2021 IEEE Globecom Workshops (GC Wkshps) | 978-1-6654-2390-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/GCWkshps52748.2021.9681973

Throughput-Efficient Blockchain for Internet-of-Vehicles

Weiquan Ni[1], Alia Asheralieva[1], Carsten Maple[2], Md Monjurul Karim[1], Dusit Niyato[3] and Qiang Yan[4]
[1]*Department of Computer Science and Engineering*, *Southern University of Science and Technology*, Shenzhen, China,
e-mail:{weiquanni17@163.com, aasheralieva@gmail.com, karim@mail.sustech.edu.cn}
[2]*Warwick Manufacturing Group*, *University of Warwick*, Coventry, UK, e-mail:{CM@warwick.ac.uk}
[3]*School of Computer Science and Engineering*, *Nanyang Technological University*, Singapore, e-mail:{dniyato@ntu.edu.sg}
[4]*WeBank*, China, e-mail:{qyan@webank.com}

***Abstract*—Internet-of-Vehicle (IoV) is empowering smart vehicles with data collection and sharing capabilities, and blockchains have been introduced to manage the IoV data due to many advantages, including decentralization, security, reliability, and scalability. Nevertheless, existing IoV blockchain models suffer from poor security against collusion attacks instigated by malicious blockchain miners typically represented by roadside units (RSUs). To address this problem, additional block verifiers, e.g., vehicles, can be recruited during block verification, which enhances security but also can lead to the reduced throughput. Therefore, in this paper, we propose a resource management scheme for IoV blockchains to enhance the system security while maximizing the throughput by optimizing contributed computing resources from RSUs and recruited vehicles. We show that the optimal strategies of RSUs and vehicles can be found through the Karush-Kuhn-Tucker (KKT) conditions and verify (using simulations) that our scheme achieves the higher throughput with enhanced security compared to the existing IoV blockchains.**

***Index Terms*—Blockchain, Internet of Vehicles, throughput, resource management, security**

I. Introduction

IoV facilitates interconnections and interactions among smart vehicles, including data exchange and storage. Traditional centralized approaches to manage data collection and sharing in IoV have many limitations, such as the single point of failure, privacy leakage, low reliability, poor scalability, and absence of transparency. Instead, a distributed ledger - blockchain, has been recently proposed to enhance security, reliability and scalability, and preserve privacy of data management in IoV [1]. In blockchains, the data collected and exchanged by the vehicles are converted into blocks and verified by decentralized blockchain miners, i.e. RSUs, based on a predefined consensus algorithm. Due to low throughput of public consensus algorithms, e.g., proof-of-work (PoW) or proof-of-stake (PoS) [2], the IoV blockchains usually adopt consortium blockchain algorithms, such as delegated PoS (DPoS) [3] or practical Byzantine Fault Tolerance (pBFT) [4]. However, such consensus algorithms have a rather poor security - less than 33% of malicious miners can be tolerated. The main reason is that due to the small size of consortium blockchains, miners can easily collude with each other to falsely verify or reject the block [3].

One possible way to reduce the number of collusion attacks instigated by malicious miners is to increase the size of the blockchain systems by introducing additional block verifiers, e.g., recruited from the vehicles in the IoV networks [6], [7]. But, the selection of newly-recruited block verifiers must be done carefully to ensure that it does not have a negative impact on the block verification delay, i.e., time to propagate and verify the new block. In particular, if the computing resources of block verifiers, including RSUs and vehicles, are very low, the block verification delay can increase by a notable margin, which will significantly reduce the throughput. Although prior works have considered the effect of recruitment of edge devices as additional block verifiers on the block verification delay [8], [9], there are currently no studies which analyse the impact of computing resources of RSUs and vehicles on the block verification delay and, consequently, throughput. Therefore, in this paper, we study how to improve the throughput while the vehicles improving the system security.

The main contributions of this paper are as follows:

- We formulate the analytical model of the IoV blockchains where the RSUs acting as miners can recruite proximate vehicles as block verifiers to enhance the system security. We establish the relationships between the computing resources of miners and newly-recruited block verifiers and block verification delay.
- We formulate the optimization problem which the system utility proportional to the throughput is maximized subject to the constraints on the maximal allowed block verification delay, and show that this problem can be transformed to the concave optimization problem.
- Based on the KKT conditions, we derive the optimal strategies of the RSUs and vehicles. Through numerical evaluations, we show that our model can achieve a more ideal tradeoff between throughput and security than the existing IoV blockchains.

II. Enhanced Block Verification for IoV Blockchains

In this section, we describe the proposed block verification process designated to enhance the blockchain security. In the process, the blockchain miners, i.e., RSUs, can recruit the proximate vehicles as additional block verifiers. As shown in Fig. 1, the process comprises five steps:

978-1-6654-2390-8/21/$31.00 ©2021 IEEE

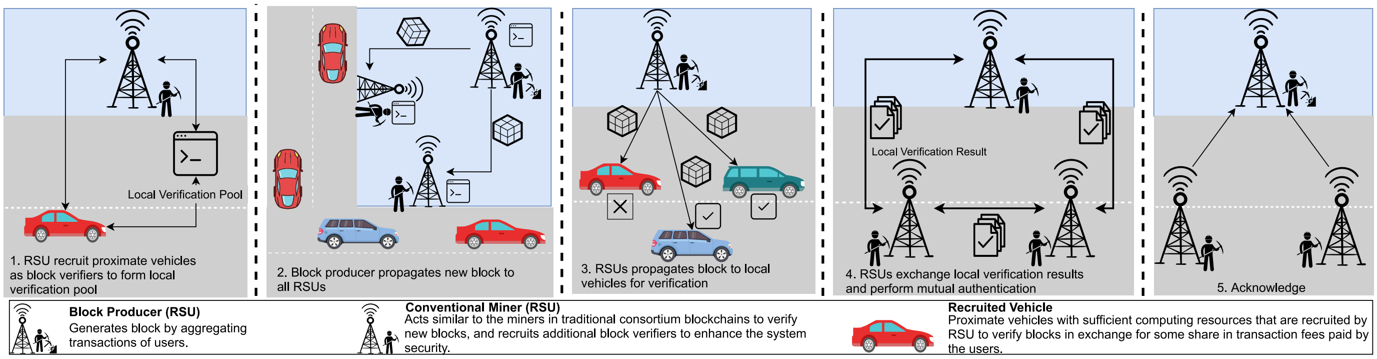

1. RSU recruit proximate vehicles as block verifiers to form local verification pool
Local Verification Pool
2. Block producer propagates new block to all RSUs
3. RSUs propagates block to local vehicles for verification
Local Verification Result
4. RSUs exchange local verification results and perform mutual authentication
5. Acknowledge
Block Producer (RSU)
Generates block by aggregating transactions of users.
Conventional Miner (RSU)
Acts similar to the miners in traditional consortium blockchains to verify new blocks, and recruits additional block verifiers to enhance the system security.
Recruited Vehicle
Proximate vehicles with sufficient computing resources that are recruited by RSU to verify blocks in exchange for some share in transaction fees paid by the users.

Fig. 1. Vehicles-enhanced block verification process for IoV blockchains.

Step 1: By observing the speed of proximate vehicles, each RSU recruits a set of vehicles that will stay in their networks during the coming block verification period, which can be forecasted by the history data of each RSU.

Step 2: The block producer converts the triggered transactions in the network into a new block, and then propagates the block to all RSUs for verification by attaching the block signature and its public key.

Step 3: Each RSU further transmits the valid block to local vehicles after verifying its integrity by inspecting the attached signature and public key. When verifies transactions inside, each block verifier checks identity of involved parties by their public keys and signatures, and whether the account balance of the sender is enough for the transaction by, for example, searching its account in the World State. Ultimately, the vehicles transmit the verification results with their signatures, public keys and the verified block headers to the local RSUs.

Step 4: Upon gathering all local verification results on the block, the local RSUs exchange and mutually authenticate the results with all other RSUs. During the authentication, the RSU firstly checks the verified block header to ensure the same verified block. Then, the validity of the received verification results are verified by their signatures and the public key of corresponding block verifiers. Finally, the RSU records it if the result is the same as its own.

Step 5: After recording more than 2/3 of the verification results in the network, the RSU sends an acknowledgement with the authentication results to the block producer, and append the valid block to the blockchain. The authentication result includes the recording verification results with total number, and information about attacks.

III. Block Verification Delays of Block Verifiers

In this section, we analyse the block verification delays of the RSUs and vehicles based on their computing resource. We consider the IoV blockchain maintained by L RSUs. Based on the common consortium blockchain algorithms, there exists only one block producer generating a new block in each round of blockchain mining. The block is then verified by $N \leq L$ RSUs. The set of RSUs verifying the block is denoted as $\mathbb{I}_R = \{R_i\}_{i=1}^N$, where R_1 specially refers to the block producer. To improve the system security, each RSU R_i recruits a group of vehicles to form a local verification pool $\mathbb{P}_i$, where $\mathcal{P}_i$ is the number of vehicles in pool i.

A. Local Block Verification Delay

Firstly, we analyse delay of block verifiers to verify the new block in the local verification pools. After producing a new block having X transactions, the block producer sends it to all RSUs for verification. Then, each RSU further transmits it to local vehicles with auxiliary verification information of size S_{aux}. After verifying the block, the verification result of size S_{vr} is returned from the vehicle to the local RSU. The local block verification delay of vehicle j in pool i is given by

block verification delay of vehicle

$$T_{ij}^L = \frac{XS_{tr}}{r_{1i}^d} + \frac{XS_{tr}+S_{aux}}{r_{ij}^d} + \frac{\alpha XS_{tr}}{c_{ij}^v} + \frac{S_{vr}}{r_{ji}^u}, \qquad (1)$$

where r_{1i}^d, r_{ij}^d and r_{ji}^u are the data transmission rate between block producer R_1 with RSU R_i, and RSU R_i with local vehicles j, respectively. S_{tr} is the average size of per transaction. α is the average computing resource for verifying per unit size of block, and c_{ij}^v is computing resource contributed from vehicle j in pool i. The local block verification terminates when the RSU aggregates all the local verification results. Thus, the local block verification delay of pool i or RSU R_i is given by

block verification delay of RSU

$$T_i^L = \underset{j\in\mathbb{P}_i}{max}\{T_{ij}^L\}, \text{ s.t. } c_{ij}^v \leq C_{ij}, \forall j \in \mathbb{P}_i, \qquad (2)$$

where the constraint means that vehicle j in pool i cannot contribute more computing resource than its capability C_{ij}. Specifically, the RSUs verify the block while transmitting it to the local vehicles, and therefore ignoring the delay of RSUs to verify the block is feasible.

B. Mutual-Authentication Delay of RSUs

During mutual-authentication, the RSUs exchange and mutually authenticate the local block verification results with other RSUs. While exchanging the verification results, the transmission time from RSU R_i to $R_{i'}$ is given by $T_{ii'}^u = (\mathcal{P}_i + 1)S_{vr}/r_{ii'}^u$, where $(\mathcal{P}_i + 1)$ is the amount of verification results in pool i, transmitted with the data transmission rate $r_{ii'}^u$. We denote the average computing resource to authenticate per verification result as δ. Thus, the time of RSU R_i to authenticate the verification results from $R_{i'}$ is given by

$$T_{ii'}^{ma} = \frac{(\mathcal{P}_{i'}+1)\delta}{c_{ii'}^{a}}, \tag{3}$$

where $c_{ii'}^{a}$ is the computing resource of R_i to authenticate the verification results from $R_{i'}$. Also, the RSUs can simultaneously authenticate verification results from all RSUs. Thus, the total computing resource of RSU R_i for the authentication cannot exceed its capacity C_i, that is, $\sum_{i' \in \mathbb{I}_R \setminus \{i\}} c_{ii'}^{a} \leq C_i$. Specifically, due to distinct computing and communication ability of local vehicles, the RSUs receive the local verification results at different times, which can be directly authenticated as long as receiving them. Thus, the time of RSU to authenticate its local verification results is negligible.

While finding the same positive verification results from more than 2/3 of block verifiers in the network, the RSU sends an acknowledgement and the authentication results to the block producer, and appends the block to the blockchain. However, the uncertain existence of malicious block verifiers makes it hard for RSUs to predict the exact time that they receive enough positive verification results. But under the assumption of more than 2/3 of honest block verifiers in the system, this can be achieved even until the RSU receives the last verification result in the network from other RSUs (i.e. the maximal delay). Thus, we generalize the mutual-authentication delay of an RSU to its maximal delay.

To authenticate verification results of pools i', the delay of RSU R_i can be divided into two parts: (1) Waiting for the verification results from RSU $R_{i'}$ ($T_{i'i}^{w}$), and (2) Authenticating the receiving results ($T_{ii'}^{ma}$). In practical scenarios, the verification pools are different in terms of the number of vehicles with various computing resources, and the time getting the block from the block producer, resulting in various local verification delays. To analyse the waiting time of RSUs to receive verification results from other RSUs, we sort the RSUs by their local block verification delays, that is, $T_1^L < T_2^L < ... < T_N^L$. Then, we analyse four cases about the time RSUs receiving verification results from other RSUs as follows.

Case 1: The RSU R_i firstly finishes and propagates its local collection of verification results to other RSUs $R_{i'}$ ($i < i'$). Then, R_i starts to wait for $R_{i'}$s' collections. However, $R_{i'}$ may finish their collections during receiving verification results from R_i, and simultaneously propagate their verification results to R_i. At this moment, R_i and $R_{i'}$ have to allocate the occupation time of the shared channel for sending and receiving information simultaneously [10]. But, the RSUs only have to communicate with each other once instead of communicating frequently while exchanging the results. Moreover, the RSUs have to fully receive the transmitted data before authentication. To reduce interference and ensure stability of data transmission in the shared channel, we assume that when two nodes have to transmit data concurrently, the node firstly proposing the transmission request occupies the channel until finishing the data transmission. In this case, $R_{i'}$ has to send its results to R_i after R_i's transmission. And the waiting time of R_i for the verification results from $R_{i'}$ is given by

$$T_{i'i}^{w_1} = T_i^L + T_{ii'}^u + T_{i'i}^u. \tag{4}$$

Case 2: The RSUs $R_{i'}$ do not finish their local collection of verification results when they finish receiving the results from R_i ($i < i'$). In such case, the waiting time of R_i to receive their verification results is given by

$$T_{i'i}^{w_2} = T_{i'}^L + T_{i'i}^u. \tag{5}$$

Case 3: The RSU R_i may be receiving verification results from other RSUs $R_{i'}$ ($i' < i$) when R_i finishes its local collection. In this case, R_i can receive the verification results from $R_{i'}$ until the receiving transmission is completed. Therefore, the waiting time of R_i to receive the verification results from $R_{i'}$ is given by

$$T_{i'i}^{w_3} = T_{i'}^L + T_{i'i}^u. \tag{6}$$

Case 4: The RSU R_i may finish receiving verification results from other RSUs $R_{i'}$ ($i' < i$) before finishing its local collection, which R_i receives verification results from $R_{i'}$ without any waiting time. However, R_i cannot authenticate the receiving results before its completion of local collection. In this special case, we replace the waiting time of R_i to receive the results from $R_{i'}$ by the waiting time for conducting authentication on the receiving results, i.e.,

$$T_{i'i}^{w_4} = T_i^L. \tag{7}$$

As long as receiving verification results from other RSUs and finishing the local collection, the RSU R_i starts to authenticate the receiving results. In general, the maximal mutual-authentication delay of RSU R_i is given by

$$T_i^M = \underset{i' \in \mathbb{I}_R \setminus \{i\}}{max} \{ \underset{w=w_1,w_2,w_3,w_4}{max} \{ T_{i'i}^{w} + T_{ii'}^{ma} \} \}. \tag{8}$$

IV. Problem Formulation and Solution

A. Problem Formulation

In this section, we formulate an optimization problem which the computing resources of RSUs and vehicles are allocated to maximize the system utility that is proportional to the throughput. After authenticating all receiving results, RSU R_i uploads an acknowledgement with the mutual-authentication results to the block producer. We denote the corresponding transmission time from RSU R_i to block producer R_1 as $T_{i1}^{ack} = (c + \frac{2}{3}(\sum_{i=1}^{N} \mathcal{P}_i + N) S_{vr}) / r_{i1}^u$, where $\frac{2}{3}(\sum_{i=1}^{N} \mathcal{P}_i + N)$ is the number of the same positive verification results. c denotes the size of acknowledgement, information of mutual-authentication results and even attacks. r_{i1}^u is the data transmission rate from R_i to R_1. In general, the total block verification delay of R_i is given by

$$T_i = T_i^M + T_{i1}^{ack}. \tag{9}$$

Substituting (4)-(8) to (9), and defining $t_{ii'}$ as the delay of R_i to authenticate verification results from $R_{i'}$, we obtain

$$t_{ii'} = max\{T_i^L + T_{ii'}^u + T_{i'i}^u + T_{ii'}^{ma}, T_{i'}^L + T_{i'i}^u + T_{ii'}^{ma}, T_i^L + T_{ii'}^{ma}\}. \tag{10}$$

Then, the total block verification delay of RSU R_i is given by

$$T_i = \max_{i' \in \mathbb{I}_R \backslash \{i\}} t_{ii'} + T_{i1}^{ack}. \tag{11}$$

Furthermore, the system utility should monotonically increase with the throughput, which is decided by the time that the block producer receives acknowledgement from more than 2/3 of RSUs, but it is unable to forecast the exact time. Therefore, we model the system utility function based on the maximal block verification delay and also minimal throughput of the system as follows:

$$U = \kappa X / \max_{i \in \mathbb{I}_R} T_i, \tag{12}$$

where $\kappa > 0$ is a benefit transformation parameter, and the higher κ means the more revenues for the system. To formulate the optimization problem, we define $\mathbf{c^v} = \{c_{ij}^v\}$ as computing resource matrix of vehicles for verifying blocks, $\mathbf{c^a} = \{c_{ii'}^a\}$ as computing resource matrix of RSUs for authenticating verification results from other pools, respectively. We formulate the optimization problem as follows:

$$\max_{\mathbf{c^v}, \mathbf{c^a}} \quad U = \kappa X / \max_{i \in \mathbb{I}_R} T_i \tag{13a}$$

$$\text{s.t.} \quad \sum_{i' \in \mathbb{I}_R \backslash \{i\}} c_{ii'}^a \leq C_i, \forall i \in \mathbb{I}_R, \tag{13b}$$

$$c_{ij}^v \leq C_{ij}, \forall j \in \mathbb{P}_i, i \in \mathbb{I}_R, \tag{13c}$$

$$T_i \leq T^{max}, \forall i \in \mathbb{I}_R. \tag{13d}$$

The constraint in (13d) is necessary to ensure that the block verification delay of each RSU does not exceed the maximal allowed block verification delay T^{max}.

B. Problem Solution

To solve the problem in (13a)-(13d), we firstly find in (11) that compared with $t_{ii'}$, the value of T_{i1}^{ack} is extremely small and negligible. Then, it is straightforward to verify that $\max_{i \in \mathbb{I}_R} T_i$ in (13a) is closely approximated by $(\sum_{i=1}^N T_i)/N$, so that the problem in (13a)-(13d) takes the form:

$$\max_{\mathbf{c^v}, \mathbf{c^a}} \quad \kappa X N / \sum_{i=1}^N T_i, \qquad \text{s.t.} (13b), (13c), (13d). \tag{14}$$

Then, the objective of (14) is minimizing the block verification delay of each RSU (T_i), in which it is intractable to minimize the items of $\max_{i' \in \mathbb{I}_R \backslash \{i\}} t_{ii'}$. Similar to [11], we transform the problem (14) to minimize the average time consuming of RSU R_i to authenticate verification results from all other RSUs, i.e., $\frac{1}{(N-1)} \sum_{i' \in \mathbb{I}_R \backslash \{i\}} t_{ii'}$. Since $t_{ii'}$ is non-differentiable with respect to c_{ij}^v and $c_{ii'}^a$. Similar to [12], we approximate $t_{ii'}$ as

$$\begin{aligned} t_{ii'} \leq & T_i^L + T_{ii'}^u + T_{i'i}^u + T_{ii'}^{ma} + T_{i'}^L + T_{i'i}^u + T_{ii'}^{ma} + \\ & T_i^L + T_{ii'}^{ma} = 3T_{ii'}^{ma} + 2T_i^L + T_{i'}^L + \Upsilon_{ii'}, \end{aligned} \tag{15}$$

where $\Upsilon_{ii'} = T_{ii'}^u + 2T_{i'i}^u$, and substitute (2) into $(3T_{ii'}^{ma} + 2T_i^L + T_{i'}^L + \Upsilon_{ii'})$. The problem of (14) transforms to

$$\begin{aligned} \max_{\mathbf{c^v}, \mathbf{c^a}} \quad & \kappa X / (\frac{1}{N(N-1)} \sum_{i=1}^N \sum_{i' \in \mathbb{I}_R \backslash \{i\}} (3T_{ii'}^{ma} + \Upsilon_{ii'}) \\ & + \frac{3}{N} \sum_{i=1}^N \max_{j_i \in \mathbb{P}_i} \{T_{ij}^L\} + \frac{1}{N} \sum_{i=1}^N T_{i1}^{ack}) \end{aligned} \tag{16a}$$

$$\begin{aligned} \text{s.t.} \quad & (13b), (13c), \\ & t_{ii'}^{UB} + T_{i1}^{ack} \leq T^{max}, \forall i \in \mathbb{I}_R, i' \in \mathbb{I}_R \backslash \{i\} \end{aligned} \tag{16b}$$

From (16a), we can find that maximizing the system utility is to minimize each $T_{ii'}^{ma}$ and $\max_{j \in \mathbb{P}_i} \{T_{ij}^L\}$, which relate to $\mathbf{c^a}$ and $\mathbf{c^v}$, respectively. However, minimizing $\max_{j \in \mathbb{P}_i} \{T_{ij}^L\}$ is also intractable. Thus, we adopt the similar approach to minimize its average, which is $(\sum_{j \in \mathbb{P}_i} T_{ij}^L)/\mathcal{P}_i$. Finally, substituting (1) and (3) into (16a), the optimization function is denoted as

$$\begin{aligned} \max_{\mathbf{c^v}, \mathbf{c^a}} \quad & \kappa X / (\Psi + \frac{3\delta}{N(N-1)} \sum_{i=1}^N \sum_{i' \in \mathbb{I}_R \backslash \{i\}} \frac{\mathcal{P}_{i'} + 1}{c_{ii'}^a} + \\ & \frac{3}{N} \sum_{i=1}^N \sum_{j \in \mathbb{P}_i} \frac{\alpha X S_{tr}}{\mathcal{P}_i c_{ij}^v}) \end{aligned} \tag{17a}$$

$$\begin{aligned} \text{s.t.} \quad & (13b), (13c), \\ & 2\frac{\alpha X S_{tr}}{c_{ij}^v} + \frac{\alpha X S_{tr}}{c_{i'j'}^v} + 3\frac{(\mathcal{P}_{i'} + 1)\delta}{c_{ii'}^a} + \Gamma_{jj'} \leq T^{max}, \\ & \forall j \in \mathbb{P}_i, i \in \mathbb{I}_R, j' \in \mathbb{P}_{i'}, i' \in \mathbb{I}_R \backslash \{i\}, \end{aligned} \tag{17b}$$

where $\Psi = \frac{1}{N(N-1)} \sum_{i=1}^N \sum_{i' \in \mathbb{I}_R \backslash \{i\}} \Upsilon_{ii'} + \frac{1}{N} \sum_{i=1}^N T_{i1}^{ack} + \frac{3}{N} \sum_{i=1}^N \sum_{j \in \mathbb{P}_i} \frac{1}{\mathcal{P}_i} (\frac{X S_{tr}}{r_{1i}^d} + \frac{(X S_{tr} + S_{aux})}{r_{ij}^d} + \frac{S_{vr}}{r_{ji}^u})$, and $\Gamma_{jj'} = \Upsilon_{ii'} + T_{i1}^{ack} + 2(\frac{X S_{tr}}{r_{1i}^d} + \frac{(X S_{tr} + S_{aux})}{r_{ij}^d} + \frac{S_{vr}}{r_{ji}^u}) + (\frac{X S_{tr}}{r_{1i'}^d} + \frac{(X S_{tr} + S_{aux})}{r_{i'j'}^d} + \frac{S_{vr}}{r_{j'i'}^u})$. It is straightforward to verify that the problem of (17a) is concave due to its concave objective, and convex constraints. In particular, the concavity of (17a) follows from the fact that the Hessian matrix of the function is negative definite [12]. Also, the problem in (17a) is the same as

$$\min_{\mathbf{c^v}, \mathbf{c^a}} \frac{3\delta}{N(N-1)} \sum_{i=1}^N \sum_{i' \in \mathbb{I}_R \backslash \{i\}} \frac{\mathcal{P}_{i'} + 1}{c_{ii'}^a} + \frac{3}{N} \sum_{i=1}^N \sum_{j \in \mathbb{P}_i} \frac{\alpha X S_{tr}}{\mathcal{P}_i c_{ij}^v}. \tag{18}$$

It is clear that (18) is a convex function related to $\mathbf{c^a}$ and $\mathbf{c^v}$, which has the same solutions as (17a) and is much easier to be solved. Hence, we use KKT conditions of (18) to find the best solutions of $\mathbf{c^a}$ and $\mathbf{c^v}$. The Lagrange function of (18) is given by (19) with the Lagrange multipliers β, θ and η.

$$\begin{aligned} \mathcal{L} = & \frac{3\delta}{N(N-1)} \sum_{i=1}^N \sum_{i' \in \mathbb{I}_R \backslash \{i\}} \frac{\mathcal{P}_{i'} + 1}{c_{ii'}^a} + \frac{3}{N} \sum_{i=1}^N \sum_{j \in \mathbb{P}_i} \frac{\alpha X S_{tr}}{\mathcal{P}_i c_{ij}^v} \\ & + \beta_{jj'} (\frac{2\alpha X S_{tr}}{c_{ij}^v} + \frac{\alpha X S_{tr}}{c_{i'j'}^v} + \frac{3\delta(\mathcal{P}_{i'} + 1)}{c_{ii'}^a} + \Gamma_{jj'} - T^{max}) \\ & + \theta_i (\sum_{i' \in \mathbb{I}_R \backslash \{i\}} c_{ii'}^a - C_i) + \eta_{ij} (c_{ij}^v - C_{ij}), \end{aligned} \tag{19}$$

To meet all KKT conditions, the following complementary Slackness conditions must be satisfied:

$$\beta_{jj'}(\frac{2\alpha X S_{tr}}{c^v_{ij}} + \frac{\alpha X S_{tr}}{c^v_{i'j'}} + \frac{3\delta(\mathcal{P}_{i'}+1)}{c^a_{ii'}} + \Gamma_{jj'} - T^{max}) = 0, \tag{20a}$$

$$\theta_i(\sum_{i' \in \mathbb{I}_R \backslash \{i\}} c^a_{ii'} - C_i) = 0, \tag{20b}$$

$$\eta_{ij}(c^v_{ij} - C_{ij}) = 0, \tag{20c}$$

Apart from this, all items regrading Lagrange multiplies should be bigger than 0, i.e., $(\beta, \theta, \eta) > 0$. And (13b)-(13d) must be satisfied at the same time. To satisfy the above conditions and first-order derivative optimality conditions, that is, $\partial\mathcal{L}/\partial c^a_{ii'} = 0$, and $\partial\mathcal{L}/\partial c^v_{ij} = 0$, we consider three possible cases:

Case 1: $\theta_i = 0$ or $\eta_{ij} = 0$. According to $\frac{\partial\mathcal{L}}{\partial c^a_{ii'}} = 0$, $\frac{\partial\mathcal{L}}{\partial c^v_{ij}} = 0$, there are not $c^a_{ii'}$ or c^v_{ij} can satisfy all KKT conditions.

Case 2: $\theta_i > 0$, $\eta_{ij} > 0$ and all $\beta_{jj'} = 0$. According to $(\partial\mathcal{L}/\partial c^a_{ii'}) = 0$ and $(\sum_{i' \in \mathbb{I}_R \backslash \{i\}} c^a_{ii'}) = C_i$, we obtain

$$\theta_i = \frac{1}{c_i^2}(\sum_{i' \in \mathbb{I}_R \backslash \{i\}} \sqrt{\frac{3(\mathcal{P}_{i'}+1)\delta}{N(N-1)}})^2, \tag{21a}$$

$$c^a_{ii'} = \sqrt{\frac{3(\mathcal{P}_{i'}+1)\delta}{N(N-1)}\frac{1}{\theta_i}}, \tag{21b}$$

$$c^v_{ij} = C_{ij}. \tag{21c}$$

Case 3: $\theta_i > 0$, $\eta_{ij} > 0$, $\beta_{jj'} > 0$. Note that $\beta_{jj'}$ is related to vehicle j in pool i with vehicle j' in pool i', with respect to the value of (c^v_{ij}, $c^v_{i'j'}$, $c^a_{ii'}$). Thus, we further divide this case into three sub-cases:

Case 3.1: There is more than one $\beta_{jj'}(j \in \mathbb{P}_i, j' \in \mathbb{P}_{i'})$ being bigger than 0. Taking $\beta_{j_1 j'_1} > 0$ and $\beta_{j_2 j'_2} > 0$ as an example, to satisfy the first Slackness condition in (20b), $[2 * (T^L_{ij_1} + T^L_{i'j'_1})] = [(T^L_{ij_2} + T^L_{i'j'_2})]$ must be satisfied. In practical scenarios, this condition is hard to realise, and thus not $c^a_{ii'}$ can satisfy all KKT conditions in this case.

Case 3.2: For each pair of verification pools i and i', there is one item being bigger than zero among all of $\beta_{jj'}(j \in \mathbb{P}_i, j' \in \mathbb{P}_{i'})$. According to the first and third conditions in (20b), we obtain $c^v_{ij} = C_{ij}$ and

$$c^a_{ii'} = \frac{3(\mathcal{P}_{i'}+1)\delta}{T^{max} - (2\frac{\alpha X S_{tr}}{C_{ij}} + \frac{\alpha X S_{tr}}{C_{i'j'}} + \Gamma_{jj'})}. \tag{22}$$

Case 3.3: For each pair of verification pools i and i', there is no more than one item being bigger than zero among all of $\beta_{jj'}(j \in \mathbb{P}_i, j' \in \mathbb{P}_{i'})$, which means that, for some pairs of verification pools, all related $\beta_{jj'} = 0$. We define the whole set of verification pools except for pool i as $\mathbb{S}_i$, in which the set of verification pools i' having one related $\beta_{jj'} > 0(j \in \mathbb{P}_i, j' \in \mathbb{P}_{i'})$ are presented as $\mathbb{S}^1_i$. Then, we can obtain

$$c^v_{ij} = C_{ij}, \tag{23a}$$

$$\theta_i = [(\sum_{i' \in \mathbb{S}_i \backslash \mathbb{S}^1_i} \sqrt{\frac{3(\mathcal{P}_{i'}+1)\delta}{N(N-1)}})/(C_i - \sum_{i' \in \mathbb{S}^1_i} c^a_{ii'})]^2, \tag{23b}$$

TABLE I
SIMULATION PARAMETERS SETTING

Parameter	Setting
Maximal computing resources of RSUs C_i	[2.5, 10] GHash/s
Maximal computing resources of vehicles C_{ij}	[25, 100] MHash/s
Transmission rates of RSUs	[1, 2] MB/s
Transmission rates of Vehicles	[500, 600] KB/s
Allowed block verification delay T^{max}	30 seconds
Size of auxiliary verification information S_{aux}	30 KB
Computing resource for verifying per unit size of block α	20 Hash/KB
Computing resource for mutually authenticating per verification result δ	30 Hash
Average size of per transaction S_{tr}	500 Byte

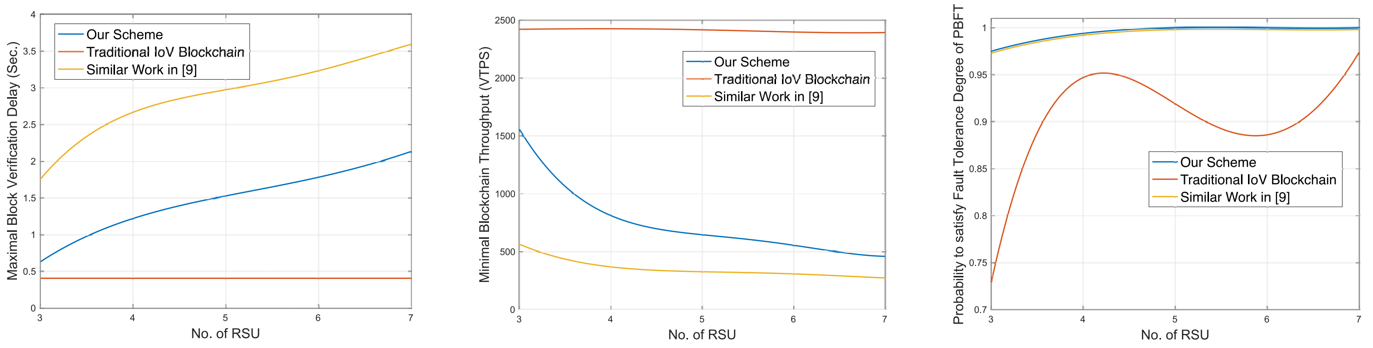
$$c^a_{ii'} = \begin{cases} \frac{3(\mathcal{P}_{i'}+1)\delta}{T^{max} - (2\frac{\alpha X S_{tr}}{C_{ij}} + \frac{\alpha X S_{tr}}{C_{i'j'}} + \Gamma_{jj'})}, & i' \in \mathbb{S}^1_i, \\ \sqrt{\frac{3(\mathcal{P}_{i'}+1)\delta}{N(N-1)}\frac{1}{\theta_i}}, & i' \in \mathbb{S}_i \backslash \mathbb{S}^1_i, \end{cases} \tag{23c}$$

$$\beta_{jj'} = \begin{cases} 0, \\ \theta_i(c^a_{ii'})^2 - \frac{3(\mathcal{P}_{i'}+1)\delta}{N(N-1)}. \end{cases} \tag{23d}$$

V. Performance Evaluation

A model of the IoV blockchain is simulated with Matlab. The model comprises 13 RSUs, in which the number of RSUs that verify the block and recruit the vehicles ranges from 3 to 7. The number of vehicles in each verification pool or RSU is (3,4,4,7,7,4,8), respectively. Also, the sizes of the block, per verification result and mutual-authentication result detail (c) are 500KB, 50KB and 20KB, respectively [13]. The data transmission rates and other information about block verifiers are put in TABLE I. To observe the performance of our scheme, we introduce the following comparable schemes: (1) *Traditional IoV blockchain*: only RSUs verify the block with the optimized computing resources. (2) *Similar work in [9]*: the work is to optimize the recruitment ratio of vehicles and transaction fee of user to maximize their respective utilities.

Firstly, we evaluate the efficiency of the compared IoV blockchain schemes in terms of main performance parameters, i.e., block verification delay, system throughput, and security. Fig. 2a and Fig. 2b show the maximal block verification delays and minimal blockchain throughputs in the compared schemes. Compared with [9], our scheme has less block verification delay, and therefore higher blockchain throughput, which is close to that of traditional IoV blockchain. Importantly, similar to [14], we exploit pBFT as an example of the consensus algorithm to evaluate the system security, which is modelled as a random sampling problem as follows: $P(\mathcal{D} \leq \mathcal{Y}) = \sum_{y=0}^{\mathcal{Y}} \binom{\mathcal{Z}}{y} p_d^y (1 - p_d)^{\mathcal{Y}-y}$. $\mathcal{D}$ is the number of malicious verifiers, while $\mathcal{Y}$ is the number of malicious verifiers that the system can tolerate with the total number of verifiers $\mathcal{Z}$. In pBFT, $\mathcal{Y} = \lfloor(\mathcal{Z}-1)/3\rfloor$. Also, p_d is the dishonest probability of each verifier, ranging from 0.1 to 0.3. Based on this, we can find the vehicles provide more security to the traditional IoV blockchain, especially the small-scale blockchains, as shown in Fig. 2c. Therefore, our scheme has a more ideal trade-off in system throughput and security compared with other schemes.



(a) (b) (c)

Fig. 2. Comparison of (a) Maximal block verification delay, (b) Minimal blockchain throughput (Verified Transactions Per Second, VTPS) and (c) System security degree based on PBFT among three schemes.

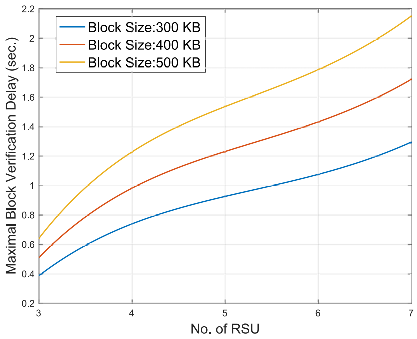


Fig. 3. Block verification delay with various block sizes and number of RSUs.

Then, the impact of the block size on the maximal block verification delay is evaluated. The examined block size is (300,400,500)KB. Also, the larger block has more transactions and leads to larger size of block verification results. Thus, the corresponding sizes of block verification results are (30,40,50)KB. From Fig. 3, the larger block has more block verification delay, especially with more RSUs and vehicles. In other words, the block verification delay is not linearly increasing with the block size, but increasing faster with large number of RSUs and vehicles.

VI. Conclusion

In this paper, we focused on the vehicles-enhanced block verification in IoV blockchains. To reduce block verification delay and improve throughput, we jointly optimize computing resources of RSUs and vehicles for block verification with the aim to maximize the system utility. Extensive experiments presented that, our computing resources optimization scheme has an ideal tradeoff between throughput and system security. In the future, machine learning etc., can also be involved to detect the malicious block verifiers.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Project No. 61950410603, the Characteristic Innovation Project No. 2021KTSCX110 of Guangdong Provincial Department of Education, EPSRC through the grants EP/R007195/1 (Academic Centre of Excellence in Cyber Security Research - University of Warwick), EP/S035362/1 (PETRAS, the National Centre of Excellence for IoT Systems Cybersecurity), EP/N510129/1 (The Alan Turing Institute) and EP/R029563/1 (Autotrust). The Corresponding Author is Alia Asheralieva.

References

[1] J. Cui et al., “Secure and efficient data sharing among vehicles based on consortium blockchain,” IEEE Trans. on Intelligent Transportation Systems, 2021.

[2] C. Lepore et al., “A survey on blockchain consensus with a performance comparison of pow, pos and pure pos,” Mathematics, vol. 8, no. 10, p. 1782, 2020.

[3] D. Mingxiao et al., “A review on consensus algorithm of blockchain,” in 2017 IEEE international conference on systems, man, and cybernetics (SMC), pp. 2567–2572, IEEE, 2017.

[4] Z. Zheng et al., “An overview of blockchain technology: Architecture, consensus, and future trends,” in 2017 IEEE international congress on big data (BigData congress), pp. 557–564, IEEE, 2017.

[5] J. Liu et al., “An improved dpos consensus mechanism in blockchain based on plts for the smart autonomous multi-robot system,” Information Sciences, vol. 575, pp. 528–541, 2021.

[6] A. Ometov et al., “An overview on blockchain for smartphones: State-of-the-art, consensus, implementation, challenges and future trends,” IEEE Access, vol. 8, pp. 103994–104015, 2020.

[7] F. A. Khan et al., “Rift: A high-performance consensus algorithm for consortium blockchain.”

[8] X. G. Yu’na et al., “6g oriented blockchain based internet of things data sharing and storage mechanism,” Journal on Communications, vol. 41, no. 10, p. 48, 2020.

[9] J. Kang et al., “Incentivizing consensus propagation in proof-of-stake based consortium blockchain networks,” IEEE Wireless Comm. Letters, vol. 8, no. 1, pp. 157–160, 2018.

[10] T. Huynh et al., “Joint downlink and uplink interference management for device to device communication underlaying cellular networks,” IEEE Access, vol. 4, pp. 4420–4430, 2016.

[11] U. Mohammad, S. Sorour, and M. Hefeida, “Task allocation for asynchronous mobile edge learning with delay and energy constraints,” arXiv preprint arXiv:2012.00143, 2020.

[12] Y. Dai et al., “Joint load balancing and offloading in vehicular edge computing and networks,” IEEE Internet of Things Journal, vol. 6, no. 3, pp. 4377–4387, 2018.

[13] J. Kang et al., “Toward secure blockchain-enabled internet of vehicles: Optimizing consensus management using reputation and contract theory,” IEEE Trans. on Vehic. Tech., vol. 68, no. 3, pp. 2906–2920, 2019.

[14] K. Lei et al., “Groupchain: Towards a scalable public blockchain in fog computing of iot services computing,” IEEE Trans. on Services Computing, vol. 13, no. 2, pp. 252-262, 2020.