The Journal of Supercomputing (2026) 82:400
https://doi.org/10.1007/s11227-026-08570-x

Comment on "An advanced blockchain-based mutual authentication technique for the Internet of vehicles environment": cryptanalysis and security vulnerabilities

Yashar Salami[1]

Received: 17 March 2026 / Accepted: 28 April 2026
© The Author(s) 2026

Abstract

The Internet of Vehicles (IoV), in which vehicles exchange safety–critical information over open wireless channels, is characterized by large-scale, distributed operation, and stringent real-time constraints, requiring efficient and reliable communication mechanisms. Recently, Suyel et al. proposed a blockchain-based mutual authentication and key agreement protocol for IoV environments (The Journal of Supercomputing, 2025, https://doi.org/10.1007/s11227-025-07934-z), claiming resistance against various attacks under the Dolev–Yao adversarial model. In this paper, we revisit the security of this protocol and demonstrate that its V2V authentication mechanism suffers from a fundamental design weakness. Specifically, the protocol verifies only the algebraic consistency of the received authentication tuple while accepting the verification key directly from the communication channel, without establishing a trustworthy binding between the claimed vehicle identity and the corresponding public key. As a consequence, an adversary can construct mathematically valid authentication messages using its own key pair while impersonating a legitimate vehicle. Based on this observation, we present several practical attacks, including vehicle impersonation, rogue public-key substitution, and message-substitution attacks. Furthermore, we show that the protocol relies on an auxiliary secure channel for final key confirmation, which contradicts the claimed Dolev–Yao threat model and weakens the overall security guarantees. To substantiate our findings, we provide a rigorous cryptanalytic evaluation combining algebraic validity analysis and BAN-logic reasoning. Our results reveal that the protocol fails to ensure reliable vehicle authentication and cannot securely attribute the established session key to the claimed peer identity. Consequently, the security claims of the target scheme do not hold under realistic adversarial conditions in IoV environments.

Keywords IoV · Dolev–Yao model · Authentication · Impersonation

✉ Yashar Salami
yashar.salami@kapadokya.edu.tr

[1] Faculty of Computer and Information Technologies, Cappadocia University, Ürgüp, Nevsehir, Turkey

Published online: 06 May 2026

1 Introduction

The Internet of Vehicles (IoV) has emerged as a crucial component of intelligent transportation systems, enabling vehicles to communicate with other vehicles, roadside infrastructure, and cloud services to improve road safety, traffic efficiency, and the driving experience [1–3]. Due to the large volume of concurrent interactions and the need for low-latency responses, IoV systems operate under strict real-time constraints and increasingly depend on cloud-based high-performance computing (HPC) and scalable distributed processing to maintain efficiency and reliability [4–6]. However, because IoV communication relies on open wireless channels, transmitted messages are inherently vulnerable to security threats such as replay, impersonation, and man-in-the-middle attacks [7, 8]. Therefore, designing secure and efficient authentication and key agreement protocols is essential to ensure the reliability and trustworthiness of IoV communication systems [9, 10].

To address these challenges, numerous authentication and key agreement protocols have been proposed for vehicular networks. Several schemes employ elliptic-curve cryptography (ECC) to achieve lightweight security for resource-constrained vehicular devices, while others integrate emerging technologies, such as blockchain, to enhance decentralization and trust management [11–13]. These approaches aim to provide mutual authentication, privacy preservation, and secure session-key establishment between communicating entities in IoV environments [14].

Very recently, Suyel et al. proposed an advanced blockchain-based mutual authentication and key agreement protocol for IoV environments [15]. In their scheme, a consortium blockchain is used to maintain distributed trust among multiple authorities, while roadside units (RSUs) and zone managers assist in authentication and validation procedures. The protocol combines ECC-based cryptographic operations with blockchain-assisted verification and claims to provide secure, efficient authentication for large-scale vehicular networks. Furthermore, the authors argue that the scheme can withstand a variety of well-known attacks, including replay, man-in-the-middle, and impersonation attacks.

However, upon careful examination of the authentication procedure, we observe several security weaknesses. In particular, the design of the authentication mechanism allows an adversary to manipulate authentication parameters transmitted over the public communication channel. As a result, the scheme fails to provide the intended security guarantees under the Dolev–Yao adversarial model. Moreover, the protocol relies on an auxiliary secure channel for the final key-confirmation step, which contradicts the claimed threat model. In this paper, we demonstrate that the protocol cannot reliably achieve secure vehicle authentication and is vulnerable to several attacks, including vehicle impersonation and public-key substitution attacks.

1.1 Contributions

The main contributions of this paper are summarized as follows:

- We revisit the V2V authentication phase of the target protocol and identify a fundamental design weakness in its verification mechanism, namely the absence of a trusted binding between the claimed temporary identity and the submitted public key.
- We show that, under the Dolev–Yao adversarial model, the target protocol is vulnerable to active attacks, including vehicle impersonation, rogue public-key substitution, and message-substitution/man-in-the-middle feasibility attacks.
- We provide a rigorous algebraic analysis demonstrating that an adversary can construct mathematically valid authentication tuples under its own key pair while falsely claiming another vehicle's identity, thereby satisfying the protocol verification equation without possessing the legitimate vehicle's credentials.
- We further support the cryptanalytic findings through BAN-logic reasoning and show that the exchanged protocol messages do not provide sufficient evidence to establish the intended authentication and session-key beliefs.
- We demonstrate that the claimed security guarantees of the target protocol are not achieved under realistic adversarial conditions, and that its reliance on an auxiliary secure channel creates a mismatch between the protocol design and its claimed Dolev–Yao threat model.

1.2 Paper organization

The remainder of this paper is organized as follows. Section 2 presents the system model and the notations used in the target protocol. Section 3 briefly reviews the authentication procedure of the target scheme. Section 4 describes the main security weaknesses and demonstrates several attacks against the protocol. Section 5 provides the informal and formal analyses, including algebraic validity analysis and BAN-logic reasoning and discusses the identified security weaknesses. Finally, Sect. 6 concludes the paper.

2 System model

In this section, we present the network model and the notations used in the target protocol.

2.1 Network model

Figure 1 illustrates the system model of the target protocol. The architecture consists of four main entities: vehicles, RSUs, zone managers (ZMs), and trusted authorities (TAs) connected through a blockchain network. Vehicles communicate with nearby vehicles and RSUs inside each zone. RSUs collect and forward local information, while each ZM supervises the RSUs within its corresponding zone. At the upper layer, multiple TAs participate in the blockchain network and

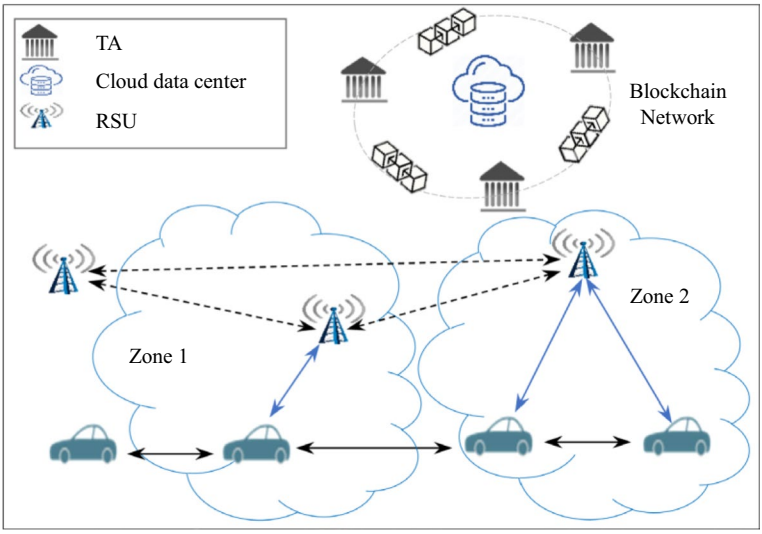

Fig. 1 Network model

maintain the distributed ledger. This model is intended to support decentralized authentication and coordination in the IoV environment.

3 Notations

Table 1 shows the notations used in the target protocol.

Table 1 Nantion used the paper

Notation	Description
$H(\cdot)$	Hash function
$Sign_{V1}$ and TS_{V1}	Signature and current timestamp of V_1
TC_V	Temporary credential of V
TID_{V1} and TID_{V2}	Temporary identity of V_1 and V_2
HPrev and HCurr	Previous and current block hash
GID_{RSU}, GID_{ZM}, and GID_V	General identity of RSU, ZM, and V
(PR_V, PU_V)	Private and public keys of the vehicle
PRRSU, PURSU)	Private and public keys of RSU
PRZM, PUZM)	Private and public key of ZM
RTSRSU, RTSZM, and RTSV	Registration timestamp of RSU, ZM, and V
MR_{TN}	Merkle root of transactions in a block
MSK_T	Master secret key of the trusted authority

4 Review of Suyel et al.'s protocol

The target protocol uses elliptic-curve cryptography over. $E_q(a, b)$ with base point G and a hash function $H(\cdot)$. The trusted authority TAuth selects the master secret key $MSK_T \in Z_q^*$ and publishes the public parameters $\{E_q(a, b), G, H(\cdot)\}$. Using these parameters, the scheme supports registration and authentication among vehicles, RSUs, zone managers, and trusted authorities. Figure 2 shows the main message flow of the target protocol.

5 Cryptanalysis of the target protocol

In this section, we present the cryptanalysis of the target protocol. We first outline the problem statement and then demonstrate several attacks against the V2V authentication mechanism, including vehicle impersonation and rogue public-key substitution attacks.

5.1 Problem statement

The protocol proposed by Suyel et al. claims security under the Dolev–Yao model, in which an adversary is assumed to have full control over the communication channel, including the ability to intercept, modify, inject, and replay transmitted messages. However, a careful analysis of the protocol reveals an inconsistency between the claimed threat model and the protocol's operational assumptions. In particular, during the V2V authentication phase, the protocol relies on a secure auxiliary channel to deliver the final key-confirmation message, while several critical authentication parameters such as PU_{V_1}, M_{V_1}, and $Sign_{V_1}$ are accepted directly from the public communication channel. Consequently, the authentication process is not established in a fully self-contained manner over the public network. If the assumed secure auxiliary channel is unavailable, an active adversary operating under the Dolev–Yao model can manipulate the exchanged authentication parameters. This design weakness enables several attacks, including vehicle impersonation, rogue public-key substitution, and message-substitution attacks. Therefore, the security guarantees claimed by the protocol cannot be reliably achieved in realistic vehicular communication environments where the existence of an external secure channel cannot generally be ensured.

5.2 Vehicle impersonation attack

During the V2V authentication phase, the initiating vehicle V_1 sends the authentication message $\{TID_{V_1}, TS_{V_1}, PU_{V_1}, M_{V_1}, Sign_{V_1}\}$. Upon receiving this message, the receiving vehicle V_2 verifies the request by checking the following equation $Sign_{V_1} \cdot G = M_{V_1} + H\left(TID_{V_1} \| TS_{V_1} \| PU_{V_1}\right) \cdot PU_{V_1}$. This verification mechanism

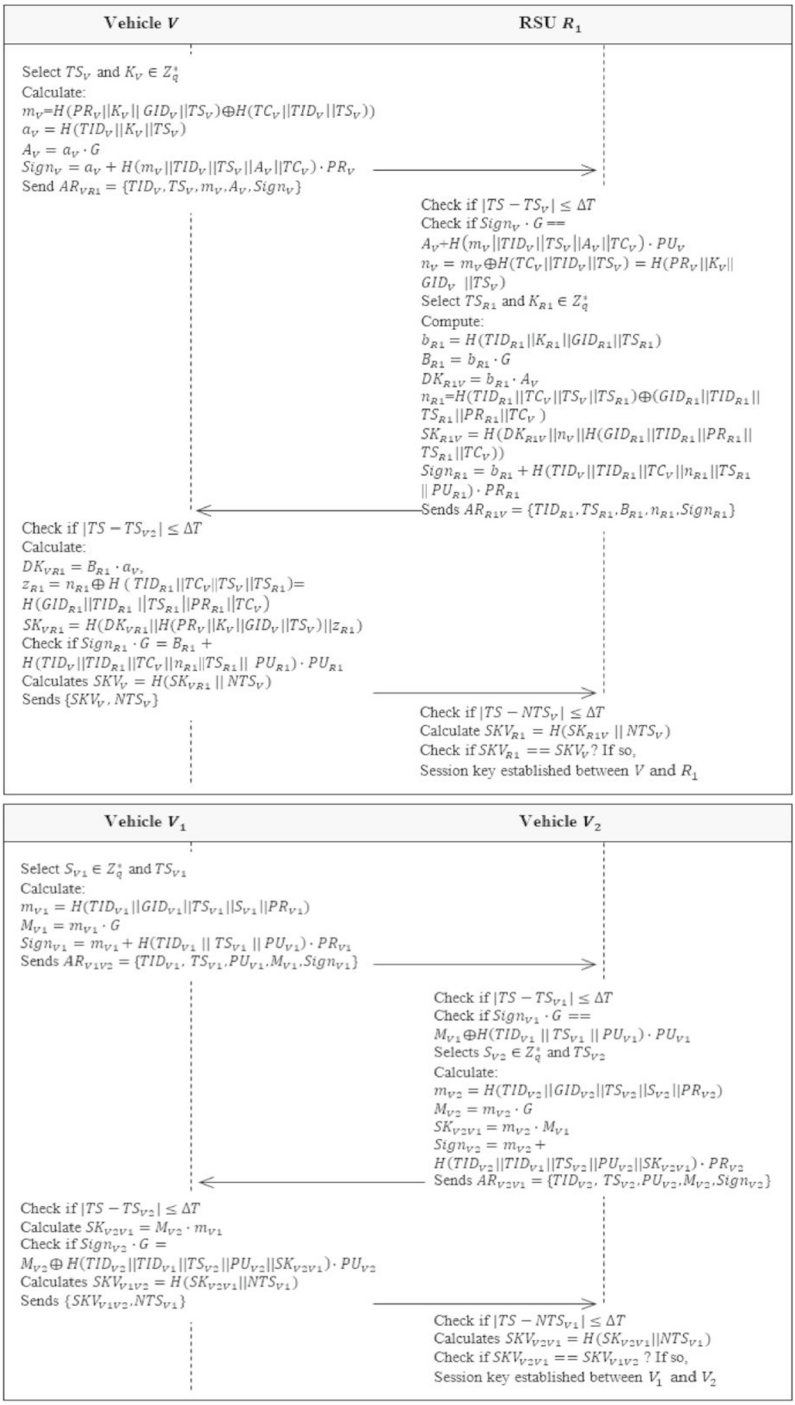

Fig. 2 Suyel et al.'s scheme

indicates that the acceptance of the authentication request depends entirely on the algebraic consistency of the received tuple. In particular, the verification procedure uses the public-key PU_{V_1} carried within the same message. It does not independently verify whether this key is the legitimately registered key associated with the claimed temporary identity TID_{V_1}. Consequently, the receiver only checks whether the received values satisfy the elliptic-curve equation, rather than confirming that the key used in the equation actually belongs to the claimed vehicle. This property enables an adversary to construct a mathematically consistent authentication tuple under its own key pair while falsely claiming the identity of another vehicle.

Step 1: adversary key generation

The adversary A First, it generates its own key pair by selecting a random private scalar s_A and computing the corresponding public-key $PU_{V_A} = s_A G$. The adversary then selects a fresh timestamp. TS_{V_A} and falsely claims the temporary identity TID_{V_1} of the target vehicle.

Step 2: Construction of the forged authentication tuple

Next, the adversary selects a random ephemeral scalar. m_A and computes the elliptic-curve point $M_{V_A} = m_A G$. Using its private key s_A, The adversary constructs the forged signature value $Sign_{V_A} = m_A + H\left(TID_{V_1} \| TS_{V_A} \| PU_{V_A}\right) \cdot s_A$. This signature follows the same algebraic structure required by the protocol verification equation.

Step 3: Transmission of the forged authentication request

The adversary sends the forged authentication request $\left\{TID_{V_1}, TS_{V_A}, PU_{V_A}, M_{V_A}, Sign_{V_A}\right\}$ to the receiving vehicle V_2.

Step 4: Signature verification at the receiver

Upon receiving the forged request, the vehicle V_2 checks the freshness of the timestamp and verifies the signature equation using the received public-key PU_{V_A}. During verification, the left-hand side of the equation becomes $Sign_{V_A} \cdot G$, which equals $\left(m_A + H\left(TID_{V_1} \| TS_{V_A} \| PU_{V_A}\right) \cdot s_A\right)G$. By the distributive property of scalar multiplication over elliptic-curve points, this expression expands to $m_A G + H\left(TID_{V_1} \| TS_{V_A} \| PU_{V_A}\right) \cdot s_A G$. Since $m_A G = M_{V_A}$ and $s_A G = PU_{V_A}$, the verification equation becomes $Sign_{V_A} \cdot G = M_{V_A} + H\left(TID_{V_1} \| TS_{V_A} \| PU_{V_A}\right) \cdot PU_{V_A}$. Therefore, the verification equation holds exactly, even though the public key used in the verification process belongs to the adversary rather than to the legitimate vehicle associated with the claimed identity. TID_{V_1}. As a result, the vehicle V_2 accepts the forged authentication tuple as valid.

Step 5: Session-key establishment

After successful verification, the vehicle V_2 proceeds with the protocol execution and computes the session-key contribution using its own ephemeral secret together with the adversary-generated value M_{V_A}. It then sends the response message to the claimed initiator.

Step 6: Session-key derivation and successful impersonation

Because the adversary knows the scalar m_A used to construct M_{V_A} It can derive the same session-key material from the response message sent by V_2.

Consequently, both the adversary and V_2 compute an identical session key. Since the forged authentication request was generated under the claimed identity TID_{V_1}, vehicle V_2 concludes that the established session key is shared with the legitimate vehicle V_1. In reality, however, the session has been established with the adversary A. Therefore, the adversary successfully impersonates the legitimate vehicle and establishes a valid session with the responder under a false identity.

5.3 Rogue public-key attack

During the V2V authentication phase, the receiving vehicle $\mathbf{V}_2$ validates the incoming authentication request by checking the following equation $Sign_{V_1} \cdot G = M_{V_1} + H\left(TID_{V_1} \| TS_{V_1} \| PU_{V_1}\right) \cdot PU_{V_1}$, where the public-key PU_{V_1} is taken directly from the received message. This verification rule implies that the validity of the authentication request depends solely on the algebraic consistency of the transmitted tuple. In particular, the verifier does not independently confirm whether the received public key corresponds to the legitimately registered key associated with the claimed temporary identity TID_{V_1}. As a consequence, the protocol verifies only that the received values satisfy the elliptic-curve verification equation. Still, it does not guarantee that the key used in this equation actually belongs to the claimed vehicle identity. This property allows an adversary to substitute its own public key for the legitimate one while still satisfying the verification equation, thereby enabling a rogue public-key attack.

Step 1: Adversary key generation
The adversary **A** first generates its own elliptic-curve key pair by selecting a random private scalar $\mathbf{s_A}$ and computing the corresponding public-key $PU_{V_A} = s_A G$. This key pair is fully controlled by the adversary.
Step 2: Identity claim and timestamp selection
Next, the adversary selects the temporary identity TID_{V_1} of a target legitimate vehicle and generates a fresh timestamp TS_{V_A}. At this stage, the adversary prepares to construct an authentication request that appears to originate from the legitimate vehicle.
Step 3: Construction of the rogue authentication tuple
The adversary selects a random ephemeral scalar $\mathbf{m_A}$ and computes the elliptic-curve point $M_{V_A} = m_A G$. Using its private key $\mathbf{s_A}$, the adversary constructs the forged signature $Sign_{V_A} = m_A + H\left(TID_{V_1} \| TS_{V_A} \| PU_{V_A}\right) \cdot s_A$. This signature follows the same algebraic structure required by the protocol verification equation.
Step 4: Transmission of the rogue authentication request
The adversary sends the forged authentication request $\left\{TID_{V_1}, TS_{V_A}, PU_{V_A}, M_{V_A}, Sign_{V_A}\right\}$ to the receiving vehicle $\mathbf{V}_2$.
Step 5: Signature verification at the receiver
Upon receiving the forged message, the vehicle V_2 verifies the freshness of the timestamp and checks the signature equation using the received public-key PU_{V_A}.

During verification, the left-hand side of the equation becomes $\mathbf{Sign_{V_A}} \cdot \mathbf{G}$, which equals $\left(m_A + H\left(TID_{V_1} \| TS_{V_A} \| PU_{V_A}\right) \cdot s_A\right)G$. By the distributive property of scalar multiplication over elliptic-curve points, this expression expands to $\mathbf{m_A G} + \mathbf{H}\left(\mathbf{TID_{V_1}} \| \mathbf{TS_{V_A}} \| \mathbf{PU_{V_A}}\right) \cdot \mathbf{s_A G}$. Since $\mathbf{m_A G} = \mathbf{M_{V_A}}$ and $\mathbf{s_A G} = \mathbf{PU_{V_A}}$, the equation becomes $\mathbf{Sign_{V_A}} \cdot \mathbf{G} = \mathbf{M_{V_A}} + \mathbf{H}\left(\mathbf{TID_{V_1}} \| \mathbf{TS_{V_A}} \| \mathbf{PU_{V_A}}\right) \cdot \mathbf{PU_{V_A}}$. Therefore, the verification equation holds exactly, even though the public key used in the verification process is controlled by the adversary. As a result, vehicle $\mathbf{V}_2$ accepts the forged authentication request as valid.

Step 6: Successful public-key substitution

After successful verification, the vehicle $\mathbf{V}_2$ proceeds with the protocol execution using the adversary-supplied public-key $\mathbf{PU_{V_A}}$ and ephemeral value $\mathbf{M_{V_A}}$. Because the verification mechanism confirms only the algebraic correctness of the received tuple, the verifier assumes that these values belong to the legitimate vehicle identified by $\mathbf{TID_{V_1}}$.

Consequently, the adversary successfully substitutes its own public key for the claimed vehicle identity, and the verifier fails to detect that the submitted key does not belong to the genuinely registered vehicle.

5.4 Message-substitution/MITM feasibility attack

The target protocol assumes that an adversary can intercept, modify, inject, and replay messages over the public communication channel. However, during the V2V authentication phase, the exchanged public parameters are accepted directly from the communication medium without an explicit authenticated binding to the claimed temporary vehicle identity. In addition, the final key-confirmation message is assumed to be transmitted through a secure auxiliary channel.

This design introduces a structural weakness. If such a secure auxiliary channel is unavailable, the protocol cannot prevent an adversary from intercepting and substituting authentication messages. As a result, the adversary can manipulate the authentication exchange and create conditions that enable man-in-the-middle-style interference during session establishment.

Step 1: Initiation of V2V authentication

A legitimate vehicle V_1 initiates the V2V authentication process by sending the authentication request $\left\{TID_{V_1}, TS_{V_1}, PU_{V_1}, M_{V_1}, Sign_{V_1}\right\}$ to vehicle V_2.

Step 2: Interception of the authentication message

The adversary A, who controls the public communication channel, intercepts this message before it reaches V_2. Consequently, the legitimate authentication request generated by V_1 is prevented from being delivered to the intended recipient.

Step 3: Construction of the substituted authentication tuple

After intercepting the message, the adversary generates its own parameters in order to construct a substituted authentication request. Specifically, the adversary

selects a private scalar s_A and computes the corresponding public-key $PU_{V_A} = s_A G$. Next, the adversary selects a random ephemeral scalar m_A and computes $M_{V_A} = m_A G$. The adversary then constructs the signature value $Sign_{V_A} = m_A + H\left(TID_{V_1} \| TS_{V_A} \| PU_{V_A}\right) \cdot s_A$, using the claimed identity TID_{V_1}, a fresh timestamp TS_{V_A}, and its own key pair. This construction follows the same algebraic structure required by the protocol verification equation.

Step 4: Delivery of the substituted authentication request

The adversary sends the substituted authentication request $\{TID_{V_1}, TS_{V_A}, PU_{V_A}, M_{V_A}, Sign_{V_A}\}$ to vehicle V_2, thereby replacing the legitimate authentication request originally sent by V_1.

Step 5: Algebraic verification at the receiver

Upon receiving the substituted message, vehicle V_2 verifies the freshness of the timestamp and checks the signature equation $Sign_{V_A} \cdot G = M_{V_A} + H\left(TID_{V_1} \| TS_{V_A} \| PU_{V_A}\right) \cdot PU_{V_A}$. Substituting the forged value of $Sign_{V_A}$ into the verification equation gives $\left(m_A + H\left(TID_{V_1} \| TS_{V_A} \| PU_{V_A}\right) s_A\right) G$. By the distributive property of scalar multiplication over elliptic-curve points, this expression expands to $m_A G + H\left(TID_{V_1} \| TS_{V_A} \| PU_{V_A}\right) s_A G$. Since $m_A G = M_{V_A}$ and $s_A G = PU_{V_A}$, the equation becomes $M_{V_A} + H\left(TID_{V_1} \| TS_{V_A} \| PU_{V_A}\right) PU_{V_A}$, which exactly matches the right-hand side of the verification equation. Therefore, the verification equation holds, and V_2 accepts the substituted authentication request as valid.

Step 6: Continuation of the authentication exchange

After accepting the substituted request, vehicle V_2 generates the response message $\{TID_{V_2}, TS_{V_2}, PU_{V_2}, M_{V_2}, Sign_{V_2}\}$ and sends it toward the claimed initiator. Because the adversary controls the communication channel, it can intercept this response as well. Consequently, vehicle V_2 continues the authentication process under the false assumption that it is interacting with the legitimate vehicle V_1, while in reality the authentication context has already been substituted by the adversary.

Step 7: Feasibility of man-in-the-middle interference

The protocol assumes that the final key-confirmation message $\{SK_{V_1V_2}, NT_{SV_1}\}$ is transmitted through a secure auxiliary channel. However, if such a channel is unavailable, the adversary can intercept, relay, modify, or replace this message. In this case, the adversary gains full control over the authentication exchange and may relay or manipulate messages between the two vehicles. Therefore, the protocol exhibits man-in-the-middle feasibility, since the adversary can substitute authentication parameters and interfere with the session establishment process without being detected.

6 Informal analysis

In this section, we present the algebraic validity analysis and the formal analysis based on BAN logic and discuss the identified security weaknesses of the target protocol.

6.1 Algebraic validity analysis

The attacks described in the previous subsections share a common structural property related to the protocol's verification mechanism. Specifically, during the V2V authentication phase, the receiver validates an authentication request by checking the elliptic-curve verification equation $\text{Sign} \cdot G = M + H(TID \| TS \| PU) \cdot PU$ where the public-key PU is taken directly from the received message. Because the protocol does not independently verify that the received public key is authentically bound to the claimed temporary identity, the verification procedure guarantees only the **algebraic consistency** of the received authentication tuple. Consequently, an adversary may construct a mathematically valid authentication tuple under an arbitrary key pair while falsely claiming another vehicle identity.

6.1.1 Lemma 1: Algebraic validity of forged authentication tuples

Lemma 1 *Let an adversary choose an arbitrary private scalar* s_A *and compute the corresponding public-key* $PU_{V_A} = s_A G$. *Let the adversary further select an ephemeral scalar* m_A *and compute* $M_{V_A} = m_A G$. *If the adversary constructs the signature value* $\text{Sign}_{V_A} = m_A + H\left(TID_{V_1} \| TS_{V_A} \| PU_{V_A}\right) \cdot s_A$, *then the authentication tuple* $\left\{TID_{V_1}, TS_{V_A}, PU_{V_A}, M_{V_A}, \text{Sign}_{V_A}\right\}$ *always satisfies the protocol verification equation used by the receiving vehicle* V_2.

Proof During verification, the vehicle V_2 checks whether $\text{Sign}_{V_A} \cdot G = M_{V_A} + H\left(TID_{V_1} \| TS_{V_A} \| PU_{V_A}\right) \cdot PU_{V_A}$. Substituting the forged value of Sign_{V_A} yields $\text{Sign}_{V_A} \cdot G = \left(m_A + H\left(TID_{V_1} \| TS_{V_A} \| PU_{V_A}\right) \cdot s_A\right) G$. Using the distributive property of scalar multiplication over elliptic-curve points, the expression expands to $m_A G + H\left(TID_{V_1} \| TS_{V_A} \| PU_{V_A}\right) \cdot s_A G$. Since $m_A G = M_{V_A}$ and $s_A G = PU_{V_A}$, The equation becomes $M_{V_A} + H\left(TID_{V_1} \| TS_{V_A} \| PU_{V_A}\right) \cdot PU_{V_A}$. Therefore, the verification equation holds exactly even though the public key used in the verification process does not belong to the legitimate vehicle associated with the claimed identity TID_{V_1}.

6.1.2 Security implication

Lemma 1 demonstrates that the protocol verification rule guarantees only the algebraic correctness of the authentication tuple, but it does not ensure that the public key used in the verification equation is authentically bound to the claimed temporary vehicle identity.

Consequently, an adversary can construct a mathematically valid authentication tuple under its own key pair while claiming the identity of another vehicle. This fundamental weakness directly enables the vehicle impersonation, rogue public key, and message-substitution attacks described in the previous subsections.

Because the receiver accepts critical public parameters directly from an untrusted communication channel without establishing an authenticated identity–key binding, an active adversary under the Dolev–Yao adversarial model can manipulate authentication parameters and interfere with session establishment undetected.

6.1.3 Theorem 1: Failure of vehicle authentication

Theorem 1 *Under the Dolev–Yao adversarial model, the target protocol fails to guarantee vehicle authentication. In particular, the protocol does not ensure that the receiving vehicle* V_2 *can correctly conclude that the established session key is shared with the legitimate vehicle* V_1.

Proof From Lemma 1, an adversary can construct a forged authentication tuple $\{TID_{V_1}, TS_{V_A}, PU_{V_A}, M_{V_A}, Sign_{V_A}\}$ that satisfies the protocol verification equation while using an adversary-controlled key pair.

Because the protocol does not verify that the received public-key PU_{V_A} is authentically bound to the claimed identity TID_{V_1}, the receiving vehicle V_2 accepts the forged authentication request as valid.

Consequently, V_2 proceeds with the protocol execution and establishes a session context under the false assumption that it is interacting with the legitimate vehicle V_1.

However, the authentication tuple was generated by the adversary using its own key pair. Therefore, the established authentication state does not correspond to the claimed identity, which violates the intended authentication goal.

Hence, the protocol cannot guarantee that the responder shares the established session with the genuine vehicle associated with the claimed identity.

6.1.4 Resulting security consequence

Theorem 1 shows that the protocol fails to provide reliable vehicle authentication. As a result, an active adversary can impersonate legitimate vehicles, substitute public keys, or manipulate authentication messages without being detected. This

weakness directly enables the vehicle impersonation attack, the rogue public-key attack, and the message-substitution/MITM feasibility attack described in the previous subsections.

6.2 BAN-logic analysis

To further substantiate the cryptanalytic findings reported in Sect. 4, we analyse the V2V authentication phase using BAN logic. The purpose of this analysis is not to rediscover the attacks, but to determine whether the exchanged protocol messages are sufficient to establish the intended authentication and session-key beliefs. In particular, we examine whether V_2 can legitimately believe that the initiator is the genuine V_1, and whether the established session key can be logically attributed to the claimed peer identity.

In the target protocol, V_1 sends the message $\{TID_{V_1}, TS_{V_1}, PU_{V_1}, M_{V_1}, Sign_{V_1}\}$ and V_2 verifies the corresponding signature equation using the same received public-key PU_{V_1}. The final key-confirmation message is further assumed to be transmitted over a secure channel.

6.2.1 Security goals

Let SK denote the session key intended to be established between V_1 and V_2. The main authentication goals are defined as follows:

$V_2| \equiv V_1 \leftrightarrow_{SK} V_2 \qquad V_2| \equiv V_1| \equiv (V_1 \leftrightarrow_{SK} V_2) \qquad V_1| \equiv V_2 \leftrightarrow_{SK} V_1$

$V_1| \equiv V_2| \equiv (V_2 \leftrightarrow_{SK} V_1)$ These goals represent session-key agreement and mutual authentication between the two communicating vehicles.

6.2.2 Idealized messages

The essential V2V message flow can be idealized as follows:

$M_1 : V_1 \rightarrow V_2 : \{TID_{V_1}, TS_{V_1}, PU_{V_1}, M_{V_1}, Sign_{V_1}\}$

$M_2 : V_2 \rightarrow V_1 : \{TID_{V_2}, TS_{V_2}, PU_{V_2}, M_{V_2}, Sign_{V_2}\}$

$M_3 : V_1 \rightarrow V_2 : \{SK_{V_1V_2}, NT_{SV_1}\}$. The critical point is that message M_1 carries both the claimed temporary identity TID_{V_1} and the verification key PU_{V_1} over the same untrusted communication channel. The acceptance decision at V_2 is therefore based on the internal algebraic consistency of the received tuple rather than on a trusted identity-to-key binding established through an authenticated source.

6.2.3 Initial assumptions

The BAN analysis adopts the following standard assumptions:

$V_2| \equiv \#(TS_{V_1}) V_1| \equiv \#(TS_{V_2}) V_2| \equiv V_1 \Rightarrow (V_1 \leftrightarrow_{SK} V_2) V_1| \equiv V_2 \Rightarrow (V_2 \leftrightarrow_{SK} V_1)$.

However, successful authentication additionally requires a trustworthy belief that the claimed temporary identity TID_{V_1} is authentically associated with the verification key PU_{V_1}. This binding is precisely the assumption undermined by the attacks presented in Sect. 4.

6.2.4 Belief analysis

After receiving the message M_1, vehicle V_2 verifies the freshness of the timestamp and checks the signature equation using the received public-key PU_{V_1}. Under this verification rule, the strongest belief that V_2 can derive is only that the sender knows the private key corresponding to the supplied public key.

This belief is strictly weaker than concluding that the sender is the legitimate vehicle represented by the identity V_1. In particular, BAN reasoning cannot soundly derive a message-origin belief of the form $V_2| \equiv V_1| \sim X$ because the protocol does not independently establish a trusted association between the claimed identity and the verification key. In the presented V2V verification process, the public key is accepted directly from the communication channel without authenticated validation.

The freshness of TS_{V_1} does not resolve this issue, since freshness indicates only recency and not legitimate origin. A freshly generated message constructed under a substituted public key may still satisfy the timestamp condition while remaining logically unrelated to the genuine V_1.

6.2.5 Failure of authentication goals

Because the origin of the initiating message cannot be soundly attributed to the legitimate vehicle V_1, vehicle V_2 cannot justifiably conclude that the established session key is shared specifically with V_1. Hence, the authentication goal $V_2| \equiv V_1 \leftrightarrow_{SK} V_2$ is not formally guaranteed.

For the same reason, the stronger goal $V_2| \equiv V_1| \equiv \left(V_1 \leftrightarrow_{SK} V_2\right)$ also fails, since second-order belief requires an already valid first-order authentication belief. The same reasoning applies symmetrically to V_1, and therefore the corresponding initiator-side goals are likewise unsupported.

6.2.6 Dependence on the secure-channel assumption

The logical weakness becomes more evident in the final confirmation step, where the protocol assumes that the message $\left\{SK_{V_1V_2}, NT_{SV_1}\right\}$ is transmitted through a secure auxiliary channel. This assumption implies that the final confidence in the established session is not derived solely from the public-network message flow, but instead depends on an external trust mechanism.

Once this auxiliary secure-channel assumption is removed, the protocol no longer provides a self-contained logical basis for authentication and key confirmation under the claimed Dolev–Yao adversarial model.

6.2.7 BAN analysis

The BAN analysis supports the practical cryptanalysis presented earlier. Although the protocol verifies the algebraic consistency of the submitted authentication tuple, it does not formally establish that the received public key is authentically bound to

the claimed temporary vehicle identity in the presence of an active adversary. Consequently, the responder cannot soundly conclude that it shares the established session key with the legitimate vehicle V_1. Therefore, the protocol fails to guarantee mutual authentication and reliable session-key belief under BAN-logic reasoning. Finally, it is worth noting that BAN logic does not explicitly model public-key substitution attacks. Nevertheless, the analysis demonstrates that the protocol messages themselves do not provide sufficient evidence to establish the intended authentication beliefs, which explains the vulnerabilities demonstrated in Sect. 4.

6.3 Discusses the identified security weaknesses

The cryptanalytic results presented in the previous sections indicate that the target protocol's security weaknesses stem from a fundamental flaw in its authentication design. In the V2V authentication phase, the verification mechanism validates only the algebraic correctness of the received authentication tuple while accepting the public key directly from the communication channel. Consequently, the protocol does not ensure a trusted binding between the claimed temporary vehicle identity and the corresponding public key. Because of this missing identity–key association, an adversary can construct mathematically valid authentication tuples using its own key pair while falsely claiming the identity of another vehicle. As demonstrated in the preceding analysis, this design weakness enables several attacks, including vehicle impersonation, rogue public-key substitution, and message substitution.

In addition, the protocol assumes the availability of a secure auxiliary channel for the final key-confirmation step. This assumption contradicts the Dolev–Yao threat model under which the protocol claims security, where all communications are considered insecure. Overall, these weaknesses indicate that the protocol cannot guarantee reliable vehicle authentication in adversarial environments. Consequently, the security claims of the target scheme are not fully supported under realistic IoV communication conditions.

7 Conclusion

In this paper, we revisited the security of the blockchain-based mutual authentication protocol for the IoV proposed by Suyel et al. Our analysis reveals a fundamental design weakness in the V2V authentication phase, in which the verification mechanism accepts the public key directly from the communication channel without establishing a trusted binding between the claimed vehicle identity and the corresponding public key. Due to this weakness, an adversary can construct algebraically valid authentication messages using its own key pair while impersonating a legitimate vehicle. Based on this observation, we demonstrated that the protocol is vulnerable to several attacks, including vehicle impersonation, rogue public-key substitution, and message-substitution attacks. In addition, the protocol relies on a secure auxiliary channel for the final key-confirmation step, which contradicts the claimed Dolev–Yao threat model. Overall, the presented analysis

shows that the protocol cannot guarantee reliable vehicle authentication in adversarial environments. Therefore, its claimed security properties do not hold under realistic IoV communication conditions. In future work, we plan to develop an improved and secure version of the proposed approach.

Author contribution Y.S. conceived the research idea, performed the cryptanalysis, developed the attack models, conducted the algebraic and BAN-logic analyses, and wrote the manuscript.

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK).

Data availability The data used to support this proposed scheme are provided within the article.

Declarations

Competing interest The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits 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 licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence 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. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Haddaji A, Ayed S, Chaari Fourati L (2024) IoV security and privacy survey: issues, countermeasures, and challenges. J Supercomput 80(15):23018–23082. https://doi.org/10.1007/s11227-024-06269-5
2. Salami Y (2024) SOBT-UF: secure offloading in blockchain infrastructure for intelligent transportation systems using 5G-enabled UAVs within a fog-edge computing federation. In 2024 19th Iranian Conference on Intelligent Systems (ICIS), IEEE, pp 217–222. https://doi.org/10.1109/ICIS64839.2024.10887460
3. Salami Y (2025) SOBV-FEF: secure lightweight data offloading base in blockchain technology for internet of vehicles enabled handover UAVs within a fog-edge federation. Sci Rep. https://doi.org/10.1038/s41598-025-31114-x
4. Kumar M, Kaur G, Rana PS (2025) Performance, portability, productivity, and security in HPC cloud: a systematic literature review. J Supercomput 81(11):1197. https://doi.org/10.1007/s11227-025-07685-x
5. Li C et al (2025) A distributed learning framework with blockchain and privacy-preserving for IoV. Appl Soft Comput 184:113710. https://doi.org/10.1016/j.asoc.2025.113710
6. Salami Y (2025) SO-ITS: a secure offloading scheme for intelligent transportation systems in federated fog-cloud. Iran J Comput Sci. https://doi.org/10.1007/s42044-025-00318-9
7. Salami Y, Khajehvand V (2020) SMAK-IOV: secure mutual authentication scheme and key exchange protocol in fog based IoV. J Comput Robot 13(1):11–20
8. Salami Y, Khajehvand V (2021) LSKE: lightweight secure key exchange scheme in fog federation. Complexity 2021:4667586. https://doi.org/10.1155/2021/4667586

9. Salami Y, Ebazadeh Y, Khajehvand V (2021) CE-SKE: cost-effective secure key exchange scheme in fog federation. Iran J Comput Sci 4(3):1–13
10. Salami Y, Khajehvand V, Zeinali E (2023) SAIFC: a secure authentication scheme for IOV based on fog-cloud federation. Secur Commun Netw 1:1–19. https://doi.org/10.1155/2023/9143563
11. Wang Y et al (2025) An intrusion detection system for Internet of Vehicles based on digital twin. J Supercomput. https://doi.org/10.1007/s11227-025-08044-6
12. Salami Y, Khajehvand V, Zeinali E (2024) SOS-FCI: a secure offloading scheme in fog–cloud-based IoT. J Supercomput 80(1):570–600. https://doi.org/10.1007/s11227-023-05499-3
13. Salami Y, Khajehvand V, Zeinali E (2024) LSMAK-IOV: lightweight secure mutual AKE scheme in fog-based IoV. In 2024 10th International Conference on Artificial Intelligence and Robotics (QICAR), IEEE, pp 1–5. https://doi.org/10.1109/QICAR61538.2024.10496659
14. Salami Y, Khajehvand V, Zeinali E (2024) A new secure offloading approach for internet of vehicles in fog-cloud federation. Sci Rep 14(1):5576. https://doi.org/10.1038/s41598-024-56141-y
15. Namasudra S, Das S, Datta S, Crespo RG, Taniar D (2025) An advanced blockchain-based mutual authentication technique for the Internet of Vehicles environment. J Supercomput. https://doi.org/10.1007/s11227-025-07934-z

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.