to appear in IEEE Vehicular Technology Conference (VTC-Fall) 2018

Privacy Preserving Distributed Stable Matching of Electric Vehicles and Charge Suppliers

Fatih Yucel*, Eyuphan Bulut*, and Kemal Akkaya†
*Dept. of Comp. Science, Virginia Commonwealth University, Richmond, VA 23284
†Dept. of Elec. and Comp. Engineering, Florida International University, Miami, FL 33174
Email: {yucelf,ebulut}@vcu.edu, kakkaya@fiu.edu

***Abstract*—The potential of electric vehicles (EV) to reduce foreign-oil dependence and improve urban air quality has triggered lots of investment by automotive companies recently and mass penetration and market dominance of EVs is imminent. However, EVs need to be charged more frequently than fossil-based vehicles and the charging durations are much longer. This necessitates in advance scheduling and matching depending on the route of the EVs. However, such scheduling and frequent charging may leak sensitive information about the users which may expose their driving patterns, whereabouts, schedules, etc. The situation is compounded with the proliferation of EV chargers such as V2V charging where there can be a lot of privacy exposure if matching of suppliers and EVs is achieved in a centralized manner. To address this issue, in this paper, we propose a privacy-preserving distributed stable matching of EVs with suppliers (i.e., public/private stations, V2V chargers) using preference lists formed by partially homomorphic encryption-based distance calculations while hiding the locations. The simulation results indicate that such a local matching of supplier and demanders can be achieved in a distributed fashion within reasonable computation and convergence times while preserving privacy of users.**

***Index Terms*—Electric vehicle charging, scheduling, privacy, Pallier homomorphic encryption, distributed stable matching, vehicular network.**

I. Introduction

Electric vehicles (EVs) have received increasing attention recently as they have the potential to provide sustainable and eco-friendly transportation systems. They can also act as energy storage systems [1] during power outages (e.g. Vehicle-to-Home) or to support renewable energy systems. Due to such potential, recently many auto companies have launched their products of many kinds of EVs, thus, a mass penetration and market dominance of EVs is expected in the upcoming years. For instance, according to a study in [2], 15 million EVs are expected to be on the roads by 2030.

Despite a disruptive increase in number of EVs is imminent, current charging infrastructure is not sufficient. Thus, there is an ongoing effort to expand the charging options for the users. Recently, several different companies have built their own charging networks (e.g., EvGo [3]). They offer charging service to EV drivers through their membership programs. They coordinate access to charging stations owned by them and provide maintenance services to keep charging stations running. While each charging network website provides the map of their own charging stations, there exist web sites (such as PlugShare [4]) that provide a complete view of all charging stations from different charging networks as well as the residential stations in an area on the map. This helps EV drivers locate available charging stations, and monitor their availability. The drivers can also check in when they charge at that station, share tips, comments, and photos, and provide snapshots of their charging experiences [4].

In order to provide more options for charging, there are also EV owners who open their residential charging stations to other EV owners and share through the charging network web sites. Similarly, Vehicle-to-Vehicle (V2V) charge sharing based solutions [5]–[9] are proposed recently to encourage EV owners with excessive charge share their charge with other EV owners in need. There are V2V charging products (e.g., Orca Inceptive [10] by Andromeda Power) in market today which are used by EV owners for charge sharing.

All these efforts for expanding the charging options are to address the frequent and long-period charging needs of EVs as opposed to fossil-driven vehicles. Specifically, in-advance scheduling of charging is needed to minimize the waiting times and thus increase the travel efficiency and driver comfort for the EV users. Obviously, this scheduling needs to consider the route of the EVs, the availability of charge suppliers (i.e., public/private charging stations, V2V chargers) and EV owners. This means scheduling may cause to leak some private information about the users during this process. With a long-term analysis of schedule and charging information (time, location) user's driving patterns and whereabouts may be exposed. For instance, for a driver charging his/her EV at two charging stations regularly (e.g., every day), it is reasonable to speculate that these two places are around driver's home and work. This could further be used by an adversary to trace the driver and commit crimes like breaking into driver's home when the driver is not at home. Similarly, marketers can send driver ads that are designed based on the habitual needs of the drivers. Such privacy threats may later hinder the successful large-scale penetration of EVs in the market as users see privacy as an important human right when using technology [11]. Thus, new EV charging approaches that hide or limit the aforementioned location and charging information are needed to ensure that this new technology will not be misused to violate users' privacy.

While a number of approaches have been proposed recently to address privacy issues in EV charging [12]–[16], they are geared mostly for charging on the power grid and within a

single charging provider. However, as the number of EVs increases and different options (e.g., mobile V2V and residential) for charge suppliers emerge, there is a need for many-to-many optimal matching for efficient resource utilization in the network. While some recent works [17]–[19] study this matching problem, they do not provide solutions for privacy-preserving matching of requesters and suppliers. However, both the requesters and some suppliers (e.g., V2V charge supplier EV, residential supplier) may not want to share their location information with the server in order not to expose their living patterns. In this paper, we address this issue and present a privacy-preserving matching of charge requester EVs with all kinds of charge suppliers.

In order to avoid the potential privacy and security pitfalls of centralized matching at a server, we propose to use distributed stable matching that utilizes the preference lists of users that are formed without having access to location information of suppliers. This is achieved by sharing a user's encrypted location and performing homomorphic computations at the supplier side. Specifically, we rely on a partially homomorphic scheme, namely Pallier, to be able to perform distance computations. The simulation results indicate that such a local matching of supplier and demanders can be achieved in a distributed fashion within reasonable computation and convergence times while preserving privacy of users.

The rest of the paper is organized as follows. In Section II, we present an overview of the proposed system. In Section III, we discuss the details of the proposed solution. In Section IV, we present our evaluation of the proposed solution. Finally, we end up with conclusion in Section V.

II. System Overview

We assume a system model shown in Fig. 1 with two sets of user groups: (i) EV owners requesting for charge, and (ii) charge suppliers (i.e., public/private charging stations, residential stations and V2V chargers). Note that there is no centralized scheduler (i.e., server) assumed in the system. We assume that requester EVs initiate a local query using a local communication technology (e.g., DSRC, LTE-direct [20]) to check if there is available suppliers[1] in their vicinity. The suppliers will collect these requests, and reply back within a reasonable decision time frame to be matched with the requester EVs based on their needs in a distributed manner. We assume that distributed stable matching will be used for the matching of suppliers and demanders. During this process, neither demanders nor suppliers will know the actual locations of each other until they are matched. This can be achieved via encryption. However, they need to know the distances for decision making and thus each user will be able to calculate the distance to the others interacted and form a preference list in the ascending order of distances. This will be achieved by performing computations on the encrypted location information. Once a demander EV is matched with a supplier, it will then learn the actual location to drive the supplier's location.

[1]Not only the V2V suppliers can be located but also the charging stations or other residential stations could be found once they are equipped with On-board-units (OBU).

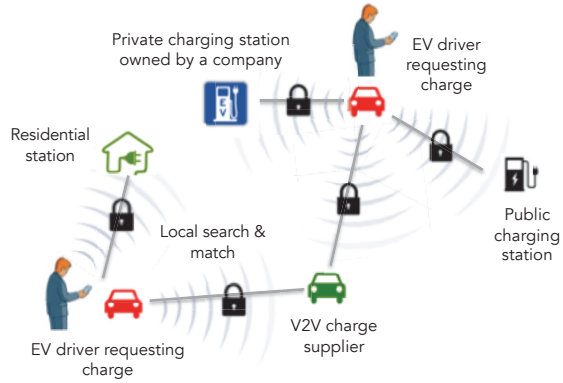

Fig. 1: Overview of the local search and match system

III. Proposed Solution

A. Stable Matching

Stable Matching problem is the problem of matching users at different sides of a bipartite graph. It is also known as the stable marriage problem in which a group of men and women with equal sizes are matched each other based on their preferences. The goal is to find a matching $\mathcal{M}$ in which each man and woman are assigned a partner and every one is satisfied with their partners. Here, each individual becomes satisfactory, so the matching is stable, if they are assigned to somebody and there does not exist a blocking pair (m, w) such that m prefers w to his current partner, and w prefers m to her current partner. This problem is initially introduced by Gale and Shapley [21] in an economic context (e.g. market matching), however it has been applied to several other domains including node deployment in wireless sensor networks [22].

It has been shown that a stable marriage always exists if the both set sizes are equal and can be found with a centralized algorithm in $\mathcal{O}(N^2)$ time [23]. Moreover, this algorithm can naturally be implemented as a distributed one [24] for which it is proved to be communication optimal [25]. A relaxed version of stable matching problem is obtained with incomplete preference lists. That is, each user on one side of the bipartite graph may consider some of the users on the other side as unacceptable, thus does not have them in its preference list. Gale Shapley algorithm has been shown to work even for this version of stable matching problem with incomplete lists (SMI). Another relaxation is when the set of users in both sides have unequal number of users. For example, when there are m men and $w < m$ women, there has to be some men not matched with a woman. However, as it has been shown in [26], with a proper stopping condition, a similar algorithm can yield a stable matching if exists.

In order to run this algorithm in a privacy preserving manner, the preference lists of users should be hidden from

Algorithm 1: Privacy Preserving Distance Calculation

1 The app generates encryption and decryption key pair of Paillier's cryptosystem: $E_{key} = (n, g)$, $D_{key} = (\lambda, \mu)$.

2 EV_R generates the following ciphertexts and broadcasts to the suppliers in the vicinity in $\mathcal{S}$.

$$\mathbb{E}(2x_i),\ \mathbb{E}(x_i^2),\ \mathbb{E}(2y_i),\ \mathbb{E}(y_i^2)$$

3 After receiving this request and associated ciphertexts, every supplier EV_S, first generates the following ciphertexts:

$$\mathbb{E}(x_j^2),\ \mathbb{E}(y_j^2)$$

4 Then, EV_S executes the following homomorphic operations and sends it back to the EV_R:

$$\mathbb{E}(2x_i)^{-x_j} = \mathbb{E}(-2x_ix_j),$$
$$\mathbb{E}(2y_i)^{-y_j} = \mathbb{E}(-2y_iy_j),$$
$$\mathbb{E}(-2x_ix_j).\mathbb{E}(x_i^2).\mathbb{E}(x_j^2) = \mathbb{E}((x_i - x_j)^2)$$
$$\mathbb{E}(-2y_iy_j).\mathbb{E}(y_i^2).\mathbb{E}(y_j^2) = \mathbb{E}((y_i - y_j)^2)$$
$$\mathbb{E}((x_i - x_j)^2).\mathbb{E}((y_i - y_j)^2) = \mathbb{E}([\mathbf{dist(i,j)}]^2)$$

5 EV_R, after receiving the ciphertext, decrypts it and computes the actual distance to the supplier, EV_S.

$$\text{dist(i,j)} = \sqrt{\mathbb{D}(\mathbb{E}([dist(i,j)]^2))}$$

others and the preference lists should be formed without knowing unnecessary information from others. For the former, running the algorithm in a distributed manner (rather than in a centralized server) will hide preference lists of users from the external entities. For the latter, we assume each demander will form its preference lists of suppliers in the ascending order of distances to them (as they may naturally prefer the closest suppliers). However, we propose that these preference lists could be obtained without knowing the actual location information of suppliers using Paillier cryptosystem [27] based homomorphic operations between the requester and supplier. We also assume that the suppliers will form a preference list of demanders in the ascending order of their distances (as they may naturally want to service closest ones first).

For PHE operations, we assume that each EV owner will get a separate pair of PHE public-private keys when the app is setup. Suppliers will know the PHE public key of users, but not the PHE private key and thus, they will not be able to decrypt the raw location information. But they will perform computations on ciphertexts resulting decryptable proper information for demanders using homomorphic properties.

B. Formation of Preference Lists

Let's denote ciphertext generated by the Pailier cryptosytem for m with $\mathbb{E}(m)$. The encrypted squared distance computation between a requester i at location $\text{loc}_i = (x_i, y_i)$ and a supplier j at location $\text{loc}_j = (x_j, y_j)$ could be achieved by:

$$\begin{aligned} dist(i,j) &= |loc_i - loc_j| = (x_i - x_j)^2 + (y_i - y_j)^2 \\ \mathbb{E}(dist(i,j)) &= \mathbb{E}(x_i^2 - 2x_ix_j + x_j^2 + y_i^2 - 2y_iy_j + y_j^2) \\ &= \mathbb{E}(x_i^2).\left(\mathbb{E}(x_i)\right)^{-2x_j}.\mathbb{E}(x_j^2).\mathbb{E}(y_i^2). \\ &\quad \left(\mathbb{E}(y_i)\right)^{-2y_j}.\mathbb{E}(y_j^2) \end{aligned} \tag{1}$$

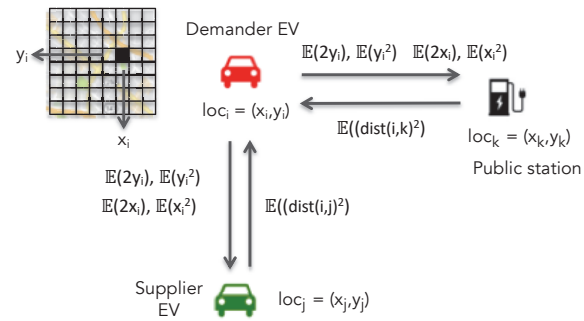

Fig. 2: The communication between the demanders and suppliers to form preference lists without knowing actual location.

If the supplier can get the encrypted values for both coordinates ($\mathbb{E}(x_i)$, $\mathbb{E}(y_i)$) and their squares ($\mathbb{E}(x_i^2)$, $\mathbb{E}(y_i^2)$) from the requester i, it can calculate $\mathbb{E}(dist(i,j))$ and send back to the requester without knowing requester's location and without releasing its location to the requester. Similarly, supplier can learn its distance to the requester using its own key pairs. This overall procedure with all communication and computation requirements is summarized in Algorithm 1.

When a requester EV needs to be charged, it sends a broadcast message to the suppliers in the vicinity with its encrypted location information (i.e., $\mathbb{E}(2x_i)$, $\mathbb{E}(x_i^2)$, $\mathbb{E}(2y_i)$, $\mathbb{E}(y_i^2)$). The suppliers that receive this request then perform necessary homomorphic operations (using requester's PHE public key and their location information) on these ciphertexts without knowing the actual requester location information. Once the encrypted distance information is obtained, it is sent back to the requester (it is a broadcast but only the requester can decrypt it). The requester then decrypts it and takes[2] the square root to obtain the actual distance. The requester waits for a predefined time and collects all the supplier information in the vicinity. Then, it forms a preference list of suppliers in the ascending order of distances (i.e., travel time). This process is also illustrated in Fig. 2.

Note that in the assumed system model, there are different supplier options including public/private stations, residential stations and V2V suppliers. Since the locations of public/private stations will be fixed and known to public, it may not be considered as privacy leakage. For residential charging stations, even though the location will be fixed, residents may still want to keep the location information private from the other requesters until they are assigned in the final matching. Similarly, in the case of V2V charger suppliers, the location of EVs can change during the day thus a location update has to be provided to the requesters each time a matching will be done. Moreover, such V2V charge suppliers may not want to release the location information to the requester EVs as it might pose daily moving patterns of drivers and can be considered as privacy leakage.

[2]Not necessary as the same list can be formed with squared distances.

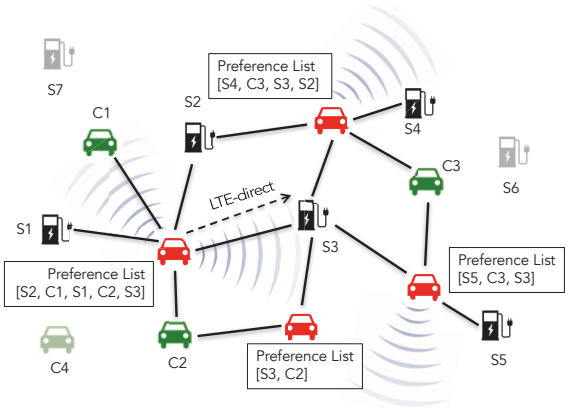

Fig. 3: Formation of incomplete preference lists through local communication.

C. Privacy Preserving Distributed Stable Matching

Once the demander EVs and suppliers form their preference lists, the matching process starts[3]. Fig. 3 shows an example scenario, where four different EVs request for charge from the stations and V2V chargers in their vicinity within the same decision time frame. Note that both the demander EVs and suppliers will have a partial view of the network. This will result in incomplete preference lists for the users. That is, some suppliers will not appear in the list of some demanders. Similarly, some demanders will not appear in the list of some suppliers. Additionally, the number of demander EVs and the supplier EVs currently available in the network could change during the day. While these bring extra challenges to the matching, thanks to the distributed computable nature of stable matching even with incomplete lists [24], the users can communicate to each other with necessary messages without releasing their preference orders and come to a negotiation on the final matching. For the unequal sets of demander and suppliers, the algorithm also stops naturally with a specific condition giving a stable matching [26] for those who are matched. If there are fewer demanders than suppliers (i.e., $d < s$), the algorithm stops when d of the suppliers have been proposed to, and if there are more demanders than suppliers, the algorithm stops when each demander is either being suspended by a supplier or being rejected by all suppliers.

Algorithm 2: DemanderEV()

```
d ← free
continue ← true
while continue do
    if d = free & list(d) ≠ ∅ then
        s ← top(list(d))
        sendMessage(propose, d, s)
        d ← s
    end
    msg ← getMessage()
    switch msg.type do
        case accept
            nothing
        end
        case delete
            list(d) ← list(d) - msg.sender
            if msg.sender = s then
                d ← free
            end
        end
        case stop
            continue ← false
        end
    endsw
end
```

Algorithm 3: Supplier()

```
s ← free
continue ← true
while continue do
    msg ← getMessage()
    switch msg.type do
        case propose
            d ← msg.sender
            if d ∉ list(s) then
                sendMessage(delete, s, d)
            else
                sendMessage(accept, s, d)
                s ← d
                for each p after m in list(w) do
                    sendMessage(delete, s, p)
                    list(w) ← list(w) - p
                end
            end
        end
        case stop
            end ← true
        end
    endsw
end
```

Algorithm 2 and 3 show the procedures run by demanders and suppliers, respectively. Each demander offers to their first preference of suppliers in their list (with a *propose* message). If that supplier has the demander in its list, it accepts to provide service to this demander (and sends an *accept* message) and deletes all other demanders that come after this accepted one in its list (and let them know via a *delete* message). If the demander gets the *accept* message, it does nothing. However, the supplier may reject the demander if it is not in supplier's list (meaning not a better option), thus sends a *delete* message to the demander. In that case, the demander, once notified with rejection, becomes free again and proposes

[3]The synchronization of the matching process by all users involved could be achieved through a predetermined schedule (e.g., collect between 0-15 sec, run matching between 15-30 sec) by all parties.

to the next supplier in its list. This process continues until a stable condition is reached, if exists. This algorithm guarantees privacy in preferences and in the final assignment [24]. That is, each requester only knows the assigned supplier, and no more information.

IV. Simulation Results

A. Experiment Setup and Metrics

In this section, we present several simulation results regarding the performance of the proposed matching algorithm. We have generated a network topology of 100 demanders and 100 suppliers in a region of size 1km by 1km. The location of the demander and suppliers are assigned with uniform distribution. Then, by changing the range, R, of the local communication technology used, we obtain different sizes for the preference lists of users. In Table I, the corresponding average list size for different R values is shown. When R=1500, all demanders can see all other suppliers and vice versa. Thus, the lists for demanders consist of all suppliers and the lists for suppliers consist of all demanders. Here, note that the lists of neighbor users on the graph will have overlapping users, thus these lists will not be independent. Such a dependence, however, will affect the convergence and the messaging overhead which are the considered metrics in the experiments. While convergence refers to the duration of time for the algorithm to be stabilized, message overhead is the number of messages exchanged between demanders and suppliers.

TABLE I: Average preference list sizes for different R

Range (R) - meter	100	250	500	750	1000	1500
Avg preference list size	2.8	15.5	48.4	80.3	97.5	100

For the PHE calculations, in general, we use 512-bit primes for p and q defined in Paillier cryptosystem. However, we also test the impact of different key sizes later. For the simulations, we use a computer with Intel core i7 processor with speed 2.5 GHz and a 16GB of memory. For every result in this section, we took the average of 100 different runs for statistical significance.

B. Performance Results

We first look at the messaging overhead in the stable matching process. Fig. 4 shows the number of messages of each type exchanged between the demanders and suppliers. As expected, the number of delete messages is higher than accept and propose messages, thus a secondary axis is used for them. The results clearly show that the number of delete messages grows linearly as the preference list size increases for the users. This is because after the first acceptance, the user sends a delete message to all others that come after the one accepted in its list. On the other hand, the number of propose and accept messages have some saturation after the list sizes become more than 20. This is mainly due to the fact that every user finds a stable matching after proposing a few users in their lists. Also, as expected, the number of propose messages is higher than accept message counts.

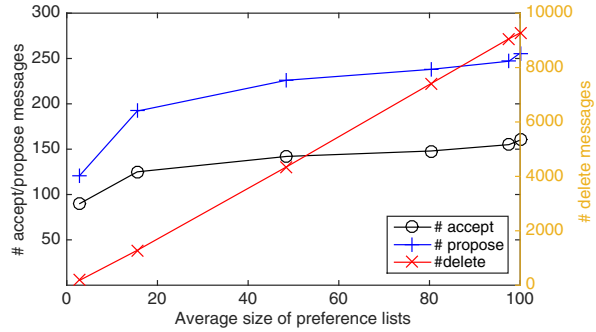

Fig. 4: The number of all messages of each type until the distributed stable process converges.

Next, we look at the duration needed for both phases of the algorithm. In the first phase, through communication with each of the nearby users (one way of communication is assumed to take 100 ms), encrypted distance values are received and decrypted. Once all of them are collected, a sorting algorithm is run to have the preference lists in the ascending order of distances. The dominating factor in this phase is the computation of Paillier operations. In the second phase, due to the multiple messages exchanged between the users throughout the distributed stable matching process, the dominating factor becomes the communication cost. In Fig. 5, the duration of these two phases are shown for different list sizes. The results show that even with complete lists, first phase takes around 1.1 sec, and the second one takes around 2.6 sec. Note that in non-privacy preserving version of the distributed stable matching, there will not be the delay due to the phase 1 encryptions but phase 2 duration will be the same. Thus, the proposed privacy preservation method brings around 25% delay overhead to the matching.

In the proposed distributed matching, since each user has a partial view of the graph and defines the preference lists accordingly, it is possible that after the matching process converges, such incomplete lists may yield some of the nodes not matched. Fig. 6 shows the average coverage obtained in the matching process, where the coverage refers to the % of demanders and suppliers matched out of all users. For example, when list size is around 48, 95% of users (meaning 95% of demanders and 95% of suppliers as we use equal number for both sets) are matched in the current process. With average list size of around 15, this ratio goes down to 88%, which is still reasonable. While this will cause some users not matched in the current round, such users will most likely be matched by the end of the next round with 98.5% probability (i.e., 1-(1-0.88)^2). Moreover, with a reasonably short duration for each round (as shown in Fig. 5), matching in the consecutive rounds can still be satisfactory for the users.

V. Conclusion

In this paper, we study the privacy preserving matching of EVs that are in need of charge with suppliers. In the

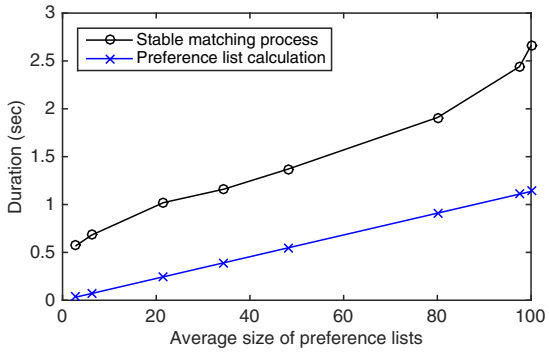

Fig. 5: Total duration of calculating preference lists (includes Paillier computations) and total convergence duration of distributed stable matching algorithm.

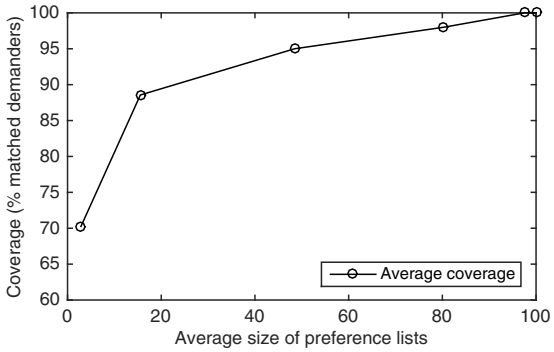

Fig. 6: Average coverage ratio of matching (% of demanders/suppliers matched out of all).

proposed system, demander EVs identify the potential suppliers in the vicinity through a local search with a peer-to-peer (P2P) communication technology such as LTE-direct or DSRC. Then, using a homomorphic encryption-based distance calculation, each demander (supplier) calculates the preference lists of the suppliers (demanders) in the increasing order of their distances. Finally, using a distributed stable matching algorithm with these preference lists, a matching is obtained such that every demander and supplier is satisfied with their assignments. Such a distributed querying and matching system in general avoids the potential privacy and security pitfalls of centralized matching at a server. The matching results satisfy all the users at the same time, thus promotes participation. Moreover, all this process is achieved without releasing the location information of users and their preference lists to one another. The simulation results show that this privacy preserving matching process can converge in a reasonable time and the computation overheads for Paillier based calculations do not affect the convergence delay profoundly as long as appropriate key sizes are selected.

References

[1] T. Markel, M. Kuss, and P. Denholm, "Communication and control of electric drive vehicles supporting renewables," in *Proc. of IEEE Vehicle Power and Propulsion Conference, VPPC'09*, 2009, pp. 27–34.

[2] E. W. Wood, C. L. Rames, M. Muratori, S. Srinivasa Raghavan, and M. W. Melaina, "National plug-in electric vehicle infrastructure analysis," National Renewable Energy Laboratory (NREL), Golden, CO (United States), Tech. Rep., 2017.

[3] EvGo, 2017. [Online]. Available: http://www.evgonetwork.com/

[4] PlugShare, 2017. [Online]. Available: https://www.plugshare.com/

[5] R. Alvaro-Hermana, J. Fraile-Ardanuy, P. J. Zufiria, L. Knapen, and D. Janssens, "Peer to peer energy trading with electric vehicles," *IEEE Intelligent Trans. Systems Magazine*, vol. 8, no. 3, pp. 33–44, 2016.

[6] J. Kang, R. Yu, X. Huang, S. Maharjan, Y. Zhang, and E. Hossain, "Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains," *IEEE Transactions on Industrial Informatics*, 2017.

[7] M. Wang, M. Ismail, R. Zhang, X. Shen, E. Serpedin, and K. Qaraqe, "Spatio-temporal coordinated v2v fast charging strategy for mobile gevs via price control," *IEEE Transactions on Smart Grid*, 2016.

[8] "AAA unveils north americas first roadside assistance truck capable of charging electric vehicles," Jul. 2011. [Online]. Available: http://newsroom.aaa.com/2011/07/ev-charging-statio/

[9] B. Roberts, K. Akkaya, E. Bulut, and M. Kisacikoglu, "An authentication framework for electric vehicle-to-electric vehicle charging applications," in *MASS REU Research in Networking and Systems Workshop*. IEEE, 2017.

[10] A. P. introduces portable DC fast charger, "Charles morris," 2013. [Online]. Available: https://chargedevs.com/newswire/andromeda-power-introduces-portable-dc-fast-charger/

[11] J. Freudiger, R. Shokri, and J.-P. Hubaux, "Evaluating the privacy risk of location-based services." in *Financial Cryptography*, vol. 7035. Springer, 2011, pp. 31–46.

[12] W. Han and Y. Xiao, "Privacy preservation for v2g networks in smart grid: A survey," *Computer Communications*, vol. 91, pp. 17–28, 2016.

[13] Z. Yang, S. Yu, W. Lou, and C. Liu, "Privacy-preserving communication and precise reward architecture for v2g networks in smart grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 697–706, 2011.

[14] Y. Cao, N. Wang, G. Kamel, and Y.-J. Kim, "An electric vehicle charging management scheme based on publish/subscribe communication framework," *IEEE Systems Journal*, 2015.

[15] S. Han, U. Topcu, and G. J. Pappas, "Differentially private distributed protocol for electric vehicle charging," in *Communication, Control, and Computing, 52nd Annual Allerton Conf. on*. IEEE, 2014, pp. 242–249.

[16] J. K. Liu, W. Susilo, T. H. Yuen, M. H. Au, J. Fang, Z. L. Jiang, and J. Zhou, "Efficient privacy-preserving charging station reservation system for electric vehicles," *The Computer Journal*, vol. 59, no. 7, pp. 1040–1053, 2016.

[17] R. Zhang, X. Cheng, and L. Yang, "Flexible energy management protocol for cooperative ev-to-ev charging," in *Global Communications Conference (GLOBECOM), 2016 IEEE*. IEEE, 2016, pp. 1–6.

[18] R. Zhang, X. Cheng, and L. Yang, "Stable matching based cooperative v2v charging mechanism for electric vehicles," in *in Proceedings of Vehicular Technology Conference (VTC Fall)*. IEEE, 2017, pp. 1–6.

[19] E. Bulut and M. Kisacikoglu, "Mitigating range anxiety via vehicle-to-vehicle social charging system," in *Proceedings of Vehicular Technology Conference (VTC Spring), IEEE*, 2017.

[20] 3GPP, "Mobile broadband standard," v12, 2015. [Online]. Available: http://www.3gpp.org/specifications/releases/68-release-12

[21] D. Gale and L. Shapley, "College admissions and stability of marriage. american mathematicas monthly, 69, 9-15," 1962.

[22] B. McLaughlan and K. Akkaya, "Coverage-based clustering of wireless sensor and actor networks," in *Pervasive Services, IEEE International Conference on*. IEEE, 2007, pp. 45–54.

[23] C. Ng and D. S. Hirschberg, "Lower bounds for the stable marriage problem and its variants," *SIAM Journal on Computing*, vol. 19, no. 1, pp. 71–77, 1990.

[24] I. Brito and P. Meseguer, "Distributed stable matching problems," in *CP*. Springer, 2005, pp. 152–166.

[25] Y. A. Gonczarowski, N. Nisan, R. Ostrovsky, and W. Rosenbaum, "A stable marriage requires communication," in *Proc. of the 26th ACM-SIAM symposium on Discrete algorithms*, 2015, pp. 1003–1017.

[26] D. McVitie and L. B. Wilson, "Stable marriage assignment for unequal sets," *BIT Numerical Mathematics*, vol. 10, no. 3, pp. 295–309, 1970.

[27] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 1999, pp. 223–238.