A Blockchain-Based Security Scheme for Vehicular Ad Hoc Networks in Smart Cities

Xue Jun Li
Department of EEE
Auckland University of Technology
Auckland, New Zealand
xuejun.li@aut.ac.nz

Maode Ma
College of Engineering
Qatar University
Qatar
mamaode@qu.edu.qa

Yong Xing Yong
School of EEE
Nanyang Technological University
Singapore
e160206@e.ntu.edu.sg

***Abstract*— The development of Vehicular Ad Hoc Networks (VANET) has brought many advantages to facilitate the deployment of the Intelligent Transportation System (ITS). However, without proper protection, VANETs can be vulnerable to severe cyber-attacks. This paper explores the threats to the VANETs and proposes a security scheme for VANETs with a Blockchain (VNB). Furthermore, the proposed VNB with Ethereum was developed. With a graphical user interface, experiments were conducted. For ad hoc communications, a vehicle can randomly select another vehicle, and VNB will authenticate the selected vehicle with the Blockchain and Trusted Authority (TA). Preliminary test results successfully proved that Blockchain can be the key technology to mitigate the security threats to VANETs.**

Keywords—vehicular ad hoc networks, blockchain, security

I. INTRODUCTION

Vehicular ad hoc networks (VANETs) play an important role in the Intelligent Transportation System (ITS). In 1999, the US Federal Communication Commission (FCC) allocated 75 MHz in the spectrum of 5.850-5.925 GHz for Dedicated Short Range Communications (DSRC) for VANETs. The communication protocol works based on Wireless Access for Vehicular Environments (WAVE) also known as IEEE 802.11p [1-3]. As ITS involves high-speed moving vehicles and requires a simple mechanism for communication, VANETs allow an arbitrary vehicle to communicate with nearby vehicles and infrastructure to disseminate real-time traffic messages.

Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications are enabled in a VANET. To achieve that, an On-Board Unit (OBU) is equipped in every vehicle and Road Side Unit (RSUs) are deployed along a road [2]. In addition, a Trusted Authority (TA) and application servers are installed. The TA is responsible for identity management and registration [4]. V2V communication enables broadcast information like vehicle speed, turning direction, etc. With the aid of a RSU, a vehicle gets safety and traffic information from the TA through V2I communication.

A VANET is designed mainly to focus on crash prevention, safety and traffic control, which is considered as one of the most prominent technologies to provide efficiency and safety for ITS [5]. Experts predicted that it can prevent up to 80% of the accidents involving non-impaired drivers. However, a VANET can be vulnerable to the following adversaries items [4]: (1) selfish drivers, who may deliberately inject false traffic information in the system to maximize their benefits; (2) eavesdroppers, who spy on other drivers to deduce their behaviour or travelling pattern for future information attacks; and (3) malicious attackers, who may use sophisticated tools to disrupt the communication in a VANET for the certain financial gain.

Apart from the abovementioned adversarial behaviours, a VANET is also vulnerable to the following types of cyber attacks: (1) Denial of Service (DoS) attack, which will make the VANET unavailable to vehicles. This can be done by jamming the channel with dummy messages, which prevents a legitimate user from sending critical safety messages. Eventually, it may lead to a system crash. (b) Privacy Violation: To prevent impersonation, the identity of the sender of a message is shown, which would reveal the driver's identity, location, actions and preference. This violates their privacy. (c) Message Alternation: This attack is done where messages are altered. For example, when a driver decides to send a hazard message, it can be changed from hazard to non-hazard message, which may lead to devastating consequences.

To address those security issues, a secure VANET requires the following features: (1) authentication to verify the identity of a user; (2) message integrity to prevent any form of tampering; (3) message non-repudiation to track down the cause of an accident; (4) entity authentication to ensure the sender of a message is active in the network; (5) access control to prevent unauthorised information access; (6) message/user information confidentiality to protect privacy; (8) availability to ensure system resilience; (9) real-time guarantee to provide collision avoidance and valid information.

These requirements motivate us to study the possibility of applying Blockchain in VANETs. The rest of this paper is organised as follows. Section II briefly reviews Blockchain and existing works on the security schemes for VANETs. Section III presents the proposed security scheme in VANETs with a Blockchain, followed by its implementation in computer simulations in Section IV. Section V presents its simulation results with discussions and Section VI concludes the paper.

II. RELATED WORK

A. Blockchain

A blockchain [6] is a decentralized, shared, distributed and append-only ledger, which is used to facilitate the process of recording transactions and tracking assets. Its name originated from the fact that it stores transaction data in blocks that are linked together to form a chain. Due to its versatility, a blockchain can be used in many industries such as financial services, insurance, government, supply chain management, healthcare, and the Internet of Things (IoT). The most popular application of blockchain is the Bitcoin.

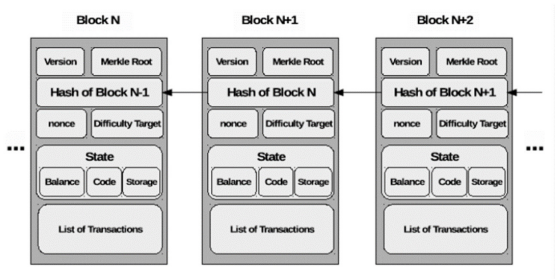


Fig. 1. The Block Structure in a Blockchain

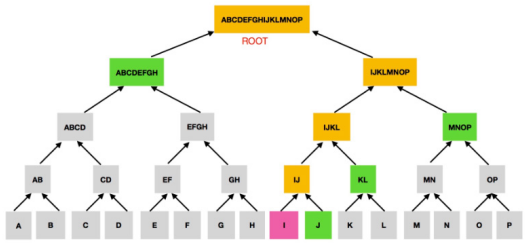


Fig. 2. A Merkle Tree Example

It is necessary to briefly revisit the terminologies in the blockchain.

Block Structure— A block in a blockchain consists of a block header and block body [7] which contains a list of

transactions as shown in Fig. 1. A block header contains metadata including version, previous bock hash, timestamp, block size, etc. Fig. 2 illustrates a Merkle tree. Every transaction in the block goes through a cryptographic hash and form an individual digest. These digests form the leaf nodes in a Merkle tree. The leaf nodes are then paired up to go through another cryptographic hash function to form a new node. The process continues until one node is left, which is called the Merkle root.

Consensus Mechanism Algorithm—One of the ways a blockchain builds trust is through the consensus of the participants in the distributed network. A consensus mechanism algorithm is to determine if a transaction should be appended into a blockchain based on the amount of consensus made by the participants. If most of the participants deem that the transaction is valid, the transaction will be appended. Otherwise, consensus cannot be reached, and the transaction will be discarded.

Noteworthily, successfully appended transactions cannot be removed from a blockchain. Therefore, it is important to understand the requirements in the consensus mechanism algorithms: (1) Agreement Seeking: For a normal transaction to get appended into the blockchain requires a minimum 51% agreement. (2) Collaborative: All participants share the same goal to ensure that the transactions in the blockchain are accurate. (3) Cooperative: All participants should work together to achieve the best result. (4) Egalitarian: Each vote from a participant has equal weight. (5) Inclusive: Every participant participating in the consensus process are important. (6) Participatory: The consensus mechanism algorithm should be designed to enable every participant to actively participate in the consensus process.

Several consensus algorithms are available for a blockchain to achieve distributed consensus [8]: Proof of Work (PoW) [9], Proof of Stake (PoS) [10], Delegated Proof of Stake (DPoS), Leased Proof of Stake (LPoS), Proof of Importance (PoI) and Practical Byzantine Fault Tolerance (PBFT) [11]. The advantage of PoW lies in its simplicity and security to find the correct nonce. However, it consumes a tremendous amount of energy to find the golden nonce. In addition, PoW can easily become centralised if miners form mining pools to achieve the 51% consensus. For PoS, it is energy efficient and prevents centralisation. Additionally, it is cheaper to implement. However, PoS is vulnerable to double spending attacks. In this paper, we employ PoW and PoS.

In general, there are three types of the blockchains, namely public blockchain, consortium blockchain and private blockchain [12].

B. Security Schemes based on Blockchain in VANETs

By prediction, the global market for VANETs will reach 33, 374 million USD by 2050 [13]. One salient feature of a VANET is its flexibility, which allows a vehicle to authenticate its neighbours through the TA before communicating with each other. However, this centralised process makes VANETs vulnerable to the DoS attacks. To address the security issues in VANETs, blockchain was proposed to be applied as a trusted platform. Furthermore, the smart contract features can be implemented to enable the automation process.

In [14], an analysis on VANETs stated that the most suitable network architecture for VANETs is peer-to-peer (P2P). With P2P, VANETs enjoy benefits like (1) high proximity, where connection to nearby peers is more reliable than distant parent node; (2) low latency, which is minimised through a DSRC protocol; (3) decentralization, under which no particular entity owns the network; (4) fault-tolerance, which makes vehicular peers immune to network outrage. A VANET also benefits from a super-peer architecture where super-peer (RSU) would provide services beyond what regular peers (vehicles) could. The super-peer architecture increases the throughput, domain-specific functionality, reliable inter-RSU connections and facilitated clustering, which allows focused local interactions while maintaining a broader connection between distinct clusters.

Fig. 3. A 3-layered Approach to Extract VANETs' Messages [14]

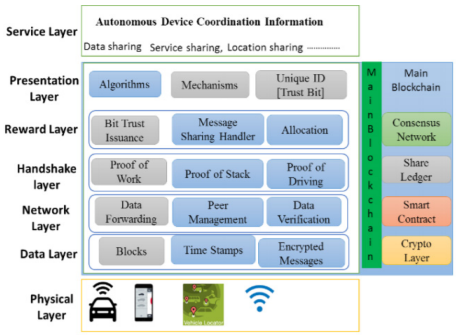


Fig. 4. A 7-layered Approach to Extract VANETs' Messages [15]

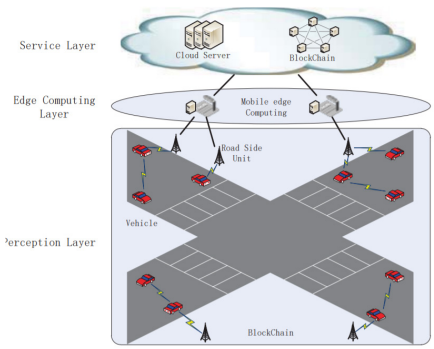


Fig. 5. A 3-layered Approach to Extract VANETs' Messages [16]

Some works have been done to incorporate a VANET with a blockchain to fully utilise the advantages of both technologies. In [14], a blockchain was used to enable P2P transactions with message integrity and security. This component constitutes four layers, namely the Application Layer, Blockchain Layer, Network Layer and the Physical Layer. All the layers are logically independent of each other, while each layer encapsulates the layer above and together will form a single message (see Fig. 3).

In [15], a security mechanism for VANETs with P2P communication was proposed to provide a secure and trusted environment with immutable database and ubiquitous data access in a secure way. Fig. 4 shows the framework of the model. It uses a blockchain for data sharing among intelligent vehicles by an Intelligent Vehicle-Trust Point (IV-TP), which enables the creation of crypto unique ID, self-executing digital contracts and details of the intelligent vehicle controlled over the blockchain cloud.

As shown in Fig. 5, a security architecture of VANETs based on a blockchain and Mobile Edge Computing (MEC) was proposed in [16], which included three layers: perceptual layer, edge computing layer, and service layer.

C. *Identified Research Gap*

As abovementioned, a blockchain was proposed to enhance message integrity in VANETs. A blockchain layer was proposed in [14], which is responsible for the hashing function. In [15], a blockchain was used to share data among intelligent vehicles by the IV-TP. Lastly, in [16], a blockchain was applied to store crucial messages. However, none of them mentioned much about the vehicle authentication procedure in VANETs, which plays an important role in VANETs. Authentication allows the system to differentiate the true users from the fake users. For example, without authentication, an attacker can hack into an anonymous account and communicate with the nearby vehicles by sending malicious information, which can lead to serious consequences.

Therefore, we aim to design and develop a VANET system with a blockchain to authenticate legitimate vehicles for the sake of security. Once a vehicle is authenticated, it will be able to gain access to the system. Otherwise, it cannot communicate with any vehicle in the VANETs.

III. Proposed Blockchain-Based Security Scheme

The proposed security scheme consists of three layers, namely the Physical Layer, the Communication Layer and the Service Layer shown in Figure 6.

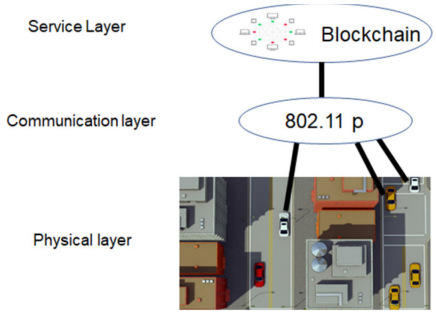


Fig. 6. Proposed Security Scheme

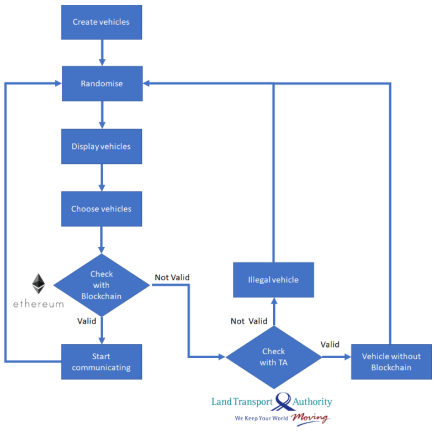


Fig. 7. Flowchart of Operation Procedure of the System

Fig. 7 shows the procedure of the proposed security scheme in VANETs with a blockchain (VNB). First, the TA, which is Land Transport Authority in Singapore, will create a set of legitimate vehicles. Next, the VNB will randomise a set of vehicles from the TA list and a set of illegitimate vehicles. The selected vehicles will then be displayed on the vehicle screen. A VNB user can then choose another vehicle for communication. Upon selecting a vehicle, the VNB will check that particular vehicle account by the blockchain, which is provided by Ethereum. If that vehicle account does exist in the blockchain, the VNB will then allow the user to communicate with that vehicle. Otherwise, the VNB will prohibit the user from communication with the one selected.

The vehicle account not in the blockchain does not pose an immediate security threat, which can be verified with the TA. If the vehicle account exists in the TA list, the VNB will inform the user that the chosen vehicle is legitimate but without a blockchain account. Otherwise, the VNB will then alert the user that the chosen vehicle is an *unknown* vehicle with potential security threats.

Upon completing one cycle to identify a vehicle as legitimate or illegitimate, the VNB will randomise the vehicle account pool and the process repeats itself. The rationale behind this is to simulate the fact that vehicles in the VANETs are always moving and the randomisation is to refresh the VNB to ensure that the vehicles displayed on the screen are real-time.

IV. Implementation

For the implementation of the proposed VBN, four main components are reqiured, including Python, Virtual Land Transport Authority as the TA, Ethereum and Graphical User Interface, which are discussed as follows

Python is the main programming language used to develop the proposed VNB due to its simplicity and its large range of libraries. The former allows developers to understand the code easily and make further improvements to the VNB in the future. The latter enables developers to import certain applications to develop the VNB. Python libraries are used to import Ethereum and Graphical User Interface.

Virtual LTA acts as the TA in the system. It will generate and store 100 valid car license plates in a dictionary. The car license plate format will begin with the letter "S" followed by two alphabetic letters, four numerical digits and ending with an alphabetic letter (e.g SMK1234E). This is the legitimate car license plate format used in Singapore. Fig. 8 shows the flow chart of the TA function.

With a valid car license plate, the car owner can use it to create a blockchain account. The TA is used to verify if the vehicle is a valid vehicle or an attacker attempting to hack the VNB. If the vehicle does not own a blockchain account, the VNB will verify the car plate number with the TA. If the plate number exists, it gives a legitimate vehicle. However, it is unable to start communication because the VNB works based on the blockchain, so the user has to create a blockchain account before communicating with another vehicle. If the plate number does not exist, it could be an attacker creating an anonymous vehicle account trying to disrupt the VNB or steal some private vehicle information.

The Ethereum [17] is a decentralized platform that allows developers to prototype blockchain applications easily by building a blockchain with a built-in Turing-complete programming language (Solidity) [18]. Solidity allows developers to write smart contracts and develop decentralized applications. The Ethereum is designed with the features of simplicity, universality, modularity, agility, non-discrimination and non-censorship [18]. Simplicity keeps protocols as simple as possible. Universality allows developers to construct any smart contract or transaction type that can be mathematically defined. The Ethereum protocol is modular and separable. When a modification is done to a protocol, the application would still be able to function. Agility allows the Ethereum protocol to be changeable. Non-discrimination and non-censorship mean that the Ethereum is designed to control harm and not attempt to oppose specific undesirable applications. Developers can run their script endless time as long as they are willing to pay the computation transaction fees.

We use Web3.py [19] to program the blockchain. It is a Python library used to interact with the Ethereum. In the VNB, the blockchain used is a private blockchain because the TA is still required to act as a middle man. To use the blockchain, the VNB will need to be connected to the network. However, the Ethereum will not be connected to the main network that incures costs. Instead, it will be connected remotely. Fig. 9 shows the flow chart on the way to connect to a blockchain remotely. First, the path must be defined before starting a connection. Once the connection is set up, it will return true and the development of the blockchain can be started.

With the connection setup, developers can use functions to create accounts, send transactions, search for accounts or transactions etc. The functions used are the *create account*

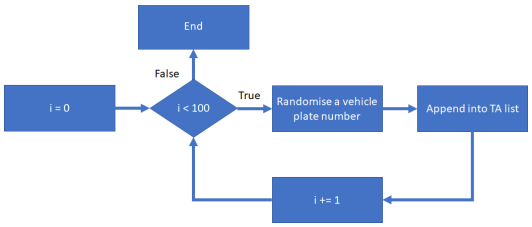


Fig. 8. Flowchart for the TA

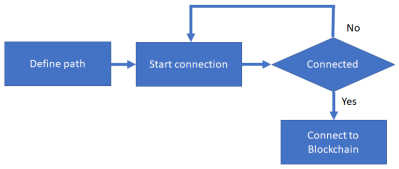


Fig. 9. Flowchart for Blockchain Connection

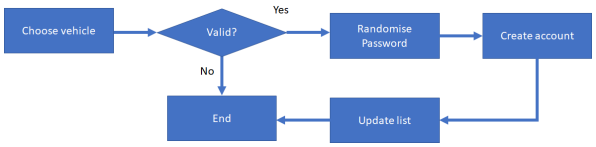


Fig. 10. Flowchart for Blockchain Account Creation

for the user with a valid car plate license and the *lookup* to check if an account exists in the blockchain. For the operation, a password is needed to create a new account. The VNB will randomise an eight-character password consisting of digits, uppercase and lowercase alphabets (e.g., *NSIHJnA9*). Once the account is created, a hash will be given as the account username (e.g., *0xac39872293fAc45C5b1EF2714e7c03bDc2F937ED*). These two items will then be stored. Fig. 10 shows the flow of the creation of an account. Fig. 11 shows the flowchart of the lookup function to check if a vehicle account exists in the blockchain. The VNB will return true if it exists and false otherwise. Hence, the input parameter is the account number.

The GUI is used to provide the user with visual interactions with the VNB. Noteworthily, the VNB shall be running in the background and the vehicle OBU will decide which vehicle to communicate with. To create the GUI, frames are used, inside which developers can create labels, textbox, buttons etc, which they want to display for the users to see and interact with.

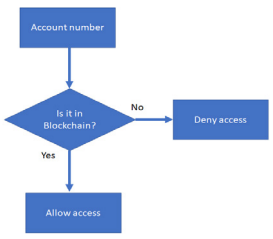


Fig. 11. Flowchart of the Lookup Function

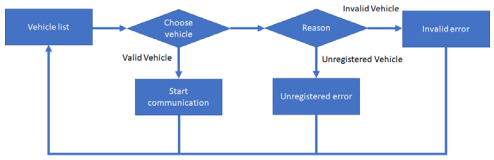


Fig. 12. Flowchart of System GUI

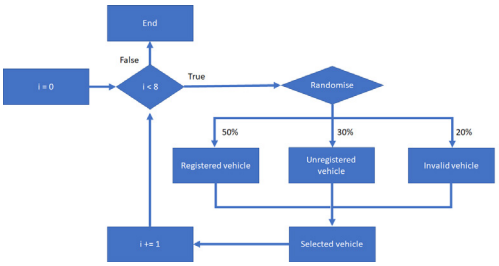


Fig. 13. Flowchart of Vehicle Randomisation

Fig. 12 shows the flowchart of the GUI, which will display the list of vehicles nearby, from which the user will act as the vehicle OBU to choose a vehicle to interact with. Upon clicking the vehicle to communicate with, the GUI will run the authentication process in the background to authenticate the selected vehicle with three possible results: (1) First, when the vehicle has a blockchain account, the authentication will be successful. Then, the user will be brought to another frame to start communicating with the vehicle. (2) Second, when the vehicle does not have a blockchain account but can be verified with the TA, the authentication will fail, the user will be brought to a frame that shows a caution alert to inform the user of the reason for the authentication failure. Then, it will bring the user back to the frame to choose another vehicle to communicate with after 2 seconds. (3) Finally, when the vehicle does not own a blockchain account and is not verified by the TA, the authentication will fail, and the action is similar to the second case. However, the difference is the level of risk. A vehicle without a blockchain account and not verified by the TA might be an attacker trying to hack the VNB. Hence, the warning to the user will be an emergent instead of a caution one.

To simulate the scenario that most vehicles are with the blockchain and a few vehicles are suspicious, a probability threshold is calculated. As shown in Fig. 13, the probability of encountering legitimate vehicles with the blockchain, a legitimate vehicle without the blockchain, and a suspicious vehicle is 50%, 30% and 20%, respectively.

V. Simulation Results and Discussions

The proposed VNB is implemented through the simulation with the GUI. To run the VNB, users are required to download an integrated development environment for Python and Geth, which is the Official Go Implementation of Ethereum protocol. With Geth, users can interact with the Ethereum. With both components, the VNB can be tested.

A. Simulation Results

Fig. 14 illustrates the welcome frame for the VNB to start. Fig. 15 shows the car plate list generated by the TA. All car plates are assigned a value of 0. Then, a random amount of the car plate will be used to create a blockchain account. After it, the username and the password are assigned.

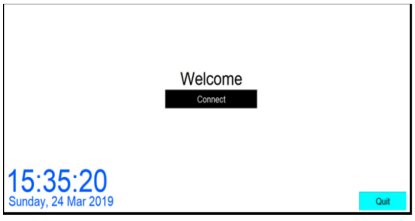


Fig. 14. Welcome Frame

{'SAR4936V': 0, 'SZT9366D': 0, 'SCG9351X': 0, 'SQO5504V': 0, 'SOS8954K': 0,
'SCG9208J': 0, 'SZG4758V': 0, 'SGI65490': 0, 'SLV9073I': 0, 'SSF5167W': 0,
'SFB2129T': 0, 'SZE4492R': 0, 'SVA8230V': 0, 'SNA3663W': 0, 'SVB7746R': 0,
'SIN2129T': 0, 'SXZ0159B': 0, 'SCQ7116F': 0, 'SUM2915Y': 0, 'SOP5892S': 0,
'SOK5804Z': 0, 'SWT4106Z': 0, 'SAT9044I': 0, 'SAF8916X': 0, 'SNQ1238C': 0,
'SIL2089C': 0, 'SJF6965X': 0, 'SYJ4244R': 0, 'SIY9839N': 0, 'SSN0279N': 0,
'SYC5126U': 0, 'SGW9303D': 0, 'SLK6002R': 0, 'SSG2419I': 0, 'SSP1031L': 0,
'SEQ0319A': 0, 'SHW0842W': 0, 'SQT4447Y': 0, 'SHJ3443A': 0, 'SXT5146J': 0,
'SMY7781W': 0, 'SFT8500N': 0, 'SPV8368G': 0, 'SOT6720M': 0, 'SVI9291U': 0,
'SNF2873B': 0, 'SLT9589D': 0, 'SKN8540D': 0, 'SCB3791E': 0, 'SRN5371Y': 0,
'SOY4091H': 0, 'SHA4356Z': 0, 'SPH5410Y': 0, 'SQG5709R': 0, 'SGS2698C': 0,
'SUR0463L': 0, 'SQT0224B': 0, 'SXZ3594T': 0, 'SEC4934T': 0, 'SBX8097L': 0,
'SIU8274R': 0, 'SNC9628V': 0, 'SOI9466R': 0, 'SDD7627C': 0, 'SBG7266B': 0,
'SFH9198O': 0, 'SEJ1682S': 0, 'SBC5949Q': 0, 'SOT8335Q': 0, 'SYY1578Z': 0,
'SUC9194U': 0, 'SXR5102E': 0, 'SMU6518S': 0, 'SCA8304Y': 0, 'SBJ0184R': 0,
'SHJ0414C': 0, 'SMX5074L': 0, 'SHJ7307C': 0, 'SHA4867G': 0, 'SED4153X': 0,
'SMD0094Q': 0, 'SMP0006Z': 0, 'SPQ9121O': 0, 'SQV0853O': 0, 'SQS9158W': 0,
'SUI8230T': 0, 'STH9417L': 0, 'SZQ7096J': 0, 'SNT6838T': 0, 'SCK3866Y': 0,
'SZH3550N': 0, 'SRW1489M': 0, 'SVR7211N': 0, 'STQ4271X': 0, 'SZR6253V': 0,
'SCZ6375F': 0, 'STA1672A': 0, 'SMB3894I': 0, 'SGG5050X': 0, 'SXL1430X': 0}

Fig. 15. Generated Vehicle List by the TA

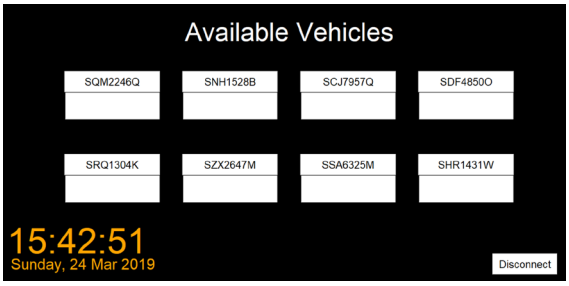
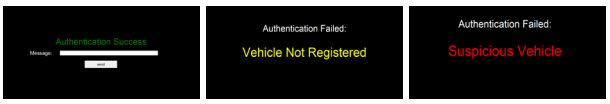


Fig. 16. Nearby Vehicles in the VANET



(a) (b) (c)

Fig. 17. Authentication Results (a) Success (b) Unregistered (c) Suspicious

Once the *Connect* button is clicked, the VNB searches for nearby vehicles. As illustrated in Fig. 16, eight vehicles appear randomly, which could be the vehicles with the blockchain, the vehicles without the blockchain and the *suspicious* vehicles, according to the probability function. Next, the screen shows eight vehicles as buttons and a *Disconnect* button. Under each vehicle button, a chat textbox is used to record the communication between the particular vehicle and the user. To communicate with the desired vehicle, the authentication process will begin after the desired vehicle button is clicked. The *Disconnect* button is used to disconnect the VNB from the VANET.

To communicate with a vehicle by clicking its button, the VNB will proceed to authenticate the vehicle through the blockchain network and the TA. When the authentication is successful, the frame shown in Fig. 17(a) is displayed. The vehicle is of low risk indicated by a green coloured text. Below the result is a textbox, which allows the user to send messages to the vehicle.

If the authentication fails, failure frames will be displayed as shown in Fig. 17(b) and Fig. 17(c). Fig. 17(b) shows the user that the vehicle is of a medium risk with the yellow coloured text. This means that this vehicle the user is trying to communicate to is a valid car but does not have a blockchain account. Fig. 17(c) shows that the vehicle is of high risk as indicated in a red text. The chosen vehicle does not have a blockchain account and is not registered with the TA. It might be an attacker who is trying to disrupt the VNB.

Finally, a vehicle will periodically scan surrounding vehicles after sending messages or authentication failure. As

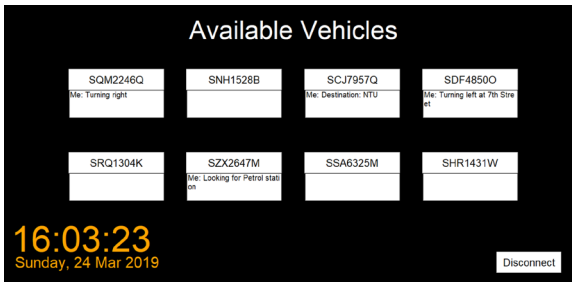


Fig. 18. Communication History

illustrated in Fig. 18, the GUI records those messages exchanged among known neighbour vehicles.

B. Discussions

The proposed VNB can correctly classify the vehicles and only allow communication when the vehicle authentication is successful. However, for practical implementation, the following items should be considered:

Memory Consumption—As the proposed VNB is built remotely, memory space is limited as all the data and files are stored in the remote device. To reduce the impact of this limitation, the VNB only allows the TA to create one hundred vehicle car plates. Furthermore, the blockchain account files will be removed periodically to release the memory space.

Decentralisation—The proposed VNB is built on a partially decentralized network as the VNB still requires a third party, which is the TA. Currently, the TA is not decentralised, which makes the proposed VNB not fully decentralised. Therefore, to create a fully decentralised system, the TA has to be implemented in a decentralised fashion.

Computation Power—Messages exchanged between vehicles are not stored as transactions in the blockchain. This is due to the computation power and the memory space of the device. To mine a transaction, miners are needed. As mentioned earlier, mining requires a large computation power. A normal vehicle is unable to provide such resources. Furthermore, each transaction requires a memory space which shall be provided for future implementation.

VI. Conclusion

A security mechanism was proposed for secure VANETs communications with the use of the blockchain technology. Subsequently, the proposed VANET with the blockhchain provided by Ethereum was developed. The VNB was used to simulate a vehicle OBU, which will scan for nearby vehicles, perform authentication and provide communication for vehicles with the blockchain accounts. In the simulation, the TA is used to store the information of all valid car plate numbers. Simulation results show that the VNB was able to differentiate different types of vehicles correctly. Despite the limitation of the proposed VNB, it demonstrates a promising security and trust scheme for autonomous vehicular networks and intelligent transportation systems in the near future for smart cities.

References

[1] G. Li, M. Ma, C. Liu, and Y. Shu, "A Lightweight Secure VANET-Based Navigation System," in *IEEE GLOBECOM'15*, San Diego, CA, USA, 2015, pp. 1-6.

[2] *ITS Standards Fact Sheets-IEEE 1609 - Family of Standards for Wireless Access in Vehicular Environments (WAVE)*, 2009.

[3] B. Li, M. Mirhashemi, X. Laurent, and J. Gao, "Wireless Access for Vehicular Environments," 2011.

[4] A.-S. K. Pathan, *Security of Self-Organizing Networks: MANET, WSN, WMN, VANET*. 2019.

[5] L. G. Giordano and L. Reggiani, *Vehicular Technologies: Deployment and Applications*. Intech Open, 2013.

[6] M. Gupta, *Blockchain For Dummies®, IBM Limited Edition*. John Wiley & Sons, Inc., 2017.

[7] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Generation Computer Systems,* vol. 82, pp. 395-411, 1 May 2018.

[8] FinTechFans. (2017). *Popular Blockchain Consensus Mechanisms Used By Cryptocurrencies Explained*. Available: https://fintechfans.com/blog/popular-blockchain-consensus-mechanisms-used-by-cryptocurrencies-explained

[9] C. Dwork and M. Naor, "Pricing via Processing or Combatting Junk Mail," Berlin, Heidelberg, 1993, pp. 139-147: Springer Berlin Heidelberg.

[10] S. King and S. Nadal, "PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake," 19 August 2012.

[11] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A Survey of Distributed Consensus Protocols for Blockchain Networks," *IEEE Communications Surveys & Tutorials,* vol. 22, no. 2, pp. 1432-1465, 2020.

[12] C. Thompson. (2016, 9 August). *The difference between a Private, Public & Consortium Blockchain*. Available: https://www.linkedin.com/pulse/difference-between-private-public-consortium-collin-thompson

[13] Statista. (2015). *Global automotive V2X market size between 2018 and 2025*. Available: https://www.statista.com/statistics/565790/global-market-size-for-v2x-in-vehicles/

[14] R. Barber, "Autonomous Vehicle Communication using Blockchain," the Sally McDonnell Barksdale Honors College, The University of Mississippi, 2018.

[15] M. Singh and S. Kim, "Blockchain Based Intelligent Vehicle Data Sharing Framework," *arXiv,* 2017.

[16] X. Zhang, R. Li, and B. Cui, "A security architecture of VANET based on blockchain and mobile edge computing," in *1st IEEE International Conference on Hot Informaiton-Centric Networking (HotICN'18)*, Shenzhen, China, 2018, pp. 258-259.

[17] Ethereum Project. (2021). *Ethereum Project*. Available: https://ethereum.org/en/

[18] V. Buterin. (2014). *A Next-Generation Smart Contract and Decentralized Application Platform*. Available: https://translatewhitepaper.com/wp-content/uploads/2021/04/EthereumOrijinal-ETH-English.pdf

[19] P. Merriam and J. Carver. (2018). *Web3.py*. Available: https://web3py.readthedocs.io/en/stable/