This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2791432, IEEE Transactions on Dependable and Secure Computing

Providing Task Allocation and Secure Deduplication for Mobile Crowdsensing via Fog Computing

Jianbing Ni, *Student Member, IEEE,* Kuan Zhang, *Member, IEEE,* Yong Yu, *Member, IEEE,* Xiaodong Lin, *Fellow, IEEE,* Xuemin (Sherman) Shen, *Fellow, IEEE*

***Abstract*—Mobile crowdsensing enables a crowd of individuals to cooperatively collect data for special interest customers using their mobile devices. The success of mobile crowdsensing largely depends on the participating mobile users. The broader participation, the more sensing data are collected; nevertheless, the more replicate data may be generated, thereby bringing unnecessary heavy communication overhead. Hence it is critical to eliminate duplicate data to improve communication efficiency, a.k.a., data deduplication. Unfortunately, sensing data is usually protected, making its deduplication challenging. In this paper, we propose a fog-assisted mobile crowdsensing framework, enabling fog nodes to allocate tasks based on user mobility for improving the accuracy of task assignment. Further, a fog-assisted secure data deduplication scheme (Fo-SDD) is introduced to improve communication efficiency while guaranteeing data confidentiality. Specifically, a BLS-oblivious pseudo-random function is designed to enable fog nodes to detect and remove replicate data in sensing reports without exposing the content of reports. To protect the privacy of mobile users, we further extend the Fo-SDD to hide users' identities during data collection. In doing so, Chameleon hash function is leveraged to achieve contribution claim and reward retrieval for anonymous mobile users. Finally, we demonstrate that both schemes achieve secure, efficient data deduplication.**

Keywords: Mobile crowdsensing, fog computing, task allocation, secure deduplication.

I. Introduction

Mobile crowdsensing [1] is a compelling paradigm that allows a large group of individuals to collaboratively sense data and extract information about social events and national phenomena with common interest using their mobile devices, e.g., smart phones, smart glasses, drones, cameras and smart vehicles. It supports an ever-increasing number of sensing applications, ranging from social recommendation, such as restaurant recommendation, vehicular navigation and parking space discovery, to environment monitoring, such as air quality measurement, noise level measurement and dam water release warning [2], [3]. With the human intelligence and user mobility, it improves the quality of sensing data, extends the scale of sensing applications, and reduces the cost on high-quality data collection.

In mobile crowdsensing, one of the main challenges is to find proper mobile users for sensing tasks to achieve efficient and scalable data collection [4]. Firstly, due to the unique requirements of sensing tasks and the user mobility, a crowdsensing server (CS-server) collects various types of information about mobile users, e.g., location, reputation and activity pattern, and thereby customizes a task allocation policy for each sensing task [5]. For example, to measure traffic congestion in downtown Toronto, the CS-server should recruit the mobile users driving on the roads in downtown Toronto. Secondly, it is hard to guarantee that the potential mobile users could receive the assigned sensing tasks and upload sensing reports in time [6]. Thirdly, to perform sensing tasks, mobile users have to travel to specific locations with a certain cost on time and travel. Therefore, there should be an effective framework for the CS-server to allocate sensing tasks to proper mobile users.

In addition, with the increasing number of participating mobile users, there are inevitably some duplicates in sensing reports [7]. For a social event or national phenomenon, mobile users in the same location may obtain the same sensing data and generate the identical or similar items in sensing reports. For example, let us consider the following two scenarios:

- To measure the air quality in an urban area, mobile users measuring at proximate points may submit the same measurements;
- To survey the satisfaction on the government in a certain area, some mobile users may make the same options and give similar comments.

In these cases, the replicate data consumes a large amount of network bandwidth and storage space. A straightforward method to reduce the overhead is to discard the redundant copies on intermediates. Nevertheless, this approach exposes the detailed sensing data, which may contain plenty of personal information about mobile users, e.g., location, references, occupations, health status and religious beliefs [8]. To preserve the privacy of mobile users, data encryption is widely used to achieve data confidentiality, but brings a huge obstacle on the detection of replicate data for intermediates. Message-locked

Part of this research work was presented in IEEE Global Communications Conference 2016 (GLOBECOM 2016). This work is supported by the National Key R&D Program of China (2017YFB0802000), National Natural Science Foundation of China (NSFC) Research Fund (61728102), NSFC Research Fund for International Young Scientists (61750110528), National Cryptography Development Fund during the 13th Five-year Plan Period (MMJJ20170216), and the Fundamental Research Funds for the Central Universities (GK201702004). (Corresponding Author: Xiaodong Lin)

Jianbing Ni and Xuemin (Sherman) Shen are with Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada N2L 3G1. Email: j25ni@uwaterloo.ca, xshen@bbcr.uwaterloo.ca.

Kuan Zhang is with Department of Electrical and Computer Engineering, University of Nebraska-Lincoln, Omaha, NE 68182 USA. Email: kuan.zhang@unl.edu.

Xiaodong Lin is with Department of Physics and Computer Science, Wilfrid Laurier University, Waterloo, Ontario, Canada N2L 3C5, Email: xlin@wlu.ca.

Yong Yu is with School of Computer Science, Shaanxi Normal University, Xi'an, 710062, China. Email: yuyong@snnu.edu.cn.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2791432, IEEE Transactions on Dependable and Secure Computing

encryption (MLE) [9] (the most prominent manifestation of which is convergent encryption) may resolve this problem, where the same plaintexts always map to the same ciphertexts. Nonetheless, MLE is inherently subject to off-line brute-force attacks [10], where adversaries can learn the sensing data by guessing the possible plaintexts in encrypted sensing reports. Furthermore, the sensing data is predictable in some mobile crowdsensing applications, such as air quality measurement, place recommendation and traffic congestion monitoring. As a result, an attacker may guess to obtain the correct sensing data in an encrypted sensing report. Therefore, it is necessary to design a secure data deduplication mechanism to allow the intermediates to detect replicate reports without violating the privacy of mobile users.

However, if the equality of sensing reports can be detected in public, anyone can predict that the mobile uses are in approximate positions or have similar preferences knowing that they submit the identical sensing reports. For this "duplicate-linking" leakage, some sensitive information would be exposed in duplicate-sensitive mobile crowdsensing applications, e.g., air quality monitoring and place recommendation. In these applications, the mobile users in the same location or with the same profiles may report the identical sensing data. If two mobile users report identical measurements, the one may predict that the other is nearby. If two patients report the same symptoms, they may have the same disease [11]. Therefore, it is important to prevent privacy disclosure from the identical reports in mobile crowdsensing. However, data deduplication and privacy preservation are inherently in conflict. On one hand, if the intermediates can detect the replicate reports, they can know that there may be some correlation between the corresponding mobile users. On the other hand, if the privacy of mobile users is perfectly preserved, it is impossible for the intermediates to perform data deduplication. Therefore, it is an open problem to design a straightforward approach to prevent "duplicate-linking" leakage in duplicate-sensitive applications.

Furthermore, once the redundant copies in sensing reports are deleted, the CS-server cannot identify the contributions of mobile users. Although the redundant copies do not contribute to the completeness of sensing results, they improve their trustworthiness [12]. The CS-server should not ignore the contributions of the mobile users whose data are replicate with others'. However, if these mobile users are rewarded, some lazy mobile users may acquire unfair benefits by replaying the sensing reports generated by other mobile users and eavesdropped on communication channels. This "duplicate-replay" attack should be abandoned to get rid of the lazy mobile users, who are unwilling to perform crowdsensing tasks, but greedy for benefits. In short, it is of significant importance to not only support secure data deduplication over sensing reports against brute-force attacks, "duplicate-linking" leakage and "duplicate-replay" attacks, but also record the contributions of mobile users fairly without exposing the sensing data.

To the end, we exploit fog computing [13] to support accurate task allocation and secure data deduplication for mobile crowdsensing, which is a new architecture providing computing, storage and networking services proximate to terminal devices with appealing properties, including location awareness, geographic distribution and low latency. With fog computing, a large number of decentralized mobile devices can self-organize to communicate and potentially collaborate with each other via a fog node located at the edge of the Internet [14]. In this paper, we propose a Fog-assisted Mobile CrowdSensing framework (Fo-MCS) that enables fog nodes to perform task allocation based on the mobility patterns of users to improve the accuracy of task assignment. Under this framework, we design a Fog-assisted Secure Data Deduplication scheme (Fo-SDD) based on BLS-Oblivious Pseudo-Random Function (BLS-OPRF) to achieve the detection of replicate data, and extend the Fo-SDD to prevent "duplicate-linking" leakage for duplicate-sensitive applications. Specifically, the main contributions of our paper are as follows:

- We develop fog-assisted task allocation based on the local information about mobile users, such as mobility patterns and preferences. Specifically, the CS-server firstly assigns the sensing tasks to the fog nodes located in the intended sensing area. Then the fog nodes, acting as geography-related local servers, further find proper mobile users to fulfill the tasks. Fo-MCS not only reduces the overhead of CS-server on task allocation, but also improves the accuracy of task assignment.
- We design a BLS-OPRF scheme based on the BLS signature [15] to generate the encryption key of sensing data and enable fog nodes to detect and delete the identical sensing data in sensing reports for saving communication bandwidth. During this process, the fog nodes can learn nothing about the reports, except the equality of sensing data. Meanwhile, we also leverage the key-homomorphic signature [16] to sign the sensing data and allow fog nodes to aggregate the signatures of mobile users. By doing so, the CS-server only receives one copy of replicate sensing reports, but learns the contributions of the mobile users who generate these replicate sensing reports.
- We balance the trade-off between data deduplication and privacy preservation against "duplicate-linking" leakage. We leverage blind signatures [17] to ensure that no one can link the sensing reports to a specific mobile user, even if the user submits the identical reports with others. Nevertheless, once the participating mobile users are anonymous, it is difficult for the CS-server to distribute benefits based on their contributions. To address this issue, we utilize Chameleon hash function [18] to enable mobile users to claim their contributions and retrieve the corresponding rewards. In addition, the misbehavior of mobile users, including double-reporting of sensing data and double-retrieving of benefits, could be detected to guarantee the fairness of mobile crowdsensing.

The remainder of this paper is organised as follows. In section II, we review related work about task allocation and data deduplication. In section III, we define the Fo-MCS framework, security threats and design goals. Then, we describe our Fo-SDD and discuss its security in section IV, followed by the extend Fo-SDD in section V. We evaluate the performance of both schemes in section VI, and finally, we conclude our paper in section VII.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2791432, IEEE Transactions on Dependable and Secure Computing

II. Related Work

To ensure sensing tasks to be fulfilled effectively, how to select mobile users to perform the tasks is critical in mobile crowdsensing. Several reputation-based task allocation mechanisms [19], [20] have been proposed to evaluate the trustworthiness of mobile users and assign tasks to the mobile users with high reputation. Wang et al. [20] proposed an anonymous reputation management scheme and a data trust assessment scheme. These schemes enforce both positive and negative reputation updates and preserve the identities of mobile users without the involvement of trusted third parties. To achieve better accuracy, Kazemi et al. [21] defined reputation scores to represent the probability that a mobile user can perform a task correctly, and a confidence level to state that a task is acceptable if its confidence is higher than a given threshold. Moreover, spatial-temporal correlation is widely used to recruit mobile users. Kazemi and Shahabi [22] focused on spatial task assignment for spatial crowdsourcing, in which the service provider allocates tasks using greedy, least location entropy priority or nearest neighbor priority algorithm based on the location of mobile users. To et al. [5] defined maximum task assignment problem and proposed three solutions to resolve this problem by utilizing spatial distribution and travel cost of mobile users. The first approach is based on local optimization strategy, while the second one improves the overall task allocation by assigning higher priority to sensing tasks located in the moving area of mobile users. The third approach incorporates the travel cost of mobile users in task assignment. To keep location privacy in spatial crowdsourcing, Shen et al. [23] proposed a secure task assignment protocol by utilizing additive homomorphic encryption. This protocol can protect the location privacy of mobile users in a semi-honest adversary model. Pournajaf et al. [4] investigated an approach for allocating crowdsensing tasks to mobile users without sharing the location of mobile users to the CS-server, and proposed a two-stage optimization method to globally optimize task allocation using cloaked location. Wang et al. [24] leveraged spatial and temporal correlation among the data in different areas to reduce the number of allocated tasks, and proposed a novel crowdsourcing task allocation framework by combining compressive sensing, Bayesian inference and active learning techniques. The CS-server can dynamically adjust the minimum number of sub-area of tasks in each cycle for ensuring data quality. In addition, due to the limited power of mobile devices, energy is also an important factor to determine the selection of mobile users. Xiong et al. [25] proposed an energy-efficient mobile crowdsensing framework that guarantees the required number of mobile users returning reports and the minimized number of redundant tasks allocated. Energy consumption on data reporting for a specific mobile user or overall users is dramatically reduced. Liu et al. [26] introduced the concept of quality-of-information (QoI) to evaluate data granularity and presented a QoI-aware energy-efficient scheme to optimize QoI for mobile crowdsensing. Xiong et al. [27] defined a spatial-temporal coverage metric for mobile crowdsensing and proposed a generic mobile crowdsensing framework in energy-efficient Piggyback crowdsensing model, which optimizes the task allocation with various incentives.

To achieve secure deduplication, Bellare et al. [9] introduced the concept of message-locked encryption and proposed several schemes against "duplicate-faking" attacks for space-efficient secure outsourced storage. However, these schemes are vulnerable to off-line guessing attack if the messages are predictable. To prevent this attack, Keelveedhi et al. [10] utilized server-aided encryption to achieve deduplicated storage and presented RSA-oblivious pseudo-random function (RSA-OPRF) to add randomness into the deterministic ciphertexts. A duplicateless encryption for simple storage (DupLESS) was proposed to offer a secure, easily-deployed solution for efficient outsourced storage supporting data deduplication. Consequently, to further enhance the security of MLE, Bellare et al. [28] presented interactive MLE, where the upload and download protocols are interactive between the client and server, to achieve security for messages that are both correlated and parameter dependent. Following Bellare et al.'s works, Li et al. [29], [30] investigated the problems of authorized data deduplication with the considering of privileges of users in a hybrid cloud, and reliable key management to ensure that the deduplicated outsoureced data is readable for multiple data owners. Wen et al. [31] further studied the reliable key management problem in secure data deduplication and leveraged group key agreement to support dynamic file update. Liu et al. [32] pointed out the client-side encryption is at odds with the practice of data deduplication and proposed a secure cross-user deduplication scheme supporting client-side encryption without the dependance on additional servers. Recently, Cui et al. [33] and Zheng et al. [34] considered the data deduplication in some emerging applications and services, such as named data networking and cloud-based media hosting. Specifically, Cui et al. [33] introduced the problem of near-duplicate detection in content-centric services and presented a secure near-duplicate detection system over encrypted in-network storage for effective resource utilization and possible traffic alleviation. Zheng et al. [34] explored the deduplication of encrypted multimedia contents for the cloud to eliminate the extra storage and bandwidth cost.

In our preliminary work [35], we proposed a fog-based deduplicated mobile crowdsensing scheme (Fo-DSC) to achieve fog-based task allocation and secure data deduplication, simultaneously. However, Fo-DSC is vulnerable to brute-force attacks and duplicate-linking leakage. In this paper, we proposed a fog-assisted secure data deduplication scheme to improve the security of Fo-DSC in mobile crowdsensing. We design a BLS-OPRF mechanism to enable fog nodes at the edge of networks to detect and delete the replicate sensing data in crowdsensing reports. The idea of BLS-OPRF is derived from the RSA-OPRF scheme in [10] based on the BLS signature [15]. Furthermore, we employ the blind signature to prevent privacy leakage of mobile users from multiple duplicates, and develop Chameleon hash function to achieve contribution claim and reward retrieval for anonymous mobile users. In addition, the communication overhead of Fo-SDD is much lower than that of the Fo-DSC.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2791432, IEEE Transactions on Dependable and Secure Computing

III. Problem Statement

In this section, we formalize Fo-MCS framework and security threats. Then, we identify design goals.

A. Fo-MCS Framework

Fo-MCS framework consists of three layers: service layer, fog layer and mobile users layer, and four entities: customers, a CS-server, fog nodes and mobile users. In the service layer, customers can be individuals or organizations. They have sensing tasks to fulfill but do not have sufficient resources to complete individually. Hence, they release these tasks on a CS-server. The CS-server provides mobile crowdsensing services for customers. It is responsible to assign sensing tasks to fog nodes based on the spatial information of tasks, process sensing reports, and distribute benefits to mobile users. In the fog layer, the fog nodes are deployed at the edge of the Internet and stretch from different network equipment, e.g., roadside units on roads, access points, gateways and edge routers. They have computing capability and storage space to provide computation and storage services to mobile users. Their responsibilities include assigning sensing tasks to mobile users on behalf of local servers, processing on sensing reports and forwarding the processed reports to the CS-server. In the mobile users layer, the mobile users perform the sensing tasks to collect data for earning rewards using their own mobile devices with the capabilities of data sensing, processing and communication.

As illustrated in Fig. 1, the whole Fo-MCS framework works as follows. A customer generates a sensing task and sends it to the CS-server, along with the rewards to attract mobile users. After obtaining the sensing task, the CS-server performs fog-assisted task allocation to assign it to mobile users. Specifically, the CS-server allocates the sensing task to the fog nodes according to the sensing area of the task and the coverage areas of fog nodes; and the fog nodes further recruit mobile users in their coverage areas to fulfill the task based on their mobility patterns and the task requirements. Then, the participating mobile users collect sensing data, generate sensing reports and submit them to the fog nodes. The fog nodes process the received sensing reports, including data deduplication and data aggregation, and forward the processed reports to the CS-server. After that, the CS-server generates a crowdsensing result for the customer based on the processed reports. Finally, the customer reads the crowdsensing result and determines the contributions of mobile users and the CS-server distributes the rewards to mobile users according to their contributions on the sensing task.

B. Security Threats

Security threats come from both external and internal attackers. The global eavesdroppers may wiretap on wireless communication channels to capture the messages exchanged between two entities, e.g., fog nodes and mobile devices. The CS-server and fog nodes are both honest-but-curious, indicating that they follow the protocols agreed with customers and mobile users honestly, but they are also interested in the

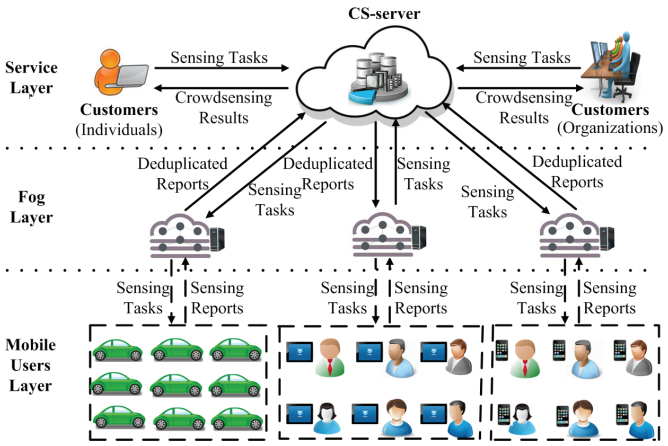


Fig. 1. Fo-MCS Framework

sensing reports generated by mobile users. The mobile users are honest to perform sensing tasks for benefits, but curious on the sensing reports submitted by other users, lazy for sensing data and greedy for benefits. Specifically, the attackers may launch the following attacks to achieve their goals:

- Brute-Force Attack: A curious entity, including the CS-server or mobile users, checks all possible sensing data or measurements with the hope of eventually obtaining the correct plaintexts in the encrypted sensing reports.
- "Duplicate-Linking" Leakage: The identical sensing reports disclose the equality of sensing data generated by mobile users. Thus, it is predictable that these mobile uses are in proximate positions or have similar profiles, such as references, habits or health status.
- "Duplicate-Replay" Attack: A lazy mobile user captures a sensing report delivered by others through eavesdropping and replays it to cheat the fog node to believe that his report is identical with a submitted one. Thus, the mobile user would be rewarded although the replayed report will be deleted by the fog node.
- Double-Reporting: A greedy mobile user may submit more sensing reports than allowed without being detected, in such a way that the user would obtain more rewards.
- Double-Retrieving: To acquire more benefits than that rewarded by the CS-server, a greedy mobile user may retrieve the rewards more than once without being detected.

In addition, the mobile users may deliver forged sensing data to cheat customers for benefits. This active attack has been discussed in [36], [37], which can be resisted by using trust management of mobile users or tasks duplication among multiple participants (to recruit multiple mobile users to collect the same data or measure the same phenomenon). Therefore, we assume that the majority of mobile users are fully trusted to perform the sensing tasks, and multiple fog nodes would not collude together, or collude with the CS-server to invade the privacy of mobile users. The customers are honest as they are the beneficiaries of mobile crowdsensing services.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2791432, IEEE Transactions on Dependable and Secure Computing

C. Design Goals

To achieve secure data deduplication under the Fo-MCS framework and resist the security threats, Fo-SDD should achieve the following design goals.

- *Secure Data Deduplication*: To save communication bandwidth, the replicate data in sensing reports should be securely deleted. Specifically, the fog nodes are able to detect and erase the replicate data without learning any information about the sensing reports. To ensure the confidentiality of sensing data, Fo-SDD should satisfy the following security goals:
  - Security against Brute-Force Attacks: The sensing data should be encrypted to prevent attackers from recovering it through brute-force attacks. Although the semantic security cannot be achieved, Fo-SDD should reach high security guarantee, except that the encrypted sensing reports expose the equality of underlying sensing data.
  - No "Duplicate-Linking" Leakage: The privacy leakage from the equality of sensing data should be prevented in duplicate-sensitive applications. A mobile user cannot predict that another user is similar with him in some aspects if they submit identical sensing reports.
  - Security against "Duplicate-Replay" Attacks: To prevent lazy mobile users from replaying the captured sensing reports, it is necessary to prove that the mobile users actually possess the sensing data if they submit the corresponding reports to fog nodes. A lazy mobile user can be detected if he replays captured sensing reports generated by others.
- *Efficient Contribution Claim*: The contributions of mobile users who submit the replicate sensing reports should not be ignored. To reduce communication overhead and record the contributions of mobile users, the fog nodes are able to aggregate the signatures on the identical sensing data generated by different mobile users. In addition, to maintain the fairness of mobile users, Fo-SDD should achieve the following goals:
  - Detection of Double-Reporting: A greedy mobile user cannot submit more sensing reports than allowed to the CS-server without being detected.
  - Detection of Double-Retrieving: A greedy mobile user cannot double-retrieve the rewards from the CS-server without being detected.

In addition, to offer sophisticated security protection on the Fo-MCS framework, we should achieve other fundamental security goals, such as the confidentiality of sensing tasks against external attackers, the authentication and integrity of sensing reports.

IV. Fo-SDD Scheme

In this section, we introduce the overview of Fo-SDD, describe the Fo-SDD in detail and discuss the security properties of Fo-SDD.

A. Overview of Fo-SDD

To resist brute-force attacks, we design a BLS-OPRF scheme to prevent attackers from guessing the predictable sensing data. Specifically, the local fog node $\mathcal{F}_j$ aids each mobile user $\mathcal{U}_i$ in its coverage area to generate the encryption key $\mathcal{S}_i$ with its secret key x_j for the sensing data $\mathcal{P}_i$. Thus, the attackers cannot compute $\mathcal{S}_i$ without x_j and thereby recover $\mathcal{P}_i$ from the target ciphertext Z_i using brute-force attacks. Unfortunately, a curious $\mathcal{F}_j$ is still able to guess $\mathcal{P}_i$ in Z_i. To prevent the brute-force attacks of $\mathcal{F}_j$, we employ mobile fog nodes to generate the encryption key $\mathcal{S}_i$ of sensing data $\mathcal{P}_i$ for $\mathcal{U}_i$. As a result, a single fog node $\mathcal{F}_j$ cannot launch brute-force attacks to recover $\mathcal{P}_i$ from Z_i. In addition, to further reduce communication overhead and achieve contribution claim, we allow each mobile user to generate a signature on the sensing data based on the key-homomorphic signature [16]. The distinguished feature of the key-homomorphic signature is that the signatures from multiple mobile users who generate replicate reports can be aggregated to be one signature, while all the public keys of these mobile users should be used to verify the validity of the aggregated signature. In this way, the communication overhead between fog nodes and CS-server is reduced and the customer can learn the identities of contributors during the verification of the aggregated signature. In addition, proxy re-encryption [38] is leveraged to realize the confidentiality of sensing tasks and allow the CS-server to efficiently delegate the decryption capability of sensing tasks to fog nodes.

B. The Detailed Fo-SDD

The Fo-SDD consists of six phases: Service-Setup, Task-Releasing, Task-Allocation, Data-Collection, Data-Deduplication and Data-Reading. The detailed Fo-SDD is described below.

1) Service-Setup: This phase is run by the CS-server to bootstrap mobile crowdsensing services. Given the security parameter λ, the CS-server defines two cyclic groups $(\mathbb{G}, \mathbb{G}_T)$ with the same prime order p, where p is λ bits. Let g be a random generator of group $\mathbb{G}$, $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be a bilinear map. $H : \{0,1\}^* \rightarrow \mathbb{G}$ and $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^\lambda$ are two full-domain hash functions, such as SHA-256 or SHA-3 [39]. $(\mathcal{SE}, \mathcal{SD})$ are the encryption and decryption algorithms of a deterministic symmetric encryption scheme. Such a scheme can be constructed from AES scheme with a fixed IV in CTR mode [39]. (SE, SD) are the encryption and decryption algorithms of the standard AES scheme [39]. The CS-server randomly chooses $s \in \mathbb{Z}_p$ as its secret key and computes $S = g^s \in \mathbb{G}$ as its public key. The DSA signature [39] is employed to achieve the integrity and authentication of sensing tasks and sensing reports during transmission.

A fog node $\mathcal{F}_j$ randomly chooses $x_j \in \mathbb{Z}_p$ as the secret key and computes $X_j = g^{x_j} \in \mathbb{G}$ as the public key.

A mobile user $\mathcal{U}_i$ randomly picks $v_i \in \mathbb{Z}_p$ as the secret key and computes $U_i = \hat{e}(g,g)^{v_i}$ as the public key.

2) Task-Releasing: When a customer $\mathcal{C}$ is willing to collect data for some purpose, such as measuring air quality, monitoring traffic condition or reconstructing indoor floor plan, $\mathcal{C}$ first

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2791432, IEEE Transactions on Dependable and Secure Computing

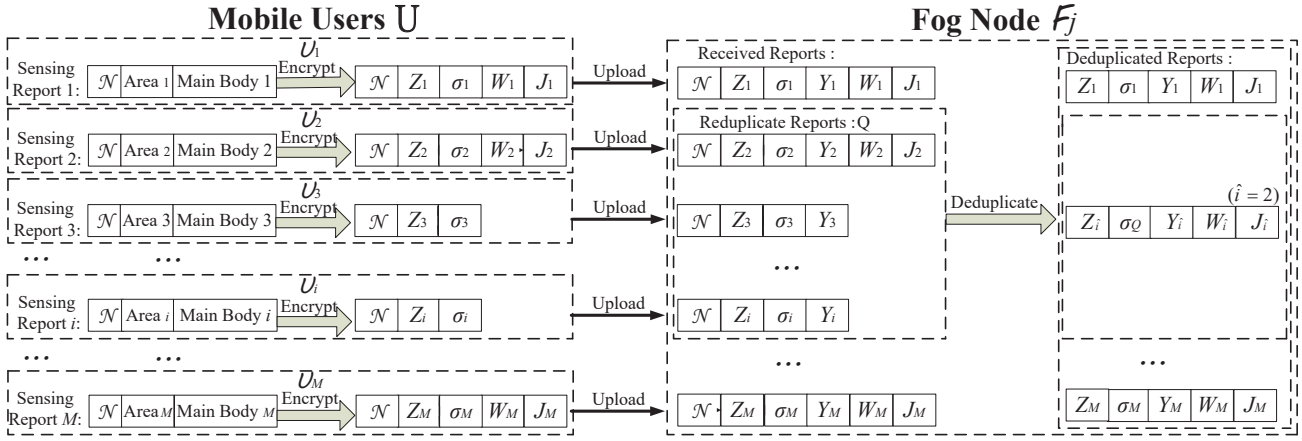


Fig. 2. Data Collection and Deduplication

generates a sensing task $\mathcal{T} = (\mathcal{T}_t, \mathcal{T}_e, \mathcal{T}_a, \mathcal{T}_b)$, indicating the goal (what to sense), the expiration time (when to sense), the sensing area (where to sense) and the benefits, respectively. Then, $\mathcal{C}$ chooses a random $k \in \mathbb{Z}_p$ to calculate a temporary public key $K = g^k$. After that, $\mathcal{C}$ encrypts $\mathcal{T}$ by randomly picking $r \in \mathbb{Z}_p$, $T \in \mathbb{G}_T$ to compute

$$(C_1, C_2, C_3) = (S^r, T\hat{e}(g,g)^r, SE(\mathcal{H}(C_2, T), \mathcal{T}_t||\mathcal{T}_e||\mathcal{T}_b)).$$

Finally, $\mathcal{C}$ sets $C_c = (C_1, C_2, C_3)$ and sends $(C_c, K, \mathcal{T}_a)$ to the CS-server.

3) Task-Allocation: Upon receiving $(C_c, K, \mathcal{T}_a)$, the CS-server first chooses $\mathcal{N} \in \mathbb{Z}_p$ as a unique identifier of $\mathcal{T}$ and picks a set of fog nodes $\mathbb{F} = \{\mathcal{F}_1, \cdots, \mathcal{F}_N\}$ located in $\mathcal{T}_a$, where N is the number of fog nodes in the set $\mathbb{F}$. Then, for each $\mathcal{F}_j \in \mathbb{F}$, the CS-server uses s and X_j to compute

$$RK_j = X_j^{\frac{1}{s}}, \qquad C'_j = \hat{e}(C_1, RK_j).$$

Finally, the CS-server sends $(\mathcal{N}, C'_j, C_2, C_3, K, \mathcal{T}_a)$ to $\mathcal{F}_j$.

When $\mathcal{F}_j$ receives $(\mathcal{N}, C'_j, C_2, C_3, K, \mathcal{T}_a)$, it first decrypts the sensing task as $T' = \frac{C_2}{(C'_j)^{1/x_j}}$ and $\mathcal{T}_t||\mathcal{T}_e||\mathcal{T}_b = SD(\mathcal{H}(C_2, T'), C_3)$. Then, $\mathcal{F}_j$ checks whether the task $\mathcal{T}$ is expired or not. If not, $\mathcal{F}_j$ recruits a set of mobile users $\mathbb{U} = \{\mathcal{U}_1, \cdots, \mathcal{U}_M\}$ to perform $\mathcal{T}$ based on the requirements of $\mathcal{T}$ and the mobility patterns of mobile users [4], [21], [25], where M is the number of mobile users in $\mathbb{U}$. For each $\mathcal{U}_i \in \mathbb{U}$, $\mathcal{F}_j$ randomly chooses $r_i \in \mathbb{Z}_p$, $R \in \mathbb{G}_T$ and encrypts $\mathcal{T}$ as

$$(D_{i1}, D_{i2}, D_{i3}) = (g^{r_i}, RU_i^{r_i}, SE(\mathcal{H}(\mathcal{N}, R), \mathcal{T}_t||\mathcal{T}_e||\mathcal{T}_b)).$$

Finally, $\mathcal{F}_j$ sets $D_i = (D_{i1}, D_{i2}, D_{i3})$ and sends $(\mathcal{N}, D_i, K, \mathcal{T}_a)$ to $\mathcal{U}_i$.

4) Data-Collection: When receiving $(\mathcal{N}, D_i, K, \mathcal{T}_a)$, $\mathcal{U}_i$ first decrypts the ciphertext D_i as $R' = \frac{D_{i2}}{\hat{e}(g, D_{i1})^{v_i}}$ and $\mathcal{T}_t||\mathcal{T}_e||\mathcal{T}_b = SD(\mathcal{H}(\mathcal{N}, R'), D_{i3})$. If this task is not expired, $\mathcal{U}_i$ starts to perform the task, collect and generate the sensing data $\mathcal{P}_i$ according to the requirements of $\mathcal{T}$. Then, $\mathcal{U}_i$ randomly chooses $s_i \in \mathbb{Z}_p$ to compute $S_i = H(\mathcal{P}_i)^{s_i}$ and sends $(\mathcal{N}, S_i)$ to $\mathcal{F}_j$. After $\mathcal{F}_j$ receives $(\mathcal{N}, S_i)$, it calculates $S'_i = S_i^{x_j}$ and returns S'_i to $\mathcal{U}_i$ (To prevent brute-force attacks from $\mathcal{F}_j$, S'_i can be generated by multiple fog nodes, that is, $S'_i = S_i^{\sum_{j \in \mathcal{M}} x_j}$, where $\mathcal{M}$ is the set of indices of $\mathcal{F}_j$ and its neighboring fog nodes). Then, $\mathcal{U}_i$ computes $\mathcal{S}_i = (S'_i)^{\frac{1}{s_i}}$ and verifies whether

$$\hat{e}(H(\mathcal{P}_i), X_j) = \hat{e}(\mathcal{S}_i, g) \tag{1}$$

holds or not. If not, $\mathcal{U}_i$ returns failure and aborts; otherwise, it calculates $Z_i = \mathcal{SE}(\mathcal{H}(\mathcal{N}, \mathcal{S}_i), \mathcal{P}_i)$. Furthermore, $\mathcal{U}_i$ chooses a random $w_i \in \mathbb{Z}_p$ and generates a signature σ_i as

$$\sigma_i = (\sigma_{i1}, \sigma_{i2}) = (g^{-w_i}, g^{v_i} H(\mathcal{N}, \mathcal{S}_i, \mathcal{P}_i)^{w_i}).$$

$\mathcal{U}_i$ sets the sensing report $\mathbb{P}_i = (Z_i, \sigma_i)$ and sends $(\mathcal{N}, \mathbb{P}_i)$ to $\mathcal{F}_j$. Upon receiving $(\mathcal{N}, \mathbb{P}_i)$, $\mathcal{F}_j$ computes $Y_i = \mathcal{H}(\mathcal{N}, Z_i)$ and checks whether Y_i exists in the database. If yes, $\mathcal{F}_j$ returns success and aborts; otherwise, it keeps $(\mathcal{N}, \mathbb{P}_i, Y_i)$ and requests $\mathcal{U}_i$ to return (W_i, J_i). $\mathcal{U}_i$ generates (W_i, J_i) by picking a random number $a_i \in \mathbb{Z}_p$ to calculate $W_i = g^{a_i}$, $a'_i = \mathcal{H}(W_i, K^{a_i})$, $J_i = SE(a'_i, \mathcal{S}_i)$ and sends (W_i, J_i) to $\mathcal{F}_j$.

It is worth pointing out that the ciphertext of sensing data $\mathcal{P}_i$ supports replicate data detection and deletion. Specifically, Y_i is a tag used to detect the duplicate of $\mathcal{P}_i$. $\mathcal{S}_i$ is derived from $(\mathcal{N}, \mathcal{P}_i)$ and used to encrypt $\mathcal{P}_i$. Therefore, $\mathcal{F}_j$ is able to detect the replicate data based on Y_i in sensing reports. The data collection and deduplication processes are shown in Fig. 2, and the information flow of data collection phase is illustrated in Fig. 3.

5) Data-Deduplication: Upon receiving $\{\mathbb{P}_1, \cdots, \mathbb{P}_M\}$ from $\mathbb{U}$, $\mathcal{F}_j$ checks whether $\{\mathbb{P}_1, \cdots, \mathbb{P}_M\}$ are replicate or not. If there are two reports, in which Y_i=$Y_{\hat{i}}$, the reports in $\mathbb{P}_i$ and $\mathbb{P}_{\hat{i}}$ are identical. If a set of reports $\{\mathbb{P}_i\}_{i \in Q}$ are identical, where Q is the set of indices of replicate reports, $\mathcal{F}_j$ aggregates the corresponding signatures $\{\sigma_i\}_{i \in Q}$ as

$$\sigma_Q = (\sigma_{Q1}, \sigma_{Q2}) = (\textstyle\prod_{i \in Q} \sigma_{i1}, \prod_{i \in Q} \sigma_{i2}).$$

Then, $\mathcal{F}_j$ keeps the first copy $Z_{\hat{i}}$ generated by $\mathcal{U}_{\hat{i}}$ who delivers $(W_{\hat{i}}, J_{\hat{i}})$ and deletes the replicate copies. $\mathcal{F}_j$ sets the sensing reports that are not replicate with others as $\{\mathbb{P}_i\}$ for each $1 \le i \le M$ and $i \notin Q$. Finally, $\mathcal{F}_j$ forwards the deduplicated reports $(\mathcal{N}, \{(Z_i, \sigma_i, W_i, J_i)\}_{i \notin Q}, Z_{\hat{i}}, \sigma_Q, W_{\hat{i}}, J_{\hat{i}})$ to the CS-server.

When receiving $(\mathcal{N}, \{(\widehat{\mathbb{P}}_i, W_i, J_i)\}_{i \notin Q}, Z_{\hat{i}}, \sigma_Q, W_{\hat{i}}, J_{\hat{i}})$ from $\mathcal{F}_j$, the CS-server forwards them to the customer $\mathcal{C}$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2791432, IEEE Transactions on Dependable and Secure Computing

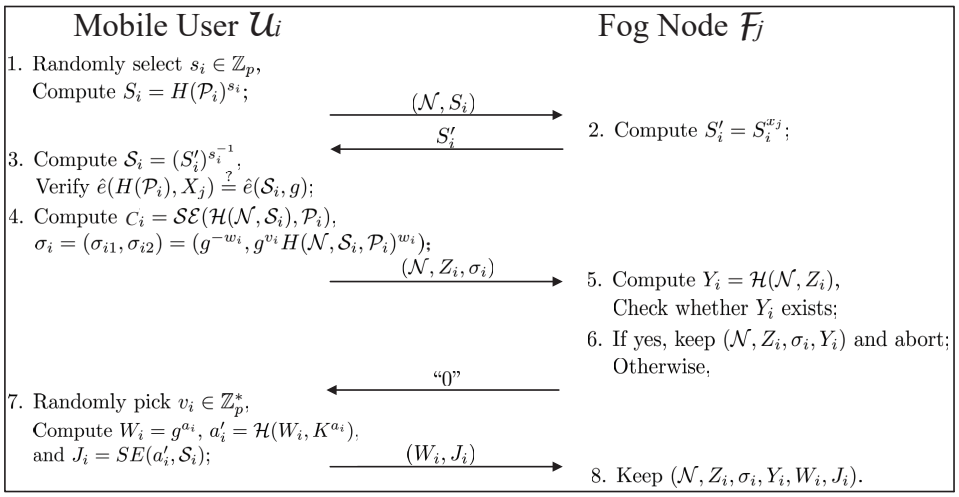


Fig. 3. Information Flow of Data Collection

6) Data-Reading: When $\mathcal{C}$ receives the deduplicated reports from the CS-server, $\mathcal{C}$ uses k to decrypt the deduplicated reports and checks the contributors (mobile users) as follows:

- For each $(Z_i, \sigma_i, W_i, J_i) \in \{(Z_i, \sigma_i, W_i, J_i)\}_{i \notin Q}$, $\mathcal{C}$ computes
$$\mathcal{S}_i = SD(\mathcal{H}(W_i, W_i^k), J_i), \quad \mathcal{P}_i = \mathcal{SD}(\mathcal{H}(\mathcal{N}, \mathcal{S}_i), Z_i).$$
After recovering all sensing reports $\{(Z_i, \sigma_i, W_i, J_i)\}_{i \notin Q}$ that are not replicate with others, $\mathcal{C}$ verifies the signatures $\{\sigma_i\}_{i \notin Q}$ by checking whether
$$\hat{e}(\prod_{i \notin Q} \sigma_{i2}, g) \prod_{i \notin Q} \hat{e}(H(\mathcal{N}, \mathcal{S}_i, P_i), \sigma_{i1}) \stackrel{?}{=} \prod_{i \notin Q} U_i. \quad (2)$$
If yes, $\mathcal{C}$ accepts the sensing data $\{\mathcal{P}_i\}_{i \notin Q}$ and learns $\{\mathcal{U}_i\}_{i \notin Q}$ are the contributors; otherwise, $\mathcal{C}$ uses the recursive divide-and-conquer approach to find and delete the corrupted data.
- For $(Z_{\hat{i}}, \sigma_Q, W_{\hat{i}}, J_{\hat{i}})$, $\mathcal{C}$ computes
$$\mathcal{S}_{\hat{i}} = SD(\mathcal{H}(W_{\hat{i}}, W_{\hat{i}}^k), J_{\hat{i}}), \quad \mathcal{P}_{\hat{i}} = \mathcal{SD}(\mathcal{H}(\mathcal{N}, \mathcal{S}_{\hat{i}}), Z_{\hat{i}}).$$
After obtaining the sensing data $\mathcal{P}_{\hat{i}}$, $\mathcal{C}$ verifies the signature σ_Q by checking whether
$$\hat{e}(\sigma_{Q2}, g)\hat{e}(H(\mathcal{N}, \mathcal{S}_{\hat{i}}, P_{\hat{i}}), \sigma_{Q1}) \stackrel{?}{=} \prod_{i \in Q} U_i. \quad (3)$$
If yes, $\mathcal{C}$ accepts the sensing data $\mathcal{P}_{\hat{i}}$ and learns $\{\mathcal{U}_i\}_{i \in Q}$ are the contributors of $\mathcal{P}_{\hat{i}}$; otherwise, $\mathcal{C}$ deletes it.

Finally, $\mathcal{C}$ obtains the sensing data $(\{\mathcal{P}_i\}_{i \notin Q}, \mathcal{P}_{\hat{i}})$ and distributes the benefits to mobile users in $\mathbb{U}$ based on their contributions.

C. Security Discussion

Secure Data Deduplication: To deduplicate sensing reports and achieve data confidentiality, a BLS-OPRF scheme is designed to compute the encryption key from the sensing data. Thus, a fog node is able to detect the replicate data based on the ciphertexts, which are identical if the sensing data is equal. The Fo-SDD not only supports the deduplication of sensing reports, but also achieves high security guarantee on sensing data.

- Security against Brute-Force Attacks: The encryption key $\mathcal{S}_i$ is secret that no adversary is able to distinguish it from a random value, except $\mathcal{F}_j$. Having $(H(\mathcal{P}_i)^{s_i}, H(\mathcal{P}_i)^{s_i x_j})$, it is hard to compute $H(\mathcal{P}_i)^{x_j}$; otherwise, the Computational Diffie-Hellman (CDH) problem [15] is intractable. Moreover, since the external attackers do not have the secret key of $\mathcal{F}_j$, they cannot guess the sensing data $\mathcal{P}_i$. Thus, the sensing data is confidential against brute-force attacks. Nonetheless, since the Decisional Diffie-Hellman (DDH) problem [15] is tractable, it is possible for $\mathcal{F}_j$ to obtain $\mathcal{P}_i$ by using brute-force attacks, that is, to guess a $\mathcal{P}_i'$ and test whether $\hat{e}(H(\mathcal{P}_i'), S_i') = \hat{e}(S_i, \mathcal{S}_i)$ holds or not. To prevent this attack from $\mathcal{F}_j$, we employ multiple neighboring fog nodes to cooperatively generate S_i' for $\mathcal{U}_i$. Thereby, a single fog node $\mathcal{F}_j$ cannot launch brute-force attacks to acquire $\mathcal{P}_i$, unless all neighboring fog nodes collude to guess. Certainly, to achieve higher security guarantee against brute-force attacks, it is possible to employ a trusted key server to generate encryption keys for all mobile users, such as a cellular service provider or network operator.
- Security against "Duplicate-Replay" Attacks: To prevent lazy mobile users from replaying other users' sensing reports, we ensure that only the mobile users possessing sensing data can generate valid sensing reports. Specifically, $\mathcal{U}_i$ needs to use the sensing data $\mathcal{P}_i$ to generate the signature σ_i, which would be verified by the customer in Data-Reading phase. If $\mathcal{U}_i$ replays a captured report Z_i without possessing $\mathcal{P}_i$, the misbehavior can be detected by the customer. Therefore, as long as the key homomorphic signature [16] is unforgeable, Fo-SDD is secure against "duplicate-replay" attacks.

Efficient Contribution Claim: To claim the contribution, $\mathcal{U}_i$ utilizes the key-homomorphic signature scheme to generate σ_i on $\mathcal{P}_i$, such that $\mathcal{C}$ can confirm whether $\mathcal{U}_i$ is the contributor

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2791432, IEEE Transactions on Dependable and Secure Computing

of $\mathcal{P}_i$ or not by verifying σ_i. If some mobile users $\{\mathcal{U}_i\}_{i\in Q}$ upload the same data $\mathcal{P}_{\hat{i}}$, $\mathcal{F}_j$ aggregates the corresponding signatures $\{\sigma_i\}_{i\in Q}$ to generate σ_Q for reducing the communication overhead from $\mathcal{F}_j$ to $\mathcal{C}$. Meanwhile, $\mathcal{C}$ is able to verify the validity of the signature σ_Q with the public keys of $\{\mathcal{U}_i\}_{i\in Q}$. Therefore, $\mathcal{C}$ approves that $\mathcal{P}_{\hat{i}}$ is generated by $\{\mathcal{U}_i\}_{i\in Q}$. Since the key-homomorphic signature [16] achieves existential unforgeability based on the CDH assumption, no attacker can forge the signatures or claim the contributions of the eligible mobile users to itself. Therefore, $\mathcal{C}$ believes that the claimed contributions of mobile users should belong to them indeed.

In addition, it is possible to detect double-reporting and double-retrieving, since both the customer and the CS-server know the identities of mobile users. The CS-server can record the identities when the mobile users deliver their reports and retrieve their rewards. Once a mobile user double-submits reports or double-retrieves rewards, the CS-server can find the misbehavior by checking the records.

V. Extended Fo-SDD Scheme

Since the symmetric encryption scheme used to encrypt sensing data in Fo-SDD is deterministic, it produces the same ciphertext from a given identical plaintext and an encryption key, when it is separately executed by different mobile users. If several mobile users generate the same sensing data $\mathcal{P}_i$, the ciphertexts are identical, that is, Z_i. As a result, any one can learn that two sensing reports are identical, and thereby predict that these mobile uses are in proximate positions or have similar profiles. To prevent privacy leakage of mobile users in duplicate-sensitive applications, we extend the Fo-SDD by means of anonymization. Specifically, we leverage the blind signature [17] to extend the Fo-SDD to prevent malicious hackers or curious entities from learning the identities of participating mobile users. Unfortunately, once the mobile users are anonymous, some problems are emerged. For example, greedy mobile users may submit more sensing reports than allowed to earn unfair benefits; it is difficult for the CS-server to distribute rewards to mobile users; and greedy mobile users may double-draw their rewards from the CS-server. Therefore, we design a contribution claim and reward retrieval mechanism from Chameleon hash function [18] to allow mobile users to claim their contributions and retrieve their rewards fairly. Meanwhile, the CS-server is able to discover the misbehavior of greedy mobile users, and thereby recover their identities.

A. Extended Fo-SDD

1) Service-Setup: The CS-server bootstraps the mobile crowdsensing services following the same procedures as those in the Fo-SDD, except that five more public parameters $(g_0, g_1, G, \mathcal{G}, \mathcal{F})$ are needed. g_0, g_1 are two random generators of group $\mathbb{G}$, G is a random value chosen from $\mathbb{G}_T$, $\mathcal{G} = \hat{e}(g, g)$ and $F : \mathbb{Z}_p \times \{0,1\}^* \to \mathbb{Z}_p$ is a pseudo-random function.

A fog node $\mathcal{F}_j$ randomly chooses $x_j \in \mathbb{Z}_p$ as the secret key and computes $X_j = g^{x_j} \in \mathbb{G}$ as the public key.

A mobile user $\mathcal{U}_i$ randomly picks $v_i \in \mathbb{Z}_p$ as the secret key and computes $U_i = \mathcal{G}^{v_i}$ as the public key. $\mathcal{U}_i$ is required to register at the CS-server to obtain an anonymous credential, which is used to access the crowdsensing services, in the following steps:

- $\mathcal{U}_i$ randomly chooses $u_i' \in \mathbb{Z}_p$ to compute $A_i = g_0^{u_i'} g_1^{v_i}$, and sends (A_i, U_i) to the CS-server, along with the following zero-knowledge proof expressed in Camenisch-Stalder notation [40]:
$$\mathcal{PK}\{(u_i', v_i) : A_i = g_0^{u_i'} g_1^{v_i} \wedge U_i = \mathcal{G}^{v_i}\}.$$
- The CS-server checks $\mathcal{PK}$ to ensure (A_i, U_i) is generated properly. It randomly picks $u_i'', e_i \in \mathbb{Z}_p$ to calculate $B_i = (gA_i g_0^{u_i''})^{\frac{1}{s+e_i}}$ and returns (B_i, u_i'', e_i) to $\mathcal{U}_i$.
- $\mathcal{U}_i$ computes $u_i = u_i' + u_i''$ and checks $\hat{e}(B_i, Sg^{e_i}) \stackrel{?}{=} \hat{e}(gg_0^{u_i} g_1^{v_i}, g)$. If yes, $\mathcal{U}_i$ maintains (B_i, e_i, u_i) along with v_i.

Finally, $\mathcal{U}_i$ obtains an anonymous credential (B_i, e_i, u_i).

2) Task-Releasing: The Task-Releasing phase is the same as that in Fo-SDD.

3) Task-Allocation: The Task-Allocation phase is the same as that in Fo-SDD.

4) Data-Collection: $\mathcal{U}_i$ follows the same operations as those in Fo-SDD to recover $\mathcal{T}_t||\mathcal{T}_e||\mathcal{T}_b$, generate $\mathcal{P}_i$, interact with the CS-server to compute $\mathcal{S}_i$ and encrypt $\mathcal{P}_i$ to obtain Z_i. Furthermore, $\mathcal{U}_i$ randomly chooses $b_i \in \mathbb{Z}_p$ to compute $Y_i = \mathcal{H}(\mathcal{N}, Z_i, t)$, $l_i = F(v_i, \mathcal{N}||t||U_i)$, $V_i = g_0^{l_i}$, $T_i = \mathcal{G}^{v_i} G^{Y_i l_i}$, $H_i = \mathcal{G}^{l_i} U_i^{b_i}$, where t denotes the current reporting period. After that, $\mathcal{U}_i$ picks a random value $a_i \in \mathbb{Z}_p$ to calculate $W_i = g^{a_i}$, $a_i' = \mathcal{H}(W_i, K^{a_i})$, and $J_i = SE(a_i', \mathcal{S}_i)$. Finally, $\mathcal{U}_i$ sends the sensing report $\mathbb{P}_i = (\mathcal{N}, Z_i, V_i, T_i, H_i, W_i, J_i)$ to $\mathcal{F}_j$, along with the following zero-knowledge proof expressed in Camenisch-Stalder notation [40]:

$$\mathcal{SPK}\left\{\begin{array}{l}(B_i, e_i, u_i, v_i, l_i, b_i) : \\ \quad \hat{e}(B_i, Sg^{e_i}) = \hat{e}(gg_0^{u_i} g_1^{v_i}, g) \wedge \\ \quad V_i = g_0^{l_i} \wedge \\ \quad T_i = \mathcal{G}^{v_i} G^{Y_i l_i} \wedge \\ \quad H_i = \mathcal{G}^{l_i} U_i^{b_i}\end{array}\right\}(Z_i).$$

5) Data-Deduplication: Upon receiving $\{\mathbb{P}_1, \cdots, \mathbb{P}_M\}$ from $\mathbb{U}$, $\mathcal{F}_j$ first verifies the validity of $\mathcal{SPK}$ and checks whether there are double-submitted reports. Given two sensing reports, $\mathbb{P}_i = (\mathcal{N}, Z_i, V_i, T_i, H_i, W_i, J_i)$ and $\mathbb{P}'_i = (\mathcal{N}, Z_i', V_i', T_i', H_i', W_i', J_i')$, $\mathcal{F}_j$ computes $Y_i = \mathcal{H}(\mathcal{N}, Z_i, t)$, $Y_i' = \mathcal{H}(\mathcal{N}, Z_i', t)$. If $V_i = V_i'$ and $Y_i \neq Y_i'$, the mobile user double-submits two reports in a time period. $\mathcal{F}_j$ recovers the public key of the greedy mobile user by computing $U_i = (\frac{(T_i)^{Y_i'}}{(T_i')^{Y_i}})^{\frac{1}{Y_i' - Y_i}}$, and deletes one of the reports. If $V_i = V_i'$ and $Y_i = Y_i'$, these reports are the same and submitted by the same mobile user. $\mathcal{F}_j$ keeps one of them. If $Y_i{=}Y_{\hat{i}}$ and $V_i \neq V_i'$, two sensing reports $\mathbb{P}_i$ and $\mathbb{P}_{\hat{i}}$ are identical, but delivered by different mobile users. $\mathbb{P}_i$ and $\mathbb{P}_{\hat{i}}$ are replicate reports. If a set of reports $\{\mathbb{P}_i\}_{i\in Q}$ are replicate, $\mathcal{F}_j$ keeps the first received copy $(Z_{\hat{i}}, W_{\hat{i}}, J_{\hat{i}})$ generated by $\mathcal{U}_{\hat{i}}$ and deletes the replicate copies. The sensing reports that are not replicate with others are $\{\mathbb{P}_i\}$ for each $1 \leq i \leq M$ and $i \notin Q$. Finally, $\mathcal{F}_j$ forwards $(\mathcal{N}, \{(Z_i, H_i, W_i, J_i)\}_{i\notin Q}, Z_{\hat{i}}, W_{\hat{i}}, J_{\hat{i}}, \{H_i\}_{i\in Q})$ to the CS-server.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2791432, IEEE Transactions on Dependable and Secure Computing

When receiving the deduplicated reports $(\mathcal{N}, \{(Z_i, H_i, W_i, J_i)\}_{i\notin Q}, Z_{\hat{i}}, W_{\hat{i}}, J_{\hat{i}}, \{H_i\}_{i\in Q})$ from $\mathcal{F}_j$, the CS-server forwards them to the customer $\mathcal{C}$.

6) Data-Reading: When $\mathcal{C}$ receives the deduplicated reports from the CS-server, $\mathcal{C}$ uses k to decrypt the deduplicated reports and distributes the rewards $\mathcal{T}_b$ to the contributors (mobile users) as follows:

- For each $(Z_i, W_i, J_i) \in \{(Z_i, W_i, J_i)\}_{i\notin Q}$, $\mathcal{C}$ computes $\mathcal{S}_i = SD(\mathcal{H}(W_i, W_i^k), J_i), \quad \mathcal{P}_i = \mathcal{SD}(\mathcal{H}(\mathcal{N}, \mathcal{S}_i), Z_i)$. After recovering $\mathcal{P}_i$, $\mathcal{C}$ checks whether
$$\hat{e}(H(\mathcal{P}_i), X_j) \stackrel{?}{=} \hat{e}(\mathcal{S}_i, g). \tag{4}$$
If yes, $\mathcal{C}$ accepts $\mathcal{P}_i$ and believes $\mathcal{U}_i$ actually generates $\mathcal{P}_i$.
- For $(Z_{\hat{i}}, W_{\hat{i}}, J_{\hat{i}})$, $\mathcal{C}$ computes $\mathcal{S}_{\hat{i}} = SD(\mathcal{H}(W_{\hat{i}}, W_{\hat{i}}^k), J_{\hat{i}}), \quad \mathcal{P}_{\hat{i}} = \mathcal{SD}(\mathcal{H}(\mathcal{N}, \mathcal{S}_{\hat{i}}), Z_{\hat{i}})$. After recovering $\mathcal{P}_{\hat{i}}$, $\mathcal{C}$ checks whether
$$\hat{e}(H(\mathcal{P}_{\hat{i}}), X_j) \stackrel{?}{=} \hat{e}(\mathcal{S}_{\hat{i}}, g). \tag{5}$$
If yes, $\mathcal{C}$ accepts $\mathcal{P}_{\hat{i}}$ and believes $\{\mathcal{U}_i\}_{i\in Q}$ actually generate $\mathcal{P}_{\hat{i}}$.

After obtaining $(\{\mathcal{P}_i\}_{i\notin Q}, \mathcal{P}_{\hat{i}})$, $\mathcal{C}$ determines the rewards that the participating mobile users can acquire based on their contributions. Suppose the mobile user with H_i can earn $\mathcal{B}_i$. $\mathcal{C}$ sends the items $\{(\mathcal{N}, H_i, \mathcal{B}_i)\}_{1\le i\le M}$ to the CS-server.

When a mobile user $\mathcal{U}_i$ retrieves the earned rewards, $\mathcal{U}_i$ sends $(\mathcal{N}, H_i)$ to the CS-server. The CS-server randomly picks $l_i' \in \mathbb{Z}_p$ and returns it to $\mathcal{U}_i$. After receiving l_i', $\mathcal{U}_i$ computes $b_i' = v_i^{-1}(v_i b_i + l_i - l_i')$ and returns b_i' to the CS-server. Then, the CS-server calculates $H_i' = \mathcal{G}^{l_i'} U_i^{b_i'}$, finds the item $(\mathcal{N}, H_i, \mathcal{B}_i)$, in which $H_i' = H_i$, and returns the corresponding rewards $\mathcal{B}_i$ to $\mathcal{U}_i$. In addition, if $\mathcal{U}_i$ tries to double-retrieve the benefit, which means that there is another b_i'' computed by $\mathcal{U}_i$ for a random challenge l_i'', the secret key of $\mathcal{U}_i$ is easy to be recovered by the CS-server as $v_i = \frac{l_i'' - l_i'}{b_i' - b_i''}$.

B. Security Discussion

The extended Fo-SDD only exposes the knowledge that some anonymous mobile users have submitted identical sensing reports. This is the best result supporting data deduplication with high security guarantee currently. Now we discuss the security properties of the extended Fo-SDD.

Secure Data Deduplication: The method to realize data deduplication in the extended Fo-SDD remains the same as that in Fo-SDD. The fog node is able to detect the replicate reports based on the ciphertexts if the sensing data is identical. The improved security of the extended Fo-SDD is analyzed as follows:

- No "Duplicate-Linking" Leakage: In the extended Fo-SDD, we use the blind signature [17] to protect the identities of mobile users and thereby prevent information leakage from the equality of sensing reports. Specifically, in Service-Setup phase, the CS-server generates the anonymous credentials for mobile users using blind signatures and each mobile user utilizes the credential to prove its capability to join crowdsensing activities in Data-Collection phase without exposing its identity. To prove the unforgeability of the credential, we assume that the zero-knowledge proof $\mathcal{SPK}$ is sound, that is, there is an extract algorithm $\mathcal{EX}$ to capture the witness used by the mobile user. In the credential generation query, an adversary can obtain a valid credential on any identity with the aid of a simulator, who can use $\mathcal{EX}$ to extract the witness from the proof. If the adversary can forge a valid credential, the simulator can utilize this credential to forge a valid blind signature within a non-negligible probability. However, since the blind signature is unforgeable under the q-Strong Diffie-Hellman (q-SDH) assumption [17], it is intractable for the adversary to forge a valid credential. Therefore, the mobile users are anonymous in the extended Fo-SDD, as long as the q-SDH assumption holds. In short, even if a curious entity can learn the equality of sensing reports, it cannot link these duplicates to specific mobile users. Therefore, there is no "duplicate-linking" leakage in the extended Fo-SDD.
- Security against "Duplicate-Replay" Attacks: To prevent "duplicate-replay" attacks, $\mathcal{F}_j$ checks whether $V_i = V_i'$ and $Y_i = Y_i'$ in two given sensing reports $\mathbb{P}_i = (\mathcal{N}, Z_i, V_i, T_i, H_i, W_i, J_i)$ and $\mathbb{P}'_i = (\mathcal{N}, Z_i', V_i', T_i', H_i', W_i', J_i')$. If $V_i = V_i'$ and $Y_i = Y_i'$, these reports are the same and from the same mobile user, such that one of the reports may be replayed. Since a greedy mobile user does not have $\mathcal{P}_i$, the user cannot obtain $\mathcal{S}_i$. Thus, the user is unable to generate a new sensing report from a captured one. The greedy mobile user only can replay the captured one, what can be detected by $\mathcal{F}_j$. To ensure all encryption keys can correctly decrypt the sensing reports, $\mathcal{C}$ verifies the recovered data by checking $\hat{e}(H(\mathcal{P}_i), X_j) \stackrel{?}{=} \hat{e}(\mathcal{S}_i, g)$. If the equation holds, the sensing data is recovered correctly. Therefore, the extended Fo-SDD is secure against "duplicate-replay" attacks.

Efficient Contribution Claim: In the extended Fo-SDD, we allow honest anonymous mobile users to claim the contributions and retrieve the rewards from the CS-server, and prevent a greedy mobile user from double-reporting the sensing data or double-retrieving rewards. The security of contribution claim is discussed as follows:

- Detection of Double-Reporting: We design a double-reporting tag $T_i = \mathcal{G}^{v_i} G^{Y_i l_i}$ for each sensing report. If a greedy mobile user $\mathcal{U}_i$ submits two sensing reports in a reporting period to the CS-server, there are two different pairs (Y_i, T_i) and (Y_i', T_i'), but the same l_i in two reports. Having (Y_i, T_i) and (Y_i', T_i'), it is easy to recover the public key of $\mathcal{U}_i$, that is, $U_i = (\frac{(T_i)^{Y_i'}}{(T_i')^{Y_i}})^{\frac{1}{Y_i' - Y_i}}$. In addition, the CS-server cannot slander an honest mobile user. To do it, a new T_i' should be computed for the CS-server. Nonetheless, it is difficult for the CS-server to compute T_i' without a valid l_i. Therefore, if the pseudo-random function F is secure, the CS-server cannot successfully slander an honest mobile user.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2791432, IEEE Transactions on Dependable and Secure Computing

TABLE I
RUN TIME OF FO-SDD (UNIT: MILLISECOND)

Phases	$\mathcal{C}$	CS-server	$\mathcal{F}_j$	$\mathcal{U}_i$
Task-Releasing	10.329	–	–	–
Task-Allocation	–	33.534	18.649	5.732
Data-Collection	–	–	4.847	193.459
Data-Deduplication	–	1.543	6.234	–
Data-Reading	794.624	–	–	–

TABLE II
RUN TIME OF THE EXTENDED FO-SDD (UNIT: MILLISECOND)

Phases	$\mathcal{C}$	CS-server	$\mathcal{F}_j$	$\mathcal{U}_i$
Task-Releasing	11.043	–	–	–
Task-Allocation	–	33.968	19.425	5.653
Data-Collection	–	–	4.275	464.649
Data-Deduplication	–	1.434	6617.835	–
Data-Reading	1435.657	–	–	–

- Detection of Double-Retrieving: Chameleon hash function [18] is employed to enable mobile users to claim their contributions and discover greedy mobile users who double-retrieve the rewards. The Chameleon hash function is $H_i = \mathcal{G}^{l_i} U_i^{b_i}$, which is a secure hash function based on Discrete Logarithm (DL) assumption in group $\mathbb{G}_T$. To retrieve the rewards, $\mathcal{U}_i$ can use its secret key v_i to open the hash function with (b_i', l_i'). However, a greedy anonymous mobile user who double-retrieves the rewards would be traced, when the CS-server has two items (b_i', l_i') and (b_i'', l_i''), that is, $v_i = \frac{l_i'' - l_i'}{b_i' - b_i''}$. Besides, it is also impossible for the CS-server to slander an honest mobile user, since it cannot generate a valid item (b_i'', l_i'') without the user's secret key v_i.

In summary, the extended Fo-SDD supports sensing data deduplication with high security guarantee, and efficient contribution claim with the detection of double-reporting mobile users or double-retrieving mobile users.

VI. PERFORMANCE EVALUATION

In this section, we evaluate the computational and communication overhead of Fo-SDD and extended Fo-SDD, and show the performance of fog-assisted task allocation.

A. Computational Overhead Evaluation

To evaluate the computational overhead, we implement the Fo-SDD and extended Fo-SDD on a notebook with Intel Core i5-4200U CPU and the clock rate is 2.29GHz and the memory is 4.00 GB. The notebook acts as the customer, the CS-server and a fog node. We also use a HUAWEI MT2-L01 smartphone with Kirin 910 CPU and 1250M memory to run the operations on mobile devices. The operation system is Android 4.2.2 and the toolset is Android NDK r8d. We use a version 5.6.1 of MIRACL library to implement number-theoretic based methods of cryptography. The Weiling pairing is utilized to realize the bilinear pairing operation and the elliptic curve is chosen with a base field size of 512 bits. The size of the parameter p is 160 bits. 50 mobile users submit sensing reports, in which 10 reports are replicate. The running time for every entity in the Fo-SDD and the extended Fo-SDD is shown in Table I and Table II, respectively. It seems that the operations in Data-Deduplication phase of the extended Fo-SDD are costly for $\mathcal{F}_j$ to deal with 50 sensing reports simultaneously. But in reality, these reports are received randomly, instead of arriving at the same time. Therefore, it is still efficient for $\mathcal{F}_j$ to respond to the mobile users in Data-Deduplication phase.

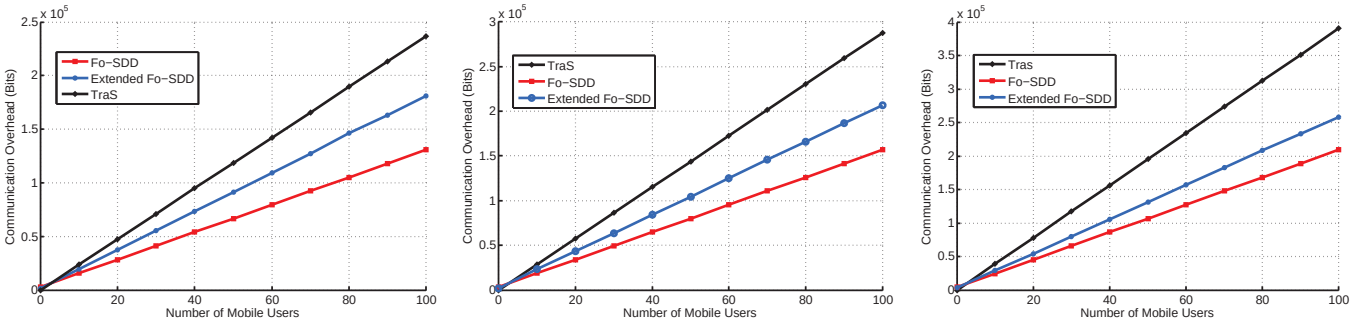
B. Communication Overhead Evaluation

We demonstrate the communication overhead of the Fo-SDD among the CS-server, $\mathcal{C}$, $\mathbb{F}$ and $\mathbb{U}$. The parameter p is set to be 160 bits. When releasing a sensing task $\mathcal{T}$, $\mathcal{C}$ sends $(C_c, K, \mathcal{T}_a)$ to the CS-server, which is $2048 + |\mathcal{T}|$ bits, where $|\mathcal{T}|$ denotes the binary length of $\mathcal{T}$. The CS-server forwards $(\mathcal{N}, C_j', C_2, C_3, K, \mathcal{T}_a)$, whose size is $2720+|\mathcal{T}|$ bits, to each fog node $\mathcal{F}_j \in \mathbb{F}$. After that, $\mathcal{F}_j$ sends $2208 + |\mathcal{T}|$-bit $(\mathcal{N}, D_i, K, \mathcal{T}_a)$ to each mobile user $\mathcal{U}_i \in \mathbb{U}$. After generating the sensing report, $\mathcal{U}_i$ needs to forward $1760 + |\mathcal{P}_i|$-bit $(\mathcal{N}, Z_i, \sigma_i, W_i, J_i)$ to $\mathcal{F}_j$, where $|\mathcal{P}_i|$ is the binary length of $\mathcal{P}_i$. $\mathcal{F}_j$ performs the data deduplication after obtaining the reports from $\mathbb{U}$, and forwards the deduplicated reports $(\mathcal{N}, \{(\widehat{\mathbb{P}}_i, W_i, J_i)\}_{i \notin Q}, Z_{\hat{\jmath}}, \sigma_Q, W_{\hat{\jmath}}, J_{\hat{\jmath}})$ to the CS server, which is of binary length $2208 + 2048(M - |Q|) + |\mathcal{P}_i|(M - |Q| + 1)$ bits, where $|Q|$ is the number of replicate reports in $\{\mathbb{P}_1, \cdots, \mathbb{P}_M\}$. Then, the CS-server sends the deduplicated reports to $\mathcal{C}$. If there is no replicate data in sensing reports, that is, $|Q| = 1$, the communication overhead between $\mathcal{F}_j$ and the CS-server is $160 + 2048M + |\mathcal{P}_i|M$ bits, as well as the burden between the CS-server and the customer $\mathcal{C}$.

The communication overhead of the extended Fo-SDD is low. The data exchanged among $\mathcal{C}$, the CS-server, $\mathcal{F}_j$ and $\mathcal{U}_i$ in Task-Releasing and Task-Allocation phases has the same length with those in the Fo-SDD. In Data-Collection phase, $\mathcal{U}_i$ needs to forward $6368 + |\mathcal{P}_i|$-bit $(\mathcal{N}, Z_i, V_i, T_i, H_i, W_i, J_i, \mathcal{SPK})$ to $\mathcal{F}_j$. $\mathcal{F}_j$ performs the data deduplication and forwards the deduplicated reports $(\mathcal{N}, \{(Z_i, H_i, W_i, J_i)\}_{i \notin Q}, Z_{\hat{\imath}}, W_{\hat{\imath}}, J_{\hat{\imath}}, \{H_i\}_{i \in Q})$ to the CS server, which is of binary length $1184 + 2048(M - |Q|) + |\mathcal{P}_i|(M - |Q| + 1) + 1024|Q|$ bits, where $|Q'|$ is the number of replicate and non-replayed reports in $\{\mathbb{P}_1, \cdots, \mathbb{P}_M\}$. Then, the CS-server forwards the deduplicated reports to $\mathcal{C}$.

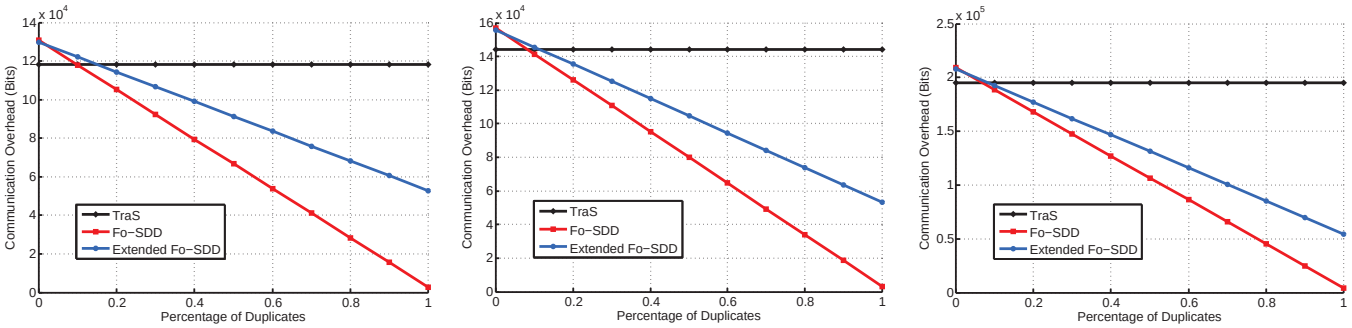
We compare Fo-SDD, extended Fo-SDD and TraS (AES [39] is used to encrypt the sensing data and DSS [39] is used to claim contributions) about the communication overhead between the CS-server and the fog nodes. Fig. 4 shows the comparison of the TraS, Fo-SDD and extended Fo-SDD, when 50% of sensing reports are replicate, and each mobile user delivers one sensing report to fog nodes. The length of sensing reports is 512 bits, 1024 bits and 2048 bits in Fig. 4(a), Fig. 4(b) and Fig. 4(c), respectively. With the increasing number of mobile users participating in crowdsensing activities, Fo-SDD and extend Fo-SDD can reduce a large number of communication overhead compared with the TraS. The Fo-SDD has the best communication efficiency as the replicate sensing data are deleted and the signatures of mobile users who report the replicate data are aggregated. Fig. 5 illustrates the comparison of the TraS, Fo-SDD and extended Fo-SDD about

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2791432, IEEE Transactions on Dependable and Secure Computing



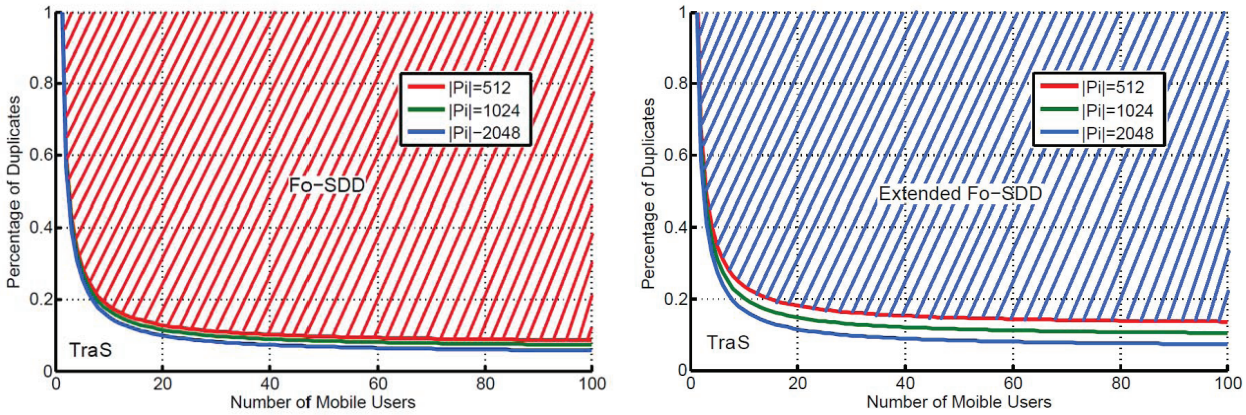
(a) Communication Overhead with $|\mathcal{P}_i|$=512 (b) Communication Overhead with $|\mathcal{P}_i|$=1024 (c) Communication Overhead with $|\mathcal{P}_i|$=2048

Fig. 4. Comparison Results on Communication Overhead between Fog and CS-server with $Q/M = 50\%$



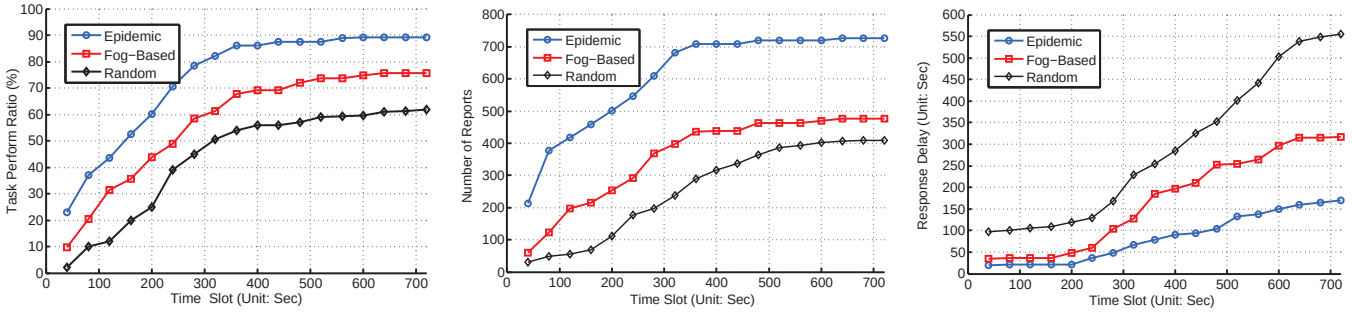
(a) Communication Overhead with $|\mathcal{P}_i|$=512 (b) Communication Overhead with $|\mathcal{P}_i|$=1024 (c) Communication Overhead with $|\mathcal{P}_i|$=2048

Fig. 5. Comparison Results on Communication Overhead between Fog and CS-server with 50 Mobile Users



(a) Relation between TraS and Fo-SDD (b) Relation between TraS and Extended Fo-SDD

Fig. 6. Relation among TraS, Fo-SDD and Extended Fo-SDD



(a) Comparison on Task Perform Ratio (b) Comparison on Number of Reports (c) Comparison on Response Delay

Fig. 7. Performance on Fog-Assisted Task Allocation

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2791432, IEEE Transactions on Dependable and Secure Computing

the communication overhead, when 50 mobile users submit 50 sensing reports. The length of reports is 512 bits, 1024 bits and 2048 bits in Fig. 5(a), Fig. 5(b) and Fig. 5(c), respectively. With the increasing percentage of replicate sensing reports, Fo-SDD and extend Fo-SDD can save plenty of communication bandwidth compared with the TraS. As shown in Fig. 4 and Fig. 5, the Fo-SDD is the most efficient scheme in three ones. Fig. 6 shows the relation among the TraS, Fo-SDD and extended Fo-SDD in terms of communication overhead under the various percentage of replicate data, the report length and the number of mobile users. In Fig. 6 (a), if the points determined by the number of mobile users and the percentage of duplicates are located in the red area, the communication overhead of Fo-SDD between the CS-server and fog nodes is lower than that of TraS; otherwise, TraS is more communication-efficient than Fo-SDD. In Fig. 6 (b), if the points determined by the number of mobile users and the percentage of duplicates are located in the blue area, the extended Fo-SDD is more efficient than the TraS on the communication overhead, and the TraS has lower communication overhead than the extended Fo-SDD, if the points are located in the opposite area. For example, if there are 60 mobile users to deliver 60 sensing reports and the percentage of duplicates is 60%, the Fo-SDD is more efficient than the TraS as shown in Fig 6(a), and the extended Fo-SDD is more efficient than the TraS as shown in Fig 6(b). In addition, the length of sensing reports does not have a big impact on the relations of communication efficiency for the TraS, Fo-SDD and extended Fo-SDD.

C. Performance of Task Allocation

We conduct a simulation to show that the fog-assisted task allocation approach can improve the accuracy of sensing tasks assignment. The simulation is conducted on Infocom06 trace [41], which formalizes the mobility pattern of mobile users. The setting is similar with the simulation in [42]. We compare the fog-assisted task allocation approach with two methods. One is epidemic allocation, in which the CS-server allocates the tasks to all the mobile users connected with and the mobile users perform the tasks straightway; the other is random allocation, where the SC server randomly chooses 5 mobile users to perform the tasks. As shown in Fig. 7(a), Fig. 7(b) and Fig. 7(d), fog-assisted method has a higher task perform ratio, receives more crowdsensing reports and has lower delay to accomplish the tasks than the random allocation. Although the epidemic method can get higher perform ratio and lower delay than the fog-assisted method, the CS-server may receive a large amount of reports that are collected out of the sensing area, which results in the waste of precious bandwidth and storage resources.

VII. Conclusions

In this paper, we have developed a fog-assisted mobile crowdsensing (Fo-MCS) framework to improve the accuracy of task allocation with the aid of fog nodes. We have also proposed a fog-assisted secure data deduplication scheme (Fo-SDD) to reduce the communication overhead between fog nodes and CS-server. The Fo-SDD enables fog nodes to detect and erase the replicate data in sensing reports, and provides high security guarantee against brute-force attacks and "duplicate-replay" attacks. To resist "duplicate-linking" leakage, we have extended the Fo-SDD to hide the identities of mobile users, such that no attacker can link the identical sensing reports to specific mobile users. In addition, we have leveraged Chameleon hash function to achieve contribution claim and reward retrieval for anonymous mobile users. Finally, we have discussed the security and efficiency of the proposed schemes and demonstrated the advantages of the Fo-MCS framework. For the further work, we will investigate location privacy preservation for mobile users in fog-assisted mobile crowdsensing.

References

[1] R.K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: Current state and future challenges," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 32–39, 2011.

[2] Q. Wang, Y. Zhang, X. Lu, Z. Wang, Z. Qin, and K. Ren, "Real-time and spatio-temporal crowd-sourced social network data publishing with differential privacy," *IEEE Trans. Dependable Secur*, to appear.

[3] H. Li, K. Ota, M. Dong, and M. Guo, "Mobile crowdsensing in software defined opportunistic networks," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 140–145, 2017.

[4] L. Pournajaf, L. Xiong, V. Sunderam, and S. Goryczka, "Spatial task assignment for crowd sensing with cloaked locations," in *Proc. MDM*, 2014, pp. 73–82.

[5] H. To, C. Shahabi, and L. Kazemi, "A server-assigned spatial crowdsourcing framework," *ACM Trans. Spatial Algorithms Syst.*, vol. 1, no. 1, pp. 1–28, 2015.

[6] X. Zhang, Z. Yang, W. Sun, Y. Liu, S. Tang, K. Xing, and X. Mao, "Incentives for mobile crowd sensing: A survey," *IEEE Commun. Surv. Tutor.*, vol. 18, no. 1, pp. 54–67, 2016.

[7] J. Wang, "When data cleaning meets crowdsourcing," *Amplab UC Berkeley*, https://amplab.cs.berkeley.edu/, 2015.

[8] Q. Li and G. Cao, "Providing privacy-aware incentives in mobile sensing systems," *IEEE Trans. Mob. Comput.*, vol. 15, no. 6, pp. 1485–1498, 2016.

[9] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-locked encryption and secure deduplication," in *Proc. EUROCRYPT*, 2013, pp. 296–312.

[10] S. Keelveedhi, M. Bellare, and T. Ristenpart, "DupLESS: server-aided encryption for deduplicated storage," in *Proc. Usenix Security*, 2013, pp. 179–194.

[11] X. Yi, B. Athman, G. Dimitrios, S. Andy, and W. Jan, "Privacy protection for wireless medical sensor data," *IEEE Trans. Dependable Secur*, vol. 13, no. 3, pp. 369–380, 2016.

[12] Q. Xu, Z. Su, S. Yu, and Y. Wang, "Trust based incentive scheme to allocate big data tasks with mobile social cloud," *IEEE Trans. Big Data*, to appear.

[13] L. M. Vaquero and L. Rodero-merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *ACM SIGCOMM Comp. Commun. Rev.*, vol. 44, no. 5, pp. 27–32, 2014.

[14] J. Ni, K. Zhang, X. Lin, and X. Shen, "Securing fog Computing for Internet of Things applications: Challenges and solutions," *IEEE Commun. Surv. Tutor.*, to appear.

[15] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," in *Proc. AisaCrypt*, 2001, pp. 514–532.

[16] Q. Wu, Y. Mu, W. Susilo, B. Qin, and J. Domingo-Ferrer, "Asymmetric group key agreement," in *Proc. EUROCRYPT*, 2009, pp. 153–170.

[17] M. H. Au, W. Susilo, and Y. Mu, "Constant-size dynamic k-TAA," in *Proc. SCN*, 2006, pp. 111–125.

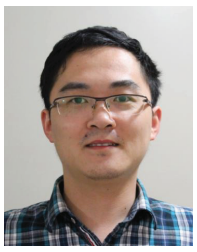
[18] A. Shamir and Y. Tauman, "Improved online/offline signature schemes," in *Proc. CRYPTO*, 2001, pp. 355–367.

[19] K. L. Huang, S.S. Kanhere, and W. Hu, "A privacy-preserving reputation system for participatory sensing," in *Proc. LCN*, 2012, pp. 10–18.

[20] X. O. Wang, W. Cheng, P. Mohapatra, and T. Abdelzaher, "Enabling reputation and trust in privacypreserving mobile sensing," *IEEE Trans. Mob. Comput.*, vol. 13, no. 12, pp. 2777–2790, 2014.

[21] L. Kazemi, C. Shahabi, and L. Chen, "Geotrucrowd: Trustworthy query answering with spatial crowdsourcing," in *Proc. ACM SIGSPATIAL GIS*, 2013, pp. 314–323.

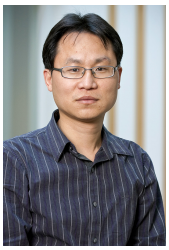
[22] L. Kazemi and C. Shahabi, "Geocrowd: Enabling query answering with spatial crowdsourcing," in *Proc. ACM SIGSPATIAL GIS*, 2012, pp. 189–198.
[23] Y. Shen, L. Huang, L. Li, X. Lu, S. Wang, and W. Yang, "Towards preserving worker location privacy in spatial crowdsourcing," in *Proc. IEEE GLOBECOM*, 2015, pp. 1–6.
[24] L. Wang, D. Zhang, A. Pathak, C. Chen, H. Xiong, D. Yang, and Y. Wang, "CCS-TA: Quality-guaranteed online task allocation in compressive crowdsensing," in *Proc. UBICOMP*, 2015, pp. 683–694.
[25] H. Xiong, D. Zhang, L. Wang, J. P. Gibson, and J. Zhu, "EEMC: Enabling energy-efficient mobile crowdsensing with anonymous participants," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 3, Article 39, 2015.
[26] C. Liu, B. Zhang, X. Su, J. Ma, W. Wang, and K. K. Leung, "Energy-aware participant selection for smartphone-enabled mobile crowd sensing," *IEEE Syst. J.*, vol. 11, no. 3, pp. 1435–1446, 2017.
[27] H. Xiong, D. Zhang, G. Chen, L. Wang, C. Gauthier, and L.E. Barnes, "iCrowd: Near optimal task allocation for piggyback crowdsensing," *IEEE. Trans. Mob. Comput.*, vol. 15, no. 8, pp. 2010–2022, 2016.
[28] M. Bellare and S. Keelveedhi, "Interactive message-locked encryption and secure deduplication," in *Proc. PKC*, 2015, 516–538.
[29] J. Li, Y. K. Li, X. Chen, P. Lee, and W. Lou, "A hybrid cloud approach for secure authorized deduplication," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 5, pp. 1206–1216, 2015.
[30] J. Li, X. Chen, M. Li, J. Li, P. Lee, and W. Lou, "Secure deduplication with efficient and reliable convergent key management," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 6, pp. 1615–1625, 2014.
[31] M. Wen, K. Ota, H. Li, J. Lei, C. Gu, and Z. Su, "Secure data deduplication with reliable key management for dynamic updates in CPSs," *IEEE Trans. Computational Social Systems*, vol. 2, no. 4, pp. 137–147, 2015.
[32] J. Liu, N. Asokan, and B. Pinkas, "Secure deduplication of encrypted data without additional independent servers," in *Proc. ACM CCS*, 2015, pp. 874–885.
[33] H. Cui, X. Yuan, Y. Zheng, and C. Wang, "Enabling secure and effective near-duplicate detection over encrypted in-network storage," in *Proc. IEEE INFOCOM*, 2016, pp. 1–9.
[34] Y. Zheng, X. Yuan, X. Wang, J. Jiang, C. Wang, and X. Gui, "Toward encrypted cloud media center with secure deduplication," *IEEE Trans. Multimedia*, vol. 19, no. 2, pp. 251–265, 2017.
[35] J. Ni, X. Lin, K. Zhang, and Y. Yu, "Secure and deduplicated spatial crowdsourcing: A fog-based approach," in *Proc. IEEE GLOBECOM*, 2016, pp. 1–6.
[36] S. Chang and Z. Chen, "Protecting mobile crowd sensing against sybil attacks using cloud based trust management system," *Mob. Inf. Syst.*, vol. 2016, Article No. 6506341, pp. 1–9, 2016.
[37] A. Faggiani, E. Gregori, L. Lenzini, V. Luconi, and A. Vecchio, "Smartphone-based crowdsourcing for network monitoring: Opportunities, challenges, and a case study," *IEEE Commun. Mag.*, vol. 52, no. 1, pp. 106–113, 2014.
[38] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," In *Proc. NDSS*, 2005, pp. 29–43.
[39] J. Katz and Y. Lindell, "Introduction to modern cryptography," CRC Press, 2014.
[40] J. Camenisch and M. Stadler, "Efficient group signature schemes for large group (extended abstract)," in *Proc. CRYPTO*, 1997, pp. 410–424.
[41] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD trace cambridge/haggle/imote/infocom," 2006.
[42] K. Zhang, X. Liang, J. Ni, K. Yang, and X. Shen, "Exploiting social network to enhance human-to-human infection analysis without privacy leakage," *IEEE Trans. Dependable Secur*, to appear.



Jianbing Ni (S'16) received the B.E. degree and the M.S. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2011 and 2014, respectively. He is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research interests are applied cryptography and information security, with current focus on cloud computing, smart grid and Internet of Things.

Kuan Zhang (S'13-M'17) received the Ph.D. degree in Electrical and Computer Engineering from the University of Waterloo, Canada, in 2016. He was also a postdoctoral fellow with Department of Electrical and Computer Engineering, University of Waterloo, Canada. Since 2017, he has been an assistant professor at the Department of Electrical and Computer Engineering, University of Nebraska-Lincoln, USA. His research interests include security and privacy for mobile social networks, e-healthcare systems, cloud computing and cyber physical systems.

Yong Yu (M'16) is currently a Professor of Shaanxi Normal University, China. He holds the prestigious one hundred talent Professorship of Shaanxi Province as well. He received his Ph.D. degree in cryptography from Xidian University in 2008. He has authored over 60 referred journal and conference papers. His research interests are cryptography and its applications, especially public encryption, digital signature, and secure cloud computing. He is an Associate Editor of Soft Computing.



Xiaodong Lin (M'09-SM'12-F'17) received the Ph.D. degree in information engineering from Beijing University of Posts and Telecommunications, Beijing, China, and the Ph.D. degree in electrical and computer engineering (with Outstanding Achievement in Graduate Studies Award) from the University of Waterloo, Waterloo, ON, Canada. He is currently an Associate Professor with the Department of Physics and Computer Science, Wilfrid Laurier University, Waterloo, ON, Canada. His research interests include wireless network security, applied cryptography, computer forensics, software security, and wireless networking and mobile computing.

Xuemin (Sherman) Shen (M'97-SM'02-F'09) received Ph.D. degrees (1990) from Rutgers University, New Jersey (USA). Dr. Shen is a University Professor, Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research focuses on resource management in interconnected wireless/wired networks, wireless network security, social networks, smart grid, and vehicular ad hoc and sensor networks. Dr. Shen served as the Technical Program Committee Chair/Co-Chair for IEEE Globecom'16, Infocom'14, IEEE VTC'10 Fall, and Globecom'07, the Symposia Chair for IEEE ICC'10. He also serves as the Editor-in-Chief for IEEE Internet of Things Journal, Peer-to-Peer Networking and Application, and IET Communications; a Founding Area Editor for IEEE Transactions on Wireless Communications. Dr. Shen received the Excellent Graduate Supervision Award in 2006, and the Outstanding Performance Award in 2004, 2007, 2010, and 2014 from the University of Waterloo, the Premier's Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada, the Distinguished Performance Award in 2002 and 2007 from the Faculty of Engineering, University of Waterloo, the Joseph LoCicero Award and the Education Award 2017 from the IEEE Communications Society. Dr. Shen is a registered Professional Engineer of Ontario, Canada, an IEEE Fellow, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, and a Distinguished Lecturer of IEEE Vehicular Technology Society and Communications Society.