# PTVC: Achieving Privacy-Preserving Trust-Based Verifiable Vehicular Cloud Computing

Cheng Huang*, Rongxing Lu†, Hui Zhu‡, Hao Hu§, and Xiaodong Lin¶
*Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada
†Faculty of Computer Science, University of New Brunswick, Fredericton, NB, Canada
‡School of Cyber Engineering, Xidian University, Xi'an, China 710126
§School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore
¶ Faculty of Business and Information Technology, University of Ontario Institute of Technology, Oshawa, ON, Canada
Email: c225huan@uwaterloo.ca, rlu1@unb.ca, zhuhui@xidian.edu.cn, hhu002@e.ntu.edu.sg, xiaodong.lin@uoit.ca

***Abstract*—With the development of intelligent transport systems (ITS) and vehicular ad hoc network (VANET), vehicular cloud computing (VCC) has been proposed to bring essential and potential benefits, such as improving traffic safety and offering computational services to road users. To make such computational services reliable and secure, the computation results from the vehicular cloud (VC) should be verifiable and the trustworthy vehicles need to be selected to form the VC with disclosure-minimizing privacy. To address these challenges, a privacy-preserving trust-based verifiable vehicular cloud computing scheme has been proposed in this paper, named PTVC. Specifically, the proposed PTVC scheme integrates the unique features of VCC and the requirements of privacy into traditional reputation system based on beta distribution, which can help differentiate the trust levels of the vehicles and preserve location privacy in the meantime. Moreover, by using the verifiable techniques, the cloud users can verify the correctness of outsourced computation while guaranteeing the privacy of their outsourced data. Detailed security analysis shows that the proposed PTVC scheme is secure and robust against several sophisticated attacks. In addition, performance evaluations via extensive simulations are also conducted, demonstrating its effectiveness.**

***Index Terms*—Vehicular Cloud, Privacy-preserving, Verifiable Computing, Trust**

## I. Introduction

Vehicular cloud computing (VCC) is a new hybrid technology based on vehicular ad hoc network (VANET), which has a remarkable impact on traffic management and other applications. VCC supports a flexible architecture with the roadside units (RSUs) deployed along the roads and on board units (OBUs) equipped in mobile vehicles, which enables not only vehicle-to-vehicle (V2V) but also vehicle-to-infrastructure (V2I) communications. However, in this paper, we consider a detailed scenario that a vehicle is traveling in a rural area where no RSUs are deployed nearby (i.e., due to the high cost of RSUs, they may only be deployed in the downtown areas), this vehicle wants to achieve an emerging computation task by asking the nearby vehicles to form a temporary vehicular cloud (VC) to provide computational resources. Yet, vehicles driven by human beings cannot always be trustworthy. For example, they may make mistakes during the computation task or just return the arbitrary results, leading to the failure of this task. One possible solution is to incorporate trust management [1] and verification technique [2], which allows each vehicle to distinguish these critical vehicles and verify the correctness of computation results.

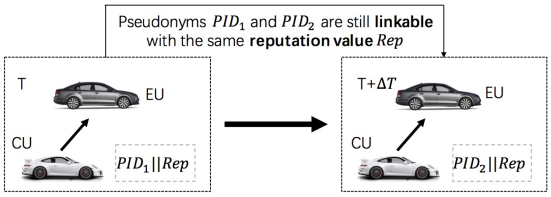



Fig. 1. Pseudonyms link due to the same reputation value

In addition, location privacy and data privacy are also vital concerns in the VCC. The general idea is that vehicles can periodically change their pseudonyms which make the location information unlinkable in a period. Nevertheless, pseudonyms are not the panacea in some particular scenario [3]. As an example shown in Fig. 1, a vehicle (the gray vehicle) asks a nearby vehicle in the VC to perform the computation task at time $T$ and $T + \Delta T$. When two candidates (the white vehicles) $PID_1$ and $PID_2$ respond to the requesting vehicle by sending their reputation values $Rep$ at time $T$ and $T+\Delta T$ respectively, the requesting vehicle can link $PID_1$ and $PID_2$ by connecting $Rep$ though its pseudonym is different. The constant reputation value breaches this vehicle's location privacy. Therefore, it is necessary to build a trust system where vehicles take its advantages without sacrificing the privacy. For the data privacy, it is evident that the cloud user's valuable data should be privacy-preserving. Hence, a privacy-preserving trust-based verifiable vehicular cloud computing scheme has been proposed in this paper, called PTVC, to address the issues as mentioned above. Specifically, the main contributions of this paper are three-fold.

• Firstly, to select the trustworthy vehicles to form the VC, we propose a privacy-preserving vehicle selection protocol based on the beta distribution and an efficient commitment scheme in [4]. With this protocol, the cloud user can effectively locate some vehicles which meet their demands while minimizing the privacy disclosure of location.

• Secondly, to verify the computation results from the VC, we propose a privacy-preserving verifiable computing protocol based on the verification techniques in [2], [5]. With this protocol, the cloud user can efficiently verify the correctness of outsourced computation while minimizing the privacy disclosure of outsourced data.

• Finally, through combining the above two protocols, we present a privacy-preserving trust-based verifiable vehicular cloud computing scheme, called PTVC. Also, we analyze the security and performance of PTVC.

The remainder of this paper is organized as follows. In Section II, we formalize the system model, the trust model, the threat model and identify the design goal. Then, we present the detailed design of our proposed PTVC scheme in Section III, followed by the security analysis and performance evaluation in Section IV and Section V, respectively. Finally, Section VI reviews some related work and Section VII draws the conclusion.

## II. Models and Design Goal

In this section, we formalize our system model, trust model, threat model and identify our design goal.

### A. System Model

Our system model consists of three entities, namely the trusted authority (TA), the immobile roadside units (RSUs) and a number of registered mobile vehicles equipped with OBUs (onboard units), as shown in Fig. 2.

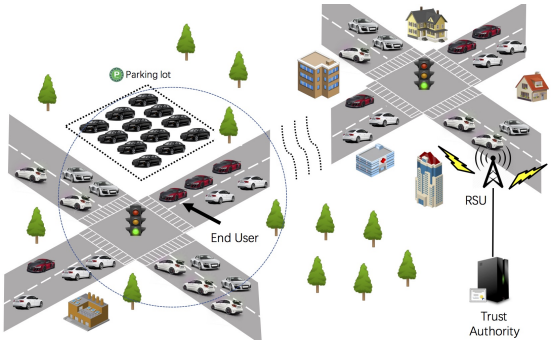



Fig. 2. System model under consideration

• *Trust Authority* (TA): TA takes charge of the registration of all RSUs and the vehicles and distributes key materials to them, which is also responsible for storing and updating reputation values of all vehicles.

• *Roadside Units* (RSUs): RSUs are subordinated by the TA and communicate with the TA through wired lines and secured channels. Equipped with wireless devices, RSUs can exchange data with the passing-by vehicles.

• *Vehicles*: Vehicles are regarded as a set of mobile/stationary nodes equipped with OBUs. Every vehicle has its reputation and can be updated through V2I communication. In this case, the vehicles can be further divided into two categories: the computing units (CUs) and the end users (EUs).

- *Computing Units* (CUs): CUs are a group of vehicles which have free computational resources. They can be organized to form a temporary vehicular cloud (VC) through the VANET network [6], [7], which can provide the services for other vehicles (EUs).
- *End Users* (EUs): EUs are the vehicles which have a requirement of heavy computing. Through the VANET network, EUs can broadcast the computation requests to the nearby vehicles (CUs), and then assign the computational tasks and outsource their data to the VC. The high-trust vehicles are chosen based on the reputation values of these vehicles. After their tasks are finished, the feedbacks of all participating vehicles (CU) will be sent to the TA.

### B. Trust Model

In this section, we make some assumptions and define the trust levels of all entities in our system model as follows.

• *Trust Authority* (TA): TA is assumed to be fully trusted with strong physical protection in our system. That is, TA is difficult to be compromised by any adversary/attacker.

• *Roadside Units* (RSUs): All RSUs are also assumed to be trusted, and they will never disclose any internal information without permissions. Although it is possible that some attackers can compromise some existing RSUs or deploy some bogus RSUs at the roadside, the TA can inspect all RSUs at a high level: once the RSUs are compromised, they will be recovered or revoked in the next time slot by the TA.

• *Vehicles (CUs)*: CUs are assumed to be honest but curious about the private outsourced data of the vehicles (EUs). Namely, they may try to disclose the EUs' private data while following the proposed protocols. Sometimes they tend to return the arbitrary computation results to the EUs and may also provide fake reputation value for EUs to collect more EUs' outsourced data.

• *Vehicles (EUs)*: EUs are assumed to be honest but curious about the location privacy of the vehicles (CUs). That is, they may faithfully follow the protocols of our proposed PTVC scheme, but may be curious and try to disclose CUs' private location information. Additionally, they are also honest to evaluate the CUs which participate in their tasks.

### C. Threat Model

Specifically, we consider that the attackers can launch the following four types of attacks to jeopardize the privacy of vehicles.

• *Reputation Spoofing Attack*: The CUs can impersonate other CUs and provide fake reputation values for the EUs to obtain more data for data analysis.

• *Data Analysis Attack*: On the one hand, the CUs can analyze the outsourced data coming from the EUs. On the other hand, they can also eavesdrop the transmission of messages and then analyze these messages.

• *Arbitrary Results Attack*: The CUs have the possibility to return the arbitrary computation results to the requesting EUs deliberately, and make the final results incorrect.

• *Pseudonyms Link Attack*: If the reputation value of a CU is directly given to the requesting EU, the location privacy of this CU may be breached since the requesting EU can still infer that it belongs to a preceding vehicle with the same reputation values even though the pseudonym of this vehicle is different.

### D. Design Goals

Under the models as mentioned above, our goal is to design a privacy-preserving trust-based verifiable computing scheme in a vehicular cloud, which should be effective in distinguishing the critical vehicles, protecting the private location information of all vehicles, and ensuring the correctness of the computation results in the meantime. Specifically, the following objectives need to be achieved: *i) Achieving the location privacy of each vehicle; ii) Guaranteeing the correctness of computation results; iii) Achieving anonymous authentication and data integrity.*

## III. Proposed PTVC Scheme

In this section, based on the verification techniques in [2], [5] and the efficient commitments scheme in [4], we propose the PTVC scheme, a privacy-preserving trust-based verifiable vehicular cloud computing scheme, which is mainly comprised of four phases: system setup, privacy-preserving trust-based vehicle selection protocol, privacy-preserving verifiable computing protocol and trust management.

### A. System Setup

TA takes charge of bootstrapping the whole system and generating the public and private keys for vehicles and road-side units (RSUs). Specifically, TA executes the bootstrap as follows.

• Given a security parameter $\xi$, TA firstly generates the bilinear parameters $(q, g, \mathbb{G}, \mathbb{G}_T, e)$ by running *Gen()* and then randomly choose a different generator $h$ of $\mathbb{G}$. Moreover, TA defines a public cryptographic hash function $H : \{0,1\}^* \rightarrow Z_q^*$, a general pseudo random function (PRF) $F_k()$ such as AES-128 algorithm, and a symmetric encryption algorithm $E_k()/D_k()$. TA also generates a $\lambda$-bits large prime $p$.

• TA defines $l$ trust levels $\{TL_1, TL_2, ..., TL_l\}$ for all reputation scores from $-100$ to 100. For instance, $TL_1$ is with $(-100, -80]$, $TL_2$ is with $[-80, -60)$,..., $TL_{10}$ is with $[80, 100)$. Then, TA selects $l$ random elements $\{y_1, y_2, ..., y_l\} \in Z_q^*$ as the master keys, and calculates the public key as $\{Y_1 = g^{y_1}, Y_2 = g^{y_2}, ..., Y_1 = g^{y_l}\}$.

• For each RSU, TA first generates an identity $RID$, and calculates its private key and public key as $(s_r, S_r)$, where $s_r$ is randomly chosen in $Z_q^*$ and $S_r = g^{s_r}$.

• Each vehicle $V_i$ first need to register itself in TA, and obtains a real identity $ID_i$ and an initial reputation score $k_i$. To protect their location privacy, TA generates $W$ pseudonyms, the corresponding key pairs and the signature of reputation values for each registered vehicle $V_i$ as follows.

- TA chooses a symmetric key $k_0$ and computes $W$ pseudonyms for $V_i$ as $\{PID_{i1} = E_{k_0}(ID_i||r_1), ..., PID_{iW} = E_{k_0}(ID_i||r_W)\}$ with random number $r_j \in Z_q^*$, where $j \in [1, W]$.
- Afterwards, TA generates the key pairs for all pseudonyms as $\{(s_{i1}, S_{i1}), ..., (s_{iW}, S_{iW})\}$, where $s_{i1}, ...., s_{iW}$ are all randomly chosen from $Z_q^*$ and $S_{i1} = g^{s_{i1}}, ..., S_{iW} = g^{s_{iW}}$.
- If the reputation value $k_i \in TL_x$, where $x \in [1, l]$, for each pseudonym $PID_{ij}$, TA makes a signature as $A_{ij} = g^{\frac{1}{y_x + k_i + H(T_i) + s_{ij}}}$, where $T_i$ is the timestamp when updating the reputation value of $V_i$.

• Finally, TA publishes the public parameters as $< q, g, \mathbb{G}, \mathbb{G}_T, e, h, p, H(), E(), D() >$.

### B. Privacy-Preserving Trust-Based Vehicle Selection Protocol

If a vehicle (EU) wants nearby vehicles (CUs) to form a temporary vehicular cloud (VC), the EU first needs to locate the high-reputation vehicles (CUs). Therefore, a privacy-preserving trust-based vehicle selection protocol has been proposed as follows.

*Step-1.* A vehicle (EU) $V_i$ sets a threshold trust level $TL'$, chooses a pseudonym $PID_{i\delta}$ and broadcasts the computation requests as $PID_{i\delta}||H(T)^{s_{i\delta}}||Req||TL'||T$, where $T$ is the current timestamp, to nearby vechicles (CUs) through VANET network.

*Step-2.* After receiving the $V_i$'s requests, if the vehicle $V_j$ has free computation resources, it first verifies whether the request is coming from the registered vehicle by computing $e(g, H(T)^{s_{i\delta}}) = e(S_{i\delta}, H(T))$. If it does hold, $V_j$ accepts the request, otherwise, rejects it. Then, $V_j$ judges whether its reputation value $k_j$ statisfies the trust level requirement of $V_i$. That is, $k_j$ is in the trust level $TL_x$, and the minimum value in $TL_x$ should be equal to or larger than the minimum value in $TL'$. If $V_j$ is eligible, it will calculate the proofs and reply to $V_i$ as $PID_{j\kappa}||H(T')^{s_{j\kappa}}||Resp||TL_x||\Pi||T'||T_j$ to $V_i$ as follows, where $T'$ is the current timestamp and $T_j$ is the timestamp that $V_j$ updates its last reputation.

- $V_j$ encrypts its reputation value as $C = g^{k_j} h^r$, where $r$ is a random number chosen in $Z_q^*$.
- $V_j$ chooses a pseudosym $PID_{j\kappa}$ and a random element $v \in Z_q^*$, and calculates as follows.

$$B = A_{j\kappa}^v = g^{\frac{v}{y_x + k_j + H(T_j) + s_{j\kappa}}} \quad (1)$$

$$D = B^{-k_j} g^v = g^{\frac{v(y_x + H(T_j) + s_{j\kappa})}{y_x + k_j + H(T_j) + s_{j\kappa}}} \quad (2)$$

- $V_j$ randomly chooses $k_j', r', v' \in Z_p^*$ and computes $C' = g^{k_j'} h^{r'}$ and $D' = B^{-k_j'} g^{v'}$.
- $V_j$ calculates the proofs $\Pi = C, B, D, s_1, s_2, s_3, \phi$ as follows.

$$\phi = H(C, B, D, C', D', H(T)^{s_{i\delta}}) \quad (3)$$

$$s_1 = k_j' + \phi \cdot k_j \bmod q, s_2 = r' + \phi \cdot r \bmod q,$$
$$s_3 = v' + \phi \cdot v \bmod q \quad (4)$$

*Step-3.* After receiving the response from $V_j$, the vehicle $V_i$ first checks whether $e(g, H(T')'^{s_{j\kappa}}) = e(S_{j\kappa}, H(T'))$.

Next, $V_i$ checks the timestamp $T_j$ and judges whether $V_j$ has a relatively new reputation. Then, to verify $V_j$'s reputation value $k_j$, it checks that $e(D,g) = e(B, Y_x \cdot S_{j\kappa} \cdot g^{H(T_j)})$ and calculates as follows.

$$\hat{C} = g^{s_1} h^{s_2} C^{-\phi} = C', \hat{D} = B^{-s_1} g^{s_3} D^{-\phi} = D' \quad (5)$$

Finally, $V_j$ checks that $\phi = H(C, B, D, \hat{C}, \hat{D}, H(T)^{s_{i\delta}})$. If it does hold, $V_j$ is chosen by $V_i$ (EU) as one computing unit (CU).

### C. Privacy-Preserving Verifiable Computing Protocol

If the vehicle (EU) wants to outsource its data to the VC without leaking privacy and receive the correct computation results eventually, it needs to encrypt its data before and be able to verify the results from the VC. Hence, a privacy-preserving verifiable computing protocol has been proposed as follows. Following the $\tau$-security definition [5], to describe this protocol precisely, we suppose there are total $U$ CUs, and even though $\tau$ CUs at most collude together, the outsourced data is still privacy-preserving. Additionally, we further assume that all transmitting packets in this protocol are encrypted with the session key and this session key is generated by the communication vehicles using their key pairs.

*Step-1.* Let $N$ and $S$ be integers, where $S < p$ and $N < p$, and the EU owns the data $m = (m_{i,j}) \in Z_p^{N*S}$, which is a collection of $S$ datasets that are to be outsourced. The outsourced function $f(x_1, x_2, ..., x_n)$ has $N$ inputs, and the degree of this function should be less than $U/\tau$. The EU then generates $N * S * \tau$ random coefficients $(f_{i,j,1}, ..., f_{i,j,\tau}) \in Z_p^{N*S}$ and $U$ different elements $a_1, ..., a_U \in Z_p^*$. For all $i \in [1, N]$ and $j \in [1, S]$, let $f_{i,j}(x) = m_{i,j} + f_{i,j,1}x + ... + f_{i,j,\tau}x^\tau$ and the EU calculates the $M_{u,i,j} = f_{i,j}(a_u)$ for each $u \in [1, U]$. Finally, the EU obtains the private outsourced data as $M = \{(M_{1,i,j}) \in Z_p^{N*S}, ..., (M_{U,i,j}) \in Z_p^{N*S}\}$ for all CUs.

*Step-2.* For each CU, the EU randomly selects an element $\beta_u \in Z_p \backslash \{1, 2, ..., s\}$ and generates a polynomial $\theta_i(x) = \theta_{i,1} + \theta_{i,2}x + ... + \theta_{i,s}x^{s-1} + t_i \cdot x^s$ for every $i \in [1, N]$ by solving the following matrix equation, which satisfies $\theta_i(j) = M_{u,i,j}$ and $\theta_i(a) = F_k(i)$.

$$\begin{pmatrix} \theta_{i,1} \\ \theta_{i,2} \\ \vdots \\ \theta_{i,s} \\ t_{u,i} \end{pmatrix} = \begin{pmatrix} M_{u,i,1} \\ M_{u,i,2} \\ \vdots \\ M_{u,i,s} \\ F_k(i) \end{pmatrix} \begin{pmatrix} 1 & 1^1 & 1^2 & \dots & 1^s \\ 1 & 2^1 & 2^2 & \dots & 2^s \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & s^1 & s^2 & \dots & s^s \\ 1 & \beta^1 & \beta^2 & \dots & \beta^s \end{pmatrix}^{-1} \quad (6)$$

The EU then sends $\{t_{u,1}, t_{u,2}, ..., t_{u,N}\} || (M_{u,i,j}) \in Z_p^{N*S}$ to the corresponding CU, and the verification key $vk_u$ can be calculated as $vk_u = f(F_k(1), F_k(2), ..., F_k(N))$.

*Step-3.* After receiving the outsourced data, the CU calculates the results as $\rho_u = \{\rho_{u,1}, \rho_{u,2}, ..., \rho_{u,S}\}$, where $\rho_{u,j} = f(M_{u,1,j}, M_{u,2,j}, ..., M_{u,N,j})$ for all $j \in [1, S]$. Then, the CU also generates a polynomial $\theta_i(x) = \theta_{i,1} + \theta_{i,2}x + ... + \theta_{i,s}x^{s-1} + t_i \cdot x^s$ for every $i \in [1, N]$ by solving the following matrix equation, and the correctness proof can be calculated as $\pi_u = f(\theta_1(x), \theta_2(x), ..., \theta_N(x))$.

$$\begin{pmatrix} \theta_{i,1} \\ \theta_{i,2} \\ \vdots \\ \theta_{i,s} \end{pmatrix} = \begin{pmatrix} M_{u,i,1} - t_i \\ M_{u,i,2} - 2^s t_i \\ \vdots \\ M_{u,i,s} - s^s t_i \end{pmatrix} \begin{pmatrix} 1 & 1^1 & 1^2 & \dots & 1^s \\ 1 & 2^1 & 2^2 & \dots & 2^s \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & s^1 & s^2 & \dots & s^s \end{pmatrix}^{-1} \quad (7)$$

Finally, the CU sends $\rho_u || \pi_u$ back to the EU.

*Step-4.* The EU accepts $\rho_u$ if $\pi_u(\beta_u) = vk_u$ and $\pi_u(j) = \rho_{u,j}$ for every $j \in [1, S]$. After verifying and receiving computation results from $U$ CUs, the EU can recover the value $\{z_1, z_2, ..., z_S\}$ by interpolating the univariate polynomial $\{G_1(x), G_2(x), ..., G_S(x)\}$ as follows.

$$\begin{pmatrix} G_1(a_1) = \rho_{1,1} & G_1(a_2) = \rho_{2,1} & \dots & G_1(a_U) = \rho_{U,1} \\ G_2(a_1) = \rho_{1,2} & G_2(a_2) = \rho_{2,2} & \dots & G_2(a_U) = \rho_{U,2} \\ \vdots & \vdots & \vdots & \vdots \\ G_S(a_1) = \rho_{1,S} & G_S(a_2) = \rho_{2,S} & \dots & G_S(a_U) = \rho_{U,S} \end{pmatrix} \quad (8)$$

That is, the EU can reconstruct the final results as $\{G_1(0) = z_1, G_2(0) = z_2, ..., G_S(0) = z_S\}$.

### D. Trust Management

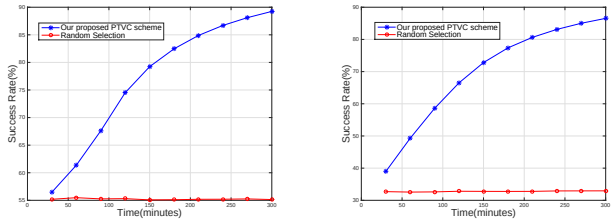
After the VC returns the computation results, the vehicle (EU) needs to give the feedbacks on the performance of all participating vehicles (CUs) to TA, and these feedbacks can help to update the reputation values of all vehicles. Specifically, the feedback for each participating CU is either positive or negative. In other words, if the CU honestly returns the correct computation results to the EU, the feedback of this CU will be positive (1). Otherwise, the feedback of this CU will be negative (0).

*Reputation Updating:* The EU $V_i$'s feedbacks are $(ID_i || FBs)$, and $FBs$ involves the pseudonyms of all particpating CUs and their scores (1 or 0). Upon receiving the feedbacks from the EU $V_i$, TA performs the following procedures to update the vehicle's reputation values: i) TA searches the reputation database to get the real identities of the all CUs' based on their pseudonyms and finds their reputation values; ii) according to the reputation rating function [8], the updated reputation value can be calculated as $k_j = Rep(r, s) * 100 = \frac{100(r-s)}{r+s+2}$ for the vehicle $V_j$, and $k_j$ is located in $y_x$; iii) for $W$ pseudonyms of $V_j$, TA will also make a new signature as $A_{j\kappa} = g^{\frac{1}{y_x + k_j + H(T_j) + s_{j\kappa}}}$, where $T_j$ is the current timestamp.

Furthermore, the frequency of updating reputation value is determined by TA. When there are more computation tasks in the system, TA will require the vehicles to update their reputation values more frequently, and vice versa. When a vehicle drives into the wireless range of the RSU, it will retrieve its updated reputation value from TA via VANET network immediately.

## IV. Security Analysis

In this section, we discuss the security properties of our proposed PTVC scheme. Specifically, our analyzes focus on the




(a) $NUM = 200$ (b) $NUM = 400$

Fig. 3. The simulation results with different settings

resilience against the attack strategies mentioned in Section II as follows.

*Resilience against Reputation Spoofing Attack*: A small fraction of vehicles (CUs) may cheat the vehicle (EU) by using the fake reputations and the fake identities in our system. Our proposed PTVC scheme is designed to detect those dishonest vehicles. According to our proposed protocols, a vehicle $V_i$ needs to be authenticated with pseudonym by computing $e(g, H(T)^{s_{i\delta}}) = e(S_{i\delta}, H(T))$. With a given pseudonym $PID_{i\delta}$, only the corresponding vehicle has the private key $s_{i\delta}$ and computes the signature $H(T)^{s_{i\delta}}$ with the current timestamp. The timestamp $H(T)$ can be used to resist the replay attack. In this case, none of the vehicles can impersonate other vehicles in our scheme. Additionally, all pseudonyms are binding with specific signature of reputation value $A_{i\delta}$. Therefore, unless the vehicle knows the master key $y_x$, it is impossible for $V_i$ to spoof other vehicles with a fake reputation value.

*Resilience against Data Analysis Attack*: The curious vehicles may analyze the outsourced data and eavesdrop the messages transmitting in our system to disclose the privacy of other vehicles. Our proposed PTVC scheme is designed to resist the data analysis attack. In the vehicle selection protocol, the reputation values and proofs are all masked with the random numbers. In the verifiable computing protocol, all transmitting data are encrypted. Moreover, the outsourced data $m = (m_{i,j}) \in Z_p^{N*S}$ is divided into $U$ pieces $M = \{(M_{1,i,j}) \in Z_p^{N*S}, ..., (M_{U,i,j}) \in Z_p^{N*S}\}$. As long as $\tau + 1$ CUs do not collude together, the outsourced data is privacy-preserving, following the security definition in [5].

*Resilience against Arbitrary Results Attack*: To cope with the incorrect computation results and detect these bad vehicles, our proposed PTVC scheme supports the EU to verify the results from other vehicles and use the feedbacks from the EU to adjust the reputation values of the participating vehicles in our system. Specifically, with the verification key $vk_u$, the correctness of results can be verified by computing $\pi_u(\beta) = vk_u$ and $\pi_u(j) = \rho_{u,j}$ for every $j \in [1, S]$. In the meantime, with the trust management, these bad vehicles' reputation values will be low and none of the EUs would like to choose these vehicles as the CUs.

*Resilience against Pseudosyms Link Attack*: Although the pseudosyms make the participating vehicles anonymous, there still exists a risk that a vehicle can be linked based on its unchanged reputation value in a long period. For instance, when a EU $V_i$ judge whether a CU $V_j$?s reputation satisfies its requirement, it does not have to know the exact reputation value of $V_j$. Instead, the $V_j$ can provide the proofs that it has the high trust level. Specifically, the $V_j$ has the signature $A_{j\kappa} = g^{\frac{1}{y_x + k_j + H(T_j) + s_{j\kappa}}}$. With the proofs $B = A_{j\kappa}^v$, $D = B^{k_j} g^v$, $\phi$, $s_1$, $s_2$, and $s_3$, $V_i$ can verifiy the proofs by computing $e(D, g) = e(B, Y \cdot S_{j\kappa} \cdot g^{H(T_j)})$ and $\phi = H(C, B, D, \hat{C}, \hat{D}, H(T)^{s_{i\theta}})$. Afterwards, $V_i$ will know $V_j$ is a high-reputation vehicle but not know its exact reputation value, i.e., it cannot link the current pseudosym with the precious pseudosyms.

## V. Performance Evaluation

In this section, we discuss the performance of PTVC scheme using a custom simulator built in Java 8. The performance metrics used in our proposed scheme are in terms of the computational cost and the success rate (i.e., the ratio of achieving the computation tasks correctly in a period).

### A. Simulation Setup

To simulate the VANET, an area of $5km * 5km$ is created with roads that are equally laid with the distance of $1km$ in between. One RSU with transmission radius of $1km$ is deployed at an intersection of the map. $NUM$ vehicles with a transmission range of $300m$ are deployed with an initial reputation 100, as shown in Fig. 3 (a). We also choose the trust levels (-4 to 5) to represent various reputation values in $[-100, 100]$, and set the threshold of trust level as 2. In our simulator, all vehicles randomly choose a destination and follow the shortest path map based movement at a constant speed $36km/h$, After reaching the destination, they will stop for 5 minutes, and then choose a new random destination to repeat the above. Each vehicle will generate a computation task every 5 minutes, and only choose the vehicles whose trust level is larger than or equal to the threshold. In addition, we assume that the proportion of bad vehicles, defined as the number of bad vehicles to the total number of vehicles, is $20\%$ and they may return the arbitrary computation results with the probablity of $50\%$. Other detailed parameters are set as follows: $\xi = 512$, $\lambda = 500$, $\tau = 2$, $N = 10$, $S = 10$ and $U = 5$, and the function is defined as the $f(x_1, x_2, ..., x_N) = \sum_{i=1}^{N}(x_i - c)^2$, where $c$ is a constant value. Furthermore, our experiment environment is a laptop with 3.1 GHz processor, 8GB RAM, and Window 7 platform.

### B. Simulation Results

We simulate the process of privacy-preserving vehicle selection protocol (Protocol1) and privacy-preserving verifiable computing protocol (Protocol2) 1000 times, and the simulation results show that the computational cost is acceptable. Specifically, the average running time (between one EU and one CU) of Protocol1 is 832ms for EU (Step-1 and Step-3) and 824ms for CU (Step-2), respectively. In Protocol2, the EU first needs to initialize the outsourced data (Step-1 and Step-2), which costs almost 1s, and then for each CU, the average

computation time (Step-3) is almost 200ms while the average verification and reconstruction time for the EU (Step-4 and Step-5) are approximately 100ms. Note that, the value of $N$, $S$, $U$ and $\tau$ can be changed according to the real computation tasks and the computation ability of nearby vehicles.

Moreover, in Fig. 3 (b) and (c), we compare the success rate with different settings, i.e., i) with/without the threshold of trust level; ii) different vehicle number $NUM = 200, 400$; iii) $20\%$ vehicles with the probability of $50\%$ to return the arbitrary computation results. From the figure, we can see that the success rate is very low when the vehicles are chosen randomly (without the threshold of trust level 2). By contrast, after adopting our proposed PTVC scheme, the success rate can be improved from $50\%$ to almost $90\%$ and $30\%$ to $85\%$, which demonstrates the effectiveness of our proposed PTVC scheme in improving the success rate. From the figure, it can also be seen that the success rate increases with the rise of vehicle number. The reason is that when there are more vehicles, more computation tasks will be generated and the bad vehicles have more chances to take the tasks and then be detected by our PTVC scheme, which finally makes them be excluded in the following computation tasks from the EUs.

## VI. Related Work

Our work is mainly related to not only the privacy preservation in VANET but also the verifiable computing in the vehicular cloud. Also, we consider the trust management as the supplement to make our PTVC scheme better. Hence, we will briefly review some related work in this section.

Many efforts have been finished [3], [9] on preserving the privacy of vehicles in VANET. Using the pseudonyms, these approaches can ensure the anonymity of all vehicles and address the privacy issues in VANET. In [3], Lu et al. consider a scenario when the pseudonyms are changed at improper location or time on the road, where the location and velocity information embedded in the safety messages could still provide a clue for the adversary to link the pseudonyms, which leads to the failure of privacy preservation. Moreover, how to achieve anonymity and trust at the same time has been a promising research issue which attracts researchers? attention in recent years, some of the works [10]–[12] have been done. For instance, in [10], the authors present a way to transfer the reputation values belonging to the same user between its different pseudonyms. This kind of transfer is manipulated by a trusted server, who also maintains a list of mappings between real user identity and all the associated pseudonyms. Another similar solution is proposed in [12], called IncngniSense. In this framework, a blind signature is used to generate periodic pseudonyms and a reputation transfer based on cloaking mechanism is presented. Different from the above existing solutions, our PTVC scheme not only considers the anonymous authentication and trust management in VANET, but also combines them with verifiable computing techniques [2], [5] to achieve privacy-preserving trust-based verifiable vehicular cloud computing.

## VII. Conclusion

In this paper, we have proposed a privacy-preserving trust-based verifiable vehicular cloud computing scheme, called PTVC. With the multiple vehicles nearby, the proposed PTVC can select the trustworthy vehicles to form the temporary vehicular cloud and verify the correctness of computation results without privacy leakage. Detailed security analysis shows that PTVC is privacy-preserving and robust against several various attacks. In addition, through extensive performance evaluation, we have also demonstrated that the proposed PTVC scheme is effective in terms of computational cost and can help improve the success rate by differentiating the bad vehicles. In future work, we intend to carry on smartphone-based/vehicle-based experiments to further verify the effectiveness of the proposed PTVC scheme. In addition, we will also exploit the security model of malicious attackers, where the attackers will not honestly follow the protocol.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61303218 and 61672411, and R. Lu also thanks for the SUG support of UNB.

## References

[1] J. Zhang, "A survey on trust management for vanets," in *25th IEEE International Conference on Advanced Information Networking and Applications, AINA 2011, Biopolis, Singapore, March 22-25, 2011*, 2011, pp. 105–112.

[2] L. F. Zhang and R. Safavi-Naini, "Batch verifiable computation of outsourced functions," *Des. Codes Cryptography*, vol. 77, no. 2-3, pp. 563–585, 2015.

[3] R. Lu, X. Lin, T. H. Luan, X. Liang, and X. S. Shen, "Pseudonym changing at social spots: An effective strategy for location privacy in vanets," *IEEE Trans. Vehicular Technology*, vol. 61, no. 1, pp. 86–96, 2012.

[4] G. Arfaoui, J. Lalande, J. Traoré, N. Desmoulins, P. Berthomé, and S. Gharout, "A practical set-membership proof for privacy-preserving NFC mobile ticketing," *PoPETs*, vol. 2015, no. 2, pp. 25–45, 2015.

[5] L. F. Zhang, R. Safavi-Naini, and X. W. Liu, "Verifiable local computation on distributed data," in *Proceedings of the Second International Workshop on Security in Cloud Computing, SCC@ASIACCS '14, Kyoto, Japan, June 3, 2014*, 2014, pp. 3–10.

[6] S. Bitam, A. Mellouk, and S. Zeadally, "Vanet-cloud: a generic cloud computing model for vehicular ad hoc networks," *IEEE Wireless Commun.*, vol. 22, no. 1, pp. 96–102, 2015.

[7] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, "A survey on vehicular cloud computing," *J. Network and Computer Applications*, vol. 40, pp. 325–344, 2014.

[8] A. Jøsang and R. Ismail, "The beta reputation system," in *Proceedings of the 15th bled electronic commerce conference*, vol. 5, 2002, pp. 2502–2511.

[9] R. Lu, X. Lin, H. Zhu, P. Ho, and X. Shen, "ECPP: efficient conditional privacy preservation protocol for secure vehicular communications," in *Proceedings of the IEEE INFOCOM 2008, Phoenix, AZ, USA, 13-18 April 2008*, 2008, pp. 1229–1237.

[10] X. O. Wang, W. Cheng, P. Mohapatra, and T. F. Abdelzaher, "Artsense: Anonymous reputation and trust in participatory sensing," in *Proceedings of the IEEE INFOCOM 2013, Turin, Italy, April 14-19, 2013*, 2013, pp. 2517–2525.

[11] Q. Li and G. Cao, "Providing efficient privacy-aware incentives for mobile sensing," in *IEEE 34th International Conference on Distributed Computing Systems, ICDCS 2014, Madrid, Spain, June 30 - July 3, 2014*, 2014, pp. 208–217.

[12] D. Christin, C. Roßkopf, M. Hollick, L. A. Martucci, and S. S. Kanhere, "Incognisense: An anonymity-preserving reputation framework for participatory sensing applications," *Pervasive and Mobile Computing*, vol. 9, no. 3, pp. 353–371, 2013.