Achieving Blockchain-based Privacy-Preserving Location Proofs under Federated Learning

Qinglei Kong*, Feng Yin*, Yue Xiao†, Beibei Li†, Xuejia Yang*, Shuguang Cui*,
*Future Network of Intelligence Institute (FNii), the Chinese University of Hong Kong, Shenzhen, China, 518172
†College of Cybersecurity, Sichuan University, Chengdu, China, 610065
Email: sallykongql@gmail.com; yinfeng@cuhk.edu.cn; yuexiao@stu.scu.edu.cn; libeibeibupt@outlook.com;
xuejiayang@link.cuhk.edu.cn; shuguangcui@cuhk.edu.cn

ICC 2021 - IEEE International Conference on Communications | 978-1-7281-7122-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICC42927.2021.9500728

***Abstract*—Federated learning-based navigation has received much attention in vehicular IoT. The intention is to employ a big number of end-users for data collection along different trajectories and perform local training of a global learning model to substitute the global positioning system (GPS) in urban areas. The prerequisites for its commercialization, however, lie in the location-dependent input data trustworthiness and participants' privacy preservation. In this paper, we propose a privacy-preserving proof-of-location mechanism using blockchain to meet these conditions. Specifically, the proposed scheme utilizes a Threshold Identity-Based Encryption (TIBE) system for the generation of secret shares, such that each anonymous location proof can only be verified with at least a threshold number of participants. In addition, the proposed scheme exploits a cuckoo filter for the secure and efficient maintenance and dissemination of location proofs. Systematic security analysis is conducted to demonstrate the fulfillment of harsh security requirements. Performance evaluations are carried out to validate the computation efficiency in comparison with an oblivious transfer (OT) protocol, which has been widely adopted for secure data acquisition.**

***Index Terms*—Federated Learning, Privacy Preservation, Location Proof, Navigation**

I. Introduction

Federated learning has recently attracted considerable attention, which harnesses the computing power of edge devices and performs local model training without data centralization [1]–[3]. With location-dependent data gathered along the moving trajectories, federated learning-based localization mechanisms have also been investigated [4], [5]. Specifically, in order to mitigate the positioning degradation caused by the global positioning system (GPS) outage, a deep neural network (DNN) model has been built with the data collected from the inertial measurement unit (IMU), and then calibrated with information collected by the GPS unit. However, there are still some challenges that remain to be addressed in terms of security and privacy.

The first challenge lies in the reliability of the data input for a navigation model. Since a navigation model relies on location-dependent data, it is particularly important that the geographic information claimed by the end devices are trustworthy, and the end devices can prove that they truly go through the claimed locations during the given period. As a potential solution to this problem, a location proof mechanism allows users to collect proofs for attesting their presences in the spatio-temporal domain, and enables service operators to validate these proofs. From the perspective of proof origins, location proofs are generated by either infrastructures (WiFi or cellular access points) or peers [6], [7]. Meanwhile, the blockchain based systems have also been designed for maintenance of location proofs [8], [9], which addresses the issue of overloading and single point of failure brought by the centralized server. Therefore, to guarantee the input data trustworthy, there is a need for the distributive maintenance and dissemination of location proofs.

The second challenge is that of the privacy preservation of location proofs. The training process under the umbrella of federated learning requires the collaboration of at least a threshold number of users, and the location proofs should only be verified if the threshold number has been satisfied, such that each user's proof can be hidden among at least the threshold number of participants. Meanwhile, since location proofs may indicate the movement trajectories of users, directly publishing user locations and proofs on the blockchain may violate the owners' identity/location privacy, and some works have been proposed in terms of security protection [10], [11]. By exploiting cryptographic techniques, user privacy protection and wormhole attack defense can also be achieved in the location proof architecture proposed in [12]. In order to ensure the integrity and collusion-resistance of location proofs, a spatial-temporal provenance (STP) proof scheme is proposed for ad-hoc mobile users in [13]. Therefore, a privacy-preserving location proof dissemination scheme for the federated learning framework is highly demanded.

In this paper, in order to address the above challenge, we propose a new privacy-preserving proof-of-location dissemination system with blockchain, for federated learning-based navigation. Specifically, the contributions of this paper are threefold as follows:

- First, we propose our privacy-preserving location proof acquisition scheme, which utilizes a threshold encryption system to hide the location proof owners and the location proofs can only be successfully verified with at least a threshold number of users.
- Second, we achieve the secure storage and dissemination of location proofs, which combines a cuckoo filter to efficiently structure the location proofs, and a permissioned blockchain with the proof-of-stake (PoS) consensus to disseminate location proofs.

- Detailed analysis is given to demonstrate that the proposed scheme satisfies the defined security properties. Meanwhile, performance evaluations are conducted to show the computational costs of the proposed scheme, and the results illustrate that the proposed scheme is computationally efficient.

The remainder of this paper is organized as follows. We first introduce our system model and blockchain framework, present our security requirements, and identify our design goals in Section II. Then we present our proposed blockchain-based proof-of-location scheme for learning-based navigation in Section III. The security analysis and performance evaluations are shown in Sections IV and V, respectively. Finally, we conclude this paper in Section VI.

II. System Model, Security Requirements, and Design Goals

In this section, we first introduce the system model of the proposed scheme, describe the blockchain framework, identify the security requirements and show the design goals.

A. System Model

In our system model, we briefly review the location proof generation and dissemination process under the exploited federated learning-based navigation framework, and one can refer to [4] for more details. Specifically, the proposed scheme consists of three kinds of entities: a group of users, base stations, and a fog server.

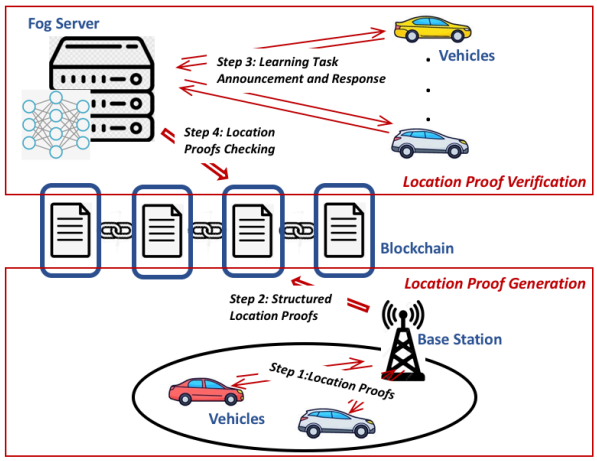

Fig. 1. Proposed Blockchain-based Proof of Location System

- *User*. A user in our proposed scheme could be either a vehicle owner or a smartphone user equipped with an IMU and a GPS unit. The users collect the sensor data, and maintain the collected data in its local storage. When a user joins a training task announced by the fog server, the local model update will be calculated according to the sensor data.
- *Base Station*. A base station in our proposed scheme could be either an RSU or a cellular base station deployed near the roadside. When a user roams under the coverage of a base station, it issues a location proof towards the user, as shown in Fig. 1 (*Step-1: Location Proofs*). Then the base station structures the received location proofs and publishes them on the blockchain, as shown in Fig. 1 (*Step-2: Structured Location Proofs*).
- *Fog Server*. In our proposed scheme, the fog server maintains a model for localized navigation. In order to periodically update a regional navigation model, the fog server identifies the spatio-temporal requirement, and publishes a training task, as shown in Fig. 1 (*Step-3: Learning Task Announcement and Response*). After receiving the responses from a threshold number of users, the fog server checks the blockchain to verify whether a user has entered the required location as he/she claims, as shown in Fig. 1 (*Step-4: Location Proofs Checking*).

Communication Model. The connections between users and base stations could be realized using 5G or Vehicular ad hoc networks (VANETs) defined in $IEEE$ 802.11p [14]. Meanwhile, the connections between users and the fog server, could also be realized through any format of wireless signaling with high bandwidth and low latency.

B. Blockchain Framework

In the blockchain framework, we describe the access rights, architecture and consensus of the blockchain.

- *Access Rights.* All the involved entities can **read** both the transaction and ledger, but only the base stations can **write**. Meanwhile, a transaction proposed by one base station needs to be verified by the rest of base stations.
- *Architecture.* A transaction is a basic component of the blockchain system, and each transaction contains the location proofs generated by a base station during a given time slot. Then the base station broadcasts the transaction towards the rest of the base stations. While the blockchain ledger has a series of blocks, the "genesis" block is created during the initialization phase, and each subsequent block is attached to the previous block. Moreover, each block contains all the transactions generated during a time slot.
- *Consensus.* We exploit the PoS consensus protocol proposed in [15], which runs a pseudo-random block proposer identification mechanism, and the probability of being identified as a proposer is proportional to the scale of their location proofs. If a base station is selected to propose a new block, it generates a new block, broadcasts the block towards the entire blockchain network, and attaches the block towards the ledger.

C. Security Requirements

In our security model, we consider the fog server is honest-but-curious. That is, the fog server will correctly follow the defined protocol, but try to infer some additional information about users. Meanwhile, as there may involve incentives for a training task, users may lie about their locations in order to join the learning task for incentives. Furthermore, we assume there is no collusion between any two entities. Based on the above

analysis, the security requirements of the proposed scheme is shown as follows.

- *Privacy preservation.* Firstly, as the proposed system exploits a blockchain to store and share the location proofs, based on the data published on the blockchain, it should not disclose any information about which user a location proof is belonging to. Secondly, for each training task announced by the fog server, a training group could be formulated and execute the navigation task, under the condition that at least a threshold number of users respond to join the task. Thirdly, the fog server could not recover the real identity of a responding user, unless a training group could be successfully formulated.
- *Location verification.* In our proposed system, as a responding user intends to attest the presence in the spatio-temporal domain, the fog server should be able to verify whether a location proof is originated from the base station as it claims to be. Then the fog server should also identify the content stored on the blockchain, to check whether the user satisfies the claimed location and time requirement.
- *Immutability.* As the proposed scheme involves verifying the correctness of the received location proofs, storing part of the location proofs at the base stations or fog servers may introduce excessive communication overheads and high computational verification complexity, especially in the context of frequent navigation task announcements. Thus, an immutable record is required to maintain and disseminate the location proofs.

D. Design Goals

Based on the aforementioned system model and security requirements, the design goals of our proposed scheme are shown as follows.

The proposed scheme should satisfy the above defined security requirements. If the proposed scheme does not take the security requirements into consideration, the following situations may occur: 1) the identity and location privacy of the origins could be violated; 2) the fog server cannot verify the correctness of the spatio-temporal information claimed by the responding users, and then the quality of the vehicle navigation learning model could not be guaranteed.

The proposed scheme should achieve the goal of high efficiency. As cryptographic operations may introduce heavy computational complexity and communication overheads towards the system, and their exploitations should be carefully considered.

III. Proposed Location Proof Generation and Verification Scheme

In this section, we present our proposed blockchain-based location proof generation and verification process for the federated learning-based navigation system.

A. System Initialization

Given a security parameter κ, a group of base stations reach an agreement on the 7-tuple bilinear parameters denoted as $\{q, \mathbb{G}, \mathbb{G}_T, e, g, g_2, h_1\}$ by running $\mathcal{G}en(\kappa)$, where $\mathcal{G}en$ denotes a probabilistic algorithm that takes a parameter κ as input, q is a prime number with $|q| = \kappa$, $\mathbb{G}$ is an additive cyclic group and $\mathbb{G}_T$ is a multiplicative cyclic group, $(g, g_2, h_1) \in \mathbb{G}$ are three generators, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is non-degenerated and computable bilinear map. More detailed information about bilinear maps can be found in [16]. Meanwhile, the base stations agree on a secure hash function, i.e., $H(\cdot) : \{0,1\}^* \to \mathbb{Z}_q^*$, and a cuckoo filter. More detailed information about the initialization of the Cuckoo filter structure can be found in [17].In addition, we assume the base stations determine a fixed time slot length ts, i.e., each base station only issues one location proof towards one user under its coverage during a single time slot.

During the registration of a base station with identity ID, the base station first chooses $k-1$ random secret numbers $\alpha_j, j \in \{1, 2, ..., k-1\}$, where $\alpha_j \in \mathbb{Z}_q^*$, and further calculates the public values $g^{\alpha_j}, j \in \{1, 2, ..., k-1\}$. Secondly, the base station generates a new transaction and publishes the initialization information $ID||(g^{\alpha_1}, g^{\alpha_2}, ..., g^{\alpha_{k-1}})$ on the blockchain, where the format of the transaction is illustrated in Fig. 3, where the payload is changed to $(g^{\alpha_1}, g^{\alpha_2}, ..., g^{\alpha_{k-1}})$.

B. Location Proof Generation

At the beginning of each time slot ts, base station ID first identifies a new random secret number $\beta \in \mathbb{Z}_q^*$, and computes a public value g^β. Then base station ID chooses another random value $s \in \mathbb{Z}_q^*$, and generates a ciphertext-tuple $C = (C_1, C_2, C_3)$ of another random value $x \in \mathbb{G}_T$, which is

$$\begin{cases} C_1 = e(g^\beta, g_2)^s \cdot x, \\ C_2 = g^s, \\ C_3 = ((g^\beta)^{ID} \cdot h_1)^s. \end{cases} \tag{1}$$

When a vehicle with identity $vid_i \in \mathbb{Z}_q^*$ roams under the coverage of base station ID, it first establishes a secure link with base station ID, and it securely sends a location proof request $vid_i||t_i$ towards base station ID, where t_i is the current timestamp, as shown in Fig. 2 (*Location Proof Request*). After that, the base station ID does the following steps:

- Base station ID first generates a new pseudo-identity for vehicle vid_i with the secret value x and timestamp t_i, which is denoted as $pid_i = vid_i \cdot H(x||t_i)$.
- Base station ID embeds pseudo-identity pid_i with the parameter tuple $(\alpha_1, \alpha_2, ..., \alpha_{k-1}, \beta)$, and derives a polynomial $f(pid_i)$, which is

$$f(pid_i) = \sum_{j=1}^{k-1} \alpha_j \cdot (pid_i)^j + \beta. \tag{2}$$

- Base station ID selects a random number $r_i \in \mathbb{Z}_q^*$ and generates a secret share tuple $ss_i = (\omega_{i,0}, \omega_{i,1})$, which is

$$\begin{cases} \omega_{i,0} = g_2^{f(pid_i)} \cdot (g^{\beta \cdot ID} \cdot h_1)^{r_i}, \\ \omega_{i,1} = g^{r_i}. \end{cases} \tag{3}$$

- Base station ID formulates a message msg_i, and it securely transmits $msg_i = pid_i || ss_i || t_i$ towards vehicle vid_i, as shown in Fig. 2 (*Location Proof Response*).

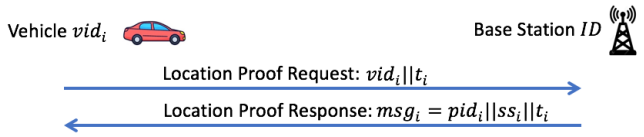

Fig. 2. Exchange of Location Proof Messages

Meanwhile, base station ID performs the following steps to structure the pseudo-identities of all the proof requesting vehicles. Specifically, the base station ID first computes a unique value $cid_i = H(vid_i||x||t_i)$ for each vehicle, and inserts each derived value cid_i into an empty Cuckoo filter CF as follows:

$$\begin{cases} f_i = fingerprint(cid_i) \\ h_1(cid_i) = hash(cid_i) \\ h_2(cid_i) = h_1(cid_i) \oplus hash(f_i) \end{cases} \tag{4}$$

Note that more detailed information about the Cuckoo filter can be found in [17].

At the end of time slot ts, base station ID formulates a transaction with the content $ID||ts||(C_1, C_2, C_3)||g^\beta||CF$, whose format is shown in Fig. 3. Then it broadcasts the transaction towards the entire system.

Identifier	Transaction Hash Digest
Body	Origin ID||ts
Payload	$(C_1, C_2, C_3)\|\|g^\beta\|\|CF$
Digital Signature	Signature of Origin ID

Fig. 3. Format of Transaction Proposed by Origin Base Station

If base station ID is also identified as the proposer base station through the PoS consensus, it also formulates a new block with the **write** access control, including all the transactions generated during the timeslot ts (as shown in Fig. 4), and it broadcasts the block towards the rest of base stations to perform **verify** access control. If the block is verified to be correct, the newly formulated block can be attached to the blockchain.

C. Location Proof Verification

During the learning task announcement phase, the fog server first identifies the target spatial range, and broadcasts a training task $ID||ts$ towards all the involved vehicles. (Note that the spatial range may include the coverage of one or multiple base stations, and we take one base station as an example.) Meanwhile, we assume a group of users $\mathcal{K}$ respond

Identifier	Block Hash Digest
Body	Proposer ID||ts
	Previous Block Hash Digest
	Proof-of-Stake Consensus
Payload	Transactions included
Digital Signature	Signature of Proposer ID

Fig. 4. Format of Block Generated by Proposer Base Station

to the training task. Specifically, if a user intends to join the task, it first establishes a secure transmission link with the fog server, and replies with the message $pid_i||ss_i, i \in \mathcal{K}$. Furthermore, the fog server searches the blockchain for the initialization block of the base station ID containing information $ID||(g^{\alpha_1}, g^{\alpha_2}, ..., g^{\alpha_{k-1}})$, and identifies the block that satisfy spatio-temporal requirement $ID||ts||(C_1, C_2, C_3)||g^\beta||CF$.

- For a user's message $pid_i||ss_i, i \in \mathcal{K}$, the fog server performs the verification process:

$$vk = (g^{\alpha_{k-1}})^{\sum_{i \in \mathcal{K}} pid_i^{k-1}} \cdot ... \cdot (g^{\alpha_1})^{\sum_{i \in \mathcal{K}} pid_i} \cdot g^{\beta \cdot k}, \tag{5}$$

and the fog server performs the verification process:

$$e(vk, g_2) \cdot e((g^\beta)^{ID} \cdot h_1, \prod_{i \in \mathcal{K}} \omega_{i,1}) \stackrel{?}{=} e(g, \prod_{i \in \mathcal{K}} \omega_{i,0}). \tag{6}$$

- If Eq. (6) is verified to be correct, the fog server derives the aggregated value (ω_0, ω_1), in which $\omega_0 = \prod_{i \in \mathcal{K}} \omega_{i,0}^{\sum_{j \in \mathcal{K}, j \neq i} \frac{-x_j}{x_i - x_j}}$ and $\omega_1 = \prod_{i \in \mathcal{K}} \omega_{i,1}^{\sum_{j \in \mathcal{K}, j \neq i} \frac{-x_j}{x_i - x_j}}$, and further performs the decryption process:

$$\hat{x} = C_1 \cdot \frac{e(C_3, \omega_1)}{e(C_2, \omega_0)}. \tag{7}$$

- Given the decryption result $\hat{x}$, the fog server recovers the real identity $vid_i, i \in \mathcal{K}$, which is

$$\widehat{vid}_i = pid_i \cdot H(\hat{x}||t_i)^{-1}. \tag{8}$$

Then the fog server recovers the value $\widehat{cid}_i = H(\widehat{vid}_i||x||t_i), i \in \mathcal{K}$, and checks the cuckoo filter CF for the existence of $\hat{f}_i = fingerprint(\widehat{cid}_i)$ in either position $h_1(\widehat{cid}_i)$ or position $h_2(\widehat{cid}_i)$. If all the k responding vehicles are verified to be correct, the fog server continues to perform the training process.

IV. Security Analysis

In this section, we analyze the security properties of the proposed scheme, in terms of privacy preservation, location verification and immutability.

The proposed scheme can achieve the security goal of privacy preservation. Firstly, we exploit TIBE for the encryption of the secret value x, which is proven to be secure against the chosen ciphertext attacks and consistency of decryptions. Given the ciphertext tuple (C_1, C_2, C_3) displayed on the blockchain, value x cannot be recovered, unless at least a threshold number of k users present their secret shares for decryption. Besides, the Cuckoo Filter kept on the blockchain

only includes the fingerprint of vid_i, which does not disclose vid_i. Secondly, a vehicle's real identity vid_i is hidden within the pseudo-identity pid_i, only pid_i and ss_i are transmitted to the fog server. If the fog server does not receive at least k secret shares, it cannot recover the value x and further discover the real identities. Thus, the security goal of privacy preservation can be well achieved in our scheme.

The proposed scheme can achieve the security goal of location verification. Firstly, our proposed scheme exploits a homomorphic threshold cryptosystem, which supports the ciphertext validity testing of each secret share. Besides, the TIBE that we exploit is also proven to be secure against the chosen identity attacks, and the origin of the secret shares can be effectively authenticated. Secondly, after the recovery of the secret value x, the fog server can recover the real identity vid_i, and further compute the value cid_i with the recovered vid_i. Based on cid_i, the fog server can check the cuckoo filter CF to verify whether identity vid_i is existed or not. Thus, the security goal of location verification can be achieved in our scheme.

The proposed scheme can achieve the security goal of immutability. The PoS consensus-based blockchain system that we exploited, is a distributed ledger constructed by base stations within a peer-to-peer network. Since the order and content stored on the blockchain is agreed by all the base stations, the secure and reliable dissemination of location proofs can be achieved in our proposed scheme.

V. Performance Evaluations

To evaluate the performance of the proposed scheme, we compare it with a k-out-of-n oblivious transfer $(OT_n^k - I)$ scheme shown in [18]. The reason we select the $OT_n^k - I$ for comparison is because it can also achieve the secure location proof acquisition and verification, without disclosing which user is under query. The main processes of the compared scheme is briefly described as follows:

- During the location proof generation phase, base station ID also generates ciphertext (C_1, C_2, C_3), as shown in Eq. (1). For each vehicle vid_i, it also generates a pseudo-identity $pid_i = vid_i \cdot H(x||t_i)$, calculates the secret share ss_i, and keeps the real identity vid_i in its local storage. Meanwhile, base station ID computes the value $\hat{cid}_i = H(vid_i||x||t_i), i \in \mathcal{K}$ for each vehicle, and it publishes $ID||ts||(C_1, C_2, C_3)||\hat{cid}_i, i \in \mathcal{K}$ on the blockchain.
- During the location proof verification phase, the fog server also receives the responses from a vehicle group $pid_i||ss_i, i \in \mathcal{K}$. Meanwhile, the fog server also recovers the value x with the received secret shares, as in Eq. (7).
- To verify the correctness of each pseudo-identity $pid_i, i \in \mathcal{K}$, the fog server chooses two polynomials $f(x) = a_0 + a_1 \cdot x + ... + a_{k-1} \cdot x^{k-1} + x^k$ and $f^{'}(x) = b_0 + b_1 \cdot x + ... + b_{k-1} \cdot x^{k-1} + x^k$, where $(a_i, b_i) \in \mathbb{Z}_q^*, i \in \{0, 1, ..., k-1\}$ and $f^{'}(x) = (x - pid_1)(x - pid_2)...(x - pid_k)$. Then the fog server computes the value $A_i = g^{a_i} \cdot h^{b_i}$, and transmits $(A_0, A_1, ..., A_{k-1})$ towards base station ID.
- After receiving $(A_0, A_1, ..., A_{k-1})$, base station ID computes $C_i = (U_i, V_i) = (g^{k_i}, vid_i \cdot B_i^{k_i}), i \in \{1, 2, ..., n\}$, where $k_i \in_R \mathbb{Z}_q$ and $B_i = g^{f(pid_i)} \cdot h^{f(pid_i)} = A_0 A_1^i ... A_{k-1}^{i^{k-1}} (gh)^{i^k}$. Then base station ID transmits $(C_1, C_2, ..., C_n)$ towards the fog server. Lastly, the fog server recovers the real identity vid_i via $vid_i = V_{pid_i} / (U_{pid_i})^{f(pid_i)}$.

We test the computational overhead of the involved cryptographic operations, which exploits the Type-A Java Pairing-Based Cryptography Library (JPBC) for bilinear pairing [19]. We do experiments using a desktop with a dual-core processor Intel(R) Core(TM) i7 − 8700 CPU @3.20GHz and 8.00GB of installed RAM on a Windows 10 Enterprise platform. Then the following cryptographic costs are derived, after testing 1000 times: one exponentiation operation in $\mathbb{G}$ is $C_e = 7.98$ ms, an exponentiation operation in $\mathbb{G}_T$ is $C_t = 0.57$ ms, and a pairing operation is $C_p = 4.49$ ms. For our proposed scheme, during the proof generation process, it takes 3 exponentiation operations in $\mathbb{G}$ and 1 exponentiation operation in $\mathbb{G}_T$ to generate the ciphertext tuple (C_1, C_2, C_3). For each vehicle vid_i, it takes 3 exponentiation operations in $\mathbb{G}$ to generate ss_i. Thus, the total computation complexity for base station ID is $(3 * n + 3) * C_e + C_t$, when the number of proof requesting vehicles is n. During the proof verification process, the fog server takes $k - 1$ exponentiation operation in $\mathbb{G}$ to generate vk, and it spends 3 pairing operations for Eq. (7). Meanwhile, the fog server also takes $2 * k$ exponentiation operations in $\mathbb{G}$ and 2 pairing operations for decryption. Therefore, the total computation complexity for the fog server is $(3 * k - 1) * C_e + 5 * C_p$. While for the traditional scheme, during the proof generation process and verification process, base station ID also takes $(3 * n + 3) * C_e + C_t$ and the fog server also spends $(3 * k - 1) * C_e + 5 * C_p$ for the verification and decryption process. Meanwhile, to verify the correctness of the pseudo-identities, the fog server also takes $2 * k * C_e$ for the generation of $(A_0, A_1, ..., A_{k-1})$, and base station ID also takes $(n * k + n) * C_e + n * C_e$ for vehicle verification.

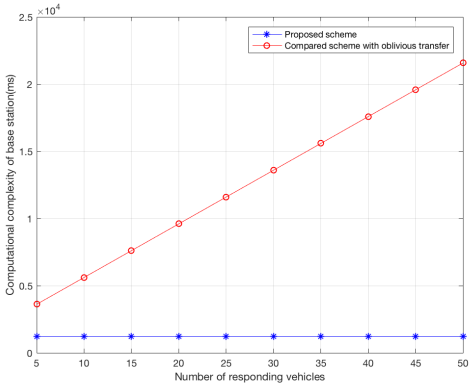

Fig. 5. Computational Complexity of Base Station ID

Fig. 5 compares and shows the computational complexity of base station ID, when the number of responding vehicles ranges between 5 to 50. In our proposed scheme, the

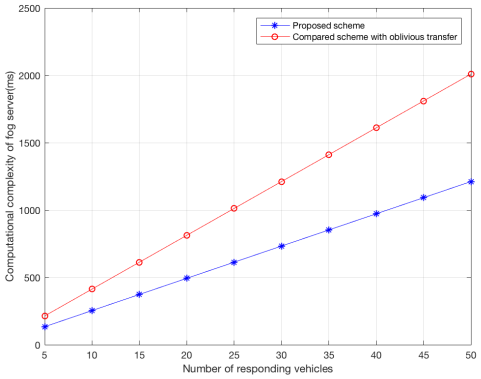

Fig. 6. Computational Complexity of Fog Server

computational cost of base station ID is fixed, while the computational cost of base station ID of the compared OT scheme increases with respect to the increase of the responding vehicles. Meanwhile, evaluation results show that the proposed scheme greatly reduces the computational cost of base station ID. Fig. 6 compares and shows the computational complexity of the fog server, when the number of responding vehicles ranges between 5 to 50. Evaluation results show that our proposed scheme greatly reduces the computational cost, in comparison with the OT scheme. For example, when the number of responding vehicles is set to be 20, the computational cost of our scheme is 493.3 ms, and the computational cost of the compared scheme is 812.5 ms.

VI. Conclusion

In this paper, we have proposed a privacy-preserving proof-of-location scheme under the federated learning-based navigation framework in vehicular IoT, which exploits a PoS blockchain for the maintenance and dissemination of location proofs. Specifically, our scheme exploits a homomorphic threshold encryption scheme for the generation of secret shares, such that the location proofs can only be verified with at least a threshold number of users. Meanwhile, to efficiently hide the identities of location proof origins on blockchain, the proposed scheme utilizes a cuckoo filter for the storage of location proofs. Furthermore, thorough security analysis is conducted to validate the security properties: privacy preservation and location verification. In performance evaluations, we compare with an oblivious transfer scheme, and results show that our scheme greatly reduces the computation overheads. For future work, we will consider the implementation of the PoS blockchain system, and test its real-field performance.

Acknowledgment

The work was supported in part by the Key Area R&D Program of Guangdong Province with grant No. 2018B030338001, by the National Key R&D Program of China with grant No. 2018YFB1800800, by Shenzhen Outstanding Talents Training Fund, and by Guangdong Research Project No. 2017ZT07X152.

References

[1] B. S. Ciftler, A. Albaseer, N. Lasla, and M. Abdallah, "Federated learning for RSS fingerprint-based localization: A privacy-preserving crowdsourcing method," in *Proceedings of the International Wireless Communications and Mobile Computing Conference (IWCMC'2020), Limassol, Cyprus, 2020*.

[2] B. Li, Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, "Deepfed: Federated deep learning for intrusion detection in industrial cyber-physical systems," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2020.

[3] Y. Sun, Y. Cui, and H. Liu, "Joint pushing and caching for bandwidth utilization maximization in wireless networks," *IEEE Transactions on Communications*, vol. 67, no. 1, pp. 391–404, 2019.

[4] F. Yin, Z. Lin, Y. Xu, Q. Kong, D. Li, S. Theodoridis, and S. Cui, "Fedloc: Federated learning framework for cooperative localization and location data processing," *CoRR*, 2020.

[5] Y. Xu, F. Yin, W. Xu, J. Lin, and S. Cui, "Wireless traffic prediction with scalable Gaussian process: Framework, algorithms, and verification," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1291–1306, Jun. 2019.

[6] Z. Zhu and G. Cao, "APPLAUS: A privacy-preserving location proof updating system for location-based services," in *Proceedings of the International Conference on Computer Communications (INFOCOM'11), Shanghai, China, 2011*.

[7] F. Boeira, M. Asplund, and M. P. Barcellos, "Vouch: A secure proof-of-location scheme for vanets," in *Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWiM'18, Montreal, QC, Canada*, 2018.

[8] W. Wu, E. Liu, X. Gong, and R. Wang, "Blockchain based zero-knowledge proof of location in iot," in *Proceedings of the IEEE International Conference on Communications, (ICC'20), Dublin, Ireland, 2020*.

[9] M. Amoretti, G. Brambilla, F. Medioli, and F. Zanichelli, "Blockchain-based proof of location," in *IEEE International Conference on Software Quality, Reliability and Security Companion, QRS Companion'18, Lisbon, Portugal, 2018*.

[10] Q. Kong, L. Su, and M. Ma, "Achieving privacy-preserving and verifiable data sharing in vehicular fog with blockchain," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2020.

[11] Q. Kong, R. Lu, M. Ma, and H. Bao, "A privacy-preserving and verifiable querying scheme in vehicular fog data dissemination," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1877–1887, 2019.

[12] W. Luo and U. Hengartner, "Veriplace: a privacy-aware location proof architecture," in *Proceedings of the International Symposium on Advances in Geographic Information Systems, ACM-GIS'10, San Jose, CA, USA, 2010*.

[13] X. O. Wang, A. Pande, J. Zhu, and P. Mohapatra, "STAMP: enabling privacy-preserving location proofs for mobile users," *IEEE/ACM Trans. Netw.*, vol. 24, no. 6, pp. 3276–3289, 2016.

[14] I. S. Association *et al.*, "802.11 p-2010-IEEE standard for information technology-local and metropolitan area networks-specific requirements-part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 6: Wireless access in vehicular environments," 2010.

[15] W. Li, S. Andreina, J. Bohli, and G. Karame, "Securing proof-of-stake blockchain protocols," in *Proceedings of the Data Privacy Management, Cryptocurrencies and Blockchain Technology - ESORICS International Workshops'17, DPM'2017 and CBT'2017, Oslo, Norway, 2017*.

[16] D. Boneh, X. Boyen, and S. Halevi, "Chosen ciphertext secure public key threshold encryption without random oracles," in *Proceeding of the Topics in Cryptology at the RSA Conference CT-RSA' 2006, San Jose, CA, USA, 2006*.

[17] B. Fan, D. G. Andersen, M. Kaminsky, and M. Mitzenmacher, "Cuckoo filter: Practically better than bloom," in *Proceedings of the ACM International on Conference on emerging Networking Experiments and Technologies, CoNEXT'14, Sydney, Australia, 2014*.

[18] C. Chu and W. Tzeng, "Efficient k-out-of-n oblivious transfer schemes," *J. UCS*, vol. 14, no. 3, pp. 397–415, 2008.

[19] A. De Caro and V. Iovino, "jpbc: Java pairing based cryptography," in *Proceedings of the 16th IEEE Symposium on Computers and Communications, ISCC 2011*, 2011, pp. 850–855.