidad Industrial de Santander, Bucaramanga, Colombia, in 1999, the M.S. degree in computer engineering from the University of Puerto Rico at Mayagüez, Mayagüez, PR, USA, in 2005, and the Ph.D. degree in computer science and engineering from the University of South Florida, Tampa, FL, USA, in 2014.

He is an Associate Professor with the Universidad del Turabo, Gurabo, PR, USA. His current research interests include crowd sensing, security, privacy, and green networking.

Luis G. Jaimes (GSM’13) received the B.S. degree in mathematics from the Universidad Industrial de Santander, Bucaramanga, Colombia, in 1999, the M.S. degree in scientific computing from the University of Puerto Rico at Mayagüez, Mayagüez, PR, USA, in 2004, and the M.S. degree in computer science and Ph.D. degree in electrical engineering from the University of South Florida, Tampa, FL, USA, in 2012 and 2015, respectively.

He is an Assistant Professor with Bethune-Cookman University, Daytona Beach, FL, USA. His current research interests include mHealth, economic incentives in wireless networks, data privacy, machine learning, and ubiquitous computing.

Miguel A. Labrador (S’97–A’01–M’01–SM’04) received the Ph.D. degree in information science (with a concentration in telecommunications) from the University of Pittsburgh, Pittsburgh, PA, USA, in 2000.

He is currently a Professor with the Department of Computer Science and Engineering, University of South Florida, Tampa, FL, USA. He authored *Topology Control in Wireless Sensor Networks* (Springer, 2009) and *Location-Based Information Systems* (CRC Press, 2010). His current research interests include energy efficient mechanisms for wireless sensor networks and location-based services.