Privacy-Preserving Autonomous Cab Service Management Scheme

Ahmed Sherif
Tennessee State University
Nashville, TN, USA
asherif@tnstate.edu

Ahmad Alsharif
Tennessee Tech. University
Cookeville, TN, USA
ahalsharif42@students.tntech.edu

Mohamed Mahmoud
Tennessee Tech. University
Cookeville, TN, USA
mmahmoud@tntech.edu

Jacob Moran
Tennessee Tech. University
Cookeville, TN, USA
jemoran@students.tntech.edu

ABSTRACT

In the autonomous vehicles era, vehicles will be an on-demand service rather than an owned product, i.e., many passengers will rely on Autonomous Cabs (ACs) in their transportation. In order to guarantee the high quality of the AC service, the AC company needs to learn the geographic distribution of the potential service requests. The best way to obtain this information is by requesting the passengers to frequently report their locations, e.g., by using their smartphones. However, learning the passengers' locations causes a serious location privacy issue. In this paper, we propose a privacy-preserving scheme for reporting location information for AC management. Data aggregation approach is used to preserve location privacy by providing the AC company with the total number of requests in each geographic area, while hiding the individual reports of the passengers. Unlike the existing aggregation schemes that do binary data addition, the used aggregation scheme does individual bits addition. Our analysis and experimental results demonstrate that the proposed scheme is efficient and can preserve location privacy.

Keywords

Privacy Preservation; Autonomous Cabs;

1. INTRODUCTION

Autonomous Vehicles (AVs) are equipped with advanced sensing and communication capabilities, navigation devices such as GPS and radar, computer vision technology, etc., to enable the vehicles to autonomously drive themselves without any human intervention [6]. According to recent reports, the Alphabet Inc's Waymo fleet of self-driving vehicles drove more than 600,000 miles in 2016 on public roads in California [9]. AVs have the potential to fundamentally improve the transportation systems by reducing crashes, assisting traffic flows, and reducing travel time.

Currently, human-driven cars are mostly personal devices, but it is anticipated that AVs will lead us from vehicles as an owned product to an on-demand service. Unlike the current cab service that can be considered a secondary mode of transportation, it is anticipated that Autonomous Cabs (ACs) will be the main means of transportation for many people. The idea is that each user will have an account in an AC company and he can use his smartphone to request an AC when needed. Due to the automation and elimination of human effort, ACs will be able to serve people anywhere at anytime.

In order to provide a high-quality service with acceptable operating cost, the AC companies need to know the geographic distribution of the potential service requests. The distribution information can be used for AC service management as follows: (1) operating enough number of ACs, where other ACs can be in service maintenance or in charge; (2) determining the parking locations of the ACs so that they can respond to requests quickly; a good service can be offered when the ACs are physically close to passengers; and (3) selecting the routes of the ACs, i.e., after dropping off a passenger, the ACs can take a route where there is a large chance that a customer close to the route requests the service. The best way to enable the AC companies to collect the information they need is by asking users to run an application on their smartphones to periodically send their location information. However, people may be reluctant to report their locations due to privacy concerns, because location information can be used to launch physical attacks on users. Reporting coarse-grained location information to provide a level of privacy protection reduces the utility of the data and cannot completely resolve the privacy concern because knowing that someone is in a ceratin area can be used to know its exact location, such as workplace and residence.

In this paper, we propose a privacy-preserving scheme for reporting location information for AC management. Data aggregation approach is used to preserve location privacy by providing the AC company with the total number of requests in each geographic area, called cell, while hiding the individual reports of the passengers. Unlike the existing ag-

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

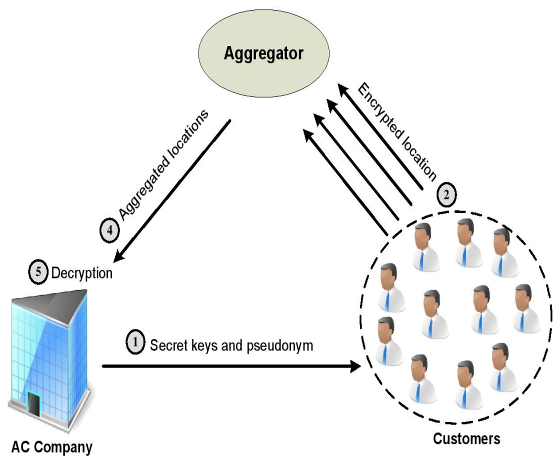

Figure 1: Network model and exchanged messages.

gregation schemes, like [10], that do binary data addition, the used aggregation scheme does individual bits addition, which is more appropriate for AC service management application. Our analysis and experimental results demonstrate that the proposed scheme is efficient and can preserve location privacy.

The remainder of this paper is organized as follows. The network and threat models are discussed in section 2. The proposed privacy-preserving autonomous cab service management scheme is discussed in section 3. The security analysis and performance evaluations are discussed in section 4. The related works are summarized in section 5. Finally, conclusions are drawn in section 6.

2. NETWORK AND THREAT MODELS

Network Model. As shown in Figure 1, the considered network model has three main entities: the AC company, users, and an aggregator. Users can use a smartphone application to report their locations. Initially, the users should communicate with the AC company to receive unique secret keys needed to encrypt their location information. In order to report his location, each user first generates an encrypted report containing the location information. Then, he sends this encrypted report to the aggregator by the smartphone application. The aggregator can be a third party server or one of the users that act as a head of a cluster of users. After receiving the encrypted reports, the aggregator aggregates the reports without revealing any private location information. Finally, the aggregator sends the aggregated reports to the AC company. The company decrypts the aggregated reports to learn the total number of users requesting the service in each area.

Threat Model. The aggregator and the AC company are considered "honest-but-curious". Specifically, they run the scheme honestly and do not aim to disrupt the proper operation of the scheme. However, they are curious to learn the location information of the individual users. Users may try to learn the other users' locations.

3. THE PROPOSED SCHEME

3.1 Aggregation Scheme

In this section, we explain an aggregation scheme based on the kNN encryption scheme [2,7,20]. The scheme has a primary user and several secondary users. The primary user is the entity that computes the keys and distributes them to the secondary users. The secondary users encrypt their data, which is in the form of a binary vector of size n, by using their keys, and then these ciphertexts are aggregated and sent to primary user. The primary user uses its key to decrypt the aggregated data. The aggregation scheme has the following phases.

Sharing Keys. The key set of the primary user is [S, M_1N_1, M_1N_2, M_2N_3, M_2N_4], where S is a binary vector of size n, and $(M_1, M_2, N_1, N_2, N_3, N_4)$ are $n \times n$ invertible matrices. For each secondary user U_i, the corresponding key set is [S, $N_1^{-1}M_i^{'}$, $N_2^{-1}M_i^{''}$, $N_3^{-1}M_i^{'''}$, $N_4^{-1}M_i^{''''}$], where $M_i{'} + M_i{''} = M_1^{-1}$, and $M_i{'''} + M_i^{''''} = M_2^{-1}$. The primary and secondary users share S and each secondary user has different ($N_1^{-1}M_i^{'}$, $N_2^{-1}M_i^{''}$) and ($N_3^{-1}M_i^{'''}$, $N_4^{-1}M_i^{''''}$) because they have different $M_i{'}$, $M_i{''}$, $M_i{'''}$ and $M_i^{''''}$.

Encryption. Each secondary user should use the binary vector S as splitting indicator to split his data vector (q_i) into two random vectors $q_i^{'}$ and $q_i^{''}$ as follows. If the j^{th} bit of S is one, $q_i'(j)$ and $q_i''(j)$ are set similar to $q_i(j)$, while if it is zero, $q_i'(j)$ and $q_i''(j)$ are set to two random numbers such that their summation is equal to $q_i(j)$. Finally, the secondary user's vector pair (q_i', q_i'') is encrypted to I_i $=[N_1^{-1}M_i^{'}q_i^{'},\ N_2^{-1}M_i^{''}q_i^{'},\ N_3^{-1}M_i^{'''}q_i^{''},\ N_4^{-1}M_i^{''''}q_i^{''}]$, where the ciphertext I_i is called an index, and $q_i^{'}$ and $q_i^{''}$ are column vectors.

Aggregation. I_i is a column vector with the size of $4n$ elements. For m users, each user U_i creates an index I_i as follows.

$$\begin{aligned} I_1 &= [a_{1,1}, a_{1,2}, \ldots, a_{1,4n}]^T \\ I_2 &= [a_{2,1}, a_{2,2}, \ldots, a_{2,4n}]^T \\ &\vdots \\ I_m &= [a_{m,1}, a_{m,2}, \ldots, a_{m,4n}]^T \end{aligned}$$

where $a_{i,j}$ indicates the j^{th} element in the i^{th} index. The aggregated index I_{agg} can be computed by summing all m users' indices as

$$\begin{aligned} I_{agg} &= \sum_{i=1}^{m} I_i \\ &= \left(\sum_{i=1}^{m} a_{i,1}, \sum_{i=1}^{m} a_{i,2}, \sum_{i=1}^{m} a_{i,3}, \ldots, \sum_{i=1}^{m} a_{i,4n} \right) \\ &= [a_1, a_2, \ldots, a_{4n}] \end{aligned}$$

Decryption. After decrypting an aggregated index, the format of the plaintext is a vector where each element gives the summation of the individual bits of one element in the binary vectors of the secondary users. To obtain the plaintext of one element, the dot product should be done between the aggregated index and another index encrypted by the primary user where the index's vector (p) has "1" in the position of the element and 0's in all other elements. This procedure is similar to measuring the similarity of the two indices used in [3,20], but we use it on the aggregated index. After creating the data vector (p), the primary user encrypts it as follows. First, it uses the binary vector S as splitting indicator to split p into two random vectors $p^{'}$ and $p^{''}$ as follows. If the j-th bit of S is zero, $p'(j)$ and $p''(j)$ are set similar to $p(j)$, and if the j-th bit of S is one, $p'(j)$ and $p''(j)$ are set to two random numbers so that their summation is equal to $p(j)$. Then, the vector pair (p', p'') is encrypted as $T = [p^{'}M_1N_1, p^{'}M_1N_2, p^{''}M_2N_3, p^{''}M_2N_4]$.

Numerical Example. To further illustrate our scheme, we provide a numerical example. Let each vector has four bits ($n = 4$) and S, M_1, M_2, N_1, N_2, N_3 and N_4 are as follows.

$S = 1001$

$$M_1 = \begin{pmatrix} 0.92 & 0.15 & 0.70 & 0.57 \\ 0.33 & 1.00 & 0.30 & 0.48 \\ 0.24 & 0.39 & 0.03 & 0.74 \\ 0.97 & 0.54 & 0.48 & 0.19 \end{pmatrix}, M_2 = \begin{pmatrix} 0.57 & 0.70 & 0.87 & 0.13 \\ 0.77 & 0.51 & 0.83 & 0.55 \\ 0.27 & 0.00 & 0.87 & 0.75 \\ 0.23 & 0.22 & 0.03 & 0.90 \end{pmatrix}$$

$$N_1 = \begin{pmatrix} 0.81 & 0.63 & 0.96 & 0.96 \\ 0.91 & 0.10 & 0.96 & 0.49 \\ 0.13 & 0.28 & 0.16 & 0.80 \\ 0.91 & 0.55 & 0.97 & 0.14 \end{pmatrix}, N_2 = \begin{pmatrix} 0.42 & 0.66 & 0.67 & 0.65 \\ 0.92 & 0.04 & 0.76 & 0.17 \\ 0.79 & 0.85 & 0.74 & 0.70 \\ 0.96 & 0.93 & 0.39 & 0.03 \end{pmatrix}$$

$$N_3 = \begin{pmatrix} 0.28 & 0.69 & 0.44 & 0.19 \\ 0.05 & 0.31 & 0.38 & 0.49 \\ 0.10 & 0.95 & 0.77 & 0.45 \\ 0.82 & 0.03 & 0.80 & 0.64 \end{pmatrix}, N_4 = \begin{pmatrix} 0.71 & 0.66 & 0.96 & 0.75 \\ 0.75 & 0.16 & 0.34 & 0.26 \\ 0.28 & 0.12 & 0.59 & 0.51 \\ 0.68 & 0.50 & 0.22 & 0.70 \end{pmatrix}$$

By using these matrices along with the binary vector S, we can calculate the keys of the primary and secondary users. The primary user's key is [S, M_1N_1, M_1N_2, M_2N_3, M_2N_4], and each secondary user's key is [S, $N_1^{-1}M_i^{'}$, $N_2^{-1}M_i^{''}$, $N_3^{-1}M_i^{'''}$, $N_4^{-1}M_i^{''''}$].
Let the bit vector of the first secondary user is $q_1 = 1101$. After splitting q_1 with S, we get

$$q_1' = (1 \quad 0.5 \quad 0.3 \quad 1)^T$$
$$q_1'' = (1 \quad 0.5 \quad -0.3 \quad 1)^T$$

Let $M_1{}', M_1{}'', M_1{}''', M_1{}''''$ of the first secondary user be

$$M_1' = \begin{pmatrix} -0.30 & -0.80 & 0.30 & 0.80 \\ -0.20 & 0.30 & -0.14 & 0.20 \\ 1.27 & 1.20 & -1.32 & -1.17 \\ 0.30 & -0.11 & 0.80 & -0.30 \end{pmatrix}, M_1'' = \begin{pmatrix} -0.57 & -0.59 & 0.74 & 1.27 \\ -0.45 & 0.81 & -0.10 & 0.13 \\ 1.00 & 0.54 & -1.00 & -1.00 \\ 0.23 & -0.04 & 0.43 & -0.46 \end{pmatrix}$$

$$M_1''' = \begin{pmatrix} -1.00 & 1.00 & -1.00 & -0.8 \\ 1.20 & -1.50 & -0.40 & 0.80 \\ 0.40 & -0.50 & 0.80 & -0.20 \\ 0.00 & -0.30 & 0.10 & 0.70 \end{pmatrix}, M_1'''' = \begin{pmatrix} -1.48 & 2.85 & -0.16 & -0.23 \\ 1.29 & -0.55 & -0.18 & 0.58 \\ 0.37 & -0.30 & 0.36 & -0.39 \\ 0.00 & -0.15 & 0.3.0 & 0.36 \end{pmatrix}$$

The index of the first secondary user is

$I_1 = [N_1^{-1}M_1'q_1', N_2^{-1}M_1''q_1', N_3^{-1}M_1'''q_1'', N_4^{-1}M_1''''q_1'']$

$= [3.26 \quad 0.49 \quad -3.19 \quad 0.32 \quad -2.79 \quad 1.18 \quad 3.94 \quad -2.49 \quad -1.52 \quad -1.09 \quad -1.76 \quad 5 \quad 3.22 \quad -1.58 \quad -0.26 \quad -1.64]^T$

By using the same procedure and assuming that the binary vector of the second secondary user is $q_2 = 1001$ After splitting q_2 with S, we get

$$q_2' = (1 \quad 0.2 \quad 0.7 \quad 1)^T$$
$$q_2'' = (1 \quad -0.2 \quad -0.7 \quad 1)^T$$

Let $M_2{}', M_2{}'', M_2{}''', M_2{}''''$ of the second secondary user be

$$M_2' = \begin{pmatrix} -0.40 & -1.00 & 1.00 & 1.07 \\ -0.30 & 0.10 & -0.10 & 0.30 \\ 1.00 & 0.74 & -2.00 & -0.17 \\ 0.20 & -0.10 & 1.00 & -0.40 \end{pmatrix}, M_2'' = \begin{pmatrix} -0.47 & -0.39 & 0.40 & 1.00 \\ -0.35 & 1.00 & -0.14 & 0.03 \\ 1.27 & 1.00 & -0.32 & -2.00 \\ 0.33 & -0.05 & 0.23 & -0.36 \end{pmatrix}$$

$$M_2''' = \begin{pmatrix} -1.48 & 2.00 & -0.16 & -1.00 \\ 1.49 & -0.05 & -0.20 & 0.38 \\ 0.60 & -0.40 & 1.10 & -0.30 \\ 0.00 & -0.20 & 0.20 & 1.00 \end{pmatrix}, M_2'''' = \begin{pmatrix} -1.00 & 1.85 & -1.00 & -0.03 \\ 1.00 & -2.00 & -0.38 & 1.00 \\ 0.17 & -0.40 & 0.06 & -0.29 \\ 0.00 & -0.25 & 0.20 & 0.06 \end{pmatrix}$$

The index of the second secondary user is

$I_2 = [N_1^{-1}M_2'q_2', N_2^{-1}M_2''q_2', N_3^{-1}M_2'''q_2'', N_4^{-1}M_2''''q_2'']$

$= [-17.58 \quad -0.05 \quad 17.17 \quad -1.09 \quad -6.38 \quad 3.23 \quad 8.88 \quad -7.18 \quad -6.41 \quad -5.23 \quad 3.75 \quad 5.17 \quad 5.17 \quad -4.38 \quad -0.04 \quad -1.93]^T$

I_1 and I_2 are aggregated to obtain the following aggregated index (I_{agg}).

$I_{agg} = [-14.32 \quad 0.44 \quad 13.98 \quad -0.77 \quad -9.17 \quad 4.41 \quad 12.82 \quad -9.67 \quad -7.93 \quad -6.32 \quad 1.99 \quad 10.17 \quad 8.39 \quad -5.96 \quad -0.3 \quad -3.57]^T$

In order to decrypt only the first element in the aggregated index, the encryption (T) of the binary vector $p = 1000$ with the primary user's key is needed. To compute T, split p with S to get:

$$\begin{pmatrix} p' \\ p'' \end{pmatrix} = \begin{pmatrix} 0.04 & 0 & 0 & 0.2 \\ 0.96 & 0 & 0 & -0.2 \end{pmatrix}$$

T is computed as follows

$T = [p'M_1N_1, p'M_1N_2, p''M_2N_3, p''M_2N_4]$

$= [0.36 \quad 0.23 \quad 0.41 \quad 0.39, \quad 0.36 \quad 0.32 \quad 0.36 \quad 0.26, \quad 0.21 \quad 1.33 \quad 1.05 \quad 0.74, \quad 1.02 \quad 0.5 \quad 1.17 \quad 0.92]$

To decrypt the first element in the aggregated index which is two in our example, we need to calculate the inner product between the aggregated index I_{agg} and T. In addition, by multiplying I_1 and T (or I_2 and T) we get 1 which is the number of common ones in the two indices. This is called a similarity measurement technique that is used in keywords search and ride sharing organization in [3, 18].

3.2 Privacy-preserving AC Service Management

System Initialization. The AC company creates its key and computes the secret keys of each user as explained in section 3. The ACs service area is divided into cells where each cell is represented by one bit in a binary vector. As shown in Figure 2, the area is divided into 42 cells (6 rows and 7 columns). $b_{i,j}$ is one bit in the location data vector that represents cell (i, j), where $b_{i,j} \in \{0, 1\}$, i and j are the row and column numbers, respectively. In order to simply explain our idea, Figure 2 shows a small area of only 42 cells, but our scheme can be used for a much larger area.

Sending Aggregated Reports. When a user reports his location, he first composes a binary vector using the format given in Figure 2. Only one bit is set to "1" which represents the location of the user. For example, a user's vector of value $[\mathbf{1}00\ldots0]$ indicates that the user's location is cell (0,0). Then, the user uses his secret key to encrypt this vector using the scheme illustrated in section 3. Finally, the user signs the index to authenticate himself and then sends

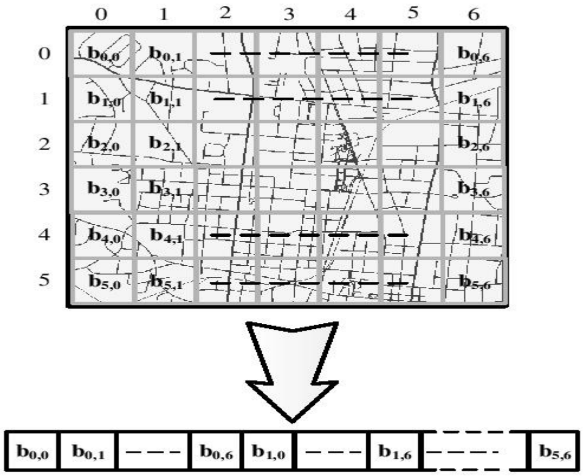
0 1 2 3 4 5 6
0 1 2 3 4 5
$b_{0,0}$ $b_{0,1}$ $b_{0,6}$
$b_{1,0}$ $b_{1,1}$ $b_{1,6}$
$b_{2,0}$ $b_{2,1}$ $b_{2,6}$
$b_{3,0}$ $b_{3,1}$ $b_{3,6}$
$b_{4,0}$ $b_{4,1}$ $b_{4,6}$
$b_{5,0}$ $b_{5,1}$ $b_{5,6}$
$b_{0,0}$ $b_{0,1}$ --- $b_{0,6}$ $b_{1,0}$ --- $b_{1,6}$ ---- $b_{5,6}$

Figure 2: Representing the city cells as binary vector.

the packet to the aggregator.

Decryption. The aggregator should use the aggregation procedure discussed in section 3 to aggregate the indices. Then, the aggregator sends the aggregated index to the AC company. The company should use its secret key to decrypt the aggregated index to learn the number of requests in each cell.

4. EVALUATIONS

4.1 Security/Privacy Analysis

Location privacy. Our scheme can preserve the users' location privacy as the users' locations are encrypted by their keys and without knowing the keys it is infeasible to decrypt the indices. The aggregator can not use the indices reported by one user at different times to know whether the user is still in the same location. This is because the users use different random numbers each time they encrypt their binary vectors. Also, our scheme can preserve the users' locations privacy from the company by reporting only the aggregated index to it. The company can only know the total number of users in each cell, but it cannot know the identities of the users in a cell.

Authentication. By signing the reports, the users can authenticate themselves to the aggregator. This can protect against impersonation attacks and prevent accepting the reports sent by external attackers.

The aggregator can not decrypt the indices of the users. The index of each user is encrypted by using the kNN encryption scheme, which is proven to be secure as long as the keys are unknown. As we explained earlier, the secret key $(S, M_1, M_2, N_1, N_2, N_3, N_4)$ is needed to decrypt the user's index. As the aggregator does not have this key, it can not decrypt the indices to know the locations of the users.

Each user can not compute other users' keys and the AC company's key. As the key of each user has $[S,\ N_1^{-1}M_i^{'},\ N_2^{-1}M_i^{''},\ N_3^{-1}M_i^{'''},\ N_4^{-1}M_i^{''''}]$, the attackers do not know the values of N_1, N_2, N_3, and N_4 to extract $M_i^{'}$ and $M_i^{''}$ to compute M_1. They also cannot extract $M_i^{'''}$ and $M_i^{''''}$ to compute M_2. As a result, the users can not compute the AC company's key from their keys. In addition, as the company uses different $M_i^{'}, M_i^{''}, M_i^{'''}$, and $M_i^{''''}$ for each user in creating its key, it is impossible to compute a user's keys as long as $M_i^{'}, M_i^{''}, M_i^{'''}, M_i^{''''}, S, N_1, N_2, N_3$, and N_4 are unknown.

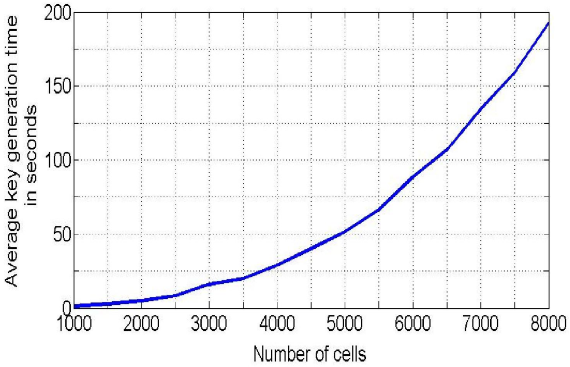

Figure 3: Average key generation time.

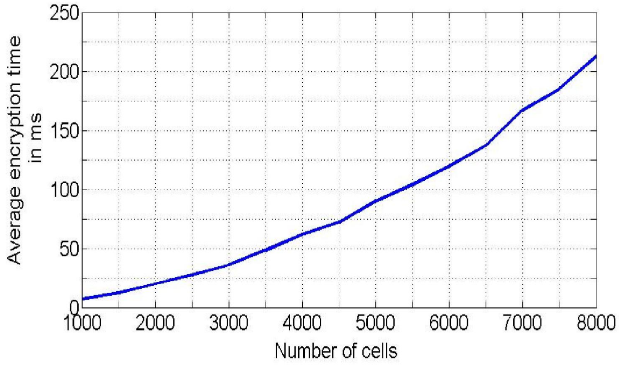

Figure 4: Average encryption time.

4.2 Performance Evaluations

We implemented our scheme using MATLAB and conducted experiments on a server with an Intel Core i7-4765T Processor @ 2.0 GHZ and 8.00 GB RAM.

Figure 3 gives the average key generation time at different numbers of cells. It can be seen that the time increases as the number of cells increases because the key size increases. Although the average key generation time is relatively long, this does not degrade the scheme performance because keys are generated at system initialization phase and are used for a long period of time.

Figure 4 gives the average encryption time for different numbers of cells. It can be seen that as the number of cells increases, the encryption time increases because the data vector size increases, and thus more multiplication operations are needed.

Figure 5 gives the average decryption time of each element in the aggregated index for different numbers of cells. It can be seen that the average decryption time increases as the number of cells increases because the size of the indices increases and thus more dot product operations are needed.

Figure 6 gives the average aggregation time for different numbers of cells and users. It can be seen that the aggregation time is very short and increases as the number of cells increases because the size of the indices increase and thus more addition operations are needed. It can also be

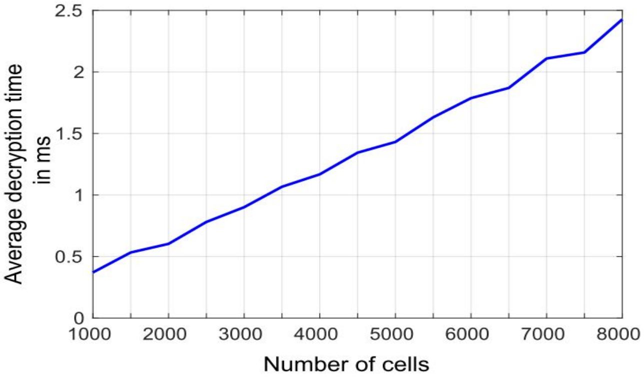

Figure 5: Average decryption time for each element in the aggregated index.

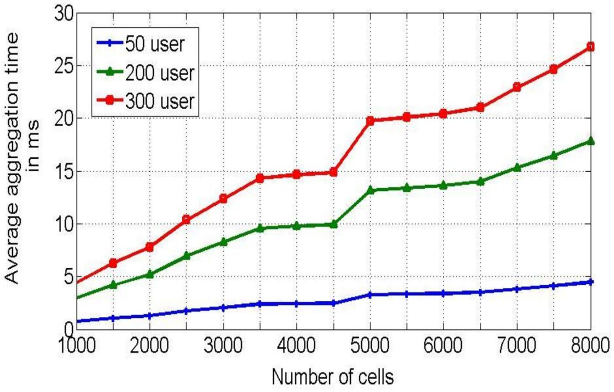

Figure 6: Average Aggregation time.

seen that more time is needed to aggregate indices for more users.

5. RELATED WORKS

Several data aggregation schemes have been proposed to allow an aggregator to aggregate data without learning the individual or the aggregated data [4]. In [4], each user splits his data into random shares; one share for each member in the group. Then, each user sends the aggregated shares to the aggregator, who obtains the final aggregation result. Since the aggregator only receives aggregated values over random shares, it does not know the original data. For the homomorphic encryption based aggregation schemes [8, 10], the multiplication of the ciphertexts can result in the ciphertext of the aggregated data. The schemes need two keys. The public key is used in encrypting the data while the private key is used in decryption. In [8], Q. Li et al. proposed an efficient scheme to obtain the Sum and Min aggregate of time-series data. The proposed scheme employs an additive homomorphic encryption to ensure that the aggregator can only obtain the sum of all users' data, without knowing individual user's data or intermediate results. However, these schemes either require cooperations between users to use the secret sharing or high computation and communication overhead in case of using homomorphic encryptions. Moreover, the schemes does binary data addition, but individual bits aggregation is used in our proposed scheme, which is more proper for reporting location information because each location can be represented by one bit.

Various schemes have been proposed to secure and preserve privacy in different wireless networks and applications, such as [1, 5, 11–17, 19, 21], but these schemes cannot be used to solve the problems addressed in this paper because they are designed for different network and threat models.

6. CONCLUSIONS

In this paper, we proposed a privacy-preserving scheme for AC service management. Data aggregation approach is used to preserve location privacy by providing the AC company with the total number of requests in each small geographic area without accessing the passengers' individual routes. Unlike the existing aggregation schemes that do binary data addition, the proposed scheme does individual bits addition, which is more appropriate for AC service management application. Our analysis have demonstrated that the proposed scheme can preserve privacy. Real experiments are conducted to measure the overhead of the proposed scheme. The given results can demonstrate that our scheme is efficient even in case of large cities that are represented by many geographic cells.

7. ACKNOWLEDGMENTS

This publication was made possible by NPRP grant number 9-055-2-022 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

8. REFERENCES

[1] A. Alsharif, S. Tonyali, M. Mahmoud, K. Akkaya, M. Ismail, and E. Serpedin. Performance Analysis of Certificate Renewal Scheme for AMI Networks. *Proc. of the 7th International Workshop on Computer Science and Engineering, Beijing, China*, pages 25–27, June 2017.

[2] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou. Privacy-preserving multi-keyword ranked search over encrypted cloud data. *INFOCOM, 2011 Proceedings IEEE, Shanghai*, pages 139–152, 2011.

[3] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou. Privacy-preserving multi-keyword ranked search over encrypted cloud data. *IEEE Transactions on Parallel and Distributed Systems*, 25(1):222–233, January 2014.

[4] F. D. Garcia and B. Jacobs. Privacy-friendly energy-metering via homomorphic encryption. *Proceedings of Security and Trust Management: 6th International Workshop, STM 2010, Greece, Athens*, pages 226–238, September 2011.

[5] S. Gunukula, A. Sherif, M. Pazos-Revilla, B. Ausby, M. Mahmoud, and X. Shen. Efficient Scheme for Secure and Privacy-Preserving Electric Vehicle Dynamic Charging System. *Proc. of IEEE ICCŠ17, Paris, France*, May 2017.

[6] T. Lassa. *The Beginning of the End of Driving*, 2014 [Online; accessed 10-October-2017].

[7] H. Li, D. Liu, Y. Dai, and T. Luan. Engineering searchable encryption of mobile cloud networks: when

QoE meets QoP. *IEEE Wireless Communications*, 22:74–80, 2015.

[8] Q. Li and G. Cao. Efficient and privacy-preserving data aggregation in mobile sensing. *Proceedings of 20th IEEE International Conference on Network Protocols (ICNP)*, pages 1–10, October 2012.

[9] P. Lienert. *Waymo's self-driving cars improve performance in California tests*, 2017 [Online; accessed 10-October-2017].

[10] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen. EPPA: An efficient and privacy-preserving aggregation scheme for secure smart grid communications. *IEEE Transactions on Parallel and Distributed Systems*, 23(9):1621–1631, September 2012.

[11] M. Mahmoud, K. Rabieh, A. Sherif, E. Oriero, M. Ismail, E. Serpedin, and K. Qaraqe.

[12] M. Mahmoud, N. Saputro, P. Akula, and K. akkaya. Privacy-preserving power injection over a hybrid ami/lte smart grid network. *IEEE Internet of Things Journal*, 2016.

[13] H. Mohammed, S. Tonyali, K. Rabieh, M. Mahmoud, and K. Akkaya. Efficient Privacy-Preserving Data Collection Scheme for Smart Grid AMI Networks. *Proc. of IEEE Globecom, Washington DC, USA*, December 2016.

[14] M. Pazos-Revilla, A. Alsharif, S. Gunukula, N. Guo, M. Mohamed, and X. Shen. Secure and privacy-preserving physical-layer-assisted scheme for ev dynamic charging system. *IEEE Transactions on Vehicular Technology*, In press 2017.

[15] K. Rabieh, M. Mahmoud, K. Akkaya, and S. Tonyali. Scalable Certificate Revocation Schemes for Smart Grid AMI Networks Using Bloom Filters. *IEEE Transactions on Dependence and Secure Computing (TDSC)*, 14:420–432, July 2017.

[16] K. Rabieh, M. Mahmoud, and M. Younis. Privacy-Preserving Route Reporting Schemes for Traffic Management Systems. *IEEE Transactions on Vehicular Technology (TVT)*, 66:2703–2713, March 2017.

[17] A. Sherif, A. Alsharif, J. Moran, and M. Mahmoud. Privacy-Preserving Ride Sharing Organization Scheme for Autonomous Vehicles in Large Cities. *Proc. of IEEE 86th Vehicular Technology Conference (VTC2017-Fall), Toronto, Canada*, September 2017.

[18] A. Sherif, K. Rabieh, M. Mahmoud, and X. Liang. Privacy-preserving ride sharing scheme for autonomous vehicles in big data era. *IEEE Internet of Things Journal, to appear*, 2016.

[19] A. Sherif, K. Rabieh, M. Mahmoud, and X. Liang. Privacy-preserving ride sharing scheme for autonomous vehicles in big data era. *IEEE Internet of Things Journal*, 4:611–618, April 2017.

[20] W. Wong, D. Cheung, B. Kao, and N. Mamoulis. Secure kNN computation on encrypted databases. *Proc. of the ACM SIGMOD International Conference on Management of Data, New York, USA*, pages 139–152, 2009.

[21] Z. Haddad and A. Alsharif and A. Sherif and M. Mahmoud. Privacy-Preserving Intra-MME Group Handover Via MRN in LTE-A Networks for Repeated Trips. *Proc. of IEEE 86th Vehicular Technology Conference (VTC2017-Fall), Toronto, Canada*, September 2017.