Vehicular Ad hoc Networks (VANETs) authentication schemes need to consider mobility and rapidly changing topologies in addition to an unreliable wireless channel communication. IEEE 1609.2, VANET security standard, suggests Elliptic Curve Digital Signature Algorithm (ECDSA) for the authentication of Vehicle-to-Vehicle (V2V) messages. However, it has the drawback of expensive computations for verification. Therefore, we provide a smart probabilistic verification strategy for the authentication of V2V communication. We propose a practical and efficient strategy that makes use of secure ECDSA but still decreases the computation time for the most relevant packets. Furthermore, we maintain conditional privacy of vehicles through a lightweight vehicle registration process with a regional registering authority. In comparison to IEEE 1609.2, our algorithm reduces the packet loss ratio that occurs due to messages expiring in the verification queue.

Key words: VANET, Broadcast, Authentication, Privacy, ECDSA, Wireless.

* email: kanika@auburn.edu
1 INTRODUCTION

VANETs are designed primarily for applications that ensure real time safety of human life. According to the latest report by National Highway Traffic Safety Administration, the year 2012 saw 33,561 human deaths per 100 million vehicle miles travelled [4], a 3.3-percent increase from year 2011. It emphasizes the importance of vehicular networking and communication for minimizing accidents and improving traffic conditions. At the same time, security of the VANET communication is necessary to avoid malicious entities sending out incorrect information.

A VANET model is shown in Figure 1. A VANET essentially comprises of road-side units (RSU) and vehicle’s on-board unit (OBU) equipped with wireless capabilities. Road-side units are fixed entities installed at the side of roads. RSUs form the infrastructure backbone of VANETs. They communicate in infrastructure (wired) mode with the internet and application servers and wirelessly with the vehicle’s OBU. On the other hand, OBUs can only communicate wirelessly either amongst themselves or with the infrastructure. In addition, OBUs are not fixed and move along with the vehicle on which they are placed. The dissemination of safety information in VANET takes place largely through broadcast [5]. The means of communication in VANETs are defined by the DSRC standards [1] for Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I) communications.

However, wireless communication is not reliable. For instance, let us consider the scenario shown in Figure 1, where an accident has taken place at the intersection. The vehicles involved in the accident immediately broadcast messages informing the neighbouring vehicles about the accident. Nonetheless, there may be a malicious vehicle which can modify the original content of the message and re-broadcast it further. The vehicles receiving erroneous information are not aware of the accident. Such circumstances contradict the purpose of VANETs. Therefore, it becomes extremely important to secure these communications. IEEE 1609.2 [2], security standard for VANET, asserts the authentication of VANET broadcasts with Elliptic Curve Digital Signature Algorithm (ECDSA).

ECDSA is a digital signature based on elliptic curve cryptography such that the elliptic curve is defined over a finite field of a prime number $p$ [18]. ECDSA with primes $p$ of sizes 224, 256 and greater are considered to be secure because solving their discrete logarithmic is hard [9]. Yet, ECDSA is computationally expensive, an ECDSA signature generation takes 4 milliseconds while the verification takes 22 milliseconds [16], on a 400 Mhz proces-
Besides, when all vehicles will be broadcasting messages at a frequency of 10 Hz, the verification queue size will increase at a rapid rate. Since the messages are valid only for a certain time period, some of them will time out waiting to be verified. Malicious vehicles can take advantage of this fact by increasing signature verification time through signature flooding of fake messages.

Therefore, smart verification strategy is required. Hence, we design a probabilistic verification method based on the distance and direction of the communicating vehicles with respect to each other. At the same time, we make available a privacy controlled mechanism, where Regional Registering Authorities (RRA) are the entities responsible for disclosing the original identities of the vehicles communicating with pseudonyms. In accordance, with the early deployment stages of VANETs, our solution does not require a strong backbone of the infrastructure entities RRA. Another advantage of our solution is that the vehicles use the information available in the broadcasts to compute the probability. Thus, it does not increase the communication overhead of the broadcasts.

The main contributions of this work are as follows. First, we develop a test-bed of vehicular communication where the vehicles are communicating
among themselves (V2V) as well as with the roadside infrastructure (V2I). Vehicle movements are taken from a realistic urban mobility trace generator, SUMO [20]. Further, the Nakagami probabilistic radio propagation model [28] is used for RF communication. Second, we implement V2V communication in the form of broadcasts. We authenticate these broadcasts with ECDSA (IEEE 1609.2 security standard). Then, we analyse the issue of packet loss taking place as a result of the long verification delay and how the packet processing ratio decreases with increasing number of broadcasting neighbours. Third, we use a lightweight scheme with inexpensive hash and XOR operations to conduct the V2I authentication process, where the vehicle and a Regional Registering Authority authenticate each other. Fourth, for V2V communication we use ECDSA coupled with a smart probabilistic verification approach to solve the above realized issue of packet loss. We also safeguard the original identity of the broadcasting vehicle by providing it a pseudonym.

The rest of the paper is organized as follows. In Section 2, we present the related work. Section 3 discusses the crucial attacks threatening the security of VANETs. Section 4 describes the construction of a realistic VANET scenario. Section 5 examines broadcast authentication properties necessary for VANETs and implements IEEE 1609.2. Section 6 explains the design goals and proposes a mobility-based probabilistic verification solution, such that it facilitates conditional privacy. Section 7 presents results and discussion. We conclude in Section 8.

2 RELATED WORK

Many researchers have discovered different threats to the security of vehicular networks and have proposed solutions for the identified issues. Below, we discuss existing schemes to deal with the security issues in vehicular networks, along with the effectiveness and performance of the schemes and their practical implementation issues.

Security of vehicular communication depends on authenticating the identity of the source sending the messages, such that it cannot deny the original transmission of the message (non-repudiation). On the other hand, to provide resilience to identity and location traceability attacks, it becomes important to restrict the original identity of the sender from being known to any of the receivers. Most of the vehicular transmission is broadcast; therefore, we further look into approaches to broadcast authentication.

Timed Efficient Stream Loss-tolerant Authentication (TESLA)[31], has
been widely used in various resource constrained wireless networks for broadcast authentication. It is built over Message Authentication Codes (MAC) using a symmetric key. However, the scheme is made asymmetric with the delayed disclosure of the symmetric key. TESLA lacks non-repudiation and immediate authentication, and requires time synchronization between the communicating entities.

IEEE 1609.2 standard for VANETs recommends the use of elliptic curve based digital signatures, ECDSA [18], since it overcomes the above drawbacks. Correspondingly, it has been identified that ECDSA are computationally expensive signatures. To control the computations, researchers have suggested many methods, such as signature amortization, single signature for multiple packets [30, 24]; omitting signatures and certificates with messages sent to a known neighbour; or omitting signature and certificate verification of messages received from known neighbours [33]. Both the above cases, signature amortization and omitting signatures from certain messages, compromise security due to lack of non-repudiation. In addition, the latter methods fail when insider attacks are made.

Use of probability has been suggested in [16] for VANETs. It is a selective authentication scheme which isolates attackers by sending recursive warning messages when false packets are detected. They assumed that all vehicles, legitimate and fake, will be sending out their original identities and mark the vehicle’s ID from which a false message is received. When vehicles are using pseudonyms, this scheme is not appropriate. Another suggested method is to accelerate signature-based broadcast authentication using inter-nodal cooperation [13]. This scheme’s effectiveness depends on the number of hops occurring for each message. For a single-hop broadcast transmission, it acts as an all verification scheme.

Bilinear operations have been used for batch verifications [36, 37] and short signatures [8]. The use of bilinear operations suggested in theory impractical assumptions [14]. Alternatively other researchers have used point multiplications for batch verification in emergency communications [17]. In [35], message verification is accomplished by vehicles with help from the Road-Side Units (RSU). All vehicles are assumed to be connected to an RSU. RSU assigns a pseudo identity and a shared symmetric secret key to each vehicle. The symmetric keyed-hash message authentication (HMAC) code is used for authentication. The receiver obtains the symmetric key from RSU to verify the message. It requires a very strong RSU backbone, assumes that there is no packet loss between RSU and OBU communication and lacks non-repudiation and immediate authentication.
<table>
<thead>
<tr>
<th>Scheme</th>
<th>Aim</th>
<th>Method</th>
<th>Drawback</th>
</tr>
</thead>
<tbody>
<tr>
<td>TESLA</td>
<td>Lightweight Message Authentication</td>
<td>MAC with delayed key disclosure</td>
<td>Non-repudiation, Immediate Authentication, time synchronization</td>
</tr>
<tr>
<td>Signature Amortization</td>
<td>Reduce overhead for Message Authentication</td>
<td>One signature for multiple packets</td>
<td>Non-repudiation</td>
</tr>
<tr>
<td>Selective Authentication</td>
<td>Reduce overhead for Message Authentication</td>
<td>Recursive warning messages</td>
<td>Not suitable with Privacy alternatives</td>
</tr>
<tr>
<td>Batch Verification/Short Signatures</td>
<td>Lightweight Message Authentication</td>
<td>Bilinear Pairing</td>
<td>Impractical assumptions [14]</td>
</tr>
<tr>
<td>RAISE</td>
<td>Privacy, Lightweight Message Authentication</td>
<td>keyed-hash message authentication (HMAC)</td>
<td>Strong RSU backbone, Non-repudiation, Immediate Authentication</td>
</tr>
<tr>
<td>Time Restricted keys</td>
<td>Privacy</td>
<td>Several Public Private key pairs</td>
<td>Storage, reloading</td>
</tr>
<tr>
<td>Group Signatures</td>
<td>Privacy</td>
<td>Bilinear Pairing</td>
<td>Impractical assumptions [14]</td>
</tr>
<tr>
<td>ID-based Signatures</td>
<td>Privacy</td>
<td>Original Identity based signature</td>
<td>Forgery attack</td>
</tr>
</tbody>
</table>
Group signatures are a privacy providing mechanism [23]. In this a limited number of vehicles form a group and select a group manager amongst themselves, such that they allow the group manager to sign for each member. Privacy is controlled since only the group manager can identify the original message originator. Signatures generation comprises of hashes or MACs. The complexity is in the process of forming groups and re-electing group managers and keeping track of neighbour information. Some group signatures [7], are generated using bilinear pairing and considered to be based on impractical assumptions, according to [14].

On the other hand, privacy can be obtained based on location coordinates [6]. It is a combination of ID-based signature [36] and vehicles physical location coordinates taken together to create a pseudonym. Message verification process has complexity similar to ECDSA verification. Moreover, the security factor seems to vary with the network density. Time restricted public-private key pairs have also been proposed as a solution to privacy issues [32]. Each key pair is also accompanied with a public key certificate, and the complete set is valid for a given time duration. The number of key pairs that can be loaded to each vehicle depends on the vehicle’s storage capacity. This scheme requires reloading of key pairs after all the existing pairs are used up.

In Table 1, we summarize the existing schemes, along with their aim, method and drawback. We notice that most of the suggested security protocols aim at reducing the time involved in the security algorithms by compromising security or through the use of additional hardware. In contrast, some schemes are theoretically proven to be secure, but have impractical assumptions. We also observe the absence of practical implementations and dynamic schemes that can adapt to changes in network density.

3 CLASSIFICATION OF ATTACKS ON VANETS

Vehicular communication is prone to numerous attacks. The study of attacks is important in VANETs because of two reasons. First, VANETs involves real-time safety of human life. Secondly, a thorough study of attacks can assist in providing proficient solutions against them.

We briefly describe five attack attributes here and then classify all the attacks on the basis of these attributes. Some of these attributes are given in [32] and [29].

- The goal of attackers can be either to disturb the working of the system (malicious) or to gain something from the attack (rational).
• An attacker’s participation can be either limited with no access to system (outsider) or the attacker can be a member of the system (insider).

• An attacker’s strategy of attack can be either to transmit fake signals or packets (active) or to silently monitor the activities on the channel (passive).

• An attacker’s mobility state can either be static or mobile. Mobile attackers can be either moving uniformly or with varying velocities.

• Scope of an attacker can be limited to a part of VANET (local) or the attacker can be affecting various parts of the VANET (extended).

Further, we discuss the most important attacks for which realizing a solutions seems necessary before real implementations of VANETs are feasible. We classify all attacks according to their goal, participation, strategy, mobility and scope.

3.1 Flooding DoS Attack
Attackers can send false messages that consume the resources of legitimate vehicles in verifying those messages. However, they will be discarded after verification although they consume time and effort for the verification of the messages. DoS attacks have malicious goal, insider or outsider participation, active strategy, static or mobile, and local or extended scope.

3.2 Replay Attack
Replaying of valid messages can be created by storing broadcast messages at time $t_i$ and releasing them at time $t_{i+1}$. This attack is highly effective with mobile attackers, such that they can accumulate messages from one physical location and distribute them in another physical location, where the messages do not hold any importance. However, the vehicles receiving the messages need to authenticate them and consume their time and resources. Replay attacks have malicious or rational goal, insider membership, active strategy, mobile, and extended scope.

3.3 False Information Attack
This attack is performed by transmitting false information on the wireless channel. For instance, in Figure 1, an attacker may broadcast false information about traffic jam on Wales Street to all the vehicles crossing the intersection. All vehicles receiving this message will try to re-route towards 144th street, causing situations such as traffic jams on 144th street. False information attacks have rational goal, insider membership, active strategy, static or mobile, and local or extended scope.
3.4 Identity Attack
In this attack, an attacker illegitimately obtains the identity of a genuine vehicle, \( V_i \), and sends out messages pretending to be \( V_i \). The attacker can obtain the identity of \( V_i \) from the messages that it broadcasts. This is because authentication schemes require the source of the message to append its ID to the broadcasted packet. Identity attacks have rational goal, insider membership, active strategy, static or mobile, and local or extended scope.

3.5 Forgery Attack
In these attacks, the attacker modifies the content of a legitimate message. This attack differs from false information where the complete message was formed by the attacker. However, in this attack the attacker only changes a few bits or the whole content of the message, so that valid information does not reach affected vehicles. Forgery attacks have rational goal, insider membership, active strategy, static or mobile, and local or extended scope.

3.6 Location Traceability Attack
Obtaining a vehicle \( V_i \) physical location is possible for an attacker by collecting a sequence of messages received from \( V_i \). Signal strength or other parameters can be used by the attacker for localizing the position of \( V_i \), if it is not already added to the message. Location traceability attacks have rational goal, insider or outsider membership, passive strategy, mobile, and extended scope.

4 REALISTIC VANET SCENARIO
Because it is not advisable to perform initial tests on a real implementation, where human life is involved, we chose to perform simulations. Our simulations setup imitates actual VANET scenarios due to the following reasons. First, we consider a simulator that can model wireless network applications and protocols. Second, our requirement is to obtain realistic mobility traces from a vehicle mobility model which can reproduce an urban environment. Third, we use a radio propagation model that accommodates the fading characteristics of the wireless channel. We further look into each of the requirements and make appropriate selections.

4.1 Network Simulator for Wireless Access in Vehicular Environments
To fulfil the first requirement, we selected an event-based network simulator, ns-2 [27]. Ns-2 includes most of the IEEE 802.11 standards, including IEEE
802.11p and the standard to add wireless access in vehicular environments (WAVE) [10]. The parameter values used for IEEE 802.11p PHY and MAC layers are given in Table 2.

<table>
<thead>
<tr>
<th>PHY/MAC LAYER</th>
<th>PARAMETER</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHY</td>
<td>Frequency</td>
<td>5.9 GHz</td>
</tr>
<tr>
<td>PHY</td>
<td>CSThresh</td>
<td>-95 dBm</td>
</tr>
<tr>
<td>PHY</td>
<td>NoiseFloor</td>
<td>-99 dBm</td>
</tr>
<tr>
<td>PHY</td>
<td>PowerMonitorThresh</td>
<td>-102 dBm</td>
</tr>
<tr>
<td>PHY</td>
<td>HeaderDuration</td>
<td>40 $\mu$s</td>
</tr>
<tr>
<td>PHY</td>
<td>PreambleCaptureSwitch</td>
<td>1</td>
</tr>
<tr>
<td>PHY</td>
<td>DataCaptureSwitch</td>
<td>1</td>
</tr>
<tr>
<td>PHY</td>
<td>BasicModulationScheme</td>
<td>BPSK</td>
</tr>
<tr>
<td>MAC</td>
<td>CWMin</td>
<td>1.5</td>
</tr>
<tr>
<td>MAC</td>
<td>CWMax</td>
<td>1023</td>
</tr>
<tr>
<td>MAC</td>
<td>SlotTime</td>
<td>13$\mu$s</td>
</tr>
<tr>
<td>MAC</td>
<td>SIFS</td>
<td>32$\mu$s</td>
</tr>
<tr>
<td>MAC</td>
<td>SymbolDuration</td>
<td>8$\mu$s</td>
</tr>
<tr>
<td>MAC</td>
<td>PLCPDataRate</td>
<td>6 Mbps</td>
</tr>
<tr>
<td>MAC</td>
<td>basicRate</td>
<td>6 Mbps</td>
</tr>
<tr>
<td>MAC</td>
<td>dataRate</td>
<td>6 Mbps</td>
</tr>
</tbody>
</table>

4.2 Mobility Model for Urban Environment

To obtain realistic mobility traces, firstly, we investigate the requirements for a realistic vehicular mobility model for an urban setting [15, 26]. Further, we examine the various urban mobility simulators [25]. In our thorough analysis, we found Simulation of Urban MOBility (SUMO), a continuously developing mobility simulator. SUMO satisfies the minimum requirements for an urban setting along with other additional ones. We highlight some features of
SUMO, used for our simulations, in Table 3. Therefore, we chose SUMO as our mobility trace generator.

### Table 3

**SUMO Features**

<table>
<thead>
<tr>
<th>SUMO Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Car Following Model</td>
<td>Krauss Model [22]</td>
</tr>
<tr>
<td>Intersection Management</td>
<td>Stochastic Turns *</td>
</tr>
<tr>
<td>Traffic Lights</td>
<td>Yes</td>
</tr>
<tr>
<td>Stop Signs</td>
<td>Yes</td>
</tr>
<tr>
<td>Speed Limitations</td>
<td>Yes</td>
</tr>
<tr>
<td>Velocity</td>
<td>Smooth †</td>
</tr>
<tr>
<td>Multilane roads</td>
<td>Yes</td>
</tr>
<tr>
<td>Lane Changing</td>
<td>Yes</td>
</tr>
<tr>
<td>Trip Generation</td>
<td>Random or Activity-based</td>
</tr>
<tr>
<td>Path Computation</td>
<td>Dijkstra</td>
</tr>
<tr>
<td>Collision free movement</td>
<td>Yes</td>
</tr>
<tr>
<td>Real Map support</td>
<td>Yes</td>
</tr>
<tr>
<td>Building Platform</td>
<td>C++</td>
</tr>
<tr>
<td>Portable</td>
<td>Yes</td>
</tr>
<tr>
<td>Ease of Installation</td>
<td>Moderate</td>
</tr>
<tr>
<td>Ease of Use</td>
<td>Hard</td>
</tr>
<tr>
<td>Visualization Tool</td>
<td>Yes</td>
</tr>
<tr>
<td>Generate traces for ns-2</td>
<td>Yes</td>
</tr>
</tbody>
</table>

SUMO, being an open source, microscopic and continuous road traffic simulator, can handle large road networks [20]. SUMO mobility model has been validated by [21]. We used SUMO to generate similar vehicles, with 5 m length, such that all vehicles maintain a minimum gap of 2.5 m between them. We set the maximum speed of vehicles to reach 20 m/s ($\approx 45$ miles/hr), similar to speed limit on county paved roads in the state of Alabama. The highest

* Vehicle chooses own speed and direction according to a probability density function when no path is previously defined

† Vehicle does not abruptly brake or accelerate
acceleration and deceleration ability of the vehicles is allowed to be 0.8 m/s\(^2\) and 4.5 m/s\(^2\), respectively. We also take care of the driver imperfection to adhere to the rules to be 50%.

4.3 Radio Propagation Model
To find a suitable radio propagation model, we consider two requirements: i) accommodation of fading characteristics of the wireless channel, and ii) compliance with ns-2. A frequently used propagation model in ns-2 simulations is Two Ray Ground model. However, it is a deterministic model which delivers same signal strength at all distances within range. On the other hand, the Nakagami model is a probabilistic propagation model.

[28] and [38] have demonstrated that the Nakagami model has a gradually fading property, such that the packet delivery ratio decreases as the distance between the sender and receiver increases. Nakagami probability distribution function has been defined by [10] as follows.

\[
f(x) = \frac{2m^m x^{2m-1}}{\Gamma(m)\Omega^m} e^{\frac{-mx^2}{\Omega}} \quad x > 0, \Omega > 0, m \geq 1/2,
\]

where \(m\) and \(\Omega\) are functions of distance. \(m\) is the fading parameter, such that increasing values of \(m\) decreases fading. \(\Omega\) is interpreted as the average received power. Due to its probabilistic nature, the Nakagami model seems to be an appropriate choice for a realistic scenario where it is required to have signal fading with the increase in distance between sender and receiver. For an urban area, we take the Nakagami parameter values from the Vehicle Safety Communications (VSC) project tasks [12].

5 VANET BROADCAST AUTHENTICATION
In VANETS, information is typically broadcasted, either in the form of WAVE Service advertisement (WSA) or Basic Safety Message (BSM). Through WSA, a WAVE device advertises the service that it offers to all other WAVE devices in vicinity, detailed in IEEE 1609.3 [3]. Usually RSUs broadcast WSA, but service offering OBUs can also do so [19]. Besides, BSM announces the status of a vehicle. All DSRC equipped OBUs transmit BSM periodically and at regular intervals. Security of WSA and BSM communication is maintained
by IEEE 1609.2. In this paper, since we focus on VANETs in the early stages of development which will not have strong RSU backbone, we will concentrate on BSM.

BSM are used for transmitting warnings or traffic related information, in addition to vehicles’ status, including vehicle’s location and velocity. The advantages of using BSM are: i) interoperability of vehicle safety applications, without standardization of the applications, ii) various vehicle safety applications that can be performed with the same message, and iii) backward-compatible with future developments due to flexible expansion of messages [2]. These safety messages are broadcasted by vehicles every 100 – 300 milliseconds [1]. BSM will generally be encapsulated in a WAVE short message (WSM) data [19]. WSM follow WAVE Short Message Protocol (WSMP), which allows a rapid transmission of messages in an environment where radio frequency varies rapidly.

Below, we look into the security, mainly authentication, of these safety messages. Since these messages are broadcasted, we first study some of the properties essential for broadcast authentication in VANETs and then examine their effects on IEEE 1609.2. Finally, we will implement and study IEEE 1609.2 on realistic VANETs and study the problem of long verification delay.

5.1 Broadcast Authentication Properties for VANETs

In view of the fact that VANETs have dynamically changing topologies and network densities, we consider the following nine properties as important characteristics of broadcast authentication techniques for VANETs.

1. Source Authentication: It basically suggests verifying the source that originates the packet.
2. Data Integrity: It confirms that the message has not been modified while in transit from the sender to the receiver.
3. Robustness to Packet Loss: It ensures that the loss of any packet will suspend/terminate the authentication process.
4. Non-Repudiation: It confirms that a source cannot deny that it has originated a message.
5. Immediate Authentication: It means that there is no delay in authenticating a received message due to the authentication process.
6. No Time Synchronization: The sender and receivers should not be required to time synchronize with each other.
7. Low Computation Overhead: Authentication should not consume exceptionally large amount of time and resources of the vehicle.

8. No Buffering Overhead: The authentication process should not require buffering of data or parameters.

9. Scalability: It should be easy to add and remove vehicles in VANET without adversely affecting the authentication process.

5.2 IEEE 1609.2 Security Standard

IEEE 1609.2 supports the cryptographic standard ECDSA [2]. ECDSA is a variant of the Digital Signature Algorithm (DSA) which uses elliptic curve cryptography. The standard defines generation and verification algorithms for digital signature using elliptic curve domain parameters. It is a Public Key infrastructure (PKI) scheme. The key pair formed is a private key, $d$, and a public key, $Q$. Elliptic curve, $E$, is defined for a finite arithmetic field $F_p$, such that $p$ is required to be an odd prime. With that, a base point $B$ is selected on the curve, $B \in E(F_p)$ of order $n$, where $n$ is a large prime.

ECDSA signature generation process consists of the following stages:

1. Per-message secret number generation, $k$ in the interval $[1, n - 1]$
2. Hash of message, $h(m)$, where $l$ are the left most bits of $h(m)$
3. Calculation of the curve point $(x, y) = k \cdot B$
4. Formation of signature $(r, s)$ as: $r = x \mod n$ and $s = k^{-1}(l + r \cdot d)$
5. If $s = 0$, goto Step 1

ECDSA signature verification process consists of the following stages:

1. Verify $r$ and $s$ are integers in the interval $[1, n - 1]$
2. Compute $w = s^{-1} \mod n$
3. Compute $u_1 = l \cdot w \mod n$ and $u_2 = r \cdot w \mod n$
4. Compute $u_1 \cdot B + u_2 \cdot Q = (x, y)$ and $v = x \mod n$
5. Signature is verified successfully if $v = r$

ECDSA is highly secure because finding the discrete logarithm of a random elliptic curve element with respect to a publicly known base point is impractical [18].
Simulating Broadcast Authentication in VANETs

For simulating IEEE 1609.2, we use a realistic VANET scenario as described in Section 4. Following the development of the scenario, we implement broadcast using the DSRC standard [1], with each broadcast valid for a given time duration. These messages contain the vehicle’s status and other traffic related information. Table 4 lists our basic simulation parameters. Using IEEE 1609.2, we authenticate each message with ECDSA signatures, such that the broadcasting vehicles authenticate each message with an ECDSA signature and each receiver verifies the message using ECDSA verification. In ECDSA, verification is on a first come first serve basis. Due to this, messages that wait in a queue for a long time to be verified are timed out.

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Urban Area</td>
<td>1000 m X 1000 m</td>
</tr>
<tr>
<td>Mobility Model</td>
<td>SUMO</td>
</tr>
<tr>
<td>Radio Propagation Model</td>
<td>Nakagami</td>
</tr>
<tr>
<td>WAVE Standard</td>
<td>IEEE 802.11p</td>
</tr>
<tr>
<td>Simulation Duration</td>
<td>250 seconds</td>
</tr>
<tr>
<td>Broadcast Interval</td>
<td>100 – 300 millisecond</td>
</tr>
<tr>
<td>Broadcast Time-To-Live</td>
<td>1000 milliseconds</td>
</tr>
</tbody>
</table>

To explain the problem of long verification delay, consider an example here. Suppose vehicle V_i is receiving messages from 10 neighbours, at an interval of 100ms. Then, in 1 second, the total number of messages received by V_i will be 100. In addition, each message has a time-to-live of, say 1000ms. V_i adds all the received messages to a first-in first-out (FIFO) queue, and verifies them in succession. Since, the time required to verify each message is 22 ms, total time required for the verification of 100 messages is 2.2 seconds. Hence, 50% of the messages will be dropped because they will be expired when they are taken out of the queue to be verified. Below we simulate some scenarios to study this problem in highly dynamic VANETs with changing network densities and dynamic topologies.

Firstly, we simulated IEEE 1609.2 with change in the network density
(number of OBUs) from 20 to 200, out of which only 50% of the vehicles are able to generate broadcasts to imitate the initial deployment phases. Yet, all the vehicles can receive and verify messages. Each received message is added to the FIFO queue, followed by a one-by-one verification. We call this Scenario 1.

Figure 2 shows the packet loss ratio of IEEE 1609.2 in Scenario 1. We notice that as we decrease the broadcast interval from 300 ms to 100 ms, the packet loss ratio increases. However, the major factor affecting the packet loss interval immensely is the network density. With 300 ms broadcast interval, packet loss ratio shows significant increase as the number of OBUs is increased to greater than 120 in the given area. However, with 200 ms and 100 ms broadcast interval, packet loss ratio becomes prominent as the number of OBUs is increased above 80 and 40, respectively. The highest value of packet loss ratio obtained in our scenario with 200 OBUs is 0.15 for 300 ms interval, 0.35 for 200 ms interval and 0.54 for 100 ms interval.

Further, we study the effect of change in number of broadcasting OBUs within the transmission range of a single receiving OBU. We vary the number of OBUs in range from 5 to 40. We perform the simulation with broadcast
intervals of 100, 200 and 300 ms. We call this Scenario 2. For this scenario, we evaluate the percentage of packets processed, i.e., the packets which were taken out from the verification queue before expiration and reached the processing phase or the probability of verification phase. IEEE 1609.2 endorses verification probability to be equal to 1.

Packet processed ratio for Scenario 2 is given in Figure 3. For 100 ms broadcast interval, packet processed ratio decreases when in-range OBUs become $>5$ such that it becomes 0.13 for 40 OBUs in-range. While for 200 ms interval, the ratio decreases after in-range OBUs become $>10$. For $15-40$ in-range OBUs, the packet processed ratio for 200 ms interval is double the value experienced at 100 ms interval. As we increase the broadcast interval further to 300 ms, the ratio rises for all in-range OBUs ending at 0.41 for 40 in-range OBUs. The packets which do time out before reaching the processing stage, are dropped. The reason is again verification delay.

All the above results indicate that IEEE 1609.2 has an important issue with vast number of packets being dropped due to delay in the verification queue. Therefore, it provides us the motivation to develop an efficient authentication scheme for VANETs.
6 DESIGN GOALS AND PROPOSED SOLUTION

As seen in the implementation of IEEE 1609.2, a large number of packets are dropped due to verification delay. Also, the packet process rate decreases as the broadcasting OBUs in-range are increased. It is certain that this scheme cannot be used in practice without modification. We also agree that ECDSA is a very secure algorithm. Therefore, after considering all these aspects and after elaborating on the design goals below, we describe the design of our scheme that is based on a probabilistic ECDSA signature verification method.

6.1 Design Goals

Analysis of the issues existing in IEEE 1609.2 leads us to design a smart probabilistic verification strategy that have the maximum security while minimizing the packet loss ratio. We maximize the packet processing ratio by verifying the most relevant messages with higher probability. The most relevant messages are the ones received from a relatively closer sender in terms of distance such that the sender is moving in a relatively neighbouring direction to the receiver.

Another important factor in designing an authentication strategy for vehicular ad hoc networks is that it should not reveal the original identity of the broadcasting OBU. This can be maintained with the use of pseudonyms. Each broadcasting OBU can append the pseudonym to the message instead of the original identity. However, the method must allow the original identity to be recovered by the Government, County, licensing or any other authority. Therefore, we use a registration mechanism where the original identities of the OBUs are stored with a regional registering authority (RRA).

6.2 Proposed Solution

To meet the design goals described above, we propose a four-phase scheme called probabilistic ECDSA signature verification (PESV). The first two phases are for V2I authentication, comprising of a preprocessing phase where the key initialization of RRA and OBU takes place, followed by a vehicle registration providing mutual authentication between the RRA and OBU including a pseudonym generation at each OBU. The next two phases are for V2V authentication, comprising of ECDSA signature generation and ECDSA probabilistic signature verification. All the four phases are discussed in detail below. Regularly occurring notation are described in Table 5.

Preprocessing

In this phase, a Trusted Authority (TA) assigns parameters to RRA and OBU. These parameters will act as initialization keys in the following phase of mu-
<table>
<thead>
<tr>
<th>NOTATION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>TA</td>
<td>Trusted Authority</td>
</tr>
<tr>
<td>RRA</td>
<td>Regional Registering Authority</td>
</tr>
<tr>
<td>OBU&lt;sup&gt;i&lt;/sup&gt;</td>
<td>Wireless On-Board Unit of i&lt;sup&gt;th&lt;/sup&gt; vehicle</td>
</tr>
<tr>
<td>ID&lt;sub&gt;OBU&lt;/sub&gt;&lt;sup&gt;i&lt;/sup&gt;</td>
<td>Original identifier of i&lt;sup&gt;th&lt;/sup&gt; OBU</td>
</tr>
<tr>
<td>ID&lt;sub&gt;TA&lt;/sub&gt;</td>
<td>Original identifier of TA</td>
</tr>
<tr>
<td>&lt;sup&gt;x&lt;/sup&gt;&lt;sup&gt;i&lt;/sup&gt;</td>
<td>secret generated by TA for i&lt;sup&gt;th&lt;/sup&gt; OBU</td>
</tr>
<tr>
<td>AID&lt;sub&gt;OBU&lt;/sub&gt;&lt;sup&gt;i&lt;/sup&gt;</td>
<td>Alias of i&lt;sup&gt;th&lt;/sup&gt; OBU for mutual authentication with RRA</td>
</tr>
<tr>
<td>R&lt;sub&gt;OBU&lt;/sub&gt;&lt;sup&gt;i&lt;/sup&gt;</td>
<td>Random Number of i&lt;sup&gt;th&lt;/sup&gt; OBU</td>
</tr>
<tr>
<td>R&lt;sub&gt;RA&lt;/sub&gt;&lt;sup&gt;i&lt;/sup&gt;</td>
<td>Random Number of RRA for i&lt;sup&gt;th&lt;/sup&gt; OBU</td>
</tr>
<tr>
<td>P&lt;sub&gt;OBU&lt;/sub&gt;&lt;sup&gt;i&lt;/sup&gt;</td>
<td>Pseudonym of i&lt;sup&gt;th&lt;/sup&gt; OBU for V2V communication</td>
</tr>
<tr>
<td>x, y</td>
<td>location coordinates of OBU</td>
</tr>
<tr>
<td>V</td>
<td>Velocity of OBU</td>
</tr>
<tr>
<td>θ</td>
<td>Direction of motion of vehicle</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. RRA is issued two system parameters; hash function, $h()$, and secret encryption key, $K$.

2. OBU is issued the following system parameters; $h()$, $p^i = h(x^i)$, $ac^i = h(x^i||ID_{TA})$, $seal^i = E_K(x^i||ID_{TA})$ and $secret^i = h(x^i||ID_{OBU})$. These parameters are generated for each OBU separately, such that $x^i$ is random number corresponding to $OBU^i$ and $ID_{TA}$ is the original identity of TA. At the same time, each OBU is loaded with an ECDSA public-private key pair and a public key certificate.

Vehicle Registration
Using the parameters assigned by TA, each vehicle’s OBU registers itself with an available RRA. This process also includes a mutual authentication between RRA and OBU. This V2I communication is required to provide conditional privacy, such that the RRA can identify the original ID of OBU whenever required. All RRAs maintain a database for the OBUs in their region. Inspired by the work [11], we use symmetric cryptography for this process.

The complete process involves the following sequence of steps, as shown in Figure 4. As a precaution to identity thefts, each $OBU^i$ interacts with RRA using an alias, $AID_{OBU}^i$. The formation of the alias consists of selecting a random number $R_{OBU}^i$, concatenating it with a system parameter, $p^i$, followed by XORing it with the original ID of OBU. The OBU further creates a code, $code^i$, for the authentication purpose and encrypts it with random number for transmission purposes, $stamp^i$. Both the creation of code and encryption make use of system parameters, $secret^i$ and $ac^i$ respectively. OBU then sends the encrypted code along with its seal and alias, to the available RRA ($\{stamp^i, seal^i, AID_{OBU}^i\}$).

When RRA receives an authentication message from an OBU, it follows a sequence of steps for authenticating $code^i$. Firstly, it decrypts $seal^i$ using secret encryption key, $K$. Values $ID_{TA}$ and $x^i$ are obtained from the decryption. Then RRA computes the symmetric encryption-decryption key, $ac^i$, which is further used to recover $code^i$ and $R_{OBU}^i$. Original ID of OBU is retrieved by XORing alias, $AID_{OBU}^i$, with the hash of system parameter, $p^i$, and random number, $R_{OBU}^i$.

At this step, RRA performs a check by matching $ID_{OBU}^i$ with its Certificate Revocation List (CRL). If a match is found, it exits the authentication process and informs the OBU that it will not take any further requests from
**OBU**

Generate Random Number $R_{OBU}^i$

Compute Alias

$AID_{OBU}^i = h(y^i \| R_{OBU}^i) \oplus ID_{OBU}$

Compute code for authentication

$code^i = h(secret^i \| R_{OBU}^i)$

Compute stamp for encrypting code

$stamp^i = E_{ac}^i (code^i \| R_{OBU}^i)$

**RRA**

Decrypt seal

$D_k (seal^i) \rightarrow (x^i \| ID_{RA})$

Compute encryption key of stamp

$ac^i = h(x^i \| ID_{RA})$

Decrypt stamp

$D_{ac} (stamp^i) \rightarrow (code^i \| R_{OBU}^i)$

Compute original OBU Identifier

$ID_{OBU}^i = k(h(x^i) \| R_{OBU}^i) \oplus AID_{OBU}^i$

Check Certificate Revocation List (CRL)

$ID_{OBU}^i? = CRL (ID_{OBU})$

If Match Found

Inform OBU

Exit Authentication

Else

Compute secret

$secret^i = h(x^i \| ID_{OBU}^i)$

Compute code

$code^i = h(secret^i \| R_{OBU}^i)$

Generate Random Number $R_{KA}^i$

Generate key

$c^i = h(R_{KA}^i \| R_{OBU}^i)$

Generate encrypted data

$data^i = E_{ac}^i (R_{KA}^i \| R_{OBU}^i)$

Decryption of data

$D_k (data^i) \rightarrow (R_{KA}^i \| R_{OBU}^i)$

Computes key

$c^i = h(R_{KA}^i \| R_{OBU}^i)$

$E_{c^i} (R_{KA}^i)$

Decrypts $E_{c^i} (R_{KA}^i)$ and Verifies $R_{KA}^i$

---

**FIGURE 4**

Vehicle Registration and Mutual Authentication between RRA and OBU
it. Otherwise, RRA continues processing further and and computes secret\(_i\) and code\(_i\) values. This completes the OBU’s registration with the RRA. All code\(_i\) are stored in its database along with their original identifiers ID\(_i\). RRA then advances to the creation of a random number \(R_{RA}^i\) for OBU\(_i\), and a key, \(c^i\). Both the random numbers, \(R_{RA}^i\) and \(R_{OBU}^i\), are sent to OBU in an encrypted form.

OBU decrypts the received cryptograph to discover \(R_{RA}^i\) and computes the value of key \(c^i\). For the final confirmation, OBU sends the RA’s random number back to it, encrypted with the key, \(c^i\). RA solves the encryption and verifies the random number to complete the mutual authentication process successfully.

Each OBU then creates a pseudonym, \(P_{OBU}^i\), at its own end for communicating with the other OBUs in the network using the values code\(_i\) and ID\(_i\). Known to RRA.

\[
P_{OBU}^i = \text{code}^i \oplus \text{ID}_{OBU}^i
\]

This allows the RRA to have control over obtaining the original identity of the OBU. At the same time, no third party can track or access the broadcasting OBUs real identity.

**Signature Generation**

According to the DSRC standard, each periodic BSM broadcast transmission includes the physical location coordinates of the vehicle, on which the OBU is installed, at the time of generation of the message, velocity of the vehicle on which the OBU is installed, and warning information. We append the OBU’s pseudonym and timestamp, \(ts\) (the time at which message was generated).

\[
\text{message} = \{x_i, y_i, V_i, \text{info}, P_{OBU}^i, ts\}
\]

Creation of the message is followed by the signature generation. We use ECDSA signature generation procedure as described in Section 5.2. Each signature generation take 4 ms on a 400 MHz processor [16]. The public key certificates accompany the message and signature in the broadcast packet.

**Signature Verification**

Signature verification is done using a probabilistic strategy. We carefully consider the method for computing the probability of verification. The factors used to estimate probability are whether the communicating vehicles are coming close or moving farther away from each other, and relative direction of motion of the communicating vehicles. For predicting the direction of motion of vehicles, we develop a mobility prediction model based on past
track history. In addition, the difference in distance between the vehicles at a present and in a future time will determine if the vehicles are coming closer or moving away. Moreover, the relative direction of motion between sender and receiver is computed using the coordinates contained in the BSM broadcast. Hence, we carry no additional information with the broadcast for our computations. Below, we describe our procedure in four sequential steps.

**Step 1: Mobility Prediction**

Here, we define two methods for computing a vehicle’s direction or mobility. The first method uses past track history. We accomplish this task by logging the physical coordinates of each vehicle in its database after a regular interval. The interval found to be most appropriate is 500 milliseconds.

Using the history of past locations, we compute the direction of motion, such that $\theta_{t(i)}$ is the direction of motion of the vehicle, $x_{t(i)}$, $y_{t(i)}$ are the position coordinates of the vehicle at time $t_i$, and $x_{t(i-1)}$, $y_{t(i-1)}$ are the position coordinates of the vehicle at time $t_{i-1}$.

$$\theta_{t(i)} = \tan^{-1}\left(\frac{y_{t(i)} - y_{t(i-1)}}{x_{t(i)} - x_{t(i-1)}}\right)$$

However, at the beginning, vehicles do not have a past track history and cannot use the above method for computation of direction of motion. Therefore, we suggest the use of a second technique using the velocity components $(V_x, V_y)$ because the vehicle is aware of its present velocity, $V$, as given below. The components are defined as $V_x = V \cos \theta$ and $V_y = V \sin \theta$. This leads to our computation of the direction of motion as follows:

$$\theta_{t(i)} = \tan^{-1}\left(\frac{V_y}{V_x}\right)$$

This method is of temporary usage for the initial stages. This is because the vehicle may experience various stops in the later stages such as red lights or due to the car following model. Once, the vehicle starts logging, it can effectively use its past track history for the computations.

**Step 2: Estimation of future coordinates**

Once the direction of motion, $\theta$, of the vehicle is obtained, from the above step, we estimate the future coordinates using the following equations:

$$x_{t(i+1)} = x_{t(i)} + V \times \cos \theta \times \Delta t$$
$$y_{t(i+1)} = y_{t(i)} + V \times \sin \theta \times \Delta t$$
where \( x_t(i), y_t(i) \) are coordinates of the vehicle at time \( t_i \), and \( x_{t(i+1)}, y_{t(i+1)} \) are physical location coordinates of the vehicle at time \( t_{i+1} \). Time interval after which the future coordinates are estimated is given by \( \Delta t \), such that \( \Delta t = t_{i+1} - t_i \). For our simulations, we take \( \Delta t \) as 1000 milliseconds.

**Step 3: Distance Computation**

We intend to know if the vehicles will be coming close to each other or will go farther from each other in future. To perceive this, we find the difference between the distance of communicating vehicles at present and in future. Distance between communicating vehicles is computed using the formula:

\[
\text{Distance}_{t(i)} = \sqrt{(x_{t(i)} - x_{t(i)})^2 + (y_{t(i)} - y_{t(i)})^2}
\]

where \( x_{t(i)}, y_{t(i)} \) are the position coordinates of the receiver at time \( t_i \), and \( x_{t(i)}, y_{t(i)} \) are the position coordinates of the sender at time \( t_i \).

\[
\text{Distance}_{t(i+1)} = \sqrt{(x_{t(i+1)} - x_{t(i+1)})^2 + (y_{t(i+1)} - y_{t(i+1)})^2}
\]

where \( x_{t(i+1)}, y_{t(i+1)} \) are the physical location coordinates of the receiver at time \( t_{i+1} \), and \( x_{t(i+1)}, y_{t(i+1)} \) are the physical location coordinates of the sender at time \( t_{i+1} \). The quantity that is important for our calculations is the difference between these two given distances.

**Step 4: Probabilistic Verification**

Our most important consideration in probability computation is whether the vehicles are coming close or moving farther away from each other, this factor is statically computed by the difference in present and future distances between the two communicating vehicles. We base our probability on this factor of distance. To model the behaviour of our probabilistic distribution associated with the independent outcomes of the difference in distance, we use exponential distribution. We also use a scaling factor based on the relative direction of motion of communicating vehicles. The complete set of equations and explanation for probability computation is given below.

If \( \text{Distance}_{t(i+1)} - \text{Distance}_{t(i)} < 0 \) //Vehicles are coming close

\[
P_v = \max(f_1(\theta), f_2(d), f_2(d)) \quad \text{such that} \quad f_1(\theta) > 0
\]

where \( f_1(\theta) = \frac{|\theta_{t(i)} - \theta_{t(i)}|}{180} \), \( f_2(d) = 1 - e^{(\text{Distance}_{t(i+1)} - \text{Distance}_{t(i)})/d} \)
\[ \text{If } \text{Distance}_{t(i+1)} - \text{Distance}_{t(i)} > 0 \quad \text{//Vehicles are moving away} \]

\[ P_v = \min(f_1'(\theta).f_2(d), f_2'(d)) \quad \text{such that } f_1'(\theta) > 0 \]

where \( f_1'(\theta) = 1 - \frac{|\theta_{rt(i)} - \theta_{st(i)}|}{180} \), \( f_2'(d) = e^{-(\text{Distance}_{t(i+1)} - \text{Distance}_{t(i)})} \)

where \( P_v \) is the probability of verification, \( f_1(\theta) \) and \( f_1'(\theta) \) are the functions of direction of motion, \( f_2(d) \) and \( f_2'(d) \) are the functions of distance, \( \theta_{rt(i)} \) is the direction of motion of receiver at time \( t(i) \), and \( \theta_{st(i)} \) is the direction of motion of sender at time \( t(i) \).

If the vehicles are coming closer, we use a max formula such that the function of distance, \( f_2(d) \), is an exponential cumulative distribution function, where the rate of change is a continuous percent increase. On the other hand, if the vehicles are moving away from each other, we use a min formula such that the function of distance, \( f_2'(d) \), is an exponential distribution, with \( f_2'(d) \) being the inverse of \( f_2(d) \). It is also called the tail of the cumulative distribution function [34]. It takes the form of an exponentially decreasing probability density function, where the rate of change is a continuous percent decrease. In both cases, function of \( \theta \) acts as a scaling factor, unless it is zero, such that \( P_v \in [0, 1] \).

Our main aim to use probabilistic verification is to speed up the verification of most relevant packets. We will implement our algorithm on the same scenarios as we implemented IEEE 1609.2 (Section 5.2). Further, we will compare the results and analyse the improvement.

7 RESULTS AND DISCUSSION

The above proposed strategy is simulated in the same realistic VANET scenario as used for the implementation for IEEE 1609.2. In the first subsection, we will provide results obtained using PESV in comparison to the IEEE 1609.2. The second subsection will compare PESV with other existing schemes and will include security analysis.

7.1 Simulations and Results

Table 4 summarizes the simulation settings. We assume that all vehicles are equipped with GPS’s to record their location coordinates and are aware of their velocity. At first, we replicate Scenario 1 as used in the simulations of Section 5.2. Then, we do an analysis of the packet loss ratio for PESV...
in comparison to IEEE 1609.2. Figure 5 shows the comparison results. We observed, for 100 ms broadcast interval, the packet loss ratio for PESV decreases by 36–38% when the OBUs in the network are between 150–200. As we decrease the frequency of broadcasts, we obtain a decrease in the packet loss ratio by 16 – 33% when messages are sent at an interval of 200 – 300 ms, such that the number of OBUs is 200.

Next, we use Scenario 2 for the computation of packet processed ratio as the number of OBUs in the transmission range of a single OBU is increased. Figure 6 displays our results. With 100 ms broadcast interval, comparing our algorithm with IEEE 1609.2, we observe that the increase in the packet processed ratio is higher when the number of in-range OBUs is between 10 – 20. Similarly for 200 and 300 ms interval, comparative increase in the packet processed ratio of our algorithm is more when in-range OBUs are between 20 – 30 and 30 – 40, respectively. Overall, the percentage increase in the packet processed ratio varies between 15 – 49%.

Finally, we present a validation of our mobility prediction model for estimating future coordinates in order to check the accuracy of our estimates. Though our aim is to build a practical and efficient authentication strategy and not a method for accurate location prediction, we do account for the small percentage of errors experienced. We record the estimated future coordinate values $x_{t(i+1)}, y_{t(i+1)}$ at time $t_i$. As we proceed with the simulations and reach a point where the time is $t_{i+1}$, we compare the estimated values with the GPS coordinates and record the error in estimation. Figure 7 shows our estimation errors in meters which varies between $1.4–2.6$ m. The cause of these errors may be traffic lights, vehicles ending their excursion or halting and the dependence of our mobility traces on the Krauss car following model.

7.2 Discussion

We compare the features of PESV with the three latest schemes (published in year 2013), in Table 6. We first list the protocol used for key agreement in V2I communication. Then, we compare the features for mutual authentication and Certificate Revocation List (CRL) check, in V2I. Further, we compare if the V2V authentication is independent or not. Finally, we discuss the adaptation of identity anonymity in both V2I as well as V2V communication.

Security Analysis

We have seen that PESV is more practical compared to IEEE 1609.2 and has the benefits of decreasing the packet loss ratio, while maximizing the percentage of packets processed. Then, we elaborate on the features that PESV
FIGURE 5
Packet Loss caused by Verification Queue delay
FIGURE 6
Packet Processed Ratio
### TABLE 6
Feature Comparison with other schemes

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Key Agreement Protocol (V2I)</th>
<th>Mutual Authentication (V2I)</th>
<th>CRL Check (V2I)</th>
<th>Identity Anonymity (V2I)</th>
<th>Independent Authentication (V2V)</th>
<th>Identity Anonymity (V2V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPAS [17]</td>
<td>Modified Diffie Hellman</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Privacy Preserving [37]</td>
<td>Bilinear pairing</td>
<td>N</td>
<td>N</td>
<td>-</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>PESV</td>
<td>Symmetric key</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>
possesses in comparison to the latest proposed schemes. Further, we analyse the security of PESV regarding its capability to prevents the most important attacks in VANETs as described in Section 3.

**Flooding DoS Attack**
This attack can be controlled to some extent due to our smart strategy for assigning probability to messages from multiple senders to indicate different level of relevance depending on their distances, whether the communicating vehicles are coming closer or moving away, and direction of motion from the sender. The probabilistic nature of PESV is able to adjust to flooding DoS attacks.

**Replay Attack**
To prevent replay attacks, we append a timestamp ($ts_\text{r}$) to each broadcast message sent by an OBU in order to keep receivers from being affected by the replay attack. PESV makes sure that each message is verified if the time to live of the message has not exceeded 1000 milliseconds from the time of generation ($ts_\text{g}$). In addition, our vehicle registration process also prevents replay attacks because of the use of random numbers in each authentication.
request sent by an OBU.

**False Information Attack**

False entities do not possess valid security credentials and hence cannot form a legitimate signature. Therefore, false information messages will be discarded by the receiving vehicle’s immediately upon verification. PESV makes sure that each OBU obtains its initial parameters from the TA and then uses them for registration with the RRA. The RRA registration can only succeed if the OBU has the secret, $x^i$ and $ID_{TA}$, encrypted using the secret key $K$, such that the RRA can re-compute it. This is because an OBU will not be aware of the secret key $K$, since it is only shared between the TA and RRA.

**Identity Attack**

PESV makes sure that identity anonymity is maintained in both kinds of communications, whether it is a V2I communication for the vehicle registration process or a V2V communication for the broadcast of vehicle status and traffic information. We make use of alias for vehicle registration formed from in such a way that it combines the hash of the secret assigned to the OBU by the TA, $p^i = h(x^i)$, and a random number, $R^i_{OBU}$, generated by the OBU. For broadcast purposes, the pseudonym is created using the code authenticated by an RRA, $code^i$, composed of the $secret^i$ received from OBU and a random number, $R^i_{OBU}$. This forbids the third party to access the original identity of the OBU. However, an RRA can do so for liability purposes.

**Forgery Attack**

As shown in Figure 1, an attacker will be interested in modifying relevant information to create an erroneous message. The attacker will succeed in doing so in the absence of valid security credentials or if the cipher is easy to modify to produce the same results. Since, we make use of ECDSA signatures it is not possible for the attacker to modify the message without affecting the value of the signature. ECDSA signature $(r,s)$ is formed with some bits of the hash of the message, $s = k^{-1}(l + r * d)$, where $l$ are the left most bits of $h(m)$ (for details, see Section 5.2). In vehicle registration process, modification of the content means modifying $h(secret^i; ||; R^i_{OBU})$. This leads to a failure in authentication.

**Location Traceability**

Each authentication request and broadcast message sent out by the OBU, in PESV, does not contain its original identifier, $ID_{OBU}$. The creation of alias
or pseudonym makes use of the random numbers, $R_{OBU}^i$, and parameters assigned by TA, $p^i$, $secret^i$. The parameters assigned by TA are unique due to the use of a unique secret that the TA generates for each OBU, $x^i$. Hence, it is not possible for an adversary to obtain these values.

8 CONCLUSION

VANET will eventually become an important part of our lives because of the benefits of traffic safety that they offer. Actual implementations of VANETs are awaiting the finalizing of security primitives for practical scenarios that ensure communications are secure. In this paper, we propose an efficient and practical authentication mechanism that complies with IEEE 1609.2 in the usage of ECDSA signature generation. However, we modify the verification with the use of a smart probabilistic strategy that accounts for the relative movement and distances of the communicating nodes. We also maintain conditional privacy with the inclusion of a lightweight mutual authentication mechanism between the infrastructure and vehicles. Our algorithm obtains up to 49\% increase in packet processed ratio with varying number of broadcasting OBUs in range and a 16 – 38\% decrease in packet loss ratio occurring due to delay in the verification queue as the broadcast frequency is increased from 100 – 300 ms.

REFERENCES


