Mutual Authentication Protocol for Secure Vehicular Platoon Admission

Rohini Poolat Parameswarath
Department of ECE
College of Design and Engineering
National University of Singapore
Singapore
rohini.p@nus.edu.sg

Biplab Sikdar
Department of ECE
College of Design and Engineering
National University of Singapore
Singapore
bsikdar@nus.edu.sg

***Abstract*—Vehicular platooning offers several advantages and plays an important role in the future of mobility. However, this technology is vulnerable to several attacks. Since vehicles can freely join and leave a platoon, it is important to ensure that only legitimate vehicles are admitted into a platoon. In this paper, we propose a mutual authentication protocol to securely admit vehicles into a platoon. The proposed protocol is built on the concepts of Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs). Each vehicle creates its DID which helps to achieve vehicle identity privacy. The vehicles must register with a Trusted Authority (TA) to join platoons. The TA issues a VC to the registered vehicle. The platoon leader and the vehicle joining the platoon verify each other's VC before the vehicle joins the platoon. The security analysis demonstrates that the proposed protocol ensures secure platoon admission and preserves the privacy of vehicles. We also provide a proof of concept implementation of the blockchain network for the proposed scheme using Ethereum. We use the online Integrated Development Environment (IDE) Remix to compile and run the smart contract written in Solidity code for the proof of concept implementation. A performance analysis of the proposed protocol shows that its computation cost is less than that of other existing schemes for platoon admission.**

***Index Terms*—Mutual authentication, privacy, security, vehicular platoon, verifiable credentials.**

I. Introduction

In a vehicular platoon, a group of vehicles move together, following one another, with coordinated driving [1], [2]. Vehicular platooning offers several advantages such as saving space on highways, improving safety, and using less fuel [3], [4]. The vehicle in the front of the platoon is called the leader vehicle and the rest of the vehicles in the platoon are called followers. All the follower vehicles strictly follow the one in front of them and maintain a constant safe distance [3].

The vehicles in a platoon exchange control messages for coordinated driving. These messages contain information about their acceleration, braking, etc. [1]. Various technologies such as the Adaptive Cruise Control (ACC) system and Vehicle-to-Vehicle (V2V) communication protocol are used in vehicular platoon communication and management [5]. Since the ACC system and V2V communication protocol introduce attack surfaces in a vehicular platoon, the vehicles in a platoon are susceptible to insider (attacks from a platoon member vehicle) as well as outsider attacks (attacks from a platoon non-member vehicle) [1], [5], [6]. To prevent the outside attack and to keep the platoon secure, only the vehicles that have been admitted to the platoon are authorized to interact with each other [6]. However, in a platoon ghost attack, an adversary may impersonate a non-existing ghost vehicle [1]. Then, the adversary will have access to the control commands from the preceding vehicles and may send these messages to its succeeding vehicles. The adversary may modify the control messages leading to collisions that pose a threat to human safety. The adversary may carry out several attacks now as an insider. Some examples of these attacks include eavesdropping to collect data, denial-of-service to prevent platooning, replay attacks, and so on [3]. Also, the vehicles in a platoon share sensitive information such as vehicle identity and location identity while exchanging messages. If an adversary eavesdrops on these messages, he/she may be able to extract such sensitive information about the vehicles [7]. To address these problems, we propose a mutual authentication protocol for vehicular platoon admission in this paper. Before a new vehicle joins the platoon, the joining vehicle and the platoon leader authenticate each other. The proposed protocol leverages the concepts of Decentralized Identifiers (DIDs) [8] and Verifiable Credentials (VCs) [9].

A. Related Work

Han et al. addressed the platoon ghost attack in [1] where the attacker gains admission into a platoon but does not join the platoon physically and acts maliciously. In other words, the attacker impersonates a ghost vehicle that does not exist in the platoon. To address this issue, Han et al. proposed a scheme that verifies the physical context of vehicles to give admission into a platoon. The underlying idea is that the unique attributes of road surfaces will be the same for two adjacent vehicles. They utilized the accelerometer data for context verification. However, this protocol is vulnerable to record and replay attacks [10]. A challenge-response verification mechanism was used for platoon verification in [10]. The verifier transmits random checkpoints to a vehicle. The vehicle should reach these checkpoints within a given time. The verifier verifies the distance to the vehicle using a radar. The scheme in

979-8-3315-1778-6/24/$31.00 ©2024 IEEE

[10] did not address the anonymity and privacy of vehicles. A secure scheme for platoon access based on elliptic curve cryptography (ECC) was proposed in [11]. However, the scheme in [11] has a high processing delay. Hence, it could face scalability challenges when there are several vehicles. Lai et al. [12] proposed a framework to securely set up a platoon. This framework is based on attribute-based encryption and contributory key agreement. Vaas et al. used trajectories of two vehicles as proof for their co-presence to form a platoon in [13]. A protocol for platoon verification using Optical Camera Communications (OCC) was proposed in [14]. The protocol verifies the communication link between two vehicles in succession in a platoon. The protocols in [1], [10], [13] require a minimum of 10 seconds and sometimes more than a minute to transmit a verification key [14]. Hence, these protocols take too long to admit a vehicle into a platoon. Further, if the line of sight is interrupted within this time frame, e.g., if the line of sight is obscured by another vehicle, the platoon verification process will fail [14].

B. Motivation and Contributions

The vehicles in a platoon are prone to insider and outsider attacks. An adversary may impersonate a ghost vehicle that does not exist in the platoon. After that, he/she can access and modify control messages to induce a collision among vehicles in the platoon. Hence, ensuring that only legitimate vehicles join a platoon is crucial to protect against malicious interference with the platoon's control messages. Further, the platoon admission process should not take too long. Also, an adversary should not be able to extract sensitive information about the vehicles by eavesdropping on the exchanged messages. Motivated by these requirements, this paper makes the following contributions:

- **A mutual authentication protocol for vehicular platoon admission that preserves the privacy of vehicles:** The proposed protocol employs the concepts of DIDs and VCs to securely admit vehicles into a platoon. Blockchain is used as the supporting platform. The joining vehicle and the platoon leader verify each other's legitimacy by verifying their VCs and mutually authenticate each other before the vehicle gets admitted into the platoon. The vehicles create and manage their identities using DIDs. DIDs are used during authentication instead of real identities to preserve the privacy of vehicles. Also, the proposed protocol enables platoon admission of vehicles in a very short period.
- **Protection from several attacks:** The proposed protocol offers protection from several attacks such as replay, eavesdropping, and impersonation.
- **Security and performance analyses:** We provide an informal security analysis to demonstrate the proposed protocol's security features. We also provide a performance analysis of the proposed protocol to show that it is computationally efficient.
- **Proof of concept implementation using Ethereum:** We also provide a proof of concept implementation of the blockchain network for the proposed scheme with Ethereum [15] smart contracts using the Remix Integrated Development Environment (IDE) [16]. The smart contract is written in the Solidity programming language.

The rest of the paper is organized as follows. In Section II, the preliminaries, system model, and adversary model are presented. In Section III, we present the proposed mutual authentication protocol for platoon admission. Then, we present a proof of concept implementation in Section IV. We provide a discussion on the protocol's security features and performance in Section V. Finally, conclusions are given in Section VI.

II. Preliminaries, System and Adversary Model

In this section, we briefly introduce the building blocks (Decentralized Identifier and Verifiable Credential) of the proposed protocol, the system model, and the adversary model.

A. Preliminaries

Decentralized Identifier: DID is a decentralized digital identifier that is created and managed by its owner [8]. A DID maps to a DID document that resides on public ledgers such as blockchains. The DID document contains information about the owner, e.g., the public key required to authenticate the DID owner [8].

Verifiable Credentials: Verifiable credentials refer to claims that can be verified cryptographically [9]. A trusted *Issuer* signs credentials about the *Holder* of the VC. Since a digital signature is used, VCs are tamper-resistant, credible, and can be verified digitally by others. The *Holder* presents the VC to another party, the *Verifier*, to prove that he/she possesses the required credentials. The *Verifier* can verify the credentials by verifying the signature of the *Issuer* associated with the VC.

B. System Model

The system model is depicted in Figure 1. We consider a platoon with a leader L and followers F_i for $i \in \{1, 2, \cdots m\}$. Vehicles can freely join or leave the platoon. If a vehicle wants to join a platoon, it sends a request to the platoon's leader. The leader and the follower vehicles are registered with a Trusted Authority (TA). The TA registers vehicles and issues verifiable credentials to the registered vehicles. Each entity has a unique DID which maps to a DID document on the blockchain. The public key associated with the DID of an entity is stored on its DID document on the blockchain. The leader vehicle, the follower vehicles, and the TA communicate over the Internet.

C. Adversary Model

We consider the scenario where an adversary carries out a ghost attack. The goal of the adversary is to get admission into the platoon and inject false control messages [10]. In this case, the adversary gets admission into a platform but does not join the platoon [1]. Thus, the adversary impersonates a ghost vehicle that does not exist in the platoon. However, by getting admission into the platoon, the adversary will have access to the control commands and can modify the control

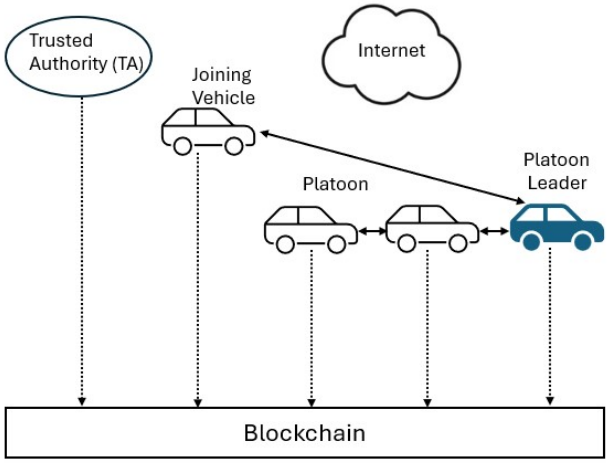

Fig. 1: System model.

messages to the vehicles following the ghost vehicle, resulting in potentially fatal accidents. The adversary may carry out several attacks now as an insider.

Also, the adversary may capture messages exchanged during the admission of a legitimate vehicle and replay them later to get admission into the platoon. He/she may generate messages impersonating a legitimate vehicle to get admission into the platoon as well. The adversary may also listen to the exchanged messages during a vehicle's platoon admission to get sensitive data such as the vehicle's identity. Then, the adversary can link specific vehicles to the exchanged messages and the platoon admission data will be available to the adversary. This poses a privacy threat to the vehicles.

III. Proposed Mutual Authentication Protocol

We now present the mutual authentication protocol for vehicular platoon admission. The proposed protocol consists of initialization, registration, and platoon admission phases. The initialization and registration phases are executed only once. The platoon admission phase is executed before admitting any follower vehicle into a platoon.

A. Initialization Phase

Step 1: The TA generates its DID, ID_{TA}, and the corresponding pair of private key (TA_{pr}) and public key (TA_{pu}). The DID document corresponding to the ID_{TA} is stored on the blockchain. The public key, TA_{pu}, is stored in this DID document of ID_{TA}. The private key, TA_{pr}, corresponding to ID_{TA} is securely stored by the TA, e.g., in its digital wallet.

Step 2: Similarly, each vehicle V_i with a vehicle identity, VID_{Vi}, generates its DID, ID_{Vi}, and the corresponding pair of private key (Vi_{pr}) and public key (Vi_{pu}). The vehicles store their DID, private key, and public key as mentioned in Step 1.

B. Registration Phase

In the registration phase, all the vehicles including the leader vehicle register with the TA. The steps involved in the vehicle registration phase are listed below:

Step 1: A vehicle V_i composes a message R_1 with a registration request, VID_{Vi}, and ID_{Vi}. V_i sends R_1 to the TA.

Step 2: Upon receiving R_1, the TA verifies whether the vehicle with an identity, VID_{Vi}, is not registered. This is performed by checking its database to see if a record for V_i already exists. If V_i is not registered, the TA generates a credential C_{Vi} and signs it with TA_{pr} to generate a VC, VC_{Vi}, for V_i. Then, the TA stores VID_{Vi}, ID_{Vi}, C_{Vi}, and VC_{Vi}. Finally, the TA composes a message R_2 with VC_{Vi} and sends it to V_i.

Step 3: V_i stores VC_{Vi}.

C. Platoon Admission Phase

When the vehicle V_i wants to join a platoon, the proposed protocol requires V_i and the platoon leader L_i to authenticate each other. The steps involved in this mutual authentication process are listed below:

Step 1: V_i composes a message M_1 with ID_{Vi} and a platoon joining request. Then, V_i sends $M_1 = \{ID_{Vi}, Req\}$ to L_i.

Step 2: Upon receiving M_1 from V_i to join the platoon, L_i generates a random number R_1. Let ID_{Li}, VC_{Li}, Li_{pu}, Li_{pr} denote L_i's DID, VC, public key, and private key, respectively. L_i gets the public key, Vi_{pu}, corresponding to ID_{Vi} from the blockchain. Subsequently, L_i encrypts $VC_{Li} \parallel R_1$ with Vi_{pu} to get $Enc[VC_{Li}]$. Finally, L_i composes M_2 = $\{ID_{Li}$, $Enc[VC_{Li}]\}$ and sends M_2 to V_i.

Step 3: Upon receiving M_2 from L_i, V_i generates a random number R_2. After that, V_i decrypts $Enc[VC_{Li}]$ with Vi_{pr} to get VC_{Li} and R_1. Then, it gets the public key, TA_{pu}, of the TA corresponding to ID_{TA} from the blockchain and verifies VC_{Li} using TA_{pu}. V_i gets the public key, Li_{pu}, corresponding to ID_{Li} from the blockchain. Subsequently, V_i encrypts $VC_{Vi} \parallel R_2$ with Li_{pu} to get $Enc[VC_{Vi}]$. V_i calculates the session key as $SK = R_1 \parallel R_2$. Finally, V_i composes M_3 = $\{Enc[VC_{Vi}]\}$ and sends M_3 to L_i.

Step 4: Upon receiving M_3, L_i decrypts $Enc[VC_{Vi}]$ with Li_{pr} to get VC_{Vi} and R_2. Then, it gets the public key, TA_{pu}, of the TA corresponding to ID_{TA} from the blockchain and verifies VC_{Vi} using TA_{pu}. V_i calculates the session key as $SK = R_1 \parallel R_2$. Thus, V_i and L_i authenticate each other by verifying their VCs and establish a session key. This session key is used for secure communication between V_i and L_i when V_i joins the platoon. The steps involved in the platoon admission phase are illustrated in Table I.

IV. Proof of Concept Implementation

In this section, we provide details of a proof of concept implementation of the blockchain network for the proposed scheme. We implemented the network with Ethereum [15] smart contracts using the Remix IDE [16]. Ethereum [15] is a

TABLE I: Platoon Admission Phase

Joining Vehicle (V_i)		**Platoon Leader** (L_i)
M_1= $\{ID_{Vi}, Req\}$		
	$\xrightarrow{M_1}$	
		Generate: R_1 Get: Vi_{pu} of ID_{Vi} from blockchain Encrypt: $(VC_{Li} \parallel R_1)$ with Vi_{pu} to get $Enc[VC_{Li}]$ M_2= $\{ID_{Li}, Enc[VC_{Li}]\}$
	$\xleftarrow{M_2}$	
Generate: R_2 Decrypt: $Enc[VC_{Li}]$ with Vi_{pr} Get: TA_{pu} of ID_{TA} from blockchain Verify: VC_{Li} using TA_{pu} Get: Li_{pu} of ID_{Li} from blockchain Encrypt: $VC_{Vi} \parallel R_2$ with Li_{pu} to get $Enc[VC_{Vi}]$ Calculate: $SK = R_1 \parallel R_2$ M_3= $\{Enc[VC_{Vi}]\}$		
	$\xrightarrow{M_3}$	
		Decrypt: $Enc[VC_{Vi}]$ with Li_{pr} Get: TA_{pu} of ID_{TA} from blockchain Verify: VC_{Vi} using TA_{pu} Calculate: $SK = R_1 \parallel R_2$

decentralized open-source blockchain system. Actions on the blockchain can be performed by using smart contracts. After compiling, the smart contract is deployed on the blockchain. Then, it is broadcast to all participating nodes on the network. The nodes validate the transaction and execute the smart contract code. The software environment Ethereum Virtual Machine (EVM) helps with executing the smart contract code. We used the online IDE Remix [16] to compile and run the smart contract written in Solidity code.

We used a personal computer with an Intel Core i5 CPU, 2.90 GHz clock, and 8 GB of RAM to do the implementation. Simulations were conducted with three nodes (node 1, node 2, and node 3) on Ethereum. The nodes were initialized using Ethereum Go client 'geth'. Ethereum accounts were also created for the nodes so that they could interact with each other through the smart contracts. The smart contracts were compiled on the Remix IDE and deployed on node 1. After that, node 1 broadcast the address of the deployed smart contract to the entire network so that the modes can interact with each other using the smart contract. These nodes are in charge of registering the users' DIDs on the blockchain. The function $addDID()$ adds a new DID to the system. The new DID and its public key are the inputs of this function. When the DID is added, a new block is created and appended to the Ethereum blockchain that contains details of the transaction of this operation. The function $checkDIDExists()$, which takes a DID as input, checks if that particular DID exists. The function $getPublicKey()$ takes a DID as input. If $did1$ is the input of $getPublicKey()$, it returns the public key associated with $did1$ which can be used to verify the signatures signed with the private key of $did1$.

V. Discussion

In this section, first, we present the security features of the proposed protocol. Then, we analyze the performance of the proposed protocol by estimating its computation cost. Finally, we compare the proposed protocol with similar schemes based on security features and computation cost.

A. Security Features

The proposed protocol achieves the following security features:

- **Secure admittance of vehicles into a platoon:** Vehicles that are registered with the TA receive VCs signed by the TA. Before admitting a vehicle into a platoon, the leader and the joining vehicle verify each other's VC and mutually authenticate each other. Thus, the protocol ensures secure admittance of vehicles into a platoon.
- **Protection from Eavesdropping Attacks**: The parameter VC_{Li} in M_2 is encrypted with the public key, Vi_{pu}, of V_i. Even if an adversary listens to M_2, he/she cannot obtain VC_{Li} as he/she does not have the corresponding private key, Vi_{pr}, to decrypt it. Similarly, the parameter VC_{Vi} in M_3 is encrypted with the public key, Li_{pu}, of L_i. The adversary cannot decrypt VC_{Vi} as he/she does not have Li_{pr}. Thus, the proposed protocol ensures protection from eavesdropping attacks.
- **Protection Against Replay Attacks**: Replay attacks involve the adversary capturing the exchanged messages between the joining vehicle and the leader, and replaying them later to get admission into the platoon. The parameter $Enc[VC_{Li}]$ in M_2= $\{ID_{Li}, Enc[VC_{Li}]\}$ contains a random number R_1 which is different in each session. Similarly, the parameter $Enc[VC_{Vi}]$ in M_3= $\{Enc[VC_{Vi}]\}$ contains a random number R_2 which is different in each session. Hence, the adversary cannot replay M_2 and M_3 to launch a replay attack.
- **Protection Against Impersonation Attack**: The adversary does not have access to VC_{Vi}. Hence, the adversary cannot compose the messages M_3= $\{Enc[VC_{Vi}]\}$ to

impersonate a legitimate vehicle to get admission into the platoon. Thus, the proposed protocol provides protection against impersonation attacks.

- **Prevention of ghost attacks:** The proposed protocol ensures that only legitimate, registered vehicles get admission into a platoon. As a result, the proposed protocol eliminates the scenario where an adversary impersonates a ghost vehicle that does not exist in the platoon. Thus, the adversary cannot do platoon ghost attacks and subsequent attacks on control commands after joining the platoon as a ghost vehicle.
- **Anonymity:** Only the TA knows the real identity, VID_{Vi}, of a vehicle. The vehicle's DID is used during platoon admission instead of its real identity, thereby maintaining the anonymity of the vehicle. Thus, the protocol keeps the vehicle anonymous while admitting it into the platoon.
- **Privacy:** Even if an adversary listens to the exchanged messages during platoon admission, he/she cannot link them to any specific vehicle due to the anonymity property discussed above. Hence, the platoon admission data is not available to an adversary, thereby preserving the privacy of vehicles.

B. Performance Analysis

Next, we analyze the computation cost incurred during the execution of the protocol while admitting a vehicle into the platoon. Let the time taken by signature verification and encryption/decryption operations be t_{verify} and t_{ed}, respectively. We use the Elliptic Curve Digital Signature Algorithm (ECDSA) for the signature generation, signature verification, encryption, and decryption operations. The time taken by the concatenation operation is negligible. From the experiments, t_{verify} and t_{ed} are 0.34 ms and 0.16 ms, respectively. Hence, the total computation cost during the platoon admission phase is 4 t_{ed} + 2 t_{verify} = 1.32 ms.

C. Comparison With Similar Schemes

Next, we compare the proposed protocol's security features and computation cost with that of similar schemes.

Comparison of Security Features: The proposed protocol enables secure admittance of vehicles into a platoon preventing ghost attacks. Also, the proposed protocol provides protection against replay, eavesdropping, and impersonation attacks and offers anonymity and privacy for vehicles. In the proposed scheme, vehicles create their own identities using DIDs without depending on any third party. This results in a high level of privacy. None of the other schemes has this feature. The schemes in [1], [10], [13] are based on physical context verification for platoon admission. Though these schemes ensure secure admittance of vehicles into a platoon, they require a minimum of 10 seconds and sometimes more than a minute to transmit a verification key [14]. Also, the line of sight between the two vehicles should not be interrupted within this time frame [14]. On the contrary, the proposed protocol does not require the vehicles to be in the line of sight. It will enable secure admittance even if the line of sight is obscured, which is quite common in road traffic. Further, the schemes in [1], [10], [14] do not maintain the anonymity and privacy of vehicles. A summary of the comparison of the security features is given in Table II.

TABLE II: Comparison Based On Security Features

Scheme	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**
Han et al. [1]	Y	N	Y	Y	Y	N	N	N
Dickey et al. [10]	Y	N	Y	Y	Y	N	N	N
Junaidi et al. [11]	Y	N	Y	Y	Y	Y	Y	Y
Lai et al. [12]	Y	N	Y	Y	Y	Y	Y	Y
Vaas et al. [13]	Y	N	Y	Y	Y	Y	Y	N
Plattner et al. [14]	Y	N	Y	Y	Y	N	N	Y
Proposed Protocol	Y	Y	Y	Y	Y	Y	Y	Y
S1: Secure admittance;								
S2: Vehicles create identities without depending on any third party;								
S3: Replay attack protection;								
S4: Eavesdropping attack protection;								
S5: Protection against impersonation attacks; **S6:** Anonymity;								
S7: Privacy; **S8:** Does not require line of sight;								

Comparison of Computation Cost: As mentioned in Section V. B, the total computation cost of the proposed protocol during the platoon admission phase is 1.32 ms. The minimum time taken by the scheme in [10] is 10 s (which is the verification time for 1 challenge) from the experiments given in [10]. The time taken by the scheme in [11] is approximately 4 t_{sign} + 4 t_{verify} + 6 t_{mul} + 2 t_h where t_{sign}, t_{mul}, and t_h represent the time taken by signature generation, scalar multiplication, and hash operations, respectively. From the experiments, t_h = 0.23 ms, t_{sign} = 0.27 ms, and t_{mul} = 1.12 ms. Hence, the time taken by the scheme in [11] is 9.62 ms. The computation cost of the scheme in [12] is 12 t_h = 2.76 ms. We have plotted the computation costs of cryptographic technique-based schemes [11] and [12] in Figure 2. Since the schemes based on physical context verification take longer than those based on cryptographic techniques, we have not plotted the computation costs of schemes based on physical context verification. From the analysis and figures, it can be noted that the computation cost of the proposed protocol is less than that of other existing schemes for platoon admission.

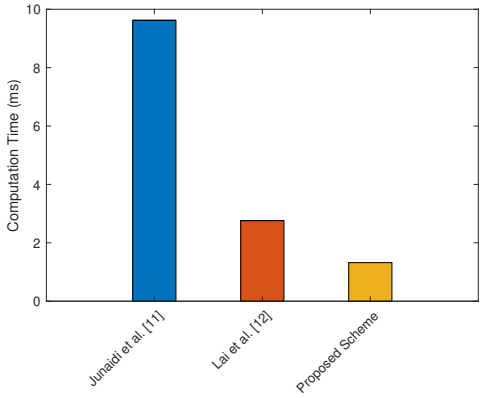

Fig. 2: Computation cost during platoon admission.

VI. Conclusion

In this paper, we proposed a mutual authentication protocol to securely admit vehicles into a platoon. The proposed protocol ensures that only registered vehicles with valid verifiable credentials are given admission into a platoon. Thus, the protocol prevents ghost attacks that pose a safety threat to vehicles in platoons. We provided a proof of concept implementation of the blockchain network for the proposed scheme which showed the practicality of the proposed protocol. We compared the computation cost of the proposed protocol with that of two other protocols for platoon admission. The comparison showed that the computation cost of the proposed protocol is less than that of other protocols. Thus, it is clear from our analysis that the proposed protocol is practical and provides secure admission of vehicles into a platoon with reasonable computation cost.

VII. Acknowledgement

This research is supported by the National Research Foundation, Singapore and Infocomm Media Development Authority under its Future Communications Research Development Programme, under grant FCP-NUS-RG-2022-019.

References

[1] J. Han, M. Harishankar, X. Wang, A. J. Chung, and P. Tague, "Convoy: Physical context verification for vehicle platoon admission," ser. HotMobile '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 73–78. [Online]. Available: https://doi.org/10.1145/3032970.3032987

[2] M. Amoozadeh, H. Deng, C.-N. Chuah, H. M. Zhang, and D. Ghosal, "Platoon management with cooperative adaptive cruise control enabled by vanet," *Vehicular Communications*, vol. 2, no. 2, pp. 110–123, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214209615000145

[3] S. J. Taylor, F. Ahmad, H. N. Nguyen, and S. A. Shaikh, "Vehicular platoon communication: Architecture, security threats and open challenges," *Sensors*, vol. 23, no. 1, p. 134, 2022.

[4] S. Ellwanger and E. Wohlfarth, "Truck platooning application," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 966–971.

[5] A. Ghosal, S. U. Sagong, S. Halder, K. Sahabandu, M. Conti, R. Poovendran, and L. Bushnell, "Truck platoon security: State-of-the-art and road ahead," *Computer Networks*, vol. 185, p. 107658, 2021.

[6] B. Ko and S. H. Son, "An approach to detecting malicious information attacks for platoon safety," *IEEE Access*, vol. 9, pp. 101 289–101 299, 2021.

[7] H. Hu, R. Lu, C. Huang, and Z. Zhang, "Tripsense: A trust-based vehicular platoon crowdsensing scheme with privacy preservation in vanets," *Sensors*, vol. 16, no. 6, p. 803, 2016.

[8] "Decentralized Identifiers (DIDs)," Online, https://www.w3.org/TR/did-core/, [Accessed: May 2024].

[9] "Verifiable Credentials Data Model 1.0," Online, https://www.w3.org/TR/vc-data-model/, [Accessed: May 2024].

[10] C. Dickey, C. Smith, Q. Johnson, J. Li, Z. Xu, L. Lazos, and M. Li, "Wiggle: Physical challenge-response verification of vehicle platooning," in *2023 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2023, pp. 54–60.

[11] D. R. Junaidi, M. Ma, and R. Su, "Secure vehicular platoon management against sybil attacks," *Sensors*, vol. 22, no. 22, p. 9000, 2022.

[12] C. Lai, R. Lu, and D. Zheng, "Spgs: a secure and privacy-preserving group setup framework for platoon-based vehicular cyber-physical systems," *Security and Communication Networks*, vol. 9, no. 16, pp. 3854–3867, 2016.

[13] C. Vaas, M. Juuti, N. Asokan, and I. Martinovic, "Get in line: Ongoing co-presence verification of a vehicle formation based on driving trajectories," in *2018 IEEE European Symposium on Security and Privacy (EuroSP)*, 2018, pp. 199–213.

[14] M. Plattner, E. Sonnleitner, and G. Ostermayer, "A security protocol for vehicle platoon verification using optical camera communications," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[15] "Welcome to Ethereum," Online,https://ethereum.org/en/, [Accessed: Apr 2024].

[16] "REMIX," Online,https://remix.ethereum.org/, [Accessed: Apr 2024].