Conditional Privacy-Preserving Authentication Protocols for Vehicular Ad Hoc Networks

Dissertation
zur Erlangung des mathematisch-naturwissenschaftlichen
Doktorgrades
"Doctor rerum naturalium"
der Georg-August-Universität Göttingen

im Promotionsprogramm Computer Science (PCS)
der Georg-August University School of Science (GAUSS)

vorgelegt von

Jiliang Li
aus Shandong, China

Göttingen, im 2019

Betreuungsausschuss

Prof. Dr. Dieter Hogrefe,
Georg-August-Universität Göttingen

Prof. Dr. Marcus Baum,
Georg-August-Universität Göttingen

Mitglieder der Prüfungskommission

Referent:	Prof. Dr. Dieter Hogrefe, Georg-August-Universität Göttingen
Korreferent:	Prof. Dr. Marcus Baum, Georg-August-Universität Göttingen
	Prof. Dr. Yusheng Ji, National Institute of Informatics, Japan

Weitere Mitglieder der Prüfungskommission

Prof. Dr. Xiaoming Fu,
Georg-August-Universität Göttingen

Prof. Dr. Delphine Reinhardt,
Georg-August-Universität Göttingen

Prof. Dr. Winfried Kurth,
Georg-August-Universität Göttingen

Tag der mündlichen Prüfung

17. May 2019

Statement

I hereby declare that I have written this thesis independently without any help from others and without the use of documents or aids other than those stated. I have mentioned all used sources and cited them correctly according to established academic citation rules.

Göttingen, June 2019

Abstract

The Conditional Privacy-Preserving Authentication (CPPA) protocol has applications in the construction of secure Vehicular Ad hoc Networks (VANETs) due to its capability to achieve both privacy preservation and authentication simultaneously. Although a number of CPPA protocols have been proposed in the literature, existing approaches generally suffer from limitations such as the security problem of system private keys, high computation requirement during certificate generation and message verification phases. To resolve these issues, this thesis firstly presents a Certificateless and Provably-Secure Conditional Privacy-Preserving Authentication (CPS-CPPA) protocol for VANETs based on the Tamper-Proof Device (TPD). To improve efficiency further, the proposed CPS-CPPA scheme added the function of batch verification. However, this thesis has found out that the CPS-CPPA protocol cannot guarantee the secrecy of one master key in practice and not withstand the forged message attack and impersonation attack. To overcome the vulnerabilities of CPS-CPPA protocol, this thesis presents an Enhanced, Certificateless and Provably-Secure Conditional Privacy-Preserving Authentication (ECPS-CPPA) protocol to be used in vehicular environments that supports both privacy and security requirements in the VANETs system. This thesis also demonstrates that the ECPS-CPPA protocol is secure against forged message attack, impersonation attack, and other existing attacks. A comparative summary shows that our ECPS-CPPA protocol has favorable computation and communication overheads in comparison to the other two recently published protocols. In the future, it is important to implement a proof of concept of this protocol in order to evaluate the real-world utility of ECPS-CPPA protocol.

Acknowledgements

I would like to express my cordial gratitude to the people who supported me during my PhD studies.

First and foremost, I cordially appreciate my first supervisor Prof. Dr. Dieter Hogrefe for his support, guidance and encouragement throughout my PhD studies. He is a very nice and knowledgeable Professor and has created an enjoyable working environment for us that stimulates initiative. It has been an great honor to be his PhD student. I appreciate all his contributions of suggestions and funding to make my PhD study productive.

I sincerely appreciate Prof. Dr. Marcus Baum and Prof. Dr. Yusheng Ji, who contributed to many discussions on my work. They have given me valuable encouragement and feedback. Furthermore, I would like to thank the thesis committee Prof. Dr. Dieter Hogrefe, Prof. Dr. Marcus Baum, Prof. Dr. Yusheng Ji, Prof. Dr. Xiaoming Fu, Prof. Dr. Delphine Reinhardt, and Prof. Dr. Winfried Kurth for investing their valuable time.

In addition, I would like to express my gratitude to all the members of Telematics group and all my co-authors including Prof. Dr. Dieter Hogrefe, Prof. Dr. Yusheng Ji, Prof. Dr. Weiguo Zhang, Prof. Dr. Kim-Kwang Raymond Choo, Dr. Saru Kumari, Prof. Dr. Joel J. P. C. Rodrigues, Prof. Dr. Muhammad Khurram Khan, and Prof. Dr. Mohammad S. Obaidat. Without their help and support, I can not complete my PhD work smoothly.

I cordially appreciate the funding sources that have made my PhD work possible. The China Scholarship Council offers me three years scholarship to support my

PhD study.

Finally, I cordially appreciate my family for all of their encouragement, love and support.

Contents

Chapter 1

Introduction

In this chapter we introduce an overview about CPPA protocols for Vehicular Ad hoc Networks (VANETs) in Section 1.1, as well as the contributions and organization of the thesis in Section 1.2 and Section 1.3 respectively.

1.1 Overview

Due to constant and rapid advancements in the development of wireless communication and network technologies, VANETs have regained renewed interest due to their capability to support vehicles with wireless devices to communicate with other vehicles and Roadside Units (RSUs) and ensure traffic safety and enhance driving efficiency [1–8]. Other benefits associated with VANETs include collision avoidance, lane merging, traffic optimization, toll collection, location-based services, infotainment, etc [9]. In the literature, such settings have also been considered Internet of Vehicles (IoV) and smart cities [10, 11].

One can think of VANETs as a combination of Mobile Ad hoc Networks (MANETs) with vehicles (e.g. cars, buses, trucks and motorcycles) and RSUs [3, 12, 13]. Unlike nodes in a Mobile Ad hoc Network (MANET), vehicles are not usually resource constrained in terms of power, storage space and computing capability. A typical Vehicular Ad hoc Network (VANET) includes Trusted Authorities (TAs) (e.g. traf-

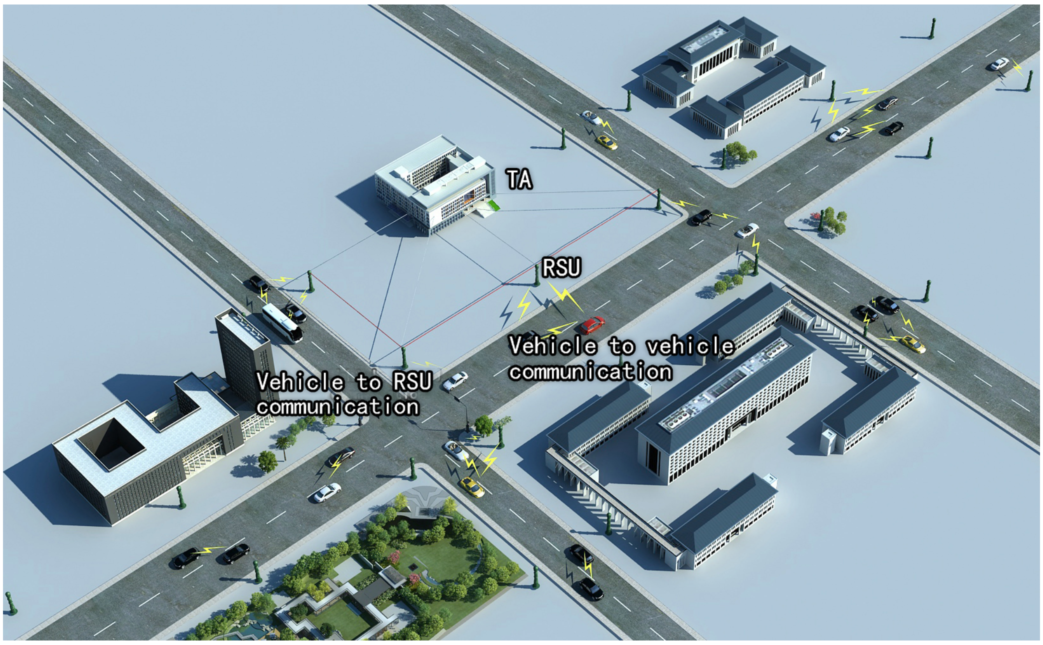
TA
RSU
Vehicle to vehicle communication
Vehicle to RSU communication

Figure 1.1: A typical structure of VANETs

fic authority centers), RSUs (e.g. placed on road sides or other installations), and Onboard Units (OBUs) equipped on vehicles [3, 14, 15] – see Figure 1.1.

Communications in VANETs, such as Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I), use Dedicated Short Range Communication (DSRC), which is a short medium range communications protocol [14]. Every vehicle could communicate with adjacent vehicles and the nearby RSUs located at the roadside through the Onboard Unit (OBU) installed in the vehicle and DSRC protocol. For example, on-vehicle OBUs periodically broadcast traffic-related information covering factors such as position, weather conditions, direction, speed, and traffic situation. Such information allow participating vehicles in the vicinity to take the required actions, for example take an alternate route to avoid a traffic accident, traffic congestion, etc [16, 17]. RSUs and other vehicles can also transmit traffic-related information (e.g. an accident that has just taken place) to the traffic administration department or other relevant department (e.g. law enforcement or fire department), so that the necessary actions can be undertaken [18]. Hence, it is not surprising that VANETs and the many variants (e.g. IoV, Intelligent Transport

Systems (ITS), and smart cities) have received recent attention [9].

Similar to other wireless networks, there are a number of other features important to VANETs, such as the following:

Security: Once attackers have controlled over the communication channels, they could easily eavesdrop, tamper, replay or even drop messages sent within VANETs. In other words, designers of VANETs need to ensure the system is secure against a wide range of attacks such as masquerading, replaying, tunneling, message modification, key and certificate replication attacks [9, 14, 18]. For example, an malicious adversary may hijack and modify the initial messages or masquerade one legitimate vehicle to broadcast 'fake' messages, resulting in chaos or traffic incidents [18]. Hence, the capability to ensure the authenticity of messages from vehicles in VANETs is crucial.

Anonymity: If the vehicle user sends his/her identity to RSUs or other vehicles without masking, a malicious attacker may track the user's routes through capturing of the messages. The leakage of routes may have real-world consequences such as physical stalking, kidnapping, and assassination (e.g. a malicious adversary intercept and replace intercepted messages with fabricated messages in order to reroute the victim's vehicles). Therefore, anonymity is another key feature in VANETs [19].

Traceability (and *conditional privacy*): If a misbehaving vehicle transmits malicious or suspicious information to RSUs or nearby vehicles, then the system needs to have the ability to identify the vehicle (and the owner) so that the vehicle (and the owner) can be taken to task (e.g. monetary penalties or other criminal sanctions). Thus, both traceability and conditional privacy are important features [18]. Conditional privacy restricts to the Trusted Authority (TA) being the only party who can extract the vehicle's real identity.

CPPA schemes such as those presented in [3, 9, 12, 18–25] can be used to achieve both security and privacy related properties within VANETs. There are, however, limitations in these existing schemes as discussed in Section 3.1.

1.2 Contributions

In this thesis, it introduces two efficient, provably-secure and anonymous CPPA solutions for VANETs in order to overcome limitations in existing CPPA schemes. To be specific, four main contributions of our work are described as below.

• First, the vulnerabilities of existing schemes are retrospected and analyzed. Meantime, several security weaknesses of these schemes are pointed out. Then, it demonstrates several previously unknown flaws in the protocols of Azees et al. [20] and Zhang et al. [26], respectively.

• Second, this thesis presents a CPS-CPPA protocol for VANETs [27]. To improve efficiency further, the proposed CPS-CPPA scheme added the function of batch verification.

• Third, this thesis points out that the CPS-CPPA protocol cannot guarantee the secrecy of one master key in practice and not withstand the forged message attack and impersonation attack. To overcome the weaknesses of the CPS-CPPA protocol, this thesis presents an ECPS-CPPA protocol for VANETs.

• Finally, we also conducted a comparison of the computation overhead and communication overhead to prove that our ECPS-CPPA scheme possesses more favorable performance compared with existing solutions for VANETs.

1.3 Organization

The rest of this thesis is organized as follows. Chapter 1 provides the overview, contributions and organization. Chapter 2 shows the background and design goals for VANETs. Chapter 3 reviews the existing studies, and especially revisits and analyzes Azees et al.'s protocol and Zhang et al.'s protocol respectively. Chapter 4 presents a CPS-CPPA protocol [27]. Chapter 5 points out the weaknesses in the CPS-CPPA protocol in Chapter 4, and then presents an ECPS-CPPA protocol. Chapter 6 summarizes the computation and communication overheads comparison. At last, Chapter 7 concludes this thesis.

Chapter 2

Preliminaries

In this chapter we introduce some preliminaries about CPPA protocols for VANETs. In particular, Section 2.1 introduces the network model and preliminaries about cryptography that used in this thesis. In Section 2.2 we list the design goals according to the existing literature.

2.1 Background

2.1.1 Network Model

As shown in Figure 2.1, the two-level network model is pretty adaptable for VANETs, in which the TA is set as the first-level, and RSUs as well as vehicles are set as the second level, respectively. The functions of these three entities are described as below.

TA : TA is fully trusted by all parties of VANETs and has sufficient computation, communication and storage capabilities. The TA is also responsible for the generation of system parameters and the registration of RSUs and vehicles. In addition, upon successful completion of their registration, the TA initially generates the security parameters for all vehicles and RSUs, and stores them into the vehicles and RSUs offline. It is capable of recovering the genuine Identity (ID) of vehicle

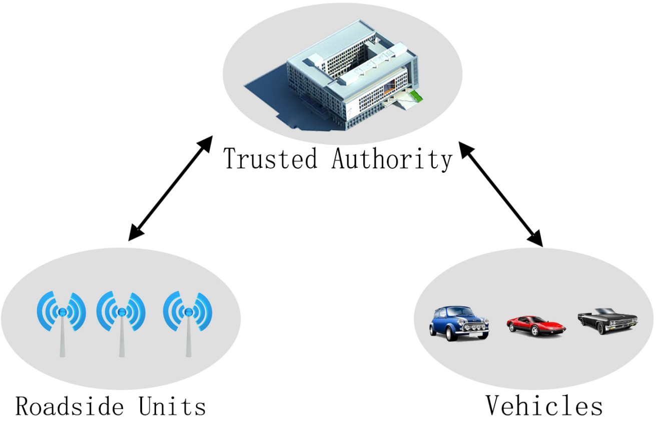

Figure 2.1: The network model of VANETs

from the transmitted message.

RSUs : RSUs are stationary infrastructures deployed on the roadside or some installations (e.g. bus stops). RSUs serve as the 'interface' between the TA and vehicles, and utilizes the DSRC [28] protocol for V2V and V2I wireless communications. It could authenticate traffic messages from vehicles and process them locally or forward them to TA. In our solution, RSUs are semi-trusted. If an RSU was compromised, then TA could detect and either reset the compromised RSU or remove/replace it.

Vehicle : Every vehicle is equipped with an OBU, which allows the vehicle to communicate wirelessly with other vehicles and RSUs using the DSRC protocol. Every OBU may have a Tamper-Proof Device (TPD) to protect stored secret information, e.g. secret keys etc.

2.1.2 Preliminaries about cryptography

Here, we will review three key cryptographic primitives, namely: bilinear pairings, Discrete Logarithm (DL) problem, and Computational Diffie-Hellman (CDH) problem [29].

Let $e : G_1 \times G_2 \rightarrow G_3$ be a rational function, where G_1, G_2, G_3 are three groups with a large prime order q. Let g_1 and g_2 respectively denote the generators of G_1

and G_2. e is called a bilinear pairing if it satisfies the below three properties:

- **Bilinearity:** For elements $g_1 \in G_1$, $g_2 \in G_2$ and $v, w \in Z_q^*$, $e(g_1^v, g_2^w) = e(g_1, g_2)^{vw}$ holds.
- **Nondegeneracy:** $e(g_1^v, g_2^w) \neq 1_{G_3}$.
- **Computability:** For any two elements $V \in G_1$, $W \in G_2$, we can compute $e(V, W)$ efficiently; that is, there is a valid algorithm to compute easily $e : G_1 \times G_2 \to G_3$.

In addition, it is known that there is no polynomial-time or efficient algorithm to resolve the below two hard problems.

- **DL Problem:** For an element $y \in G_1$ (or $y' \in G_2$), the DL challenge is to be able to compute $x \in Z_q^*$ such that $y = g_1^x$ (or $y' = g_2^x$) holds.
- **CDH Problem:** For two elements g_1^a, $g_1^b \in G_1$ (or g_2^a, $g_2^b \in G_2$) with two unknown elements $a, b \in Z_q^*$, the CDH problem is to compute $g_1^{a \cdot b} \in G_1$ (or $g_2^{a \cdot b} \in G_2$).

2.2 Design Goals

Based on the literature [16, 18–20, 23, 24, 27, 30–43], a secure and efficient CPPA solution for VANETs is supposed to satisfy the following requirements or goals.

Identity Privacy Preservation: RSUs, vehicles and third-party participants are not capable of extracting the vehicle's actual identity from the messages transmitted from any vehicle.

Message Authentication and Integrity: Every message transmitted by a vehicle should be authenticated by the receivers such as RSUs and other vehicles, and the receivers are capable of detecting any modification or fabrication of received messages.

Traceability: The TA is the only entity capable of extracting the vehicle's actual ID when the need arises (e.g. a complaint against a misbehaving vehicle).

Unlinkability: RSUs, vehicles and third-party participants are not capable of tracing the vehicle's behavior by analyzing its transmitted messages. That is, they cannot link and decide if two messages are transmitted from the identical vehicle.

Secrecy of Master Key: Although every vehicle or RSU is installed with a TPD, the highly-motivated attacker can extract the data memorized in the device by power analysis techniques [44]. Therefore, it is very essential to preserve the master key of VANETs system safely.

Resilient to Message Modification Attack: The adversary may transmit modified information around the VANETs system in order to achieve his/her specific goal. For example, an adversary would transmit fake/modified traffic information to his/her nearby vehicles for the sake of obtaining an optimal traffic route. Therefore, the modified messages are not supposed to pass the verification by the receivers (e.g. other vehicles and RSUs)

Resilient to Impersonation Attack: Such attacks are generally targeted at other legal vehicles. They are executed by sending fake messages to other vehicles in which the adversary attempts to masquerade as a trusted vehicle.

Resilient to Replay Attack: The replay attack is a form of network attack in which the transmitted information is fraudulently or maliciously delayed or repeated. Thus, the secure VANETs system should withstand such attack.

Full Batch Verification: It is not efficient for the receiver to authenticate the authenticity of received messages one by one, therefore, full batch verification is a necessary property in which the receiver could verify the legality of multiple messages from vehicles simultaneously.

No Map-to-Point Operation: It is expensive and complicated to execute the map-to-point operation, and consequently, it will degrade the performance of the VANETs system. Therefore, map-to-point operation is supposed to be avoided in a CPPA scheme for VANETs.

No Certificates Management: The overhead and complexity of certificates management increase with the number of registered vehicles. Besides, it is important to authenticate the legality of certificate prior to accept. To guarantee better feasibility and performance of vehicular system, it is capable of supporting no certificates management in the design of a CPPA scheme.

No Verifier Table: To avoid governance issue and attacks relating to verifier table, a CPPA protocol for VANETs must be capable of supporting no verifier table.

Provable Security: The security of cryptographic protocol is supposed to be proved via a widely acknowledged security model [45]. In another word, without the preciseness of a security proof, the customers would not be sure of the security of the cryptographic system. Therefore, a CPPA scheme is supposed to be proved securely under a security model.

Chapter 3

Existing Studies

In this chapter, Section 3.1 reviews the existing studies on CPPA protocols. Specifically, to vividly understand the weaknesses in the existing CPPA protocols, we choose two typical CPPA protocols [20, 26] to revisit and analyze. Section 3.2 reviews and analyzes the CPPA protocol of Azees et al. [20]. In Section 3.3, we revisit and demonstrate the insecurity of Zhang et al' CPPA protocol [26]. Finally, the summary is presented in Section 3.4.

3.1 State of the Art

This section briefly reviews existing literature on CPPA schemes designed for VANETs.

In 2006, Gamage et al. [21] introduced an ID-based ring signature solution to ensure privacy for VANETs applications. However, the presented approach does not provide traceability and this implies a lack of conditional privacy. A year later in 2007, Raya et al. [9] introduced a CPPA solution using anonymous certificates. Specifically, to mask the vehicle's real identity, a large number of key pairs and corresponding certificates based on Public Key Infrastructure (PKI) are preloaded into the memory space of vehicles' OBUs and the OBU randomly chooses a pair of private/public key that can be used for authentication. This imposes storage requirements for each vehicle (e.g. to store its private/public key pairs and corresponding certificates), and TA (e.g. to store all vehicles' certificates). For a large system with vehicles constantly joining and leaving, it is not a trivial task to search for and identify a misbehaving vehicle in practice. In 2008, a new CPPA solution using bilinear pairing is designed by Lu et al. [23]. In this solution, the RSU sends a temporary anonymous certificate to the vehicle which passes by the region of the RSU. The RSUs also provide the vehicles a fresh anonymous certificate periodically to enforce conditional privacy. Nevertheless, this solution has a low efficiency. In the same year, Lin et al. [46] provided a privacy-preserving protocol utilizing group signature technique, which provides traceability. However, in Lin et al.'s solution, each vehicle has to store the revocation list to avoid communicating with the 'blacklisted' vehicles. Therefore, as the number of revoked vehicles increases, the checker will need to spend considerably amount of time on the verification stage alone. This is clearly not practical.

In 2008, Zhang et al. [25] constructed an ID-based batch authentication protocol based on pairing-based cryptography. In their approach, both vehicles and RSUs do not need to store any certificate. Moreover, their solution provides batch verification for multiple messages. In other words, this CPPA solution overcomes the limitation in the approaches of Raya et al. [9] and Lu et al. [23]. Nevertheless, in the approach of Zhang et al. [25], a long-term system master secret s is embedded in the vehicle's TPD, which could be extracted by an adversary (e.g. via side-channel attacks [47]), particularly when the adversary has physical access to the TPD.

In 2009, Jiang et al. [22] presented an authentication protocol based on the binary authentication tree (BAT), in which the RSU could quickly differentiate the fabricated messages from the legitimate ones. However, Shim [12] demonstrated that an adversary can successfully forge an aggregate signature on two bogus messages in the scheme of Jiang et al. [22]. Shim [3] also introduced a CPPA solution using Pseudo-Identity (PID)-based signature for secure VANETs. Liu et al. [48], however, revealed that Shim's solution in [3] has an error in the batch verification stage. In 2013, Li and Liu introduced a lightweight identity authentication scheme for VANETs to improve the efficiency of the authentication process while concealing the sensitive information of the vehicle simultaneously [49]. Then, Lee and Lai proposed a secure batch verification protocol with group testing for VANETs [50]. In 2015, He et al. [18] proposed an ID-based CPPA solution for VANETs utilizing Schnorr's signature [51]. In He et al.'s solution, the system's private key is preloaded on the vehicle's TPD. In other words, the proposed solution suffers from the same limitation as the solution of Zhang et al. [25]. In 2016, Oulhaci et al. also designed a secure and distributed certification system framework for security message authentication in VANETs, which is against fake public-key certification [52]. In the same year, Lee et al. use the Chinese remainder theorem to design a safer and quicker batch key-agreement protocol for establishing communication channels [53]. More recently in 2016 and 2017, Shao et al. [19] and Azees et al. [20] introduced a group signature-based CPPA solution for VANETs and an authentication solution based on short-time anonymous certificates and public keys, respectively. The proposed solution of Azees et al. [20] does not support batch verification. In addition, the adversary against Azees et al.'s protocol cannot resist bogus message attack, framing attack and sybil attack. The reason of suffering from the above attacks is because the authors use a temporarily generated number as the private key to sign traffic message, which is a invalid signature and easily counterfeited by adversary. In 2017, Zhang et al. [26] gave a new distributed aggregate privacy-preserving authentication protocol for vehicular ad hoc networks. In their protocol, one RSU is responsible for a subgroup of VANETs and holds a private key used to produce secret shares for vehicles. Although, they give some assumptions guaranteeing that no other items can learn the secrets in a vehicle's TPD, if a vehicle is corrupted in one RSU, the private key of the RSU would be calculated by the malicious adversary. Later, Zhang et al. gives a novel method to establish cryptographic mix-zones which resist malicious attackers and reinforce privacy protection in VANETs [54]. In 2018, Asaar et al. proposed a novel ID-based message authentication protocol via proxy vehicles (ID-MAP) [55], which cannot preserve the security of master key either.

3.2 Review and Analysis of Azees et al.'s CPPA Protocol

3.2.1 Azees et al.'s CPPA Protocol: A Revisit

We will now briefly review Azees et al.'s CPPA protocol [20]. The protocol has two anonymous authentication procedures, namely, the authentication scheme for the vehicle and the authentication scheme for the RSU. The anonymous authentication procedure for an RSU is similar to that for a vehicle; thus, we will only review the authentication procedure for a vehicle. This procedure has six sub-stages, namely, system initialization stage, vehicle registration and key generation stage, anonymous certificate generation of vehicle, vehicle signature generation, verification stage, and traceability stage.

System Initialization: Using the bilinear parameters (G_1, G_2, G_3, e, q), the TA computes the system parameters as shown below. TA chooses two random numbers $a, b \in Z_q^*$ as the master keys, generates $A_1 = g_1^a$ and $B_1 = g_1^b$, and chooses a cryptographic hash function $\hbar : \{0,1\}^* \to Z_q^*$. Finally, TA sets the system parameters $param = (q, e, g_1, g_2, G_1, G_2, G_3, A_1, B_1, H)$ public.

Vehicle Registration and Key Generation: In the registration stage, the vehicle needs to provide relevant user information, such as name, license plate number, address, and contact number to the TA. In the key generation stage, TA obtains the vehicle user u_i's original identity OID_{u_i}. Then, TA computes dummy identities DID_{u_i} for u_i. To compute u_i's dummy identity, the TA selects a nonce $n_i \in Z_q^*$ and computes $DID_{u_i} = g_1^{n_i+a}$. Then, the TA chooses a nonce $v_i \in Z_q^*$, generates $T_i = g_1^{\overline{\frac{1}{v_i+a+b}}}$ and stores $(OID_{u_i}, DID_{u_i}, T_i^b)$ corresponding to u_i in the database of the tracking list. The TA returns the authorization key $AK = (DID_{u_i}, T_i, E_i)$ to u_i in an offline manner (e.g. a smart card), where $E_i = g_1^{-n_i}$. Upon receiving AK from the TA, u_i stores it in the vehicle's TPD.

Anonymous Certificate Generation of Vehicle: Once u_i participates in the system, u_i generates the anonymous certificates using AK as shown below.

- u_i chooses a range of random numbers $r_1, r_2, \cdots, r_l \in Z_n^*$, $l \leq n$ and computes $Y_k = g_2^{r_k}$ for $k = 1, 2, \cdots, l$.

- u_i randomly chooses $\mu, k_1, k_2 \in Z_q^*$ and computes $\gamma_u = B_1^{\mu}$, $\gamma_v = T_i \cdot A_1^{\mu}$, $\lambda = (\mu + r_k) \bmod q$, $\lambda_1 = \gamma_u^{\mu+k_1}$, $\lambda_2 = \frac{\gamma_u^{\mu+k_1}}{\gamma_v^{\mu+k_2}}$. After computing $\gamma_u, \gamma_v, \lambda, \lambda_1, \lambda_2$, u_i computes the challenger $c = \hbar(DID_{u_i}||A_1||B_1||E_i||\gamma_u||\gamma_v||Y_k||\lambda_1||\lambda_2)$, $\sigma_1 = (r_k - k_1) \bmod q$, and $\sigma_2 = (r_k - k_2) \bmod q$.

- Finally, the user generates $Cert_k = \{Y_k||E_i||DID_{u_i}||\gamma_u||\gamma_v||c||a||\sigma_1||\sigma_2\}$ as the anonymous certificate.

Remark-1: The product of E_i and DID_{u_i} for every vehicle user is constant, namely, $E_i \times DID_{u_i} = g_1^{-n_i} \times g_1^{n_i+a} = g_1^a = A_1$ and A_1 (as well as B_1) is a public parameter generated in the system initialization stage. The generations of parameters $\{Y_k, \lambda_1, \lambda_2, \gamma_u, \gamma_v, \lambda, \sigma_1, \sigma_2, c, Cert_k\}$ are based on the randomly selected numbers $\{r_k, \mu, k_1, k_2\} \in Z_q^*$, whereby the challenger $c = \hbar(DID_{u_i}||A_1||B_1||E_i||\gamma_u||\gamma_v||Y_k||\lambda_1||\lambda_2)$, and $Cert_k$ is the set of $\{Y_k||E_i||DID_{u_i}||\gamma_u||\gamma_v||c||\lambda||\sigma_1||\sigma_2\}$. Thus, a malicious adversary $\mathcal{A}$ can also randomly choose new $\{r_A, \mu', k_1', k_2'\} \in Z_q^*$ and produce fabricated $\{Y_A, \lambda_1', \lambda_2', \gamma_u', \gamma_v', \lambda', \sigma_1', \sigma_2', c_A, Cert_A\}$.

Vehicle Signature Generation: To verify the integrity of message M, u_i generates the signature $sig = g_1^{\frac{1}{r_k + \hbar(M)}}$ and transmits $msg = (M||sig||Y_k||Cert_k)$ to nearby vehicles and RSUs.

Remark-2: The authors only use the ephemeral value r_k to sign the message, where r_k is also used in the challenger c and certificate $Cert_k$. Thus, $\mathcal{A}$ can easily fabricate $sig_A = g_1^{\frac{1}{r_A + \hbar(M_A)}}$, where M_A is the message that $\mathcal{A}$ wishes to broadcast and r_A is the one used in Remark 1.

Verification: Upon receiving $msg = (M||sig||Y_k||Cert_k)$, the receiver will verify the challenger c and the integrity of message in the following steps.

1. The receiver computes

$$N_i = E_i \times DID_{u_i} = g_1^{-n_i} \times g_1^{n_i+a} = g_1^a = A_1,$$

$$\lambda_1' = \frac{\gamma_u^{\lambda}}{\gamma_u^{\sigma_1}} = \frac{\gamma_u^{\mu+r_k}}{\gamma_u^{r_k-k_1}} = \gamma_u^{\mu+r_k-r_k+k_1} = \gamma_u^{\mu+k_1} = \lambda_1,$$

$$\lambda_2' = \frac{\gamma_u^{\lambda} \cdot \gamma_v^{\sigma_2}}{\gamma_u^{\sigma_1} \cdot \gamma_v^{\lambda}} = \frac{\gamma_u^{\mu+r_k} \cdot \gamma_v^{r_k-k_2}}{\gamma_u^{r_k-k_1} \cdot \gamma_v^{\mu+r_k}}$$

$$= \frac{\gamma_u^{\mu+r_k-r_k+k_1}}{\gamma_v^{\mu+r_k-r_k+k_2}} = \frac{\gamma_u^{\mu+k_1}}{\gamma_v^{\mu+k_2}} = \lambda_2.$$

Then, the challenger computes $c = \hbar(DID_{u_i}||N_i||B_1||E_i||\gamma_u||\gamma_v||Y_k||\lambda_1'||\lambda_2')$ and inspects if c is equal to c'. If the verification is successful, then the receiver verifies the sender and accepts $\{Y_k||Cert_k\}$; otherwise, the receiver interrupts the session.

2. The receiver authenticates the integrity of traffic message by checking the correctness of $e(sig, Y_k \cdot g_2^{\hbar(M)}) = e(g_1, g_2)$. Note that $e(sig, Y_k \cdot g_2^{\hbar(M)}) = e(g_1^{\frac{1}{r_k+\hbar(M)}}, g_2^{r_k} \cdot g_2^{\hbar(M)}) = e(g_1^{\frac{1}{r_k+\hbar(M)}}, g_2^{r_k+\hbar(M)}) = e(g_1, g_2)^{\frac{1}{r_k+\hbar(M)} \cdot r_k+\hbar(M)} = e(g_1, g_2)$. Therefore, the receiver accepts msg.

Remark-3: Although all the equations are correct and $\{A_1, \lambda_1, \lambda_2\}$ can be revealed by a simple computation, the verification is weak since the parameters $\{Y_k, \lambda_1, \lambda_2, \gamma_u, \gamma_v, \lambda, \sigma_1, \sigma_2, c, Cert_k\}$ are produced based on randomly selected numbers, and thus, there is no reliable public parameter used in the verification that is generated by the TA and one that cannot be fabricated by $\mathcal{A}$. In other words, $\mathcal{A}$ can also produce these parameters and successfully pass the required verification (see Section 3.2.2).

Traceability: Once a malicious vehicle transmits a fabricated or modified message to mislead others, the TA can utilize its anonymous certificate $Cert_k = \{Y_k||E_i||DID_{u_i}||\gamma_u||\gamma_v||c||\lambda||\sigma_1||\sigma_2\}$ to compute the value as follows.

$$\frac{\gamma_v^b}{\gamma_u^a} = \frac{(T_i \cdot A_1^{\mu})^b}{(B_1^{\mu})^a} = \frac{T_i^b \cdot (A_1^{\mu})^b}{(B_1^{\mu}a)} = \frac{T_i^b \cdot g_1^{a\mu b}}{g_1^{\mu ab}} = T_i^b$$

Thus, the TA will know the vehicle's identity by matching the value T_i^b in the tracking list and proceed to remove the malicious vehicle from the network/system.

3.2.2 Security Flaws

According to Remarks 1-3 in the preceding section (see Section 3.2.1), we will now explain how the protocol is not secure against the four attacks described in Sections 3.2.2.1 to 3.2.2.3 as below.

3.2.2.1 Bogus Message Attack

$\mathcal{A}$ can transmit a fabricated message by the following steps.

1. $\mathcal{A}$ captures a signed message $msg = (M||sig||Y_i||Cert_i)$ sent by some vehicle user u_i, where $Cert_i = \{Y_i||E_i||DID_{u_i}||\gamma_u||\gamma_v||c||\lambda||\sigma_1||\sigma_2\}$ is the anonymous certificate of u_i and $\{DID_{u_i}, E_i\}$ will be used in step 2. Then, $\mathcal{A}$ randomly chooses a number r_A and computes $Y_A = g_2^{r_A}$, where g_2 is the generator of group G_2.

2. For the corresponding short time certificate $Cert_A$, $\mathcal{A}$ randomly chooses $\gamma_{u'}$, $\gamma_{v'}$, μ', $k'_1, k'_2 \in Z_q^*$ and computes $\lambda' = (\mu' + r_A) \bmod q$, $\lambda'_1 = {\gamma_{u'}}^{\mu'+k'_1}$ and $\lambda'_2 = \frac{{\gamma_{u'}}^{\mu'+k'_1}}{{\gamma_{v'}}^{\mu'+k'_2}}$. Then, $\mathcal{A}$ generates the forged challenger $c_A = \hbar(DID_{u_i}||A_1||B_1||E_i||\gamma_{u'}||\gamma_{v'}||Y_A||\lambda'_1||\lambda'_2)$, $\sigma'_1 = (r_A - k'_1)$, $\sigma'_2 = (r_A - k'_2)$ as well as the corresponding certificate $Cert_A = \{Y_A||E_i||DID_{u_i}||\gamma'_u||\gamma'_v||c_A||\lambda'||\sigma'_1||\sigma'_2\}$, whereby $\{DID_{u_i}, E_i\}$ are involved in the captured $Cert_i = \{Y_i||E_i||DID_{u_i}||\gamma_u||\gamma_v||c||\lambda||\sigma_1||\sigma_2\}$.

   Remark-1′: As discussed in **Remark-1**, since the generations of parameters $\{Y_k, \lambda_1, \lambda_2, \gamma_u, \gamma_v, \lambda, \sigma_1, \sigma_2, c, Cert_k\}$ are based on the randomly selected numbers $\{r_k, \mu, k_1, k_2\} \in Z_q^*$, where the challenger $c = \hbar(DID_{u_i}||A_1||B_1||E_i||\gamma_u||\gamma_v||Y_k||\lambda_1||\lambda_2)$, and $Cert_k$ is the set of $\{Y_k||E_i||DID_{u_i}||\gamma_u||\gamma_v||c||\lambda||\sigma_1||\sigma_2\}$, $\mathcal{A}$ can also randomly choose new $\{r_A, \mu', k'_1, k'_2\} \in Z_q^*$ and produce fabricated $\{Y_A, \lambda'_1, \lambda'_2, \gamma'_u, \gamma'_v, \lambda', \sigma'_1, \sigma'_2, c_A, Cert_A\}$. Additionally, since the generation of $\gamma_u = B_1^{\mu}$ and $\gamma_v = T_i \cdot A_1^{\mu}$ are both based on μ and there is no detection on γ_u and γ_v in the stage of verification, $\mathcal{A}$ only needs to choose random $\gamma_{u'}$ and $\gamma_{v'}$ instead of computing them by μ', which decreases the computation cost.

3. $\mathcal{A}$ forges a message M_A, generates the corresponding signature $sig_A = g_1^{\frac{1}{r_A+\hbar(M_A)}}$, and then sends the message $msg_A = (M_A||sig_A||Y_A||Cert_A)$ to nearby RSUs or vehicles.

   Remark-2′: As discussed in **Remark-2**, $\mathcal{A}$ can successfully fabricate $sig_A = g_1^{\frac{1}{r_A+\hbar(M_A)}}$, where M_A is the message that $\mathcal{A}$ wishes to broadcast and r_A is the one selected in step 1.

4. After receiving $msg_A = (M_A||sig_A||Y_A||Cert_A)$ from $\mathcal{A}$, the receiver computes

$$N_i = E_i \times DID_{u_i} = A_1,$$

$$\lambda_1'' = \frac{\gamma_{u'}^{\lambda'}}{\gamma_{u'}^{\sigma_1}} = \frac{\gamma_{u'}^{\mu'+r_A}}{\gamma_{u'}^{r_A-k_1'}} = \gamma_{u'}^{\mu'+r_A-r_A+k_1'} = \gamma_{u'}^{\mu'+k_1'} = \lambda_1',$$

$$\lambda_2'' = \frac{\gamma_{u'}^{\lambda'} \cdot \gamma_{v'}^{\sigma_2'}}{\gamma_{u'}^{\sigma_1'} \cdot \gamma_{v'}^{\lambda'}} = \frac{\gamma_{u'}^{\mu'+r_A} \cdot \gamma_{v'}^{r_A-k_2'}}{\gamma_{u'}^{r_A-k_1'} \cdot \gamma_{v'}^{\mu'+r_A}}$$

$$= \frac{\gamma_{u'}^{\mu'+r_A-r_A+k_1'}}{\gamma_{v'}^{\mu'+r_A-r_A+k_2}} = \frac{\gamma_{u'}^{\mu'+k_1'}}{\gamma_{v'}^{\mu'+k_2}} = \lambda_2'.$$

   Then, the challenger computes $c_A' = \hbar(DID_{u_i}||N_i||B_1||E_i||\gamma_{u'}||\gamma_{v'}||Y_A||\lambda_1''||\lambda_2'')$; thus, it is trivial to note that $c_A' = c_A$.

5. Finally, the receiver authenticates the integrity of traffic message by checking the correctness of $e(sig, Y_A \cdot g_2^{\hbar(M_A)}) = e(g_1, g_2)$. Clearly, $e(sig, Y_A \cdot g_2^{\hbar(M_A)}) = e(g_1^{\frac{1}{r_A+\hbar(M_A)}}, g_2^{r_A} \cdot g_2^{\hbar(M_A)}) = e(g_1^{\frac{1}{r_A+\hbar(M_A)}}, g_2^{r_A+\hbar(M_A)}) = e(g_1, g_2)^{\frac{1}{r_A+\hbar(M_A)} \cdot r_A+\hbar(M_A)} = e(g_1, g_2)$. Hence, the receiver accepts the forged message msg_A.

   Remark-3′: As discussed in **Remark-3**, since the equations required for the recovery (i.e., $N_i = A_1$, $\lambda_1' = \lambda_1$, and $\lambda_2' = \lambda_2$) and verification (i.e., c_A and $e(sig, Y_A \cdot g_2^{\hbar(M_A)}) = e(g_1, g_2)$) are correct and the parameters $\{Y_k, \lambda_1, \lambda_2, \gamma_u, \gamma_v, \lambda, \sigma_1, \sigma_2, c, Cert_k\}$ are produced based on randomly selected numbers, $\mathcal{A}$'s fabricated $\{Y_A, \lambda_1', \lambda_2', \gamma_u', \gamma_v', \lambda', \sigma_1', \sigma_2', c_A, Cert_A\}$ can also successfully pass all verification steps.

6. If the TA determines that M_A is a forged message, it will recover the ID_{u_i} by computing $\frac{\gamma_{v'}^{b}}{\gamma_{u'}^{a}}$. However, the latter is a nonce, and hence, it is not useful. Therefore, this protocol does not provide message authentication or traceability.

3.2.2.2 Framing Attack

1. $\mathcal{A}$ executes step 1 in Section 3.2.2.1, computes $\lambda_1 = \frac{\gamma_u^{\lambda}}{\gamma_u^{\sigma_1}}$, $\lambda_2 = \frac{\gamma_u^{\lambda} \cdot \gamma_v^{\sigma_2}}{\gamma_u^{\sigma_1} \cdot \gamma_v^{\lambda}}$ and generates the forged challenger $c_A = \hbar(DID_{u_i}||A_1||B_1||E_i||\gamma_u||\gamma_v||Y_A||\lambda_1||\lambda_2)$, as well as the corresponding certificate $Cert_A = \{Y_A||E_i||DID_{u_i}||\gamma_u||\gamma_v||c_A||\lambda||\sigma_1||\sigma_2\}$.

2. $\mathcal{A}$ forges M_A, generates a corresponding signature $sig_A = g_1^{\frac{1}{r_A+\hbar(M_A)}}$, and sends $msg_A = (M_A||sig_A||Y_A||Cert_A)$ to nearby RSUs or vehicles.

3. After receiving $msg_A = (M_A||sig_A||Y_A||Cert_A)$ from $\mathcal{A}$, the verification of $\{c_A||Y_A||Cert_A\}$ will show that it is valid, since there is no detection procedure for the modification on Y_i and $\mathcal{A}$ does not modify u_i's parameters $\{E_i||DID_{u_i}||\gamma_u||\gamma_v||\lambda||\sigma_1||\sigma_2\}$. Then, the receiver will verify the integrity of the message by checking the correctness of $e(sig.Y_A \cdot g_2^{\hbar(M_A)}) = e(g_1, g_2)$.

4. If the TA later determines that M_A is forged, it will recover the identity ID_{u_i} by computing $\frac{\gamma_v^b}{\gamma_u^a} = T_i^b$ to be the identity for the originator of the forged message. However, u_i is not the true attacker, and hence, u_i has been the victim of a framing attack.

Remark-4′: The difference between the framing attack and bogus message attack lies in the choice of (γ_u, γ_v). If $\mathcal{A}$ uses (γ_u, γ_v) in the captured message $Cert_i = \{Y_i||E_i||DID_{u_i}||\gamma_u||\gamma_v||c||\lambda||\sigma_1||\sigma_2\}$ of vehicle user u_i and broadcasts malicious information, then the TA would recover the adversary as the innocent u_i. If $\mathcal{A}$ uses the pair of randomly selected ($\gamma_{u'}$, $\gamma_{v'}$), then no user will be detected (see step 6 in Section 3.2.2.1).

3.2.2.3 Sybil Attack and Replay Attack

As described in Sections 3.2.2.1 and 3.2.2.2, it is possible for $\mathcal{A}$ to carry out a Sybil attack against the Azees et al.'s protocol [20]. In addition, the protocol is not resilient to the replay attack because the protocol does not detect and guarantee message freshness. Since the replayed messages could successfully pass the verification procedure, Sybil attacks with a replay attack can be carried out and this can result in real consequences.

3.2.2.4 Lack of batch authentication stage

Although the authors in [20] evaluated the computational cost in the batch authentication of multiple messages, it is only a single message verification. We observe that the protocol lacks a specific and efficient batch authentication process, and the importance of the batch authentication in secure communication is also explained in [56].

3.3 Review and Analysis of Zhang et al.'s CPPA Protocol

3.3.1 Zhang et al.'s CPPA Protocol: A Revisit

There are mainly five phases in Zhang et al.'s CPPA protocol [26], which are described as follows.

3.3.1.1 System Setup

In this phase, the root trusted authority (TA) executes the below steps to initialize the system parameters.

1. The root TA generates a bilinear map: $\hat{e} : G_1 \times G_2 \rightarrow G_3$, where G_1, G_2, G_3 are cyclic groups with prime order q, g_1 and g_2 are generators of G_1 and G_2 separately.

2. The root TA picks $a, b \in Z_q^*$ as its master secrets, and computes $y = g_2^a$, $e = g_1^b$ as its master public keys. a is utilized to launch certificates for RSUs and b is utilized to set up a secure channel between TA and an RSU or a vehicle

3. The root TA selects $E_\pi(.)/D_\pi(.)$ and hash functions $H_0(\cdot) : \{0,1\}^* \rightarrow G_1$, $H_1(\cdot) : \{0,1\}^* \rightarrow Z_q^*$, $H_{2_{key}}(\cdot) : \{0,1\}^* \rightarrow \{0,1\}^l$, $H_3(\cdot) : \{0,1\}^* \rightarrow \Gamma$ and $H_4(\cdot) : \{0,1\}^* \rightarrow \{0,1\}^{l'}$, where $H_{2_{key}}(\cdot)$ is a keyed hash, the key space of key is $\{0,1\}^*$, and Γ is the key space of π. Selects Λ from the key space of key.

4. The root TA keeps a, b, Λ secretly and preloads $Params = \{\hat{e}, q, G_1, G_2, G_3, g_1, g_2, H_0(\cdot), H_1(\cdot), H_{2_{key}}(\cdot), H_3(\cdot), E_\pi(.)/D_\pi(.)\}$.

3.3.1.2 RSU Setup

In this phase, every roadside unit (RSU) R_j's private-public key pair and certificate are generated.

1. R_j picks $a_j, b_j \in Z_q^*$ and computes $y_j = g_2^{a_j}$, $e_j = g_1^{b_j}$, whereby (a_j, b_j) are R_j's private keys and (y_j, e_j) are R_j's public keys. a_j is responsible for producing shares for the vehicles, and b_j is responsible for establishing a secure channel between R_j and a vehicle.

2. R_j submits (y_j, e_j) and its identifying information (e.g. R_j's validity period of its public key and location information) to the root TA by a secure channel. TA issues the short-term certificate $Cert_{R_j} = (ID_{R_j}, (y_j, e_j), Sig_j)$ to R_j, where Sig_j is a signature on $(ID_{R_j}, (y_j, e_j))$. $Cert_{R_j}$ is broadcast within R_j's communication range.

3.3.1.3 Vehicle Setup

The vehicle V_i is supposed to be initialized before joining a VANET.

1. The root TA computes V_i's PID $PID_{V_i} = H_{2_\Lambda}(ID_{V_i}||VP_i)$ and selects an authentication key λ_i, where ID_{V_i} is V_i's genuine identity and VP_i is the validity period.
2. TA stores the $\{Params, PID_{V_i}, \lambda_i\}$ into V_i's TPD and adds $\{ID_{V_i}, VP_I, PID_{V_i}, \lambda_i\}$ to member list ML.

3.3.1.4 Member Secrets Generation

In this phase, a vehicle will obtain the member secrets from its nearest RSU as following steps.

1. V_i firstly verifies the validity of $Cert_{R_j}$ from R_j. If Sig_j in $Cert_{R_j}$ is not valid under the master public key y, it terminates; elsewise, V_i extracts the identity ID_{R_j} and the public keys (y_j, e_j) from $Cert_{R_j}$. V_i selects a random value $\theta \in Z_q^*$, computes $f = g_1^\theta$, $\pi_{i1} = H_3(f, e_j, e_j^\theta, ID_{R_j}, T_t)$, and $\pi_{i2} = H_3(f, e, e^\theta, ID_{R_j}, T_t)$, where T_t is the current time-stamp. V_i computes $\varrho_j = E_{i2}(\lambda_i, T_t)$ and transmits $(f, ID_{R_j}, \varrho_j, T_t)$ to R_j.
2. After receiving $(f, ID_{R_j}, \varrho_j, T_t)$ form V_i, R_j firstly checks whether T_t is fresh or not. If T_t is fresh, R_j forwards $(f, ID_{R_j}, \varrho_j, T_t)$ to TA via a secure channel, elsewise, it terminates.
3. Upon receiving $(f, ID_{R_j}, \varrho_j, T_t)$ from R_j, TA computes $\pi_{i2} = H_3(f, e, f^b, ID_{R_j}, T_t)$ and $D_{\pi_{i2}}(\varrho_j) = (\lambda_i', T_i')$. If λ_i' does not exist in the tuple $\{ID_{V_i}, VP_I, PID_{V_i}, \lambda_i\}$ of ML like $\lambda_i \neq \lambda_i'$ or $T_t \neq T_t'$ or VP_i is expired, it terminates; elsewise, it issues 1 to R_j via the secure channel.
4. Once receiving 1 from TA, R_j computes $\pi_{i1} = H_3(f, e_j, f^{b_j}, ID_{R_j}, T_t)$ and selects an authorized period T_p and two member secrets (α_j, β_j) satisfying $a_j = \alpha_j \cdot \beta_j$. Then, it continues computing $\hbar_{R_j} = H_{2_{\pi_{i1}}}(T_p, \alpha_j, \beta_j)$ and $\varrho' = E_{\pi_{i1}}(T_p, \alpha_j, \beta_j, \hbar_{R_j})$ and broadcasts $(H_4(f), \varrho_j')$.
5. Upon receiving $(H_4(f), \varrho_j')$, V_i computes $D_{\pi_{i1}}(\varrho_j') = (T_p, \alpha_j, \beta_j, \hbar_{R_j})$ and then authenticates if $\hbar_{R_j} = H_{2_{\pi_{i1}}}(T_p, \alpha_j, \beta_j)$. If it is correct, it sets the member secrets and the authorized period in the TPD to be (α_j, β_j) and T_p; elsewise,

it terminates.

3.3.1.5 Vehicle Signature

1. V_i generates a public pseudo-identity $PPID_{i,t} = H_4(PID_{V_i}, T_t)$, where PID_{V_i} is its pseudo-identity and T_t is the time-stamp.
2. V_i computes $pid_{i,t,0} = H_0(PPID_{i,t}, 0)$, $pid_{i,t,1} = H_0(PPID_{i,t}, 1)$, $s'_{i,t,0} = pid^{\alpha_j}_{i,t,0}$, $s_{i,t,0} = s'^{\beta_j}_{i,t,0}$, $s'_{i,t,1} = pid^{\alpha_j}_{i,t,1}$, $s_{i,t,1} = s'^{\beta_j}_{i,t,1}$, sets $s_{i,t} = (s_{i,t,0}, s_{i,t,1})$ as the one-time signature key of V_i.
3. V_i computes $\sigma_{i,t} = s_{i,t,0}s^{h_i}_{i,t,1}$ as the signature, where $h_i = H_1(M_i, PPID_{i,t}, Cert_{R_j})$, and broadcast $(M_i, PPID_{i,t}, \sigma_{i,t})$.
4. Finally, to make the member secrets stored in the TPD update locally [57], choose a random $r \in Z^*_q$, sets $\alpha_j = r\alpha_j$ and $\beta_j = r^{-1}\beta_j$ and set ($\alpha_j = \alpha'_j$, $\beta_j = \beta'_j$) as the new secret values.

3.3.1.6 Batch Message Verification

Upon receiving multiple messages $\{m_1, PPID_{i,j_1}, \sigma_1\}$, $\{m_2, PPID_{i,j_2}, \sigma_2\}$, . . ., $\{m_n, PPID_{i,j_n}, \sigma_n\}$ sent by vehicles of the same/neighboring groups, the verifier verify the validity of those messages via the below steps.

1. The verifier divides the public pseudo-identities into l sets $S_1 = \{PPID_{1,j_1}, \ldots, PPID_{t_1,j_{t_1}}\}$, $S_2 = \{PPID_{t_1+1,j_{t_1+1}}, \cdots, PPID_{t_2,j_{t_2}}\}$, . . ., $S_1 = \{PPID_{t_{l-1}+1,j_{t_{l-1}+1}}, \cdots, PPID_{n,j_n}\}$.
2. The verifier computes the aggregate signature $\Omega = \Pi^n_{i=1}\sigma_i$.
3. The verifier computes $h_i = H_1(M_i, PPID_{i,t}, cert_{R_k})$, $pid_{i,0} = H_0(PPID_{i,j_i}, 0)$ and $pid_{i,1} = H_0(PPID_{i,j_i}, 1)$ for $PPID_{i,j_i} \in S_k$.
4. The verifier checks whether $\hat{e}(\Omega, g_2) = \prod^l_{j=1} \hat{e}(\prod_{i \in S'_j} id_{i,0}id^{h_i}_{i,1}, y_j)$. If it is correct, outputs 1; elsewise outputs 0.

3.3.2 Security Flaws

In this subsection, we will propose three vulnerabilities against Zhang et al.'s solution, which are presented as below.

3.3.2.1 Leakage of RSU's private key

Assuming the adversary $\mathcal{A}$ has registered a vehicle in R_j with $\{Params, PID_{V_i}, \lambda_i\}$ stored in its TPD. According to step 5 of 3.3.1.4, after receiving $(T_p, \alpha_j, \beta_j, \hbar_{R_j})$ successfully, $\mathcal{A}$ could compute R_j's the private key $a_j = \alpha_j \cdot \beta_j$, which is based on the step 4 of 3.3.1.4.

3.3.2.2 Forged Message Attack

An attacker $\mathcal{A}$ can transmit a forged message by the following steps.

1. $\mathcal{A}$ chooses a random number as $\overline{PPID_{i,t}}$ and two new member secrets $(\overline{\alpha_j}, \overline{\beta_j})$, where r is random value, $\overline{\alpha_j} = r\alpha_j$ and $\overline{\beta_j} = r^{-1}\beta_j$.
2. $\mathcal{A}$ computes $\overline{pid_{i,t,0}} = H_0(\overline{PPID_{i,t}}, 0)$, $\overline{pid_{i,t,1}} = H_0(\overline{PPID_{i,t}}, 1)$, $\overline{s'_{i,t,0}} = \overline{pid_{i,t,0}}^{\overline{\alpha_j}}$, $\overline{s_{i,t,0}} = \overline{s'_{i,t,0}}^{\overline{\beta_j}}$, $\overline{s'_{i,t,1}} = \overline{pid_{i,t,1}}^{\overline{\alpha_j}}$, $\overline{s_{i,t,1}} = \overline{s'_{i,t,1}}^{\overline{\beta_j}}$, sets $\overline{s_{i,t}} = (\overline{s_{i,t,0}}, \overline{s_{i,t,1}})$ as the one-time signature key of $\mathcal{A}$.
3. $\mathcal{A}$ computes $\overline{\sigma_{i,t}} = \overline{s_{i,t,0}} \cdot \overline{s_{i,t,1}}^{\overline{h_i}}$ as the signature, where $\overline{h_i} = H_1(M_A, \overline{PPID_{i,t}}, Cert_{R_j})$, and broadcast $(M_A, \overline{PPID_{i,t}}, \overline{\sigma_{i,t}})$.
4. Upon receiving the message $(M_A, \overline{PPID_{i,t}}, \overline{\sigma_{i,t}})$, the verifier finds out $\mathcal{A}$'s corresponding $Cert_{R_j}$ and computes $\overline{h_i} = H_1(M_A, \overline{PPID_{i,t}}, Cert_{R_j})$, $\overline{pid_{i,0}} = H_0(\overline{PPID_{i,t}}, 0)$ and $\overline{pid_{i,1}} = H_0(\overline{PPID_{i,t}}, 1)$.
5. The verifier checks whether $\hat{e}(\overline{\sigma_{i,t}}, g_2) = \hat{e}(\overline{pid_{i,0}} \cdot \overline{pid_{i,1}}^{\overline{h_i}}, y_j)$. We can verify the correctness as below.

$$
\begin{aligned}
\hat{e}(\overline{\sigma_{i,t}}, g_2) =& \hat{e}(\overline{s_{i,t,0}} \cdot \overline{s_{i,t,1}}^{\overline{h_i}}, g_2) \\
=& \hat{e}(\overline{s'_{i,t,0}}^{\overline{\beta_j}} \cdot \overline{s'_{i,t,1}}^{\overline{\beta_j} \cdot \overline{h_i}}, g_2)
\end{aligned}
$$

$$
\begin{aligned}
&=\hat{e}(\overline{pid_{i,t,0}}^{\overline{\alpha_j}\cdot\overline{\beta_j}} \cdot \overline{pid_{i,t,1}}^{\overline{\alpha_j}\cdot\overline{\beta_j}\cdot\overline{h_i}}, g_2)\\
&=\hat{e}(\overline{pid_{i,t,0}}^{a_j} \cdot \overline{pid_{i,t,1}}^{a_j\cdot\overline{h_i}}, g_2)\\
&=\hat{e}(\overline{pid_{i,t,0}} \cdot \overline{pid_{i,t,1}}^{\overline{h_i}}, g_2^{a_j})\\
&=\hat{e}(\overline{pid_{i,t,0}} \cdot \overline{pid_{i,t,1}}^{\overline{h_i}}, y_j)
\end{aligned}
$$

3.3.2.3 Impersonation Attack

$\mathcal{A}$ can forge a message with one intercepted user's pseudo-identity $PPID_{i,t}$ through the following steps.

1. $\mathcal{A}$ intercepts the transmitted message $\{M_i, PPID_{i,j}, \sigma_i\}$ and sets $\overline{PPID_{i,t}} = PPID_{i,j}$.
2. $\mathcal{A}$ extracts the T_t from M_i, which is used to forge a new message M_A, and then executes the other steps as those in 5.2.2.
3. If the TA later detects that M_A is a fake message, it will find out the real identity ID_{V_i} by checking whether $PPID_{i,t} = H_4(PID_{V_i}, T_t)$, where PID_{V_i} is in the tuple $\{ID_{V_i}, VP_I, PID_{V_i}, \lambda_i\}$ on ML. Hence, the adversary $\mathcal{A}$ could mount the framing attack on the honest V_i.

3.4 Summary

In this chapter, we reviewed the existing studies on CPPA protocols. In particular, we studied the anonymous CPPA protocols of Azees et al. [20] and Zhang et al [26]. designed for VANETs, respectively. We revealed previously unknown attacks against them, and more importantly identified design flaws in their protocol. Specifically, in Azees et al.'s protocol, randomly-selected numbers are used to produce all other parameters without binding these numbers to an identity. In addition, there is no reliable public verification. Hence, an attacker can easily exploit these design flaws to carry the four attacks we showed in this chapter.

Zhang et al.'s protocol is not against the attacks from malicious adversary, since the RSU sends its private signature key a_j to vehicles with an easy variation.

Chapter 4

CPS-CPPA: Certificateless and Provably-Secure Conditional Privacy-Preserving Authentication Protocol

4.1 Motivation

To resolve the vulnerabilities of existing schemes that are retrospected and analyzed, especially, the flaws in the protocols of Azees et al. [20] and Zhang et al. [26]. This chapter presents a CPS-CPPA protocol for VANETs with the function of batch verification [27].

4.2 The CPS-CPPA Protocol

The proposed CPS-CPPA protocol [27] consists of two parts, namely: an anonymous CPS-CPPA solution for the vehicle and an ID-based CPS-CPPA solution for the RSU. For each part, there are five stages, i.e., system parameters setup stage, enrollment stage, message signing stage, single message verification, and batch messages verification.

Table 4.1: The working flow of anonymous CPS-CPPA protocol for vehicle

Stages	TA	Vehicle V_i	RSU_j
System Initialization	TA selects two master keys $a, b \in Z_q^*$ TA computes $A_{pub} = g^a$ and $B_{pub} = g^b$ as public keys TA generates and broadcasts system parameters $Params = \{q, G, g, A_{pub}, B_{pub}, H_0, H_1, H_2, H_3\}$		
Vehicle Enrollment	TA $\xleftarrow{\{ID_i\},\ offline}$ V_i TA selects random numbers $\{k_{i,1}, \cdots, k_{i,z}\} \in Z_q^*$ TA computes $PK_{i,l} = g^{k_{i,l}}$, $l \in \{1, \cdots, z\}$ TA computes $PID_{i,l} = ID_i \oplus H_0(PK_{i,l}^b, B_{pub})$, $l \in \{1, \cdots, z\}$ TA computes $sk_{i,l} = a \cdot H_1(PID_{i,l})$, $l \in \{1, \cdots, z\}$. TA $\xrightarrow{\{Params, PID_i^*, SK_i^*, PK_i^*\},\ offline}$ V_i V_i stores $\{Params, PID_i^*, SK_i^*, PK_i^*\}$ into its TPD		
Signature Generation		V_i selects $PID_{i,l}$, $sk_{i,l}$, $PK_{i,l}$ V_i randomly chooses $r_i \in Z_q^*$ V_i computes $R_i = g^{r_i}$ V_i generates the current timestamp T_i V_i computes $h_i = H_2(M_i, PID_{i,l}, PK_{i,l}, R_i, T_i) \in Z_q^*$ V_i generates the signature $Sig_i = (H_3(R_i) - sk_{i,l} \cdot h_i) \cdot r_i^{-1}$ $V_i \xrightarrow{Msgs=\{M_i, PID_{i,l}, PK_{i,l}, R_i, T_i, Sig_i\}} RSU_j$	
Single Authentication			RSU_j checks whether T_i is fresh RSU_j computes $H_1(PID_{i,l}) \in Z_q^*$ RSU_j computes $h_i = H_2(M_i, PID_{i,l}, PK_{i,l}, R_i, T_i) \in Z_q^*$ RSU_j checks if $R_i^{Sig_i} \cdot A_{pub}^{H_1(PID_{i,l}) \cdot h_i} = g^{H_3(R_i)}$ holds
Batch Authentication			Vehicles $\xrightarrow{Msgs_1, Msgs_2, \cdots, Msgs_n}$ RSU_j RSU_j checks whether $\{T_1, T_2, \cdots, T_n\}$ are fresh RSU_j computes $H_1(PID_{i,l}) \in Z_q^*$ for $i = 1, \cdots, n$ RSU_j computes $h_i = H_2(M_i, PID_{i,l}, PK_{i,l}, R_i, T_i) \in Z_q^*$ for $i = 1, \cdots, n$ RSU_j checks if $g^{\sum_{i=1}^n (\varrho_i \cdot H_3(R_i))} = \prod_{i=1}^n R_i^{\varrho_i \cdot Sig_i} \cdot A_{pub}^{\sum_{i=1}^n (\varrho_i \cdot H_1(PID_{i,l}) \cdot h_i)}$
Traceability	TA receives reported $Msgs = \{M_i, PID_{i,l}, PK_{i,l}, R_i, T_i, Sig_i\}$ TA computes $ID_i = PID_{i,l} \oplus H_1(PK_{i,l}^b, B_{pub})$		

4.2.1 Anonymous CPS-CPPA Protocol for Vehicle

The details of proposed CPS-CPPA protocol for vehicle [27] are described as below, and the working flow is also illustrated in Table 4.1.

System Parameters Setup: Prior to the arrangement of VANETs, TA generates the system parameters $Params$ as follows:

1. Given a security parameter $k \in Z^{+}$, TA generates a prime q and a group G of the order q, where g is a generator of G. TA also chooses five cryptographic hash functions $H_0 : G \times G \rightarrow \{0,1\}^*$, $H_1 : \{0,1\}^* \rightarrow Z_q^*$, $H_2 : \{0,1\}^* \times \{0,1\}^* \times G \times G \times \{0,1\}^* \rightarrow Z_q^*$, $H_3 : G \rightarrow Z_q^*$ and $H_4 : G \times \{0,1\}^* \rightarrow Z_q^*$,.

2. TA selects a random number $a \in Z_q^*$ and sets $A_{pub} = g^a$, where a is a master secret key for private key extraction and is only known to TA. Similarly, TA chooses a random number $b \in Z_q^*$ and sets $B_{pub} = g^b$, where b is a master secret key for traceability and is only known to TA.

3. Finally, TA publishes system parameters $Params = \{q, G, g, A_{pub}, B_{pub}, H_0, H_1, H_2, H_3, H_4\}$.

Enrollment for Vehicle: Utilizing the Pseudo-Identities (PIDs) that are uniquely associated with the corresponding real identities allows us to achieve anonymous conditional privacy-preserving authentication in our solution.

1. A legitimate vehicle V_i transmits information including its unique identity ID_i (e.g. the vehicle user's personal identity, vehicle's license plate number etc.) to TA. Upon confirming the validity of ID_i, TA selects a group of private random numbers $\{k_{i,1}, k_{i,2}, \cdots, k_{i,z}\} \in Z_q^*$ and computes the corresponding public values $PK_i^* = \{PK_{i,1}, PK_{i,2}, \cdots, PK_{i,z}\}$, where $PK_{i,l} = g^{k_{i,l}}$ and $l \in \{1, 2, \cdots, z\}$.

2. TA generates a group of PIDs for V_i as $PID_i^* = \{PID_{i,1}, PID_{i,2}, \cdots, PID_{i,z}\}$, where $PID_{i,l} = ID_i \oplus H_0(PK_{i,l}^b, B_{pub})$ and $l \in \{1, 2, \cdots, z\}$. Hence, the real identity ID_i of vehicle V_i is masked in the pseudo-IDs PID_i^*.

3. After computing the PID_i^*, TA computes private keys $SK_i^* = \{sk_{i,1}, sk_{i,2}, \cdots, sk_{i,z}\}$, where $sk_{i,l} = a \cdot H_1(PID_{i,l})$ and $l \in \{1, 2, \cdots, z\}$.

4. Finally, TA sends system parameters $Params$ and z triple sets of $\{PID_i^*, SK_i^*, PK_i^*\}$ to vehicle V_i via a secure channel delivering a TPD for V_i. It is assumed that the adversary $\mathcal{A}$ cannot extract any information from the vehicle's TPD, even if $\mathcal{A}$ has registered one vehicle.

Vehicle Message Signing: In order to guarantee message authentication and integrity, each message issued by a vehicle should be signed and verified before it is accepted by the RSUs or other vehicles. The signature on one traffic-related message M_i by V_i is explained as follows.

1. V_i randomly selects a private key $sk_{i,l}$, a corresponding $PK_{i,l}$ and pseudo-identity $PID_{i,l}$ from the sets SK_i^*, PK_i^* and PID_i^* separately. Then, V_i chooses a random $r_i \in Z_q^*$ and computes $R_i = g^{r_i}$, $h_i = H_2(M_i, PID_{i,l}, PK_{i,l}, R_i, T_i) \in Z_q^*$, $Sig_i = (H_3(R_i) - sk_{i,l} \cdot h_i) \cdot r_i^{-1}$, where the generation of Sig_i is based on [58], and T_i is the current timestamp that supports the freshness of a valid signed message.

2. Then, V_i issues the signature message $Msgs = \{M_i, PID_{i,l}, PK_{i,l}, R_i, T_i, Sig_i\}$ to a nearby RSU.

Single Message Verification: Once the receiver (i.e. RSU or other vehicles) has received a single message signed by V_i, RSU will authenticate the message in order to ensure that the sender is a legitimate user rather than an adversary impersonating some legitimate user.

1. After receiving $Msgs = \{M_i, PID_{i,l}, PK_{i,l}, R_i, T_i, Sig_i\}$ signed by V_i, the receiver checks the freshness of timestamp T_i. The verifier drops the message if it is not fresh.

2. If T_i is valid, the receiver then computes $H_1(PID_{i,l})$, $h_i = H_2(M_i, PID_{i,l}, PK_{i,l}, R_i, T_i) \in Z_q^*$ and verifies whether $R_i^{Sig_i} \cdot A_{pub}^{H_1(PID_{i,l}) \cdot h_i} = g^{H_3(R_i)}$. If the equation is satisfied, then the receiver accepts the validity of the message M_i; otherwise, the receiver rejects it.

Batch Messages Verification: When there are a large number of vehicles in the communication range of the receiver, single message authentication may result in higher computation overhead due to verification delay. Therefore, this paper also presents a batch verification method so that the receiver can efficiently verify multiple messages at the same time. This will significantly decrease verification delay. In addition, the small exponent test technology [48, 59, 60] is adopted in the batch messages verification in order to guarantee the non-repudiation of signatures. Upon receiving n messages $\{M_1, PID_{1,l}, PK_{1,l}, R_1, T_1, Sig_1\}$,

$\{M_2, PID_{2,l}, PK_{2,l}, R_2, T_2, Sig_2\}$, $\cdots$, $\{M_n, PID_{n,l}, PK_{n,l}, R_n, T_n, Sig_n\}$ simultaneously, the receiver uses $Params = \{q, G, g, A_{pub}, B_{pub}, H_0, H_1, H_2, H_3, H_4\}$ to authenticate batch messages, as below.

1. The receiver checks the freshness of $\{T_1, T_2, \cdots, T_n\}$, and rejects the messages that are not fresh.

2. The receiver randomly selects n numbers $\{\varrho_1, \varrho_2, \cdots, \varrho_n\}$, where $\varrho_i \in_R [1, 2^m]$ for $i = 1, 2, \cdots, n$ and $m = 80$ is typically adequate [48,59,60].

3. The receiver computes $H_1(PID_{i,l})$, $h_i = H_2(M_i, PID_{i,l}, PK_{i,l}, R_i, T_i) \in Z_q^*$ for $i \in \{1, 2, \cdots, n\}$ and checks whether the below verification equation holds.

$$g^{\sum_{i=1}^{n}(\varrho_i \cdot H_3(R_i))} = \prod_{i=1}^{n} R_i^{\varrho_i \cdot Sig_i} \cdot A_{pub}^{\sum_{i=1}^{n}(\varrho_i \cdot H_1(PID_{i,l}) \cdot h_i)}.$$

If it is equal, then the receiver accepts the messages; otherwise, the receiver rejects the messages.

The correctness of the batch messages verification is demonstrated as below:

$\prod_{i=1}^{n} R_i^{\varrho_i \cdot Sig_i} \cdot A_{pub}^{\sum_{i=1}^{n}(\varrho_i \cdot H_1(PID_{i,l}) \cdot h_i)}$

$= \prod_{i=1}^{n}(R_i^{\varrho_i \cdot Sig_i} \cdot A_{pub}^{\varrho_i \cdot H_1(PID_{i,l}) \cdot h_i})$

$= \prod_{i=1}^{n}((g^{r_i})^{\varrho_i \cdot (H_3(R_i) - sk_{i,l} \cdot h_i) \cdot r_i^{-1}} \cdot (g^a)^{\varrho_i \cdot H_1(PID_{i,l}) \cdot h_i})$

$= \prod_{i=1}^{n}(g^{r_i \cdot \varrho_i \cdot (H_3(R_i) - (a \cdot H_1(PID_{i,l})) \cdot h_i) \cdot r_i^{-1}} \cdot g^{a \cdot \varrho_i \cdot H_1(PID_{i,l}) \cdot h_i})$

$= \prod_{i=1}^{n}(g^{r_i \cdot r_i^{-1} \cdot \varrho_i \cdot (H_3(R_i) - (a \cdot H_1(PID_{i,l})) \cdot h_i)} \cdot g^{\varrho_i \cdot a \cdot H_1(PID_{i,l}) \cdot h_i})$

$= \prod_{i=1}^{n}(g^{\varrho_i \cdot H_3(R_i) - \varrho_i \cdot a \cdot H_1(PID_{i,l}) \cdot h_i} \cdot g^{\varrho_i \cdot a \cdot H_1(PID_{i,l}) \cdot h_i})$

$= \prod_{i=1}^{n}(g^{\varrho_i \cdot H_3(R_i) - \varrho_i \cdot a \cdot H_1(PID_{i,l}) \cdot h_i \cdot + \varrho_i \cdot a \cdot H_1(PID_{i,l}) \cdot h_i})$

$= \prod_{i=1}^{n} g^{\varrho_i \cdot H_3(R_i)}$

$= g^{\sum_{i=1}^{n}(\varrho_i \cdot H_3(R_i))}$

Table 4.2: The working flow of ID-based CPS-CPPA protocol for RSU

Stages	TA	Vehicle V_i	RSU_j
RSU Enrollment	$TA \xleftarrow{\{ID_{rsu_j}\},\ offline} RSU_j$ TA selects random numbers $\{x_{j,1}, \cdots, x_{j,z}\} \in Z_q^*$ TA computes $Y_{j,l} = g^{x_{j,l}}$, $l \in \{1, \cdots, z\}$ TA computes $rsk_{j,l} = a \cdot H_4(Y_{j,l}, RID_j)$, $l \in \{1, \cdots, z\}$. $TA \xrightarrow{\{Params, RID_j, RSK_j^*, Y_j^*\},\ offline} RSU_j$		RSU_j stores $\{Params, RID_j, RSK_j^*, Y_j^*\}$ into its TPD
Signature Generation			RSU_j selects $rsk_{j,l}$, $Y_{j,l}$ RSU_j randomly chooses $w_j \in Z_q^*$ RSU_j computes $W_j = g^{w_j}$ RSU_j generates the current timestamp T_j RSU_j computes $rh_j = H_2(M_j, RID_j, Y_{j,l}, W_j, T_j) \in Z_q^*$ RSU_j generates $Rsig_j = (H_3(W_j) - RSK_{j,l} \cdot rh_j) \cdot w_j^{-1}$ $V_i \xleftarrow{Msgs=\{M_j, RID_j, Y_{j,l}, W_j, T_j, Rsig_j\}} RSU_j$
Single Authentication		V_i checks whether T_j is fresh V_i computes $rh_j = H_2(M_j, RID_j, Y_{j,l}, W_j, T_j) \in Z_q^*$, V_i computes $H_1(Y_{j,l}, RID_j) \in Z_q^*$ V_i checks if $W_j^{Rsig_j} \cdot A_{pub}^{H_4(Y_{j,l}, RID_j) \cdot rh_j} = g^{H_3(W_j)}$ holds	
Batch Authentication		$V_i \xleftarrow{Msgs_1, Msgs_2, \cdots, Msgs_t} RSU_j$ V_i checks whether $\{T_1, T_2, \cdots, T_t\}$ are fresh V_i computes $rh_j = H_2(M_j, RID_j, Y_{j,l}, W_j, T_j) \in Z_q^*$ for $i = 1, \cdots, t$ V_i computes $H_4(Y_{j,l}, RID_j) \in Z_q^*$ for $i = 1, \cdots, t$ V_i checks if $\prod_{j=1}^{t} W_j^{\varsigma_j \cdot Sig_j} \cdot A_{pub}^{\sum_{j=1}^{t}(\varsigma_j \cdot H_1(Y_{j,l}, RID_j) \cdot rh_j)} = g^{\sum_{j=1}^{t}(\varsigma_j \cdot H_3(W_j))}$	

4.2.2 ID-based CPS-CPPA Protocol for RSU

The RSUs are supposed to present their real identities when sending signed messages, since they belong to the infrastructure and are not subject to the privacy issue. The details of the proposed CPS-CPPA protocol for RSU [27] are shown as follows (see also Table 4.2). The system parameters setup stage in ID-based CPPA solution for RSU is the same as those described in 4.2.1; thus, this section omits this stage in the discussion that follows.

Enrollment for RSU: TA generates a unique identity RID_j for each RSU, which includes its corresponding location information. Then, TA computes private keys for RSU as follows.

1. For a given RSU's identity RID_j, TA selects a group of private random numbers $\{x_{j,1}, x_{j,2}, \cdots, x_{j,z}\} \in Z_q^*$ and computes the corresponding public values $Y_j^* = \{Y_{j,1}, Y_{j,2}, \cdots, Y_{j,z}\}$, where $Y_{j,l} = g^{x_{j,l}}$ and $l \in \{1, 2, \cdots, z\}$.

2. TA computes private keys $RSK_j^* = \{RSK_{j,1}, RSK_{j,2}, \cdots, RSK_{j,z}\}$, where $RSK_{j,l} = a \cdot H_4(Y_{j,l}, RID_j)$ and $l \in \{1, 2, \cdots, z\}$.

3. Finally, TA sends $Params$ and $\{RID_j, RSK_j^*, Y_j^*\}$ to RSU via a secure channel. Then, RSU stores its private key $\{RSK_j^*, Y_j^*\}$ with its corresponding identity RID_j into its storage memory.

RSU Message Signing: In the event when an RSU broadcasts location-based traffic information to nearby vehicles, the signature upon traffic message M_j generated by the RSU is as follows:

1. RSU chooses a private key $RSK_{j,l}$ from the set RSK_j^*, a corresponding $Y_{j,l}$ from the set Y_j^*, a random $w_j \in Z_q^*$ and computes $W_j = g^{w_j}$, $rh_j = H_2(M_j, RID_j, Y_{j,l}, W_j, T_j) \in Z_q^*$, and $Rsig_j = (H_3(W_j) - RSK_{j,l} \cdot rh_j) \cdot w_j^{-1}$, whereby T_j is the current timestamp which supports the freshness of a valid signed message.

2. Then, RSU broadcasts the signature message $Msgs = \{M_j, RID_j, Y_{j,l}, W_j, T_j, Rsig_j\}$ to nearby vehicles.

Single Message Verification: When a vehicle V_i receives single message signed by an RSU, V_i will have to authenticate the message in order to ensure the legitimacy of RSU.

1. After receiving $Msgs = \{M_j, RID_j, Y_{j,l}, W_j, T_j, Rsig_j\}$ signed by the RSU, V_i checks the freshness of timestamp T_j and drops the message if T_j is not fresh.

2. If T_j is valid, then V_i computes $rh_j = H_2(M_j, RID_j, Y_{j,l}, W_j, T_j) \in Z_q^*$, $H_4(Y_{j,l}, RID_j)$ and verifies whether $W_j^{Rsig_j} \cdot A_{pub}^{H_4(Y_{j,l}, RID_j) \cdot rh_j} = g^{H_3(W_j)}$. If the equation is satisfied, then V_i accepts the validity of the message M_j; otherwise, V_i rejects it.

Batch Messages Verification: To handle the situation when a vehicle receives multiple signed messages from RSUs in a time interval, a batch verification method is also presented. This allows the vehicle to efficiently verify multiple messages from vehicles at the same time. Specifically, after receiving t messages $\{M_1, RID_1, Y_{1,l}, W_1, T_1, Rsig_1\}$, $\{M_2, RID_2, Y_{2,l}, W_2, T_2, Rsig_2\}$, $\cdots$, $\{M_t, RID_t, Y_{t,l}, W_t, T_t, Rsig_t\}$ simultaneously, the vehicle verifies them using the below steps.

1. The vehicle checks the freshness of $\{T_1, T_2, \cdots, T_t\}$, and rejects the messages if some of them are not fresh.

2. The vehicle randomly selects t numbers $\{\varsigma_1, \varsigma_2, \cdots, \varsigma_t\}$, where $\varsigma_j \in_R [1, 2^m]$ for $j = 1, 2, \cdots, t$ and $m = 80$ is typically adequate [48,59,60].

3. The vehicle computes $rh_j = H_2(M_j, RID_j, Y_{j,l}, W_j, T_j) \in Z_q^*$, $H_4(Y_{j,l}, RID_j)$ for $j \in \{1, 2, \cdots, t\}$ and checks whether the below verification equation holds.

$$\prod_{j=1}^{t} W_j^{\varsigma_j \cdot Rsig_j} \cdot A_{pub}^{\sum_{j=1}^{t}(\varsigma_j \cdot H_4(Y_{j,l}, RID_j) \cdot rh_j)} = g^{\sum_{j=1}^{t}(\varsigma_j \cdot H_3(W_j))}.$$

If it is equal, then the vehicle accepts the messages; otherwise, the vehicle rejects the messages.

The correctness of the batch messages verification is demonstrated, as follows.

$$\begin{aligned}
&\textstyle\prod_{j=1}^{t} W_j^{\varsigma_j \cdot Rsig_j} \cdot A_{pub}^{\sum_{j=1}^{t}(\varsigma_j \cdot H_4(Y_{j,l}, RID_j) \cdot rh_j)} \\
&= \textstyle\prod_{j=1}^{t}(W_j^{\varsigma_j \cdot Rsig_i} \cdot A_{pub}^{\varsigma_j \cdot H_4(Y_{j,l}, RID_j) \cdot rh_j}) \\
&= \textstyle\prod_{j=1}^{t}((g^{w_j})^{\varsigma_j \cdot (H_3(W_j) - sk_{i,l} \cdot rh_j) \cdot w_j^{-1}} \cdot (g^{a})^{\varsigma_j \cdot H_4(Y_{j,l}, RID_j) \cdot rh_j}) \\
&= \textstyle\prod_{j=1}^{t}(g^{w_j \cdot \varsigma_j \cdot (H_3(W_j) - (a \cdot H_4(Y_{j,l}, RID_j)) \cdot rh_j) \cdot w_j^{-1}} \cdot g^{a \cdot \varsigma_j \cdot H_4(Y_{j,l}, RID_j) \cdot rh_j}) \\
&= \textstyle\prod_{j=1}^{t}(g^{w_j \cdot w_j^{-1} \cdot \varsigma_j \cdot (H_3(W_j) - (a \cdot H_4(Y_{j,l}, RID_j)) \cdot rh_j)} \cdot g^{\varsigma_j \cdot a \cdot H_4(Y_{j,l}, RID_j) \cdot rh_j}) \\
&= \textstyle\prod_{j=1}^{t}(g^{\varsigma_j \cdot H_3(W_j) - \varsigma_j \cdot a \cdot H_4(Y_{j,l}, RID_j) \cdot rh_j} \cdot g^{\varsigma_j \cdot a \cdot H_4(Y_{j,l}, RID_j) \cdot rh_j}) \\
&= \textstyle\prod_{j=1}^{t}(g^{\varsigma_j \cdot H_3(W_j) - \varsigma_j \cdot a \cdot H_4(Y_{j,l}, RID_j) \cdot rh_j \cdot + \varsigma_j \cdot a \cdot H_4(Y_{j,l}, RID_j) \cdot rh_j}) \\
&= \textstyle\prod_{j=1}^{t} g^{\varsigma_j \cdot H_3(W_j)} \\
&= g^{\sum_{j=1}^{t}(\varsigma_j \cdot H_3(W_j))}
\end{aligned}$$

4.3 Security Proofs

In this section, it will demonstrate that the presented anonymous CPS-CPPA protocol for vehicle achieves the security and privacy requirements outlined in Section 2.2 [27]. We does not give further analysis on the ID-based CPPA protocol for RSU, since the process of proof and analysis is similar to that of the presented anonymous CPS-CPPA protocol for vehicle as below.

4.3.1 Security Model

The definition of security for our proposed solution is given by a game executed between a polynomial-time adversary $\mathcal{A}$ and a challenger $\mathcal{I}$. In the game, $\mathcal{A}$ mounts a number of oracle queries to $\mathcal{I}$ as follows, which can be requested adaptively.

$Setup$: This query simulates the initialization of the VANETs system. When receiving this query, $\mathcal{I}$ creates the master keys and $Params$, and returns $Params$ to $\mathcal{A}$.

H_i: After $\mathcal{A}$ sends the query with the information I, $\mathcal{I}$ selects a random number $\pi_i \in Z_q^*$, stores (I, π_i) in the list L_{H_i} and returns π_i to A, where $i = 0, 1, 2, 3$.

$GenerateVehicle$: Upon receiving the vehicle V_i's identity ID_i, $\mathcal{I}$ produces V_i's pseudo-identities PID_i^*, private keys SK_i^*, public values PK_i^* and stores $\{ID_i, PID_i^*, SK_i^*, PK_i^*\}$ in the list $L_{vehicle}$.

$CorruptVehicle$: Upon receiving the vehicle V_i's identity ID_i, $\mathcal{I}$ transmits $\{PID_i^*, SK_i^*\}$ to $\mathcal{A}$.

$Signature$: Upon receiving $\mathcal{A}$'s message M and pseudo-identity PID_i, $\mathcal{I}$ generates and returns the corresponding signature message $Msgs$ to $\mathcal{A}$.

Upon executing the aforementioned queries, $\mathcal{A}$ fabricates a signature Sig_i^* of a traffic message M_i^* associated with V_i^*'s identity ID_i^*.

$\mathcal{A}$ wins the above experiment if all the below conditions are fulfilled.

1) Sig_i^* is legitimate, namely: $Verification(M^*, V_i^*, ID_i^*, Sig_i^*) = 1$.

2) $\mathcal{A}$ has not executed a $CorruptVehicle$ query associated with V_i^*'s identity ID_i^*.

3) $\mathcal{A}$ has not executed a $Signature$ query associated with V_i^*'s pseudo-identity PID_i^* and message M_i^*.

Let the function $Adv_{\Omega_1,\mathcal{A}}^{CPS-CPPA}$ denote the advantage of $\mathcal{A}$ in breaking conditional privacy-preserving authentication of the presented CPS-CPPA solution Ω_1.
Definition 1. *The proposed CPS-CPPA solution Ω_1 is chosen-identity and chosen-message secure, if for any polynomial-time adversary $\mathcal{A}$, the function $Adv_{\Omega_1,\mathcal{A}}^{CPS-CPPA}$ is negligible.*

4.3.2 Provable Security

Based on Definition 1, the chosen-identity and chosen-message security of the CPS-CPPA solution using random oracles are proved.
Theorem 1. *Assuming that the underlying DL problem is intractable, the CPS-CPPA solution for VANETs is secure in the random oracle model.*

Proof. Assume that a polynomial-time adversary $\mathcal{A}$ could fabricate a valid signature message $Msgs = \{M_i, PID_i, PK_i, R_i, T_i, Sig_i\}$ by a non-negligible advantage ε, then the challenger $\mathcal{I}$ can solve the DL problem with a non-negligible advantage through executing the $\mathcal{A}$ as a subroutine. Let $A_{pub} = g^a$ be an instance of the DL problem, and the aim of the $\mathcal{I}$ is to compute a. First, I issues $Params = \{q, G, g, A_{pub}, B_{pub}, H_0, H_1, H_2, H_3\}$ to $\mathcal{A}$, and $\mathcal{A}$ performs random oracle queries adaptively simulated by $\mathcal{I}$ as below.

$\mathbf{H_0}$ **Oracle**: $\mathcal{I}$ maintains a list L_{H_0} in the form of $\{\Theta, B_{pub}, \pi_0\}$, which is empty initially. When $\mathcal{A}$ issues a query $\{\Theta, B_{pub}\}$ to $\mathcal{I}$, $\mathcal{I}$ checks whether the tuple $\{\Theta, B_{pub}, \pi_0\}$ is in the list L_{H_0}. If so, $\mathcal{I}$ issues $\pi_0 = H_0(\Theta, B_{pub})$ to $\mathcal{A}$, otherwise, $\mathcal{I}$ selects a random nonce $\pi_0 \in Z_p$, issues $\pi_0 = H_0(\Theta, B_{pub})$ to $\mathcal{A}$ and appends $\{\Theta, B_{pub}, \pi_0\}$ to the list L_{H_0}.

$\mathbf{H_1}$ **Oracle**: $\mathcal{I}$ maintains a list L_{H_1} in the form of $\{\Upsilon, \pi_1\}$, which is empty initially. When $\mathcal{A}$ issues a query Υ to $\mathcal{I}$, $\mathcal{I}$ checks whether the tuple $\{\Upsilon, \pi_1\}$ is in the list

L_{H_1}. If so, $\mathcal{I}$ issues $\pi_1 = H_1(\Upsilon)$ to $\mathcal{A}$, otherwise, $\mathcal{I}$ selects a random nonce $\pi_1 \in Z_p$, issues $\pi_1 = H_1(\Upsilon)$ to $\mathcal{A}$ and appends $\{\Upsilon, \pi_1\}$ to the list L_{H_1}.

$\mathbf{H_2}$ Oracle: $\mathcal{I}$ maintains a list L_{H_2} in the form of $\{M_i, PID_i, PK_i, R_i, T_i, \pi_2\}$, which is empty initially. When $\mathcal{A}$ issues a query $\{M_i, PID_i, PK_i, R_i, T_i\}$ to $\mathcal{I}$, $\mathcal{I}$ checks whether the tuple $\{M_i, PID_i, PK_i, R_i, T_i, \pi_2\}$ is in the list L_{H_2}. If so, $\mathcal{I}$ issues $\pi_2 = H_2(M_i, PID_i, PK_i, R_i, T_i)$ to $\mathcal{A}$, otherwise, $\mathcal{I}$ selects a random nonce $\pi_2 \in Z_p$, issues $\pi_2 = H_2(M_i, PID_i, PK_i, R_i, T_i)$ to $\mathcal{A}$ and appends $\{M_i, PID_i, PK_i, R_i, T_i, \pi_2\}$ to the list L_{H_2}.

$\mathbf{H_3}$ Oracle: $\mathcal{I}$ maintains a list L_{H_3} in the form of $\{R_i, \pi_3\}$, which is empty initially. When $\mathcal{A}$ issues a query $\{R_i\}$ to $\mathcal{I}$, $\mathcal{I}$ checks whether the tuple $\{R_i, \pi_3\}$ is in the list L_{H_2}. If so, $\mathcal{I}$ issues $\pi_3 = H_3(R_i)$ to $\mathcal{A}$, otherwise, $\mathcal{I}$ selects a random nonce $\pi_3 \in Z_p$, issues $\pi_3 = H_3(R_i)$ to $\mathcal{A}$ and appends $\{R_i, \pi_3\}$ to the list L_{H_3}.

GenerateVehicle Oracle: $\mathcal{I}$ maintains a list $L_{vehicle}$ in the form of $\{ID_i, k_i, PK_i, PID_i, SK_i\}$ which is empty initially. Once $\mathcal{A}$ sends this query to $\mathcal{I}$, $\mathcal{A}$ checks whether the tuple $\{ID_i, k_i, PID_i, sk_i, PK_i\}$ is in the list $L_{vehicle}$. If so, $\mathcal{I}$ returns PK_i to $\mathcal{A}$; otherwise $\mathcal{I}$ executes the steps as below.

1) If $ID_i = ID_i^*$, $\mathcal{I}$ selects three random numbers k_i, π_0 and π_1, computes $PK_i = g^{k_i}$ and holds $\{PID_i, SK_i\}$. $\mathcal{I}$ stores $\{ID_i, k_i, PID_i, sk_i, PK_i\}$, $\{\Theta, B_{pub}, \pi_0\}$ and $\{\Upsilon, \pi_1\}$ in the lists $L_{vehicle}$, L_{H_0} and L_{H_1} respectively. At last, $\mathcal{I}$ returns PK_i to $\mathcal{A}$.

2) If $ID_i \neq ID_i^*$, $\mathcal{I}$ selects three random numbers k_i, π_0 and π_1, computes $PK_i = g^{k_i}$, $PID_i = ID_i \oplus \pi_0$, $SK_i = a \cdot \pi_1$. $\mathcal{I}$ stores $\{ID_i, k_i, PID_i, sk_i, PK_i\}$, $\{\Theta, B_{pub}, \pi_0\}$ and $\{\Upsilon, \pi_1\}$ in the lists $L_{vehicle}$, L_{H_0} and L_{H_1} respectively and finally returns PK_i to $\mathcal{A}$.

CorruptVehicle Oracle: $\mathcal{A}$ cannot mount this inquiry in Ω_1, because we assume that the adversary $\mathcal{A}$ cannot extract any information from the vehicle's TPD, even if $\mathcal{A}$ has registered one vehicle.

Signature Oracle: Upon receiving $\mathcal{A}$'s query with message M_i and pseudo-identity PID_i, $\mathcal{I}$ selects two random numbers r_i, π_2, π_3 and computes $R_i = g^{r_i}$ and $Sig_i = (\pi_3 - SK_i \cdot \pi_2) \cdot r_i^{-1}$. $\mathcal{I}$ stores $\{M_i, PID_i, PK_i, R_i, T_i, \pi_2\}$ to the list L_{H_2}, $\{R_i, \pi_3\}$ to the list L_{H_3} and returns the signature message $Msgs = \{M_i, PID_i, PK_i, R_i, T_i, Sig_i\}$ to $\mathcal{A}$.

Finally, $\mathcal{A}$ outputs a signature message $\{M_i, PID_i, PK_i, R_i, T_i, Sig_i\}$ to $\mathcal{I}$ with PID_i. If $PID_i \neq PID_i^*$, then $\mathcal{I}$ aborts the game. $\mathcal{I}$ checks whether the below equation is correct.

$$R_i^{Sig_i} \cdot A_{pub}^{H_1(PID_i) \cdot h_i} = g^{H_3(R_i)} \tag{4.1}$$

If it is not correct, then $\mathcal{I}$ interrupts the game. Based on the forking lemma in [61], if the challenger repeats the procedure with a different selection H_2, then $\mathcal{A}$ can output another legitimate signature message $\{M_i, PID_i, PK_i, R_i, T_i, Sig_i'\}$ with the advantage $\varepsilon' \geq \frac{1}{9}$. Thus, the following equation is obtained:

$$R_i^{Sig_i'} \cdot A_{pub}^{H_1(PID_i) \cdot h_i'} = g^{H_3(R_i)} \tag{4.2}$$

According to the above two equations, the following equations are obtained:

$$R_i^{Sig_i - Sig_i'} = A_{pub}^{H_1(PID_i) \cdot (h_i' - h_i)} \tag{4.3}$$

$$R_i^{Sig_i \cdot h_i' - Sig_i' h_i} = g^{H_3(R_i) \cdot (h_i' - h_i)} \tag{4.4}$$

Hence, based on Equations 4.3 and 4.4, the following equations could be respectively obtained.

- $R_i^{Sig_i - Sig_i'} = A_{pub}^{H_1(PID_i) \cdot (h_i' - h_i)}$, $g^{r_i \cdot (Sig_i - Sig_i')} = g^{a \cdot H_1(PID_i) \cdot (h_i' - h_i)}$

$$r_i \cdot (Sig_i - Sig_i') = a \cdot H_1(PID_i) \cdot (h_i' - h_i) \tag{4.5}$$

- $R_i^{Sig_i \cdot h_i' - Sig_i' h_i} = g^{H_3(R_i) \cdot (h_i' - h_i)}$, $g^{r_i (Sig_i \cdot h_i' - Sig_i' h_i)} = g^{H_3(R_i) \cdot (h_i' - h_i)}$

$$r_i \cdot (Sig_i \cdot h_i' - Sig_i' \cdot h_i) = H_3(R_i) \cdot (h_i' - h_i) \tag{4.6}$$

According to Equations 4.5 and 4.6, $\mathcal{I}$ outputs $H_3(R_i) \cdot H_1(PID_i)^{-1}(Sig_i - Sig_i') \cdot (Sig_i \cdot h_i' - Sig_i' \cdot h_i)^{-1}$ as the result of the DL problem. However, this is a contradiction with the hardness of the DL problem in G. Consequently, this completes the proof.

□

4.3.3 Security and Attributes Analysis

Identity Privacy Preservation: In the enrollment stage, the vehicle's genuine identity is concealed in the $PID_i^* = \{PID_{i,1}, PID_{i,2}, \cdots, PID_{i,z}\}$ by TA, where $PID_{i,l} = ID_i \oplus H_0(PK_{i,l}^b, B_{pub})$ and $l \in \{1, 2, \cdots, z\}$. To reveal the real identity ID_i from $PID_{i,l} = ID_i \oplus H_0(PK_{i,l}^b, B_{pub})$, $\mathcal{A}$ needs to compute $PK_{i,l}^b = g^{k_{i,l} \cdot b}$ based on $PK_{i,l} = g^{k_{i,l}}$ and $B_{pub} = g^b$. This, however, contradicts the hardness of CDH problem. Thus, the CPS-CPPA solution for VANETs preserves identity privacy.

Message Authentication and Integrity: Upon receiving $Msgs = \{M_i, PID_{i,l}, PK_{i,l}, R_i, T_i, Sig_i\}$ from V_i, the verifier (one nearby RSU or vehicle) verifies the correctness of $R_i^{Sig_i} \cdot A_{pub}^{H_1(PID_{i,l}) \cdot h_i} = g^{H_3(R_i)}$ in order to check the message's validity and integrity. Based on Theorem 1 in Section 4.3.2, there is no polynomial-time adversary $\mathcal{A}$ that could fabricate a legal message when the DL problem is hard. Thus, $\mathcal{A}$ cannot obtain the master private key of TA and generates legitimate information for message authentication and integrity.

Traceability: In the pseudo-identity generation and private key extraction stage, the vehicle's genuine identity is in the pseudo-IDs $PID_i^* = \{PID_{i,1}, PID_{i,2}, \cdots, PID_{i,z}\}$, where $PID_{i,l} = ID_i \oplus H_0(PK_{i,l}^b, B_{pub})$ and $l \in \{1, 2, \cdots, z\}$. By knowing the master secret key b of the VANETs system, TA could extract the real identity $ID_i = PID_{i,l} \oplus H_0(PK_{i,l}^b, B_{pub})$. Consequently, the function of traceability is provided by the proposed CPS-CPPA solution.

Unlinkability: TA selects a group of private random numbers $\{k_{i,1}, k_{i,2}, \cdots, k_{i,z}\} \in Z_q^*$ in the enrollment stage and the vehicle also chooses random $r_i \in Z_q^*$ in the message signing stage, where $PID_i^* = \{PID_{i,1}, PID_{i,2}, \cdots, PID_{i,z}\}$, $PID_{i,l} = ID_i \oplus H_0(PK_{i,l}^b, B_{pub})$, $SK_i^* = \{sk_{i,1}, sk_{i,2}, \cdots, sk_{i,z}\}$, $R_i = g^{r_i}$, $h_i = H_2(M_i,$

$PID_i, PK_{i,l}, R_i, T_i) \in Z_q^*$, $Sig_i = (H_3(R_i) - sk_{i,l} \cdot h_i) \cdot r_i^{-1}$. Due to the randomness of $k_{i,1}$ and r_i, the vehicle could generate random identities and signatures from which the adversary cannot find the connection between two anonymous identities or two signatures (i.e. not able to determine whether they are sent by the same vehicle). Thus, our CPS-CPPA solution achieves unlinkability.

Resilient to Message Modification Attack: Each vehicle user broadcasts an anonymous signature message to nearby RSUs and other vehicles in the format $Msgs = \{M_i, PID_{i,l}, PK_{i,l}, R_i, T_i, Sig_i\}$. $\mathcal{A}$ has the capability to change the content of M_i after eavesdropping on the wireless medium. To protect the integrity of the message, a vehicle's signature on M_i is generated as $Sig_i = (H_3(R_i) - sk_{i,l} \cdot h_i) \cdot r_i^{-1}$, where T_i is the current timestamp and $R_i = g^{r_i}$, $h_i = H_2(M_i, PID_{i,l}, PK_{i,l}, R_i, T_i) \in Z_q^*$. Since the private key SK_i is only known by the particular vehicle, no attacker can generate a valid signature. Besides, the private key SK_i is changed periodically. Thus, the presented CPS-CPPA solution for VANETs is secure against message modification attack.

Resilient to Impersonation Attack: To execute an impersonation attack, $\mathcal{A}$ has to be able to generate valid $Msgs = \{M_i, PID_{i,l}, PK_{i,l}, R_i, T_i, Sig_i\}$, where $R_i^{Sig_i} \cdot A_{pub}^{H_1(PID_{i,l}) \cdot h_i} = g^{H_3(R_i)}$. Based on Theorem 1, $\mathcal{A}$ cannot fabricate such a signature message. RSUs and other vehicles can check the legality of messages through verifying the correctness of the aforementioned equation. Thus, the proposed CPS-CPPA solution for VANETs could resist the impersonation attack.

Resilient to Replay Attack: Timestamp T_i is included in the signature message $Msgs = \{M_i, PID_{i,l}, PK_{i,l}, R_i, T_i, Sig_i\}$ and is also included in the generation of Sig_i. Thus, the verifier could detect a replay attack when T_i is no longer fresh. Thus, the proposed CPS-CPPA solution for VANETs is replay attack resilience.

Full Batch Verification: According to the function of batch verification in Section 4.2.1, upon receiving n messages $\{M_1, PID_{1,l}, PK_{1,l}, R_1, T_1, Sig_1\}$, $\{M_2, PID_{2,l}, PK_{2,l}, R_2, T_2, Sig_2\}$, $\cdots$, $\{M_n, PID_{n,l}, PK_{n,l}, R_n, T_n, Sig_n\}$ from different vehicles during the same time interval, RSUs could verify their legitimacy simultaneously.

No Map-to-Point Operation: It is expensive and complicated to execute the map-to-point operation, and map-to-point operation is avoided in the CPS-CPPA scheme for VANETs.

No Certificate Management: In the CPS-CPPA solution for vehicle, neither vehicle nor RSUs store any certificates for message verification. The vehicle only needs to memorize the system parameters $Params$ and $\{PID_i^*, SK_i^*, PK_i^*\}$, where $PID_i^* = \{PID_{i,1}, PID_{i,2}, \cdots, PID_{i,z}\}$, $PID_{i,l} = ID_i \oplus H_0(PK_{i,l}^b, B_{pub})$, $SK_i^* = \{sk_{i,1}, sk_{i,2}, \cdots, sk_{i,z}\}$, $sk_{i,l} = a \cdot H_1(PID_{i,l})$, $PK_i^* = \{PK_{i,1}, PK_{i,2}, \cdots, PK_{i,z}\}$, $PK_{i,l} = g^{k_{i,l}}$, and $l \in \{1, 2, \cdots, z\}$ generated by the TA. Therefore, TA does not need to manage any certificate.

No Verifier Table: The adversary is not capable of stealing any verifier table since there is no verifier table maintained by RSUs or vehicles. Therefore, the presented CPS-CPPA solution for VANETs is stolen verifier table attack resilience.

Provable Security: The security proof of the cryptographic scheme is widely adopted by cryptography protocols, so that the customers (e.g. individuals, companies and governments etc.) would believe the security of the cryptographic system. Therefore, the presented CPS-CPPA scheme is proved securely under a security model.

4.4 Overheads Analysis

The overheads analysis on the CPS-CPPA protocol will be done in chapter 6 with the ECPS-CPPA together.

4.5 Summary

VANETs will be increasingly popular and potentially be more interconnected with our fabrics of society. For example, in the future, sensors on vehicles may be used to collect our body data that can be linked to healthcare and other relevant industries in order to deliver appropriate services. Security and privacy will remain two of several key research topics in such applications, at least in the foreseeable future.

In this chapter, it presented an efficient and anonymous CPPA scheme based on the TPD, which can be utilized in safety-related VANETs applications. It then proved the security of the proposed solution. However, there is a weakness in the CPS-CPPA protocol, which would be discussed in Chapter 5.

Chapter 5

ECPS-CPPA: Enhanced, Certificateless and Provably-Secure Conditional Privacy-Preserving Authentication Protocol

5.1 Motivation

In chapter 4, the CPS-CPPA attempts to resolve the weaknesses in the existing CPPA protocols, especially those weaknesses in Azees et al.'s protocol [20] and Zhang et al. 's protocol [26]. The CPS-CPPA protocol provides properties successfully, such as the message authentication and integrity, identity-preserving protection, traceability, un-linkability and batch verification. But this chapter in Section 5.2 will point out that the CPS-CPPA cannot guarantee the security of master key a in practice, and not resist modification forged message attack as well as impersonation attack.

To improve the CPS-CPPA protocol further, this chapter in Section 5.3 presents an ECPS-CPPA protocol to be used in vehicular environments that supports both privacy and security requirements in the VANETs system, and in Section 5.4 we also demonstrate that our ECPS-CPPA protocols secure against modification attack, impersonation attack, and other existing attacks and is certificateless.

5.2 Security Analysis of CPS-CPPA Protocol

In this section, we will propose three vulnerabilities against our CPS-CPPA solution, which are presented as below.

5.2.1 Leakage of Master Secret Key

Assuming the adversary $\mathcal{A}$ has registered a vehicle in TA with $\{PID_A^*, SK_A^*, PK_A^*\}$ stored in its TPD, where $PID_A^* = \{PID_{A,l} = ID_A \oplus H_0(PK_{A,l}^b, B_{pub}), l \in \{1, \cdots, z\}\}$, $SK_A^* = \{sk_{A,l} = a \cdot H_1(PID_{A,l}), l \in \{1, \cdots, z\}\}$ and $PK_A^* = \{PK_{A,l} = g^{k_{A,l}}, l \in \{1, \cdots, z\}\}$. Although the storage device is assumed to be unassailable, a highly motivated adversary can extract the information $\{PID_A^*, SK_A^*, PK_A^*\}$ stored in the device by power analysis techniques [44].

According to $\{PID_{A,l}, sk_{A,l}, PK_{A,l}\}$ and $sk_{A,l} = a \cdot H_1(PID_{A,l})$, the master key is calculated as $a = sk_{A,l} \cdot H_1(PID_{A,l})^{-1}$. And then the malicious adversary $\mathcal{A}$ mount other attacks such as forged message attack and impersonation attack as follows.

5.2.2 Forged Message Attack

The attacker $\mathcal{A}$ can transmit a forged message by the following steps.

1. $\mathcal{A}$ generates three random numbers $r_A \in Z_q^*$, $PID_A \in Z_q^*$, PK_A, and computes $R_A = g^{r_A}$ and secret key $sk_A = a \cdot H_1(PID_A) \bmod q$.

2. $\mathcal{A}$ computes $h_A = H_2(M_A, PID_A, PK_A, R_A, T_A) \in Z_q^*$, and $Sig_A = (H_3(R_A) - sk_A \cdot h_A) \cdot r_A^{-1} \bmod q$, where M_A is a message upon traffic status and T_A is the current time-stamp. Then, the vehicle broadcasts $\{M_A, PID_A, T_A, R_A, Sig_A\}$ to nearby RSUs and vehicles.

3. After receiving $\{M_A, PID_A, T_A, R_A, c_A\}$ from $\mathcal{A}$, the verifier checks the freshness of T_A, apparently, it is easy to produce fresh time-stamp.

4. The verifier computes $H_1(PID_A)$, $h_A = H_2(M_A, PID_A, PK_A, R_A, T_A) \in Z_q^*$ and accepts the message because the equation $R_A^{Sig_A} \cdot A_{pub}^{H_1(PID_A)\cdot h_A} = g^{H_3(R_A)}$ holds. It is easy to verify as following steps. Due to$R_A = g^{r_A}$, $A_{pub} = g^a$, $sk_A = a \cdot H_1(PID_A) \bmod q$, $Sig_A = (H_3(R_A) - sk_A \cdot h_A) \cdot r_A^{-1} \bmod q$, we could get that

$$
\begin{aligned}
& R_A^{Sig_A} \cdot A_{pub}^{H_1(PID_A)\cdot h_A} \\
&= (g^{r_A})^{(H_3(R_A)-sk_A\cdot h_A)\cdot r_A^{-1}} \cdot (g^a)^{H_1(PID_A)\cdot h_A} \\
&= g^{(H_3(R_A)-sk_A\cdot h_A)} \cdot (g^a)^{H_1(PID_A)\cdot h_A} \\
&= g^{(H_3(R_A)-a\cdot H_1(PID_A)\cdot h_A)} \cdot (g^a)^{H_1(PID_A)\cdot h_A} \\
&= g^{(H_3(R_A)-a\cdot H_1(PID_A)\cdot h_A)+a\cdot H_1(PID_A)\cdot h_A} \\
&= g^{H_3(R_A)}
\end{aligned}
$$

5.2.3 Impersonation Attack

$\mathcal{A}$ can forge a message with one intercepted user's pseudo-identity $PID_{i,l}$ through the following steps.

1. $\mathcal{A}$ intercepts the transmitted message $Msgs = \{M_i, PID_{i,l}, PK_{i,l}, R_i, T_i, Sig_i\}$ sent from vehicle V_i.

2. $\mathcal{A}$ uses $\{PID_{i,l}, PK_{i,l}\}$, to replace his/her $\{PID_A, PK_A\}$.

3. $\mathcal{A}$ executes the other steps as those in 5.2.2.

4. If the TA later finds out that M_A is a fake message, it will recover the identity ID_i by computing $ID_i = PID_{i,l} \oplus H_0(PK_{i,l}^b, B_{pub})$ as the genuine identity for the fake message, although V_A is the genuine signer for that. Hence, the adversary $\mathcal{A}$ could mount the framing attack on the honest V_i.

5.3 The ECPS-CPPA Protocol

To overcome the vulnerabilities analyzed in the previous subsection 5.2, we redesign an ECPS-CPPA solution for VANETs based on CPS-CPPA [27] in this section. The ECPS-CPPA protocol also consists of two parts, namely: an anonymous ECPS-CPPA solution for vehicle and an ID-based ECPS-CPPA solution for RSU.

5.3.1 ECPS-CPPA Protocol for Vehicle

The details of ECPS-CPPA protocol for vehicle are described as below, and the working flow is also illustrated in Table 5.1.

System Parameters Setup: TA generates the system parameters $Params$ as follows:

1. Depending on the security parameter $k \in Z^+$, TA generates a prime q and a group G of the order q, where g is a generator of G. TA also chooses three cryptographic hash functions $H_0 : G \times G \to \{0,1\}^*$, $H_1 : G \times \{0,1\}^* \to Z_q^*$, and $H_2 : \{0,1\}^* \times G \times \{0,1\}^* \to Z_q^*$.

2. TA picks a random number $a \in Z_q^*$ and sets $A_{pub} = g^a$, where a is a master secret key for private key extraction and is only known to TA.

3. TA chooses a random number $b \in Z_q^*$ and sets $B_{pub} = g^b$, where b is a master secret key for traceability and is only known to TA.

4. TA publishes system parameters $Params = \{q, G, g, A_{pub}, B_{pub}, H_0, H_1, H_2\}$.

Enrollment for Vehicle:

1. The vehicle V_i transmits his/her identity ID_i including the owner's personal information and the vehicle's information to TA. Upon confirming the validity of ID_i, TA selects a group of private random numbers $\{k_{i,1}, k_{i,2}, \cdots, k_{i,z}\} \in Z_q^*$ and computes the corresponding public values $PK_i^* = \{PK_{i,1}, PK_{i,2}, \cdots, PK_{i,z}\}$, where $PK_{i,l} = g^{k_{i,l}}$ and $l \in \{1, 2, \cdots, z\}$.

Table 5.1: The working flow of ECPS-CPPA protocol for vehicle

Stages	TA	Vehicle V_i	RSU_j
System Initialization	TA selects two master keys $a, b \in Z_q^*$ TA computes $A_{pub} = g^a$ and $B_{pub} = g^b$ as public keys TA generates and broadcasts system parameters $Params = \{q, G, g, A_{pub}, B_{pub}, H_0, H_1, H_2, H_3\}$		
Vehicle Enrollment	TA $\xleftarrow{\{ID_i\},\ offline}$ V_i TA selects random numbers $\{k_{i,1}, \cdots, k_{i,z}\} \in Z_q^*$ TA computes $PK_{i,l} = g^{k_{i,l}}$, $l \in \{1, \cdots, z\}$ TA computes $PID_{i,l} = ID_i \oplus H_0(B_{pub}^{k_{i,1}}, PK_{i,l})$, $l \in \{1, \cdots, z\}$ TA computes $sk_{i,l} = a \cdot H_1(PK_{i,l}, PID_{i,l}) + k_{i,l}$, $l \in \{1, \cdots, z\}$. TA $\xrightarrow{\{Params, PID_i^*, SK_i^*, PK_i^*\},\ offline}$ V_i V_i stores $\{Params, PID_i^*, SK_i^*, PK_i^*\}$ into its TPD		
Signature Generation		V_i selects $PID_{i,l}$, $sk_{i,l}$, $PK_{i,l}$ V_i randomly chooses $r_i \in Z_q^*$ V_i computes $R_i = g^{r_i}$ V_i generates the current timestamp T_i V_i computes $h_i = H_2(M_i, R_i, T_i) \in Z_q^*$ V_i generates the signature $Sig_i = (r_i - sk_{i,l} \cdot h_i)$ V_i $\xrightarrow{Msgs=\{M_i, PID_{i,l}, PK_{i,l}, R_i, T_i, Sig_i\}}$ RSU_j	
Single Authentication			RSU_j checks whether T_i is fresh RSU_j computes $H_1(PK_{i,l}, PID_{i,l})$ RSU_j computes $h_i = H_2(M_i, R_i, T_i) \in Z_q^*$ RSU_j checks if $g^{Sig_i} \cdot A_{pub}^{H_1(PK_{i,l}, PID_{i,l}) \cdot h_i} \cdot PK_{i,l}^{h_i} = R_i$ holds
Batch Authentication			Vehicles $\xrightarrow{Msgs_1, Msgs_2, \cdots, Msgs_n}$ RSU_j RSU_j checks whether $\{T_1, T_2, \cdots, T_n\}$ are fresh RSU_j computes $H_1(PK_{i,l}, PID_{i,l})$ for $i = 1, \cdots, n$ RSU_j computes $h_i = H_2(M_i, R_i, T_i) \in Z_q^*$ for $i = 1, \cdots, n$ RSU_j checks if $g^{\sum_{i=1}^{n}(\varrho_i \cdot Sig_i)} \cdot A_{pub}^{\sum_{i=1}^{n}(\varrho_i \cdot H_1(PK_{i,l}, PID_{i,l}) \cdot h_i)} \cdot \prod_{i=1}^{n} PK_{i,l}^{(\varrho_i \cdot h_i)} =? \prod_{i=1}^{n} R_i^{\varrho_i}$
Traceability	TA receives reported $Msgs = \{M_i, PID_{i,l}, PK_{i,l}, R_i, T_i, Sig_i\}$ TA computes $ID_i = PID_{i,l} \oplus H_1(PK_{i,l}^b, PK_{i,l})$		

2. TA generates a group of pseudo-IDs for V_i as $PID_i^* = \{PID_{i,1}, PID_{i,2}, \cdots, PID_{i,z}\}$, where $PID_{i,l} = ID_i \oplus H_0(B_{pub}^{k_{i,1}}, PK_{i,l})$ and $l \in \{1, 2, \cdots, z\}$. Hence, the real identity ID_i of vehicle V_i is concealed in the pseudo-IDs PID_i^*.

3. After computing the PID_i^*, TA computes private keys $SK_i^* = \{sk_{i,1}, sk_{i,2}, \cdots, sk_{i,z}\}$, where $sk_{i,l} = a \cdot H_1(PK_{i,l}, PID_{i,l}) + k_{i,l}$ and $l \in$

$\{1, 2, \cdots, z\}$.

4. TA sends system parameters $Params$ and z triple sets of $\{PID_i^*, SK_i^*, PK_i^*\}$ to vehicle V_i via a secure channel delivering a TPD for V_i. We assume that every vehicle has installed an intrusion detection system which will alarm the vehicle's owner when the adversary $\mathcal{A}$ is trying to intrude the vehicle's TPD. We also assume that the adversary $\mathcal{A}$ can extract the information inside the TPD of the vehicle registered by $\mathcal{A}$.

Vehicle Message Signing: The signature on one traffic-related message M_i by V_i is explained as below.

1. V_i randomly picks a triple of ($PID_{i,l}$, $sk_{i,l}$, $PK_{i,l}$) from the sets $(PID_i^*, SK_i^*, PK_i^*)$ separately. Then, V_i selects a random $r_i \in Z_q^*$ and calculates $R_i = g^{r_i}$, $h_i = H_2(M_i, R_i, T_i) \in Z_q^*$, $Sig_i = (r_i - sk_{i,l} \cdot h_i)$, wherethe generation of Sig_i is based on [51], and T_i is the current timestamp.

2. Then, V_i issues the signature message $Msgs = \{M_i, PID_{i,l}, PK_{i,l}, R_i, T_i, Sig_i\}$ to a nearby RSU or other vehicles.

Single Message Verification: Once the receiver has received a single message signed by V_i, it will verify the message as follows.

1. After receiving $Msgs = \{M_i, PID_{i,l}, PK_{i,l}, R_i, T_i, Sig_i\}$ signed by V_i, the receiver checks the freshness of timestamp T_i. The verifier drops the message if it is not fresh.

2. If T_i is valid, the receiver then computes $H_1(PK_{i,l}, PID_{i,l})$, $h_i = H_2(M_i, R_i, T_i) \in Z_q^*$ and verifies whether $g^{Sig_i} \cdot A_{pub}^{H_1(PK_{i,l}, PID_{i,l}) \cdot h_i} \cdot PK_{i,l}^{h_i} = R_i$. If the equation is satisfied, then the receiver accepts the validity of the message M_i; otherwise, the receiver drops it.

Batch Messages Verification: Upon receiving n messages $\{M_1, PID_{1,l}, PK_{1,l}, R_1, T_1, Sig_1\}$, $\{M_2, PID_{2,l}, PK_{2,l}, R_2, T_2, Sig_2\}$, $\cdots$, $\{M_n, PID_{n,l}, PK_{n,l}, R_n, T_n, Sig_n\}$ simultaneously, the receiver uses $Params = \{q, G, g, A_{pub}, B_{pub}, H_0, H_1, H_2\}$ to authenticate batch messages, as follows.

1. The receiver checks the freshness of $\{T_1, T_2, \cdots, T_n\}$, and drops the messages

that are not fresh.

2. The receiver randomly selects n numbers $\{\varrho_1, \varrho_2, \cdots, \varrho_n\}$, where $\varrho_i \in_R [1, 2^m]$ for $i = 1, 2, \cdots, n$ and $m = 80$ is typically adequate [48,59,60].

3. The receiver computes $H_1(PK_{i,l}, PID_{i,l})$, $h_i = H_2(M_i, R_i, T_i) \in Z_q^*$ for $i \in \{1, 2, \cdots, n\}$ and checks whether the below verification equation holds.

$$g^{\sum_{i=1}^{n}(\varrho_i \cdot Sig_i)} \cdot A_{pub}^{\sum_{i=1}^{n}(\varrho_i \cdot H_1(PK_{i,l},PID_{i,l}) \cdot h_i)} \cdot \prod_{i=1}^{n} PK_{i,l}^{(\varrho_i \cdot h_i)} = \prod_{i=1}^{n} R_i^{\varrho_i}$$

   If it is equal, then the receiver accepts the messages; otherwise, the receiver rejects the messages.

   The correctness of the batch messages verification is demonstrated as follows:

$$g^{\sum_{i=1}^{n}(\varrho_i \cdot Sig_i)} \cdot A_{pub}^{\sum_{i=1}^{n}(\varrho_i \cdot H_1(PK_{i,l},PID_{i,l}) \cdot h_i)} \cdot \prod_{i=1}^{n} PK_{i,l}^{(\varrho_i \cdot h_i)}$$

$$= \prod_{i=1}^{n}(g^{(\varrho_i \cdot Sig_i)} \cdot A_{pub}^{(\varrho_i \cdot H_1(PK_{i,l},PID_{i,l}) \cdot h_i)} \cdot PK_{i,l}^{(\varrho_i \cdot h_i)})$$

$$= \prod_{i=1}^{n}(g^{(\varrho_i \cdot (r_i - sk_{i,l} \cdot h_i))} \cdot \left(g^a\right)^{(\varrho_i \cdot H_1(PK_{i,l},PID_{i,l}) \cdot h_i)} \cdot \left(g^{k_{i,l}}\right)^{(\varrho_i \cdot h_i)})$$

$$= \prod_{i=1}^{n} g^{\varrho_i \cdot (r_i - sk_{i,l} \cdot h_i + a \cdot H_1(PK_{i,l},PID_{i,l}) \cdot h_i + k_{i,l} \cdot h_i)}$$
$$= \prod_{i=1}^{n} g^{\varrho_i \cdot (r_i - (a \cdot H_1(PK_{i,l},PID_{i,l}) + k_{i,l}) \cdot h_i + a \cdot H_1(PK_{i,l},PID_{i,l}) \cdot h_i + k_{i,l} \cdot h_i)}$$
$$= \prod_{i=1}^{n} g^{\varrho_i \cdot (r_i - (a \cdot H_1(PK_{i,l},PID_{i,l}) + k_{i,l}) \cdot h_i + a \cdot H_1(PK_{i,l},PID_{i,l}) \cdot h_i + k_{i,l} \cdot h_i)}$$

$$= \prod_{i=1}^{n} g^{\varrho_i \cdot (r_i - a \cdot H_1(PK_{i,l},PID_{i,l}) \cdot h_i - k_{i,l} \cdot h_i + a \cdot H_1(PK_{i,l},PID_{i,l}) \cdot h_i + k_{i,l} \cdot h_i)}$$

$$= \prod_{i=1}^{n} g^{\varrho_i \cdot (r_i)}$$

$$= \prod_{i=1}^{n} R_i^{\varrho_i}$$

5.3.2 ECPS-CPPA Protocol for RSU

The details of the ECPS-CPPA protocol for the RSU are shown as follows (see also Table 5.2), where this subsection omits the system setup stage since it is already depicted in 5.3.1.

Table 5.2: The working flow of ECPS-CPPA protocol for RSU

Stages	TA	Vehicle V_i	RSU_j
RSU Enrollment	TA $\xleftarrow{\{ID_{rsu_j}\},\ offline}$ RSU_j TA selects random numbers $\{x_{j,1},\cdots,x_{j,z}\}\in Z_q^*$ TA computes $Y_{j,l}=g^{x_{j,l}}$, $l\in\{1,\cdots,z\}$ TA computes $rsk_{j,l}=a\cdot H_1(Y_{j,l},RID_j)+x_{j,l}$, $l\in\{1,\cdots,z\}$. TA $\xrightarrow{\{Params,RID_j,RSK_j^*,Y_j^*\},\ offline}$ RSU_j		RSU_j stores $\{Params,RID_j,RSK_j^*,Y_j^*\}$ into its TPD
Signature Generation			RSU_j selects $rsk_{j,l}$, $Y_{j,l}$ RSU_j randomly chooses $w_j\in Z_q^*$ RSU_j computes $W_j=g^{w_j}$ RSU_j generates the current timestamp T_j RSU_j computes $h_j=H_2(M_j,W_j,T_j)\in Z_q^*$ RSU_j generates the signature $Rsig_j=(w_j-rsk_{j,l}\cdot h_j)$ V_i $\xleftarrow{Msgs=\{M_j,RID_j,Y_{j,l},W_j,T_j,Rsig_j\}}$ RSU_j
Single Authentication		V_i checks whether T_j is fresh V_i computes $H_1(Y_{j,l},RID_j)$ V_i computes $h_j=H_2(M_j,W_j,T_j)\in Z_q^*$ V_i checks if $g^{Rsig_j}\cdot A_{pub}^{H_1(Y_{j,l},RID_j)\cdot h_j}\cdot Y_{j,l}^{h_j}=W_j$ holds	
Batch Authentication		V_i $\xrightarrow{Msgs_1,Msgs_2,\cdots,Msgs_t}$ RSU_j V_i checks whether $\{T_1,T_2,\cdots,T_t\}$ are fresh V_i computes $H_1(Y_{j,l},RID_j)$ for $i=1,\cdots,t$ V_i computes $h_j=H_2(M_j,W_j,T_j)\in Z_q^*$ for $i=1,\cdots,t$ V_i checks if $g^{\sum_{j=1}^{n}(\varsigma_j\cdot Rsig_j)}\cdot A_{pub}^{\sum_{j=1}^{n}(\varsigma_j\cdot H_1(Y_{j,l},RID_j)\cdot h_j)}\cdot\prod_{j=1}^{n}Y_{j,l}^{(\varsigma_j\cdot h_j)}=\prod_{j=1}^{n}W_j^{\varsigma_j}$	

Enrollment for RSU: TA generates a unique identity RID_j for each RSU, and computes the private key for RSU as follows.

1. For a given RSU's identity RID_j, TA selects a group of private random numbers $\{x_{j,1},x_{j,2},\cdots,x_{j,z}\}\in Z_q^*$ and computes the corresponding public values $Y_j^*=\{Y_{j,1},Y_{j,2},\cdots,Y_{j,z}\}$, where $Y_{j,l}=g^{x_{j,l}}$ and $l\in\{1,2,\cdots,z\}$.

2. TA computes private keys $RSK_j^*=\{rsk_{j,1},rsk_{j,2},\cdots,rsk_{j,z}\}$, where $rsk_{j,l}=a\cdot H_1(Y_{j,l},RID_j)+x_{j,l}$ and $l\in\{1,2,\cdots,z\}$.

3. Finally, the TA sends $Params$ and $\{RID_j,RSK_j^*,Y_j^*\}$ to RSU via a secure channel. Then, RSU stores $\{RSK_j^*,Y_j^*\}$ with its corresponding identity RID_j into its storage memory.

RSU Message Signing: The signature on a traffic-related message M_j generated by the RSU is as follows:

1. RSU chooses a private key $rsk_{j,l}$ from the set RSK_j^*, a corresponding $Y_{j,l}$ from the set Y_j^*, a random $w_j \in Z_q^*$ and computes $W_j = g^{w_j}$, $h_j = H_2(M_j, W_j, T_j) \in Z_q^*$, and $Rsig_j = (w_j - rsk_{j,l} \cdot h_j)$, whereby T_j is the current timestamp which supports the freshness of a valid signed message.

2. Then, RSU broadcasts the signature message $Msgs = \{M_j, RID_j, Y_{j,l}, W_j, T_j, Rsig_j\}$ to nearby vehicles.

Single Message Verification: V_i will have to verify the signed message from RSU in order to ensure the legitimacy of RSU.

1. After receiving $Msgs = \{M_j, RID_j, Y_{j,l}, W_j, T_j, Rsig_j\}$ signed by the RSU, V_i checks the freshness of timestamp T_j and drops the message if T_j is not fresh.

2. If T_j is valid, then V_i computes $h_j = H_2(M_j, W_j, T_j) \in Z_q^*$, $H_1(Y_{j,l}, RID_j)$ and verifies whether $g^{Rsig_j} \cdot A_{pub}^{H_1(Y_{j,l}, RID_j) \cdot h_j} \cdot Y_{j,l}^{h_j} = W_j$. If the equation is satisfied, then V_i accepts the validity of the message M_j; otherwise, V_i rejects it.

Batch Messages Verification: After receiving t messages $\{M_1, RID_1, Y_{1,l}, W_1, T_1, Rsig_1\}$, $\{M_2, RID_2, Y_{2,l}, W_2, T_2, Rsig_2\}$, $\cdots$, $\{M_t, RID_t, Y_{t,l}, W_t, T_t, Rsig_t\}$ simultaneously, the vehicle verifies them using the following steps.

1. The vehicle checks the freshness of $\{T_1, T_2, \cdots, T_t\}$, and rejects the messages if some of them are not fresh.

2. The vehicle randomly selects t numbers $\{\varsigma_1, \varsigma_2, \cdots, \varsigma_t\}$, where $\varsigma_j \in_R [1, 2^m]$ for $j = 1, 2, \cdots, t$ and $m = 80$ is typically adequate [48,59,60].

3. The vehicle computes $h_j = H_2(M_j, W_j, T_j) \in Z_q^*$, $H_1(Y_{j,l}, RID_j)$ for $j \in \{1, 2, \cdots, t\}$ and checks whether the below verification equation holds.

$$g^{\sum_{j=1}^{n}(\varsigma_j \cdot Rsig_j)} \cdot A_{pub}^{\sum_{j=1}^{n}(\varsigma_j \cdot H_1(Y_{j,l}, RID_j) \cdot h_j)} \cdot \prod_{j=1}^{n} Y_{j,l}^{(\varsigma_j \cdot h_j)} = \prod_{j=1}^{n} W_j^{\varsigma_j}$$

If it is equal, then the vehicle accepts the messages; otherwise, the vehicle rejects the messages.

The correctness of the batch messages verification is demonstrated, as follows.

$$g^{\sum_{j=1}^{n}(\varsigma_j\cdot Rsig_j)}\cdot A_{pub}^{\sum_{j=1}^{n}(\varsigma_j\cdot H_1(Y_{j,l},RID_j)\cdot h_j)}\cdot\prod_{j=1}^{n}Y_{j,l}^{(\varsigma_j\cdot h_j)}$$

$$=\prod_{j=1}^{n}(g^{(\varsigma_j\cdot Rsig_j)}\cdot A_{pub}^{(\varsigma_j\cdot H_1(Y_{j,l},RID_j)\cdot h_j)}\cdot PK_{i,l}^{(\varsigma_j\cdot h_j)})$$

$$=\prod_{j=1}^{n}(g^{(\varsigma_j\cdot(r_i-rsk_{j,l}\cdot h_j))}\cdot(g^{a})^{(\varsigma_j\cdot H_1(Y_{j,l},RID_j)\cdot h_j)}\cdot(g^{x_{j,l}})^{(\varsigma_j\cdot h_j)})$$

$$=\prod_{j=1}^{n}g^{\varsigma_j\cdot(r_i-rsk_{j,l}\cdot h_j+a\cdot H_1(Y_{j,l},RID_j)\cdot h_j+x_{j,l}\cdot h_j)}$$

$$=\prod_{j=1}^{n}g^{\varsigma_j\cdot(r_i-(a\cdot H_1(Y_{j,l},RID_j)+x_{j,l})\cdot h_j+a\cdot H_1(Y_{j,l},RID_j)\cdot h_j+x_{j,l}\cdot h_j)}$$

$$=\prod_{j=1}^{n}g^{\varsigma_j\cdot(r_i-(a\cdot H_1(Y_{j,l},RID_j)+x_{j,l})\cdot h_j+a\cdot H_1(Y_{j,l},RID_j)\cdot h_j+x_{j,l}\cdot h_j)}$$

$$=\prod_{j=1}^{n}g^{\varsigma_j\cdot(r_i-a\cdot H_1(Y_{j,l},RID_j)\cdot h_j-x_{j,l}\cdot h_j+a\cdot H_1(Y_{j,l},RID_j)\cdot h_j+x_{j,l}\cdot h_j)}$$

$$=\prod_{j=1}^{n}g^{\varsigma_j\cdot(r_i)}$$

$$=\prod_{j=1}^{n}R_i^{\varsigma_j}$$

5.4 Security and Soundness Proofs

In this section, it will demonstrate that the ECPS-CPPA protocol for vehicle achieves design goals outlined in subsection 2.2 of Chapter 2. We does not give further analysis on the ECPS-CPPA protocol for RSU, since the process of proof and analysis is similar to that of the ECPS-CPPA protocol for vehicle as follows.

5.4.1 Security Model

The security model for the ECPS-CPPA protocol is as same as that in subsection 4.3.1 of Chapter 4 [27].

5.4.2 Provable Security

Let the function $Adv_{\Omega_2,\mathcal{A}}^{ECPS-CPPA}$ denote the advantage of $\mathcal{A}$ in breaking conditional privacy-preserving authentication of the presented ECPS-CPPA solution Ω_2.
Definition 2. *The ECPS-CPPA solution Ω_2 is chosen-identity and chosen-message secure, if for any polynomial-time adversary $\mathcal{A}$, the function $Adv_{\Omega_2,\mathcal{A}}^{ECPS-CPPA}$ is negligible.*

Based on Definition 2, the chosen-identity and chosen-message security of the ECPS-CPPA solution using random oracles are proved.
Theorem 2. *Assuming that the underlying DL problem is intractable, the ECPS-CPPA solution for VANETs is secure in the random oracle model.*

Proof. Assume that a polynomial-time adversary $\mathcal{A}$ can fabricate a valid signature message $Msgs = \{M_i, PID_i, PK_i, R_i, T_i, Sig_i\}$ by a non-negligible advantage ε, then the challenger $\mathcal{I}$ can resolve the DL problem with a non-negligible advantage through executing the $\mathcal{A}$ as a subroutine. Let $A_{pub} = g^a$ be an instance of the DL problem, and the aim of the $\mathcal{I}$ is to compute a. First, I issues $Params = \{q, G, g, A_{pub}, B_{pub}, H_0, H_1, H_2, H_3\}$ to $\mathcal{A}$, and $\mathcal{A}$ performs random oracle queries adaptively simulated by $\mathcal{I}$ as below.

$\mathbf{H_0}$ **Oracle**: $\mathcal{I}$ maintains a list L_{H_0} in the form of $\{G, G, \pi_0\}$, which is empty initially. When $\mathcal{A}$ issues a query $\{\Theta, B_{pub}\}$ to $\mathcal{I}$, $\mathcal{I}$ checks whether the tuple $\{G, G, \pi_0\}$ is in the list L_{H_0}. If so, $\mathcal{I}$ issues $\pi_0 = H_0(\Theta, B_{pub})$ to $\mathcal{A}$, otherwise, $\mathcal{I}$ selects a random nonce $\pi_0 \in Z_p$, issues $\pi_0 = H_0(\Theta, B_{pub})$ to $\mathcal{A}$ and appends $\{G, G, \pi_0\}$ to the list L_{H_0}.

$\mathbf{H_1}$ **Oracle**: $\mathcal{I}$ maintains a list L_{H_1} in the form of $\{G, PID_i, \pi_1\}$, which is empty initially. When $\mathcal{A}$ issues a query Υ to $\mathcal{I}$, $\mathcal{I}$ checks whether the tuple $\{G, PID_i, \pi_1\}$ is in the list L_{H_1}. If so, $\mathcal{I}$ issues $\pi_1 = H_1(\Upsilon)$ to $\mathcal{A}$, otherwise, $\mathcal{I}$ selects a random nonce $\pi_1 \in Z_p$, issues $\pi_1 = H_1(\Upsilon)$ to $\mathcal{A}$ and appends $\{G, PID_i, \pi_1\}$ to the list L_{H_1}.

$\mathbf{H_2}$ **Oracle**: $\mathcal{I}$ maintains a list L_{H_2} in the form of $\{M_i, R_i, T_i, \pi_2\}$, which is empty initially. When $\mathcal{A}$ issues a query $\{M_i, R_i, T_i\}$ to $\mathcal{I}$, $\mathcal{I}$ checks whether the tuple $\{M_i, R_i, T_i, \pi_2\}$ is in the list L_{H_2}. If so, $\mathcal{I}$ issues $\pi_2 = H_2(M_i, R_i, T_i)$ to $\mathcal{A}$, otherwise, $\mathcal{I}$ selects a random nonce $\pi_2 \in Z_p$, issues $\pi_2 = H_2(M_i, R_i, T_i)$ to $\mathcal{A}$ and appends $\{M_i, R_i, T_i, \pi_2\}$ to the list L_{H_2}.

GenerateVehicle Oracle: $\mathcal{I}$ maintains a list $L_{vehicle}$ in the form of $\{ID_i, k_i, PID_i, sk_i, PK_i\}$ which is empty initially. Once $\mathcal{A}$ sends this query to $\mathcal{I}$, $\mathcal{A}$ checks whether the tuple $\{ID_i, k_i, PID_i, sk_i, PK_i\}$ is in the list $L_{vehicle}$. If so, $\mathcal{I}$ returns PK_i to $\mathcal{A}$; otherwise $\mathcal{I}$ executes the steps as below.

1) If $ID_i = ID_i^*$, $\mathcal{I}$ selects three random numbers k_i, π_0 and π_1, computes $PK_i = g^{k_i}$ and holds $\{PID_i, SK_i\}$. $\mathcal{I}$ stores $\{ID_i, k_i, PID_i, sk_i, PK_i\}$, $\{G, G, \pi_0\}$ and $\{G, PID_i, \pi_1\}$ in the lists $L_{vehicle}$, L_{H_0} and L_{H_1} respectively. At last, $\mathcal{I}$ returns PK_i to $\mathcal{A}$.

2) If $ID_i \neq ID_i^*$, $\mathcal{I}$ selects three random numbers k_i, π_0 and π_1, computes $PK_i = g^{k_i}$, $PID_i = ID_i \oplus \pi_0$, $sk_i = a \cdot \pi_1 + k_i$. $\mathcal{I}$ stores $\{ID_i, k_i, PID_i, sk_i, PK_i\}$, $\{G, G, \pi_0\}$ and $\{G, PID_i, \pi_1\}$ in the lists $L_{vehicle}$, L_{H_0} and L_{H_1} respectively and finally returns PK_i to $\mathcal{A}$.

CorruptVehicle Oracle: $\mathcal{A}$ inquiries $\{ID_i, k_i, PID_i, sk_i, PK_i\}$ from $L_{vehicle}$ and, $\mathcal{I}$ issues $\{PID_i, sk_i\}$ to $\mathcal{A}$.

Signature Oracle: Upon receiving $\mathcal{A}$'s query with message M_i and pseudo-identity PID_i, $\mathcal{I}$ selects two random numbers r_i, π_2, π_3 and computes $R_i = g^{r_i}$ and $Sig_i = (r_i - sk_i \cdot \pi_2)$. $\mathcal{I}$ stores $\{M_i, R_i, T_i, \pi_2\}$ to the list L_{H_2}, and returns the signature message $Msgs = \{M_i, PID_i, PK_i, R_i, T_i, Sig_i\}$ to $\mathcal{A}$.

Finally, $\mathcal{A}$ outputs a signature message $\{M_i, PID_i, PK_i, R_i, T_i, Sig_i\}$ to $\mathcal{I}$ with PID_i. If $PID_i \neq PID_i^*$, then $\mathcal{I}$ aborts the game. $\mathcal{I}$ checks whether the below equation is correct.

$$g^{Sig_i} \cdot A_{pub}^{H_1(PK_i,PID_i) \cdot h_i} \cdot PK_i^{h_i} = R_i \tag{5.1}$$

If it is not correct, then $\mathcal{I}$ interrupts the game. Based on the forking lemma in [61], if the challenger repeats the procedure with a different selection H_1, then $\mathcal{A}$ can output another legitimate signature message $\{M_i, PID_i, PK_i, R_i, T_i, Sig_i'\}$ with the advantage $\varepsilon' \geq \frac{1}{9}$. Thus, the following equation is obtained:

$$g^{Sig_i'} \cdot A_{pub}^{H_1(PK_i,PID_i)' \cdot h_i} \cdot PK_i^{h_i} = R_i \tag{5.2}$$

According to the above two equations, the following equations are obtained:

$$g^{Sig_i - Sig_i'} = A_{pub}^{(H_1(PK_i,PID_i)' - H_1(PK_i,PID_i))\cdot h_i} \tag{5.3}$$

$$Sig_i - Sig_i' = a \cdot (H_1(PK_i, PID_i)' - H_1(PK_i, PID_i)) \cdot h_i \tag{5.4}$$

According to Equations 5.4, $\mathcal{I}$ outputs $(Sig_i - Sig_i') \cdot ((H_1(PK_i, PID_i)' - H_1(PK_i, PID_i)) \cdot h_i)^{-1}$ as the result of the DL problem. The advantage that $\mathcal{I}$ solves the DL problem can be analyzed via the following events [27].

1) E_{pid} denotes the event that PID_i and PID_i^* are equal.

2) E_{forge} denotes the event that $\mathcal{A}$ can forge two legitimate signatures.

Let N_{H_1} denotes the number of H_1 oracle queries executed in the above experiments. Thus, it can be got that $Prob[E_{pid}] = \frac{1}{N_{H_1}}$, $Prob[E_{forge}|E_{pid}] \geq \frac{1}{9} \cdot \varepsilon$ and the advantage that $\mathcal{A}$ can solve the DL problem is as blow.

$$\begin{aligned} Prob[E_{forge} \wedge E_{pid}] &= Prob[E_{forge}|E_{pid}] \cdot Prob[E_{pid}] \\ &= \geq \frac{1}{9} \cdot \varepsilon \cdot \frac{1}{N_{H_1}} = \frac{\varepsilon}{9N_{H_1}}. \end{aligned}$$

Therefore $\mathcal{I}$ solves the DL problem with a non-negligible advantage $\frac{\varepsilon}{9N_{H_1}}$ due to the non-negligible ε and bounded N_{H_1}. However, this is a contradiction with the hardness of the DL problem in G. Consequently, this completes the proof.

□

5.4.3 Security and Attributes Analysis

Identity Privacy Preservation: In the enrollment stage, the vehicle user's identity ID_i is masked in the form of $PID_{i,l} = ID_i \oplus H_1(B_{pub}^{k_{i,1}}, PK_{i,l})$ by the TA, where $PK_{i,l} = g^{k_{i,l}}$ generated by TA, and $k_{i,1}$ is a random number chosen by TA. To reveal the vehicle user's identity ID_i from PID_i, $\mathcal{A}$ needs to compute $B_{pub}^{k_{i,l}} = g^{b \cdot k_{i,l}}$ based on $PK_{i,l} = g^{k_{i,l}}$ and $B_{pub} = g^b$. This, however, is contradictive with the hardness of CDH problem. Thus, our ECPS-CPPA protocol for VANETs safeguards the user′s identity privacy.

Message Authentication and Integrity: Upon receiving $Msgs = \{M_i, PID_{i,l}, PK_{i,l}, R_i, T_i, Sig_i\}$ from u_i, the receiver authenticates the equation $g^{Sig_i} \cdot A_{pub}^{H_1(PK_{i,l}, PID_{i,l}) \cdot h_i} \cdot PK_{i,l}^{h_i} = R_i$ so as to inspect the message′s legitimacy. According to Theorem 2 in Section 5.4.2, any polynomial-time attacker $\mathcal{A}$ cannot counterfeit a legal signature upon traffic message because of the difficult DL problem. Therefore, $\mathcal{A}$ is not capable of extracting the master key of TA and generating valid signing for message verification.

Traceability: In the enrollment stage, the vehicle′s genuine identity is masked in the pseudo-IDs $PID_i^* = \{PID_{i,1}, PID_{i,2}, \cdots, PID_{i,z}\}$, where $PID_{i,l} = ID_i \oplus H_1(B_{pub}^{k_{i,1}}, PK_{i,l})$ and $l \in \{1, 2, \cdots, z\}$. By knowing the master secret key b of the VANETs system, TA could extract the real identity $ID_i = PID_{i,l} \oplus H_0(PK_{i,l}^b, PK_{i,l})$. Consequently, the function of traceability is provided by the ECPS-CPPA solution.

Unlinkability: TA selects a group of private random numbers $\{k_{i,1}, \cdots, k_{i,z}\} \in Z_q^*$ in the enrollment stage and the vehicle also chooses random $r_i \in Z_q^*$ in the message signing stage, where $PID_i^* = \{PID_{i,1}, \cdots, PID_{i,z}\}$, $PID_{i,l} = ID_i \oplus H_1(B_{pub}^{k_{i,1}}, PK_{i,l})$, $SK_i^* = \{sk_{i,1}, \cdots, sk_{i,z}\}$, $R_i = g^{r_i}$, $h_i = H_2(M_i, R_i, T_i) \in Z_q^*$, $Sig_i = (r_i - sk_{i,l} \cdot h_i)$. Due to the randomness of $k_{i,1}$ and r_i, the vehicle could generate random identities and signatures from which the adversary cannot find the connection between two anonymous identities or two signatures (i.e. not able to determine whether they are sent by the same vehicle). Therefore, the ECPS-CPPA protocol achieves unlinkability.

Secerecy of Master Key: In the ECPS-CPPA protocol, the TPD does not store the master key a directly. The secret key is generated as $sk_{i,l} = a \cdot H_1(PK_{i,l}, PID_{i,l}) + k_{i,l}$, where the master key a is protected by the random selected number $k_{i,l}$. Even though the adversary $\mathcal{A}$ have maliciously extracted $\{sk_{i,l}, PK_{i,l}, PID_{i,l}\}$, $\mathcal{A}$ cannot compute a out since it is impossible to determine two variables in one equation.

Resilient to Message Modification Attack: Every TPD of vehicle broadcasts the message tuple $Msgs = \{M_i, PID_{i,l}, PK_{i,l}, R_i, T_i, Sig_i\}$ to nearby RSUs and other vehicles. $\mathcal{A}$ has the capability to change the content of M_i after eavesdropping on the wireless medium. In order to protect the integrity of the message, a vehicle's signature on M_i is generated as $Sig_i = (r_i - sk_{i,l} \cdot h_i)$, where $h_i = H_2(M_i, R_i, T_i) \in Z_q^*$. Since the private key $sk_{i,l}$ is only known by the particular vehicle, no attacker can generate a valid signature. Besides, the private key is changed periodically. Thus, the ECPS-CPPA solution for VANETs is secure against message modification attacks.

Resilient to Impersonation Attack: To mount masquerading attacks, $\mathcal{A}$ is supposed to produce legal $Msgs = \{M_i, PID_{i,l}, PK_{i,l}, R_i, T_i, Sig_i\}$. According to Theorem 2, $\mathcal{A}$ cannot counterfeit a legal signature successfully. The receivers could inspect the legitimacy of the received message by authenticating the equation in 5.3.1. Thus, the ECPS-CPPA solution for VANETs could resist the impersonation attack.

Resilient to Replay Attack: Timestamp T_i is involved in the signature tuple $Msgs = \{M_i, PID_{i,l}, PK_{i,l}, R_i, T_i, Sig_i\}$ and is also included in the generation of signature Sig_i. Therefore, the receiver could check the replay attack once T_i is overtime. Therefore, the ECPS-CPPA protocolfor VANETs is able to resist replay attack.

Full Batch Authentication: According as the batch authentication in Section 5.3.1, upon receiving n messages $Msgs = \{M_i, PID_{i,l}, PK_{i,l}, R_i, T_i, Sig_i\}$ for $i = 1, \cdots, n$, from different vehicles simultaneously, the receivers are capable of authenticating the validity at one time.

No Map-to-Point Operation: The expensive and complicated map-to-point operation is also avoided in the ECPS-CPPA scheme for VANETs.

No Certificate Management: In the ECPS-CPPA solution for vehicle, neither vehicle nor RSUs store any certificates for message verification. The vehicle only needs to memorize the system parameters $Params$ and $\{PID_i^*, SK_i^*, PK_i^*\}$, where $PID_{i,l} = ID_i \oplus H_1(B_{pub}^{k_{i,1}}, PK_{i,l})$, $sk_{i,l} = a \cdot H_1(PK_{i,l}, PID_{i,l}) + k_{i,l}$, $PK_{i,l} = g^{k_{i,l}}$ for $l \in \{1, \cdots, z\}$ generated by the TA. Therefore, TA does not need to manage any certificate.

No Verifier Table: The presented ECPS-CPPA solution for VANETs is stolen verifier table attack resilience since there is no verifier table maintained by RSUs or vehicles.

Provable Security: The ECPS-CPPA scheme is proved securely under the random oracle model.

5.5 Summary

In this chapter, the proposed CPS-CPPA protocol for VANETs is analyzed. We have revealed previously unknown attacks against the CPS-CPPA protocol, and more importantly identified design flaws in this protocol. Specifically, in the CPS-CPPA protocol, the master key a is not practically protected, since the highly-motivated adversary can easily compute the master key a, and then mount the forged message attack and impersonation attack. To resolve these weaknesses, we presented an enhanced CPPA protocol for safety-related VANETs applications and then demonstrated the security of the ECPS-CPPA solution.

Chapter 6

Overheads Evaluation

In this chapter, it analyzes the performance of the proposed two solutions as well as those of [18, 20], in terms of computation and communication overheads.

6.1 Computation Overheads

Notations used are as follows:

1. $\hat{e} : G_1 \times G_1 \rightarrow G_2$ denotes a bilinear pairing.

2. T_{bp} denotes the run time required for a bilinear pairing operation $\hat{e}(\hat{U}, \hat{V})$, where $\hat{U}, \hat{V} \in G_1$.

3. T_{sm-bp} denotes the runtime for a scale multiplication operation about the bilinear pairing in G_1.

4. T_{pa-bp} denotes the runtime for a point addition operation about the bilinear pairing in G_1.

5. T_{sm-ecc} denotes the runtime for a scale multiplication operation about the Elliptic-Curve Cryptography (ECC) in an additive group G.

6. T_{pa-ecc} denotes the runtime for a point addition operation about the ECC in an additive group G.

7. T_h denotes the time required for running a cryptographic hash function operation.

For a fair evaluation, the same run time in He et al.'s evaluation [18] is used – see Table 6.1. The above cryptographic operations are executed using MIRACL [62].The hardware platform is an Intel I7-4770 processor with 3.40 GHz clock frequency, 4 gigabytes memory and runs Windows 7 operating system [18].

Table 6.1: Run time of multiple cryptographic operations

Cryptographic Operation	**Running Time (milliseconds)**
T_{bp}	4.211 ms
T_{sm-bp}	1.709 ms
T_{pa-bp}	0.0071 ms
T_{sm-ecc}	0.442 ms
T_{pa-ecc}	0.0018 ms
T_h	0.0001 ms

Let MSG, SV and BV signify the message signature generation, single authentication, and batch authentication, separately.

Table 6.2 presents a comparative summary for the computation costs. In addition, Fig.6.1 visually illustrates the comparative results in MSG and SV stages and Fig.6.2 particularly shows the wholly comparative results in BV stage.

Within the MSG stage of Azees et al.'s protocol [20], the OBU executes 4 scalar multiplication operations about the bilinear pairing, 2 point addition operations about the bilinear pairing and 2 cryptographic hash function operations. Therefore, the runtime of this stage is $4T_{sm-bp}$ + $2T_{pa-bp}$ + $2T_h$ $\approx$ 6.8504 milliseconds (ms). Within the SV stage, the OBU executes 2 bilinear pairing operations, 5 scalar multiplication operations about the bilinear pairing, and 2 point addition operations about the bilinear pairing. Therefore, the runtime of this stage is $2T_{bp}$ + $5T_{sm-bp}$ + $2T_{pa-bp}$ $\approx$ 16.9812 ms. Within the BV stage, the OBU executes (n + 1) bilinear pairing operations, ($5n$) scalar multiplication operations about the bilinear pairing and ($2n$) point addition operations about the bilinear pairing. Hence, the runtime of this stage is (n + 1)T_{bp} + $(5n)T_{sm-bp}$ + $(2n)T_{pa-bp}$ $\approx$ 12.77 n + 4.211 ms.

Table 6.2: Computation costs: a comparative summary (Unit: millisecond)

Protocols	MSG **Stage**	SV **Stage**	BV **Stage**
He et al.'s protocol [18]	$3T_{sm-ecc}$ + $3T_h$ $\approx$ 1.3263	$3T_{sm-ecc}$ + $2T_{pa-ecc}$ + $2T_h$ $\approx$ 1.3298	$(2n$ + 2$)T_{sm-ecc}$ + $(2n)T_{pa-ecc}$ + $(2n)T_h$ $\approx$ 0.8878 n + 0.884
Azees et al.'s protocol [20]	$4T_{sm-bp}$ + $2T_{pa-bp}$ + $2T_h$ $\approx$ 6.8504	$2T_{bp}$ + $5T_{sm-bp}$ + $2T_{pa-bp}$ $\approx$ 16.9812	$(n$ + 1$)T_{bp}$ + $(5n)T_{sm-bp}$ + $(2n)T_{pa-bp}$ $\approx$ 12.77 n + 4.211
CPS-CPPA protocol	$1T_{sm-ecc}$ + $2T_h$ $\approx$ 0.4422	$3T_{sm-ecc}$ + $1T_{pa-ecc}$ + $2T_h$ $\approx$ 1.328	(n+2)T_{sm-ecc} + $(n)T_{pa-ecc}$ + $(2n)T_h$ $\approx$ 0.444 n + 0.884
ECPS-CPPA protocol	$1T_{sm-ecc}$ + $1T_h$ $\approx$ 0.4421	$3T_{sm-ecc}$ + $2T_{pa-ecc}$ + $2T_h$ $\approx$ 1.3298	(2n+2)T_{sm-ecc} + $(2n)T_{pa-ecc}$ + $(2n)T_h$ $\approx$ 0.8878 n + 0.884

Within the MSG stage of our ECPS-CPPA protocol, the OBU executes 1 scalar multiplication operation about the ECC, 1 cryptographic hash function operations. Hence, the runtime of this stage is $1T_{sm-ecc}$ + $1T_h$ $\approx$ 0.4421 ms. Within the SV stage, the OBU executes 3 scalar multiplication operations about the ECC, 2 point addition operations about the ECC and 2 cryptographic hash function operations. Thus, the runtime of this stage is $3T_{sm-ecc}$ + $2T_{pa-ecc}$ + $2T_h$ $\approx$ 1.3298 ms. Within the BV stage, the OBU executes (2n+2) scalar multiplication operations about the ECC, 2n point addition operations about the ECC and 2n cryptographic hash function operations. Thus, the runtime of this stage is (2n+2)T_{sm-ecc} + $(2n)T_{pa-ecc}$ + $(2n)T_h$ $\approx$ 0.8878 n + 0.884 ms. Thus, the computation overhead in the MSG, SV and BV stages of out ECPS-CPPA are lower than those of Azees et al.'s protocol (see Fig.6.1 and 6.2 detailly).

In a similar way, the computation overhead in the MSG stage of our ECPS-CPPA protocol are more favorable than that of [18] (see Fig.6.1), and the computation overheads in the SV and BV stages of our ECPS-CPPA protocol are equal to those of [18] respectively (see Fig.6.1, Fig. 6.2 detailly). Although our CPS-CPPA protocol owns the least computation overheads compared with our ECPS-CPPA protocol, it has a weakness issue about the master key in practice.

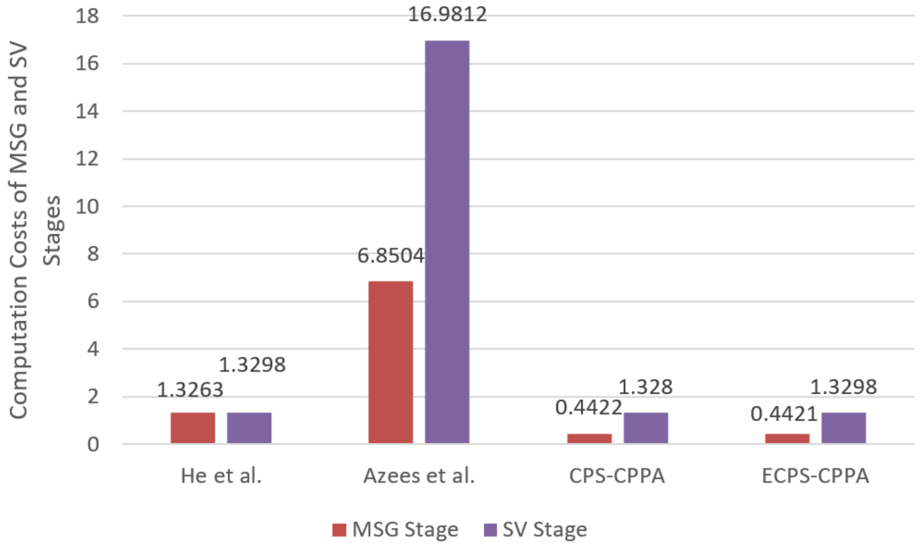

Figure 6.1: Computation costs of MSG and SV stages

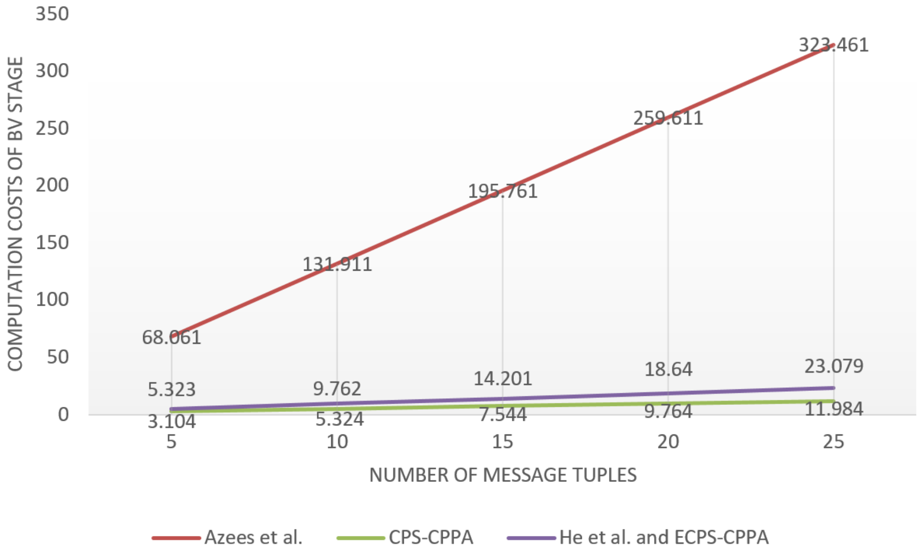

Figure 6.2: Computation costs of BV stage

6.2 Communication Costs

This subsection will evaluate the communication costs. Let the sizes of the element in G_1, the element in G, the element in Z_q^*, timestamp and the value of hash function be 128 bytes, 40 bytes, 20 bytes, 4 bytes and 20 bytes, respectively [18]. The messages of traffic status are not considered in the communication comparison, since they are similar in size. Table 6.3 gives a comparative summary.

Table 6.3: Communication costs: a comparative summary (Unit: byte)

Protocols	**Sending of one message**	**Sending of n messages**
He et al.'s protocol [18]	124	124n
Azees et al.'s protocol [20]	848	848n
CPS-CPPA protocol	124	124n
ECPS-CPPA protocol	124	124n

In He et al.'s solution [18], the vehicle transmits its signature messages $\{AID_i, T_i, R_i, M_i\}$ to the verifier, where $AID_i = \{AID_{i,1}, AID_{i,2}\}$, $AID_{i,1} \in G$, $AID_{i,2} \in Z_q$, $R_i \in G$, $\sigma_i \in Z_q$ and T_i is a timestamp. Thus, the communication overhead is 2 × 40 + 2 × 20 + 4 = 124 bytes. In Azees et al.'s protocol [20], the vehicle transmits the signature tuple $\{sig||Y_k||Cert_k\}$ to the verifier, where $Cert_k = \{Y_k||E_i||DID_{u_i}||\gamma_u||\gamma_v||c||\lambda||\sigma_1||\sigma_2\}$, $\{sig, E_i, DID_{u_i}, \gamma_u, \gamma_v, Y_k\} \in G_1$, $\{\lambda, \sigma_1, \sigma_2\} \in Z_q^*$, c is a hash value. Thus, the communication cost is 6 × 128 + 4 × 20 = 848 bytes. In our CPS-CPPA and ECPS-CPPA protocols, the vehicle sends its signature tuple $Msgs = \{M_i, PID_{i,l}, PK_{i,l}, R_i, T_i, Sig_i\}$ to the verifier, where $\{PK_{i,l}, R_i\} \in G$, t_i is the timestamp, and $\{PID_{i,l}, Sig_i\} \in Z_q^*$. Thus, the communication overhead is 2 × 40 + 2 × 20 + 4 = 124 bytes.

Therefore, our ECPS-CPPA protocol for VANETs owns a favorable communication overhead, compared with other protocols [18, 20].

6.3 Summary

In this chapter, we compared the proposed two CPPA protocols with the other two existing CPPA protocols on the aspects of computation overhead and communication overhead. The results show that our ECPS-CPPA protocol owns favorable computation and communication costs. In addition, our CPS-CPPA protocol owns the least computation overheads compared with our ECPS-CPPA protocol, but there is a weakness issue about the master key in practice.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

With the potential of VANETs in applications ranging from smart cities to smart campuses to battlefields, and so on, designing efficient security and privacy solutions for VANETs will be increasingly important. It is also important to study the soundness of proposed solutions to ensure that we are able to identify any vulnerabilities and limitations in these solutions prior to them being deployed in a real-world setting.

In this thesis, we studied the existing CPPA protocol for VANETs in order to improve the CPPA further. For example, we made the cryptanalysis on Azees et al.'s CPPA protocol and Zhang et al.'s CPPA protocol respectively in Chapter 3, which revealed previously unknown attacks (i.e. bogus message attack, framing attack, Sybil attack, and replay attack) against protocols. Specifically, in Azees et al.'s protocol, randomly-selected numbers are used to produce all other parameters without binding these numbers to an identity. Besides, there is no reliable public verification. Therefore, the highly-motivated adversary is easily capable of exploiting the weaknesses to mount the four attacks we showed in chapter 3. In Zhang et al.'s CPPA protocol, the RSU issues its secret key a_j to vehicles with a simple variation, which make the protocol not resilient with malicious attacks.

To solve the flaws of existing CPPA protocols, especially, those in the protocols of Azees et al. [20] and Zhang et al. [26]. Chapter 4 introduces a CPS-CPPA protocol for VANETs with the function of batch verification, which can be utilized in safety-related VANETs applications. It then proved the security of the proposed CPS-CPPA solution. However, we revealed previously unknown attacks against our CPS-CPPA protocol for VANETs, and more importantly identified design flaws in this protocol. Specifically, in the CPS-CPPA protocol, the master key a is not practically safeguarded, where the malicious adversary can easily compute the master key a from his/her registered vehicle's TPD, and then make the modification attack and impersonation attack successfully. To overcome these weaknesses, Chapter 5 presents an ECPS-CPPA protocol for safety-related VANETs applications and demonstrated the security of the ECPS-CPPA version. In addition, we compare the ECPS-CPPA protocol with the other two CPPA protocols on the aspects of communication and computation overheads .

7.2 Future Work

Upon the CPPA protocols, there are still some remaining issues that needed to be studied in the future. Firstly, in the ECPS-CPPA protocol, TA stores a limited number (i.e. z) of parameter triples in the vehicle's TPD, which means that the number of PIDs are limited. That is, when a vehicle has used up the PIDs for a long time, the driver has to apply new PIDs from TA. Otherwise, the vehicle is vulnerable to the PID linking attack, since the vehicle has been using the PIDs repetitively for a long time. So, is it possible to generate a random PID for every signature, which should also be traced by TA? If it is not possible to design such a CPPA protocol in principle, the efficient and secure PID changing strategies are necessary. Secondly, TA keeps two master keys a and b, but it is impossible for all the vehicles to register in one TA spot. So the management and transport of the master keys are an crucial issue.

This paper is mainly about the theory design for VANET, which is the first and fundamental step. In the future, the simulation before the real-world evaluation is very necessary. Then, a prototype of the presented solutions for real-world evaluation is going to be implemented in order to re-define the presented protocols, for example within a closed environment (e.g. within the campus grounds of the authors' institutions in Germany and U.S.).

Bibliography

[1] Hoa La Vinh and Ana Rosa Cavalli. Security attacks and solutions in vehicular ad hoc networks: a survey. *International journal on AdHoc networking systems (IJANS)*, 4(2):1–20, 2014.

[2] Fengzhong Qu, Zhihui Wu, Fei-Yue Wang, and Woong Cho. A security and privacy review of vanets. *IEEE Transactions on Intelligent Transportation Systems*, 16(6):2985–2996, 2015.

[3] Kyung-Ah Shim. Cpas: An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks. *IEEE Transactions on Vehicular Technology*, 61(4):1874–1883, 2012.

[4] Shiang-Feng Tzeng, Shi-Jinn Horng, Tianrui Li, Xian Wang, Po-Hsian Huang, and Muhammad Khurram Khan. Enhancing security and privacy for identity-based batch verification scheme in vanets. *IEEE Transactions on Vehicular Technology*, 66(4):3235–3248, 2017.

[5] Tat Wing Chim, Siu-Ming Yiu, Lucas CK Hui, and Victor OK Li. Specs: Secure and privacy enhancing communications schemes for vanets. *Ad Hoc Networks*, 9(2):189–203, 2011.

[6] Georgios Karagiannis, Onur Altintas, Eylem Ekici, Geert Heijenk, Boangoat Jarupan, Kenneth Lin, and Timothy Weil. Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. *IEEE communications surveys & tutorials*, 13(4):584–616, 2011.

[7] Frank Kargl, Panagiotis Papadimitratos, Levente Buttyan, Michael Müter,

Elmar Schoch, Bjorn Wiedersheim, Ta-Vinh Thong, Giorgio Calandriello, Albert Held, Antonio Kung, et al. Secure vehicular communication systems: implementation, performance, and research challenges. *IEEE Communications magazine*, 46(11):110–118, 2008.

[8] Abdelwahab Boualouache, Sidi-Mohammed Senouci, and Samira Moussaoui. A survey on pseudonym changing strategies for vehicular ad-hoc networks. *IEEE Communications Surveys & Tutorials*, 2017.

[9] Maxim Raya and Jean-Pierre Hubaux. Securing vehicular ad hoc networks. *Journal of computer security*, 15(1):39–68, 2007.

[10] Mohammad Reza Jabbarpour, Houman Zarrabi, Rashid Hafeez Khokhar, Shahaboddin Shamshirband, and Kim-Kwang Raymond Choo. Applications of computational intelligence in vehicle traffic congestion problem: a survey. *Soft Computing*, 22(7):2299–2320, 2018.

[11] Zhenyu Zhou, Caixia Gao, Chen Xu, Yan Zhang, Shahid Mumtaz, and Jonathan Rodriguez. Social big-data-based content dissemination in internet of vehicles. *IEEE Transactions on Industrial Informatics*, 14(2):768–777, 2018.

[12] Kyung-Ah Shim. Reconstruction of a secure authentication scheme for vehicular ad hoc networks using a binary authentication tree. *IEEE Transactions on Wireless Communications*, 12(11):5386–5393, 2013.

[13] Jun Song, Fan Yang, Kim-Kwang Raymond Choo, Zhijian Zhuang, and Lizhe Wang. Sipf: A secure installment payment framework for drive-thru internet. *ACM Transactions on Embedded Computing Systems (TECS)*, 16(2):52, 2017.

[14] Sherali Zeadally, Ray Hunt, Yuh-Shyan Chen, Angela Irwin, and Aamir Hassan. Vehicular ad hoc networks (vanets): status, results, and challenges. *Telecommunication Systems*, 50(4):217–241, 2012.

[15] Daniel Jacobs, Kim-Kwang Raymond Choo, M-Tahar Kechadi, and Nhien-An Le-Khac. Volkswagen car entertainment system forensics. In *Trustcom/BigDataSE/ICESS, 2017 IEEE*, pages 699–705. IEEE, 2017.

[16] Richard Gilles Engoulou, Martine Bellaïche, Samuel Pierre, and Alejandro Quintero. Vanet security surveys. *Computer Communications*, 44:1–13, 2014.

[17] Jiafu Wan, Daqiang Zhang, Shengjie Zhao, Laurence Yang, and Jaime Lloret. Context-aware vehicular cyber-physical systems with cloud support: architecture, challenges, and solutions. *IEEE Communications Magazine*, 52(8):106–113, 2014.

[18] Debiao He, Sherali Zeadally, Baowen Xu, and Xinyi Huang. An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks. *IEEE Transactions on Information Forensics and Security*, 10(12):2681–2691, 2015.

[19] Jun Shao, Xiaodong Lin, Rongxing Lu, and Cong Zuo. A threshold anonymous authentication protocol for vanets. *IEEE Transactions on vehicular technology*, 65(3):1711–1720, 2016.

[20] Maria Azees, Pandi Vijayakumar, and Lazarus Jegatha Deboarh. Eaap: efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks. *IEEE Transactions on Intelligent Transportation Systems*, 18(9):2467–2476, 2017.

[21] Chandana Gamage, Ben Gras, Bruno Crispo, and Andrew S Tanenbaum. An identity-based ring signature scheme with enhanced privacy. In *SecureComm*, pages 1–5. Citeseer, 2006.

[22] Yixin Jiang, Minghui Shi, Xuemin Shen, and Chuang Lin. Bat: A robust signature scheme for vehicular networks using binary authentication tree. *IEEE Transactions on Wireless Communications*, 8(4):1974–1983, 2009.

[23] Rongxing Lu, Xiaodong Lin, Haojin Zhu, P-H Ho, and Xuemin Shen. Ecpp: Efficient conditional privacy preservation protocol for secure vehicular communications. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 1229–1237. IEEE, 2008.

[24] Rongxing Lu, Xiaodong Lin, Tom H Luan, Xiaohui Liang, and Xuemin Shen.

Pseudonym changing at social spots: An effective strategy for location privacy in vanets. *IEEE transactions on vehicular technology*, 61(1):86–96, 2011.

[25] Chenxi Zhang, Rongxing Lu, Xiaodong Lin, P-H Ho, and Xuemin Shen. An efficient identity-based batch verification scheme for vehicular sensor networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 246–250. IEEE, 2008.

[26] Lei Zhang, Qianhong Wu, Josep Domingo-Ferrer, Bo Qin, and Chuanyan Hu. Distributed aggregate privacy-preserving authentication in vanets. *IEEE Transactions on Intelligent Transportation Systems*, 18(3):516–526, 2017.

[27] JiLiang Li, Kim-Kwang Raymond Choo, WeiGuo Zhang, Saru Kumari, Joel JPC Rodrigues, Muhammad Khurram Khan, and Dieter Hogrefe. Epa-cppa: An efficient, provably-secure and anonymous conditional privacy-preserving authentication scheme for vehicular ad hoc networks. *Vehicular Communications*, 13:104–113, 2018.

[28] Byungjin Ko, Haengju Lee, and Sang Hyuk Son. Gps-less localization system in vehicular networks using dedicated short range communication. In *Embedded and Real-Time Computing Systems and Applications (RTCSA), 2016 IEEE 22nd International Conference on*, pages 106–106. IEEE, 2016.

[29] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

[30] Debiao He, Neeraj Kumar, Kim-Kwang Raymond Choo, and Wei Wu. Efficient hierarchical identity-based signature with batch verification for automatic dependent surveillance-broadcast system. *IEEE Transactions on Information Forensics and Security*, 12(2):454–464, 2017.

[31] Ahren Studer, Fan Bai, Bhargav Bellur, and Adrian Perrig. Flexible, extensible, and efficient vanet authentication. *Journal of Communications and Networks*, 11(6):574–588, 2009.

[32] Jonathan Petit, Florian Schaub, Michael Feiri, and Frank Kargl. Pseudonym

schemes in vehicular networks: A survey. *IEEE communications surveys & tutorials*, 17(1):228–255, 2015.

[33] Brijesh Kumar Chaurasia and Shekhar Verma. Conditional privacy through ring signature in vehicular ad-hoc networks. In *Transactions on computational science XIII*, pages 147–156. Springer, 2011.

[34] Panagiotis Panos Papadimitratos, Ghita Mezzour, and Jean-Pierre Hubaux. Certificate revocation list distribution in vehicular communication systems. In *Proceedings of the fifth ACM international workshop on VehiculAr Inter-NETworking*, pages 86–87. ACM, 2008.

[35] Björn Wiedersheim, Zhendong Ma, Frank Kargl, and Panos Papadimitratos. Privacy in inter-vehicular networks: Why simple pseudonym change is not enough. In *Wireless On-demand Network Systems and Services (WONS), 2010 Seventh International Conference on*, pages 176–183. IEEE, 2010.

[36] Abdelwahab Boualouache, Sidi-Mohammed Senouci, and Samira Moussaoui. Vlpz: The vehicular location privacy zone. *Procedia Computer Science*, 83:369–376, 2016.

[37] Abdelwahab Boualouache and Samira Moussaoui. Tapcs: Traffic-aware pseudonym changing strategy for vanets. *Peer-to-Peer Networking and Applications*, 10(4):1008–1020, 2017.

[38] Xinxin Liu, Han Zhao, Miao Pan, Hao Yue, Xiaolin Li, and Yuguang Fang. Traffic-aware multiple mix zone placement for protecting location privacy. In *INFOCOM, 2012 Proceedings IEEE*, pages 972–980. IEEE, 2012.

[39] Rongxing Lu, Xiaodong Lin, Tom H Luan, Xiaohui Liang, and Xuemin Shen. Anonymity analysis on social spot based pseudonym changing for location privacy in vanets. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1–5. IEEE, 2011.

[40] Yuanyuan Pan and Jianqing Li. Cooperative pseudonym change scheme based on the number of neighbors in vanets. *Journal of Network and Computer Applications*, 36(6):1599–1609, 2013.

[41] Hesiri Weerasinghe, Huirong Fu, Supeng Leng, and Ye Zhu. Enhancing unlinkability in vehicular ad hoc networks. In *Intelligence and Security Informatics (ISI), 2011 IEEE International Conference on*, pages 161–166. IEEE, 2011.

[42] JiLiang Li, WeiGuo Zhang, Saru Kumari, Kim-Kwang Raymond Choo, and Dieter Hogrefe. Security analysis and improvement of a mutual authentication and key agreement solution for wireless sensor networks using chaotic maps. *Transactions on Emerging Telecommunications Technologies*, 29(6):e3295, 2018.

[43] JiLiang Li, WeiGuo Zhang, Vivek Dabra, Kim-Kwang Raymond Choo, Saru Kumari, and Dieter Hogrefe. Aep-ppa: An anonymous, efficient and provably-secure privacy-preserving authentication protocol for mobile services in smart cities. *Journal of Network and Computer Applications*, 134:52–61, 2019.

[44] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *Annual International Cryptology Conference*, pages 388–397. Springer, 1999.

[45] David Galindo, Javier Herranz, and Eike Kiltz. On the generic construction of identity-based signatures with additional properties. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 178–193. Springer, 2006.

[46] Xiaodong Lin, Xiaoting Sun, Pin-Han Ho, and Xuemin Shen. Gsis: A secure and privacy-preserving protocol for vehicular communications. *IEEE Transactions on vehicular technology*, 56(6):3442–3456, 2007.

[47] Yusuke Nozaki, Yoshiya Ikezaki, and Masaya Yoshikawa. Tamper resistance of iot devices against electromagnnetic analysis. In *Future of Electron Devices, Kansai (IMFEDK), 2016 IEEE International Meeting for*, pages 1–2. IEEE, 2016.

[48] Joseph K Liu, Tsz Hon Yuen, Man Ho Au, and Willy Susilo. Improvements on an authentication scheme for vehicular sensor networks. *Expert Systems with Applications*, 41(5):2559–2564, 2014.

[49] Jung-Shian Li and Kun-Hsuan Liu. A lightweight identity authentication

protocol for vehicular networks. *Telecommunication Systems*, 53(4):425–438, 2013.

[50] Cheng-Chi Lee and Yan-Ming Lai. Toward a secure batch verification with group testing for vanet. *Wireless networks*, 19(6):1441–1449, 2013.

[51] Claus-Peter Schnorr. Efficient signature generation by smart cards. *Journal of cryptology*, 4(3):161–174, 1991.

[52] Tiziri Oulhaci, Mawloud Omar, Fatiha Harzine, and Ines Harfi. Secure and distributed certification system architecture for safety message authentication in vanet. *Telecommunication Systems*, 64(4):679–694, 2017.

[53] Cheng-Chi Lee, Yan-Ming Lai, and Pu-Jen Cheng. An efficient multiple session key establishment scheme for vanet group integration. *IEEE Intelligent Systems*, 31(6):35–43, 2016.

[54] Lei Zhang. Otibaagka: a new security tool for cryptographic mix-zone establishment in vehicular ad hoc networks. *IEEE Transactions on Information Forensics and Security*, 12(12):2998–3010, 2017.

[55] Maryam Rajabzadeh Asaar, Mahmoud Salmasizadeh, Willy Susilo, and Akbar Majidi. A secure and efficient authentication technique for vehicular ad-hoc networks. *IEEE Transactions on Vehicular Technology*, 67(6):5409–5423, 2018.

[56] Neetesh Saxena, Hong Shen, Nikos Komninos, Kim-Kwang Raymond Choo, and Narendra S Chaudhari. Bvpsms: A batch verification protocol for end-to-end secure sms for mobile users. *IEEE Transactions on Dependable and Secure Computing*, 2018.

[57] Eike Kiltz and Krzysztof Pietrzak. Leakage resilient elgamal encryption. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 595–612. Springer, 2010.

[58] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4):469–472, 1985.

[59] Shi-Jinn Horng, Shiang-Feng Tzeng, Yi Pan, Pingzhi Fan, Xian Wang, Tianrui

Li, and Muhammad Khurram Khan. b-specs+: Batch verification for secure pseudonymous authentication in vanet. *IEEE Transactions on Information Forensics and Security*, 8(11):1860–1875, 2013.

[60] Zhang Jianhong, Xu Min, and Liu Liying. On the security of a secure batch verification with group testing for vanet. *International Journal of Network Security*, 16(5):351–358, 2014.

[61] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of cryptology*, 13(3):361–396, 2000.

[62] Shamus Software Ltd. [online]. available: http://www.shamus.ie/index.php?page=home, accessed may 1, 2015., 2005.

Acronyms

CPPA	Conditional Privacy-Preserving Authentication	ix
CPS-CPPA	Certificateless and Provably-Secure Conditional Privacy-Preserving Authentication	x
ECPS-CPPA	Enhanced, Certificateless and Provably-Secure Conditional Privacy-Preserving Authentication	x
RSU	Roadside Unit	x
VANETs	Vehicular Ad hoc Networks	1
VANET	Vehicular Ad hoc Network	1
RSUs	Roadside Units	1
MANETs	Mobile Ad hoc Networks	1
MANET	Mobile Ad hoc Network	1
TAs	Trusted Authorities	1
IoV	Internet of Vehicles	1
OBUs	Onboard Units	2
V2V	Vehicle-to-Vehicle	2
V2I	Vehicle-to-Infrastructure	2
DSRC	Dedicated Short Range Communication	2
OBU	Onboard Unit	2
ITS	Intelligent Transport Systems	2
TA	Trusted Authority	3
ID	Identity	5
TPD	Tamper-Proof Device	6
DL	Discrete Logarithm	6
CDH	Computational Diffie-Hellman	6

PKI	Public Key Infrastructure	11
PID	Pseudo-Identity	12
PIDs	Pseudo-Identities	29
ECC	Elliptic-Curve Cryptography	59

List of Figures

List of Tables