Private Pooling: A Privacy-Preserving Approach for Mobile Collaborative Sensing

Kevin Wiesner, Michael Dürr, and Markus Duchon

Mobile and Distributed Systems Group
Ludwig-Maximilian-Universität München
Oettingenstr. 67, 80538 Munich, Germany
firstname.lastname@ifi.lmu.de
http://www.mobile.ifi.uni-muenchen.de/

Abstract. Due to the emergence of embedded sensors in many mobile devices, mobile and people-centric sensing has become an interesting research field. A major aspect in this field is that quality and reliability of measurements highly depend on the device's position and sensing context. A sound level measurement, for instance, delivers highly differing values whether sensed from inside a pocket or while carried in a user's hand. Mobile collaborative sensing approaches try to overcome this problem by integrating several mobile devices as information sources in order to increase sensing accuracy. However, sharing data with other devices for collaborative sensing in return raises privacy concerns. By exchanging sensed values and context events, users might give away sensitive data, which should not be linkable to them. In this paper, we present a new mobile collaborative sensing protocol, *Private Pooling*, which protects the users' privacy by decoupling the data from its contributors in order to allow for anonymous aggregation of sensing information.

Keywords: Mobile Collaborative Sensing, Privacy, Ad hoc sharing.

1 Introduction

The recent development of mobile phones brought up various devices with embedded sensors. One popular example is the iPhone 4 that has a built-in gyrometer, accelerometer, proximity, and an ambient light sensor. In addition it has also integrated hardware for positioning and navigation (A-GPS, digital compass, Wi-Fi, Cellular) as well as for image and sound capturing (camera, microphone). The current Android reference [1] also supports various embedded sensors, for instance temperature and pressure sensors. It can be foreseen that upcoming mobile devices will come along with even more kinds of sensors. Possible future sensors might be environmental sensors (e.g. air pollution sensors), weather sensors (e.g. humidity sensors), as well as health sensors (e.g. heart rate sensors). Conceptual designs for smartphones with such a variety of sensing hardware already exist, for instance with the Nokia Eco Sensor Concept [17].

This development led to the research field of mobile or people-centric sensing networks (PCSN). In contrast to wireless sensor networks (WSN) where sensor

R. Prasad et al. (Eds.): MOBISEC 2011, LNICST 94, pp. 52–63, 2012.

are typically statically distributed, people-centric sensing (PCS) deals with mobile sensors embedded in user devices. In addition to mobility, we identified four main aspects in which PCSNs differ from WSNs:

1. Energy: In typical sensor networks, energy constraints are one of the main challenges [7]. However, as PCSN leverage user devices, energy is not as constrained as in typical WSNs. The life-time of static sensors is usually limited by the energy supply, whereas mobile phones can be recharged more easily. Thus, the challenge in PCSN is not the technical energy supply, but rather the willingness of users to share this limited resource. A possible solution to motivate users to share their resources is e.g. shown in [20].
2. Large-scale deployment: As more and more sensors will be built into regular mobile phones, there will be a huge amount of sensors that could be tapped for sensing applications. Large-scale sensor deployments are usually constrained by hardware costs, however, in PCSNs there are no such costs as users pay for it by buying mobile phones.
3. Sensor diversity: WSNs typically consist of a very homogeneous set of hardware, i.e., normally similar hardware is used throughout the network. In contrast, PCSNs consists of a very heterogeneous set of hardware as all different kinds of mobile devices could participate. This leads to differences in quality and accuracy of measurements that need to be coped with.
4. Privacy: As sensors are carried around by users, measurements reflect users' activities and locations. Thus, one main challenge of PCS is to conduct measurements and nevertheless preserve user privacy.

Another research field that builds upon embedded sensors are mobile context-aware systems, where built-in sensors of mobile devices are exploited to infer user context. For instance, many systems use a built-in accelerometer and gyrometer to determine the current user activity or setting [6,18]. This inferred information is then often used to adapt which and how information is presented, or to automate some processes with the goal to ease a user's life.

In either mentioned application field, quality and reliability of measurements highly depends on the device's sensing context, i.e., its position in relation to the information source [16]. That means, it is crucial where and how a mobile phone is carried by the user. The accelerometer reports different movement patterns when the phone is carried around in the pants pocket, compared to measurements conducted while carried in the handbag. A sound level measurement, for instance, delivers highly differing values if sensed with a mobile phone inside a bag compared to results where the device was carried in the hand. A context-aware application could for instance reason that a person is in a quiet room, while actually being in a crowded public space, just because the phone's microphone is in a pocket and sound patterns hence differ significantly.

Mobile collaborative sensing approaches try to overcome this problem. Instead of relying on one single value that might deliver distorted measurements, those concepts integrate several mobile sensing devices as information sources. The initiator of the request receives the collected information, and can then reach

a decision taking the different sensed valued into account. If several phones detect the same event, the probability that the event happened increases. If enough measurements are available, it is possible to statistically eliminate some of the measurements. Thereby, more robust and reliable results can be achieved.

A very important aspect of mobile sensing is the protection of privacy, as participating users might reveal sensitive information by providing their phone-based measurements. Collaborative sensing approaches even intensify these privacy concerns, as information is shared with anybody that requests it and thus provided data could be misused. To prevent this, shared information about sensed values or detected context events should be exchanged in an anonymized way. For ad hoc sharing of mobile sensors, user intervention is not reasonable, thus user privacy needs to be automatically adhered.

We present a new mobile collaborative sensing method, *Private Pooling*, which describes an approach to exchange sensed information anonymously by incorporating concepts from multi-party computation protocols. Private Pooling allows for aggregated sensing and collaborative context information exchange among co-located phones, without revealing the link between information and its contributor.

The remainder of this paper is structured as follows. In Section 2, a motivating example is given, and Section 3 outlines some fundamentals. Section 4 then describes our concept and proposed solution, followed by a discussion in Section 5. Section 6 describes the current prototype implementation, and Section 7 gives an overview of related work and projects. Finally, in Section 8 we summarize our results and conclude with an outlook on future work.

2 Motivation

In this section, we want to emphasize the need for a privacy-preserving protocol in the field of mobile collaborative sensing. A first question could be, why we aim for a protocol instead of letting users decide which information they want to share. There exist two reasons why we favor automated sharing: (1) Mobile collaborative sensing is supposed to be an ubiquitous technology, that should merge in the background. If people are required to actively accept sharing their sensor data, users could get annoyed by responding to these requests. The benefit of mobile collaborative sensing increases with the number of participants. It should be as easy as possible to participate in order to foster a large-scale deployment (cf. [12]). (2) The response time of participating phones should be very short, so that a requesting user can use the gathered information with only a minimum delay. Waiting for users to respond to requests whether they would like to share their data or not could protract the whole process. However, unobtrusive sharing of information also leads to a loss of control which information is shared with whom and when. Attackers could exploit unintentionally shared data. There are several use cases, where this loss of control could be problematic. In the following, we focus on one short illustrative example:

Alice, Bob, Carol, and Mallory are all at the same train station, and reside within radio range of Mallory, so that they can communicate directly with him. An application on Mallory's phone now wants to use collaborative sensing to infer the current context. Even though the term "context" typically refers to a description of "a situation and the environment a device or user is in" [19], in this example for the sake of simplicity we assume that context is a textual description of the environment the user is in. Further, we assume that the context recognition is part of the application, and that the application provides some context model, as our approach focuses on information exchange rather than on information reasoning. A possible model delivered could consist of a determined context in combination with a certainty that this context is correct ({context: certainty}). The application sends out a context aggregation requests, and gathers responses of the others. Bob and Carol might both return a simple model containing {train station : 100%}. Since Alice likes jewelry, she often visits a near-by jewelry store, which is well-known for its first-class gem. Her phone already learned that being in that area at that time often means that she is in the jewelry store, and thus her context response could look like the following: {train station: 82%, jewelry: 18%}. Alice's response contains highly sensitive information, which she does not want Mallory to know about. The application's privacy problem could thus lead to a real-world security risk for Alice, if Mallory knew that this information was contributed by her. Mallory could for instance reason that Alice is wealthy and use this information with criminal intent.

It becomes apparent that there is a strong need for privacy when collaborative sensing is applied. One way to solve this would be to blur or reduce the data that is shared. Alice could only send {train station: 82%} so that no sensitive data would leak. This raises two problems: First, this approach reduces the quality of data and therefore the usefulness might decrease as well. Second, it is impossible to automatically infer which parts are sensitive and which are not. Thus, our approach focuses on separating the data from the participants, i.e., shared data must not be linkable to the user who shared it. In our example, Mallory would receive a result {train station: 94%, jewelry: 6%}, but could not determine which user contributed which part.

3 Preliminaries

As pointed out in the previous section, the goal of our approach is to allow for sensor data exchange over insecure wireless connections without revealing individually contributed information. Users should be able to contribute their information without any privacy concerns, as nobody should be able to discover which part of the collected data was contributed by whom.

In the following, we first identify the requirements that a solution should fulfill. Subsequently, we shortly explain which concepts our approach is derived from.

3.1 Requirements

In this section we outline which requirements an optimal solution should fulfill. Mobile collaborative sensing should be employed for more accurate and robust sensing. Thus it should leverage the exchange of sensor data without harming users by revealing sensitive information. In this setting, we identified three main requirements:

***Privacy-Preserving*:** The protection of the participants' privacy is the main requisite, as already motivated before.

***Decentralization*:** If users are in a setting with no or only limited connectivity, e.g. in a building, it should still be possible to receive the data of surrounding mobile phones, without depending on a central backend server.

***Lossless Information Exchange*:** An optimal solution should be lossless, that means available data should be completely shared in best quality.

3.2 Background

Our concept was designed with the previous requirements in mind. One area where the mentioned aspects are already considered for data exchange is the field of secure multi-party computation (MPC). The aim of MPC is to provide secure solutions for joint computation with private inputs from multiple parties, typically by leveraging public-key cryptography. There exist several MPC approaches including approaches for secure joint computation, joint signatures, elections over the Internet, and private database access [10,9]. Further approaches are outlined in Section 7. Our concept of *Private Pooling* is based on a solution for the so-called "millionaires' problem". It originally refers to a situation where two millionaires want to know who of them is richer without revealing their wealth to their counterpart. Yao [21] proposed a solution for this problem. For our concept an adapted version of this problem is used as a basis.

Grosskreutz, Lemmen, and Rüping [11] describe a simple solution to allow multiple millionaires to find out how much money they own all together without revealing the individual's wealth. In Figure 1, the concept of this approach is illustrated. The first participant generates a random number (rnd) between 0 and M, where M is the upper bound of the outcome. Then he adds the value of his assets ($v1$), and sends this combined value modulo M to participant 2. As the random number is unknown to participant 2, he cannot determine the actual wealth of participant 1. From now on, the participants simply add their wealth to the aggregated sum (modulo M). The last participant returns the value, consisting of the random number and all participants' values, to participant 1, who can then subtract the initially added random value (rnd) and finally receives the value of the total wealth of all millionaires. This approach obviously only works in the semi-honest model, that means, participants try to find out as much as possible but all stick to the protocol. It does not work in case of a shared medium, as every communication can be overheard by all users.

In the following section, we outline our adaptation of this protocol to the mobile and collaborative sensing scenario.

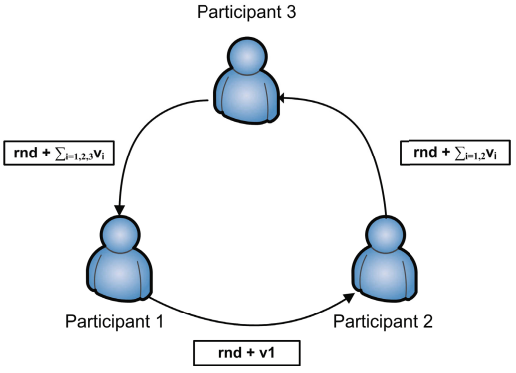


Fig. 1. Secure multi-party sum computation

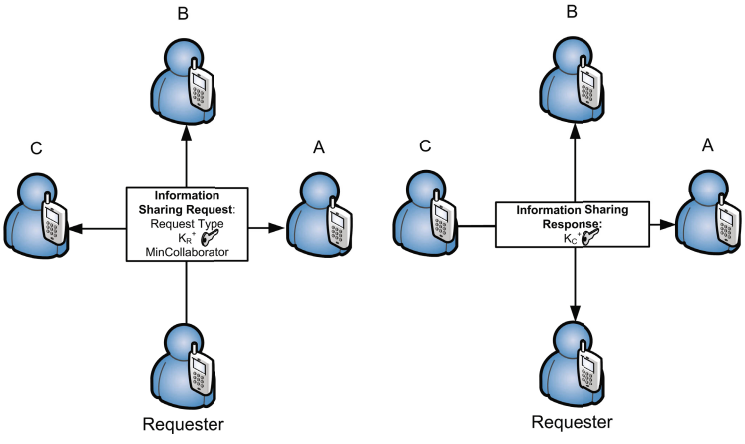
4 Concept

Our proposed concept, *Private Pooling*, is a protocol that enables privacy-preserving information exchange over wireless connections. This is reached by establishing secured circular communication and decoupling shared information from its contributor.

Our main contribution is the concept of secured circular communication, which refers to a circular communication order of participating users in such a way that communication between two users cannot be eavesdropped by other users in the vicinity. This adaptation of the aforementioned millionaires example allows us to enforce a certain, specified communication order, even in the mobile and wireless scenario. In combination with additional randomized data, it can be ensured that shared information cannot be backtracked to the users that contributed it, as in the aforementioned millionaires example.

Private Pooling consists of three phases: (1) Sharing Request Phase, (2) Sharing Response Phase, and (3) Information Aggregation Phase. In the following we explain all phases in detail:

(1) *Sharing Request Phase*: A collaborative sensing activity starts with the Sharing Request Phase (see Figure 2a). A user that wants to gather information of surrounding users (in the following called requester), broadcasts an Information Sharing Request (*ISReq*). The *ISReq* contains the type of information that the requester wants to collect, and thus notifies all users in his vicinity about his interest. Further, the *ISReq* specifies the minimum number of participants (*MinCollaborators*). The reason for this is to notify users about the privacy level of this sharing request (*ISReq*). The more users participate, the more difficult it is to link some data to a specific user. The usage of *MinCollaborators* will be explained later on. Collected information by only two other users can be more easily backtracked to the contributing users than if it is an information pool gathered by 20 users. Further, the *ISReq* contains the requester's public key to enable secure communication at a later phase.



(a) *Sharing Request* (b) *Sharing Response (User C)*

Fig. 2. Communication flow for initial phases

(2) *Sharing Response Phase*: In the second step, all users willing to participate, respond to the *ISReq* with an Information Sharing Response (*ISRes*). By sending an *ISRes* participants indicate that they are willing to contribute their sensed data to the data collection initiated by the requester (Figure 2b shows this process for user C only, but A and B need to respond in a similar way). The public key of the user is included in this *ISRes*, so that possible future messages addressed to him can be encrypted. As the *ISReq*, the *ISRes* is broadcasted too. Thereby, other participating users learn which contributors are within their reach, and can store their public keys so that they can send encrypted messages in the future. Users that are not in reach, are not able to exchange keys, and consequently will not be able to communicate directly, even if they happen to be within radio range later on.

(3) *Information Aggregation Phase*: This last step is the actual collaborative part where sensed data is collected. If a requester receives enough *ISRes*, i.e. if $\sum$ ISRes >= MinCollaborators, the requester can start the Information Aggregation Phase by sending out an Information Aggregation Request (*IAReq*). An *IAReq* consists of two main parts: the sensed data and aggregator information. The latter contains a list of users that are willing to participate (*scheduled participants*) signed with the private key (K_R) of the requester and an unsigned list of users that already contributed their data to this collection (*completed participants*). By signing the list of scheduled participants, each user can verify that this list was not modified by any other user than the requester. However, the second list has to be updated by each user, and could be also subject to forgeries. But in contrast to the former, the completed participant list cannot be modified in such a way that the modifying user would benefit from his changes himself. For the sake of simplicity, we will use the term *aggregator list* in the following for the remaining users that still need to participate ($P_{\text{scheduled}} \setminus P_{\text{completed}}$).

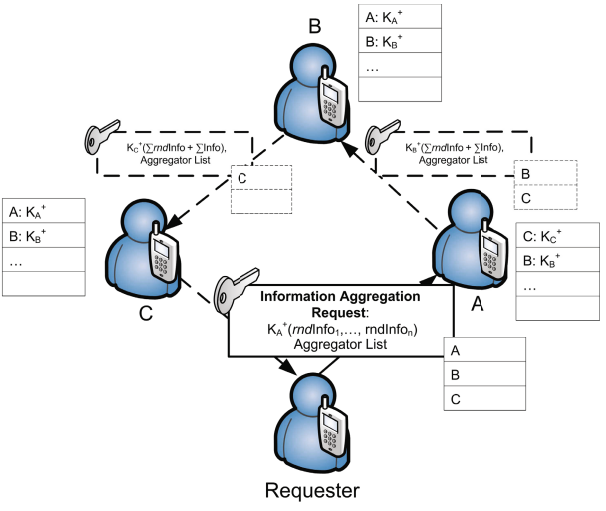


Fig. 3. Communication flow in the *Information Aggregation Phase*

Since the requester's direct successor knows that he is the first participant (because he receives the *IAReq* from the same user that broadcasted the *ISReq*), the requester does not contribute any real data in order to protect his privacy. Instead he generates n random data sets. Random data sets are generated from previously collected data sets of other users mixed with own former data, in order to provide highly realistic but uncritical data. If the requester, for instance, already participated in k previous sharing processes, he received d_k data sets each time (depending on his position in this process). Combined with p own data sets, he posses $p + \sum_{i=1}^{k} d_i$ data sets, from which he randomly chooses n to forward to the first participant. In case of a collection of temperature data, this could look like the following: $\{14\,°\text{C}, 15\,°\text{C}, 23\,°\text{C}\}$.

Subsequently, one user in the aggregator list is chosen as the first recipient, i.e. in our example in Figure 3 user A. The data is then encrypted with A's public key, and finally the *IAReq* is sent. As user A is the first participating user, A first checks whether the previously announced number of *MinCollaborators* is less or equal to the length of the received aggregator list. *MinCollaborators* specifies the minimum number of participants, and thus is a privacy indicator. If the length of the aggregator list does not comply with this number, the users privacy could be comprised, and therefore the *IAReq* is rejected. If the *IAReq* complies with the previously broadcasted *ISReq*, A selects one of the users (from which he received an *ISRes*) from the aggregator list as successor. The selection of the user is random, that means, there is no predetermined order in which the data is collected.

User A adds his sensed data, e.g. a temperature of 18 °C, to the received data from the requesters, and forwards the *IAReq* to the next participant, in our case user B. B would then remove his predecessor's (in this case A's) entry from the aggregator list (that means, he adds A to the list of completed participants),

and would receive the following data: $\{14\,°C, 15\,°C, 18\,°C, 23\,°C\}$. The following users add their sensor data as well, and forward the data collection to a remaining participant as long as the aggregator list consists of more than only their own entry. If no more other entries exist in the aggregator list, the last user (here: user C) adds his data to the collected data and forwards it back to the requester using the requester's public key. The last step of the protocol is the requester's removal of the initially added random data sets and the contribution of his own data, which leads to the real data collection. Assuming that the requester receives $\{14\,°C, 15\,°C, 18\,°C, 20\,°C, 20\,°C, 23\,°C\}$ from user C and measured a temperature of 17 °C himself, this would lead to the following final data set: $\{17\,°C, 18\,°C, 20\,°C, 20\,°C\}$.

In a further optional step, the requester could broadcast the collected data to all participants, as only the initiator knows the result. However, this might not be necessary in many cases, as the participating users might not be interested in the data at that moment. For that reason, the default is that collected information is not broadcasted. If participants are interested in receiving the aggregated information, they have to indicate it by setting a *BroadcastFlag*. In that way, the initiator knows whether a broadcast is necessary or not.

During this phase, it might happen that a user has not received any public key of the remaining users on the aggregator list during the *Sharing Response Phase*. Or the remaining participants are not in transmission range in the *Information Aggregation Phase* and cannot be reached anymore. In either case, this participant is not able to forward the collected data and sends an negative acknowledgement (*NACK*) to his predecessor. The predecessor then tries to forward the *IAReq* to another user on the aggregator list. If no other user is available, the *NACK* is further forwarded to preceding users until a successful communication order can be found or the requester receives the *NACK*. In the latter case, the requester simply triggers a new *ISReq* to overcome this problem.

5 Analysis and Discussion

First, we review the communication complexity of our approach: A naive approach where participants directly reply to the initiator's request requires n messages to gather measurements of n participants (*ISReq+(n-1)ISRes*). Our approach in contrast requires *2n* messages, as it first establishes a secured circular communication before actual measurements are transmitted. Even though it is a considerable increase, it remains scalable due to its linear complexity (*O(n)*). The communication overhead arises mainly from the circular data collection. Instead of sending each measurement directly to the requester, measurement values are collected in a specified order, which leads to an increase of utilized data bits. For n participants and m random values, this leads to $nm + \frac{n(n-1)}{2}$ values (the first message contains m values, the following *m+1* values, etc.) that are sent around, compared to *n-1* values in the naive approach.

Second, we outline some shortcomings and limitations of our approach: Our protocol is designed for the semi-honest model, as the approach outlined in

Section 3.2. If two malicious participants collaborate and exchange information they received from others, it could be possible to determine the input of individuals. In our example (Figure 3), participant A and C could exchange and compare the received information and thus extract B's input. Another problem in this context could arise, if a requester simulates n users and responds to his own request with *(n-1) ISRes* messages. Other users would then believe that there are *(n-1)* users participating, even though they might be the only real contributor. This would allow the requester to simply extract the contributed data. In the current version, we do not address those problems, as we focused on providing anonymity for the use case of participants sticking to the protocol. Our concept tries to avoid that sensitive information could be leaked even without active involvement of malicious users. However, in our future work we intend to extend this scenario for the malicious model as well.

6 Implementation

We built a first prototype as a proof of concept. Therefore, we developed an Android application that provides some sensor data (i.e. temperature readings) and collects data from co-located phones. For discovering nearby phones and the communication between those, the *haggle* API [2] was utilized. Haggle provides a network architecture for opportunistic communication and enables content exchange based on interests with a publish/subscribe messaging system. Thus, *ISReq* and *ISRes* can be broadcasted, as all phones register to follow the *ISChannel*. Messages in the information aggregation phase are sent using individual channels for each user. To secure communication, we used public key encryption provided through the standard android reference. In our future work we plan on using this prototype to evaluate our proposed protocol in empirical studies.

7 Related Work

As this paper combines aspects from mobile collaborative sensing as well as secure multi-party computation, relevant related work can also be found in these two research areas.

Mobile collaborative sensing is an approach which arose from mobile sensing research, often named people-centric sensing [5]. There has been a lot of work done on applications or systems that leverage embedded sensors of mobile phones [15,8,13,3], however most of the work focuses on individually sensed measurements. The concept of combining multiple information sources and cooperating sensors originates from static sensor networks, where approaches such as sensor fusion and aggregation have already been intensively studied. In the field of mobile phone-based sensing, only few works have examined collaborative sensor usage. One approach was proposed by Miluzzo et al. [14]: Their collaborative reasoning framework "Darwin Phones" allows for pooling of evolved models with neighboring phones, as well as collaborative inference for jointly observed events. Thereby, the authors want to improve the model development and aim

on making the inference more robust. Another example is presented by Bao and Choudhury [4]: They designed a phone application that collaboratively recognizes socially interesting events and automatically clusters video recordings around these detected events.

However, the mentioned mobile collaborative sensing approaches have not really addressed privacy and security issues as highlighted in this paper. Therefore, we proposed a protocol that is based on secure multi-party computation (MPC) concepts. MPC is a research field of cryptography, with the aim to enable secure execution of distributed computing tasks, without the need for a trusted third party. Therefore it usually leverages public-key cryptography. Yao [21] proposed a solution for the so-called "millionaires' problem", where two millionaires want to know who of them is richer without revealing their wealth to their counterpart. Other MPC concepts address for instance the issue of sharing a secret among multiple parties, computing random bit choices by multiple parties, or oblivious transfers. An overview of MPC problems and applications can be found in [9].

To the best of our knowledge, the utilization of multi-party approaches for wireless ad-hoc communication in the field of mobile collaborative sensing has not been studied yet, and thus makes a unique contribution to this field.

8 Conclusion and Future Work

This paper explores a new approach for mobile collaborative sensing. The proposed "Private Pooling" protocol leverages anonymous ad-hoc sensing data collection of co-located mobile phone users. In order to achieve a secure and privacy-preserving information exchange, our concept is based on a multi-party computation approach and prevents others from linking shared data to the person that shared this data. Thereby, we allow for a privacy-preserving information exchange without reducing quality of contributed data. We also discussed shortcomings of the current protocol and outlined our current implementation.

The next steps will be to further improve our protocol to make it more robust (e.g. also in case of the malicious model) and to conduct empirical studies using our prototype.

References

1. Android reference (December 2010), `http://developer.android.com/reference/`
2. haggle - A content-centric network architecture for opportunistic communication (2010), `http://code.google.com/p/haggle/`
3. Abdelzaher, T., Anokwa, Y., Boda, P., Burke, J., Estrin, D., Guibas, L., Kansal, A., Madden, S., Reich, J.: Mobiscopes for human spaces. IEEE Pervasive Computing 6, 20–29 (2007)
4. Bao, X., Choudhury, R.R.: Movi: mobile phone based video highlights via collaborative sensing, pp. 357–370 (2010)
5. Campbell, A.T., Eisenman, S.B., Lane, N.D., Miluzzo, E., Peterson, R.A.: People-centric urban sensing. In: Proceedings of WICON 2006, p. 18. ACM, New York (2006)

6. Choudhury, T., Borriello, G., Consolvo, S., Haehnel, D., Harrison, B., Hemingway, B., Hightower, J., Klasnja, P.P., Koscher, K., LaMarca, A., Landay, J.A., LeGrand, L., Lester, J., Rahimi, A., Rea, A., Wyatt, D.: The mobile sensing platform: An embedded activity recognition system. IEEE Pervasive Computing 7, 32–41 (2008)
7. Dargie, W., Poellabauer, C.: Fundamentals of Wireless Sensor Networks: Theory and Practice. Wiley (2010)
8. Das, T., Mohan, P., Padmanabhan, V.N., Ramjee, R., Sharma, A.: Prism: platform for remote sensing using smartphones. In: Proceedings of MobiSys 2010, pp. 63–76. ACM, New York (2010)
9. Du, W., Atallah, M.J.: Secure multi-party computation problems and their applications: a review and open problems, pp. 13–22 (2001)
10. Goldwasser, S.: Multi party computations: past and present. In: Proceedings of PODC 1997, pp. 1–6. ACM, New York (1997)
11. Grosskreutz, H., Lemmen, B., Rüping, S.: Privacy-preserving data-mining. Informatik-Spektrum 33, 380–383 (2010)
12. Lane, N.D., Eisenman, S.B., Musolesi, M., Miluzzo, E., Campbell, A.T.: Urban sensing systems: opportunistic or participatory? In: Proceedings of HotMobile 2008, pp. 11–16. ACM, New York (2008)
13. Lu, H., Pan, W., Lane, N.D., Choudhury, T., Campbell, A.T.: Soundsense: scalable sound sensing for people-centric applications on mobile phones. In: Proceedings of MobiSys 2009, pp. 165–178. ACM, New York (2009)
14. Miluzzo, E., Cornelius, C.T., Ramaswamy, A., Choudhury, T., Liu, Z., Campbell, A.T.: Darwin phones: the evolution of sensing and inference on mobile phones. In: Proceedings of MobiSys 2010, pp. 5–20. ACM, New York (2010)
15. Miluzzo, E., Lane, N., Eisenman, S., Campbell, A.: CenceMe: Injecting Sensing Presence into Social Networking Applications, pp. 1–28 (2007)
16. Miluzzo, E., Papandrea, M., Lane, N.D., Lu, H., Campbell, A.T.: Pocket, Bag, Hand, etc.-Automatically Detecting Phone Context through Discovery, `http://www.cs.dartmouth.edu/miluzzo/papers/miluzzo-phonesense10.pdf`
17. Nokia. Nokia eco sensor concept, `http://www.nokia.com/environment/devices-and-services/devices-and-accessories/future-concepts/eco-sensor-concept`
18. Raento, M., Oulasvirta, A., Petit, R., Toivonen, H.: Contextphone: A prototyping platform for context-aware mobile applications. IEEE Pervasive Computing 4, 51–59 (2005)
19. Schmidt, A., Beigl, M., Gellersen, H.-W.: There is more to context than location. Computers & Graphics 23(6), 893–901 (1999)
20. Teske, H., Furthmüller, J., Waldhorst, O.P.: A Resilient and Energy-saving Incentive System for Resource Sharing in MANETs. In: Proceedings of KiVS 2011, Dagstuhl, Germany. OASIcs, vol. 17, pp. 109–120 (2011)
21. Yao, A.C.: Protocols for secure computations. In: Annual IEEE Symposium on Foundations of Computer Science, pp. 160–164 (1982)