A privacy-preserving cloud computing system for creating participatory noise maps

George Drosatos*, Pavlos S. Efraimidis*, Ioannis N. Athanasiadis*, Ellie D'Hondt† and Matthias Stevens†,‡

*Electrical & Computer Engineering Dept.,
Democritus University of Thrace, Xanthi, Greece
E-mail: {gdrosato, pefraimi, iathan}@ee.duth.gr

†BrusSense Team, Department of Computer Science,
Vrije Universiteit Brussel, Brussels, Belgium
E-mail: {eldhondt, mstevens}@vub.ac.be

‡ExCiteS group, Dept. of Civil, Environmental and Geomatic Engineering,
University College London, London, United Kingdom
E-mail: m.stevens@ucl.ac.uk

***Abstract*—Participatory sensing is a crowd-sourcing technique which relies both on active contribution of citizens and on their location and mobility patterns. As such, it is particularly vulnerable to privacy concerns, which may seriously hamper the large-scale adoption of participatory sensing applications. In this paper, we present a privacy-preserving system architecture for participatory sensing contexts which relies on cryptographic techniques and distributed computations in the cloud. Each individual is represented by a personal software agent, which is deployed on one of the popular commercial cloud computing services. The system enables individuals to aggregate and analyse sensor data by performing a collaborative distributed computation among multiple agents. No personal data is disclosed to anyone, including the cloud service providers. The distributed computation proceeds by having agents execute a cryptographic protocol based on a homomorphic encryption scheme in order to aggregate data. We show formally that our architecture is secure in the Honest-But-Curious model both for the users and the cloud providers.**

Our approach was implemented and validated on top of the NoiseTube system [1], [2], which enables participatory sensing of noise. In particular, we repeated several mapping experiments carried out with NoiseTube, and show that our system is able to produce identical outcomes in a privacy-preserving way. We experimented with real and simulated data, and present a live demo running on a heterogeneous set of commercial cloud providers. The results show that our approach goes beyond a proof-of-concept and can actually be deployed in a real-world setting. To the best of our knowledge this system is the first operational privacy-preserving approach for participatory sensing. While validated in terms of NoiseTube, our approach is useful in any setting where data aggregation can be performed with efficient homomorphic cryptosystems.

***Keywords*-Privacy-preserving computation; Cloud computing; Mobile sensing; Participatory sensing; Noise Mapping; Environmental monitoring; Citizen science.**

I. Introduction

Participatory sensing [3], [4] appropriates everyday devices such as mobile phones to acquire information about the physical world (and the people in it) at a granularity which is very hard to achieve otherwise. A crucial component of participatory sensing systems is *geolocation*, i.e., labelling data with geographical coordinates, as it facilitates the visualisation and management of the potentially enormous amounts of data gathered. This is particularly important in the context of *NoiseTube* [1], [2], a participatory sensing system and service[1] designed to monitor and map noise pollution. Indeed, it would be practically impossible to produce noise maps on the basis of sound level measurements, gathered quasi-continuously as contributors walk the streets, without automatic geolocation of measurements by means of GPS.

However, location traces constitute sensitive personal information. In small-scale deployments, in which individual contributors know or trust each other, the disclosure of such information may be acceptable. However, in larger-scale deployments, involving more contributors and possibly coordinated by some authority, trust relationships tend to be much weaker and contributors may be uncomfortable about the type of information that is collected, and with whom it is shared. Hence, scaling up a participatory sensing project inherently increases privacy concerns, which in turn can severely hamper the project from reaching its goals.

In this paper, we present a privacy-preserving extension of NoiseTube. Our system, called *NoiseTubePrime*, relies on privacy-preserving distributed computation in the cloud and is oriented towards coordinated noise mapping campaigns set up by citizens and/or authorities. NoiseTubePrime differs from earlier work [5], [6], [7] on privacy-preserving mobile sensing systems in that it is at the same time simple, safe, verified, and transparent to end-users.

II. The NoiseTubePrime System

In order to remedy privacy concerns when creating collective noise maps in terms of the existing NoiseTube service [1], [2], we propose a solution relying on a privacy-preserving distributed computation algorithm for generating

[1] http://www.noisetube.net

grid-based noise maps for a target area and time-frame. At the basis of this algorithm lies a privacy-preserving cryptographic protocol for secure multi-party computations [8], with as input current or archived datasets of geolocated sound level measurements gathered by multiple users. Computation is executed by software agents "living" in the cloud.

Concretely, each user is represented by a personal, cloud-based software agent which acts as a mediator. Such an agent temporarily stores encrypted user data, takes part in the generation of participatory maps on the user's behalf, while also crucially preserving his/her privacy. All data transmitted by users to NoiseTubePrime agents is encrypted. In this way we overcome privacy issues related to how the cloud service provider might treat the data. Cloud deployment ensures that agents are continuously online and have adequate computational resources. In this way users do not need to operate agents on their mobile phones or personal servers.

An architectural diagram of the NoiseTubePrime system is shown in Figure 1. A typical scenario for using it proceeds as follows. Suppose a particular entity, be it an authority or a citizens' organisation, is interested to map a local area during a time span of interest (e.g. Friday night in a pub area). The initiative taker(s) then organise a measurement campaign in which a group of citizens use the NoiseTube system to gather geolocated sound level measurements in the specific geographical region and time period. The campaign proceeds through the following steps:

(a) To collect data about noise pollution, on an individual basis or in collaboration with others, users download the NoiseTube client application for their mobile device (e.g. from the Android Market)[2]. By default measurement data is stored locally on each user's mobile phone. Users should set up their personal cloud agent, which should register to a *Directory Service* (DS) for the virtual network topology we deploy. Each user mandates his/her NoiseTubePrime agent to take part in existing or future campaigns, following a user-specified privacy policy.

(b) At some point in time the NoiseTube service announces a new campaign, i.e., the need for data to be collected in a certain area and period. Users are invited to participate through their agents, and agent policy may include information on how agents should respond to such requests. For instance, agents may choose to participate to campaigns based on whether their owners plan to collect data in the specific region or not, or have collected relevant data before[3]. A deadline is set for all agents interested in contributing to register via the DS.

(c) When a user agrees to join a campaign (through the mediating agent), his/her mobile device inspects the user's local dataset for measurements that satisfy the given constraints, and uses this data to generate and encrypt the contribution of the user. The encrypted contribution is then handed over to the user's NoiseTubePrime agent in the cloud, as soon as connectivity is available. This data upload operation takes place once per user and campaign/computation, and from that point on the user's mobile device is no longer involved in the computation.

(d) Each NoiseTubePrime agent manages a user's private data in the form of an encrypted map for the area of interest. Maps are encrypted with a public key that is either used across the system, is specific for the campaign in question, or for a certain time period. Agents only use the encrypted data to participate in the generation of collective maps, as allowed by user policy.

(e) After the announced deadline has passed the NoiseTube service initiates the distributed computation in the cloud. Note that agents from different users can be hosted on different cloud services. A list of the participating agents is retrieved from the DS. Agents are organized into a virtual network topology in which distributed computations take place. This may be a simple ring topology or something more sophisticated such as a tree for time-critical computations. One of the agents is selected to operate as the root-node for the specific computation via an appropriate request.

(f) The root-node coordinates a distributed computation that generates the specified noise map. This algorithm is detailed in the following Section III.

(g) When agent interactions for the distributed computation are over, the NoiseTube service receives an encrypted aggregate noise map without any trace of the personal data of individual users. The NoiseTube service, using its private key, decrypts the received data to obtain the requested noise map, which is then made available accordingly. Members of the campaign can log on to the service to visualise and explore the resulting noise maps. In this way, a user's private information is not disclosed at any stage of participatory noise mapping process.

[2]We should note that the NoiseTubePrime functionality is not yet incorporated in NoiseTube application that is currently available for download.

[3]Hence the computations may involve both past and current data.

III. The Privacy-Preserving Computation

In this section, we describe the cryptographic protocol that is implemented by the NoiseTubePrime agents for calculating noise maps in a privacy-preserving way. The communication between agents in our protocol is performed over secure sockets (SSL/TLS). The protocol is secure in the *Honest-But-Curious* (HBC) model (see Section IV). We also assume that the cloud providers are honest-but-curious, and that they do not collude with NoiseTube to reveal users' data. In any case, the later threat can be addressed by deploying a threshold decryption scheme.

A. The PrivNoiseMap Problem Definition

The main goal of our work is to generate aggregate noise maps without violating the privacy of participants.

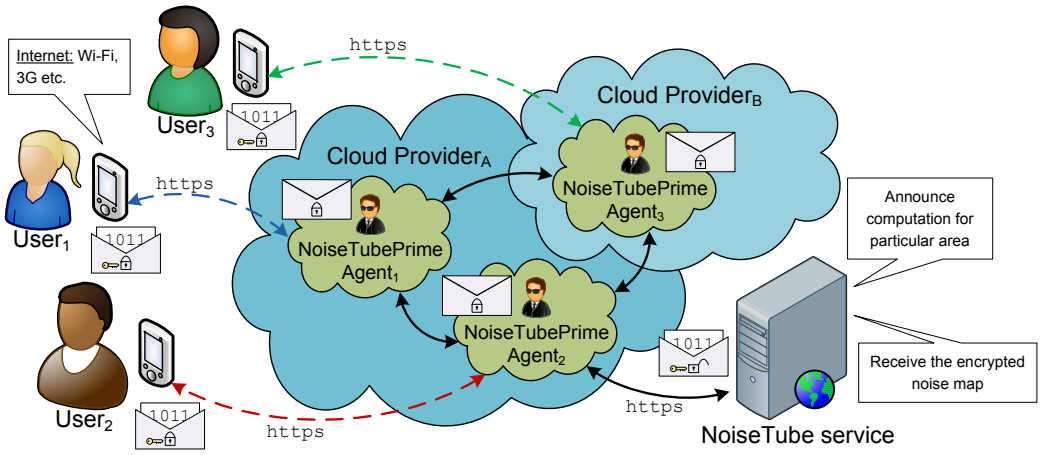

Figure 1. The general architecture of our system.

The personal data which is needed for the computation are the sound level measurements associated with the user location and time-stamp. To formalize the problem addressed in this work, we define the abstract *PrivNoiseMap* problem for the privacy-preserving computation of participatory noise maps related to a particular measurement campaign. NoiseTubePrime is then an approach and associated system that solves the PrivNoiseMap problem. An instance of PrivNoiseMap consists of:

- **N users** $u_1, u_2, \ldots, u_N$ and their geolocated, timestamped sound level measurements, where N is the number of participants that have expressed interest in the campaign.
- **Input:** The geographical area of interest (defined by minimum and maximum latitudes and longitudes) together with the cell dimensions (e.g. 20 m×20 m) of a grid covering that area, the time intervals of interest, the deadline for the distributed computation and a public encryption key.
- **Output:** The aggregate noise map with the required statistical information per grid cell. No personal data is disclosed during the computation.

B. The Distributed Protocol

We present a protocol for a privacy-preserving computation that solves the PrivNoiseMap problem. The protocol does not disclose any locations or sound level measurements of any participants; only the final aggregate noise map is revealed at the end of the computation.

Initially, the NoiseTube service announces that a specific campaign is planned. The announcement includes the campaign name, the area and time period of interest, the public encryption key and the response deadline.

When the campaign's deadline is reached each NoiseTubePrime agent, registered with the DS for that specific campaign, receives a request for the distributed computation as well as a corresponding deadline. Within the deadline, each agent communicates with the user's mobile device, and asks for any data that is relevant to the specific computation instance. The user is involved in the particular computation according to his/her privacy policy. User privacy policy can be quite sophisticated and it is not the focus of this work. In case the user participates, data relevant for the campaign is encrypted at the client side in the form of a personal aggregate map using the campaign's designated public key. The structure of each personal aggregate map is shown in Figure 2. It covers the whole geographical area of the campaign, not only the sub-area the user traversed. Each grid element corresponds to an area for which two values are computed: the number of measurements in the particular area (E_c), and the sum of measurements (E_s), in our case sound levels in dB(A). By the announced deadline, each NoiseTubePrime agent has received the encrypted personal aggregate map of it's user (as in Figure 2) in case connectivity was possible with the mobile device.

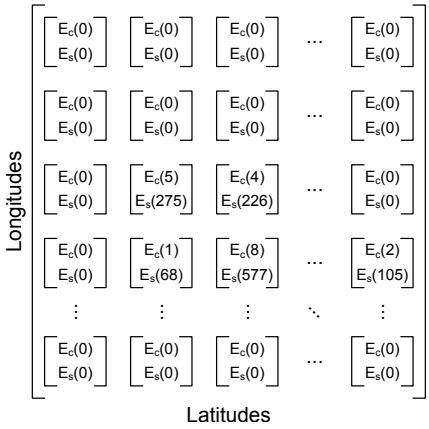

Figure 2. An example of an aggregate noise map.

When the computation deadline has been met, the distributed computation between the participating personal agents can start. Initially, the NoiseTube service selects one of the participating agents as the root-node and sends it a

request to commence the map computation. Then, the root-node agent begins the computation. The computation is performed across the agent topology, for example a simple ring topology, which provides a virtual distributed computation platform. Each agent receives the aggregate map from its predecessor, multiplies each value pair (E_s,E_c) with its own corresponding value pair, andThen the result is forwarded to the successor agent in the topology, which repeats the same steps. This computation exploits the homomorphic property of the Paillier cryptosystem, as is explained below.

At the end of the computation, the aggregate encrypted map is returned to the root-node which then forwards it to the NoiseTube service for decryption. The NoiseTube service receives the aggregate map, decrypts it with the private key, and calculates the measurement average for each grid element by dividing $D(E_s)/D(E_c)$. This produces the decrypted aggregate noise map, where for each element of the grid we have calculated the average noise value and the number of measurements that support it.

To avoid side-channel privacy leaks, a user can participate even without having data for a particular computation, by submitting a private encrypted map of zero values. In this way, not even his/her own agent is aware of the fact that the user does not have data for the particular computation. Similarly, when the mobile device cannot establish contact with the NoiseTubePrime agent, the agent may participate in the computation with a private encrypted map of zero values. In this way the agent does not need to opt out from the ring, while the final result is the same and at the same time the privacy of its owner is protected.

The appropriate network topology depends on several factors like the number of participating agents, the requirements for tolerance on network failures and the limitations on the execution time. However, in this work execution time was not critical, and our experiments with several clouds turned to be fast enough, thus we adopted a simple ring topology, and did not investigate this issue any further.

C. Cryptographic Tools

The *Paillier cryptosystem* [9] is a probabilistic asymmetric cryptographic algorithm for public key cryptography. The security of Paillier is implied by the Decisional Composite Residuosity Assumption (DCRA). In NoiseTubePrime, we use the additive homomorphic encryption property of the Paillier cryptosystem for calculating aggregate data in a privacy-preserving manner. Using homomorphic encryption one can perform a specific algebraic operation on the plaintext by performing a (possibly different) algebraic operation on the ciphertext. The additive homomorphic property of the Paillier cryptosystem, if the public key is modulo m, is shown in the following equation:

$$\mathcal{E}(x_1) \cdot \mathcal{E}(x_2) = \mathcal{E}(x_1 + x_2 \bmod m)$$

We exploit the above property to calculate the measurement sum and the number of measurements in each grid element, which, when decrypted, can be divided to calculate the average value. We note that our method would work also for other functions computable with additive operations, such as covariance or frequency distribution. Such capabilities are presented for example in [10].

IV. Security

In this subsection, we demonstrate that the proposed protocol does preserve the privacy of participants. The security holds for the HBC model both for the users and for the cloud providers. An honest-but-curious party (adversary) [11] follows the prescribed computation protocol properly, but may keep intermediate computation results, e.g. messages exchanged, and try to deduce additional information from them other than the protocol result.

In the NoiseTubePrime protocol, the information exchanged by agents is both aggregated and encrypted; thus, honest-but-curious party cannot infer any private information. The security of the Paillier cryptosystem and its homomorphic property ensures that the personal data is not disclosed and cannot be associated with any particular user. To prove the privacy attribute of the protocol, we show that it satisfies the criterion of k-anonymity [12]. In the context of this work, k-anonymity means that no less than k individual users can be associated with a particular measurement value. The NoiseTubePrime protocol offers k-anonymity in the sense that the result computed at the end of the protocol cannot be attributed to any of the N participated agents, i.e. $k = N$, even if the list of participating users is known.

To summarize, the key security features of NoiseTubePrime protocol are:

- Each NoiseTubePrime agent receives an encrypted grid from the previous node. It cannot obtain information about the contents of the map, because the ciphertexts are encrypted with Paillier encryption.
- None of the cloud providers can obtain any information about the private content stored or computed by the agents, because all data and computations are in encrypted form.
- Each node alters the ciphertexts of the computation. Even the nodes that do not have data to participate multiply the ciphertexts with an encrypted number '0', which is the neutral element of the additive homomorphic property of Paillier. Again it is impossible to detect that an agent contributed with a grid consisting only of zeros.
- At the end of the protocol, only the aggregate noise map is revealed. As a result, no individual can be associated with his/her own measurements contributed in the computation. Consequently, the proposed protocol preserves k-anonymity for $k = N$, where N is the number of all participants that took part in the computation.

V. Related Work

Privacy protection in mobile sensing systems – participatory or opportunistic [13] – has recently attracted the interest of the scientific community. Because users of a participatory sensing system play an active role in the data collection process – unlike users of an opportunistic sensing system – it has been argued that they should also be actively engaged in privacy-related decisions [14], e.g. where and when to measure and what to share which whom. It has also been argued that, in order to protect user privacy and increase their negotiating power, data collection and data sharing should be decoupled by introducing a *personal data vault* that stores a user's data in a secure manner (i.e. encrypted), from which he/she can then selectively share subsets with various services or campaigns [15]. This idea is one of the ingredients of the NoiseTubePrime system presented above.

A comprehensive approach for opportunistic sensing is presented in [5]. The area under consideration is divided into appropriate regions (tessellation procedure), which have to be sufficiently large to preserve user anonymity. The NoiseTubePrime approach is simpler and does not require any specific area division to preserve user privacy.

A very interesting related work is the PriSense system [7] which is based on a data slicing technique [16], and can offer functionality comparable to NoiseTubePrime for additive aggregation functions. However, the homomorphic encryption-based approach of NoiseTubePrime is simpler – no data scattering has to take place – and seems to be more general since homomorphic encryption is not limited to additive functions. Moreover, due to its simplicity, the NoiseTubePrime approach should be less error-prone.

In [6] the authors use advanced algorithmic techniques like sketches and approximate set cover to compute approximate statistic results in a privacy-preserving way. While theoretically interesting, in our opinion this approach is too complicated to be applied in practical, real-world settings, as opposed to NoiseTubePrime.

VI. Experimental Results

To evaluate our system, we developed a NoiseTubePrime prototype that implements the privacy-preserving protocol for calculating participatory noise maps in the cloud. The prototype can be separated into two parts. The first is the mobile application, which runs on users' devices, and the second is the NoiseTubePrime agent community, which runs on several cloud providers.

At the mobile device side, we implemented our solution on the Android platform (Android v2.2 "Froyo"), using Java. We have chosen Android but there is no reason why our solution could not be ported to other mobile application platforms (e.g. Java ME/CLDC or Apple iOS). NoiseTubePrime agents were also implemented in Java, as Java Web Servlets (WAR). They were deployed on several cloud infrastructure providers, namely Google App Engine, CloudBees, and Amazon EC2, without important differences in the implementation[4].

At the current stage of development, NoiseTubePrime agents and the Android client application do not have all functionalities that were presented in the previous sections. However, we have fully implemented and tested the protocol for the cryptographic distributed computations in a realistic setting. The communication among the NoiseTubePrime agents and between the Android client and the cloud agent is performed over `https`, which provides encrypted communication with secure sockets (SSL/TLS). Both applications implement the Paillier cryptosystem primitives for encrypting/decrypting data and performing calculations.

To demonstrate NoiseTubePrime functionality, we also implemented an online demo (http://polis.ee.duth.gr:8080/privnoise), which is publicly available for evaluation. This demo is implemented with the Google Web Toolkit and has both a client and a server side. For this purpose, we used real sound level measurements collected in July 2010 by volunteering citizens in a 1 km^2 area in the city of Antwerp, Belgium, as part of an experiment described in [17], [18], [2]. A screenshot of the demo during the execution of an experiment is shown in Figure 3. Each grid element corresponds to an area of 40 m×40 m and the key size of Paillier cryptosystem was selected to be 512 bits. A preliminary experimental evaluation confirmed the viability of our approach. For example, for a map with 35×35=1225 grid elements the execution times are approximately 21 s for each mobile device to prepare the encrypted map and 1.2 s for each agent to participate in the computation.

VII. Conclusion

This paper presents a new, privacy-preserving architecture for the creation of participatory noise maps. This architecture is called NoiseTubePrime and builds on the NoiseTube system [1], [2]. NoiseTubePrime allows aggregate noise maps to be generated from data collected by multiple users without disclosing their location traces. The resulting maps are exactly the same as those generated with conventional aggregation methods, as applied in NoiseTube. In our approach, each user is represented by a cloud-based personal software agent. The protection of privacy is achieved by using cryptographic techniques and performing a distributed computation within the network of agents. The distributed computation is performed on encrypted data and no personal data items are disclosed to anyone, including the cloud providers, at any time. Finally, we developed a prototype implementation and presented experimental results using a heterogeneous set of commercial cloud providers, confirming the viability and the efficiency of the proposed solution.

Our future plans are to develop a stable version of the NoiseTubePrime implementation and demonstrate its use

[4] *Google App Engine:* http://appengine.google.com, *CloudBees:* http://www.cloudbees.com, *Amazon EC2:* http://aws.amazon.com/ec2

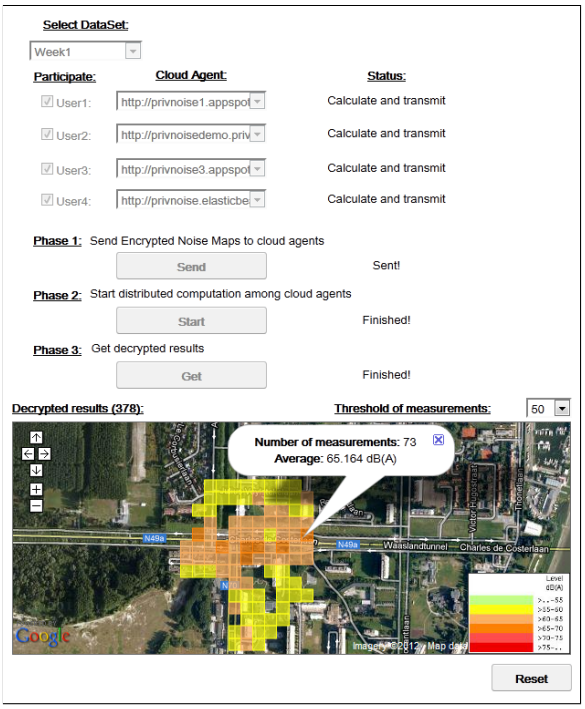

Figure 3. A screenshot of our demo.

for local campaigns, extending the platform towards more statistical parameters. Current and future NoiseTube users should be oblivious to these privacy extensions insofar as possible. We would also like to experiment with scaling up the size of campaigns, in order to investigate if our approach would hold in city-wide mapping scenarios.

Finally we want to stress that the proposed system and architecture is independent of the noise domain, and can thus potentially be applied to other participatory sensing campaigns or platforms. The only constraint is that the parameters of interest can be computed with efficient homomorphic cryptosystems.

Acknowledgments

The research of G. Drosatos and P. S. Efraimidis leading to their contribution has received funding from the E.U. FP7/2007-2013 under grant agreement no 264226: SPICE. This paper reects only the views of the authors - The Union is not liable for any use that may be made of the information contained. Ellie D'Hondt is supported by the InnovIris, the Brussels Institute for Research and Innovation.

References

[1] N. Maisonneuve, M. Stevens, and B. Ochab, "Participatory noise pollution monitoring using mobile phones," *Information Polity*, vol. 15, no. 1-2, pp. 51–71, Aug. 2010.

[2] Matthias Stevens, "Community memories for sustainable societies: The case of environmental noise," Ph.D. dissertation, Vrije Universiteit Brussel, June 2012, advised by Luc Steels and Ellie D'Hondt.

[3] J. A. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava, "Participatory sensing," in *WSW '06, held at ACM SenSys '06*, Oct. 2006.

[4] E. Paulos, "Citizen science: Enabling participatory urbanism," in *Handbook of Research on Urban Informatics: The Practice and Promise of the Real-Time City*. Information Science Reference, IGI Global, 2009, ch. 28, pp. 414–436.

[5] A. Kapadia, N. Triandopoulos, C. Cornelius, D. Peebles, and D. Kotz, "AnonySense: Opportunistic and privacy-preserving context collection," in *Pervasive '08*, ser. LNCS, vol. 5013. Springer Berlin/Heidelberg, May 2008, pp. 280–297.

[6] L. Becchetti, L. Filipponi, and A. Vitaletti, "Opportunistic privacy preserving monitoring," in *PhoneSense '10, held at ACM SenSys '10*, Nov. 2010, pp. 51–55.

[7] J. Shi, R. Zhang, Y. Liu, and Y. Zhang, "PriSense: Privacy-preserving data aggregation in people-centric urban sensing systems," in *INFOCOM '10)*. IEEE, Mar. 2010, pp. 1–9.

[8] A. C.-C. Yao, "Protocols for secure computations (extended abstract)," in *FOCS '82*. IEEE, Nov. 1982, pp. 160–164.

[9] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *EUROCRYPT '99*, ser. LNCS, vol. 1592. Springer, 1999, pp. 223–238.

[10] G. Drosatos and P. Efraimidis, "Privacy-preserving statistical analysis on ubiquitous health data," in *TrustBus '11*, ser. LNCS. Springer, 2011, vol. 6863, pp. 24–36.

[11] A. Acquisti, S. Gritzalis, C. Lambrinoudakis, and S. De Capitani di Vimercati, *Digital privacy*. Auerbach Publications, Taylor & Francis Group, 2008.

[12] V. Ciriani, S. Capitani di Vimercati, S. Foresti, and P. Samarati, "κ-anonymity," in *Secure Data Management in Decentralized Systems*, ser. Advances in Information Security. Springer, 2007, vol. 33, pp. 323–353.

[13] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *J. IEEE Commun. Magazine*, vol. 48, no. 9, pp. 140–150, 2010.

[14] K. Shilton, "Four billion little brothers?: Privacy, mobile phones, and ubiquitous data collection," *J. Commun. ACM*, vol. 52, pp. 48–53, Nov. 2009.

[15] D. Estrin, "Participatory sensing: Applications and architecture," *J. IEEE Internet Computing*, vol. 14, no. 1, pp. 12–14, 2010.

[16] W. He, X. Liu, H. Nguyen, K. Nahrstedt, and T. Abdelzaher, "PDA: Privacy-preserving data aggregation in wireless sensor networks," in *INFOCOM '07*. IEEE, 2007, pp. 2045–2053.

[17] E. D'Hondt and M. Stevens, "Participatory noise mapping," in *Pervasive '11 as demo*, June 2011, pp. 33–36.

[18] E. D'Hondt, M. Stevens, and A. Jacobs, "Participatory noise mapping works! An evaluation of participatory sensing as an alternative to standard techniques for environmental monitoring," Dec. 2011, Preprint. [Online]. Available: http://www.noisetube.net/publications/partnoisemaps.pdf