IET Research Journals

IET Journals
The Institution of Engineering and Technology

Private Routing and Ride-Sharing Using Homomorphic Encryption

ISSN 1751-8644
doi: 0000000000
www.ietdl.org

Farhad Farokhi[1,2,*] *Iman Shames*[2] *Karl H. Johansson*[3]

[1] *CSIRO's Data61, Docklands, Australia*
[2] *Department of Electrical and Electronic Engineering at the University of Melbourne, Parkville, Australia*
[3] *School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden*
* *E-mail: farhad.farakhi@unimelb.edu.au*

Abstract: A framework for private and secure communication and interaction between agents interacting in transportation services is developed. An agent, i.e., a user, can ask questions or submit queries regarding whether the other agents, i.e., drivers, use a desired road at specific times of the day in an encrypted fashion. We develop the framework using semi-homomorphic encryption (namely, the Paillier's encryption method) to enables algebraic manipulation of plain data without the need for decryption using appropriate computations over the encrypted data. Strong privacy and security guarantees are proved for the agents. Subsequently, the semi-homomorphic encryption method is utilized to develop privacy-aware ride-sharing and routing algorithms without the need for disclosing the origin and destination of the user.

1 Introduction

1.1 Motivation

Advances in communication technology have created new opportunities in the context of the shared economy. An example of such advances is collaborative driving including ride-sharing and heavy-duty vehicle platooning to reduce fuel consumption, greenhouse emissions, and costs for commuters or truck fleets [1]. The rewards of these new technologies come at the cost of the erosion of privacy within society. For instance, ride-sharing applications can require and record detailed information from their customers and service providers. This information can be later used for targeted advertisements or be sold to a third party for profit. The situation can even get worse in commercial examples due to the competitive nature of the users. For instance, although heavy-duty vehicle platooning has shown promises in small-scale experiments, it has not yet been adopted widely. In addition to technological and legal barriers, this could be caused by the commercial nature of the fleet owners (that are often competing for the same clients) resulting in their unwillingness to share private data, e.g., routes and travel times of their vehicles, even if cooperation reduces their operative costs. This motivates the need for developing private and secure match-making services to facilitate effective coordination among competing companies, such as fleet owners, for widespread adoption of these new technologies. Such services are not limited to heavy-duty vehicle platooning or ride-sharing applications but can be justified in many other scenarios, such as collaborative logistics and energy markets (or, even completely unrelated to the topic of this paper, online dating services). Another example of a popular, yet privacy eroding, online service is routing based on real-time traffic estimates. This service requires the users to provide their origins and destinations to the server so that the application can find the fastest (or the most efficient in another appropriate measure) route. Therefore, new approaches for online routing that can deliver services with privacy guarantees are required.

1.2 Related Work

The problem of privacy in transportation systems has attracted much attention recently [2–13]. Some of those studies rely on anonymization and differential privacy for retaining privacy. Anonymization is proved to be insufficient for privacy in many transportation systems [14, 15]. Further, differential privacy may not be preferred for routing and ride-sharing as the corrupted data can lose its essential characteristics (e.g., the extracted shortest path might no longer be the shortest for the user). Other studies are involved with the traffic estimation using real-time location measurements of the participants and they propose strategies that can keep the identity of the users or the location of their private places, such as home or work addresses, hidden [12]. In this paper, in contrast to those studies, we investigate the problem regarding the use of real-time traffic data and not its generation.

The first part of the paper on ride-sharing shares ideas with private searching in streaming data [16–20]. In private searching in streaming data, we are interested in determining if certain important keywords have been utilized in private encrypted messages, e.g., messages or e-mails. If the keywords do not appear, the content of the messages itself is not of special interest. In private searching in streaming data, the privacy guarantees are often one-sided with guarantees provided to protect the privacy of data owners. In this paper, however, both sides (i.e., the questioning agents and the responding agent) require privacy guarantees.

The problem of creating location-based services with privacy guarantees has been studied in the past [21–30]. Those studies provide mechanisms for identifying places of interest (possibly of specific types) close to a location without revealing the exact address. Some of those studies rely on homomorphic encryption techniques, e.g., [26], while others rely on adding dummy locations, e.g., [25]. Those papers, however, do not investigate the problem of routing the vehicles following the shortest path from their current locations. Homomorphic encryption has been previously utilized to develop privacy-preserving ticketing and billing for transport [27, 31, 32] and for ride-sharing in [33–37]. Again, those studies do not address privacy-preserving routing.

Homomorphic encryption has been used to ensure security and privacy, especially against eavesdropping agents, within networked control and estimation [38–42]. Those studies, however, address the difficulties associated with the use of encryption in real-time feedback loop and do not focus on developing a framework for private coordination among multiple, possibly competing, agents. These studies also do not provide two-sided privacy guarantees, which is required for this paper as discussed above.

This paper is closely related to secure multi-party computation dedicated to developing methods for multiple agents to jointly evaluate a function for their inputs while keeping the inputs private (from each other). In addition to using homomorphic encryption for secure multi-party computation [43, 44], other approaches have been developed that rely on secret sharing or other forms of encryption.

A branch of such studies rely on Yao's protocol (originally introduced for secure two-party computation) [45]. The protocol provides a method for evaluating a Boolean circuit without any party being able to observe the bits that flow through the circuit during the evaluation. Yao's protocol has been proved to be secure [46] and can be efficiently implemented with a computational complexity that is linear in the number of inputs [47]. In private routing, when dealing with procedures that are not Boolean, the efficiency of Yao's protocol is limited. This is because the function needs to be rewritten in a Boolean form (which is, of course, possible when dealing with integer numbers). However, finding the most efficient representation of a function in the Boolean form for efficiently implementing Yao's protocol [48] is not trivial [49].

Finally, a preliminary version of this paper was presented as a conference paper in [52]. The conference paper only contained results on private ride-sharing and did not address the problem of private routing, which is a substantial part of the contributions of this paper.

1.3 Contributions

In this paper, at first, we develop a framework for secure and private communication between two agents. In the context of the heavy-duty vehicle platooning, which was presented in an early version of this paper [52], the agents are the fleet owners. However, in the case of ride-sharing, which is the topic of this paper, the agents can be the commuters and the road users. In this framework, an agent can submit an encrypted query or ask an encrypted question regarding whether the other agents use a particular road at a given time of the day. This is done in such a way that the other agents can provide their responses without knowing the content of the question or query.

We develop this framework using semi-homomorphic encryption, particularly, the Paillier's encryption method [53]. This is because semi-homomorphic encryption allows algebraic manipulation on the plaintext, which is often required for responding to the encrypted query, without the need of decryption [54]. This category of encryption techniques makes it possible for the second agent (i.e., the one receiving the encrypted question) to respond by appropriate manipulations of the encrypted question. Here, Paillier's encryption method, as an example of semi-homomorphic encryption methodology, is utilized [53]; however, the idea of this paper can be developed based on many other homomorphic encryption methods; see, e.g., [54].

We prove strong privacy and security guarantees for the developed framework. First, the agents cannot extract any private information from the agent who submits the query. Furthermore, the amount of information that the questioning agent can extract from all the other agents is bounded. A deceitful agent, that is submitting an encrypted query or asking an encrypted question, can at most extract the answer to two questions regarding the activities of the other agents. This is often negligible in comparison with the number of possible questions. These privacy guarantees, however, come at the price of increasing the computational load of the users, due to the need for encryption. In the paper, this secure communication platform is subsequently generalized to distributed coordination mechanisms among many agents.

The semi-homomorphic encryption method is then used to develop a privacy-aware routing algorithm. The users submit queries to the server containing the current state of the traffic on the roads in a transportation network without disclosing the identity of the roads in question. This allows the users to locally run shortest path algorithms, such as Dijkstra's algorithm [55, pp. 595-600], to find the most desirable path between origins and destinations without revealing the locations to the server containing the traffic data. It can be shown that the server cannot identify the location of the origin and the destination of the user better than a random number generator can guess. This regained privacy comes at the price of increased computational complexity as the burden of reporting the weight of an edge becomes linearly dependent on the number of nodes in the transportation network (which can be staggeringly high). To overcome this problem, a method for trimming down computational complexity and investigating the trade-off between complexity and privacy is developed.

In summary, we make the following contributions in this paper:

- Developing a framework for communication between two agents for ride-sharing using homomorphic encryption with strong privacy and security guarantees;
- Generalizing the framework to distributed coordination among many agents;
- Using homomorphic encryption for developing privacy-aware routing algorithms;
- Proposing a method for trimming down the computational complexity of privacy-aware routing algorithms in order to investigate the trade-off between complexity and privacy.

1.4 Outline

The remainder of this paper is organized as follows. The Paillier's encryption is introduced, as an example of a semi-homomorphic encryption method, in Section 2. A framework for private ride-sharing is developed in Section 3. Subsequently, the problem of private routing is investigated in Section 4. Numerical examples are presented in Section 5. Finally, Section 6 concludes the paper.

2 Semi-Homomorphic Encryption

In this subsection, the Paillier's encryption method is briefly introduced [53]. The semantic security of the Paillier's encryption method follows from the Decisional Composite Residuosity Assumption [53]. This assumption requires that the problem of deciding whether there exists $y \in \mathbb{Z}_{N^2}$ with $x = y^N \bmod N$ for integers $N \in \mathbb{Z}$ and $x \in \mathbb{Z}_{N^2}$ is computationally hard; see [53, 54] for more information. Note that, for any $N \in \mathbb{N}$, $\mathbb{Z}_N$ is the set of integers modulo N.

Now, we can describe the Paillier's encryption method. We must first generate public and private keys. Select large prime numbers p and q such that $\gcd(pq, (p-1)(q-1)) = 1$, where $\gcd(a, b)$ refers to the greatest common divisor of a and b. This condition is satisfied with a high probability if the prime numbers are selected randomly and independently. The public key is $N = pq$. The public key is used for encryption and can be shared with all the parties that need to perform computations. The private key is (λ, μ) with $\lambda = \mathrm{lcm}(p-1, q-1)$ and $\mu = \lambda^{-1} \bmod N$, where $\mathrm{lcm}(a, b)$ is the least common multiple of a and b. The private key must only be available to the entity that decrypts the data. We can encrypt a plain message $t \in \mathbb{Z}_N$ by computing $E(t; r) = (N+1)^t r^N \bmod N^2$, where r is randomly selected with uniform probability from $\mathbb{Z}_N^* := \{x \in \mathbb{Z}_N | \gcd(x, N) = 1\}$. For decryption of $c \in \mathbb{Z}_{N^2}$, we must follow the mapping $D(c) = L(c^\lambda \bmod N^2)\mu \bmod N$, where $L(x) = (x-1)/N$. By construction, $D(E(t; r)) = t$ for all $r \in \mathbb{Z}_N^*$ and all $t \in \mathbb{Z}_N$, which points to the correctness of the Paillier's encryption method [53].

The Paillier's encryption is a semi-homomorphic encryption method. Therefore, without decrypting encrypted data, we can perform addition on the corresponding plain data. The Paillier's encryption also allows us to multiply an encrypted number with a plaintext without decryption. Note that multiplication of ciphertext by a plaintext can be done by successive summation of the ciphertext. The fact that we cannot multiply two encrypted numbers together implies that the Paillier's encryption is only semi-homomorphic. In the next two sections, we use the additive-homomorhic property as well as multiplication by plaintext to devise secure and private ride-sharing and routing algorithms. This is shown in the following proposition.

Proposition 2.1 ([53]). *The following identities hold:*

1. *For all* $r, r' \in \mathbb{Z}_N^*$ *and* $t, t' \in \mathbb{Z}_N$ *such that* $t + t' \in \mathbb{Z}_N$, $E(t; r)E(t'; r') \bmod N^2 = E(t + t'; rr')$;
2. *For all* $r \in \mathbb{Z}_N^*$ *and all* $t, t' \in \mathbb{Z}_N$ *such that* $tt' \in \mathbb{Z}_N$, $E(t; r)^{t'} \bmod N^2 = E(t't; r^{t'})$.

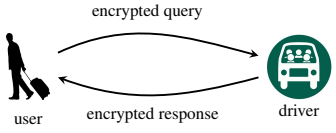

Fig. 1: Communication structure between a user and a driver in private ride-sharing. The user and the driver can only enquire about each other's interests and logistical constraints under strict privacy constraints. The user does not leak any private information and a deceitful user can at most extract information on the interests of the drivers about at most two pairs of roads and time windows.

Proposition 2.1 states that summation and multiplication of the plain data can be performed on the encrypted data as the computation $E(t;r)E(t';r') \bmod N^2$ on the encrypted data corresponds to addition $t+t'$ and $E(t;r)^{t'} \bmod N^2$ corresponds to multiplication tt'. In what follows, we use these properties to create a method for secure and private ride-sharing.

3 Private Ride-Sharing

In this section, we present the problem formulation and the results for ride-sharing (although the results can be readily used in other coordination contexts). This allows us to pose concrete questions and provide meaningful privacy and security guarantees.

Assume that driver $i \in \mathcal{F} := \{1,\dots,F\}$ travels over various roads on a transportation network in set $\mathcal{P}$ and at various time intervals of the day in set $\mathcal{T}$ (based on their individual preferences). It is assumed that $|\mathcal{P}| < \infty$ and $|\mathcal{T}| < \infty$. In this paper, we discretize the time of the day (e.g., in one-hour windows) in order to have a finite number of time windows. We use the set of integers $\mathcal{W} := \{1,\dots,|\mathcal{P}||\mathcal{T}|\}$ to capture all the possible combinations of roads and time windows (all combinations in which a driver might be interested in traveling). There is a one-to-one relationship between $\mathcal{W}$ and $\mathcal{P}\times\mathcal{T}$, i.e., $\mathcal{W}$ is isomorphic to $\mathcal{P}\times\mathcal{T}$.

We want to create a framework for secure and private communication among the users and drivers for the users to identify potential vehicles for ride-sharing. The matching is possible if the drivers and user exchange times and roads over which they travel. This exchange would, however, violate their privacy. Therefore, we aim to create a communication platform for the users and drivers to enquire about each other's other interests and logistical constraints under strict privacy constraints. It is shown that the enquiring agent does not leak any private information (i.e., the drivers cannot realize the road and the time window of interest of the enquiring user). In addition, even with the most sophisticated maneuvers, a deceitful enquiring user can only extract information on the interests of the drivers about at most two pairs of roads and time windows, which is negligible considering the sheer number of possibilities.

3.1 Secure and Private Communication Framework

In this section, the communication is restricted to two agents: a user (enquiring about the possibility of traveling on a specific time window and road) and a driver. Figure 1 illustrates the communication structure between the user and the driver in private ride-sharing. The user submits an encrypted query to the driver. The driver computes the response to the query (without knowing the query or the response) and provides the encrypted response to the user. The user then decrypts the response, which could point to the possibility of ride-sharing between them. This setup is subsequently generalized to develop a distributed coordination mechanism in the next subsection.

Assume that the user wants to know if the driver is traveling on the path and the time of the day associated with $w \in \mathcal{W}$ while it does not want the driver to know w explicitly, at least not before confirming that they can ride-share. The user can construct an encrypted vector

Algorithm 1 Procedure SUBMITQUERY for the user.

input: $w, \mathcal{W}$
output: x
procedure SUBMITQUERY(w,$\mathcal{W}$)
  # Computed by the user
  for $i \in \mathcal{W}$ **do**
    if $i = w$ **then**
      $x_i \leftarrow E(1;r_i)$
    else
      $x_i \leftarrow E(0;r_i)$
    end if
  end for
  return x
end procedure

Algorithm 2 Procedure RETURNRESPONSE for the driver.

input: $x, \overline{\mathcal{W}}, N$
output: y
procedure SUBMITQUERY(x,$\overline{\mathcal{W}}$,N)
  # Computed by the driver
  $y \leftarrow 1$
  for $i \in \overline{\mathcal{W}}$ **do**
    Select v_i randomly from $\{1,\dots,N-1\}$
    $y \leftarrow y(x_i^{v_i} \bmod N^2) \bmod N^2$
  end for
  return y
end procedure

$x \in \mathbb{Z}^{|\mathcal{W}|}$ such that the i-th element of x is

$$x_i = \begin{cases} E(1;r_i), & i = w, \\ E(0;r_i), & \text{otherwise.} \end{cases} \tag{1}$$

The random element r_i in the encryption $E(1;r_i)$ or $E(0;r_i)$ ensures that, with a high probability, ciphertexts of repeated 1 and 0 remain different[1]. The user transmits encrypted vector x to the driver. The driver then computes

$$y = \left(\prod_{j\in\overline{\mathcal{W}}} x_j^{v_j} \bmod N^2 \right) \bmod N^2,$$

where x_j are encryption of zeros and ones, defined in (1), v_j is randomly selected from $\{1,\dots,N-1\}$ with uniform probability, and $\overline{\mathcal{W}} \subseteq \mathcal{W}$ is the set of all paths and times over which the driver is traveling. The procedures for the user and driver are summarized in Algorithms 1 and 2. The following proposition proves that the decryption $D(y)$ indicates if the user and the driver can share a ride or not, in fact, pointing to the correctness of the proposed method.

Proposition 3.1. *If the user and the driver, respectively, use Algorithms* 1 *and* 2*, then* $D(y) \neq 0$ *if the user and the driver can ride share, i.e., if* $w \in \overline{\mathcal{W}}$*, and* $D(y) = 0$ *otherwise.*

Proof: The proof follows from the construction of the vector x and the application of Proposition 2.1. □

In this paper, we are interested in security and privacy from the perspective of eavesdropping. The driver can create unintelligible outputs by ignoring the received encrypted vector x; however, the detection and mitigation of false detection injection attacks are out

[1] *Assuming that both p and q are of the length of 1024 bits, the public key is of length 2048 bits. Therefore, the probability of selecting the same r twice even in a vector of millions of elements is smaller than* 10^{-1000}.

of our scope. In the remainder of this section, it is proved that eavesdropping attacks are computationally expensive (in fact, impossible by appropriate selection of the security parameters). We need to make the following definition for assessing the security and privacy of the proposed method from the perspective of the user. This definition is known in the encryption literature as semantic security.

Definition 3.1. *Let the driver propose $w_1, w_2 \in \mathcal{W}$. The user chooses at random w from $\{w_1, w_2\}$ with equal probability and sends x constructed using Algorithm* 1. *The driver can based on its knowledge of x (and w_1, w_2) guess w. This guess is denoted by w'. The driver's advantage*[1] *is given by* $\mathrm{Adv}(k) := |\mathbb{P}\{w = w'\} - 1/2|$, *where k denotes the security parameter, e.g.,* $\min(p, q)$ *in the Paillier's technique. The proposed strategy is defined to be private if* Adv *is negligible*[2].

Definition 3.1 states that the proposed strategy is private if the driver cannot guess preference w supplied by the user any better than a pure random number generator.

Proposition 3.2. *Under the Decisional Composite Residuosity Assumption, Algorithm* 1 *is private in the sense of Definition* 3.1.

Proof: The proof follows the semantic security of the Paillier's encryption method [53]. □

The driver's privacy and security guarantees are weaker because, by construction, if the user and the driver, respectively, use Algorithms 1 and 2, the user can successfully determine if the driver travels on the path and in the time window associated with w. This implies that, even in the best of situation, some private information from the driver is leaked. We show that the user can potentially extract more information by not following Algorithm 1. This points to a slightly deeper erosion of privacy. However, we prove that the information that the user can extract, under the sophisticated attacks, is extremely limited.

Assume that the user does not follow Algorithm 1 and constructs x such that $x_i = E(\tilde{x}_i; r_i)$, $\forall i$, for $\tilde{x}_i \in \mathbb{Z}_N$. We can prove the following proposition.

Proposition 3.3. *If $x_i = E(\tilde{x}_i; r_i)$, $\forall i$, for $\tilde{x}_i \in \mathbb{Z}_N$, then*

$$D(y) = \left(\sum_{w \in \mathcal{W}} \tilde{x}_i v_i z_i \right) \bmod N, \tag{2}$$

where $z_i = 1$ if $i \in \overline{\mathcal{W}}$ (the driver travels on the road and the time associated with $i \in \mathcal{W}$) and $z_i = 0$ otherwise.

Proof: The proof follows from the construction of the vector x, $\tilde{x}_i, \forall i$, and the application of Proposition 2.1. □

Proposition 3.3 states the user must solve $D(y) = (\sum_{w \in \mathcal{W}} \tilde{x}_i v_i z_i) \bmod N$ to extract z_i for all i. Note that the user can introduce the change of variable $\xi_i = v_i z_i$ and instead solve $D(y) = (\sum_{w \in \mathcal{W}} \tilde{x}_i \xi_i) \bmod N$. Evidently, $z_i = 1$ if $\xi_i \neq 0$ and $z_i = 0$ otherwise. Define

$$\Xi := \left\{ \xi \in \mathbb{Z}_N^{|\mathcal{W}|} \,\middle|\, D(y) = \left(\sum_{w \in \mathcal{W}} \tilde{x}_i \xi_i \right) \bmod N \right\}. \tag{3}$$

This set captures all the solutions of the linear equation $D(y) = (\sum_{w \in \mathcal{W}} \tilde{x}_i \xi_i) \bmod N$.

[1] *The advantage captures the superiority of the performance in comparison to a pure random number generator.*

[2] *$f : \mathbb{N} \to \mathbb{R}_{\geq 0}$ is negligible if there exists $n_c \in \mathbb{N}$, $\forall c \in \mathbb{N}$, such that $f(n) \leq 1/n^c, \forall n \geq n_c$ [16].*

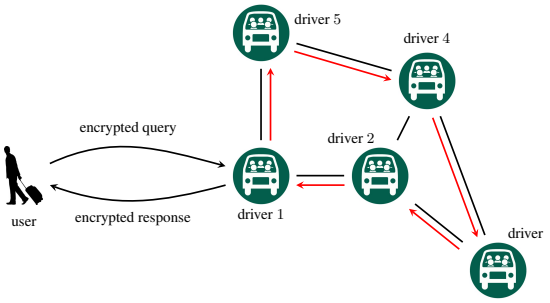

Fig. 2: Communication structure for distributed coordination between a user and multiple drivers in private ride-sharing. The black lines show undirected edges among the drivers for communication. The red arrows show a walk on the graph among the drivers used for responding to the encrypted query of the user.

Proposition 3.4. *Let $t = |\{i \,|\, \tilde{x}_i \neq 0\}| > 1$. Then $|\Xi| \geq (N-1)^{t-1}$ if there exists i such that $\gcd(\tilde{x}_i, N) = 1$.*

Proof: See Appendix A. □

Proposition 3.4 shows that, even if two $\tilde{x}_i$ are non-zero, $|\Xi|$ or the number of solutions of $D(y) = (\sum_{w \in \mathcal{W}} \tilde{x}_i \xi_i) \bmod N$ is larger than $N - 1$. The number of the solutions grows even larger as more $\tilde{x}_i$ become non-zero. This is because t, which captures the number of the non-zero $\tilde{x}_i$, appears as an exponent. Since the security of the encryption relies on the public key N being extremely large[1], the user must check a huge number of solutions. This is numerically impractical.

Proposition 3.5. *Let $t = |\{i \,|\, \tilde{x}_i \neq 0\}| > 2$. Then $|\Xi| \geq 2(N-1)^{t-2}$ if there does not exist i such that $\gcd(\tilde{x}_i, N) = 1$.*

Proof: See Appendix B. □

Proposition 3.5 shows that, by smart planning (selecting $\tilde{x}_i$ such that $\gcd(\tilde{x}_i, N) \neq 1$), the user can only realize if the driver has any interest in the roads and the time windows associated with two entries of $\mathcal{W}$ instead of one by following Algorithm 1. Since in practice $|\mathcal{W}|$ is large, this might not matter.

Remark 3.1 (Computational Complexity)**.** *Algorithm* 1 *requires $|\mathcal{W}|$ encryption operations, where encryption has a cost that is non-linear in N. Because the size of N is constant, the cost of the encryption is also constant (albeit a large constant). This implies that the computational complexity of Algorithm* 1 *is $\mathcal{O}(|\mathcal{W}|)$. Algorithm* 2 *requires $|\overline{\mathcal{W}}|$ exponentiations and multiplications. As a result, the computational complexity of Algorithm* 2 *scales as $\mathcal{O}(|\overline{\mathcal{W}}|)$. Finally, note that, in practice, $|\overline{\mathcal{W}}| \ll |\mathcal{W}|$. Thus the combined computational complexity of Algorithms* 1 *and* 2 *is $\mathcal{O}(|\mathcal{W}|)$.*

Remark 3.2 (Secure Multi-Party Computation)**.** *Note that the function that is being computed in this section is Boolean. Let ζ_w^{user} be a Boolean variable that takes the value true if the user wants to enquire about the availability of the driver for traveling on the time window and road associated with w and takes the value false otherwise. Similarly, define ζ_w^{driver} to be a Boolean variable that takes the value true if $w \in \overline{\mathcal{W}}$ and takes the value false otherwise. The desired output in this case is given by $\bigvee_{w \in \mathcal{W}} (\zeta_w^{\text{user}} \wedge \zeta_w^{\text{driver}})$. This allows the required computations to be implemented using Yao's protocol. Doing so, the user can garble (i.e., encrypt) the*

[1] *Most often p, q are selected as prime numbers with the length of 1024 bits pointing to that $N = \mathcal{O}(2^{2048})$.*

circuit and the driver can receive the encrypted inputs and evaluate the outcome through oblivious transfer, e.g., 1–2 oblivious transfer [56]. The computational complexity of this methodology is $\mathcal{O}(|\mathcal{W}|)$ [47], which is identical to our proposed methodology using semi-homomorphic encryption (see Remark 3.1). This is, in fact, the case because Yao's protocol can be implemented using semi-homomorphic encryption (the methodology of this paper is an implementation of Yao's protocol using Paillier's encryption).

Remark 3.3 (Brute-force Attack)**.** *An adversary who wishes to obtain $\overline{\mathcal{W}}$ corresponding to the driver could query sequentially to check if $w \in \overline{\mathcal{W}}$ for all $w \in \mathcal{W}$ or not. This attack has a worst-case complexity that is polynomial in $|\mathcal{W}|$, specifically $\mathcal{O}(|\mathcal{W}|^2)$. Such a brute-force attack is, in fact, feasible for any secure multi-party computation algorithm that provides a correct answer, such as Yao's protocol. To avoid this, the answers must be statistically corrupted by noise to ensure differential privacy [57]; however, that generates false positive matches between users and drivers, which drastically reduces the utility of the application. To avoid brute-force attacks, it is assumed that the number of function computations allowed by the policy is restricted to a number that is much smaller than $|\mathcal{W}|$.*

3.2 Distributed Coordination

Now, we use the results of the previous section to construct a distributed coordination mechanism for the users and drivers. We use an undirected graph $\mathcal{G}_\mathcal{C}$ with the vertex set $\mathcal{F}$ (i.e., the drivers) and the edge set $\mathcal{E}_\mathcal{C} \subseteq \mathcal{F} \times \mathcal{F}$ in order to model the communication structure between the drivers. This graph must not be mistaken with the transportation network, i.e., the roads over which the vehicles travel. A walk over $\mathcal{G}_\mathcal{C}$ is a sequence of possibly repeated vertices $\mathcal{L} = (v_0, \dots, v_k)$ such that $(v_i, v_{i+1}) \in \mathcal{E}_\mathcal{C}$ with $0 \le i \le k-1$. Figure 2 illustrates the communication structure for distributed coordination between a user and multiple drivers in private ride-sharing. The black lines show undirected edges $\mathcal{E}_\mathcal{C}$ among the drivers $\mathcal{F}$ for communication. The red arrows show a walk on the graph among the drivers used for responding to the encrypted query of the user. We make the following standing assumption.

Assumption 3.1. *$\mathcal{G}_\mathcal{C}$ is connected.*

Due to Assumption 3.1, any driver can communicate with any other driver successfully. Let $\mathcal{L} = (v_0, \dots, v_k)$ be a walk over $\mathcal{G}_\mathcal{C}$ such that the drivers v_0 and v_k can communicate with the user. Because the user wants to check the possibility of ride-sharing with all the drivers in $\mathcal{F}$ (and not a subset of them), the walk $\mathcal{L}$ must span all the vertices of the graph. The existence of the walk is guaranteed by Assumption 3.1. We use this property to develop an algorithm for the drivers to collaboratively respond to the encrypted query of the user.

A user wants to check if any driver operates over the path and the time window associated with $w \in \mathcal{W}$. The user can follow Algorithm 1 to construct the encrypted vector x. The user subsequently submits x to v_0 for coordination. All drivers in the walk $\mathcal{L}$ follow Algorithm 3 to respond to the query of the user. In this algorithm, $\overline{\mathcal{W}}_j \subseteq \mathcal{W}$ is the set of all times and paths over which driver j travels. We can show that, if the user follows Algorithm 1 and the drivers in $\mathcal{L}$ follow Algorithm 3, the provided encrypted response is correct.

Proposition 3.6. *If the user follows Algorithm 1 and all the drivers in $\mathcal{L}$ follow Algorithm 3, then $D(y) \neq 0$ if any of the drivers in $\mathcal{L}$ uses the path and time window associated with w, and $D(y) = 0$ otherwise.*

Proof: The proof follows from the application of Proposition 2.1. □

We can prove a similar result to Proposition 3.2 for the enquiring user in the distributed coordination case as well following the semantic security of the Paillier's encryption method. Therefore, we only focus on the privacy guarantees of the drivers in the remainder of this section.

Algorithm 3 Procedure DISTRESPONSE for the drivers in the walk $\mathcal{L}$ to distributedly respond to the query of the user.

input: $x, \mathcal{L}, (\overline{\mathcal{W}}_j)_{j\in\mathcal{L}}, N$
output: y
procedure DISTRESPONSE(x,$\mathcal{L}$,$(\overline{\mathcal{W}}_j)_{j\in\mathcal{L}}$,$N$)
  # Computed by the drivers in $\mathcal{L}$
  for $j = v_0, \dots, v_{k-1}$ **do**
    for $i \in \overline{\mathcal{W}}_j$ **do**
      Select ω_i randomly in $\{1, \dots, \lfloor N/(|\mathcal{L}|-2)\rfloor\}$
      $x_i \leftarrow x_i^{\omega_i} \bmod N^2$
    end for
  end for
  $y \leftarrow 1$
  for $i \in \overline{\mathcal{W}}_{v_k}$ **do**
    Select ω_i randomly in $\{1, \dots, \lfloor N/(|\mathcal{L}|-2)\rfloor\}$
    $y \leftarrow y(x_i^{\omega_i} \bmod N^2) \bmod N^2$
  end for
  return y
end procedure

Proposition 3.7. *If, for all i, $x_i = E(\tilde{x}_i; r_i)$ for some integer $\tilde{x}_i \in \mathbb{Z}_N$, then*

$$D(y) = \Big(\sum_{w\in\mathcal{W}} \tilde{x}_i v_i \Big(\sum_{j\in\mathcal{L}\setminus\{\ell\}} z_i^j \Big) \Big) \bmod N, \tag{4}$$

where $z_i^j = 1$ if driver $j \in \mathcal{L}$ travels on the path and the time window associated with $i \in \mathcal{W}$ and $z_i^j = 0$ otherwise.

Proof: The proof is similar to that of Proposition 3.3. □

Similarly, following Proposition 3.7, the enquiring user must solve the linear equation

$$D(y) = \Big(\sum_{w\in\mathcal{W}} \sum_{j\in\mathcal{L}\setminus\{\ell\}} \tilde{x}_i \xi_i^j \Big) \bmod N,$$

where $z_i^j = 1$ if $\xi_i^j \neq 0$ and $z_i^j = 0$ otherwise. Construct the set of all possibilities

$$\Xi := \Big\{ (\xi_i^j)_{j\in\mathcal{L}\setminus\{\ell\}} \in \mathbb{Z}_N^{|\mathcal{W}|(|\mathcal{L}|-2)} \,\Big|\, D(y) = \Big(\sum_{w\in\mathcal{W}} \sum_{j\in\mathcal{L}\setminus\{\ell\}} \tilde{x}_i \xi_i^j \Big) \bmod N \Big\}. \tag{5}$$

The following result can be proved regarding the size of the set Ξ extending Propositions 3.4 and 3.5 to the distributed situation.

Proposition 3.8. *The following two statements hold:*

- *Let $t = |\{i \,|\, \tilde{x}_i \neq 0\}| > 1$. Then $|\Xi| \ge (|\mathcal{L}|-2)(N-1)^{t-1}$ if there exists i such that $\gcd(\tilde{x}_i, N) = 1$.*
- *Let $t = |\{i \,|\, \tilde{x}_i \neq 0\}| > 2$. Then $|\Xi| \ge 2(|\mathcal{L}|-2)^2(N-1)^{t-2}$ if there does not exist i such that $\gcd(\tilde{x}_i, N) = 1$.*

Proof: The proof follows a similar line of reasoning as in Propositions 3.4 and 3.5. □

Proposition 3.8 states that the privacy guarantees of the drivers is stronger than the privacy guarantees in the case of two agents (in the previous subsection). This is because the responses of all the drivers get mixed and the user cannot identify the drivers that have responded positively.

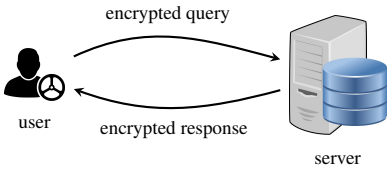

Fig. 3: Communication structure between a road user and a server containing real-time traffic information.

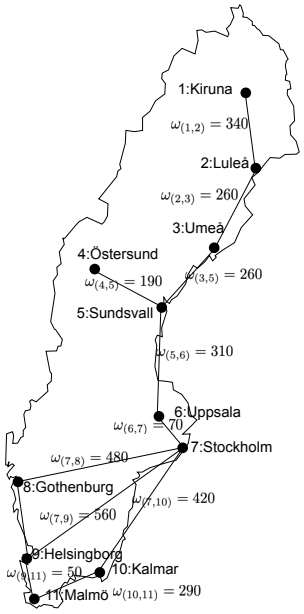

Fig. 4: Example of a transportation network in Sweden modeled by a graph.

4 Private Routing

In this section, the transportation network is modeled by a directed graph $\mathcal{G} = (\mathcal{V}_\mathcal{G}, \mathcal{E}_\mathcal{G})$, where $\mathcal{V}_\mathcal{G} := \{1, \ldots, |\mathcal{V}_\mathcal{G}|\} \subseteq \mathbb{N}$ denotes the set of vertices (e.g., intersections) and $\mathcal{E}_\mathcal{G} \subseteq \mathcal{V}_\mathcal{G} \times \mathcal{V}_\mathcal{G}$ denotes the edges (e.g., roads or road segments)[1]. A weight is associated to each edge $e \in \mathcal{E}_\mathcal{G}$ and denoted by $\omega_e \in \mathbb{N}$. The assumption that the weights are integer numbers is without loss of generality as the weights can always be multiplied by a large integer constant. The weights can denote the average travel time on the roads if our desire is to seek the shortest path in time from source $s \in \mathcal{V}_\mathcal{G}$ to destination $d \in \mathcal{V}_\mathcal{G}$. However, the weights can also show the length of the road or the amount of toll if, respectively, it is desired to find the shortest path in distance or the cheapest travel option. It is assumed that the graph $\mathcal{G}$ is available to both the server and the users. However, the weights $(\omega_e)_{e\in\mathcal{E}_\mathcal{G}}$ are stored only in a database on the server. Finally, the source s and the destination d are only known by user. Figure 4 shows an example of a transportation network modeled by a graph. Here, the graph is undirected, i.e., $(i,j) \in \mathcal{E}_\mathcal{G}$ if and only if $(j,i) \in \mathcal{E}_\mathcal{G}$. The weights $\omega_{(j,i)} = \omega_{(i,j)}$ captures the distance between the cities in kilometres.

To figure out the shortest path between the source and the destination, the user can use Dijkstra's algorithm or dynamic programming [55, pp. 595–600]. The algorithm can be implemented on the server in which case the user needs to transmit the location of the source and the destination to the server. This infringes on the privacy of the user. Alternatively, the algorithm can run locally by the user in which case it needs to query the server containing the database of weights in the vicinity of s and d (with a large-enough radius of inclusion). This approach violates the privacy of the user as the server can figure out the source and the destination (from the sequence of the queried edges). In this paper, the aim is to develop a mechanism using semi-homomorphic encryption, so that the user can query the weight of any edge $e \in \mathcal{E}_\mathcal{G}$ without revealing the identity of the edge e to the server. Figure 3 illustrates the communication structure between a road user and a server containing real-time traffic information.

[1] *It is pivotal to not mistake the transportation network $\mathcal{G}$ with the communication graph $\mathcal{G}_\mathcal{C}$ among the drivers in the previous section.*

Algorithm 4 Procedure REPORTINGWEIGHTS for reporting an encrypted copy of the weights of the edges.

input: $\tilde{x}, \mathcal{G}$
output: $\tilde{y}$
procedure REPORTINGWEIGHTS($\tilde{x}$,$\mathcal{G}$)
  # Computed by the server
  for $\ell \in \mathcal{V}_\mathcal{G}$ **do**
    $\tilde{y}_\ell \leftarrow 1$
    for $k \in \mathcal{V}_\mathcal{G} : w_{\ell k} \neq 0$ **do**
      $\tilde{y}_\ell \leftarrow \tilde{y}_\ell (\tilde{x}_k)^{w_{\ell k}} \bmod N^2$
    end for
  end for
  return $\tilde{y}$
end procedure

Algorithm 5 Procedure EXTRACTINGWEIGHTS for extracting the weights of the edges.

input: $i, \mathcal{G}$
output: $(\omega_{(j,i)})_{(j,i)\in\mathcal{E}_\mathcal{G}}$
procedure EXTRACTINGWEIGHTS(i,$\mathcal{G}$)
  # Computed by the user
  for $\ell \in \mathcal{V}_\mathcal{G}$ **do**
    if $\ell = i$ **then**
      $\tilde{x}_\ell \leftarrow E(1; r_k)$
    else
      $\tilde{x}_\ell \leftarrow E(0; r_k)$
    end if
  end for
  $\tilde{y} \leftarrow$ REPORTINGWEIGHTS($\tilde{x}$)
  $\omega_{(j,i)} \leftarrow D(\tilde{y}_j) \quad \forall j \in \mathcal{V}_\mathcal{G} : (j,i) \in \mathcal{E}_\mathcal{G}$
  return $(\omega_{(j,i)})_{(j,i)\in\mathcal{E}_\mathcal{G}}$
end procedure

4.1 *Private and Secure Communication Framework*

Define the matrix $W = (w_{ij})$ such that $w_{ij} = \omega_{(j,i)}$ if $(j,i) \in \mathcal{E}_\mathcal{G}$ and $w_{ij} = 0$ otherwise. Let, for any $i \in \mathcal{V}_\mathcal{G}$, $x^{(i)}$ denote a vector, where $x_i^{(i)} = 1$ and $x_j^{(i)} = 0$ for $j \neq i$. Calculate

$$y = Wx, \tag{6}$$

where $x = x^{(i)}$ if the user is interested in knowing the weights of all the edges that originate from vertex $i \in \mathcal{V}_\mathcal{G}$, i.e., all the edges $e \in \mathcal{E}_\mathcal{G}$ such that $e = (i,j)$ for some $j \in \mathcal{V}_\mathcal{G}$. Then, y_j is equal to $\omega_{(i,j)}$ if $(i,j) \in \mathcal{E}_\mathcal{G}$ or equal to zero otherwise. A method must be developed to compute the multiplication in (6) on the server without revealing i. This is presented in the following proposition.

Proposition 4.1. *Let $\tilde{x}_\ell = E(x_\ell; r_\ell)$ for all $\ell \in \mathcal{V}_\mathcal{G}$. Calculate $\tilde{y}_\ell$ in $\ell \in \mathcal{V}_\mathcal{G}$ according to*

$$\tilde{y}_\ell = \prod_{k\in\mathcal{V}_\mathcal{G}} (\tilde{x}_k)^{w_{\ell k}} \bmod N^2.$$

Then, $y_\ell = D(\tilde{y}_\ell)$ for all $\ell \in \mathcal{V}_\mathcal{G}$.

Proof: See Appendix C. □

The calculations for which the server is responsible are summarized in Algorithm 4 while the computations that the user needs to perform are shown in Algorithm 5

Remark 4.1 (Computational Complexity)**.** *The server is required to perform $\mathcal{O}(|\mathcal{V}_\mathcal{G}|d_\mathcal{G})$ exponentiations and multiplications, where $d_\mathcal{G}$ is the maximum degree[1] of the graph $\mathcal{G}$. Assuming that the size of the weights is independent of the number of vertices and edges in the graph, the computational complexity of Algorithm* 4 *is of the order of $\mathcal{O}(|\mathcal{V}_\mathcal{G}|d_\mathcal{G})$. The computational complexity of the encryption part of Algorithm* 5 *in lines 3–9 is of the order of $\mathcal{O}(|\mathcal{V}_\mathcal{G}|)$. This is because what being encrypted is only binary. The computational complexity of the decryption part of Algorithm* 5 *in line 11 is of the order of $\mathcal{O}(d_\mathcal{G}c)$, where c is the cost of decrypting a ciphertext generated by Paillier's method. This part is not a function of the size of the underlying graph. Therefore, the computational complexity of both Algorithms* 4 *and* 5 *scales linearly with the size of the graph. This might not be desirable as the size of the graph (i.e., the underlying transportation network) grows. For large transportation networks, the server and user can agree to focus on a smaller graph (subgraph of the original transportation network containing source and destination) at the cost of reducing the privacy guarantees (by revealing neighborhoods in which source and destination are located); see Subsection* 4.2 *for capturing the trade-off between privacy and computational complexity.*

In the remainder of this subsection, the privacy guarantees of the algorithm are formally analyzed. Assume that an adversary can eavesdrop on the communications of the user with the server. Note that the adversary can even be the server itself. Therefore, the adversary is assumed to have access to both $\tilde{x}$ and $\tilde{y}$ (which is evidently a function of $\tilde{x}$).

Definition 4.1. *Let the adversary propose two nodes $i_1, i_2 \in \mathcal{V}_\mathcal{G}$. The user chooses at random i from $\{i_1, i_2\}$ with equal probability and sends $\tilde{x}$ constructed using Algorithm* 5*. The adversary can based on its knowledge of $\tilde{x}$ (and i_1, i_2) guess i. This guess is denoted by i'. The driver's advantage is given by $\mathrm{Adv}(k) := |\mathbb{P}\{i = i'\} - 1/2|$, where k denotes the security parameter (similar to the previous section, e.g., $\min(p, q)$ in Paillier's technique). The proposed strategy is defined to be private if Adv is negligible.*

Proposition 4.2. *Under the Decisional Composite Residuosity Assumption, Algorithm* 5 *is private in the sense of Definition* 4.1*.*

Proof: The proof follows from the semantic security of Paillier's encryption under the Decisional Composite Residuosity Assumption [53]. □

In general, the user can choose any node $i' \in \mathcal{V}_\mathcal{G}$ to find the weights of the edges. In this case, Proposition 4.2 shows that it is not possible for the adversary (at least as security parameter approaches infinity) to figure out the nodes in which the user is interested just from its communications with the server. In fact, the adversary cannot provide a guess than is better than flipping a ($|\mathcal{V}_\mathcal{G}|$ sided) coin, i.e., $\mathbb{P}\{i = i'\} \to 1/|\mathcal{V}_\mathcal{G}|$ as the security parameter grows.

Finally, note that if the user can submit queries of the form (6) in a private manner, it can query the online database for the state of the traffic on the roads connecting its origin and destination and use Dijkstra's algorithm to find the shortest path.

[1] *Note that, in general, $d_\mathcal{G}$ is of the order of $|\mathcal{E}_\mathcal{G}|$; however, in transportation system, $d_\mathcal{G}$ is always much smaller than that as intersections rarely contain more than four roads.*

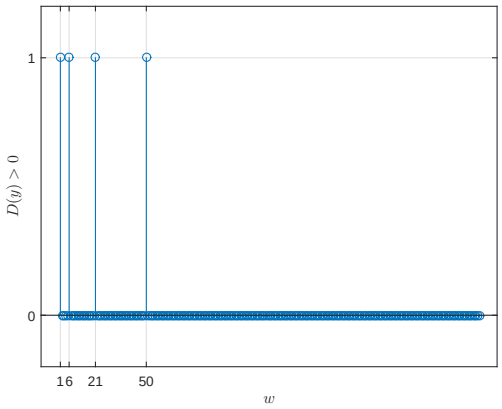

Fig. 5: The outcome of Algorithms 1 and 2 on if $D(y) > 0$ or not for various roads and time windows associated with w.

4.2 Improving the Computational Complexity

The computational complexity of the algorithms that are used by the server and the user grow linearly with $|\mathcal{V}_\mathcal{G}|$. This makes it difficult to directly use this procedure for real transportation systems as the number of nodes can be staggeringly high. To be able to circumvent this issue, in this section, the problem is restricted to a subgraph $\overline{\mathcal{G}} \subseteq \mathcal{G}$. It is important to select a subgraph $\overline{\mathcal{G}}$ that is fully connected and contains both the source and the destination (because otherwise there might not exist a path between the nodes). If the search is restricted to $\overline{\mathcal{G}}$, following the results of Proposition 4.2, the probability that an adversary can guess the identity of the nodes to which the user is interested is equal to $\mathbb{P}\{i' = i\} = 1/|\mathcal{V}_{\overline{\mathcal{G}}}|$. Noting that $|\mathcal{V}_{\overline{\mathcal{G}}}| \leq |\mathcal{V}_\mathcal{G}|$, this results in a weaker privacy guarantee for the user. To balance the privacy requirement and the reduction in the computational complexity, the following integer program can be used:

$$\min_{\overline{\mathcal{G}} \subseteq \mathcal{G}} \quad |\mathcal{V}_{\overline{\mathcal{G}}}|, \tag{7a}$$

$$\text{s.t.} \quad \overline{\mathcal{G}} \subseteq \mathcal{G} \text{ is connected}, \tag{7b}$$

$$s \in \mathcal{V}_{\overline{\mathcal{G}}}, \tag{7c}$$

$$d \in \mathcal{V}_{\overline{\mathcal{G}}}, \tag{7d}$$

$$|\mathcal{V}_\mathcal{G}| \leq \varrho |\mathcal{V}_{\overline{\mathcal{G}}}|, \tag{7e}$$

where (7e) is the privacy erosion constraint ensuring that the ratio of $\mathbb{P}\{i' = i\}$ after and before the reduction is bounded by a constant $\varrho > 1$. This constant is a design parameter. Evidently, the optimization problem (7) is always feasible (because $\mathcal{G}$ always satisfies the constraints), thus it admits a solution.

5 Numerical Example

In this section, we consider an example in which user aims to find a driver for ride-sharing. There are $|\mathcal{P}| = 10$ roads. The time of the day is discretized into $|\mathcal{T}| = 24$ one-hour windows. Therefore, $|\mathcal{W}| = 240$. Assume that the driver travels on the roads and the time windows associated with $w = 1, 6, 21, 50$.

Figure 5 shows the outcome of Algorithms 1 and 2 for various queries submitted by the user with a key length of 128 bits. The vertical axis of Figure 5 is one if the decryption $D(y) > 0$ and zero otherwise. We can clearly see that $D(y) > 0$ only for $w = 1, 6, 21, 50$. Therefore, the proposed algorithms allow the user and the driver to correctly coordinate their actions without revealing private information to each other.

The secure communication channel comes at the price of computational complexity. Figure 6 shows the computation time and

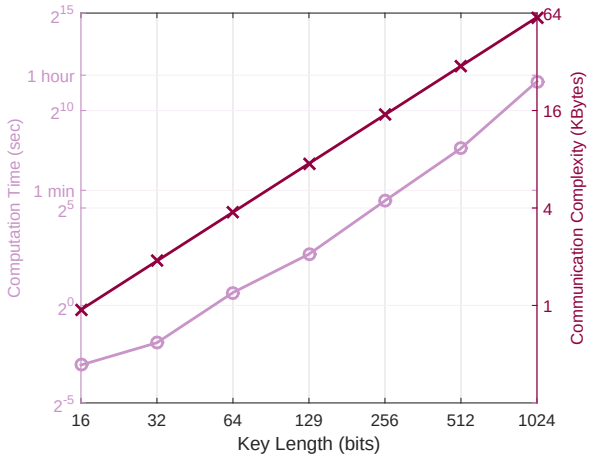

Fig. 6: The computation time (—⊖—) and the communication burden (—×—) associated with executing Algorithms 1 and 2 versus the key length.

the communication burden of executing Algorithms 1 and 2 for various key lengths. The computation is done with Python programming language on Windows 7 over a PC with Intel(R) i7-4770 CPU at 3.40GHz and 16GB of RAM[1]. The computation time and the amount of data the user needs to communicate to the driver rapidly increase with increasing key length. The computational time (in seconds) for Algorithms 1 and 2 scales as $\mathcal{O}(k^{2.44})$ with key length k. Therefore, the computational complexity of the proposed algorithms grows polynomially with the key length. The communication burden is a linear function of the key length because the size of the integers that must be transmitted grows linearly with key length.

National Institute of Standards and Technology (NIST) recommends the use of a key length of 2048 bits for factoring-based asymmetric encryption algorithms[2]. This recommendation is to ensure that brute-force attacks are not physically possible during the life-time of the services and is based on projections of computing technologies. For privacy-preserving policies, however, such a high standard might be unnecessary. To demonstrate this, consider RSA, which is a similar encryption methodology and also a semi-homomorphic encryption relying on hardness of prime number factorization. It is a long time since the first time that RSA encryption (relying on polynomial factorization) was attacked using a brute-force methodology; see RSA Challenge[3]. Factorization of 430 and 463 bit numbers has been shown to take approximately 1000 and 2000 MIPS[4]-years of computing time, respectively [59]. It must be noted that 1 MIPS-years is approximately 31.5 trillion instructions[5]. The computer used for the numerical analysis presented in this paper can compute 0.12 trillion instructions per second. Thus, factorization of 430 and 463 bit numbers takes approximately 73 and 146 hours. These numbers are certainly not safe for use in finance or military applications[6]. However, for privacy-preserving ride-sharing, they probably provide strong-enough guarantees. This is because by the time that a malicious driver breaks the code, the user is in a different location.

[1] *We developed a dedicated digital engine for computations based on Paillier encryption using Altera Cyclone V FPGA that was 25 times faster than the computation times in Figure* 6 *[58].*
[2] `https://www.keylength.com/en/4/`, *accessed on 2018.*
[3] `https://en.wikipedia.org/wiki/RSA_Factoring_Challenge`
[4] *MIPS stands for mega instructions per second*
[5] *(*10^6 *instructions/second)* $\times$ *(86400 seconds/day)* $\times$ *(365 days/year)* $\approx$ *31.5 trillion instructions*
[6] *Otherwise, we must change our credit cards every 3 days.*

6 Conclusions and Future Work

In this paper, we developed a private framework for ride-sharing and online routing using the Paillier's encryption method. We proved strong privacy and security guarantees for the users and the drivers. We used numerical simulations to discuss the feasibility of the framework. Future studies can focus on developing a coordination algorithm among the users and the drivers that can accommodate adjustments to departure times and routes for increasing ride-sharing potential. Another approach is to utilize secret sharing in which a secret is divided into multiple shares and each agent receives one share, which appears random to the receiving party. Then, appropriate computations on the secret shares can be performed to evaluate the final answer [50, 51].

Acknowledgements

The work of F. Farokhi was supported by the veski Fellowship from the State Government of Victoria, facilitating this collaboration. The work of I. Shames was supported by a grant (MyIP: ID6874) from the Defence Science and Technology Group (DSTG). The work of K. H. Johansson was supported by Knut och Alice Wallenbergs Foundation, Swedish Foundation for Strategic Research, and Swedish Research Council.

A Proof of Proposition 3.4

For i such that $\gcd(\tilde{x}_i, N) = 1$, there exists $\tilde{x}_i^{-1} \bmod N$. Thus $\xi_i = (D(y) - \sum_{j\neq i} \tilde{x}_i^{-1}\tilde{x}_j\xi_j) \bmod N$. Therefore, all $(\xi_j)_{j\neq i}$ are free variables, i.e., for any selection of $(\xi_j)_{j\neq i}$, there exists ξ_i that satisfies the linear equation $D(y) = \sum_{w\in\mathcal{W}} \tilde{x}_i\xi_i \bmod N$. This points to that the number of solutions of the linear equation modulo (which is equal to $|\Xi|$) is equal to the number of all the possible choices of $(\xi_j)_{j\neq i}$.

B Proof of Proposition 3.5

If there does not exist i such that $\gcd(\tilde{x}_i, N) = 1$, we can construct two sets where in the first one $\tilde{x}_i$ is divisible by q and in the second $\tilde{x}_i$ is divisible by p (note that $\tilde{x}_i$ cannot be divisible by both as otherwise it will be larger than $\mathrm{lcm}(p, q) = pq = N$). Let these sets be denoted by $\mathcal{J}_1$ and $\mathcal{J}_2$, respectively. In this case, we can write

$$D(y) = q\Bigg(\sum_{j\in\mathcal{J}_1} \xi_j \underbrace{\left(\frac{\tilde{x}_j}{q}\right)}_{\tilde{x}'_j}\Bigg) + p\Bigg(\sum_{j\in\mathcal{J}_2} \xi_j \underbrace{\left(\frac{\tilde{x}_j}{p}\right)}_{\tilde{x}'_j}\Bigg) \bmod N.$$

Noting that $\gcd(p, q) = 1$ (since p and q are prime numbers), this equation can be separated into

$$\alpha = \sum_{j\in\mathcal{J}_1} \xi_j\tilde{x}'_j \bmod N, \tag{8a}$$

$$\beta = \sum_{j\in\mathcal{J}_2} \xi_j\tilde{x}'_j \bmod N, \tag{8b}$$

where $\alpha = D(y)\bar{\alpha}$ and $\beta = D(y)\bar{\beta}$ with $\bar{\alpha}$ and $\bar{\beta}$ denoting Bézout coefficients, i.e., $\bar{\alpha}q + \bar{\beta}p = 1$. There are only two Bézout coefficients that satisfy $|\bar{\alpha}| < p$ and $|\bar{\beta}| < q$ [60, Proposition 13, p. 60]. The number of solutions to (8a) can be lower bounded with the same line of reasoning as in Proposition 3.4 by $(N-1)^{|\mathcal{J}_1|-1}$. Similarly, the number of solutions to (8b) can be lower bounded by $(N-1)^{|\mathcal{J}_2|-1}$. This concludes the proof.

C Proof of Proposition 4.1

We can use Item 2 in Proposition 2.1 to show that

$$\begin{aligned}(\tilde{x}_k)^{w_{\ell k}} &= E(x_k; r_k)^{w_{\ell k}} \\ &= E(w_{\ell k} x_k; r_k^{w_{\ell k}}).\end{aligned} \tag{9}$$

Further, using (9) and Item 1 in Proposition 2.1 results in

$$\begin{aligned}\prod_{k\in\mathcal{V}_\mathcal{G}} (\tilde{x}_k)^{w_{\ell k}} &= \prod_{k\in\mathcal{V}_\mathcal{G}} E(w_{\ell k} x_k; r_k^{w_{\ell k}}) \\ &= E\Big(\sum_{k\in\mathcal{V}_\mathcal{G}} w_{\ell k} x_k; \prod_{k\in\mathcal{V}_\mathcal{G}} r_k^{w_{\ell k}}\Big) \\ &= E\Big(y_\ell; \prod_{k\in\mathcal{V}_\mathcal{G}} r_k^{w_{\ell k}}\Big).\end{aligned}$$

This concludes the proof.

D References

1 Besselink, B., Turri, V., van de Hoef, S.H., Liang, K.Y., Alam, A., Mårtensson, J., et al.: 'Cyber-physical control of road freight transport', *Proceedings of the IEEE*, 2016, **104**, (5), pp. 1128–1141
2 Hoh, B., Gruteser, M., Herring, R., Ban, J., Work, D., Herrera, J.C., et al. 'Virtual trip lines for distributed privacy-preserving traffic monitoring'. In: Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services. MobiSys '08. (New York, NY, USA, 2008. pp. 15–28
3 Fries, R.N., Gahrooei, M.R., Chowdhury, M., Conway, A.J.: 'Meeting privacy challenges while advancing intelligent transportation systems', *Transportation Research Part C: Emerging Technologies*, 2012, **25**, pp. 34–45
4 Yu, R., Kang, J., Huang, X., Xie, S., Zhang, Y., Gjessing, S.: 'Mixgroup: Accumulative pseudonym exchanging for location privacy enhancement in vehicular social networks', *IEEE Transactions on Dependable and Secure Computing*, 2016, **13**, (1), pp. 93–105
5 Jaworski, P., Edwards, T., Moore, J., Burnham, K. 'Cloud computing concept for intelligent transportation systems'. In: Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on. (IEEE, 2011. pp. 391–936
6 Hoh, B., Iwuchukwu, T., Jacobson, Q., Work, D., Bayen, A.M., Herring, R., et al.: 'Enhancing privacy and accuracy in probe vehicle-based traffic monitoring via virtual trip lines', *IEEE Transactions on Mobile Computing*, 2012, **11**, (5), pp. 849–864
7 Timpner, J., Schürmann, D., Wolf, L.: 'Trustworthy parking communities: helping your neighbor to find a space', *IEEE Transactions on Dependable and Secure Computing*, 2016, **13**, (1), pp. 120–132
8 Troncoso, C., Danezis, G., Kosta, E., Balasch, J., Preneel, B.: 'Pripayd: Privacy-friendly pay-as-you-drive insurance', *IEEE Transactions on Dependable and Secure Computing*, 2011, **8**, (5), pp. 742–755
9 Cui, L., Xie, G., Qu, Y., Gao, L., Yang, Y.: 'Security and privacy in smart cities: Challenges and opportunities', *IEEE Access*, 2018, **6**, pp. 46134–46145
10 Kargl, F., Friedman, A., Boreli, R. 'Differential privacy in intelligent transportation systems'. In: Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks. (ACM, 2013. pp. 107–112
11 Canepa, E.S., Claudel, C.G. 'A framework for privacy and security analysis of probe-based traffic information systems'. In: Proceedings of the 2nd ACM international conference on High confidence networked systems. (, 2013. pp. 25–32
12 Farokhi, F., Shames, I. 'Preserving privacy of agents in participatory-sensing schemes for traffic estimation'. In: Proceedings of the 55th IEEE Conference on Decision and Control. (, 2016.
13 Ogden, K.: 'Privacy issues in electronic toll collection', *Transportation Research Part C: Emerging Technologies*, 2001, **9**, (2), pp. 123–134
14 De.Montjoye, Y.A., Hidalgo, C.A., Verleysen, M., Blondel, V.D.: 'Unique in the crowd: The privacy bounds of human mobility', *Scientific reports*, 2013, **3**, pp. 1376
15 Gao, J., Sun, L., Cai, M.: 'Quantifying privacy vulnerability of individual mobility traces: A case study of license plate recognition data', *Transportation Research Part C: Emerging Technologies*, 2019, **104**, pp. 78–94
16 Ostrovsky, R., Skeith, E.W.: 'Private searching on streaming data', *Journal of Cryptology*, 2007, **20**, (4), pp. 397–430
17 Yi, X., Bertino, E. 'Private searching for single and conjunctive keywords on streaming data'. In: Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society. (ACM, 2011. pp. 153–158
18 Yi, X., Bertino, E., Vaidya, J., Xing, C.: 'Private searching on streaming data based on keyword frequency', *IEEE Transactions on Dependable and Secure Computing*, 2014, **11**, (2), pp. 155–167
19 Boneh, D., Waters, B. 'Conjunctive, subset, and range queries on encrypted data'. In: Vadhan, S.P., editor. Theory of Cryptography: 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007. Proceedings. (Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. pp. 535–554
20 Vora, A.V., Hegde, S.: 'Keyword-based private searching on cloud data along with keyword association and dissociation using Cuckoo filter', *International Journal of Information Security*, 2019, **18**, (3), pp. 305–319
21 Myles, G., Friday, A., Davies, N.: 'Preserving privacy in environments with location-based applications', *IEEE Pervasive Computing*, 2003, **2**, (1), pp. 56–64
22 Beresford, A.R., Stajano, F.: 'Location privacy in pervasive computing', *IEEE Pervasive Computing*, 2003, **2**, (1), pp. 46–55
23 Bamba, B., Liu, L., Pesti, P., Wang, T. 'Supporting anonymous location queries in mobile environments with privacygrid'. In: Proceedings of the 17th international conference on World Wide Web. (, 2008. pp. 237–246
24 Yi, X., Paulet, R., Bertino, E. 'Nearest neighbor queries with location privacy'. In: Homomorphic Encryption and Applications. (Springer International Publishing, 2014. pp. 81–99
25 Kido, H., Yanagisawa, Y., Satoh, T. 'An anonymous communication technique using dummies for location-based services'. In: Proceedings of International Conference on Pervasive Services. (, 2005. pp. 88–97
26 Paulet, R., Kaosar, M.G., Yi, X., Bertino, E.: 'Privacy-preserving and content-protecting location based queries', *IEEE Transactions on Knowledge and Data Engineering*, 2014, **26**, (5), pp. 1200–1210
27 Kerschbaum, F., Lim, H.W., Gudymenko, I. 'Privacy-preserving billing for e-ticketing systems in public transportation'. In: Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society. (ACM, 2013. pp. 143–154
28 Ardagna, C.A., Cremonini, M., di Vimercati, S.D.C., Samarati, P.: 'An obfuscation-based approach for protecting location privacy', *IEEE Transactions on Dependable and Secure Computing*, 2011, **8**, (1), pp. 13–27
29 Badsha, S., Yi, X., Khalil, I., Liu, D., Nepal, S., Bertino, E. 'Privacy preserving location recommendations'. In: International Conference on Web Information Systems Engineering. (Springer, 2017. pp. 502–516
30 Yang, S., Tang, S., Zhang, X.: 'Privacy-preserving k nearest neighbor query with authentication on road networks', *Journal of Parallel and Distributed Computing*, 2019, **134**, pp. 25–36
31 Li, H., Dan, G., Borisov, N., Nahrstedt, K., Gunter, C.A. 'Janus: Privacy-preserving billing for dynamic charging of electric vehicles'. (University of Illinois at Urbana-Champaign, 2019. `https://www.ideals.illinois.edu/handle/2142/104027`
32 Rajendran, B., Pandey, A.K., Bindhumadhava, B. 'Secure and privacy preserving digital payment'. In: 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI). (IEEE, 2017. pp. 1–5
33 Aïvodji, U.M., Gambs, S., Huguet, M.J., Killijian, M.O.: 'Meeting points in ridesharing: A privacy-preserving approach', *Transportation Research Part C: Emerging Technologies*, 2016, **72**, pp. 239–253
34 Yu, H., Jia, X., Zhang, H., Yu, X., Shu, J.: 'Psride: Privacy-preserving shared ride matching for online ride hailing systems', *IEEE Transactions on Dependable and Secure Computing*, 2019,
35 Aïvodji, U.M., Huguenin, K., Huguet, M.J., Killijian, M.O. 'SRide: A privacy-preserving ridesharing system'. In: Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks. (ACM, 2018. pp. 40–50
36 Hallgren, P., Orlandi, C., Sabelfeld, A. 'PrivatePool: Privacy-preserving ridesharing'. In: 2017 IEEE 30th Computer Security Foundations Symposium (CSF). (IEEE, 2017. pp. 276–291
37 Pagnin, E., Gunnarsson, G., Talebi, P., Orlandi, C., Sabelfeld, A.: 'TOPPool: Time-aware optimized privacy-preserving ridesharing', *Proceedings on Privacy Enhancing Technologies*, 2019, **2019**, (4), pp. 93–111
38 Farokhi, F., Shames, I., Batterham, N. 'Secure and private cloud-based control using semi-homomorphic encryption'. In: Proceedings of the 6th IFAC Workshop on Distributed Estimation and Control in Networked Systems. (, 2016.
39 Kogiso, K., Fujita, T. 'Cyber-security enhancement of networked control systems using homomorphic encryption'. In: Proceedings of the 54th Annual Conference on Decision and Control. (, 2015. pp. 6836–6843
40 Lin, Y., Farokhi, F., Shames, I., Nešić, D. 'Secure control of nonlinear systems using semi-homomorphic encryption'. In: 2018 IEEE Conference on Decision and Control (CDC). (IEEE, 2018. pp. 5002–5007
41 Alexandru, A.B., Pappas, G.J. 'Encrypted LQG using labeled homomorphic encryption'. In: Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems. (ACM, 2019. pp. 129–140
42 Cheon, J.H., Han, K., Hong, S.M., Kim, H.J., Kim, J., Kim, S., et al.: 'Toward a secure drone system: Flying with real-time homomorphic authenticated encryption', *IEEE access*, 2018, **6**, pp. 24325–24339
43 Damgård, I., Pastro, V., Smart, N., Zakarias, S. 'Multiparty computation from somewhat homomorphic encryption'. In: Advances in Cryptology–CRYPTO 2012. (Springer, 2012. pp. 643–662
44 Gentry, C., Halevi, S., Smart, N.P. 'Homomorphic evaluation of the aes circuit'. In: Advances in Cryptology–CRYPTO 2012. (Springer, 2012. pp. 850–867
45 Yao, A.C. 'Protocols for secure computations'. In: Foundations of Computer Science, 1982. SFCS'08. 23rd Annual Symposium on. (IEEE, 1982. pp. 160–164
46 Lindell, Y., Pinkas, B.: 'A proof of security of yao's protocol for two-party computation', *Journal of Cryptology*, 2009, **22**, (2), pp. 161–188
47 Lindell, Y., Pinkas, B. 'An efficient protocol for secure two-party computation in the presence of malicious adversaries'. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. (Springer, 2007. pp. 52–78
48 Kolesnikov, V., Schneider, T. 'Improved garbled circuit: Free XOR gates and applications'. In: International Colloquium on Automata, Languages, and Programming. (Springer, 2008. pp. 486–498
49 Kreuter, B., Shelat, A., Shen, C.H. 'Billion-gate secure computation with malicious adversaries.'. In: USENIX Security Symposium. vol. 12. (, 2012. pp. 285–300

50 Chaum, D., Crépeau, C., Damgard, I. 'Multiparty unconditionally secure protocols'. In: Proceedings of the 20th Annual ACM Symposium on Theory of Computing. (ACM, 1988. pp. 11–19
51 Kamm, L., Willemson, J.: 'Secure floating point arithmetic and private satellite collision analysis', *International Journal of Information Security*, 2015, **14**, (6), pp. 531–548
52 Farokhi, F., Shames, I., Johansson, K.H. 'Private and secure coordination of matchmaking for heavy-duty vehicle platooning'. In: Proceedings of the IFAC World Congress. (, 2017.
53 Paillier, P. 'Public-key cryptosystems based on composite degree residuosity classes'. In: Stern, J., editor. Advances in Cryptology — EUROCRYPT '99: International Conference on the Theory and Application of Cryptographic Techniques Prague, Czech Republic, May 2–6, 1999 Proceedings. (Berlin, Heidelberg: Springer Berlin Heidelberg, 1999. pp. 223–238
54 Yi, X., Paulet, R., Bertino, E.: 'Homomorphic Encryption and Applications'. Springer Briefs in Computer Science. (Springer International Publishing, 2014)
55 Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: 'Introduction To Algorithms'. (MIT Press, 2001)
56 Even, S., Goldreich, O., Lempel, A.: 'A randomized protocol for signing contracts', *Communications of the ACM*, 1985, **28**, (6), pp. 637–647
57 Dwork, C. 'Differential privacy: A survey of results'. In: International Conference on Theory and Applications of Models of Computation. (Springer, 2008. pp. 1–19
58 Tran, J., Farokhi, F., Cantoni, M., Shames, I.. 'Implementing homomorphic encryption based secure feedback control'. (, 2019. preprint. arXiv preprint arXiv:1902.06899.
59 Cavallar, S., Dodson, B., Lenstra, A., Leyland, P., Lioen, W., Montgomery, P.L., et al. 'Factorization of RSA-140 using the number field sieve'. In: International Conference on the Theory and Application of Cryptology and Information Security. (Springer, 1999. pp. 195–207
60 Thierry, V.: 'Handbook of mathematics'. (Books on Demand, 2015)