A Privacy-Preserving Proximity Testing Using Private Set Intersection for Vehicular *Ad-Hoc* Networks

Liping Zhang, Wenhao Gao, Shukai Chen, Wei Ren, *Member, IEEE*, Kim-Kwang Raymond Choo, *Senior Member, IEEE*, and Neal N. Xiong, *Senior Member, IEEE*

***Abstract*—Proximity testing technologies have been of increasing importance in vehicular *ad-hoc* networks (VANETs), especially in location-based services. However, there exist several known challenges in most existing proximity testing methods. For instance, during proximity testing, how to protect the location privacy of users, guarantee the fairness trait of both communication parties, and reduce computational costs is challenging. In this article, we present an efficient privacy-preserving proximity testing scheme using private set intersection (PSI) and differential privacy. In our design, a Chebyshev-based PSI is constructed to achieve location privacy with low energy consumption during the proximity testing process. Furthermore, geo-indistinguishability is employed in our scheme to generate virtual points as inputs set of PSI, which further protects the location privacy from exposure and provides resistance to collusion attacks. Fairness requirement is also satisfied in our scheme. The performance evaluation shows that the proposed scheme achieves good efficiency and is suitable for VANETs.**

***Index Terms*—Chebyshev chaotic maps, privacy protection, private set intersection (PSI), proximity testing.**

Manuscript received September 30, 2021; revised November 11, 2021; accepted December 1, 2021. Date of publication December 10, 2021; date of current version July 11, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62172303 and Grant 61972366, in part by the Open Research Project of the Hubei Key Laboratory of Intelligent GeoInformation Processing under Grant KLIGIP-2019B09 and Grant 2021B06, in part by the Provincial Key Research and Development Program of Hubei under Grant 2020BAB105, in part by the Foundation of Henan Key Laboratory of Network Cryptography Technology under Grant LNCT2020-A01, in part by the Foundation of Key Laboratory of Network Assessment Technology, Chinese Academy of Sciences under Grant KFKT2019-003, and in part by the Foundation of Guizhou Provincial Key Laboratory of Public Big Data under Grant 2019BDKFJJ003 and Grant 2019BDKFJJ011. The work of Kim-Kwang Raymond Choo was supported by the Cloud Technology Endowed Professorship. Paper no. TII-21-4270. *(Corresponding author: Wei Ren.)*

Liping Zhang, Wenhao Gao, and Shukai Chen are with the School of Computer Science, China University of Geosciences, Wuhan 430074, China (e-mail: carolyn321@163.com; gwh19971023@163.com; chenshukai@cug.edu.cn).

Wei Ren is with the School of Computer Science, China University of Geosciences, Wuhan 430074, China, with the Henan Key Laboratory of Network Cryptography Technology, Zhengzhou 450001, China, and also with the Key Laboratory of Network Assessment Technology, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China (e-mail: weirencs@cug.edu.cn).

Kim-Kwang Raymond Choo is with the Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX 78249 USA (e-mail: raymond.choo@fulbrightmail.org).

Neal N. Xiong is with the Department of Mathematics and Computer Science, Northeastern State University, Tahlequah, OK 74464 USA (e-mail: xiongnaixue@gmail.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2021.3133566.

Digital Object Identifier 10.1109/TII.2021.3133566

I. Introduction

IN INTELLIGENT transportation systems, location-based services (LBSs) rely on accurate positioning data to support the various requirements of vehicle users and services [1]. Through LBS, users can obtain nearby information that they are interested in, such as traffic status, nearest gas station, and nearby users in the vehicular *ad-hoc* networks (VANETs) [2].

Proximity testing is an essential technique in LBS, as it enables a user to determine whether any other vehicle is within the specific geometric range [3]. During the proximity testing process, the service provider (SP) collects the location information for all vehicles participating in the LBS. When a vehicle requests proximity testing, the SP will find other vehicles in the vicinity, and then the proximity testing process will be performed between the respective vehicles. Subsequently, the vehicle that initiates the request for proximity testing can find all the nearby users through the above process. However, existing proximity testing services may be vulnerable to different attacks. For instance, when proximity testing is operated in insecure public networks [4], the adversary could eavesdrop on location information, which could be used to facilitate other nefarious activities such as information theft (e.g., to obtain vehicle user's other sensitive information such as home addresses, lifestyle information, and social connections) and frauds [5].

Given the importance of the topic, a number of security solutions for proximity testing have been presented in the literature. Examples include security schemes [6]–[8], approaches using location tags to hide the real location with the aim of providing privacy protection, and so on. There have also been approaches that rely on homomorphic encryption [9] and elliptic curves [10] to ensure communication security in proximity testing. Another alternative approach is the use of the grid-and-hash approach to conceal the locations [11]. In the scheme outlined in [12], proximity regions (instead of using vehicle location region) are utilized to avoid location exposure. Furthermore, to achieve

location privacy, private equality test (PET) based proximity testing schemes, such as those described in [7] and [8], have been proposed to test the equality of private elements. In recent years, some latest technologies, such as blockchain technology [13]–[16] and physically unclonable functions [17], [18], have been proposed to enhance privacy protection, and their application in proximity testing needs to be explored.

However, in many existing privacy-preserving proximity testing approaches, some known limitations have been revealed. For example, collusion attacks are not considered in approaches, such as those presented in [7] and [8]. In other words, a curious user may collude with the SP to compromise other users' location privacy [19]. Furthermore, in some approaches, users participating in proximity testing are not informed of the testing results at the same time, which results in unfairness [6], [20]. In addition, future solution designers need to consider minimizing computational costs in proximity testing (and consequently lowering energy consumption), particularly if these solutions are to be deployed in real-world applications.

To solve the above discussed challenges, we propose an efficient privacy-preserving proximity testing scheme using Chebyshev-based PSI (PTCP). Specifically, in our approach, we have the following.

1) We design an efficient Chebyshev polynomial algorithm that meets the security requirements advocated by Chen *et al.* [21] and Yoshioka *et al.* [22]. Based on our proposed algorithm, a novel Chebyshev-based private set intersection (PSI) scheme is constructed for proximity testing to achieve location privacy protection. Since only Chebyshev polynomial operations are involved in PSI design, our scheme incurs minimal energy consumption.
2) To enhance security, geo-indistinguishability is employed to generate virtual points for vehicles. In our scheme, the cloaking coordinates of these virtual points are mapped to their corresponding grids and then these grids' tags form the input set of our PSI protocol. Hence, even in the event that an adversary compromises the PSI scheme, he/she cannot obtain the actual vehicle location. In addition, the adoption of virtual points can protect the scheme from collusion attacks.
3) In our design, the SP sends the results of proximity testing to both communication parties at the same time, which meets the fairness requirement of proximity testing.

The rest of this article is organized as follows. Sections II and III introduce the related approaches and preliminaries, respectively. Our proposed PTCP scheme is described in Section IV and its security is discussed in Section V. The performance of our scheme is evaluated in Section VI. Finally, Section VII concludes this article.

II. Related Work

In recent years, various approaches have been applied in VANETs to achieve privacy-preserving proximity testing, such as cryptography and PET. PET is an important method that has been widely used for proximity testing. In the scheme of Narayanan *et al.* [7], PET was employed to construct proximity testing schemes under different privacy levels. In the scheme of Kotzanikolaou *et al.* [8], an improved PET protocol was proposed to reduce computational burden during proximity testing.

There are some other attempts to protect location privacy during proximity testing. In the scheme of Huang *et al.* [12], fast scale product technique and differential privacy were employed to achieve privacy-preserving proximity testing among multiple users. Qiu *et al.* [23] also introduced the concept of differential privacy in their design and then further utilized location tags to perturb users' real location. However, although these schemes [7], [8], [12], [23] have made some efforts to enhance the security during proximity testing, the fairness requirement is not satisfied, which may lead to the dominance of one communication side. Moreover, these schemes [7], [8], [12], [23] also suffered from collusion attacks. Under this case, the adversary may collude with one communication party to obtain the other party's privacy. Therefore, a privacy-preserving proximity testing scheme should provide fairness and resistance to collusion attacks.

To resist collusion attacks, Choi *et al.* [9] employed homomorphic encryption in their design without a trusted third party. But fairness was still not satisfied in their scheme [9]. To meet the security requirements, elliptic curve encryption was employed in the design of Sakib *et al.* [10] design to achieve proximity testing with privacy protection. Although their scheme [10] realizes fairness and can resist collusion attacks, the energy consumption of their scheme is still high due to the usage of time-consuming operations such as elliptic curve point multiplication operations.

Blockchain is another popular technique that could be used to protect location privacy in VANETs. Liao *et al.* [14] proposed a framework incorporating blockchain and smart contracts to facilitate fair task offloading in VANETs. Yang *et al.* [15] employed the blockchain technique to construct a multidomain vehicular authentication architecture. In the scheme of Li *et al.* [16], the blockchain technique was utilized to record users' real information and issue certificates. Although blockchain technology has the potential to provide a new possibility for location privacy protection, whether it is suitable for proximity testing is still an open issue. In the design of a privacy-preserving proximity testing scheme, more details should be considered for practical applications, such as security, accuracy, and energy consumption. Therefore, how to apply the existing blockchain technique in proximity testing still needs further research.

PSI is an effective cryptographic method to obtain the intersection of two sets [24]. Via PSI, the information of elements that are not in the intersection is not revealed. Currently, PSI has been widely used in various fields, such as feature matching [25], social networking [26], and cloud computing [27]. For example, in VANETs, PSI has been utilized to realize feature matching without exposing the unmatched features [25]. Inspired by this, we introduce the concept of PSI in our design to achieve location privacy during proximity testing.

III. Preliminaries

In this section, we review the basic concepts of differential privacy for LBSs, PSI, the system model, and the adversary

TABLE I
NOMENCLATURE

Notation	Definition
ID_i	Identity of the i^{th} vehicle
ID_{SP}	Identity of the location-based service provider
v	The seed of Chebyshev polynomial
p	A large prime
$H(\cdot)$	Secure one-way hash function
ε	The privacy budget of differential privacy
σ_i, a_j	High-entropy random numbers
μ	The side length of the grids
s	Private key of SP
pub_{SP}	Public key of SP
k_i	Private key of i^{th} vehicle
pub_i	Public key of i^{th} vehicle
T_γ	Timestamp
(x_i, y_i)	The i^{th} vehicle's real location coordinates
(cx_i, cy_i)	The i^{th} vehicle's cloaking location coordinates
E_k	Symmetric encryption algorithm with the key k
D_k	Symmetric decryption algorithm with the key k
$\mathcal{Z}$	All possible reported locations
$\mathbb{R}$	The set of all real numbers
$\|\|$	Concatenation operation
$\oplus$	Exclusive-or operation

model. The notations employed in this article are described in Table I.

A. Differential Privacy for Location-Based Services

Geo-indistinguishability is an expansion of differential privacy in location-based systems to protect users' real location by allowing approximate information. In this mechanism, a bivariate version of the Laplace function is utilized to generate noises to perturb the original user location.

Definition 1 (Geo-Indistinguishability) [28]: A mechanism K achieves ε-geo-indistinguishability if and only if for all x, x'

$$K(x)(Z) \leq e^{\varepsilon d(x,x')} K\left(x'\right)(Z) \tag{1}$$

where x, x' refer to two different location points, $d(x, x')$ represents the Euclidean distance between x and x', and ε is the privacy budget that controls the level of differential privacy [28]. The $K(x)(Z)$ can be rewritten as $P(Z|x)$, where P is the conditional probability. Each observation is $Z \subseteq \mathcal{Z}$ where $\mathcal{Z}$ represents all possible reported locations. After adding a planar Laplacian noise, the real location can be reported as a cloaking location. Given the parameter $\varepsilon \in \mathbb{R}^+$ and the real location $x_0 \in \mathbb{R}^2$, the probability density function of Laplacian noise centered at the location x_0 on any other point $x \in \mathbb{R}^2$ is given as (2).It can also be represented as polar coordinate model as shown in (3), where r is the distance between x and x_0, and θ is the glope angle formed by the line xx_0 with respect to the horizontal axis of the Cartesian coordinate system. In order to cloak the real location effectively, r should be calculated as (4), where W_{-1} is the Lambert W function (the –1 branch), p should be chosen from [0,1), and θ is uniformly chosen from $[0, 2\pi)$

$$D_\varepsilon\left(x_0\right)(x) = \frac{\varepsilon^2}{2\pi} e^{-\varepsilon d(x_0, x)} \tag{2}$$

$$D_\varepsilon(r, \theta) = \frac{\varepsilon^2}{2\pi} r e^{-\varepsilon r} \tag{3}$$

$$r = C_\varepsilon^{-1}(p) = -\frac{1}{\varepsilon}\left(W_{-1}\left(\frac{p-1}{e}\right) + 1\right). \tag{4}$$

B. Private Set Intersection

PSI was first introduced by Freedman *et al.* [27], [29]; it is an effective cryptographic method that allows two sets to compute the intersection without exposing the information out of the intersection [27]. Currently, PSI has been used in various fields, such as feature matching [25], social networking [26], and cloud computing [27].

C. System Model

The system model is composed of three types of entities, namely a trusted authority (TA), an SP, and a set of registered vehicles $V = \{V_1, V_2, \ldots, V_n\}$.

Trusted authority: The TA is a powerful entity, fully trusted by other entities, that is responsible for initializing the entire system and generating system parameters.

Service provider: The SP is a core entity whose primary function is to store the protected location of the vehicle secretly and respond to the proximity testing request from the vehicles.

Vehicle (V): Each legal vehicle V_i uploads its protected location information to the SP. Any vehicle can request proximity testing to the SP, and only the corresponding vehicle can obtain the result of proximity testing.

D. Adversary Model

We suppose that entities in the model are honest-but-curious, which means that each vehicle follows the rules of the protocol strictly while also trying to learn as much location information of other vehicles as possible. We also assume that adversaries cannot break the chaotic map-based discrete logarithm problem (CMDLP) [30] within probabilistic polynomial time. Similar to the adversary model adopted in the scheme of Kotzanikolaou *et al.* [8], the active attacks are not considered in our adversary model. So we assume that adversaries have the ability to obtain all the messages transmitted between the entities, but they cannot modify the exchanged messages or inject fake messages to impersonate the legal vehicle.

IV. Proposed Scheme

In this section, we present a privacy-preserving proximity testing scheme using Chebyshev-based PSI (PTCP). In our design, PSI and geo-indistinguishability are employed to achieve location privacy during the proximity testing process. In addition, the geo-indistinguishability property is utilized to perturb the actual location to achieve further privacy protection.

Algorithm 1: Chebyshev Polynomial Algorithm.

$>>$: right logical shift
$*$: matrix multiplication
$\lfloor n \rfloor$: the integer part of n
Input: n, x, N
Output: result $= T_n(x) \bmod N$

$a \leftarrow \begin{bmatrix} 2x & -1 \\ 1 & 0 \end{bmatrix}$, ans $\leftarrow \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
$n \leftarrow n - 1$
while $\lfloor n \rfloor > 0$ **do**
  if n is odd **then**
    ans $\leftarrow$ ans $* a \bmod N$
  end if
  $n \leftarrow n >> 1$
  $a \leftarrow a * a \bmod N$
end while
result $\leftarrow$ ans[0][0] $\cdot\, x +$ ans[0][1] $\bmod\ N$

To achieve high security with low computational costs, Chebyshev polynomials are adopted in our PSI-based proximity testing scheme. To meet the security requirements, in our proposed Chebyshev polynomial algorithm, a large prime number n is adopted, and the modulus N is set as a strong prime number that satisfies $N - 1 = 2p_1$ and $N + 1 = 2p_2$. In addition, a square-matrix-based binary exponentiation algorithm is proposed in our design to compute the Chebyshev polynomials, which reduces the time complexity. We also adopt the matrices given as (5) and (6) to describe the Chebyshev polynomials instead of the conventional recursive form. The nth power modulus N of the matrix $\left[\begin{smallmatrix} 2x & -1 \\ 1 & 0 \end{smallmatrix}\right]$ can be calculated by the binary power algorithm in $O(\log n)$ time complexity. Therefore, our proposed Chebyshev polynomials algorithm achieves low computational costs and satisfies the security requirements proposed by Yoshioka *et al.* [22] and Chen *et al.* [21]. The detailed steps of our proposed algorithm are shown in Algorithm 1.

$$\begin{bmatrix} T_{n+1}(x) \\ T_n(x) \end{bmatrix} = \begin{bmatrix} 2x & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} T_n(x) \\ T_{n-1}(x) \end{bmatrix} \bmod N \tag{5}$$

$$\begin{aligned} \begin{bmatrix} T_{n+1}(x) \\ T_n(x) \end{bmatrix} &= \begin{bmatrix} 2x & -1 \\ 1 & 0 \end{bmatrix}^2 \begin{bmatrix} T_{n-1}(x) \\ T_{n-2}(x) \end{bmatrix} \\ &= \begin{bmatrix} 2x & -1 \\ 1 & 0 \end{bmatrix}^n \begin{bmatrix} T_1(x) \\ T_0(x) \end{bmatrix} \bmod N. \end{aligned} \tag{6}$$

Based on our proposed Chebyshev algorithm, we further construct a proximity testing scheme by using Chebyshev-based PSI. The proposed PTCP consists of three main phases: system initialization phase, vehicle location uploading phase, and proximity testing phase. In the system initialization phase, several parameters are generated and a 2-D map is established to normalize the location of vehicles. Next, in the vehicle location uploading phase, the cloaking coordinates of vehicles are uploaded to SP. Consequently, proximity testing is performed between vehicles that are within a certain search range.

A. System Initialization Phase

In this phase, the whole system is initialized. Specifically, the TA generates some parameters for the Chebyshev polynomial algorithm and constructs a 2-D map to normalize the location of vehicles. The SP computes its private and public keys for the subsequent phases of our PTCP scheme. The detailed steps of this phase are illustrated as follows.

Step S1: The TA first selects a random $v \in (-\infty, +\infty)$ as the seed of the Chebyshev polynomial and a large prime number p as the modulus. Then, it chooses a secure one-way hash function $H() : \{0,1\}^* \rightarrow \{0,1\}^l$. After that, the TA divides the map into multiple nonoverlapping grids of side length μ, and each grid is assigned with a grid tag by order. Finally, the TA publishes the grid map information and public parameters as $\{p, v, H, \varepsilon, \mu\}$.

Step S2: The SP chooses a high entropy random integer $s \in \mathbb{Z}_p^*$ as its private key and computes its corresponding public key pub_{SP} as in the following equation:

$$\text{pub}_{\text{SP}} = T_s(v) \bmod p. \tag{7}$$

B. Vehicle Location Uploading Phase

In this phase, the vehicle uses its private key to hide the real location and uploads the cloaking coordinates to SP.

Step V1: Each vehicle V_i chooses a high entropy random integer k_i as its private key and computes its public key pub_i as (8). Then, the vehicle V_i selects a high entropy random $\sigma_i \in [0, 1)$ and computes r_i as (11) using σ_i and the privacy budget ε. After that, the vehicle V_i can compute the cloaking coordinate (cx_i, cy_i) using its Cartesian coordinate (x_i, y_i) and the computed r_i as in (9) and (10). This cloaking coordinate (cx_i, cy_i) of the vehicle V_i is mapped to a grid with tag, tag_i. After that, the vehicle V_i computes $T_{k_i}(\text{pub}_{\text{SP}})$ using its private key k_i and then it constructs k_{ic} as (12) using the computed $T_{k_i}(\text{pub}_{\text{SP}})$, the timestamp T_γ, the SP's identity ID_{SP}, and its own identity ID_i. Finally, it encrypts message $\{(cx_i, cy_i), \text{ID}_{\text{SP}}, \text{ID}_i, T_\gamma\}$ using k_{ic} to generate C_i as (13) and then sends the message $\{C_i, \text{ID}_i, T_\gamma\}$ to the SP

$$\text{pub}_i = T_{k_i}(v) \bmod p \tag{8}$$

$$cx_i = x_i + r_i \cos\theta, \theta \in [0, 2\pi) \tag{9}$$

$$cy_i = y_i + r_i \sin\theta, \theta \in [0, 2\pi) \tag{10}$$

$$r_i = -\frac{1}{\varepsilon}\left(W_{-1}\left(\frac{\sigma_i - 1}{e}\right) + 1\right) \tag{11}$$

$$k_{ic} = H\left(T_{k_i}\left(\text{pub}_{\text{SP}}\right) \parallel \text{ID}_{\text{SP}} \parallel \text{ID}_i \parallel T_\gamma\right) \tag{12}$$

$$C_i = E_{k_{ic}}\left((cx_i, cy_i) \parallel \text{ID}_{\text{SP}} \parallel \text{ID}_i \parallel T_\gamma\right). \tag{13}$$

Step V2: After receiving all uploaded data, for each vehicle V_i, the SP computes $T_s(\text{pub}_i)$ using its private key s and calculates k_{ci} as (14) using the computed $T_s(\text{pub}_i)$, its own identity ID_{SP}, and the received message $\{\text{ID}_i, T_\gamma\}$. Then, the SP decrypts the received C_i via the computed k_{ci} to obtain (cx_i, cy_i), ID_{SP}, ID_i, and the timestamp T_γ as (15). After that, the SP checks the

validity of the decrypted $\mathrm{ID_{SP}}$ using its own identity. Next, it verifies the freshness of the timestamp and checks whether the decrypted ID_i is equal to the received one. If the verification fails, the SP rejects the uploading location request; otherwise, it secretly stores $((cx_i, cy_i), \mathrm{ID}_i)$ in its database. When this step is finished, the vehicle location uploading phase is completed

$$k_{ci} = H\left(T_s\left(\mathrm{pub}_i\right) \parallel \mathrm{ID_{SP}} \parallel \mathrm{ID}_i \parallel T_\gamma\right) \tag{14}$$

$$D_{k_{ci}}(C_i) = (cx_i, cy_i) \parallel \mathrm{ID_{SP}} \parallel \mathrm{ID}_i \parallel T_\gamma. \tag{15}$$

C. Proximity Testing Phase

In this phase, if a vehicle wants to know whether another vehicle is within a search range without exposing its location information, it needs to perform the following steps.

Step P1: The vehicle V_i first decides a search area SA that consists of multiple location grids. Then, the vehicle V_i computes its cloaking testing range R_i as (16) using the radius l_{SA} of the search area SA, the public parameter σ_i, and the privacy budget ε. After that, the vehicle V_i computes k_{ic} as (17) using the computed $T_{k_i}(\mathrm{pub_{SP}})$, the SP's identity $\mathrm{ID_{SP}}$, its own identity ID_i, and the timestamp T_γ. Consequently, it constructs the CR_i as (18) by encrypting the message $\{R_i, \mathrm{ID_{SP}}, \mathrm{ID}_i, T_\gamma\}$ via the computed key k_{ic}. Finally, the vehicle V_i sends message $\{\mathrm{CR}_i, \mathrm{ID}_i, T_\gamma\}$ to the SP to inform it of the cloaking testing range R_i

$$R_i = l_{\mathrm{SA}} - \frac{1}{\varepsilon}\left(W_{-1}\left(\frac{\sigma_i - 1}{e}\right) + 1\right) \tag{16}$$

$$k_{ic} = H\left(T_{k_i}\left(\mathrm{pub_{SP}}\right) \parallel \mathrm{ID_{SP}} \parallel \mathrm{ID}_i \parallel T_\gamma\right) \tag{17}$$

$$\mathrm{CR}_i = E_{k_{ic}}\left(R_i \parallel \mathrm{ID_{SP}} \parallel \mathrm{ID}_i \parallel T_\gamma\right). \tag{18}$$

Step P2: After receiving the message from the vehicle V_i that requests for proximity testing, the SP calculates k_{ci} as (19) using the computed $T_s(\mathrm{pub}_i)$, the received message $\{\mathrm{ID}_i, T_\gamma\}$, and its own identity $\mathrm{ID_{SP}}$. Then, it can decrypt the received CR_i via the computed k_{ci} to obtain the cloaking testing range R_i, $\mathrm{ID_{SP}}$, ID_i, and the timestamp T_γ as (20). After that, the SP verifies the validity of the decrypted $\mathrm{ID_{SP}}$ via its own identity. Next it checks the freshness of the timestamp and checks whether the decrypted ID_i is equal to its received one. If the verification fails, the SP terminates proximity testing; otherwise, it computes the distance between the cloaking coordinate (cx_i, cy_i) of the vehicle V_i and the cloaking coordinate (cx_j, cy_j) of each vehicle V_j (the SP can obtain (cx_j, cy_j) from its database). Then the SP can figure out all vehicles (V') within the cloaking testing range R_i according to the computed distance. Next, for each $V_j \in V'$, the SP chooses a high entropy random integer a_j and uses it to compute $T_{a_j}(v) \bmod p$ and $T_{a_j}(\mathrm{pub}_j)$. After that, the SP constructs C_{cj} as (21) using the decrypted ID_i and the computed $T_{a_j}(\mathrm{pub}_j)$. Finally, it sends message $\{T_{a_j}(v), C_{cj}\}$ to the vehicle V_j for proximity testing

$$k_{ci} = H\left(T_s\left(\mathrm{pub}_i\right) \parallel \mathrm{ID_{SP}} \parallel \mathrm{ID}_i \parallel T_\gamma\right) \tag{19}$$

$$D_{k_{ci}}(\mathrm{CR}_i) = R_i \parallel \mathrm{ID_{SP}} \parallel \mathrm{ID}_i \parallel T_\gamma \tag{20}$$

$$C_{cj} = \mathrm{ID}_i \oplus T_{a_j}(\mathrm{pub}_j). \tag{21}$$

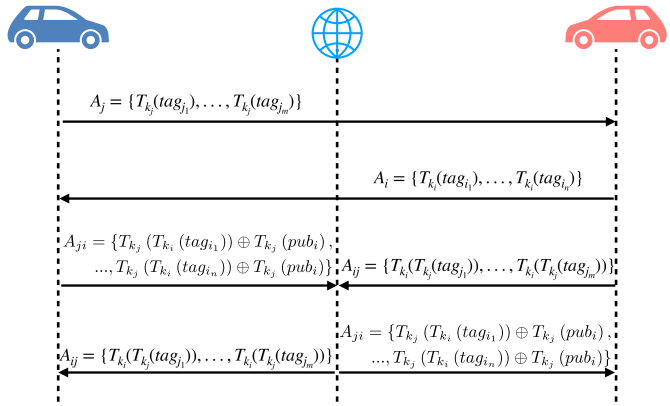

Fig. 1. Our proposed PSI-based proximity testing.

Step P3: After receiving the message from the SP, privacy-preserving proximity testing is performed using Chebyshev-based PSI as shown in Fig. 1. And the detailed procedure is illustrated as follows.

1) Upon receiving $\{T_{a_j}(v), C_{cj}\}$ from the SP, the vehicle V_j computes $T_{k_j}(T_{a_j}(v))$ using its private key k_j, and then it can obtain ID_i as (22) to know who initiates the request for proximity testing. If the vehicle V_j agrees to perform proximity testing with the vehicle V_i, it then finds out every grid tag_{j_m} within its search area and next computes the corresponding $T_{k_j}(\mathrm{tag}_{j_m})$ for each tag_{j_m} by encrypting tag_{j_m} via its private key k_j. Finally, the vehicle V_j sends the message $\{A_j\}$ as (23) to the vehicle V_i

$$\mathrm{ID}_i = C_{cj} \oplus T_{k_j}(T_{a_j}(v)) \tag{22}$$

$$A_j = \{T_{k_j}(\mathrm{tag}_{j_1}), \ldots, T_{k_j}(\mathrm{tag}_{j_m})\}. \tag{23}$$

2) After receiving the message $\{A_j\}$ from vehicle V_j, the vehicle V_i first finds every grid tag_{i_n} within its search area and computes the corresponding $T_{k_i}(\mathrm{tag}_{i_n})$ for each tag_{i_n} by encrypting tag_{i_n} via its private key k_i. Then, it sends the message $\{A_i\}$ as (24) to the vehicle V_j

$$A_i = \{T_{k_i}(\mathrm{tag}_{i_1}), \ldots, T_{k_i}(\mathrm{tag}_{i_n})\}. \tag{24}$$

3) After receiving the message $\{A_i\}$ from the vehicle V_i, the vehicle V_j first computes a series of $T_{k_j}(T_{k_i}(\mathrm{tag}_{i_n}))$ using its private key k_j for each $T_{k_i}(\mathrm{tag}_{i_n})$ in A_i. Next, it computes $T_{k_j}(T_{k_i}(\mathrm{tag}_{i_n})) \oplus T_{k_j}(\mathrm{pub}_i)$ using the computed $T_{k_j}(\mathrm{pub}_i)$ via its private key k_j and $T_{k_j}(T_{k_i}(\mathrm{tag}_{i_n}))$. Then, V_j sends the message $\{A_{ji}\}$ as (25) to the SP. Meanwhile, the vehicle V_i computes $T_{k_i}(T_{k_j}(\mathrm{tag}_{j_m}))$ using its private key k_i for each $T_{k_j}(\mathrm{tag}_{j_m})$ in A_j. And then it transmits the message $\{A_{ij}\}$ as (26) to the SP

$$A_{ji} = \{T_{k_j}\left(T_{k_i}\left(\mathrm{tag}_{i_1}\right)\right) \oplus T_{k_j}\left(\mathrm{pub}_i\right), \ldots, T_{k_j}\left(T_{k_i}\left(\mathrm{tag}_{i_n}\right)\right) \oplus T_{k_j}\left(\mathrm{pub}_i\right)\} \tag{25}$$

$$A_{ij} = \{T_{k_i}(T_{k_j}(\mathrm{tag}_{j_1})), \ldots, T_{k_i}(T_{k_j}(\mathrm{tag}_{j_m}))\}. \tag{26}$$

4) Finally, after receiving the message $\{A_{ij}\}$ and $\{A_{ji}\}$, respectively, from the vehicle V_i and the vehicle V_j,

the SP distributes $\{A_{ij}\}$ to V_j and $\{A_{ji}\}$ to V_i at the same time. Next, the vehicle V_j checks whether each $T_{k_i}(T_{k_j}(\text{tag}_{j_m}))$ in A_{ij} is equal to the corresponding $T_{k_j}(T_{k_i}(\text{tag}_{i_n}))$. Meanwhile, the vehicle V_i computes every $T_{k_j}(T_{k_i}(\text{tag}_{i_n}))$ using the corresponding $T_{k_j}(T_{k_i}(\text{tag}_{i_n})) \oplus T_{k_j}(\text{pub}_i)$ in A_{ji} and the computed $T_{k_i}(\text{pub}_j)$ via its private key k_i. Then, it can determine whether each computed $T_{k_j}(T_{k_i}(\text{tag}_{i_n}))$ is equal to the corresponding $T_{k_i}(T_{k_j}(\text{tag}_{j_m}))$. Thus, both the vehicle V_i and the vehicle V_j will learn whether their input location sets are overlapped at the same time, which means that they both obtained the proximity testing result without exposing their location privacy.

V. Security Analysis

In this section, we first analyze the security of our proposed PSI protocol and then discuss the fairness property of the proximity testing. There are two types of adversaries that should be considered in the running protocol. 1) An external adversary $\mathcal{A}^{\text{ext}}$ represents all entities in VANET except the vehicles running the protocol. The external adversary $\mathcal{A}^{\text{ext}}$ has the ability to obtain all the messages transmitted between vehicles and the SP. Then the external adversary $\mathcal{A}^{\text{ext}}$ tries to compromise the private input of any communication party. The external adversary $\mathcal{A}^{\text{ext}}$ also wants to know whether the private input of two communication parties is overlapped or not. 2) An internal adversary $\mathcal{A}^{\text{int}}$ represents an honest-but-curious communication party in the PSI protocol. The internal adversary $\mathcal{A}^{\text{int}}$ has its secrets and can obtain all the messages exchanged between another participant and the SP. The internal adversary $\mathcal{A}^{\text{int}}$ attempts to find the private input of another participant. In addition, the internal adversary can also collude with the SP.

A security experiment $DistExp^{\text{ext}}$ is adopted to formalize the private input indistinguishability against the external adversary $\mathcal{A}^{\text{ext}}$. In this experiment, $\mathcal{A}^{\text{ext}}$ can query an oracle $\mathcal{O}$ with inputs: the low-entropy set of all possible grid areas $\mathbb{G}$, the transmitted message $\{A_i, A_j, A_{ij}, A_{ji}\}$, and the public parameters $\{\text{pub}_i, \text{pub}_j\}$ of the vehicle V_i and the vehicle V_j, trying to compromise the private inputs of vehicles. If the oracle $\mathcal{O}$ cannot distinguish the private input from the set $\mathbb{G}$, then the output of $DistExp^{\text{ext}}$ is 0; otherwise, the output is 1.

We used a security experiment $DistExp^{\text{int}}$ to formalize the private input indistinguishability against the internal adversary $\mathcal{A}^{\text{int}}$, where $\mathcal{A}^{\text{int}}$ is an honest-but-curious communication participant trying to obtain the private input of the other party. Compared to $DistExp^{\text{ext}}$, in experiment $DistExp^{\text{int}}$, the internal adversary $\mathcal{A}^{\text{int}}$ can query an oracle $\mathcal{O}$ with extra input: either vehicle V_i's private key k_i or vehicle V_j's private key k_j. If the oracle $\mathcal{O}$ cannot distinguish the other party's private input from the set $\mathbb{G}$, then the output of $DistExp^{\text{ext}}$ is 0; otherwise, the output is 1.

A. Provide Private Input Indistinguishability

Theorem 1: In our proposed PSI protocol, the advantage that an adversary can distinguish the private input of either party from the set $\mathbb{G}$ of all possible private inputs (tags) is described as follows:

$$\text{Adv}\left(\mathcal{A}^{\text{ext}}\right) = \left|\Pr\left[DistExp^{\text{ext}}\left(|\mathbb{G}|\right) = 1\right] - \frac{1}{|\mathbb{G}|}\right| \leq \varepsilon$$

where ε is negligible.

Proof: For the external adversary $\mathcal{A}^{\text{ext}}$, we assume that $\mathcal{A}^{\text{ext}}$ has compromised every tag, $\text{tag}_a \in \mathbb{G}$ and try to guess the real location of vehicles. However, without the knowledge of vehicle's private key (k_i or k_j), the external adversary $\mathcal{A}^{\text{ext}}$ cannot determine whether each of his/her guessed tags is correct or not by using the obtained messages. When the external adversary $\mathcal{A}^{\text{ext}}$ attempts to obtain vehicle's private key (k_i or k_j) from the transmitted messages $\{A_i, A_j, A_{ij}, A_{ji}\}$, he/she will face the extended CMDLP [30]. Hence, the advantage that the external adversary $\mathcal{A}^{\text{ext}}$ can distinguish the correct tag from the set $\mathbb{G}$ is negligible.

For internal adversaries, three cases are considered.

Case 1: The input set of the internal adversary $\mathcal{A}^{\text{int}}$ (the vehicle V_i) and the vehicle V_j are disjoint.

In this case, the additional knowledge of V_j's private key k_j does not enhance the advantage to the internal adversary $\mathcal{A}^{\text{int}}$ to compromise the vehicle V_i's private key k_i in comparison with external adversaries. Therefore, under Case 1, the PTCP scheme provides private input indistinguishability against internal adversaries.

Case 2: The input set of the internal adversary $\mathcal{A}^{\text{int}}$ (the vehicle V_i) and the vehicle V_j are overlapped.

In this case, the internal adversary $\mathcal{A}^{\text{int}}$ is able to obtain some of the intersection of both input sets (the internal adversary $\mathcal{A}^{\text{int}}$ and the vehicle V_j). However, the real location (x, y) of the vehicle V_j is masked by a cloaking coordinate (cx, cy), which is assigned with a grid tag, tag_i, of the vehicle V_j. As given in Section III, (cx, cy) is generated using geo-indistinguishability property. Since $\frac{P(Z|x)}{P(Z|cx)} \leq e^{\varepsilon d(x,cx)}$ is guaranteed, differential privacy of the real location (x, y) is realized. Hence, even if the internal adversary compromises the correct tag_i, he/she cannot figure out the real location of vehicle V_j using the cloaking coordinate.

Case 3: The internal adversary $\mathcal{A}^{\text{int}}$ (the vehicle V_i) colludes with SP.

In this case, the internal adversary $\mathcal{A}^{\text{int}}$ is able to obtain the correct cloaking coordinate (cx_i, cy_i) of the vehicle V_j from SP. The proof is essentially the same as in Case 2. Since the internal adversary $\mathcal{A}^{\text{int}}$ could learn nothing about the real location of the vehicle V_j from the cloaking coordinate, the collusion attack cannot succeed in our proposed scheme. Hence, the advantage that the internal adversary $\mathcal{A}^{\text{int}}$ can distinguish the correct tag from the set $\mathbb{G}$ is negligible.

According to the above analysis, the PTCP scheme provides private input indistinguishability.

B. Provides Private Input Overlapping Undecidability

A security experiment $OverlapExp^{\text{ext}}$ is employed to formalize the private input overlapping undecidability against the

external adversary $\mathcal{A}^{\text{ext}}$. In this experiment, $\mathcal{A}^{\text{ext}}$ can query an oracle $\mathcal{O}$ with inputs: the low-entropy set of all possible grid areas $\mathbb{G}$, the transmitted message $\{A_i, A_j, A_{ij}, A_{ji}\}$, and the public parameters $\{\text{pub}_i, \text{pub}_j\}$ of the vehicle V_i and the vehicle V_j, trying to determine whether the input sets are overlapped or not. If the oracle $\mathcal{O}$ cannot be determined, then the output of $OverlapExp^{\text{ext}}$ is 0; otherwise, the output is 1.

Theorem 2: In our proposed PSI protocol, the advantage that an adversary can decide the overlapping of the input sets (tags) of the vehicle V_i and the vehicle V_j is described as follows:

$$\text{Adv}\left(\mathcal{A}^{\text{ext}}\right) = \left|\Pr\left[OverlapExp^{\text{ext}}\left(|\mathbb{G}|\right) = 1\right] - \frac{1}{2}\right| \leq \varepsilon$$

where ε is negligible.

Proof: To determine the overlapping of the input sets of the vehicle V_i and the vehicle V_j, the external adversary $\mathcal{A}^{\text{ext}}$ queries the oracle $\mathcal{O}$ with inputs: the low-entropy set of all possible grid areas $\mathbb{G}$, the message $\{A_i, A_{ij}\}$ from the vehicle V_i, the message $\{A_j, A_{ji}\}$ from the vehicle V_j, and the public parameters $\{\text{pub}_i, \text{pub}_j\}$. However, without the knowledge of secret key (k_i and k_j) of the vehicle V_i and the vehicle V_j, the external adversary $\mathcal{A}^{\text{ext}}$ cannot obtain the input sets of both the vehicle V_i and the vehicle V_j due to the CMDLP [30]. In addition, the external adversary $\mathcal{A}^{\text{ext}}$ cannot construct a correct $T_{k_j}(\text{pub}_i)$ without the knowledge of the private key k_j of vehicle V_j. So the oracle $\mathcal{O}$ cannot determine whether the input sets are overlapped or not, and it will output 0. That means the external adversary $\mathcal{A}^{\text{ext}}$ fails to determine whether the two input sets from the vehicle V_i and the vehicle V_j are overlapped or not. Therefore, the PTCP scheme provides private input overlapping undecidability.

C. Resistance of Collusion Attacks

According to the adversary model in our PTCP scheme, the internal adversary may attempt to collude with the SP to obtain the location information of another communication party. Assume that Bob is the internal adversary and it launched collusion attacks with the SP to get the location information of Alice (another communication party). However, although Bob is able to obtain the correct cloaking coordinate through collusion with the SP, it still cannot get the real location of Alice from the cloaking coordinate. The proof is essentially the same as in Case 3 Section V-A. Therefore, the PTCP scheme can resist collusion attacks.

D. Fairness Assurance

The fairness property in proximity testing means that the sender and receiver can obtain the result of proximity testing at the same time. In our proposed scheme, only when the SP has received the message A_{ij} and A_{ji} from V_i and V_j could it then distribute A_{ij} to V_j and A_{ji} to V_i, respectively. This process ensures that both vehicles are informed of the testing results at the same time. Therefore, the PTCP scheme provides fairness property.

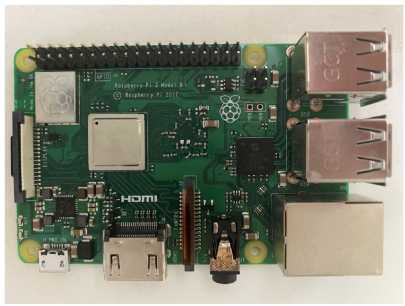

Fig. 2. Raspberry Pi 3B+.

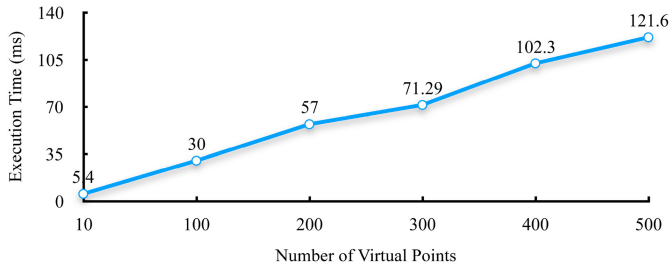

Fig. 3. Execution time of virtual points generation with different scales.

VI. Performance Evaluation

This section introduces the detailed experimental setup and compares our PTCP scheme with other related schemes [12], [23] in terms of computational costs.

In our experiments, the PTCP scheme and other two related works [12], [23] were all simulated on Apple clang version 11.0.0 using C++ under the OpenSSL Library [31], GMP Library [32], and PBC Library [33]. The hardware platform is a MacBook Pro with Intel (R) Core (TM) i7-8569 U CPU @ 2.8 GHz, 16 GB RAM, and the operating system is macOS Catalina 10.15.7. To simulate the devices with limited computational power, the experiments were also performed on a Raspberry Pi 3B+ with a 1-GB LPDDR2 SDRAM and a BCM2837B0 system-on-chip (SoC) of 1.4 GHz frequency as given in Fig. 2.

In our PTCP scheme, the computational costs mainly include two parts: virtual points generation and PSI execution.

A. Virtual Points Generation

We first simulated the process of generating virtual points in our PTCP scheme. In our experiments, the parameters ε and σ were set to 0.01 and 0.1, respectively. Fig. 3 illustrates the execution time of virtual points generation with different scales. As shown in Fig. 3, the execution time of this process smoothly goes up from 54 to 121.6 ms as the number of virtual points increases from 10 to 500. That means the generation of the virtual points of our proposed method is still efficient even if the scale of virtual points increases to a relatively high level.

TABLE II
EXECUTION TIME OF CRYPTOGRAPHIC OPERATIONS

Cryptographic Operations	Execution Time	
	Intel i7-8569U (ns)	BCM2837B0 (ns)
EC Point Multiplication(128 bits)	204009	5966983
EC Point Multiplication(160 bits)	237009	7417198
EC Point Multiplication(256 bits)	435009	11697781
EC Point Multiplication(512 bits)	803834	23440564
EC Point Addition(128 bits)	4009	165217
SHA1(128 bits)	2009	68551
AES128 Encryption(128 bits)	13009	50061
AES128 Decryption(128 bits)	16609	81415
Chebyshev Polynomial(128 bits)	179000	2573966
Chebyshev Polynomial(160 bits)	215000	3812347
Chebyshev Polynomial(256 bits)	329000	5489946
Chebyshev Polynomial(512 bits)	653000	11583372

B. Chebyshev-Based PSI

To evaluate our proposed Chebyshev-based PSI, we first analyze the efficiency of our proposed Chebyshev polynomial algorithm by simulating different cryptographic operations including Chebyshev polynomial (128, 160, 256, and 512 b), EC point multiplication (128, 160, 256, and 512 b), EC point addition (128 b), SHA1 (128 b), and advanced encryption standard (AES) encryption/decryption (128 b). In our experiments, each cryptographic operation was performed 100 times and the average value was computed to obtain a more convincing result. Table II presents the experimental results of different cryptographic operations on the Raspberry Pi and the MacBook. Obviously, the Chebyshev polynomial operations are more efficient than the EC point multiplication operation, and the execution time of the Chebyshev polynomial operation is also acceptable on Raspberry Pi 3B+.

Based on our proposed Chebyshev polynomials algorithm, we performed some simulations on our proposed PSI protocol. In our proposed PSI protocol, the input set size of vehicle V_i and the vehicle V_j is n and m, respectively. Then the vehicle V_j needs to perform $m+1$ Chebyshev polynomial operations to obtain ID_i and A_j. For the vehicle V_i, n Chebyshev polynomial operations are required to construct A_i, and another n Chebyshev polynomial operations are needed to obtain $T_{k_i}(T_{k_j}(\mathrm{tag}_{j_m}))$. In addition, the vehicle V_i and the vehicle V_j are required to perform m and $2n$ Chebyshev polynomial operations to construct A_{ij} and A_{ji}, respectively, at the same time.

In our simulation experiments, we set $m=n$, the total time of our proposed PSI protocol is close to $(5m+1)T_c$, where T_c denotes the execution time of a Chebyshev polynomial operation. Table III illustrates the execution time of our proposed PSI protocol with different set sizes. As shown in Table III, our proposed PSI protocol costs 208.89 ms with set size 2^8 and 3258.43 ms with set size 2^{12}. When the set size increases to 2^{16}, the execution time is 59 341.22 ms. So, our proposed PSI protocol satisfies the low energy requirements in real application scenarios.

TABLE III
EXECUTION TIME OF OUR PROPOSED PSI PROTOCOL

Set size	2^8	2^{12}	2^{16}
Execution time (ms)	208.89	3285.43	59341.22

TABLE IV
COMPARISON OF COMPUTATIONAL COSTS

Search Area (m)	Execution Time (ms) of PPPT [23]	Execution Time (ms) of EPPD [12]	Execution Time (ms) of Ours
200	487.65	975.31	274.44
300	528.19	1109.20	406.59
400	609.88	1173.48	586.74
500	729.50	2480.28	620.05
600	1067.09	2522.22	630.21
700	1422.44	2512.98	731.87
800	1625.74	2786.98	853.84
900	1837.85	4594.62	935.16
1000	2276.16	4688.90	1016.48

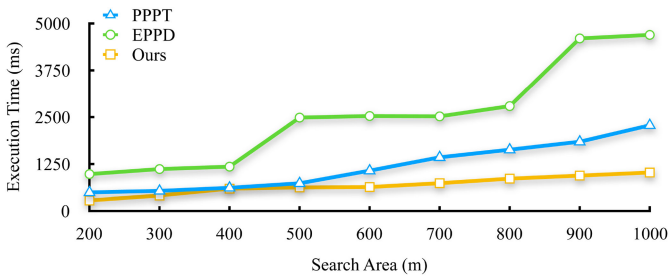

Fig. 4. Execution time comparison between our PTCP scheme and others.

C. Computational Costs Comparison With Other Proximity Testing Scheme

In addition, we further performed several simulation experiments to compare the execution time of the related schemes [12], [23] and our PTCP scheme. In our experiments, 1000 users were randomly deployed in a square area with a side length of 10 km. As shown in Table IV and Fig. 4, the execution time of the schemes of Qiu *et al.* [23] and Huang *et al.* [12] keeps stable when the search area is between 200 and 400 m, while that of our PTCP scheme increases from 274.44 to 586.74 ms in this duration. When the search area is 400 m, the execution time of our scheme is close to that of the scheme of Qiu *et al.* [23] at around 600 ms. When the search range is between 400 and 1000 m, the execution time of Qiu *et al.*'s scheme [23] increases steadily from 609.88 to 2276.16 ms. Notably, the execution time of the scheme of Huang *et al.* [12] remains the highest overall among these three schemes ([12], [23], and our PTCP scheme). Besides, there are two sharp increases of this scheme's

execution time [12]: one is at 400 m and another is at 800 m. Obviously, our scheme requires less execution time especially when the search range increases. Therefore, our PTCP scheme is more suitable for practical use since the search range for proximity testing is generally not limited to 1000 m in real applications.

D. Communication Cost

In our experiments, the users' ID is 64 b, the timestamp is 32 b, the output of AES encryption is 128 b, and the output of Chebyshev polynomial is 128 b. During vehicle location uploading phase, each vehicle sends the message $\{C_i, \mathrm{ID}_i, T_\gamma\}$ to the SP, and the communication cost for each vehicle is 224 b. In proximity testing phase, if a vehicle V_i initiates proximity testing, it requires to transmit 224 b messages $\{\mathrm{CR}_i, \mathrm{ID}_i, T_\gamma\}$ to the SP. Furthermore, when a vehicle V_i wants to perform proximity testing with a vehicle V_j, it needs to send 256 b message $\{T_{a_j}(v), C_{cj}\}$ to V_j. After that, during PSI, the transmitted message between the vehicle V_i and the vehicle V_j is $\{A_i, A_j, A_{ij}, A_{ji}\}$. So the communication cost is $256(m+n)$ b, where m, n refer to the input set size of V_i and V_j, respectively. Therefore, the total communication cost for proximity testing phase is $256(m+n)+480$ b.

E. Accuracy of Our PTCP Scheme

In our PTCP scheme, geo-indistinguishability and grid dividing are utilized to protect the real location. But they may also affect the accuracy of proximity testing in some circumstances.

Geo-indistinguishability affects the result of proximity testing via the cloaking coordinate and the cloaking testing range. In our scheme, during vehicle location uploading phase, each vehicle V_i computes the cloaking coordinate (cx_i, cy_i) using its real location (x, y). Then, in proximity testing phase, the vehicle V_i computes the cloaking testing range R_i using the radius l_{SA} of its search area and the control parameter σ_i. After that, the SP computes the distance from (cx_i, cy_i) to other vehicles. Therefore, the SP can figure out all vehicles within the cloaking testing range according to the computed distance. However, since the real location and the search area radius are perturbed according to geo-indistinguishability, some vehicles that should be considered may not be included in the input set of proximity testing.

To analyze the effects of geo-indistinguishability, we simulated the proximity process with different $\sigma_i \in [0, 1)$. In the experiments, 10 000 users were randomly distributed in a 40 km $\times$ 40 km square area. Meanwhile, we set the privacy budget $\varepsilon = 2$, the side length of a grid $\mu = 500$ m, and the search range radius for each vehicle $l_{\mathrm{SA}} = 2500$ m. As shown in Fig. 5, the accuracy of proximity testing keeps high and smoothly increases to 99.94% with σ_i going up to 0.99.

Furthermore, grid dividing also affects the accuracy of proximity testing via the side length of each grid and the search area radius of vehicles. Fig. 6 shows an example of when the result of proximity testing is incorrect. As shown in Fig. 6, the search area of two vehicles is not overlapped, but their location sets possess one same grid. Under this case, our PTCP scheme will obtain an incorrect result that their location sets are overlapped.

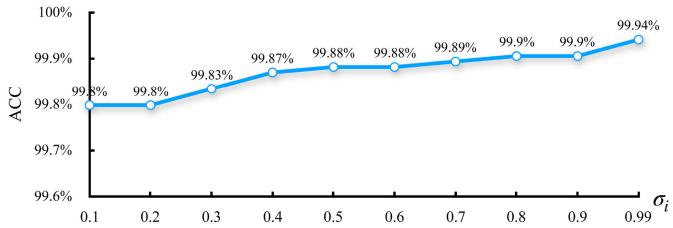

Fig. 5. Average accuracy (ACC) as σ_i increases.

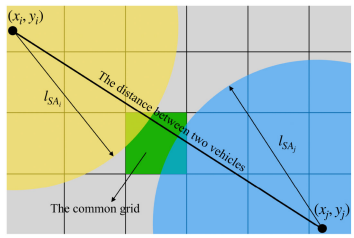

Fig. 6. Examples when the result of proximity testing is incorrect.

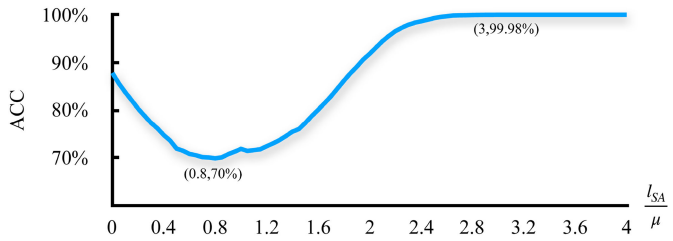

Fig. 7. Average accuracy (ACC) as $\frac{l_{\mathrm{SA}}}{\mu}$ increases.

To analyze the effects of grid dividing on the testing accuracy, massive simulation experiments were performed with different $\frac{l_{\mathrm{SA}}}{\mu} \in [0, 4]$, where l_{SA} refers to the search range radius and μ refers to the grid side length. For each $\frac{l_{\mathrm{SA}}}{\mu}$, the average accuracy (ACC) is calculated by counting the number of errors among $1,000,000$ proximity testing. As shown in Fig. 7, when $\frac{l_{SA}}{\mu}$ increases from 0 to 0.8, ACC decreases to 70.0%. But when $\frac{l_{SA}}{\mu}$ is over 0.8, ACC stably goes up and keeps high at more than 99.98% ($\frac{l_{SA}}{\mu} > 3$).

Notably, when the search area of two vehicles is overlapped, their location sets must share at least one grid. Under this case, our proposed PTCP scheme can reach accurate testing. In addition, the experimental results also show that when $\frac{l_{\mathrm{SA}}}{\mu}$ increases, the input size goes up. Then, location privacy is better guaranteed, but the computational and communication overhead will cost more. Therefore, in real application scenarios, the search area radius l_{SA} and the grid side length μ should be carefully chosen. Our experimental results show that when $\frac{l_{\mathrm{SA}}}{\mu}$ is 3, a tradeoff among the security, the accuracy, the computational cost, and the communication cost will be reached.

VII. Conclusion

In this article, we proposed an efficient privacy-preserving proximity testing scheme for VANETs. In our design, geo-indistinguishability was employed to generate virtual points that are used as the input set of PSI. Then, a novel Chebyshev-based PSI scheme was constructed for proximity testing to achieve location privacy protection with low energy consumption. In addition, the fairness requirement of proximity testing was satisfied in our scheme since the testing results were sent to both communication parties at the same time. The security analysis demonstrated that our scheme provides location protection and can resist collusion attacks. The experimental results showed that our scheme outperforms other related schemes and is a robust proximity testing method with privacy protection for VANETs.

References

[1] V. K. Yadav, S. Verma, and S. Venkatesan, "Efficient and secure location-based services scheme in VANET," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13 567–13 578, Nov. 2020.

[2] B. Luo, X. Li, J. Weng, J. Guo, and J. Ma, "Blockchain enabled trust-based location privacy protection scheme in VANET," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2034–2048, Feb. 2020.

[3] M. Li, N. Cao, S. Yu, and W. Lou, "FindU: Privacy-preserving personal profile matching in mobile social networks," in *Proc. IEEE Conf. Comput. Commun.*, 2011, pp. 2435–2443.

[4] W. He, X. Liu, and M. Ren, "Location cheating: A security challenge to location-based social network services," in *Proc. 31st Int. Conf. Distrib. Comput. Syst*., 2011, pp. 740–749.

[5] M. Li *et al.*, "Large-scale third-party library detection in android markets," *IEEE Trans. Softw. Eng.*, vol. 46, no. 9, pp. 981–1003, Sep. 2020.

[6] Y. Zheng, M. Li, W. Lou, and Y. T. Hou, "Location based handshake and private proximity test with location tags," *IEEE Trans. Dependable Secure Comput.*, vol. 14, no. 4, pp. 406–419, Jul./Aug. 2017.

[7] A. Narayanan, N. Thiagarajan, M. Lakhani, M. Hamburg, and D. B. Boneh, "Location privacy via private proximity testing," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2011, pp. 1–17.

[8] P. Kotzanikolaou, C. Patsakis, E. Magkos, and M. Korakakis, "Lightweight private proximity testing for geospatial social networks," *Comput. Commun.*, vol. 73, pp. 263–270, 2016.

[9] K. Choi and M. Kim, "Locap: Privacy-preserving location proximity protocol," in *Proc. 9th Int. Conf. Ubiquitous Future Netw.*, 2017, pp. 994–998.

[10] M. N. Sakib and C.-T. Huang, "Privacy preserving proximity testing using elliptic curves," in *Proc. 26th Int. Telecommun. Netw. Appl. Conf.*, 2016, pp. 121–126.

[11] L. Šikšnys, J. R. Thomsen, S. Šaltenis, M. L. Yiu, and O. Andersen, "A location privacy aware friend locator," in *Proc. Int. Symp. Spatial Temporal Databases*, 2009, pp. 405–410.

[12] C. Huang, R. Lu, H. Zhu, J. Shao, A. Alamer, and X. Lin, "EPPD: Efficient and privacy-preserving proximity testing with differential privacy techniques," in *Proc. IEEE Int. Conf. Commun.*, 2016, pp. 1–6.

[13] S. Zou, J. Xi, H. Wang, and G. Xu, "CrowdBLPS: A blockchain-based location-privacy-preserving mobile crowdsensing system," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 4206–4218, Jun. 2020.

[14] H. Liao, Y. Mu, Z. Zhou, M. Sun, Z. Wang, and C. Pan, "Blockchain and learning-based secure and intelligent task offloading for vehicular fog computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4051–4063, Jul. 2021.

[15] Y. Yang, L. Wei, J. Wu, C. Long, and B. Li, "A blockchain-based multi-domain authentication scheme for conditional privacy preserving in vehicular ad-hoc network," *IEEE Internet Things J.*, early access, doi: 10.1109/JIOT.2021.3107443.

[16] B. Li, R. Liang, D. Zhu, W. Chen, and Q. Lin, "Blockchain-based trust management model for location privacy preserving in VANET," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3765–3775, Jun. 2021.

[17] F. L. Ţiplea and C. Hristea, "PUF protected variables: A solution to RFID security and privacy under corruption with temporary state disclosure," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 999–1013, 2021, doi: 10.1109/TIFS.2020.3027147.

[18] G. Bansal, N. Naren, V. Chamola, B. Sikdar, N. Kumar, and M. Guizani, "Lightweight mutual authentication protocol for V2G using Physical Unclonable Function," *IEEE Trans. Veh. Technol.*, vol. 69, no. 7, pp. 7234–7246, Jul. 2020.

[19] H. Xiong, Y. Bao, X. Nie, and Y. I. Asoor, "Server-aided attribute-based signature supporting expressive access structures for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1013–1023, Feb. 2020.

[20] W. Wang, A. Liu, Z. Li, X. Zhang, Q. Li, and X. Zhou, "Protecting multi-party privacy in location-aware social point-of-interest recommendation," *World Wide Web*, vol. 22, no. 2, pp. 863–883, 2019.

[21] F. Chen, X. Liao, T. Xiang, and H. Zheng, "Security analysis of the public key algorithm based on Chebyshev polynomials over the integer ring ZN," *Inf. Sci.*, vol. 181, no. 22, pp. 5110–5118, 2011.

[22] D. Yoshioka, "Security of public-key cryptosystems based on Chebyshev polynomials over $\mathbb{Z}/p^k\mathbb{Z}$," *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 67, no. 10, pp. 2204–2208, Oct. 2020.

[23] Y. Qiu and M. Ma, "A privacy-preserving proximity testing for location-based services," in *Proc. IEEE Glob. Commun. Conf.*, 2018, pp. 1–6.

[24] A. Abadi, S. Terzis, R. Metere, and C. Dong, "Efficient delegated private set intersection on outsourced private datasets," *IEEE Trans. Dependable Secure Comput.*, vol. 16, no. 4, pp. 608–624, Jul./Aug. 2019.

[25] X. Wang, X. Kuang, J. Li, J. Li, X. Chen, and Z. Liu, "Oblivious transfer for privacy-preserving in VANET's feature matching," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4359–4366, Jul. 2021.

[26] Y. Qian, X. Xia, and J. Shen, "A profile matching scheme based on private set intersection for cyber-physical-social systems," in *Proc. IEEE Conf. Dependable Secure Comput.*, 2021, pp. 1–5.

[27] Y. Wang, Q. Huang, H. Li, M. Xiao, S. Ma, and W. Susilo, "Private set intersection with authorization over outsourced encrypted datasets," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 4050–4062, Jul. 2021, doi: 10.1109/TIFS.2021.3101059.

[28] X. Dong, T. Zhang, D. Lu, G. Li, Y. Shen, and J. Ma, "Preserving geo-indistinguishability of the primary user in dynamic spectrum sharing," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 8881–8892, Sep. 2019.

[29] M. J. Freedman, K. Nissim, and B. Pinkas, "Efficient private matching and set intersection," in *Proc. Int. Conf. Theory Appl. Cryptogr. Tech.*, 2004, pp. 1–19.

[30] D. Abbasinezhad-Mood, A. Ostad-Sharif, S. M. Mazinani, and M. Nikooghadam, "Provably secure escrow-less Chebyshev chaotic map-based key agreement protocol for vehicle to grid connections with privacy protection," *IEEE Trans. Ind. Informat.*, vol. 16, no. 12, pp. 7287–7294, Dec. 2020.

[31] "Openssl cryptography and ssl/tls toolkit," [EB/OL]. Accessed: Jul. 14, 2021. [Online]. Available: https://www.openssl.org/

[32] "The gnu multiple precision arithmetic library," [EB/OL]. Accessed: Jul. 14, 2021. [Online]. Available: https://gmplib.org/

[33] "Dan boneh's publications by topic," [EB/OL]. Accessed: Jul. 14, 2021. [Online]. Available: http://crypto.stanford.edu/ dabo/pubs/pubsbytopic.html

Liping Zhang received the Ph.D. degree in information security from the Huazhong University of Science and Technology, Wuhan, China, in 2009.

She is currently an Associate Professor of Information and Network Security with the China University of Geosciences, Wuhan. She has authored or coauthored more than 30 research papers, most of which are refereed international journal papers including IEEE/ACM/IET journal papers. She is the Principal grant holder of three externally funded research projects. Her research interests include network security, key management and distribution, and privacy protection.

Wenhao Gao received the B.Sc. degree in information security, in 2020, from the China University of Geosciences, Wuhan, China, where he is currently working toward the postgraduate research degree in computer science.

His research interests include secure multiparty computation, communications security, and network security.

Shukai Chen received the B.Sc. degree in network engineering from PLA Army Engineering University, Nanjing, China, in 2020. He is currently working toward the postgraduate research degree in electronic and information engineering (computer science) with the China University of Geosciences, Wuwan, China.

His research interests include ECG authentication, communications security, and network security.

Wei Ren (Member, IEEE) received the Ph.D. degree in computer science from the Huazhong University of Science and Technology, Wuhan, China, in 2006.

He is currently a Full Professor with the School of Computer Science, China University of Geosciences, Wuhan. He was with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL, USA, in 2007 and 2008, the School of Computer Science, University of Nevada Las Vegas, Las Vegas, NV, USA, in 2006 and 2007, and the Department of Computer Science, The Hong Kong University of Science and Technology, Hong Kong, in 2004 and 2005. He has authored or coauthored more than 100 refereed papers, one monograph, and four textbooks.

Dr. Ren is the recipient of ten patents and five innovation awards. He is a Distinguished Member of the China Computer Federation.

Kim-Kwang Raymond Choo (Senior Member, IEEE) received the Ph.D. degree in information security from the Queensland University of Technology, Brisbane, Australia, in 2006.

He currently holds the Cloud Technology Endowed Professorship with The University of Texas at San Antonio (UTSA), San Antonio, TX, USA.

Dr. Choo is the founding Coeditor-in-Chief for *ACM Distributed Ledger Technologies: Research and Practice*, and the founding Chair of IEEE TEMS Technical Committee on Blockchain and Distributed Ledger Technologies. He is an ACM Distinguished Speaker and IEEE Computer Society Distinguished Visitor (2021–2023), and a Web of Science's Highly Cited Researcher (Computer Science—2021, Cross-Field—2020). In 2015, he and his team won the Digital Forensics Research Challenge organized by the University of Erlangen-Nuremberg, Erlangen, Germany. He was the recipient of the 2019 IEEE Technical Committee on Scalable Computing Award for Excellence in Scalable Computing (Middle Career Researcher), the British Computer Society's 2019 Wilkes Award Runner-up, the Fulbright Scholarship in 2009, the 2008 Australia Day Achievement Medallion, and the British Computer Society's Wilkes Award in 2008. He is also the recipient of best paper awards from *IEEE Systems Journal* in 2021, *IEEE DSC* in 2021, *IEEE Consumer Electronics Magazine* for 2020, *Journal of Network and Computer Applications* for 2020, *EURASIP Journal on Wireless Communications and Networking* in 2019, *IEEE TrustCom 2018*, and *ESORICS 2015*; the IEEE Blockchain 2019 Outstanding Paper Award; and the best student paper awards from *Inscrypt 2019* and *ACISP 2005*.

Neal N. Xiong (Senior Member, IEEE) received the Ph.D. degree in sensor system engineering from Wuhan University, Wuhan, China, in 2007, and the Ph.D. degree in dependable communication networks from the Japan Advanced Institute of Science and Technology, Nomi, Japan, in 2008.

He is currently an Associate Professor (5th year) with the Department of Mathematics and Computer Science, Northeastern State University, OK, USA. Before he attended Northeastern State University, for about ten years, he worked with Georgia State University, Atlanta, GA, USA; Wentworth Technology Institution, Boston, MA, USA; and Colorado Technical University, Colorado Springs, CO, USA, where he was a Full Professor for about five years. He has authored or coauthored more than 200 international journal papers and more than 100 international conference papers. Some of his works were published in IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE or ACM transactions, *ACM Sigcomm* workshop, IEEE INFOCOM, IEEE International Conference on Distributed Computing Systems, and International Parallel and Distributed Processing Symposium. His research interests include cloud computing, security and dependability, parallel and distributed computing, networks, and optimization theory.

Dr. Xiong has been a General Chair, Program Chair, Publicity Chair, Program Committee member, and Organizing Committee Member of more than 100 international conferences and a Reviewer for about 100 international journals, including IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS (Park: A/B/C), IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE TRANSACTIONS ON MOBILE COMPUTING, and IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS. He is currently serving as an Editor-in-Chief, Associate Editor, or Editorial Board Member for more than ten international journals, including Associate Editor for IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, Associate Editor for *Information Science*, Editor-in-Chief for *Journal of Internet Technology (JIT)*, and Editor-in-Chief for *Journal of Parallel & Cloud Computing (PCC)*, and as a Guest Editor for more than ten international journals, including *Sensor*, *Wireless Networks*, and *Mobile Networks and Applications*. He was the recipient of the Best Paper Award in the 10th IEEE International Conference on High-Performance Computing and Communications (HPCC-08) and the Best Student Paper Award in the 28th North American Fuzzy Information Processing Society Annual Conference (NAFIP S2009). He is the Chair of the "Trusted Cloud Computing" Task Force, IEEE Computational Intelligence Society (CIS).