IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 December 2024, accepted 11 December 2024, date of publication 26 December 2024, date of current version 2 January 2025.

Digital Object Identifier 10.1109/ACCESS.2024.3523358

RESEARCH ARTICLE

Pseudo-Random Identification and Efficient Privacy-Preserving V2X Communication for IoV Networks

SUJASH NASKAR[1], (Member, IEEE), GERHARD HANCKE[2], (Fellow, IEEE), TINGTING ZHANG[1], AND MIKAEL GIDLUND[1], (Senior Member, IEEE)

[1]Department of Computer and Electrical Engineering (DET), Mid Sweden University, 852 30 Sundsvall, Sweden
[2]Department of Computer Science, City University of Hong Kong, Hong Kong

Corresponding author: Sujash Naskar (sujash.naskar@miun.se)

This work was supported by the Project "Next Generation Industrial Internet of Things (NIIT)," funded by the Swedish Knowledge Foundation (KKS).

ABSTRACT The advancement of Internet of Vehicles (IoV) technologies has significantly enhanced road safety and transportation efficiency through smart traffic management and precise control systems. With the advent of 5G and beyond, vehicles within the IoV ecosystem can seamlessly communicate with various smart entities (X) using V2X (Vehicle-to-Entity) communications. However, the openness of IoV networks and the exponential growth of V2X links have expanded potential attack surfaces, increasing the risk of security and privacy breaches. In response to these challenges, this article proposes a privacy-preserving and secure communication framework for IoV networks, addressing critical security challenges in V2X communication. By leveraging lightweight cryptographic mechanisms such as hash functions, quadratic residuosity, and Legendre symbols, the proposed scheme ensures secure authentication, group key sharing, and pseudonym management within IoV networks. The proposed scheme's security and privacy features, along with its correctness, have been rigorously validated against various security threats where other state-of-the-art schemes fail. Comprehensive performance analysis demonstrates that our scheme completes authentication in a fraction of a millisecond, significantly outperforming existing approaches. The design simplicity and efficiency of the proposed authentication structure make it highly suitable for real-world IoV applications.

INDEX TERMS Internet of Vehicles, vehicular ad-hoc networks, preserving privacy, V2X communication, security attacks, anonymous authentication.

I. INTRODUCTION

The rapid development and significant advancements in Internet of Vehicles (IoV) technologies, together with Fifth-Generation (5G) wireless technology, have enabled vehicles to communicate with other entities (V2X) faster and more reliably [1]. In IoV, a vehicle is equipped with an On-Board-Unit (OBU), which is responsible for security in all Vehicle to Entity (V2X) communications. This infrastructure includes Road Side Units (RSU) and Trusted Authority (TA). As the name suggests, the TA is assumed to be trusted and centrally controls all security and privacy-related communications. RSUs are placed by the roadsides physically. It interacts with the vehicles and TA using a public channel like the Internet. RSU gathers information about local traffic, weather, and any alerts or emergencies and provides the information to the vehicles with authenticity and confidentiality. Each vehicle (V) must first be authenticated by the Trusted Authority (TA) before it is allowed to communicate with other entities in the IoV network [2]. The entities (X) in IoV can be any smart device or unit communicating with the vehicle(s), e.g., smart parking, mobile devices, smart charging station etc. Each of these entities (X) has a Tamper Protected Unit (TPU), which is similar to OBU and dedicated to secure V2X communication for the entity [3]. Authentication and all other communications happen through the internet; therefore,

The associate editor coordinating the review of this manuscript and approving it for publication was Huan Zhou.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

enforcing security in these communications is mandatory. So far, many cryptography-based security solutions have been proposed in this area, but with the rapid growth of 5G and 6G technologies and an increasing number of attack surfaces [4], those solutions are no longer safe or efficient. Preserving user privacy in automated vehicles is a very crucial and difficult task [5], as each time a vehicle communicates, its identity needs to be anonymous so that an adversary can not trace the vehicle to gather sensitive private information such as the original identity of the vehicle-user, its traffic routes, etc. Compromising sensitive private information allows the adversary [6] to perform illegal activities or pre-planned road accidents or robberies. With a weak security model, an adversary can gain access to a vehicle's control system and perform life-threatening terror attacks. Complex and inefficient security schemes are useless for IoV networks as the communication is time-critical [7], and the security scheme needs to be fast and lightweight. Thus, there is a great need to develop new security solutions to guarantee the safety of autonomous transportation with 5G and beyond technologies [2].

In this research work, we have focused on developing an efficient anonymous authentication technique with promising privacy guarantees and a group key-sharing technique for broadcasts of common information within a group of vehicles with strong security. We have also proposed a general and common method to share a token between two communicating parties for direct V2X communication using only anonymous identity, regardless of the type of X. By minimizing the values for the crucial parameters (CPs) such as execution time, complexity, communication time, the storage and energy costs, and maximizing the scalability facts and types of attacks that can be prevented, our research goals are as follows:

- A fast, anonymous, pseudo-random-identity-based privacy-preserving authentication between each vehicle to TA using lightweight cryptos such as general cryptographic hash, prime multiplication, and pseudo-random identity generation.
- Use of Quadratic residuosity to share a group key (K) of variable lengths (n) and use it for broadcast from RSU/TA only.
- A general method of token sharing by using hash functions and random nonce values to establish V2X communication.
- A promising privacy perseverance and guaranteed security validity for each phase of communication in the proposed scheme.

The organization of the paper is as follows: Our contributions, together with a generalized survey on IoV architecture, privacy, and security needs, and the limitations present in related works, are presented in section II. Section III describes all the necessary system components, assumptions, and methodologies for the proposed scheme are described. The proposed scheme is presented in section IV, and the correctness and security validity are described in section V.

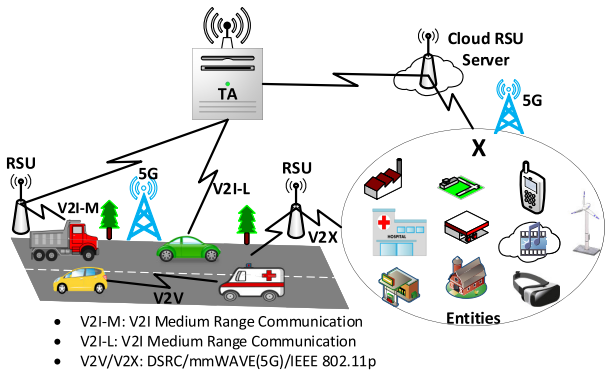

FIGURE 1. Example of a generalized IoV network.

In section VI, the proposed scheme is analyzed against various different security attacks. Section VII represents the performance analysis, and section VIII concludes the final remarks with the future directions and summarizes the research work.

II. LITERATURE REVIEW AND CONTRIBUTIONS

The architecture of IoV, described in various literature, and the different security and privacy techniques used to ensure safety in IoV communications are presented in this section.

A. RELATED WORKS AND LIMITATIONS

IoV networks enable seamless communication between vehicles and facilities such as smart parking stations, EV charging stations, smart grids, and smart homes. Advancements like V2V-based EV charging reservation systems proposed by Cao et al. [8] and cooperative edge caching using multi-agent reinforcement learning in IoV proposed by Zhou et al. [9] highlight the focus on optimizing communication and resource management in vehicular networks. These contributions further underscore the enormous possibilities of V2X communication in enhancing efficiency and scalability in IoV systems. However, identity and location privacy of vehicles in V2X is often overlooked, which is a critical and challenging task in IoV, as also pointed out by [3], [4], and [10]. Over the years, various methods have been proposed to enhance privacy, including anonymous pseudorandom identity-based (PRID) authentication, blockchain-based authentication [11], [12], [13], machine learning-based intrusion detection systems (IDS), and encryption-based mechanisms [14], [15], [16], [17]. PRID authentication uses anonymous temporary identities mapped with cryptographic functions, allowing only the TA and intended unit to verify identities [18], [19]. While PRID techniques are resource-efficient and suitable for time-critical IoV applications [20], poor implementation of mapping and cryptographic functions can lead to security vulnerabilities. Thus, a firmly secured and efficient PRID-based IoV authentication scheme with provable security and privacy guarantees is still lacking in the literature.

Islam et al. [19] proposed a password-based random identification scheme for privacy-preserving authentication

TABLE 1. Comparative study on secure communication facilities provided by various schemes.

Schemes	F1	F2	F3	F4
Debiao et al. [27]	✓	✓	✗	✗
Hafizul et al. [19]	✓	✓	✓	✗
Jing et al. [23]	✓	✗	✓	✗
Seyed et al. [28]	✓	✗	✓	✗
Trupil et al. [22]	✓	✗	✗	✓
Chien et al. [18]	✓	✗	✗	✗
Yujue et al. [29]	✓	✗	✗	✗
Himani et al. [26]	✓	✗	✗	✓
Pandi et al. [5]	✓	✗	✓	✓
Shiang et al. [3]	✓	✗	✗	✗
Proposed	✓	✓	✓	✓

Note: For different communications, some abbreviations are used as F1: Authentication; F2: Use of Temporary IDs; F3: Sharing of Group Keys; F4: Establishing Secure V2X Communication; ✓: Provide; ✗: Not Provide.

and group key agreements, claiming it to be lightweight. However, it is vulnerable to non-repudiation, spoofing, and traceability attacks, and requires an exhaustive database search by the TA for each authentication, making it inefficient. Vijayakumar et al. [5], introduced a privacy-preserving authentication and key-sharing scheme using pseudo-random identity and public-private key agreements. Although effective, it is unsuitable for time-critical IoV applications due to the high computational load from generating and verifying multiple keys and signatures, and it is susceptible to DoS attacks, non-repudiation issues, and message replay. Chen et al. [18] presented a pseudo-random identity-based scheme to address privacy issues in Ying and Nayak [21]'s proposal, claiming protection against RSU impersonation and traceability attacks. However, the scheme remains vulnerable due to the impractical assumption of a fully secure TA-RSU communication channel and lacks solutions for V2X security and privacy. The IoVCom scheme by Limbasiya and Das [22] uses hash functions and ECC for security in V2X communications, offering lightweight and fast authentication. However, it requires substantial memory to store global and local parameters and faces scalability challenges. The use of identical keys across all RSUs makes the system vulnerable, as compromising one RSU could lead to a total system failure. The solutions proposed in [3], [23], [24], and [25] follow batch verification techniques for faster V2X communication, but these suffer from privacy issues, high communication complexity, and bandwidth overhead. Sikarwar and Das [26] improved upon these with a common authentication approach for all V2X communications using one-way hash and bilinear mapping, providing scalability and security. However, the scheme requires each system component to handle numerous public parameters, limiting scalability when components are added or removed. In Table 1, we compare the security and privacy functionalities provided by our proposed scheme and the existing schemes.

B. OUR CONTRIBUTIONS

With the goal of proposing a lightweight privacy-preserving authentication protocol for a secure V2X communication framework in IoV networks, our main research contributions are detailed below:

1) Our proposed scheme achieves fast and lightweight authentication by using single-time prime multiplication, division, and six general cryptographic hash operations. RSU computations are minimized to a single cryptographic hash per authentication, and the group key-sharing process eliminates RSU involvement, making the process faster and more reliable for time-critical IoV applications.
2) To enhance privacy and prevent traceability attacks, we implemented a pseudo-random identity assignment phase where the TA generates unlinkable random identities for vehicles, ensuring the privacy of original identity and location-based traceability is preserved. We also design a group key-sharing process from TA to the verified system entities to facilitate traffic broadcasts securely. The group key-sharing process leverages quadratic residues (qrs) and non-residues (nqrs), which can be pre-computed and reused, making it highly efficient.
3) For V2X communications, the scheme uses a common token-sharing approach with secret tokens of variable lengths independent of TA's direct involvement. Additionally, the token-sharing process requires only a single cryptographic hash and XOR operations, with just 40 bytes of data exchanged to establish direct V2X communication. This reduces the TA's computational and communication overhead while improving scalability.

III. SYSTEM MODEL AND PRELIMINARIES

In this section, all the necessary system components, the network model, the adversarial assumptions, and the methodologies valid and used in our proposed scheme are described in brief.

A. NETWORK MODEL

As IoV is a combined form of VANET architecture together with IoT applications, we follow the general network model, which is also valid for [25], [30]. The IoV network, therefore, consists of the following entities:

1) TRUSTED AUTHORITY (TA)

We assume that there is a trusted authority (TA) that is reliable and can not be compromised [2], [13]. TA is the central authority that oversees the overall security framework, authentication, key sharing, and malicious activities within the network. TA is presumed to have sufficiently large computational and storage resources to efficiently handle vehicle authentication, secure data management, communication logs, key management, etc. The TA operates over a backhaul network, a high-bandwidth, secure infrastructure connecting RSUs and other distributed entities to the central authority. This network defines TA's communication range and can span through large geographic regions using cellular technologies like 3GPP LTE (4G) and 5G [31], which offer low latency, high bandwidth, and reliability. These capabilities enable the

TA to maintain scalability and manage the overall security framework effectively in large-scale IoV deployments.

2) ROAD SIDE UNITS (RSUs)

The RSU components can be physically placed by the roadsides in specific regions and are mainly responsible for gathering and sharing local traffic and other information. RSUs act as intermediaries between the Trusted Authority (TA) and other entities, facilitating communication and extending the TA's effective range. RSUs follow short-range communication standards such as IEEE 802.11p (DSRC) [32], with a communication range of up to 1 km, and 3GPP Release 14 C-V2X, which provides ranges of 300-500 meters in urban areas and up to 2 km in open environments [5], [31]. These standards ensure reliable and secure interactions with vehicles and other IoV entities. RSUs are connected to the TA via the backhaul network using cellular technologies like LTE (4G) or 5G. Similar to the TA, RSUs must be equipped with sufficient computational and storage resources to manage real-time authentication of a large number of vehicles, validate pseudonyms, and maintain lists of vehicles operating within their regions. Alternative technologies like cloud and edge computing and satellite communication can also be adapted in RSUs to mitigate frequency challenges.

3) VEHICLES (V) AND SMART ENTITIES (X)

Apart from TA, RSU, and the vehicles (V), any other entity in the system is categorized as X and, therefore, referred to in V2X communication. Each vehicle in the system has an OBU chip that needs to be embedded into a tamper-protected area to ensure protection against physical attacks. All the other entities in X, such as infrastructure, smart parking areas, smart grid, etc., have a tamper-protected unit (TPU) safely placed in their system. The OBU and TPU chips are responsible for all security-related computation and communication with TA, RSU, and the other entities in X. Silimar to RSUs, vehicles also follow short-range communication standards [33] such as IEEE 802.11p (DSRC) or the 3GPP-C-V2X.

4) HIERARCHICAL MODEL

The proposed scheme adopts a hierarchical model to efficiently manage network operations and distribute tasks across the described entities as presented in Figure 2. The TA serves as the root of trust, managing critical administrative functions such as pseudonym issuance, cryptographic credential management, and system-wide security management. RSUs perform delegated localized responsibilities, including pseudonym verification, real-time authentication, and maintaining communication logs for V2X interactions. Vehicles and other smart entities primarily interact with RSUs for authentication and communication, which ensures that the TA's workload is minimized and real-time operations are efficiently handled at the local level.

To become a legitimate user, all the IoV components register with TA using a secure channel offline only once before deployment. After that, all the communications between these components and TA happen through public channels. The group key is temporary and only used once for a specific session of communication with a specific group. The group key is restricted to use only for broadcasts from TA/RSU. It can not be used directly for V2X and other communications. V2X communication is only allowed when there is a valid token between the sender and the receiver. The token is similar to a symmetric key that can be used for digital signatures and encryption of communicating messages. The token is temporary, and RSU keeps a record of the V2X communicating parties in a communication log.

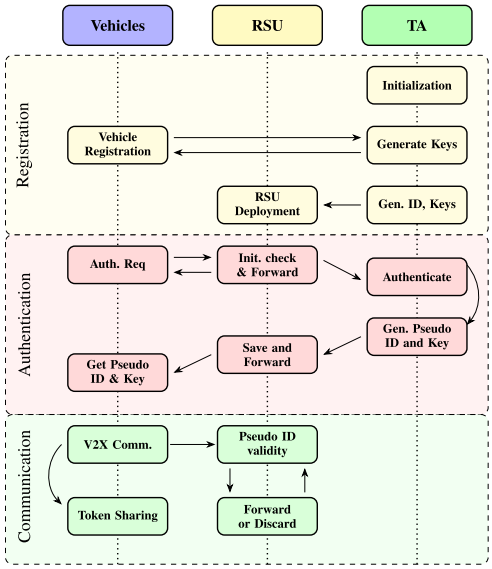

FIGURE 2. Control flow diagram illustrating the phases of the proposed protocol and interactions between Vehicles, RSUs, and the TA.

B. THREAT MODEL

The adversarial assumptions follow the same model as mentioned by [2], [6], and [7]. We assume that the adversary knows the communication protocol and can send and receive messages directly to all communication parties using the public channel. The adversary has full knowledge of the cryptographic approaches used in the scheme. An adversary in IoV can be both a valid legitimate user or an unauthorized user/eavesdropper performing both active and passive attacks. In passive attacks, the adversary tries to get original information that is private or sensitive, such as the vehicle's original identity, its traveling routes, current location, etc. In active attacks, the adversary not only attempts to access sensitive information but also alters or manipulates it, potentially causing planned actions that result in severe consequences, such as accidents or traffic disruptions. A non-credible but valid user in the system can act as an internal adversary to gather private information for other users by performing privacy theft attacks using the global or local

variables and their 'legitimate' status. It can also perform impersonation attacks and repudiation attacks by hiding its original identity and pretending to be some other user in the system. An unauthorized entity, also known as an external adversary, can intercept any communication message from the channel and investigate the message to find similarities and linkability to perform identity thefts, traceability attacks, reply attacks, and man-in-middle attacks (MiMA). The external adversary can try to guess the group key or the token by looking at the shared values to perform offline guessing attacks and even brute force attacks. We assumed that the adversary has huge computational power but not infinite power to compute hard computation problems, such as discrete logarithm problems, in a reasonable amount of time.

Once the network and the adversarial models are defined, we now present the cryptographic methodologies used in our proposed scheme in brief.

C. ONE-WAY-HASH-FUNCTION

A one-way hash function (H) is an irreversible cryptographic function that processes input data (x) of any size into fixed-size blocks (l) and produces a fixed-size output digest (y); $H(x) \rightarrow y$. It is designed to be computationally infeasible to (i) recover the original message (x) from its hash (y) or (ii) find two distinct messages (x_1, x_2) that produce the same hash (y). This can be expressed as $H : \{0,1\}^* \rightarrow \{0,1\}^l$. Common hash functions like SHA-1, SHA-256, SHA-384, and SHA-512 have been widely used in cryptography for decades, offering provable security, irreversibility, and no message digest collisions. The hash value (y) is often denoted as $h(x)$, a unique random string derived from the input message (x). In our proposed scheme, we use two general hash functions, H (e.g., SHA-256) and $H1$ (e.g., SHA-384), to generate hash values of different sizes.

D. QUADRATIC RESIDUES

Quadratic residuosity is considered a hard-cryptographic approach and used in modern cryptographic solutions, especially in private information retrieval (PIR) protocols, and can be used in privacy-preserving security approaches [34], [35]. Let m be a positive integer, then an integer a is a quadratic residue modulo m if $gcd(a, m) = 1$, i.e., they are co-primes and the congruence $x^2 \equiv a \ (mod\ m)$ has a solution. Computationally, it is an NP-hard problem to find if the equation $x^2 \equiv a \ (mod\ m)$ has a solution for any given value of a and m in polynomial time, where the factorization of m ($m = p * q$) is unknown and p, q are large odd primes. This is also referred to as the Quadratic residuosity problem [36]. To generate quadratic residue (a) modulo m, where the factorization $m = p * q$ is already known, we need to find x such that the following equations have solutions:

$x^2 \equiv a \ (mod\ p)$ and $x^2 \equiv a \ (mod\ q)$. For any a if $x^2 \not\equiv a (mod\ p)$, then we can say that a is quadratic non-residue of m.

TABLE 2. List of symbols used in proposed scheme.

Notation	Description
VID	Original identity of OBU/TPU user
p, q	Large Odd Primes
V_RID	Random nonce, $V_RID = rp \| \| sqn$
R_ID	Original identity of RSU
RP	Secret value of RSU
Tds	Time threshold between Hello-types
sqn	Random sequence number
$rv(i), rp$	Pseudo Random Values; $i = 1, 2..$
rp_{n-bits}	First $n - bits$ of rp
δTS	Maximum allowed time threshold
$H(), H1()$	One way hash functions
$h_{(i)}, h_{H1}$	Different hash values; $i = 1, 2, 3..$
$TS(i)$	Current time-stamp; $i = 1, 2, 3..$
$IV(i)$	Intermediate values; $i = 1, 2, 3..$
$h_5^{V_i}/h_5^{X_i}$	Temporary identity of Vehicle V_i or entity X_i
$OBU[i]$	On-Board-Unit of Vehicle i
$TPU[i]$	Tamper Protected Unit of Entity i
$qrs, nqrs$	Quadratic residues and non-residues
S.Info, R.Info	Sender and Receiver in V2X communication
T_k	Randomly generated token.
$\oplus$	XOR operation
$\|\|$	Concatenation operation

For a large odd prime p, there are exactly $(p-1)/2$ quadratic residues (qrs) modulo p and exactly $(p-1)/2$ quadratic non-residues ($nqrs$) modulo p [37].

E. LEGENDRE SYMBOL

Legendary symbol is a multiplicative function used to check if a given integer a is a quadratic residue (qr) or non-residue (nqr) modulo p, where p is an odd prime number. The Legendre symbol is denoted as $\left(\frac{a}{p}\right)$ defined by:

$$\left(\frac{a}{p}\right) = \begin{cases} 0 \ \ if & a \equiv 0 \ (mod\ p) \\ 1 \ \ if & a \equiv 1 \ (mod\ p), i.e.\ a\ is\ qr\ modulo\ p \\ -1 \ \ if & a \not\equiv 1 \ (mod\ p), i.e.\ a\ is\ nqr\ modulo\ p \end{cases}$$

By using Euler's criterion, we compute:

$$\left(\frac{a}{p}\right) = a^{(p-1)/2} \ (mod\ p).$$

For any received value of a, if this satisfies, then a is qr modulo p otherwise a is nqr modulo p.

IV. PROPOSED SCHEME

Once all the preliminaries and the methodologies are set, the actual proposed scheme is presented in this section. The scheme is divided into five phases, namely: 1) Registration phase, 2) Anonymous Authentication phase, 3) Pseudo Identity Assignment phase, 4) Group Key Sharing phase, and 5) V2X Token Sharing and Communication phase. The different notations used in the proposed scheme are listed in Table 2.

A. REGISTRATION PHASE

To become a legitimate user and a valid component in the IoV network, each vehicle, all the RSUs, and the entities X in V2X must be registered with TA before they are publicly

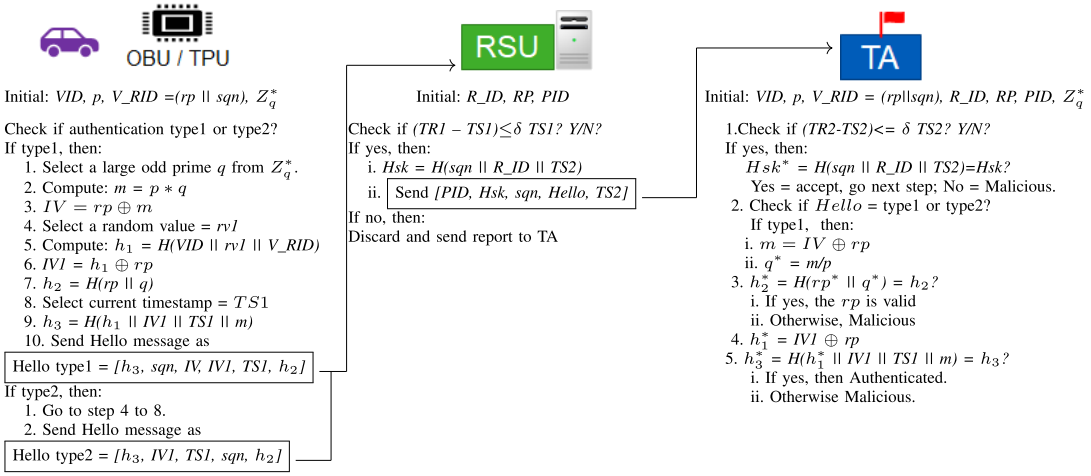

FIGURE 3. Authentication phase.

deployed. The TA initially performs the registration process only once, with the vehicles equipped with OBUs and the entities equipped with TPUs. All the value sharing during the registration process happens through a secure communication channel offline. The description of the phase is given as follows.

1) VEHICLE REGISTRATION

1) The TA first selects a large prime number (P) and a finite field Z_q^*.
2) The TA selects a unique identity VID, which serves as the vehicle's original identity and consists of privacy-related information about the owner, such as name, address, contact info., issuing date, license, etc.
3) TA also generates a random nonce named V_RID for the vehicle, which has two separate parts, a pseudo-random value part (rp) and a random sequence number (sqn) part. $V_RID = rp||sqn$.
4) Then, TA shares the parameters VID, p, V_RID, Z_q^* with the OBU chip using a secure channel and then attaches the OBU chip in a tamper-protected area in the vehicle.
5) TA saves these parameters in its own database by maintaining a table of records.

2) TPU AND ENTITY (X) REGISTRATION

OBU and TPU chips are basically the same and responsible for security-related communications. The registration process of a TPU is the same as OBU; only the VID consists of information specific to the entity type, which could be a smart home, smart parking, smart grid, wireless sensors, etc.

3) RSU REGISTRATION

1) To initially register an RSU, the TA first selects a unique identity R_ID. Then, it generates a pseudo-random value RP as a shared secret value between the TA itself and the RSU.
2) TA assigns unique PID to the RSU, which will be public after the RSU deployment.
3) All these values are shared with a TPU chip through a secure communication channel and embedded into a tamper-protected area of RSU. The values are stored in TA's database, which is securely maintained as a record table.

B. ANONYMOUS AUTHENTICATION PHASE

All the registered vehicles and other entities in IoV are obligated to get authenticated by the TA before they can communicate with other units to send or receive messages. The proposed privacy-preserving anonymous authentication technique is the same for all vehicles (OBU) and other entities 'X' (TPU). Therefore, we only present it for a vehicle (OBU), as shown in Figure 3. This phase is divided into two parts, namely 1) the authentication message generation phase and 2) the authentication message verification phase.

1) AUTHENTICATION MESSAGE GENERATION PHASE

Whenever a vehicle enters an RSU region, it sends an authentication message or 'Hello' message to the TA via that RSU. There are two types of 'Hello' messages in our scheme: Hello-type1 and Hello-type2. The vehicle sends a Hello-type1 message if the authentication is initial or the time difference between two Hello-type1 messages (δTS) is greater than or equal to a pre-specified time threshold (Tds), i.e., if $\delta TS \geq Tds$, otherwise it sends a Hello-type2 message. Note that Hello-type1 and Hello-type2 are both authentication messages generated by the OBU based on the time threshold Tds. The threshold is determined based on the number of vehicle users, the application area, and priority-based information.

1) Following Figure 3, the process of generating a Hello-type1 authentication message begins as the OBU selects a large prime number q from the field Z_q^*.
2) Then it computes $m = p * q$, and $IV = rp \oplus m$, where p is the preshared secret prime number. The rest

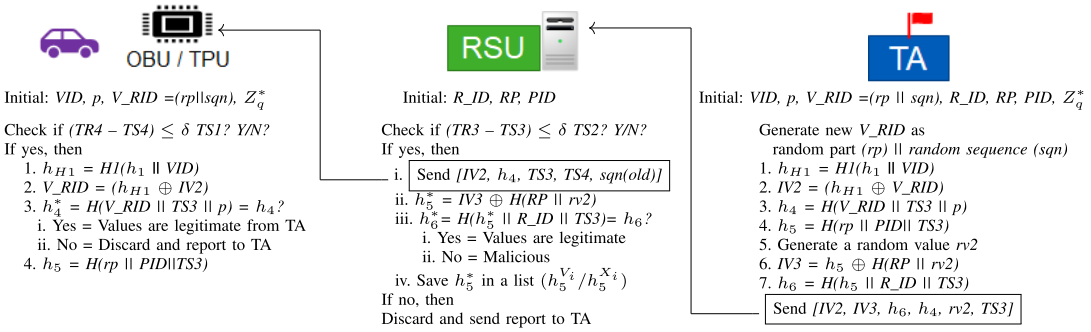

FIGURE 4. Pseudo ID assignment phase.

of the process is the same for both Hello-type1 and Hello-type2.

3) The OBU then generates a random value as $rv1$ and compute the hash $h_1 = H(VID||rv1||V_RID)$. It also computes an intermediate value $IV1 = h1 \oplus rp$ and another hash value $h_2 = H(rp||q)$.
4) After that, the OBU selects current timestamp $TS1$ and take the hash value $h_3 = H(h_1||IV1||TS1||m)$ and create the Hello-type1 $= [h_3, IV1, TS1, m, sqn, h_2]$ or it creates Hello-type2 message as Hello-type2 $= [h_3, IV1, TS1, sqn, h_2]$
5) OBU of the vehicle sends the 'Hello' message to the RSU using the RSUs public identity PID through a public channel.
6) Assume that the message is received by the RSU at time $TR1$, then RSU checks if $(TR1 - TS1) \leq \delta TS1$ holds, where $\delta TS1$ is the maximum allowed time threshold between vehicle and the RSU. If the condition holds, RSU accepts the message as valid and proceeds to the next step. Otherwise, it rejects the message and considers it invalid.
7) For a valid 'Hello' message, RSU generates a hand-shake as $Hsk = H(sqn||R_ID||TS2)$ and sends the message $M : [PID, Hsk, Hello, TS2]$ to the TA where $TS2$ is the current timestamp generated by the RSU.

2) AUTHENTICATION MESSAGE VERIFICATION PHASE

1) Assume that TA received the message M at time $TR2$. TA accepts M as valid for the next step if $(TR2-TS2) \leq \delta TS2$, where $\delta TS2$ is the maximum threshold value between RSU and TA.
2) For a valid M, TA first computes $Hsk^* = H(sqn||RID||TS2)$ where $sqn, TS2$ are taken from M and RID is a pre-shared secret value. Then, it checks if $Hsk^* = Hsk$ or not. If yes, it goes to the next step, otherwise TA now considers that M has come from a malicious RSU or an adversary and starts investigating.
3) From M, TA takes the Hello message and checks if it is a type1 or type2 message. For a type1 message, TA first gets $m = IV \oplus rp$ and selects corresponding p from its database, and computes $q^* = m/p$. For a Hello-type2, this step is simply skipped.
4) In the next step, TA first verifies if the sequence number (sqn) is valid or not. To do so, TA goes to the database table and takes a copy of the rp value corresponding to that sequence number. Then it computes $h_2^* = H(rp^*||q^*)$ and check if $h_2^* = h_2$ or not. If yes, it confirms that the sequence number is valid; otherwise, it is malicious.
5) If the sequence number is valid, then TA needs to check if it came from the original vehicle or not by first computing $h_1^* = IV1 \oplus rp$ and then $h_3^* = H(h_1^*||IV1||TS1||m)$ and check if $h_3^* = h_3$ or not. If it is a match, then authentication is successful, and TA will proceed to the next phase. Otherwise, authentication fails.

C. PSEUDO ID ASSIGNMENT PHASE

After each successful authentication, the TA assigns a new value for V_RID by generating and sharing it with the vehicle safely. This new value of V_RID is used in the next authentication process in a new RSU region. At the same time, a temporary identity for that vehicle is safely shared with the RSU, which is used in token sharing in V2X communication later in phase five. All this communication happens using the same public channel as presented in Figure 4 and described below.

1) COMPUTATIONS AT TA

1) The TA first generates a pseudo-random value for the rp and a random sequence number for the sqn to generate new V_RID.
2) Then TA computes $h_{H1} = H1(VID \;||\; h_1)$ where, $H1$ is the second hash function selected and h_1 is the hash value taken from authentication phase. Then it computes an intermediate value $IV2 = (h_{H1} \oplus V_RID)$ where V_RID is newly generated. It also computes a hash $h_4 = H(V_RID||TS3||p)$ where TS3 is the current timestamp generated by TA.
3) TA computes $h_5 = H(rp||PID||TS3)$ which serves as the temporary identity. It also generates a random value $rv2$ and compute intermediate value $IV3$ as $IV3 = h_5 \oplus H(rv2 \;||\; RP)$ and also $h_6 = H(h_5||R_ID||TS3)$.
4) TA sends $[IV_2, IV_3, h_6, h_4, rv2, TS3]$ in a message to the RSU using the public channel.

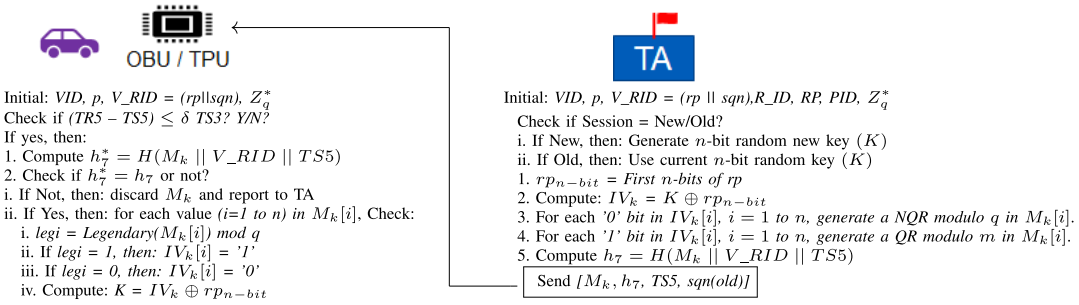

FIGURE 5. Group key sharing phase.

2) COMPUTATIONS AT RSU

1) Assume that RSU received the message at time TR3, it checks if $(TR3 - TS3) \leq \delta TS2$; if it holds, it accepts the message as valid; otherwise, discard the message.
2) RSU generates the current timestamp TS4 and sends $[IV2, h_4, TS3, TS4, sqn]$ to the vehicle. Note, here, the sqn is the same as RSU received it in the last Hello message from that vehicle.
3) RSU computes back the temporary identity as $h_5^* = IV3 \oplus H(rv2||RP)$. To confirm the legitimacy of the value and to ensure that the value came from TA, it computes $h_6 = H(h_5^*||R_ID||TS3)$ and check if $h_6^* = h_6$ or not? If it matches, then the values and the TA are legitimate, and RSU saves h_5^* in a list $(h_5^{V_i}/h_5^{X_i})$; otherwise, it discards the values and reports to TA.

3) COMPUTATIONS AT VEHICLE

1) Assume that vehicle received the message at time TR4, it checks if $(TR4 - TS4) \leq \delta TS1$. If it holds, then the vehicle accepts the message as valid. It can precompute $h_{H1}^* = H1(VID||h_1)$ and computes back V_RID as $V_RID^* = IV2 \oplus h_{H1}^*$ instantly.
2) To check the legitimacy of the value, the vehicle computes $h_4^* = H(V_RID^*||TS3||p)$ and checks if $h_4^* = h_4$. If yes, then the value is accepted and stored, otherwise, it reports back to TA.
3) The vehicle can compute the temporary identity as $h_5 = H(rp||PID||TS3)$ and save it to use in phase five.

D. GROUP KEY SHARING WITH VEHICLES

In this phase, a common key is generated and shared with a group of vehicles in an RSU region by the TA, as presented in Figure 5. After successful authentication, the process of sharing the common key with a vehicle begins as follows.

1) TA first checks if the vehicle is joining an existing session or entering a new session of broadcasts and information sharing. For a new session, the TA generates a $n - bit$ random vector (K) as the common key K. It uses the current existing $n - bit$ key K for an existing session.
2) TA takes the first $n-bits$ of current rp value as rp_{n-bits} and compute $IV_k = K \oplus rp_{n-bits}$.
3) For each $'0'$-bit in IV_k, TA generates a non-quadratic residue (nqr) modulo q and for each $'1'$-bit in IV_k, TA generates a quadratic residue qr modulo m. Let M_k be the set of n values generated for the value IV_k.
4) TA now computes a hash value h_7 and send it to the vehicle in a message containing $[M_k, h_7, TS5, sqn(old)]$ where $TS5$ is the current timestamp generated by the TA.
5) Assuming that the message is received by the vehicle at time $TR5$, it checks if $(TR5 - TS5) \leq \delta TS3$, where $\delta TR3$ is the maximum allowed delay between a vehicle and the TA.
6) The vehicle computes $h_7^* = H(M_k||V_RID||TS5)$. Then, it checks if $h_7^* = h_7$. If it matches, then the vehicle accepts the message and goes to the next step; otherwise, it discards the message and reports to the TA.
7) The vehicle unfolds the message, and for each received value in M_k, it checks if it is a qr or nqr modulo q by using the Legendre symbol. For each qr value, it assigns a binary $'1'$ in that position, and for each nqr, it assigns a binary $'0'$ in that position. Finally, it gets back the vector for IV_k as IV_k^*.
8) The vehicle takes the first $n-bits$ of current rp value as rp_{n-bits}^* and computes back the key $K = IV_K \oplus rp_{n-bits}^*$ for receiving the broadcast messages from RSU or the TA.

We assume that the key K is shared with the RSU from TA using some public-private key encryption technique.

E. V2X TOKEN SHARING AND COMMUNICATION PHASE

This phase allows vehicles to communicate with each other and other entities in IoV by following a common approach of sharing a token using the RSU only once. Regardless of the type of X, this phase allows vehicles to establish safe and direct communication with any other entity. After pseudo ID assignment, each vehicle V_i or entity X_i is assigned with a temporary identity $h_5^{V_i}$ or $/h_5^{X_i}$, and it is also shared with the corresponding RSU in that region. The RSU then makes a $list - X$ by assigning random names corresponding to each pseudo ID.

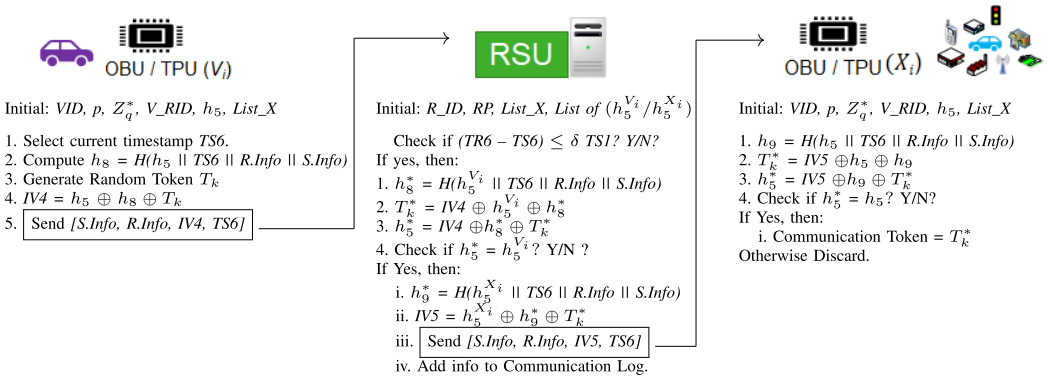

FIGURE 6. V2X token sharing phase.

Once the $list - X$ is created, RSU shares this list as a broadcast to all the entities in its range. If any vehicle V_i wants to communicate with any entity X_i, it initiates the token-sharing process as follows:

1) As presented in Figure 6, the vehicle V_i (sender) selects the current timestamp $TS5$ and compute $h_8 = H(h_5||TS5||R.Info||S.Info)$ where $R.Info$ is the receivers name and $S.Info$ is it's own name from $list - X$.
2) Then it generates a pseudo-random value as a token T_k and computes $IV1 = h_5 \oplus h_8 \oplus T_k$. The vehicle then sends $[S.Info, R.Info, IV1, TS5]$ to the RSU in a message.
3) The RSU first checks the timestamp as $(TR6 - TS6) \leq \delta TS1$? If yes, then using $S.Info$ RSU selects $h_5^{V_i}$ of the sender from the list and computes $h_8^* = H(h_5^{V_i}||TS6||R.Info||S.Info)$ and $T_k^* = IV1 \oplus h_5^{V_i} \oplus h_8^*$.
4) RSU also computes $h_5^* = IV1 \oplus h_8^* \oplus T_k^*$ and checks if $h_5^* = h_5^{V_i}$. If they match, then it confirms that the sender and the shared values are legitimate.
5) Now, from the $R.Info$, RSU selects $h_5^{X_i}$ of the receiver from the list and compute $h_9^* = H(h_5^{X_i}||TS6||R.Info||S.Info)$ and $IV2 = h_5^{X_i} \oplus h_9^* \oplus T_k^*$.
6) RSU then send $[S.Info, R.Info, IV2, TS6]$ to the receiving entity X_i. At the same time, RSU maintains a V2X communication log, where it puts the timestamp $TS6$, the sender and receiver's temporary identities, and the token T_k. This communication log is shared with the TA periodically and in a secure way.
7) Once received from the RSU, the receiver X_i now computes $h_9 = H(h_5||TS6||R.Info||S.Info)$ and generates back the token $T_k^* = IV2 \oplus h_5 \oplus h_9$.
8) Then the receiver X_i computes $h_5^* = IV2 \oplus h_9 \oplus T_k^*$ and check if h_5^* match with its own h_5. If it is a match, it confirms that the message indeed came from RSU and accepts the token T_k.

Once the token is shared between the sender (V_i) and the receiver (X_i) via the RSU, direct V2X communication is established. The token can be used as a symmetric key to encrypt or decrypt messages or to digitally sign a message based on the security and privacy needs and the type of communication.

V. SECURITY VALIDATION AND CORRECTNESS

After presenting the proposed scheme, this section discusses its security validity and correctness. We first verify the correctness and validity of each phase before analyzing potential security attacks. The security at the end of each phase is assessed against its beginning, ensuring each phase meets its intended goals. We then present claims and proofs to support the scheme's unconditional security and correctness.

Claim 1: The registration phase is fully secured from adversarial interference, and the values are shared/stored correctly.

Proof: During the registration phase, all communications between the TA and the OBU/TPU occur through a secure offline channel. The TA generates the values VID, V_RID, p, and Z_q^* and shares them securely with the OBU/TPU. The OBU/TPU is then placed in a tamper-protected area within the vehicle or another entity, safeguarding these values from physical attacks. The secure channel and offline mode prevent any adversarial interference, ensuring that the parameters VID, V_RID, p, and Z_q^* for a user (V_i/X_i), as well as RP and RID for an RSU, remain safe and secret during registration. The TA stores these values in a secret database accessible only to itself. As a fully trusted entity in IoV, the TA ensures that these secret parameters are never leaked.

Claim 2: All the values, shared in an authentication message through the public channel, give no information about the shared secret values of VID, rp, p, Z_q^*, R_ID and also h_1.

Proof: A Hello-type1 authentication message contains $[h_3, IV1, TS1, IV, sqn, h_2]$, and Hello-type2 contains $[h_3, IV1, TS1, sqn, h_2]$. Since h_3 and h_2 are irreversible hash values, they reveal nothing about the original values of h_1, rp, and q. The $IV1$ value is computed as $IV1 = h_1 \oplus rp$, where rp is a secret derived from V_RID, and h_1 is calculated as $h_1 = H(\text{VID} || rv1 || V_RID)$ by the OBU/TPU. The value h_1 is not directly shared but XOR-ed with rp, making $IV1$ a random value that doesn't reveal h_1 or rp, the same is applied for IV. An adversary cannot compute h_3 without knowing h_1,

which is secret and unique for each authentication due to the random $rv1$.

The sqn is shared by the OBU/TPU in the authentication message. The secret parameter V_RID consists of rp and sqn, and sharing sqn does not reveal the current value of rp, as both are randomly generated and assigned by the TA during registration. The RSU uses sqn from the authentication message to compute the handshake $hsk = H(sqn \,||\, R_ID \,||\, TS2)$, where R_ID is a shared secret between the TA and RSU from the registration phase. Since hsk is an irreversible hash value and only the TA knows R_ID, it can verify the legitimacy of the RSU. Thus, after verification, all values for VID, rp, p, Z_q^*, R_ID, and h_1 remain secure as expected.

Claim 3: After each successful verification, a new value of sqn and rp (V_RID) is shared with the OBU/TPU secretly.

Proof: When a user is successfully verified, the TA performs a pseudo ID assignment phase and shares a new V_RID with the OBU/TPU. This new V_RID consists of two parts, rp and sqn, with TA generating random values for each. These values ($rp||sqn$) are shared as the new V_RID in a way that prevents the RSU or an adversary from deducing sqn or rp during transmission. The TA computes a second hash function $H1$ to produce $h_{H1} = H1(VID||h_1)$, a random value matching the size of V_RID. Since h_1 is unique and secret for each authentication (as shown in claim 2), it is used to compute h_{H1}. The new V_RID is then shared with the OBU/TPU as $IV2 = h_{H1} \oplus V_RID$. The OBU/TPU can easily compute $h_{H1}^* = H1(VID||h_1)$, retrieve V_RID^* as $IV2 \oplus h_{H1}^* = V_RID^*$, and update sqn and rp for the next authentication. Since h_1 and VID are known only to the OBU/TPU and TA, the V_RID from $IV2$ can only be retrieved by the corresponding OBU/TPU. This ensures that rp and sqn are safely shared with the OBU/TPU at the end of the pseudo ID assignment phase.

Claim 4: Sending sqn openly to TA in an authentication message does not cause linkability, and the value of rp is still secret.

Proof: During registration, TA assigns a secret V_RID (containing rp and sqn) to the OBU/TPU (claim 1). While sqn is shared openly during authentication, it reveals nothing about rp (claims 1 and 2). After each authentication, a new V_RID with random rp and sqn is securely shared with the OBU/TPU (claim 3). Since sqn is randomly assigned and secret, it cannot be linked between different authentication sessions. Even if an adversary intercepts $IV2$ and sqn at different times, they cannot derive any meaningful information about rp. Thus, exposing sqn during authentication does not compromise rp or any other secret parameters.

Claim 5: The pseudorandom identity generated by h_5 gives no information about the secret parameters VID, p, V_RID, Z_q^*, RP to any other entity, and h_5 is only known by TA, corresponding RSU, and the OBU/TPU.

Proof: During the pseudo ID assignment phase, TA generates a pseudo ID h_5 for a vehicle V_i or entity X_i. The value of rp is secret and known only to V_i, X_i, and TA (claims 2, 3, 4). The ID h_5 is independent of secret parameters like VID, p, and Z_q^*. To securely transmit h_5 to the RSU, TA computes $IV3 = h_5 \oplus H(rv2||RP)$, where RP is the RSU's secret and $rv2$ is random, making $IV3$ reveal nothing about h_5 or other secrets. The RSU can retrieve h_5 using $IV3 \oplus H(p||rv2)$. The OBU/TPU computes $h_5 = H(rp||PID||TS3)$ using the secret rp. Thus, h_5 remains known only to the OBU/TPU, RSU, and TA, without revealing any secret parameters. Therefore, all shared parameters VID, p, V_RID, Z_q^*, RP, and RID remain secure at the end of the pseudo ID assignment phase.

TABLE 3. Security comparison for existing state of the art schemes with proposed scheme.

Attacks	[28]	[22]	[18]	[26]	[27]	[29]	Proposed
A1	Yes	No	Yes	Yes	Yes	Yes	Yes
A2	No	No	No	Yes	No	No	Yes
A3	Yes	Yes	Yes	Yes	Yes	Yes	Yes
A4	Yes	Yes	Yes	Yes	Yes	Yes	Yes
A5	Yes	Yes	Yes	Yes	Yes	Yes	Yes
A6	Yes	Yes	Yes	No	Yes	Yes	Yes
A7	No	No	No	Yes	No	No	Yes
A8	No	Yes	No	Yes	No	No	Yes
A9	NA	Yes	NA	No	NA	No	Yes

Note: For different attack types, some abbreviations are used as A1: User Privacy Theft Attack, A2: Traceability Attack; A3: Impersonation Attack; A4: Reply Attack; A5: MiMA; A6: Modification Attack; A7: Non-Repudiation; A8: Sybil Attack, A9: Collision Induction; Yes: Secure; No: Vulnerable; NA: Not Applicable.

Claim 6: The group key sharing message containing M_k provides no information about the common key K or the secret parameters to unauthorized users or adversaries.

Proof: For an n-bit group key K, TA generates a sequence of n values as a mix of qrs and $nqrs$ modulo q. As the factorization ($m = p * q$) is known only to the OBU/TPU and TA since p and q are secret parameters from the field Z_q^*. Only TA can derive q by dividing m by P. An adversary would need to know P and q to determine if a value in M_k is qr or nqr. Since P is secret (claims 1, 2, 3, 5), only the intended OBU/TPU can verify M_k and retrieve IV_k. TA computes $h_7 = H(M_k||V_RID||TS5)$ for secure transmission, where V_RID is secret (claim 3). The OBU/TPU verifies M_k by computing $h_7^* = H(M_k||V_RID||TS5)$. The hash h_7 reveals nothing about V_RID or other parameters during transmission. Since VID and Z_q^* are not used in key sharing, the message with M_k, h_7, and $TS5$ does not disclose any information about the key, VID, P, V_RID, Z_q^*, or q.

Claim 7: The token sharing phase is completely dependent on the pseudo-identity h_5 and independent of using any of the secret parameters like VID, p, V_RID, Z_q^*.

Proof: The token-sharing process begins when the RSU sends the $list - X$ with temporary names to all OBU/TPU units within its range. Each OBU/TPU knows only its pseudo identity h_5 (claim 5) and the names in $list - X$. The OBU/TPU computes $h_8 = H(h_5||TS5||R.Info||S.Info)$ and sends $IV4 = h_5 \oplus h_8 \oplus T_k$ to the RSU. Since h_5 is secret, $IV4$ is random and does not reveal the token T_k. The RSU verifies the sender by recomputing h_5^* and checking it against its stored value. If valid, the RSU shares the token with the intended receiver X_i using $R.Info$ and $h_5^{X_i}$ from $list - X$. The RSU computes

$h_9^* = H(h_5^{X_i}||TS6||R.Info||S.Info)$ and $IV5 = h_5^{X_i} \oplus h_9^* \oplus T_k^*$. As h_9^* is irreversible and $IV5$ is random, T_k remains secure. The pseudo-identity h_5 is known only to the corresponding OBU/TPU, RSU, and TA (claim 5), ensuring that all secret parameters VID, p, V_RID, Z_q^* are never shared. This process ensures the security of all secret parameters throughout token sharing, supporting the claim.

VI. PRIVACY AND SECURITY ANALYSIS

In this section, we first analyze the privacy guarantees provided by the proposed scheme. The scheme is designed to prioritize privacy at every communication step, ensuring that sensitive user and vehicle information remains protected against unauthorized access and potential tracing or linkage by adversaries. We then evaluate the security strength of the proposed scheme against the designed threat model, considering both active and passive attacks. Potential attack surfaces in each phase of the scheme are identified and analyzed, with a comparison of the defense capabilities against known security and privacy attacks in IoV, as shown in Table 3. This comprehensive analysis demonstrates the scheme's robustness against a wide range of adversarial strategies.

A. PRIVACY GUARANTEES

1) IDENTITY PRIVACY

The scheme ensures that an adversary A cannot access sensitive user information such as the vehicle's original ID (VID), owner's name, or address. During authentication, irreversible hash values (h_2, h_3) and pseudo-random values ($IV1, sqn$) are used in Hello-type1 and Hello-type2 messages, which reveal nothing about the VID or pseudo ID (V_RID). Additionally, the hash value (h_1), derived from VID, $rv1$, and V_RID, is not shared directly, preventing adversaries from inferring original identities.

2) LOCATION AND TRACEABILITY

To prevent adversaries from tracking a vehicle's routes or movements, the scheme dynamically generates random sequence numbers (sqn) and pseudonyms for every session. Messages transmitted at different times (M_{t1} at $t1$ and M_{t2} at $t2$) use distinct and unlinkable parameters. This ensures that no adversary can trace user movements or location-based information across sessions.

3) UNLINKABILITY

Pseudonyms used in the scheme are unlinkable due to the dynamic pseudo ID assignment phase. The Trusted Authority (TA) generates a unique pseudo ID (h_5) for each session, shared only with the RSU and the vehicle. During V2X token sharing, a random name is assigned to h_5 for each sender or receiver. These measures ensure that messages and pseudonyms from different sessions cannot be linked to the same vehicle or user.

4) MINIMAL EXPOSURE OF SENSITIVE DATA

Sensitive information such as the vehicle's original identity or travel routes is neither transmitted in plaintext nor stored in intermediate systems like RSUs. Anonymized identifiers and encrypted tokens are used for all V2X communications. Even when broadcasting traffic updates or group keys, the scheme ensures that no data shared includes traceable components.

B. SECURITY ATTACKS ANALYSIS

1) IMPERSONATION ATTACK

An adversary (A) must not be able to impersonate a legitimate user, RSU, or TA to gain unauthorized access to the network. To do so, A would need knowledge of the finite field Z_q^*, the original ID (VID), the shared secret value (p), the current sequence number (sqn), and the pseudo ID (V_RID). Since the OBU/TPU is in a tamper-protected area and values are securely shared during registration, A cannot physically obtain these secrets. Even if A captures a message (M) and tries to guess the secret parameters, it will fail because h_3 includes a hash value h_1 with a random $rv1$ and current V_RID (claim 1). If A tries to guess sqn, TA will detect it when verifying $h_2^* = h_2$ (claim 4). Even if A guesses sqn, it will fail when TA checks $h_3^* = h_3$, which relies on secret parameters. To impersonate a valid RSU, A would need the secret parameters R_ID and RP, stored in a tamper-protected area, making them inaccessible.

During the pseudo ID assignment phase, A might try to deceive users by posing as the TA. However, users verify $h_4^* = h_4$, and RSU checks $h_6^* = h_6$ using secret parameters and current timestamps (claims 2, 3, 5), ensuring TA legitimacy. Similarly, while sharing the group key message M_k, $h_7^* = h_7$ is checked to verify the TA. As claim 3 confirms, only the legitimate user and TA know the current V_RID, ensuring the TA's legitimacy. Thus, our scheme fully protects against impersonation attacks by A.

2) REPLY ATTACK

In a reply attack, an adversary A captures a transmitted message M and attempts to reuse it to gain unauthorized access. Our scheme counters this by including timestamps in every message exchanged between the user, RSU, and TA. These timestamps are integrated into the hash values, preventing A from altering them during transmission. Each received message's timestamp is checked against a predefined threshold time δTS. If A resends a captured message with a new timestamp, TA will detect it, as the original timestamp is part of the hash h_3 used in authentication, which A cannot replicate without knowing all secret values ($VID, V_RID, p, rv1, rp$). This applies to all messages in our scheme, where each M contains randomly generated, non-guessable values, ensuring $M_{t1} \neq M_{t2}$ and preventing reply attacks.

3) MIMA (MAN IN THE MIDDLE ATTACK)

In a MiMA, an adversary A intercepts messages to perform cryptanalysis and extract information. Our scheme is protected from MiMA attacks through the use of irreversible

hash values and timestamps. Since the hash is irreversible, analyzing M reveals nothing about the secret values VID, p, Z_q^*, or the current V_RID (claim 2). Additionally, A cannot predict the next sequence number as they are randomly assigned and known only to the TA (claims 3 and 4). To obtain the group key from M_k, A would need to factorize m into its secret primes p and q, which is infeasible. Moreover, q changes over time, preventing A from linking M_k at time $t1$ with M_k at time $t2$ (claim 6). The threshold value δts further ensures that linking $M_k(t1)$ and $M_k(t2)$ is impossible.

4) MODIFICATION ATTACK

A modification attack occurs when an adversary A intercepts and illegally alters a message M from the public channel. Changing the hash values in M is impossible without knowing all secret parameters. Modifying the sequence number sqn is detected during verification, as TA computes $h_2^* = H(rp||q)$ to match the received h_2, which A cannot compute without knowing rp and q. If M_k is modified, the OBU detects it immediately by checking $h_7^* = h_7$, as h_7 is based on the pseudo-identity V_RID and the current timestamp. Any modification attempt also requires time for alteration and retransmission, which will be detected by the receiver when the timestamp is checked against the maximum allowed threshold, rendering modification attacks ineffective.

5) NON-REPUDIATION

In IoV networks, non-repudiation is a significant challenge, particularly in V2X communications where a valid user U_m can act as an adversary, sending false information and performing illegal activities. Our scheme addresses this through a secure token-sharing phase via RSU. Once the token (T_k) is shared, it can be used for digital signatures or as an encryption-decryption key, ensuring message security and legitimacy. The token is known only to the sender and receiver (claim 7), and messages are accepted only if signatures match or decryption is successful, preventing both sender and receiver from denying their actions. The RSU records V2X communications and updates the TA in case of disputes. The TA can then investigate and identify the original sender and receiver, as pseudo-identities (h_5) are linked to original identities in the TA's database.

6) SYBIL ATTACK

In a Sybil attack, A aims to overload system resources by sending false messages. Knowing the communication protocol, A can generate fake authentication messages with random sequence numbers (sqn). However, in our scheme, the TA quickly detects and eliminates incorrect sqn values by checking the hash without needing to complete the entire authentication process. Similarly, if A sends false messages M_k^* for group-key derivation, the vehicle checks the TA's legitimacy through the hash h_7 and discards fake messages before processing.

During V2X token sharing, A might send false messages with fake sender names to overload the RSU. However, the RSU uses the temporary identity (h_5) for each user and verifies the sender's legitimacy using a hash (h_8^*) and XOR operations. Fake messages are discarded quickly, preventing A from significantly impacting system resources. This fast verification process is a key strength of our scheme.

7) COLLISION INDUCTION

Collisions in automated vehicles can occur if they fail to quickly communicate emergency alerts to surrounding vehicles. Fast, direct communication is essential to minimize RSU or TA involvement and prevent adversaries from sending fake emergency alerts that could cause collisions. Our scheme ensures the secure and rapid sharing of emergency alerts, with a token-sharing time of 0.0044 milliseconds (based on a hash time of 0.0011 milliseconds, as shown in Table 5. This quick token-sharing process allows vehicles to securely send alerts and avoid collisions, making our scheme effective against collision induction.

8) BRUTE-FORCE ATTACK

In a brute-force attack, an adversary attempts to guess values and parameters through repeated trials. This approach is infeasible for large values, as solving such problems in polynomial time is computationally impractical. In our scheme, even if A tries to guess VID, p, or Z_q^*, the large size of these values makes it infeasible. Even if these were somehow guessed, the frequently updated V_RID remains out of reach within a short time frame. Additionally, brute-forcing a 128-bit group key would require 2^{127} attempts, far beyond current technological capabilities. Thus, our scheme is secure against brute-force attacks.

C. FORMAL SECURITY ANALYSIS

We also adapt the random oracle model based formal security analysis as mentioned in [15] to formally prove our scheme's security strength. All the entities in the scheme act as polynomial-time Turing machines, and a challenger $\mathcal{C}$ simulates the operations of the proposed scheme. The adversary $\mathcal{A}$ interacts with the challenger $\mathcal{C}$ with the following queries:

- Hash (q_h): The hash query q_h represents the adversary's ability to make queries to the hash function and determine the likelihood of hash collisions.
- Send (q_s): The send query q_s represents interactions initiated by the adversary by manipulating instances Π_P^t where $\mathcal{A}$ sends messages to $\mathcal{C}$ and observes the responses. This term is significant in the protocol analysis, particularly in how the adversary might influence or manipulate messages to break the scheme.
- Execute (q_e): The executed query q_e allows the adversary to observe the execution of the protocol instances Π_P^t between honest parties passively. This helps the adversary to gather information that might be used to compromise the protocol.

- Test(Π_V^t, Π_{LI}^t): By utilizing the test query, $\mathcal{A}$ tries to differentiate between random secrets and the group-specific secret key. This query is allowed only once, and upon receiving the test query, $\mathcal{C}$ makes a decision by tossing an unbiased coin $c : \{0, 1\}$. If $c = 1$, then $\mathcal{C}$ replies to $\mathcal{A}$ with the epoch key. Otherwise, $\mathcal{C}$ chooses a random value 0, 1 and returns it as output to $\mathcal{A}$.

Theorem 1: Considering $\text{Adv}_{\mathcal{A}}$ as the probability advantage for adversary $\mathcal{A}$, and relying on secure hash functions and quadratic residuosity assumptions, the security bound integrates query-based advantages for $\mathcal{A}$ into the following comprehensive bound:

$$\text{Adv}_{\mathcal{A}} \leq \frac{q_h^2}{2^l} + \frac{(q_s + q_e)^2}{p} + \frac{q_s}{2^{l-1}} + 2 \cdot \text{Adv}_A^{\text{QRA}} \quad (1)$$

- $\frac{q_h^2}{2^l}$: This term represents the collision probability for the hash function queries, where q_h is the number of hash queries and 2^l is the output length of the hash function.
- $\frac{(q_s+q_e)^2}{p}$: This term captures the combined effect of the send queries (q_s) and execute queries (q_e) related to the order p.
- $\frac{q_s}{2^{l-1}}$: This term accounts for the probability of finding collisions or pre-images in hash functions related to send queries.
- $\text{Adv}_A^{\text{QRA}}$: This term represents the adversary's advantage in breaking the quadratic residuosity assumption (QRA).

To prove Theorem 1, consider the following game sequence G_i, $i \in (1, 4)$ with some interconnected events.

- **Game G_0:** Game0 simulates a real-world attack scenario under the Random Oracle Model. In this game, the adversary attempts to break the protocol's security by interacting with $\mathcal{C}$ in a manner consistent with a genuine attack. Let $\Pr[Succ_{\mathsf{G}_0}]$ denote the probability of success in G_0, which represents the likelihood that the adversary achieves its objective within the constraints of the game. The adversary's advantage, denoted as $Adv(A)$, is computed as:

$$Adv(A) = |2 \cdot \Pr[Succ_{G0}] - 1| \quad (2)$$

  This advantage quantifies the deviation of the adversary's performance from random guessing, providing insight into the effectiveness of the adversary's attacks against the protocol.
- **Game G_1:** $\mathcal{A}$ generates an authentication message that passes the verification process vehicle receives a pseudonym. As the V2I authentication messages are signed and encrypted, the success probability of G_1 is similar to G_0. The probability that $\mathsf{G}_1 \Rightarrow win_{\mathcal{A}}$ can be determined with the following event:
  Event E1: The adversary uses the hash-query, execute-query, and send-query to successfully acquire the authentication parameters used in the authentication process that are protected by the hardness of the collision resistance property of hash functions. The success probability of this event can be acquired using the birthday paradox:

$$Succ_{E1} \leq \frac{q_h^2}{2^{l+1}} + \frac{(q_s + q_e)^2}{2p} + \frac{q_s}{2^l} \quad (3)$$

  Therefore, the success probability of $\mathsf{G}_1 \Rightarrow win_{\mathcal{A}}$ from Eqs. 3 can be computed as:

$$Succ_{\mathsf{G}_1} - Succ_{\mathsf{G}_0} \leq \frac{q_h^2}{2^{l+1}} + \frac{(q_s + q_e)^2}{2p} + \frac{q_s}{2^l} \quad (4)$$

- **Game G_2:** G_2 and G_1 are inseparable as to successfully acquire a group key from communication $\mathcal{A}$ first needs to win G_1 to get authenticated successfully. The probability that $\mathsf{G}_2 \Rightarrow win_{\mathcal{A}}$ is:

$$Succ_{\mathsf{G}_2} - Succ_{\mathsf{G}_1} \leq \text{Adv}_A^{\text{QRA}} \quad (5)$$

  where the term $\text{Adv}_A^{\text{QRA}}$ denotes the probability of the adversary successfully evaluates the quadratic residues by randomly guessing within the entire search space of size 2^m.
- **Game G_3:** This game is indistinguishable from G_2 where $\mathcal{A}$ successfully forges a token to be shared with any authenticated entity. To forge a fake token, $\mathcal{A}$ must acquire pseudonyms to pass verification at RSU. Therefore,

$$Succ_{\mathsf{G}_2} = Succ_{\mathsf{G}_3} \quad (6)$$

- **Game G_4:** Once the test query is executed, the adversary has to determine if all the secret parameters and the keys for are correct or not. The probability that $\mathcal{A}$ successfully guesses the correct bits is:

$$Succ_{\mathsf{G}_4} = 1/2 \quad (7)$$

Now, from eqs. 2, 7 we get:

$$\frac{1}{2} Adv_{\mathcal{A}} = \left| \Pr[Succ_{\mathsf{G}_0}] - \frac{1}{2} \right| = \left| \Pr[Succ_{\mathsf{G}_0}] - \Pr[Succ_{\mathsf{G}_4}] \right| \quad (8)$$

By simplifying eqs. (4) to (6) and (8) we finally get:

$$Adv_{\mathcal{A}} \leq \frac{q_h^2}{2^l} + \frac{(q_s + q_e)^2}{p} + \frac{q_s}{2^{l-1}} + 2 \cdot \text{Adv}_A^{\text{QRA}} \quad (9)$$

This negligible probability advantage protects our scheme against feasible polynomial-time attacks performed by $\mathcal{A}$. Therefore, our scheme is provably secure, grounded in established cryptographic principles and the defined adversarial model.

VII. PERFORMANCE ANALYSIS

The applied methodology to analyze the proposed scheme performance follows the exact same method adapted by Zhang et al. [23] He et al. [27] Pournaghi et al. [28] and many others. Using this well-established methodology, the performance of the proposed scheme is analyzed by measuring crucial parameters (CPs) such as computation cost and communication overhead in every stage of communication.

TABLE 4. Execution times of cryptographic operations.

Operation	Description	Timing (ms)
T_{div}	Division of two large primes	≈ 0.0021
T_{mul}	Multiplication of two large primes	≈ 0.0018
T_{exp}	Exponentiation operation	≈ 0.0083
T_{E-add}	Point addition on curve `secp256k1`	≈ 0.0326
T_{E-mul}	Point multiplication on curve `secp256k1`	≈ 0.8069
T_h	Hash operation	≈ 0.0011
T_{leg}	Legendre symbol check	≈ 0.0137
T_{bp}	Bilinear-pairing of two groups	≈ 3.8721

A. ANALYSIS OF COMPUTATION COST

The computation cost is measured in the total number of cryptographic operations performed in a certain phase. This strategy follows the same methodology adopted by Wei et al [38], Feng et al. [39], Naskar et al. [40] and various other IoV authentication scheme. Notably, this simulation strategy focuses solely on the cryptographic functions utilized in the proposed scheme to determine their average execution times. Each cryptographic operation is individually executed to gauge the average computation time. Importantly, vehicle speed or mobility parameters do not impact the cryptographic execution time. Rather, the execution time is fully dependent on the simulation platform and available resources, such as computation power and memory as presented below.

B. SIMULATION SETUP

To compute the execution times of the cryptographic functions, we have used an Intel i7-6500U @ 2.50GHz CPU with two cores and four logical processors having 16GB of physical memory(RAM). We have used a Linux virtual environment and implemented the cryptographic functions in C programming language using the OpenSSL cryptographic library [41] and the PCB library [42]. The security parameter is set to $128-bits$ security. We have used the standard cryptographic curve "$secp256k1$"for comparative study, which provides a security level of $128-bits$. The execution times for all cryptographic operations are listed in Table 4.

During the authentication phase, two important steps are performed: (i) authentication message generation and (ii) authentication message verification. To generate a Hello-type1 authentication message, the OBU/TPU has to compute $(1*T_{mul}+3*T_h)$ computations. For a Hello-type2 message, only three hash operations are performed. The RSU computes a single hash value (T_h) to generate a handshake (Hsk) and attach it to the authentication message. To perform type1 verification, TA computes three hash $(3*T_h)$ as hsk^*, h_2^*, h_3^* and one division $(1*T_{div})$ operation, so total computation is $(3*T_h+1*T_{div})$. Similarly, for type2 verification, it performs $(3*T_h)$ computations only. Therefore, from generating a type1 authentication message at OBU/TPU to its verification at TA costs $(6*T_h+1*T_{mul}+1*T_{div})$ computations, and for type-2, it costs $(6*T_h)$ computations only. The computation costs for the authentication phase are presented in Table 5. During the pseudo ID assignment phase, TA computes five hash operations, and RSU computes two hash values. The OBU/TPU computes h_{H1}, h_4^* for getting and verifying the value of V_RID and also h_5 to generate the pseudo ID. So, to finish the pseudo ID assignment phase, the total computations needed at TA, RSU, and OBU/TPU is $9*T_h$ as presented in Table 5.

During the group key-sharing phase, the TPU/OBU needs to compute one hash function to verify the legitimacy of the received values in M_k and the sender TA. Then it computes the Legendre symbol check for regenerating the intermediate value IV_k to get the $n-bit$ long key K. So it takes $(1*T_h+n*T_{leg})$ computations to get the $n-bit$ key from TA. Checking for Legendre symbol $leg()$ to verify if a value $v \in qr/nqr$ based on the result $leg(v)=\{-1/1\}$ is a very fast and lightweight process. To establish a V2X communication, the token-sharing phase must be performed. This process involves one hash operation (h_8) at the sender, two hash operations (h_8^*, h_9^*) at the RSU, and one hash operation at the receiver. In total, the token-sharing process requires $4\times T_h = 0.0044$ ms to complete. This extremely fast execution enables any RSU in the network to handle a maximum token-sharing rate of $227,272$ tokens/second, allowing RSUs to achieve high scalability in managing massive V2X communications.

C. COMMUNICATION COST

The communication cost specifies the size of the messages sent during authentication, key sharing, and V2X-token sharing. In our proposed scheme, the size of p can be considered as 33 $bytes$, VID is 66 bytes and V_RID is 48 $bytes$ in which 32 $bytes$ are for rp part and 16 bytes are for sqn part. A timestamp is of 4 $bytes$, and for the first hash function (H), SHA-256 produces a hash value of 32 $bytes$; the second hash function ($H1$) SHA-384 produces a hash value of 48 $bytes$.

A type-1 authentication message consists of two 32 $bytes$ hash values (h_2, h_3), one timestamp $TS1$ (4 $bytes$), the sequence number (16 $bytes$) and the value of $m=66$ $bytes$. Therefore, the total size of a type-1 authentication message is $(3*32+16+4+66)=182$ $bytes$. Similarly, for a type-2 authentication, the size of the message will be much less as m is not present in type-2; therefore, only $(3*32+16+4)=116$ $bytes$ are sent to the TA from the OBU/TPU for type-2 authentication message. Table 5 represents the message sizes for different phases. TA shares a message $[IV2, IV3, h_6, h_4, rv2, TS3]$ at the pseudo ID assignment phase to the RSU. The size of $IV2$ is of 48 $bytes$ using SHA-384; $IV3, h_6, h_v$ are of 32 $bytes$ each and $rv2$ is of 28 $bytes$. Therefore, TA shares $(48+3*32+28+4)=176$ $bytes$ with RSU and RSU shares $(48+32+2*4+16)=104$ $bytes$ to the OBU/TPU.

During group key sharing, TA generates a mix of 128 values containing qrs and $nqrs$ for a key of example length $128-bits$. TA selects the mix of qrs and $nqrs$ such that the average size of the values is within two bytes. This is valid as p, q are very large prime numbers, and selecting only 128 values with a mix of qrs and $nqrs$ based on a Legendre symbol check is an easy task. Also, once these 128 values are selected, they can be reused to send another group key with a different sequence within the specified time slot δTS.

TABLE 5. Computation costs for different phases of the proposed scheme.

Phase	Operation	Computations	Total	Time (ms)	Message Size (bytes)
Authentication	1. Authentication Message Generation	Hello-type1: $1*T_{mul}+3*T_h$ Hello-Type2: $3*T_h$	type1: $6*T_h+1*T_{mul}+1*T_{div}$	0.0105	182
	2. Authentication Message Verification	Hello-type1: $1*T_{div}+3*T_h$ Hello-type2: $3*T_h$	type2: $6*T_h$	0.0066	116
Pseudo ID Assignment	1. Operations by TA 2. Operations at RSU 3. Operations at OBU/TPU	$5*T_h$ $2*T_h$ $2*T_h$	$9*T_h$	0.0099	TA $\rightarrow$ RSU: 176 RSU $\rightarrow$ Vehicle: 104
Sharing Group Key	Computing the n-bit key by OBU/TPU	$1*T_h+n*T_{leg}$	$1*T_h+n*T_{leg}$	$0.0011+n*0.0137$	304, $K=128$ bits
V2X Token Sharing	1. Sender token generation 2. RSU one-time Computation 3. Receiver token regeneration	$1*T_h$ $2*T_h$ $1*T_h$	$4*T_h$	0.0044	40

Therefore, the maximum size of the group key message is $(128*2+32+16)=304$ *bytes* to share a key of $128-bit$ long. Here, the extra 32 *bytes* is for the hash value h_7, and 16 *bytes* is for the old sequence number. Similarly, for a $80-bit$ key, it sends $(80*2+32+16)=208$ *bytes* and for $64-bit$ key, it sends $(64*2+32+16)=176$ *bytes* only.

During the V2X token-sharing process, each OBU/TPU is assigned a 2 *bytes* long name in $list-X$. So, $S.Info$, $R.Info$ is of 2 *bytes* each and $IV4$ is of 32 *bytes*. Therefore, the size of the token sharing message is $(2*2+32+4)=40$ *bytes*. The size of the token can be from some bits $(80-bit, 128-bit)$ to a few bytes long, and it does not affect $IV4$ until it is $\leq$ 32 *bytes*. This provides more flexibility in selecting the tokens depending on the communication entities in various X and security and privacy requirements. Clearly, the communication overhead to perform authentication, key-sharing, and V2X token sharing in our scheme is significantly less, making it very efficient in time-critical IoV communications.

It is notable that parameters like channel path loss and transmission latency, directly tied to the physical channel, are excluded from our analysis to focus solely on cryptographic and security-related computation and communication in this study.

D. COMPARATIVE STUDIES

In order to analyze the computational and communication efficiency gain of the proposed scheme in comparison with the existing schemes, we perform a comprehensive comparative study. For this purpose, we have selected the state-of-the-art elliptic curve (EC)-based authentication schemes proposed in [23] and [27] and the bilinear pairing (BP)-based schemes proposed in [3], [28], and [29]. To evaluate the computation costs for the comparative schemes, we have used the same hardware setup as mentioned in Section VII-B to execute the EC and BP cryptographic operations listed in Table 4 for comparative fairness. As presented in Table 6, the proposed scheme requires significantly lower computational costs to perform authentication of vehicles than the state-of-the-art schemes. The main reason behind such significant computational efficiency gain is the use of general one-way-hash functions and simple multiplication and division operations in our scheme. The proposed scheme requires only 116 bytes of data to be sent by the vehicle to be able to get verified, which is significantly lower than state-of-the-art comparative schemes (Table 6). The authentication verification bandwidth defined by the number of messages that are verified per second is presented in Figure 7, which shows that the proposed scheme significantly outperforms the comparative schemes by verifying a large number of authentication messages.

TABLE 6. Computation cost comparison for performing authentication in various schemes with our scheme.

Scheme	Computations for Authentication	Time (ms)	Message bytes
Jing et al. [23]	$5*T_{E-mul}+1*T_{E-add}+3*T_h$	4.0704	200
Tzeng et al [3]	$2*T_{bp}+2*T_{E-mul}$	9.3580	196
Debiao et al. [27]	$6*T_{E-mul}+5*T_h+2*T_{E-add}$	5.4869	260
Yujue et al. [29]	$2*T_{bp}+15*T_{exp}+11*T_h+5*T_{E-mul}$	11.9153	388
Seyed et al. [28]	$3*T_{bp}+5*T_{E-mul}+4*T_h$	15.6541	320
Proposed Scheme	type1: $6*T_h+1*T_{mul}+1*T_{div}$ type2: $6*T_h$	0.0105 0.0066	182 116

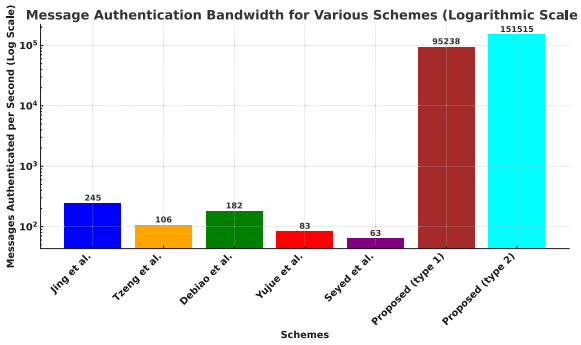

FIGURE 7. Authentication bandwidth by number of messages verified per second for different schemes in logarithmic scale.

E. DISCUSSIONS

The overall insights of the proposed scheme in terms of robust security strength, efficiency, reliability, and scalability are discussed below:

1) The proposed scheme uses a hierarchical architecture where the TA manages high-level tasks, while

RSUs handle localized operations like authentication and token sharing, ensuring scalability and reducing the TA's tasks. Lightweight cryptographic operations, such as hash functions and pseudo-random identities, enable fast processing, with authentication and token sharing completed within a fraction of milliseconds and minimal communication overhead. Independent phases and time-based thresholds ensure efficient request handling, allowing concurrent processing and robust security, making the scheme suitable for large-scale, real-time V2X communications. In an unlikely scenario, if TA becomes temporarily unavailable, the hierarchical design ensures that real-time operations, such as pseudonym authentication and V2X communication, remain unaffected, as they are managed independently by RSUs. Only centralized tasks like registering new vehicles, assigning pseudonyms, and updating revocation lists may experience temporary interruptions until the TA is restored.

2) The proposed scheme is built on mathematically proven cryptographic techniques with well-defined security guarantees. The computational complexity of breaking these techniques far exceeds feasible limits under current and foreseeable technology. Combined with the layered security architecture, the scheme ensures robust protection against known attacks, providing a secure and efficient solution for IoV networks.
3) The proposed scheme seamlessly integrates with existing IoV standards like IEEE 802.11p, DSRC, and C-V2X, ensuring compatibility with current infrastructures, backward compatibility with legacy systems, and scalability for technologies like 5G-based V2X. Leveraging the hierarchical architecture and lightweight cryptographic operations, our scheme is specifically designed for seamless integration into resource-constrained IoV environments. Cryptographic computations like hash operations, modular arithmetic, and XOR are computationally efficient, requiring less than 1 ms per operation, while memory requirements are kept very low to accommodate pseudonyms, session keys, and temporary values.

VIII. CONCLUSION

In this paper, we first highlight the critical importance of preserving privacy in IoV communications, reviewing various approaches proposed by researchers, along with their strengths and limitations. We then introduce a pseudo-random identity-based anonymous lightweight and privacy-preserving authentication scheme to address critical security challenges in IoV communication. By leveraging pseudonym-based authentication and dynamic key-sharing mechanisms, the scheme ensures secure and scalable V2X communication while preserving user privacy. Each phase of our proposed scheme is designed to be both computationally and communicatively efficient, making it fast and lightweight. The scheme is designed to meet real-world IoV implementation requirements, ensuring its practicality and applicability in diverse IoV environments. We provide a detailed analysis to demonstrate the scheme's security validity and effectiveness against various attacks where other schemes have failed. However, our future goal is to extend the proposed scheme to support a decentralized approach while maintaining a significantly fast execution and simple authentication structure.

REFERENCES

[1] C. R. Storck and F. Duarte-Figueiredo, "A survey of 5G technology evolution, standards, and infrastructure associated with vehicle-to-everything communications by Internet of Vehicles," *IEEE Access*, vol. 8, pp. 117593–117614, 2020.

[2] X. Shen, R. Fantacci, and S. Chen, "Internet of Vehicles [scanning the issue]," *Proc. IEEE*, vol. 108, no. 2, pp. 242–245, Feb. 2020.

[3] S.-F. Tzeng, S.-J. Horng, T. Li, X. Wang, P.-H. Huang, and M. K. Khan, "Enhancing security and privacy for identity-based batch verification scheme in VANETs," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3235–3248, Apr. 2017.

[4] K. Ren, Q. Wang, C. Wang, Z. Qin, and X. Lin, "The security of autonomous driving: Threats, defenses, and future directions," *Proc. IEEE*, vol. 108, no. 2, pp. 357–372, Feb. 2020.

[5] P. Vijayakumar, M. Azees, V. Chang, J. Deborah, and B. Balusamy, "Computationally efficient privacy preserving authentication and key distribution techniques for vehicular ad hoc networks," *Cluster Comput.*, vol. 20, no. 3, pp. 2439–2450, Sep. 2017.

[6] H.-K. Kong, M. K. Hong, and T.-S. Kim, "Security risk assessment framework for smart car using the attack tree analysis," *J. Ambient Intell. Humanized Comput.*, vol. 9, no. 3, pp. 531–551, Jun. 2018.

[7] S. Eichler, C. Schroth, and J. Eberspächer, "Car-to-car communication," in *Proc. VDE-Kongress-Innov. Eur.*, vol. 1, 2006, pp. 35–40.

[8] Y. Cao, T. Jiang, O. Kaiwartya, H. Sun, H. Zhou, and R. Wang, "Toward pre-empted EV charging recommendation through V2V-based reservation system," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 51, no. 5, pp. 3026–3039, May 2021.

[9] H. Zhou, K. Jiang, S. He, G. Min, and J. Wu, "Distributed deep multi-agent reinforcement learning for cooperative edge caching in Internet-of-Vehicles," *IEEE Trans. Wireless Commun.*, vol. 22, no. 12, pp. 9595–9609, Dec. 2023.

[10] T. Garg, N. Kagalwalla, P. Churi, A. Pawar, and S. Deshmukh, "A survey on security and privacy issues in IoV," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 5, p. 5409, Jun. 2020.

[11] R. Kumar, P. Kumar, R. Tripathi, G. P. Gupta, and N. Kumar, "P2SF-IoV: A privacy-preservation-based secured framework for Internet of Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 22571–22582, Nov. 2022.

[12] G. S. Aujla, A. Singh, M. Singh, S. Sharma, N. Kumar, and K. R. Choo, "BloCkEd: Blockchain-based secure data processing framework in edge envisioned V2X environment," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 5850–5863, Jun. 2020.

[13] E. S. Ali, M. K. Hasan, R. Hassan, R. A. Saeed, M. B. Hassan, S. Islam, N. S. Nafi, and S. Bevinakoppa, "Machine learning technologies for secure vehicular communication in Internet of Vehicles: Recent advances and applications," *Secur. Commun. Netw.*, vol. 2021, pp. 1–23, Mar. 2021.

[14] 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.

[15] A. Kumar Yadav, M. Shojafar, and A. Braeken, "IVFAS: An improved Vehicle-to-Fog authentication system for secure and efficient fog-based road condition monitoring," *IEEE Trans. Veh. Technol.*, vol. 73, no. 9, pp. 12570–12584, Sep. 2024.

[16] Y. Zhou, H. Tan, and K. C. A. A. Iroshan, "A secure authentication and key agreement scheme with dynamic management for vehicular networks," *Connection Sci.*, vol. 35, no. 1, Dec. 2023, Art. no. 2176825.

[17] Q. Xie, Z. Ding, and P. Zheng, "Provably secure and anonymous V2I and V2V authentication protocol for VANETs," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 7, pp. 1–10, May 2023.

[18] C.-M. Chen, B. Xiang, Y. Liu, and K.-H. Wang, "A secure authentication protocol for Internet of vehicles," *IEEE Access*, vol. 7, pp. 12047–12057, 2019.
[19] S. H. Islam, M. S. Obaidat, P. Vijayakumar, E. Abdulhay, F. Li, and M. K. C. Reddy, "A robust and efficient password-based conditional privacy preserving authentication and group-key agreement protocol for VANETs," *Future Gener. Comput. Syst.*, vol. 84, pp. 216–227, Jul. 2018.
[20] Y. Sun, L. Wu, S. Wu, S. Li, T. Zhang, L. Zhang, J. Xu, Y. Xiong, and X. Cui, "Attacks and countermeasures in the Internet of Vehicles," *Ann. Telecommun.*, vol. 72, nos. 5–6, pp. 283–295, Jun. 2017.
[21] B. Ying and A. Nayak, "Anonymous and lightweight authentication for secure vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 10626–10636, Dec. 2017.
[22] T. Limbasiya and D. Das, "IoVCom: Reliable comprehensive communication system for Internet of vehicles," *IEEE Trans. Dependable Secur. Comput.*, vol. 18, no. 6, pp. 2752–2766, Nov. 2021.
[23] J. Zhang, J. Cui, H. Zhong, Z. Chen, and L. Liu, "PA-CRT: Chinese remainder theorem based conditional privacy-preserving authentication scheme in vehicular ad-hoc networks," *IEEE Trans. Dependable Secur. Comput.*, vol. 18, no. 2, pp. 722–735, Mar. 2021.
[24] C. Lee and Y.-M. Lai, "Toward a secure batch verification with group testing for VANET," *Wireless Netw.*, vol. 19, no. 6, pp. 1441–1449, Jan. 2013.
[25] J. Liu, Q. Li, H. Cao, R. Sun, X. Du, and M. Guizani, "MDBV: Monitoring data batch verification for survivability of Internet of vehicles," *IEEE Access*, vol. 6, pp. 50974–50983, 2018.
[26] H. Sikarwar and D. Das, "Towards lightweight authentication and batch verification scheme in IoV," *IEEE Trans. Dependable Secur. Comput.*, vol. 19, no. 5, pp. 3244–3256, Sep. 2022.
[27] 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.
[28] S. M. Pournaghi, B. Zahednejad, M. Bayat, and Y. Farjami, "NECPPA: A novel and efficient conditional privacy-preserving authentication scheme for VANET," *Comput. Netw.*, vol. 134, pp. 78–92, Apr. 2018.
[29] Y. Wang, Y. Ding, Q. Wu, Y. Wei, B. Qin, and H. Wang, "Privacy-preserving cloud-based road condition monitoring with source authentication in VANETs," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 7, pp. 1779–1790, Jul. 2019.
[30] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen, "ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications," in *Proc. IEEE 27th Conf. Comput. Commun.*, Apr. 2008, pp. 1229–1237.
[31] S. Chen, J. Hu, Y. Shi, Y. Peng, J. Fang, R. Zhao, and L. Zhao, "Vehicle-to-everything (v2x) services supported by LTE-based systems and 5G," *IEEE Commun. Standards Mag.*, vol. 1, no. 2, pp. 70–76, May 2017.
[32] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," *Proc. IEEE*, vol. 99, no. 7, pp. 1162–1182, Jul. 2011.
[33] J. Jenefa and E. A. M. Anita, "Identity-based message authentication scheme using proxy vehicles for vehicular ad hoc networks," *Wireless Netw.*, vol. 27, no. 5, pp. 3093–3108, Jul. 2021.
[34] C. Cocks, "An identity based encryption scheme based on quadratic residues," in *Cryptography and Coding*. Cham, Switzerland: Springer, 2001, pp. 360–363.
[35] A.-M. Nica, *Quadratic Residues and Applications in Cryptography*. New York, NY, USA: Academic, 2020.
[36] K. H. Rosen, *Elementary Number Theory and its Applications*, vol. 1. Reading, MA, USA: Addison-Wesley, 2005, pp. 288–325.
[37] D. A. Burgess, "The distribution of quadratic residues and non-residues," *Mathematika*, vol. 4, no. 2, pp. 106–112, Dec. 1957.
[38] L. Wei, J. Cui, H. Zhong, I. Bolodurina, and L. Liu, "A lightweight and conditional privacy-preserving authenticated key agreement scheme with multi-TA model for fog-based VANETs," *IEEE Trans. Dependable Secur. Comput.*, vol. 20, no. 1, pp. 422–436, Jan. 2023.
[39] X. Feng, Q. Shi, Q. Xie, and L. Wang, "P2BA: A privacy-preserving protocol with batch authentication against semi-trusted RSUs in vehicular ad hoc networks," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 3888–3899, 2021.
[40] S. Naskar, C. Brunetta, G. Hancke, T. Zhang, and M. Gidlund, "A scheme for distributed vehicle authentication and revocation in decentralized VANETs," *IEEE Access*, vol. 12, pp. 68648–68667, 2024.
[41] E. Käsper, "Fast elliptic curve cryptography in OpenSSL," in *Financial Cryptography and Data Security*. Cham, Switzerland: Springer, 2012, pp. 27–39.
[42] L. Ben, "On the implementation on pairing-based cryptosystems," Ph.D. dissertation, Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA, 2007.

SUJASH NASKAR (Member, IEEE) received the B.Sc. degree in computer science and the M.Sc. degree from the University of Calcutta, Kolkata, India, in 2017 and 2019, respectively. He is currently pursuing the Ph.D. degree in IoT security and privacy with STC Research Group, Mid Sweden University, Sundsvall, Sweden. From November 2022 to July 2023, he was a Guest Research Fellow with Simula UiB, Bergen, Norway. His research interests include privacy and security management for vehicular ad hoc networks (VANET), especially developing lightweight privacy-preserving authentication schemes for VANET. His awards and honors include receiving the prestigious Ericsson grant for a long-term research visit.

GERHARD HANCKE (Fellow, IEEE) received the B.Eng. and M.Eng. degrees in computer engineering from the University of Pretoria, South Africa, in 2002 and 2003, respectively, and the Ph.D. degree in computer science from the University of Cambridge, U.K., in 2009. Previously, he was a Researcher with the Smart Card and IoT Security Centre, and as a Teaching Fellow with the Department of Information Security, Royal Holloway, University of London. He is currently a Professor with the Department of Computer Science, City University of Hong Kong. His research interests include security, reliable communication, and distributed sensing for the Industrial Internet of Things.

TINGTING ZHANG received the B.Sc. and M.Sc. degrees in computer science and engineering from Fudan University, Shanghai, China, in 1982 and 1984, respectively, and the Ph.D. degree in computer science and engineering from Linköping University, Linköping, Sweden, in 1993. She is currently a Professor of computer engineering with the Department of Information and Communication Systems, Mid Sweden University, Sundsvall, Sweden. Her current research work is in the areas of wireless sensor networks and security.

MIKAEL GIDLUND (Senior Member, IEEE) received the Licentiate of Engineering degree in radio communication systems from the KTH Royal Institute of Technology, Stockholm, Sweden, in 2004, and the Ph.D. degree in electrical engineering from Mid Sweden University, Sundsvall, Sweden, in 2005. From 2008 to 2015, he was a Senior Principal Scientist and a Global Research Manager of Wireless Technologies with ABB Corporate Research, Västerås, Sweden. From 2007 to 2008, he was a Project Manager and a Senior Specialist with Nera Networks AS, Bergen, Norway. From 2006 to 2007, he was a Research Engineer and a Project Manager with Acreo AB, Hudiksvall, Sweden. Since 2015, he has been a Professor of computer engineering with Mid Sweden University. He holds more than 20 patents (granted and pending) in the area of wireless communication. His current research interests include wireless communication and networks, wireless sensor networks, access protocols, and security. He is an Associate Editor of IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS and IEEE JOURNAL OF EMERGING AND SELECTED TOPICS IN INDUSTRIAL ELECTRONICS.