Received August 6, 2019, accepted August 13, 2019, date of publication August 23, 2019, date of current version September 6, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2937182

A Decentralized Lightweight Authentication and Privacy Protocol for Vehicular Networks

SHIMAA A. ABDEL HAKEEM[1,2], **MOHAMED A. ABD EL-GAWAD**[1,3], **AND HYUNGWON KIM**[1], **(Member, IEEE)**

[1]Electronics Engineering Department, Chungbuk National University, Cheongju 28644, South Korea
[2]Electronics Research Institute (ERI), Giza 12622, Egypt
[3]National Telecommunication Institute, Cairo 11768, Egypt

Corresponding author: HyungWon Kim (hwkim@cbnu.ac.kr)

This work was supported in part by the IITP Grant through the Korean Government, under the development of wide area driving environment awareness and cooperative driving technology which are based on V2X wireless communication under Grant R7117-19-0164, and in part by the Center for Integrated Smart Sensors through the Ministry of Science, ICT and Future Planning as Global Frontier Project, South Korea, under Grant CISS-2019.

ABSTRACT Recently many security protocols have been proposed for road safety applications in Vehicle-to-everything (V2X) communications. Most of them, however, do not fully satisfy the requirements of light-weight and fast processing, which are special properties for V2X. Most of the previous authentication protocols assume that a Certificate Authority (CA) is present within the communication range from all the vehicles, which is not practical for moving vehicles. We propose a light-weight security protocol for authentication and privacy protection for V2X. It employs two security hardware devices, Biometric Device (BD) and Tamper Proof Device (TPD), which verifies the driver and securely keeps the keys, respectively. It decentralizes the CA's tasks by locally generating pseudo-identity and private keys to preserve privacy and provide authentication in Vehicle-to-Vehicle (V2V) communication. In addition, we propose an authentication signature protocol using a notion of hash-chain key generation. We implemented the proposed key generation and authentication protocol using NS-3 simulator. Our extensive simulations demonstrated that the proposed authentication protocol significantly enhances the security level while protecting the conditional privacy of vehicles by utilizing anonymous identities. The proposed protocol has a 20% $\sim$ 85% less communication overhead compared with the previous protocols.

INDEX TERMS Hash-chain, MAC algorithm, privacy, two-factor authentication, biometric, tamperproof.

I. INTRODUCTION

Vehicle-to-everything (V2X) is receiving increasing attention due to the recent progress in autonomous driving technologies. Each vehicle is equipped with Onboard Unit (OBU) to communicate with other vehicles and with Road Side Units (RSUs) which are available on the road. Dedicated Short-Range Communications (DSRC) standard [1] is utilized to exchange periodically reporting messages between vehicles in large scale networks. Recently, vehicle technologies and applications are targeting road safety with high data exchanging rate to provide alerts about any upcoming crashes in addition to providing mobility information of the surrounding vehicles which enhances the safety of automatic driving. V2X high mobility conditions make vehicles more susceptible to attacks. Hence, V2X critically needs a very strong security system which allows vehicles to communicate safely and securely. Many recently proposed security protocols are trying to enhance the security level by satisfying the requirements and challenges of existing V2X security standards [2]. V2X privacy and authentication are the most important issues which are the primary concerns in this paper. Message and driver's authentication are of the core elements for any V2X security system that allow the drivers to access the network resources at any time. Being authorized would allow a driver to exchange messages and alerts with the vehicles in the driver's communication range.

Preserving the vehicle's privacy by hiding its real identity using pseudo-identities cannot fully hide the vehicles from being tracked as linking two pseudo-identities can ease the discovery of the real identity. Exposing vehicles privacy would allow malicious vehicles to gain access to the private

The associate editor coordinating the review of this article and approving it for publication was Peter Langendorfer.

information of other vehicles as well as to the whole V2X system information. In this paper, we are trading off between hiding the real identity of the vehicle and allowing the authorities to keep track of this vehicle under some special conditions like the commitment of a crime or any misbehaving behavior. Non-repudiation is the security property which ensures that no vehicle can deny sending a special message [3]. Any security protocol must support authentication, privacy, and non-repudiation.

Many V2X security protocols are depending on traditional Public Key Infrastructure (PKI) to support message authentication and integrity. In PKI, message signing is done using a digital signature algorithm in addition to a certificate to authenticate the sender. PKI based solutions are providing a high-security level, however, they are still experiencing high communication overhead due to the large size of the attached certificate in comparison with the original message size [4]. The critical issue concerning previous Conditional Privacy-Preserving Authentication (CPPA) protocols is the dependence of these protocols on the Certificate Authority (CA), which causes high traffic density at the CA and a single failure point.

V2X security is moving towards providing some decentralized authentication and privacy solutions. Recently, the Two-Factor authentication is widely used with online banking, web-based systems and many communication systems [5]. Two-Factor authentication grants service access after passing a two-stage user identification process. In the first stage, each user applies his own credentials then verifies the second credentials that are received from the authentication server as a result of the successful passing of the first stage (e.g., a security code sent over the phone). According to standards and previous related work [6], Tamper Proof devices (TPDs) are recommended to be installed on vehicles, since it is generally assumed impossible to compromise TPDs. According to WAVE standard [1], each OBU is equipped with a Hardware Security Module (HSM), a type of TPD, which keeps the master keys secure. A Biometric Device (BD) is also recommended to be installed per each OBU and can provide better safety standards [7].

TPD and BD will be used as CA's agents to provide security services instead of frequently direct communication to the CA. Using TPD and BD would decentralize the CA's work and reduce the communication burden on the CA.

In this paper, we propose a decentralized security protocol that employs a two-factor authentication concept and a lightweight message signing protocol. The proposed protocol utilizes a Message Authentication Code (MAC) [8] to sign and verify each transmitted message via sharing one common pre-generated hash chain (i.e., a chain of keys). In the signing process, the sender vehicle calculates a MAC value using a randomly selected chain key, then attaches the obtained MAC value and the index of the selected key to the message. Accordingly, the receiver vehicle uses the received key index to extract the corresponding chain key and verifies the received MAC value.

The rest of this paper is organized as follows: Section II provides the previous V2X security-related implementations. Section III explains the addressed security requirements and mathematical preliminaries. In section IV, we introduce the system model and network structure. Section V explains in details the proposed protocol. In Section VI, security correctness proof is given. Section VII compares the performance of the proposed protocol with other previous works via NS3 simulations. Conclusions are given in Section VIII.

II. RELATED WORK

V2X security has been studied extensively, and hence, a discussion of the studies reported in the literature will be discussed in this section. One of the most important solutions was the Public Key Infrastructure (PKI) which was standardized by IEEE1609.2 as the security solution for all V2X safety applications. Although PKI provides message authentication, integrity, and non-repudiation, it has many drawbacks. The major drawbacks of PKI are due to processing delay and communication overhead. The reason is that PKI uses a pair of keys: a private key to sign the message and a public key that is attached with a trusted certificate to allow the receiver to verify the message. Due to the downsides of PKI, Adrian Perrig *et al.* [9], proposed an efficient multicast secure source authentication protocol known as TESLA. TESLA protocol is based on MAC algorithms with delayed symmetric key disclosure. TESLA uses symmetric key cryptography which is faster than using a digital signature. TESLA solution got a wide acceptance as an authentication protocol with low communication overhead but it still suffers from the non-repudiation problem.

Another security protocol called VAST protocol has been reported in [10]. The VAST targets of different V2X applications and provides multi-hop authentication. VAST is based on a mix of TESLA++ (i.e., an updated version of TESLA) and the Elliptic Curve Digital Signature Algorithm (ECDSA) signatures. TESLA++ has introduced an efficient solution for preventing Denial of Service (DoS) memory attacks, however, it doesn't provide multi-hop authentication and non-repudiation. The VAST protocol does not depend on infrastructure which makes it high applicable in rural areas or areas with early deployment phases of V2X. The most critical drawback of this protocol is the long-time delay consumed by the message verification process.

Huang *et al.* [11] proposed the Anonymous Batch Authenticated and Key Agreement (ABAKA) protocol which supports the multiple-request authentication from different vehicles and establishes different session keys with each vehicle at the same time. ABAKA protocol applies ECDSA protocol to efficiently verify a batch of requests using only one verification operation and negotiate a distinct session key with each vehicle through one broadcasting message. Jia *et al.* [12] proposed an Efficient Privacy-Preserving Authentication Protocol (EPAS) for V2X by adopting the ECDSA signature and batch verification to provide effective and light authentication. EPAS based emergency communication employs two different design protocols, the first protocol is based on the V2D communication in which the received messages can be verified in a batch by the Disaster Relief Authority (DRA) or can be verified separately. The EPAS second protocol is based on vehicle group communication, as vehicles during the rescue process need to communicate with each other for timely triggered exchanged information.

Kamat *et al.* [13] adopted a new secure pseudonym and identity-based protocol for V2X to provide a high security and authentication level over traditional PKI and symmetric solutions. Their protocol validity does not need high storage requirements which solves the storage problem of PKI.

In [14], Xiaodong lin, *et al.* introduced a privacy-preserving and security solution for V2X (GSIS). This protocol is based on identity-based signature and group signatures techniques which provide anonymous and liability authentication but it suffers from a linear increase in the verification time while the number of revoked vehicles increases.

Lu *et al.* [15] presented a novel adaptive privacy-preserving framework with ID-based authentication for V2X. The self-generated and adaptive pseudonyms are used as vehicles identifiers instead of their real identities. The mentioned framework is based on two different signature protocols: the first one is the ID-based signature (IBS) protocol and the second one is the ID-based Online/Offline signature (IBOOS) protocol. In Vehicle-to-Infrastructure (V2I) communication, IBS protocol is used for authentication, while for Vehicle-to-Vehicle (V2V) IBOOS protocol is used. Reusability is one of the good results of this framework as it can be reused with new IBOOS and IBS protocols for improving security performance.

Zhang *et al.* [16] proposed a novel Identity-Based Batch Signature Verification (IBV) Protocol for V2X. This solution depends on RSUs to verify multiple and distinct received signatures at the same time which reduces the total verification time. Biswas *et al.* [17] introduced an ID-based authentication protocol for safety applications in V2X. The proxy signatures are used by the mentioned protocol to provide authentication flexibility. This protocol uses the location information as signer's ID to sign and verify the proxy signatures. In this protocol, the message is valid for the only single zone which is considered as a critical problem.

Bayat *et al.* [18] proposed a new Conditional Privacy-Preserving Authentication (CPPA) protocol based on bilinear pairing to improve the identity-based CPPA protocol in V2X. However, this protocol cannot prevent message modification attack in which an adversary can repeat the transmission of a previous message after modifying its content. Bayat et al.'s protocol incorporates a Tamper Proof Device (TPD) which is a hardware security module device to store the vehicles pseudonyms and the secret keys of each vehicle.

In He et al.'s protocol [19], the VANET system's secret key is kept in TPD to prevent attackers from compromising and controlling the whole system. Using one secret system key is considered as a weak point of He et al.'s protocol that allows the adversaries to gather all running cryptographic operations. Once the master private key is compromised, the adversary could control the whole system and causes a lot of damage.

In Wang et al.'s protocol [20], a lightweight and efficient strong privacy-preserving (LESPP) authentication protocol was proposed for securing VANET communication using lightweight MAC tags and symmetric encryption. LESPP attaches a MAC signature per each message to allow the receiver verifying the message with a low power computation and low memory overhead, however, this protocol can't resist the single point failure problem.

Recently, Zhou *et al.* [21] protocol introduced an efficient privacy and authentication protocol for V2X based on the key separation approach. Their protocol uses two different types of secret keys which are updated periodically. The first key is issued by a security organization (e.g., CA) while the second key is issued by the vehicle driver. The messages construction is based on Elliptic Curve Cryptographic (ECC) security which makes it more efficient and decreases computation rather than using bilinear pairing. However, this protocol supports the installation of TPD devices in each vehicle, it still suffers from modification attacks and system key single point of failure. It has an assumption that the key generation organization is assumed to be fully trusted but if this organization is compromised the whole system will be attacked.

Many security solutions are based on one-system secret keys like 2FLIP protocol which has been proposed by Wang *et al.* [22]. 2FLIP provides a light-weight protocol for authentication and privacy, however, its revocation process has some limitations. For instance, for multi-drivers' vehicle, 2FLIP cannot recognize which driver misbehaves, and consequently revokes the whole vehicle as a one-unit which leads to unfair revocation. Another limitation is that the CA must hash all vehicles pseudo-identities and store them in vehicles table which is considered a great overhead. 2FLIP depends on a one-system secret key in all cryptographic operations which makes it vulnerable to key compromising attacks.

In this paper, we offer an effective security solution where we utilize the BD and TPD devices to generate random keys and dynamic pseudo-identities. One of the widely used TPD is IBM4758 cryptographic coprocessor [23], which can store the cryptographic material, sign messages and verify the signature. Using its hardware, TPD can make it extremely difficult for hackers to compromise the cryptograph material kept and calculated only inside the hardware. The proposed protocol supports privacy, integrity, and authentication using a light-weight MAC algorithm and a pre-generated hash table. Avoiding the system key, our protocol eliminates the single point of failure issue which manifests in previous protocols [18], [22]. The proposed protocol also introduces a notion of a distinct sequence of hash elements to speed up its messages authentication process without sacrificing the security level.

Conventional MAC algorithms require sending the signed key together with the MAC value calculated over the message [24] to allow the receiver to verify the message. The proposed algorithm doesn't require sending the key to improve the security level. Instead, all vehicles store the same hash chain table of *n* key elements. Each sender randomly chooses one key to sign the message and attaches only the index of the key to the message as a pointer to the key. Using these improved methods, the proposed protocol can significantly improve the computation speed, increase the security level, and reduce the network overhead.

III. THE PROPOSED PROTOCOL CONTRIBUTION AND THE FUNDAMENTAL SECURITY PRELIMINARIES

In these sections, we summarize the target security goals that the proposed protocol can satisfy.

A. PROTOCOL CONTRIBUTION

1. Self-generation of Pseudo-identity: Using in-vehicle TPD and BD, each vehicle generates *n* anonymous identities and the corresponding private keys without frequent access to CA.
2. Hash-chain Table Generation: The novelty of our protocol is the offline generation of a common hash chain table which is stored in all vehicles before the system initialization. The use of distinct hash-based random keys for authentication and integrity check can increase the security level and avoid the attacks that compromise the key.
3. Driver Biometric Authentication: The use of BD enhances the security level by allowing only the authorized drivers to access the vehicle using their biometric information, and then activating the TPD to join the V2X system.
4. Practical Revocation of Misbehavior's Certificate: We propose an efficient certificate revocation protocol that can invalidate only the misbehaving drivers instead of the vehicle entirely. In contrast, the previous protocols revoke the vehicle entirely instead of only the misbehaving drivers by invalidating the whole vehicle as one unit, which can cause unnecessary over-restriction, especially for the car-sharing applications.
5. Strong Message Integrity: Each message includes a short MAC signature and pseudo-identity to ensure the integrity and authentication of each message. At the receiver side, each vehicle calculates the signature and compares it with the attached signature. The receiver then accepts the message, only if the signatures are equal.
6. Non-repudiation and Privacy Preservation: The proposed protocol satisfies the security requirements of preserving privacy while at the same time tracking misbehaving vehicles. The use of the biometric information prevents the drivers from denying sending a harmful message, which fulfills the non-repudiation requirements.
7. Traceability: Anonymity is introduced in many protocols by using an anonymous identity to hide the real identity of each vehicle at the cost of sacrificing the conditional traceability. In contrast, our protocol, while fully supporting anonymity, provides the CA with the ability to track and revoke the misbehavior vehicle by mapping the pseudo-identity of each vehicle with its real identity. This way, it ensures that no vehicle can deny the generation of harmful messages.
8. Periodic Hash Chain Updates: The proposed protocol provides a method of periodic key updates which is scheduled by the CA at the initialization phase.

B. ELLIPTIC CURVE AND HASH CHAIN PRELIMINARIES

As our proposed solution is based on the elliptic curve cryptography and the MAC generation using hash chains, we describe the basic operations of them in this subsection.

1) THEORY OF ELLIPTIC CURVE

An elliptic curve algorithm is expressed in a cubic equation form as in eq. (1).

$$y^2 + axy + by = x^3 + cx^2 + dx + e \tag{1}$$

Here, d, c, b, a and e are real numbers. In the ECC system, the equation is defined as the form of Eq. (2), and Eq. (3).

From Eq. (2), Eq (a, b) represents the formula to calculate the elliptic curve with domain parameters a, b over a prime finite field Fq, where $q > 3$ and a, $b \in Fq$ according to Eq. (3).

$$Eq(a, b) : y^2 = x^3 + ax + b(mod\ p) \tag{2}$$

$$4a^3 + 27b^2(mod\ p) \neq 0 \tag{3}$$

Given an integer $n \in F_q$ and a point $P \in Eq(a, b)$, nP defines scalar multiplication over $Eq(a, b)$. In our protocol, we used a non-singular elliptic curve E/Fq which is described using Eq. (2) and (3). The point P on the elliptic curve forms an additive cyclic group G with order q, while it consists of all points on the elliptic curve E and the point at infinity. In general, the security level of ECC depends on how difficult it is to deduce the elliptic curve coefficients as described below [25].

2) ECC COMPUTATIONAL PROBLEMS

Definition 1 (Elliptic Curve Discrete Logarithm Problem (ECDLP)): Given $Q, R \in E$, find an integer $k \in Z_q^*$ such that $R = kQ$.

Definition 2 (Decisional Diffie–Hellman Problem (DDHP)): Given (P, aP, bP, cP) for any $a, b, c \in Z_q^*$, decide whether or not $cP = abP$, i.e., decide whether $c = ab\ mod\ q$ or not.

Up to now, there is no polynomial algorithm that can solve the two aforementioned problems [26]. In this paper, the proposed protocol exploits, instead of bilinear pairing, the point of multiplication over *ECC* which reduces the computational cost.

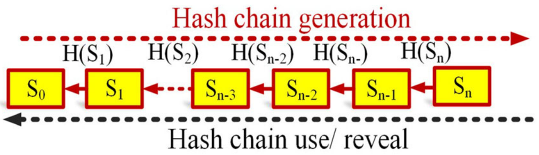


FIGURE 1. Hash chain generation and revealing.

3) HASH CHAIN GENERATION

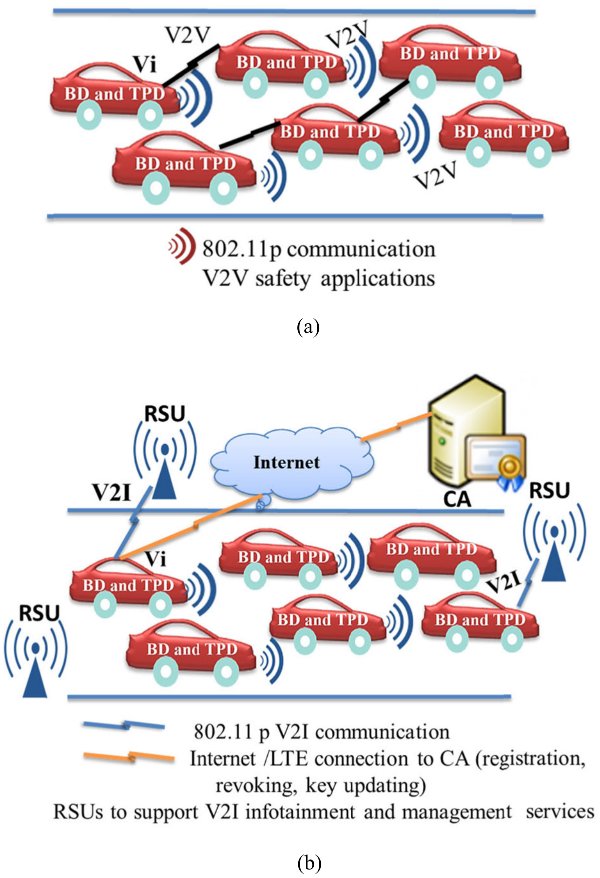
A set of hash functions $h(\cdot)$ is used to generate the One Time Passwords (OTPs). A hash function is generally defined as a one-way function $y = h(x)$ for a given input x. It is known that it is extremely difficult to predict the original input x from the hashed value y. The use of hash chains has been first introduced by Lamport [27]. Later, it was enhanced to implement an *OTP* system by applying the hash function $h(\cdot)N$ times starting with a seed (s) to generate a hash chain of length N. The result of such recursive hash chain generation produces a set of keys S_i illustrated in Fig 1. Each of the private keys is used for calculating MAC value over messages [24].

IV. SYSTEM MODEL

A. NETWORK MODEL

In this paper, we target two vehicular communication modes (V2V and V2I). Vehicle to Infrastructure (V2I) mode is supported only when the RSU exists to allow vehicles connection to the CA. The network system model consists of a CA, a set of RSUs installed on the roads and a varying number of vehicles on the road. The role of CA includes the initialization phase authentication, the system key updating process, and the vehicles revoking. The system model is not fully depending on the CA since vehicles are equipped with BD and TPD devices for authentication parameters generation. As shown in Fig. 2a and Fig. 2b, the proposed network model can support authentication for messages and identities in any vehicular communication application especially the multi-drivers service.

It is assumed that CA is a fully trusted certificate authority organization with unlimited memory storage and computation resources. CA handles many tasks like vehicles registration, key management, and conditional traceability of vehicles. In addition, we assume that the RSUs are deployed on the roadside and can communicate with a CA using a wireline internet connection. An RSU has a wide communication range and enough memory storage to offer security services to the running vehicles on the road. Each vehicle is equipped with an OBU which includes a hardware security module called TPD device with addition to BD device. An OBU in each vehicle periodically broadcasts a standard safety message consisting of the current time, the vehicle's position, speed, direction, acceleration, and road traffic events. These safety messages can provide the driver of each vehicle warning of upcoming dangers or can assist autonomous vehicles to avoid accidents.



(a)

(b)

FIGURE 2. (a) The proposed network system model (V2V communication mode) (b) The proposed network system model (V2I communication mode).

B. ADVERSARY MODEL

An adversary is an entity attempting to compromise the security by monitoring or modifying the data transferred through the communication channel. The adversary may have powerful communication abilities and computation resources. An adversary can also tamper messages, drop packets, replace the original messages, and delay the transmission of messages. As we previously discussed, in the proposed method, the cryptographic materials and secret keys are saved only in TPDs. An adversary may disguise itself by using another vehicle's valid identity to send harmful or false messages without being detected. The adversary may try to use a stolen TPD to impersonate other drivers and generate a large number of invalid or legitimate messages to disrupt the V2X services. We assume that the adversary can affect the system with some defined attacks as explained in section (VI). We assume that the adversary can generate replay attack, key modification attack, DOS attack, and message modification attacks.

V. THE PROPOSED PROTOCOL

The proposed protocol initially generates a hash chain using an initial secret key. Using a pre-defined hash algorithm, the TPD in each vehicle securely generates a hash chain of n elements that represent the signing keys. A vehicle

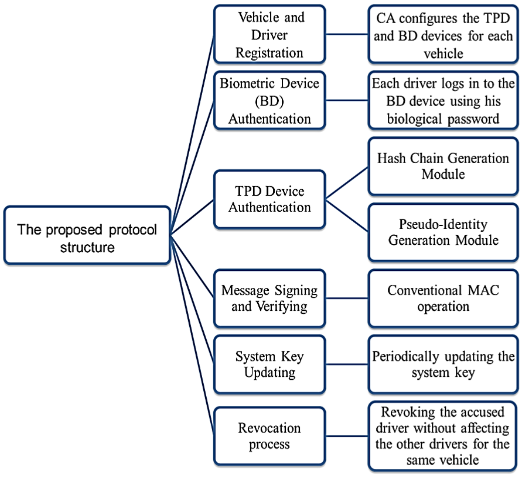

The proposed protocol structure
Vehicle and Driver Registration
CA configures the TPD and BD devices for each vehicle
Biometric Device (BD) Authentication
Each driver logs in to the BD device using his biological password
TPD Device Authentication
Hash Chain Generation Module
Pseudo-Identity Generation Module
Message Signing and Verifying
Conventional MAC operation
System Key Updating
Periodically updating the system key
Revocation process
Revoking the accused driver without affecting the other drivers for the same vehicle

FIGURE 3. The proposed protocol full structure.

signs each message using one of these keys and attaches the obtained signature to the message and only the index i for the key K_i. K_i is one hash element from the generated hash chain instead of the actual key. At the receiver side, only registered vehicles that have the pre-generated hash table can retrieve the signing key K_i from the index i and verify the signature. The receiver calculates the MAC value over the received message and accepts the message only if the calculated MAC matches the received one. The proposed protocol is composed of the following processes: vehicle and driver registration, BD device authentication, TPD device authentication, message authentication, system key update, and certificate revocation as described in Fig 3.

Table 1 describes the notations used in the proposed algorithms. In the following subsections, we provide a detailed description of each process of the proposed protocol.

A. VEHICLE AND DRIVER REGISTRATION

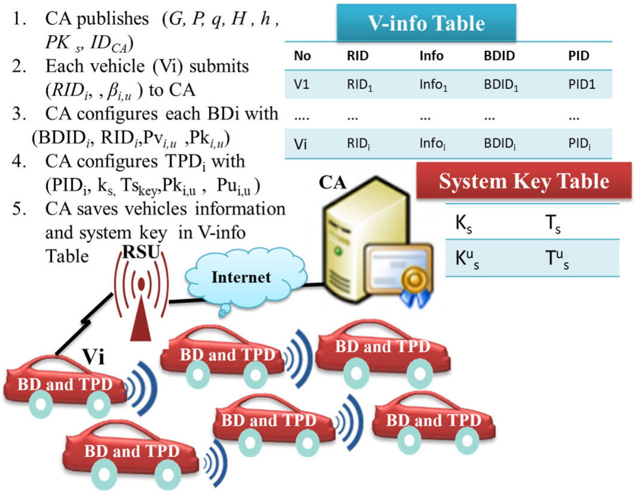
We assume that each vehicle using the proposed protocol is equipped with a bd device and all legitimate drivers are registered at the ca with their biometric information. to support multiple drivers effectively for the same vehicle as in car-sharing applications, our method requires each driver to register at the ca individually. once the ca approves the driver registration, it configures bd with the following two functions:

- Password verification function $Pv_{i,u}$: verifies the driver's login to the BD.
- Password maintenance function $Pk_{i,u}$: updates the password when the driver biological information changed.

CA configured the BD device with $Pv_{i,u}$, $Pk_{i,u}$ to verify and keep the driver biological password, then keep evidence of each biological password for future tracking. In Fig. 4, the CA as a managing authority, picks a two large prime numbers p, q and a non-singular elliptic curve E defined by Eq. (2).

TABLE 1. The system notations and abbreviations.

Notations	Descriptions
CA	Certificate Authority
$Vehcile_i$	The ith vehicle
TPD_i	Tamper Proof Device of $Vehcile_i$
BD_i	Biometric Device of $Vehcile_i$
$Pu_{i,u}$	Biological Password Updater for driver u of $Vehcile_i$
$Pk_{i,u}$	Biological Password Keeper for driver u of Vehcile$_i$
$Pv_{i,u}$	Biological Password Verifier for driver u of Vehcile$_i$
s	Secret Random Value
PK_s	Full system public key , $PK_s = sP$
G	G is acyclic additive group of order q with a generator P
SID_{CA}	The secret identity of the CA and $SID_{CA} = sH(ID_{CA})$
K_i	The generated hash chain elements,
k_s	System secret key, $k_s =$ h (s|| SID_{CA})
$h(.)$	A collision-free one- way hash function as h: $\{0,1\}^1 \in Z_q^*$
$H(.)$	A MapToPoint hash function as $H: \{0,1\}1 \in G$
RID_i	Real identity vehicle$_i$
$\beta_{i,u}$	The hashed version of biological password for driver u of vehicle$_i$, $\beta_{i,u}=h(pw_{i,u})$
$BDID_i$	The Biometric Device Identity of $Vehcile_i$
Ts_{key}	The time stamp of the current secret system key k_s
PID_{init}	The initial pseudo-identity of $Vehcile_i$, configured by the CA
Sig_{ki}	The calculated MAC value over m$_i$ using the secret system key k_s $sig_{ki} = mac_{ki}(PID_i \|\| m_{i,j} \|\| Ts)$
PID_i	The pseudo- identity of $Vehcile_i$
k_s^u	The updated version of system key k_s , CA will generate a new key k_s^u
ID_{CA}	Identity of Certificate Authority
T_s^u	the time stamp of the updated secret system key k_s^u
$Enc\,k(.)$	Encryption function using k as the key like DES or AES algorithms
||	Concatenation operation of messages
MAC	Message Authentication Code, HMAC
$\oplus$	Exclusive or operation (XOR)


1. CA publishes ($G, P, q, H, h, PK_s, ID_{CA}$)
2. Each vehicle (Vi) submits (RID_i, , $\beta_{i,u}$) to CA
3. CA configures each BDi with ($BDID_i$, RID_i,$Pv_{i,u}$,$Pk_{i,u}$)
4. CA configures TPD_i with (PID_i, k_s, Ts_{key},$Pk_{i,u}$, $Pu_{i,u}$)
5. CA saves vehicles information and system key in V-info Table

V-info Table

No	RID	Info	BDID	PID
V1	RID_1	$Info_1$	$BDID_1$	PID1
....	...	...	...	...
Vi	RID_i	$Info_i$	$BDID_i$	PID_i

System Key Table

K_s	T_s
K^u_s	T^u_s

FIGURE 4. CA and vehicles registration steps.

Through a secure channel (e.g., submit information personally), each driver must provide the vehicle's real identity, RID_i, his biological password, $\beta_{i,u}$, and the vehicle information, Info$_i$ (e.g., manufacturing date, serial number, model and vehicle owner). Once the CA verifies and approves the

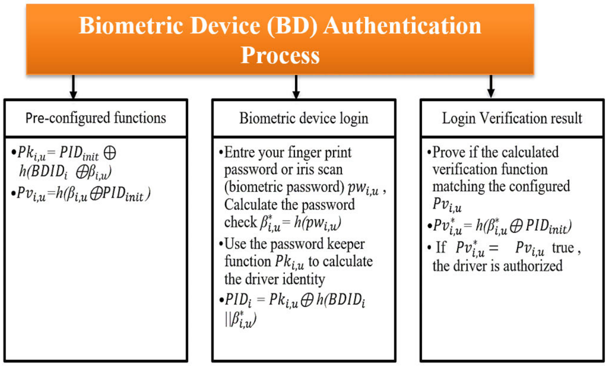

Biometric Device (BD) Authentication Process
Pre-configured functions
•$Pk_{i,u}= PID_{init} \oplus h(BDID_i \oplus \beta_{i,u})$
•$Pv_{i,u}=h(\beta_{i,u} \oplus PID_{init})$
Biometric device login
•Entre your finger print password or iris scan (biometric password) $pw_{i,u}$, Calculate the password check $\beta^*_{i,u}= h(pw_{i,u})$
•Use the password keeper function $Pk_{i,u}$ to calculate the driver identity
•$PID_i = Pk_{i,u} \oplus h(BDID_i || \beta^*_{i,u})$
Login Verification result
•Prove if the calculated verification function matching the configured $Pv_{i,u}$
•$Pv^*_{i,u}= h(\beta^*_{i,u} \oplus PID_{init})$
• If $Pv^*_{i,u} = Pv_{i,u}$ true , the driver is authorized

FIGURE 5. Biometric device (BD) login process.

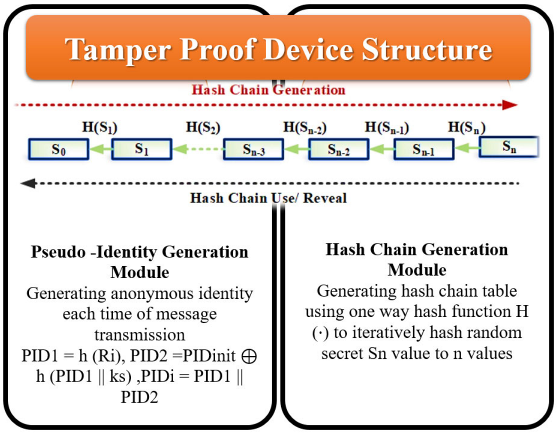

Tamper Proof Device Structure
Hash Chain Generation
$H(S_1)$
$H(S_2)$
$H(S_{n-2})$
$H(S_{n-1})$
$H(S_n)$
S_0
S_1
S_{n-3}
S_{n-2}
S_{n-1}
S_n
Hash Chain Use/ Reveal
Pseudo -Identity Generation Module
Generating anonymous identity each time of message transmission
PID1 = h (Ri), PID2 =PIDinit ⊕ h (PID1 || ks) ,PIDi = PID1 || PID2
Hash Chain Generation Module
Generating hash chain table using one way hash function H (·) to iteratively hash random secret Sn value to n values

FIGURE 6. Tamper proof device (TPD) modules.

submitted information, it assigns a driver initial pseudo-identity, PID_{init}, and a BD biometric device identity, $BDID_i$. For each vehicle, CA then stores $\{RID_i, PID_{init}, BDID_i, Info_i\}$ in the vehicle database (v-info table). CA configures BD_i with parameters $\{BDID_i, RID_i, Pv_{i,u}, Pk_{i,u}\}$, which is transferred to TPD_i for the corresponding driver. Finally, TPD_i stores $\{PID_{init}, k_s, Ts_{key}, Pki, u$. where, Ts_{key} defines the current timestamp of the secret system key.

B. BIOMETRIC DEVICE AUTHENTICATION

Before driving, each driver logs in to the biometric device, *BD*, using his biological password, $pw_{i,u}$ (e.g. a fingerprint scan or iris scan). $Pv_{i,u}$ and $Pk_{i,u}$ that are previously configured by the CA are used to verify and keep the driver's biological identity. Each BD_i verifies the driver identity by calculating the biometric login information $\{\beta^*_{i,u}, PID_{init}, Pv^*_{i,u}\}$ using the verifying functions described above. If the driver is a legitimate user, the BD automatically activates the TPD and allows the driver to send and receive messages. Fig. 5 shows the BD_i login calculation steps.

C. TPD DEVICE AUTHENTICATION

In this subsection, we explain the preparation steps of the TPD device to start signing and verifying the transmitted messages. TPD speeds up the required processing by providing an offline generation of pseudo-identities and hash elements as shown in Fig 6. The TPD operations are described below.

1) PSEUDO-IDENTITY GENERATION MODULE

To provide anonymity, each TPD generates a group of pseudo-identities with each identity composed of two parts PID^1 and PID^2. These pseudo-identities can hide the real identity of the vehicle from other vehicles and prevent the tracking attacks. We define the structure of the pseudo-identity in a way that allows only the CA to retrieve the real identity at any time. In the revocation subsection (subsection B), we show how the CA maps the real identity of the vehicle to any of its pseudo-identities. Eq. 4, 5, and 6 show the required logical operations to generate the pseudo-identities.

$$PID^1 = h(R_i) \tag{4}$$

$$PID^2 = PID_{init} \oplus h(PID^1||k_s) \tag{5}$$

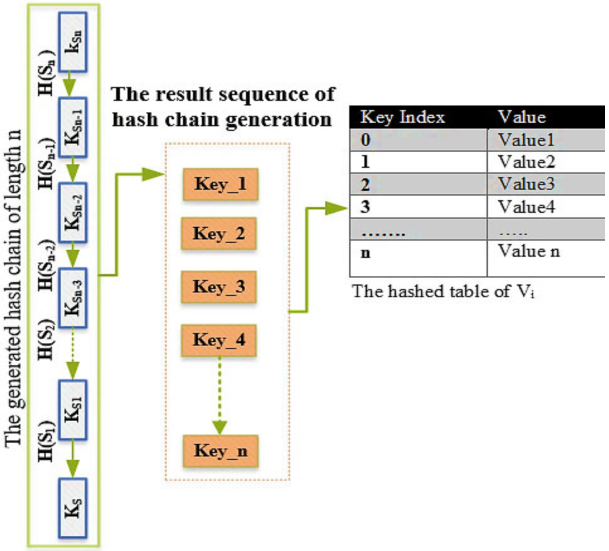
$$PID_i = PID^1||PID^2 \tag{6}$$

Here $R_i \in z^*$ is a random number for each generated pseudo-identity. PID_i consists of a dynamic part PID^1 and static part PID^2; PID^1 is a hashing value of a random number R_i, while PID^2 is calculated by an XOR of the initial pseudo-identity of each vehicle and the concatenation of PID^1 and system key k_s. PID_i in this way satisfies the anonymity condition and provides a practical revoking function. The static part PID^2 allows the CA to find the real information of drivers using PID_{init} as an index for the vehicle information table, while the dynamic part PID^1 hides the real identity of the vehicle to support anonymity.Each vehicle generates n pseudo-identities before starting communication to save the generation time for PIDs on the fly.

In this paper, we propose an efficient and practical revoking method that can eliminate the search time for long certificate blacklists. The proposed revoking method, unlike conventional solutions, does not require periodic updates for the certificate lists at all vehicles. The proposed method allows the CA to revoke the misbehaving vehicle by sending a revoking message to the TPD device of the misbehaving vehicle. When a misbehaving vehicle identity (PID_i is reported to the CA, it uses Eq. (5) and (6) to find the value of PID_{init}, then uses PID_{init} to search the v-info table and map it to the real vehicle identity. The proposed pseudo-identities generation method can achieve anonymity and supporting efficient revoking.

2) HASH CHAIN GENERATION MODULE

During the vehicle registration phase, the CA has stored the secret system key, k_s, in the OBU of each vehicle. Using the installed secret key k_s as a seed, this module pre-calculates a hash of n elements by iteratively calculating a pre-stored one-way hash function, $H()$.



Key Index	Value
0	Value1
1	Value2
2	Value3
3	Value4
....	.....
n	Value n

FIGURE 7. The hash chain generation using system key k_S.

The pre-calculated hash chain is stored in each registered vehicle. Each element of the chain is used later as a signing key for message authentication. Fig 7. illustrates an example of the hash chain generation process. Each vehicle regenerates the hash chain when the CA updates the secret system key, k_s.

D. MESSAGE SIGNING AND VERIFYING

After a successful driver login and hash key generation, the TPD device can sign and verify each transmitted message. Eq. 7 defines the message signature formula that is similar to a conventional MAC operation.

$$Sig_{ki}{=}mac_{ki}(PID_i||m_{i,j}||Ts) \tag{7}$$

The parameters of Eq. 7 are described as follows:

$\boldsymbol{mac_{ki}}$: The MAC value for message m using a signing key of $\boldsymbol{ki}$.

$\boldsymbol{Sig}_{ki}$: The output signature of the MAC operation is truncated to only 12 bytes. Although the actual length of the hashed values using HMAC-SHA-256 algorithm is 20 bytes, according to the mentioned MAC truncation operation in [28], we keep only the least significant 12 bytes.

$\boldsymbol{m_{i,j}}$: The transmitted message from the vehicle v_i to a vehicle v_j

$\boldsymbol{PID_i}$: The pseudo-identity of the sender vehicle, v_i, that allows other vehicles to verify the sender vehicle legitimacy.

$\boldsymbol{T_s}$: the current timestamp.

The sender vehicle, v_i, attaches the obtained signature, Sig_{ki}, and the index k_{index} of the signing key, the timestamp, T_s, and the sender pseudo-identity, PID_i, to the transmitted message as depicted in Fig 8.

When the receiver vehicle, v_j, receives the safety-related message $\{PID_i, Sig_{ki}, m_{i,j}, k_{index}, T_s\}$, v_j checks the freshness of timestamp, Ts.

$\boldsymbol{PID_i}$ (20 bytes)	$\boldsymbol{Sig_{ki}}$ (12 bytes)	$\boldsymbol{m_{i,j}}$	$\boldsymbol{k_{index}}$ (4bytes)	$\boldsymbol{T_s}$ (4bytes)

FIGURE 8. The proposed message format.

If T_s is invalid, v_j rejects the message; otherwise, v_j verifies the signature of the received message. v_j queries the stored hash table using the received key index, k_{index}. Then, v_j calculates the signature of the received message, Sig^*_{ki}, as in Eq. 8.

$$Sig^*_{ki} = mac_{ki}(PID_i||m_{i,j}||Ts) \tag{8}$$

If the calculated signature, Sig^*_{ki}, is equal to the received one, Sig_{ki}, the receiver vehicle, v_j, accepts the message.

E. THE PERIODIC UPDATING OF THE SYSTEM KEY

The proposed solution assumes that the secret system key, k_s, is physically secured by the BD and TPD devices. For instance, if an illegitimate user tries to access the BD with wrong biometric information, the TPD after the specified number of trials flushes all stored information including the system key, k_s., Nevertheless, we can enhance the security level of the proposed system by periodically updating the system key. This periodic update takes place according to a predefined agreement between CA and all registered vehicles. The update operations are described as follows:

- CA generates a new key k^u_s.
- Encrypt the updated key k^u_s using the current system key k_s as follows:

$$C = Enc_{k_s}(k^u_s||ID_{CA}||T^u_s) \tag{9}$$

- CA signs the encrypted key using its secret identity SID_{CA}, $sig_c = sign_{SID_{CA}}(C)$, and then broadcasts the signed encrypted key, followed by the signature (C, sig_c, T^u_s) to all vehicles in the network.
- The vehicles that received the key first verify the timestamp, T^u_s, by checking its freshness and periodicity.
- The vehicles then verify the signature over the message using the public key of CA, PK_s. The vehicles, then, extract the new key by decrypting C using the old system key, k_s.
- After updating the system key, the TPD device of each vehicle generates a new hash chain of length n using the new system key, k^u_s, and stores k^u_s as shown in Fig 9.

F. EFFICIENT VEHICLE REVOCATION PROCESS

The most common means of certificate revocation methods in V2X is the use of Certificate Revocation List (CRL). In this method, a CA sends a list of the revoked certificate to all registered vehicles. Each vehicle needs to continuously obtain the updated CRL list from the CA, whereby it checks whether a certificate is revoked or not by searching for the certificate's ID in this list. Checking the vehicles' revoked certificates using the CRL method leads to long delays based on the CRL list size. The CRL list size increases as the number of revoked

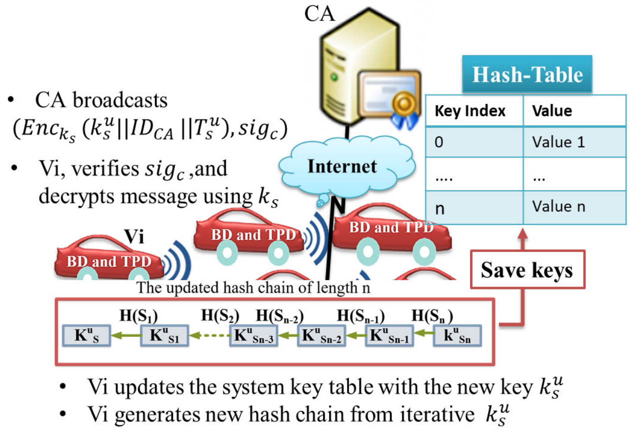

CA
Hash-Table
Key Index | Value
0 | Value 1
.... | ...
n | Value n
• CA broadcasts $(Enc_{k_s}(k_s^u||ID_{CA}||T_s^u), sig_c)$
• Vi, verifies sig_c ,and decrypts message using k_s
Internet
Vi
BD and TPD
BD and TPD
BD and TPD
Save keys
The updated hash chain of length n
$H(S_1)$
$H(S_2)$
$H(S_{n-2})$
$H(S_{n-1})$
$H(S_n)$
K^u_S
K^u_{S1}
K^u_{Sn-3}
K^u_{Sn-2}
K^u_{Sn-1}
k^u_{Sn}
• Vi updates the system key table with the new key k_s^u
• Vi generates new hash chain from iterative k_s^u

FIGURE 9. The system key and hash chain updating process.

certificates grow. The CRL list size can range between several bytes to megabytes as mentioned in [29]. The conventional CRL method is not suitable for large V2X networks, as it incurs excessive overhead. In contrast to the traditional CRL revocation mechanisms, this paper introduces a simple and efficient revocation protocol.

In this paper, we are not focused on misbehavior detection techniques. Instead, we utilize existing protocols for detecting misbehaviors. Interested readers, can refer to such a previous work of [30]. Once a vehicle is detected as misbehavior to be revoked, the CA decides to isolate this vehicle from the system by notifying the other vehicles with a revocation list. As an example of misbehavior detection scenario, if a vehicle receives false information or too many requests from a specific vehicle within a short time, it reports the pseudo-identity PID_i of this vehicle as misbehavior to the CA. We propose an efficient revocation protocol to revoke only the accused driver without affecting the other drivers for the same vehicle. CA has a database of all vehicles information and driver's real identities. Once CA receives the invalid vehicle identity PID_i, CA extracts PID_{init} and finds the real driver identity by searching v-info table – a fast search engine based on contents addressed memory (CAM) [31]. CA Extracts PID_{init} using Eq. (4) $\sim$ (6), and maps it to the real vehicle identity RID_i and the biological password $\beta_{i,u}$ of the driver.

CA directly communicates with the TPD device of the misbehaving vehicle by sending a signed a revocation message that contains the initial pseudo-identity of the revoked vehicle (PID_{init}, Sig_{rev}). Here Sig_{rev} is a digital signature of CA, which signs initial pseudo-identity PID_{init} by CA's secret identity (SID_{CA})

$$Sig_{rev} = Sign_{SID_{CA}}(PID_{init}) \tag{10}$$

When the TPD in the revoked vehicle receives this message, it verifies the legitimacy of the sender, if it's legitimate, TPD erases all the cryptographic materials previously configured in the TPD_i during the registration phase. Whereas the previous work called 2FLIP protocol [22] revokes the whole vehicle regardless of the misbehaving driver, our protocol can revoke the misbehaving driver individually without affecting the other drivers for the same vehicle.

VI. SECURITY ANALYSIS OF THE PROPOSED PROTOCOL

In this section, we analyze the security functions provided by the proposed protocol compared with recent security protocols that are based on TPD and BD devices.

A. SECURITY REQUIREMENTS ANALYSIS

In this subsection, we show how the proposed protocol can accomplish the required security functions: driver authentication, message authentication, non-repudiation, privacy-preserving, unlinkability, traceability, and system update.

1) DRIVER AUTHENTICATION

BD device verifies the user legitimacy by comparing the biometric information of each driver with the pre-stored biometric credentials. Based on the verifying function $Pv_{i,u}$ and the initial pseudo-identity PID_{init}, each BD device calculates the following: $\beta^*_{\mathrm{i,u}} = h(pw_{i,u})$, $Pv^*_{i,u}$ =h $(\beta^*_{i,u} \oplus PID_{init})$. If $Pv^*_{i,u}{==}\,Pv_{i,u}$, then the driver is authorized to login the TPD. The authentication phase between the BD and TPD increases the security level by preventing unauthorized drivers from sending messages or activating the TPD.

2) MESSAGE AUTHENTICATION

We proposed a novel MAC algorithm based on the hash chain table and one common secret key. The traditional MAC algorithm requires sending the MAC key with the signed message. In contrast, the proposed protocol introduces a pre-generated hash table that allows each vehicle to send only the key index. Each receiving vehicle uses this key index to look up the hash table and verify the received message. Using a hash chain table with indexed keys increases the immunity to key compromising attacks, and also can avoid disclosing any information about the key. Message authentication steps are described below: A sender vehicle v_i calculates a MAC signature over the transmitted message using HMAC-SHA256 algorithm $Sig_i = mac_{ki}(PID_i||m_{i,j}||Ts)$.

- v_i attaches $Key_{index}T_s$ and PID_i to the message
- Received vehicle v_j verifies the time stamp Ts to avoid the replay attacks. Ts defines the validity time of the pseudo-identity. v_j accepts the message if Ts is valid.
- Searching the hash table using the attached key index for k_i, v_j calculates the signature over the received message using $Sig_i^* = mac_{ki}(PID_i||m_{i,j}||Ts)$. Then v_j compares the calculated MAC value against the received one.
- v_i accepts the message if $Sig_i^* == Sig_i$ otherwise, reject the message. Thus, attackers cannot modify or alter a valid signature in polynomial time. Hence, message authentication and integrity are ensured.

3) NONREPUDIATION

Each message is transmitted with an attachment of pseudo-identity which is generated by the corresponding TPD of each vehicle. The pseudo-identity is generated using the vehicle's initial identity, the system key, and a timestamp.

Due to this combination, no vehicle or driver can deny the transmission of a message through its corresponding TPD. With the help of timestamp, it can never deny the time of message generation. Thus, nonrepudiation is guaranteed.

4) PRIVACY-PRESERVING

We proposed a method of using pseudo-identity PID_i to hide the real identity of each vehicle, which allows them to communicate anonymously and thus avoid vehicle tracking. Using BD and TPD allows each driver to authenticate his real identity by entering the biological password. If the identity of the driver is legitimate, the driver can activate TPD and sends messages without exposing the real identity of the vehicle. In our proposed protocol, we achieve the privacy of each vehicle, while allowing the CA to track the misbehaving vehicles using our revocation process. Even if BD or TPD are stolen, the driver's critical information is still preserved, as the activation of both devices requires the actual driver biometric information.

5) UNLINKABILITY

Instead of attaching the long size high-cost pseudonym certificates with each message to provide anonymity. Our protocol can preserve the vehicle's privacy using a short size low-cost random numbers called pseudo-identities to hide the real vehicle identity. While the conventional protocols periodically change pseudonym certificates like [32], [33] according to a predefined change and swapping mechanism, our proposed protocol dynamically generate the pseudo-identity numbers on the fly. After verifying the biometric information of each driver, the transmitter TPD device is activated to generate a dynamic identity per each message, while the receiving TPD authenticates the sender's message integrity. Using dynamic pseudo-identity in the proposed protocol can prevent any adversary from linking two identities and discovering the sender's real identity at a much lower computation overhead than the conventional protocols. The proposed protocol uses extensive pseudo-identity change which makes it hard for the attacker to link the newly changed pseudo-identity with the old one which preserves both the location and identity privacy. Due to the inherent weakness in the V2X standard's beaconing protocol, an attacker may be able to track a vehicle position. Our protocol, however, never reveals the real identity of a vehicle, since the real identity is securely stored only in the V-info table at CA. Each pseudo-identity is composed of two parts: The first part is $PID^1 = h(R_i)$ with a random number R_i changing every time a vehicle transmits a message. The second part is $PID_{init} \oplus h(PID^1||k_s)$ where a fixed value PID_{init} allows CA to track the vehicle, while the full pseudo-identity $PID_i = PID^1||PID^2$ makes it extremely hard to link it with the previous pseudo-identity. The proposed pseudo-identity structure can satisfy the anonymity since it allows only CA to retrieve the identity to process vehicle tracking and revocation.

6) TRACEABILITY

In the proposed protocol, CA extracts the real identities of the vehicles by searching the vehicle information table (v-info table). The dynamic pseudo-identity PID_i of the vehicle is obtained by using Eq. (4) $\sim$ (6). CA extracts the initial vehicle identity and finds the real identity RID_i by looking up the v-info table. The CA's conditional traceability is satisfied, while vehicle privacy is still preserved.

7) SECURE SYSTEM UPDATE

We introduce a periodic system key updating protocol to protect the system from key compromising attacks by allowing only the registered vehicles to generate a new hash chain table. CA generates a new system key k_s^u, and encrypts it using $C = Enc_{k_s}(k_s^u||ID_{CA}||T_s^u)$ with the current system key k_s and the current key timestamp T_s^u. CA signs the encrypted message using the secret identity SID_{CA}, attach a signature sig_c, and broadcasts it to all registered vehicles. Only the registered vehicles can verify the signature, decrypt the key message using the previous system key, and then update the hash chain table.

B. RESISTANCE TO ATTACKS

The proposed protocol is secure against many well-known attacks described below:

1) REPLAY ATTACK

The proposed protocol ensures the freshness of each transmitted message by attaching the current timestamp of the TPD device. For example, its message format is $\{PID_i, Sig_{ki}, m_{i,j}, Key_{index}, Ts\}$, where Ts is the attached time stamp. All vehicles must be synchronized to support accurate time stamps and resist against the replay attacks. The synchronization of OBU's clock can be provided by GPS devices.

2) MODIFICATION ATTACK

In the proposed protocol, the message integrity is achieved by attaching asignature $Sig_{ki}= mac_{ki}(PID_i||m_{i,j}||Ts)$, which is calculated using the HMAC algorithm with a random key. The receiver recalculates the signature over the received message using $Sig_i^* = mac_{ki}(PID_i||m_{i,j}||Ts)$. Only when signature Sig_i^* is identical to the transmitted signatureSig_i, the receiver accepts the message. In this way, by verifying the signature of each message our protocol ensures the message integrity and prevents-message alteration.

3) SYSTEM KEY COMPROMISING ATTACKS

In this paper, we proposed a Message Authentication method that employs the key index for each message. By attaching the key index to the message, it allows the receivers to lookup the pre-stored hash table in the receiver vehicle without

TABLE 2. The security comparison of the proposed protocol compared with the related mentioned protocols.

Security Functions and Attacks Security Properties	He et al. [19]	Zhou et al.[21]	Bayat et al. [18]	Wang et al. [20]	Proposed Protocol
Authentication and integrity	✓	✓	✓	✓	✓
Nonrepudiation	✓	✓	✓	✓	✓
Privacy preserving	✓	✓	✓	✓	✓
Unlinkability	✓	✓	✓	✓	✓
Traceability	✕	✕	✕	✓	✓
system update	✕	✕	✕	✓	✓
Resistance to Replay attack	✓	✓	✓	✓	✓
Resistance to Modification attack	✕	✓	✕	✓	✓
Resistance to key compromising attacks	✕	✕	✕	✕	✓
Resistance DOS attacks	✕	✕	✕	✓	✓
Resistance Man in the middle attack	✕	✕	✕	✕	✓

exposing the actual key. In contrast, the conventional MAC algorithm requires sending the key with a signature to allow the receivers to verify the message, which makes it vulnerable to sniffing key attacks.

4) DOS ATTACKS

The proposed protocol can enhance the resistance against denial of service (DOS) attacks. Conventional methods like TESLA protocol [9], often require a large buffer to store the incoming packets for a long time until the sender discloses the key to allow the receiver to verify the messages. In our protocol, each TPD device stores a hash chain of n entries during the initialization phase, which requires a significantly smaller buffer and lower calculation overhead for MACs than the conventional methods.

5) MAN IN THE MIDDLE ATTACK

In the proposed protocol, it is extremely difficult to retrieve the system key by sniffing the key index since only the registered vehicles can map the key indices to the hash key values. In the proposed protocol, furthermore, even if the attacker sniffs the initial pseudo-identity of any vehicle, it cannot retrieve any information about the real identity of the vehicle. In summary, the security level of the proposed protocol is compared against the previous work [18]–[21] that support the same network topology and also employ a TPD device.See Table 2.

VII. PERFORMANCE EVALUATION

To evaluate the performance of the proposed protocol, we implemented it in an NS3 simulator using a cryptography library called MIRACL [34]. To demonstrate the performance comparison, we also have implemented a set of previous protocols using the same NS-3 simulator platform [35]. We conducted an extended set of simulations using a wide range of example networks. We first compare the performance of the proposed protocol with the previous works [18]–[22] in terms of communication overhead and computation time. We then measured the average message delay and the average message loss ratio for all considered protocols.

A. COMPUTATION OVERHEAD OF TPD BASED PROTOCOLS

Table 3 shows the average execution time of primary cryptographic operations in our simulator. The simulations are conducted in a hardware platform employing an Intel Core I7-4770 processor with 3.40 GHz clock, and the main memory of 4 GB. In the simulations, we adopted the same experimental environment as [19] to make a direct comparison. Many V2X security protocols such as [18] are based on bilinear pairing to achieve their security requirements.

Since they are based on symmetric bilinear pairing to generate the cryptographic material with a security level of 80 bits, it would take 2^{80} trials for a hacker to break the security strength of these generated elements. The bilinear pairing can be defined by Eq (11).

$$e : G_1 \times G_1 \rightarrow G_T, \tag{11}$$

Here, G_1 indicates an additive group constructed by a generator P with order q on a supersingular elliptic curve $E : y^2 = x^3 + x \ mod\ p$ with degree 2. Here p denotes a 512-bit prime number, while q indicates a Solinas prime number of 160 bits. On the other hand, many other security protocols such as [19] are still using the traditional elliptic curve to construct their solutions While the bilinear pairing methods suffer from high computations cost due to their excessive complexity, other approaches such as [20], [21], and our proposed protocol provide lightweight solutions. To compare with the previous works based on bilinear pairing and ECC, we implemented the simulator using a security level of 80 bits.

In addition, the simulator employs an additive group G of order q, which is expressed by a non-singular elliptic curve $E : y^2 = x^3 + ax + b \ mod\ p$, where p, q are 160-bit prime numbers and a, b belongs to the finite field z_q^*.

The computation cost of [19] requires three ECC multiplication and three hashing to sign one message with a total time cost, $3T_{M-ECC}+3T_h = 1.338$ ms. The verification cost for [19] is calculated using three ECC multiplication, two-point addition, and two hashing operations. Therefore, the computation cost of one message is $3T_{M-ECC}+ 2T_{a-ECC}+ 2T_h = 1.3356$ ms. The above calculation ignores the computation time for concatenation and XOR operations since their computation time is negligibly short.

The computation cost of the following security protocols [18], [20], and [21], have been analyzed in the same manner using the listed calculations in Table 4. The proposed protocol requires only negligible hash and MAC operations

TABLE 3. The definition and processing time of the primary cryptographic operations of our NS3 Simulator.

Cryptographic operations	Definitions and Abbreviations	Average Execution Time (ms)
Bilinear Pairing	*TP*: the time needed to perform one bilinear pairing operation	4.3000
Scalar Multiplication in Bilinear Pairing	*TM-P*: the time needed to perform one multiplication over bilinear pairing	1.7000
Scalar Multiplication in ECC	*TM-ECC*: the time needed to perform one multiplication using ECC	0.4400
Map-to-point Hash Function in Bilinear Pairing	*Th-P*: the time for hash function to map a string to a point in group	4.4160
Message Authentication Code operation	*TMAC:* the time defined for one mac operation using HMAC algorithm	0.0167
Encryption Operation	*Tenc*: the time to perform one encryption operation using AES-128	4.0274
Decryption Operation	*T dec* : the time to perform one decryption operation using AES-128	4.1524
Hashing Operation	Th: the time defined for one hash function operation using SHA-256 algorithm	0.006
Small Scalar Multiplication in Bilinear Pairing	*TSM−P*: the execution time of a small scale multiplication operation xi · P related to the bilinear pairing, which is used in the small exponent test, where P ∈ G1, vi is a small random integer in [1, 2t] and t is a small integer number.	.0535
Small Scalar Multiplication in ECC	*TSM-ECC*: the execution time of small-scale multiplication operation in ECC xi · P where P ∈ G and vi is a small random integer in [1, 2t] and t is a small integer number.	.0139
Addition Operation in Bilinear Pairing	*Ta-P*: the execution time of addition operation using a bilinear pairing	0.0071
Addition Operation in ECC	*Ta-ECC*: the execution time of addition operation using based on ECC	0.0018
Digital signature signing time	TDSA: message signing time using digital signature algorithm with 1024 bits	0.45
Digital signature Verifying time	TDSAV: message verifying time using digital signature algorithm with 1024 bits	0.52

to authenticate messages with a little computation time. Any receiver can verify the message by calculating the MAC value using the key retrieved from an already stored hash chain. We assume that the BD device initially does the driver's authenticity then it activates the TPD to send and receive messages.

According to the standard DSRC communication requirements of V2X, each vehicle broadcasts beacon message every 100 or 300 ms. Table 5 compares the computation speed in term of the number of messages that can be signed and verified per second. It shows that the proposed protocolachieves two orders of magnitude higher speed than the previous methods that we experimented. To calculate the number of messages that need to be verified by each vehicle, we consider a high-vehicle-density scenario of 180 vehicles within a 300 m communication range. Assume that each vehicle sends a packet every 300 ms, so each vehicle must verify about 600–2000 messages per second [36]. From Table 5, we find that [19] and the proposed protocol can verify 600 messages per second, while only the proposed protocol can verify 2000 messages per second.

TABLE 4. Computation time comparison for message signature and verification.

VANET Security Protocols	Message Signing (ms)	Message Verification (ms)
He et al. [19]	*3TM-ECC +3Th =1.338*	*3TM-ECC + 2Ta-ECC + 2Th = 1.3356*
Wang et al. [20]	*Th + TMAC +Tenc= 4.0501*	*Th + TMAC + Tdec =4.1751*
Bayat et al. [18]	*5TM-P+Ta-P+2Th+ Th-P =12.9351*	*3 TP + TSM−P + Th-P + Th =17.3755*
Zhou et al. [21]	*2TM-ECC +4Th =.904*	*7TM-ECC + 2Ta-ECC + 4Th =3.1076*
Proposed	*TMAC = 0.0167*	*TMAC = 0.0167*

TABLE 5. Comparison of computation time for signature and verification.

VANET Security Protocols	The number of signed messages per second	The number of verified messages per second
He et al. [19]	747	748
Wang et al. [20]	246	239
Bayat et al. [18]	77	57
Zhou et al. [21]	1106	321
Proposed	59880	59880

B. COMMUNICATION OVERHEAD OF TPD BASED PROTOCOLS

In this section, we present the communication cost analysis of the proposed protocol and compare it with the previous work [18]–[21]. In this paper, we compare two cryptography methods: one based on bilinear pairing and the other based on the traditional elliptic curve. According to the previously mentioned parameters of the bilinear and elliptic methods, we calculate the size of the cryptographic elements. For the bilinear method, we used an additive group G1 to support 128 bytes elements size. For the elliptic curve method, we used an elliptic group G to support 40 bytes element size, In addition, we assume that the output size of the hash function is 20 bytes, the size of elements in the finite field Z_q^* is 20 bytes, and the timestamp size is 4 bytes.

To compare the overhead with the protocols in [18]–[21] in Table 6, we analyzed the message structures of the 4 previous protocols. The message structure of [19] is expressed by Eq (12).

$$\sigma_i|\,|AID_i|\,|R_i||T_i \tag{12}$$

TABLE 6. Message communication cost of the proposed scheme and TPD based protocols.

VANET Security Protocols	He et al. [19]	Wang et al. [20]	Bayat et al. [18]	Zhou et al. [21]	Proposed protocol
Communication Overhead (Bytes)	144	60	280	104	40

Eq. (13) contains a signature $\sigma_i \in Z_q^*$, a pseudo-identity AID_i, a random number $Ri \in G$, and a timestamp T_i. Here, $AID_i = \{AID_i^1, AID_i^2\}$, which belongs to the elliptic curve group G, The total communication overhead of one message in the previous method of [19] is calculated by Eq.12 as $20{+}40 \times 3 + 4 = 144$ bytes.

In [18], the broadcast message parameters are expressed using Eq (13).

$$mac_{m,ts}||EPID_{i,ts} \tag{13}$$

Here, $mac_{m,ts}$ represents a MAC signature of 20 bytes over the message, while $EPID_{i,ts} \in G$ represents the vehicle pseudo-identity of 40 bytes. The total communication overhead of sending one message, therefore, is 60 bytes according to Eq. (13).

For the protocol proposed in [18], the parameters for transmitted messages are calculated using Eq (14):

$$ID_i|\,|\sigma_i|\,|T_i \tag{14}$$

Here, a pseudo-identity $ID_i \in G1$ consists of two parts (ID_i^1, ID_i^2): a message signature $\sigma i \in Z_{q,}^*$ and a timestamp T_i. Its communication overhead of one message is $128 \times 2 + 20 + 4 = 280$ bytes.

In [21], the transmitted message structure is calculated using Eq (15):

$$U_i|\,|\sigma_i|\,|w_i|\,|\theta_i|\,|T_i \tag{15}$$

Here, θi, wi, σi are random numbers that belong to Z_q^*, T_i indicates the timestamp, while $U_i \in$ G denotes a random number. The total communication overhead for the message of Eq. (15) is $40 + 20 \times 3 + 4 = 104$ bytes.

For our proposed protocol, the transmitted message structure is represented by Eq (16).

$$PID_i|\left|Sig_{ki}\right||Key_{index}||T_s \tag{16}$$

Here, $PID_i \in Z_q^*$ represents the pseudo-identity, while Sig_{ki} indicates the 12 bytes truncated signature over the message. Key_{index} represents the index of the hashed key, while T_S represents the time stamp. The total communication overhead of one message is $20{+}12{+}4{+}4 = 40$ bytes.

Table 6 compares the communication overhead of the 5 protocols. The proposed protocol exhibits 20% ∼ 85% lower communication overhead than the previous methods in Table 6.

TABLE 7. Computation time comparison for 2FLIP and the proposed protocol.

Computation cost (ms)	2FLIP [22]	Proposed protocol
Signing process - V2V message	$7Th + TMAC = .0587$	$TMAC = .006$
Verification process - V2V message	$2Th + TMAC = .0287$	$TMAC = .006$
Signing process - key update message	$Th + 2TP + TDSA + Tenc = 13.1534$	$Tenc + TDSA = 4.4774$
Verification process - key update message	$Tdec + TDSAV = 4.6724$	$Tdec + TDSAV = 4.6724$
Signing process - Revoking message	$2TP = 8.6$	$TDSA = 0.45$
Verification process - Revoking message	$TDSAV = 0.52$	$TDSAV = 0.52$

TABLE 8. 2FLIP and the proposed protocol message communication cost.

Communication cost (Bytes)	2FLIP [22]	Proposed protocol
V2V message	47	40

C. OVERHEAD OF BD BASED PROTOCOLS

In this section, we compare the proposed protocol with the 2FLIP protocol [22]. 2FLIP is a protocol that utilizes biometric device authentication. To calculate the computation time and communication overhead of 2FLIP, we use the cryptographic calculations listed in Table 3.

Table 7 summarizes the computation time of the proposed protocol and 2FlIP for the following seven functions: message signing, verifying, key updating, hash chain updating, and vehicle revocation.

In 2FLIP, the message structure is calculated using Eq (17).

$$PID_{i,ts}||\sigma_{i,ts}||T_s||m \tag{17}$$

Here, $PID_{i,ts}$ represents the dynamic-identity of each vehicle and $\sigma_{i,ts}$ indicates the MAC signature over a message m. The signing computation cost of one message in 2FLIP is $(7Th + TMAC)$, while the verifying cost is $(2Th + TMAC)$ as mentioned in [22].

The proposed protocol uses one MAC signature $(TMAC)$ for signing and verifying messages. Similarly, the computation times of the key update and revocation are listed in Table 7.

It can be observed from Table 7 that the computation time of the proposed protocol is substantially shorter than the 2FLIP protocol in the functions for message signing, verifying, key updating and revoking. The communication cost of the proposed protocol and 2FLIP is listed in Table 8. In 2FLIP, a message consists of $\{PID_{i,ts}, \sigma_{its}, T_s, m\}$ with $PID_{i,ts}$ of 23 bytes, $\sigma_{i,ts}$ of 20 bytes, and timestamp of 4 bytes leading to a size of 47 bytes. The security function coverage is summarized in Table 9.

From Table 9, we can observe that 2FLIP cannot support some security functions such as traceability,

TABLE 9. The security function coverage of the proposed protocol compared with 2flip.

Security Functions	2FLIP [22]	Proposed protocol
Authentication and integrity	✓	✓
Nonrepudiation	✓	✓
Privacy preserving	✓	✓
Unlinkability	✓	✓
Traceability	✗	✓
system update	✓	✓
Resistance to Replay attack	✓	✓
Resistance to Modification attack	✗	✓
Resistance to key compromising attacks	✗	✓
Resistance DOS attacks	✓	✓
Resistance Man in the middle attack	✓	✓
Efficient revoking mechanism	✗	✓

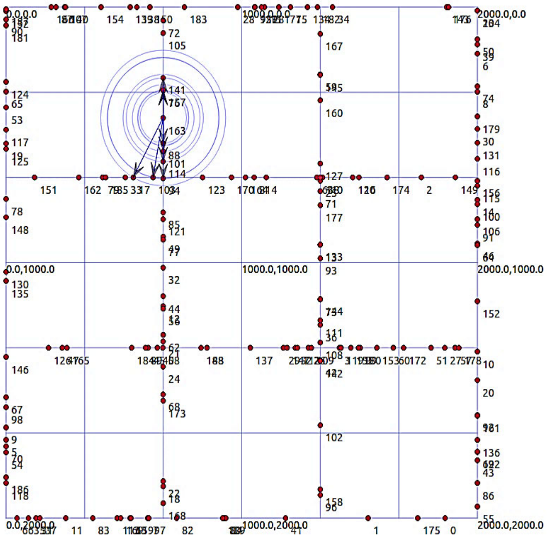


FIGURE 10. Manhattan grid scenario corresponding to a square area of size 2000 × 2000 m2.

resistance to modification attacks, and key compromising attacks. 2FLIP uses a single system key for vehicles authentication which makes 2FLIP vulnerable to key compromising attacks. In contrast, the proposed protocol employs n hashed distinct keys without exposing the system key, and thus it can resist system key attacks.

D. SIMULATION RESULTS

We have conducted an extensive set of simulations to compare the proposed protocol performance with previous protocols of [18]–[22]. We have implemented the proposed protocol using the miracle library on the NS-3 simulator. We implemented all cryptographic operations described in the above section including signing and verifying the messages for the previous protocols as well as the proposed protocol. We conducted simulations using Manhattan grid mobility to model an urban scenario of vehicular networks. Fig. 10 shows the simulated network topology that is a square area of 2000x2000 m^2. Vehicles travel at a speed in the range of [40]–[50] km/h along the roads. The simulations assume that the vehicles are equipped with IEEE802.11p radios and communicate over a two-ray ground propagation channel. All simulation parameters are listed in Table 10. The performance of each protocol has been measured in terms of transmission delay and message loss ratio, which are described below.

TABLE 10. Simulation parameters.

Simulation parameter	Value
Simulator	Ns-3.26
City simulation range	2000m x 2000m
Communication range	250 m
Simulation time	100 s
Channel bandwidth	6 Mbps
Buffer size	8 Kbytes
Broadcast interval	0.1 s
Vehicle speed	40-50 km/h
Mobility model	Manhattan Grid
Propagation loss model	Two-ray ground-reflected propagation
TX power	20 dBm
Data payload size	200 bytes
Wireless technology	IEEE802.11p

1) AVERAGE TRANSMISSION DELAY

It represents the average end-to-end delay consisting of the singing time, transmission time, and verifying time. It is calculated using Eq. (18):

$$\begin{aligned}\text{AvgMsgdelay} = & \frac{1}{N_{SA}.Msgsent_n.K_n} \\ & .\sum_{n\in SA}\sum_{m=1}^{Msgsent_n}\sum_{k=1}^{kn} \\ & .\left(t_{sign}^{nm}+t_{trans}^{nmk}+t_{verifing}^{nmk}\right) \\ & .\left(1+Length_k\right) \qquad (18)\end{aligned}$$

Here, SA represents the total simulation area, N_{SA} is the total number of vehicles in SA, $Msgsent_n$ is the number of messages sent by vehicle n, and K_n is the number of vehicles located within the one-hop communication range of vehicle n.

t_{sign}^{nm} represents the signing time of message m by vehicle n, t_{trans}^{nmk} is the transmission time of message m sent from vehicle n to vehicle k, $t_{verifing}^{nmk}$ is the verifying time of message m received by vehicle k from vehicle n, and $Length_k$ is the buffer length of vehicle k.

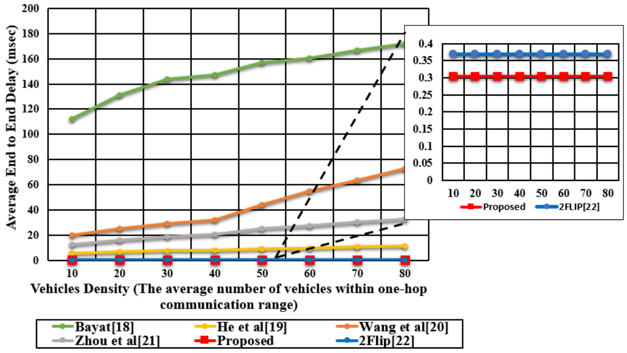


FIGURE 11. The impact of vehicle density on the average message End2End.

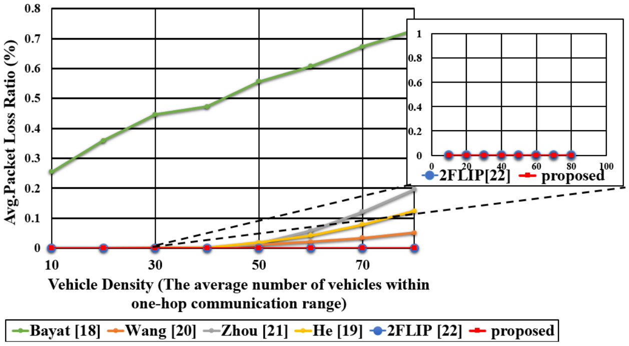


FIGURE 12. The impact of vehicle density on the packet average loss ratio.

2) AVERAGE MESSAGE LOSS RATIO

From the security application's perspective, the received message might be dropped due to the limited size of the application buffer. As the signature verification function consumes time, the application buffer may overflow if the message arrival rate is higher than the message processing rate. Thus, we calculate the average loss ratio by Eq. (19).

$$AvgMsgLR = \frac{1}{N_{SA}} \sum_{n=1}^{N_{SA}} \frac{Msg^n_{dropped}}{\sum_{k=1}^{kn} Msg^n_{arrived}} \tag{19}$$

Here, $Msg^n_{dropped}$ denotes the number of messages dropped at the application buffer of vehicle n, while $Msg^n_{arrived}$ denotes the total number of messages received by vehicle k.

In this paper, we do not consider the message loss due to the wireless medium. Instead, we only consider the message loss caused by the application buffer overflow at the receiver vehicles. Fig. 11 illustrates the simulation results of the average transmission delay over a wide range of the number of vehicles in the network. While Bayat [18], He [19], Zhou [21] show a steep increase in the average transmission delay as the number of vehicles in the network increases, the other protocols exhibit relatively flat delay. The proposed protocol gives a significantly shorter delay than all the previous protocols tested.

It is important to keep the end-to-end delay as low as possible especially in V2X reporting and safety applications. The average message delay of Bayat *et al.*, [18] protocol shows the longest delay, which is due to the fact that its pairing operation consumes a lot of singing and verifying delay as listed in Table 5. The end-to-end delay of the other protocols [19]–[21] are relatively close since their protocols employ ECC cryptographic operations, and thus consume less singing and verifying time than pairing operations. Fig. 11 also shows that the proposed protocol slightly outperforms 2FLIP, since it has a lower computation cost for both signing and verification processes (see Table 8). For instance, for a vehicle density of 80 vehicles, the proposed protocol incurs a delay of 304 μsec, while 2FLIP incurs around 368 μsec. As shown in Table 7, for the computation cost of singing one message, the proposed protocol consumes only TMAC = 0.006, whereas 2FLIP consumes 7Th + TMAC = 0.0587. In addition, at the receiver side, the proposed protocol consumes only TMAC = 0.006, whereas 2FLIP consumes 2Th + TMAC = 0.0287. Fig. 12 illustrates the average packet loss ratio for the protocols that are implemented and tested.

In Fig. 12, for a vehicle density of 80 vehicles, [18] shows the highest loss ratio (72%). Its excessive loss ratio is due to the buffer overflow that happens because its message arrival rate is much higher than the message verification rate. Therefore, the loss ratio is proportional to the computation cost of signature verification. In the same context, He *et al.* [19], Wang *et al.* [20] and Zhou *et al.* [21] show lower loss ratios as they have lower computation cost for signature verification. The proposed protocol and 2FLIP provide zero losses for the range of vehicles densities and the application buffer size of 8 Kbytes in the simulations that we have tested. As the signature verification requires only 304 μsec, the application buffer does not overflow for any of the simulated vehicle densities. The lightweight verification method of the proposed protocol substantially reduces the processing delay, and consequently, it can prevent the message loss entirely.

The above simulation results demonstrate that the hash chain based key generation and the lightweight signature computation of the proposed protocol can significantly reduce both computation and network overhead without compromising the security level.

VIII. CONCLUSION

In this paper, we proposed a decentralized light-weight authentication protocol for V2V communications. Our protocol preserves privacy by using self-generation of pseudo-identities instead of traditional digital certificates. Moreover, the proposed protocol integrates the BD and TPD security devices with the pre-stored shared hash chain of authentication keys. The hardware security devices play the role of CA agents, and generate pseudo-identities and corresponding private keys to authenticate the messages and keep the driver's privacy. The use of pre-stored shared hash chain offered a new message integrity mechanism that calculates a signature over each message using the hashed keys without disclosing the keys. The proposed protocol satisfies important security requirements such as anonymity, unlinkability, and conditional traceability. It also provided a misbehavior revocation mechanism that reports the misbehaviors' pseudo IDs, so the CA can determine the real identity of from the pseudo IDs. In the extensive simulation experiments, our protocol outperformed the recently published protocols in the computation cost of message authentication and verification. We have shown that the proposed protocol can sign 60,000 messages per second which is up to 55 times higher speed than the previous protocols tested. Therefore, we conclude that the proposed protocol is well suited to time-critical applications such as large scale V2X networks. The revocation method introduced in this paper supports the multi-driver revocation scenario since it revokes only the accused driver by allowing the CA to revoke only the driver's pseudo-identity.

REFERENCES

[1] *An Overview of the DSRC/WAVE Technology*. Accessed: Aug. 26, 2019. [Online]. Available: https://eudl.eu/doi/10.1007/978-3-642-29222-4_38

[2] H. Hasrouny, A. E. Samhat, C. Bassil, and A. Laouiti, "VANet security challenges and solutions: A survey," *Veh. Commun.*, vol. 7, pp. 7–20, Jan. 2017. doi: 10.1016/j.vehcom.2017.01.002.

[3] D. Mathew and H. A. Roy, "A survey on different privacy-preserving authentication schemes in VANET," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 396, no. 1, 2018, Art. no. 012033. doi: 10.1088/1757-899x/396/1/012033.

[4] A. Wasef, R. Lu, X. Lin, and X. Shen, "Complementing public key infrastructure to secure vehicular ad hoc networks [security and privacy in emerging wireless networks]," *IEEE Wireless Commun.*, vol. 17, no. 5, pp. 22–28, Oct. 2010. doi:10.1109/mwc.2010.5601954.

[5] J. Zhang, X. Tan, X. Wang, A. Yan, and Z. Qin, "T2FA: Transparent two-factor authentication," *IEEE Access*, vol. 6, pp. 32677–32686, 2018. doi: 10.1109/access.2018.2844548.

[6] R. G. Engoulou, M. Bellaïche, S. Pierre, and A. Quintero, "VANET security surveys," *Comput. Commun.*, vol. 44, pp. 1–13, May 2014.

[7] C. Hodge and M. R. Singer, "Telematics framework for federal agencies: Lessons from the marine corps fleet," Nat. Renew. Energy Lab., Golden, CO, USA, Tech. Rep. NREL/TP-5400-70223, 2017.

[8] H. Krawczyk, M. Bellare, and R. Canetti, *HMAC: Keyed-Hashing for Message Authentication*, document RFC: 2104, 1997, pp. 1–11.

[9] A. Perrig and J. D. Tygar, "TESLA broadcast authentication," in *Secure Broadcast Communication*. Boston, MA, USA: Springer, 2003.

[10] A. Studer, F. Bai, B. Bellur, and A. Perrig, "Flexible, extensible, and efficient VANET authentication," *J. Commun. Netw.*, vol. 11, no. 6, pp. 574–588, Dec. 2009.

[11] J.-L. Huang, L.-Y. Yeh, and H.-Y. Chien, "ABAKA: An anonymous batch authenticated and key agreement scheme for value-added services in vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 60, no. 1, pp. 248–262, Jan. 2011.

[12] X. Jia, X. Yuan, L. Meng, and L. Wang, "EPAS: Efficient privacy-preserving authentication scheme for VANETs-based emergency communication," *J. Softw.*, vol. 8, no. 8, pp. 1–8, 2013.

[13] P. Kamat, A. Baliga, and W. Trappe, "Secure, pseudonymous, and auditable communication in vehicular ad hoc networks," *Secur. Commun. Netw.*, vol. 1, no. 3, pp. 233–244, 2008.

[14] X. Lin and R. Lu, "GSIS: Group signature and ID-based signature-based secure and privacy-preserving protocol," in *Vehicular Ad Hoc Network Security and Privacy*. Piscataway, NJ, USA: IEEE, 2015, pp. 21–49. doi: 10.1002/9781119082163.ch2.

[15] H. Lu, J. Li, and M. Guizani, "A novel ID-based authentication framework with adaptive privacy preservation for VANETs," in *Proc. Comput., Commun. Appl. Conf.*, Jan. 2012, pp. 345–350.

[16] C. Zhang, R. Lu, X. Lin, P.-H. Ho, and X. Shen, "An efficient identity-based batch verification scheme for vehicular sensor networks," in *Proc. IEEE 27th Conf. Comput. Commun. (INFOCOM)*, Apr. 2008, pp. 246–250.

[17] S. Biswas and J. Misic, "Deploying proxy signature in VANETs," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2010, pp. 1–6.

[18] M. Bayat, M. Barmshoory, M. Rahimi, and M. R. Aref, "A secure authentication scheme for VANETs with batch verification," *Wireless Netw.*, vol. 21, no. 5, pp. 1733–1743, 2015.

[19] D. He, S. Zeadally, B. Xu, and X. Huang, "An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 12, pp. 2681–2691, Dec. 2015.

[20] M. Wang, D. Liu, L. Zhu, Y. Xu, and F. Wang, "LESPP: Lightweight and efficient strong privacy preserving authentication scheme for secure VANET communication," *Computing*, vol. 98, no. 7, pp. 685–708, 2014.

[21] Y. Zhou, S. Liu, M. Xiao, S. Deng, and X. Wang, "An efficient V2I authentication scheme for VANETs," *Mobile Inf. Syst.*, vol. 2018, Dec. 2018, Art. no. 4070283.

[22] F. Wang, Y. Xu, H. Zhang, Y. Zhang, and L. Zhu, "2FLIP: A two-factor lightweight privacy-preserving authentication scheme for VANET," *IEEE Trans. Veh. Technol.*, vol. 65, no. 2, pp. 896–911, Feb. 2016,

[23] J. G. Dyer, M. Lindemann, R. Perez, R. Sailer, L. V. Doorn, and S. W. Smith, "Building the IBM 4758 secure coprocessor," *Computer*, vol. 34, no. 10, pp. 57–66, Oct. 2001.

[24] X. Lin, X. Sun, X. Wang, C. Zhang, P. H. Ho, and X. Shen, "TSVC: Timed efficient and secure vehicular communications with privacy preserving," *IEEE Trans. Wireless Commun.*, vol. 7, no. 12, pp. 4987–4998, Dec. 2008.

[25] P. D. Pingle, "A survey of latest trends in cryptography and elliptic curve cryptography," *Int. J. Sci. Res. Educ.*, vol. 4, no. 5, pp. 5294–5301, 2016. doi: 10.18535/ijsre/v4i05.03.

[26] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comput.*, vol. 48, no. 177, pp. 203–209, 1987. doi: 10.2307/2007884.

[27] L. Lamport, "Password authentication with insecure communication," *Commun. ACM*, vol. 24, no. 11, pp. 770–772, Nov. 1981.

[28] D. Eastlake, *HMAC SHA TSIG Algorithm Identifiers*, document RFC: 4635, 2006.

[29] G. Rigazzi, A. Tassi, R. J. Piechocki, T. Tryfonas, and A. Nix, "Optimized certificate revocation list distribution for secure V2X communications," in *Proc. IEEE 86th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2017, pp. 1–7.

[30] K. Benahmed, M. Merabti, and H. Haffaf, "Distributed monitoring for misbehaviour detection in wireless sensor networks," *Secur. Commun. Netw.*, vol. 6, no. 4, pp. 388–400, 2012.

[31] S. A. Naby, S. Arslan, and H. Kim, "IKE hardware engine based on CAM for concurrent processing of massive user sessions," in *Proc. 13th Int. Comput. Eng. Conf. (ICENCO)*, Dec. 2017, pp. 154–159.

[32] J. Petit, F. Schaub, M. Feiri, and F. Kargl, "Pseudonym schemes in vehicular networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 228–255, 1st Quart., 2014.

[33] A. Boualouache, S.-M. Senouci, and S. Moussaoui, "A survey on pseudonym changing strategies for vehicular ad-hoc networks," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 770–790, 1st Quart., 2017.

[34] GitHub. (Jun. 6, 2018). *miracl/MIRACL*. Accessed: Mar. 7, 2019. [Online]. Available: https://github.com/miracl/MIRACL

[35] Nsnam. *NS-3.29 NS-3: ns3::MinstrelHt WiFi Manager Class Reference*. Accessed: Mar. 7, 2019. [Online]. Available: https://www.nsnam.org/releases/ns-3-29/

[36] Y. Saleh, F. Mahmood, and B. Abderrahim, "Performance of beacon safety message dissemination in Vehicular Ad hoc NETworks (VANETs)," *J. Zhejiang Univ.-Sci. A*, vol. 8, no. 12, pp. 1990–2004, Nov. 2007. [Online]. Available: https://link.springer.com/article/10.1631/jzus.2007.A1990

SHIMAA A. ABDEL HAKEEM was born in Egypt. She received the B.S. and M.S. degrees in communication and electronic engineering from the University of Fayoum, Egypt, in 2010 and 2014, respectively. From 2011 to 2015, she was a Network Administrator in Fayoum University Network Project. She has been a Research Assistant at the Electronics Research Institute, Egypt, since 2016. She is currently pursuing the Ph.D. degree with the Mixed-Signal Integrated System Laboratory, Chungbuk National University. Her research interests include wireless sensor networks, security protocols, vehicular ad hoc Networks, and routing protocols. She was a recipient of the Korean Information and Communication Conference Best Paper Award for her paper "Efficient Vehicular Network Authentication Using Aggregate Message Authentication Code", in 2018.

MOHAMED A. ABD EL-GAWAD received the B.Sc. degree in electrical engineering from Assiut University, Assiut, Egypt, in 2006, and the M.Sc. degree in electrical engineering from the Arab Academy for Science and Technology, Cairo, Egypt, in 2013. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Chungbuk National University, South Korea. From 2008 to 2015, he was with the National Telecommunication Institute, Egypt, where he conducted professional training as well as research on wireless communications. Since 2016, he has been with the MSIS Laboratory, Cheongju, South Korea, as a Student Researcher. His current research interests include wireless sensor networks and vehicular communications.

HYUNGWON KIM (M'95) received the B.S. and M.S. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), in 1991 and 1993, respectively, and the Ph.D. degree in electrical engineering and computer science from the University of Michigan, Ann Arbor, MI, USA, in 1999. In 1999, he joined Synopsys Inc., Mountain View, CA, USA, where he developed electronic design automation software. In 2001, he joined Broadcom Corporation, San Jose, CA, USA, where he developed various network chips, including a Wifi gateway router chip, a network processor for 3G, and 10 gigabit ethernet chips. In 2005, he founded Xronet Corporation, a Korea based wireless chip maker, as a CTO and CEO, he managed the company to successfully develop and commercialize wireless baseband and RF chips and software, including WiMAX chips supporting IEEE 802.16e and Wifi chips supporting IEEE 802.11a/b/g/n. Since 2013, he has been with the Department of Electronics Engineering, Chungbuk National University, Cheongju, South Korea, as an Associate Professor. His current research interests include the areas of sensor read-out circuits, touch screen controller SoC, wireless sensor networks, wireless vehicular communications, mixed signal SoC designs for low power sensors, and bio-medical sensors.