Int. J. of Multimedia Intelligence and Security

Privacy-preserving Data Deduplication in Edge-assisted Mobile Crowdsensing

Yili Jiang, Kuan Zhang, and Yi Qian

Department of Electrical and Computer Engineering
University of Nebraska-Lincoln
Omaha, NE, USA
E-mail: yilijiang@huskers.unl.edu
E-mail: kuan.zhang@unl.edu
E-mail: yi.qian@unl.edu

Rose Qingyang Hu

Department of Electrical and Computer Engineering
Utah State University
Logan, UT, USA
E-mail: rose.hu@usu.edu

Abstract: Mobile crowdsensing enables the collaborative data collection between mobile workers and centralized cloud server. When sensing data from the surrounding environment, workers in the same location may generate the identical data report. Although edge intelligence is integrated to remove the redundant data by comparing the report content, disclosing the sensing data to the edge nodes results in severe privacy leakage. To detect and remove duplicated data without revealing the content, encryption based data deduplication schemes are the main solutions. However, the existing schemes have high computational cost due to heavy cryptographic primitives. In this work, we propose a pairing-based data deduplication scheme with lower computational cost. The proposed scheme guarantees both secure data deduplication and secure contributor identification. In addition, by deploying proxy re-encryption, the privacy of task location is preserved. The experimental results demonstrate that the proposed scheme achieves better computational efficiency than the other schemes.

Keywords: Privacy preservation, data deduplication, edge intelligence, computational efficiency, mobile crowdsensing.

Biographical notes:
Yili Jiang received her B.S. degree in Electrical and Information Engineering from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2014. She is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering at University of Nebraska-Lincoln. Her research interests include cybersecurity, information privacy, intelligent Internet of Things, and machine learning.
Kuan Zhang is an assistant professor in Department of Electrical and Computer Engineering at University of Nebraska–Lincoln, USA. In 2016, He received his Ph.D. degree from University of Waterloo, Canada in Electrical and Computer Engineering. He has published over 100 papers in journals and conferences. He was the recipient of Best Paper Award in IEEE WCNC 2013, Securecomm 2016 and ICC 2020. His research interests include cyber security, big data, Internet-of-things, cloud/edge computing.

Yi Qian received a Ph.D. degree in electrical engineering from Clemson University, South Carolina in 1996. He is currently a professor in the Department of Electrical and Computer Engineering, University of Nebraska-Lincoln (UNL). His research interests include communications and systems, and information and communication network security. Prof. Yi Qian is a Fellow of IEEE. He was previously Chair of the IEEE Technical Committee for Communications and Information Security. He was the Technical Program Chair for IEEE International Conference on Communications 2018. He serves on the Editorial Boards of several international journals and magazines, including as the Editor-in-Chief for IEEE Wireless Communications. He was a Distinguished Lecturer for IEEE Vehicular Technology Society. He is currently a Distinguished Lecturer for IEEE Communications Society.

Rose Qingyang Hu is a Professor in the Electrical and Computer Engineering Department and Associate Dean for research of College of Engineering at Utah State University. She also directs Communications Network Innovation Lab at Utah State University. Her current research interests include next-generation wireless system design and optimization, Internet of Things, cyber physical system, mobile edge computing, V2X communications, artificial intelligence in wireless networks, wireless system modeling and performance analysis. Prof. Hu received the B.S. degree from the University of Science and Technology of China, the M.S. degree from New York University, and the Ph.D. degree from the University of Kansas. Besides a decade academia experience, she has more than 10 years of R&D experience with Nortel, Blackberry, and Intel as a Technical Manager, a Senior Wireless System Architect, and a Senior Research Scientist, actively participating in industrial 3G/4G technology development, standardization, system level simulation, and performance evaluation. She has published extensively in top IEEE journals and conferences and also holds numerous patents in her research areas. Prof. Hu is currently serving on the editorial boards of the IEEE Transactions on Wireless Communications, the IEEE Transactions on Vehicular Technology, the IEEE Communications Magazine and the IEEE Wireless Communications. She also served as the TPC Co-Chair for the IEEE ICC 2018. She is an IEEE Communications Society Distinguished Lecturer Class 2015-2018 and a recipient of prestigious Best Paper Awards from the IEEE GLOBECOM 2012, the IEEE ICC 2015, the IEEE VTC Spring 2016, and the IEEE ICC 2016. Prof. Hu is a Fellow of IEEE and a member of Phi Kappa Phi Honor Society.

1 Introduction

With the development of smart devices and cloud/edge computing, mobile crowdsensing (MCS) has been a promising paradigm to encourage the mobile workers to participate in data collection. Equipped with intelligent devices, such as smartphones, cameras, tablets, and so on, the mobile workers (also known as mobile individuals, mobile nodes) in MCS have sufficient abilities to collect data from their surrounding environment, being able to facilitate diverse applications. For example, smartphones and cameras can sense the spatial data of a tourist area to assist with plane reconstruction and passenger flow management Ni et al. (June 2020). The build-in sensors of a vehicle can collect road information for traffic monitoring, parking vacancy discovery, and road surface condition inspection Basudan et al. (2017), C. Wang et al. (2019), Abdul Rahman et al. (2019), Bock et al. (2020). Instead of deploying and maintaining sensor devices in specific area, MCS recruits mobile workers

to collect data. Therefore, compared with traditional sensor networks, MCS has significant financial advantages, especially for large-scale or short-term data collection.

Despite the appealing benefits, MCS faces the following challenges in terms of system efficiency and privacy preservation.

1) In the classical MCS, the mobile workers collect and upload data to the centralized cloud server for data aggregation. The remote data transmissions and centralized data processing not only consume excessive communication/computational resources, but also result in longer latency to the system. To solve the problem, edge computing is integrated into MCS Ni et al. (2017), Lamaazi et al. (2020). Specifically, by deploying edge nodes closer to the mobile workers, the sensed data is transmitted to the edge nodes instead of the remote cloud server. The edge nodes then aggregate the data and upload the results to the cloud server. In this way, the communication/computational overheads in the system are reduced.

2) Since most of tasks in MCS are location-dependent, workers need to upload their locations to the cloud server for task competition and task allocation. However, location information is sensitive for mobile workers. Disclosing the real-time locations to an untrusted cloud server may lead to trajectory tracking, threatening the personal security of mobile workers. Therefore, preserving location privacy of workers is essential in MCS. Shen et al. (2015) and Sucasas et al. (2020) proposed to deploy encryption techniques to protect the workers' locations in task allocation. P. Zhou et al. (2019), Z. Wang et al. (2019), and Wang et al. (2021) employed differential privacy to preserve location privacy for mobile workers. Specifically, before uploading locations to the cloud server, mobile workers add noise data to their location coordinates to obfuscate the information. However, considering the fact that workers perform task within the task area, once the task location is disclosed, the approximate locations of workers are revealed. Thus, to protect the workers' locations, it is also essential to protect the task location. Unfortunately, few of the existing schemes can achieve privacy preservation of task location.

3) The mobile workers may generate duplicated reports Li et al. (2021). For instance, when measuring the users' experiences with a media service (e.g., web browsing, image downloading, video watching) L. Zhou et al. (2019), the workers may choose the same Quality-of-Experience (QoE) level and return the same feedback. Thus, the generated report could be identical. The transmission of the duplicated reports results in unnecessary resource consumption. To save the communication resource, the edge node detects and discards the identical reports in data aggregation. However, detecting the identical reports requires the edge node to read and compare the content of the reports, disclosing the sensed data to the edge node. Since the sensed data may contain sensitive information of workers (e.g., locations, health data, web browsing history, etc.), disclosing the report content to the edge node violates the privacy of workers. To tackle this problem, encryption based deduplication is proposed, where the edge nodes can detect identical reports based on the ciphertext Cui et al. (2016), Zheng et al. (2017). In addition, although the duplicated reports are deleted, the contributions of the workers who generate the duplicated report should not be neglected. To record the contributors, Ni et al. (2016) and Jiang et al. (2018) designed secure signature aggregation in the encryption-based deduplication scheme. Although the above encryption based schemes guarantee secure data deduplication and contributor identification, they have high computational costs, resulting in system efficiency degradation.

From the above challenges, we are motivated to propose a privacy-preserving data deduplication scheme for edge-assisted MCS (EMCS). The proposed scheme improves system efficiency while guaranteeing privacy preservation of task location. The main contributions of this paper are summarized as follows.

- We propose a pairing-based scheme for privacy-preserving data deduplication in EMCS. By employing proxy re-encryption, the task location is confidential from the honest-but-curious cloud server and edge nodes, achieving the privacy preservation of task location. In addition, the scheme enables efficiency signature aggregation and verification in data deduplication, reducing the computational costs of the system.
- We provide detailed discussion about the achieved security and privacy. Besides task location preservation, the proposed scheme achieves secure data deduplication and secure contributor identification.
- We provide both theoretical analysis and experimental discussion about the computational efficiency of the proposed scheme. Compared with the other schemes, the experimental results show that our proposed scheme reduces the computational time significantly.

The rest of this paper is organized as follows. In section 2, we discuss the related work. In section 3, we describe the system model, security model and design goals. After that, we provide the preliminaries in section 4 and describe the proposed scheme in section 5. The security discussion of the proposed scheme is provided in section 6. Subsequently, we discuss the experimental results in section 7 and conclude the work in section 8.

2 Related Work

In this section, we discuss the related work for location privacy preservation and secure data deduplication in EMCS.

- *Location privacy preservation*: Shen et al. (2015) proposed a homomorphic encryption based scheme to protect location information from the honest-but-curious cloud server in task allocation. Benefiting from the additive property of homomorphic encryption, the cloud server is able to allocate location-dependent task to the workers without knowing their real locations. Sucasas et al. (2020) designed a group signature based scheme for location privacy preservation. Especially, the workers are divided into groups based on their locations. The workers in the same group share a group key which can be used to sign messages on behalf of a group. Thus, the cloud server can only verify that the worker is from a group but cannot reveal its real location. However, the above encryption based schemes have high computational overhead. To improve the computation efficiency, Z. Wang et al. (2019) proposed a differential privacy based scheme, where the mobile workers obfuscate their locations by introducing noise data into the location coordinates. Similar schemes are proposed by P. Zhou et al. (2019) and Wang et al. (2021). Although the location information of mobile workers is preserved in these schemes, the task location is public to the system. Considering the fact the mobile workers execute task within the task area, the disclosing of task location reveals the approximate locations of mobile workers.

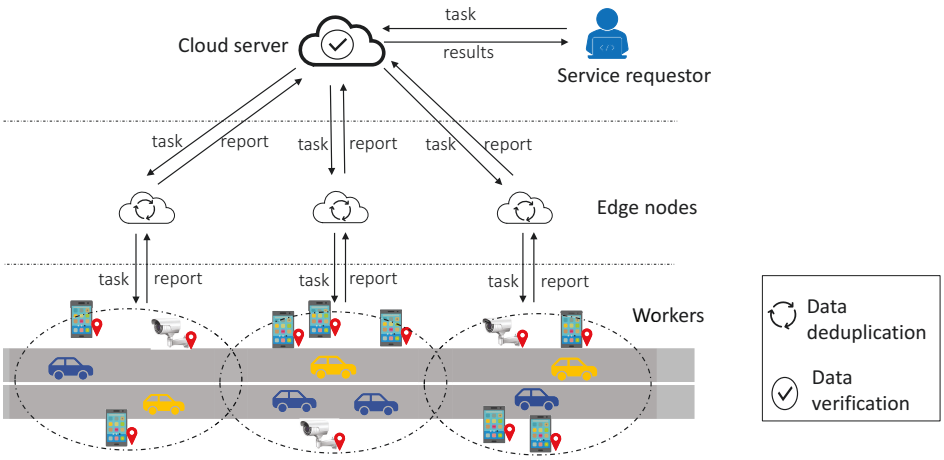


Figure 1: System Model.

- *Secure data deduplication*: To achieve secure data deduplication, Ni et al. (2016) proposed a message-locked encryption based scheme for data deduplication. In their scheme, the edge nodes can detect and remove duplicated reports without revealing the content of the report, guaranteeing data confidentiality. In addition, even the duplicated reports are deleted, the contributors who generate the duplicated reports can be identified and aggregated by the edge nodes. Jiang et al. (2018) improved the computational efficiency in the scheme of Ni et al. (2016) by applying symmetric key based encryption instead of asymmetric key based encryption. Subsequently, Ni et al. (May 2020) enhanced the security of their previous scheme by protecting from brute-force attack and duplicate-replay attack. The security improvement has sacrifice in computational efficiency. Jiang et al. (2021) also enhanced the security of their previous scheme. Similarly, the security improvement has sacrifice in computational efficiency.

According to the above discussions, protecting task location and improving the computational efficiency of secure data deduplication are challenging in EMCS. In this paper, we design a pairing-based scheme for privacy-preserving deduplication. In the proposed scheme, the task location is confidential from the cloud server and edge nodes. Moreover, the data deduplication achieves better computational efficiency compared with the existing schemes.

3 System Model

In this section, we introduce the system overview, security model, and design goals.

3.1 System Overview

Fig. 1 shows the architecture and workflows of the edge-assisted mobile crowdsensing. Generally, the system consists of four types of entities, including the service requestor, cloud server, edge nodes, and mobile workers. After receiving a task (e.g., collect the traffic

data in a specific area) from the service request, the cloud server recruits mobile workers to sense the data with the assistance of edge nodes. The mobile workers utilize the smart devices (e.g., smart phones) to collect the data and upload the data report to the cloud server. The details of the four types of entities and workflows are described as follows.

- *Service requestor*: The service requestor can be a company, or an organization that aims to collect sufficient data for data mining or data analysis. For instance, a company intends to analyze the city-scale traffic flow by collecting the real-time traffic data. However, the service requestor may have limited labor and financial budget to maintain a large-scale sensor network, being unable to collect sufficient data by itself. Therefore, the service requestor outsources the task to the cloud server.

- *Cloud server*: The cloud server provides services for the service requestor. Since the cloud server has powerful computational and communication resources, it can recruit mobile workers from the task area to collect data. After receiving the collected data, it aggregates data and returns the results to the service requestor. For the simplification of expression, we may use “cloud" to denote “cloud server" in this paper.

- *Edge nodes*: The edge nodes provide assistance for cloud in task allocation, worker recruitment, information aggregation, and data transmission. The edge nodes are deployed closer to the mobile workers and are assumed to have sufficient computational/communication resources.

- *Mobile workers*: The mobile workers are individuals that are equipped with smart devices and various sensors. When assigned a piece of task, the mobile workers sense data from their surroundings and upload the data reports to edge nodes.

Without loss of generality, we consider the following steps in the system model.

- *Task allocation*: When a service requestor releases a task to the cloud, the cloud allocates the task to mobile workers by cooperating with the edge nodes. Specifically, the cloud releases the task to the edge nodes that are located in the task area. The edge nodes then select mobile workers within its coverage to perform the task. When multiple workers are interested in the task, they upload their information to the edge nodes for competition. Generally, workers that have good reputations and higher sensing abilities are more likely to be recruited. It is noted that how to select the best winners is not the goal of this paper. For the stage of task allocation in this work, we focus on the privacy preservation rather than winner selection.

- *Data collection*: Once a mobile worker is selected to perform the task, it senses data from its surrounding environment following the task requirements. Based on the sensed data, the mobile worker generates a data report and uploads the report to the corresponding edge node. The data reports from different workers can be identical when the workers are working in the same area. For instance, when multiple workers collect data of air quality in the same area, the generated reports can be identical since the sensing results are the same.

- *Data deduplication*: After receiving data reports from mobile workers, the edge node aggregates the reports to reduce communication overhead. In details, the edge node may receive numerous data reports from the workers in large-scale crowdsensing. If the edge node transmits all the reports to the remote cloud server, the communication

costs can be heavy. As the different workers may upload the identical reports, the communication costs can be significantly reduced if the redundant reports are removed. Thus, in this step, the edge node checks the received data reports and deletes the duplicated reports. The deduplicated reports are then uploaded to the cloud server.

- *Data verification*: After receiving data reports from the edge nodes, the cloud server performs authentication to verify that the reports are from legitimate mobile workers. If the reports pass the verification, the cloud aggregates into the final report and returns to the service requestor. Otherwise, the cloud discards the illegitimate reports.

- *Data reading*: After receiving the final report, the service requestor reveals the sensed data from the final report. With the sensed data, the service requestor can further execute data mining or data analysis. Based on the quality of the sensed data, the service requestor issues rewards to the cloud and the cloud distributes the rewards to the workers who contribute to the reports.

3.2 Security Model

- The service requestor is fully trusted. It always provides reliable information to the system and performs activities honestly.

- The cloud and the edge nodes are honest-but-curious. Specifically, they follow the rules to perform activities honestly. However, they are curious about the task location and the sensed data. They may attempt to reveal the sensed data or the task location by analyzing the received reports. The collusion between the cloud and edge nodes are not considered in this paper.

- The mobile workers may be malicious to launch duplicate-replay attack. Specifically, the malicious workers attempt to get rewards without performing data collection. To achieve this, a worker eavesdrops and captures the data reports that are sent from the others to the cloud/edge nodes. The worker then uploads the captured report to cheat the cloud/edge nodes that he/she generates the identical report with the others and contributes to the task.

3.3 Design Goals

- *Privacy preservation of task location*: The task location is confidential from the cloud and edge nodes. The task location can be sensitive for both service requestor and mobile workers. For instance, a hospital may collect the health data from the residents living in downtown to study the relationship of noise and mental health. Once the task location is revealed, the approximate locations of mobile workers are exposed due to the fact that the workers perform task within the task area. Since the location is private for mobile workers, the location privacy leakage may reduce their enthusiasm in participating the task. Therefore, it is essential for hospital to protect the task location from being reveled by the cloud and edge nodes.

- *Secure data deduplication*: The edge nodes are able to delete redundant data reports without revealing the content of the reports. In other words, the edge nodes can perform data deduplication. However, the content of the data is confidential from the edge nodes.

Table 1 Notation Definitions

Variable	Definition
(pk, sk)	key pair of public key and private key
l	security parameter, which is a large integer
$\mathbb{Z}_p^\star$	$\{1, 2, 3, ..., p-1\}$
H_1, H_2	hash functions
$\mathcal{C}$	cloud server
$\mathcal{E}_i$	edge node
$\mathcal{W}_j$	mobile worker
$\mathcal{R}$	service requestor
M	the number of edge nodes
N	the number of mobile workers
T	task content
Q	the set of indices of the selected edge nodes
U	the set of indices of the selected mobile workers
D	the set of indices of the duplicated reports
$\|D\|$	the number of the duplicated reports
Enc	the symmetric key based encryption
Dec	the symmetric key based decryption

- *Secure contributor identification*: The cloud and edge nodes can collaborate to aggregate the contributors of the data reports. On the one hand, to provide fairness in MCS, although the redundant reports are deleted in data deduplication, the workers who generate the redundant reports should still be identified as the contributors. On the other hand, when a malicious worker uploads a captured report from another worker, the cloud and edge nodes are able to detect this. Thus, the malicious worker cannot cheat the system that he/she is a contributor. All the real contributors can be identified without revealing the content of their sensing reports.

- *Efficient computation*: The complete scheme should be computational efficient. Especially, when considering large scale or real-time data collection in crowdsensing, heavy computational overhead leads to long latency and data inaccuracy. Therefore, efficient computation is essential for the system.

4 Preliminaries

In this section, we review the bilinear pairing and proxy re-encryption that are used to design our proposed scheme.

Bilinear pairing: Let $\mathbb{G}_1$ be a multiplicative cyclic group with a prime order p. $\mathbb{G}_T$ is a multiplicative cyclic group with the same order p. A mapping $\hat{e}$: $\mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$ is a bilinear pairing that satisfies the following properties.

- Bilinearity: $\hat{e}(g^a, h^b) = \hat{e}(g, h)^{ab}$, given $g, h \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_p^\star$.

- Non-degeneracy: $\hat{e}(g, g) \neq 1$, where $g \in \mathbb{G}_1$ and $g \neq \infty$.

- Computability: For all $g, h \in \mathbb{G}_1$, $\hat{e}(g, h)$ is efficiently computable.

Proxy re-encryption: Proxy re-encryption is a promising way for an untrusted proxy to convert a ciphertext of Alice to a ciphertext for Bob without revealing the plaintext. Assume that the private key and public key of Alice are (sk_a, pk_a), where $sk_a = a \in \mathbb{Z}_p^\star, pk_a = g^a$. Similarly, the key pair of Bob is (b, g^b). The details of proxy re-encryption involve the followings.

- Alice encrypts the message m by computing $c_a = (g^{ak}, m\hat{e}(g, g)^k)$, where k is a random number in $\mathbb{Z}_p^\star$. c_a is uploaded to and stored at the proxy.
- When Alice intends to share the message m with Bob, she computes $rk = g^{b/a}$ with Bob's public key and sends rk to the proxy.
- The proxy computes $c_b = (\hat{e}(g, g)^{bk}, m\hat{e}(g, g)^k)$, where $\hat{e}(g, g)^{bk} = \hat{e}(g^{ak}, g^{b/a})$. c_b is transmitted to Bob.
- Bob can reveal m with his private key by computing $m = m\hat{e}(g, g)^k / (\hat{e}(g, g)^{bk})^{1/b}$. Thus, the message is shared from Alice to Bob without exposing to the proxy.

5 Proposed Scheme

In this section, we first introduce the overview of the proposed scheme. Then we describe the scheme in details. The notations used though this paper are listed in Table 1.

5.1 Overview of the Proposed Scheme

As illustrated in Fig. 2, the proposed scheme includes stages of initialization, task allocation, data collection, data deduplication, data verification, and data reading. In the stage of initialization, all the entities in the system generate their key pair. In addition, the edge nodes encrypt their locations and upload the ciphertext $\mathbf{C}_{e_i}^1$ to the cloud. In the task allocation, when the service requestor launches a task, the edge nodes collaborate with the cloud to further encrypt the ciphertext $\mathbf{C}_{e_i}^1$ into a new ciphertext. By decrypting the new ciphertext, the service requestor receives the locations of edge nodes. Then the service requestor selects edge nodes whose coverage is within the task location and returns the list of the selected edge nodes to the cloud server. The cloud then releases the task requirements to the selected edge nodes and the edge nodes allocate task to the recruited workers. In the stage of data collection, the recruited workers collect data from the surrounding environment and upload the sensing reports to the edge nodes. Then in the stage of data deduplication, the edge nodes delete the duplicated reports and aggregate the signatures of the workers who generate the identical reports. In the stage of data verification, the cloud server verifies the aggregated signatures and identifies the contributors of sensing task. After that, the cloud sends the final report to the service requestor. In the stage of data reading, the service requestor reveals the content of the final report and achieves the collected data.

5.2 Details of the Proposed Scheme

(1) *Initialization*:

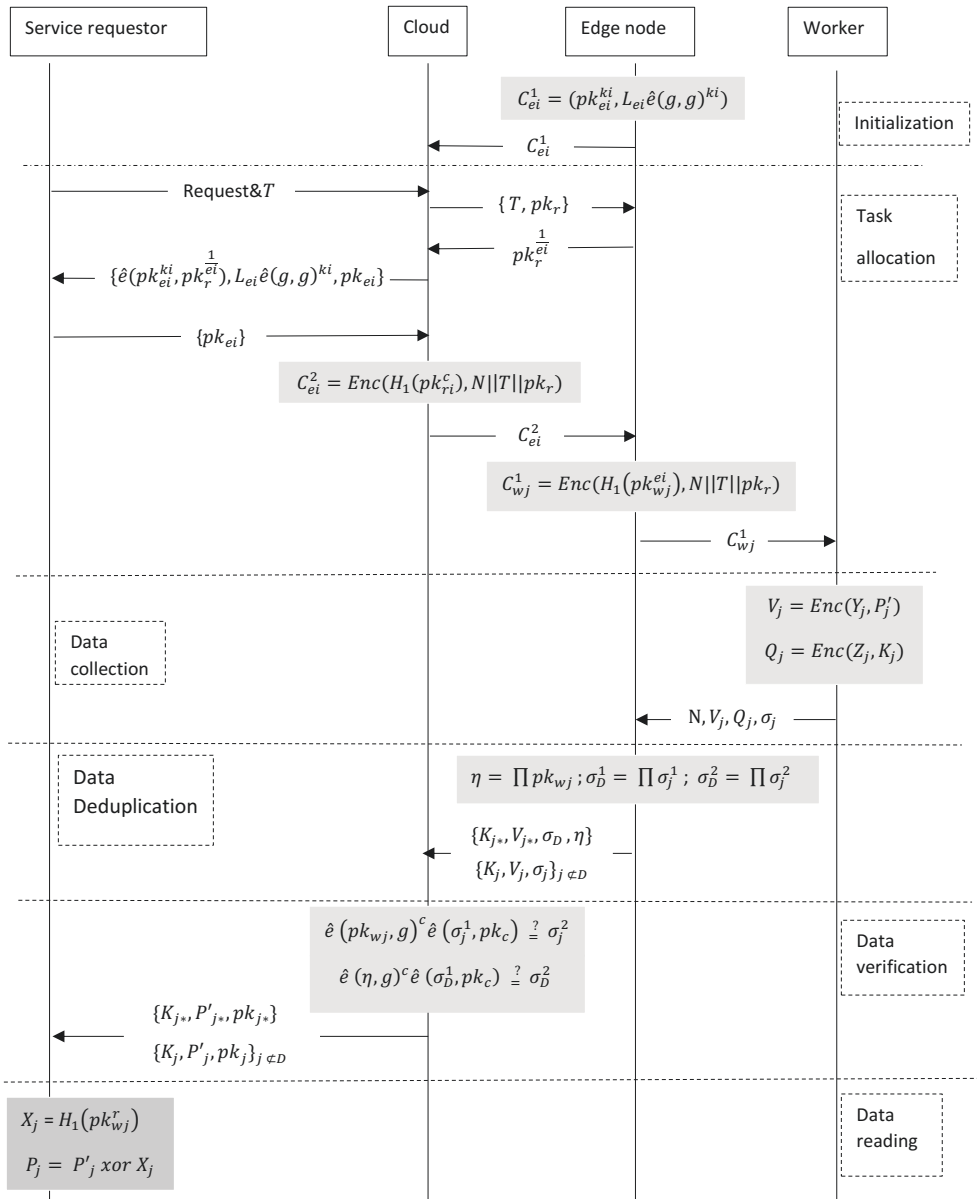


Figure 2: Proposed protocol.

Let $\mathbb{G}_1$ be a multiplicative cyclic group with a prime order $p > 2^l$, where l is the security parameter. $\mathbb{G}_T$ denotes a multiplicative cyclic group with the same order p. g is a generator of $\mathbb{G}_1$ and $\hat{e}$: $\mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$ is a bilinear pairing operator. The system parameters $(l, p, \mathbb{G}_1, \mathbb{G}_T, \hat{e}, H_1, H_2)$ are public to all entities, where H_1 and H_2 are two secure hash functions that $H_1 : \mathbb{G}_1 \rightarrow \mathbb{Z}_p^\star$, $H_2 : \{0,1\}^\star \rightarrow \mathbb{Z}_p^\star$. $\{0,1\}^\star$ denotes a sequence of binary numbers.

With the system parameters, cloud server $\mathcal{C}$ chooses a random number $c \in \mathbb{Z}_p^\star$ as its private key sk_c and calculates $pk_c = g^c$ as the corresponding public key. For each registered edge node $\mathcal{E}_i (i \in [1, M])$, it chooses a random number $e_i \in \mathbb{Z}_p^\star$ as its private key sk_{e_i} and generates $pk_{e_i} = g^{e_i}$ as the corresponding public key. For each registered mobile worker $\mathcal{W}_j (j \in [1, N])$, it chooses a random number $w_j \in \mathbb{Z}_p^\star$ as its private key sk_{w_j} and generates $pk_{w_j} = g^{w_j}$ as the corresponding public key. For all the entities, they publish their public key to the system and keep their private key as a secret.

For each edge node $\mathcal{E}_i$, its location and coverage area are denoted as L_{e_i}. After key generation, $\mathcal{E}_i$ randomly choose an integer $k_i \in \mathbb{Z}_p^\star$ and encrypts L_{e_i} as follows,

$$\mathbf{C}_{e_i}^1 = (pk_{e_i}^{k_i}, L_{e_i} \hat{e}(g, g)^{k_i}). \tag{1}$$

$\mathbf{C}_{e_i}^1$ is sent to $\mathcal{C}$ and stored at $\mathcal{C}$ for location dependent task allocation.

(2) *Task Allocation*:

- When a service requestor $\mathcal{R}$ plans to outsource a sensing task to $\mathcal{C}$, it first determines the task content T. T describes the required data type (e.g., health data, traffic flow, air quality, and so on), sensing hours, and reward policy. It is noted that the task content T does not include the information of task location. $\mathcal{R}$ then chooses a random number $r \in \mathbb{Z}_p^\star$ as its temporary private key sk_r, generates $pk_r = g^r$ as the temporary public key, and publishes pk_r to the system. After that, the $\mathcal{R}$ sends the service request along with T to $\mathcal{C}$.
- After receiving T, $\mathcal{C}$ broadcasts $\{T, pk_r\}$ to all edge nodes.
- For each edge node, it calculates $pk_r^{\frac{1}{e_i}}$ and sends to $\mathcal{C}$. $\mathcal{C}$ then forwards $\{\hat{e}(pk_{e_i}^{k_i}, pk_r^{\frac{1}{e_i}}), L_{e_i} \hat{e}(g, g)^{k_i}, pk_{e_i}\}, \forall i \in [1, M]$ to $\mathcal{R}$.
- $\mathcal{R}$ reveals L_{e_i} with its private key. The correctness of the decryption is proved as

$$\frac{L_{e_i} \hat{e}(g, g)^{k_i s k_r}}{\hat{e}(pk_{e_i}^{k_i}, pk_r^{\frac{1}{e_i}})} = \frac{L_{e_i} \hat{e}(g, g)^{k_i r}}{\hat{e}(g^{e_i k_i}, g^{\frac{r}{e_i}})} = \frac{L_{e_i} \hat{e}(g, g)^{k_i r}}{\hat{e}(g, g)^{k_i r}} = L_{e_i}. \tag{2}$$

  Based on the L_{e_i}, $\mathcal{R}$ selects edge nodes whose coverage area is in the task area. A list $\{pk_{e_i}\}, \forall i \in Q$ is returned to $\mathcal{C}$. Q is the set of indices of the selected edge nodes.
- $\mathcal{C}$ generates a unique identity N for the task T and calculates $H_1(pk_{e_i}^c)$ as a session key for $\mathcal{E}_i$. Then $\mathcal{C}$ encrypts $(\mathsf{N}||T||pk_r)$ with the session key by computing $\mathbf{C}_{e_i}^2 = \mathsf{Enc}(H_1(pk_{e_i}^c), \mathsf{N}||T||pk_r)$. $\mathbf{C}_{e_i}^2$ is sent to $\mathcal{E}_i$.
- After receiving $\mathbf{C}_{e_i}^2$, $\mathcal{E}_i$ reveals $(\mathsf{N}||T||pk_r)$ by computing $\mathsf{Dec}(H_1(pk_c^{e_i}), \mathbf{C}_{e_i}^2)$. The correctness of the decryption is proved when the symmetric key $H_1(pk_c^{e_i}) = H_1(pk_{e_i}^c)$ is proved as follows,

$$H_1(pk_c^{e_i}) = H_1(g^{ce_i}) = H_1(pk_{e_i}^c). \tag{3}$$

- $\mathcal{E}_i$ broadcasts T to mobile workers within its coverage. When there exists task competition, $\mathcal{E}_i$ selects winners based on their reputations and reward expectations. Subsequently, $\mathcal{E}_i$ calculates $H_1(pk_{w_j}^{e_i})$ as a session key for $\mathcal{W}_j$, and encrypts $(\mathsf{N}||T||pk_r)$ with the session key by computing

$$\mathbf{C}_{w_j}^1 = \mathsf{Enc}(H_1(pk_{w_j}^{e_i}), \mathsf{N}||T||pk_r), \tag{4}$$

where $j \in U$ and U is the set of indices of the selected mobile workers. $\mathbf{C}_{w_j}^1$ is sent to $\mathcal{W}_j$.

(3) *Data Collection*:

- For each selected mobile worker $\mathcal{W}_{j\in U}$, it reveals $(\mathsf{N}||T||pk_r)$ by computing $\mathsf{Dec}(H_1(pk_{e_i}^{w_j}), \mathbf{C}_{w_j}^1)$. The correctness of the decryption is proved when the symmetric key $H_1(pk_{e_i}^{w_j}) = H_1(pk_{w_j}^{e_i})$ is proved as follows,

$$H_1(pk_{e_i}^{w_j}) = H_1(g^{w_j e_i}) = H_1(pk_{w_j}^{e_i}). \tag{5}$$

- Each $\mathcal{W}_{j\in U}$ collects data based on the requirements in T and generates the sensing report $P_{j\in U}$. To protect P_j, $\mathcal{W}_j$ calculates the following

$$\begin{cases} X_j = H_1(pk_r^{w_j}) \\ Y_j = H_1(pk_c^{w_j}) \\ Z_j = H_1(pk_{e_i}^{w_j}) \\ P_j' = P_j \oplus X_j \\ K_j = H_2(N||P_j) \\ V_j = \mathsf{Enc}(Y_j, P_j') \\ Q_j = \mathsf{Enc}(Z_j, K_j). \end{cases} \tag{6}$$

- To ensure the authentication, $\mathcal{W}_j$ generates the signature σ as follows,

$$\begin{cases} \sigma_j^1 = g^{-w_j} g^{Y_j} \\ \sigma_j^2 = \hat{e}(g^{Y_j}, pk_c) \\ \sigma_j = (\sigma_j^1, \sigma_j^2). \end{cases} \tag{7}$$

- $\mathcal{W}_j$ returns the message $\{\mathsf{N}, V_j, Q_j, \sigma_j\}$ to $\mathcal{E}_i$.

(4) *Data Deduplication*:

When the $\mathcal{E}_i$ receives messages from the corresponding workers, it first removes the duplicated data and then aggregates the signatures of the corresponding workers as follows.

- $\mathcal{E}_i$ recovers Z_j by computing $Z_j = H_1(pk_{w_j}^{e_i})$. After that, it reveals K_j by calculating $\mathsf{Dec}(Z_j, Q_j)$. For all $j \in U$, $\mathcal{E}_i$ can find the duplicated data by comparing K_j. Then $\mathcal{E}_i$ randomly selects $j^\star \in D$ where D is a set of indices of the duplicated report. For the duplicated data, only $K_{j^\star}$ is remained and the others are removed to save communication/storage costs.

- For all $j \in D$, $\mathcal{E}_i$ aggregates the signature σ_j as follows,

$$\begin{cases} \eta = \prod_{j \in D} pk_{w_j} \\ \sigma_D^1 = \prod_{j \in D} \sigma_j^1 \\ \sigma_D^2 = \prod_{j \in D} \sigma_j^2 \\ \sigma_D = (\sigma_D^1, \sigma_D^2). \end{cases} \tag{8}$$

- $\mathcal{E}_i$ returns $\{K_{j^\star}, V_{j^\star}, \sigma_D, \eta\}$ and $\{K_j, V_j, \sigma_j\}_{\forall j \notin D}$ to the cloud $\mathcal{C}$. In addition, $\mathcal{E}_i$ adds $(K_{j^\star}, \{pk_{w_j}\}_{\forall j \in D})$ to its record.

(5) *Data Verification*:

After receiving the messages from $\mathcal{E}_i$, $\mathcal{C}$ first checks the validity of the signature by verifying

$$\hat{e}(pk_{w_j}, g)^{sk_c} \hat{e}(\sigma_D^1, pk_c) \stackrel{?}{=} \sigma_D^2, \tag{9}$$

$$\hat{e}(\eta, g)^{sk_c} \hat{e}(\sigma_D^1, pk_c) \stackrel{?}{=} \sigma_D^2. \tag{10}$$

If Eq. (9) does not hold, the message $\{K_j, V_j, \sigma_j\}$ is discarded. otherwise, $\mathcal{C}$ calculates $Y_j = H_1(pk_{w_j}^c)$ and $P_j' = \mathsf{Dec}(Y_j, V_j)$.

If Eq. (10) does not hold, the message $\{K_{j^\star}, V_{j^\star}, \sigma_D\}$ is discarded. otherwise, $\mathcal{C}$ calculates $Y_{j^\star} = H_1(pk_{w_{j^\star}}^c)$ and $P_{j^\star}' = \mathsf{Dec}(Y_{j^\star}, V_{j^\star})$.

$\mathcal{C}$ returns $\{K_{j^\star}, P_{j^\star}', pk_{j^\star}\}$ and $\{K_j, P_j', pk_j\}_{\forall j \notin D}$ to the service requestor $\mathcal{R}$.

(6) *Data Reading*:

The service requestor $\mathcal{R}$ can read the collected data as follows.

$$X_j = H_1(pk_{w_j}^r); \tag{11}$$

$$P_j = P_j' \oplus X_j; \tag{12}$$

$$K_j \stackrel{?}{=} H_2(\mathsf{N} || P_j), \tag{13}$$

where $j \notin D$ or $j = j^\star$. If Eq. (13) holds, P_j is valid. Otherwise, P_j is discarded.

When P_j is valid, the service requestor $\mathcal{R}$ returns $\{K_j, pk_j\}$ and the rewards to $\mathcal{C}$. $\mathcal{C}$ further forwards $\{K_j, pk_j\}$ to the corresponding $\mathcal{E}_i$. If $j \notin D$, $\mathcal{E}_i$ identifies the contributor with the public key pk_j and distributes rewards accordingly. If $j = j^\star$, $\mathcal{E}_i$ searches its record with $K_{j^\star}$ and determines the list of public keys $\{pk_{w_j}\}_{\forall j \in D}$. With these public keys, the contributors are identified although their duplicated reports are removed. After that, $\mathcal{E}_i$ issues corresponding rewards to the contributors.

6 Security Discussion

6.1 Correctness Proof

The correctness of the signature verification is proved as follows.

For all $j \notin D$,

$$\begin{aligned}
&\hat{e}(pk_{w_j}, g)^{sk_c}\hat{e}(\sigma_j^1, pk_c)\\
=&\hat{e}(pk_{w_j}, g)^{c}\hat{e}(g^{-w_j}g^{Y_j}, g^{c})\\
=&\hat{e}(g^{Y_j}, g^{c})\\
=&\sigma_j^2.
\end{aligned} \tag{14}$$

For the aggregated signature,

$$\begin{aligned}
&\hat{e}(\eta, g)^{sk_c}\hat{e}(\sigma_D^1, pk_c)\\
=&\hat{e}(\prod_{j\in D} pk_{w_j}, g)^{sk_c}\hat{e}(\prod_{j\in D}\sigma_j^1, g^{c})\\
=&\hat{e}(\prod_{j\in D} pk_{w_j}, g)^{c}\hat{e}(\prod_{j\in D}\sigma_j^1, g^{c})\\
=&\hat{e}(\prod_{j\in D} g^{w_j}, g^{c})\hat{e}(\prod_{j\in D} g^{-w_j}g^{Y_j}, g^{c})\\
=&\hat{e}(\prod_{j\in D} g^{Y_j}, g^{c}) = \prod_{j\in D}\sigma_j^2\\
=&\sigma_D^2.
\end{aligned} \tag{15}$$

6.2 Security Discussion

- *Privacy preservation of task location*: In the proposed scheme, the entire task location is confidential from the cloud and edge nodes. Specifically, the cloud transfers the ciphertext of edge nodes' locations to the service requestor. The service requestor then selects edge nodes that are located in the task area for task allocation. Thus the task location is not released to the cloud and edge nodes in the whole process. In addition, since the locations of edge nodes are encrypted by proxy re-encryption Ateniese et al. (2006), which is proved to be confidential under Decisional Diffie-Hellman (DDH) hard problem, the cloud is unable to reveal the location of edge nodes through $\mathbf{C}_{e_i}^1$. Therefore, the cloud cannot obtain the complete task location unless colluding with all the selected edge nodes. Moreover, one selected node cannot reveal the entire task location without colluding with all the other selected edge nodes. Thus, the privacy preservation of task location is achieved.

- *Secure data deduplication*: In the data deduplication, the edge nodes can detect the identical reports by comparing K_j. Since K_j is the hash result of the report P_j, the edge nodes is unable to recover the P_j by analyzing K_j due to the one-way property of the hash function. Therefore, the edge nodes are able to identify the redundant reports without revealing the report content. Thus, the secure data deduplication is achieved.

Table 2 Computational Time

	Service requestor side	Cloud side	Edge side	Worker side
Initialization	-	-	T_p+T_e+T_m	-
Task allocation	T_e+T_m	T_p+T_e	(N+2)T_e	-
Data collection	-	-	-	N(6T_e+T_p+T_m)
Data deduplication	-	-	NT_e+3|D|T_m	-
Data verification	-	(N-|D|+1)(T_m+2T_p+T_e)	-	-
Data reading	(N-|D|+1)T_e	-	-	-

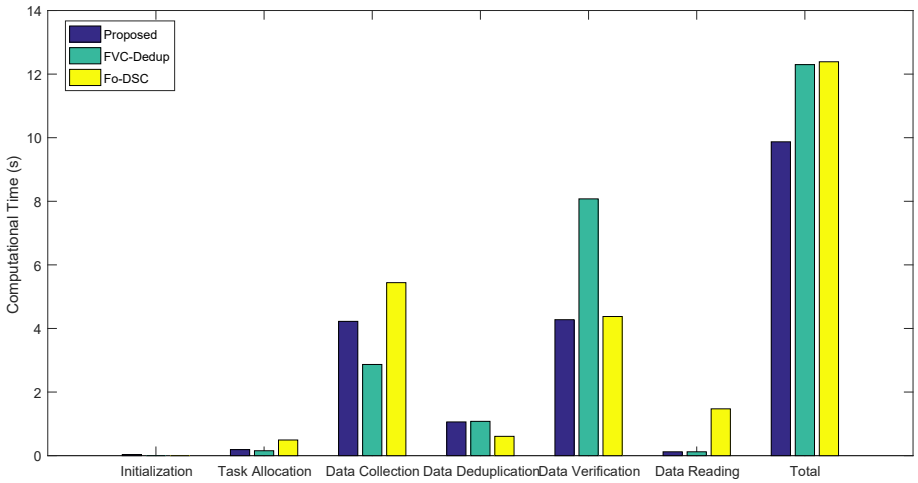
Notes: "T_e" : "operation time of one exponentiation"; "T_m" : "operation time of one multiplication"; "T_p" : "operation time of one pairing"; "N" : "the number of mobile workers"; "$|D|$" : "the number of duplicated reports".

- *Secure contributor identification*: Based on the description of the secure contributor identification in the section 3.3, we discuss the corresponding security in two parts. 1). In the proposed scheme, the edge nodes delete the duplicated reports and aggregate the signatures of the workers who generate the identical reports. The aggregated signatures can be verified by the cloud using the public key of the workers. Although the duplicated reports are removed, with the public keys, the corresponding contributors can be identified. 2). When a malicious worker captures the message $\{V_j, Q_j, \sigma_j\}$ from another worker and uploads to the edge node, the edge node first decrypts Q_j by computing $Z_j = H_1(pk_{w_j}^{e_i})$, where pk_{w_j} is the public key of the original worker. If the malicious worker claims that he/she generates the report, the edge node cannot compute the correct Z_j with the public key of the malicious worker. Without the correct Z_j, the correct K_j cannot be recovered. Then the report will not be classified as a redundant report. After that, in the stage of data verification, the cloud verifies the signature of the message. As shown in Eq. (14), the verification requires the public key of the original worker as well. Given the public key of the malicious worker, the verification fails and the report is then discarded. Thus, a malicious worker cannot pretend to be a contributor via uploading the captured message. In addition, the content of P_j is credential from the cloud and edge nodes in the data deduplication and data verification. Therefore, the secure contributor identification is achieved.

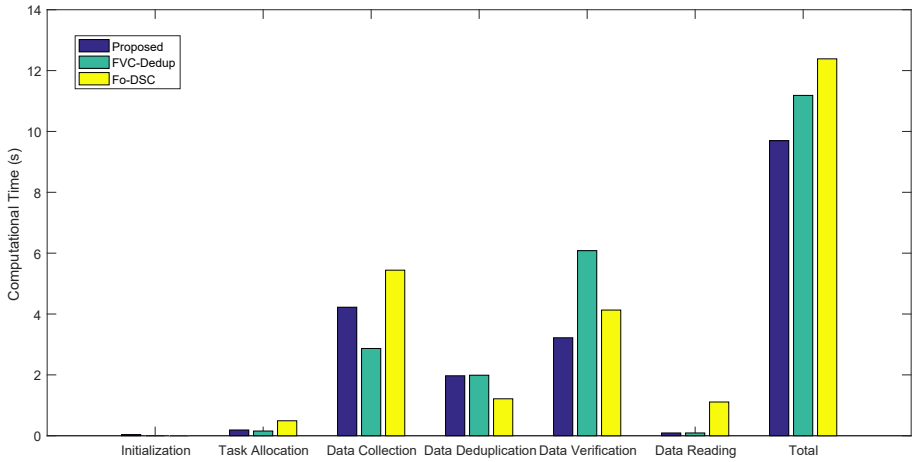
7 Experimental Results

In this section, we evaluate the computational efficiency of the proposed scheme by comparing with the scheme Fo-DSC of Ni et al. (2016) and the scheme FVC-Dedup of Jiang et al. (2021). We implement the proposed scheme on the experimental platform that is constructed on the Pairing-Based Cryptography Library (https://crypto.stanford.edu/pbc/) with the Linux system over an Intel(R) Core(TM) i7-4770 3.4GHz processor and 16GB memory. In this part, we first analyze the computational time consumption. We then show the computational efficiency in simulation experiments and discuss the comparison results. The numerical results in the following are based on the average values of 1000 simulation runs.

Table 2 shows the computational time when considering one cloud server and one edge node. In the results, T_e represents the operation time of one exponentiation. T_m represents the operation time of one multiplication. T_p represents the operation time of one pairing.



(a) N=100, |D|/N=20%.

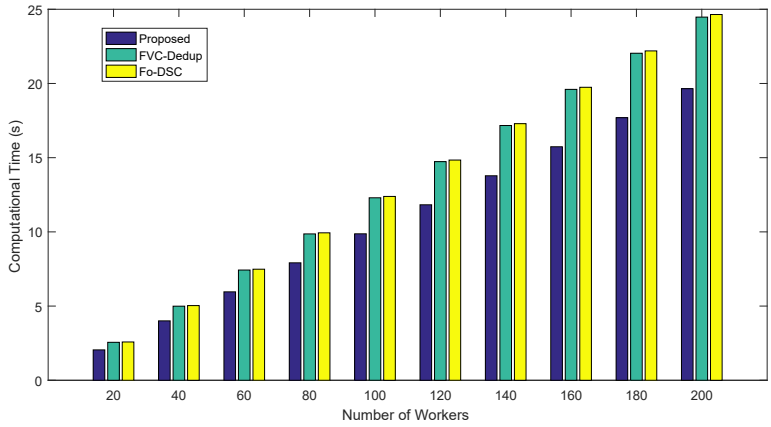


(b) N=100, |D|/N=40%.

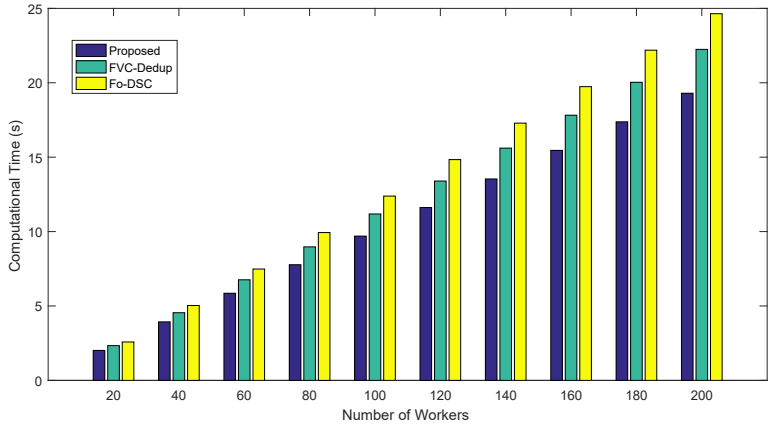
Figure 3: Computational time in different stages.

As shown in Table 2, considering the number of mobile workers is N and the number of duplicated reports is $|D|$, the computation time at different stages varies with N and $|D|$.

Fig. 3 shows the results of the computational time in different stages. Compared with the schemes of FVC-Dedup and Fo-DSC, even though our proposed scheme has higher computational time than FVC-Dedup in the stages of initialization and data collection, it has less computational time in the other stages. In addition, the total time cost of the system is significantly reduced. It indicates that our scheme achieves improved computation efficiency of the system with slight performance sacrifice in data collection.



(a) |Dl/N=20%.



(b) |Dl/N=40%.

Figure 4: Computational time in total.

Fig. 4 illustrates the performance in term of total computational time. It shows that the total computational time is proportional to the increase number of mobile workers. In the two cases of $|D|/N = 20\%$ and $|D|/N = 40\%$, our proposed scheme outperforms both FVC-Dedup and Fo-DSC. In addition, with the increase number of mobile workers, the advantages of our proposed scheme are more significant. It indicates that our proposed scheme achieves better performance in large-scale crowdsensing, being more suitable in practice.

8 Conclusion

In this work, we have proposed a privacy-preserving scheme for data deduplication in EMCS. The proposed scheme improves computational efficiency while preserving the privacy of task location. We have provided detailed discussion to show that the scheme is

correct and guarantees both secure data deduplication and secure contributor identification. We have analyzed the computational time of the proposed scheme from the view of theory. The experimental results have demonstrated that the proposed scheme has higher computational efficiency than the existing schemes.

Acknowledgement

This work was partially supported by National Science Foundation under grants CNS-2007995 and CNS-2008145.

References

S. Abdul Rahman, A. Mourad and M. El Barachi. (2019) 'An Infrastructure-Assisted Crowdsensing Approach for On-Demand Traffic Condition Estimation', *IEEE Access*, vol. 7, pp. 163323-163340, 2019.

G. Ateniese, K. Fu, M. Green, and S. Hohenberger. (2006) 'Improved proxy re-encryption schemes with applications to secure distributed storage', *ACM Transactions on Information and System Security*, vol. 9, no. 1, pp.1–30, Feb. 2006.

S. Basudan, X. Lin and K. Sankaranarayanan. (2017) 'A Privacy-Preserving Vehicular Crowdsensing-Based Road Surface Condition Monitoring System Using Fog Computing', *IEEE Internet of Things Journal*, vol. 4, no. 3, pp. 772-782, June 2017.

F. Bock, S. Di Martino and A. Origlia. (2020) 'Smart Parking: Using a Crowd of Taxis to Sense On-Street Parking Space Availability', *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 2, pp. 496-508, Feb. 2020.

H. Cui, X. Yuan, Y. Zheng and C. Wang. 2016 'Enabling secure and effective near-duplicate detection over encrypted in-network storage', *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, 2016, pp. 1-9.

S. Jiang, J. Liu, M. Duan, L. Wang and Y. Fang. (2018) 'Secure and Privacy-Preserving Report De-duplication in the Fog-Based Vehicular Crowdsensing System', *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1-6.

S. Jiang, J. Liu, Y. Zhou and Y. Fang. (2021) 'FVC-Dedup: A Secure Report Deduplication Scheme in a Fog-assisted Vehicular Crowdsensing System', *IEEE Transactions on Dependable and Secure Computing*, pp.1-14, 2021.

H. Lamaazi, R. Mizouni, S. Singh and H. Otrok. (2020) 'A Mobile Edge-Based CrowdSensing Framework for Heterogeneous IoT', *IEEE Access*, vol. 8, pp. 207524-207536, 2020.

J. Li, Z. Su, D. Guo, K. -K. R. Choo, Y. Ji and H. Pu. (2021) 'Secure Data Deduplication Protocol for Edge-Assisted Mobile CrowdSensing Services', *IEEE Transactions on Vehicular Technology*, vol. 70, no. 1, pp. 742-753, Jan. 2021.

J. Ni, X. Lin, K. Zhang and Y. Yu. (2016) 'Secure and Deduplicated Spatial Crowdsourcing: A Fog-Based Approach', *2016 IEEE Global Communications Conference (GLOBECOM)*, 2016, pp. 1-6.

J. Ni, A. Zhang, X. Lin and X. S. Shen. (2017) 'Security, Privacy, and Fairness in Fog-Based Vehicular Crowdsensing', *IEEE Communications Magazine*, vol. 55, no. 6, pp. 146-152, June 2017.

J. Ni, K. Zhang, Y. Yu, X. Lin and X. S. Shen. (2020) 'Providing Task Allocation and Secure Deduplication for Mobile Crowdsensing via Fog Computing', *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 3, pp. 581-594, 1 May 2020.

J. Ni, K. Zhang, Q. Xia, X. Lin and X. S. Shen, "Enabling Strong Privacy Preservation and Accurate Task Allocation for Mobile Crowdsensing," in IEEE Transactions on Mobile Computing, vol. 19, no. 6, pp. 1317-1331, 1 June 2020.

Y. Shen, L. Huang, L. Li, X. Lu, S. Wang and W. Yang. (2015) 'Towards Preserving Worker Location Privacy in Spatial Crowdsourcing', *2015 IEEE Global Communications Conference (GLOBECOM)*, 2015, pp. 1-6.

V. Sucasas, G. Mantas, J. Bastos, F. Damião and J. Rodriguez. (2020) 'A Signature Scheme with Unlinkable-yet-Accountable Pseudonymity for Privacy-Preserving Crowdsensing', *IEEE Transactions on Mobile Computing*, vol. 19, no. 4, pp. 752-768, 1 April 2020,

C. Wang, Z. Xie, L. Shao, Z. Zhang and M. Zhou. (2019) 'Estimating Travel Speed of a Road Section Through Sparse Crowdsensing Data', *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 9, pp. 3486-3495, Sept. 2019.

L. Wang, D. Yang, X. Han, D. Zhang and X. Ma. (2021) 'Mobile Crowdsourcing Task Allocation with Differential-and-Distortion Geo-Obfuscation', *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 2, pp. 967-981, 1 March 2021.

Z. Wang, J. Hu, R. Lv, J. Wei, Q. Wang, D. Yang and H. Qi. (2019) 'Personalized Privacy-Preserving Task Allocation for Mobile Crowdsensing', *IEEE Transactions on Mobile Computing*, vol. 18, no. 6, pp. 1330-1341, 1 June 2019.

Y. Zheng, X. Yuan, X. Wang, J. Jiang, C. Wang and X. Gui. (2017) 'Toward Encrypted Cloud Media Center With Secure Deduplication', *IEEE Transactions on Multimedia*, vol. 19, no. 2, pp. 251-265, Feb. 2017.

L. Zhou, D. Wu, X. Wei and Z. Dong. (2019) 'Seeing Isn't Believing: QoE Evaluation for Privacy-Aware Users', *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 7, pp. 1656-1665, July 2019.

P. Zhou, W. Chen, S. Ji, H. Jiang, L. Yu and D. Wu. (2019) 'Privacy-Preserving Online Task Allocation in Edge-Computing-Enabled Massive Crowdsensing', *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7773-7787, Oct. 2019.

The Pairing-Based Cryptography Library, https://crypto.stanford.edu/pbc/