A Blockchain-Assisted Lightweight Privacy Preserving Authentication Protocol for Peer-to-Peer Communication in Vehicular Ad-hoc Network

Sharon Justine Payattukalanirappel (✉ sharonjustine.22phd11005@iiitkottayam.ac.in)
Indian Institute of Information Technology Kottayam
Panchami V Vamattathil
Indian Institute of Information Technology Kottayam
Mohammed Ziyad C Cheeramthodika
Indian Institute of Information Technology Kottayam

Research Article

Keywords: Peer-to-peer communication, VANET, Blockchain, Handover authentication

Posted Date: February 15th, 2024

DOI: https://doi.org/10.21203/rs.3.rs-3919847/v1

Additional Declarations: No competing interests reported.

A Blockchain-Assisted Lightweight Privacy Preserving Authentication Protocol for Peer-to-Peer Communication in Vehicular Ad-hoc Network

Sharon Justine Payattukalanirappel[1*], Panchami V Vamattathil[2] and Mohammed Ziyad C Cheeramthodika[3]

CSE- Cyber Security Department, Indian Institute of Information Technology Kottayam, Valavoor, Kottayam, 686635, Kerala, India.

*Corresponding author(s). E-mail(s): sharonjustine.22phd11005@iiitkottayam.ac.in;

Abstract

Vehicular Ad-Hoc Network (VANET), provides considerable real-time traffic information services that enhance safety and traffic effectiveness. However, as most of the VANET systems are centralized in nature prone to single-point failure, vulnerable to attacks and there will be reasonable latency in communication. In this paper, while considering the resource-constrained nature of VANET, a lightweight privacy-preserving authentication scheme for peer-to-peer communication using blockchain (DLPA) is proposed. We have designed and deployed smart contracts using Public blockchain to resist the vehicle impersonation attack, to identify illegal vehicle's identity and thereby non-repudiation will be achieved. Vehicle-to-Vehicle (V2V) authentication and peer-to-peer communication are attained without the involvement of a Trusted Authority (TA) and to eliminate the trusted third party who is responsible for generating the key. Furthermore, DLPA has achieved handover authentication of vehicles so that vehicles need not be re-authenticated when they enter into a new Road Side Unit (RSU) limit. The proposed scheme is implemented in different Ethereum powered test networks using Remix IDE to demonstrate the feasibility and to analyze the performance of the smart contract in terms of transaction cost and execution cost. In addition to that, security proof and analysis are performed to unveil that our proposed scheme preserves the privacy of the communicating parties, semantic security of the session key, and resistance against various known threats and attacks. Finally, the performance analysis of the scheme is done by calculating the communication and computation costs. While analyzing the result, the proposed protocol has a minimal cost when compared with other blockchain-based authentication schemes in VANET.

Keywords: Peer-to-peer communication, VANET, Blockchain, Handover authentication

1 Introduction

The Vehicular Ad-hoc Network is an intelligent transportation network, which consists of vehicles and Road Side Unit (RSU) connected using wireless sensor network. In the VANET architecture, each vehicle is incorporated with an On Board Unit (OBU) which makes the communication of vehicles possible [1]. VANET communication includes Vehicle to Vehicle (V2V) communication, Vehicle to RSU (V2RSU) communication, RSU to RSU (RSU2RSU) communication and so on [2]. In this paper, we are concentrating on V2V and V2RSU communication.

In VANET, the communication is through insecure wireless open communication channel, thus it makes the communication vulnerable towards threats. Any adversaries can monitor the data as well as they can fabricate the data. Since the data being transferred is highly sensitive and confidential in nature, to ensure security and privacy in the VANET communication authentication schemes are required. In the IoT based VANET scenario, it is inevitable to propose a lightweight authentication protocol for the resource constrained devices in VANET. Latency cannot be tolerated in the VANET scenario hence the proposed protocol dependency towards the central authority has been minimised. Whenever the vehicle moves from one RSU limit to another RSU limit the vehicle again need to authenticate. In this protocol, the authenticated vehicle details are updated in the Blockchain [19][20] hence avoiding the need for re-authentication. The decentralization of the protocol can be achieved by using Blockchain technology which is immutable in nature[21].

The cryptographic primitives used in this protocol are Elliptic Curve Cryptography (ECC), hash operations and bitwise operations. ECC is one of the public key cryptosystem. The ECC use analog of the discrete logarithm problem [3] for hardness thus makes it more secure than other public key cryptosystems. ECC based encryption and decryption is used in this protocol.security. Latency in Vehicular Ad-hoc Network cannot be tolerated. Hence, a peer-to-peer secure, privacy-preserving lightweight authentication scheme with minimal dependency on the central authority is proposed.

The major research contribution of this paper are as follows:

- Firstly, the security and performance of the existing systems analyzed and found that the the communication and computation cost of the existing systems are high, security and the privacy of the communication is not preserved, vulnerable to various attacks.
- Secondly,to achieve minimal dependency towards central authority, a Peer-to-peer, Decentralized, Lightweight, Privacy preserving Authentication protocol (DLPA) is proposed for Vehicular Ad-hoc Network (VANET).

- The formal security analysis is performed using BAN logic and Scyther tool. The informal security analysis is also shown to provide the idea of the protocol's self-reliance against various other attacks.
- The proposed authentication protocol, is implemented using smart contract in Ethereum IDE remix and the gas consumption for each phase is calculated.
- A handover authentication is also provided in this protocol using Blockchain.
- The performance evaluation of the proposed protocol is done and it is compared with the existing authentication protocols.

The reminder of the article is organized as follows. Section 2 reviews the literature of the existing works. The network model of the is included in section 3. The section 4 consists of the discussion of the proposed protocol. The formal and informal analysis is shown in section 5 and section 6 respectively. Section 7 includes the implementation of smart contract in Ethereum IDE remix and performance analysis is discussed and plotted in section 8. Finally the conclusion and future scope are included in section 9.

2 Related Work

In this section we discuss the existing works related to the Blockchain based authentication protocols in Vehicular Ad-hoc Network (VANET).

Recently, Wei, Lu, et al. [4] proposed a decentralized authenticated key agreement scheme based on smart contract for securing vehicular ad-hoc networks which is a smart contract based VANETs authenticated key agreement scheme. Smart contract deployed on a public Blockchain and Trusted Authority (TA) is not required for key generation. The RSU form the Blockchain network and the smart contract run on Blockchain. In this key agreement protocol bloom filters are used and the filters check whether the public keys are generated by registered vehicles or not. The Bloom filters used in this protocol provides an extra latency. In this protocol four Blockchain transactions are created, one transaction is for vehicle registration and another transaction is for registering the public key of the vehicle and the third transaction is in order to verify the legality of the road side unit and final transaction is to register the public key of RSU. In the authentication and key agreement phase any third party can capture message send by the sender and can fabricate the message and send to the receiver. Hence man in the middle attack is possible in this protocol.

In 2021, Xu, Zisang, et al.[5] proposed a Blockchain-based Roadside Unit-assisted authentication and key agreement Protocol for Internet of Vehicles. In this protocol a Blockchain based authentication and key agreement protocol is designed for the multi-Trusted Authority network model. Here the Trusted Authority forms the Blockchain network. Data centers are there to store the entire IoV information. The registration information of all vehicular network is stored in the data center and all TA's jointly maintain a ledger storing a parameter, which can be regarded as a pointer or block identifier. Any TA can find the vehicle information from Blockchain using the parameter. Authentication is performed with the help of Trusted Authority which increase the overall Latency of the scheme. The entire data is stored in a centralized data center which may be a single point of failure.

In the same year Bagga, Palak, et al.[6] proposed a batch authentication protocol for Internet of Vehicles which is based on Blockchain technology. The system consists of Trusted Authority, Road Side Unit, Vehicle, Fog Server, and Cloud Server. Vehicle to vehicle authentication and batch authentication is performed. Batch authentication is vehicle to RSU authentication, where a cluster of vehicles are authenticated under a particular RSU. TA delivers certificates consisting of identity, public key, private key to all vehicles and RSU. The fog node receives a partial block from the RSU which consists of list of transactions and their compact signature. The fog server receives the compact signature and checks the validity of the signature, if valid then that partial block will be forwarded to cloud server. The cloud server converts the partial block to complete block. The cloud server do mining on the block using practical Byzantine Fault Tolerance(pBFT) consensus algorithm. The addition Cloud Server makes the system more complex and this increases the overall latency of the system.

Son, Seunghwan, et al. [7] proposed vehicle to infrastructure handover authentication protocol for VANET based on Blockchain technology. The system consists of TA, RSU, vehicle and Blockchain. The TA deploys RSU and issues a smart card for vehicles in the registration phase. RSU authenticate the vehicle before communication. Blockchain of the scheme is constituted by TA and RSU. After the registration the TA uploads the information about the pseudo-identity of the registered vehicle to the Blockchain. During authentication RSU verifies whether the vehicle is registered or not. When vehicle to infrastructure authentication is over RSU uploads the transaction which consists of vehicle's temporal identity, a random number and RSU's signature to the Blockchain. While considering the VANET scenario V2V communication is also important which is not mentioned in this paper.

In 2020 Lin, Chao, et al.[8] proposed a Blockchain-based conditional privacy-preserving authentication protocol for vehicular ad-hoc networks. The system consists of Certificate Authorities (CA), RSU, vehicle and Blockchain network. The CA is a trusted entity which is responsible for managing the certificate of vehicles and RSU's public key. The CA sign the certificate and embedded it in to the transaction. The RSU use the Dedicated Short Range Communication (DSRC) protocol to communicate with On Board Unit. In this protocol RSU form the Blockchain network. RSU act as a full node for storing all transaction in the Blockchain. Real time generation of public and private key is done in this protocol. DSRC protocol is used in this protocol is very susceptible towards jamming attacks.

Li, Xinghua, et al.[9] proposed an unlinkable key agreement scheme with collusion resistance for VANETs. The system defines a two layer architecture where the upper layer consists of the TA that form the Blockchain network. The RSU and vehicles constitutes the lower layer. The TA selects several random numbers to randomize vehicular identity and calculates the inverse value. Homomorphic encryption is used to encrypt the random number and each inverse value to generate multiple ciphertexts. The inverse value along with the ciphertext constitutes a ticket which is shared by Blockchain network. The inverse value is used to verify the legitimacy of the vehicle pseudonym. The system provides unlikability. Before communication an anonymous authentication credential is generated. The computational overhead for the system is high.

An efficient authentication scheme over Blockchain for fog computing enabled Internet of Vehicles proposed by Eddine, Merzougui Salah, et al.[10]. The system consists of RSU, OBU, TA, Certification Authority (CA) , Blockchain Manager (BM), Authentication Manager (AM).The TA is a central secured authority has the responsibility to register the OBU, and RSU. TA is responsible for initializing and publishing public parameters for cryptographic functions. The CA is also a trusted authority which is responsible for updating certification in each fog area. The BM perform the authentication of OBU and it manages the Blockchain. The AM write down the results of the authentication in large public register. The AM and BM associated to form the Blockchain. Fog area composed of fog service providers for providing services to targeted users. In this system it need to rely on multiple Trusted Authority.

Yao, Yingying, et al. [11] proposed a Blockchain assisted anonymous authentication for vehicular for service. The system has Audit Department (AD), Service Manager (SM), Witness Peer, RSU, OBU, Consortium Blockchain. AD is fully trusted authority which is responsible for the registration of OBUs and SMs and also it has the responsibility of tracing the vehicles. SM manages all the vehicular fog devices and they authenticate OBUs. The witness peer writes the results to the public ledger through a consensus algorithm. The consortium Blockchain made up of SM and witness peers of all regions. In this system also multiple trusted authorities are there. The entire system need to rely on these trusted parties.

From the above literature analysis, it is found that most of the existing systems have high computational and communication overhead. An optimal value for both communication as well as computation cost for a system is rare. In some systems more than one trusted third parties are available, so that the user and the infrastructure need to trust all these third parties. Due to this dependency the communication overhead and the latency will be high.Some of the existing systems use centralized database which lack transparency and also the centralized database is a single point of failure. The proposed protocol addresses these issues.

3 Network model

The VANET communication includes the Vehicle to vehicle communication which is indicated by V2V and the vehicle to Road Side Unit communication which is indicated by V2RSU. Figure 1 shows the network model of the proposed authentication protocol. The model consists of vehicles, RSU, TA, Blockchain.

- The vehicles consists of On Board Unit (OBU) which is responsible for the communication of vehicle with other vehicles and RSU
- RSU will be placed in the road sides. In VANET the entire communication area is divided in to geographical regions. An RSU will be there for each geographical region.
- The TA manages all the registrations. Before initiating the communication, vehicles as well as the RSU should register with the TA. For the authentication between the entities TA is not required.
- The registered vehicles and RSU details will be uploaded to the Blockchain. And also after vehicle to RSU authentication the vehicle details will be updated to the

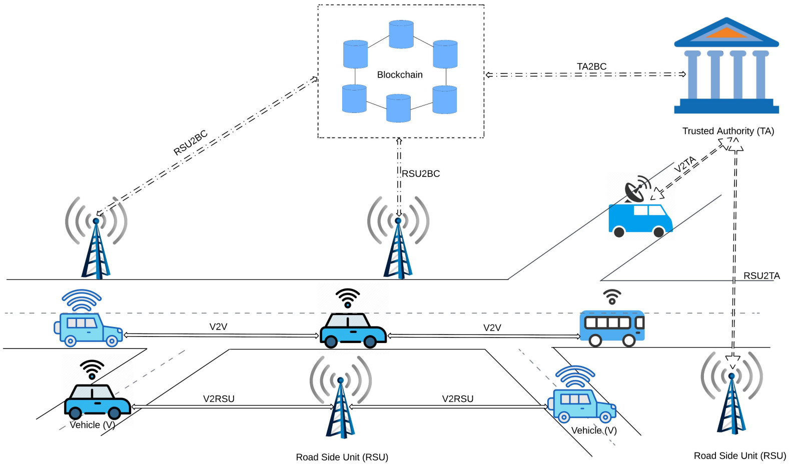


Fig. 1 Network model for the VANET authentication

Blockchain in order to make the handover authentication possible. When the vehicle moves from one RSU range to another RSU range there is no need to re-authenticate with the new RSU, the RSU in communication with the vehicle can check whether the vehicle is already authenticated with some other RSU, using the details in the Blockchain. A particular vehicle want to perform authentication with an RSU only once, later the handover authentication can be performed using the details of the authenticated vehicle uploaded in the Blockchain.

4 Proposed Protocol

The proposed authentication protocol is categorised in to two phases

- The registration of vehicle and RSU with the TA and uploading the registration details in to the Blockchain.
- Authentication between vehicle to vehicle and between vehicle and RSU. After the authentication between vehicle and RSU the authenticated vehicle information is stored in the Blockchain, which enables the handover authentication.

The ECC based encryption is denoted in this paper as $C \longleftarrow (M, K)$ where C is the ciphertext, M is the message and K is the Key. The ECC based decryption is denoted as $M \longleftarrow (C, K)$.

4.1 Registration Process

In the registration process the vehicle and the RSU need to register with the TA.

Table 1 Notations used in the proposed DLPA scheme

Symbols	Descriptions
V_i	Vehicle i
V_j	Vehicle j
V_x	Vehicle x
RSU_i	Road Side Unit i
TA	Trusted Authority
PU	Public element
PR	Secret element
M_i	Message
VN	Vehicle Number
$\mathcal{T}_1, \mathcal{T}_2$	Timestamp
$\mathcal{T}_c$	Current Timestamp
UID	Unique id
X, A, B, S, C, K D, L, A^*, S^*, K^*	Variables
SK	Session Key
h	SHA-256 hash function
$\bigoplus$	XOR operation
$\|\|$	Concatenation
E	ECC based lightweight encryption
D	ECC based decryption
N	Nonce

For the vehicle registration, initially the vehicle need to generate a random number. The random number act as a seed . The private key for the vehicle is generated using this seed. The public key for the vehicle PU_{Vi} is generated using the Elliptic Curve point multiplication of the private key of vehicle PR_{Vi} with P. P is the ECC generator. The vehicle number of the registering vehicle is concatenated with the public key of the vehicle $(VN_{Vi}||PU_{Vi})$ and this will be send along with the timestamp to the TA. When the message reaches at the TA, the TA generates the current timestamp and verify the timestamp sent by the vehicle, $\mathcal{T}_1 - \mathcal{T}_c| \leq \Delta\mathcal{T}$. The TA generate a unique id for the vehicle $UID_{Vi} \longleftarrow PU_{Vi} \bigoplus h(VN_{Vi}||PR_{TA})$. The unique id concatenated with the public key of trusted authority and encrypted using the public key of Vi. This message X_{Vi} along with the timestamp will be sent to the vehicle. The Vi verify the timestamp by generating the current timestamp and decrypt X_{Vi} to get the unique id. The TA stores the $(UID_{Vi}||VN_{Vi})$ to the Blockchain. The vehicle registration process is depicted in Table 2.

As like vehicle registration, RSU also generate a random number which act as seed to generate the private key of the RSU. The public key of the RSU is generated by $PU_{RSUi} \longleftarrow PR_{RSUi}.P$. Where P is the ECC generator. RSU send its public key along with the timestamp to the TA. The TA verifies the timestamp by generating the current timestamp, $Verify\,|\mathcal{T}_1 - \mathcal{T}_c| \leq \Delta\mathcal{T}$. The TA generates an unique id for RSU, $UID_{RSUi} \longleftarrow (PR_{TA}||PU_{RSUi})$. The unique id is concatenated with the public key of TA, and the hash value is taken which is encrypted using the public key of RSU, $X_{RSUi} \longleftarrow E((UID_{RSUi}||PU_{TA}), PU_{RSUi})$. X_{RSUi} along with the timestamp is sent back to RSU. The RSU verifies the timestamp and decrypt X_{RSUi}, UID_{RSUi} stored in the Blockchain. The protocol for the RSU registration is shown in Table 3

Table 2 Vehicle registration protocol

Vi	TA
Generate a random number seed.	
Private key for vehicle Vi is generated using seed	
$Public\,key\,of\,V_i,\ PU_{Vi} \longleftarrow PR_{Vi}.P$	
P is the ECC generator	
$M_1 \longleftarrow (VN_{Vi}\|\|PU_{Vi})$	
$\xrightarrow{\{M_1.\mathcal{T}_1\}}$	$Verify\,\|\mathcal{T}_1 - \mathcal{T}_c\| \leq \Delta\mathcal{T}$
	Generate an unique ID for Vi
	$UID_{Vi} \longleftarrow PU_{Vi} \bigoplus h(VN_{Vi}\|\|PR_{TA})$
	$X_{Vi} \longleftarrow E((UID_{Vi}\|\|PU_{TA}), PU_{Vi})$
	$M_2 \longleftarrow X_{Vi}$
	$\xleftarrow{\{M_2, \mathcal{T}_2\}}$
$Verify\,\|\mathcal{T}_2 - \mathcal{T}_c\| \leq \Delta\mathcal{T}$	
Decrypt X_{Vi}	
$(UID_{Vi}\|\|PU_{TA}) \longleftarrow D(X_{Vi}, PR_{Vi})$	
	$(UID_{Vi}\|\|VN_{Vi})$ stored in the Blockchain

The unique id generated for the vehicle and RSU by the TA is stored in the Blockchain. Hence anyone can validate the registered vehicles and RSU in the VANET.

4.2 Authentication Process

The authentication process happen between two vehicles and between RSU and vehicle. In the proposed protocol there is no need for the third party that is the TA for the authentication between entities in the VANET.

In the vehicle to vehicle (V2V) authentication, the vehicle generate a random number N_i and it perform some computation. Vehicle Vi calculates A_{Vi}, $A_{Vi} \longleftarrow h(N_i||PU_{Vi})$. A_{Vi} is encrypted using the public key of V_j and the encrypted value is assigned to a variable B_{Vi}. Then compute $S_{Vi} \longleftarrow B_{Vi} \bigoplus h(A_{Vi})$. S_{Vi} is encrypted with private key of Vi. A message will be created with B_{Vi} and C_{Vi}. The message along with the timestamp is sent to Vj. The receiver verifies the timestamp $|\mathcal{T}_1 - \mathcal{T}_c| \leq \Delta\mathcal{T}$. B_{Vi} is decrypted using the private key of Vj and C_{Vi} is decrypted using the public key of Vi. Then calculates $h(A_{Vi}) \longleftarrow B_{Vi} \bigoplus S^*_{Vi}$. For authentication vj check if $h(A_{Vi}) = h(A^*_{Vi})$. Then computes $K_{Vj} \longleftarrow PU_{Vj} \bigoplus h(UID_{Vj}) \bigoplus h(A_{Vi})$. K_{Vj} is encrypted with private key of Vj. The unique id of Vj is encrypted with Vi's public key. A session key is also generated $SK_{ViVj} \longleftarrow h(K_{Vj}||UID_{Vj})$. A message created with D_{Vj} and L_{Vj} and it will be sent to Vi along with the timestamp. The Vi verifies the timestamp by generating the current timestamp. Decrypt

Table 3 RSU registration protocol

RSUi	TA
Generate a random number seed.	
Private key for Road Side Unit RSUi is generated using seed	
Public key of RSU_i, $PU_{RSUi} \longleftarrow PR_{RSUi}.P$ P is the ECC generator	
$M_3 \longleftarrow (PU_{RSUi})$	
$\xrightarrow{\{M_3.\mathcal{T}_1\}}$	
	$Verify\,|\mathcal{T}_1 - \mathcal{T}_c| \leq \Delta\mathcal{T}$
	Generate an unique ID for RSUi
	$UID_{RSUi} \longleftarrow h(PR_{TA}\|\|PU_{RSUi})$
	$X_{RSUi} \longleftarrow E((UID_{RSUi}\|\|PU_{TA}), PU_{RSUi})$
	$M_4 \longleftarrow X_{RSUi}$
	$\xleftarrow{\{M_4, \mathcal{T}_2\}}$
$Verify\,|\mathcal{T}_2 - \mathcal{T}_c| \leq \Delta\mathcal{T}$	
Decrypt X_{RSUi}	
$(UID_{RSUi}\|\|PU_{TA}) \longleftarrow D(X_{RSUi}, PR_{RSUi})$	
	UID_{RSUi} stored in the Blockchain

$L_{Vj}, K^*_{Vj} \longleftarrow D(L_{Vj}, PU_{Vj})$. Then decrypt $D_{Vj}, UID^*_{Vj} \longleftarrow D(D_{Vj}, PR_{Vi})$. Calculate $h(UID_{Vj}) \longleftarrow K^*_{Vj} \bigoplus PU_{Vj} \bigoplus h(A_{Vi})$. Check if $h(UID_{Vj}) = h(UID^*_{Vj})$ if the value is equal then Vi and Vj is mutually authenticated. Then the session key will be generated $SK_{ViVj} \longleftarrow h(K^*_{Vj}||UID_{Vj})$. The V2V authentication protocol is shown in Table 4.

As like V2V authentication, in vehicle to RSU (V2RSU) authentication initially a random number Nx will be generated by vehicle Vx. Calculate $A_{Vx} \longleftarrow h(N_x||PU_{Vx})$. A_{Vx} is encrypted using the public key of $RSUx$ which is assigned to a variable B_{Vx}. Calculate S_{Vx}, $S_{Vx} \longleftarrow B_{Vx} \bigoplus h(A_{Vx})$. S_{Vx} is encrypted using the secret key of Vx and assigned to a variable C_{Vx} . A message will be created with B_{Vx} and C_{Vx} and sent to $RSUx$ along with the timestamp. The $RSUx$ verifies $T_1 - \mathcal{T}_c| \leq \Delta\mathcal{T}$. Decrypt B_{Vx}, $A^*_{Vx} \longleftarrow D(B_{Vx}, PR_{RSUx})$. Then decrypt C_{Vx}, $S^*_{Vx} \longleftarrow D(C_{Vx}, PU_{Vx})$. Calculate $h(A_{Vx}) \longleftarrow B_{Vx} \bigoplus S^*_{Vx}$. Then checks $h(A_{Vx}) = h(A^*_{Vx})$. If the value is equal then store the vehicle's public key to the Blockchain. Hence the handover authentication of vehicles when the vehicle moves from one RSU limit to another RSU limit become possible. K_{RSUx} is calculated $K_{RSUx} \longleftarrow PU_{RSUx} \bigoplus h(UID_{RSUx}) \bigoplus h(A_{Vx})$, and encrypted $L_{RSUx} \longleftarrow E(K_{RSUx}, PR_{RSUx})$. Unique id of $RSUx$ is encrypted using the public key of Vx, $D_{RSUx} \longleftarrow E(UID_{RSUx}, PU_{Vx})$. Session key is generated

Table 4 V2V authentication protocol

Vi	Vj
Generates a random number $\mathcal{N}_i$	
Computes:	
$A_{Vi} \longleftarrow h(N_i\|\|PU_{Vi})$	
$B_{Vi} \longleftarrow E(A_{Vi}, PU_{Vj})$	
$S_{Vi} \longleftarrow B_{Vi} \bigoplus h(A_{Vi})$	
$C_{Vi} \longleftarrow E(S_{Vi}, PR_{Vi})$	
$M_5 \longleftarrow (B_{Vi}, C_{Vi})$	
$\xrightarrow{\{M_5, \mathcal{T}_1\}}$	
	$Verify\, \|\mathcal{T}_1 - \mathcal{T}_c\| \leq \Delta\mathcal{T}$
	$Decrypt\ B_{Vi}, A^*_{Vi} \longleftarrow D(B_{Vi}, PR_{Vj})$
	$Decrypt\ C_{Vi}, S^*_{Vi} \longleftarrow D(C_{Vi}, PU_{Vi})$
	$h(A_{Vi}) \longleftarrow B_{Vi} \bigoplus S^*_{Vi}$
	check if $h(A_{Vi}) = h(A^*_{Vi})$
	$K_{Vj} \longleftarrow PU_{Vj} \bigoplus h(UID_{Vj}) \bigoplus h(A_{Vi})$
	$D_{Vj} \longleftarrow E(UID_{Vj}, PU_{Vi})$
	$L_{Vj} \longleftarrow E(K_{Vj}, PR_{Vj})$
	$SK_{ViVj} \longleftarrow h(K_{Vj}\|\|UID_{Vj})$
	$M_6 \longleftarrow (D_{Vj}, L_{Vj})$
	$\xleftarrow{\{M_6, \mathcal{T}_2\}}$
$Verify\, \|\mathcal{T}_2 - \mathcal{T}_c\| \leq \Delta\mathcal{T}$	
$Decrypt\ L_{Vj}, K^*_{Vj} \longleftarrow D(L_{Vj}, PU_{Vj})$	
$Decrypt\ D_{Vj}, UID^*_{Vj} \longleftarrow D(D_{Vj}, PR_{Vi})$	
$h(UID_{Vj}) \longleftarrow K^*_{Vj} \bigoplus PU_{Vj} \bigoplus h(A_{Vi})$	
$if\ h(UID_{Vj}) = h(UID^*_{Vj})$	
mutually authenticated	
$SK_{ViVj} \longleftarrow h(K^*_{Vj}\|\|UID_{Vj})$	

between the vehicle and RSU, $SK_{VxRSUx} \longleftarrow h(K_{RSUx}||UID_{RSUx})$. A message is created with D_{RSUx} and L_{RSUx}. The message and the timestamp generated is sent back to Vx. The vehicle Vx initially verifies the timestamp. Then decrypt L_{RSUx}, $K^*_{RSUx} \longleftarrow D(L_{RSUx}, PU_{RSUx})$ and also the vehicle decrypt D_{RSUx}, $UID^*_{RSUx} \longleftarrow D(D_{RSUx}, PR_{Vx})$. Vx calculates $h(UID_{RSUx}) \longleftarrow K^*_{VRSUx} \bigoplus PU_{RSUx} \bigoplus h(A_{Vx})$. Then it checks if $h(UID_{RSUx}) = h(UID^*_{RSUx})$. If the value of hash functions is equal then the vehicle and RSU become mutually authenticated. Then the session key generated $SK_{VxRSUx} \longleftarrow h(K^{*}_{RSUx}||UID_{RSUx})$. The V2RSU authentication protocol is depicted in Table 5.

5 Formal security analysis

5.1 Formal security analysis using BAN logic

BAN logic is formal verification technique which consists of a set of rules that analyze the exchange of information in protocols[12]. The proposed authentication protocol use the BAN principals, statements and the rule of inference to validate the set of goals in the scheme. The symbols and the corresponding description is shown in Table 6.

5.1.1 Goals

Goal G1:

$$V_j \mid\equiv \#(M_5)$$

Goal G2:

$$V_i \mid\equiv \#(M_6)$$

Goal G3:

$$V_j \triangleleft A_{Vi}$$

Goal G4:

$$V_j \triangleleft S_{Vi}$$

Goal G5:

$$V_i \triangleleft UID_{Vj}$$

Goal G6:

$$V_i \triangleleft K_{Vj}$$

Goal G7:

$$V_i \mid\equiv V_i \overset{SK_{ij}}{\rightleftharpoons} V_j$$

Goal G8:

$$V_j \mid\equiv V_i \overset{SK_{ij}}{\rightleftharpoons} V_j$$

Eight goals are set for the proposed authentication scheme. The first and second goals shows the freshness of the messages sent to Vj and Vi respectively. Goal 3 means that Vj receives A_{Vi}. Goal 4 indicates Vj receives S_{Vi}. Goal 5 and 6 shows that UID_{Vj} and K_{Vj} received by V_i. Goal 7 and 8 indicates the session key establishment between V_i and V_j.

5.1.2 Assumptions and idealizations

Assumption A1.

$$V_j \mid\equiv \#(\mathcal{T}_1)$$

Assumption A2.

$$V_i \mid\equiv \#(\mathcal{T}_2)$$

Assumption A3.

$$V_i \mid\equiv \#(h(K_{Vj}||UID_{Vj}))$$

Assumption A4.

$$V_j \mid\equiv \#(h(K_{Vj}||UID_{Vj}))$$

Assumption A5.

$$V_i \mid\equiv V_i \xleftrightarrow{SK_{ij}} V_j$$

Assumption A6.

$$V_j \mid\equiv V_i \xleftrightarrow{SK_{ij}} V_j$$

The idealization of messages in the authentication protocol are as follows:
Idealization of M_5:

$V_i \longrightarrow V_j : [\ \mathcal{T}_1, \{A_{Vi}\}_{PU_{Vj}}, \{S_{Vi}\}_{PR_{Vi}}]$
Idealization of M_6:

$V_j \longrightarrow V_i : [\ \mathcal{T}_2, \{UID_{Vj}\}_{PU_{Vi}}, \{K_{Vj}\}_{PR_{Vj}}]$

5.1.3 The proof of BAN logic

This section shows the proof of goals in BAN logic.

S_1: From Assumption A1 and using R1 $\frac{V_j|\equiv\#(\mathcal{T}_1)}{V_j|\equiv\#(\mathcal{T}_1,M_5)}$

Hence$[V_j \mid\equiv \#(M_5)]$ **[Goal G1 proved]**

S_2: From Assumption A2 and using R1 $\frac{V_i|\equiv\#(\mathcal{T}_2)}{V_i|\equiv\#(\mathcal{T}_2,M_6)}$

Hence $[V_i \mid\equiv \#(M_6)]$ **[Goal G2 proved]**

S_3: From R2 $\frac{V_j|\equiv \xmapsto{PU_{Vj}} V_j, V_j \triangleleft \{A_{Vi}\}_{PU_{Vj}}}{V_j \triangleleft A_{Vi}}$
Hence

$$V_j \triangleleft A_{Vi}$$

[Goal G3 proved]

S_4: From R3 $\frac{V_j|\equiv \xmapsto{PU_{Vi}} V_j, V_j \triangleleft \{S_{Vi}\}_{PR_{Vi}}}{V_j \triangleleft S_{Vi}}$
Hence

$$V_j \triangleleft S_{Vi}$$

[Goal G4 proved]

S_5: From R2 $\frac{V_i|\equiv \xmapsto{PU_{Vi}} V_i, V_i \triangleleft \{UID_{Vj}\}_{PU_{Vi}}}{V_i \triangleleft UID_{Vj}}$
Hence

$$V_i \triangleleft UID_{Vj}$$

[Goal G5 proved]

S_6: From R3 $\frac{V_i|\equiv \overset{PU_{Vj}}{\longmapsto} V_j, V_i \triangleleft \{K_{Vj}\}_{PR_{Vj}}}{V_i \triangleleft K_{Vj}}$
Hence

$$V_i \triangleleft K_{Vj}$$

[Goal G6 proved]

S_7: From Assumption A3, A5 and R4

$$\frac{V_i|\equiv\#(h(K_{Vj}||UID_{Vj})), V_i|\equiv V_j|\equiv h(K_{Vj}||UID_{Vj})}{V_i|\equiv V_i \overset{SK}{\rightleftharpoons} V_j}$$

Hence $V_i |\equiv V_i \overset{SK}{\rightleftharpoons} V_j$

[Goal G7 proved]

S_8: From Assumption A4, A6 and R4

$$\frac{V_j|\equiv\#(h(K_{Vj}||UID_{Vj})), V_j|\equiv V_i|\equiv h(K_{Vj}||UID_{Vj})}{V_j|\equiv V_j \overset{SK}{\rightleftharpoons} V_i}$$

Hence $V_j |\equiv V_j \overset{SK}{\rightleftharpoons} V_i$

[Goal G8 proved]

5.2 Formal security analysis using Scyther

Scyther is an automated security protocol verification tool[13]. Scyther works under perfect cryptography assumption hence assumed that the adversary will not learn anything from the hashed and encrypted data[14]. The proposed authentication protocol is written in Security Protocol Description Language (SPDCL). The scyther tool simulate the adversary model using the Dolev-Yao model and performs different claims. Figure 2 shows the simulation result after verifying all claims between vehicle and RSU. The simulation result shows that all claims satisfied by the proposed authentication protocol.

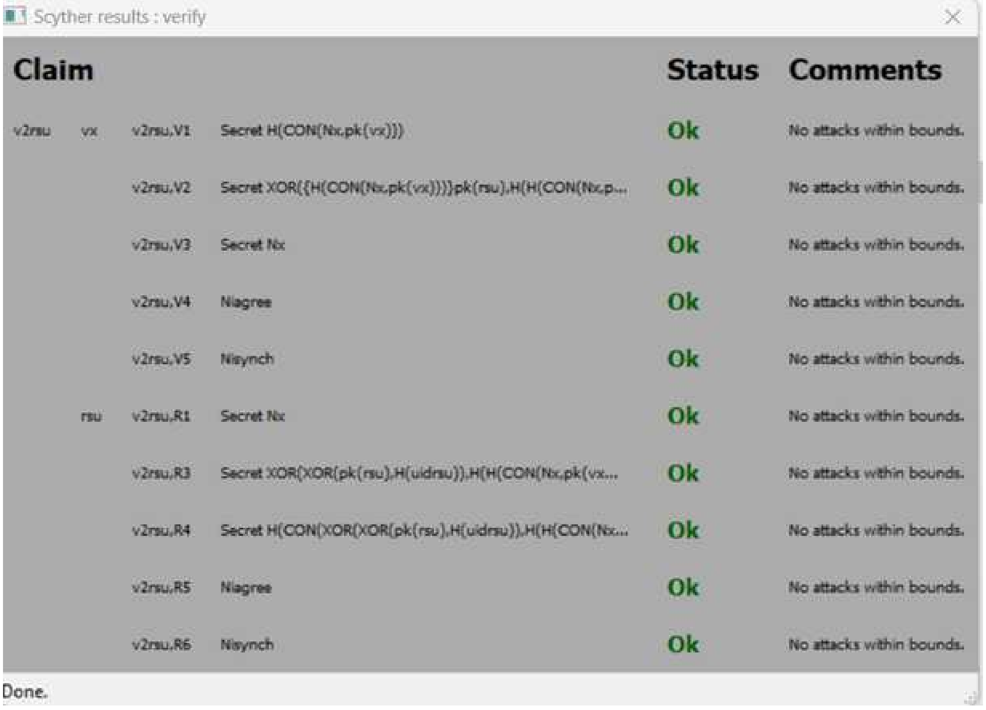
6 Informal security analysis

6.1 Mutual authentication

While communicating between two participants, both of them need to compute authenticators. In the V2RSU scenario the RSU need to compute $h(A_{Vx})$ after receiving authenticator from vehicle.. And the vehicle need to compute $h(UID_{RSU})$ after receiving the authenticator from RSU. Both parties verifies the received and the calculated values are same. In this way the mutual authentication between two parties happen.

6.2 Unlinkability

For each session the random number Ni is generated by the session initiator. Hash value of Ni will be taken some other computation and encryption will be done on the hashed value. Hence it is difficult for the Adv to derive relation between the messages sent during the communication between participants.



Scyther results : verify

Claim				Status	Comments
v2rsu	vx	v2rsu,V1	Secret H(CON(Nx,pk(vx)))	Ok	No attacks within bounds.
		v2rsu,V2	Secret XOR({H(CON(Nx,pk(vx)))}pk(rsu),H(H(CON(Nx,p...	Ok	No attacks within bounds.
		v2rsu,V3	Secret Nx	Ok	No attacks within bounds.
		v2rsu,V4	Niagree	Ok	No attacks within bounds.
		v2rsu,V5	Nisynch	Ok	No attacks within bounds.
	rsu	v2rsu,R1	Secret Nx	Ok	No attacks within bounds.
		v2rsu,R3	Secret XOR(XOR(pk(rsu),H(uidrsu)),H(H(CON(Nx,pk(vx...	Ok	No attacks within bounds.
		v2rsu,R4	Secret H(CON(XOR(XOR(pk(rsu),H(uidrsu)),H(H(CON(Nx...	Ok	No attacks within bounds.
		v2rsu,R5	Niagree	Ok	No attacks within bounds.
		v2rsu,R6	Nisynch	Ok	No attacks within bounds.

Done.

Fig. 2 Security analysis using scyther

6.3 Insider attack

Some third party who is an adversary *Adv* and pretends to be a genuine user and tries to capture the data then *Adv* cannot perform the attack. Because the data hold either public values, or hashed value of public and private elements or the data posses encrypted messages.

6.4 Privacy and anonymity

All the messages send between the parties in encrypted form. Hence Adv cannot obtain any private data. In the communication between two parties, identities of the parties are not shared. Hence privacy is preserved. For each session different nonce value and timestamp is generated, each time the message will be different. Since each message has different value it is difficult for an *Adv* to trace the identity of the user from the message sent.

6.5 Reply attack

The adversary *Adv* forge or delay the message from the sender during communication. However in each communication the timestamp is also send along with the message and at the receiver end verifies $|\mathcal{T}_i - \mathcal{T}_c| \leq \Delta\mathcal{T}$. Where $\mathcal{T}_i$ is the timestamp at which the data has been sent. $\mathcal{T}_c$ is the current timestamp. If the subtraction of these timestamp is less than the threshold value then the receiver accepts otherwise rejects. Hence the reply attack can be avoided.

6.6 Man in the middle attack

In the proposed protocol even the adversary Adv captures all the messages between the participants, since some private elements of the parties is also incorporated with the message the Adv cannot initiate a Man in the middle attack. All the registered vehicle information are stored in the Blockchain so the Adv cannot perform this attack.

6.7 Denial of service attack

The adversary Adv eavesdrop the communication between participants and hold the message, however the denial of service attack is not possible. Since the receiver in the communication checks the freshness of the message each time, by checking the timestamp sent by the sender. Message M_i can be hold by the Adv for some time making M_i out of service. But due to this verification $|\mathcal{T}_i - \mathcal{T}_c| \leq \Delta\mathcal{T}$. Where $\mathcal{T}_i$, this attack won't work.

6.8 Distributed denial of service (DDoS) attack

The proposed protocol run in Ethereum which has high transaction fees and high gas consumption. While using Blockchain the user need to pay, which is actually a huge amount. Hence the Adv will not initiate DDoS attack, because the Adv have to pay to perform the attack.

6.9 Stolen-verifier attack

In the communication between V_x and RSU_x. The messages sent from vehicle to RSU is in encrypted form. $B_{Vx} \longleftarrow E(A_{Vx}, PU_{RSUx})$, $C_{Vx} \longleftarrow E(S_{Vx}, PU_{RSUx})$. Even the Adv decrypt the message, the attacker may not able to find the message because it is in hashed form. From RSU to vehicle also the hashed messages in encrypted form is transferred through the internet. Hence even the attacker stolen the entire database as a whole, the Adv may not able to get the identity of each user.

6.10 Offline password guessing attack

The communication initiating party generate a random number Ni and perform hash operation and encryption on it. It is infeasible for Adv to perform the offline password guessing attack.

6.11 Forward and backward secrecy

The public keys used in our scheme is random in one session, resulting in that the final output session key is independent. The leakage of any of the messages will not compromise the messages created in another session.

6.12 Impersonation attack

The impersonation attack is not possible in the proposed protocol because both the parties generate some secret information and the secret information are random for each session. Hence *Adv* cannot perform impersonation.

6.13 Comparison of functionality and security requirements

The proposed system is compared with other authentication protocol based on their functionality ans security features. The proposed DLPA satisfies all the functionality and has resistance towards major security attacks when compared with other authentication protocols. From the literature of other papers the functionality and the security of those papers are studied. DLPA evaluated based on the informal security analysis and other schemes from literature review papers. The comparison is shown in Table 8.

7 Implementation

The proposed lightweight authentication scheme (DLPA) is implemented using the Ethereum IDE Remix. Remix is a web based IDE which consists of deployment tool for developing and managing the smart contract life cycle[17]. A smart contract written to simulate the authentication protocol in the solidity programming language. The algorithm for registration of RSU, registration of vehicle and authentication are given. Algorithm 1 is given for the registration of RSU, Algorithm 2 is for vehicle registration. Algorithm 3 is for the authentication. The vehicle is already registered or not is given by Algorithm 4. The Algorithm 5 is for checking whether the RSU is registered or not.

Algorithm 1 shows the registerRSU function. The input given is the rsuId and the output will be bool which shows the RSU is registered. There are some require statements are needed $msg.sender \neq address(0)$, $rsuMapping[msg.sender].rsuAddress \neq msg.sender$ means already registered, $bytes(rsuId).length > 0$, $vehicleMapping[msg.sender].isRegistered = false$, a vehicle is already registered with this address. Do $rsuMapping(msg.sender)rsuAddress = msg.sender$, $rsuMapping[msg.sender].id = rsuId$, then $rsuMapping[msg.sender].isRegistered = true$. The RSU that want to register in the VANET, the address and the Id is stored in the Blockchain and make that RSU as registered.

registerVehicle function is the Algorithm 2. The input of the function are uniqueId, vehicleNumber and the output is bool, the vehicle is registered by providing the vehicle number and unique id. There are some require statements are needed $msg.sender \neq address(0)$, $bytes(uniqueId).length > 0$, $bytes(vehicleNumber).length > 0$, $vehicleMapping[msg.sender].vehicleAddress! = msg.sender$ already registered, $rsuMapping[msg.sender].isRegistered == false$, an RSU is already registerd with this address. Perform $vehicleMapping[msg.sender].vehicleAddress = msg.sender$, $vehicleMapping[msg.sender].vehicleID = string(abi.encodePacked(uniqueId, vehicleNumber))$, the vehicle id and the unique number is concatenated and stored in Blockchain

Algorithm 1 registerRSU Function

```
Input rsuId
Output bool
require $msg.sender \neq address(0)$
require $rsuMapping[msg.sender].rsuAddress \neq msg.sender$
require $bytes(rsuId).length > 0$
require $vehicleMapping[msg.sender].isRegistered = false$
$rsuMapping(msg.sender)rsuAddress = msg.sender$
$rsuMapping[msg.sender].id = rsuId$
$rsuMapping[msg.sender].isRegistered = true$
return bool
```

$vehicleMapping[msg.sender].isRegistered = true$ and make the vehicle status as registered.

Algorithm 2 registerVehicle Function

```
Input uniqueId, vehicleNumber
Output bool
require $msg.sender \neq address(0)$
require $bytes(uniqueId).length > 0$
require $bytes(vehicleNumber).length > 0$
require $vehicleMapping[msg.sender].vehicleAddress \neq msg.sender$
require $rsuMapping[msg.sender].isRegistered == false$
$vehicleMapping[msg.sender].vehicleAddress = msg.sender$
$vehicleMapping[msg.sender].vehicleID = string(abi.encodePacked (uniqueId, vehicleNumber))vehicleMapping[msg.sender].isRegistered = true$
return bool
```

Algorithm 3 shows the authenticate function, which shows the handover authentication whenever the vehicle moves from one RSU limit to another RSU limit then if the vehicle is already authenticated with an RSU that information will be stored in the Blockchain hence there is no need to re-authenticate with another RSU. The input to the function is rsuHash, vehicleHash. The output of the function is bool whether the vehicle is authenticated with RSU. The require statements are $bytes(rsuHash).length > 0$, $bytes(vehicleHash).length > 0$,$vehicleMapping[msg.sender].isRegistered == true$ means the vehicle should register before authentication. Do $currentVehicleAddress == vehicleMapping[msg.sender].vehicleAddress$. If $keccak256(abi.encodePacked(rsuHash)) == keccak256(abi.encodePacked (vehicleHash))authVehicleMapping[currentVehicleAddress].authString = string (abi.encodePacked(msg.sender, vehicleMapping[msg.sender].vehicleID,$
The keccac256 hash function is used to check whether vehicle hash value match with the RSU calculated hash value which is already stored in the Blockchain during the authentication are same then

do $authVehicleMapping[currentVehicleAddress].isAuthenticated = true$, return the vehicle status as authenticated. else $authVehicleMapping[currentVehicleAddress].isAuthenticated = false$, set the vehicle authenticated with RSU as false.

Algorithm 3 authenticate Function

```
**Input** rsuHash, vehicleHash
**Output** bool
**require** $bytes(rsuHash).length > 0$
**require** $bytes(vehicleHash).length > 0$
**require** $vehicleMapping[msg.sender].isRegistered == true$
$currentVehicleAddress = vehicleMapping[msg.sender].vehicleAddress$
**if** $keccak256(abi.encodePacked(rsuHash)) == keccak256(abi.encodePacked(vehicleHash))$ **then**
    $authVehicleMapping[currentVehicleAddress].authString =$
    $string(abi.encodePacked(msg.sender, vehicleMapping[msg.sender].vehicleID$
    $authVehicleMapping[currentVehicleAddress].isAuthenticated = true$
**else**$authVehicleMapping[currentVehicleAddress].isAuthenticated = false$
**end if**
```

The function checkIfVehicleRegistered is shown as Algorithm 4. Input is the vehicle address and output is bool. The function returns $vehicleMapping[vehicleAddress].isRegistered$, whether the vehicle is already registered or not.

Algorithm 4 checkIfVehicleRegistered Function

```
**Input** vehicleAddress
**Output** bool
**return** $vehicleMapping[vehicleAddress].isRegistered$
```

Algorithm 5 is checkIfRSURegistered function. The input to the function is rsu address. The function returns $rsuMapping[rsuAddress].isRegistered$.

Algorithm 5 checkIfRSURegistered Function

```
**Input** rsuAddress
**Output** bool
**return** $rsuMapping[rsuAddress].isRegistered$
```

The function isVehicleAuthenticated Function is shown in Algorithm 6. Input to the function is vehicle address. It require $(vehicleMapping[vehicleAddress].isRegistered == trueState$. The function returns the value $authVehicleMapping[vehicleAddress].isAuthenticated$.

Table 5 V2RSU authentication protocol

Vx	RSUx
Generates a random number N_x	
Computes:	
$A_{Vx} \longleftarrow h(N_x \Vert PU_{Vx})$	
$B_{Vx} \longleftarrow E(A_{Vx}, PU_{RSUx})$	
$S_{Vx} \longleftarrow B_{Vx} \bigoplus h(A_{Vx})$	
$C_{Vx} \longleftarrow E(S_{Vx}, PR_{Vx})$	
$M_7 \longleftarrow (B_{Vx}, C_{Vx})$	
$\xrightarrow{\{M_7, \mathcal{T}_1\}}$	
	$Verify\ \lvert \mathcal{T}_1 - \mathcal{T}_c \rvert \leq \Delta\mathcal{T}$
	$Decrypt\ B_{Vx}, A^*_{Vx} \longleftarrow D(B_{Vx}, PR_{RSUx})$
	$Decrypt\ C_{Vx}, S^*_{Vx} \longleftarrow D(C_{Vx}, PU_{Vx})$
	$h(A_{Vx}) \longleftarrow B_{Vx} \bigoplus S^*_{Vx}$
	check if $h(A_{Vx}) = h(A^*_{Vx})$
	StorePU_{Vx} in Blockchain
	$K_{RSUx} \longleftarrow PU_{RSUx} \bigoplus h(UID_{RSUx}) \bigoplus h(A_{Vx})$
	$D_{RSUx} \longleftarrow E(UID_{RSUx}, PU_{Vx})$
	$L_{RSUx} \longleftarrow E(K_{RSUx}, PR_{RSUx})$
	$SK_{VxRSUx} \longleftarrow h(K_{RSUx} \Vert UID_{RSUx})$
	$M_8 \longleftarrow (D_{RSUx}, L_{RSUx})$
	$\xleftarrow{\{M_8, \mathcal{T}_2\}}$
$Verify\ \lvert \mathcal{T}_2 - \mathcal{T}_c \rvert \leq \Delta\mathcal{T}$	
$Decrypt\ L_{RSUx}, K^*_{RSUx} \longleftarrow D(L_{RSUx}, PU_{RSUx})$	
$Decrypt\ D_{RSUx}, UID^*_{RSUx} \longleftarrow D(D_{RSUx}, PR_{Vx})$	
$h(UID_{RSUx}) \longleftarrow K^*_{VRSUx} \bigoplus PU_{RSUx} \bigoplus h(A_{Vx})$	
$if\ h(UID_{RSUx}) = h(UID^*_{RSUx})$	
mutually authenticated	
$SK_{VxRSUx} \longleftarrow h(K^*_{RSUx} \Vert UID_{RSUx})$	

Algorithm 6 isVehicleAuthenticated Function

```
Input vehicleAddress
Output bool
require ($vehicleMapping[vehicleAddress].isRegistered == trueState$
return $authVehicleMapping[vehicleAddress].isAuthenticated$
```

Table 6 BAN logic notations

Symbols	Descriptions
V_i, V_j	Principals
M_5, M_6	Message
SK	Session Key
$V_i \mid\equiv V_j$	V_i believes V_j
$V_i \triangleleft V_j$	V_i sees V_j
$V_i \mid\sim V_j$	V_i once said V_j
$\#(X)$	X is fresh
$V_i \overset{SK}{\leftrightharpoons} V_j$	V_i and V_j have a shared session key SK
$\{X\}_k$	X is encrypted by a key k

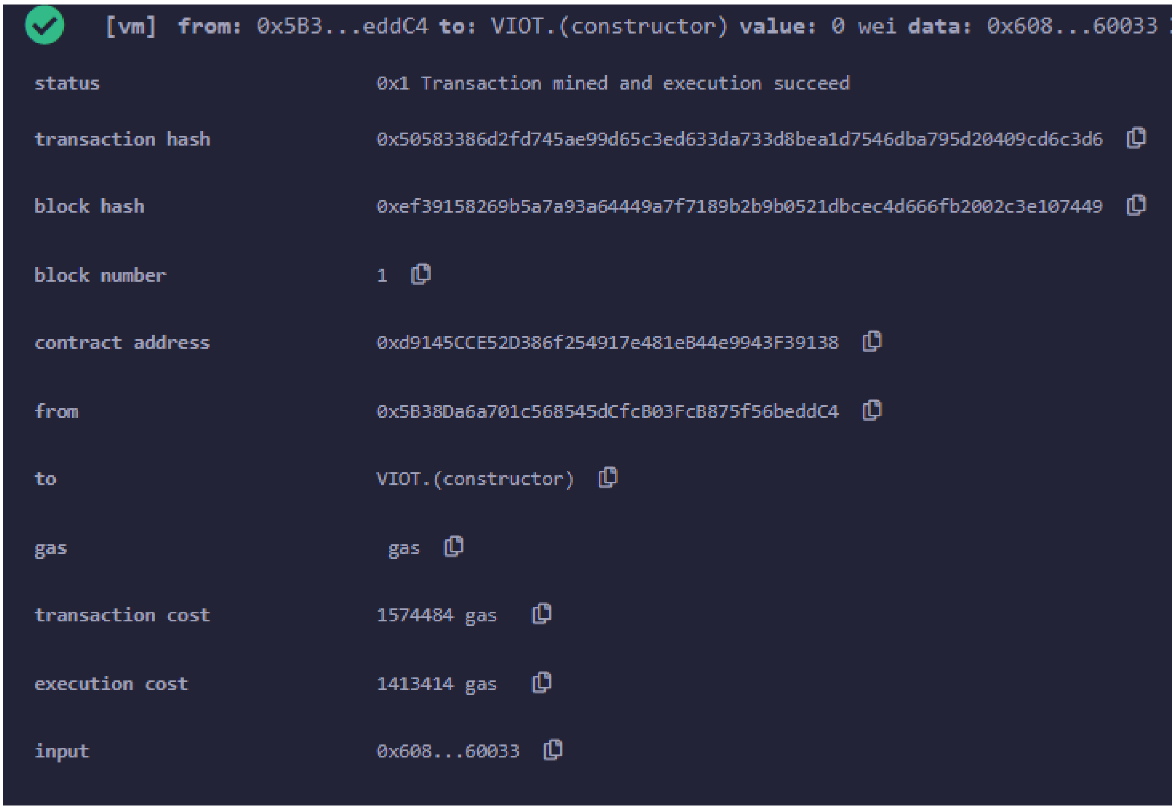
Table 7 BAN logic rules

Rule	Description
R1: $\frac{A\mid\equiv\#(X)}{A\mid\equiv\#(X,Y)}$	A believes that if X is fresh then (X,Y) is fresh
R2: $\frac{A\mid\equiv\overset{k}{\longmapsto}B, A\triangleleft\{X\}_k}{A\triangleleft X}$	A believes that k is public key then and A receives X encrypted using the public key then A receives X
R3: $\frac{A\mid\equiv\overset{k}{\longmapsto}B, A\triangleleft\{X\}_{k-1}}{A\triangleleft X}$	A believes that k is public key then and A receives X encrypted using the corresponding secret key then A receives X
R4: $\frac{A\mid\equiv\#(X), A\mid\equiv B\mid\equiv X}{A\mid\equiv A\overset{k}{\leftrightharpoons}B}$	A believes that if X is fresh and A believes B believes on X then A believes that A and B have a session key k

Table 8 Comparison of functionality and security requirements of proposed DLPA with other authentication schemes

Authentication Schemes	SR1	SR2	SR3	SR4	SR5	SR6	SR7	SR8	SR9	SR10	SR11	SR12
Feng, Xia, et al.[15]	Y	Y	-	Y	Y	-	-	-	-	-	-	-
Son, Seunghwan, et al.[7]	Y	-	-	Y	Y	Y	N	N	Y	Y	Y	Y
Lu, Zhaojun, et al.[16]	N	-	Y	-	Y	Y	-	-	-	-	-	Y
Eddine, Merzougui Salah, et al[10]	Y	-	-	Y	Y	Y	-	Y	-	-	Y	-
Lin, Chao, et al.[8]	Y	Y	N	Y	Y	Y	-	Y	Y	-	-	Y
Xu, Zisang, et al.[5]	N	-	-	Y	Y	-	-	-	-	-	Y	Y
Bagga, Palak, et al.[6]	N	-	Y	-	Y	Y	-	-	-	-	-	Y
Li, Xinghua, et al.[9]	Y	Y	-	-	-	-	N	N	-	-	-	-
Wei, Lu, et al.[4]	Y	-	-	-	Y	Y	Y	-	-	-	Y	-
Nandy, Tarak, et al.[17]	Y	N	Y	Y	Y	N	-	-	Y	Y	N	N
Proposed DLPA	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Y':Yes, 'N':No, '-':Not applicable, SR1:Mutual authentication, SR2:Unlinkability,SR3:Insider attack, SR4:Privacy and anonymity, SR5: Reply attack, SR6: Man in the middle attack, SR7: Denial of service attack, SR8: Distributed denial of service (DDoS attack), SR9: stolen-verifier attack, SR10: Offline password guessing attack, SR11: Forward and Backward secrecy, SR12: Impersonation attack



```
[vm]  from: 0x5B3...eddC4 to: VIOT.(constructor) value: 0 wei data: 0x608...60033
status                  0x1 Transaction mined and execution succeed
transaction hash        0x50583386d2fd745ae99d65c3ed633da733d8bea1d7546dba795d20409cd6c3d6
block hash              0xef39158269b5a7a93a64449a7f7189b2b9b0521dbcec4d666fb2002c3e107449
block number            1
contract address        0xd9145CCE52D386f254917e481eB44e9943F39138
from                    0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to                      VIOT.(constructor)
gas                      gas
transaction cost        1574484 gas
execution cost          1413414 gas
input                   0x608...60033
```

Fig. 3 Deployment of smart contract

The screenshot of the output while deploying the smart contract is given in the Figure 3. The gas consumption for deploying the smart contract and other details of the smart contract deployment are given in the output of the screenshot. For different functions in the smart contract the cost which is the gas consumption will be different and also with different values of input given to the function the gas consumption may vary.

8 Performance analysis

The performance analysis of the proposed lightweight (DLPA) is done and compared with other authentication protocols. The lightweight feature of the proposed authentication protocol can be analyzed using the communication cost and computation cost. The performance of the proposed authentication scheme is compared with other authentication schemes [4],[5],[6],[7],[10]. Table 9 shows the communication cost and Table 10 shows the computation cost. In some of the fields in the table it is shown like NA which means in that particular paper either V2V or V2RSU protocol is not

Table 9 Communication cost in bits

Authentication Scheme	V2V	V2RSU
Wei, Lu, et al.[4]	1664	1920
Xu, Zisang, et al.[5]	NA	4416
Bagga, Palak, et al.[6]	2912	2912
Son, Seunghwan, et al.[7]	NA	2880
Eddine, Merzougui Salah, et al[10]	2560	NA
Our DLPA Scheme	640	640

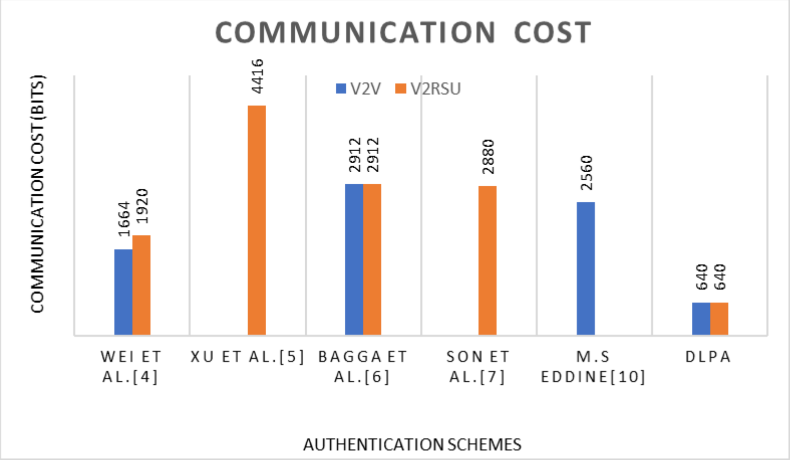


Fig. 4 Communication Cost Analysis

mentioned. From the performance analysis it is evident that the proposed scheme outperformed other schemes in terms of communication cost and computation cost.

8.1 Communication cost analysis

The messages transferred between the vehicle and between vehicle and RSU during authentication is calculated which is the communication cost. In the V2V authentication there are two communication between the vehicles V_i and V_j. In the first communication from V_i and V_j the message consists of M_5,and T_1. M_5 composed of two parameters B_{Vi} and C_{Vi}. Both of the parameters are stored with the output of an encryption. The time stamp is considered as 64 bits. Hence for the first communication the communication cost is 128+128+64=320 bits. In the V2V authentication protocol the communication from V_j to V_i consists of M_6,and T_2. M_6 consist of two parameters which are the encrypted values. The total cost for this communication is 128+128+64=320 bits. Hence for the V2V communication the total communication cost is 640 bits. For V2RSU communication also the same case as V2V hence the communication cost for V2RSU is also 640 bits. The communication cost analysis is shown in Table 9. The graphical representation of the communication cost is given in Figure 4.

Table 10 Computation cost in ms

Authentication Scheme	V2V	V2RSU
Wei, Lu, et al.[4]	4.964	4.964
Xu, Zisang, et al.[5]	NA	0.434
Bagga, Palak, et al.[6]	7.893	17.017
Son, Seunghwan, et al.[7]	NA	6.024
Eddine, Merzougui Salah, et al[10]	5.91505	NA
Our DLPA Scheme	0.23972	0.13166

8.2 Computation cost analysis

The computation cost is the total time taken to execute the functions used in the authentication scheme. The cryptographic functions used in the proposed authentication scheme are Hash function, ECC based encryption, and XOR operation. For the execution of XOR operation it only takes negligible value hence it can be avoided while calculating the computation cost. In the V2V authentication both the parties participating are mobile then for encryption we took the value 0.02415 ms and for hash we took the value 0.01163 ms. In the V2V authentication it consists of 5 hash functions, 4 encryption 4 decryption and 4 XOR operation. XOR function value is not added. Hence the total computation cost will be 4 hash+8 encryption, (5*0.01163)+(8*0.02415)=0.25135 ms. In the case of V2RSU authentication one of the participant is stationary (desktop system) and another participant is mobile. For desktop system the encryption value is 0.0023 ms and for hash function the value is 0.0013 ms. In the vehicle (mobile) part there are 3 hash operations and 4 ECC based encryption. Then the cost for mobile part is (3*0.01163)+(4*0.02415)=0.13149 ms. The cost for desktop system is the sum of 2 hash and 4 encryption, (2*0.0013)+(4* 0.0023)=0.0118 ms. Total computation cost for V2RSU authentication is 0.13149+0.0118=0.14329 ms. The computation cost analysis is given in Table 10. The graphical analysis of the computation cost is given in Figure 5.

8.3 Blockchain practical feasibility analysis

To verify the feasibility of the proposed DLPA, implemented the system in Ethereum IDE remix. The execution cost and the transaction cost of implementing each function is given in the Table 10. The execution cost is the amount of computational resources used to perform the computations in the the smart contract. Transaction cost is the total cost of submitting a transaction to the Ethereum network. The transaction cost comprises of both the execution cost and additional costs like transaction data storage, state changes, and other operations. checkIfRSUREgistered, checkIfVehicleRegistered, isVehicleAuthenticated, are the read only or view functions in the implemented smart contract. The read only functions does not make any state change so that there is no transaction is mined for the same. Hence there is no transaction cost for the read only functions in the smart contract.

Table 11 Implementation cost of smart contract functions

Function	Execution Cost	Transaction Cost
registerVehicle	74520	96472
registerRSU	73001	94509
checkIfRSURegistered	2964	-
checkIfVehicleRegistered	2920	-
isVehicleAuthenticated	5162	-
authenticate	55404	78548

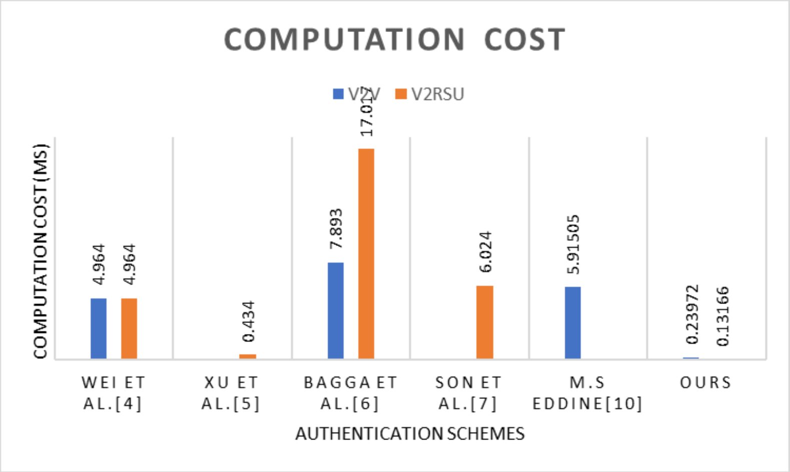


Fig. 5 Computation Cost Analysis

9 Conclusion and Future Scope

In this paper, a lightweight blockchain-enabled authentication and key agreement scheme (DLPA) is proposed. Vehicle driver's anonymity is preserved in this scheme by using pseudo-id to users. In the DLPA scheme, as the third party is eliminated we could achieve peer-to-peer communication thereby reducing the delay in communication. In addition to that The proposed DLPA scheme there is no need for re-authentication when vehicles move from from RSU unit to another. We have done the formal and informal security analysis and proved that our DLPA has resistance against known attacks. While analyzing the performance of other authentication protocols based on the communication and computation cost, either the communication or computation cost will be high, there is no such system that has optimum value for both. The proposed system exhibits good performance with minimum value for communication and computation cost when compared with other authentication protocols. In the future, we will design a batch authentication protocol for VANET.

Declarations

Ethics Approval

Not Applicable

Data Availability

No data was used for the research described in the article.

Funding

No funding is used for the research described in the article

Consent to publish

Not Applicable

References

[1] Lin, Chao, et al. ”BCPPA: A blockchain-based conditional privacy-preserving authentication protocol for vehicular ad hoc networks.” IEEE Transactions on Intelligent Transportation Systems 22.12 (2020): 7408-7420.

[2] D. B. Rawat, B. B. Bista, G. Yan and S. Olariu, ”Vehicle-to-Vehicle Connectivity and Communication Framework for Vehicular Ad-Hoc Networks,” 2014 Eighth International Conference on Complex, Intelligent and Software Intensive Systems, Birmingham, UK, 2014, pp. 44-49, doi: 10.1109/CISIS.2014.7.

[3] Koblitz, Neal. “Elliptic curve cryptosystems.” Mathematics of Computation 48 (1987): 203-209.

[4] Wei, Lu, et al. ”A Decentralized Authenticated Key Agreement Scheme Based on Smart Contract for Securing Vehicular Ad-hoc Networks.” IEEE Transactions on Mobile Computing (2023).

[5] Xu, Zisang, et al. ”A blockchain-based roadside unit-assisted authentication and key agreement protocol for internet of vehicles.” Journal of Parallel and Distributed Computing 149 (2021): 29-39.

[6] Bagga, Palak, et al. ”Blockchain-based batch authentication protocol for Internet of Vehicles.” Journal of Systems Architecture 113 (2021): 101877.

[7] Son, Seunghwan, et al. ”Design of blockchain-based lightweight V2I handover authentication protocol for VANET.” IEEE Transactions on Network Science and Engineering 9.3 (2022): 1346-1358.

[8] Lin, Chao, et al. ”BCPPA: A blockchain-based conditional privacy-preserving authentication protocol for vehicular ad hoc networks.” IEEE Transactions on Intelligent Transportation Systems 22.12 (2020): 7408-7420.

[9] Li, Xinghua, et al. ”An unlinkable authenticated key agreement with collusion resistant for VANETs.” IEEE Transactions on Vehicular Technology 70.8 (2021): 7992-8006.

[10] Eddine, Merzougui Salah, et al. "EASBF: An efficient authentication scheme over blockchain for fog computing-enabled internet of vehicles." Journal of Information Security and Applications 59 (2021): 102802.

[11] Yao, Yingying, et al. "BLA: Blockchain-assisted lightweight anonymous authentication for distributed vehicular fog services." IEEE Internet of Things Journal 6.2 (2019): 3775-3784.

[12] Burrows, Michael, Martin Abadi, and Roger Needham. "A logic of authentication." ACM Transactions on Computer Systems (TOCS) 8.1 (1990): 18-36.

[13] Adeli, Morteza, et al. Xperbp: a cloud-based lightweight mutual authentication protocol." Peer-to-Peer Networking and Applications (2023): 1-18.

[14] Cremers, Cas JF. "The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols: Tool Paper." International conference on computer aided verification. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[15] Feng, Xia, et al. "An efficient privacy-preserving authentication model based on blockchain for VANETs." Journal of Systems Architecture 117 (2021): 102158.

[16] Lu, Zhaojun, et al. "A blockchain-based privacy-preserving authentication scheme for VANETs." IEEE Transactions on Very Large Scale Integration (VLSI) Systems 27.12 (2019): 2792-2801.

[17] Nandy, Tarak, et al. "An enhanced lightweight and secured authentication protocol for vehicular ad-hoc network." Computer Communications 177 (2021): 57-76.

[18] Jain, Shashank Mohan. "Introduction to Remix IDE." A Brief Introduction to Web3: Decentralized Web Fundamentals for App Development. Berkeley, CA: Apress, 2022. 89-126.

[19] Bitcoin, Nakamoto S. "Bitcoin: A peer-to-peer electronic cash system." (2008).

[20] Lu, Yang. "The blockchain: State-of-the-art and research challenges." Journal of Industrial Information Integration 15 (2019): 80-90.

[21] Li, Xuehan, et al. "Bdra: Blockchain and decentralized identifiers assisted secure registration and authentication for vanets." IEEE Internet of Things Journal (2022).

[22] Feng, Qi, et al. "BPAS: Blockchain-assisted privacy-preserving authentication system for vehicular ad hoc networks." IEEE Transactions on Industrial Informatics 16.6 (2019): 4146-4155.

[23] Tandon, Righa, Ajay Verma, and P. K. Gupta. "D-BLAC: A dual blockchain-based decentralized architecture for authentication and communication in

VANET." Expert Systems with Applications 237 (2024): 121461.

[24] Dwivedi, Sanjeev Kumar, et al. "Design of Blockchain and ECC-Based Robust and Efficient Batch Authentication Protocol for Vehicular Ad-Hoc Networks." IEEE Transactions on Intelligent Transportation Systems (2023).

[25] Wu, Anmulin, Yajun Guo, and Yimin Guo. "A decentralized lightweight blockchain-based authentication mechanism for Internet of Vehicles." Peer-to-Peer Networking and Applications (2023): 1-14.

[26] Masud, Mehedi, et al. "A user-centric privacy-preserving authentication protocol for IoT-AmI environments." Computer Communications 196 (2022): 45-54.

[27] Wang, Weizheng, et al. "Blockchain and PUF-based lightweight authentication protocol for wireless medical sensor networks." IEEE Internet of Things Journal 9.11 (2021): 8883-8891.

[28] Shukla, Saurabh, et al. "Identification and authentication in healthcare internet-of-things using integrated fog computing based blockchain model." Internet of Things 15 (2021): 100422.

[29] Lu, Zhaojun, et al. "BARS: A blockchain-based anonymous reputation system for trust management in VANETs." 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE). IEEE, 2018.

[30] Sutrala, Anil Kumar, et al. "On the design of conditional privacy preserving batch verification-based authentication scheme for internet of vehicles deployment." IEEE Transactions on Vehicular Technology 69.5 (2020): 5535-5548.

[31] Ali, Ikram, et al. "A blockchain-based certificateless public key signature scheme for vehicle-to-infrastructure communication in VANETs." Journal of Systems Architecture 99 (2019): 101636.

[32] Chen, Jiageng, Mohammad Saiful Islam Mamun, and Atsuko Miyaji. "An efficient batch verification system and its effect in a real time VANET environment." Security and Communication Networks 8.2 (2015): 298-310.

[33] Zhang, Jing, et al. "DBCPA: Dual blockchain-assisted conditional privacy-preserving authentication framework and protocol for vehicular ad hoc networks." IEEE Transactions on Mobile Computing (2022).