



Secure and Lightweight ECC-Based Mutual Authentication and Key-Exchange Protocol for Internet of Vehicles

Chirayu Anchan¹, Hema Shekhawat^{*1} , and Jitendra Singh Yadav¹ 

¹Department of Computer and Communication Engineering, Manipal University
Jaipur, Rajasthan, India

chirayu.229303079@mu.manipal.edu, *hemashekhawat@gmail.com,
jitendrasingh.yadav@jaipur.manipal.edu

Abstract. The Internet of Vehicles (IoV) enables continuous data exchange between vehicles, roadside units, and remote servers to support safety-critical applications. Because these communications occur over open wireless channels, they are inherently exposed to security threats such as replay attacks, impersonation, and topological tracking. Existing lightweight authentication protocols struggle to balance computational efficiency with robust privacy, frequently failing to provide mutual authentication or adequately protect vehicle identity. To address these vulnerabilities, this paper proposes a secure, lightweight mutual authentication and key-exchange protocol based on Elliptic Curve Cryptography. The protocol introduces an asymmetric identity masking technique to protect the vehicle's identity through an ephemeral masking key and establishes a forward-secure session key utilizing a multi-ephemeral cryptographic exchange. The protocol's resilience is evaluated through both informal analysis and rigorous formal verification using the Scyther symbolic model checker. Operating under the Dolev-Yao adversary model, the verification proves absolute session key secrecy and synchronization across an unbounded number of concurrent sessions. Performance evaluations demonstrate a total CPU average execution time of approximately 11.9 milliseconds. This minimal computational overhead confirms the protocol's high efficiency and suitability for the resource-constrained Electronic Control Units characteristic of modern IoV environments.

Keywords: Internet of Vehicles - elliptic curve cryptography - mutual authentication - key exchange - dual pseudonymity - V2X security - lightweight protocol.

1 Introduction

The rapid increase of connected and autonomous vehicles has given rise to the Internet of Vehicles (IoV), a large-scale network in which vehicles interact with each other (V2V), with roadside infrastructure (V2I), and with remote cloud

servers (V2C) [6, 12]. These communications support safety-critical applications such as emergency braking, collision avoidance, and real-time traffic management. Because IoV communications occur over open wireless channels, they are inherently vulnerable to a range of security threats. An adversary may eavesdrop on transmitted messages to recover session keys or track vehicle movements, replay previously captured valid messages to mount impersonation attacks, or act as a man-in-the-middle to modify exchanged data [13]. Protecting against these threats while preserving the real-time constraints of IoV applications is a significant challenge. Traditional public-key cryptographic schemes based on RSA require 3072-bit keys for 128-bit security and involve expensive modular exponentiation operations, making them impractical for the embedded electronic control units (ECUs) found in modern vehicles. Elliptic Curve Cryptography (ECC) achieves equivalent security with only 256-bit keys and operations that are several times more efficient, making it the preferred choice for constrained IoV environments [8]. A key limitation of many existing IoV authentication protocols is their failure to adequately protect vehicle privacy without incurring heavy communication overheads. A representative example is the lightweight ECC-based protocol proposed by Tewari et. al. [5], which provides mutual authentication for IoT devices but relies on static identity parameters that can leave nodes vulnerable to correlation attacks. The proposed protocol addresses this gap by extending Tewari et. al [5] design with an asymmetric identity masking technique for the vehicle. The vehicle's identity is protected via an ephemeral masking key, transmitted as $C_v = ID_v \oplus K_{mask}$ [11]. Furthermore, the session key derivation is redesigned to consolidate ephemeral parameters, eliminating the need to transmit the vehicle's key-exchange point R_1 separately [14]. This paper makes the following contributions:

- We propose a strengthened four-message mutual authentication and key-exchange protocol based on ECC over NIST P-256 that introduces an asymmetric identity masking mechanism, protecting the vehicle's identity from tracking and correlation attacks over open wireless channels.
- We introduce a compact session key derivation using the shared point $K = R'_1 \cdot r_2 = r_s \cdot r_1 \cdot r_2 \cdot G$, which consolidates ephemeral parameters to reduce overall communication overhead.
- We provide a comprehensive informal security analysis demonstrating resistance to replay attacks, impersonation, man-in-the-middle attacks, eavesdropping, vehicle tracking, and known session key attacks.
- We formally verify the protocol using the Scyther symbolic model checker under the Dolev-Yao model, proving absolute session key secrecy and synchronization across an unbounded number of concurrent sessions.
- We evaluate the protocol's performance, reporting an approximate 11.9 ms CPU execution time, demonstrating that the robust security guarantees are achieved with minimal computational overhead suitable for IoV ECUs.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 presents cryptographic preliminaries. Section 4 defines the system and threat model. Section 5 describes the proposed protocol. Section 6

presents the security analysis. Section 7 evaluates performance. Section 8 concludes the paper.

2 Related Work

Authentication in vehicular networks has been studied extensively. Early proposals relied on RSA-based PKI, but the overhead of certificate management proved prohibitive for real-time V2X scenarios [7,9]. ECC-based protocols significantly reduced overhead. Basic ECDH achieves forward-secure key agreement in two rounds with only two scalar multiplications, but provides neither mutual authentication nor privacy protection [10]. Vasudev et al. [2] proposed a lightweight mutual authentication protocol for V2V communication in IoV using ECC, enabling vehicles to authenticate with a trusted authority. While the scheme is computationally efficient, it has been shown to have design weaknesses including vulnerability to vehicle-server impersonation attacks and lack of forward secrecy [3]. Furthermore, the server identity is used statically in all session tokens, making the server traceable across sessions. Tewari and Gupta [5] proposed a lightweight mutual authentication protocol based on ECC specifically targeting IoT devices, which shares the closest design philosophy with the proposed protocol. Their scheme achieves mutual authentication and session key establishment using a partial private key issuance mechanism similar to the one adopted here. However, the Tewari–Gupta protocol has three notable limitations. First, the server identity ID_s is used directly in all authentication tokens, making the server traceable across sessions. Second, the session key derivation requires the ephemeral point R_1 to be transmitted separately, increasing communication overhead unnecessarily. Third, the protocol does not provide server-side pseudonymity. The LIA-V2X scheme [4] proposed a lightweight identity authentication protocol based on 256-bit ECC for V2X communication, incorporating ECDH key agreement and anonymous authentication with batch verification support. While LIA-V2X provides strong vehicle anonymity, it focuses exclusively on vehicle-side privacy and does not address server-side identity protection, leaving the infrastructure endpoint identifiable to a passive adversary. The proposed protocol addresses the limitations of all three schemes by introducing dual pseudonymity for both vehicle and server, eliminating unnecessary transmission of R_1 , and providing comprehensive forward secrecy. To the best of our knowledge, the proposed protocol is the first ECC-based IoV authentication scheme to simultaneously protect both endpoint identities with per-session pseudonyms.

3 Cryptographic Preliminaries

3.1 Elliptic Curve Cryptography

Let $\mathbb{F}_p$ be a prime field. An elliptic curve E over $\mathbb{F}_p$ is the set of points (x, y) satisfying the Weierstrass equation: $y^2 \equiv x^3 + ax + b \pmod{p}$, $4a^3 + 27b^2 \not\equiv 0$

together with a point at infinity $\mathcal{O}$. The points on E form an additive abelian group. The scalar multiplication $k \cdot P$ is the core ECC operation. Security relies on the Elliptic Curve Discrete Logarithm Problem (ECDLP):

Definition 1 (ECDLP). *Given points P and $Q = k \cdot P$ on E , it is computationally infeasible to find the integer scalar k . No sub-exponential algorithm is currently known for the ECDLP on generic curves.*

This work utilizes the NIST P-256 (secp256r1) curve, which provides 128-bit security using 256-bit keys. Let G denote the base generator point and q denote the prime group order [1, 15].

The protocol employs SHA-256 as its cryptographic hash function $H(\cdot)$, modeled as a random oracle producing 256-bit outputs. Furthermore, to ensure explicit and unambiguous specification, the protocol utilizes the exclusive-OR (XOR) bitwise operation, denoted by $\oplus$, and string concatenation, denoted by $\|$, to bind identities and temporal parameters securely prior to hashing.

4 System Model and Threat Model

4.1 System Model

The proposed IoV authentication architecture consists of the following principals:

- **Vehicle V :** A registered smart vehicle equipped with a resource-constrained On-Board Unit (OBU) capable of performing basic ECC operations and wireless communication.
- **Server S :** A highly secure, centralized cloud server responsible for vehicle registration, identity management, and mutual authentication.
- **Road Side Units (RSU) and Fog Servers (FS):** FS : Infrastructure nodes deployed at the network edge. In the context of this specific authentication protocol, the RSUs and FSs act strictly as transparent communication relays. They facilitate the routing of messages over the public wireless channel between the Vehicle and the Server but do not possess the cryptographic material required to decrypt payloads, verify tokens, or participate in the session key derivation.

Prior to deployment, the Server registers each vehicle through a secure, out-of-band provisioning step to distribute long-term credentials. During active operation, the authentication path routes from the Vehicle, through the untrusted RSU and FS relays, to the Server.

4.2 Threat Model

This paper adopts the standard Dolev-Yao adversary model. The adversary $\mathcal{A}$ has absolute control over the public wireless communication channel (including the transparent RSU and FS relays). $\mathcal{A}$ can intercept, record, replay, modify, and inject messages. However, $\mathcal{A}$ cannot compromise the secure storage of the Vehicle or Server to extract long-term private keys, nor can $\mathcal{A}$ solve the Elliptic Curve Discrete Logarithm Problem (ECDLP) or break the underlying cryptographic hash functions in polynomial time.

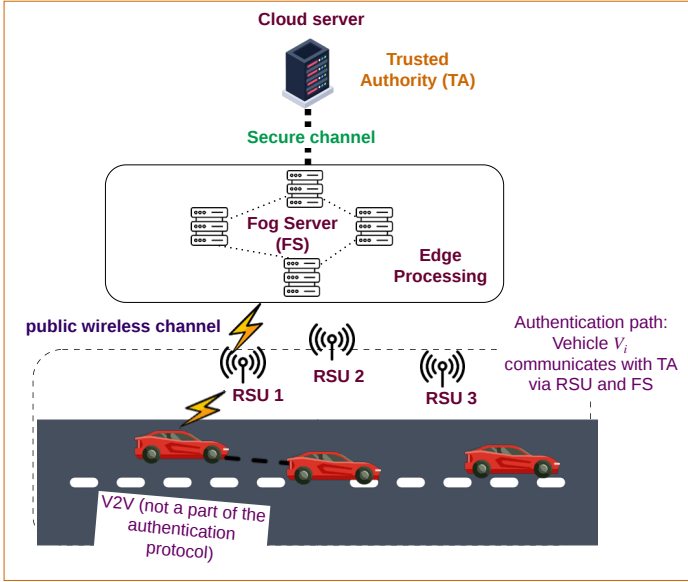


Fig. 1. System model

4.3 Security Goals

1. **Mutual Authentication:** Both parties verify each other.
2. **Session Key Security:** A fresh session key is known only to the two authenticated parties.
3. **Forward Secrecy:** Compromise of long-term keys does not expose previously established session keys.
4. **Replay Resistance:** Captured messages cannot be reused.

Table 1 defines the notation used throughout the paper.

5 Proposed Protocol

5.1 Key Setup and Registration

- **Server setup:** The server selects $r_s \in [1, q - 1]$ and computes $P_s = r_s \cdot G$.
- **Vehicle setup:** The vehicle selects a master private scalar $r_{sg} \in [1, q - 1]$ and computes $P_{sg} = r_{sg} \cdot G$.
- **Registration (one-time, secure channel):** The server issues a partial private key to the vehicle: $h_i = H(ID_v || P_s)$ $s_i = (r_s + h_i \cdot r_{sg}) \pmod{q}$. The tuple (s_i, P_s) constitutes the vehicle's long-term credential. The server retains r_{sg} (keyed by ID_v) for use during authentication. The server public

Table 1. Notation

Symbol	Description	Symbol	Description
ID_v, ID_s	Vehicle and server real identities	r_1, r_2, r_3	Ephemeral random scalars per session
r_s	Server long-term private scalar	R_1	Vehicle ephemeral public point
P_s	Server public key	R'_1	Server challenge point
r_{sg}	Vehicle master private scalar	R_2	Vehicle ephemeral public point
P_{sg}	Vehicle master public key	R_5	Server proof point
s_i	Partial private key issued to vehicle	K	Shared session point
h_i	Hash binding vehicle identity to P_s	τ_i	Timestamp in message i
r_p	Ephemeral nonce for identity masking	Δ	Maximum acceptable clock skew
R_v	Vehicle ephemeral masking point	SK	Shared session key
K_{mask}	Masking key for vehicle identity	$H(\cdot)$	SHA-256 hash function
C_v	Encrypted vehicle identity	$\oplus$	XOR operation

key P_s is distributed to all vehicles via a secure out-of-band channel prior to deployment.

- **Vehicle identity masking:** To protect ID_v on the channel, the vehicle uses asymmetric masking based on the server's public key. Before each session, the vehicle picks a fresh ephemeral scalar $r_p \in [1, q - 1]$ and computes: $R_v = r_p \cdot GK_{mask} = H(r_p \cdot P_s)C_v = ID_v \oplus K_{mask}$.

Only the server can recover ID_v by computing $K_{mask} = H(r_s \cdot R_v)$ and then $ID_v = C_v \oplus K_{mask}$, since $r_s \cdot R_v = r_s \cdot r_p \cdot G = r_p \cdot P_s$. An eavesdropper observing C_v and R_v cannot recover ID_v without r_s , which requires solving the ECDLP.

5.2 Authentication Protocol

Fig. 2 illustrates the full protocol flow. Every message carries a timestamp τ_i . Upon receipt, the receiver verifies $|T_{current} - \tau_i| < \Delta$. If this check fails, the message is silently discarded.

Message 1 ($V \rightarrow S$): The vehicle computes the masked identity C_v along with its ephemeral masking point and sends: $M_1 = \langle C_v, R_v, ID_s, \tau_1 \rangle$ (Note: ID_s acts as a routing identifier for the public infrastructure. The ephemeral point R_1 has been eliminated from this message to reduce communication overhead).

Server decrypts ID_v and computes challenge: Upon receiving M_1 , the server decrypts the identity: $K_{mask} = H(r_s \cdot R_v)ID_v = C_v \oplus K_{mask}$. It then looks up r_{sg} from its database using the recovered ID_v . To initiate the session key derivation, the server selects a fresh ephemeral scalar $r_1 \in [1, q - 1]$, computes $R_1 = r_1 \cdot G$, and generates the challenge point $R'_1 = r_s \cdot R_1$. The server securely binds this challenge to the vehicle's request using a hash token: $A_0 = H(ID_v || R'_1 || R_v || \tau_2)$.

Message 2 ($S \rightarrow V$): $M_2 = \langle A_0, R'_1, \tau_2 \rangle$.

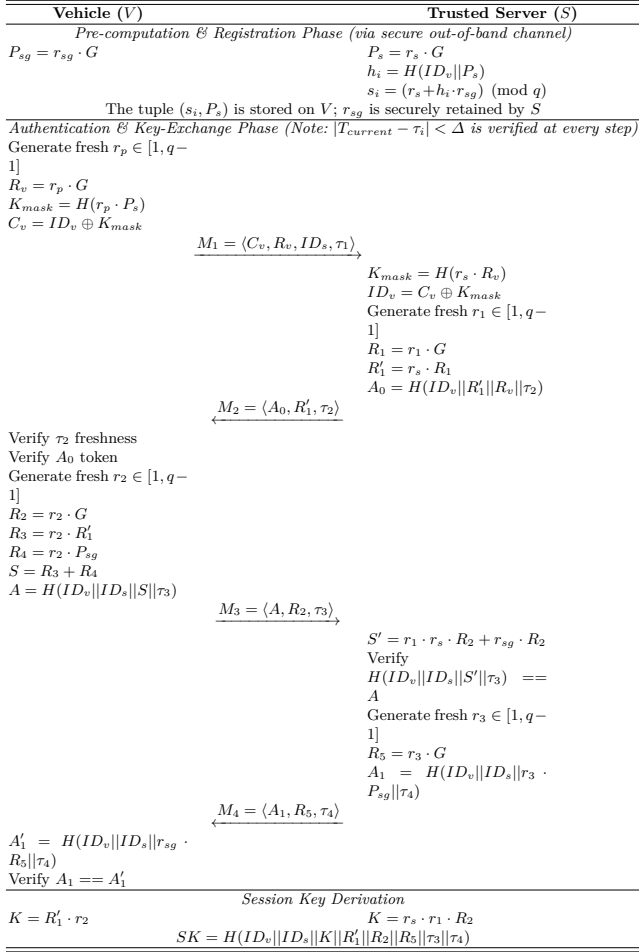


Fig. 2. Proposed ECC-based mutual authentication and key-exchange protocol

Message 3 ($V \rightarrow S$): The vehicle receives M_2 , verifies the timestamp, and validates the token A_0 to ensure the server correctly processed its masked identity. The vehicle then selects its session ephemeral scalar $r_2 \in [1, q - 1]$ and computes $R_2 = r_2 \cdot G$. To prove possession of its master private scalar, it computes: $R_3 = r_2 \cdot R'_1$, $R_4 = r_2 \cdot P_{sg}$, $S = R_3 + R_4$, $A = H(ID_v || ID_s || S || \tau_3)$ and sends $M_3 = \langle A, R_2, \tau_3 \rangle$.

Server verification of M_3 : The server receives R_2 and reconstructs the proof using its knowledge of r_1 and r_s : $S' = r_1 \cdot r_s \cdot R_2 + r_{sg} \cdot R_2$. By scalar multiplication linearity: $S' = (r_1 r_s + r_{sg}) \cdot R_2 = r_2 \cdot (r_s \cdot R_1 + P_{sg}) = r_2 \cdot R'_1 + r_2 \cdot P_{sg} = S$. The server accepts the authentication if $H(ID_v || ID_s || S' || \tau_3) = A$.

Message 4 ($S \rightarrow V$): The server selects a final ephemeral scalar $r_3 \in [1, q - 1]$, computes $R_5 = r_3 \cdot G$, and: $A_1 = H(ID_v || ID_s || r_3 \cdot P_{sg} || \tau_4)$ and sends $M_4 =$

$\langle A_1, R_5, \tau_4 \rangle$. Vehicle verification of $M_4: A'_1 = H(ID_v || ID_s || r_{sg} \cdot R_5 || \tau_4)$. Since $r_{sg} \cdot R_5 = r_{sg} \cdot r_3 \cdot G = r_3 \cdot P_{sg}$, the check $A'_1 = A_1$ succeeds only if the server is genuine. Mutual authentication is complete.

5.3 Session Key Derivation

Both parties independently compute the shared point: $K = r_s \cdot r_1 \cdot r_2 \cdot G$. The vehicle computes $K = R'_1 \cdot r_2 = r_s \cdot r_1 \cdot r_2 \cdot G$ using R'_1 received in M_2 and its own r_2 . The server computes $K = r_s \cdot r_1 \cdot R_2 = r_s \cdot r_1 \cdot r_2 \cdot G$ using R_2 received in M_3 and its own r_s and r_1 . Both yield the same point without ever transmitting K on the open channel. The session key is then derived as: $SK = H(ID_v || ID_s || K || R'_1 || R_2 || R_5 || \tau_3 || \tau_4)$. Because r_1 and r_2 are ephemeral and K is never transmitted, SK achieves forward secrecy. The inclusion of both identities and all session-specific points ensures SK is unique to each session.

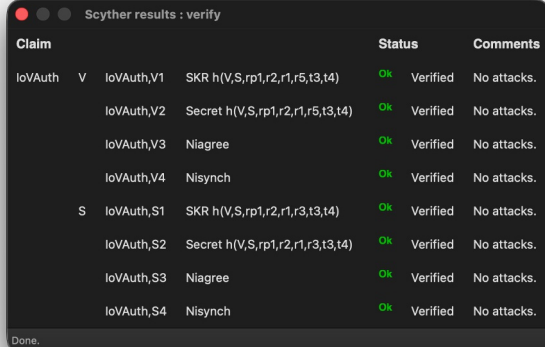
6 Security Analysis

6.1 Formal Verification with Scyther

To provide rigorous, empirical validation of the informal claims, the proposed protocol was modeled in the Security Protocol Description Language (SPDL) and analyzed using the Scyther formal verification tool [17]. The tool evaluated the protocol against a standard Dolev-Yao adversary capable of arbitrary message interception, modification, and injection [16]. Scyther systematically verified eight critical security claims across both the Vehicle and Server roles, including Session Key Secrecy (SKR), Secret, Non-injective Agreement (Nia-gree), and Non-injective Synchronization (Nisynch). During the iterative modeling phase, Scyther's unbounded multi-protocol analysis initially identified a parallel session mixing vulnerability where an adversary could force a session mismatch by mixing M_1 transmissions from different runs. This vulnerability was successfully remediated in the final design by introducing the hash token $A_0 = H(ID_v || R'_1 || R_v || \tau_2)$ in Message 2. This token cryptographically binds the server's newly generated challenge (R'_1) directly to the vehicle's masked identity and ephemeral request, neutralizing the mixing vector. Following this architectural update, Scyther verified all claims with no attacks found under an unbounded number of concurrent sessions. Fig. 3 summarises the final verification results. There is also an While Scyther provides profound assurances of logical routing security, it assumes cryptographic primitives function as perfect black boxes. Future work will expand this threat model to include physical side-channel evaluations and formal proofs utilizing the extended Canetti and Krawczyk (eCK) model to analyze resilience against Ephemeral Secret Leakage (ESL) attacks.

6.2 Informal Analysis

The protocol's resilience is evaluated against a Dolev-Yao adversary.



Scyther results : verify

Claim	Status	Comments
loVAuth V loVAuth,V1 SKR $h(V,S,rp1,r2,r1,r5,t3,t4)$	Ok Verified	No attacks.
loVAuth,V2 Secret $h(V,S,rp1,r2,r1,r5,t3,t4)$	Ok Verified	No attacks.
loVAuth,V3 Niagree	Ok Verified	No attacks.
loVAuth,V4 Nisynch	Ok Verified	No attacks.
S loVAuth,S1 SKR $h(V,S,rp1,r2,r1,r3,t3,t4)$	Ok Verified	No attacks.
loVAuth,S2 Secret $h(V,S,rp1,r2,r1,r3,t3,t4)$	Ok Verified	No attacks.
loVAuth,S3 Niagree	Ok Verified	No attacks.
loVAuth,S4 Nisynch	Ok Verified	No attacks.

Done.

Fig. 3. Scyther formal verification output

1. **Replay Attack:** Every message carries a timestamp τ_i verified within window Δ . A replayed message outside this window is rejected, ensuring freshness.
2. **Impersonation Attack** Forging M_3 requires computing $S = r_2 \cdot R'_1 + r_2 \cdot P_{sg}$. The term $r_2 \cdot P_{sg}$ requires knowledge of r_{sg} ; recovering it from P_{sg} is the ECDLP. Forging M_4 requires computing $r_3 \cdot P_{sg}$ for a self-chosen $R_5 = r_3 \cdot G$, again an ECDLP instance.
3. **Man-in-the-Middle Attack** Tokens A and A_1 cryptographically bind both pseudonyms, ephemeral points, and timestamps. Without the long-term secrets r_s or r_{sg} an adversary cannot forge valid tokens.
4. **Eavesdropping and Session Key Security** The shared point $K = r_s \cdot r_1 \cdot r_2 \cdot G$ is never transmitted. Recovering K from the observed values R'_1 and R_2 requires solving the Computational Diffie-Hellman (CDH) problem, which is at least as hard as the ECDLP.
5. **Vehicle and Server Tracking** The vehicle's real identity ID_v is encrypted as $C_v = ID_v \oplus K_{mask}$ using an ephemeral masking key $K_{mask} = H(r_p \cdot P_s)$ that changes every session. An eavesdropper observing C_v and R_v cannot recover ID_v without solving the ECDLP to obtain r_s from P_s . The server identity ID_s is sent in plaintext in M_1 , which is a deliberate design choice since roadside infrastructure endpoints are typically publicly known. The vehicle is fully unlinkable across sessions since r_p is fresh each time.
6. **Known Session Key Attack** All ephemeral scalars r_1, r_2, r_3 are independently generated per session. Consequently SK is statistically independent across sessions: compromising one session key provides no information about others.
7. **Forward Secrecy** Even if long-term keys r_s or r_{sg} are later compromised, past session keys cannot be recovered because the ephemeral scalars are discarded and K is never exposed on the channel.

7 Performance Evaluation

Table 2. Summary of Performance Evaluation Metrics per Protocol Run

Evaluation Category	Parameter	Value
Cryptographic Operations	Vehicle (ScM / PA / H)	4 / 1 / 4
Cryptographic Operations	Server (ScM / PA / H)	3+ / 1 / 3
Cryptographic Operations	Total (ScM / PA / H)	7+ / 2 / 7
Execution Time	User CPU / System CPU	≈ 21.2 ms / ≈ 3.73 ms
Execution Time	Total CPU / Wall-clock	≈ 11.9 ms / $\approx 11.9 \pm 0.48$ ms
Communication Overhead	M_1 ($V \rightarrow S$) / M_2 ($S \rightarrow V$)	72 bytes / 73 bytes
Communication Overhead	M_3 ($V \rightarrow S$) / M_4 ($S \rightarrow V$)	104 bytes / 104 bytes
Communication Overhead	Total Payload	353 bytes

7.1 Computational Complexity

To evaluate the computational overhead, a prototype of the proposed protocol was implemented utilizing the *pycryptodome* cryptographic library in Python. The implementation executed ECC operations over the NIST P-256 curve and utilized SHA-256 for all cryptographic hashing. To isolate the true cryptographic execution cost and eliminate application initialization latency, the performance was benchmarked across multiple batch iterations to establish a deterministic baseline. The benchmarking results demonstrate a total CPU execution time of approximately 11.9 ms, with a negligible standard deviation of ± 0.485 ms. As detailed in Table 2, the computational cost is dominated by the required scalar multiplications (SCM), while point additions (PA) and hash evaluations (H) contribute minimal latency. This ≈ 11.9 ms total execution time confirms that the protocol operates well within the real-time constraints required for safety-critical IoV communications on resource-constrained ECUs as shown in Table 2.

7.2 Communication Overhead

Bandwidth efficiency is critical in high-density V2X environments. Assuming an identity string (ID) length of 4 bytes and a timestamp (τ) length of 4 bytes, the payload sizes can be deterministically calculated. Utilizing ECC point compression, each transmitted point (R_v, R'_1, R_2, R_5) occupies 33 bytes. Each SHA-256 hash token (A_0, A, A_1) and masking block (C_v) occupies 32 bytes. By shifting the generation of the initial key derivation scalar r_1 to the server, the proposed protocol effectively eliminates the transmission of R_1 from the vehicle's initial request. Message 1 (M_1) strictly transmits the 32-byte masked identity C_v , the 33-byte masking point R_v , a 4-byte server routing identity ID_s , and a 4-byte timestamp,

resulting in a highly compact 73-byte initial payload. The total communication overhead across all four rounds is approximately 353 bytes as shown in Table 2. This minimal footprint fits comfortably within the Maximum Transmission Unit (MTU) payload limits of IEEE 802.11p and C-V2X (5G PC5) standards without requiring packet fragmentation. Table 3 compares the proposed protocol against related work.

Table 3. Comparison of Performance and Security Properties

Protocol	ScM	Rds	M.A.	V-Priv	S-Priv	Fwd.Sec.	Replay	MitM
Vasudev [2]	5	3	✓	×	×	×	✓	✓
Tewari [5]	6	4	✓	×	×	✓	✓	✓
LIA-V2X [4]	6	3	✓	✓	×	✓	✓	✓
Proposed	7+	4	✓	✓	✓	✓	✓	✓

ScM=ECC scalar multiplications, Rds=Communication Rounds, M.A.=Mutual Auth, V-Priv=Vehicle Privacy, S-Priv=Server Privacy, Fwd.Sec.=Forward Secrecy

8 Conclusion

This paper proposed and formally verified a secure, lightweight ECC-based mutual authentication and key-exchange protocol for the Internet of Vehicles (IoV). By extending existing lightweight frameworks to include an asymmetric identity masking technique, the proposed protocol successfully protects the vehicle’s identity from topological tracking over open wireless channels. Furthermore, the redesign of the session key derivation—achieving a forward-secure shared secret via $K = r_s \cdot r_1 \cdot r_2 \cdot G$ optimizes communication overhead. Rigorous formal verification using the Scyther tool under the Dolev-Yao model confirmed the protocol’s resilience against replay, impersonation, and parallel session mixing attacks across an unbounded number of concurrent sessions. Critically, the protocol achieved these security guarantees with a total CPU average execution time of approximately 11.9 milliseconds. This minimal computational footprint demonstrates that the protocol is highly practical and readily deployable on the resource-constrained ECUs standard in modern vehicular networks. Future work will focus on implementing and benchmarking the protocol on embedded hardware architectures, such as the ARM Cortex-M. Additionally, subsequent iterations of this research will explore advanced threat models, including resilience against ESL attacks , and the potential integration of PQC primitives to ensure long-term security against emerging quantum computing capabilities.

References

1. H. Shekhawat and D. S. Gupta, "Quantum-safe lattice-based mutual authentication and key-exchange scheme for the smart grid," in *Proc. 15th Int. Conf. Computing Communication and Networking Technologies (ICCCNT)*, IEEE, 2024, DOI: 10.1109/ICCCNT61001.2024.10723943. Also available in *Trans. Emerging Telecommun. Technol.*, vol. 35, no. 7, p. e5017, 2024.
2. H. Vasudev, V. Deshpande, D. Das, and S. K. Das, "A lightweight mutual authentication protocol for V2V communication in Internet of Vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 6709–6717, Jun. 2020.
3. V. Kumar, A. Kumari, S. Jangirala, and H. Pratap, "Design flaws and cryptanalysis of a lightweight mutual authentication protocol for V2V communication in Internet of Vehicles," in *Proc. Int. Conf. Advanced Computing and Intelligent Technologies (ICACIT)*, Springer, 2022, pp. 125–135.
4. Y. Liu, X. Zhang, and W. Chen, "LIA-V2X: A lightweight identity mutual and batch authentication protocol for V2X communication in Internet of Vehicles," *arXiv preprint arXiv:2312.14895*, 2023.
5. A. Tewari and B. B. Gupta, "A lightweight mutual authentication protocol based on elliptic curve cryptography for IoT devices," *Int. J. Advanced Intelligence Paradigms*, vol. 9, nos. 2/3, pp. 111–121, 2017.
6. L. Cai, Y. Liu, T. Tian, H. Zhang, and X. Pan, "A survey of the internet of vehicles: Architecture, enabling technologies, and test fields," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4131–4155, 2021.
7. D. J. Bernstein and T. Lange, "Post-quantum cryptography," *Nature*, vol. 549, pp. 188–194, 2017.
8. M. Azees, P. Vijayakumar, and L. J. Deboarh, "EAAP: Efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2467–2476, 2017.
9. M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *Journal of Computer Security*, vol. 15, no. 1, pp. 39–68, 2007.
10. D. Hankerson, A. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*. Springer, 2004.
11. C. Zhang, P.-H. Ho, and J. Tapolcai, "On batch verification with group communications," *Wireless Networks*, vol. 17, no. 8, pp. 1851–1865, 2011.
12. J. Shen, T. Zhou, X. Chen, J. Li, and W. Susilo, "Anonymous and traceable group data sharing in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 4, pp. 912–925, 2018.
13. J. Zhang, H. Zheng, Z. Guo, and H. Cui, "A new lightweight authentication scheme for IoV in 5G networks," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 9099–9109, 2019.
14. X. Li, J. Liu, M. S. Obaidat, P. Nepal, and R. Amin, "A privacy preserving three-factor authentication protocol for e-health clouds," *The Journal of Supercomputing*, vol. 76, pp. 3826–3849, 2020.
15. National Institute of Standards and Technology, "Digital Signature Standard (DSS)," *FIPS PUB 186-4*, 2013.
16. D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.
17. C. J. F. Cremers, "The Scyther Tool: Verification, falsification, and analysis of security protocols," in *Proc. Int. Conf. Computer Aided Verification (CAV)*, 2008, pp. 414–418.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

