FVC-Dedup: A Secure Report Deduplication Scheme in a Fog-assisted Vehicular Crowdsensing System

Shunrong Jiang*, Jianqing Liu⋆, Yong Zhou*, and Yuguang Fang§
*Department of Information Science, China University of Mining and Technology, Xuzhou 221116, China
⋆Department of Electrical and Computer Engineering, The University of Alabama in Huntsville, AL, USA
§Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, 32611, USA
Email:jsywow@gmail.com, jianqing.liu@uah.edu, yzhou@cumt.edu.cn, fang@ece.ufl.edu

***Abstract*—It is observed that modern vehicles are becoming more and more powerful in computing, communications, and storage capacity. By interacting with other vehicles or with local infrastructures (i.e., fog) such as road-side units, vehicles and fog devices can collaboratively provide services like crowdsensing in an efficient and secure way. Unfortunately, it is hard to develop a secure and privacy-preserving crowdsensing report deduplication mechanism in such a system. In this paper, we propose a scheme FVC-Dedup to address this challenge. Specifically, we develop cryptographic primitives to realize secure task allocation and guarantee the confidentiality of crowdsensing reports. During the report submission, we improve the message-lock encryption (MLE) scheme to realize privacy-preserving report deduplication and resist the fake duplicate attacks. Besides, we construct a novel signature scheme to achieve efficient signature aggregation and record the contributions of each participant fairly without knowing the crowdsensing data. The security analysis and performance evaluation demonstrate that FVC-Dedup can achieve secure and privacy-preserving report deduplication with moderate computing and communication overhead.**

I. Introduction

With the rapid advancement of Intelligent Transportation Systems (ITS), modern vehicles are equipped with wireless communication devices and onboard sensors [2, 3], such as GPS, cameras, and onboard units (OBUs), to enable vehicular crowdsensing [4, 5]. With the use of various sensing devices and OBUs, vehicles could report the driving information (e.g., real-time speed, location, and driving direction) periodically, as well as opportunistically provide road conditions, weather reports, and traffic situation for traffic signal control, transportation planning, etc. Furthermore, with the improvement on computing and data storage capabilities, fog computing [6–8] provides an emerging paradigm that significantly relaxes the limitations of the information interactions between the physical and cyber worlds. These attractive features lead to the booming development of fog-assisted vehicular crowdsensing systems (FVCS).

FVCS could be utilized in most of the mobile crowdsensing applications such as environmental monitoring (e.g., air quality measurement and noise level measurement), recreational applications (e.g., travel assistance and parking recommendation), and societal deployments (e.g., urban lifestyle and mobility) [9–14]. The reason is that FVCS integrates fog computing, in addition to inheriting the advantages of mobile crowdsensing, with special features such as communication efficiency, location awareness, etc. With FVCS, naturally distributed vehicles could be organized spontaneously to communicate and cooperate with one another to execute sensing tasks through fog nodes situated at the edge of the Internet and significantly boost the performance of ITS without excessive investments. Unfortunately, security and privacy pose serious design challenges because onboard sensors gather real-time data from the environments in the surroundings, which may contain privacy sensitive information [15, 16]. As a result, how to design privacy-preserving vehicular crowdsensing schemes has become an important yet challenging design issue.

Moreover, FVCS is geo-distributed and location-aware. There are unavoidably some duplicated reports because of vehicles' overlap sensing in FVCS [17]. Particularly, in scenarios such as traffic congestion monitoring and air quality monitoring, vehicles that are in close proximity (e.g., within a few hundred meters) acquire almost the same crowdsensing data and submit almost identical reports, resulting in significant redundant traffic. As such, it is always a good idea to lower the redundant traffic by deduplicating identical reports (the so-called *deduplication*) while retaining the proof of work in crowdsourcing in order to reward the contributions of these participants [18, 19]. However, we should not sacrifice the security and privacy for achieving crowdsensing report deduplication.

To achieve security in crowdsensing report deduplication, MLE [20] may be one of the solutions. In MLE, the same plaintext is always mapped to the same tag, and the plaintext is encrypted by a randomly chosen key. MLE is unfortunately vulnerable to off-line brute-force attacks [21], and adversaries are able to acquire the crowdsensing data through inferring possible plaintexts from the encrypted crowdsensing reports. Thus, we should improve MLE to guarantee the data security and privacy of the reports. Moreover, if it is possible to detect the equality of crowdsensing reports in public, attackers could falsify a duplicated report to get the reward without performing the task or compromise the proposed scheme in the aggregated signature verification phase [1]. To prevent such a fake duplicate attack, we need to achieve privacy-preserving report deduplication in FVCS. Furthermore, after secure and privacy-preserving report deduplication (without

A preliminary version was presented at IEEE GLOBECOM, 2018[1].

exposing side-channel identifiers of vehicles to the fog nodes), only one copy of the duplicated reports is returned to the crowdsensing server. Attackers may claim that they are part of the contributors by replaying the crowdsensing reports generated by other honest participants, or retrieve rewards more than once [19]. Thus, how to identify the real contributors for the duplicated reports is worthwhile to be investigated. In summary, it is vital to realize secure and privacy-preserving report deduplication, as well as to record the contributions of vehicles fairly without leaking the crowdsensing data.

Motivated by the above observations, in this paper, we propose an efficient and privacy-preserving report deduplication scheme, namely, FVC-Dedup, for FVCS. Our main contributions for FVC-Dedup are summarized as follows.

- We improve the MLE scheme to realize privacy-preserving crowdsensing report deduplication that prevents the fake duplicate attack. Specifically, we hide the tag part of the ciphertext in the process of report submission to guarantee that a fog node can only check if the crowdsensing reports are identical without learning other information.
- To record the contributions of the participants whose reports are deduplicated, we improve the identity-based batch multi-signature scheme (IBMS) [22, 23] to support anonymous signature aggregation. Therefore, fog nodes can achieve efficient and anonymous signature aggregation. Meanwhile, the crowdsensing server can ensure secure aggregation verification and record the contributions of each participant.
- To recover the real contributors while detecting dishonest participants, we construct an efficient method based on cryptographic primitives to ensure that each contributor can get the corresponding reward only once.

The remainder of the paper is organized as follows. Section II summarizes the related works. The system model and design goals are illustrated in Section III. Section IV describes our FVC-Dedup scheme. Section V and Section VI demonstrate the security analysis and the performance evaluation of FVC-Dedup, respectively. Finally, our paper is concluded in Section VII.

II. Related Work

In this section, we first review some secure and privacy-preserving schemes for crowdsensing report deduplication. Particularly, we look into studies on MLE which is a crucial technique used in data deduplication.

Crowdsensing report deduplication: To realize precise task allocation together with secure crowdsensing reports deduplication, Ni et al. [4] introduced a fog-based crowdsensing architecture. Considering real-time navigation applications in fog-based VANETs (vehicular ad hoc networks), Wang et al. developed a secure and privacy-preserving scheme based on cryptographic primitives [9]. In the scenario of road surface condition monitoring, Basudan et al. [24] studied the security and privacy on data transmissions. Moreover, they offered a new solution by leveraging a constructed certificateless aggregate signcryption algorithm. Ni et al. [18] investigated the requirements to achieve security and fairness in fog-based vehicular crowdsensing, and designed their scheme based on MLE. However, the fake duplicate attacks and fair reward distribution were not considered in their work. To deal with the drawbacks of [18], an improved version, named as Fo-DSC, was designed in [19] to protect mobile users from privacy leakage in privacy-sensitive applications by using the BLS-oblivious pseudo-random function. However, their design incurred huge computation overhead at the fog node because of its complex cryptographic primitives. In our preliminary work [1], we presented a secure fog-based deduplication scheme to simultaneously realize fog-based task allocation and report deduplication for vehicular crowdsensing systems without considering the fairness in the reward process. Basudan et al. [25] proposed an efficient deduplicated reporting scheme in fog-assisted vehicular crowdsensing based on a certificateless aggregate signcryption scheme. Besides, the fairness was guaranteed among the vehicles whose reports were deduplicated. The deduplication operation was time-consuming due to pairing operations. To reduce the overhead due to outsourcing wireless sensing data to the cloud, Zhang et al. [26] presented two variable-sized block-level deduplication schemes based on Rabin fingerprinting with deterministic tags and random tags, respectively. Sharma et al. [27] proposed a four-layer architecture for fog-assisted cluster-based industrial IoT to address task allocation and secure data deduplication. Particularly, they adopted SHA-3 to generate the hash values for the verification of data deduplication at a fog node. We summarize the existing crowdsensing report deduplication schemes and their features in TABLE I.

Message-Lock Encryption (MLE): To prevent the brute-force attacks in convergent encryption (CE) [28], Bellare et al. [20] proposed the MLE concept and designed a randomized convergent encryption scheme as an implementation of MLE. Keelveedhi et al. [21] introduced DupLESS to encrypt data with message-based keys acquired from a key-server based on an MLE scheme and an oblivious PRF protocol (i.e., RSA-OPRF) [29]. In contrast to the traditional deterministic CE, DupLESS worked better against possible brute-force attacks. Xu et al. [30] improved CE to design a leakage-resilient and cross-user client-side deduplication scheme by Proof-of-Ownership (PoW) for encrypted files in the cloud storage. To avoid inferring ciphertext components from messages, Abadi et al. [31] designed two schemes that satisfy new security notions for MLE with lock-dependent messages. Unfortunately, it was time-consuming to verify the equality for random tags. To deal with this disadvantage, Jiang et al. [32] proposed an interactive scheme by leveraging a random decision tree, in which the dynamic operations of the decision tree were also considered. Bellare et al. [33] proposed an interactive MLE scheme, which can detect correlated and parameter-dependent messages for secure deduplication during the interactions (uploads and downloads) between a client and a server. Cui et al. [34] studied the near-duplicate detection for encrypted in-network storage. Besides, they presented a secure and effective system by bridging locality sensitive hashing, multi-key searchable encryption, and Yao's garbled circuits.

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.2021.3069944, IEEE Transactions on Dependable and Secure Computing

TABLE I
COMPARISON OF SECURE CROWDSENSING REPORT DEDUPLICATION SCHEMES.

Scheme	Application	Technology	Privacy-aware deduplication[1]	Fairness
Wang et al. [9]	Navigation	MLE+Group Signature	×	×
Basudan et al. [24]	Road monitoring	Signcryption	×	×
Ni et al. [18]	Crowdsensing	MLE	×	×
Ni et al. [19]	Crowdsensing	BLS-oblivious	✓	✓
Jiang et al. [1]	Crowdsensing	MLE	✓	×
Basudan et al. [25]	Crowdsensing	Signcryption	×	✓
Zhang et al. [26]	Wireless sensing	Rabin fingerprinting	×	×
FVC-Dedup	Crowdsensing	MLE+IBMS	✓	✓

[1] Privacy-aware deduplication means realizing deduplication against the fake duplicate attacks without compromising the privacy of reporters.

III. System Model and Preliminaries

A. System Model

The system model of FVC-Dedup contains four entities: customers $\mathcal{C}_i$, the service cloud $\mathcal{SC}$, fog nodes $\mathcal{F}_i$ at the road side, and vehicles $\mathcal{V}_i$, as described in Fig. 1.

- Customers $\mathcal{C}_i$ could be organizations or individuals, which generate crowdsensing tasks according to different locations (or areas of interest) and submit their tasks to $\mathcal{SC}$.
- The service cloud $\mathcal{SC}$ offers crowdsensing services to $\mathcal{C}_i$. It allocates tasks to $\mathcal{F}_i$ based on the location information, followed by collecting, verifying, and processing of the crowdsensing reports from $\mathcal{V}_i$ through $\mathcal{F}_i$. Finally, it returns the crowdsensing results to $\mathcal{C}_i$. Moreover, $\mathcal{SC}$ provides rewards to vehicles based on their contributions.
- Fog nodes $\mathcal{F}_i$ are distributed at the edge of the network, e.g., co-located with or installed on the road-side units (RSUs) along the road. Served as intermediaries between $\mathcal{V}_i$ and $\mathcal{SC}$, and they connect with $\mathcal{V}_i$ and $\mathcal{SC}$ using wireless and wired links, respectively. Moreover, they could acquire information about the mobile vehicles in their one-hop communication range [18] and are in charge of distributing tasks to $\mathcal{V}_i$ according to task requirements, performing data deduplication and signature aggregation on crowdsensing reports, and forwarding the deduplicated reports to $\mathcal{SC}$.
- Mobile vehicles $\mathcal{V}_i$ perform tasks that are obtained from $\mathcal{F}_i$ using their own sensing-enabled devices. Besides, $\mathcal{V}_i$ could get rewards from $\mathcal{SC}$ based on their contributions.

As shown in Fig. 1, the whole system operates as follows: $\mathcal{C}_i$ produces a sensing task and submits it to $\mathcal{SC}$, along with the reward to be offered as in step 1. Upon receiving the sensing task, $\mathcal{SC}$ carries out fog-assisted task allocation and assigns it to $\mathcal{V}_i$. Specifically, $\mathcal{SC}$ assigns the sensing task to fog nodes $\mathcal{F}_i$ based on the locations and the coverage ranges of fog nodes as in step 2. Then, $\mathcal{F}_i$ further selects vehicles within their coverage ranges to complete the task according to the task requirements and their mobility patterns [35, 36] as in step 3. After that, the participating vehicles collect sensed data, generate crowdsensing reports, and submit them to $\mathcal{F}_i$ as in step 4. Upon receiving requested reports, $\mathcal{F}_i$ performs the report deduplication and aggregation operations, and then forwards the processed reports to $\mathcal{SC}$ as in step 5. After the verification operation is passed, $\mathcal{SC}$ generates the crowdsensing results for $\mathcal{C}_i$ as in step 6. $\mathcal{C}_i$ decrypts the crowdsensing results and determines the contribution of each result. Finally, $\mathcal{SC}$ credits rewards to the vehicles based on their contributions.

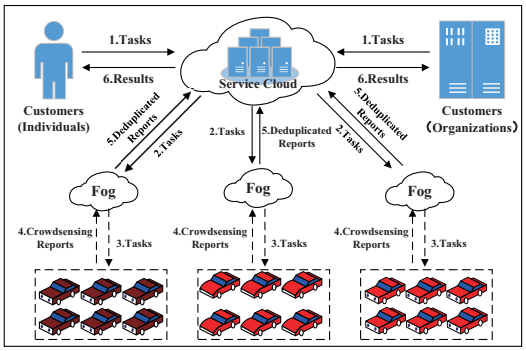


Fig. 1. The system model.

B. Threat Models

In the above system model, both internal and external attackers could bring in security threats to the whole system. For example, external attackers may attempt to get important information about tasks, report data, and other aspects by eavesdropping on wireless communication channels. They can also launch attacks such as replay attacks, forgery attacks, and impersonation attacks. Internal attackers may include $\mathcal{V}_i$, $\mathcal{F}_i$, and $\mathcal{SC}$, which are honest-but-curious[1]. This implies that they will follow the proposed protocols, but they are also curious about the crowdsensing data submitted by vehicles. Besides, $\mathcal{F}_i$ and $\mathcal{V}_i$ may be interested in the relationship between the identity and the report of each honest participating vehicle in the report deduplication phase. *In this paper, we do not consider the collusion attacks among $\mathcal{V}_i$, $\mathcal{F}_i$, and $\mathcal{SC}$.* Moreover, a selected $\mathcal{V}_i$ submits crowdsensing reports for rewards honestly, but may be lazy to conduct the crowdsensing tasks. This implies that they may launch a fake duplicate attack. Specifically, these vehicles may forge the crowdsensing data collected by honest participants to get rewards. Note that external attackers can also launch such attacks. Furthermore, during the reward retrieval, a greedy vehicle may offer more contribution proofs through submitting more crowdsensing

[1]Here, we assume $\mathcal{F}_i$ is honest-but-curious (or semi-trusted), which is consistent with the assumption in most related works. Although a compromised $\mathcal{F}_i$ can get the detailed task information, its impact on the whole system is quite limited. Besides, with the help of other security measures, the compromised $\mathcal{F}_i$ could be detected and fixed quickly.

reports than what is needed in order to obtain more rewards or retrieve rewards more than once.

C. Design Goals

To design secure and privacy-preserving report deduplication in our fog-assisted vehicular crowdsensing system and prevent security breaches, FVC-Dedup should set the design goals as follows:

- *Secure task allocation*: Only the legitimate vehicles[2] $\mathcal{V}_i$ selected by $\mathcal{F}_i$ can decipher the detail of task T.
- *Secure crowdsensing report deduplication*: To lower the communication overhead of crowdsensing report while not leaking the privacy of the report, $\mathcal{F}_i$ should delete the duplicated crowdsensing reports without learning the actual crowdsensing data. Furthermore, external/internal attackers should not be able to launch a fake duplicate attack to cheat on $\mathcal{C}_i$ without being detected.
- *Privacy-preserving crowdsensing report deduplication*: During the report deduplication process, the privacy leakage from the crowdsensing data should be prevented. Specifically, vehicle $\mathcal{V}_i$ cannot distinguish the differences (identity or report data) between two crowdsensing reports (except the one from itself), while fog node $\mathcal{F}_i$ cannot distinguish the differences in identities between two crowdsensing reports.
- *Secure contribution aggregation*: The participants should be rewarded based on their contributions. Hence, in the report deduplication operation, $\mathcal{F}_i$ could aggregate the signatures of the identical crowdsensing reports from different $\mathcal{V}_i$ to record their contributions. Besides, FVC-Dedup should also maintain the fairness for contributors by accomplishing the following goals [19]:
  - *Double-reporting detection*: Vehicle $\mathcal{V}_i$ is not able to submit more crowdsensing reports than what is needed without being detected.
  - *Double-retrieving detection*: Vehicle $\mathcal{V}_i$ is not able to redeem the same reward twice from $\mathcal{SC}$ without being detected.

Additionally, FVC-Dedup should meet other fundamental security goals, such as the integrity and authentication for crowdsensing reports.

D. Preliminaries

Definition III.1. *Computational Diffie-Hellman problem (CDH problem) [37]: Given (g, g^a, g^b) for a randomly-chosen generator $g \in \mathbb{G}_1$ and random $a, b \in \mathbb{Z}_q^*$ as well as a pairing $e(\cdot) : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$, compute the value g^{ab}.*

CDH hardness assumption: For an algorithm $\mathcal{A}$, we define the advantage in solving the CDH problem as

$$Adv_{\mathcal{A}} = \Pr[\mathcal{A}(g, g^a, g^b) = g^{ab}]. \quad (1)$$

[2]They are legal registered participants (passing the identity authentication) of our system and are not revoked in the certificate revocation list (CRL) by $\mathcal{SC}$ [9, 16].

TABLE II
NOTATIONS

Notations	Descriptions
$\mathcal{SC}$	The service cloud
$\mathcal{F}_i$	The ith fog node
$\mathcal{C}_i$	The ith customer
$\mathcal{V}_i$	The ith mobile vehicle
$SK_{\mathcal{F}_i}/PK_{\mathcal{F}_i}$	The secret/public key of $\mathcal{F}_i$
$SK_{\mathcal{V}_i}/PK_{\mathcal{V}_i}$	The secret/public key of $\mathcal{V}_i$
$H_1(\cdot)$	The hash function $H_1 : \{0,1\}^* \to \mathbb{G}_1$
$H_2(\cdot)$	The hash function $H_2 : \{0,1\}^* \to \mathbb{Z}_q^*$
$H_3(\cdot)$	The hash function $H_3 : \mathbb{G}_1 \to \{0,1\}^*$
$\mathsf{Enc}(\cdot)/\mathsf{Dec}(\cdot)$	The AES-128 encryption/decryption
$\mathcal{SIG}(\cdot)/\mathcal{VER}(\cdot)$	The ECDSA signature/verfication algorithm

The CDH hardness assumption states that $Adv_{\mathcal{A}}$ is a negligible probability[3] for any probabilistic polynomial-time algorithm $\mathcal{A}$ [38].

IV. FVC-Dedup Scheme

In this section, we will illustrate FVC-Dedup based on the following phases. Table II shows the notations employed in this paper.

A. Overview

According to the workflow of crowdsourcing in FVCS, we should ensure secure task allocation and privacy-preserving report deduplication with fair contribution aggregation. Specifically, to prevent leaking the task to attackers, we construct secure channels between different legitimate participants with secret keys. Thus, the task can be issued to the legitimate vehicles $\mathcal{V}_i$ through fog node $\mathcal{F}_i$. To resist the fake duplicate attack (especially launched by the lazy participants) during the report deduplication, we hide the tag part in MLE and ensure that only $\mathcal{C}_i$ can decrypt the crowdsensing data. In so doing, $\mathcal{F}_i$ can perform report deduplication and ensure the correctness of the aggregated signature for the duplicated reports. Moreover, to prevent leaking the relationship between the identity and the crowdsensing report of each participating vehicle to $\mathcal{F}_i$, we improve the signature algorithm [22, 23] by adding an obfuscation part that can also be used for verification during the reward retrieval by $\mathcal{SC}$. Thus, the proposed scheme can resist attacks launched by greedy participants and record the contributions fairly. Overall, secure and privacy-preserving report deduplication can be realized in FVCS.

Based on the above discussion, we design our improved IBMS, which is composed of the following algorithms [22, 23]:

- $(msk, mpk) \leftarrow \mathsf{Setup}(1^\lambda)$. This algorithm takes a security parameter λ as input to generate a master secret/public key msk/mpk.
- $(SK_{ID_i}) \leftarrow \mathsf{Extract}(msk, ID_i)$. This algorithm takes a participant[4] identity ID_i and the master secret key msk as input to generate the participant' private key SK_{ID_i}.

[3]$Adv_{\mathcal{A}} \leq 1/poly(\lambda)$, where $poly(\cdot)$ is a positive polynomial and λ is a secure parameter.

[4]Here, participant refers to vehicle in our context.

Experiment $\text{Exp}_{\mathcal{A}}[\textit{IBMS}, \lambda]$:
$(msk, mpk) \leftarrow \mathsf{Setup}(1^\lambda)$
For $i = 1, \cdots, q_E$
$SK_{ID_i} \leftarrow \mathcal{A}(mpk, ID_i)$
For $j = 1, \cdots, q_S$,
$\sigma_{i,m_j} \leftarrow \mathsf{Sign}(SK_{ID_i}, m_j)$
$(ID^*, \sigma^*) \leftarrow \mathcal{A}(mpk, ID_1, \sigma_{1,m_1}, \cdots, \sigma_{1,m_n}, \cdots, ID_{\mathcal{Q}}, \sigma_{\mathcal{Q},m_1}, \cdots, \sigma_{1,m_n})$
$1 \leftarrow \mathsf{Verify}(mpk, m, IS^*, \sigma^*)$, where $IS^* = IS \cup \{ID^*\}$ and $\sigma^* = \sigma_m \cup \{\sigma_{*,m}\}$
If $ID^* \notin IS$ and $\sigma_{*,m} \notin \sigma_m$, output "1" else "0".

Fig. 2: The experiment $\text{Exp}_{\mathcal{A}}[\textit{IBMS}, \lambda]$.

- $(\sigma_m) \leftarrow \mathsf{MSign}(\{SK_{ID_1}, \cdots, SK_{ID_{\mathcal{Q}}}\}, mpk, m)$. This algorithm is a multisignature protocol run by a group of participants $\{ID_1, \cdots, ID_{\mathcal{Q}}\}$ who intend to sign the same message m. The protocol can be divided into two phases: First, each participant ID_i executes $(\sigma_{i,m}) \leftarrow \mathsf{Sign}(SK_{ID_i}, m)$ which takes the same message m, the secret SK_{ID_i} as input and outputs an individual identity-based signature $\sigma_{i,m}$. Then, an aggregate algorithm $(\sigma_m) \leftarrow \mathsf{Agg}(\{\sigma_{1,m}, \cdots, \sigma_{\mathcal{Q},m}\}, mpk)$ is performed by a trusted party which outputs a multisignature σ_m for each participant.
- $(0, 1) \leftarrow \mathsf{Verify}(mpk, m, IS, \sigma_m)$. This algorithm is run by the verifier to verify whether the aggregate signature σ_m is a valid multisignature on message m on behalf of the set of identities $IS = \{ID_1, \cdots, ID_{\mathcal{Q}}\}$.

Definition IV.1. *Let* $\textit{IBMS} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{MSign}, \mathsf{Verify})$ *be a signature scheme. For any* $\lambda \in \mathbb{Z}_q^*$*, integer* $\mathcal{Q}$ *and running in time at most* t[5] *in the experiment* $\text{Exp}_{\mathcal{A}}[\textit{IBMS}, \lambda]$ *as shown in* Fig. 2*, we define the advantage* $Adv_{\mathcal{A}}(\textit{IBMS}, \lambda)$ *of an adversary* $\mathcal{A}$*, making at most* q_E *adaptive key extraction queries and* q_S *adaptive signature queries in the above experiment against* $\textit{IBMS}$ *as*

$$Adv_{\mathcal{A}}(\textit{IBMS}, \lambda) = \Pr[\text{Exp}_{\mathcal{A}}[\mathsf{IBMS}, \lambda] = 1]. \tag{2}$$

$\textit{IBMS}$ *scheme is* $(t, q_E, q_S, \mathcal{Q}, 1/poly(\lambda))$*-secure, if*

$$Adv_{\mathcal{A}}(\textit{IBMS}, \lambda) \leq 1/poly(\lambda), \tag{3}$$

where $1/poly(\cdot)$ *is a negligible function of its input.*

B. Our FVC-Dedup

1) System Initialization: With the public parameters ($\mathbb{G}_1$, $\mathbb{G}_T$, $e(\cdot)$, g, q, $H_1(\cdot)$, $H_2(\cdot)$, $H_3(\cdot)$), where $e(\cdot) : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$, in which $\mathbb{G}_1$ and $\mathbb{G}_T$ are bilinear groups of a prime order q. g is a random generator of $\mathbb{G}_1$. $\mathcal{SC}$ selects $s \in \mathbb{Z}_q^*$ at random as its master secret key $SK_{\mathcal{SC}}$ and calculates $PK_{\mathcal{SC}} = g^s$ as its master public key. Besides, for each registered vehicle $\mathcal{V}_i$ with identity $ID_{\mathcal{V}_i}$, $\mathcal{SC}$ calculates the signature secret key $SSK_{\mathcal{V}_i} = H_1(ID_{\mathcal{V}_i})^s$.

[5] t is an upper bound of running time with $1/poly(\lambda)$ for the experiment because the scheme can be broken in 2^λ times at most.

To generate the secure channel during the task allocation, each fog node $\mathcal{F}_i$ randomly selects $SK_{\mathcal{F}_i} = \theta_i \in \mathbb{Z}_q^*$ as the secret key and calculates $PK_{\mathcal{F}_i} = g^{\theta_i}$ as the public key. Besides, each registered vehicle $\mathcal{V}_i$ also randomly chooses $SK_{\mathcal{V}_i} = u_i \in \mathbb{Z}_q^*$ as the secret key and generates $PK_{\mathcal{V}_i} = g^{u_i}$ as the public key. The elliptic curve digital signature algorithm (ECDSA) [39] is used to guarantee the integrity and authentication during the task allocation.

2) Task Allocation: Once a customer $\mathcal{C}_i$ plans to launch a task T before the expiration time T_e based on the location loc, she/he randomly selects $c_i \in \mathbb{Z}_q^*$ and computes a temporary public key g^{c_i}. Then, she/he conceals T by calculating $\mathcal{T} = T \oplus H_3(PK_{\mathcal{SC}} || PK_{\mathcal{SC}}^{c_i})$. Finally, $\mathcal{C}_i$ sends the message as shown below to $\mathcal{SC}$.

$$\mathcal{C}_i \rightarrow \mathcal{SC} : loc, T_e, \mathcal{T}, g^{c_i}; \tag{4}$$

Upon receiving the task request, $\mathcal{SC}$ calculates $\mathcal{T} \oplus H_3(PK_{\mathcal{SC}} || (g^{c_i})^s)$ to recover T. $\mathcal{SC}$ sets $N \in \mathbb{Z}_q^*$ as a unique identifier of T and chooses several fog nodes $\mathcal{F}_i$ based on loc. After that, $\mathcal{SC}$ randomly selects $\phi_i \in \mathbb{Z}_q^*$ and computes $\widetilde{\mathcal{T}}_i = T \oplus H_3(PK_{\mathcal{F}_i} || PK_{\mathcal{F}_i}^{\phi_i})$. Finally, $\mathcal{SC}$ forwards $(N, T_e, \widetilde{\mathcal{T}}_i, g^{\phi_i}, g^{c_i})$ to each chosen $\mathcal{F}_i$.

When $\mathcal{V}_i$ wants to perform a crowdsensing task, $\mathcal{V}_i$ first submits its profile $VP_i = (TY_{\mathcal{V}_i}, AR_{\mathcal{V}_i}, TE_{\mathcal{V}_i})$ with ECDSA to the current fog node $\mathcal{F}_i$. Here, $TY_{\mathcal{V}_i}$ is a task type of interest, $AR_{\mathcal{V}_i}$ is the interested area, and $TE_{\mathcal{V}_i}$ is the validity period. Upon receiving these profiles, $\mathcal{F}_i$ first checks the CRL to ensure the legality of $\mathcal{V}_i$. If $\mathcal{V}_i$ is legitimate, it then verifies the signature. If all verifications pass, $\mathcal{F}_i$ records these profiles. Once $\mathcal{F}_i$ recovers $(N, T_e, T, g^{\phi_i}, g^{c_i})$ of the task T, $\mathcal{F}_i$ selects a set of $\mathcal{V}_i$ where $T \in TY_{\mathcal{V}_i}$, $T_e \in TE_{\mathcal{V}_i}$, and $loc \in AR_{\mathcal{V}_i}$ as the candidates to perform T [35, 36]. Besides, $\mathcal{F}_i$ randomly selects $f_i \in \mathbb{Z}_q^*$ and calculates g^{f_i}. Then, $\mathcal{F}_i$ computes $k_i = H_3(PK_{\mathcal{V}_i} || PK_{\mathcal{V}_i}^{f_i})$ and $K_i = \mathsf{Enc}(k_i, T_e || T || g^{c_i})$. Finally, $\mathcal{F}_i$ sends (N, g^{f_i}, K_i) with the signature to $\mathcal{V}_i$.

3) Data Collection: When $\mathcal{V}_i$ receives the task request message, $\mathcal{V}_i$ will execute the operations as follows:

- $\mathcal{V}_i$ calculates $k_i' = H_3(PK_{\mathcal{V}_i} || (g^{f_i})^{u_i})$ and $\mathsf{DEC}(k_i', K_i)$ to get the task T, T_e, and g^{c_i};
- $\mathcal{V}_i$ verifies the validity of the signature by ECDSA;
- After passing the verification, $\mathcal{V}_i$ begins to collect data based on T and gets the crowdsensing data P_i. In order to protect P_i, $\mathcal{V}_i$ selects $v_i \in \mathbb{Z}_q^*$ at random, and calculates

$$\begin{cases} J_i = g^{v_i} \\ \mathcal{X}_i = H_2(N || P_i) \\ \mathcal{Y}_i = H_2(N || \mathcal{X}_i) \\ L_i = H_3(g^{c_i} || (g^{c_i})^{v_i}) \\ \mathcal{Z}_i = \mathsf{Enc}(L_i, P_i) \\ t_i = H_3(g^{f_i} || (g^{f_i})^{v_i}) \\ \mathcal{P}_i = \mathcal{Y}_i \oplus t_i \end{cases} \tag{5}$$

- To ensure authenticity and integrity of the report, $\mathcal{V}_i$ randomly chooses $w_i, a_i \in \mathbb{Z}_q^*$, and calculates

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.2021.3069944, IEEE Transactions on Dependable and Secure Computing

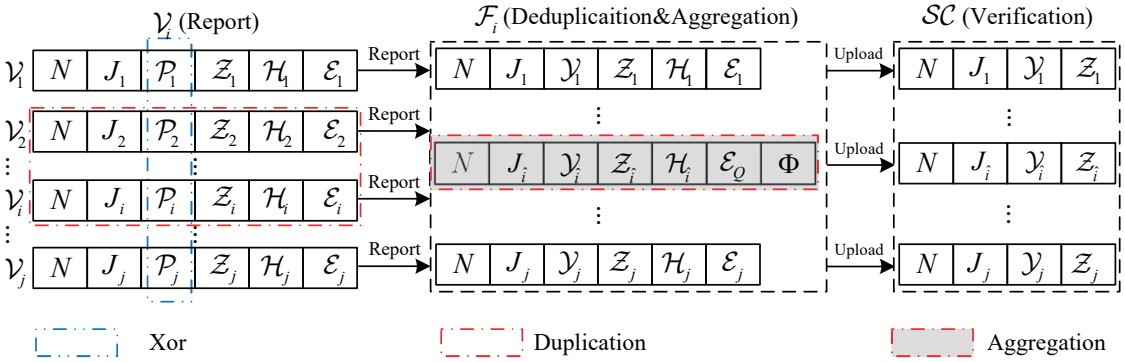


Fig. 3. The crowdsensing report deduplication process.

$$\begin{cases} \mathcal{A}_i = g^{w_i} \\ \mathcal{B}_i = g^{\mathcal{Y}_i} \\ \mathcal{D}_i = SSK_{\mathcal{V}_i}^{a_i} \mathcal{B}_i^{w_i} \\ \mathcal{E}_i = (\mathcal{A}_i, \mathcal{D}_i) \\ \mathcal{H}_i = H_1(ID_{\mathcal{V}_i})^{a_i} \end{cases} \tag{6}$$

- Finally, $\mathcal{V}_i$ returns the following message to $\mathcal{F}_i$ before T_e[6]:

$$\mathcal{V}_i \rightarrow \mathcal{F}_i : N, J_i, \mathcal{P}_i, \mathcal{Z}_i, \mathcal{H}_i, \mathcal{E}_i. \tag{7}$$

4) Report Deduplication: Once $\mathcal{F}_i$ gets the crowdsensing reports from various $\mathcal{V}_i$ (assume that there are n reports), it will perform a data deduplication operation and a signature aggregation operation as shown in Fig. 3:

- For each J_i, $\mathcal{F}_i$ calculates $t_i = H_3(g^{f_i} || J_i^{f_i})$ and $\mathcal{Y}_i = \mathcal{P}_i \oplus t_i$. After that, $\mathcal{F}_i$ detects the duplicated report according to $\mathcal{Y}_i$. Here, we assume Q as the set of duplicated reports.
- In order to record contributions of the vehicles that submit duplicated reports, $\mathcal{F}_i$ aggregates the corresponding signature as

$$\mathcal{E}_Q = (\prod_{i \in Q} \mathcal{A}_i, \prod_{i \in Q} \mathcal{D}_i), \quad \Phi = e(\prod_{i \in Q} \mathcal{H}_i, PK_{\mathcal{SC}}). \tag{8}$$

- Finally, $\mathcal{F}_i$ selects one copy $\{\mathcal{Y}_{\widehat{i}}, \mathcal{Z}_{\widehat{i}}\}$ $(\widehat{i} \in Q)$ at random from the duplicated report set Q and returns the following message to $\mathcal{SC}$:

$$\mathcal{F}_i \rightarrow \mathcal{SC} : N, \{J_j, \mathcal{Y}_j, \mathcal{Z}_j, \mathcal{H}_j, \mathcal{E}_j\}_{j \notin Q}, \{J_i, \mathcal{H}_i\}_{i \in Q}, \mathcal{Y}_{\widehat{i}}, \mathcal{Z}_{\widehat{i}}, \mathcal{E}_Q, \Phi. \tag{9}$$

Notice that $\{J_i, \mathcal{H}_i\}_{i \in Q}$ are used for the *Reward and Revocation* phase.

5) Report Verification: When $\mathcal{SC}$ receives the aggregated reports, it can verify the validity of signatures by checking if

$$e(\prod_{i \in Q} \mathcal{D}_i, g) \stackrel{?}{=} e(\prod_{i \in Q} \mathcal{A}_i, g^{\mathcal{Y}_{\widehat{i}}}) \cdot \Phi. \tag{10}$$

[6]If $\mathcal{V}_i$ moves out of the range of $\mathcal{F}_j$, it only needs to report the result to the nearest fog node (e.g., $\mathcal{F}_j$). Then, $\mathcal{F}_i$ could authorize $\mathcal{F}_j$ to execute the deduplication operation by sending f_i to $\mathcal{F}_j$.

The valid verification of the signature is described as follows:

$$\begin{aligned} e(\prod_{i \in Q} \mathcal{D}_i, g) =& e(\prod_{i \in Q} SSK_{\mathcal{V}_i}^{a_i} \mathcal{B}_i^{w_i}, g) \\ =& e(\prod_{i \in Q} H_1(ID_{\mathcal{V}_i})^{sa_i} g^{\mathcal{Y}_{\widehat{i}} w_i}, g) \\ =& e(\prod_{i \in Q} H_1(ID_{\mathcal{V}_i})^{sa_i}, g) \cdot e(\prod_{i \in Q} g^{\mathcal{Y}_{\widehat{i}} w_i}, g) \\ =& e(\prod_{i \in Q} g^{w_i}, g^{\mathcal{Y}_{\widehat{i}}}) \cdot e(\prod_{i \in Q} \mathcal{H}_i, g^s) \\ =& e(\prod_{i \in Q} \mathcal{A}_i, g^{\mathcal{Y}_{\widehat{i}}}) \cdot \Phi. \end{aligned} \tag{11}$$

For other signatures $(1 \leq j \leq n, j \notin Q)$, it can verify the validity of signatures by checking if

$$e(\mathcal{D}_j, g) \stackrel{?}{=} e(\mathcal{A}_j, g^{\mathcal{Y}_j}) \cdot e(\mathcal{H}_j, PK_{\mathcal{SC}}). \tag{12}$$

The valid verification of the signature is described as follows:

$$\begin{aligned} e(\mathcal{D}_j, g) =& e(SSK_{\mathcal{V}_j}^{a_j} \mathcal{B}_j^{w_j}, g) \\ =& e(H_1(ID_{\mathcal{V}_j})^{sa_j} g^{\mathcal{Y}_j w_j}, g) \\ =& e(H_1(ID_{\mathcal{V}_j})^{sa_j}, g) \cdot e(g^{\mathcal{Y}_j w_j}, g) \\ =& e(g^{w_j}, g^{\mathcal{Y}_j}) \cdot e(H_1(ID_{\mathcal{V}_j})^{a_j}, g^s) \\ =& e(\mathcal{A}_j, g^{\mathcal{Y}_j}) \cdot e(\mathcal{H}_j, PK_{\mathcal{SC}}). \end{aligned} \tag{13}$$

After the signature verification is passed, $\mathcal{SC}$ forwards the valid reports to $\mathcal{C}_i$ as follows:

$$\mathcal{SC} \rightarrow \mathcal{C}_i : \{J_j, \mathcal{Y}_j, \mathcal{Z}_j\}_{j \notin Q}, J_{\widehat{i}}, \mathcal{Y}_{\widehat{i}}, \mathcal{Z}_{\widehat{i}}. \tag{14}$$

6) Report Decryption: Once $\mathcal{C}_i$ receives the crowdsensing reports, she/he computes:

$$\begin{cases} L_i' = H_3(g^{c_i} || J_i^{c_i}) \\ P_i' = \mathsf{Dec}(L_i', \mathcal{Z}_i) \\ \mathcal{X}_i' = H_2(N || P_i') \\ \mathcal{Y}_i' = H_2(N || \mathcal{X}_i') \end{cases} \tag{15}$$

Then, she/he checks whether $\mathcal{Y}_i' \stackrel{?}{=} \mathcal{Y}_i$. If it does not hold, she/he drops the report. Otherwise, she/he will accept P_i'.

7) Reward and Revocation: To distribute the reward and revoke internal attackers during the report deduplication operation, the following operations are performed:

- In *Task Allocation*, $\mathcal{F}_i$ calculates $K_i = \mathsf{Enc}(k_i, T_e||T||g^{\phi_i}||g^{c_i})$ and sends (N, g^{f_i}, K_i) with the signature to $\mathcal{V}_i$.
- In *Data Collection*, $\mathcal{V}_i$ calculates $m_i = H_3(g^{\phi_i}||(g^{\phi_i})^{v_i})$ and $\mathcal{M}_i = \mathsf{Enc}(m_i, a_i)$. Finally, $\mathcal{V}_i$ sends the message: $\mathcal{V}_i \rightarrow \mathcal{F}_i : N, J_i, \mathcal{M}_i, \mathcal{P}_i, \mathcal{Z}_i, \mathcal{H}_i, \mathcal{E}_i$.
- In *Report Deduplication*, $\mathcal{F}_i$ returns the identifiers of selected $\mathcal{V}_i$ and $\mathcal{M}_i$ to $\mathcal{SC}$.
- In *Report Verification*, $\mathcal{SC}$ calculates $m_i = H_3(g^{\phi_i}||J_i^{\phi_i})$ and $a_i = \mathsf{Dec}(m_i, \mathcal{M}_i)$. Thus, $\mathcal{SC}$ can recover each contributor by checking whether $H_1(ID_{\mathcal{V}_i})^{a_i} \stackrel{?}{=} \mathcal{H}_i$ after passing report verification. Besides, it can also identify the internal attackers who fail the report verification.

Notice that, for the greedy vehicles (e.g., $\mathcal{V}_j$) in T, $\mathcal{SC}$ first delays offering distribute the reward to $\mathcal{V}_j$, ignores the report results of $\mathcal{V}_j$, and records $\mathcal{V}_j$ in the greedy list using a counter. When the counter at $\mathcal{V}_j$ reaches a certain value (e.g., 10), $\mathcal{SC}$ adds $\mathcal{V}_j$ into the CRL and broadcasts the updated CRL to each $\mathcal{F}_i$. According to our designed scheme, $\mathcal{F}_i$ selects a set of suitable $\mathcal{V}_i$ which are not in the CRL to perform tasks. Thus, we can reduce the impact caused by the greedy vehicles.

V. Security Analysis

We first present the security proof about our constructed signature scheme derived from [22, 23]. Then, we present the detailed security analysis according to different security requirements.

Theorem V.1. *If the CDH problem is $(t_\varepsilon, \varepsilon)$-hard*[7]*, our signature scheme IBMS is $(t', q_{H_1}, q_{H_2}, q_E, q_S, \mathcal{Q}, \varepsilon')$*[8]*-secure against existential forgery under an adaptive chosen-message and an adaptive chosen-ID attack, for any t' and ε' satisfying*

$$\begin{cases} \varepsilon' \geq \frac{(q_E+q_S+\mathcal{Q}-1)^2 e^2}{4}\varepsilon \\ t' \leq t_\varepsilon + (2q_{H_1} + q_{H_2} + 5\mathcal{Q}q_S)t_{E_{\mathbb{G}_1}}, \end{cases} \tag{16}$$

where $t_{E_{\mathbb{G}_1}}$ is the time to execute one exponentiation operation over $\mathbb{G}_1$.

Proof: Let $\mathcal{C}$ be a challenger, $\mathcal{A}$ be an adversary capable of breaking the proposed signature scheme under an adaptive chosen-message attack. Thus, we have $Adv_{\mathcal{A}}(\mathit{IBMS}) = \varepsilon'$. Assume that $\mathcal{C}$ is given an instance (g, g^α, g^β) of the CDH problem in $\mathbb{G}_1$. We show how $\mathcal{C}$ can use $\mathcal{A}$ to solve the CDH problem.

Initialization: $\mathcal{C}$ sets $g_1 = g^\alpha$, chooses the system parameter $para = (\mathbb{G}_1, \mathbb{G}_T, e(\cdot), g, g_1, H_1, H_2)$, and gives $para$ to $\mathcal{A}$. Besides, we set H_1 and H_2 as random oracles controlled by $\mathcal{C}$.

Training: At any time, $\mathcal{A}$ can query H_1, H_2, **Extract**, and **Sign**. $\mathcal{C}$ answers $\mathcal{A}$'s queries as follows:

[7]This means that no t_ε-time algorithm has advantage ε in solving the CDH hardness problem.

[8]In the security model, the adversary can make q_{H_1} hash queries on H_1 hash function, q_{H_2} hash queries on H_2 hash function, q_E adaptive key extraction queries, and q_S adaptive signature queries, respectively.

H_1-queries: To respond to H_1-queries, $\mathcal{C}$ maintains an initially empty list H_1^{list}. When $\mathcal{A}$ queries H_1 with $ID_{\mathcal{V}_i}$, $\mathcal{C}$ executes:

- If there is a tuple $(ID_{\mathcal{V}_i}, \mu_i, id_{\mathcal{V}_i}, SSK_{\mathcal{V}_i}, H_1coin_i)$ on H_1^{list}, $\mathcal{C}$ responds with $id_{\mathcal{V}_i}$ as the answer;
- Otherwise, $\mathcal{C}$ picks a coin $H_1coin_i \in \{0,1\}$ with $\Pr[H_1coin_i = 1] = \delta$, selects $\mu_i \in \mathbb{Z}_q^*$ and proceeds as follows:
  - If $H_1coin_i = 0$, set $id_{\mathcal{V}_i} = g^{\mu_i}$, $SSK_{\mathcal{V}_i} = g_1^{\mu_i}$, add $(ID_{\mathcal{V}_i}, \mu_i, id_{\mathcal{V}_i}, SSK_{\mathcal{V}_i}, H_1coin_i)$ into H_1^{list}, and respond with $id_{\mathcal{V}_i}$ as the answer;
  - Else set $id_{\mathcal{V}_i} = g^{\beta\mu_i}$, $SSK_{\mathcal{V}_i} = NULL$, add $(ID_{\mathcal{V}_i}, \mu_i, id_{\mathcal{V}_i}, SSK_{\mathcal{V}_i}, H_1coin_i)$ into H_1^{list}, and respond with $id_{\mathcal{V}_i}$ as the answer.

H_2-queries: To respond to H_2-queries, $\mathcal{C}$ maintains an initially empty list H_2^{list}. Upon input m_i, $\mathcal{C}$ proceeds as follows:

- If there is a tuple $(m_i, \nu_i, \mathcal{B}_i, H_2coin_i)$ on H_2^{list}, $\mathcal{C}$ returns $\mathcal{B}_i$ as the answer;
- Otherwise, $\mathcal{C}$ picks a coin $H_2coin_i \in \{0,1\}$ with $\Pr[H_2coin_i = 1] = \delta$, selects $\nu_i \in \mathbb{Z}_q^*$ and proceeds as follows:
  - If $H_2coin_i = 0$, set $H(m_i) = \nu_i$, $\mathcal{B}_i = g^{\nu_i}g^\alpha$, add $(m_i, \nu_i, \mathcal{B}_i, H_2coin_i)$ into H_2^{list}, and return B_i as the answer;
  - Else set $H(m_i) = \nu_i$, $\mathcal{B}_i = g^{\nu_i}$, add $(m_i, \nu_i, \mathcal{B}_i, H_2coin_i)$ into H_2^{list}, and return $\mathcal{B}_i$ as the answer.

Extract queries: Upon input an identity $ID_{\mathcal{V}_i}$, $\mathcal{C}$ first makes an H_1-queries on $ID_{\mathcal{V}_i}$, then recovers $(ID_{\mathcal{V}_i}, \mu_i, id_{\mathcal{V}_i}, SSK_{\mathcal{V}_i}, H_1coin_i)$ from H_1^{list}. If $H_1coin_i = 0$, $\mathcal{C}$ returns $SSK_{\mathcal{V}_i} = g_1^{\mu_i}$ as the answer; otherwise, it aborts.

Sign queries: Upon input $(ID_{\mathcal{V}_i}, m_1, ..., m_n)$, $\mathcal{C}$ first makes an H_1-queries on $ID_{\mathcal{V}_i}$ and finds $(ID_{\mathcal{V}_i}, \mu_i, id_{\mathcal{V}_i}, SSK_{\mathcal{V}_i}, H_1coin_i)$ on H_1^{list}, and for $1 \leq j \leq n$, asks an H_2-queries on m_j and recovers $(m_j, \nu_j, \mathcal{B}_j, H_2coin_j)$ from H_2^{list}, then:

- If $H_1coin_i = 0$, apply the signature algorithm to generate a signature;
- Else if $H_2coin_j = 0$ for all $1 \leq j \leq n$, select $\nu_j, w_j, a_j \in \mathbb{Z}_q^*$, calculate $\mathcal{A}_j = g^{w_j}g^{-\beta\mu_i a_j}$ and $\mathcal{D}_{i,j} = \mathcal{A}_j^{\nu_j} g_1^{w_j}$, and output $(\mathcal{A}_j, \mathcal{D}_{i,j})$ for $1 \leq j \leq n$;
- Else abort.

Notice that all responses to **Sign** queries are valid. Specifically, $(\mathcal{A}_j, \mathcal{D}_{i,j})$ are valid signatures on $(m_1, ..., m_n)$ for $ID_{\mathcal{V}_i}$ because

$$\begin{aligned} & e(\mathcal{A}_i, \mathcal{B}_i) \cdot e(id_{\mathcal{V}_i}^{a_j}, g_1) \\ =& e(g^{w_j}g^{-\beta\mu_i a_j}, g^{\nu_j}g^\alpha) \cdot e(g^{\beta\mu_i a_j}, g_1) \\ =& e(g,g)^{(w_j-\beta\mu_i a_j)(\nu_j+\alpha)} \cdot e(g,g)^{\beta\mu_i\alpha} \\ =& e(g,g)^{\nu_j(w_j-\beta\mu_i a_j)+w_j\alpha} \\ =& e(g^{(w_j-\beta\mu_i a_j)^{\nu_j}}, g_1^{w_j}) \\ =& e(\mathcal{D}_{i,j}, g). \end{aligned} \tag{17}$$

Forgery: $\mathcal{A}$ outputs an identity set $L = \{ID_1^*, ..., ID_{\mathcal{Q}}^*\}$, a message m^*, and a multi-signature $\mathcal{E}_{\mathcal{Q}}^* =$

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.2021.3069944, IEEE Transactions on Dependable and Secure Computing

$(\prod_{i\in\mathcal{Q}}\mathcal{A}_i^*, \prod_{i\in\mathcal{Q}}\mathcal{D}_i^*)$ where $\mathcal{E}_{\mathcal{Q}}^*$ is a valid multi-signature on m^* under $\{ID_1^*, ..., ID_{\mathcal{Q}}^*\}$.

To acquire the solution of the CDH problem, $\mathcal{C}$ proceeds with the following steps:

- For $1 \le i \le \mathcal{Q}$, make H_1-queries on ID_i^* and recovers $(ID_{\mathcal{V}_i}^*, \mu_i^*, id_{\mathcal{V}_i}^*, SSK_{\mathcal{V}_i}^*, {H_1coin_i}^*)$ from H_1^{list};
- Make H_2-queries on m^*, and recover $(m_i^*, \nu_i^*, \mathcal{B}_i^*, {H_2coin_i}^*)$ from H_2^{list}.

For simplicity, we only consider the situation that ${H_1coin_1}^* = {H_2coin_j}^* = 1$ $(1 \le j \le \mathcal{Q})$ and ${H_1coin_i}^* = 0$ for $2 \le i \le \mathcal{Q}$; otherwise, $\mathcal{C}$ aborts. If $\mathcal{C}$ does not abort, this implies $id_1^* = g^{\beta\mu_1^*}$, $\mathcal{B}_j^* = g^{\nu_j^*}$, and $id_i^* = g^{\mu_i^*}$. Obviously, we have:

$$e(\prod_{i\in\mathcal{Q}}\mathcal{D}_i, g) = e(\prod_{i\in\mathcal{Q}}\mathcal{A}_i, \prod_{i\in\mathcal{Q}}\mathcal{B}_i)\cdot e(\prod_{i\in\mathcal{Q}}\mathcal{H}_i, g_1), \tag{18}$$

which implies

$$e(\prod_{i\in\mathcal{Q}}\mathcal{D}_i, g) = e(\prod_{i\in\mathcal{Q}}\mathcal{A}_i, g^{\sum_{i=1}^{\mathcal{Q}}\nu_i^*})\cdot e(g^{\beta\mu_1^* a_1 + \sum_{i=2}^{\mathcal{Q}} a_i\mu_i^*}, g_1). \tag{19}$$

Thus, we have

$$g^{\alpha\beta} = (\prod_{i\in\mathcal{Q}}\mathcal{D}_i \cdot \prod_{i\in\mathcal{Q}}\mathcal{A}_i^{-\sum_{i=1}^{\mathcal{Q}}\nu_i^*} \cdot g_1^{-\sum_{i=2}^{\mathcal{Q}} a_i\mu_i^*})^{-\mu_1^* a_1} \tag{20}$$

as the solution to the CDH problem.

It remains to calculate the probability that solves the given instance for the CDH problem. To do so, three events are required for $\mathcal{C}$ to succeed:

- Event ε_1: $\mathcal{C}$ does not abort due to any of $\mathcal{A}$'s queries;
- Event ε_2: $\mathcal{E}_{\mathcal{Q}}^*$ is a valid and nontrivial multi-signature on m^* under $\{ID_1^*, ..., ID_{\mathcal{Q}}^*\}$;
- Event ε_3: event ε_2 occurs, and ${H_1coin_1}^* = H_2coin_j^* = 1$ and ${H_1coin_i}^* = 0$ for $2 \le i \le \mathcal{Q}$.

$\mathcal{C}$ succeeds if all the above events occur. The probability $\Pr[\varepsilon_1 \cap \varepsilon_2 \cap \varepsilon_3]$ could be expressed as:

$$\Pr[\varepsilon_1 \cap \varepsilon_2 \cap \varepsilon_3] = \Pr[\varepsilon_1]\Pr[\varepsilon_2|\varepsilon_1]\Pr[\varepsilon_3|\varepsilon_2 \cap \varepsilon_1] \tag{21}$$

Claim 1: The probability that $\mathcal{C}$ does not abort due to $\mathcal{A}$'s **Extract** queries and **Sign** queries is more than $(1-\delta)^{q_E+q_S}$.

Claim 2: If $\mathcal{C}$ does not abort due to $\mathcal{A}$'s **Extract** queries and **Sign** queries, $\mathcal{A}$'s view is identical to its view in the real attack. Therefore, $\Pr[\varepsilon_2|\varepsilon_1] \ge Adv_{\mathcal{A}}(\mathit{IBMS}) \ge \varepsilon'$.

Claim 3: The probability that $\mathcal{C}$ does not abort after $\mathcal{A}$ outputs a nontrivial and valid forgery is at least $\delta^{\mathcal{Q}+1}(1-\delta)^{\mathcal{Q}-1}$. Hence $\Pr[\varepsilon_3|\varepsilon_2 \cap \varepsilon_1] \ge \delta^{\mathcal{Q}+1}(1-\delta)^{\mathcal{Q}-1}$.

Thus, $\mathcal{C}$ generates the correct answer with the following probability

$$\begin{aligned}\Pr[\varepsilon_1 \cap \varepsilon_2 \cap \varepsilon_3] &\ge \delta^{\mathcal{Q}+1}(1-\delta)^{q_E+q_S+\mathcal{Q}-1}\varepsilon' \\ &\ge \frac{2^{\mathcal{Q}+1}}{(q_E+q_S+\mathcal{Q}-1)^{\mathcal{Q}+1}e^2}\varepsilon' \ge \varepsilon.\end{aligned} \tag{22}$$

Therefore, we have $\varepsilon' \ge \frac{(q_E+q_S+\mathcal{Q}-1)^2 e^2}{4}\varepsilon$. The corresponding running time is at most $t_\varepsilon + (2q_{H_1} + q_{H_2} + 5\mathcal{Q}q_S)t_{E_{\mathbb{G}_1}}$. ■

A. Secure task allocation

During the task allocation process, the $\mathcal{SC}$ assigns task T to fog nodes $\mathcal{F}_i$ according to location information. When $\mathcal{F}_i$ receives T, it selects a set of $\mathcal{V}_i$ based on the requirements of T, and calculates $k_i = H_3(PK_{\mathcal{V}_i}||PK_{\mathcal{V}_i}^{f_i})$ and $K_i = \mathsf{Enc}(k_i, T_e||T||g^{c_i})$ with the ECDSA signature. Only the chosen $\mathcal{V}_i$ can compute $k_i' = H_3(PK_{\mathcal{V}_i}||(g^{f_i})^{u_i})$ and $\mathsf{DEC}(k_i', K_i)$ to obtain task T and g^{c_i}. Since it is a CDH problem for attackers to find $SK_{\mathcal{V}_i}$ from $PK_{\mathcal{V}_i}$, FVC-Dedup can achieve secure task allocation.

B. Secure crowdsensing report deduplication

During the crowdsensing report process, fog node $\mathcal{F}_i$ provides secure report deduplication without disclosing crowdsensing data P_i. The details are given below.

To achieve data confidentiality and report deduplication, MLE scheme is adopted to encrypt the crowdsensing data. With MLE scheme, fog node is able to decide if two reports are identical by comparing the tag parts of the ciphertexts, and retain only one copy of the duplicated reports to decrease the communication overhead. Nevertheless, external attackers or lazy mobile vehicles $\mathcal{V}_i$ may forge a duplicated report according to tag $\mathcal{Y}_i$ of the MLE [18]. Thus, the aggregated signature verification will fail in this case. In [18], $\mathcal{C}_i$ will drop the signature without recording the contributions of vehicles or adopt the recursive divide-and-conquer method to search and delete incorrect signatures, which greatly reduces the efficiency of aggregated signature verification. To overcome this drawback, each allocated vehicle calculates $t_i = H_3(g^{f_i}||(g^{f_i})^{v_i})$ and $\mathcal{P}_i = \mathcal{Y}_i \oplus t_i$ to hide the tag part of crowdsensing reports. Thus, only $\mathcal{F}_i$ can calculate $\mathcal{Y}_i$ and verify if some anonymous vehicles have submitted identical crowdsensing reports. To construct the correct t_i, attackers should get v_i from g^{v_i} which is a CDH problem [37]. Hence, attackers cannot forge a duplicated report in FVC-Dedup.

During the report submission phase, each vehicle $\mathcal{V}_i$ encrypts the crowdsensing reports P_i using MLE scheme by randomly choosing v_i and $L_i = H_3(g^{c_i}||(g^{c_i})^{v_i})$. Since MLE is demonstrated to be secure under the PRV-CDA model [20], in which the ciphertext of an unpredictable message is indistinguishable from a random string of the same length, our scheme could ensure the confidentiality of crowdsensing reports.

C. Privacy-preserving crowdsensing report deduplication

During the crowdsensing report deduplication process, the privacy leakage comes from the crowdsensing data and identity. To ensure the data confidentiality, we adopt AES-128 to encrypt the crowdsensing data while the corresponding key is generated by $L_i = H_3(g^{c_i}||(g^{c_i})^{v_i})$. To construct the correct key, attackers should get c_i from g^{c_i} (v_i from g^{v_i}) which is a CDH problem [37]. Thus, we can ensure data confidentiality. Besides, to achieve crowdsensing report deduplication, we use MLE to generate the tag part. However, in this way, it will leak the equality relationship of crowdsensing reports. To prevent this leakage, we calculate $t_i = H_3(g^{f_i}||J_i^{f_i})$ and $\mathcal{P}_i = \mathcal{Y}_i \oplus t_i$

TABLE III
THE COMPUTATION OVERHEAD OF FVC-DEDUP.

Phase	Entity	Computation Overhead	Communication Overhead (bits)
Task Allocation	$\mathcal{F}_i(\mathcal{F}_i \rightarrow \mathcal{V}_i)$	$(n+1)T_{exp} + nT_{aes}$	$n(S_{aes} + S_{\mathbb{G}_1})$
Data Collection	$\mathcal{V}_i(\mathcal{V}_i \rightarrow \mathcal{F}_i)$	$(8T_{exp} + 2T_{aes} + T_{mul})$	$(4S_{\mathbb{G}_1} + S_{aes} + S_{H_2})$
Report deduplication	$\mathcal{F}_i(\mathcal{F}_i \rightarrow \mathcal{SC})$	$nT_{exp} + 3(\lvert Q\rvert - 1)T_{mul} + T_{par}$	$(4n - 2\lvert Q\rvert + 3)S_{\mathbb{G}_1} + (n - \lvert Q\rvert)(S_{aes} + S_{H_2}) + S_{aes} + S_{\mathbb{G}_T}$
Report Verification	$\mathcal{SC}(\mathcal{SC} \rightarrow \mathcal{C}_i)$	$(n - \lvert Q\rvert + 1)(2T_{par} + T_{exp} + \widehat{T_{mul}})$	$(n - \lvert Q\rvert + 1)(S_{\mathbb{G}_1} + S_{aes} + S_{H_2})$
Report Decryption	$\mathcal{C}_i$	$(n - \lvert Q\rvert + 1)(T_{exp} + T_{aes})$	N/A
Reward and Revocation	$\mathcal{SC}(\mathcal{F}_i \rightarrow \mathcal{SC})$	$n(2T_{exp} + T_{aes})$	nS_{aes}

[1] n and Q is the number of chosen vehicles and the number of duplicated reports, respectively.

to hide the tag part of crowdsensing reports. Thus, only $\mathcal{F}_i$ can calculate $\mathcal{Y}_i$ and verify if some anonymous vehicles have submitted identical crowdsensing data. To construct the correct t_i, attackers should get v_i from g^{v_i} which is a CDH problem. Hence, we prevent the leakage of linking the identical crowdsensing reports to a specific vehicle. Furthermore, in *Data Collection* phase, $\mathcal{V}_i$ submits the crowdsensing report to $\mathcal{F}_i$ without exposing its identity. Specifically, we construct the IBMS scheme by adding a_i for each $\mathcal{D}_i = SSK_{\mathcal{V}_i}^{a_i}\mathcal{B}_i^{w_i}$ and $\mathcal{H}_i = H_1(ID_{\mathcal{V}_i})^{a_i}$, each one can verify the validity of these signatures without exposing vehicles' identities and only $\mathcal{SC}$ can recover such information. In short, even if $\mathcal{F}_i$ can learn that two crowdsensing reports are identical, it cannot link each report to the identity of its reported vehicle. Therefore, no attacker can link a crowdsensing report to a specific vehicle. With these techniques, we can achieve privacy-preserving crowdsensing report deduplication.

D. Secure contribution aggregation

To record the contributions of duplicated reports from $\mathcal{V}_i$, we construct the IBMS scheme from [23] to sign each message, and verify the aggregated signature at the $\mathcal{SC}$. However, attackers easily construct an incorrect signature according to the tag part $\mathcal{Y}_i$ of the ciphertexts, causing failed aggregated signature verification. To resist this attack, we calculate $\mathcal{P}_i = \mathcal{Y}_i \oplus t_i$ to hide the tag parts of ciphertexts. In this way, only $\mathcal{F}_i$ can find the correct $\mathcal{Y}_i$ and execute the signature aggregation operation. Moreover, the signature is proven to be secure under the CDH assumption. External attackers or the internal (lazy) vehicles cannot forge duplicated reports. Thus, our scheme can ensure secure contribution to a successful aggregation.

To protect the privacy during the report deduplication, we construct the IBMS scheme by adding a_i for each $\mathcal{D}_i = SSK_{\mathcal{V}_i}^{a_i}\mathcal{B}_i^{w_i}$ and $\mathcal{H}_i = H_1(ID_{\mathcal{V}_i})^{a_i}$. Thus, $\mathcal{V}_i$ submits its crowdsensing report to $\mathcal{F}_i$ without exposing its identity $ID_{\mathcal{V}_i}$ or $H_1(ID_{\mathcal{V}_i})$. However, to verify the validity of these signatures and identify each contributing vehicle, several important information including the identifies of selected $\mathcal{V}_i$, $\mathcal{M}_i = \mathsf{Enc}(m_i, a_i)$ and $\mathcal{H}_i$ are sent to $\mathcal{SC}$. Thus, only $\mathcal{SC}$ can obtain the identity of each vehicle and distribute the corresponding reward by checking whether $H_1(ID_{\mathcal{V}_i})^{a_i} \stackrel{?}{=} \mathcal{H}_i$ after passing report verification. Therefore, the behavior of double-reporting and double-retrieving of a greedy vehicle can be detected.

VI. PERFORMANCE ANALYSIS

A. Complexity Analysis

We quantify the overhead of our FVC-Dedup by the communication and computation overhead. Here, T_{mul}, T_{exp}, T_{par}, and $\widehat{T_{mul}}$ refer to the time to execute one point multiplication over $\mathbb{G}_1$, one exponentiation operation over $\mathbb{G}_1$, one pairing operation over $\mathbb{G}_T$, and one point multiplication over $\mathbb{G}_T$, respectively. We use T_{aes} to denote the running time to perform one AES-128 encryption/decryption operation over $\mathbb{Z}_q^*$. Notice that we omit the computation overhead of hash computation which is not time-consuming compared with other operations. For the communication overhead analysis, we use S_{aes}, S_{H_2}, S_{H_3}, $S_{\mathbb{G}_1}$ (S_{H_1}), and $S_{\mathbb{G}_T}$ to represent the size of the ciphertext encrypted by AES-128, the size of H_2, the size of H_3, the size of $\mathbb{G}_1$, and the size of $\mathbb{G}_T$, respectively. The detailed performance analysis is listed as follows.

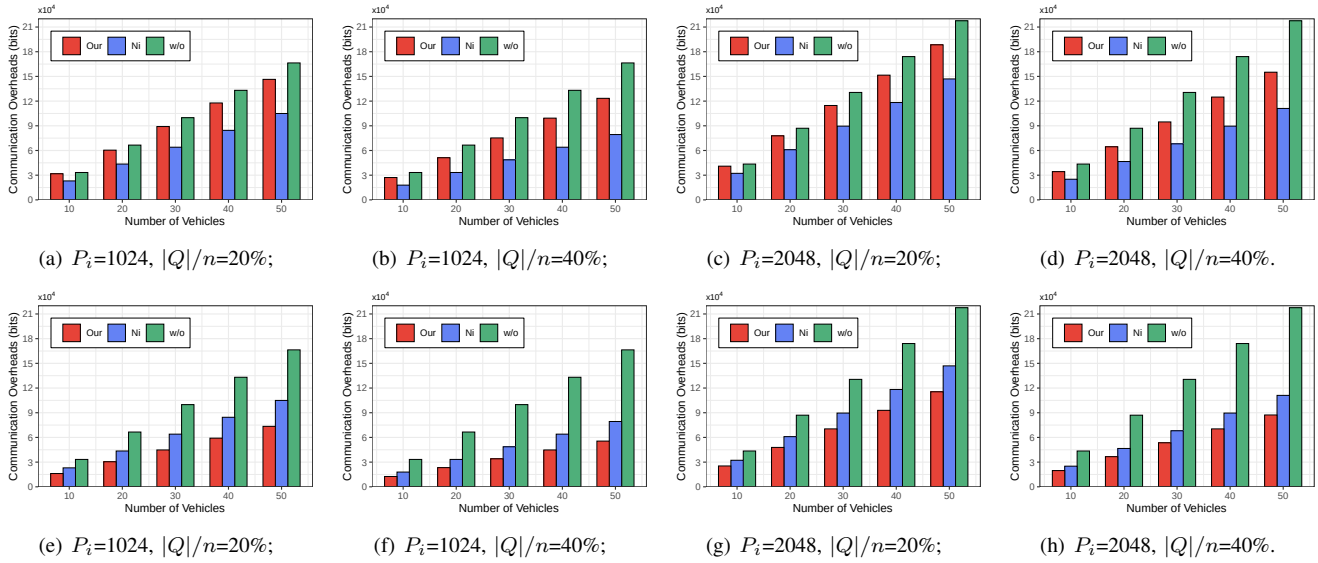
Task Allocation: In this phase, $\mathcal{F}_i$ computes $k_i = H_3(PK_{\mathcal{V}_i}||PK_{\mathcal{V}_i}^{f_i})$ and encrypts $K_i = \mathsf{Enc}(k_i, T_e||T||g^{c_i})$, which consumes $T_{exp}+T_{aes}$. After that, $\mathcal{F}_i$ sends (N, g^{f_i}, K_i) for each chosen $\mathcal{V}_i$, and the corresponding communication overhead is about $(S_{aes} + S_{\mathbb{G}_1})$. Thus, for n chosen vehicles, the corresponding computation overhead and communication overhead are $(n+1)T_{exp} + nT_{aes}$ and $n(S_{aes} + S_{\mathbb{G}_1})$, respectively.

Data Collection: During this phase, $\mathcal{V}_i$ first calculates $k_i' = H_3(PK_{\mathcal{V}_i}||(g^{f_i})^{u_i})$ and $\mathsf{DEC}(k_i', K_i)$ to get $T_e||T||g^{c_i}$, which takes $T_{exp}+T_{aes}$. After finishing the data collection, it should execute the data encryption and the signature operation, which altogether needs $(7T_{exp} + T_{aes} + T_{mul})$. Finally, $\mathcal{V}_i$ sends the report to $\mathcal{F}_i$, and the corresponding communication overhead is about $(4S_{\mathbb{G}_1} + S_{aes} + S_{H_2})$. Therefore, the total overhead is about $(8T_{exp} + 2T_{aes} + T_{mul})$ for computation overhead and $(4S_{\mathbb{G}_1} + S_{aes} + S_{H_2})$ for communication overhead.

Report deduplication: Here we assume that there are n reports sent to $\mathcal{F}_i$ and the number of duplicated reports is $|Q|$. When $\mathcal{F}_i$ receives these reports, it should calculate $t_i = H_3(g^{f_i}||J_i^{f_i})$ and $\mathcal{Y}_i = \mathcal{P}_i \oplus t_i$, which takes nT_{exp}. Then, it carries out the signature aggregation for the duplication data, which requires $3(|Q| - 1)T_{mul} + T_{par}$. Finally, $\mathcal{F}_i$ forwards these messages to $\mathcal{SC}$, and the corresponding communication overhead is about $(n - |Q|)(4S_{\mathbb{G}_1} + S_{aes} + S_{H_2}) + (2|Q| + 3)S_{\mathbb{G}_1} + S_{aes} + S_{\mathbb{G}_T}$. Hence, the total computation overhead and communication overhead is about $nT_{exp} + 3(|Q| - 1)T_{mul} + T_{par}$ and $(4n - 2|Q| + 3)S_{\mathbb{G}_1} + (n - |Q|)(S_{aes} + S_{H_2}) + S_{aes} + S_{\mathbb{G}_T}$, respectively.

Report Verification: Upon receiving the report message, $\mathcal{SC}$ executes the signature verification, which consumes $(n -$

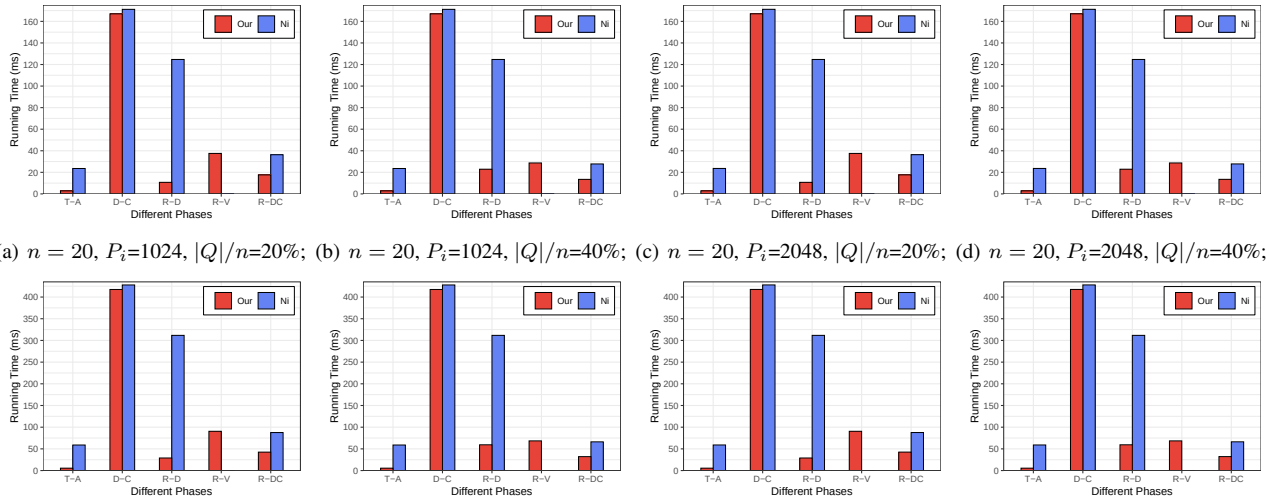
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2021.3069944, IEEE Transactions on Dependable and Secure Computing



(a) P_i=1024, $|Q|/n$=20%; (b) P_i=1024, $|Q|/n$=40%; (c) P_i=2048, $|Q|/n$=20%; (d) P_i=2048, $|Q|/n$=40%.

(e) P_i=1024, $|Q|/n$=20%; (f) P_i=1024, $|Q|/n$=40%; (g) P_i=2048, $|Q|/n$=20%; (h) P_i=2048, $|Q|/n$=40%.

Fig. 4. The communication overhead in different phases ($\mathcal{F}_i \rightarrow \mathcal{SC}$ (a)-(d) and $\mathcal{SC} \rightarrow \mathcal{C}_i$(e)-(h)).



(a) $n = 20$, P_i=1024, $|Q|/n$=20%; (b) $n = 20$, P_i=1024, $|Q|/n$=40%; (c) $n = 20$, P_i=2048, $|Q|/n$=20%; (d) $n = 20$, P_i=2048, $|Q|/n$=40%;

(e) $n = 50$, P_i=1024, $|Q|/n$=20%; (f) $n = 50$, P_i=1024, $|Q|/n$=40%. (g) $n = 50$, P_i=2048, $|Q|/n$=20%; (h) $n = 50$, P_i=2048, $|Q|/n$=40%.

Fig. 5. The computation overhead in different phases for different schemes.

$|Q|)(2T_{par} + T_{exp} + \widehat{T_{mul}})$ for no duplicated reports and $2T_{par} + T_{exp} + \widehat{T_{mul}}$ for the aggregated signature. The size of the message forwarding to $\mathcal{C}_i$ is about $(n - |Q| + 1)(S_{\mathbb{G}_1} + S_{aes} + S_{H_2})$. As a result, the total overhead is $(n-|Q|+1)(2T_{par}+T_{exp}+\widehat{T_{mul}})$ for computation overhead and $(n - |Q| + 1)(S_{\mathbb{G}_1} + S_{aes} + S_{H_2})$ for communication overhead.

Report Decryption: To get the report content, $\mathcal{C}_i$ should decrypt the received messages and hence the corresponding computation overhead is about $(n - |Q| + 1)(T_{exp} + T_{aes})$.

Reward and Revocation: To distribute the reward and revoke the internal attackers, $\mathcal{F}_i$ sends $\mathcal{M}_i$ to $\mathcal{SC}$, and hence the corresponding communication overhead is about nS_{aes}. Then, $\mathcal{SC}$ checks the validity, which takes about $n(2T_{exp} + T_{aes})$.

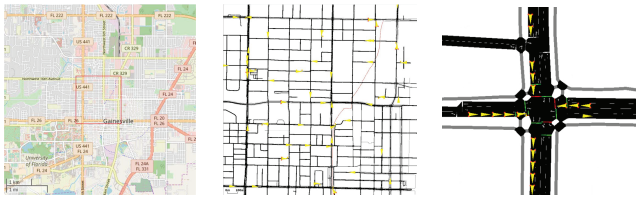
Finally, we summarize the corresponding overhead in Table III.

B. Simulation Evaluation

1) System Implementation: We implement FVC-Dedup using the PBC library [40] with Type A pairing parameters equivalent to 1024-bit Discrete Logarithm security. The sizes of q and $S_{\mathbb{G}_1}$ are 160 bits and 512 bits, respectively. Moreover, we assume that the size of each task T is 1280 bits and the length for crowdsensing data P_i is 1028-bit or 2048-bit. We implement the simulation on a computer with Intel(R) Core(TM) i9-8950HK CPU of 2.90 GHz and 8 GB memory with Ubuntu 18.04.

To display the deduplication operational efficiency, we first list the communication overhead between different entities ($\mathcal{SC} \rightarrow \mathcal{C}_i$ and $\mathcal{F}_i \rightarrow \mathcal{SC}$) and select two typical schemes: Ni's scheme (Ni's extended Fo-SDD scheme in [19])[9] and w/o scheme (our scheme without deduplication operation) as

[9]Notice that in our comparison study, we omit several complex operations in Ni' scheme [19] such as those for the zero-knowledge proof.



(a) Network of roads in Gainesville; (b) The mobile network imported into SUMO; (c) Vehicles in the mobile network.

Fig. 6. Mobility scenario produced by SUMO.

TABLE IV
SIMULATION PARAMETERS

Parameters	Values
Simulation area	2 km× 2 km
Speed of vehicles	30 km/h-40 km/h
Simulation duration	100 s
Wireless data rate	OfdmRate6Mbps
Wireless protocol	802.11p
Transmission power	25 dBm
Channel bandwidth	10 MHz
Propagation loss model	Nakagami

baselines for comparison. Fig. 4 shows the corresponding communication overhead under different lengths for crowdsensing data ($P_i = 1024$ and 2048) and different duplication rates ($|Q|/n = 20\%$ and 40%). As shown in Fig. 4(a), the message size from $\mathcal{F}_i$ to $\mathcal{SC}$ linearly increases as the number of reported vehicles increases for these three schemes, and Ni's scheme is the most efficient. The reason is that the fog node only needs to send the encrypted crowdsensing reports without signatures after the verification operations in their scheme. Interestingly, it is observed that in Fig. 4(b), the communication overhead reduces when the duplication rate of $|Q|/n$ is increased to 40%, compared to Fig. 4(a) (20%). Moreover, the size of the crowdsensing data P_i also has an impact on the communication overhead, as illustrated in Fig. 4(c) and Fig. 4(a).

Fig. 4(e)-Fig. 4(h) show the corresponding communication overhead for $\mathcal{SC} \rightarrow \mathcal{C}_i$. Obviously, the communication overhead of our FVC-Dedup is much lower than that of the other two schemes in the case when more reports are submitted to the fog node or when a larger sized crowdsensing data is submitted, which is consistent with the results shown in Fig. 4(a)-Fig. 4(d). The major reason is that we execute the signature verification at $\mathcal{SC}$. Moreover, comparing Fig. 4(f) with Fig. 4(e) (or comparing Fig. 4(h) with Fig. 4(g)), with the increasing duplication rate, the gap between FVC-Dedup and Ni's scheme is smaller.

Next, we study the computation overhead in different phases in comparison with that in Ni's scheme. The simulation results under different parameter settings are shown in Fig. 5. The simulation parameters include the number of reporting vehicles ($n = 20$ and 50), the duplication rate ($|Q|/n = 20\%$ and 40%), and the size of crowdsensing data ($P_i = 1024$ and 2048). Here T-A, D-C, R-D, R-V, R-DC refer to ***Task Allocation***, ***Data Collection***, ***Report deduplication***, ***Report Verification***, and ***Report Decryption***, respectively. Note that the total computation overhead of n vehicles is recorded in the D-C phase. In our study, we assume that the computation overhead is mainly attributed to the operations over $\mathbb{G}_1$ and $\mathbb{G}_T$, captured by T_{exp}, T_{mul}, and T_{par}, respectively.

Through comparison between Fig. 5(a) and Fig. 5(b) (or between Fig. 5(c) and Fig. 5(d)), we observe that our FVC-Dedup incurs less computation overhead, and the advantage of FVC-Dedup is more apparent with the increase in the duplication rate. The reason is that the total computation overhead for FVC-Dedup decreases with increasing $|Q|/n$, whereas the decrease of Ni's scheme only happens in the R-DC phase. Since the fog node should firstly verify each signature of each report in their scheme, in which the duplication rate only affects the R-DC phase. Obviously, the computation overhead of our scheme is significantly reduced at the fog node (in R-D phase). Besides, in FVC-Dedup, the computation overhead in R-D phase increases with increasing $|Q|/n$ while the corresponding overhead in R-V phase decreases.

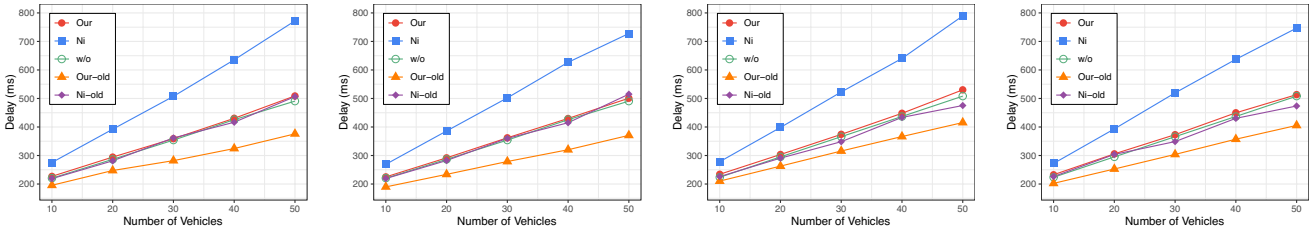
As described in Fig. 5(a) and Fig. 5(c), with the increasing size of crowdsensing data P_i, the computation overhead has limited growth for both schemes. This is because the additional computation overhead caused by the increased P_i is reflected in AES-128 and hash function operations, which has limited effect on the computation overhead. On the contrary, the duplication rate brings a more obvious effect on computation overhead as shown in Fig. 5(d). Moreover, the computation overhead in different phases almost increase linearly as the number of reporting vehicles increases, as shown in Fig. 5(a) and Fig. 5(f).

2) Networking Implementation: Finally, we evaluate the average delay in crowdsensing tasking with NS3 for FVC-Dedup. We define the average delay for a task as the average time interval between two events, namely, task generation and result acquisition. Since many factors, i.e., the performance of devices and the difficulty of tasks, will influence the specific sensing time, we omit the sensing time for convenience. Besides, OpenStreetMap is adopted to export the network of roads in Gainesville and the highlighted area ($2km \times 2km$) as displayed in Fig. 6(a) is selected. After that, the area is imported into SUMO as the mobile network shown in Fig. 6(b). Accordingly, Fig. 6(c) illustrates the deployment of vehicles over the mobile network.

Based on the system model, we set the wired link rate between $\mathcal{C}_i$ and $\mathcal{SC}$ to 10 Mbps and the wired link propagation delay as 50 ms. With respect to the wired communication between $\mathcal{SC}$ and $\mathcal{F}_i$, we set the corresponding bandwidth and link delay as 100 Mbps and 20 ms, respectively. Furthermore, the IEEE 802.11p protocol is utilized to establish physical links between $\mathcal{F}_i$ and $\mathcal{V}_i$. TABLE IV shows the corresponding parameters. To investigate the impact of the number of reporting vehicles on the delay, we vary n from 10 to 50 with an increment of 10 under different duplication rates ($|Q|/n = 20\%$ and 40%) and different sizes of crowdsensing data ($P_i = 1024$ and 2048). Besides, we compare FVC-Dedup scheme with Ni's scheme [19], w/o scheme, our old scheme [1], and Ni's old scheme [18].

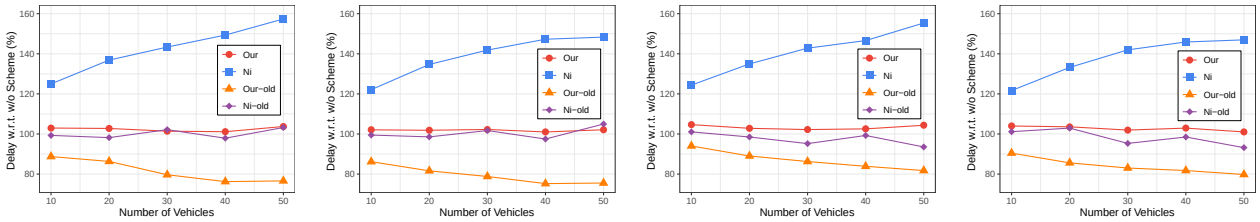
According to the implementation, the transmission delay

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.2021.3069944, IEEE Transactions on Dependable and Secure Computing



(a) $P_i = 1024$, $|Q|/n$=20%; (b) $P_i = 1024$, $|Q|/n$=40%. (c) $P_i = 2048$, $|Q|/n$=20%; (d) $P_i = 2048$, $|Q|/n$=40%.

Fig. 7. The average time delay of implementing crowdsensing tasks.



(a) $P_i = 1024$, $|Q|/n$=20%; (b) $P_i = 1024$, $|Q|/n$=40%. (c) $P_i = 2048$, $|Q|/n$=20%; (d) $P_i = 2048$, $|Q|/n$=40%.

Fig. 8. The corresponding delay w.r.t. w/o scheme.

is mainly caused by the wireless communication overhead between $\mathcal{F}_i$ and $\mathcal{V}_i$. Fig. 7 displays the average task delay with varied impact factors. As shown in Fig. 7(a), when $P_i = 1024$ and $n = 10$, the task delay is about 226.9 ms for FVC-Dedup. As a comparison, it takes 275.5 ms for Ni's scheme, 220.4 ms for w/o scheme, 195.6 ms for our old scheme, and 218.9 ms for Ni's old scheme. When $n = 50$, the corresponding average task delay is about 509.1 ms, 772.2 ms, and 490.7 ms, 375.9 ms, and 506.5 ms, respectively. Obviously, the average task delay exhibits an almost linear increase as more vehicles report crowdsensing results for these schemes. Despite the increment, FVC-Dedup displays higher efficiency in terms of time delay and higher increasing rate than Ni's scheme. Interestingly, the w/o scheme is more efficient than FVC-Dedup. This is because the deduplication operation needs additional computation overhead while the corresponding reduced communication overhead has limited effect on delay. However, with increasing duplication rate, the efficiency for FVC-Dedup is more obvious, even better than the w/o scheme as demonstrated in Fig. 7(b) and Fig. 7(d), respectively. Besides, comparing Fig. 7(b) with Fig. 7(a) (or comparing Fig. 7(d) with Fig. 7(c)), the average task delay decreases with the increasing duplication rate, even the impact is limited. It is reasonable that the reduced delay caused by the deduplication operations is limited because of the small-sized crowdsensing data and the large bandwidth between $\mathcal{SC}$ and $\mathcal{F}_i$. Moreover, comparing Fig. 7(c) with Fig. 7(a), with the increasing size of crowdsensing data, the average task delay has limited growth for all five schemes. That is because the additional computation and communication overhead caused by the increasing size of P_i has limited impacts on the delay. Finally, we also present the corresponding delay increasing ratio comparing with w/o scheme as demonstrated in Fig. 8.

Finally, we also study the average delay under different vehicular speed. Here, we let a vehicle start from $0m/s$ and gradually increase to $30m/s$. As shown in Fig. 9-Fig. 11, vehicular speed has little influence on the delay in crowdsensing tasks. Besides, the communication overhead has limited impact on delay when the size of P_i is small.

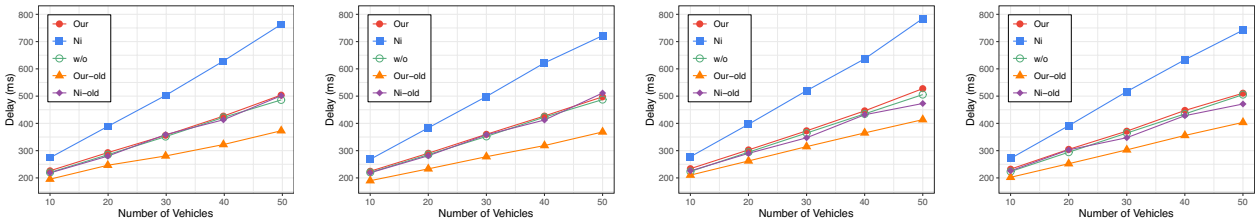
VII. Conclusion

In this paper, we have systematically investigated crowdsensing report deduplication in FVCS, and presented our FVC-Dedup scheme to support efficient and privacy-preserving report deduplication. To achieve secure task allocation and ensure crowdsensing report confidentiality, we have used the cryptographic primitives to generate the secure key for AES-128. To protect privacy in FVC-Dedup, we have improved MLE to realize privacy-preserving crowdsensing report deduplication, which can hide the identities of contributing vehicles from fog nodes and resist the fake duplicate attacks efficiently. Besides, we have employed an aggregated signature algorithm to achieve efficient signature aggregation and verification. Finally, we have constructed a reward retrieval method to reward the real contributors while detecting the greedy participants. The security analysis shows that FVC-Dedup is capable of accomplishing the design goals in FVCS. Moreover, we have implemented FVC-Dedup to evaluate the performance. It has been demonstrated that our proposed scheme is efficient and outperforms existing schemes from extensive simulation results.

VIII. Acknowledgement

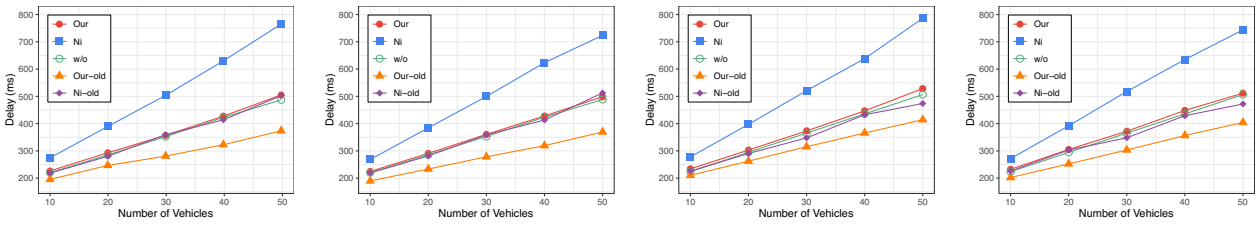
This work was supported in part by the National Natural Science Foundation of China (No.61702231, No.U1764263) and the Natural Science Foundation of Jiangsu Province (BK20170556). The work of Y. Fang was supported in part by US National Science Foundation under grant IIS-1722791.

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.2021.3069944, IEEE Transactions on Dependable and Secure Computing



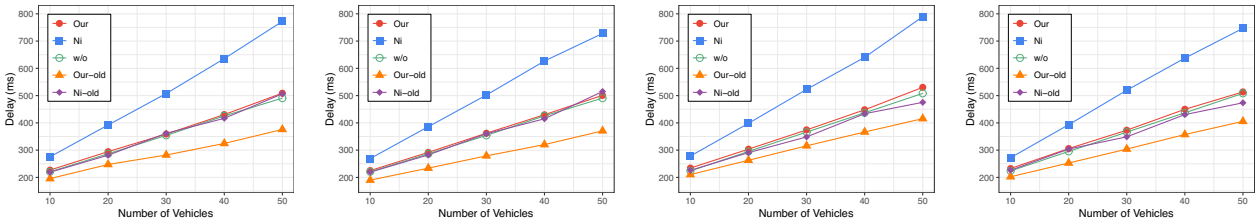
(a) $P_i = 1024$, $|Q|/n$=20%; (b) $P_i = 1024$, $|Q|/n$=40%. (c) $P_i = 2048$, $|Q|/n$=20%; (d) $P_i = 2048$, $|Q|/n$=40%.

Fig. 9. The average delay for crowdsensing tasks ($0m/s - 10m/s$).



(a) $P_i = 1024$, $|Q|/n$=20%; (b) $P_i = 1024$, $|Q|/n$=40%. (c) $P_i = 2048$, $|Q|/n$=20%; (d) $P_i = 2048$, $|Q|/n$=40%.

Fig. 10. The average delay for crowdsensing tasks ($10m/s - 20m/s$).



(a) $P_i = 1024$, $|Q|/n$=20%; (b) $P_i = 1024$, $|Q|/n$=40%. (c) $P_i = 2048$, $|Q|/n$=20%; (d) $P_i = 2048$, $|Q|/n$=40%.

Fig. 11. The average delay for crowdsensing tasks ($20m/s - 30m/s$).

REFERENCES

[1] S. Jiang, J. Liu, M. Duan, L. Wang, and Y. Fang, "Secure and privacy-preserving report de-duplication in the fog-based vehicular crowdsensing system," in *IEEE Global Communications Conference*, 2018, pp. 1–6.

[2] J. Timpner, D. Schurmann, and L. Wolf, "Trustworthy parking communities: Helping your neighbor to find a space," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 1, pp. 120–132, 2016.

[3] H. Ding, C. Zhang, Y. Cai, and Y. Fang, "Smart cities on wheels: A newly emerging vehicular cognitive capability harvesting network for data transportation," *IEEE Wireless Communications*, vol. PP, no. 99, pp. 1–10, 2017.

[4] J. Ni, A. Zhang, X. Lin, and X. Shen, "Security, privacy, and fairness in fog-based vehicular crowdsensing," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 146–152, 2017.

[5] Y. Gong, L. Wei, Y. Guo, C. Zhang, and Y. Fang, "Optimal task recommendation for mobile crowdsourcing with privacy control," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 745–756, 2017.

[6] I. Stojmenovic, "Fog computing: A cloud to the ground support for smart things and machine-to-machine networks," in *Telecommunication Networks and Applications Conference (ATNAC), 2014 Australasian*. IEEE, 2014, pp. 117–122.

[7] C. Zhu, J. Tao, G. Pastor, Y. Xiao, Y. Ji, Q. Zhou, Y. Li, and A. Ylajaaski, "Folo: Latency and quality optimized task allocation in vehicular fog computing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4150–4161, 2019.

[8] J. Jiang, L. Tang, K. Gu, and W. Jia, "Secure computing resource allocation framework for open fog computing," *The Computer Journal*, 2020.

[9] L. Wang, G. Liu, and L. Sun, "A secure and privacy-preserving navigation scheme using spatial crowdsourcing in fog-based vanets," *Sensors*, vol. 17, no. 4, 2017.

[10] A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, D. Kliazovich, and P. Bouvry, "A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 3, pp. 2419–2465, 2019.

[11] D. E. Boubiche, M. Imran, A. Maqsood, and M. Shoaib, "Mobile crowd sensing-taxonomy, applications, challenges, and solutions," *Computers in Human Behavior*, pp. 352–370, 2019.

[12] H. R. Arkian, A. Diyanat, and A. Pourkhalili, "Mist: Fog-based data analytics scheme with cost-efficient resource provisioning for iot crowdsensing applications," *Journal of Network and Computer Applications*, vol. 82, pp. 152–165, 2017.

[13] P. Yang, N. Zhang, S. Zhang, K. Yang, L. Yu, and X. Shen, "Identifying the most valuable workers in fog-assisted spatial crowdsourcing," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1193–1203, 2017.

[14] O. Consortium, "Openfog reference architecture for fog computing," *OPFRA001*, vol. 20817, p. 162, 2017.

[15] J. Sun, C. Zhang, Y. Zhang, and Y. Fang, "An identity-based security system for user privacy in vehicular ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 9, pp. 1227–1239, 2010.

[16] S. Jiang, X. Zhu, and L. Wang, "An efficient anonymous batch authentication scheme based on hmac for vanets," *IEEE*

Transactions on Intelligent Transportation Systems, vol. 17, no. 8, pp. 2193–2204, 2016.

[17] J. Wang, "When data cleaning meets crowdsourcing," https://amplab.cs.berkeley.edu/when-data-cleaning-meets-crowdsourcing/, 2015.

[18] J. Ni, X. Lin, K. Zhang, and Y. Yu, "Secure and deduplicated spatial crowdsourcing: A fog-based approach," in *IEEE Global Communications Conference*, 2017, pp. 1–6.

[19] J. Ni, K. Zhang, Y. Yong, X. Lin, and X. Shen, "Providing task allocation and secure deduplication for mobile crowdsensing via fog computing," *IEEE Transactions on Dependable & Secure Computing*, vol. 17, no. 3, pp. 581–594, 2020.

[20] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-locked encryption and secure deduplication," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2013, pp. 296–312.

[21] S. Keelveedhi, M. Bellare, and T. Ristenpart, "Dupless: server-aided encryption for deduplicated storage," in *Presented as part of the 22nd USENIX Security Symposium (USENIX Security 13)*, 2013, pp. 179–194.

[22] C. Gentry and Z. Ramzan, "Identity-based aggregate signatures," in *Public Key Cryptography-PKC 2006*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 257–273.

[23] L. Zhang, Q. Wu, B. Qin, and J. Domingo-Ferrer, "Provably secure one-round identity-based authenticated asymmetric group key agreement protocol," *Information sciences*, vol. 181, no. 19, pp. 4318–4329, 2011.

[24] S. Basudan, X. Lin, and K. Sankaranarayanan, "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, 2017.

[25] S. Basudan, A. Alamer, X. Lin, and K. Sankaranarayanan., "Efficient deduplicated reporting in fog-assisted vehicular crowdsensing," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2018, pp. 463–469.

[26] Y. Zhang, H. Su, M. Yang, D. Zheng, F. Ren, and Q. Zhao, "Secure deduplication based on rabin fingerprinting over wireless sensing data in cloud computing," *Security and Communication Networks*, pp. 1–12, 2018.

[27] S. Sharma and H. Saini, "Fog assisted task allocation and secure deduplication using 2fbo and mowo in cluster-based industrial iot (iiot)," *Computer Communications*, vol. 152, no. 2, pp. 187–199, 2020.

[28] J. Douceur, A. Adya, W. Bolosky, S. Dan, and M. Theimer, "Reclaiming space from duplicate files in a serverless distributed file system," in *in Proceedings of the 22nd International Conference on Distributed Computing Systems, IEEE*, 2002, pp. 617–624.

[29] M. Naor and O. Reingold, "Number-theoretic constructions of efficient pseudo-random functions," *Journal of the Acm*, vol. 51, no. 2, pp. 231–262, 2004.

[30] J. Xu, E.-C. Chang, , and J. Zhou, "Weak leakage-resilient client-side deduplication of encrypted data in cloud storage," in *ASIA CCS*, 2013, pp. 195–206.

[31] M. Abadi, D. Boneh, I. Mironov, A. Raghunathan, and G. Segev, "Message-locked encryption for lock-dependent messages," in *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, 2013, pp. 374–391.

[32] T. Jiang, X. Chen, Q. Wu, J. Ma, W. Susilo, and W. Lou, "Towards efficient fully randomized message-locked encryption," in *Information Security and Privacy - 21st Australasian Conference, ACISP 2016, Melbourne, VIC, Australia, July 4-6, 2016, Proceedings, Part I*, 2016, pp. 361–375.

[33] M. Bellare and S. Keelveedhi, "Interactive message-locked encryption and secure deduplication," in *PKC*, 2015, pp. 516–538.

[34] H. Cui, X. Yuan, Y. Zheng, and C. Wang, "Enabling secure and effective near-duplicate detection over encrypted in-network storage," in *IEEE INFOCOM*, 2016, pp. 1–9.

[35] L. Kazemi, C. Shahabi, and C. Lei, "Geotrucrowd: trustworthy query answering with spatial crowdsourcing," in *Acm Sigspatial International Conference on Advances in Geographic Information Systems*, 2013, pp. 314–323.

[36] L. Pournajaf, X. Li, V. Sunderam, and S. Goryczka, "Spatial task assignment for crowd sensing with cloaked locations," in *IEEE International Conference on Mobile Data Management*, vol. 1, 2014, pp. 73–82.

[37] B. Dan, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in *Advances in Cryptology-ASIACRYPT 2001*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 514–532.

[38] R. Canetti, *Decisional Diffie-Hellman Assumption*. Boston, MA: Springer US, 2005, pp. 140–142.

[39] D. Hankerson, S. Vanstone, and A. Menezes, "Guide to elliptic curve cryptography," vol. 22, no. 03, 2004.

[40] https://crypto.stanford.edu/pbc/.