Blockchain-based Secure Authentication Protocol for Vehicular Ad-hoc Networks

Abstract

Vehicular Ad Hoc Networks (VANETs) facilitate communication among vehicles and roadside units (RSUs) to enhance road safety and traffic efficiency. However, ensuring the security and privacy of communications in VANETs remains a significant challenge. In this paper, we propose a Blockchain-based Conditional Privacy-Preserving Authentication (BCPPA) protocol for VANETs to address these challenges. The protocol leverages Blockchain technology for secure data storage and employs smart contracts for authentication and revocation processes. Furthermore, we introduce a key derivation algorithm to alleviate the burden of key pre-storing in vehicle On-Board Units (OBUs). Our protocol utilizes modified Elliptic Curve Digital Signature Algorithm (ECDSA) with batch verification to enhance verification efficiency in VANETs. We provide a detailed description of the protocol, along with security and performance analysis, demonstrating its effectiveness in ensuring secure and privacy-preserving communications in VANETs.

Keywords: Blockchain, ROR model, Elliptic curve cryptography, Vehicular Adhoc Network(VANET), Smart Contract.

1. Introduction

VEHICULAR Ad Hoc Network (VANET) is a self-organized ad hoc network where cars and roadside units (RSUs) are often connected via wireless communications. An On-Board Unit (OBU) (some wireless communication device) is installed in every participating vehicle, allowing it to connect with other adjacent cars and RSUs. For data exchange, the RSUs can further link to the backbone network, for instance via the Internet. A typical VANET network model includes a traffic control center(TCC), RSU, vehicle, and internet. There are three basic ways to communicate: a wired or wireless link, vehicle-to-vehicle communication, and vehicle-to-RSU communication. Vehicles and/or RSUs are connected to the Internet through a wired or wireless connection, while the other two wireless communications are managed by the Dedicated Short Range Communication (DSRC) protocol to provide short-range communication [1]. Using the OBUs and DSRC, cars may interact with one another or with RSUs to provide information on the current road traffic circumstances, such as the weather and traffic situation, as well as their driving statuses, such as their location and speed. Implementing a prompt reaction (such as rerouting to minimize traffic building), can assist the cars in efficiently avoiding traffic jams or potential traffic accidents [2]. TCC may get these traffic alerts from the RSUs through the Internet and promptly take the necessary action (e.g. adjusting traffic lights). However, certain possible security issues should not be disregarded, particularly when using wireless communication because it is more prone to attack than conventional connection. Attackers may target this transportation system, for instance, and try to stir up civil discontent by intercepting, altering, replaying, or deleting transmission signals. To prevent impersonation or malicious alteration, it is important to assure the authenticity, validity, and integrity of sent communications. Successful assaults have the potential to cause deaths in the real world.

We also need to take into account preserving the privacy of automobiles (and their owners/drivers), even if message authentication can help to minimize some of these assaults. A vehicle's identification will be known, for instance, when it shares its traffic status with another RSU or vehicle. Such data might be mined by an attacker to determine the path taken by the vehicle. Additionally, in accordance with the IEEE Standard [3], cars often broadcast signals concerning their current driving state and road traffic circumstances on a recurring basis every 100 to 300 milliseconds. The traceability of cars is facilitated by the broadcast message's frequency. Obviously, there might be

issues with safety and privacy.

Conditional privacy-preserving authentication (CPPA) is one of the technologies that have been suggested to allow secure communications on VANETs [4]. The privacy of the vehicle should be conditionally secured in a CPPA protocol in the context of VANET. This suggests that the car is anonymous to most entities, however, a reliable entity can discover the vehicle's true identity. This makes it possible to identify misbehaving vehicles (such as those that have delivered fake traffic updates) and penalize them appropriately.

PKI-based [5] and ID-based [6, 7, 8] are two main categories into which existing CPPA protocols for VANETs may be divided. The latter group doesn't have the problems associated with key/certificate pre-loading and revocation that PKI-based protocols have, and certain schemes, like those in [1, 5, 9], also offer batch verification to boost speed. However, these ID-based solutions lead to brand-new issues including the difficulty of revocation of the private key for the vehicle. These problems, along with others like the necessity of concept hardware and frequent contacts, are still present in the recently proposed blockchain-based CPPA (BCPPA) protocols [10]. We are thus driven to provide a successful PKI-based BCPPA protocol that resolves the aforementioned problems.

1.1. Related Works

Raya and Hubaux [11] suggested the idea of CPPA to solve security and privacy issues in VANETs. Additionally, they provided an actual CPPA protocol that makes use of anonymous certificates and may be implemented using a modified PKI. To achieve anonymous authentication (covering the true identity of the vehicle), several public/private key pairs and matching certificates are pre-loaded onto cars' OBUs. The vehicle should select a public/private key pair at random when it wants to publish its traffic status in order to authenticate messages using signatures. However, this will result in large storage costs (i.e., expenses associated with keeping keys and certificates) for both vehicles and the applicable authority, as well as substantial costs associated with carrying out key and certificate revocation. Lu et al. [4] presented a unique CPPA protocol using RSU-based anonymous certificates to address the aforementioned shortcomings. An RSU will be seen when the car arrives there. A temporary anonymous certificate for authentication can be obtained. Although one can occasionally get conditional privacy Online RSUs are heavily used for signing and verifying digital signatures, obtaining fresh anonymous certificates, and other tasks. In VANETs, this is not productive. Similar costs are incurred by the CPPA procedures provided by Zhang et al. [5] and RSUs for certificates in both cars and RSUs. In fact, one can see from the literature that key/certificate management complexity is a prevalent problem in current CPPA protocols. Thus, attempts to create ID-based CPPA protocols have been made, including those that use ID-based signatures [12], software-based solutions [8], pseudoID-based solutions [1], and others [13]. All of these protocols either concentrate on enhancing certain already-existing solutions to meet the necessary security criteria or on enhancing CPPA's functionality to accommodate VANET applications. Most of these protocols, however, are either unsuitable for multi-cloud environments or depend on optimal hardware. Zhang et al. [14] offered a Chinese Remainder Theorem-based CPPA Scheme for resolving the first problem, while Zhang et al. [9] developed a novel CPPA scheme employing multiple trusted authority one-time identity-based aggregate signatures, both of which merely calls for plausible tamper-proof devices. As an alternative to the latter, Cui et al. [26] created a CPPA protocol that is resilient and adaptable and can accommodate the VANETs' growing need for a variety of services. However, there is still one common intractability of revoking vehicle private keys in various ID-based methods, a subject that is comparatively understudied.

Several Blockchain-based CPPA (BCPPA) protocols have been suggested to address the issues with PKI-based solutions, such as the difficulty of revoking certificates and the lack of transparency of trusted authorities. To propose a novel BCPPA protocol with privacy protection and effective certificate revocation, for instance, Lu et al. [15] integrated blockchain and Merkle Patricia Tree. However, this protocol necessitates frequent interactions between vehicles and the certificate authority in order to generate anonymous certificates. An ID-based BCPPA protocol with traceable anonymity was designed by Zheng et al. [10] using pseudonym technology, however, it requires optimal hardware and cannot withstand a corrupted certificate authority.

1.2. Introduction of Blockchain Technology

The blockchain has a verifiable and accurate record of every single transaction that has ever happened. The most commonly known application of blockchain technology is Bitcoin which is a decentralized peer-to-peer digital currency. Even though the digital currency bitcoin is very contentious, the underlying blockchain has run without a glitch and has a wide range of uses in both the financial and non-financial sectors. The central hypothesis of the blockchain enables distributed agreement in the online digital environment. An immutable record can be created in a public ledger so that interested parties can be sure that a digital event happened. It sets the path for the transformation from a centralized to a decentralized open, and scalable digital economy. The potential of this cutting-edge technology is tremendous, and the revolution in this area has just started. We can also use blockchain technology in health care to maintain medical data security and integrity. In this paper, we store medical data in the blockchain. Figure 1 depicts the blockchain transaction mechanism.

Implementing blockchain technology in ad-hoc vehicle networks can offer several benefits, including enhanced security, transparency, and decentralized control. Here's a conceptual overview of how blockchain technology could be applied in such a scenario:

Decentralized Network: Ad-hoc vehicle networks are often characterized by dynamic and decentralized communication among vehicles. Blockchain can facilitate a distributed ledger that records transactions and interactions among vehicles without the need for a central authority.

Secure Communication: Blockchain's cryptographic features can ensure secure communication between vehicles. Transactions and messages can be encrypted, and the decentralized nature of the blockchain makes it resistant to tampering.

Smart Contracts: Smart contracts, self-executing contracts with the terms of the agreement directly written into code, can be used to automate and enforce certain conditions in ad-hoc vehicle networks. For example, smart contracts could automatically execute payments for tolls, parking, or other services.

Identity and Authentication: Blockchain can provide a secure and immutable way to manage identities and authentication in ad-hoc networks. Each vehicle could have a unique identity stored on the blockchain, and cryptographic keys could be used for secure access and communication.

Data Integrity and Transparency: The distributed nature of the blockchain ensures that data is stored across multiple nodes. This improves data integrity and transparency, as any attempt to manipulate information would require the consensus of the majority of nodes.

Supply Chain and Maintenance: Blockchain can be used to create a transparent and traceable record of a vehicle's maintenance history. This ensures that all relevant parties have access to accurate and up-to-date information, reducing the risk of fraud and improving overall safety.

Tokenization and Transactions: Cryptocurrencies or tokens on the blockchain can be used for transactions within the network. This could include payments for services, tolls, or even a form of incentive system to encourage positive behavior within the ad-hoc network. Consensus Mechanisms: Choosing an appropriate consensus mechanism is crucial for the functioning of the blockchain. Depending on the requirements of the ad-hoc network, consensus mechanisms like Proof of Work (PoW), Proof of Stake (PoS), or other consensus algorithms can be selected. Scalability and Performance: It's essential to consider the scalability and performance of the blockchain network, especially in a dynamic and fast-paced environment like ad-hoc vehicle networks. Choosing or developing a blockchain solution that can handle a high volume of transactions efficiently is crucial. Implementing blockchain technology in ad-hoc vehicle networks presents challenges but also opens up possibilities for creating more secure, transparent, and efficient systems for transportation and communication. Careful consideration of the specific requirements and collaboration between stakeholders are key to successful implementation.

1.3. Motivation and Contribution

Blockchain, a distributed ledger technology [16], offers secure data storage capabilities suitable for storing sensitive information such as certificates or system settings. This stored data can be accessed by vehicles or Roadside Units (RSUs) for authentication purposes. In this work, we explore the utilization of smart contracts to establish relationships among relevant information and facilitate revocation when necessary. Additionally, we propose a key derivation algorithm aimed at eliminating the need for pre-storing a large number of keys in vehicle On-Board Units

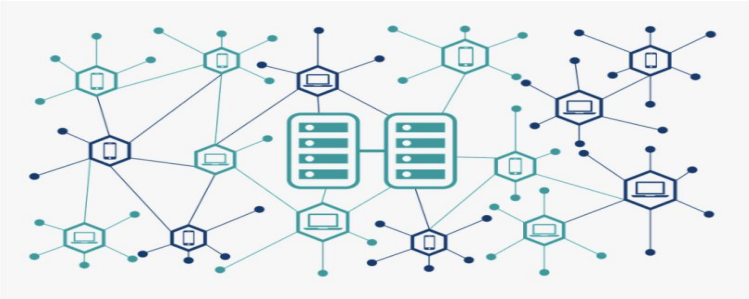

Figure 1: Blockchain Mechanism

(OBUs). This solution addresses the challenges associated with escrow, ensuring that regularly updated plans rely on confidential information and maintain a reasonable OBU overhead.

Furthermore, we introduce a specific protocol, Blockchain-based Certificateless Public Key Encryption with Aggregate (BCPPA), which leverages ECDSA, a widely used digital signature technique in Public Key Infrastructure (PKI) systems. Our protocol incorporates a modified version of ECDSA that supports batch verification, aimed at reducing verification costs in Vehicular Ad-Hoc Networks (VANETs). Notably, our BCPPA protocol can accommodate additional signatures while still benefiting from batch verification. Finally, we present security and performance evaluations to demonstrate the feasibility and effectiveness of our proposed approach.

1.4. Paper Organization

The remainder of this paper is organized as follows: In Section 2, we provide essential preliminaries necessary to describe the proposed framework. Section 3 elaborates on the proposed framework and the underlying system architecture. Section 4 presents a comprehensive security assessment of the proposed framework. The performance analysis of the suggested framework is detailed in Section 5. Lastly, we conclude our work, summarizing the contributions and potential future directions.

2. Preliminaries

In this part, We describe the important mathematical preliminaries related to our proposed work.

2.1. Elliptic Curve Cryptography

An elliptic curve cryptography is a type of asymmetric key or public key cryptography that uses an elliptic curve over a large finite field. ECC can deliver greater security and improved performance with a lower key size as compared to current public-key cryptography. Suppose p be a big prime number, $u, v \in F_p$, and $4u^3 + 27v^2 \neq 0(modp)$. Then, a nonsingular elliptic curve $E_p(u, v)$ over a finite field F_p is defined by the equation as below:

$$E_p(u, v) : y^2 = x^3 + ux + v(modp)$$

Suppose G is a point, which we call a base point, on the elliptic curve $E_p(u, v)$. Then, scalar multiplication operation on $E_p(u, v)$ is defined as
$m \cdot G = G + \cdots + G(m\ times)$,
where m is a non-negative scalar lie in F_p. The following issues serve as the foundation for the security ECC.

2.2. Elliptic Curve Discrete Logarithm Problem (ECDLP)

The Elliptic Curve Discrete Logarithm Problem (ECDLP) is a fundamental computational problem in elliptic curve cryptography. Given two points G_1 and G_2 on an elliptic curve $E_p(a, b)$, where G_2 is equal to scalar multiplication of G_1 by an integer m such that $G_2 = mG_1$, the task is to find the scalar m. Mathematically, this problem can be expressed as finding m given G_1, G_2, and the curve parameters a, b, and p. Formally:

$$G_2 = mG_1$$

where $G_1, G_2 \in E_p(a, b)$ and $m \in F_p$. The ECDLP is considered hard, meaning that there is no known efficient algorithm to solve it. This hardness property forms the basis of security for many cryptographic schemes based on elliptic curves.

2.3. Elliptic Curve Diffie-Hellman Problem (ECDHP)

The Elliptic Curve Diffie-Hellman Problem (ECDHP) is another important computational problem in elliptic curve cryptography, building upon the Discrete Logarithm Problem. In ECDHP, we consider three points G_1, xG_1, and yG_1 lying on an elliptic curve $E_p(u, v)$. The challenge is to compute xyG_1 given G_1, xG_1, and yG_1. Mathematically, the problem can be stated as follows:

Given G_1, xG_1, and yG_1, find xyG_1.

The security of the Elliptic Curve Diffie-Hellman key exchange protocol relies on the assumption that the ECDHP is computationally hard. This means that there is no efficient algorithm known to compute xyG_1 from G_1, xG_1, and yG_1 without knowing the discrete logarithms of x and y.

2.4. One-Way Collision-Resistant Hash Function

A one-way collision-resistant hash function, as formally defined by Brown et al. in 2005 [17], is a cryptographic function that maps an input of arbitrary length to a fixed-length output, typically denoted as $H : \{0, 1\}^* \rightarrow \{0, 1\}^n$. This function is deterministic, meaning the same input will always produce the same output, and its output is a binary string of length n.

Let's break down the formal definition and properties of a one-way collision-resistant hash function:

- **Definition**: Given an arbitrary length input string $g \in \{0, 1\}^*$, the hash function $H(g)$ produces a binary string in $\{0, 1\}^n$.
- **Randomized Procedure**: The hash function operates as a randomized procedure, meaning it should be computationally infeasible to predict the output for a given input without actually performing the computation.
- **Collision Resistance**: A hash function is considered collision-resistant if it is computationally difficult to find two distinct inputs c and d such that $H(c) = H(d)$. In other words, it should be challenging for an adversary to find a collision pair (c, d) where $c \neq d$ but their hash values are the same.
- **Adversary's Opportunity**: The security of a hash function against collisions is measured by the adversary's advantage in finding collisions. The adversary's advantage $Adv_A^{HASH}(t)$, where t represents the execution time, is defined as the probability that the adversary $\mathcal{A}$ successfully finds a collision pair (c, d) when given the opportunity to interact with the hash function. Mathematically, it is expressed as:

$$Adv_A^{HASH}(t) = \text{Prob}\left[(c, d) \xleftarrow{R} A : c \neq d \text{ and } H(c) = H(d)\right]$$

  where $(c, d) \xleftarrow{R} A$ denotes that the adversary $\mathcal{A}$ randomly selects the pair (c, d).
- **Security Requirement**: A hash function is considered collision-resistant if its advantage against collision-finding attacks is negligible. Formally, if $Adv_A^{HASH}(t) \leq \epsilon_{HASH}$ for any sufficiently small $\epsilon_{HASH} \geq 0$, then the hash function $H(g)$ is considered resistant to collisions.

For example, let's consider a hash function H with a 256-bit output length. Any input message g, regardless of its length, will be mapped to a fixed-size output string of 256 bits. The security of this hash function lies in the computational difficulty of finding two different messages that produce the same hash output. If an adversary's advantage in finding such collisions is negligible, the hash function is considered collision-resistant.

3. Proposed Protocol and System Architecture

In this section, we discuss the proposed framework and its system architecture

3.1. Architecture of Proposed Protocols

In Fig. 2, the system model is displayed. The proposed system consists of the following entities: Cloud Assisted Server(CAS), Road Side Unit(RSU), and Adhoc Vehicles(AV). The description of important entities in this framework is as follows:

1. CAS: CAS is a trusted authority. It is responsible for the registration of other entities in the system, including RSU, AV. CAS also generates cryptographic parameters, initializes the blockchain network, and deploys smart contracts to the blockchain. The information of registered entities, mainly their public key information, is written into smart contracts and deployed over the blockchain.
2. RSU: The DSRC protocol is used by the RSU, a roadside infrastructure, to connect with OBUs. It also performs the function of a complete node, which means that it stores all of the blockchain's transaction data and offers the APIs needed to retrieve transactions and activate chained smart contracts (e.g. the test chain rinkeby of Ethereum). Here, we assume that RSUs are trustworthy entities and would not offer fake APIs.
3. Vehicle: The vehicle has an internal processing unit called an OBU that can handle the DSRC protocol and is tamper-proof. The OBU in our concept is viable in this case since the secrets that are kept there may be updated regularly. Each OBU holds a private seed that is used by a key derivation algorithm to generate the vehicle's one-time private key, thereby avoiding the need to keep a large number of private keys. The OBU communicates its traffic status to neighboring cars and RSUs often while the vehicle is in motion. Here, the OBU primarily engages in V2R communication with the RSU to retrieve transactions and engages in V2V communication with other vehicles.
4. Smart Contract: A blockchain is made up of so-called transactions, which are stored in an unchangeable, irrefutable, and verifiable manner by the Blockchain Network. For example, instead of preloading all the certificates in the OBUs, we embed public certificates into the transaction so that the vehicles may receive the goal certificates from the blockchain. Here, we suggest leveraging an established public blockchain (like Ethereum) for our idea, which anybody may join and help develop. RSUs join this network as complete nodes, providing services (such as obtaining transactions and initiating the smart contract) for adjacent cars, as was already indicated.

3.2. Adversary Model (including both internal and external adversaries)

We employ the Dolev-Yao (DY) model [18] to assess the security of the proposed protocol. In this model, an attacker can exploit insecure channels to intercept, delete, and modify communications. The adversary is endowed with the following capabilities:

- Impersonation, forgery attacks, and man-in-the-middle (MITM) attacks can be executed by the adversary.
- Through power analysis techniques, the adversary can obtain the identity of a legitimate ad-hoc vehicle and gather all associated information.
- A valid or privileged ad-hoc vehicle registered within the system may act as a malicious adversary.

Furthermore, we consider the Canetti-Krawczyk (CK) model [19], which imposes more stringent conditions compared to the DY model. According to the CK paradigm, a malicious adversary can potentially compromise secure data such as private keys, passwords, identities, session secrets, etc.

3.3. Proposed Schemes

System startup, registration, login and authentication, password and biometrics update, and revocation phase are the five phases of the proposed approach. Table **??** lists the primary notations used in the system. Figure 2 illustrates the system's process.

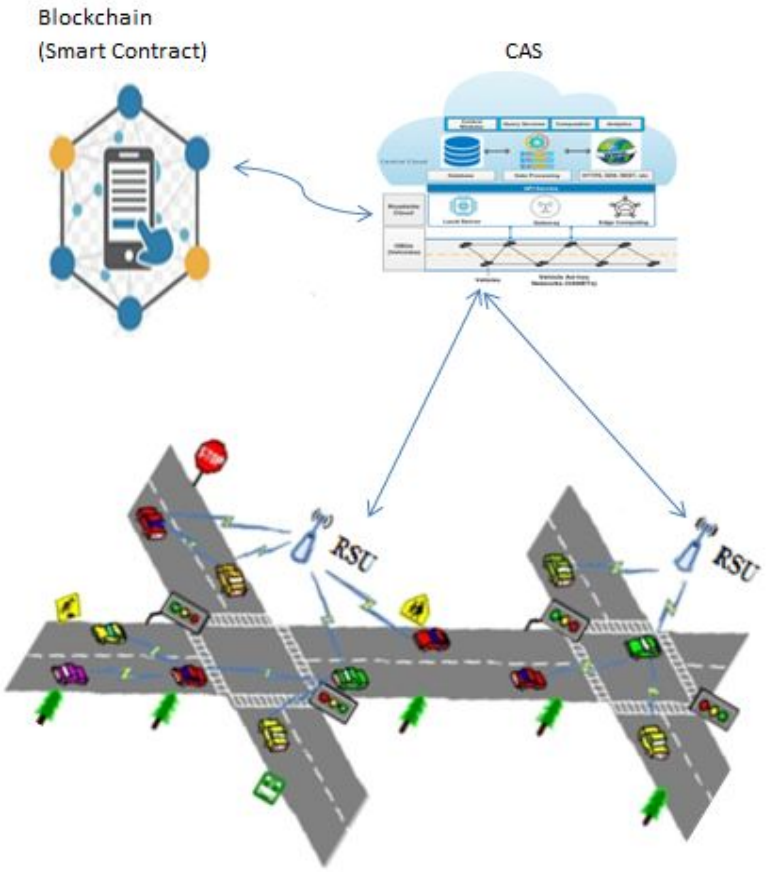

Figure 2: Proposed System Model

3.3.1. Initialization Phase

In this stage, CA sets up the system environment, deploys the smart contracts, creates the blockchain, and configures the cryptographic settings.

1) System Parameters Generation: Assume that q is a prime number of length l-bit, where l-bit is the security parameter. A cyclic group G of order n is preferred by RC over an elliptic curve E/F_q. G is generated at point P. RC assigns $P_{pub} = sP$ as its public key and selects $s \in Z_n$ randomly to serve as its secret key. RC releases the public parameters (P, G, P_{pub}) while keeping s a secret.

2) Blockchain Initialization: RC either starts its blockchain consortium or joins an already-existing one like Hyperleger Fabric. Keep in mind that in a consortium blockchain network, each blockchain node has been verified by the developer before joining and is thus presumed to be trustworthy.

3) Smart Contract Deployment: The smart contracts are sent to RC, which then executes them on the blockchain. The miners will provide a unique address to a smart contract after they have verified it. Permitted nodes can access the smart contract and use its features by submitting a transaction to the address.

3.3.2. Registration Phase

In this phase, FN, CS, and the AV register with the CAS. For each entity, CAS produces a set of public-private key pairs. By executing the smart contract, the private keys are communicated in secret while the public key details are posted on the blockchain. Each smart device registers with the RSU of the fog domain to which it is subject.

1) AV Registration: To register AV with the CAS through a secure channel we follow the following steps:

1. AV submits a registration request including his identity ID_v to the CAS.
2. CAS receives ID_v, chooses a random number $a_v \in Z_q$ and calculates $PK_v = a_v g, PID_v = h(PK_v||ID_v), R_v = a_v + h(PID_v||R_v), T_v = R_v g$. Then CAS sends (PK_v, R_v) to the smart contract. The smart contract inserts(PK_v, PID_v, T_v) into their database.
3. On receiving (PK_v, R_v), AV computes $R_v.g = PK_v + h(PID_v||PK_v)P_{pub}$,chooses password pw_v. AV calls the fuzzy extractor F = (Gen, Rec), imprints her biometric template B_v, and obtains (α_v, β_v), $Gen(B_v)$. Now, AV computes $V_1 = h(ID_v||pw_v||\alpha_v) \bigoplus R_v$ and $pw^*{}_v = h(pw_v||alpha_v||R_v)$. AV stores $(V_1, pw^*{}_v, \beta_v)$ in the memory of her mobile device or on a smart card.

Table 1: AV Registration Phase via Secure Channel

AV	**CAS**	**Smart Contract**
Selects ID_v		
Sends(ID_v)		
$\cdots\cdots\cdots\cdots\cdots\rightarrow$		
	Receives ID_v	
	Choose $a_v \in Z_q^*$	
	Calculates $PK_v = a_v g$	
	Compute $PID_v = h(PK_v||ID_v)$	
	Compute $R_v = a_v + h(PID_v||PK_v)$	
	Compute $T_v = R_v g$	
	Verify smart contract$\cdots\cdots\cdots\cdots\cdots\rightarrow$	Insert $PKT(AV||PID_v||T_v)$
	Sends $\{PK_v, R_v\}$	Stores (ID_v, PID_v, T_v) in PKT
	$\leftarrow\cdots\cdots\cdots\cdots\cdots$	
Receives $\{PK_v, R_v\}$		
Compute $R_v g = PK_v + h(PID_v||PK_v)P_{pub}$		
Chooses password PW_v		
Imprint bio metric B_v		
Generate $(\alpha_u, \beta_u) \leftarrow Gen(B_v)$		
Computes $V_1 = h(ID_v||PW_v||\alpha_u) \bigoplus R_v$		
Computes $PW_v^* = h(PW_v||\alpha_v||R_v)$		
Stores (V_1, PW_v^*, β_v)		

2) RS-unit Registration: In this phase, each RSU registers with the CAS as follows:

1. RSU sends its identity ID_{RS} to CAS
2. on receiving ID_{RS}, CAS randomly picks $b \in Z_q$ and computes $PK_{RS} = b.g, PID_{RS} = h(ID_{RS}||PK_{RS}), R_{RS} = b + h(PID_{RS}||PK_{RS}), T_{RS} = R_{RS}.g$. Now, CAS sends (R_{RS}, PK_{RS}) to RSU via a secure channel. CAS invokes $PKT(RS, PID_{RS}, T_{RS})$ to upload RSU pseudo-identity PID_{RS} and the public key information to the blockchain.
3. On receiving the key information (R_{RS}, PK_{RS}), RS computes $R_{RS}.g = PK_{RS} + h(PID_{RS}||PK_{RS})P_{pub}$ and stores R_{RS}

   The above process is depicted in Table 2.

3.3.3. Login and Authentication Phase

Authentication Between AV and RSU: AV logs into RSU and negotiates the session key as follows:

1. AV inputs the identity ID'_v, password pw'_v, and imprints the fingerprint B'_v on the mobile device equipped with fingerprint sensor. AV reproduces $\alpha_v \leftarrow Rec(B'_v, \beta_v)$ and computes $V'_1 = h(ID'_v||pw'_v||\alpha_v) \bigoplus R_v{}'$. Then, it verifies $V'_1 = V_1$, if it is true, then calculates $pw_v{}^{*\prime} = h(pw'_v||\alpha'_v||R'_v)$ and verifies $pw_v{}^{*\prime} = pw_v{}^*$. If not, AV terminates the process.
2. AV computes $PID_{RS} = h(ID_{RS})$, and invokes $QueryPKT(RS, PID_{RS})$ to obtain RSU public key T_{RS}.
3. AV randomly chooses $c \in Z_q^*$ and calculates $N = c.g, D_v = h(cT_{RS}||T_1) \bigoplus PID_v, P_v = c + h(PID_v||N||T_1)$, where T_1 is current timestamp. AV sends (N, D_v, P_v, β_v) to RSU.

Table 2: RS-Unit Registration Phase via Secure Channel

RSU	**CAS**	**Smart Contract**
Selects ID_{RS}		
Sends(ID_{RS}) · · · · · · · · · · · · · · · · →		
	Receives ID_{RS}	
	Choose $b \in Z_q^*$;	
	Compute $PK_{RS} = b.g$	
	Compute $PID_{RS} = h(ID_{RS} \| PK_{RS})$	
	Compute $R_{RS} = b + h(PID_{RS} \| PK_{RS})$	
	Compute $T_{RS} = R_{RS}.g$	
	Verify smart contract· · · · · · · · · · · · · · · · →	Insert $PKT(RS \| PID_{RS} \| T_{RS})$
	Sends$\{R_{RS}, PK_{RS}\}$	Stores (RS, PID_{RS}, T_{RS}) in PKT
	← · · · · · · · · · · · · · · · ·	
Receives $\{R_{RS}, PK_{RS}\}$		
Computes $R_{RS}.g = PK_{RS} + h(PID_{RS} \| PK_{RS})P_{pub}$		
Store R_{RS}		

4. On receiving AV's login request, RSU recovers AV's pseudo-identity $PID_v = D_v \bigoplus h(R_{rs}.N \| T_1)$. RSU invokes $QueryPKT(Eu, PID_u)$ to extract T_v from the PKT. RSU calculates $P_v g = N + h(PID_v \| N \| T_2) T_v$.
5. RSU randomly chooses $d \in z_q^*$ and computes $M = dg, P_r s = h(PID_{rs} \| d.M \| T_2)$,, where T_2 is the current timestamp. RSU sends (M, P_{RS}, T_2) to AV and sets $SK_{RV} = h(d.N \| R_{RS}.T_v \| PID_v \| PID_{RS})$ as the session key.
6. On receiving RSU response, AV calcultes $P_{RS} = h(PID_{RS} \| c.M \| T_2)$ and checks whether $P_{RS} = P_v$, If not, AV ends the process. Otherwise, AV sets $SK_{VR} = h(cM \| R_v.T_{RS} \| PID_v \| PID_{RS})$ as the session key.

The above process is depicted in table 3. The correctness of the scheme ($e.g. SK_{VR} = SK_{RV}$) is guaranteed by the equation $cM = dN = cdg$ and $R_v.T_{RS} = R_{RS}.T_v = R_V.R_{RS}g$

Table 3: Authentication between AV and RSU

AV	**Smart Contract**	**RSU**
Inputs ID'_v and PW'_v		
Imprint bio-metric B'_v		
Receive $\alpha_v \leftarrow$ Rec(B'$_v$,β_v)		
Compute $V'_1 = h(ID'_v \| PW'_v \| \alpha'_u) \bigoplus R_v$		
Verifies $V'_1 = V_1$, if yes;		
Calculates $PW^{*\prime}_v = h(PW'_v \| \alpha_v \| R_v{}^*)$		
Verifies $PW^{*\prime}_v = PW^*_v$		
Obtains · · · · · · · · · · · · · · · · →		
	$T_{RS} \leftarrow$ Query $PKT(RS, PID_{RS})$	
	← · · · · · · · · · · · · · · · ·	
Selects $c \in Z_q^*$		
Compute $N = c.g$;		
Computes $D_v = h(c.T_{RS} \| T_1) \oplus PID_v$		
Computes $P_v = c + h(PID_v \| N \| T_1)$		
Sends $\{N, D_v, P_v, \beta_v, T_1\}$ to RSU		
· · · · · · · · · · · · · · · · →		
		Receives $\{N, D_v, P_v, \beta_v, T_1\}$
		Verify timestamp $T_1 - T_2 \leq \triangle T$
		Calculates $PID_v = D_v \bigoplus h(R_{RS} N \| T_1)$
	Query $PKT(AV, PID_v) \rightarrow T_v$	Obtains T_v from smart contract
	· · · · · · · · · · · · · · · · →	Calculates $P_v.g = N + h(PID_v \| N \| T_2).T_v$
		Selects $d \in Z_q^*$
		Calculates $M = d.g$
		Calculate $dn = d.N$
		Calculates $P_{RS} = h(PID_{RS} \| dn \| T_2)$
		Computes $SK_{RV} = h(dn \| R_{RS}.T_v \| PID_v \| PID_{RS})$
		Sends $\{M, P_{RS}, T_2\}$ to AV
		← · · · · · · · · · · · · · · · ·
Receives $\{M, P_{RS}, T_2\}$		
Verify timestamp $T_2 - T_3 \leq \triangle T$		
Computes $L = c.M$		
Computes $P_{RS} = h(PID_{RS} \| L \| T_2)$		
Verifies $P_{RS} \stackrel{?}{=} P_v$, if yes		
Calculates $SK_{VR} = h(L \| R_v.T_{RS} \| PID_v \| PID_{RS})$		
Checks $SK = SK_{VR} = SK_{RV}$		

3.4. Password and Biometric Update Phase

AV can update her password or biometrics or both locally without interaction with CAS. The process is as follows.

- AV inputs the old password pw_v and imprints the old biometric B_v on the mobile device.
- AV computes $\alpha_v = Rec(B_v, \beta_v), R_v = V_1 \bigoplus h(ID_v||pw_v||\alpha_v)$.
- If $pw_v^* = h(pw_v||\alpha_v||R_v)$ holds. AV is allowed to choose a new password pdw'_u and imprint the new biometrics B'_u
- AV computes $(\alpha'_v, \beta'_v) = GEN(B'_v), (V_1)' = h(ID_v||pw'_u||\alpha'_u) \bigoplus V_1, pw_v^* = h(pw'_v||\alpha'_v||s_v)$
- AV stores (pw_v^*, V_1', β'_v) in their memory.

4. Security Analysis

The security analysis of a scheme reveals how much stronger a scheme is than others and which attacks are not usable. We present the analysis of the security of the suggested approach in both informal and formal ways, as follows:

4.1. Informal Security analysis

In this section, we look at the proposed frameworks for security and efficiency. The proposed framework withstands some well-known attacks and performs more efficiently.

4.1.1. Mutual Authentication

The security of digital signatures and MACs ensures mutual authentication between AV and RSU. AV creates electronic signatures using its permanent private keys R_v sent out by the CAS. The password and biometrics are used to safeguard R_v, which is kept on the mobile device. As long as the adversary is unable to give both a password and a biometric, the private key will remain unavailable even if they manage to get their hands on the AV device. The blockchain stores the appropriate public key (PID_v, T_v) that is used to validate the signature. The blockchain's immutability ensures the legitimacy of the public key, and the Schnorr signature algorithm employed in the suggested approach has gained widespread acceptance for its security. $¶_{RS}$, which is a MAC with dN as the key, ensures the legitimacy of RSU identification. As long as the underlying signature scheme is impossible to counterfeit and CDHP is challenging, the adversary can't access the password for AV or RSU.

4.1.2. Session Key Security

Since $cM = dN = cdg$ and $R_v.T_{RS} = R_{RS}.T_v = R_V.R_{RS}\,g$, it is evident that RSU and AV will share the same session key if the protocol is successfully implemented. To compute cdg without knowing c or d, however, is computationally impossible given the DHP's difficulty. Without abP, the attacker would have a difficult time generating the right session key, which is guaranteed by the hash function's ability to withstand collisions.

4.1.3. Single-Sign-On:

All authorized nodes may access the public-key data for AV and RSU stored on the blockchain. Neither the AV nor RSU must keep any information on the other's organization on file. Any RSU can be accessed by an AV when it successfully registers with CAS.

4.1.4. Forward secrecy:

The session key $SK_{RV} = SK_{VR} = h(cdg||R_v.T_{RS}||PID_v||PID_{RS})$ was used during the authentication between AV and RSU. Assume the private keys R_{RS} and R_v have both been compromised and the attacker can get all messages $(N, M, P_{RS}, P_v, DID_v, T_1, T_2)$ sent through the public channel. Given $N = cg, M = dg$, it is impossible to compute $cM = dN = cdg$ without first knowing c or d since DHLP is so difficult. The value of cdg cannot be ascertained even if the adversary acquires P_v, P_{RS}, R_v, and R_{RS}. Therefore, even if the RSU and AV long-term keys are compromised, the prior session keys are secure.

4.1.5. User and Device Anonymity:

The hash of the AV's real identity with a random number yields AV's pseudo-identification PID, which is stored on the blockchain. It is challenging to link the PID with the true identity of AV because of the one-way nature of the hash algorithm. The true identity of AV has never been revealed during the entire procedure. Even this PID is dynamically disguised (D_v) throughout the authentication procedure transmission. As a result, tracking the user based on her activity is challenging. As a result, user anonymity may be ensured.

4.1.6. Untraceability:

Due to the use of random numbers, it is impossible to identify any connections between messages in various sessions, whether in the mutual authentication between AV and RSU. Furthermore, AVs have not revealed their true identities to the public throughout the authentication procedure. Each session's communication includes a new version of their identifying information DID_u. It is challenging for attackers or unaffiliated users to link many sessions to a particular user or device.

4.1.7. No Online Trust Authority:

In the suggested architecture, CAS is only required to create public and private keys for the entities participating in the registration phase and initialize the system. It doesn't take part in the RSU-to-AV authentication procedure. As a result, CAS does not have to remain online during authentication.

4.1.8. Replay Attack:

Timestamps and random numbers are used in the protocols to prevent replay attacks.

4.1.9. Denail-of-Service Attack:

The decentralized nature of blockchain can make DoS attacks against specific entities ineffective

4.1.10. Man in Middle Attack(MITM):

The man-in-the-middle attack is likewise impossible since the ECC-based signature scheme's unforgeability and the biometric imprints make it impossible for any adversary to create lawful verification messages (P_v, P_{RS} *between AV and RSU*).

4.1.11. Eavesdropping Attack:

In accordance with the eavesdropping attack, $\mathcal{A}$ is able to intercept any messages sent across an insecure channel. $\mathcal{A}$ can therefore intercept messages. But under the suggested protocol, each round of authentication uses a different random number generator and a hash function to secure all the parameters. Thus, neither $\mathcal{A}$ gets any parameter nor the user's identity is discovered. Additionally, then $\mathcal{A}$ is unable to compute $SK_VR = h(c.M||R_v.T_{RS}||PID_v||PID_{RS})$. Therefore, $\mathcal{A}$ is unable to gain PID_v, PID_{RS}, T_{RS} and R_v.

4.1.12. Unlinkability:

Two significant privacy issues are location and user identity. The AV identity and other relevant facts must be kept secret from the opponent. The adversary cannot determine the AV identity in the proposed protocol since we utilize the anonymous identity ID'_v and also hide it by the hash value. To further secure privacy, each session makes use of a unique temporary identity, or ID'_v is unlinkable, thus outsiders cannot determine who is speaking with RSU. The essence associated with two executions of the protocol that are different or identical is unknown to the adversary. Therefore, the proposed approach safeguards users' privacy and stops the disclosure of their identities.

4.2. Formal Security Analysis

To assess the formal security of the suggested scheme, we employed the established technique referred to as the Random Oracle Model (ROM). The ROM security framework has found application in contemporary cryptography for the past two decades. Its inception can be attributed to Bellare and Rogaway in 1994. Within the ROM model, an underlying assumption is the presence of a publicly accessible oracle denoted as H, available to all users, including potential adversaries.

4.2.1. Random Oracle Model

As per the proposed protocol, the ad-hoc vehicles (AV) and the roadside units (CAS) are the focal entities. They execute the tasks outlined below in alignment with the provided specifications.

(a) **Participants:** We assume that $\theta_{AV}^{t_1}$ and $\theta_{CAS}^{t_2}$ indicate the incidents t_1 and t_2 of AV and CAS, respectively, these are referred to as oracles.

(b) **Freshness:** We contemplate that $\theta_{AV}^{t_1}$ and $\theta_{CAS}^{t_2}$ will remain novel, provided that the shared communication session key SK is not revealed to $\mathcal{A}$.

(c) **Partnering:** The counterpart of an occurrence $\theta_{AV}^{t_1}$ related to AV is the incident $\theta_{CAS}^{t_2}$ associated with CAS, and vice versa. In this context, we label $\theta_{CAS}^{t_2}$ as the counterpart of AV. A singular partial transcript encompasses all exchanges between AV and $\theta_{CAS}^{t_2}$, identified as the communication session ID $sid_{AV}^{t_2}$ for the ongoing session in which AV is engaged.

(d) **Adversary:** Under the ROM model, the adversary $\mathcal{A}$ exercises absolute authority over all communications. The adversary possesses the capability to modify and access all exchanged messages, as well as the ability to generate new messages and inject them into the system. The subsequent inquiries are among the actions that $\mathcal{A}$ will be capable of executing:

CorruptSC($\theta_{AV}^{t_1}$): In the event of a user misplacing their smart card or identity credentials, the adversary emulates a smart card loss attack on ID_{AV} and unveils the data stored within it.

Execute($\theta^{t_1}, \theta^{t_2}$): $\mathcal{A}$ initiates this query to replicate an eavesdropping attack, thereby inducing the exchange of communications between two reliable network users.

Send(θ^{t_1}, M): This query depicts an active attack scenario, wherein the adversary has the capability to dispatch a message M to a user involved in an occurrence of θ^{t_1}. Subsequently, the adversary seeks to obtain a response message.

Test(θ^t): This process faithfully reproduces the ROM model and ensures semantic security for the session key SK exchanged between AV and CAS, rendering them indistinguishable. A fair coin, denoted as C, is flipped to initiate the experiment. To determine the outcome of the (Test) query, a coin is flipped in secrecy from ($\mathcal{A}$). The event θ^t yields SK when $c = 1$, or a random integer if $c = 0$. Subsequently, during a Test query by $\mathcal{A}$, the validity and freshness of SK are established; otherwise, the output is null.

The objective of $\mathcal{A}$ is to compromise the suggested scheme denoted as *BSAPVAN*. The probability of this event is denoted as $Prob[event_{succ}]$, where $event_{succ}$ signifies the event of $\mathcal{A}$ successfully breaking the game. The measure of $\mathcal{A}$'s success in compromising the security of the proposed protocol is computed as $Adv_{BSAPVAN}^{ake} = 2Prob[event_{succ_0}] - 1$. Within the ROM model, the BSAPVAN protocol is considered secure for authentication if $Adv_{BSAPVAN}^{ake} \leq \delta$ holds true, for all sufficiently small $\delta > 0$. In accordance with the ROM model's design, both participants and the adversary are provided with a one-way hash function denoted as $h(.)$.

Semantic Security of Session Key (SK): The adversary, denoted as $\mathcal{A}$, conducts a sequence of tests with the objective of distinguishing the legitimate session key (SK) from instances of a random key. A binary value c must align with the outcome of the Test query, and $\mathcal{A}$ is granted the ability to make multiple test queries using the random oracle model denoted as H. At the culmination of the experiment series, $\mathcal{A}$ provides an estimation for the binary value c'. Success is achieved if $c' = c$.

Theorem 1: Assuming $\mathcal{A}$ operates as an adversarial entity within a ROM framework, constrained by a polynomial time limit of t, and directing its efforts towards the proposed *BSAPVAN* scheme. With D representing a uniformly distributed password dictionary and no compromise of node integrity by the adversary, the probability of the attacker successfully compromising the session key's security can be calculated as follows:

$$ADV_{BSAPVAN}^{ake}(\mathcal{A}) \leq \frac{{q_h}^2}{|HASH|} + \frac{q_{send}}{2^{l-1}.|d|} + 2.ADV_{G_p}^{ECDLP}(t)$$

Here, $ADV_{G_p}^{ECDLP}(t)$, q_h, $|HASH|$, d and q_{send} are the advantage of the adversary of solving the Elliptic Curve Discrete Logarithm Problem (ECDLP) over GF_p with respect to the elliptic curve equation, the number of hash queries sent, range of the hash function's, the size of d and the number of hash queries sent to the ROM model, respectively.

Proof: G_i represents a collection of games indexed by integers from 0 to 4, with each game assessing the ad-

versary's success in the guessing process denoted by the $succ_i$ bit. The objective of the proof is to illustrate that the adversary $\mathcal{A}$ possesses minimal advantage over the *BSAPVAN* protocol when it comes to undermining the security of the session key.

- **Game** G_0**:** The adversary $\mathcal{A}$ directs its actual attack towards the proposed *BSAPVAN* framework within the ROM model in the game G_0. As the bit b is initially chosen randomly in this game, we now have:

$$ADV^{ake}_{BSAPVAN} = 2Prob[event_{succ_0}] - 1 \tag{1}$$

- **Game** G_1 : The adversary $\mathcal{A}$ achieved this by launching an eavesdropping attack in the form of an Execute(θ^t, θ^l) query to the oracle. The outcome of the Execute(θ^t, θ^l) query is then compared by the adversary using a Test oracle, which assesses whether the query output corresponds to the actual session key or a random integer.

  The suggested scheme $BSAPVAN$ calculates the session key $SK_{RV} = h(d.N||R_{RS}.T_v||PID_v||PID_{RS})$. The game G_1 will not provide success result to the adversary $\mathcal{A}$, because of two following reasons given below,

  Argument-1: If SK_{RV} is known to the attacker, a counterfeit of a session key SK_{RV} can be created. According to the ROM model, all participants, including adversary $\mathcal{A}$, possess knowledge of the values of d, N equation. Despite this, the adversary $\mathcal{A}$ is not able to theft the initial session key SK_{RV} without being aware of the values for c and d. The two random variables in this case, c and d, are freshly formed for each login session, whereas R_{RS} is a distinct random value established at the time of the registration process. The reasoning states that the adversary cannot produce a fake session key SK_{RV}.

  Argument-2: The adversary $\mathcal{A}$ tries to identify the V_1' from the login request. Due to the ad-hoc vehicle maintenance of the secret key SK_{RV}, this is not possible. For the opponent, ($\mathcal{A}$) in-game G_1, the likelihood of success won't rise. G_1 is then equal to G_0, and the corresponding probability is also equal.

$$Prob(event_{succ_0}) = Prob(event_{succ_1}) \tag{2}$$

- **Game** G_2 : With the addition of the Hash and send oracle simulations, the game G_2 differs from G_1. This game represents an active attack in which the adversary $\mathcal{A}$ attempts to mislead a player into believing a message created by it. A constantly asks the Hash oracle to look for collisions. Note that if the adversary gets V_1 and D_v. V_1 is associated with the password of the ad-hoc vehicle and D_V is associated with the random variable. Therefore, if $\mathcal{A}$ asks the Send oracle, there won't be any collisions. Using SHS's (2005) birthday paradox, we obtain

$$|Prob(event_{succ_1}) - Prob(event_{succ_2})| \leq \frac{q_h^2}{2.|HASH|} \tag{3}$$

- **Game** G_3 : The game $G3$ is a simulation of the corrupt system node(Adhoc vehicle node) oracle and models the smart card loss attack. If the password has poor entropy, the adversary may conduct an online dictionary attack using the information gathered from the identity. Although the adversary is able to extract the V_1 value, because of the collision-resistant hash function, the original password PW could not be recovered, even with the known values of R_v and α_u. If the system should limit the number of incorrect logins or password entries, the likelihood might be calculated as

$$|Prob(event_{succ_2}) - Prob(event_{succ_3})| \leq \frac{q_{send}}{2^l.|D|} \tag{4}$$

- **Game** G_4 : Game G_4 simulates an attack in which the attacker has taken over the server via a corrupt system node(fog node) oracle, and has obtained the identity ID_v of the legitimate user AV. The suggested protocol

BSAPVAN generates the SK by using d, N. Let $Adv_{G_p}^{ECDLP}(t)$ be the adversary's advantage in the experiment. The session key $SK_{RV} = h(d.N||R_{RS}.T_v||PID_v||PID_{RS})$ generated using d, N, R_{RS}, PID_v and PID_{RS}. The adversary requires d, N for EC point calculation.

$$|Prob(event_{succ_3}) - Prob(event_{succ_4})| \leq Adv_{G_p}^{ECDLP}(t) \tag{5}$$

According to the last Game G_4 , It is obvious that

$$Prob[event_{succ_4}] = 1/2 \tag{6}$$

From the equation 1 to 5 we have

$$Adv_{BSAPVAN}^{ake} \leq \frac{{q_h}^2}{|HASH|} + \frac{q_{send}}{2^{l-1}.|d|} + 2.Adv_{G_p}^{ECDLP}(t) \tag{7}$$

In Game G4, the corruptSC oracle replicates the attacks that occur when someone steals an identity (assume that adversary A) under the supposition that the ECDLP problem is computationally infeasible for a probabilistic polynomial time, $Adv_{BSAPVAN}^{ake}(\mathcal{A})$ is extremely probable in this game, and this proposed protocol BSAPVAN is secure. Even when sensitive data from the identity is leaked, the session key SK_{RV} is secure based on ECDLP problem. Consequently, the suggested technique preserves perfect forward secrecy authentication and is secure.

5. Performance Analysis

Here we describe the functionality feature and security comparison, computation cost, communication cost, and storage overhead of the suggested framework and related schemes [20, 21, 22, 23, 24] of similar context.

5.1. Functionality Feature and Security Comparison

We compare the proposed protocol's functionality performance and security with other protocols Yang et al [20], Feng et al. [21], Zhang et al. [22], Feng et al. [23] and Yang et al. [24]. The comparative security analysis of the proposed scheme shows that the proposed scheme defends against all possible attacks within the bounds (shown in Table 4).

Table 4: Security Features Comparison

Security features	**Yang et al [20]**	**Feng et al. [21]**	**Zhang et al. [22]**	**Feng et al. [23]**	**Yang et al. [24]**	**Proposed**
Mutual authentication	✓	✓	✓	✓	✓	✓
Anonymity	✓	×	✓	✓	✓	✓
Untracebility	✓	✓	✓	×	✓	✓
Session key security	×	×	×	×	×	✓
Perfect forward secrecy	×	✓	×	×	×	✓
Denail of service attack	✓	✓	✓	×	×	✓
Unlinkability	✓	×	✓	×	✓	✓
Eavesdropping attack	×	×	×	×	×	✓
Man-in-the-middle attack	✓	×	×	×	✓	✓
Replay attack	✓	✓	✓	✓	✓	✓
Single-sign-in	×	✓	✓	×	×	✓
No online trust authority	×	✓	✓	×	×	✓
ROR Model	×	×	✓	×	×	✓

Table 5: Cryptographic operations execution time

Notations	Explanation	Value in Millisecond Seconds (ms)
T_H	Hash Function	0.002
T_{SM}	Elliptic curve Scalar point multiplication	0.601
T_A	Elliptic curve point addition	0.051
T_{EP}	Exponentiation in the group G.	3.85
T_{ver}	Time for retrieval.	0.32
T_{ret}	Time for verification	0.62
T_P	Time of bilinear pairing operation	39.872

Table 6: Computation Cost Comparison

Protocol	Vehicle	RSU	Total Operations	Total Cost (ms)
Yang et al [20]	$5T_H + 6T_M + 4T_A$	$4T_H + 4T_M + T_A$	$7T_M + 6T_H + 5T_A$	6.283
Feng et al. [23]	-	-	$T_H + T_M + T_{ver} + T_{ret}$	1.543
Yang et al.[24]	$2T_M + 2T_M$	$1T_A + 3T_M$	$1T_A + 5T_M + 2T_M$	3.07672
Proposed	$4T_M + 6T_H$	$6T_M + 3T_H$	$10T_M + 9T_H$	6.0818

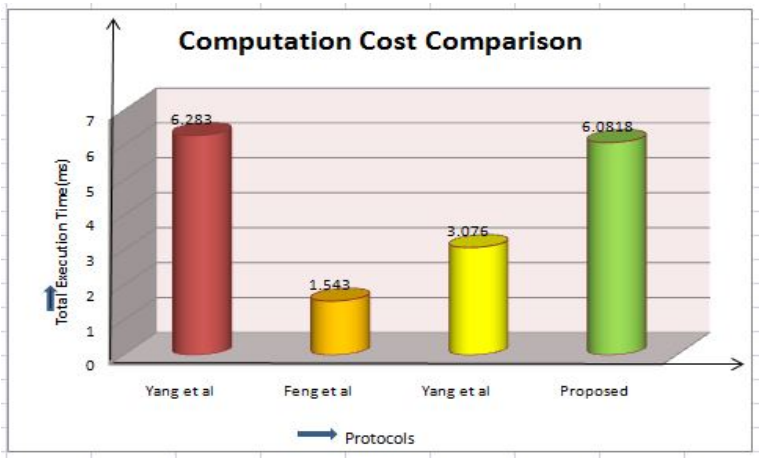

Figure 3: Computation Cost

5.2. Computation cost

In this section, we assess the computational overhead of our scheme by measuring the execution time of various basic cryptographic operations using the JPBC library. Subsequently, we analyze the authentication overhead for several associated schemes, as outlined in Table 6. Finally, we conduct a comparison of our scheme's batch authentication overhead with that of other related schemes. The details of these evaluations are provided in the below table 5:

5.3. Communication Cost

In this study, we evaluate the communicational expenses of the proposed scheme in contrast to existing methods [20, 21, 22, 23, 24]. This assessment is based on the volume of messages exchanged between parties and the bit count associated with each transmitted or received message. Our analysis assumes various parameters: a hash function length of 256 bits, a 32-bit random number, a 160-bit identity length, a 32-bit timestamp length, a 160-bit elliptic point, a 160-bit MAC message, a 160-bit multiplicative group, 1024 bits for signatures, and 1288 bits for single message authentication tokens.

To assess the additional load imposed by the proposed scheme and comparable protocols within a comparable framework, we conducted an extensive analysis and summarized the outcomes in Table 7. Our investigation reveals that, regarding communication expenses, the proposed scheme outperforms other schemes, as depicted in Figure 4.

Our objective is to diminish communication overhead while maintaining both security and performance standards. By curtailing communication costs, our proposed scheme presents notable advantages over prevailing protocols. The efficacy, coupled with the robustness and security of our approach, positions it as a valuable contribution to the contemporary literature on authentication schemes.

Table 7: Communication cost comparison

Protocol	Number of Messages	Total Cost(bits)
Yang et al [20]	3	4928
Feng et al. [23]	2	2576
Yang et al. [24]	1	1728
Proposed	2	1216

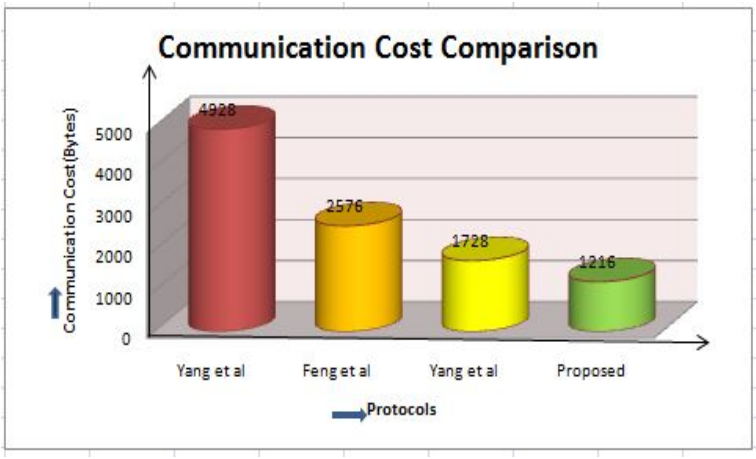

Figure 4: Communication cost

5.4. Storage overhead

In this study, we evaluate the storage expenses of the proposed scheme in contrast to existing methods [20, 21, 22, 23, 24]. The storage overhead involves the amount of memory required to store the key and its related parameters. Our analysis assumes various parameters: a hash function length of 256 bits, a 32 bits random number, a 160-bit identity length, a 32-bit timestamp length, a 160-bit elliptic point, a 160-bit MAC message, a 160-bit multiplicative group, 1024 bits for signatures, and 1288 bits for single message authentication tokens. Our proposed scheme's minimal storage overhead brings numerous benefits, such as diminished memory demands, heightened system efficiency, and cost-effectiveness. Leveraging memory resources efficiently renders our scheme an optimal choice for deployment in environments with resource constraints.

Table 8: Storage Overhead

Protocol	Total Cost (bits)
Yang et al [20]	2304
Feng et al. [23]	1024
Yang et al. [24]	1024
Proposed	736

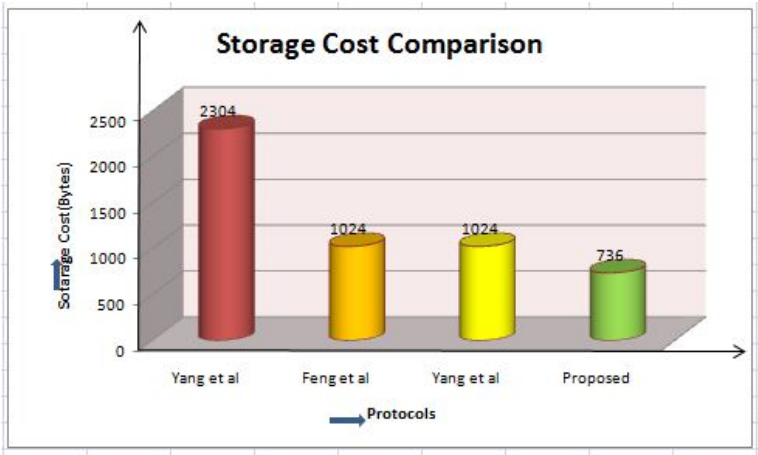

Figure 5: storage overhead

6. Conclusion

In conclusion, this paper presents the Blockchain-based Conditional Privacy-Preserving Authentication (BCPPA) protocol tailored for Vehicular Ad Hoc Networks (VANETs), addressing the critical challenges of security and privacy in vehicular communications. By harnessing Blockchain technology and smart contracts, the protocol offers a robust framework for secure data storage and efficient authentication processes. Moreover, the incorporation of a key derivation algorithm mitigates the need for extensive key pre-storing in vehicle On-Board Units (OBUs), enhancing the scalability and usability of the protocol. Leveraging a modified Elliptic Curve Digital Signature Algorithm (ECDSA) with batch verification, BCPPA optimizes verification efficiency in VANETs while ensuring cryptographic integrity. Through comprehensive security and performance analyses, we have demonstrated the efficacy of BCPPA in safeguarding VANET communications against various threats while preserving user privacy. As VANETs continue to evolve, BCPPA stands as a promising solution to foster safer and more secure vehicular communication environments.

Declarations

- **Ethics approval and consent to participate:**

  The manuscript is not currently being considered for publication elsewhere.

- **Consent for publication:**

  All authors are agreed for publication.

- **Availability of data and materials:**

  Not applicable.

- **Competing interests:**

  The authors declare that they have no competing interests.

- **Funding:**

  The authors did not receive support from any organization for the submitted work.

- **Authors' contributions:**

  Samiulla Itoo devised the project, the main conceptual ideas and proof outline. Samiulla Itoo worked out almost all of the technical details, and performed the calculations for the suggested experiment. Samiulla Itoo and Ram Baksh worked out the algorithm for Authentication, with help from Musheer Ahmad. Samiulla Itoo and Ram Baksh performed the security Analysis results of the proposed algorithm. Samiulla Itoo wrote the manuscript and all authors reviewed it.

- **Acknowledgements:**

  Not applicable.

References

[1] D. He, S. Zeadally, B. Xu, X. Huang, An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks, IEEE Transactions on Information Forensics and Security 10 (12) (2015) 2681–2691.

[2] A. Boukerche, H. A. Oliveira, E. F. Nakamura, A. A. Loureiro, Vehicular ad hoc networks: A new challenge for localization-based systems, Computer communications 31 (12) (2008) 2838–2849.

[3] N. M. Rabadi, Implicit certificates support in ieee 1609 security services for wireless access in vehicular environment (wave), in: The 7th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2010), IEEE, 2010, pp. 531–537.

[4] R. Lu, X. Lin, H. Zhu, P.-H. Ho, X. Shen, Ecpp: Efficient conditional privacy preservation protocol for secure vehicular communications, in: IEEE INFOCOM 2008-The 27th Conference on Computer Communications, IEEE, 2008, pp. 1229–1237.

[5] C. Zhang, X. Lin, R. Lu, P.-H. Ho, Raise: An efficient rsu-aided message authentication scheme in vehicular communication networks, in: 2008 IEEE international conference on communications, IEEE, 2008, pp. 1451–1457.

[6] N.-W. Lo, J.-L. Tsai, An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks without pairings, IEEE Transactions on Intelligent Transportation Systems 17 (5) (2015) 1319–1328.

[7] M. Bayat, M. Barmshoory, M. Rahimi, M. R. Aref, A secure authentication scheme for vanets with batch verification, Wireless networks 21 (5) (2015) 1733–1743.

[8] T. W. Chim, S.-M. Yiu, L. C. Hui, V. O. Li, Specs: Secure and privacy enhancing communications schemes for vanets, Ad Hoc Networks 9 (2) (2011) 189–203.

[9] L. Zhang, Q. Wu, J. Domingo-Ferrer, B. Qin, C. Hu, Distributed aggregate privacy-preserving authentication in vanets, IEEE Transactions on Intelligent Transportation Systems 18 (3) (2016) 516–526.

[10] D. Zheng, C. Jing, R. Guo, S. Gao, L. Wang, A traceable blockchain-based access authentication system with privacy preservation in vanets, IEEE Access 7 (2019) 117716–117726.

[11] M. Raya, J.-P. Hubaux, Securing vehicular ad hoc networks, Journal of computer security 15 (1) (2007) 39–68.

[12] C. Zhang, R. Lu, X. Lin, P.-H. Ho, X. Shen, An efficient identity-based batch verification scheme for vehicular sensor networks, in: IEEE INFOCOM 2008-The 27th Conference on Computer Communications, IEEE, 2008, pp. 246–250.

[13] J. K. Liu, T. H. Yuen, M. H. Au, W. Susilo, Improvements on an authentication scheme for vehicular sensor networks, Expert Systems with Applications 41 (5) (2014) 2559–2564.

[14] J. Zhang, J. Cui, H. Zhong, Z. Chen, L. Liu, Pa-crt: Chinese remainder theorem based conditional privacy-preserving authentication scheme in vehicular ad-hoc networks, IEEE Transactions on Dependable and Secure Computing 18 (2) (2019) 722–735.

[15] Z. Lu, Q. Wang, G. Qu, H. Zhang, Z. Liu, A blockchain-based privacy-preserving authentication scheme for vanets, IEEE Transactions on Very Large Scale Integration (VLSI) Systems 27 (12) (2019) 2792–2801.

[16] C. Lin, D. He, X. Huang, M. K. Khan, K.-K. R. Choo, A new transitively closed undirected graph authentication scheme for blockchain-based identity management systems, IEEE Access 6 (2018) 28203–28212.
[17] D. R. Brown, Generic groups, collision resistance, and ecdsa, Designs, Codes and Cryptography 35 (1) (2005) 119–152.
[18] D. Dolev, A. Yao, On the security of public key protocols, IEEE Transactions on information theory 29 (2) (1983) 198–208.
[19] R. Canetti, H. Krawczyk, Analysis of key-exchange protocols and their use for building secure channels, in: International conference on the theory and applications of cryptographic techniques, Springer, 2001, pp. 453–474.
[20] Y. Yang, L. Wei, J. Wu, C. Long, B. Li, A blockchain-based multidomain authentication scheme for conditional privacy preserving in vehicular ad-hoc network, IEEE Internet of Things Journal 9 (11) (2021) 8078–8090.
[21] Q. Feng, D. He, S. Zeadally, K. Liang, Bpas: Blockchain-assisted privacy-preserving authentication system for vehicular ad hoc networks, IEEE Transactions on Industrial Informatics 16 (6) (2019) 4146–4155.
[22] J. Zhang, Y. Jiang, J. Cui, D. He, I. Bolodurina, H. Zhong, Dbcpa: Dual blockchain-assisted conditional privacy-preserving authentication framework and protocol for vehicular ad hoc networks, IEEE Transactions on Mobile Computing.
[23] X. Feng, K. Cui, H. Jiang, Z. Li, Ebas: An efficient blockchain-based authentication scheme for secure communication in vehicular ad hoc network, Symmetry 14 (6) (2022) 1230.
[24] Y. Yang, D. He, H. Wang, L. Zhou, An efficient blockchain-based batch verification scheme for vehicular ad hoc networks, Transactions on Emerging Telecommunications Technologies 33 (5) (2022) e3857.