sensors

MDPI

Article

Incentive Mechanism for Privacy-Preserving Collaborative Routing Using Secure Multi-Party Computation and Blockchain

Chaojie Wang [1] and Srinivas Peeta [1,2,*]

[1] School of Civil and Environmental Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA; chaojie.wang@gatech.edu
[2] H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA
* Correspondence: peeta@gatech.edu; Tel.: +1-(404)-894-2243

Abstract: Traffic congestion results from the spatio-temporal imbalance of demand and supply. With the advances in connected technologies, incentive mechanisms for collaborative routing have the potential to provide behavior-consistent solutions to traffic congestion. However, such mechanisms raise privacy concerns due to their information-sharing and execution-validation procedures. This study leverages secure Multi-party Computation (MPC) and blockchain technologies to propose a privacy-preserving incentive mechanism for collaborative routing in a vehicle-to-everything (V2X) context, which consists of a collaborative routing scheme and a route validation scheme. In the collaborative routing scheme, sensitive information is shared through an off-chain MPC protocol for route updating and incentive computation. The incentives are then temporarily frozen in a series of cascading multi-signature wallets in case vehicles behave dishonestly or roadside units (RSUs) are hacked. The route validation scheme requires vehicles to create position proofs at checkpoints along their selected routes with the assistance of witness vehicles using an off-chain threshold signature protocol. RSUs will validate the position proofs, store them on the blockchain, and unfreeze the associated incentives. The privacy and security analysis illustrates the scheme's efficacy. Numerical studies reveal that the proposed incentive mechanism with tuned parameters is both efficient and implementable.

Keywords: incentive mechanism; secure multi-party computation; blockchain; privacy; collaborative routing

Citation: Wang, C.; Peeta, S. Incentive Mechanism for Privacy-Preserving Collaborative Routing Using Secure Multi-Party Computation and Blockchain. *Sensors* **2024**, *24*, 542. https://doi.org/10.3390/s24020542

Academic Editors: Helena Rifà-Pous and Radek Fujdiak

Received: 7 December 2023
Revised: 8 January 2024
Accepted: 10 January 2024
Published: 15 January 2024

1. Introduction

Traffic congestion occurs as a result of an imbalance in demand and supply on a spatio-temporal scale. The well-known Braess' Paradox demonstrates that relying solely on supply side solutions that focus on increasing the capacity of existing infrastructure without regard for traveler behavior may have a negligible or even detrimental effect on the performance of traffic networks. Demand-side strategies employ behavioral interventions to encourage shifts in travel mode, travel routes, and departure times [1,2]. Of these, en route re-routing is the most challenging problem due to the inherent dynamics and randomness. Over the last few decades, effective behavioral intervention strategies have been developed, such as toll and incentive mechanisms, using model-based [3–7] or model-free [8–10] approaches to influence travelers' en route behavior and thereby alleviate traffic congestion. However, these intervention strategies frequently overlook individual-level heterogeneity, rely on multiple user classes to reflect distinct behavioral patterns, and suffer from computational tractability issues associated with centralized computation.

Emerging connected technologies enable information sharing among vehicles and infrastructure through vehicle-to-everything (V2X) communications in a connected traffic environment [11–15], facilitating more informed collaborations among connected vehicles (CVs) in the re-routing process [16–18]. Li et al. [19] proposed a routing method facilitating

the navigation through passing time windows in a connected vehicle environment; this has informed the direction of our study. Additionally, Li et al. [20] developed a self-evolving routing method, which introduces a novel formulation in the spatial domain that resolves the mismatch between routing and planning found in conventional studies. This spatial domain-based planning method represents a key contribution to the field of cooperative routing. Nevertheless, individual heterogeneities are still not captured in these studies.

Wang et al. [21] proposed a novel incentive mechanism for collaborative routing in a connected and autonomous vehicular environment, which leverages individual heterogeneity in the route preferences to enhance the system performance while ensuring user satisfaction. A decentralized computational framework was developed to enable efficient network-level deployment. On the other hand, individual heterogeneity necessitates that each passenger discloses their personal preferences (e.g., value of time) as model inputs, which can represent sensitive information that can be exploited. Also, as collaborative routing reveals model outputs, travelers' updated routes, and incentives to other travelers, it raises privacy issues. Moreover, validation of execution (i.e., validating whether each vehicle travels on the route selected in the incentive mechanism) can cause a substantial computation burden and result in more privacy leaks if traditional centralized methods like sharing GPS traces are used. Further, such privacy risks may act as a barrier to participation in the system for societally vulnerable groups (e.g., sharing a low value of time may indicate a low income level), further impairing mobility equity.

Blockchain has been gaining enormous attention since the whitepaper on Bitcoin [22]. A blockchain is a decentralized ledger with tamper resistance ensured by cryptography. The popularity of blockchain is not limited to cryptocurrencies. Due to its decentralized nature, it has demonstrated its promise in Internet of Things and vehicular ad hoc networks [23–25]. However, currently, a blockchain is mainly used to record and share public information such as traffic incidents [26] rather than sensitive personal information in transportation applications. In other studies [27], a blockchain is also used to store virtual credits (which quantify the level of trust that users can place in certain users based on their historical behavior). However, route preferences and historical travel routes are sensitive information that can be potentially leaked even if anonymous identities are used in blockchains. As for the anonymity paradox of Bitcoin, though every Bitcoin account is anonymous, identity information can still be leaked through transaction pattern analysis because every transaction is transparent. The historical travel routes associated with an account can be used to deduce that user's travel pattern and potentially leak their real-world identity. Therefore, on-chain computation alone does not enable a privacy-preserving incentive mechanism for collaborative routing.

Secure multi-party computation (MPC) is a subfield of cryptography which allows a group to jointly compute a function without disclosing any participants' private inputs. A few studies [28,29] leverage MPC in intelligent transportation systems to address privacy concerns. However, none of them address its potential for collaborative routing.

To address privacy concerns in collaborative routing for CVs, this study first proposes a privacy-preserving incentive mechanism based on the decentralized mechanism developed in our previous work [21]. Specifically, a collaborative routing scheme is developed to enable travelers to update routes and compute associated incentives following an MPC protocol without disclosing their value of time. Then, a blockchain-based route validation scheme is proposed to securely validate travelers' whereabouts at checkpoints along selected routes with the assistance of nearby vehicles (i.e., witness vehicles) while allowing them to conceal their trajectories in the blockchain. Combining on-chain and off-chain cryptographic protocols, the proposed incentive mechanism protects sensitive personal information throughout peer vehicle collaborations and prevents malicious parties from conducting pattern-analysis attacks on the blockchain, ensuring that the entire collaborative routing process adheres to a high standard of privacy.

It is worth noting that the privacy concerns mentioned earlier are not specifically associated with the collaborative routing strategy presented in [21]. Rather, they stem from the inherent nature of personalization. To effectively account for the diverse heterogeneities of individual users, personalization necessitates the incorporation of users' unique characteristics and preferences to generate tailored outputs that cater to their specific interests. Consequently, while [21] primarily addresses the computational efficiency of enabling personalization within the routing context, the current study emphasizes the implementation of structured privacy protection techniques to mitigate the privacy challenges associated with personalization.

The rest of the paper is organized as follows. Section 2 provides an overview of the proposed incentive mechanism. Section 3 presents detailed protocols of the mechanism. Section 4 discusses numerical studies. Section 5 concludes the paper by summarizing contributions and future directions.

2. Mechanism Overview

This study's problem context is similar to that of our prior study [21]. As depicted in Figure 1, the major objective is to encourage vehicles to reroute collaboratively during their trips in order to improve system performance (specifically, reducing total travel time in this study). When assigning vehicles, the system performance evaluation is based on vehicles' estimated travel times. Given the difficulty of precisely estimating long-term travel times in the real world, the method will be executed repeatedly throughout the whole horizon of interest. In each iteration, vehicles reroute based on precisely predicted travel times inside the local range (shown by the gray line in Figure 1) and approximations of travel times outside the local range. The decentralized incentive system in [21] follows a hierarchical architecture. Vehicles with the same local origin–destination (OD) pairs are grouped together (the temporary destinations within the defined local range in [21]). First, a route flow assignment model finds the optimal route flows for all vehicle groups in each iteration (with the same local OD pairs). Then, using these optimal flows and the value of time for each vehicle as inputs, a vehicle assignment model assigns vehicles to various routes within each vehicle group. Then, an envy-free procedure produces incentives for every vehicle in the group to ensure participation willingness and behavioral honesty. Notably, in [21], the participation willingness did not account for the potential disadvantages associated with individuals' privacy concerns about sharing sensitive data during the process. Similarly, whereas behavioral honesty implies that users' utilities are maximized when they reveal their real value of time, the utility functions did not account for the negative consequences of disclosing the value of time.

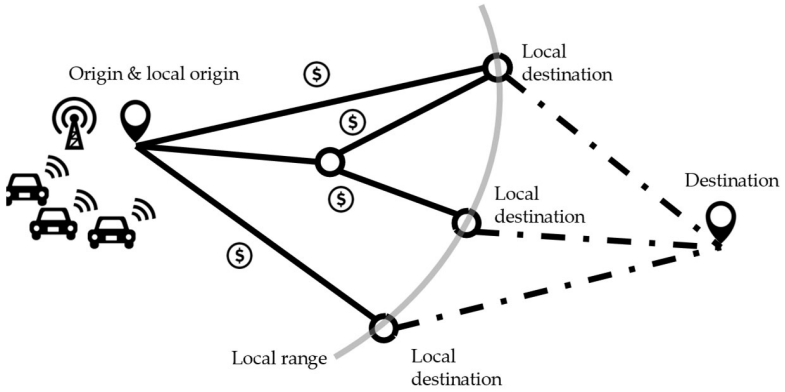

Figure 1. Nudging re-routing behavior with incentives for a local system-optimal assignment (the solid lines and circles denote the road links and nodes within the local range; the dashed lines are the remaining routes from the local destinations to the destination; and dollar signs represent the incentives on the four local routes in the figure).

For instance, there are four local routes for the group of vehicles in Figure 1. At the end of the current iteration, the vehicles will be provided with four option bundles (incentives calculated by the mechanism are bonded to the corresponding routes). Choosing a certain route determines the bonded amount of incentives. The incentives are designed such that all vehicles picking the bundles that optimize their individual utilities generate a local system-optimal assignment [21]. However, these benefits come with a price in terms of privacy. The vehicle route assignment and incentive calculation depend on vehicles sharing their values of time. And there seems to be no way to prevent incentive scams in which a vehicle claims to choose the option with the highest incentive but later travels on the route with the shortest travel time.

Figure 2 depicts the conceptual structure of the privacy-preserving incentive mechanism for collaborative routing, which consists of a secure collaborative routing scheme at the origin (or local origin), and a route validation scheme at the checkpoints along the selected route. Both schemes employ on-chain and off-chain operations (depending on whether they need logging information on the blockchain for the record), ensuring that just the bare minimum amount of data are safely stored on the blockchain (against pattern-analysis attacks). The collaborative routing scheme leverages MPC to securely execute the protocols defined by the vehicle route assignment model and incentive model (steps 2 and 3 in the hierarchical framework) in [21]. Users will not receive the incentives corresponding to the routes they choose at the origin. Instead, the incentives will be temporarily frozen, which means they will be sent to a series of multi-signature wallets/accounts from the traffic operator's account. The incentives in a multi-signature wallet require m-out-of-n signatures to become redeemable, where m is the minimum number of signatures required, n is the total number of account holders of the wallet, and both m and n are determined when the wallet is created. Apart from the vehicles receiving these incentives, the account holders of a multi-signature wallet also include one or two verifiers and a mediator. The multi-signature wallets are designed in a cascading manner, such that the frozen incentives can only become redeemable when the user passes the route validation checkpoints along the selected route in order.

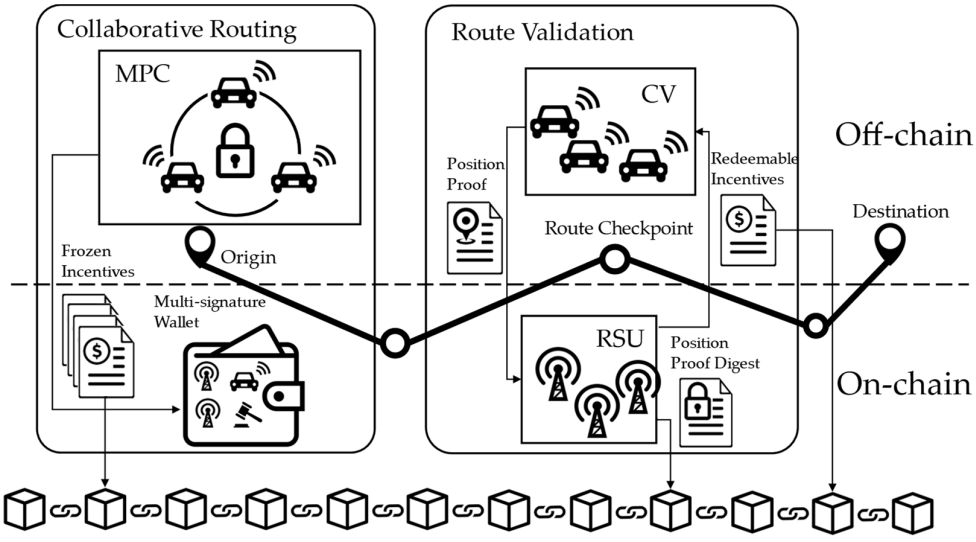

Figure 2. Conceptual structure of the privacy-preserving incentive mechanism for collaborative routing.

When the vehicle is around a checkpoint on the selected route, instead of using GPS information (which can be easily forged by malicious vehicles), it needs to follow the proposed route validation scheme to generate a position proof with the assistance of witness vehicles (nearby vehicles willing to sign on the position proof) and send it to roadside units (RSUs) to verify. The RSUs in the network will verify the proof independently and

reach a consensus following the practical byzantine fault tolerance (PBFT) algorithm. If the position proof is valid, a digest of the proof (which is a fixed-size representation of the contents of the proof) will be included in the current block of a consortium blockchain. Meanwhile, the verifier at the checkpoint will sign two multi-signature transactions: one makes the frozen incentives associated with the current checkpoint redeemable for the user; the other one is for the frozen incentives associated with the next checkpoint, which will remain frozen for the time being and requires one more signature to be redeemable. The user does not have to redeem the redeemable incentives (by signing their own signature on the multi-signature transactions and sending them out) at the time he/she receives them. Note that the transactions will be included in the consortium blockchain only when the incentives are redeemed. Therefore, users can hide their trajectories on the blockchain by redeeming the incentives from the checkpoints from multiple trips in arbitrary order.

Leveraging the tamper resistance of the on-chain information while keeping the sensitive information off-chain, the proposed incentive mechanism eliminates both direct privacy leaks (e.g., sharing the value of time in computation) and indirect privacy exposure (e.g., historical trajectories inferred from the transaction pattern or plain position proofs recorded in the blockchain). It also achieves high-standard security in terms of potentially malicious behavior from both the user side and the infrastructure (i.e., RSUs).

3. Preliminaries

3.1. Hash Functions

Hash functions are fundamental components in cryptography. Ideally, a hash function yields the following properties: (i) it is collision-free; that is, for a hash function $H(\cdot)$, it is infeasible to find x_1 and x_2 such that $H(x_1) = H(x_2)$; (ii) it is hiding, which means, given a hash value $y_1 = H(x_1)$, it is infeasible to find the corresponding x_1; and (iii) it is quick to compute the hash value $H(x)$ for any input x. Therefore, hash functions are handy tools for verifying the integrity of messages transmitted through V2X communications. We can determine whether the message is changed by comparing the hash values of messages (usually with a small and fixed length, and thus labeled message digests) calculated before and after the transmission, no matter how large the raw message is.

3.2. Blockchain

From the data structure perspective, a blockchain is essentially a data block list linked by hash pointers. As shown in Figure 3, each block consists of a hash pointer and block data. Unlike normal pointers, hash pointers not only consist of the storing address of the previous block but also the hash value of its content. Consequently, any changes to the data on the blockchain will result in changes to the hash pointer of the following block, which in turn will result in further changes to the blocks up until the most recent hash pointer. As the manipulation of on-chain data is easily detectable, the on-chain data are deemed immutable. In real-world applications, the data in each block are stored in Merkle trees [30] such that they can be retrieved, and this process also utilizes hash pointers to ensure data integrity.

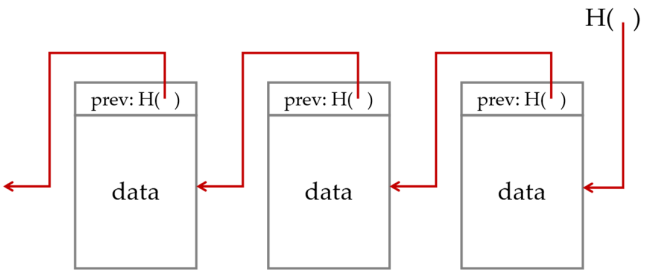

Figure 3. Blockchain and hash pointers (represented by the red arrows).

3.3. Elliptic Curve Cryptography

Elliptic Curve Cryptography is a public-key cryptographic approach based on the algebraic structure of elliptic curves over finite fields [30]. Given a point P (also referred to as the generator) on an elliptic curve (EC) $y^2 = x^3 + ax + b$ and a secret key sk, the public key is generated using the elliptic curve scalar multiplication $pk = sk \cdot P$, which essentially means successively adding P along the elliptic curve [30] to itself repeatedly (which implies that sk is a scalar value while pk is a point). The reliability of ECC is based on the one-wayness of the EC scalar multiplication, which means that it is infeasible to solve sk given pk.

3.4. Encryptions and Digital Signatures

Encryptions can be categorized into symmetric and asymmetric encryptions. In this study, we focus on public-key (asymmetric) encryption; that is, public keys are used for encryption ($e = Enc(m,\ pk)$, where m is the message, Enc is the encryption scheme, and e is the ciphertext), while secret keys are used for decryption: $m = Dec(e,\ sk)$ [31]. In the smart vehicle context, a pair of public and secret keys can be created for each vehicle, with the public keys revealed to all as an identity and the secret keys kept by the vehicle itself. Messages sent to certain vehicles can be encrypted using their public identity/key, such that only the vehicle with the corresponding secret key can decrypt them.

Similar to public-key encryptions, digital signature schemes also use public/secret key pairs to sign signatures and verify signatures. However, in the context of digital signatures, secret keys are used for signing ($sig = Sign(m,\ sk,\ r)$, where sig is the digital signature, $Sign$ is the signing scheme, and r is the randomness added to the message to prevent the signature from being re-used), while public keys are used for verification, $Verify(m,\ pk,\ sig)$ [32]. In this way, vehicles can sign on to the messages they want to disseminate with their secret keys so that anyone who received the messages can validate the authenticity and integrity of messages with the senders' public keys.

3.5. MPC

In this study, MPC based on secret sharing is used. Secret sharing refers to constructing secret shares for each private input of the participants, such that each participant holds parts of secret shares, which contain no meaningful information regarding the original private inputs separately, but, together, can reconstruct the original private inputs [33]. For instance, to create secret shares for one-bit private input $x \in \{0,\ 1\}$, one arbitrary bit is chosen, $x_2 \in \{0,\ 1\}$, then $x_1 = x \oplus x_2$ ($\oplus$ denotes the "xor" binary operation, e.g., $0 \oplus 0 = 0,\ 0 \oplus 1 = 1,\ 1 \oplus 0 = 1$, and $1 \oplus 1 = 0$) and x_2 form valid secret shares of x, as no information of x is inferred with either x_1 or x_2 alone, but together, they can reconstruct x because $x = x_1 \oplus x_2$.

Computing the outputs of a function F with private inputs using secret sharing-based MPC consists of the following steps: (i) represent F as a Boolean circuit C; (ii) generate secret shares of the private inputs of C and disseminate them to all players; (iii) evaluate C gate by gate ("gate" here refers to the Boolean gate), such that secret sharing is valid for each wire ("wires" connect the Boolean gates and transmit the outputs of upstream gates to the downstream gates as inputs); and (iv) reconstruct the function outputs on the output wires. For example, suppose two vehicles want to report to an off-ramp RSU regarding how many vehicles in total are taking the off-ramp without revealing their trips to each other using MPC. They can use a binary adder as shown in Figure 4. The Boolean circuit on the left side of Figure 4 takes two bits x and y as private indicators of whether two vehicles will take the off-ramp. The circuit consists of an "and" gate and an "xor" gate. The output consists of two bits p and z, which can form a binary representation of the total number of vehicles taking the off-ramp. Both x and y can be secret-shared, as Figure 4 shows such that vehicles 1 and 2 hold $(x_1,\ y_1)$ and $(x_2,\ y_2)$, respectively. Here, we illustrate how secret sharing remains valid on the output wire of the "xor" gate. Vehicle 1 applies the "xor" operation on x_1 and y_1 to get $z_1 = x_1 \oplus y_1$ and, similarly, vehicle 2 obtains

$z_2 = x_2 \oplus y_2$. Since $z_1 \oplus z_2 = (x_1 \oplus y_1) \oplus (x_2 \oplus y_2) = (x_1 \oplus x_2) \oplus (y_1 \oplus y_2) = x \oplus y = z$; that is, we can reconstruct z with z_1 and z_2 while inferring no information about z solely with z_1 or z_2. Therefore, secret sharing holds for the "xor" gate. And secret sharing for the "and" gate also exists [34], though it is not as intuitive as that for the "xor" gate. Therefore, vehicles 1 and 2 can send all the secret shares (p_1, z_1) and $(p_2, z_{2)}$, respectively, to the RSU to compute the total number of vehicles taking the off-ramp.

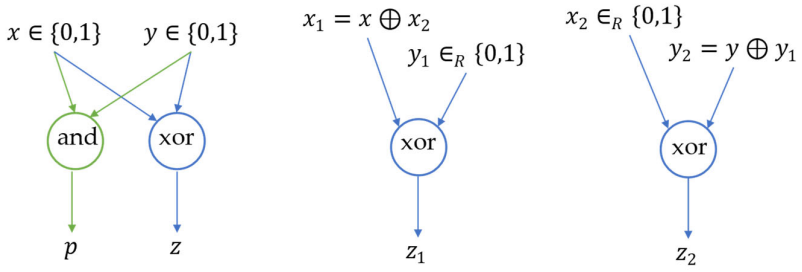

Figure 4. Example of secret sharing-based MPC.

For more complicated F, as in our proposed schemes, there are MPC compilers that can generate the corresponding circuit C, which is how we generate the MPC protocols in our numerical studies.

4. Privacy-Preserving Incentive Mechanism

Before presenting the details of the proposed incentive mechanism, this section starts with an introduction of the main entities: a trusted authority (TA), CVs, and RSUs, and how they are involved in the proposed incentive mechanism.

- Trusted Authority: The TA plays two essential roles in the proposed mechanism: the identity manager ($\mathcal{I}$) is responsible for generating identities (public key and private key pairs) and relating vehicles' pseudonyms to their real identities, and the mediator ($\mathcal{M}$) unfreezes the frozen incentives to either refund the traffic operator when users behave dishonestly or transact the incentives to users when RSUs behave maliciously (are hacked). Therefore, TA is assumed to be fully secure and trusted. Note that though both $\mathcal{I}$ and $\mathcal{M}$ function in a centralized manner, little computation or communication burden is introduced because vehicles only request identities once with $\mathcal{I}$ when they participate in the mechanism for the first time. $\mathcal{M}$ only interacts with vehicles and RSUs under malicious behaviors, which are rare because the malicious party can be traced and penalized.
- CVs: Each CV is assumed to be able to communicate with RSUs and nearby vehicles using V2X technology. There are initiators/leaders $\mathcal{L}^R$, $\mathcal{L}^V$, which initiate the collaborative routing scheme and the route validation scheme, respectively, and corresponding followers $\mathcal{F}^R, \mathcal{F}^V$ The study does not consider attacks in the communication process, implying communication channels are assumed to be secure. Also, each vehicle is assumed to have a tamper-resistant device with which to store sensitive information, including a secret key and value of time securely (vehicle hardware side attacks are not considered).
- RSUs: RSUs also play multiple roles in the proposed mechanism. Some RSUs serve as checkpoint signers ($\mathcal{S}$) in the route validation protocol, signing on multi-signature transactions when vehicles pass the validation at the checkpoints. And all RSUs are the nodes of the consortium blockchain, with some authorized RSUs ($\mathcal{V}$) verifying the incentive transactions and position proofs and undertaking the consensus work to generate new blocks. The RSUs are semi-trusted and can be potentially malicious. However, we assume that only a small percentage of RSUs are malicious, which is widely accepted in other consortium blockchain applications [35,36].

4.1. Collaborative Routing Scheme

The collaborative routing scheme consists of an off-chain MPC and an on-chain incentive freezing process. The off-chain MPC takes the optimal route flows as public inputs and the value of time of vehicles in the local vehicle group as private inputs to generate route suggestions and associated incentive amounts for vehicles in the group. With the envy-freeness analysis in [21], rational users will always choose the suggested routes. After confirmation from users, the incentives will be frozen in a series of cascading multi-signature wallets for each vehicle until they pass further route validations.

First, we describe how $\mathcal{G}$ generates pseudonyms for vehicles that participate in the mechanism for the first time using an Elliptic Curves Cryptography (ECC) based combined-public key (CPK) scheme. Identity-based CPK derives public keys (pseudonyms) from real-world identities; hence, it does not require certificates as traditional public key encryptions, reducing the key management burden [37]. In our case, the public keys are derived from the VIN (Vehicle Identification Number) of vehicles. The identity generation process is as follows:

1. Select an elliptic curve $\mathcal{C}$. Let $\mathcal{H}$ be an addition group of points on $\mathcal{C}$, and let P be the generator of $\mathcal{H}$. $q \in \mathcal{H}$ is an order of $\mathcal{H}$ (most encryption/signature schemes in this study are based on ECC; please refer to [16] for ECC basics).
2. Generate an m-length master private key vector $\mathcal{X} = (x_1, x_2, \ldots, x_m)$, where x_i are randomly selected from $\mathbb{Z}_q$.
3. Generate the corresponding public key vector $\mathcal{Y} = (y_1, y_2, \ldots, y_n)$, where $y_i = x_i \cdot P$.
4. Using $\mathcal{X}$ and $\mathcal{Y}$, generate the private key and public key for each vehicle as follows:

$$sk_{ID} = \sum_{i=1}^{m} h_i(ID)x_i \bmod q, \tag{1a}$$

$$pk_{ID} = \sum_{i=1}^{m} h_i(ID)y_i, \tag{1b}$$

   where $h_i(ID)$ is the i th bit of the digest of the vehicle's VIN generated by $H_0 : \{0, 1\}^* \to \{0, 1\}^m$. It is trivial to see that $pk_{ID} = sk_{ID} \cdot P$ holds, i.e., (pk_{ID}, sk_{ID}) is a valid pair of ECC keys.
5. Send vehicles their private keys through secure communication channels along with the following information as the public parameters of the cryptographic system $(\mathcal{H}, q, \mathcal{Y}, (H_0, H_1, H_2, H_3, H_4, H_5, H_6), E)$, where E is a symmetric encryption protocol (given cyphertext $y = E_k(x)$; we can obtain the plaintext $x = E_k^{-1}(y)$), and H_i represents hash functions used in the schemes.

Using the pseudonyms, vehicles can initiate and participate in the collaborative routing scheme, which can be described as follows:

1. One vehicle (defined as the leader of the collaborative routing scheme $\mathcal{L}^R$) initiates a request for collaborative route updating by sending nearby vehicles the message $\{ID_L, des, sig_L\}$, where ID_L is the VIN of $\mathcal{L}^R$, des is the destination identifier, and $sig_L = sign(sk_L, H_1(des))$ is the signature that $\mathcal{L}^R$ signs using its secret key sk_L on the digest of des, $H_1(des)$.
2. Vehicles heading to the same destination and interested in joining the collaborative routing scheme (defined as the followers of the collaborative routing scheme $\mathcal{F}^R$), after verifying the request ($isValid(pk_L, des, sig_L)$, each vehicle can reply a signed bit 1, $sig_{ID} = sign(sk_{ID}, 1)$ together with its VIN ID to $\mathcal{L}^R$.
3. $\mathcal{L}^R$ collects the responses from $\mathcal{F}^R$, verifies the responses ($isValid(pk_{ID}, 1, sig_{ID})$, and counts the total number of vehicles participating in this session, n_s (which is the demand of this specific OD pair).
4. $\mathcal{L}^R$ reports n_s to the RSU nearby, which will update the flows related to this OD pair iteratively together with other RSUs in a distributed manner following the route flow assignment model in [21].
5. At the same time, $\mathcal{L}^R$ and $(n_s - 1)$ $\mathcal{F}^R$ start establishing the communication network for the MPC protocol. $\mathcal{L}^R$ produces a participation confirmation message,

$\{< ID_1, ID_2, \ldots, ID_{n_s} >, < sig_{ID_1}, sig_{ID_2}, \ldots, sig_{ID_{n_s}} >\}$, which generates an order for all participants.

6. After $\mathcal{F}^R$ receive the confirmation message, they start creating secret shares of their value of time. λ_{ID_i} is denoted as the value of time of the ith vehicle ($i \in [1, 2, \ldots, n_s]$ in the confirmation message. The vehicle creates secret shares s_{ij}, $j \in [1, 2, \ldots, n_s]$ for the jth vehicle and sends it.
7. After the RSU receives optimal route flows, it broadcasts the information as public inputs of the MPC protocol.

Note that step 6 takes the most time out of the entire process as there are $n_s(n_s - 1)$ messages sent. However, this happens while RSUs are solving for the optimal flows, which is also the most time-consuming step in the hierarchical framework in [21], which mitigates the influence of step 6′s relatively long computation time.

With all required private and public inputs, the vehicles can execute the MPC protocol to produce the private outputs, which consists of their updated routes and corresponding incentives. However, MPC protocols are pre-compiled, which means that they have a fixed number of inputs, while the number of vehicles participating in the collaborative routing scheme varies in the real world. Hence, the vehicle assignment model and incentive mechanism proposed in [21] are modified as follows. Assume that the MPC protocol requires N vehicles to collaboratively update their routes (N is the maximum number of participants allowed in the scheme, determined by step 6 in practice). According to *Lemma 3* in [21], the vehicle assignment is to sort vehicles' value of time. We can create $(N - n_s)$ fake vehicles with zero value of time to complement the number of inputs required by the MPC protocol. In this way, the updated routes of the participants are the same as the ones they are supposed to obtain in the vehicle route assignment model. When determining the incentives, the protocol assumes that fake vehicles take a fake route with the same travel time as the longest travel time of all real routes. According to *Lemma 4* in [21], the adjustment incentives of the fake vehicles are zero and the real vehicles' adjustment incentives are the same as those they are supposed to obtain from the incentive mechanism in [21]. The details of the MPC protocol for route/incentive assignment (Algorithm 1) are described as follows.

Algorithm 1. MPC protocol for route/incentive assignment

Private input: individual value of time $\lambda_i, i = 1, 2, \ldots, n_s$.
Public input: travel times and optimal route flows for each route T_k, f_k, $k \in K$, K is the route id set ($\sum f_k = n_s$).
Public output: incentives p_k for each route $k \in K$.

1. Sort λ_i (denote smallest as $\lambda_{\min}$ and T_k add $N - n_s$ vehicles with $\lambda_i = 0$, and add $N - n_s$ flow to route with largest travel time.
2. Duplicate T_k for f_k times such that there are N travel times in total.
3. Assign the vehicle with the rth largest value of time, $\lambda^{(r)}$, to the route with rth shortest travel time $T^{(r)}$ (denoted as $\eta^{(r)}$).
4. Compute $p^{(1)} = \frac{1}{n_s}\sum_{j=2}^{N}\sum_{m=2}^{j}\lambda^{(m)}\left(T^{(m)} - T^{(m-1)}\right)$, and $p^{(r)} = \frac{1}{n_s}\sum_{j=2}^{N}\sum_{m=2}^{j}\lambda^{(m)}\left(T^{(m)} - T^{(m-1)}\right) - \sum_{m=2}^{r}\lambda^{(m)}\left(T^{(m)} - T^{(m-1)}\right)$ as if there were N vehicles.

Return $\eta^{(r)}$ and $p^{(r)}$ to the vehicle with $\lambda^{(r)}$.

The MPC protocol also generates the outputs required for the traffic manager to freeze incentives. Figure 5 shows an example of the process of freezing incentives. The route that the vehicle takes has four checkpoints A, B, C, and D. The amount of incentives that the vehicle receives for this trip is divided into four parts a_A, a_B, a_C, and a_D, which correspond to the four segments of the route divided by the checkpoints. To ensure that the vehicle follows the route, the traffic manager does not send the incentives to the vehicle directly at the origin. Instead, it sends the segment incentives to a series of cascading

multi-signature wallets, which require multiple signatures to be authorized to transfer. The wallet corresponding to the first checkpoint is a two-out-of-three wallet, which requires at least two signatures from three wallet holders: the signer at checkpoint A, $\mathcal{S}_A$, the vehicle, and the mediator $\mathcal{M}$. In normal operations, $\mathcal{S}_A$ signs on the transaction after the vehicle passes the route verification at checkpoint A, which makes the incentives associated with segment OA redeemable for the vehicle; it can sign on the transaction to meet the two-out-of-three signature requirement when it wants to redeem the incentives. $\mathcal{M}$ only plays a role when malicious behaviors are detected. Either the traffic manager or the vehicle can submit evidence to let $\mathcal{M}$ sign on the transaction to either refund the frozen incentives to the traffic manager or transmit the frozen incentives to the vehicle. The other wallets are three-out-of-four wallets, which require at least three signatures from four wallet holders: the signer at the upstream checkpoint, the signer at the current checkpoint, the vehicle, and the mediator.

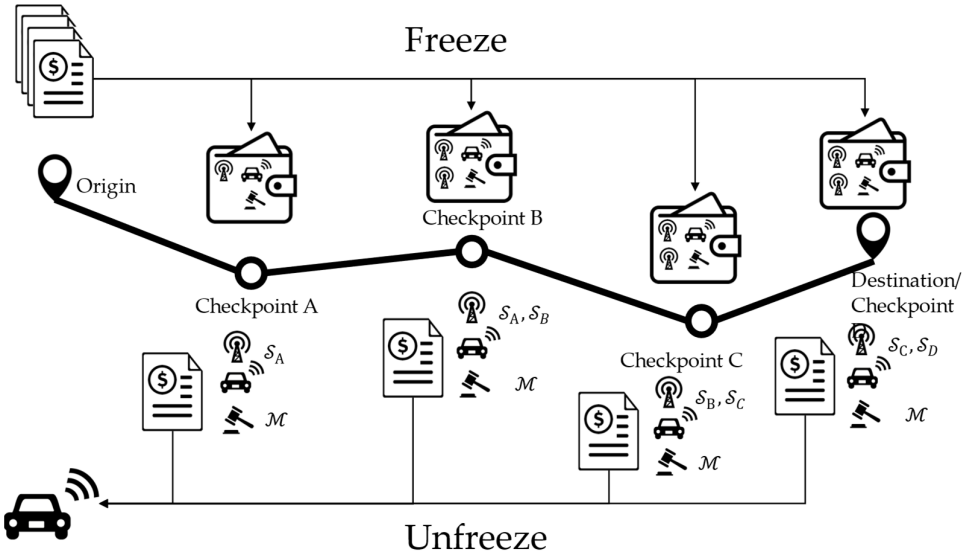

Figure 5. Freezing and unfreezing incentives in cascading multi-signature wallets.

Notably, the secret shares of the private inputs and outputs of Algorithm 1 can be fed into Algorithm 2 to skip the step of generating secret shares of the inputs of Algorithm 2. The outputs from Algorithm 2 are a series of cascading incentive freezing transactions to be signed by the traffic manager. They are marked as public because the signed transactions will be published on-chain to freeze the incentives for each segment in the corresponding multi-signature wallets. Since the signed transactions are recorded in the blockchain, the contents of Add_{s_j} are open to everyone. However, the sensitive trip information is protected twofold. First, it is almost impossible to tell which transaction is generated for which segment incentives for whose trip, because the address of a multi-signature wallet does not explicitly show the wallet holders' identities. As (3) shows, it is a hash of a piece of code. The public keys of the involved vehicle, checkpoints, and the mediator only appear in the code. Recall that hash functions are hiding; it is impossible to reconstruct the code using the wallet address (which is the hash of the code). It is also hard to enumerate the combinations of the wallet holders if each RSU/checkpoint has tens of valid identities registered at $\mathcal{I}$. Second, each trip is divided into multiple segments, which makes identifying the entire trip of a certain vehicle exponentially harder, since it requires unhashing the wallet addresses of all involved transactions.

Algorithm 2. MPC protocol for incentive freezing

Private input: individual route choices $\eta_i \in K, i = 1, 2, \ldots, n_s$
Public input: incentives for route k denoted as p_k, lengths l_{s_i}, $s_i \in S_k$ where S_k is the segment set of route k, and PK_{c_j}, the public key set of signer $\mathcal{S}_{c_j}$ at checkpoint $c_j \in C_k$ along route $k \in K$, where C_j is the checkpoint set along route k.
Public output: transactions $TRANS_{s_i}$, $s_i \in S_k$, $k \in K$, which are to be signed by the traffic manager to freeze segment incentives.

1. For each individual $i = 1, 2, \ldots, n_s$: calculate

$$p_{s_j} = \frac{l_{s_j}}{\sum_{s_j \in S_{\eta_i}} l_{s_j}} p_{\eta_i},\ s_j \in S_{\eta_i}. \tag{2}$$

2. To freeze segment incentive p_{s_j}, generate the following transaction $TRANS_{s_j}$.
   From: the traffic manager's address (i.e., its public key)
   To: the address of the multi-signature wallet

$$Add_{s_j} = H_2(script(pk_i,\ pk_m,\ pk_r,\ pk_m)),\ s_j \in S_{\eta_i}, \tag{3}$$

where $pk_r \in PK^b_{s_j}$, $pk_m \in PK^e_{s_j}$ ($PK^b_{s_j}$ and $PK^e_{s_j}$ are the public key sets of the signer at the beginning and end of segment s_j, respectively), and $script(\cdot)$ is the payment script that is used to validate the transaction.
   Amount: p_{s_j}
3. **Return** Add_{s_j}, $s_j \in S_{\eta_i}$ to vehicle i.

4.2. Route Validation Scheme

To verify that the vehicle is at a certain checkpoint CP, $\mathcal{S}_{CP}$ needs a position proof. Instead of using traditional GPS information (which can be easily tampered with), the position proof is generated by the witness vehicles at checkpoint CP. A reasonable assumption entails the presence of a sufficient number of vehicles in proximity to the checkpoints. Consequently, the approach delineated in this section holds significance under congested conditions, where V2X communication and privacy protection are predominantly required. To protect the privacy of the witness vehicles, threshold cryptographic techniques are used in the route validation scheme. Specifically, the signatures of real witness vehicles are mixed with other fake signatures to protect their pseudonyms from being tracked. The route validation scheme can be divided into three stages: a vehicle needing a position proof initiating a request for route validation, witness vehicles replying to the request, and verifiers validating the position proof. The first stage is as follows.

1. The vehicle that wants to prove that it is at checkpoint CP (defined as the leader of the route validation scheme $\mathcal{L}^V$) generates a plaintext pos describing its current position (e.g., "on xxx link near checkpoint CP at time yyy").
2. $\mathcal{L}^V$ requests the number of witnesses required, n_w, from nearby RSUs and calculates the total number of signatures on the position proof as $n_p = \lceil n_w/\eta \rceil$, where $\eta \in (0, 1]$ is the privacy protection parameter (for larger η, the pseudonyms of witness vehicles are mixed with fewer fake identities, and thus they will receive less privacy protection). That is, there should be at least n_w nearby vehicles that witness $\mathcal{L}^V$'s presence at CP and they should sign on the position proof. Meanwhile, there should be another $n_p - n_w$ fake signature on the position proof as well to protect the privacy of these witness vehicles. Otherwise, anyone can learn from the position proof that these vehicles themselves are near CP at this specific time. Therefore, $\mathcal{L}^V$ needs to generate and include $n_p - n_w$ fake signatures in the route validation request such, that the witness vehicles can generate signatures that cannot be distinguished from the fake signatures.
3. $\mathcal{L}^V$ generates $n_p - n_w$ fake IDs from the feasible ID set, $\Omega_f = \left\{ID_1, ID_2, \ldots, ID_{n_p - n_w}\right\}$ and the corresponding public keys, $pk_{ID_j} = \sum_{i=1}^{m} h_i(ID_j) y_i$, $\forall j \in \left\{1, 2, \ldots, n_p - n_w\right\}$.

4. $\mathcal{L}^V$ generates the fake signatures for the fake vehicles. For $ID_i \in \Omega_f$, it selects $a_i,\ b_i \in \mathbb{Z}_q$ and computes:

$$A_i = a_i \cdot P + b_i \cdot pk_{ID_i}, \tag{4a}$$

$$\beta_i = -b_i^{-1} A_i[x], \tag{4b}$$

$$m_i = \alpha_i \beta_i, \tag{4c}$$

where $A_i[x]$ is the x coordinate of point A_i. Note that $(A_i,\ \beta_i)$ is a valid EC Elgamal signature of m_i, because $m_i \cdot P = A_i[x] \cdot pk_{ID_i} + \beta_i \cdot A_i$. The fake signatures are created but $\mathcal{L}^V$ has no control over the corresponding m_i.

5. $\mathcal{L}^V$ generates $n_p - n_w$ different indexes $\kappa_i = H_3(A_i) \in \mathbb{Z}_q,\ i \in \{1,\ 2,\ \ldots, n_p - n_w\}$ for further Lagrange polynomial interpolation.
6. $\mathcal{L}^V$ initiates the request of route validation at checkpoint CP by sending the following message to the nearby vehicles: $\left\{(ID_i,\ m_i,\ \kappa_i)_{i=1,\ 2,\ \ldots,\ n_p - n_w},\ pos,\ n_p,\ n_w\right\}$

Upon receiving the route validation request, nearby vehicles that identify $\mathcal{L}^V$ at checkpoint CP will reply to the request with a signed message back, becoming witness vehicles in $\mathcal{L}^V$'s position proof. A witness vehicle $\mathcal{F}^V$ generates its signature as follows:

1. $\mathcal{F}^V$ constructs a polynomial f with $n_p - n_w$ degrees defined on Galois field $GF(2^{n_l})$ $(A_i[x] \in GF(2^{n_l}))$ using Lagrange interpolation, such that $f(0) = H_4(des)$ and $f(\kappa_i) = E_k(m_i),\ i = 1,\ 2,\ \ldots,\ n_p - n_w$, where $k = H_5(n_p||n_w)$.
2. $\mathcal{F}^V$ chooses a random index $\kappa \notin \left\{\kappa_1,\ \kappa_2, \ldots,\ \kappa_{n_p - n_w}\right\}$ and generates $m = E_k^{-1}(f(\kappa))$.
3. $\mathcal{F}^v$ randomly selects $c \in \mathbb{Z}_q$ and generates the EC Elgamal signature $(A,\ \beta)$ of m, where $A = c \cdot P,\ \beta = (m - sk \cdot A[x])c^{-1}$.
4. $\mathcal{F}^v$ replies to the route validation request by sending a message, $\{ID,\ m,\ \kappa,\ (A,\ \beta)\}$, to $\mathcal{L}^V$.

Once $\mathcal{L}^V$ receives more than n_w responses from the nearby vehicles, it aggregates the fake and collected signatures to generate a position proof $\mathcal{P}$ and sends it to the verifiers $\mathcal{V}$:

$$\mathcal{P} = \left\{(ID_i,\ m_i,\ \kappa_i,\ (A_i,\ \beta_i))_{i=1,\ 2,\ \ldots,\ n_p},\ pos,\ n_p, n_w\right\}.$$

Each $\mathcal{V}$ then verifies the position proof $\mathcal{P}$ as follows:

1. $\mathcal{V}$ generates the public keys $pk_{ID_j} = \sum_{i=1}^{m} h_i(ID_j) y_i, \forall j \in \{1,2,\ldots,n_p\}$.
2. $\mathcal{V}$ verifies whether the signatures (A_i, β_i) are valid by checking whether $m_i \cdot P = A_i[x] \cdot pk_{ID_i} + \beta_i \cdot A_i$ holds for $i \in \{1,2,\ldots,n_p\}$.
3. $\mathcal{V}$ randomly selects $n_p - n_w$ tuples from $(ID_i, m_i, \kappa_i, (A_i, \beta_i))_{i=1,2,\ldots,n_p}$, reconstructs the polynomial f using Lagrange interpolation, such that $f(0) = H_4(des)$ and $f(\kappa_j) = E_k(m_j)$, where $k = H_5(n_p||n_w)$, and verifies whether $f(k_i) = E_k(m_i)$ holds for all $i \neq j$.
4. $\mathcal{V}$ accepts the position proof if it passes all verifications.

To prevent malicious behavior by a single $\mathcal{V}$, the position proof is sent to all $\mathcal{V}$ in the RSU network, and the PBFT algorithm [38] is used to generate a consensus mechanism. The presence of malicious nodes will not impact the final consensus when the number of malicious nodes is less than one-third of the participating nodes. When consensus is achieved that the position proof is valid, the digest of the position proof, $H_6(\mathcal{P})$, instead of the plaintext $\mathcal{P}$, is written to the latest block of the blockchain. $\mathcal{S}_{CP}$ will sign on the unfreezing transaction after verifying that the position is in the blockchain (which indicates that the position proof has been verified by the majority of the verifiers).

4.3. Privacy and Security Analysis

To ensure that the proposed schemes can prevent privacy leaks and ensure consistency between the routes that vehicles choose and the ones they take, this section analyzes the

potential privacy leaks in the scheme steps and discusses how malicious behavior can be managed in the incentive mechanism.

First, the proposed schemes are privacy-preserving in both the collaborative routing and the route validation processes. In the collaborative routing scheme, $\mathcal{L}^R$ and $\mathcal{F}^R$ hide their real identities with pseudonyms when sending messages. Vehicles receiving messages only know that some vehicles are heading to *des* but cannot connect these messages to nearby vehicles. Also, as the messages are kept off-chain, it is hard to connect the pseudonyms to real identities through pattern analysis. The messages sent to RSUs are aggregated information, n_s. Therefore, no individual privacy information is leaked when RSUs compute optimal route flows. When computing updated routes and incentives thatfollow the MPC protocol, vehicles only send secret shares of their value of time to other vehicles, and outputs are generated by each vehicle using these shares. No meaningful information can be inferred from incomplete shares. In the on-chain incentive freezing process, incentives are sent to different multi-signature wallets. Note that $\mathcal{S}$ of multi-signature wallets do not need to be the RSUs near the corresponding checkpoints (because verifying position proofs is a cryptographic process independent of the RSU position). Therefore, wallet addresses only provide random signers' public keys, which cannot be used for privacy pattern analysis. In route validation schemes, the pseudonyms of $\mathcal{L}^V$ and $\mathcal{F}^V$ are mixed with fake pseudonyms in position proofs, which provides additional privacy protections, as position proofs are sent to all RSUs for PBFT consensus. Before logging into the blockchain, the position proofs are hashed to ensure no position/route information can be inferred from pattern analysis of the information in the blockchain.

Behavioral honesty can also be illustrated for both the collaborative routing and the route validation processes. Given that the MPC protocol ensures no privacy leakage risks, vehicles participating in collaborative routing have no privacy concerns. And [21] has shown that under this condition, vehicles will behave honestly (i.e., provide genuine inputs to the collaborative routing scheme) to maximize their own utilities. In the route validation scheme, first, the cryptographic tools used in the scheme design mitigate common types of attacks. The identity-based asymmetric key generation mitigates Sybil attacks, in which a single entity operates multiple fake identities simultaneously to undermine the system by gaining the most influence in the network. Also, the EC digital signature algorithm widely applied in the proposed scheme ensures that signatures cannot be forged, and messages are tamper-resistant. Therefore, adversaries cannot launch replay attacks in the route validation scheme by re-sending messages they received before. Also, $\mathcal{L}^V$ cannot generate a fake position proof by forging more than $n_p - n_w$ signatures. Because $\mathcal{L}^V$ has no control over the m_i corresponding to fake signatures, it can forge at most $n_p - n_w$ signatures to identify $n_p - n_w$ points on $GF\left(2^l\right)$ (there is an additional point $(0,\ H_4(des))$) and determine a polynomial with degree $n_p - n_w$. If it forges more signatures, it cannot ensure that the corresponding m_i is on the polynomial, which can be easily detected by $\mathcal{V}$.

5. Numerical Studies

Simulation studies are conducted to illustrate the performance of the privacy-preserving incentive mechanism. First, we show the correctness of the proposed incentive mechanism. Then, the computational efficiency of the collaborative routing scheme and the route validation scheme is analyzed under different privacy protection settings. The MPC protocols are implemented in SCALE-MAMBA (https://github.com/KULeuven-COSIC/SCALE-MAMBA (accessed on 9 January 2024)) and MP-SPDZ (https://github.com/data61/MP-SPDZ (accessed on 9 January 2024)), and the route validation scheme is implemented using Python.

To validate the correctness of the MPC protocol, the example network (see Figure 6) in [21] is used to illustrate that the MPC protocol can calculate the same incentives with proper settings. Twenty vehicles depart from node 13 to node 16 in the network. There are three local destinations, nodes 21, 22, and 23, and four alternative routes connecting nodes 13 and 16, as listed in Table 1 and illustrated in Figure 6. The desired flows and the

corresponding travel costs of the four routes are provided by the route flow assignment model in [21]. The values of time () for the individual vehicles are shown in Table 2. With these inputs, the implemented MPC scheme can generate the same vehicle route assignment results η_i, $i = 1, \ldots, 20$ (the id of the route that vehicle i should take) as in [21]. In terms of the incentives, when the integer representation precision of the MPC protocol is set as 15-bit fixed point numbers with a 5-bit decimal part, the output incentives $p_i^{(5,\,15)}$ are different from the ones calculated in [21] (although $\sum_{i=1}^{20} p_i^{(5,\,15)} = 0$ holds as $\sum_{i=1}^{20} p_i = 0$). If the precision is increased to 31-bit fixed point numbers with a 16-bit decimal part, the output incentives $p_i^{(16,\,31)} = p_i$, $i = 1, \ldots, 20$ (i.e., the implemented MPC scheme can calculate the same incentives as in [21] under this setting). The total data exchanged among the vehicles when executing the MPC scheme increase from 135.478 MB to 180.649 MB in this case, which is acceptable in the real world.

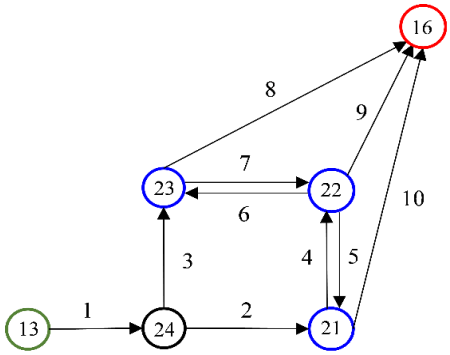

Figure 6. Example local road map from [21] (the blue circles are local destinations; the green and red circles are the origin and final destination; and the link/node IDs are denoted in the figure).

Table 1. Alternative routes with desired route flows and travel costs.

Route ID	Links	Flow	Travel Cost
1	1-2-10	8	289.961
2	1-2-4-9	2	290.086
3	1-3-7-9	1	331.512
4	1-3-8	9	331.500

Table 2. Vehicle incentives comparison.

Vehicle ID	λ_i	η_i	a_i	p_i	$p_i^{(5,15)}$	$p_i^{(16,31)}$
1	0.80	1	0.000	−12.401	−12.375	−12.401
2	0.91	1	0.000	−12.401	−12.375	−12.401
3	0.45	4	24.786	12.385	12.313	12.385
4	0.46	4	24.786	12.385	12.313	12.385
5	0.72	1	0.000	−12.401	−12.375	−12.401
6	0.64	2	0.080	−12.321	−12.281	−12.321
7	0.54	4	24.786	12.385	12.313	12.385
8	0.84	1	0.000	−12.401	−12.375	−12.401
9	0.61	2	0.080	−12.321	−12.281	−12.321
10	0.42	4	24.786	12.385	12.313	12.385
11	0.60	4	24.786	12.385	12.313	12.385
12	1.00	1	0.000	−12.401	−12.375	−12.401
13	0.40	4	24.786	12.385	12.313	12.385
14	0.43	4	24.786	12.385	12.313	12.385
15	0.87	1	0.000	−12.401	−12.375	−12.401
16	0.76	1	0.000	−12.401	−12.375	−12.401
17	0.23	4	24.786	12.385	12.313	12.385
18	0.71	1	0.000	−12.401	−12.375	−12.401
19	0.49	4	24.786	12.385	12.313	12.385
20	0.15	3	24.788	12.387	12.313	12.387

Next, we validate the computational efficiency of the collaborative routing scheme. Due to our modifications (fake vehicles and routes) to the incentive mechanism, the compiled MPC protocol can be executed by fewer vehicles than the predefined number of parties, N. Figure 7 shows how the computational time of different stages changes with the number of participating vehicles (n_s) given a compiled 8-party MPC protocol ($N = 8$). The computational times of the input and output stages increase with n_s because fake vehicles' inputs are pre-determined and do not require outputs. The computation time for the computation stage slightly increases as the number of fake vehicles decreases.

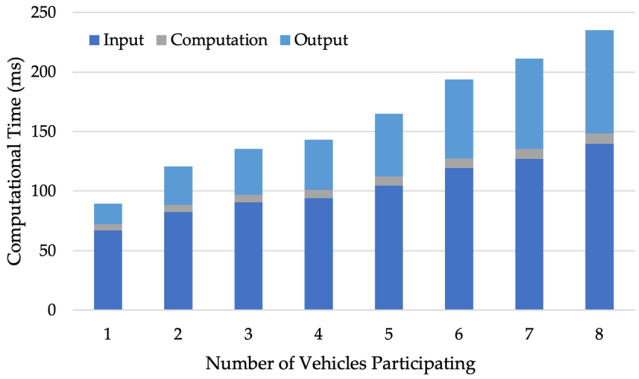

Figure 7. Computational times of the collaborative routing scheme.

Figure 8 shows how the computational time of the output stage changes with N and n_s ($n_s \leq N$). It increases significantly with an increase in the number of MPC parties. To enable real-time implementation, the vehicle group size should be limited to 10. When more than 10 vehicles want to participate, they can be assigned into multiple vehicle groups with the same OD. The flow updating model can accommodate such settings. Also, the simulations were conducted on one desktop with one thread, while in real implementation, outputs can be generated parallelly on all participating vehicles, which can reduce the total computational time as well. Given the exponential increase in the MPC scheme computational time with the increase in MPC parties, a potential refinement is to subdivide the collaborative routing problem into smaller portions. This process would enable the MPC schemes to be executed with a reduced number of parties involved in each smaller subproblem. The results obtained from these smaller subproblems can then be aggregated by another MPC scheme. Structuring MPC in this hierarchical way is likely to address the computational issue associated with a large number of MPC parties.

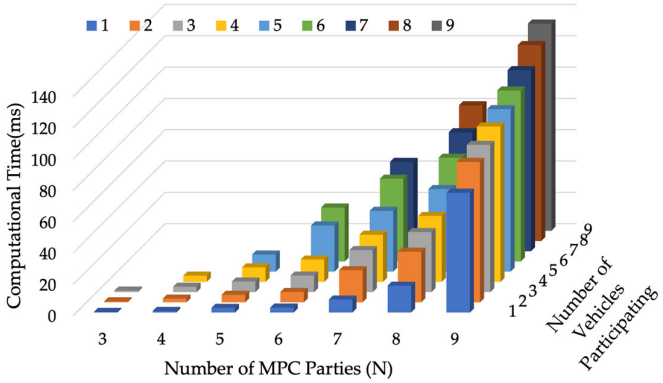

Figure 8. MPC output generation times under different N and n_s.

Next, the computational feasibility of the route validation scheme is evaluated. The results in Figure 9 seem counter-intuitive at first glance; the time the leader vehicle takes to generate the request, the time that witness vehicles take to reply to the leader, and the total

computational time all decrease as the number of witness vehicles increases. Since n_p is fixed, an increase in the number of witness vehicles will result in a decrease in the number of fake signatures that the leader generates; thus, the leader request time is reduced. Also, when witness vehicles generate the replies, the most time-consuming step is constructing the polynomial based on all fake signatures, which takes more time as the number of fake signatures increases. Therefore, the reply times of witness vehicles indicated by the yellow bars in Figure 9 decrease as the number of witness vehicles increases.

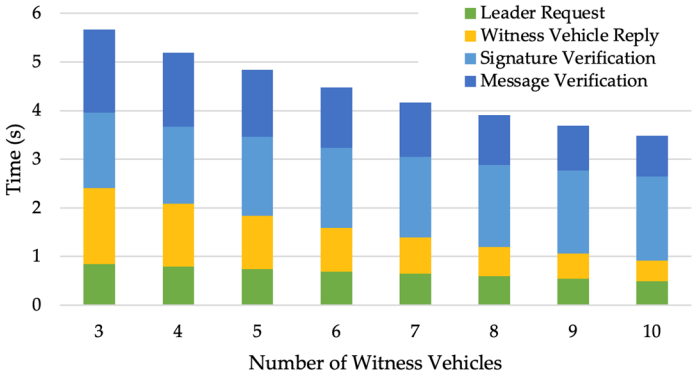

Figure 9. Computational times of the route validation scheme with different numbers of witness vehicles when $n_p = 20$.

To evaluate how the value of the privacy protection parameter η influences the efficiency of the route validation scheme, we compare the computational time of simulations with different η when the number of required witness vehicles $n_w = 5$. Figure 10 shows that the computational time decreases significantly as η increases, especially the time that witness vehicles take to generate a reply message and the message verification time of $\mathcal{V}$.

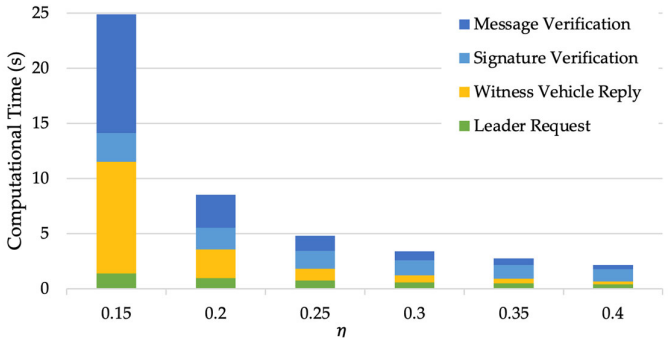

Figure 10. Computational times of the route validation scheme under different values of privacy protection parameter η.

It should be noted that the collaborative routing scheme and route validation scheme encompass a variety of critical cryptographic operations, including the EC Elgamal key generation algorithm and signature algorithm. The time complexity of these components is substantially influenced by the parameters of the cryptographic settings, such as the order q of the addition group $\mathcal{H}$. This section predominantly focused on analyzing the effects of parameters that hold greater relevance to the transportation context. In practical applications, the selection of these parameters must strike a balance between privacy security and computational efficiency. Securing privacy is certainly a crucial aspect, but the emphasis should equally be on computational feasibility, especially for vehicular applications. Generally, enhancing privacy protection could imply a potential trade-off in computational efficiency. For instance, using a smaller η affords better privacy protection for witness vehicles, as more fake signatures are blended into the position proof. However, this could concurrently increase the computational times, as indicated in Figure 10, thus

impacting the route validation scheme's efficiency. Hence, a careful trade-off must be enabled in practice.

6. Conclusions

This study offers several significant contributions. First, collaborative routing facilitates personalization by accounting for user heterogeneities, leading to increased privacy concerns. To address this issue, the proposed method combines MPC with collaborative routing, thereby enabling privacy-preserving collaboration and showcasing a potential solution to privacy concerns associated with personalization. Second, the study introduces a V2V-based position proof approach as an alternative to the widely used GPS, which has raised concerns regarding the sharing of privacy-sensitive information. This alternative allows users to verify their travel history without disclosing their historical positions, a characteristic that has not been achieved previously. Third, the study presents a novel on-chain/off-chain structure that capitalizes on the tamper-resistance property of on-chain data while maintaining sensitive privacy pattern information off-chain. This design offers valuable insights into harnessing the benefits of blockchain technology while circumventing privacy risks associated with its inherent transparency. It should be emphasized that the suggested approach extends significantly beyond the scope of [21], as the primary objective of the current study is to address potential privacy breaches arising from personalization within transportation systems. The MPC framework may be adapted to alternative application contexts involving personalized demand-side solutions. Furthermore, the position verification technique can be employed in additional applications necessitating the sharing of travel history, thereby safeguarding user privacy.

Potential directions for future research include: (i) making position proofs reusable for witness vehicles to reduce duplication of verification; (ii) refining the MPC structure to allow more vehicles in one vehicle group; and (iii) incorporating the incentive mechanism into the broader intelligent transportation system to form a sustainable incentive ecosystem, where users can spend the incentives they gain, such that pseudonyms do not need to be connected to bank accounts, further protecting privacy.

Author Contributions: Conceptualization, C.W. and S.P.; methodology, C.W.; software, C.W.; validation, C.W.; formal analysis, C.W.; investigation, C.W. and S.P.; resources, S.P.; data curation, C.W.; writing—original draft preparation, C.W.; writing—review and editing, S.P.; visualization, C.W.; supervision, S.P.; project administration, S.P.; funding acquisition, S.P. All authors have read and agreed to the published version of the manuscript.

Funding: This study is supported by the National Science Foundation (NSF) Smart and Connected Communities (SCC) grant #12125390, and by Georgia Institute of Technology. Any errors or omissions remain the sole responsibility of the authors.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Göransson, L.; Goop, J.; Unger, T.; Odenberger, M.; Johnsson, F. Linkages between Demand-Side Management and Congestion in the European Electricity Transmission System. *Energy* **2014**, *69*, 860–872. [CrossRef]
2. Bompard, E.; Carpaneto, E.; Chicco, G.; Gross, G. The Role of Load Demand Elasticity in Congestion Management and Pricing. In Proceedings of the IEEE Power Engineering Society Transmission and Distribution Conference, Seattle, WA, USA, 16–20 July 2000; Volume 4.
3. Lombardi, C.; Picado-Santos, L.; Annaswamy, A.M. Model-Based Dynamic Toll Pricing: An Overview. *Appl. Sci.* **2021**, *11*, 4778. [CrossRef]
4. Laval, J.A.; Cho, H.W.; Muñoz, J.C.; Yin, Y. Real-Time Congestion Pricing Strategies for Toll Facilities. *Transp. Res. Part B Methodol.* **2015**, *71*, 19–31. [CrossRef]

5. Kordonis, I.; Dessouky, M.M.; Ioannou, P.A. Mechanisms for Cooperative Freight Routing: Incentivizing Individual Participation. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 2155–2166. [CrossRef]
6. Liu, Y.; Nie, Y. A Credit-Based Congestion Management Scheme in General Two-Mode Networks with Multiclass Users. *Netw. Spat. Econ.* **2017**, *17*, 681–711. [CrossRef]
7. Tan, Z.; Gao, H.O. Hybrid Model Predictive Control Based Dynamic Pricing of Managed Lanes with Multiple Accesses. *Transp. Res. Part B Methodol.* **2018**, *112*, 113–131. [CrossRef]
8. Shukla, A.; Bhattacharya, P.; Tanwar, S.; Kumar, N.; Guizani, M. DwaRa: A Deep Learning-Based Dynamic Toll Pricing Scheme for Intelligent Transportation Systems. *IEEE Trans. Veh. Technol.* **2020**, *69*, 12510–12520. [CrossRef]
9. Zhu, F.; Ukkusuri, S.V. A Reinforcement Learning Approach for Distance-Based Dynamic Tolling in the Stochastic Network Environment. *J. Adv. Transp.* **2015**, *49*, 247–266. [CrossRef]
10. Qiu, W.; Chen, H.; An, B. Dynamic Electronic Toll Collection via Multi-Agent Deep Reinforcement Learning with Edge-Based Graph Convolutional Networks. *IJCAI Int. Jt. Conf. Artif. Intell.* **2019**, *2019*, 4568–4574. [CrossRef]
11. Wang, H.; Hao, W.; So, J.; Xiao, X.; Chen, Z.; Hu, J. A Faster Cooperative Lane Change Controller Enabled by Formulating in Spatial Domain. *IEEE Trans. Intell. Veh.* **2023**, *8*, 4685–4695. [CrossRef]
12. Hu, J.; Lei, M.; Wang, H.; Wang, M.; Ding, C.; Zhang, Z. Lane-level Navigation Based Eco-approach. *IEEE Trans. Intell. Veh.* **2023**, *8*, 2786–2796. [CrossRef]
13. Wang, H.; Wang, X.; Li, X.; Wu, X.; Hu, J.; Chen, Y.; Li, Y. A Pathway Forward: The Evolution of Intelligent Vehicles Research on IEEE T-IV. *IEEE Trans. Intell. Veh.* **2022**, *7*, 918–928. [CrossRef]
14. Wang, H.; Hu, J.; Feng, Y.; Li, X. Optimal control-based highway pilot motion planner with stochastic traffic consideration. *IEEE Intell. Transp. Syst. Mag.* **2022**, *15*, 421–436. [CrossRef]
15. Wang, H.; Lai, J.; Zhang, X.; Zhou, Y.; Li, S.; Hu, J. Make space to change lane: A cooperative adaptive cruise control lane change controller. *Transp. Res. Part C Emerg. Technol.* **2022**, *143*, 103847. [CrossRef]
16. Mahajan, N.; Hegyi, A.; Hoogendoorn, S.P.; van Arem, B. Design Analysis of a Decentralized Equilibrium-Routing Strategy for Intelligent Vehicles. *Transp. Res. Part C Emerg. Technol.* **2019**, *103*, 308–327. [CrossRef]
17. Du, L.; Han, L.; Chen, S. Coordinated Online In-Vehicle Routing Balancing User Optimality and System Optimality through Information Perturbation. *Transp. Res. Part B Methodol.* **2015**, *79*, 121–133. [CrossRef]
18. Du, L.; Han, L.; Li, X.Y. Distributed Coordinated In-Vehicle Online Routing Using Mixed-Strategy Congestion Game. *Transp. Res. Part B Methodol.* **2014**, *67*, 1–17. [CrossRef]
19. Li, X.; Liu, W.; Qiao, J.; Li, Y.; Hu, J. An Enhanced Semi-Flexible Transit Service with Introducing Meeting Points. *Netw. Spat. Econ.* **2023**, *23*, 487–527.
20. Li, X.; Wang, T.; Xu, W.; Hu, J. A Novel model for designing a demand-responsive connector (drc) transit system with consideration of users' preferred time windows. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 2442–2451. [CrossRef]
21. Wang, C.; Peeta, S.; Wang, J. Incentive-Based Decentralized Routing for Connected and Autonomous Vehicles Using Information Propagation. *Transp. Res. Part B Methodol.* **2021**, *149*, 138–161. [CrossRef]
22. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 19 December 2020).
23. Rathee, G.; Sharma, A.; Iqbal, R.; Aloqaily, M.; Jaglan, N.; Kumar, R. A Blockchain Framework for Securing Connected and Autonomous Vehicles. *Sensors* **2019**, *19*, 3165. [CrossRef] [PubMed]
24. Li, Y.; Ouyang, K.; Li, N.; Rahmani, R.; Yang, H.; Pei, Y. A Blockchain-Assisted Intelligent Transportation System Promoting Data Services with Privacy Protection. *Sensors* **2020**, *20*, 2483. [CrossRef] [PubMed]
25. Firdaus, M.; Rahmadika, S.; Rhee, K.H. Decentralized Trusted Data Sharing Management on Internet of Vehicle Edge Computing (Iovec) Networks Using Consortium Blockchain. *Sensors* **2021**, *21*, 2410. [CrossRef] [PubMed]
26. Yang, Y.T.; der Chou, L.; Tseng, C.W.; Tseng, F.H.; Liu, C.C. Blockchain-Based Traffic Event Validation and Trust Verification for VANETs. *IEEE Access* **2019**, *7*, 30868–30877. [CrossRef]
27. Kang, J.; Xiong, Z.; Niyato, D.; Ye, D.; Kim, D.I.; Zhao, J. Toward Secure Blockchain-Enabled Internet of Vehicles: Optimizing Consensus Management Using Reputation and Contract Theory. *IEEE Trans. Veh. Technol.* **2019**, *68*, 2906–2920. [CrossRef]
28. Liu, M.; Cheng, L.; Gu, Y.; Wang, Y.; Liu, Q.; O'Connor, N.E. MPC-CSAS: Multi-Party Computation for Real-Time Privacy-Preserving Speed Advisory Systems. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 5887–5893. [CrossRef]
29. Ying, Z.; Cao, S.; Liu, X.; Ma, Z.; Ma, J.; Deng, R.H. PrivacySignal: Privacy-Preserving Traffic Signal Control for Intelligent Transportation System. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 16290–16303. [CrossRef]
30. Hankerson, D.; Vanstone, S.; Menezes, A. *Guide to Elliptic Curve Cryptography*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2006.
31. Paar, C.; Pelzl, J. Chapter 9—Elliptic Curve Cryptosystems. In *Understanding Cryptography—A Textbook for Students and Practitioners*; Springer: Berlin/Heidelberg, Germany, 2010.
32. Fang, W.; Chen, W.; Zhang, W.; Pei, J.; Gao, W.; Wang, G. Digital Signature Scheme for Information Non-Repudiation in Blockchain: A State of the Art Review. *EURASIP J. Wirel. Commun. Netw.* **2020**, *2020*, 56.
33. Evans, D.; Kolesnikov, V.; Rosulek, M. A Pragmatic Introduction to Secure Multi-Party Computation. *Found. Trends®Priv. Secur.* **2018**, *2*, 70–246. [CrossRef]

34. Kolesnikov, V. Gate Evaluation Secret Sharing and Secure One-Round Two-Party Computation. In Proceedings of the 11th International Conference on the Theory and Application of Cryptology and Information Security, Chennai, India, 4–8 December 2005; Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer: Berlin/Heidelberg, Germany, 2005; Volume 3788 LNCS.
35. Lei, K.; Zhang, Q.; Xu, L.; Qi, Z. Reputation-Based Byzantine Fault-Tolerance for Consortium Blockchain. In Proceedings of the International Conference on Parallel and Distributed Systems (ICPADS), Singapore, 11–13 December 2018. [CrossRef]
36. Xiao, Y.; Zhou, C.; Yang, Z. Improved Practical Byzantine Fault Tolerance Algorithm Based on Supply Chain. In Proceedings of the 2022 6th International Conference on Electronic Information Technology and Computer Engineering, Xiamen, China, 21–23 October 2022; pp. 1904–1912. [CrossRef]
37. Girish; Phaneendra, H.D. Identity-Based Cryptography and Comparison with Traditional Public Key Encryption: A Survey. *Int. J. Comput. Sci. Inf. Technol.* **2014**, *5*, 5521–5525.
38. Castro, M.; Liskov, B. Practical Byzantine Fault Tolerance and Proactive Recovery. *ACM Trans. Comput. Syst. (TOCS)* **2002**, *20*, 398–461. [CrossRef]