Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.065717

ARTICLE

Unleashing the Potential of Metaverse in Social IoV: An Authentication Protocol Based on Blockchain

Tsu-Yang Wu[1,2,3], Haozhi Wu[3], Maoxin Tang[1,2], Saru Kumari[4] and Chien-Ming Chen[1,2,*]

[1]School of Artificial Intelligence, Nanjing University of Information Science & Technology, Nanjing, 210044, China
[2]Jiangsu Provincial Key Laboratory of Culture and Tourism for Research on the Application Technology of Metaverse Cultural Tourism Scenarios (Nanjing University of Information Science and Technology), Nanjing, 210044, China
[3]College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, 266590, China
[4]Department of Mathematics, Chaudhary Charan Singh University, Meerut, 250004, India

*Corresponding Author: Chien-Ming Chen. Email: chienmingchen@ieee.org

Received: 20 March 2025; Accepted: 09 May 2025; Published: 03 July 2025

ABSTRACT: As a model for the next generation of the Internet, the metaverse—a fully immersive, hyper-temporal virtual shared space—is transitioning from imagination to reality. At present, the metaverse has been widely applied in a variety of fields, including education, social entertainment, Internet of vehicles (IoV), healthcare, and virtual tours. In IoVs, researchers primarily focus on using the metaverse to improve the traffic safety of vehicles, while paying limited attention to passengers' social needs. At the same time, Social Internet of Vehicles (SIoV) introduces the concept of social networks in IoV to provide better resources and services for users. However, the problem of single interaction between SIoV and users has become increasingly prominent. In this paper, we first introduce a SIoV environment combined with the metaverse. In this environment, we adopt blockchain as the platform of the metaverse to provide a decentralized environment. Concerning passengers' social data may contain sensitive/private information, we then design an authentication and key agreement protocol called MSIoV-AKA to protect the communications. Through formal security verifications in the real-or-random (ROR) model and using the AVISPA (Automated Validation of Internet Security Protocols and Applications) tool, we firmly verify the security of the protocol. Finally, detailed comparisons are made between our protocol and robust protocols/schemes in terms of computational cost and communication cost. In addition, we implement the MSIoV-AKA protocol in the Ethereum test network and Hyperledger Sawtooth to show the practicality.

KEYWORDS: Authentication; key agreement; SIoV; metaverse; blockchain

1 Introduction

With the progressive development of the Internet of Things (IoT) [1] and Artificial Intelligence (AI) [2,3], Metaverse technology [4], hailed as the next generation of the internet, is rapidly on the rise. The metaverse is a digital world with immersive experiences that integrates virtual and real worlds to a high degree. Constructed upon technologies like Extended Reality (XR), 5G, Artificial Intelligence (AI), and data processing [5], the metaverse is capable of offering users 3D immersive and personalized experiences. In addition, blockchain is one of the key technologies of the metaverse [6–8]. Blockchain technology not only furnishes substantial computational resources for the metaverse but also enables users to transition seamlessly among different virtual worlds. Given the decentralization of blockchain, it effectively circumvents the single-point-of-failure issue. Consequently, blockchain technology assumes a crucial role within the

metaverse. Users enter the metaverse via immersive devices and create virtual avatars. Through these avatars, users can communicate in real-time and interact with residents of other virtual worlds, experiencing a sense of presence in virtual reality. Additionally, metaverse offers a variety of virtual experiences and activities, such as virtual socializing, virtual business, and virtual tourism, allowing users to enjoy diverse entertainment and social interactions in the virtual world.

The Internet of Vehicles (IoV) [9–11] connects vehicles, road infrastructure, and IoT to facilitate information exchange and data sharing between vehicles and infrastructures. In IoV, vehicles upload driving information to obtain corresponding services, such as collision warnings, traffic congestion alerts, and personalized navigation. However, despite the improvements IoV brings to traffic safety and efficiency, it lacks social interaction among users.

To address this issue, many researchers have proposed Social Internet of Vehicles (SIoV) [12,13] to enhance social interaction among users. The Social Internet of Vehicles (SIoV) incorporates the concepts of IoV and Social Networks, thereby enabling the perception of associations among various entities, including people, vehicles, and roads. Within the SIoV frameworks, users can establish social relationships with others during their travels, forming a social network in the context of IoV. Consequently, SIoV can provide various social services for passengers, such as opportunities for working, studying, playing games, or watching videos together with fellow passengers in the vehicle. However, with the explosive growth of the scale of the SIoV networks, the problems of privacy leakage, data sharing, insufficient computing and storage capabilities have become increasingly prominent [14–16]. For this reason, many researchers have adopted blockchain technology to address these problems. The decentralized technology of blockchain can avoid a single point of failure and reduce the risk of data leakage. Additionally, the consensus mechanism of blockchain can motivate nodes to contribute computing and storage resources. Therefore, blockchain has become a crucial technology in the development of SIoV [17–19].

Although SIoV improves the social experience of users to a certain extent, it is still insufficient in the face of a large number of users' social demands. In recent years, many researchers [20,21] have proposed to use the metaverse to improve the functionalities of IoV. However, few researchers have focused on how to ensure the security of the social and entertainment needs of passengers. To address these issues, we propose a SIoV architecture in a metaverse environment. This architecture utilizes blockchain as the underlying platform for the metaverse. Blockchain can provide computational power for the metaverse and store transaction data generated within the metaverse. On the other hand, blockchain can establish a comprehensive economic system that connects the virtual world with the real world. In Fig. 1, we present the diagram of SIoV within the metaverse environment. When vehicles in the SIoV connect to the metaverse, passengers in the vehicles can enter the metaverse by wearing immersive devices. The social data generated by passengers is transmitted to Road Side Units (RSUs) via On-Board Units (OBUs), which then forward the data to metaverse companies. These companies provide services to passengers through servers deployed on blockchain or cloud infrastructure. Compared to traditional SIoV, the SIoV in the metaverse offers passengers more diverse and realistic social interactions.

However, when the metaverse satisfies the social needs of users, it also faces many network security threats. Since the vehicle and RSU are situated in public, communication between vehicles, RSUs, and metaverse service companies occurs over public channels. Consequently, attackers might fabricate or alter communication information and endeavor to initiate a variety of security attacks directed at passengers. Through these security vulnerabilities, attackers can obtain the private information of targeted passengers and potentially use it to gain access to the metaverse and deceiving other users. To tackle the aforementioned issues, leveraging the SIoV architecture within the metaverse environment, we further put forward an

authentication and key agreement protocol, namely MSIoV-AKA. The contributions and summary of this paper are as follows:

1. We propose a novel SIoV architecture in a metaverse environment, enriching the social entertainment experience for passengers beyond the traditional SIoV framework. Compared to traditional SIoV, the SIoV in the metaverse offers passengers more diverse and realistic social interactions.
2. In consideration of passenger privacy and security within the metaverse SIoV architecture, we design the MSIoV-AKA protocol using Shamir's Secret Sharing [22].
3. To verify the security of the MSIoV-AKA protocol, we make the formal analysis in the Real-or-Random (ROR) model and the AVISPA tool.
4. To comprehensively evaluate the performance of the MSIoV-AKA protocol, we first conducted a comparative analysis with several existing protocols in terms of computation and communication costs. The results demonstrate that our protocol significantly reduces computation overhead, while the communication cost remains at a comparable level.
5. We also tested the protocol on Hyperledger Sawtooth and Ethereum. In Sawtooth, with 30 blocks and varying node numbers, we measured computation time and latency. On Ethereum, we recorded the Gas cost of each protocol phase to verify practical feasibility.

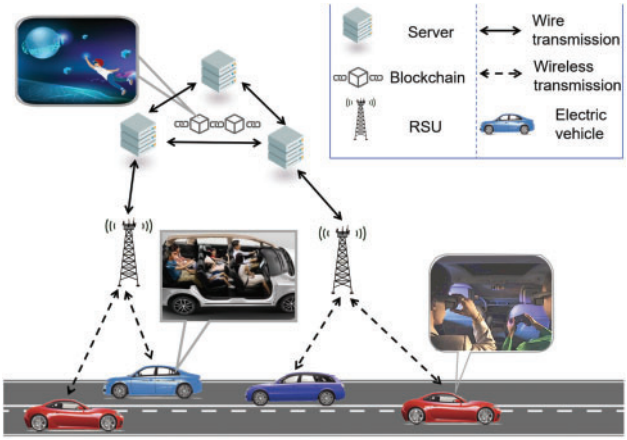


Figure 1: An integrated environment combining SIoV with metaverse

The subsequent sections of this paper are structured in a logical sequence as follows: Section 2 is related work. Section 3 delves into the attacker model and the goals of our protocol. Section 4 presents our proposed protocol. Section 5 provides a comprehensive security analysis of the protocol, evaluating its robustness against potential threats. Section 6 shows the comparisons of the performance between different protocols. Finally, Section 7 offers a concise conclusion.

2 Related Work

In recent years, researchers have proposed many secure schemes to protect the privacy and security in IoV and SIoV. In the year 2017, Mohit et al. [23] presented a vehicle authentication protocol that was based on wireless sensor networks. They confidently claimed that this protocol had the ability to resist impersonation attacks and stolen smart card attacks, thereby offering a certain level of security. However, the research landscape evolved, and in 2018, Yu et al. [24] made a significant discovery. They found that the previously proposed protocol was, in fact, not resistant to impersonation attacks, thus revealing a potential vulnerability. In light of this finding, Yu et al. took proactive steps to address this weakness and proposed an enhanced AKA

protocol. They emphasized that this new protocol achieved mutual authentication and anonymity. In 2020, Sadri and Rajabzadeh Asaar [25] further investigated and found that the protocol proposed by Yu et al. was not resistant to sensor capture attacks and impersonation attacks. Consequently, Sadri et al. proposed a new IoV authentication protocol, with the claim that it could provide more comprehensive security features, thus contributing to the advancement of secure vehicle authentication systems. In 2021, Jiang et al. [26] proposed an anonymous authentication mechanism and a blockchain-based data sharing scheme. Jiang et al. claimed that the scheme can protect user anonymity and unlinkability. In 2024, Esfahani et al. [27] introduced an AKA protocol aimed at connecting IoT devices in SIoV, a crucial step towards improving the reliability and security of communication among diverse IoT-enabled entities in the social vehicular context. This protocol overcomes the high computational cost issue of existing solutions through group authentication.

Since the term "metaverse" made its debut in the novel *Snow Crash*, the emergence of virtual world platforms has been on a steady rise. At the same time, the security of the virtual world environment has been discussed in some research. In 2016, O'Brolchain et al. [28] pointed out that users communicate through public channels and servers, which leads to significant threats to user privacy in virtual spaces. In 2018, Falchuk et al. [29] classified privacy into several types, including personal information privacy, behavioral privacy, and communication privacy. In response to the possible privacy violations in the social metaverse, Falchuk et al. proposed by creating a confusing effect. Through the confusing effect, the attacker's knowledge of the user's avatar activities, location, properties, interests and other information can be reduced. In 2020, De Guzman et al. [30] offered a comprehensive elucidation of the security and privacy requirements when users interact with virtual objects, laying the groundwork for subsequent research in this area. In 2022, Ryu et al. [31] proposed a mutual authentication scheme based on elliptic curve cryptography (ECC). This scheme not only provides secure communication between users and servers but also demonstrates resilience against offline dictionary guessing attacks, impersonation attacks, and man-in-the-middle attacks, thereby enhancing the security of user-server interactions. In the same year, Zhang et al. [32] further proposed a low-latency AKA protocol specifically tailored for metaverse-based EIoT power trading systems, addressing the unique requirements of this emerging application domain. In 2023, Yang et al. [33] introduced a biometric-based dual-factor authentication protocol incorporating chameleon signatures. This protocol represents a notable advancement as it ensures the verifiability of both virtual and physical identities of virtual characters, effectively safeguarding the integrity of the metaverse environment. Yang et al. claimed that it can successfully resist impersonation attacks and replay attacks. Also in 2023, Thakur et al. [34] proposed a certificateless encryption framework for metaverse identity verification. By leveraging ECC and fuzzy extractors to achieve mutual authentication between users. Thakur et al. claimed that it can resist replay attacks and impersonation attacks, further enhancing the security posture of the metaverse. In 2024, Gupta et al. [35] utilized convolutional neural networks to propose a lightweight encryption protocol to protect the metaverse. The protocol can provide secure mutual authentication between users and metaverse infrastructure.

3 System Model and Attacker Model

3.1 System Model

Our model involves three entities: vehicle V_i, roadside units RSU_j, and $Server$. The system model is illustrated in Fig. 2, and the detailed description follows.

1. Vehicle (V_i): We consider private vehicles rather than public transportation. Vehicles are equipped with immersive devices to access the metaverse, thereby supporting social interactions among passengers. Vehicles can communicate through OBU and nearby RSUs.
2. Passenger (U_c): We focus on social interactions among passengers within the same vehicle. Passengers can enter the metaverse using immersive devices in the vehicle. When there are more than one

passengers in the vehicle, they can engage in various social activities in the metaverse, such as playing games and watching movies. Because passengers in private vehicles are mostly family or friends, we consider that passengers in the same vehicle trust each other [12].

3. Road side Unit (RSU_j): As the intermediate node between the vehicle and the server, RSU is generally deployed on both sides of the road. Due to limited computational and storage capabilities, the RSU primarily handles message transmission. Being located in an open environment, the RSU is susceptible to attacks. In addition, RSU is considered a semi-trusted entity.
4. Server ($Server$): The server with a blockchain network maintains the blockchain. It is also responsible for the registration of vehicles, passengers, and RSUs. By invoking smart contracts, the server can also package passengers' transactions into blocks and upload them to the blockchain. Here, we consider the server as a trusted entity.
5. Blockchain (BC): The metaverse service company deploys the metaverse on the $Server$ integrated with blockchain. By invoking smart contracts deployed on the blockchain, passengers' identity information can be registered or updated. Therefore, the blockchain should be robust, secure, and support smart contract functionality.

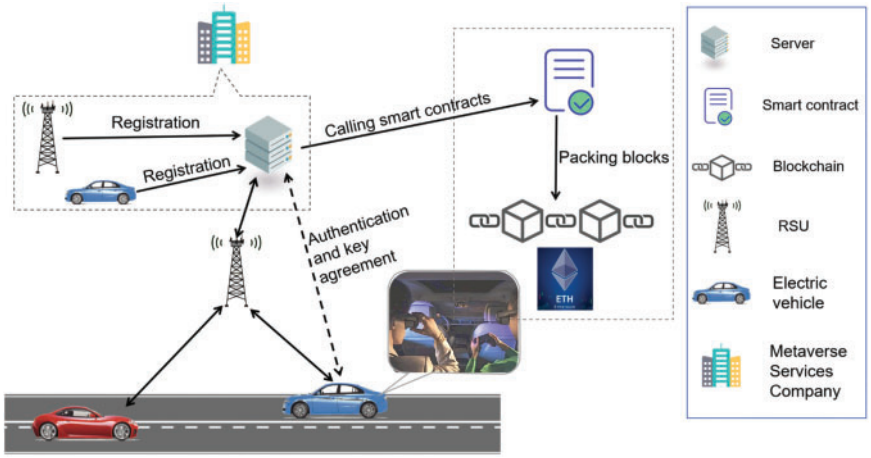


Figure 2: System model of SIoV with metaverse

According to the roles and functionalities of the entities mentioned above, the process of our system model is as follows. When passengers in a vehicle wish to use metaverse services, both the vehicle and the passengers should register with the server. After registration, passengers can log into the metaverse through immersive devices in the vehicle. Subsequently, the server integrated with blockchain technology in the metaverse service company will authenticate the passengers' identities and establish a session key. Through the session key, passengers can securely communicate with the server and invite passengers in the same vehicle to socialize in the metaverse, such as playing games.

3.2 Attacker Model

In this paper, we base on the Dolev-Yao (DY) model [36] and the Canetti-Krawczyk (CK) model [37] to define the capabilities of the attacker ($\mathcal{A}$). The specific capabilities are as follows:

(1) $\mathcal{A}$ is capable of eavesdropping, intercepting, tampering and replaying information transmitted over public channels.

(2) $\mathcal{A}$ has the capacity to physically seize the OBU, with the result that the information stored within it can be extracted.
(3) $\mathcal{A}$ can physically capture the RSU and extract the information stored in the RSU.
(4) $\mathcal{A}$ may obtain any temporary information about any of the vehicle, the RSU, and the server.
(5) $\mathcal{A}$ is able to query transactions recorded in the ledger in the blockchain, but cannot propose transactions or disrupt the blockchain system [38,39].
(6) $\mathcal{A}$ is restricted from accessing the private key stored in the server and the registration information of the communicating entity, ensuring the integrity and confidentiality of the communication process.
(7) Because passengers in the same private vehicle are usually family members or friends, passengers trust each other. Therefore, we assume that there is no malicious attacker among the passengers in the vehicle. In other words, $\mathcal{A}$ can be an outsider attacker but not an insider attacker.
(8) For (2), (3), and (4), $\mathcal{A}$ just performs one of these attacks, but not all of them.

4 The Proposed MSIoV-AKA Protocol

In this section, we propose an authentication and key agreement protocol, specifically designated as MSIoV-AKA. This protocol is systematically divided into four sequential phases: the pre-deployment phase, the registration phase, the login and authentication phase, and the passenger dynamic adding phase. In Table 1, we offer a comprehensive explanation of the notation utilized in the MSIoV-AKA protocol.

Table 1: Notations table

Symbol	Description
V_i, RSU_j	i-th vehicle, and j-th RSU, respectively
$Server$	Deployed metaverse servers
U_c	c-th passenger in the vehicle.
VID_i, ID_c, GID_j	Identities of V_i, U_c, RSU_j
$PVID_i, PID_c$	Pseudo identity of V_i, U_c
SK	Session key
k_s	Secret key of $Server$
$f(x)$	Galois field polynomial with degree t.

4.1 Pre-Deployment Phase

At this stage, the main task is to initialize some parameters for the communication entities. Firstly, vehicles and RSUs have a unique identifier set at the factory. In our protocol, the identifier is used as the device's ID. The ID of vehicles and RSUs are denoted as VID_i and GID_j, respectively. Then, the server selects a private key k_s and a large prime number q, secure hash functions $h_1: \{0,1\}^* \rightarrow \{0,1\}^l$, $h_2: \{0,1\} \rightarrow Z_p^*$. Finally, the server exposes the parameters $\{h_1, h_2, q\}$, and securely stores $\{k_s\}$.

4.2 Registration Phase

4.2.1 V_i Registration

Before passengers want to use metaverse social services in a vehicle V_i, they need to register with $Server$.

(1) First, the passenger P_c selects the identity ID_c and the password PW_c. Then, V_i sends VID_i and ID_c to $Server$.

(2) When $Server$ receives the message, it computes $PID_c = h_1(ID_c||VID_i||k_s), PVID_i = h_1(VID_i||k_s), R_i = h_1(s_0||k_s||VID_i)$, and $x_c = h_2(ID_c||VID_i)$. $Server$ constructs the $t-1$ degree polynomial:

$$f(x) = s_0 + a_1 x + \cdots + a_{t-1}x^{t-1} \tag{1}$$

where the constant term s_0 is the secret value, $a_1, a_2 \ldots a_{t-1}$ are elements in the Galois Field $GF(2^q)$. $Server$ then computes y_c and stores $\{s_0, a_1, a_2 \ldots a^{t-1}, VID_i, PVID_i, ID_c, PID_c\}$. Finally, $Server$ sends $\{PID_c, PVID_i, R_i, y_c\}$ to V_i.

(3) After V_i receives the message, it computes $Z_c = h_1(ID_c||PW_c||R_i||VID_i)$. Finally, V_i stores $\{PVID_i, R_i, Z_c, PID_c, y_c\}$ into its OBU.

4.2.2 RSU_j Registration

(1) RSU_j transmits $\{GID_j\}$ to $Server$ via a secure channel.

(2) When $Server$ receives the message, it selects the random number r_j and computes $PGID_j = h_1(GID_j||r_j||k_s), GT_j = h_1(r_j||K_s)$. Finally, Server stores $\{PGID_j, GID_j, r_j\}$ and sends $\{PGID_j, GT_j\}$ to RSU_j.

(3) After receiving the message, RSU_j calculates $GP_j = GT_j \oplus h_1(PGID_j||GID_j)$. Finally, RSU_j stores $\{PGID_j, GP_j\}$ into its database.

4.3 Login and Authentication Phase

During the journey, passengers (U_c) in the vehicle (V_i) can socialise through the metaverse server ($Server$). Firstly, U_c needs to complete the login and authentication in the V_i. Then, through RSU_j, $Server$ completes the authentication of V_i and U_c. Finally, $Server$ establishes a session key with the U_c. Fig. 3 illustrates the overall process of user login and authentication. The detailed procedure is described as follows:

(1) Firstly, the U_c in the V_i input his ID_c and PW_c. Then, V_i computes $Z_c^* = h_1(ID_c||PW_c||R_i||VID_i)$ and checks that $Z_c^* \stackrel{?}{=} Z_c$. If it passes the validation, V_i selects the random number α and the timestamp T_1 to compute $HID_i = h_1(ID_1||ID_2||...||ID_n||VID_i), I_1 = \alpha \oplus h_1(R_i||HID_i||T_1), I_2 = PID_1||PID_2||...||PID_n, V_1 = h_1(VID_i||I_2||\alpha||T_1)$. Finally, V_i sends the $\{PVID_i, I_1, I_2, V_1, T_1\}$ to RSU_j.

(2) After RSU_j receives the message, it checks $|T_1 - T_c| \leq \Delta T$. If T_1 is valid, RSU_j picks the random number β and timestamp T_2 to calculate $GT_j = GP_j \oplus h_1(PGID_j||GID_j), I_3 = \beta \oplus GT_j, V_2 = h_1(\beta||PGID_j||T_2)$. Finally, RSU_j sends $\{M_1, PGID_j, I_3, V_2, T_2\}$ to $Server$.

(3) When $Server$ receives the message, it checks $|T_2 - T_c| \leq \Delta T$. If T_2 is valid, $Server$ calculates $GT_j = h_1(r_j||k_s), \beta = I_3 \oplus GT_j, V_2^* = h_1(\beta||PGID_j||T_2)$ and then checks that $V_2 \stackrel{?}{=} V_2$. If equal, $Server$ retrieves ID_c, VID_i based on $PID_c, PVID_i$. As shown in Fig. 4, $Server$ performs $Algorithm1$ to reconstruct s_o to calculate $R_i = h_1(s_0||k_s||VID_i), HID_i = h_1(ID_1||ID_2||...||ID_n||VID_i), \alpha = I_1 \oplus h_1(R_i||HID_i||T_1), V_1^* = h_1(VID_i||I_2||\alpha||T_1)$. $Server$ checks that $V_1 \stackrel{?}{=} V_1$. If equal, $Server$ picks random number γ and timestamp T_3 to compute $SK = h_1(s_0||\alpha||\beta||\gamma||HID_i), I_3 = (\beta||\gamma) \oplus h_1(s_0||HID_i||R_i||\alpha), V_3 = h_1(SK||I_3||T_3)$. Finally, $Server$ sends $\{I_3, V_3, T_3\}$ to RSU_j.

(4) RSU_j receives the message and checks $|T_3 - T_c| \leq \Delta T$. If T_3 is valid, RSU_j picks T_4 and sends $\{M_3, T_4\}$ to V_i.

(5) When V_i receives the message, it checks $|T_4 - T_c| \leq \Delta T$. If T_4 is valid, V_i computes $x_c = h_2(ID_c||VID_i)$ and executes $Algorithm1$ to obtain s_0. Then, V_i computes $(\beta||\gamma) = I_3 \oplus h_1(s_0||\alpha||\beta||\gamma||HID_i)$, $SK = h_1(s_0||\alpha||\beta||\gamma||HID_i)$, and $V_3^* = h_1(SK||I_3||T_3)$. Finally, V_i checks $V_3 \stackrel{?}{=} V_3$. If equal, it indicates that SK is valid.

$Vehicle(V_i)/User(U_c)$	RSU_j	$Server$
User $U_c(c \in [1,n])$ input ID_c, PW_c		
$Z_c^* = h_1(ID_c \| PW_c \| R_i \| VID_i)$		
Check $Z_c^* \stackrel{?}{=} Z_c$		
Select α, T_1		
$HID_i = h_1(ID_1 \| ID_2 \| \ldots \| ID_n \| VID_i)$		
$I_1 = \alpha \oplus h_1(R_i \| HID_i \| T_1)$		
$I_2 = h_1(PID_1 \| PID_2 \| \ldots \| PID_n)$		
$V_1 = h_1(VID_i \| I_2 \| \alpha \| T_1)$		
$M_1 = \{PVID_i, PID_c, I_1, I_2, V_1, T_1\} \longrightarrow$		
	Check $\lvert T_1 - T_c \rvert \le \Delta T$	
	Select β, T_2	
	$GT_j = GP_j \oplus h_1(PGID_j \| GID_j)$	
	$I_3 = \beta \oplus GT_j$	
	$V_2 = h_1(\beta \parallel PGID_j \| T_2)$	
	$M_2 = \{M_1, PGID_j, I_3, V_2, T_2\} \longrightarrow$	
		Check $\lvert T_2 - T_c \rvert \le \Delta T$
		$GT_j = h_1(r_j \| k_s)$
		$\beta = I_3 \oplus GT_j$
		$V_2^* = h_1(\beta \| PGID_j \| T_2)$
		Check $V_2^* \stackrel{?}{=} V_2$
		Retrieve ID_c, VID_i according to $PID_c, PVID_i$
		$R_i = h_1(s_0 \parallel k_s \| VID_i)$
		$HID_i = h_1(ID_1 \| ID_2 \| \ldots \| ID_n \| VID_i)$
		$\alpha = I_1 \oplus h_1(R_i \parallel HID_i \| T_1)$
		$V_1^* = h_1(VID_i \| I_2 \| \alpha \| T_1)$
		Check $V_1^* \stackrel{?}{=} V_1$
		Select γ, T_3
		$SK = h_1(s_0 \| \alpha \parallel \beta \| \gamma \| HID_i)$
		$I_3 = (\beta \| \gamma) \oplus h_1(s \| HID_i \| R_i \| \alpha)$
		$V_3 = h_1(SK \| I_3 \| T_3)$
		$M_3 = \{I_3, V_3, T_3\} \longleftarrow$
	Check $\lvert T_3 - T_c \rvert \le \Delta T$	
	Select T_4	
	$M_4 = \{M_3, T_4\} \longleftarrow$	
Check $\lvert T_4 - T_c \rvert \le \Delta T$		
$x_c = h_2(ID_c \| VID_i)$		
Execute *Algorithm* 1, and obtain s_0		
$(\beta \| \gamma) = I_3 \oplus h(s_0 \| HID_i \| R_i \| \alpha)$		
$SK = h(s_0 \| \alpha \| \beta \| \gamma \| HID_i)$		
$V_3^* = h(SK \| I_3 \| T_3)$		
Check $V_3^* \stackrel{?}{=} V_3$		

Figure 3: Login and authentication phase

Algorithm 1 : *Reconstruction of* s_0

1. Input $(x_c, y_c), c \in [1,n]$
2. $s = 0$
3. $Y_m = 0$
4. for $m = 1$ to w
5. $\cdots X_m = 1$
6. $\cdots$ for $n = 1$ to w
7. $\cdots\cdots$ if $(n\ !=m)$
8. $\cdots\cdots\cdots X_m = X_m \cdot \frac{-x_n}{x_n - x_m}$
9. $\cdots Y_m = y_m \cdot X_m$
10. $\cdots s = s + Y_m$
11. return s

Figure 4: s_0 reconstruction algorithm

4.4 Passenger Dynamic Adding Phase

If a new passenger also wants to use the metaverse service, the passenger needs to perform the dynamic addition phase. Compared to the vehicle registration phase, the dynamic passenger addition phase allows for fast passenger registration.

(1) The new passenger P_c' chooses its ID_c', PW_c'. Then, V_i sends $\{ID_c', VID_i\}$ to $Server$.
(2) When $Server$ receives the message, it calculates $PID_c' = h_1(ID_c'||VID_i||k_s)$, $x_c' = h_2(ID_c'||VID_i)$, $y_c' = f(x_c')$. Then, $Server$ stores $\{ID_c', PID_c'\}$ and sends $\{PID_c', y_c'\}$ to V_i.
(3) After V_i receives the message, it computes $Z_c' = h_1(ID_c'||PW_c'||R_i||VID_i)$ and stores $\{Z_c', PID_c', y_c'\}$ in V_i.

4.5 Password Update Phase

To ensure the security of passenger credentials in long-term operation scenarios and to meet the compliance requirements for password rotation, the MSIoV-AKA protocol introduces a password update phase. This phase is triggered only after a successful login-authentication session between the passenger U_c and the vehicle V_i. The $Server$ never stores plaintext or reversible password images, adhering to the principle of minimal trust.

Let the old and new passwords be denoted as PW_c^{old} and PW_c^{new}, respectively. A new 128-bit random number δ and a timestamp T_5 are involved.

(1) U_c enters PW_c^{old} and PW_c^{new} on the in-vehicle terminal.
(2) V_i computes $Z_c^* = h_1(ID_c \parallel PW_c^{old} \parallel R_i \parallel VID_i)$ and verifies $Z_c^* \stackrel{?}{=} Z_c$. If the check fails, the update is rejected.
(3) V_i selects a random value δ and records the current timestamp T_5 to calculate $Z_c^{new} = h_1(ID_c \parallel PW_c^{new} \parallel R_i \parallel VID_i)$.
(4) Finally, V_i returns Z_c^{new} to U_c.

5 Security Analysis

5.1 Formal Security Analysis

In this section, we perform a comprehensive formal security analysis of the MSIoV-AKA protocol. By employing different games, we calculated the probability of an attacker ($\mathcal{A}$) compromising MSIoV-AKA protocol.

Our protocol consists of three entities, namely V_i, RSU_j, and $Server$. In the following, we use $\Pi_{V_i}^x$, $\Pi_{RSU_j}^y$, and Π_{Server}^z to represent the instance of x-th vehicle, y-th RSU, and z-th Server. In addition, we use a variety of query operations to simulate the real attacks by $\mathcal{A}$. In the following, we will demonstrate various query operations.

1. $Execute(E)$: By this query, $\mathcal{A}$ has the ability to obtain all messages that are transmitted over the public channel, thereby gaining complete access to the information flowing.
2. $Send(E, M_i)$: By this query, $\mathcal{A}$ can send a message to any entity $E = \{\Pi_{V_i}^x, \Pi_{RSU_j}^y, \Pi_{Server}^z\}$.
3. $Hash(String)$: By this query, $\mathcal{A}$ can obtain the hash value of any string.
4. $Corrupt(OBU)$: By this query, $\mathcal{A}$ can extract the information stored in the vehicle.
5. $Test(E)$: By tossing an unbiased coin (c), $\mathcal{A}$ guesses the session key. If $c = 1$, $\mathcal{A}$ can obtains the session key. If $c = 0$, $\mathcal{A}$ obtains a random string.

Theorem 1: *Under the ROR model, the probability that $\mathcal{A}$ breaks MSIoV-AKA protocol $\mathcal{P}$ in polynomial time ξ is $Adv_{\mathcal{A}}^{\mathcal{P}}(\xi) \leq \frac{q_h^2}{|Hash|} + 2 \cdot C' \cdot q_{send}^{s'}$. Here, $|Hash|$, q_h, and q_{send}, denote the range space of the hash function, the number of hash queries and the number of send queries, respectively.*

We define four games: GM_0-GM_3 to simulate $\mathcal{A}$'s attack process. In the proof process, $Succ_{\mathcal{A}}^{GM_i}(\xi)$ represents the probability of $\mathcal{A}$'s success in GM_i. $Adv_{\mathcal{A}}^{\mathcal{P}}(\xi)$ represents $\mathcal{A}$'s advantage in breaking the protocol. Below, we demonstrate the specific proof process.

GM_0: In GM_0, $\mathcal{A}$ starts the real attack by flipping an unbiased coin c. Thus, we can obtain

$$Adv_{\mathcal{A}}^{\mathcal{P}}(\xi) = |2Pr[Succ_{\mathcal{A}}^{GM_0}(\xi)] - 1|. \tag{2}$$

GM_1: GM_1 adds the $Execute(E)$ query to GM_0. By executing the $Execute(E)$ query, $\mathcal{A}$ can successfully eavesdrop on M_1, M_2, M_3, and M_4 sent over the public channel, enabling it to access the content of these messages. At this point, $\mathcal{A}$ tries to compute $SK = h_1(s_0||\alpha||\beta||\gamma||HID_i)$. However, $\mathcal{A}$ cannot obtain s_0, α, β, γ, and HID_i, so $\mathcal{A}$ cannot compute SK. Therefore, the probability of GM_1 equals GM_0,

$$Pr[Succ_{\mathcal{A}}^{GM_1}(\xi)] = Pr[Succ_{\mathcal{A}}^{GM_0}(\xi)]. \tag{3}$$

GM_2: Based on GM_1, GM_2 adds $Send(\cdot)$ and $Hash(\cdot)$ queries. In this case, $\mathcal{A}$ cannot tamper with the information transmitted over the public channel because the hash function protects the authentication value. In addition, the authentication value contains random numbers, which are of great significance in preventing collisions in the hash function, thus ensuring the integrity and security of the system. By considering the birthday paradox, we can deduce

$$|Pr[Succ_{\mathcal{A}}^{GM_2}(\xi)] - Pr[Succ_{\mathcal{A}}^{GM_1}(\xi)]| \leq \frac{q_h^2}{2|Hash|}. \tag{4}$$

GM_3: Different from GM_2, GM_3 removes the $Hash(\cdot)$ query and adds the $Corrupt(OBU)$ query. By $Corrupt(OBU)$ query, $\mathcal{A}$ can get $\{PVID_i, R_i, Z_c, PID_c, y_c\}$ stored by OBU. However, due to the absence of parameters ID_c, PW_c, VID_i, $\mathcal{A}$ still cannot calculate $SK = h_1(s_0||\alpha||\beta||\gamma||HID_i)$. $\mathcal{A}$ can only crack the semantic security of MSIoV-AKA by guessing the user's password. According to the Zipf's Law [40], the maximum probability of $\mathcal{A}$ guessing the password is $C' \cdot q_{send}^{s'}$, where C' and s' are two constants. Thus,

$$|Pr[Succ_{\mathcal{A}}^{GM_3}(\xi)] - Pr[Succ_{\mathcal{A}}^{GM_2}(\xi)]| \leq C' \cdot q_{send}^{s'}. \tag{5}$$

Finally, $\mathcal{A}$ can only guess bit c via the $Test(\cdot)$ query to win the game. Thus, we have

$$Pr[Succ_{\mathcal{A}}^{GM_3}(\xi)] = \frac{1}{2}. \tag{6}$$

According to GM_0 to GM_3, we can obtain

$$\begin{aligned} \frac{Adv_{\mathcal{A}}^{\mathcal{P}}(\xi)}{2} &= \left| Pr[Succ_{\mathcal{A}}^{GM_0}(\xi)] - \frac{1}{2} \right| \\ &= |Pr[Succ_{\mathcal{A}}^{GM_0}(\xi)] - Pr[Succ_{\mathcal{A}}^{GM_3}(\xi)]| \\ &= |Pr[Succ_{\mathcal{A}}^{GM_1}(\xi)] - Pr[Succ_{\mathcal{A}}^{GM_3}(\xi)]| \\ &\leq \sum_{i=1}^{2} |Pr[Succ_{\mathcal{A}}^{GM_{i+1}}(\xi)] - Pr[Succ_{\mathcal{A}}^{GM_i}(\xi)]| \end{aligned} \tag{7}$$

$$\leq \frac{q_h^2}{2|Hash|} + C' \cdot q_{send}^{s'}.$$

Finally, we can obtain

$$Adv_{\mathcal{A}}^{\mathcal{P}}(\xi) \leq \frac{q_h^2}{|Hash|} + 2 \cdot C' \cdot q_{send}^{s'}. \tag{8}$$

5.2 Formal Security Verification Using AVISPA

AVISPA is a commonly used validation tool. It uses the on-the-fly model checker (OFMC) and the constraint logic-based attack searcher (CL-AtSe) to verify the security of the MSIoV-AKA protocol. In Fig. 5, we show the simulation results from OFMC and CL-AtSe. In the OFMC analysis, when the node depth is 12, it took 6.07 s to access 2704 nodes. For CL-AtSe, the translation time was 0.09 s. Both results show that the proposed protocol is secure.

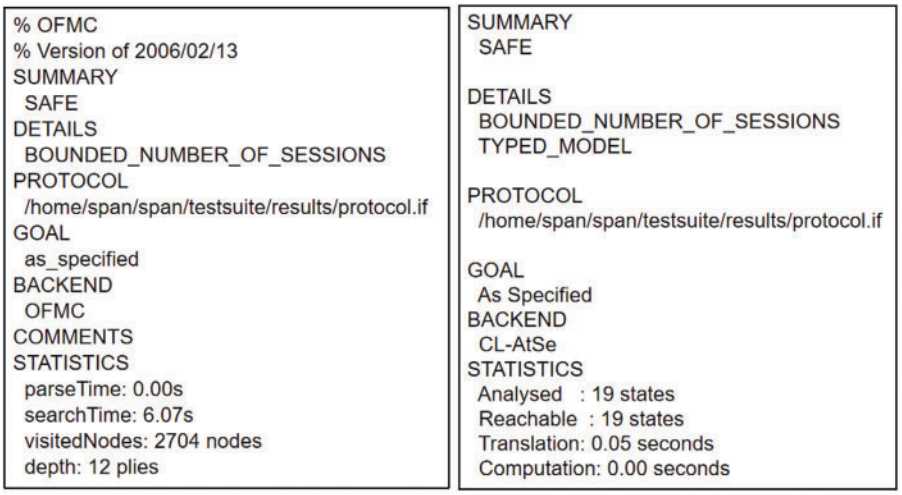


Figure 5: The verification results using OFMC and CL-AtSe

6 Performance Comparisons and Simulations

In this section, we conduct a comparative analysis between our proposed protocol and four representative robust protocols/schemes [41–44]. The comparison primarily focuses on two key metrics: computational cost and communication cost, aiming to comprehensively evaluate the efficiency of our protocol. Furthermore, we deploy smart contracts on both Ethereum and Hyperledger platforms to assess their runtime performance and validate the practical feasibility of the proposed scheme.

6.1 The Comparisons of Computational Costs and Communication Costs

In this subsection, we use three different devices to simulate the entities summarized in Table 2. The Honor 70 phone is used to act as the OBU of the vehicle, the Xiaomi 14 phone plays the role of the RSU, and a Lenovo computer is used to simulate the server. In Table 3, we show the computational costs of the used operations. Here, T_h means the execution time to run the hash function, T_{puf} is the execution time to run the PUF operation, T_{ecc} is the execution time to run the point scalar multiplication operation in ECC, and T_{cm}

is the execution time to run for Chebyshev polynomial operation. In Table 4, we show the compared results with four representative robust protocols/schemes [41–44]. The computational costs of these are significantly higher than our protocol.

Table 2: Equipment configuration parameters

	Honor 70	**Xiaomi 14**	**LenDovo laptop**
Operating system	Magic UI 6.1 (based on Android 12)	Android 14	Windows 10
Running memory	12 G	16 G	16 G
CPU	Qualcomm Snapdragon 778 G plus @2.5 GHz	Qualcomm Snapdragon 8 Gen 3 @ 3.3 GHz	Intel(R) Core(TM) i7-13700 CPU

Table 3: The execution time of operations in the three entities

Symbols	V_i	RSU_j	$Server$
T_h	41.2 µs	17.9 µs	8.76 µs
T_{puf}	52.21 µs	29.1 µs	10.18 µs
T_{ecc}	308.66 µs	236.32 µs	196.5 µs
T_{cm}	102.89 µs	78.73 µs	65.5 µs

Table 4: The comparisons of computational costs

Protocols	V_i	RSU_j	$Server$	**Total**
Modarres and Sarbishaei [41]	$12T_h + T_{puf} \approx 546.61$ µs	$3T_h \approx 53.7$ µs	$13T_h \approx 113.88$ µs	714.19 µs
Al Sibahee et al. [42]	$10T_h \approx 412$ µs	$7T_h \approx 125.3$ µs	$13T_h \approx 113.88$ µs	651.1 µs
Tomar and Tripathi [43]	$8T_h + 6T_{cm} \approx 946.94$ µs	$5T_h + 6T_{cm} \approx 561.88$µs	$8T_h + 10T_{cm} \approx 725.08$ µs	2233.9 µs
Awais et al. [44]	$7T_h + 4T_{ecc} \approx 1523.04$ µs	$4T_h + 5T_{ecc} \approx 1253.2$ µs	$9T_h + 6T_{ecc} \approx 1257.84$ µs	4034.08 µs
Our Protocol	$8T_h \approx 329.6$ µs	$2T_h \approx 35.8$ µs	$9T_h \approx 78.84$ µs	444.24 µs

According to [45], we define the length of identity, password, PUF challenge value, hash value, random number, timestamp and point in ECC as 128, 128, 128, 256, 256, 32, 160 bits, respectively. In the following, we show the communication cost of our protocol as an example. In our protocol, the transmitted messages are $\{PVID_i, I_1, I_2, V_1, T_1, PGID_j, I_3, V_2, T_2, I_3, V_3, T_3, T_4\}$, where $\{T_1, T_2, T_3, T_4\}$ is the timestamp, and $\{PVID_i, I_1, I_3, V_1, PGID_j, I_3, V_2, I_3, V_3\}$ is the hash value. Therefore, the communication cost of our protocol is 4544 bits. The compared results of communication costs are summarized in Table 5.

Table 5: The cost comparison of communication

Protocol	V_i	RSU_j	$Server$	Total
Modarres and Sarbishaei [41]	896 bits	1536 bits	640 bits	3072 bits
Al Sibahee et al. [42]	1056 bits	800 bits	1728 bits	3584 bits
Tomar and Tripathi [43]	1184 bits	3264 bits	1312 bits	5760 bits
AWais et al. [44]	672 bits	1824 bits	1184 bits	3680 bits
Our Protocol	1312 bits	2688 bits	544 bits	4544 bits

6.2 Feasibility Analysis of Blockchain

To validate the feasibility of our protocol, we executed our protocol on Ethereum and Hyperledger.

6.2.1 Ethereum-Based Implementation

In order to figure out the gas cost, we put the smart contract of our protocol on the Ethereum Test Network called Sepolia. The configurations of our implementation are as follows: Development environment: Remix, Language: Solidity, Compiler: 0.8.25+commit.b61c2a91, Ethereum wallet: MetaMask 11.14.0, Test network: Sepolia. First, we connected Remix and Sepolia using the MetaMask plugin in Google Chrome. Then, we use them to deploy and invoke our smart contract.

In Fig. 6, we show the Sepolia testnet transaction details. Fig. 6a–d represents the invocation results of the contract of vehicle registration, the contract of RSU registration, the contract of server submission, and the contract of passenger dynamic addition, respectively. Based on the exchange rate on 18 August 2024, we consider 1 Ether = 2644.279 USD. In Table 6, we show the cost of deployment and invocation for the four contracts. The results indicate that the cost of deployment is the highest; however, it only needs to be executed once. Although the contract needs to be invoked multiple times, the cost of invoking is low. Therefore, the gas cost of our protocol is acceptable in practical applications.

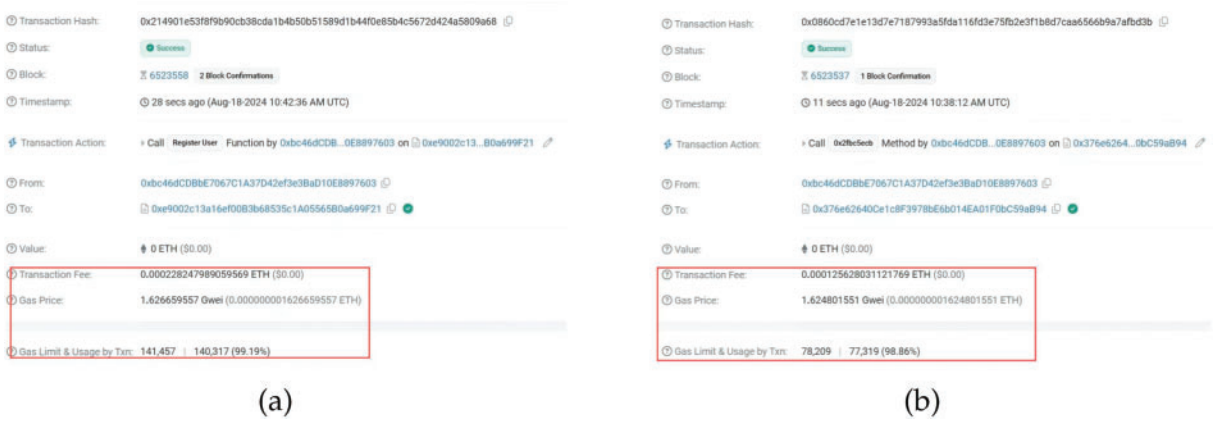


Figure 6: (Continued)

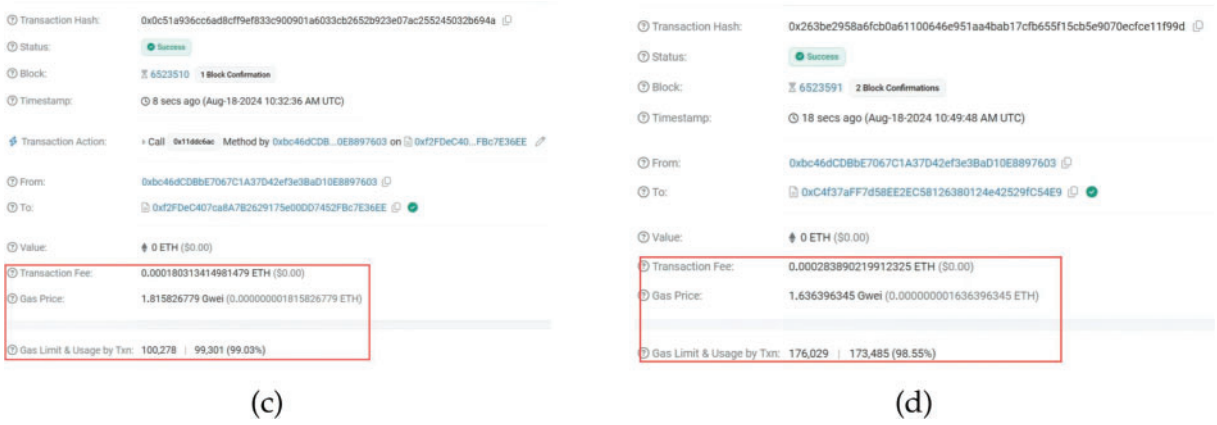


Figure 6: Sepolia testnet transaction details

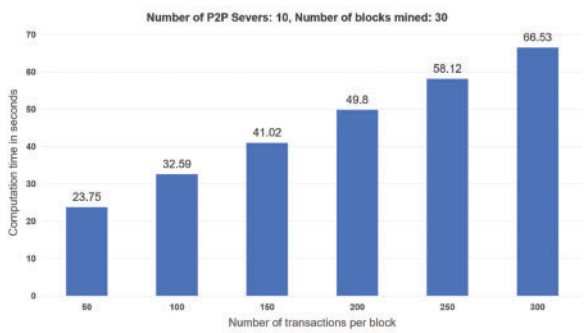
Table 6: Smart contract gas cost

Function	Contract deployment/Execution		
	Gas	Ether	USD
Vehicle registration	319,144/141,457	0.000,512/0.000,228	1.354/0.603
RSU registration	327,208/78,209	0.000,542/0.000,126	1.433,3/0.333
Server submission	594,236/176,209	0.000,966/0.000,284	2.554/0.751
Passenger dynamic addition	296,688/100,278	0.000,577/0.000,180	1.526/0.476

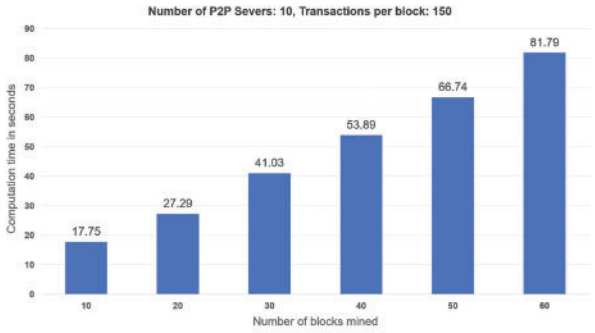
6.2.2 Hyperledger-Based Implementation

In order to conduct a comprehensive analysis of the time cost associated with packing and uploading blocks, we carry out experiments on the Hyperledger Sawtooth blockchain platform. Our experimental configurations and environments are described as follows: Operating system: Ubuntu 20.04.6 LTS, CPU: 11th Gen Inter(R) Core(TM) i9-11900 @ 2.50 GHz, RAM: 16G, Platform: Hyperledger Sawtooth, Development Environment: Pycharm 2023.3.5 (Community edition), Programming Language: python, Consensus Algorithm: Practical Byzantine Fault Tolerance (PBFT). Specifically, we used Docker 26.2.4 to create 10 containers to simulate blockchain nodes. Each node has an intkey transaction processor, a REST API service, a transaction processor, a validation server, and a PBFT engine. In addition, we pair the nodes to form a peers network. Through PBFT, each node can confirm transactions and reach consensus. Finally, the nodes package the transactions into blocks and upload them to the blockchain.

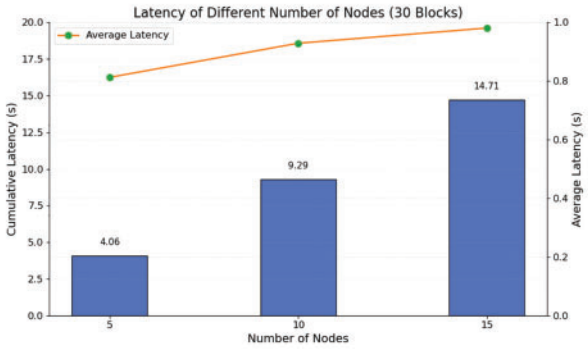
Here, we evaluate the performance of MSIoV-AKA protocol in three cases. Case 1 is shown in Fig. 7a. We assume that there are 30 blocks to be packed and uploaded. We compared the time consumed for packaging and uploading blocks containing different numbers of transactions. Case 2 is shown in Fig. 7b. We assume that each block contains 150 transactions. We compared the time consumed for packaging and uploading different numbers of blocks. Fig. 7c shows that the left vertical represents the cumulative latency for processing 30 blocks under different numbers of nodes, illustrated by the bar chart. The right vertical axis shows the average latency per block, depicted as a line chart.



(a) Case 1: Varying transactions per block



(b) Case 2: Varying number of blocks mined



(c) The latency time of the blockchain with different numbers of nodes

Figure 7: Hyperledger-based blockchain simulation results

7 Conclusion

In this paper, we have introduced a novel "SIoV combined with Metaverse" environment and have defined the system model and attacker model. Based on this environment, an authentication and key agreement protocol using blockchain called MSIoV-AKA is proposed. The formal security analysis in the RoR model and the AVISPA tool are used to verify the security of MSIoV-AKA. Finally, the theoretical comparisons of computational/communication costs and the feasibility analysis of blockchain provide evidence that our MSIoV-AKA is suitable in practice.

Acknowledgement: Not applicable.

Funding Statement: This work was supported by the Startup Foundation for Introducing Talent of Nanjing University of Information Science and Technology and Natural Science Foundation of Shandong Province, China (Grant no. ZR202111230202).

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Tsu-Yang Wu and Chien-Ming Chen; methodology, Tsu-Yang Wu and Haozhi Wu; validation, Maoxin Tang; formal analysis, Haozhi Wu and Maoxin Tang; investigation, Saru Kumari and Chien-Ming Chen; data curation, Saru Kumari and Chien-Ming Chen; writing—original draft preparation, Tsu-Yang Wu, Haozhi Wu, Maoxin Tang, Saru Kumari, and Chien-Ming Chen; writing—review and editing, Tsu-Yang Wu, Haozhi Wu, Maoxin Tang, Saru Kumari, and Chien-Ming Chen. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data are contained within the article.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Wu TY, Wu H, Kumari S, Chen CM. An enhanced three-factor based authentication and key agreement protocol using PUF in IoMT. Peer Peer Netw Appl. 2025;18(2):83. doi:10.1007/s12083-024-01839-z.
2. Li Y, Chen C, Zhang Y, Liu W, Lyu L, Zheng X, et al. Ultrare: enhancing receraser for recommendation unlearning via error decomposition. Adv Neural Inform Process Syst. 2023;36:12611–25.
3. Xue X, Mei Y, Zhao B, Zhang M. Adaptive similarity feature construction for ontology matching via multi-layer hybrid genetic programming. IEEE Trans Evol Comput. 2025;1. doi:10.1109/tevc.2025.3547578.
4. Dong S, Liu M, Abbas K. The metaverse review: exploring the boundless ream of digital reality. Comput Mater Contin. 2024;81(3):3451–98. doi:10.32604/cmc.2024.055575.
5. Chen C, Zhang Y, Li Y, Wang J, Qi L, Xu X, et al. Post-training attribute unlearning in recommender systems. ACM Transact Informat Syst. 2024;43(1):1–28. doi:10.1145/3701987.
6. Huynh-The T, Gadekallu TR, Wang W, Yenduri G, Ranaweera P, Pham QV, et al. Blockchain for the metaverse: a review. Fut Generat Comput Syst. 2023;143(9):401–19. doi:10.1016/j.future.2023.02.008.
7. Nguyen CT, Hoang DT, Nguyen DN, Xiao Y, Niyato D, Dutkiewicz E. Metashard: a novel sharding blockchain platform for metaverse applications. IEEE Transact Mobile Comput. 2023;23(5):4348–61. doi:10.1109/tmc.2023.3290955.
8. Rafique W, Qadir J. Internet of everything meets the metaverse: bridging physical and virtual worlds with blockchain. Comput Sci Rev. 2024;54(4):100678. doi:10.1016/j.cosrev.2024.100678.
9. Wang H, Zhang F, Shen Z, Liu P, Liu K. Blockchain-based IVPPA scheme for pseudonym privacy protection in internet of vehicles. J Network Intell. 2024;9(2):1260–77.
10. Su H, Dong S, Zhang T. A hybrid blockchain-based privacy-preserving authentication scheme for vehicular Ad Hoc networks. IEEE Transact Vehic Technol. 2024;73(11):17059–72. doi:10.1109/tvt.2024.3424786.
11. Lu X, fang D. Remote vehicle exhaust detection based on integrated deep learning models for environmental monitoring. J Netw Intellig. 2025;10(1):527–40.
12. Silva R, Iqbal R. Ethical implications of social internet of vehicles systems. IEEE Int Things J. 2018;6(1):517–31. doi:10.1109/jiot.2018.2841969.
13. Campolo C, Molinaro A, Iera A. A reference framework for social-enhanced Vehicle-to-Everything communications in 5G scenarios. Comput Netw. 2018;143(6):140–52. doi:10.1016/j.comnet.2018.07.010.
14. Chen Y, Zhou T, Zhou J, Cao Z, Dong X, Choo KKR. SAVE: efficient privacy-preserving location-based service bundle authentication in self-organizing vehicular social networks. IEEE Transact Intellig Transport Syst. 2021;23(8):11752–66. doi:10.1109/tits.2021.3106783.
15. Li Y, Tao X, Zhang X, Xu J, Wang Y, Xia W. A DAG-Based reputation mechanism for preventing peer disclosure in SIoV. IEEE Int Things J. 2022;9(23):24095–106. doi:10.1109/jiot.2022.3189108.
16. Mohanty SK, Tripathy S. SIoVChain: time-lock contract based privacy-preserving data sharing in SIoV. IEEE Transact Intellig Transport Syst. 2022;23(12):24071–82. doi:10.1109/tits.2022.3192566.
17. Javaid U, Sikdar B. A secure and scalable framework for blockchain based edge computation offloading in social internet of vehicles. IEEE Transact Vehic Technol. 2021;70(5):4022–36. doi:10.1109/tvt.2021.3060002.
18. Zhang L, Zhang Y, Wu Q, Mu Y, Rezaeibagha F. A secure and efficient decentralized access control scheme based on blockchain for vehicular social networks. IEEE Int Things J. 2022;9(18):17938–52. doi:10.1109/jiot.2022.3161047.
19. Xia Z, Man J, Gu K, Li X, Huang L. Conditional data-sharing privacy-preserving scheme in blockchain-based social internet of vehicles. IEEE Transact Sustain Comput. 2025;10(2):378–95. doi:10.1109/tsusc.2024.3452228.
20. Liu L, Feng J, Wu C, Chen C, Pei Q. Reputation management for consensus mechanism in vehicular edge metaverse. IEEE J Select Areas Communicat. 2023;42(4):919–32. doi:10.1109/jsac.2023.3345382.
21. Kang J, Luo X, Nie J, Wu T, Zhou H, Wang Y, et al. Blockchain-based pseudonym management for vehicle twin migrations in vehicular edge metaverse. IEEE Int Things J. 2024;11(21):34254–69. doi:10.1109/jiot.2024.3404559.

22. Bansal G, Sikdar B. Achieving secure and reliable UAV authentication: a shamir's secret sharing based approach. IEEE Transact Netw Sci Eng. 2024;11(4):3598–610. doi:10.1109/tnse.2024.3381599.
23. Mohit P, Amin R, Biswas G. Design of authentication protocol for wireless sensor network-based smart vehicular system. Vehicular Communicat. 2017;9(1):64–71. doi:10.1016/j.vehcom.2017.02.006.
24. Yu S, Lee J, Lee K, Park K, Park Y. Secure authentication protocol for wireless sensor networks in vehicular communications. Sensors. 2018;18(10):3191. doi:10.3390/s18103191.
25. Sadri MJ, Rajabzadeh Asaar M. A lightweight anonymous two-factor authentication protocol for wireless sensor networks in Internet of Vehicles. Int J Commun Syst. 2020;33(14):e4511. doi:10.1002/dac.4511.
26. Jiang Y, Shen X, Zheng S. An effective data sharing scheme based on blockchain in vehicular social networks. Electronics. 2021;10(2):114. doi:10.3390/electronics10020114.
27. Esfahani A, Decouchant J, Völp M, Mumtaz S, Konstantin Igorevich K. SI-AKAV: secure integrated authentication and key agreement for cellular-connected IoT devices in vehicular social networks. Trans Emerg Telecomm Technol. 2024;35(4):e4279. doi:10.1002/ett.4279.
28. O'Brolcháin F, Jacquemard T, Monaghan D, O'Connor N, Novitzky P, Gordijn B. The convergence of virtual reality and social networks: threats to privacy and autonomy. Sci Eng Ethics. 2016;22(1):1–29. doi:10.1007/s11948-014-9621-1.
29. Falchuk B, Loeb S, Neff R. The social metaverse: battle for privacy. IEEE Technol Soc Magaz. 2018;37(2):52–61. doi:10.1109/mts.2018.2826060.
30. De Guzman JA, Thilakarathna K, Seneviratne A. Security and privacy approaches in mixed reality: a literature survey. ACM Comput Surv. 2019;52(6):1–37. doi:10.1145/3359626.
31. Ryu J, Son S, Lee J, Park Y, Park Y. Design of secure mutual authentication scheme for metaverse environments using blockchain. IEEE Access. 2022;10(1):98944–58. doi:10.1109/access.2022.3206457.
32. Zhang X, Huang X, Yin H, Huang J, Chai S, Xing B, et al. LLAKEP: a low-latency authentication and key exchange protocol for energy internet of things in the metaverse era. Mathematics. 2022;10(14):2545. doi:10.3390/math10142545.
33. Yang K, Zhang Z, Youliang T, Ma J. A secure authentication framework to guarantee the traceability of avatars in metaverse. IEEE Transact Inform Foren Secur. 2023;18:3817–32 doi: 10.1109/tifs.2023.3288689.
34. Thakur G, Kumar P, Chen CM, Vasilakos AV, Anchana, Prajapat S. A robust privacy-preserving ECC-based three-factor authentication scheme for metaverse environment. Comput Communicat. 2023;211(9):271–85. doi:10.1016/j.comcom.2023.09.020.
35. Gupta BB, Gaurav A, Arya V. Fuzzy logic and biometric-based lightweight cryptographic authentication for metaverse security. Appl Soft Comput. 2024;164(1):111973. doi:10.1016/j.asoc.2024.111973.
36. Dolev D, Yao A. On the security of public key protocols. IEEE Transact Inform Theory. 1983;29(2):198–208. doi:10.1109/tit.1983.1056650.
37. Canetti R, Krawczyk H. Analysis of key-exchange protocols and their use for building secure channels. In: International Conference on the Theory and Applications of Cryptographic Techniques. Berlin/Heidelberge: Springer; 2001. p. 453–74.
38. Zhang Y, Li B, Wu J, Liu B, Chen R, Chang J. Efficient and privacy-preserving blockchain-based multifactor device authentication protocol for cross-domain IIoT. IEEE Int Things J. 2022;9(22):22501–15. doi:10.1109/jiot.2022.3176192.
39. Chen CM, Xiong Z, Wu TY, Kumari S, Alenazi MJF. Protecting virtual economies: a blockchain-based anti-phishing authentication protocol for metaverse applications. IEEE Internet Things J. 2025;1. doi:10.1109/jiot.2025.3554788.
40. Wang D, Cheng H, Wang P, Huang X, Jian G. Zipf's law in passwords. IEEE Transact Inform Foren Secur. 2017;12(11):2776–91. doi:10.1109/tifs.2017.2721359.
41. Modarres AMA, Sarbishaei G. An improved lightweight two-factor authentication protocol for IoT applications. IEEE Transact Indus Inform. 2022;19(5):6588–98. doi:10.1109/tii.2022.3201971.

42. Al Sibahee MA, Nyangaresi VO, Abduljabbar ZA, Luo C, Zhang J, Ma J. Two-factor privacy preserving protocol for efficient authentication in internet of vehicles networks. IEEE Int Things J. 2023;11(8):14253–66. doi:10.1109/jiot.2023.3340259.
43. Tomar A, Tripathi S. A chebyshev polynomial-based authentication scheme using blockchain technology for fog-based vehicular network. IEEE Transact Mobile Comput. 2024;23(10):9075–89. doi:10.1109/tmc.2024.3357599.
44. Awais SM, Yucheng W, Mahmood K, Badar HMS, Kharel R, Das AK. Provably secure fog-based authentication protocol for VANETs. Comput Netw. 2024;246(1):110391. doi:10.1016/j.comnet.2024.110391.
45. Rafique F, Obaidat MS, Mahmood K, Ayub MF, Ferzund J, Chaudhry SA. An efficient and provably secure certificateless protocol for industrial Internet of Things. IEEE Transact Indus Inform. 2022;18(11):8039–46. doi:10.1109/tii.2022.3156629.