HAL
open science

Enforcing Privacy in Participatory Sensing Systems

Dimitrios Tsolovos

To cite this version:

Dimitrios Tsolovos. Enforcing Privacy in Participatory Sensing Systems. Middleware Doctoral Symposium 2018, ACM, Dec 2018, Rennes, France. hal-01910067

HAL Id: hal-01910067

https://inria.hal.science/hal-01910067

Submitted on 31 Oct 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Enforcing Privacy in Participatory Sensing Systems

Dimitrios Tsolovos
Inria & University of Versailles St. Quentin-en-Yvelines
dimitrios.tsolovos@inria.fr

Abstract

Mobile Participatory Sensing systems provide useful data gathered from mobile devices that would otherwise require expensive deployments of sensor networks. This type of data can identify participants and infer important private information. In this Ph.D. thesis, we investigate a decentralized approach assisted by Trusted Execution Environment enabled hardware to enforce the control of the participants on the usage of the collected personal data series.

CCS Concepts • **Security and privacy → Privacy-preserving protocols**; **Distributed systems security**;

Keywords privacy, mobile participatory sensing, trusted execution environments, distributed systems

1 Introduction

Mobile Participatory Sensing (MPS) systems collect spatio-temporal data and other sensor readings which can vary from environmental and health data, to any other type of data that mobile phones can collect. This data, typically, has the form of continuous time series and is sensitive since it can be used to identify participants and infer important information about their interests, location or medical conditions. MPS systems are distributed by nature and yet in most applications, participants must report the data they have collected to a server and thus "re-centralize" it. This approach assumes, by construction, that individuals do not question the honesty of the hosting company nor its capacity to defeat severe attacks, since centralization creates in essence a massive honeypot.

In recent studies, authors implement Privacy Enhancing Techniques (PET) such as k-anonymity and differential privacy to protect user privacy. The main idea is that a trusted entity in the architecture, will be responsible for anonymizing user data. In the approaches of [2] and [10], the authors use a central server while in [8], the authors let the mobile phone service provider handle this. In [3], participants are called to exchange their measurements with each other before reporting them back to the server. This way, the link between them and their data is broken. This needs to be coupled with other PETs as the link can be reestablished by analyzing a participant's history and combining it with external knowledge. For a survey of recent studies in centralized MPS we refer the reader to [1]. Such centralized approaches fail to provide guarantees that user data will not be misused either intentionally, by negligence or in case of a successful attack against the server. Trusting third parties to provide confidentiality and integrity preserving computations is not a convincing solution.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Middleware'18 Doctoral Symposium, Rennes, France
© 2018 ACM. 978-x-xxxx-xxxx-x/YY/MM…$15.00
DOI: 10.1145/nnnnnnn.nnnnnnn

2 Problem Statement

A privacy aware architecture for MPS systems should enforce that participants have control over their data, while enabling rich and secure computations over data from multiple participants. More specifically, it should ensure confidentiality of user data and of the intermediate results of the computations. Additionally, it should provide integrity guarantees that user data is used only for purposes that the participant has explicitly consented to, and that the results of these computations are correct. From the point of view of the application, it should provide guarantees that participants will provide real untampered data. Functional requirements include continuous computations over time series, scalability in terms of both the number of participants, and the amount of data.

Trusted Execution Environments (TEEs) are an emerging technology which provides users with confidentiality and integrity guarantees that their data will remain protected. Through their attestation mechanisms they can ensure users that the result of a computation is the product of the given function. However, simply using a TEE enabled server which collects all user data is not possible due to their limitations in terms of available memory. Furthermore, TEEs themselves can suffer from side channel attacks [12] which can lead to the leakage of private data.

Keeping data decentralized is a step on the right direction but is not sufficient. Important functions of MPS systems such as task assignment which is the process of selecting suitable participants to collect data, and rich computations on this collected data, require *re-centralization*. The problem then lies in providing a completely decentralized framework for MPS systems that enables the functionality of typical centralized MPS systems while respecting the above requirements. The main challenge when designing this framework is achieving efficient and secure distributed computations. TEEs can provide local integrity guarantees but the integrity of the global computation is not guaranteed and needs to be verified.

3 Related work

Distributed computations lie at the core of the problem. Existing techniques which enable privacy preserving distributed computations such as Secure Multiparty Computations (SMC) [13] and gossip protocols [5] have certain drawbacks that prevent their use. More specifically, the former does not scale well with the number of parties, while the latter cannot be used for general computations. In [7], the authors use a DHT and a blockchain to build a decentralized personal data management system which ensures users can keep control of their data. Data processing however, is not considered. In [9], the authors propose a decentralized architecture for real time traffic statistics. They propose protocols in which participants can collaborate to simulate the functionality of a trusted third party. Their approach provides some interesting ideas specific to traffic analysis. Whether these can be applied to other usages needs to be investigated. In [6] the authors propose a framework for running MapReduce computations in the cloud. They use trusted Intel SGX enabled processors which handle parts of the computations. In their

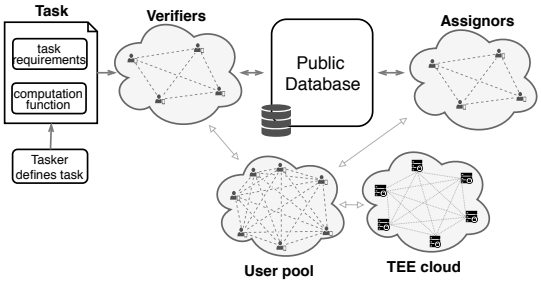

Figure 1. A high level overview of our proposal.

proposal though, they do not consider side channel attacks. Additionally, MapReduce is not optimized for stream processing which is the case for many MPS tasks. A study, closely related to ours, is [4] where the authors propose an architecture for MPS systems where users hold devices equipped with secure hardware with the goal of protecting the location privacy of participants. Moreover, they propose protocols for efficient data aggregation. However, task assignment is not considered, and location is not the only privacy sensitive information that participants provide in such systems.

While there are studies concerning decentralized computations, specific functions of an MPS system are not addressed. To the best of our knowledge, there does not exist a complete architecture for general decentralized privacy aware MPS systems.

4 Our approach

In Figure 1 we present a high level architecture of our proposed framework. It is comprised of a set of users, a public database and a cloud of devices enabled with TEEs. The tasking process can be divided into three sub-processes, the task definition, the task verification and the task distribution. A task will contain two pieces of information. The requirements in terms of sensors, time, locations of interest number of participants etc. and a function to be computed over the collected data. Once the tasking entity defines a task, a probabilistic selection mechanism will select a number of users (verifiers) from the pool of available users who will test the task for well-known threats. This mechanism will use the TEE cloud to provide the required integrity guarantees and ensure that malicious users are excluded. A different selection process selects the assignors who will assign the verified tasks to the available users based on their compatibility with the task description.

These two processes will assume the role that a central server would have in a centralized architecture. The results of these processes will be published on the public database in a way that will enable the users to verify their integrity without impeding the other user's privacy. The public database basically works as a bulletin board, where information about the system is published. A distributed process executed by the users to verify that this public database has not been tampered with will also need to be developed.

After the task has been distributed to the participants, they will then perform the required measurements specified in the task and save the collected data on their devices. With the assistance of a set of TEE enabled devices and cryptographic techniques, the participants will collaborate to execute the defined function. The result can then be published to the public database, or in case of private tasks, the result can be sent back to the tasker.

This approach allows users to keep data on their devices. When that data is requested, a verification process will ensure that data will only be used for specified purposes. The architecture avoids the re-centralization of data by keeping all processes distributed. However, there are several research questions that remain unanswered. How will the two selection mechanisms work? How will the task verification and assignment be performed? Finally, how will our system use TEEs to perform distributed computations?

5 Evaluation plan

The proposed framework will be evaluated both in terms of privacy, as well as its *efficiency* in terms of computing performance. More specifically, in terms of privacy evaluation we can measure the *data leakage* in case of a successful attack during the computations phase. That is, the amount of private data exposed to a user who should not have access to it compared to the total amount of data involved in the computation. With this we can measure the *cost-to-benefit ratio* of a successful attack. The performance of the task verification scheme based on the number of malicious users involved, and their effect on the outcome of the process. To measure the efficiency of the framework we will have to evaluate the performance of the task assignment compared to an offline optimal protocol, and the efficiency of the computations based on the amount of data. The various protocols can also be compared with systems (both centralized and distributed) from the state of the art. Possible use cases that could be tested on the framework include the collection of personal health data (e.g. steps walked, average resting heart rate etc.) and its comparison with the rest of the participants, or the generation of a noise map with data collected by Ambiciti [11].

6 Conclusion

The decentralized approach we present, can potentially provide users with guarantees that their data will not be misused and will remain private. The aim of this Ph.D. thesis is to examine if a fully decentralized privacy aware solution for MPS systems is a viable solution or if certain compromises are necessary in order to achieve acceptable privacy while maintaining important functionalities.

Acknowledgments

This thesis is funded by the Inria project CityLab@Inria and is co-supervised by Nicolas Anciaux and Valérie Issarny.

References

[1] Delphine Christin. 2016. Privacy in mobile participatory sensing: current trends and future challenges. *Journal of Systems and Software* 116 (2016), 57–68.
[2] Cory Cornelius et al. 2008. Anonysense: privacy-aware people-centric sensing. In *6th international conference on Mobile systems, applications, and services*. ACM.
[3] Delphine Christin et al. Privacy-preserving collaborative path hiding for participatory sensing applications. In *IEEE MASS 2011*.
[4] Dai Hai Ton That et al. 2016. PAMPAS: Privacy-Aware Mobile Participatory Sensing Using Secure Probes. In *ACM SSDBM 2016*.
[5] David Kempe et al. 2003. Gossip-based computation of aggregate information. In *IEEE Symposium on Foundations of Computer Science*.
[6] Felix Schuster et al. VC3: Trustworthy data analytics in the cloud using SGX. In *IEEE Symposium on Security and Privacy (SP) 2015*.
[7] Guy Zyskind et al. Decentralizing privacy: Using blockchain to protect personal data. In *IEEE Security and Privacy Workshops 2015*.
[8] Hien To et al. 2014. A framework for protecting worker location privacy in spatial crowdsourcing. *VLDB Endowment* 7, 10 (2014).
[9] Joshua WS Brown et al. Haze: Privacy-preserving real-time traffic statistics. In *21st ACM SIGSPATIAL 2013*.
[10] Khuong Vu et al. 2012. Efficient algorithms for k-anonymous location privacy in participatory sensing. In *IEEE INFOCOM*. IEEE, 2399–2407.
[11] Ventura Raphaël et al. 2017. Estimation of urban noise with the assimilation of observations crowdsensed by the mobile application Ambiciti. (2017).
[12] Yuanzhong Xu et al. Controlled-channel attacks: Deterministic side channels for untrusted operating systems. In *IEEE Symposium on Security and Privacy 2015*.
[13] Oded Goldreich. 1998. Secure multi-party computation. *Manuscript*. (1998).