

# FLIP: An Efficient Privacy-preserving Protocol for Finding Like-minded Vehicles on the Road

Rongxing Lu<sup>†</sup>, Xiaodong Lin<sup>‡</sup>, Xiaohui Liang<sup>†</sup>, and Xuemin (Sherman) Shen<sup>†</sup>

<sup>†</sup>Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

<sup>‡</sup>Faculty of Business and Information Technology, University of Ontario Institute of Technology,  
Oshawa, Ontario, Canada L1H 7K4

Email: {rxlu, x27liang, xshen}@bbr.uwaterloo.ca; xiaodong.lin@uoit.ca

**Abstract**—Vehicle chatting is one of the most promising applications in VANETs, which allows like-minded vehicles to chat on the topics of common interest on the road. However, there exist some newly emerging privacy challenging issues in vehicle chatting application, such as how to find a like-minded vehicle on the road and how to prevent one’s interest privacy (IP) from others who are not like-minded? In this paper, to tackle these challenging issues, we propose an efficient privacy-preserving finding like-minded vehicle protocol (FLIP), and apply the provable security technique to demonstrate its security. In addition, extensive simulations are also conducted to examine its practical considerations, i.e., the relation between the expected IP-preserving level and the delay of finding like-minded vehicles on the road.

**Keywords**— VANET, vehicle chatting, interest privacy, finding like-minded vehicle, provable security

## I. INTRODUCTION

Vehicular Ad hoc Networks (VANETs), as a special instantiate of mobile ad hoc network, have been subject to extensive research efforts not only from the government, but also from the academia and automobile industry in recent years [1]. In VANET, each vehicle is equipped with OnBoard Unit (OBU) device, which allows them communicate with each other, i.e., vehicle-to-vehicle (V-2-V) communication, as well as to the Roadside Units (RSUs), i.e., vehicle-to-infrastructure (V-2-I) communication. Compared with traditional ad hoc network, the hybrid of V-2-V and V-2-I communications makes VANETs more promising, and can provide a board of safety-related (e.g., emergence report, collision warning) and non-safety-related (vehicle chatting, downloading and sharing files on the road) applications close to our daily lives. Due to these salient applications, VANETs have increasingly attractive to the public.

Vehicle chatting is one of the most promising applications in VANETs, which allows vehicles moving along the same road to chat with each other on some topics of common interest, for the purpose of passing the time during the commute or asking for a help on the road [2]. However, the success of vehicle chatting application in VANET still hinges up the fully understanding and managing the security and privacy challenges that the public concerns, for example, the identity privacy, location privacy, and interest privacy. Because VANET is usually implemented in civilian scenarios, where the locations of vehicles are tightly related to people who

drive them. If the vehicle chatting application discloses the vehicle’s identity privacy and location privacy, it can’t be accepted by the public. In recent years, these two kinds of privacy have been deeply discussed in VANETs [1], [3], [4]. However, to the best of our knowledge, the interest privacy, as a special privacy requirement in vehicle chatting application, has not been explored. Therefore, how to identify a vehicle who is like-minded and establish a shared session key for secure chatting, and how to prevent other vehicles who are not like-minded from knowing one vehicle’s interest have become two newly emerging privacy challenges in vehicle chatting application.

In this paper, to address the above challenging privacy issues in vehicle chatting application, we propose an efficient privacy-preserving finding like-minded vehicle protocol, called FLIP, which allows two vehicles with the common interest to identify each other and establish a shared session key, and at the same time, protects their Interest-Privacy (IP) from other vehicles who don’t have the same interest on the road. Specifically, the contribution of this paper are two-fold.

- Firstly, we propose an efficient IP-preserving FLIP protocol aiming at vehicle chatting application in VANET, and formalize its security model as well. Then, we apply the provable security technique [5] to validate its security within the defined model.
- Secondly, we develop a custom simulator built in Java to measure the relation between the IP-preserving level and the delay for finding the like-minded vehicle. The extensive simulation results show that, after setting a required IP-preserving level, a vehicle can find a like-minded vehicle within an expected time.

The remainder of this paper is organized as follows. In Section II, we introduce the system model and design goal, as well as the security model of FLIP. In Section III, we recall some preliminaries including the elliptic curve group and complexity assumption. Then, we present our IP-preserving FLIP protocol in Section IV, followed by its security analysis and performance evaluation in Section V and Section VI, respectively. We also discuss the related work in Section VII. Finally, we draw our conclusions in Section VIII.

## II. SYSTEM MODEL AND DESIGN GOAL

In this section, we define the problem by formalizing the system model and identifying our design goal.

### A. System Model

We consider a VANET in a city environment, which consists of a large number of vehicles  $\mathcal{V} = \{V_1, V_2, \dots\}$  and a single offline trusted authority (TA), as shown in Fig. 1. Since we confine our problem to the scenario where vehicles find like-minded vehicles with common interest on the road without the assistance of RSUs, we do not include RSUs in our current model, although they are still deployed to support V-2-I communication.

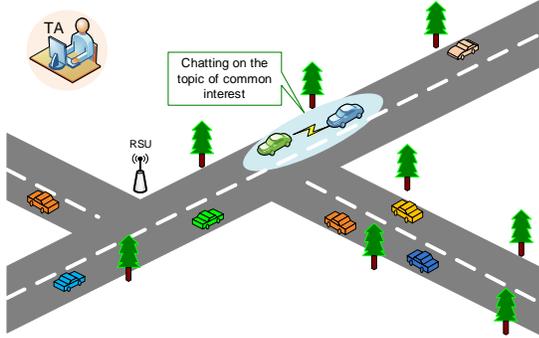


Fig. 1. System model under consideration

- Trust Authority (TA): TA is a trustable and powerful entity. The responsibility of TA is in charge of management of the whole network, for example, initializing the system, registering the vehicles in the system by assigning a finite set of pseudo-IDs and the corresponding key materials to each vehicle. Note that TA is an offline entity, which is not directly involved in the V-2-V communications.
- Vehicles  $\mathcal{V} = \{V_1, V_2, \dots\}$ : Each vehicle  $V_i \in \mathcal{V}$  is equipped with the OBU device, which allows them to communicate with each other for sharing some information of common interest. Different from the mobile nodes in the general ad hoc network, the OBU device in VANET has no power-constrained issue and at the same time, is equipped with powerful computational and communication capabilities. According to [6], the medium used for communications among the neighboring vehicles is 5.9 GHz Dedicated Short Range Communication (DSRC) identified as IEEE 802.11p, and the transmission range of each vehicle is 300 m. When two vehicles  $V_a$ , and  $V_b \in \mathcal{V}$  are within their transmission range, they can chat on the topics of common interest on the road.

### B. Design Goal

1) *Security requirements and design goal*: Security and privacy are always of vital importance to the flourish of vehicle chatting application in VANET. Without the guarantee of vehicle's privacy including identity privacy [7], location

privacy and interest privacy, vehicle chatting application can't be widely accepted by the public. Therefore, it is essential to protect vehicle's privacy. Specifically, the following security requirements should be ensured in vehicle chatting application: i) vehicle's real identity should be protected; ii) vehicle's location privacy should be guaranteed; and iii) vehicle's interests should be protected against others who doesn't have the common interest.

In regard of the former two security requirements, each vehicle can use pseudo-ID to conceal the real identity, and periodically change multiple pseudo-IDs to achieve the location privacy [1], [3]. However, for the third security requirement, vehicles should use some IP-preserving protocols to find other vehicle who has the common interest on the road. Concretely, when a vehicle  $V_a$  wants to talk with another vehicle  $V_b$  nearby, if  $V_b$  has the common interest with  $V_a$ ,  $V_a$  and  $V_b$  can establish a shared session key used for secure chatting on the topics of common interest. However, if  $V_b$  doesn't have the common interest with  $V_a$ , neither  $V_a$  nor  $V_b$  can know the counterpart's interest.

To satisfy the above security requirements, our design goal is to develop an efficient privacy-preserving finding like-minded vehicle protocol (FLIP) in VANET environment. With FLIP, vehicles who have the common interest can establish a shared session key without violating IP to others who have non-common interest. To subtly check the security of FLIP protocol in terms of IP-preserving, we should formally define its security model as follows.

2) *Security model of FLIP*: To model all possible leakages of IP in finding like-minded vehicle protocol on the road, we define the security model of FLIP by borrowing some ideas from security model of authenticated key exchange (AKE) protocols [8] to describe some possible attacks. Specifically, in the model, the vehicles do not deviate from the FLIP protocol, while an adversary  $\mathcal{A}$ , whose attack capabilities are modelled by a set of pre-defined oracle queries, can passively monitor and/or actively control all the inter-vehicle communications. We assume that two vehicles  $V_a$  and  $V_b$  participate in FLIP for common interest  $I_\alpha \in \mathcal{I} = \{I_1, I_2, \dots, I_k\}$ . Each of them has several instances called oracles involved in distinct executions of FLIP, where the common interest  $I_\alpha$  varies in different executions. We denote an instance  $s$  of  $V_i \in \{V_a, V_b\}$  by  $\Pi_{V_i}^s$  for an integer  $s \in \mathbb{N}$ , and use the notation  $\Pi_{V_a, V_b}^s$  to define the  $s$ -th instance  $V_a$  executing FLIP with  $V_b$  on the common interest  $I_\alpha^s$ , where  $\alpha \in \mathbb{N}$  and  $1 \leq \alpha \leq k$ .

ADVERSARIAL MODEL: We allow the adversary  $\mathcal{A}$  to access to all transcripts in the FLIP. All oracles only communicate with each other via  $\mathcal{A}$ . The adversary  $\mathcal{A}$  can replay, modify, delay, interleave or delete transcripts.

- $\text{Execute}(\Pi_{V_a, V_b}^s)$ : This query models passive attacks, where  $\mathcal{A}$  accesses an honest execution of FLIP between  $V_a$  and  $V_b$  by eavesdropping.
- $\text{SendReq}(\Pi_{V_a}^s, *)$ : This query models  $\mathcal{A}$  can send a transcript  $m$  to the requestor-instance  $\Pi_{V_a}^s$ , and get back the answer of  $\Pi_{V_a}^s$  by following FLIP. The adversary  $\mathcal{A}$  can use this query to perform active attacks by modifying

### III. PRELIMINARIES

In this section, we introduce some preliminaries, including some used notations, elliptic curve group and complex assumption, on basis of our proposed FLIP protocol.

#### A. Notations

Let an integer  $k \in \mathbb{N}$ , then  $1^k$  is the string of  $k$  1s. If  $x, y$  are two strings, then  $x||y$  is the concatenation of  $x$  and  $y$ . If  $S$  is a finite set,  $s \xleftarrow{R} S$  denotes sampling an element  $x$  uniformly at random from  $S$ . If  $\mathcal{A}$  is a randomized algorithm,  $y \leftarrow \mathcal{A}(x_1, x_2, \dots)$  means that  $\mathcal{A}$  has inputs  $x_1, x_2, \dots$  and outputs  $y$ .

#### B. Elliptic Curve Group and Complex Assumption

Let  $p$  be a large prime, (eg.  $p > 3$ ). Randomly choose two field elements  $a, b \in \mathbb{F}_p$  and define the elliptic curve equation  $\mathbf{E} : y^2 = x^3 + ax + b \pmod{p}$  over  $\mathbb{F}_p$ , where  $4a^3 + 27b^2 \neq 0 \pmod{p}$ . The cardinality of  $\mathbf{E}$  should be divisible by a large prime number with regard to the security issue raised by Pohlig and Hellman [9]. Let  $P = (x_P, y_P)$  be a generator point over  $\mathbf{E}(\mathbb{F}_p)$  whose order is a large prime number  $q$ , where  $P \neq \mathcal{O}$ , and  $\mathcal{O}$  denotes the point at infinity. Then,  $\mathbb{G} = \langle P \rangle$  is an efficient elliptic curve group of order  $q$ . In the following, we define the quantitative notion of the complexity of the problem underlying the proposed protocol, namely the Computational Diffie-Hellman (CDH) Problem.

*Definition 1: (CDH Problem)* The Computational Diffie-Hellman (CDH) problem in  $\mathbb{G}$  is as follows: given  $(P, xP, yP) \in \mathbb{G}$  for unknown  $x, y \in \mathbb{Z}_q^*$ , compute  $xyP \in \mathbb{G}$ .

*Definition 2: (CDH Assumption)* Let  $\mathcal{A}$  be an adversary that takes an input of  $(P, xP, yP) \in \mathbb{G}$  for unknown  $x, y \in \mathbb{Z}_q^*$ , and returns a new element  $Z \in \mathbb{G}$ . We consider the following random experiment.

*Experiment*  $\mathbf{Exp}_A^{\text{CDH}}$

$$x, y \xleftarrow{R} \mathbb{Z}_q^*, Z \leftarrow \mathcal{A}(P, xP, yP)$$

$$\text{if } Z = xyP \text{ then } b \leftarrow 1 \text{ else } b \leftarrow 0$$

$$\text{return } b$$

We define the corresponding success probability of  $\mathcal{A}$  in solving the CDH problem via  $\mathbf{Succ}_A^{\text{CDH}} = \Pr[\mathbf{Exp}_A^{\text{CDH}} = 1]$ . Let  $\tau \in \mathbb{N}$  and  $\epsilon \in [0, 1]$ . We say that the CDH is  $(\tau, \epsilon)$ -secure if no polynomial algorithm  $\mathcal{A}$  running in time  $\tau$  has success  $\mathbf{Succ}_A^{\text{CDH}} \geq \epsilon$ .

### IV. OUR PROPOSED FLIP PROTOCOL

In this section, we present our efficient privacy-preserving finding like-minded vehicle protocol (FLIP), which mainly consists of two parts: system initialization and privacy-preserving finding like-minded vehicle on the road.

#### A. System Initialization

In system initialization phase, the Trusted Authority (TA) first initializes the whole system by running the following steps. Given the security parameter  $l$ , TA generates an elliptic curve group  $\mathbb{G} = \langle P \rangle$ , where the generator  $P$  has a large prime order  $q$  with  $|q| = l$ . Then, TA chooses a random

and inserting the transcript of the protocol to identify the IP of the requestor  $V_a$ . A query  $\text{SendReq}(\Pi_{V_a}^s, \text{init})$  initializes the protocol, and thus the adversary  $\mathcal{A}$  receives the transcripts sent out by  $V_a$  to  $V_b$ .

- $\text{SendRes}(\Pi_{V_b}^s, *)$ : This query models  $\mathcal{A}$  can send a transcript  $m$  to the responder-instance  $\Pi_{V_b}^s$ , and get back the answer of  $\Pi_{V_b}^s$  by following FLIP. The adversary  $\mathcal{A}$  can use this query to perform active attacks by modifying and inserting the transcript of the protocol to identify the IP of the responder  $V_b$ .
- $\text{Reveal}(\Pi_{V_a, V_b}^s)$ : This query models the known session key attack. The adversary  $\mathcal{A}$  can get access to an old session key that has been previously established. Once  $\Pi_{V_a, V_b}^s$  is valid and holds some session key, then  $\Pi_{V_a, V_b}^s$  will send the session key and the common interest  $I_\alpha^s$  to  $\mathcal{A}$  when it receives the query.
- $\text{Corrupt}(V_i)$ : This query models exposure of the private key corresponding to  $\text{pid}_i$  held by  $V_i \in \{V_a, V_b\}$  to the adversary  $\mathcal{A}$ . In reality, the scenarios that  $V_i$  may discard some outdated pseudo-ID and its corresponding key materials are modelled by this query.
- $\text{Test}(\Pi_{V_a, V_b}^s)$ : This query is used to define the advantage of the adversary  $\mathcal{A}$ . When the adversary  $\mathcal{A}$  queries on an instance  $\Pi_{V_a, V_b}^s$  based on the common interest  $I_\alpha^s$ , where  $1 \leq \alpha \leq k$ ,  $\mathcal{A}$  is given either the actual session key or a random value drawn from the session key space, according to a random bit  $\beta \in \{0, 1\}$ , i.e., actual session key is given when  $\beta = 0$  and a random value is drawn when  $\beta = 1$ . The  $\text{Test}$  query can be asked at most once by  $\mathcal{A}$ .

**FRESHNESS:** The freshness is a useful notion, which identifies the session keys about which the adversary  $\mathcal{A}$  ought not to know anything since  $\mathcal{A}$  has not revealed any oracles that have accepted the session key and has not corrupted  $V_i \in \{V_a, V_b\}$ . An oracle  $\Pi_{V_a, V_b}^s$  is said fresh if i)  $\Pi_{V_a, V_b}^s$  has accepted a session key and  $\Pi_{V_a, V_b}^s$  has not been asked for a  $\text{Reveal}$  query; ii) No  $\text{Corrupt}$  query has been asked before a query of the form  $\text{SendReq}(\Pi_{V_a}^s, *)$  or  $\text{SendRes}(\Pi_{V_b}^s, *)$ .

**DEFINITION OF SECURITY:** The security of FLIP is defined using the following game, played between  $\mathcal{A}$  and a collection of  $\Pi_{V_a, V_b}^s$  oracle for vehicles  $V_a, V_b$  and  $s \in \mathbb{N}$ , where  $V_a, V_b$  are first assigned pseudo-IDs and the corresponding key materials, respectively.

- In the game,  $\mathcal{A}$  may ask some queries and get back the answers from the corresponding oracles.
- At certain point,  $\mathcal{A}$  asks a  $\text{Test}$  query to a fresh oracle, and outputs its guess  $\alpha'$  for  $\alpha$ , where  $1 \leq \alpha \leq k$ , and  $\beta'$  for the bit  $\beta$  in the  $\text{Test}$  query.

The success of  $\mathcal{A}$  in the game is quantified in terms of  $\mathcal{A}$ 's advantage in distinguishing whether  $\mathcal{A}$  guesses the correct common interest  $I_\alpha^s$ , and receives a real session key or not, i.e., its ability guessing  $\alpha, \beta$ . We define  $\mathcal{A}$ 's advantages as

$$\mathbf{Adv}_P^\alpha(\mathcal{A}) = k \cdot \Pr[\alpha = \alpha'] - 1, \mathbf{Adv}_P^\beta(\mathcal{A}) = 2 \cdot \Pr[\beta = \beta'] - 1$$

We say that the FLIP is secure if both  $\mathbf{Adv}_P^\alpha(\mathcal{A})$  and  $\mathbf{Adv}_P^\beta(\mathcal{A})$  are negligible.

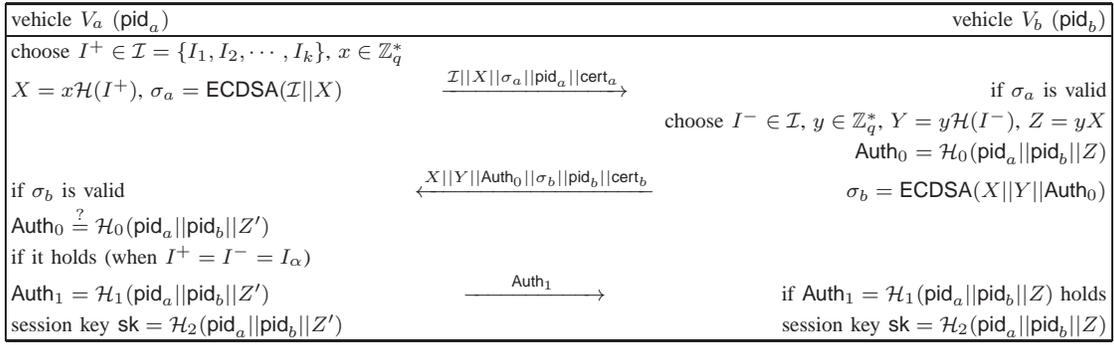


Fig. 2. Proposed Privacy-preserving Finding Like-minded Vehicle Protocol

number  $s \in \mathbb{Z}_q^*$  as the *master key* and compute the corresponding system public key  $P_{pub} = sP$ . In addition, TA also chooses four secure hash functions  $\mathcal{H}, \mathcal{H}_0, \mathcal{H}_1$ , and  $\mathcal{H}_2$ , where  $\mathcal{H} : \{0, 1\}^* \rightarrow \mathbb{G}$  and  $\mathcal{H}_i : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$ , for  $i = 0, 1, 2$ . In the end, TA publishes the public system parameters *params* as  $\{\mathbb{G}, P, q, P_{pub}, \mathcal{H}, \mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2\}$  and keeps the *master key* secretly.

When a vehicle  $V_i \in \mathcal{V}$  itself to the system, TA first checks the vehicle  $V_i$ 's validity. If  $V_i$  is valid, TA generates a family of pseudo-IDs and the corresponding key materials for  $V_i$  using Algorithm 1. In such a way,  $V_i$  can constantly change its pseudo-IDs to achieve identity privacy and location privacy on the road.

---

#### Algorithm 1 Vehicle Registration Algorithm

---

1: **procedure** VEHICLEREGISTRATION  
**Input:** a verified vehicle  $V_i \in \mathcal{V}$   
**Output:** a family of pseudo-IDs and the corresponding key materials  
2: choose a family of unlinkable pseudo-IDs  $\text{PID} = \{\text{pid}_1, \text{pid}_2, \dots\}$   
3: **for** each pseudo-ID  $\text{pid}_i \in \text{PID}$  **do**  
4:     randomly choose a private key  $x_i \in \mathbb{Z}_q^*$   
5:     compute the corresponding public key  $Y_i = x_iP$   
6:     assert  $(\text{pid}_i, Y_i)$  with certificate  $\text{cert}_i$  signed by TA with  $s$   
7: **end for**  
8: **return** all tuples  $(\text{pid}_i, x_i, Y_i, \text{cert}_i)$  to  $V_i$   
9: **end procedure**

---

#### B. Privacy-preserving Finding Like-minded Vehicle

When a vehicle  $V_a \in \mathcal{V}$  is on the road and wants to find a like-minded vehicle  $V_b \in \mathcal{V}$  on the common interest  $I_\alpha$  nearby, as shown in Fig. 2, they will run the following steps to establish a shared session key  $\text{sk}$  regarding the common interest  $I_\alpha$ .

*Step 1.*  $V_a$  first sets an interest set  $\mathcal{I}$ , which consists of  $k$  kinds of interests  $\{I_1, I_2, \dots, I_k\}$ , where  $V_a$ 's actual interest  $I^+$  is involved. Then,  $V_a$  chooses a random number  $x \in \mathbb{Z}_q^*$ , computes  $X = x\mathcal{H}(I^+)$ , and uses the ECDSA algorithm to make a signature  $\sigma_a = \text{ECDSA}(\mathcal{I}||X)$  on  $\mathcal{I}||X$  with regard to the pseudo-ID  $\text{pid}_a$  and the certificate  $\text{cert}_a$ . In the end,  $V_a$  broadcasts the request  $\mathcal{I}||X||\sigma_a||\text{pid}_a||\text{cert}_a$  to the nearby vehicles.

*Step 2.* Upon receiving the request  $\mathcal{I}||X||\sigma_a||\text{pid}_a||\text{cert}_a$ , a nearby vehicle  $V_b$  first checks the validity of  $\sigma_a$  with

$\text{pid}_a||\text{cert}_a$ . If it is invalid,  $V_b$  neglects the request. Otherwise,  $V_b$  chooses his interest  $I^- \in \mathcal{I}$  and a random number  $y \in \mathbb{Z}_q^*$ , computes  $Y = y\mathcal{H}(I^-)$ ,  $Z = yX$ , and  $\text{Auth}_0 = \mathcal{H}_0(\text{pid}_a||\text{pid}_b||Z)$ . In addition,  $V_b$  makes a signature  $\sigma_b = \text{ECDSA}(X||Y||\text{Auth}_0)$  on  $X||Y||\text{Auth}_0$  with regard to the pseudo-ID  $\text{pid}_b$  and the certificate  $\text{cert}_b$ , and returns the response  $X||Y||\text{Auth}_0||\sigma_b||\text{pid}_b||\text{cert}_b$  to  $V_a$ . Note that, in the protocol,  $V_b$  is only allowed to make at most one response for the same request.

*Step 3.* After receiving the responder  $V_b$ 's response  $X||Y||\text{Auth}_0||\sigma_b||\text{pid}_b||\text{cert}_b$ , the requestor  $V_a$  first checks the validity of  $\sigma_b$  with  $\text{pid}_b||\text{cert}_b$ . If it is invalid,  $V_a$  neglects the response. Otherwise,  $V_a$  computes  $Z' = xY$ , and checks whether  $\text{Auth}_0 \stackrel{?}{=} \mathcal{H}_0(\text{pid}_a||\text{pid}_b||Z')$ . If it holds,  $V_a$  computes and sends  $\text{Auth}_1 = \mathcal{H}_1(\text{pid}_a||\text{pid}_b||Z')$  to  $V_b$ , and calculate the session key  $\text{sk} = \mathcal{H}_2(\text{pid}_a||\text{pid}_b||Z')$ .

*Step 4.* When  $V_b$  receives  $\text{Auth}_1 = \mathcal{H}_1(\text{pid}_a||\text{pid}_b||Z')$ , he checks whether  $\text{Auth}_1 \stackrel{?}{=} \mathcal{H}_1(\text{pid}_a||\text{pid}_b||Z)$ . If it holds,  $V_b$  calculates the session key  $\text{sk} = \mathcal{H}_2(\text{pid}_a||\text{pid}_b||Z)$ . If  $I^+ = I^- = I_\alpha$  for some  $1 \leq \alpha \leq k$ ,  $V_a$  and  $V_b$  have the shared session key  $\text{sk}$ , i.e., the vehicle  $V_a$  successfully finds an like-minded vehicle  $V_b$  on the road.

**Correctness.** If the interests  $I^+$  and  $I^-$  are same, i.e.,  $I^+ = I^- = I_\alpha \in \mathcal{I}$ , then  $\mathcal{H}(I^-) = \mathcal{H}(I^+)$ ,

$$Z' = xY = xy\mathcal{H}(I^-) = xy\mathcal{H}(I_\alpha) = yx\mathcal{H}(I^+) = yX = Z$$

and both the authenticators  $\text{Auth}_0 = \mathcal{H}_0(\text{pid}_a||\text{pid}_b||Z)$ ,  $\text{Auth}_1 = \mathcal{H}_1(\text{pid}_a||\text{pid}_b||Z')$  and the session key  $\text{sk}$  are valid. However, if  $I^+ \neq I^-$ , then  $\mathcal{H}(I^-) \neq \mathcal{H}(I^+)$  and

$$Z' = xY = xy\mathcal{H}(I^-) \neq xy\mathcal{H}(I^+) = yX = Z$$

which indicates that  $\text{Auth}_0$ , and  $\text{Auth}_1$  are not valid, and the shared session key  $\text{sk}$  can't be established. Therefore, the correctness of the proposed FLIP protocol follows. Note that, the responder  $V_b$  can only response once for the same request. Otherwise, by successive responses to the same request, the requestor's IP can be guessed by non-like-minded vehicles. In reality, the requestor  $V_a$  usually can detect whether a nearby vehicle  $V_b$  has responded more than once based on  $V_b$ 's relative location and other correlative information on the road. Thus, the successive-response attack can be prevented.

**Efficiency.** The proposed FLIP protocol is very efficient in terms of computational costs. Let  $T_{\text{mul}}$  denote the time to perform one point multiplication in  $\mathbb{G}$ , and  $T_{\text{sig}}$  and  $T_{\text{ver}}$  the times of ECDSA signing and verification, respectively. Since these operations dominate the speed of FLIP, we neglect others such as hash operations in measure of FLI. Then, only  $2T_{\text{mul}} + T_{\text{sig}} + T_{\text{ver}}$  are required at both the requestor  $V_a$  and the responder  $V_b$  in the proposed FLIP protocol. Based on the results in [10], we know that  $T_{\text{mul}} \approx 2.92$  ms,  $T_{\text{sig}} \approx 2.92$  ms,  $T_{\text{ver}} \approx 3.87$  ms for a 224-bit ECDSA with the MIRACL cryptographic lib [11] running on a 3GHz Pentium IV system. Then, the computational costs are only  $2T_{\text{mul}} + T_{\text{sig}} + T_{\text{ver}} \approx 12.63$  ms. In addition, since the 802.11p physical layer offers different bitrates, ranging from 3 to 27 Mbps, from which we can choose [6]. Therefore, the transcripts in the proposed FLIP protocol can be fast exchanged between the requestor  $V_a$  and the responder  $V_b$ .

## V. SECURITY ANALYSIS

In this section, we will demonstrate the IP can be protected against non-like-minded vehicles without collusion in the proposed FLIP protocol. Note that, since the ECDSA signature is unforgeable, all transcripts in FLIP are detectable if they are altered by the adversary. Therefore, we should only consider an adversary can't break the proposed FLIP protocol without altering the transcripts.

*Theorem 1:* Let  $\mathcal{A}$  be an adversary against the proposed FLIP protocol in the random oracle model [5], where the hash functions  $\mathcal{H}$ ,  $\mathcal{H}_0$ ,  $\mathcal{H}_1$ , and  $\mathcal{H}_2$  behave as random oracles. Assume that  $\mathcal{A}$  has the advantage  $\text{Adv}_{\mathcal{P}}^{\alpha}(\mathcal{A}) = \epsilon$  to guess the correct interest  $I_{\alpha}$ , and the advantage  $\text{Adv}_{\mathcal{P}}^{\beta}(\mathcal{A}) = \epsilon$  break the proposed FLIP protocol without altering the transcripts, within the running time  $\tau$ , after several oracles defined in the adversarial model. Then, there exist  $\epsilon' \in [0, 1]$  and  $\tau' \in \mathbb{N}$  as follows

$$\epsilon' = \text{Succ}_{\mathcal{A}}^{\text{CDH}} \geq \frac{\epsilon}{q_s q_{H_2}}, \quad \tau' \leq \tau + \Theta(\cdot)$$

such that the CDH problem can be solved with probability  $\epsilon'$  and within time  $\tau'$ , where  $\Theta(\cdot)$  is the time complexity for the simulation,  $q_{H_2}$  is the total number of  $\mathcal{H}_2$  oracle queries, and  $q_s$  is the total number of session instances  $\Pi_{V_a, V_b}^1, \Pi_{V_a, V_b}^2, \dots, \Pi_{V_a, V_b}^{q_s}$ .

*Proof:* Since the adversary  $\mathcal{A}$  can, with non-negligible advantage  $\text{Adv}_{\mathcal{P}}^{\beta}(\mathcal{A})$ , break the proposed FLIP protocol, we can use  $\mathcal{A}$ 's attack capabilities to construct another algorithm  $\mathcal{B}$  to solve the CDH problem. In specific,  $\mathcal{B}$  is given a random instance of the CDH problem  $(P, xP, yP)$ , where  $x, y \in \mathbb{Z}_q^*$ . Then,  $\mathcal{B}$  runs  $\mathcal{A}$  as a subroutine and simulates the attack environment required by  $\mathcal{A}$ .

At first, for each vehicle  $V_i \in \{V_a, V_b\}$  involved in the FLIP protocol,  $\mathcal{B}$  sets  $V_i$ 's pseudo-ID  $\text{pid}_i$ , generates valid key materials by choosing a random number  $\tilde{x}_i \in \mathbb{Z}_q^*$  as the private key, computes the corresponding public key  $\tilde{Y}_i = \tilde{x}_i P$ ,

as well as the certificate  $\text{cert}_i$  with the resort of TA. Then,  $\mathcal{B}$  interacts with  $\mathcal{A}$  and simulates all the instances with queries of oracles **SendReq**, **SendRes**, **Execute**, **Reveal**, **Corrupt**, and **Test**. In order to make use of  $\mathcal{A}$ 's attack capability,  $\mathcal{B}$  first guesses  $\gamma$  such that  $\mathcal{A}$  asks the **Test** query in the  $\gamma$ -th session. Because there are total  $q_s$  session instances, the probability for successful guessing  $\gamma$  is  $1/q_s$ . Besides the above oracles,  $\mathcal{B}$  should also simulates the random oracles  $\mathcal{H}$ ,  $\mathcal{H}_0$ ,  $\mathcal{H}_1$ , and  $\mathcal{H}_2$  by maintaining the lists  $\mathcal{H}$ -list,  $\mathcal{H}_0$ -list,  $\mathcal{H}_1$ -list and  $\mathcal{H}_2$ -list to deal with the identical queries as follows.

- ▷ **sim- $\mathcal{H}$**   
On input of an interest  $I_i \in \mathcal{I} = \{I_1, I_2, \dots, I_k\}$   
choose a fresh random number  $r_i \xleftarrow{R} \mathbb{Z}_q^*$   
compute  $H_i = r_i P$ , set  $\mathcal{H}(I_i) = H_i$   
add  $(I_i, r_i, H_i)$  to  $\mathcal{H}$ -list  
return  $H_i$
- ▷ **sim- $\mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2$**   
On input of  $\text{pid}_a || \text{pid}_b || Z_i$  on  $\mathcal{H}_j$ ,  $j \in \{0, 1, 2\}$   
choose a fresh random number  $u_{ij} \xleftarrow{R} \mathbb{Z}_q^*$   
set  $\mathcal{H}_j(\text{pid}_a || \text{pid}_b || Z_i) = u_{ij}$   
add  $(\text{pid}_a || \text{pid}_b || Z_i, u_{ij})$  to  $\mathcal{H}_j$ -list  
return  $u_{ij}$
- ▷ **sim-SendReq**( $\Pi_{V_a}^s$ , init)  
**if**  $s = \gamma$  **then**  
randomly choose an interest  $I_{\alpha} \in \mathcal{I}$   
obtain the tuple  $(I_{\alpha}, r_{\alpha}, H_{\alpha})$  in  $\mathcal{H}$ -list  
compute  $X = r_{\alpha} x P$ ,  $\sigma_{\alpha} = \text{ECDSA}(\mathcal{I} || X)$   
return  $\mathcal{I} || X || \sigma_{\alpha} || \text{pid}_a || \text{cert}_a$   
**else if**  $s \neq \gamma$  **then**  
randomly choose an interest  $I_i \in \mathcal{I}$ , and  $x_i \in \mathbb{Z}_q^*$   
obtain the tuple  $(I_i, r_i, H_i)$  in  $\mathcal{H}$ -list  
compute  $X = x_i H_i$ ,  $\sigma_{\alpha} = \text{ECDSA}(\mathcal{I} || X)$   
return  $\mathcal{I} || X || \sigma_{\alpha} || \text{pid}_a || \text{cert}_a$
- ▷ **sim-SendRes**( $\Pi_{V_b}^s$ ,  $\mathcal{I} || X || \sigma_{\alpha} || \text{pid}_a || \text{cert}_a$ )  
**if**  $s = \gamma$  **then**  
choose the same interest  $I_{\alpha}$  identified with  $s$   
obtain the tuple  $(I_{\alpha}, r_{\alpha}, H_{\alpha})$  in  $\mathcal{H}$ -list  
compute  $Y = r_{\alpha} y P$ , set  $Z = \perp$   
choose a fresh random number  $u_{\alpha 0} \xleftarrow{R} \mathbb{Z}_q^*$   
set  $\text{Auth}_0 = \mathcal{H}_0(\text{pid}_a || \text{pid}_b || Z) = u_{\alpha 0}$   
add  $(\text{pid}_a || \text{pid}_b || Z, u_{\alpha 0})$  to  $\mathcal{H}_0$ -list  
compute  $\sigma_b = \text{ECDSA}(X || Y || \text{Auth}_0)$   
return  $X || Y || \text{Auth}_0 || \sigma_b || \text{pid}_b || \text{cert}_b$   
**else if**  $s \neq \gamma$  **then**  
choose the same interest  $I_i$  identified with  $s$   
obtain the tuple  $(I_i, r_i, H_i)$  in  $\mathcal{H}$ -list  
chooses  $y_i \in \mathbb{Z}_q^*$ , set  $Y = y_i H_i$ ,  $Z = x_i y_i H_i$   
choose a fresh random number  $u_{i 0} \xleftarrow{R} \mathbb{Z}_q^*$   
set  $\text{Auth}_0 = \mathcal{H}_0(\text{pid}_a || \text{pid}_b || Z) = u_{i 0}$   
add  $(\text{pid}_a || \text{pid}_b || Z, u_{i 0})$  to  $\mathcal{H}_0$ -list  
compute  $\sigma_b = \text{ECDSA}(X || Y || \text{Auth}_0)$   
return  $X || Y || \text{Auth}_0 || \sigma_b || \text{pid}_b || \text{cert}_b$
- ▷ **sim-SendReq**( $\Pi_{V_a}^s$ ,  $X || Y || \text{Auth}_0 || \sigma_b || \text{pid}_b || \text{cert}_b$ )

**if**  $s = \gamma$  **then**  
 obtain the tuple  $(\text{pid}_a || \text{pid}_b || Z, u_{\alpha 0})$  in  $\mathcal{H}_0$ -list  
 on condition that  $\text{Auth}_0 = u_{\alpha 0}$   
 choose a fresh random number  $u_{\alpha 1} \xleftarrow{R} \mathbb{Z}_q^*$   
 set  $\text{Auth}_1 = \mathcal{H}_1(\text{pid}_a || \text{pid}_b || Z) = u_{\alpha 1}$   
 add  $(\text{pid}_a || \text{pid}_b || Z, u_{\alpha 1})$  to  $\mathcal{H}_1$ -list  
 return  $\text{Auth}_1$   
**else if**  $s \neq \gamma$  **then**  
 obtain the tuple  $(\text{pid}_a || \text{pid}_b || Z, u_{i0})$  in  $\mathcal{H}_0$ -list  
 on condition that  $\text{Auth}_0 = u_{i0}$   
 choose fresh random numbers  $u_{\alpha 1}, u_{\alpha 2} \xleftarrow{R} \mathbb{Z}_q^*$   
 set  $\text{Auth}_1 = \mathcal{H}_1(\text{pid}_a || \text{pid}_b || Z) = u_{i1}$   
 add  $(\text{pid}_a || \text{pid}_b || Z, u_{i1})$  to  $\mathcal{H}_1$ -list  
 set session key  $\text{sk} = \mathcal{H}_2(\text{pid}_a || \text{pid}_b || Z) = u_{i2}$   
 add  $(\text{pid}_a || \text{pid}_b || Z, u_{i2})$  to  $\mathcal{H}_2$ -list  
 return  $\text{Auth}_1$

- ▷ **sim-Execute** $(\Pi_{V_a, V_b}^s)$   
 successively simulate  $\text{SendReq}(\Pi_{V_a}^s, \text{init})$ ,  
 $\text{SendRes}(\Pi_{V_b}^s, \mathcal{I} || X || \sigma_a || \text{pid}_a || \text{cert}_a)$ , and  
 $\text{SendReq}(\Pi_{V_a}^s, X || Y || \text{Auth}_0 || \sigma_b || \text{pid}_b || \text{cert}_b)$
- ▷ **sim-Reveal** $(\Pi_{V_a, V_b}^s)$  with  $s \neq \gamma$   
 obtain the tuple  $(\text{pid}_a || \text{pid}_b || Z, u_{i2})$  in  $\mathcal{H}_2$ -list  
 and the interest  $I_i$  identified with  $s$   
 return  $\text{sk} = u_{i2}$  and  $I_i$
- ▷ **sim-Corrupt** $(V_i)$  with  $V_i \in \{V_a, V_b\}$   
 return the private key  $\tilde{x}_i$  of  $V_i$  with respect to  $\text{pid}_i$
- ▷ **sim-Test** $(\Pi_{V_a, V_b}^s)$  with  $s = \gamma$   
 randomly flip a coin  $\beta \in \{0, 1\}$   
 choose a random number  $u_{\alpha 2} \xleftarrow{R} \mathbb{Z}_q^*$   
 return  $u_{\alpha 2}$

After receiving  $u_{\alpha 2}$  from  $\text{Test}(\Pi_{V_a, V_b}^s)$ , the adversary  $\mathcal{A}$  guesses  $\alpha' \in \{1, 2, \dots, k\}$  for  $\alpha$  and  $\beta' \in \{0, 1\}$  for  $\beta$ , and returns  $(\alpha', \beta')$  to  $\mathcal{B}$ . Then, we analyze  $\mathcal{A}$ 's successful guess probability on  $\alpha'$  and  $\beta'$ .

First, we consider the transcripts  $X = x\mathcal{H}(I_\alpha)$  and  $Y = y\mathcal{H}(I_\alpha)$  with unknown  $x, y \in \mathbb{Z}_q^*$ . Because  $\mathbb{G}$  is a cyclic group, we can see there always exist other  $x_i, y_i \in \mathbb{Z}_q^*$  such that  $X = x_i\mathcal{H}(I_i)$  and  $Y = y_i\mathcal{H}(I_i)$ . Therefore, the transcripts  $(X, Y)$  can be linked to each interest  $I_i \in \{I_1, I_2, \dots, I_k\}$  equally. Therefore, we can know  $\Pr[\alpha' = \alpha] = \frac{1}{k}$  and

$$\text{Adv}_{\mathcal{P}}^\alpha(\mathcal{A}) = k \cdot \Pr[\alpha = \alpha'] - 1 = 0, \quad \text{i.e., } \varepsilon = 0$$

Second, we analyze the advantage probability  $\text{Adv}_{\mathcal{P}}^\beta(\mathcal{A})$  on guessing the correct  $\beta$ , where

$$\Pr[\beta = \beta'] = \frac{1}{2} + \frac{\text{Adv}_{\mathcal{P}}^\beta(\mathcal{A})}{2}$$

Let  $\text{E}$  denote the event that  $\text{pid}_a || \text{pid}_b || Z$  has been queried by  $\mathcal{A}$  on  $\mathcal{H}_2$  oracle, where  $Z = r_\alpha xyP$ . If the event  $\text{E}$  does not occur,  $\mathcal{B}$  has no idea on the session key  $\text{sk}$ , then

$$\Pr[\beta' = \beta | \neg \text{E}] = \frac{1}{2}$$

and

$$\begin{aligned} \Pr[\beta = \beta'] &= \Pr[\beta = \beta' | \text{E}] \cdot \Pr[\text{E}] + \Pr[\beta = \beta' | \neg \text{E}] \cdot \Pr[\neg \text{E}] \\ &= \Pr[\beta = \beta' | \text{E}] \cdot \Pr[\text{E}] + \frac{1}{2} \cdot (1 - \Pr[\text{E}]) \\ &\leq \Pr[\text{E}] + \frac{1}{2} \cdot (1 - \Pr[\text{E}]) \\ &= \frac{1}{2} + \frac{\Pr[\text{E}]}{2} \end{aligned}$$

Therefore, based on the above relations, we have

$$\Pr[\text{E}] \geq \text{Adv}_{\mathcal{P}}^\beta(\mathcal{A})$$

Because  $\mathcal{H}_2$ -list contains  $q_{H_2}$  entries, we can pick up the correct  $\text{pid}_a || \text{pid}_b || Z$ , where  $Z = r_\alpha xyP$ , with the success probability  $1/q_{H_2}$ . Then, by computing  $Z/r_\alpha = xyP$ , where  $r_\alpha$  is included in the entry  $(I_\alpha, r_\alpha, H_\alpha)$  in  $\mathcal{H}$ -list, we can get the CDH challenge  $xyP$ . Combining the probability  $1/q_s$  for guessing the correct  $\gamma$ , we have

$$\epsilon' = \text{Succ}_{\mathcal{A}}^{\text{CDH}} \geq \frac{\text{Adv}_{\mathcal{P}}^\beta(\mathcal{A})}{q_s q_{H_2}} = \frac{\epsilon}{q_s q_{H_2}}$$

In addition, we can obtain the claimed bound for  $\tau' \leq \tau + \Theta(\cdot)$  in the above simulation. In summary, the IP can be protected in the proposed FLIP protocol. Thus, the proof is completed. ■

## VI. PERFORMANCE EVALUATION

In the proposed FLIP protocol, to prevent successive-guessing attack from non-like-minded vehicle, the responder  $V_b$  is only allowed to respond once for the same request. Therefore, the larger the Interest Set  $\mathcal{I} = \{I_1, I_2, \dots, I_k\}$  that the requestor  $V_a$  chooses, the harder the actual interest  $I_\alpha \in \mathcal{I}$  can be guessed by non-like-minded vehicle, and thus the IP can be protected. However, when the set  $\mathcal{I}$  becomes large, multiple interests  $\mathcal{I}' = \{I_\alpha, I_\beta, I_\gamma, \dots\}$  of the like-minded vehicle  $V_b$  could belong to  $\mathcal{I}$ . Then, it is hard for  $V_b$  to choose the correct  $I_\alpha \in \mathcal{I}'$ , which thus causes the long delay for finding the like-minded vehicle. Therefore, in this section, we use a custom simulator built in Java to study how the interest set  $\mathcal{I}$  affects the delay for finding the like-minded vehicle on the road. In specific, the performance metric used in the evaluation is the average delay for finding the like-minded vehicle, denoted by  $\text{FD}$ , which is defined as the average time between when the requestor  $V_a$  sends a request and when  $V_b$  successfully finds a like-minded vehicle  $V_b$  on the road.

### A. Simulation Settings

We consider a large number of vehicles  $\mathcal{V} = \{V_1, V_2, \dots\}$  are moving on a multi-lane same-direction road with velocity varying from 40 km/h to 80 km/h. Consider other vehicles passing-by a vehicle  $V_a \in \mathcal{V}$  follows a Poisson process, and the inter-passing-by time  $t_a$  has an exponential distribution with the mean  $1/\lambda$ . In the simulation, the vehicle  $V_a$  will broadcast the request with Interest set  $\mathcal{I}$  of different size  $|\mathcal{I}|$  varying from 1 to 10, to find the like-minded vehicle.

The detailed parameter settings in the simulations are summarized in Table I. We perform the simulations for the specified interest set size  $|\mathcal{I}|$  varying from 1 to 10 with increment of 1. For each case, we run the simulation 10,000 times, and the average FD is reported.

TABLE I  
SIMULATION SETTINGS

Parameter	Setting
Simulation area, duration	a multi-lane same-direction road, 1 hour
Vehicle velocity, transmission	40 km/h - 80 km/h, 300 m
Mean passing-by rate $\lambda$	[20/h, 40/h, 60/h, 80/h, 100/h, 120/h]
Interest set size $ \mathcal{I} $	[1, 2, $\dots$ , 10]

### B. Simulation Results

Fig. 3 shows the average FD under the different  $|\mathcal{I}|$  and  $\lambda$  within 1 hour. From the figure, we can see, the larger the  $|\mathcal{I}|$ , the longer the average FD; but at the same time, the average FD can be reduced with the increase of  $\lambda$ . Therefore, by setting a proper size of  $|\mathcal{I}|$  on considering of  $\lambda$ , a vehicle  $V_a$  can find a like-minded vehicle within an expected time on the road while keeping his IP from non-like-minded vehicles.

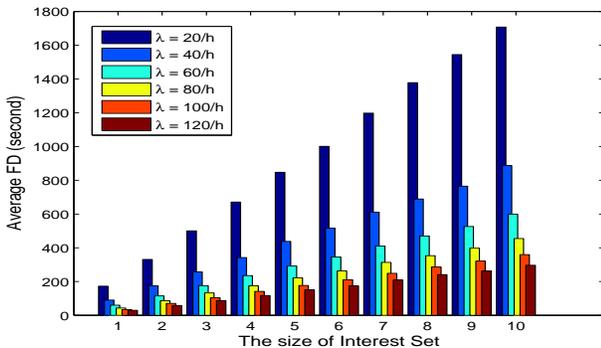


Fig. 3. The average FD in different interest set size  $|\mathcal{I}|$  within 1 hour

### VII. RELATED WORK

Recently, several research works have been reported [12], [13], which are closely related to the proposed FLIP protocol, but focus on other scenarios different from VANETs. Atallah and Du [12] present secure multi-party computation problems and several privacy-preserving applications including privacy-preserving database query, intrusion detection, data mining, and geometric computation. Although they take some initial attempts to tackle these problems, their solutions are less than satisfactory because a semi-trusted third party is required to be involved in the privacy-preserving computations. Based on the homomorphic encryption, Zhong et al. [13] propose three protocols called Louis, Lester and Pierre to resolve the nearby friend problem, where a mobile user can determine whether or not one of his friend is in a nearby location in a privacy-preserving way. However, Louis still requires an online semi-trusted third party, and both Louis and Lester are found insecure [14], [15]. Recently, Chatterjee et al. [14] propose a new efficient protocol for the nearby friend problem

without resorting to a semi-honest third party. However, due to lack of authentication, their protocol could suffer from replay attack and man-in-the-middle attack, and thus can't be directly applied in VANET scenarios. Different from the above works, our proposed FLIP protocol can provide mutual authentication and establish a shared session key between two like-minded vehicles. But more importantly, it is a provably secure protocol suitable for VANET scenarios.

### VIII. CONCLUSIONS

Secure finding like-minded vehicles protocol (FLIP) can protect vehicle's IP and is of vital importance to the success of vehicle chatting application on the road, yet it hasn't been paid enough attention in VANET. In this paper, based on the elliptic-curve technique, we have proposed an efficient IP-preserving FLIP protocol. With the provable security technique, the proposed FLIP protocol has been demonstrated to be secure in the VANET scenarios. In addition, extensive simulations have also been conducted to its practical considerations, i.e., how to balance the level of IP-preserving and the delay of finding like-minded vehicles on the road. Because the proposed FLIP protocol keeps each other's IP-preserving if two vehicles don't have the common interest, it can be widely accepted by the public.

### REFERENCES

- [1] X. Lin, X. Sun, P.-H. Ho, and X. Shen, "GSIS: A secure and privacy-preserving protocol for vehicular communications," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3442–3456, 2007.
- [2] S. Smaldone, L. Han, P. Shankar, and L. Iftode, "Roadspeak: enabling voice chat on roadways using vehicular social networks," in *SocialNets '08*, 2008, pp. 43–48.
- [3] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen, "ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications," in *INFOCOM 2008*, Phoenix, April 2008, pp. 1229–1237.
- [4] R. Lu, X. Lin, and X. Shen, "Spring: A social-based privacy-preserving packet forwarding protocol for vehicular delay tolerant networks," in *INFOCOM 2010*, San Diego, California, USA, March 2010, pp. 1–9.
- [5] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *CCS '93*, 1993, pp. 62–73.
- [6] P. Shankar, T. Nadeem, J. Rosca, and L. Iftode, "Cars: Context-aware rate selection for vehicular networks," in *ICNP 2008*, 2008, pp. 1–12.
- [7] Z. Chai, Z. Cao, and R. Lu, "Efficient password-based authentication and key exchange scheme preserving user privacy," in *WASA*, 2006, pp. 467–477.
- [8] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *EUROCRYPT*, 2000, pp. 139–155.
- [9] S. C. Pohlig and M. E. Hellman, "An improved algorithm for computing logarithms over  $GF(p)$  and its cryptographic significance," *IEEE Transactions on Information Theory*, vol. IT-24, pp. 106–110, 1978.
- [10] X. Sun, "Anonymous, secure and efficient vehicular communications," Master Thesis, University of Waterloo, 2007.
- [11] S. Software, "Miracl library," available at <http://www.shamus.ie/index.php?page=Elliptic-Curve-point-multiplication>.
- [12] M. Atallah and W. Du, "Secure multi-party computation problems and their applications: a review and open problems," in *NSPW*, 2001, pp. 13–22.
- [13] G. Zhong, I. Goldberg, and U. Hengartner, "Louis, lester and pierre: Three protocols for location privacy," in *PET 2007*, ser. LNCS 4776, 2007, pp. 62–76.
- [14] S. Chatterjee, K. Karabina, and A. Menezes, "A new protocol for the nearby friend problem," in *Cryptography and Coding*, ser. LNCS 5921, 2009, pp. 236–251.
- [15] A. Gupta, M. Saini, and A. Mathuria, "Security analysis of the louis protocol for location privacy," in *COMSNETS'09*, 2009, pp. 200–207.