International Journal of Safety and Security Engineering
Vol. 16, No. 6, June, 2026, pp. 1169-1181
Journal homepage: http://iieta.org/journals/ijsse

A Timed Efficient Stream Loss-Tolerant Authentication Protocol for IoV Cybersecurity

Duaa Sh. Jassim, Mohammed H. Ahmed*

Department of Computer Science, College of Education, Mustansiriyah University, Baghdad 10011, Iraq

Corresponding Author Email: mohammedalbawi@uomustansiriyah.edu.iq

https://doi.org/10.18280/ijsse.160601

Received: 27 March 2026
Revised: 28 April 2026
Accepted: 30 May 2026
Available online: 30 June 2026

Keywords:
TESLA protocol, IoV environment, Ascon-AEAD algorithm, VANET, mutual authentication

ABSTRACT

Vehicular Ad Hoc Networks (VANETs) play a vital role in enabling Vehicle-to-Infrastructure (V2I) communication by exchanging safety messages and traffic information in smart transportation systems. But authentication, confidentiality, and minimal delay pose a major challenge because of the broadcast nature of wireless channels and the resource-constrained nature of the vehicular environment. This paper proposes a lightweight authentication and confidentiality protocol, MA_ETESLA, which extends the TESLA protocol with mutual authentication and uses the Ascon encryption algorithm for message confidentiality. We investigated the security features of the proposed protocol using the formal verification tool SCyther and its performance using the NS-3 simulator at various vehicle speeds. This showed that the proposed protocol was efficient in terms of computational requirements, communication overhead (99.75 bytes per message), delay and packet delivery ratio (PDR) (99%) in the tested scenarios. Taken together, the results show that the proposed protocol is a viable and efficient way to enable secure V2I communications.

1. INTRODUCTION

The Internet of Things (IoT) is a dynamic network infrastructure that allows physical devices, like sensors, actuators, embedded systems and basic devices, to capture, exchange and act on information about other vehicles and roadside infrastructure (V2I) without human interaction [1]. The Internet of Vehicles (IoV) is a vital part of the IoT that comprises a network of internet-enabled vehicles that can include a range of sensors that share data with other vehicles and roadside infrastructure (V2I) [2]. The IoV is one of the most promising technologies of this fast-paced technology age [3]. It has changed the entire vehicle market, both in terms of vehicle performance and connectivity [4]. Vehicle management and data sharing is leading vehicle communications, but also introduces major vulnerabilities to network and cyber security that could impact data security, privacy, integrity and availability, and then the safety of vehicle occupants [5]. IoV vehicles are exposed to many vulnerabilities associated with the internet. Hazardous and unsafe conditions expose vehicles to significant security risks during their interactions with infrastructure and cloud computing [6]. Verifying the identity and integrity of the broadcast source is a major challenge for networks, requiring assurance that the received data originates from an authorized source and has not been tampered with during transmission [7]. A single security vulnerability in the network can have serious consequences, necessitating the implementation of stringent security measures, such as encryption and regular security audits, to protect against unauthorized access and data breaches. Source verification is a top priority [8]. A range of specialized security protocols are used to enhance mutual authentication and secure data transmission between vehicles and road infrastructure (RSU) within the network. A common mechanism is the Time-Tolerant Effective Data Authentication Protocol (TESLA), an effective authentication protocol used in resource-limited and time-constrained broadcasts, such as the IoV [9].

While TESLA provides strong source authentication and ensures data integrity in broadcast communications, it does not guarantee data confidentiality, as its core design focuses solely on authentication, not encryption. Messages transmitted over these networks are vulnerable to eavesdropping or data leaks, posing a threat to vehicle systems where personal information is exchanged [10]. However, in addition to a lack of confidentiality, TESLA's current systems in vehicle IoT environments still suffer from several limitations. These include the delay in authentication caused by the key-scan method, the timing and repeated transmission attack, and the lack of adaptability to highly dynamic vehicle scenarios. Additionally, existing solutions tend to focus on authentication, while overlooking the need for integrated encryption mechanisms that are lightweight and suitable for the low-resource environment of vehicles. Although TESLA 's system has many fixes, there is a need for integrated solutions that provide authentication, confidentiality, and efficiency for vehicle IoT systems. The majority of existing studies focus on individual aspects, leading to partial security solutions not fully tailored to real-time Vehicle-to-Infrastructure (V2I) communications. To overcome these issues, we need an efficient secure broadcast system that provides adequate authentication and light encryption to protect Basic Safety

Messages (BSM), while maintaining low computational and communication overhead.

The remainder of the research is divided as follows: Section 2 provides a review of related studies. The methodology is presented in Section 3, the design of the proposed system in Section 4, and the analysis and discussion of the results in Section 5; the conclusion of this research is reached in Section 6.

2. RELATED WORKS

2.1 TESLA-based authentication schemes

Many studies have focused on improving the TESLA protocol to enhance broadcast authentication in the IoV environment, given its efficiency in supporting group communications with limited resources.

A lightweight VANET broadcast authentication scheme based on TESLA and Bloom Filter achieved very low verification latency (below 0.01 ms) and improved privacy through mass alias changes. However, it relies on RSUs, is subject to Bloom Filter false positives, and does not fully address time synchronization [11]. A lightweight certificateless authentication system for VCS networks integrated the TESLA protocol with authentication tokens and a CRT-based motion prediction mechanism to achieve instant authentication. The result from this work achieves lower verification time, about 80 bytes of communication overhead per message, and better scalability than ECDSA and PBA. However, it suffers from design complexity and depends on the accuracy of motion prediction models [12]. Enhanced Inf-TESLA was proposed to provide continuous and low-cost authentication in IoT networks, overcoming key-chain expiration and authentication interruption in standard TESLA. The protocol achieves about 0.5 ms lower authentication delay, reduced MAC generation time, and O(n) complexity, while supporting continuous authentication without resynchronization. However, it does not provide data confidentiality since it relies on authentication without encryption [13]. A TESLA-based authentication and privacy scheme using a Cuckoo Filter (ITESLA-CF) was proposed to reduce memory and verification costs while supporting broadcast authentication in VANETs. The proposed protocol provides high packet delivery ratio (PDR) (97.56%), low latency (≈161 ms at 50 km/h) and low routing overhead (≈13.64%), demonstrating its suitability for high-traffic environments [14]. Based on the above review, existing studies mainly focus on individual objectives such as authentication, privacy, or decentralization. In contrast, the proposed MA_ETESLA framework aims to jointly provide lightweight mutual authentication, confidentiality, and low-latency performance in IoV environments.

2.2 Blockchain based authentication

Blockchain technology emerges as a promising solution for decentralized authentication for IoVs. Some studies have suggested blockchain-based authentication protocols to achieve a high level of security and transparency. A blockchain-based CTA protocol using ECC, PUF, and biometric keys was introduced for cross-authority authentication in IoV networks. The total computation cost was ≈21.4 ms while providing strong security, perfect forward secrecy, and high privacy with performance comparable to traditional techniques [15]. A blockchain-based privacy-preserving authentication protocol (PBAG) was presented to avoid direct blockchain interaction during authentication and CRL verification. Experimental results achieve an average fulfillment time of 0.36 ms, over 63.7% improvement in authentication time, and a message loss rate below 9.6%, making it suitable for time-critical IoV applications [16]. More efficient solutions are still needed that balance the advantages of blockchain with the performance requirements of the IoV environment, especially in time-sensitive applications. Despite these advantages, blockchain adoption in an IoV environment faces significant challenges, most notably the high latency resulting from verification and consensus processes and the increased communication and computational load, making it less suitable for applications requiring immediate response. While some solutions attempt to mitigate this impact by avoiding direct network interaction, this may come at the expense of certain security features.

2.3 Lightweight authentication schemes

Many studies have focused on developing lightweight authentication systems to suit the IoV environment, which requires rapid response and high resource efficiency. LBVP, a self-certified batch authentication protocol for fog-based vehicular networks, achieves a low total authentication cost of about 7.289 ms and a communication overhead of 248 bytes (396 bytes overall), making it suitable for time-critical and high-density vehicular environments [17]. The SSVC protocol was presented for enabling reliable and secure transmission in VANET. The aim of this study is to decrease the latency and enhance the transmission efficacy of the network. Initially, a network is made by a number of vehicles, and neighbor finding is executed through the WAVE protocol [18]. The UMBP protocol was proposed for multi-hop emergency message deployment in an urban VANET environment to address message delays and broadcast storms resulting from limited transmission range. Simulation results demonstrate that UMBP reduces single-hop time and improves propagation speed and reception rate compared to previous protocols [19]. The PLVA protocol was introduced for V2I authentication in VANETs to achieve low verification latency and reduced computational cost while preserving route privacy. The scheme outperforms digital signature–based methods and shows better performance than LVAP, EAAP, and Tzeng when authenticating up to 50 vehicles. However, its reliance on BGN homomorphic cryptography leads to high computational overhead when scaling to around 1,000 RSUs, limiting its suitability for large-scale deployments [20]. The Virtual Trustability Data Transmission (VTD) protocol was proposed as a trust-based routing model for VANETs to mitigate false data propagation. Performance evaluation shows that VTD achieves a PDR of 94% with an average packet failure rate of 6%, along with improved bandwidth utilization and network scalability compared to conventional VANET schemes [21]. Many of these systems prioritize efficiency at the expense of security, potentially making them vulnerable to attacks or failing to provide adequate data confidentiality protection.

2.4 Privacy-preserving schemes

Many studies have focused on developing mechanisms to preserve user privacy in the IoV environment. Due to the

sensitivity of data exchanged between vehicles and infrastructure, protocols have been proposed that strike a balance between privacy and conditional traceability in emergency situations. PPA6-IoV, a six-step privacy-preserving V2V authentication protocol based on symmetric session keys, achieves low communication overhead (≈2112 bits) and low computation cost (≈2.11 ms), with 39% and 91% improvements over existing techniques [22]. An SDN-based PAS scheme using BLS signatures was introduced for authentication and privacy in VANETs. This work achieved low computational cost (≈12.6 ms), low communication overhead (≈172 bytes), and reduced latency and packet loss compared to EAAP, TAAP, and LIAP, making it suitable for high-density networks [23]. A lightweight ECC-based authentication protocol was provided for broadcast emergency messages in VANETs using a hierarchical management framework. This work enables direct V2V authentication without RSUs, while supporting privacy and conditional traceability. This research achieves very low latency (≈0.365 ms for signing and ≈0.71 ms for verification) making it well suited for time-critical safety applications [24]. Many of these solutions face challenges related to increased computational costs or complex key management, especially when scaling up to a large number of vehicles or nodes. Also, some of these systems consider privacy but not the response time or efficiency, which may hinder their use in time-critical applications. In this regard, there is a need for solutions that adequately address confidentiality while ensuring system efficiency. While previous studies have demonstrated the protocol's effectiveness in authenticating broadcast messages, the proposed system addresses several shortcomings of previous approaches. Specifically, we propose a secure message broadcasting system for IoV environments based on an enhanced TESLA Protocol. The key contributions of this research are summarized as follows:

• Enhanced security in IoV environments using the modified TESLA Protocol.

• Integration of the TESLA Protocol with the Ascon-AEAD algorithm to ensure the security and confidentiality of vehicle messages.

• Providing V2I authentication, gradually eliminating the need for prior road infrastructure authentication.

• Verifying the effectiveness of the proposed system using Scyther software and evaluating its performance using NS-3.

3. THEORETICAL BEDGROUND

This section introduced the theoretical foundation that substantiate the proposed security system.

3.1 TESLA protocol

The TESLA protocol is a lightweight and efficient broadcast authentication protocol based on symmetric cryptography with asymmetric security features and is widely used in wireless sensor networks due to its packet loss resistance. In TESLA, the sender appends a Message Authentication Token (MAC) to each packet using a secret key and reveals this key after a predefined delay, enabling receivers to authenticate the cached messages [25]. However, TESLA requires imprecise time synchronization, prior knowledge of the key disclosure table, and an authenticated one-way key chain, as initial synchronization and exchange of security parameters typically rely on asymmetric cryptography [26].

TESLA uses a so-called one-way key chain. built by successively applying a one-way function F (.)More specifically, the i $-$ th key K_i is computed as $K_i = F(K_{i+1})$. The transmitter splits the time into intervals of equal duration where the i $-$ th time interval is associated with the i $-$ th key, K_i. Furthermore, relying on a different one-way function F' (·) a key K_i' is derived from K_i as $K_i' = F'(K_i)$. The authenticity of all messages sent during the i-th time interval is protected with a MAC computed via key K_i'. For example, if a message M j is sent during the i $-$ th time interval, the sender computes a MAC, MAC(K_i', M_j) and appends it to the message. Hence, the transmitted packet P_j is as follows [27, 28].

$$P_j = \{M_j \| \text{MAC}(K_i', M_j)\} \quad (1)$$

TESLA provides efficient broadcast authentication using symmetric encryption and MACs with low computational cost, relying on delayed key detection inaccurate time synchronization. and a unidirectional key chain for forward security [29].

3.2 Ascon encryption algorithm

In modern cryptosystems, encryption algorithms are essential for protecting important information [30]. Ascon is a family of encryption algorithms that includes authenticated ciphers (Ascon-AEAD128 and Ascon-AEAD128a), cryptographic hash (Ascon-Hash256), and extendable output functions (Ascon-XOF128 and Ascon-CXOF128). It is designed to provide high security, efficiency, and suitability for a wide range of cryptographic applications. It is primarily targeted at lightweight environments such as embedded systems and low-power devices but is also suitable for general-purpose use.

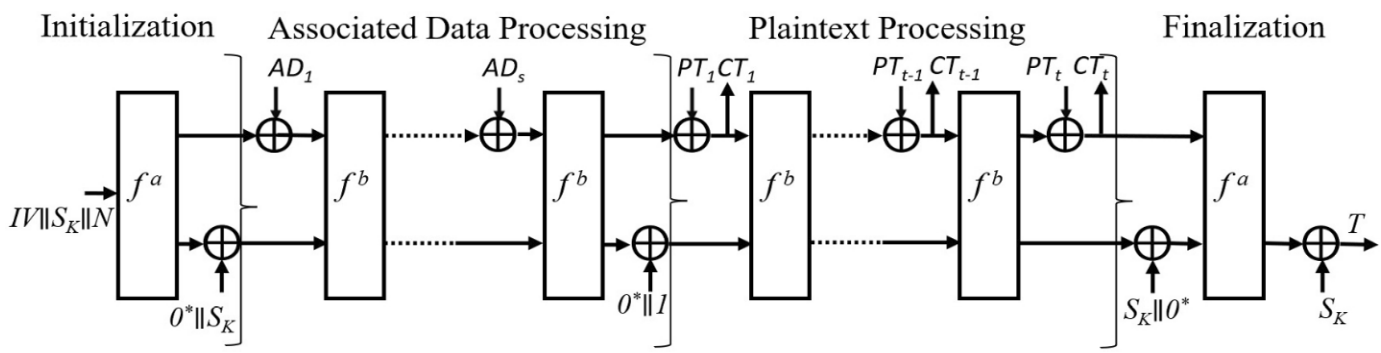

Figure 1. Ascon-AEAD128 encryption stages [31]

Ascon entered the CAESAR (competition for authenticated encryption: security, applicability, and robustness) competition and was a finalist in that competition due to high efficiency and security capability. It was later chosen as the winner of the NIST Lightweight Cryptography competition. Ascon is a symmetric-key encryption algorithm that offers encryption and authentication to resist numerous cryptographic attacks.

Ascon refers to an authenticated encryption algorithm that is a stream-based algorithm and is used to combine encryption and authentication. Ascon is divided into three primary variants, Ascon-AEAD128, Ascon-AEAD128a, and Ascon-80pq, that vary in some operational specifics but are similar to each other in their structure. An Ascon type design is constructed on a sponge formation and has a permutation core. The input data (key, plaintext, and associated data (AD)) are absorbed by the sponge construction and squeezed out results (ciphertext and authentication tag). Ascon uses a 128-bit state, divided into two main parts: one for the message and one for the AD. Figure 1 represents the inner stages of the encryption and decryption operations of the cipher [32].

4. PROPOSED AUTHENTICATION SCHEME (MA_ETESLA)

The proposed system adopts a V2I communication model consisting of three entities: vehicles, roadside units (RSUs), and a trusted authority for initial registration and key management. Vehicles generate and transmit authenticated messages, while RSUs receive, verify, and decrypt the messages. A mutual authentication is established between a vehicle and an RSU.

The system (MA_ETESLA) provides a secure and efficient solution for V2I communication, achieving mutual authentication, confidentiality and integrity without using public-key encryption. The system leverages TESLA-certified delayed key disclosure and Ascon-AEAD-certified encryption to provide efficient communication in vehicle networks. It involves the following steps: system configuration phase, vehicle registration phase, time synchronization phase, mutual authentication (V2I) phase, and secure message encryption phase and message transmission.

4.1 System initialization phase

In this phase, the road infrastructure (RSU) and all vehicles negotiate the TESLA system parameters used for secure communication. The parameters are the start time, the interval length, the key disclosure delay, the clock skew tolerance and the keychain length. Using these parameters, each vehicle and the infrastructure independently generate a one-way TESLA hash chain {}, where the first value of the hash chain is the commit value for the chain. This phase creates a common TESLA system configuration and enables each entity to create its own independent key chain for authentication.

4.2 Vehicle registration phase

After the system initialization phase, a vehicle registration step is performed to ensure trust before connectivity is established during operation. During this step, each vehicle's identity and TESLA's commitment value are registered with the trusted authority.

After generating its one-way TESLA key chain $\{K_0, \dots, K_{N-1}\}$, each vehicle submits its identifier (ID_v) and commitment key $C = K_0$ to the RSU, as shown in Eq. (2).

$$RegReq = \{ID_v \parallel C \parallel Timestamp\} \tag{2}$$

The RSU verifies the legitimacy of the vehicle through a secure registration channel and stores the binding ($ID_v \rightarrow C$). After successful verification, the RSU responds with a registration acknowledgment, as shown in Eq. (3).

$$RegAck = \{Status\} \tag{3}$$

The registered commitment values are later used to verify the disclosed TESLA keys during runtime authentication.

4.3 Synchronization phase

The synchronization phase is performed between the vehicle and the Roadside Unit (RSU) to ensure the correct operation of the TESLA protocol. Initially, the vehicle records its local time (t_v) and generates a random nonce, which is included in an initial synchronization request sent to the RSU.

Upon receiving this request, the RSU replies with the required TESLA timing parameters, including the reference start time T_0, the interval duration T_{int}, the key disclosure delay d, the key chain length N, and its TESLA commitment value. The response also echoes the received nonce and includes the RSU transmission timestamp t_S.

Based on the exchanged timestamps and the nonce, the vehicle estimates the upper bound of the synchronization error and determines a valid time window for TESLA interval computation. This enables loose time synchronization without requiring precise clock alignment, as shown in Eqs. (4) and (5).

$$\text{RSU} \rightarrow v\text{: } \{\text{Sender time } t_S, \text{Nonce}\}_{K_S^{-1}} \tag{4}$$

$$v \rightarrow \text{RSU: } \{\text{Receiver time } t_v, \text{Nonce}\}_{\boldsymbol{K_S^{-1}}} \tag{5}$$

After each vehicle receives a digitally signed response containing the time parameters and information required from the road infrastructure (RSU), the vehicle calculates the maximum possible synchronization error based on the exchanged timestamps. This procedure aims to ensure that the TESLA protocol's time security requirement is met under conditions of imprecise synchronization.

For the vehicle, the maximum possible transmission time (RM_{tt}) from the road infrastructure is estimated based on the receiver's local time and the received timestamp, using Eq. (6).

$$RM_{tt} = (t_v - t_R + t_S) \tag{6}$$

where, t_v refers to transmission times of sending Nonce to (RSU), t_R refers to receipt of the signed reply from the road infrastructure and t_S refers to road infrastructure time. For The road infrastructure, the maximum time received is calculated using the following equation.

$$SM_{tt} = (t_S - t_{SS} + t_v) \tag{7}$$

where, SM_{tt} refers to the maximum possible transmission time of the road infrastructure, t_S refers to the time of sending nonce to the vehicle, and t_{SS} refers to receipt of the signed

reply from the vehicle steps on Eq. (6) and Eq. (7). This procedure is used to define a valid time window for message acceptance and to enforce the time security condition in the TESLA protocol, thus achieving sufficient time synchronization without the need for precise clock adjustments or constant resynchronization. Unlike the traditional TESLA protocol, where only the receiver verifies the time, the proposed system applies the same synchronization procedure to both the vehicle and the Roadside Unit (RSU). This symmetrical design enables mutual synchronization and improves the system's suitability for dynamic V2I communication environments (Figure 2).

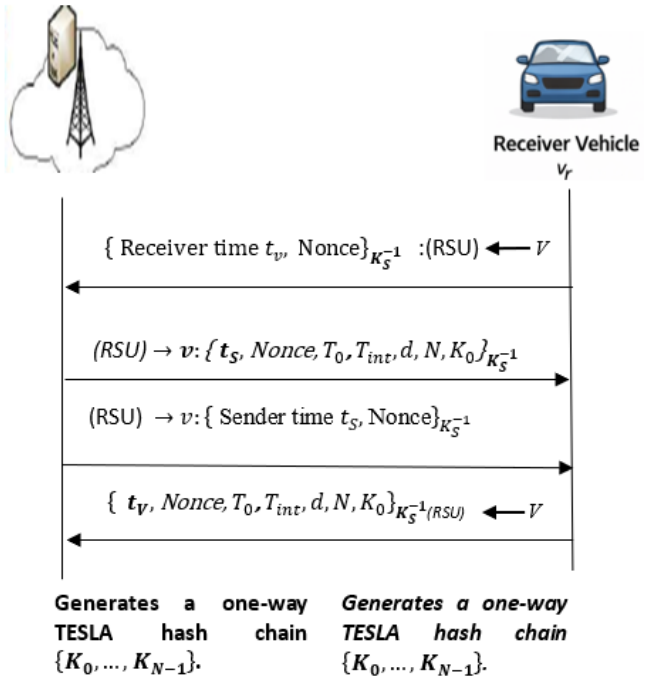

Figure 2. Transaction exchange diagram

4.4 Implementation of mutual authentication

After the synchronization phase is complete, security messages are sent between the vehicles and the road infrastructure. Previously, only the vehicles could transmit the message, and the road infrastructure would verify its authenticity. In the proposed system, mutual authentication is implemented, enabling both the vehicle and the road infrastructure to implement TESLA 's steps on both sides. When the road infrastructure sends a message (m_i) to vehicle v, it generates the TESLA key ($K_i s$) for the current time period and derives the authentication key ($K_i' s$ = F'($K_i s$)). Then the road infrastructure calculates the message's authentication token (MAC($K_i' s$, m_i)) and sends the packet (P = $\{m_i \| \text{MAC}(K_i' s, m_i)\}$) to vehicle. Upon receipt, vehicle temporarily stores the message and its MAC token without immediately accepting it, as the appropriate TESLA key has not yet been revealed. After a detection delay of d time intervals, the road infrastructure sends a subsequent detection message containing the TESLA key $K_i s$ to Vehicle ($P = \{m_i \| \text{MAC}(K_i' s, m_i) \| K_{i-d}\}$). Upon receiving this key, Vehicle verifies it against its stored commit key using the one-way keychain property, thus validating the ko_s. If the key verification is successful, Vehicle retrieves the authentication key, recalculates the cached message's MAC address, and compares it to the received MAc_i.

The message is accepted only if the two MAC addresses match, otherwise, it is rejected. To complete mutual authentication, the same sequence of operations is performed in reverse The road infrastructure authenticates incoming messages from Vehicle using its TESLA keychain, and the messages. After a period, the key is detected, the authentication code is recalculated, and the message is accepted or rejected accordingly. Through these mirrored operations, continuous mutual authentication is achieved between the vehicles v and the road infrastructure (RSU) as shows on Algorithm 1.

Algorithm 1: Mutual TESLA-based authentication between the vehicles and the road infrastructure (RSU).

Step1: The road infrastructure (S) and Vehicles (v) store each other's commitment keys ko_s and ko_v , (After synchronization phase).
Step2: Initialize receive buffers indexed by TESLA interval
Step3: S determines the current TESLA interval i
Step4: S selects TESLA key $K_i s$
Step5: S derives authentication key $K_i' s$ = F'($K_i s$)
Step6: S computes MAc_i = MAC($K_i' s$, m_i)
Step7: S sends packet Pi = { $m_i \| MAc_i$ } to v
Step8: v receives Pi and buffers (i, m_i, MAc_i)
Step9: After(d) intervals, S sends disclosed TESLA key $K_i s$ to v // Delayed key disclosure
Step10: v verifies $K_i s$ by checking $H^i(K_i s) = ko_s$
Step11: If key verification fails Then
Reject buffered messages for interval i
Step12: end if
Step13: v derives $K_i' s$ = F'($K_i s$)
Step14: v recomputes MAc_i^* = MAC($K_i' s$, m_i)
Step15: if $MAc_i^* = MAc_i$ then Accept message $m_i$as authentic
Else
Reject message
Step16: end if
Step17: v repeats Steps 3–7 to send authenticated messages to S. // Mutual Authentication
Step18: I repeats Steps 8–16 to verify messages from v
Step19: Mutual authentication is established between the vehicles and the road infrastructure (RSU).
Step20: end.

4.5 Ascon-based message encryption in the proposed system

In the proposed system, message encryption is performed using the Ascon authenticated encryption algorithm, by strictly following the encryption steps defined when a vehicle or the road infrastructure intends to transmit a message, the plaintext message is processed through the Ascon encryption stages. Initially, the transmitting vehicle or the road infrastructure prepares the encryption inputs, which include the secret key K, a unique nonce N, the associated data A, and the original plaintext message m_i . The associated data represents protocol-related information such as headers or identifiers, which are authenticated but not encrypted. The Encryption steps using the ASCON algorithm is illustrated on Figure 3.

(1) State initialization

First, the internal state S is initialized using the initialization vector, the secret key, and the nonce as follows:

$$S \leftarrow IV_{k,r,a,b} \| K \| N \quad (8)$$

In the initialization, (a) round of the round transformation

(p) are applied to the initial state followed by an (XOR) of the secret key k as following:

$$S \leftarrow p^a(S) \oplus (0^{320-k} \parallel K) \quad (9)$$

This step ensures that the encryption of the message m_i depends on the secret key and a unique nonce value.

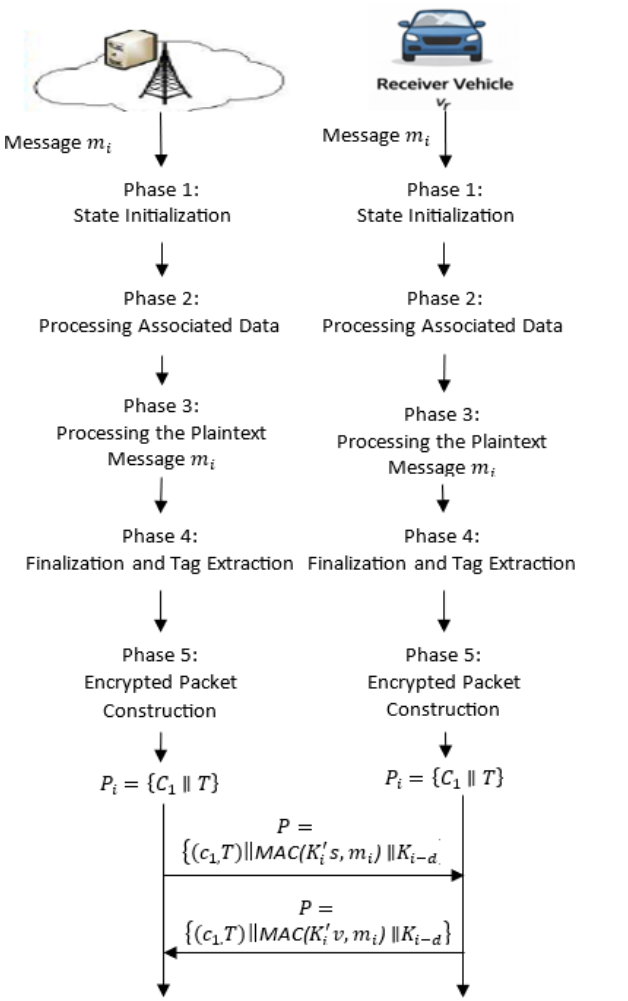

Figure 3. Encryption stages diagram

(2) Processing associated data

If associated data (A)exists (such as vehicle identifiers or protocol information), Ascon processes the associated data A in blocks of r bits it appends a single 1 and the smallest number of 0s to A to obtain a multiple of r bits and split it into s blocks of r bits as shown on Eq. (10). If A is empty, no padding is applied and s = 0:

$$A_1, \dots, A_s \leftarrow r\text{-bit blocks of } A \parallel 1 \parallel 0^* \quad (10)$$

Each block A_i with i = 1, …, s is XORed to the first bits Sr of the state S, followed by an application of the b -round permutation pb to S as following:

$$S \leftarrow p^b\big((S_r \oplus A_i) \parallel S_c\big) \quad (11)$$

After all associated data blocks are processed, domain separation is applied based on Eq. (12).

$$S \leftarrow S \oplus (0^{319} \parallel 1) \quad (12)$$

(3) Processing the plaintext message m_i

The message m_i is treated as the plaintext and divided into blocks as the following:

$$P_1, \dots, P_t \leftarrow r\text{-bit blocks of } m_i \parallel 1 \parallel 0^* \quad (13)$$

For each plaintext block except the last one, the ciphertext is computed based on Eqs. (14) and (15). Then the state is updated based on Eq. (16).

$$S_r \leftarrow S_r \oplus P_i \quad (14)$$

$$C_i \leftarrow S_r \quad (15)$$

$$S \leftarrow p^b(S) \quad (16)$$

For the final block is computed based on Eqs. (17) and (18).

$$S_r \leftarrow S_r \oplus P_t \quad (17)$$

$$\tilde{C}_t \leftarrow [S_r]_{|m_i| mod r} \quad (18)$$

As a result, the message m_i is transformed into the ciphertext C.

(4) Finalization and tag extraction

After encrypting the message m_i, the finalization phase is performed based on Eq. (19) then the authentication tag is extracted using Eq. (20).

$$S \leftarrow p^a\big(S \oplus (0^r \parallel K \parallel 0^{320-r-k})\big) \quad (19)$$

$$T \leftarrow S_{128} \oplus K_{128} \quad (20)$$

(5) Encrypted packet construction

Finally, the transmitted encrypted packet is constructed based on Eq. (21). The secure generation and transmission using (TESLA + Ascon-AEAD) are illustrated on Algorithm 2.

$$P_i = \{C_i \parallel T\} \quad (21)$$

Algorithm 2: Secure V2I message generation and transmission (TESLA + Ascon-AEAD)
Input: interval index i, The road infrastructure IDs, the vehicles ID v, TESLA key K_i, disclosure delay d, Ascon key K, nonce, N_i **Output**: authenticated and encrypted V2I packet **Step 1:** Generate the safety message m_i for the current TESLA interval. **Step 2:** Encrypt m_i using Ascon-AEAD with associated data A_i(e.g., IDs, ID v, i) to obtain:$(C_i, T_i) \leftarrow$ AsconEnc(K, N_i, A_i, m_i) **Step 3:** Compute the TESLA authentication code using the interval key K_i' derived from K_i, then: MAC$_i \leftarrow$ MAC(K_i', m_i) **Step 4:** Attach the disclosed TESLA key K_{i-d} (if available) to enable delayed verification. **Step 5:** Construct and transmit the V2I packet:$P_i = \{ C_i \parallel T_i \parallel$ MAC$_i \parallel K_{i-d} \}$ **Step 6:** end.

5. PERFORMANCE ANALYSIS

The proposed system was evaluated through three important axes: formal analysis using the Scyther formal protocol testing tool, informal analysis, and evaluation of the system's performance in vehicle networks using the NS-3.35 simulator.

5.1 Formal analysis of the proposed protocol security

The proposed MA_ETESLA protocol was formally analyzed using the Scyther verification tool. Two communicating entities were modeled: the vehicle as the sender and the Roadside Unit (RSU) as the receiver. The protocol execution consists of three main steps: transmission of the TESLA commitment value K0, transmission of the encrypted authenticated message, and delayed disclosure of the TESLA key K1.

The verification was conducted under the standard Dolev–Yao attacker model, where the adversary has full control over the public communication channel and may intercept, replay, modify, or inject messages. However, secure cryptographic primitives such as hash functions, MAC generation, and Ascon authenticated encryption are assumed to be computationally secure.

The analysis focused on verifying message confidentiality, agreement between communicating parties, synchronization freshness, and mutual authentication using the claims Secret, Alive, Weakagree, Niagree, and Nisynch. As shown in in Table 1, all specified claims were successfully verified, and no attacks were found within the explored state space.

Table 1. Results of SCYTHER testing for modified TESLA protocol

Claim			Status	Comments
Sender2	Secret_Hidden_1	OK	Verified	
Sender3	Secret K1	OK	Verified	
Sender4	Secret M	OK	Verified	
Sender5	Secret K0	OK	Verified	No attacks
Sender6	Alive	OK	Verified	
Sender7	Weakagree	OK	Verified	
Sender8	Niagree	OK	Verified	
Sender9	Nisynch	OK	Verified	
Receiver2	Secret_Hidden_2	OK	Verified	
Receiver3	Secret M	OK	Verified	
Receiver4	Secret K1	OK	Verified	
Receiver5	Secret K0	OK	Verified	No attacks
Receiver6	Alive	OK	Verified	
Receiver7	Weakagree	OK	Verified	
Receiver8	Niagree	OK	Verified	
Receiver9	Nisynch	OK	Verified	

The results of the verification using Scyther are shown in Table 1. All claims were verified and no attacks were reported, suggesting that no potential vulnerabilities exist in the considered threat model.

The confidentiality claims ensure that secret values, such as the message (M), the commitment value (K0) and the delayed key (K1), are not disclosed to untrusted parties. Moreover, the Nisynch claims confirm synchronization between the vehicle and Roadside Unit (RSU) in the protocol. The Niagree claims also confirm agreement between the parties on the data and the protocol session. From these results, the proposed MA_ETESLA protocol ensures formal confidentiality, authentication, synchronization, and agreement properties, and is shown to be resistant to replay, impersonation and message modification attacks in the threat model.

5.1.1 MA_ETESLA protocol template in Security Protocol Description Language

Scyther uses a dedicated protocol specification language known as the Security Protocol Description Language (SPDL) to model security protocols. SPDL defines protocol roles, parameters, message exchanges, and security claims, enabling formal verification of protocol correctness and security properties. The MA_ETESLA protocol is simulated as a communication process between two entities: the vehicle and the Roadside Unit (RSU). The communication protocol involves exchanging a commit, a secure message, and the subsequent issuance of a TESLA key, as described below.

```
1 hashfunction H;
2 hashfunction MAC;
3 secretaeadk:Function;
4 inversekeys(aeadk,aeadk);
5 usertypeMessage;
6 protocolTeslaAscon(Sender,Receiver)
7 {
8  role Sender
9  {
10    freshM:Message;
11    freshK1:Nonce;
12    var K0:Nonce;
13    send_1(Sender,Receiver,K0);
14    send_2(Sender,Receiver,{M,
15 MAC(M,K1)}
16       aeadk(Sender,Receiver));
17    send_3(Sender,Receiver,K1);
18    match(K0, H(K1));
19    claim(Sender,Secret,M);
20    claim(Sender,Nisynch);
21    claim(Sender,Niagree);
22  }
23  Role Receiver
24  {
25    var K0:Nonce;
26    var K1:Nonce;
27    var M:Message;
28    recv_1(Sender, Receiver, K0);
29    recv_2(Sender,Receiver,{M,
30 MAC(M,K1)}
31       aeadk(Sender,Receiver));
32    recv_3(Sender,Receiver,K1);
33    match(K0, H(K1));
34    claim(Receiver,Secret,M);
35    claim(Receiver,Nisynch);
36    claim(Receiver,Niagree);
37  }
38 }
```

The proposed MA_ETESLA protocol was formally modeled in SPDL and analyzed using the Scyther validator. Lines 1–4 define the main cryptographic primitives, including the hash function H, the MAC function, and the shared Ascon-AEAD key. Lines 8–22 define the sender role (vehicle). The sender generates a fresh message m and a delayed disclosure key K_1, then transmits the commitment value K_0, followed by the encrypted authenticated message containing m and $MAc\ (m, K_1)$. Finally, the sender reveals K_1, enabling delayed authentication. The relationship between the commitment value and the disclosed key is verified through the one-way hash condition $k_0 = H(K_1)$.

The sender specifies security claims related to confidentiality, synchronization, and agreement through the Secret, Nisynch, and Niagree claims.Lines 23–38 define the receiver role (RSU). The receiver first obtains the commitment value K0, then receives the encrypted protected message, and

finally receives the disclosed key K_1. After verifying the condition $K_0 = H(K_1)$. The receiver accepts the message as authentic and validates the same security properties, including confidentiality, synchronization, and agreement.

5.2 Informal security analysis of the MA_ETESLA protocol

This section provides an informal security analysis of the proposed protocol and explains how it meets the key security requirements for V2I communication environments.

(1) Confidentiality

In the proposed protocol, each transmitted security message (M) is encrypted before transmission using the Ascon-AEAD128 algorithm. Therefore, even if an attacker intercepts, the channel, the original text cannot be recovered without the shared secret key between the vehicle and the Roadside Unit (RSU). This overcomes the shortcomings of the traditional TESLA protocol, where messages are typically transmitted as plain text.

(2) Message Integrity

Message integrity is ensured through two layers of protection. First, the Ascon-AEAD algorithm generates an authentication tag for the encrypted payload. Second, the TESLA protocol uses a Message Authentication Token (MAC) generated with a K1 deferred detection key. Any modification to the ciphertext or authentication data results in verification failure and message rejection.

(3) Source Verification

Source verification is performed using a one-way TESLA hash and a delayed key disclosure mechanism. The sender first broadcasts the commit value K0, then later reveals the authentication key K1. The receiver verifies that K0 = H(K1). Since only the legitimate sender knows the correct key at the time of transmission, unauthorized parties cannot generate a valid Message Authentication Token (MAC).

(4) Rebroadcast Resistance

Rebroadcast resistance is achieved by binding each message to a specific time slot in TESLA. Messages received outside of this slot are rejected. Additionally, Ascon uses random values (nonces) and associated data to maintain message freshness. Therefore, packets that have been successfully intercepted cannot be reused.

(5) Protection Against Broker Attacks

Although an attacker can intercept or alter messages on the communication channel, successful forgery remains impossible. Before the key is revealed, the attacker does not know the TESLA authentication key. Furthermore, any alteration invalidates the Ascon authentication token. Therefore, altered messages are detected and discarded.

(6) Key Security

TESLA authentication keys are generated using a one-way hash chain, meaning that previously compromised keys cannot be used to extract subsequent keys. Therefore, compromising an old key does not reveal subsequent keys. Furthermore, the contents of messages remain protected by Ascon encryption.

5.3 Performance evaluation

This section evaluates the efficiency of the proposed (MA_ETESLA) system when applied to BSMs in V2I communication environments using the NS-3.35 network simulator. The main simulation parameters are summarized in Table 2.

The simulation scenario represents an urban V2I environment. In the baseline setup, 45 vehicles were deployed, while additional scalability experiments increased the number of vehicles up to 80 in order to examine the impact of network density. Vehicle speeds ranged from 50 to 100 km/h under a dynamic mobility pattern.

Each vehicle generated one protected message during every TESLA interval (Tint = 0.2 s), corresponding to 5 messages per second. The TESLA disclosure delay was configured as d = 2. The wireless channel was configured using IEEE 802.11p with a data rate of 6 Mbps, channel bandwidth of 10 MHz, transmission power of 23 dBm, and a range. propagation loss model with an effective communication range of 500 m.

All simulations were executed on a laptop running Ubuntu 22.04 LTS, equipped with a 13th Generation Intel® Core™ i7-13620H processor and 16 GB RAM. For comparative evaluation purposes, the reference protocols were selected from published studies that used identical vehicle scenarios in terms of vehicle speeds, network size, and evaluation criteria. However, some studies used different simulation platforms and varying implementation assumptions, which may lead to slight differences in the reported results. Therefore, the comparisons presented should be interpreted as relative performance indicators under identical conditions.

Table 2. Simulation parameters

Parameters	Values
Platform	Ubuntu 22.04 LTS
Tool used	NS- 3.35
Wireless protocol	802:11p
Number of vehicles	(45-80)
Number of RSUs	(1-2)
Simulation time	60 seconds
Simulation area	1000 m

5.3.1 Communication overhead

Figure 4 illustrates the additional communication overhead for transmitting a single BSM message, measuring the number of extra bytes required to secure the transmitted messages. In a scenario of 45 vehicles traveling at 50 km/h, the MA_ETESLA achieved an additional communication overhead of 99.75 bytes, comprising: the message authentication code/algorithm (HMAC-SHA256) = 32 bytes, a random value (Nonce) = 16 bytes, a TESLA key = 32 bytes, an authentication flag (ASCON) = 16 bytes, and the key index (i) = 4 bytes. Therefore, the additional security cost per message can be expressed as: 32 + 16 + 16 + 32 + 4 = 100 bytes. The average communication cost is calculated as follows: = (Total security bytes) / (Number of messages) = 99.75 bytes, compared to 294 bytes [11], 396 bytes [17], 204 bytes [24], 264 bytes [22], and 172 bytes [23]. The results indicate that the proposed system requires fewer additional bytes than the compared systems. This improvement is primarily attributed to the use of lightweight encryption processes and integrated authentication mechanisms. Reducing the additional communication burden contributes to better bandwidth utilization and less channel congestion.

5.3.2 Computational cost of signing and verifying Basic Safety Messages

The average time taken to compute MAC signatures and MAC verifications for BSM messages in the systems under consideration, in milliseconds, is shown in Table 3. This is the time required to compute the security measures in each system.

We observe that the time required is dependent on the

security method used in different systems. For instance, system [12] takes 0.18128 milliseconds for signature and 0.03691 milliseconds for verification, since it involves several MAC operations. On the other hand, system [11] only requires 0.02 milliseconds for verification due to the use of a single MAC process. In contrast, system [24] has the highest processing cost of 0.365 milliseconds for signing and 0.7106 milliseconds for verification. Among the systems, the proposed MA_ETESLA scheme has a comparatively low processing cost, at 0.017800 milliseconds for signature and 0.017500 milliseconds for verification. This is due to the low-cost MAC.

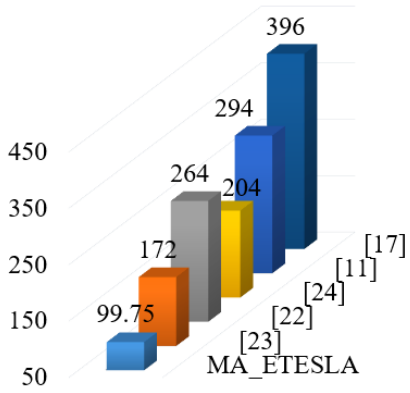

Figure 4. Communication overhead of Basic Safety Messages (BSM)

Table 3. Signing and verifying cost of Basic Safety Messages (BSM) in milliseconds

Measures	[11]	[12]	[24]	MA_ETESLA
Signing cost (ms)	No	0.18128	0.365	0.017800
Verifying cost (ms)	0.02	0.03691	0.7106	0.017500

5.3.3 Performance evaluation under varying vehicle speeds

This section evaluates how well the proposed system performs for different speed ranges (50 km/h to 100 km/h). Tables 4, 5, and 6 provide results for 3 main performance metrics: PDR; Data Transfer Rates; and Transmission Delay, showing each of these metrics for the proposed system against other protocols.

PDR is used to evaluate the reliability of message delivery in a vehicle internet network. This metric represents the percentage of packets successfully received compared to the total number of packets sent during the simulation time.

$$(PDR\%) = \frac{g_{-rx}}{g_{-tx}} * 100 \quad (22)$$

where, g_{-rx} = Number of packages successfully consumed, g_{-tx} = Total number of packets sent

Table 4 and Figure 5 illustrate a comparison of PDR values between MA_ETESLA protocol and other systems at various vehicle speeds ranging from 50 to 100 km/h. The results show that the MA_ETESLA protocol achieved high PDR values at low and medium speeds, reaching 99.89% at 50 km/h and 99.31% at 70 km/h. The value gradually decreased with increasing speed, reaching 91.64% at 100 km/h.

Compared to studies [14, 18, 19], the MA_ETESLA protocol demonstrated higher PDR values at most of the tested speeds. At 50 km/h, the MA_ETESLA protocol recorded 99.89%, compared to 97.56% [14], 93.53% [18], and 90.56% [19]. At 70 km/h, the MA_ETESLA protocol achieved 99.31%, compared to 92.26% [14], and 84.84% [18], and 82.08% [19], respectively. At 100 km/h, the MA_ETESLA protocol maintained a relatively high performance of 91.64%, close to 91.20% [14], and higher than 82.93% [18] and 78.69% [19].

These results indicate that the proposed system maintained a relatively high delivery rate across various speeds, with the difference compared to some comparable methods becoming smaller at higher speeds. The gradual decrease in PDR with increasing speed can be explained by the increased topological variations and shorter contact time between vehicles and infrastructure.

Table 4. Analysis of packet delivery ratio (PDR) results for the proposed system with other protocols

Speed (km/h)	[14]	[18]	[19]	MA_ETESLA
50	97.56	93.53	90.56	99.89
60	96.71	92.68	89.29	99.66
70	92.26	84.84	82.08	99.31
80	92.05	86.75	82.51	97.02
90	90.77	83.78	79.75	94.62
100	91.20	82.93	78.69	91.64

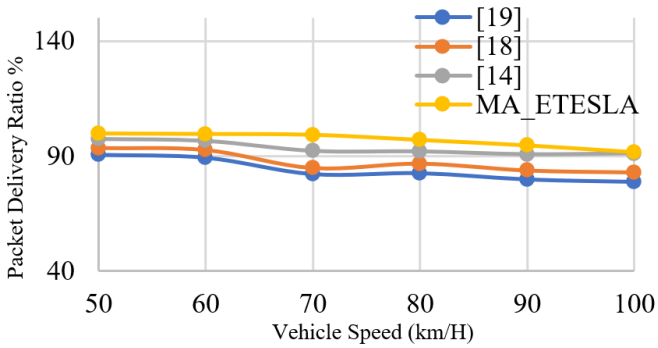

Figure 5. Packet delivery ratio (PDR) analysis of proposed system model

Throughput is a metric used to evaluate data transmission efficiency in a vehicle network, representing the amount of data successfully received per unit time. In this work, throughput is expressed in kilobytes per second (KB/s).

$$\text{Throughput} = \frac{\text{Total Successfully Received Data (KB)}}{\text{Simulation Time (s)}} \quad (23)$$

Total successfully received data (KB) represents the amount of data that was successfully delivered to the recipient. Where the simulation time (s) is the total simulation time.

Table 5 and Figure 6 illustrate the throughput comparison between the proposed system and other systems at various vehicle speeds ranging from 50 to 100 km/h.

Table 5. Analysis of throughput results for the proposed system with other protocols (KB/S)

Speed (km/)	[14]	[18]	[19]	MA_ETESLA
50	90,956.06	90,563.60	86,344.75	109,800.00
60	90,563.63	89,647.88	84,807.65	102,560.00
70	90,825.24	90,759.83	86,148.52	96,040.00
80	90,400.08	89,867.81	86,475.57	103,146.67
90	90,007.63	89,582.48	84,807.65	99,600.00
100	90,105.74	89,615.18	86,671.79	106,800.00

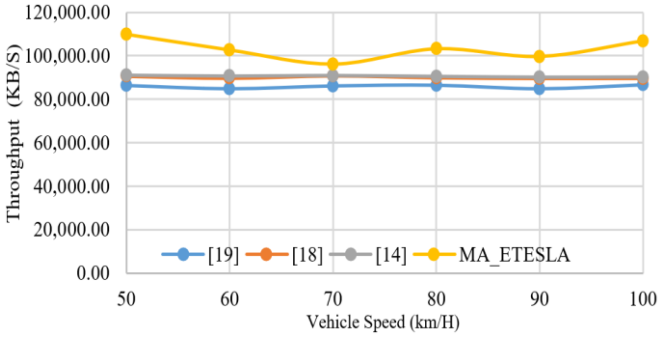

Figure 6. Throughput analysis of proposed system model

The results showed that the proposed MA_ETESLA system achieved throughput values ranging from (96,040.00 KB/s to 109,800.00 KB/s) across all tested speeds, generally higher than the comparable systems. At 50 km/h, the proposed system recorded 109,800.00 KB/s, compared to 90,956.06 KB/s in [14], 90,563.60 KB/s in [18], and 86,344.75 KB/s in [19]. At 100 km/h, the proposed system maintained a value of 106,800.00 KB/s. While other systems recorded lower values, these results indicate slight fluctuations in data transmission rates across speeds in vehicle communication environments due to variations in link duration, channel conflicts, and packet retransmission behavior. However, the proposed system maintained good data transmission efficiency under different traffic conditions, which is attributed to the low communication load and the use of lightweight security processes that helped reduce packet loss

Table 6 and Figure 7 compare the average transmission delay of the proposed system with other protocols under different vehicle speeds (50 km/h to 100 km/h). The delay metric was calculated as the average time difference between packet transmission and successful reception.

Table 6. Analysis of transmission delay for the proposed system with other protocols

Speed (km/h)	[14]	[18]	[19]	MA_ETESLA
50	161.38	205.80	283.53	61.16
60	186.36	241.89	308.51	60.65
70	216.90	266.87	347.38	54.15
80	244.66	300.18	372.36	53.74
90	269.65	336.27	405.67	60.59
100	302.96	364.03	430.66	61.01

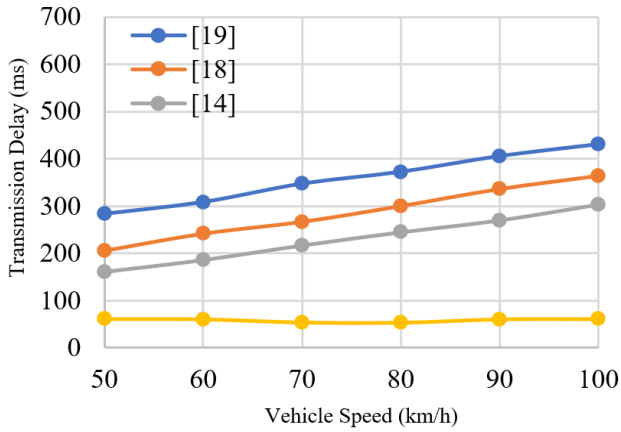

Figure 7. Transmission delay analysis of proposed system model

$$D_{avg} = \frac{\Sigma_{i=1}^{N}(tr_i - ts_i)}{N} * 1000 \quad (24)$$

where, D_{avg} = average transmission delay, N = number of successfully delivered packets, tr_i = packet reception time after verification, ts_i = packet transmission time.

The results indicate that the proposed MA_ETESLA protocol outperformed the other protocols in terms of delay for all speed ranges. At 50 km/h, the proposed system recorded an average delay of 61.16 ms, compared to 161.38 ms for method [14], 205.80 ms for method [18], and 283.53 ms for method [19]. At 70 km/h, the delay was reduced to 54.15 ms, compared to 216.90 ms, 266.87 ms, and 347.38 ms for the other protocols.

At higher speeds, the proposed method showed a consistent performance, with 53.74 ms delay at 80 km/h, 60.59 ms at 90 km/h and 61.01 ms at 100 km/h. By contrast, the other protocols exhibited much higher delays that increased as the speed increased. These findings suggest that MA_ETESLA enables faster and secure message delivery and lower delay under the dynamic vehicle network environment, and is thus suitable for vehicle safety applications that require low delay.

Figure 8 compares the computational cost of the proposed MA_ETESLA system with several authentication protocols. Computational cost is defined as the total time required to execute the encryption (t_{enc}), MAC (t_{mac}), decryption (t_{dec}), and MAC verification operations ($t_{ver-mac}$) associated with message protection. This metric reflects the computational load resulting from the implementation of security mechanisms, with lower values indicating faster execution and lower resource consumption. In the proposed system, the computational cost was calculated as the following:

$$T_{total} = t_{enc} + t_{mac} + t_{dec} + t_{ver-mac} \quad (25)$$

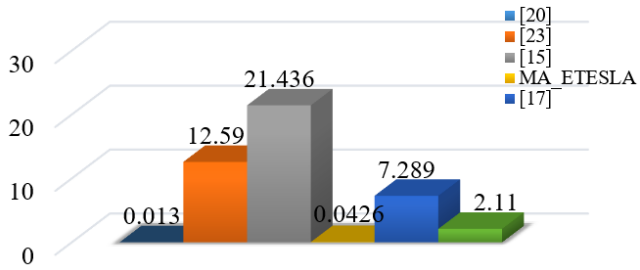

Figure 8. Computation cost analysis of proposed system model

The proposed system has a response time of 0.0426 ms, which is a lower value that suggests fast processing and is suitable for real-time communication and vehicle-based IoT systems. As shown in Figure 9, protocol [15] had the highest response time of 21.436 ms, followed by protocol [23] with 12.59 ms and then protocol [17] with 7.289 ms. Such values can be justified by the use of complex security techniques by these systems, like public-key encryption, digital signature, blockchain, or multiple authentication factors. The response time of protocol [22] was 2.11 ms, which is lower than some of the other systems but still higher than the proposed system. The lowest value of 0.013 ms was recorded for protocol [20], which uses a simple authentication mechanism with only a few encryption processes. Overall, the proposed MA_ETESLA system was able to record one of the lowest computational costs, while offering security and authentication features by

combining the Ascon-AEAD algorithm and TESLA mechanism. These findings suggest that the proposed system strikes a good trade-off between security and efficiency, and is therefore a suitable choice for highly dynamic networks.

Figure 9 compares the average packet loss in the proposed system with that of protocol [12] at different vehicle speeds. The packet loss was calculated from the delivery ratio as follows:

$$\text{PLR} = \frac{100 - PDR_{delivery}}{100} \quad (26)$$

where, $PDR_{delivery}$ = Number of packages received successfully.

Results from Figure 10 shows that the average packet loss ratio increases progressively with increasing vehicle speed for both the proposed system and the reference system [12] within the speed range of 0 to 25 km/h. This increase is attributed to the fact that higher speeds lead to faster changes in network architecture and reduced communication time between vehicles, which increases the likelihood of link breaks and packet loss during transmission. The results also show that the proposed system achieves similar performance to protocol [12] at low speeds, while a slight difference is observed at high speeds. At 25 km/h, the proposed system recorded a value of approximately 0.13, compared to a value of approximately 0.12 for protocol [12]. These results indicate that the proposed system maintains an acceptable level of reliability with increasing mobility in all cases. Overall, the figure shows that both systems are affected by the increase in speed, and that the packet loss ratio gradually increases as the network dynamics increase.

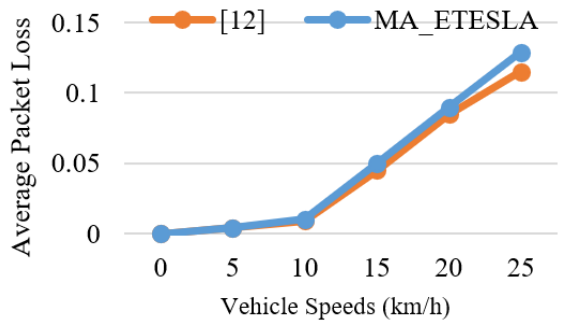

Figure 9. Impact of vehicles' speed on average packet loss ratio

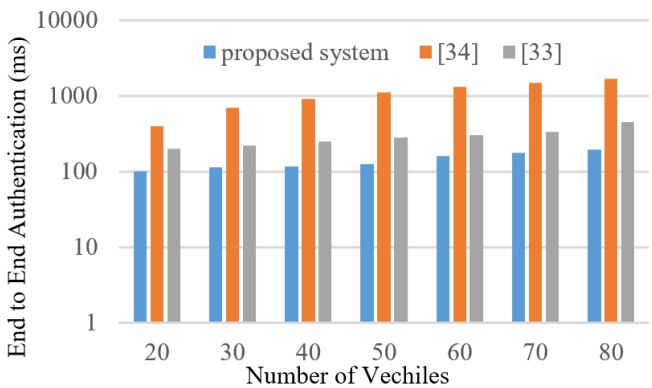

Figure 10. End-to-end authentication in milliseconds (ms)

5.3.4 End-to-end authentication delay

The total authentication time delay is used to measure the total time required from the moment a message is created by the sender until it is verified and accepted by the receiver. This is measured at varying vehicle densities ranging from 20 to 80 vehicles. Figure 10 compares the overall authentication time of the proposed system with the two reference systems [16, 33]. The results show that authentication time gradually increases with the number of vehicles in all systems due to increased competition for the wireless channel and higher network load. However, the proposed system maintained the lowest authentication time across all tested scenarios.

At a density of 20 vehicles, the proposed system recorded a time of 100.26 ms, compared to 200 ms and 400 ms for systems, respectively. At a density of 80 vehicles, the time was 295.61 ms, compared to 450 ms and 1700 ms for systems, respectively [16, 33].

These results indicate that the proposed system offers better scalability and higher efficiency in dense vehicle environments, thanks to lightweight authentication mechanisms that reduce computational load and response time.

6. CONCLUSION

This research proposed a lightweight security system for V2I communication by integrating the TESLA and Ascon-AEAD Algorithm-based Protocol (MA_ETESLA). The proposed system provides message-level mutual authentication and message confidentiality while meeting the time and resource limitations of the IOV environment. The ETESLA protocol was formally verified using SCYTHER, and the results showed it to be secure against standard attacks and to have the lowest signing and verifying costs with minimal processing time. Simulation experiments showed that the MA_ETESLA protocol has an efficient performance compared to existing protocols, where the PDR was 99.89% and the transmission latency was 61.16 ms at a vehicle speed of 50 km/h and the transmission response time was 61.01 ms at a vehicle speed of 100 km/h. The proposed MA_ETESLA protocol reduces authentication time by approximately 45% and provides a faster response time for verification processes. Nonetheless, it will remain a crucial area for future research if a large-scale deployment in real-world systems experiences time alignment issues and challenges pertaining to the leveraging of TESLA keychains due to high levels of network dynamics.

ACKNOWLEDGMENT

We would like to thank Mustansiriyah University for invaluable assistance in our research.

REFERENCES

[1] Abbood, F., Ayed, L. (2025). Efficient lightweight cryptographic framework for securing medical images in IoT systems. Ingénierie des Systèmes d'Information, 30(4): 845-853. https://doi.org/10.18280/isi.300402

[2] Azis, A., Azis, Krisbiantoro, D. (2023). Internet of things (IoT) innovation and application to intelligent governance systems: A case study on DISHUB for Transport vehicles, JOIV: International Journal on Informatics Visualization, 7(1): 193-199.

http://dx.doi.org/10.30630/joiv.7.1.1282
[3] Mishra, P., Singh, G. (2025). Internet of vehicles for sustainable smart cities: Opportunities, issues, and challenges. Smart Cities, 8(3): 93. https://doi.org/10.3390/smartcities8030093
[4] Mande, S., Ramachandran, N. (2024). A comprehensive survey on challenges and issues in V2X and V2V communication in 6g future generation communication models. Ingénierie des Systèmes d'Information, 29(3): 951-960. https://doi.org/10.18280/isi.290315
[5] Mohammed, A., Shibeeb, A., Ahmed, M. (2022). Image cryptosystem for IoT devices using 2-D Zaslavsky chaotic map. International Journal of Intelligent Engineering & Systems, 15(2): 543-553. https://doi.org/10.22266/ijies2022.0430.48
[6] Alam, T. (2024). Data privacy and security in autonomous connected vehicles in smart city environment. Big Data and Cognitive Computing, 8(9): 95. https://doi.org/10.3390/bdcc8090095
[7] Khan, M.A.A., Kaidi, H.M. (2023). A comprehensive survey of machine learning techniques in next-generation wireless networks and the Internet of Things. Ingenierie des Systemes d'Information, 28(4): 959-967 https://doi.org/10.18280/isi.280416
[8] Wijaya, M.C. (2024). Security analysis of SQL injection attacks on multimedia and journal-services sites using concatenated input validation and parsing method (CIVP). Ingenierie des Systemes d'Information, 29(5): 1915-1924. https://doi.org/10.18280/isi.290523
[9] Boyce, S. Mangwala, M., Chuma, J. (2019). Modified timed efficient stream loss-tolerant authentication to secure power line communication. International Journal of Electrical and Computer Engineering, 9(4): 2281-2295. https://doi.org/10.11591/ijece.v9i4
[10] Muhammad, M., Safdar, G. (2025). PV+ TESLA: A secure integrated approach to reduce message verification delay in V2V networks. Journal of Network and Systems Management, 33(2): 32. https://doi.org/10.1007/s10922-025-09910-7
[11] Bao, S., Hathal, W., Cruickshank, H., Sun, Z., Asuquo, P., Lei, A. (2017). A lightweight authentication and privacy-preserving scheme for VANETs using TESLA and Bloom Filters. Information & Communications Technology Express, 4(4): 221-227. https://doi.org/10.1016/j.icte.2017.12.001
[12] Hathal, W., Cruickshank, H., Sun, Z., Maple, C. (2020). Certificateless and lightweight authentication scheme for vehicular communication networks. IEEE Transactions on Vehicular Technology, 69(12): 16110-16125. https://doi.org/10.1145/3583781.3590249
[13] Eledlebi, K., Alzubaidi, A.A., Yeun, C.Y., Damiani, E., Mateu, V., Al-Hammadi, Y. (2022). Enhanced Inf-TESLA protocol: A continuous connectivity and low overhead authentication protocol via IoT devices. IEEE Access, 10: 54912-54921. https://doi.org/10.1109/ACCESS.2022.3177268
[14] Theodore, S., Gandhi, K., Palanisamy, V. (2023). A novel lightweight authentication and privacy-preserving protocol for vehicular ad hoc networks. Complex & Intelligent Systems, 9(3): 2981-2991. https://doi.org/10.1109/APCI65531.2025.11136928
[15] Xie, Q., Sun, Z., Xie, Q., Ding, Z. (2023). A cross-trusted authority authentication protocol for internet of vehicles based on blockchain. IEEE Access, 11: 97840-97851. https://doi.org/10.1109/ACCESS.2023.3308601
[16] Feng, X., Cui, K., Wang, L., Liu, Z., Ma, J. (2024). PBAG: A privacy-preserving blockchain-based authentication protocol with global-updated commitment in IoVs. IEEE Transactions on Intelligent Transportation Systems, 25(10): 13524-13545. https://doi.org/10.1109/TITS.2024.3399200
[17] Zhang, X., Zhong, H., Cui, J., Bolodurina, I., Liu, L. (2022). LBVP: A lightweight batch verification protocol for fog-based vehicular networks using self-certified public key cryptography. IEEE Transactions on Vehicular Technology, 71(5): 5519-5533. https://doi.org/10.1109/TVT.2022.3157960
[18] SathyaNarayanan, P. (2019). A sensor enabled secure vehicular communication for emergency message dissemination using cloud services. Digital Signal Processing, 85: 10-16. https://doi.org/10.1016/j.dsp.2018.06.003
[19] Krishnakumari, S., Arunadevi, P. (2017). UMBP: Multi-hop emergency alert-based intelligent transportation in VANET. Elysium Journal, 4(2): 1-8.
[20] Lv, S., Liu, Y. (2022). PLVA: Privacy-preserving and lightweight V2I authentication protocol. IEEE Transactions on Intelligent Transportation Systems, 33(7): 6633-6639. https://doi.org/10.1109/TITS.2021.3059638
[21] Ruban, C., Paramasivan, B. (2020). VTD protocol: A cluster based trust model for secure communication in VANET. International Journal of Intelligent Engineering & Systems, 13(1): 35-45. https://doi.org/10.22266/ijies2020.0229.04
[22] Jamal Ibrahim, S., Beitollahi, H. (2024). PPA6-IoV: A six-step privacy-preserving authentication protocol for the internet of vehicles. IEEE Access, 12: 168120-168134. https://doi.org/10.1109/ACCESS.2024.3459948
[23] Deng, X., Gao, T., Guo, N., Qi, J., Zhao, C. (2022). PAS: Privacy-preserving authentication scheme based on SDN for VANETs. Applied Sciences, 12(9): 4791. https://doi.org/10.3390/app12094791
[24] Aocheng, Y., Yao, J., Wei, W. (2025). A lightweight security authentication protocol for VANET traffic emergency message broadcasting. In 2025 10th International Symposium on Advances in Electrical, Electronics and Computer Engineering (ISAEECE), Xi'an, China, pp. 293-304. https://doi.org/10.1109/ISAEECE66033.2025.11159882
[25] Gauhar Fatima, S., Kausar Fatima, S., Mohd Ali, S. (2019). A security protocol for wireless sensor networks. International Journal of Advanced Research in Engineering and Technology, 10(2): 155-174.
[26] Li, L., Teng, S., Zhao, Y., An, D., Xi, H. (2025). Experimental study on a Tesla turbine integrated into a Mini-ORC system using R245fa as working fluid. Energy, 324: 136067. https://doi.org/10.1016/j.energy.2025.136067
[27] Wu, Z., Zhang, Y., Liu, L., Yue, M. (2020). TESLA-based authentication for BeiDou civil navigation message. China Communications, 17(11): 194-218. https://doi.org/10.23919/JCC.2020.11.016
[28] Wimpenny, G., Lázaro, F., Šafář, J., Raulefs, R. (2025). A pragmatic approach to VDES authentication. Journal of the Institute of Navigation, 72(1). https://doi.org/10.33012/navi.681

[29] Liu, C., Dong, W., Ma, M., Xiao, S. (2025). Design and implementation of multiple authentication schemes on navigation messages. China Communications, 22(9): 264-288. https://doi.org/10.23919/JCC.fa.2024-0050.202509
[30] Kareem, A., Ahmed, M., Albermany, S. (2023). An efficient image encryption method based on enhanced Josephus problem and a non invertible economic map. International Journal of Computing, 22(4): 493-501. https://doi.org/10.47839/ijc.22.4.3357
[31] Khan, S., Inayat, K., Muslim, F., Shah, Y.A., Ur Rehman, M.A., Khalid, A., Imran, M., Abdusalomov, A. (2024). Securing the IoT ecosystem: ASIC-based hardware realization of Ascon lightweight cipher. International Journal of Information Security, 23: 3653-3664. https://doi.org/10.1007/s10207-024-00904-1
[32] Kaiser, A., Hoiness, G. (2025). Throughput of ASCON compared with popular IoT encryption algorithms. Military Cyber Affairs, 8(1).
[33] Yang, A. (2020). Delegating authentication to edge: A decentralized authentication architecture for vehicular networks. IEEE Transactions on Intelligent Transportation Systems, 23(2): 1284-1298. https://doi.org/10.1109/TITS.2020.302400