Preserving Location Privacy and Accurate Task Allocation in Edge-assisted Mobile Crowdsensing

Yili Jiang[1], Kuan Zhang[1], Yi Qian[1], and Rose Qingyang Hu[2]
[1]Department of Electrical and Computer Engineering, University of Nebraska-Lincoln, USA
[2]Department of Electrical and Computer Engineering, Utah State University, USA

***Abstract*—Mobile crowdsensing enables collaborative data sensing between cloud server and mobile nodes. To participate in the sensing task, mobile nodes upload their locations to the centralized cloud for task allocation. However, revealing locations to an untrusted cloud results in privacy leakage, such as trajectories tracking and home address exposal, threatening the personal security. Obfuscation and cryptography based schemes are two main solutions to protect the location privacy. However, these schemes may either degrade the accuracy of task allocation or rely on some strong assumptions. Thus, how to protect location privacy without strong assumptions while remaining high accuracy in task allocation is challenging. In this paper, we propose a secure protocol for edge-assisted mobile crowdsensing, which removes the assumption that the cloud cannot collude with mobile nodes. Specifically, we deploy homomorphic encryption among service requestor, cloud server and edge nodes in a collaborative manner. Benefiting from the additive property of the cryptosystem, the cloud is able to securely calculate the mobile node's travel distance while knowing nothing about the mobile mode's location and task location. Based on the protocol, two types of location-dependent task allocation, travel distance based task allocation and spatial distribution based task allocation, can be implemented with location privacy preservation. Experimental results show the effectiveness of our work in task allocation. In addition, comprehensive privacy discussion indicates that the proposed protocol is secure from the collusion between cloud and mobile nodes, while preserving the task location and location privacy of mobile nodes.**

***Index Terms*—location privacy, task allocation, edge intelligence, mobile crowdsensing.**

I. Introduction

The development of smart devices and cloud/edge computing have promoted the newly-emerged mobile corowdsensing (MCS). In MCS, the centralized service provider outsources the sensing tasks to mobile nodes that are equipped with intelligent devices, such as smartphones, cameras, and build-in sensors. Since the equipment has sufficient computing/communication abilities, the mobile nodes in MCS facilitate data collection for various applications. For instance, smartphones can sense surrounding spatial data to assist with parking vacancy discovery [1]. Vehicles utilize the on-board units and global positioning systems to collect road information, contributing to traffic monitoring and road surface condition inspection [2] [3] [4]. Compared with traditional sensor networks, MCS has appealing advantages in financial cost, attracting great attention from both academia and industry.

Despite the superior benefits, MCS is subject to challenges in terms of system efficiency and privacy violations. 1) In large-scale MCS, remote data transmissions and centralized data processing consume excess communication/computational resources, resulting in lower time efficiency. To tackle this problem, edge intelligence is utilized to deploy communication and computational resources closer to mobile nodes, improving the system efficiency [5] [6]. 2) To assign sensing tasks, the service provider may recruit mobile nodes based on their locations. However, for the mobile nodes, disclosing their location information to an untrusted service provider leads to potential risks of privacy leakage, such as trajectories tracking, home/work address exposal, commute route disclosure, and so on. Since these risks threaten personal security, preserving location privacy in MCS is essential.

To protect location privacy in MCS, the state-of-the-art solutions mainly rely on obfuscation and cryptographic techniques. For instance, differential privacy (DP) is widely deployed to obfuscate the location coordinates by introducing randomized noise data [7] [8]. However, this approach degrades data utility due to the randomness, reducing the accuracy in task allocation. Without loss of data utility, secure protocols for task allocation are proposed [9]. Shen *et al.* [10] proposed a secure protocol by utilizing homomorphic encryption and Yao's garbled circuits. Although the protocol protects the mobile nodes' locations from the semi-trusted service provider, the task location is public to all entities in the system. Since the mobile nodes work in the area of task, their approximate location can be derived once the task location is revealed. To protect the task location, Jiang *et al.* [11] designed a symmetric key generator and proposed a privacy-preserving protocol based on the designed key generator, guaranteeing both enhanced privacy preservation and accurate task allocation without considering the collusion between cloud and mobile nodes. Under the collusion, preserving both mobile nodes' location and task location without degrading data utility remains challenging in location-dependent MCS.

We are motivated to propose a secure protocol to protect both the task location and mobile nodes' locations in edge-assisted MCS. The protocol is independent of the assumption that the service provider cannot collude with the mobile nodes. Compared with DP-based schemes, our approach maintains high data utility for location-dependent task allocation. The main contributions of this paper are summarized as follows.

- Deploying additively homomorphic encryption, we propose a secure protocol for edge-assisted MCS, which removes the assumption that the service provider cannot collude with the mobile nodes. The proposed protocol

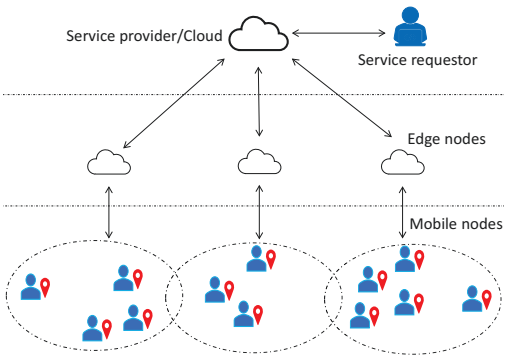


Fig. 1. System model.

preserves the task location and the location privacy of mobile nodes while remaining high data utility for location-dependent task allocation.

- Under the proposed protocol, we implement two vital types of location-dependent task allocation in MCS: travel distance based recruitment and spatial distribution based recruitment. In addition, formal correctness proof and detailed privacy discussion of the proposed protocol are provided.
- We conduct simulation experiments to evaluate the two types of task allocation under our proposed protocol. Besides, we provide a comprehensive feature comparison with the existing work. The performance comparison indicates that the proposed protocol supports the location-dependent task allocation with high data utility while preserving location privacy preservation.

The rest of this paper is organized as follows. In Section II, the system model and design goals are provided. In Section III, the preliminary is described. After that, the proposed protocol and task allocation implementation are discussed in Section IV. Detailed privacy discussion is provided in Section V. Simulation results are evaluated in Section VI and conclusions are provided in Section VII.

II. System Model

A. System Overview

Four types of entities (service requestor, cloud server, edge nodes, and mobile nodes) are normally involved in edge-assisted MCS (EMCS) as shown in Fig. 1. The detailed descriptions of each type of entity are presented as follows.

- *Service requestor:* The service requestor refers to an individual or an organization that intends to collect data from particular locations and execute data mining to support intelligent applications. For instance, a company may sense spatial data of a tourist area for plane reconstruction and passenger flow management. However, the service requestor may have limited financial budget to deploy and maintain large-scale sensor devices, or have limited computational ability to perform data mining over tremendous data. Consequently, the service requestor outsources the task to the service provider.

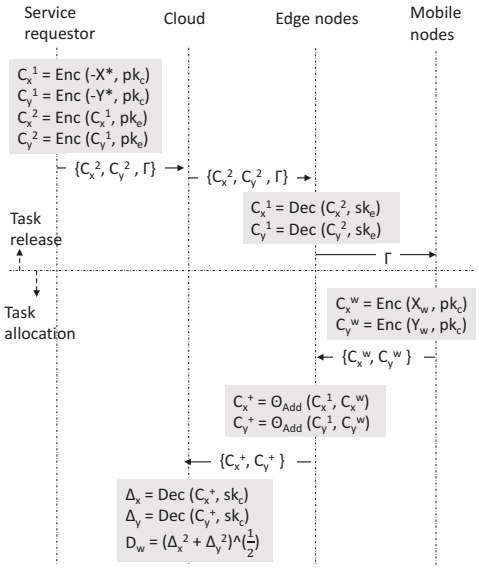


Fig. 2. Proposed protocol.

- *Cloud server:* The cloud server plays a role of service provider that has powerful computational and storage resources. It releases the task to public and recruits mobile nodes to collect data from the task area. After receiving the data, the cloud server extracts conductive information and aggregates them into a sensing report. The sensing report is returned to the cloud server. For the simplification of expression, we may use "cloud" to denote "cloud server".
- *Edge Nodes:* Edge nodes are deployed closer to mobile nodes, assisting the cloud server in task release, mobile nodes recruitment, and data transmission. The edge nodes are assumed to have sufficient communication and computational resources.
- *Mobile Nodes:* Equipped with smart devices, mobile nodes have abilities to sense data from their surroundings. When a mobile node is interested in a task, it sends its location to the cloud. The mobile nodes that are close to the task area have a high probability to be recruited.

Without loss of generality, three stages are involved in a conventional MCS system as follows. In this work, we mainly focus on the first two stages.

- *Task release:* The service requestor sends task and budget information to the cloud. The cloud then releases the task to the edge nodes and each edge node broadcasts to mobile nodes within its coverage.
- *Task allocation:* Mobile nodes submit their locations to the edge nodes for task competition. The cloud cooperates with edge nodes to assign tasks to the selected winners.
- *Data aggregation:* Mobile nodes upload the sensed data to the corresponding edge node. Cooperating with the edge nodes, the cloud performs data mining and generates final report for the service requestor.

TABLE I
NOTATION DEFINITIONS

Variable	Definition
(pk, sk)	key pair of public key and private key
Γ	task content
lcm	least common multiple
gcd	greatest common divisor
ϵ	security parameter, which is a large integer
$\mathbb{Z}_n^\star$	$\{1, 2, 3, ..., n-1\}$
$(X^\star, Y^\star)$	task location
(X_w, Y_w)	location of mobile node
D_w	travel distance of mobile node

B. Privacy Model

- The service requestor is fully trusted. It always provides real task information and performs activities honestly.
- The cloud and edge nodes are semi-trusted. On the one hand, it means that they honestly provide services following the rules. However, they are curious about the location of mobile nodes. Therefore, the cloud and edge nodes are possible to be attackers to infer the mobile nodes' locations. On the other hand, the cloud cannot collude with the edge nodes to violate the system.
- The mobile nodes could be malicious to reveal the legitimate mobile nodes' locations. To achieve this goal, these malicious mobile nodes can collude with the cloud to exchange the received information.

C. Design Goals

Under the privacy model, the following designing goals are considered in this work.

- *Location privacy preservation of mobile nodes*: The location information of a mobile node is protected from the cloud, edge nodes, service requestor and other mobile nodes. In other words, the location of a mobile node is confidential for all the other entities in the system.
- *Location privacy preservation of task area*: The task location is protected from cloud, edge nodes, and mobile nodes. Considering the fact that the recruited mobile nodes work within task area, their approximate locations can be inferred indirectly once the task location is disclosed. Therefore, task location preservation is required.
- *Accurate task allocation*: In task allocation, although the mobile nodes' locations are confidential, the cloud is able to perform location-dependent recruitment with higher accuracy.

III. PRELIMINARY

In this section, we review the Paillier cryptosystem [12], which is the design basis of our proposed protocol. As a classical additively homomorphic encryption (AHE) system, Paillier cryptosystem is able to perform operations on ciphertext to implement the additive operation of plaintext. For instance, given two plaintext m_1, m_2 and the corresponding ciphertext $c_1 = Enc(m_1), c_2 = Enc(m_2)$, a cloud server can calculate $Enc(m_1 + m_2) = c_1 \odot c_2$, where $\odot$ denotes an operation. Subsequently, by decrypting $Enc(m_1 + m_2)$, the cloud server can obtain the value of $m_1 + m_2$ without accessing the plaintext m_1 and m_2. Specifically, multiple probabilistic polynomial time algorithms are involved in the Paillier AHE as follows.

- $\mathsf{KeyGen}(\epsilon) \rightarrow (pk, sk)$: A pair of public key and private key (pk, sk) can be generated given the security parameter ϵ. Specifically, let us choose two prime numbers p, q that satisfy $p > 2^\epsilon, q > 2^\epsilon$. We set $n = pq$ and have $\lambda = \theta(n) = lcm(p-1, q-1)$, where $\theta()$ is the Carmichael's function. Randomly choose $g \in \mathbb{Z}_{n^2}^\star$ with $gcd(L(g^\lambda \bmod n^2), n) = 1$, where $L(u) = (u-1)/n$. $pk = (n, g)$ and $sk = \lambda$ are returned as the key pair.
- $\mathsf{Enc}(m, pk) \rightarrow c$: This algorithm generates the ciphertext c for plaintext m with the public key pk. In details, it randomly chooses $r \in \mathbb{Z}_n^\star$ and computes $c = g^m r^n \bmod n^2$. c is returned as the ciphertext of m.
- $\mathsf{Dec}(c, sk) \rightarrow m$: This algorithm recovers the plaintext m with the corresponding private key sk following

$$m = \frac{L(c^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} \bmod n. \tag{1}$$

- $\odot\ (c_1, c_2) \rightarrow c_+$: Calculate the ciphertext of $m_1 + m_2$ with the ciphtertext c_1, c_2. Specifically, given $c_1 = \text{Enc}(m_1, pk)$ and $c_2 = \text{Enc}(m_2, pk)$, we can calculate

$$\begin{aligned} c_+ &= \text{Enc}(m_1 + m_2, pk) && \text{(2a)} \\ &= g^{m_1+m_2}(r_1 r_2)^n \bmod n^2 && \text{(2b)} \\ &= g^{m_1} r_1^n g^{m_2} r_2^n \bmod n^2 && \text{(2c)} \\ &= c_1 c_2 \bmod n^2. && \text{(2d)} \end{aligned}$$

  Therefore, in this algorithm, the operation $\odot$ is expressed as $\odot(c_1, c_2) = c_1 c_2 \bmod n^2$. The Paillier AHE is secure from chosen plaintext attack [13].

IV. PROPOSED PROTOCOL

A. Overview of the Proposed Protocol

As shown in Fig. 2, the proposed protocol ensures privacy preservation for the stages of task release and task allocation. In task release, the service requestor encrypts the task location with the cloud's public key. The ciphertext of the task location is further encrypted with the edge node's public key. Along with the task content, all the ciphertexts are transmitted to the edge nodes through cloud. The edge node then decrypts the ciphertext with its private key and broadcasts the task content to the mobile nodes. In task allocation, the mobile nodes encrypt their locations with the cloud's public key and send to the corresponding edge node. The edge node performs operation $\odot$ over the ciphtertexts of task location and mobile node's location, and sends the results to the cloud. The cloud decrypts the results with its private key. Thus, the cloud is able to calculate the travel distance of a mobile node without revealing its real locations. Consequently, the cloud can recruit mobile nodes based on their travel distances.

B. Details of the Proposed Protocol

In this part, we introduce the details of the proposed protocol as follows. Although we consider multiple edge nodes and a set of mobile nodes in the system, without loss of generality, we describe the protocol under one edge node and one mobile node for simplicity.

1) All entities in the system generate their key pair with algorithm $\mathsf{KeyGen}(\epsilon)$. Denote the key pair of service requestor by $(pk_r = (n_r, g_r), sk_r = \lambda_r)$, the key pair of cloud by $(pk_c = (n_c, g_c), sk_c = \lambda_c)$, the key pair of an edge node by $(pk_e = (n_e, g_e), sk_e = \lambda_e)$, the key pair of a mobile node by $(pk_w = (n_w, g_w), sk_w = \lambda_w)$. Each entity publishes its public key to the system and keeps its private key secretly.
2) Given a task location $(X^\star, Y^\star)$, where $X^\star, Y^\star$ refer to the longitude and latitude respectively, the service requestor encrypts the task location with the cloud's public key by calculating

$$C_x^1 = \mathsf{Enc}(-X^\star, pk_c) \tag{3a}$$
$$= g_c^{-X^\star} r_1^{n_c} \bmod n_c^2; \tag{3b}$$

$$C_y^1 = \mathsf{Enc}(-Y^\star, pk_c) \tag{4a}$$
$$= g_c^{-Y^\star} r_1^{n_c} \bmod n_c^2. \tag{4b}$$

Consequently, the service requestor encrypts C_x^1 with the edge node's public key by calculating

$$C_x^2 = \mathsf{Enc}(C_x^1, pk_e) \tag{5a}$$
$$= g_e^{C_x^1} r_2^{n_e} \bmod n_e^2; \tag{5b}$$

$$C_y^2 = \mathsf{Enc}(C_y^1, pk_e) \tag{6a}$$
$$= g_e^{C_y^1} r_2^{n_e} \bmod n_e^2. \tag{6b}$$

The service requestor sends $\{C_x^2, C_y^2, \Gamma\}$ to the cloud and the cloud broadcasts it to the edge node. Γ is the task content, describing the type of sensed data (e.g., traffic data, health data, parking availability), reward policy, sensing hours, and so on.
3) After receiving $\{C_x^2, C_y^2, \Gamma\}$, the edge node first decrypts C_x^2, C_y^2 with its private key, receiving $C_x^1 = \mathsf{Dec}(C_x^2, sk_e)$, $C_y^1 = \mathsf{Dec}(C_y^2, sk_e)$. Then it broadcasts the task content Γ to the mobile nodes within its coverage.
4) If a mobile node is interested in the task, it encrypts its location coordinates (X_w, Y_w) with the cloud's public key by calculating

$$C_x^w = \mathsf{Enc}(X_w, pk_c) \tag{7a}$$
$$= g_c^{X_w} r_w^{n_c} \bmod n_c^2; \tag{7b}$$

$$C_y^w = \mathsf{Enc}(Y_w, pk_c) \tag{8a}$$
$$= g_c^{Y_w} r_w^{n_c} \bmod n_c^2. \tag{8b}$$

The mobile node then submits $\{C_x^w, C_y^w\}$ to the corresponding edge node.
5) The edge node calculates $C_x^+ = \odot(C_x^w, C_x^1) = \mathsf{Enc}(X_w - X^\star)$ and $C_y^+ = \odot(C_y^w, C_y^1) = \mathsf{Enc}(Y_w - Y^\star)$. $\{C_x^+, C_y^+\}$ is returned to the cloud.
6) The cloud decrypts $\{C_x^+, C_y^+\}$ with its private key, obtaining $\mathsf{Dec}(C_x^+, sk_c) = \Delta_x$, $\mathsf{Dec}(C_y^+, sk_c) = \Delta_y$, where $\Delta_x = X_w - X^\star$, $\Delta_y = Y_w - Y^\star$. Then the cloud is able to calculate the travel distance of the mobile node following

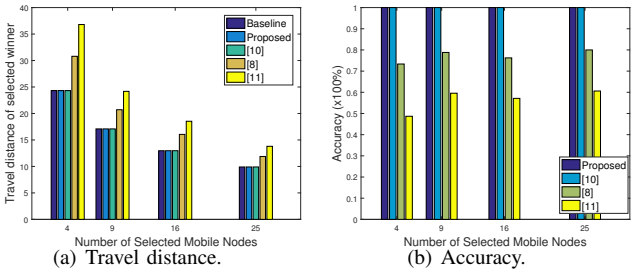
$$D_w = (\Delta_x^2 + \Delta_y^2)^{\frac{1}{2}}. \tag{9}$$

Subsequently, based on the travel distance, the cloud selects winners to perform the sensing task.

C. Location-dependent Task Allocation

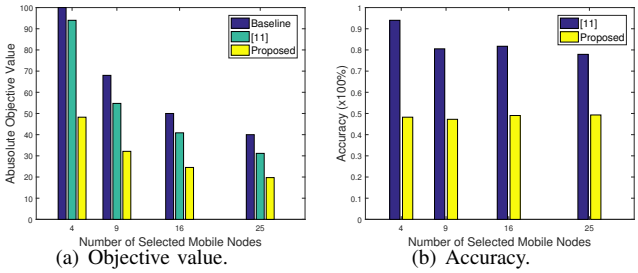
In this part, we introduce two vital recruitment systems for location-dependent task allocation: travel distance based recruitment and spatial distribution based recruitment. 1) In MCS, mobile nodes have travel costs while sensing data within a task area. The travel costs are in proportion to the travel distance. In addition, shorter travel distance results in lower latency in data collection. Thus, travel distance based recruitment is popular in MCS. 2) Spatial distribution based recruitment is important for large-scale MCS, where the cloud may recruit a set of mobile nodes. Since two mobile nodes may have overlapping sensing coverage, redundant data could be collected, leading to unnecessary costs and lower data utility. Therefore, spatial distribution is significant to avoid these issues. Under our proposed protocol, we describe how the two schemes are implemented with location privacy preservation.

- *Travel distance based recruitment* [8] [14]: In this type of recruitment system, the service provider prefers to select candidate that has the minimum travel distance to the task area for performing the sensing task. Based on the proposed protocol, the cloud is able to derive the travel distance D_w for each mobile node without revealing its locations. By comparing D_w, the cloud can determine the candidate whose travel distance is the minimum. Therefore, this recruitment system can be implemented under our proposed protocol.
- *Spatial distribution based recruitment* [11]: In this recruitment system, the locations of recruited mobile nodes are expected to be distributed as even as possible. This recruitment scheme can be implemented under our proposed protocol as the following.
  - The service provider divides the task area into K subarea, where K is the number of mobile nodes required to recruit.
  - For each subarea, the service provider encrypts its locations $(X_k^\star, Y_k^\star)(k \in [1, K])$ following the steps 2) of the protocol.
  - Each mobile node encrypts its location and sends the ciphertext to the edge node following the step 4) of the protocol.
  - The edge nodes perform operation $\odot$ on the location ciphertexts of mobile nodes for each subarea following the step 5) of the protocol.



(a) Travel distance. (b) Accuracy.

Fig. 3. Travel distance based task allocation



(a) Objective value. (b) Accuracy.

Fig. 4. Spatial distribution based task allocation

– The cloud can derive D_w^k, which is the travel distance of the mobile node to the subarea $S_k, \forall k \in [1, K]$. In each subarea, the cloud chooses the candidate that has the minimum travel distance as a winner. Then K winners are selected for the task area. The spatial distribution of the winners is approximately even.

V. Correctness and Privacy Discussion

In this section, we provide the correctness proof and privacy discussion of the proposed protocol.

A. Correctness Proof

Theorem 1: Given two large prime numbers p, q, set $n = pq$ and $\lambda = lcm(p-1, q-1)$. For any $g \in \mathbb{Z}^\star_{n^2}$, we have $g^\lambda \equiv 1 \bmod n$ and $g^{n\lambda} \equiv 1 \bmod n^2$ [12].

The proposed protocol is correct. Specifically, only the edge node can decrypt the ciphertext $\{C_x^2, C_y^2\}$ and only the cloud can decrypt the ciphertext $\{C_x^+, C_y^+\}$. Given the ciphertexts $\{C_x^1, C_y^1\}$, $\{C_x^2, C_y^2\}$, $\{C_x^w, C_y^w\}$ as shown in the above, the correctness at edge side is ensured by the following equations.

$$\mathsf{Dec}(C_x^2, sk_e) \tag{10a}$$
$$= \frac{L((C_x^2)^{\lambda_e} \bmod n_e^2)}{L(g_e^{\lambda_e} \bmod n_e^2)} \bmod n_e \tag{10b}$$
$$= \frac{L(g_e^{\lambda_e C_x^1} r_2^{\lambda_e n_e}) \bmod n_e^2)}{L(g_e^{\lambda_e} \bmod n_e^2)} \bmod n_e \tag{10c}$$
$$= \frac{L(g_e^{\lambda_e C_x^1} \bmod n_e^2)}{L(g_e^{\lambda_e} \bmod n_e^2)} \bmod n_e \tag{10d}$$
$$= \frac{n_e \eta C_x^1 + 1 - 1 \bmod n_e^2}{n_e \eta + 1 - 1 \bmod n_e^2} \bmod n_e \tag{10e}$$
$$= C_x^1, \tag{10f}$$

where Eq. (10c)-(10e) are derived from Theorem 1. Similarly, the edge node can achieve $\mathsf{Dec}(C_y^2, sk_e) = C_y^1$. To perform the algorithm $\odot()$, the edge node calculates the following over the ciphertext.

$$C_x^+ = \odot(C_x^w, C_x^1) \tag{11a}$$
$$= C_x^w C_x^1 \bmod n_c^2 \tag{11b}$$
$$= g_c^{X_w} r_w^{n_c} g_c^{-X^\star} r_1^{n_c} \bmod n_c^2 \tag{11c}$$
$$= g_c^{X_w - X^\star} (r_w r_1)^{n_c} \bmod n_c^2. \tag{11d}$$

Similarly, $C_y^+ = g_c^{Y_w - Y^\star} (r_w r_1)^{n_c} \bmod n_c^2$ is achieved.

Given the ciphertext $\{C_x^+, C_y^+\}$, the correctness at cloud side is ensured as follows.

$$\mathsf{Dec}(C_x^+, sk_c) \tag{12a}$$
$$= \frac{L((C_x^+)^{\lambda_c} \bmod n_c^2)}{L(g_c^{\lambda_c} \bmod n_c^2)} \bmod n_c \tag{12b}$$
$$= \frac{L(g_c^{\lambda_e (X_w - X^\star)} (r_w r_1)^{\lambda_c n_c}) \bmod n_c^2)}{L(g_c^{\lambda_c} \bmod n_c^2)} \bmod n_c \tag{12c}$$
$$= \frac{L(g_c^{\lambda_c (X_w - X^\star)} \bmod n_c^2)}{L(g_c^{\lambda_c} \bmod n_c^2)} \bmod n_c \tag{12d}$$
$$= \frac{n\theta (X_w - X^\star) + 1 - 1 \bmod n_e^2}{n\theta + 1 - 1 \bmod n_e^2} \bmod n_e \tag{12e}$$
$$= X_w - X^\star = \Delta_x. \tag{12f}$$

Similarly, the cloud can decrypt C_y^+ and receive Δ_y.

B. Privacy Discussion

- *Location privacy of mobile nodes*: In the proposed protocol, the mobile nodes encrypt their locations with the cloud's public key and generate $\{C_x^w, C_y^w\}$. For the other entities in the system, only the corresponding edge node can access $\{C_x^w, C_y^w\}$. Since the Paillier AHE is secure from chosen plaintext attack [13], the edge node cannot decrypt the ciphertext. Thus, the location privacy of mobile nodes is preserved.
- *Location privacy of task area*: The service provider decides the task location and generates the ciphertext $\{C_x^1, C_y^1\}$ with the cloud's public key. Since $\{C_x^1, C_y^1\}$ is further encrypted to $\{C_x^2, C_y^2\}$ with the edge node's public key, only the edge node can access $\{C_x^1, C_y^1\}$. However, since the Paillier AHE is secure from chosen plaintext attack, the edge node cannot decrypt $\{C_x^1, C_y^1\}$ to obtain the task location. At the cloud side, the cloud only receives the ciphertext $\{C_x^+, C_y^+\}$. Although the cloud can calculate the travel distance D_w based on the ciphertext, it cannot derive the task location or mobile node's location. In addition, although the cloud is able to collude with the malicious mobile nodes, the task location is preserved since the mobile nodes have no knowledge about $\{C_x^1, C_y^1\}$. Therefore, the task location is preserved.

TABLE II
FEATURES COMPARISON.

Features	Privacy Preservation			Task allocation	
	Location of mobile nodes	Task location	Secure from collusion of cloud and mobile nodes	Support travel distance based task allocation	Support spatial distribution based task allocation
Wang *et al.* [8]	✓	×	✓	✓	×
Ni *et al.* [9]	✓	✓	✓	×	×
Shen *et al.* [10]	✓	×	✓	✓	×
Jiang *et al.* [11]	✓	✓	×	✓	✓
He *et al.* [14]	×	×	×	✓	×
Proposed	✓	✓	✓	✓	✓

Notes: "✓" represents "satisfy"; "×" represents "not satisfy".

VI. PERFORMANCE EVALUATIONS

In this section, we first evaluate the performance over the above two types of task allocation by comparing them with [8], [10] and [11]. We then provide a comprehensive feature comparison with other research works. To simulate the task allocation, the targeted area is a parking lot in Omaha, NE, USA (the area is approximately 200 meters $\times$ 200 meters). 60 candidates are considered in the task area and each candidate is randomly distributed in spatial.

Fig. 3 shows the performance of travel distance based task allocation. Compared with [8] and [11], the proposed scheme and [10] perform the best and achieve the highest accuracy. This is benefited from the additive property of homomorphic encryption, which remains high data utility for task allocation.

Fig. 4 describes the performance of spatial distribution based task allocation. In the comparison, we deploy the absolute objective value in [11] to evaluate the performance. It is defined as the absolute difference between the minimum travel distance and the minimum winner-to-winner distance. A higher value indicates better performance. As shown in Fig. 4, our proposed performs worse than [11], since only one parameter (travel distance) is considered when recruiting mobile nodes under the proposed protocol. In other words, our performance is degraded due to a lack of consideration of other spatial parameters. Even so, it should be noted that our proposed is able to support spatial based task allocation.

Table II shows the features comparison of the proposed with the other work. It is noted that our proposed satisfies all the features in the comparison. In addition, the proposed scheme outperforms [10] since it protects the task location and supports spatial based task allocation. Another crucial observation is that compared with [11], the proposed scheme is secure from the collusion of cloud and mobile nodes, although it has performance degradation in spatial based task allocation.

VII. CONCLUSION

In this paper, we have proposed an additively homomorphic encryption based protocol for EMCS. The protocol supports travel distance based task allocation and spatial distribution based task allocation, while preserving location privacy. We have provided detailed proof and discussion to show the correctness and privacy preservation of the protocol. The simulation results have demonstrated that our proposed protocol achieves high accuracy in task allocation while remaining location privacy preservation. In future work, we will build a testbed for the proposed protocol to evaluate the accuracy of task allocation and time efficiency in practical.

ACKNOWLEDGMENT

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

REFERENCES

[1] F. Bock, S. Di Martino, and A. Origlia, "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, 2020.

[2] C. Wang, Z. Xie, L. Shao, Z. Zhang, and M. Zhou, "Estimating travel speed of a road section through sparse crowdsensing data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 9, pp. 3486–3495, 2019.

[3] 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.

[4] S. Abdul Rahman, A. Mourad, and M. El Barachi, "An infrastructure-assisted crowdsensing approach for on-demand traffic condition estimation," *IEEE Access*, vol. 7, pp. 163 323–163 340, 2019.

[5] J. Li, Z. Su, D. Guo, K.-K. R. Choo, Y. Ji, and H. Pu, "Secure data deduplication protocol for edge-assisted mobile crowdsensing services," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 1, pp. 742–753, 2021.

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

[7] M. Yang, T. Zhu, Y. Xiang, and W. Zhou, "Density-based location preservation for mobile crowdsensing with differential privacy," *IEEE Access*, vol. 6, pp. 14 779–14 789, 2018.

[8] Z. Wang, J. Hu, R. Lv, J. Wei, Q. Wang, D. Yang, and H. Qi, "Personalized privacy-preserving task allocation for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 6, pp. 1330–1341, 2019.

[9] J. Ni, K. Zhang, Q. Xia, X. Lin, and X. S. Shen, "Enabling strong privacy preservation and accurate task allocation for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 19, no. 6, pp. 1317–1331, 2020.

[10] Y. Shen, L. Huang, L. Li, X. Lu, S. Wang, and W. Yang, "Towards preserving worker location privacy in spatial crowdsourcing," in *2015 IEEE Global Communications Conference (GLOBECOM)*, 2015, pp. 1–6.

[11] Y. Jiang, K. Zhang, Y. Qian, and L. Zhou, "P2AE: Preserving privacy, accuracy, and efficiency in location-dependent mobile crowdsensing," *IEEE Transactions on Mobile Computing*, pp. 1–16, 2021.

[12] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology — EUROCRYPT '99*, J. Stern, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 223–238.

[13] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1333–1345, 2018.

[14] S. He, D.-H. Shin, J. Zhang, and J. Chen, "Near-optimal allocation algorithms for location-dependent tasks in crowdsensing," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3392–3405, 2017.