IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 7 July 2024, accepted 21 July 2024, date of publication 29 July 2024, date of current version 8 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3434691

RESEARCH ARTICLE

Module Lattice-Based Post Quantum Secure Blockchain Empowered Authentication Framework for Autonomous Truck Platooning

DHARMINDER CHAUDHARY[1], (Member, IEEE), P. SANTHI[2], (Member, IEEE), M. S. P. DURGARAO[1], (Member, IEEE), A. PADMAVATHI[1], (Member, IEEE), MOHAMMAD MEHEDI HASSAN[3], (Senior Member, IEEE), AND BADER FAHAD ALKHAMEES[3], (Member, IEEE)

[1]Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Chennai 601103, India
[2]Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, Tamil Nadu 600062, India
[3]Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia

Corresponding author: Dharminder Chaudhary (manndharminder999@gmail.com)

This work was supported by King Saud University, Riyadh, Saudi Arabia, through the Researchers Supporting Project under Grant RSP2024R493.

ABSTRACT Truck platooning uses networking technology and automated driving support systems to join multiple trucks in a group. When these vehicles interact for particular journey stages, such as on highways, they autonomously maintain a predefined, tight spacing among themselves. Platooning improves transportation by making better use of highways, delivering cargo faster, and minimizing congestion in traffic. Therefore, safety and platooning are two important attributes of an intelligent truck system. This paper discusses a quantum-safe blockchain-empowered authentication mechanism for autonomous truck platooning. The proposed idea is to use blockchain to combine multiple nodes and ensure authenticity with the help of an aggregation technique. The proposed design ensures authenticity to the system due to hard assumptions:1) Learning With Error (LWE) and 2) Short Integer Solution (SIS) on random module generated lattice. This paper uses operations over module lattices to be more efficient than general lattices. We can perform operations on module lattices with the help of fast algorithms for polynomial arithmetic.

INDEX TERMS Blockchain, signature, aggregate signature, truck platooning, cryptography.

I. INTRODUCTION

Truck platooning is a technique used in transportation and logistics where multiple trucks travel closely together in a convoy, with the vehicles electronically linked to synchronize their movements. The leading truck in the platoon, often referred to as the "platoon leader" or "lead truck," sets the pace and controls the acceleration, braking, and steering, while the following trucks, known as "follower trucks," automatically adjust their speed and maintain a safe following distance. The platooning of trucks, facilitated by vehicular ad hoc networks (VANET), is anticipated to have a substantial influence on the freight sector. This process involves connecting several trucks through vehicle automation and wireless communication. Platoon members can collectively maneuver in an automatic method by communicating information on parameters like speed, direction, and acceleration between vehicles. This leads to reduced fuel usage, increased road capacity, and, crucially, enhanced safety [1]. To be precise, a platoon member can maintain a substantially lower distance

The associate editor coordinating the review of this manuscript and approving it for publication was P. Venkata Krishna.

(such as 0.6 sec) behind the vehicle in front of it compared to the traditional two-second gap a human driver would keep. This notable decrease in distance has the potential to enhance lane capacity by up to 4,250 vehicles per hour [2], which is twice the current lane capacity. The potential advantages also encompass improved fuel economy resulting from the shorter following distances. According to the Japan Intelligent Transport System Energy project, maintaining a 10-meter spacing at a speed of 80 km/h within the platoon could potentially lead to a fuel saving of approximately 13 percent [1]. Truck platooning offers drivers the chance to relinquish their active driving responsibilities. Automated driving systems have the potential to respond more quickly and evaluate changing traffic situations with greater precision compared to humans. For truck platooning, two frameworks are available: the opportunistic approach and the road train technique. The road train approach relies on forming groups or platoons of trucks with the same beginning and ending sites. The VANET framework is better suited for the opportunistic approach. Within this model, a system is required in this model to discover possible platooning possibilities among trucks from different fleets and to equally assign duties within the platoon. Despite exhibiting mixed fleet platooning, the Grand Cooperative Driving Challenge (GCDC) competition setting did not address safety and confidentiality problems [3]. According to our understanding, this is the first research to provide a privacy-centered and efficient verification approach for grouping trucks with distinct fleets, where the trucks are owned and operated by different firms. So far, there has been no investigation on platooning system security in scenarios involving mixed fleets, with an emphasis on defending against assaults and preventing unauthorized access to vehicle details or fleet specifications. However, public awareness of these challenges is growing. System security must confirm a vehicle's identification before enabling it to join a platoon. This protection ensures the platoon's integrity and protects against any potential threats linked with impersonation. Concurrently, truck owners may be hesitant to share confidential vehicle information with untrustworthy third parties and other players due to privacy concerns. For instance, Participation in the system would be discouraged if competitors were able to analyze a company's truck platooning data and uncover its intricate logistics procedures. Thus, exploiting the benefits of truck platooning systems in a mixed fleet context is hampered by the combined task of preserving truck owners' privacy and providing a way for dynamic identification verification across various trucks. Blockchain possesses valuable attributes that can facilitate a robust, flexible, and decentralized authentication framework in the context of dynamic truck platooning. Since its original application in the Bitcoin protocol as a decentralized ledger for digital money [4], blockchain, a strong and distributed network technology, has piqued the curiosity of researchers. In recent times, there has been a growing body of research highlighting the immense possibilities of blockchain technology in different areas apart from cryptocurrency. These include Intelligent Transport System [5], [6], [7], smart cities [8], [9], [10] and healthcare [11], [12], [13]. Notably, blockchain technology is divided into two categories: permissionless and permissioned. These variants share fundamental characteristics like as immutability, network dependability, and data traceability. As a result, they are promising candidates for supporting a dynamic truck platooning system. Permissionless blockchain systems, on the other hand, are designed to provide complete transparency, which might cause privacy concerns when such a system processes and retains sensitive user information. In the case of Bitcoin, the complete ledger is open to the public and every party can access it. There's no provision to set different levels of authorization or establish direct access controls within the chain. Furthermore, the agreement techniques employed in decentralized blockchains, like proof of work, allow unrestricted involvement. These methods are typically centered around tokens and deliberately incorporate sophisticated computations to safeguard the network against potential risks. Such qualities result in significant inefficiencies, excessive bandwidth and energy use, and environmental issues. A permissioned blockchain, on the other hand, enables the incorporation of varying levels of authorization and measures for regulating access to secure data within the chain. This offers a structure appropriate for systems where data ownership and confidentiality are paramount. Nevertheless, the features of a permissioned blockchain solely offer security for sensitive user data after it has been recorded on the ledger and does not guarantee privacy throughout the verification process. To address this problem, several researchers have turned to cryptographic methods, including adaptations of the blockchain, in innovative ways to safeguard the data before it is stored [14], [15], [16].

II. RELATED STUDIES AND BACKGROUND

Blockchain in transportation is an effective distribution method by creating new intelligent means of transportation that are safe, reliable and distributed. For example, establishing secure and reliable data sharing for V2V and V2I communication has been an important research topic in recent years [17]. Guo et al. [18] proposed an architecture for blockchain including an event recording system in dynamic autonomous vehicles environment [19]. In addition, they proposed the blockchain for dynamic control proposed for ongoing more than one communication with the Intelligent Transportation System. The blockchain architecture helps us to provide a decentralized traffic communication system and maintains data integrity and authenticity while participating in Internet of Vehicles [20]. However, the networking topology of Internet of Vehicles is dynamic, and it becomes very challenging to ensure integrity and authenticity. Researchers have been continuously trying to analyze the potential of blockchain blocks in designing the required framework for the last five years. Lin et al. [21] introduced the concept of conditional privacy in the blockchain to ensure authentication with the help of PKI based signature and Ethereum blockchain, to provide secure communication in the Vehicular Adhoc Networking system. They have tried to show that using smart contracts with the public Ethereum takes more cost for vehicles due to the limited amount paid to miners. In the year 2019, Feng et al. [22] comes with a new idea blockchain-based privacy-preserving system for vehicular communications. This system enables an automatic authentication mechanism for Virtual Asset Network (VANET), and it provides protection to private vehicles. However, this system cannot ensure support for shared proof of consent. Yao et al. [23] introduced anonymous blockchain architecture to ensure authenticity in the distributed environment for the Internet of Vehicles communication system. This mechanism is only responsible for simulating the consensus algorithms and it does not take care of other computational costs associated with the maintenance of blockchain. Kaur et al. [24] introduced a new framework to provide authenticity to sensitive data and it is helpful in transformation using blockchain, but it lacks blockchain testing. Liu et al. [25] introduced another framework supporting group-based blockchain architecture because moving cars have decentralized dynamic topology. The authors recommend mixing the work safely but find evidence that the device is very difficult to work with. They plan to use blockchain technology to exchange information between vehicles in groups to ensure partner privacy and security and allow high-speed data sharing. Ying et al. [26] introduced a new mechanism to generate smart contracts that allow users to perform transactions at lower cost, but the process of verification is hampered by Ethereum's confirmation being slower than another permissioned blockchain. Later, they introduced an advanced payment model based on blockchain for time saving, and to speed up authentication. To improve city traffic and reduce the number of accidents, the authors of [27] proposed a vehicle model that can drive on a single track. The model is based on smart contracts that allow payments between board members and members to be made via blockchain, preventing bad money and fraud. After going through the literature, we have tried to analyze previous research related to post quantum blockchain authentication. Blockchain-assisted authentication mechanism is helpful for various practical applications in the automotive industry. Most of the research lacks in supporting authenticated decentralized blockchains [1], [23], [24], [25]. Private blockchain assisting security of vehicles [23], [24], [25], [27] have been used to provide privacy to them. Therefore, we have analysed security and privacy challenges in the vehicular communication system, and it is very helpful in autonomous truck platooning formation with mixed fleets. We have gone through the related work on post quantum blockchains [28], [29], [30], [31], [32], and implimented it in the autonomous truck platooning.

III. MOTIVATION AND CONTRIBUTION

We are motivated by the existing research [28], [29], [30], [31], [32] helped a lot in designing post-quantum blockchain-based truck platooning. The motivation for developing quantum-safe blockchain architectures stems from the potential threat posed by quantum computers to traditional cryptographic algorithms. Quantum computers can solve certain mathematical problems, such as integer factorization and discrete logarithms, much faster than classical computers. These algorithms form the basis of many cryptographic schemes used in blockchain technology and other security protocols. In order to construct an efficient structure, we have used identity-based key extraction along with a module lattice-based aggregation of signatures. The module lattice-based approach to signature aggregation involves representing the set of signatures as vectors in a lattice. Each signature is treated as a lattice point, and the aggregation process aims to find a lattice point that combines these signatures in a way that preserves their individual contributions. Security study shows that the suggested technique is resistant to a number of quantum threats.

IV. PRELIMINARIES

Post-quantum cryptography refers to cryptographic algorithms and protocols designed to remain secure against attacks by quantum computers. Quantum computers have the potential to break many of the cryptographic schemes currently used to secure digital communication and data, such as RSA and ECC, by exploiting their ability to efficiently solve certain mathematical problems like integer factorization and discrete logarithms. To prepare for the advent of quantum computers and ensure the security of our digital infrastructure, several preliminary steps are essential in the development and adoption of post-quantum cryptography. Essential tools for understanding the subject matter are provided in this part of the article in the form of terms and symbols. Let us take $\mathbb{R}$ set of reals, and $\mathbb{Z}$ set of integers. Let $\mathbb{R} = \frac{\mathbb{Z}[x]}{x^n+1}$ be ring consisting of polynomials at most n degree [33], [34], [35], and $\mathbb{R}_q = \frac{\mathbb{Z}_q[x]}{x^n+1}$ be finite quotient ring. Module lattice extends the concept of the general lattice by introducing an additional algebraic structure.

Definition 1: A **lattice** $\mathbb{L}$ in $\mathbb{R}^n$ is a discrete algebraic subgroup of $\mathbb{R}^n$. This is formed by basis $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$, where $\mathbf{b}_i \in \mathbb{R}^n$, as $\mathbb{L} = \left\{\sum_{i=1}^{n} z_i \mathbf{b}_i \mid z_i \in \mathbb{Z}\right\}$.

A **module** over R generalizes the concept of a vector space, where the field is a ring R. The R-module M is a group with respect to addition equipped with an action of R on M. A **module lattice** is an R-module having extra structure like a lattice in some sense. In lattice-based cryptography, we do computation on module generated lattice over the ring of integers modulo a polynomial $(f(x))$ which is $R = \mathbb{Z}[x]/\langle f(x)\rangle$. An R-module M possesses a basis $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_k\}$ such that arbitrary member in M can be written $\sum_{i=1}^{k} r_i \mathbf{b}_i$, where $r_i \in R$.

A. EXAMPLE: RING-LEARNING WITH ERRORS (RING-LWE)

The Ring-Learning With Errors assumption is operated over module generated lattices. The assumption is parameterized by a ring $R = \mathbb{Z}[x]/\langle f(x)\rangle$ and involves:

B. SECRET KEY

A secret polynomial $s(x) \in R$.

C. ERROR POLYNOMIAL

An error polynomial $e(x) \in R$ with small coefficients sampled from n dimensional discrete Gaussian or another appropriate distribution.

D. PUBLIC KEY

A uniformly sampled polynomial $a(x) \in R$ is public parameter. The public key is:

$$b(x) = a(x) \cdot s(x) + e(x).$$

The computational hardness of the Ring-LWE assumption depends upon the difficulty to distinguish the polynomial $b(x)$ from a uniformly random polynomial in R.

Definition 2: Let us consider a ring R, and **module** M over R with a scalar multiplication over $R \times M \rightarrow M$ satisfying axioms for all $r, s \in R$ and $m, n \in M$:

1) $(r + s)m = rm + sm$
2) $r(m + n) = rm + rn$
3) $(rs)m = r(sm)$
4) $1_R m = m$, where 1_R is unity of R

Definition 3: A **module lattice** $\mathbb{L}$ over R can be generated by $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_k\}$, where $\mathbf{b}_i \in \mathbb{R}^n$ and $k \leq n$, as $\mathbb{L} = \left\{\sum_{i=1}^{k} r_i\mathbf{b}_i \mid r_i \in R\right\}$.

Theorem 4: Let R be an Abelian ring with identity, then R-module lattice $\mathbb{L}$ in $\mathbb{R}^n$ is a discrete algebraic subgroup of $\mathbb{R}^n$.

Example 5: Let $R = \mathbb{Z}[x]/(x^2 + 1)$ and members of R are represented by $a + bx$ where $a, b \in \mathbb{Z}$ and $x^2 = -1$. A basis for such a module lattice in $\mathbb{R}^2$ should be $\{(1, 0), (0, 1)\}$.

Lemma 6: Let $\mathbb{L}$ be a module lattice over R with $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_k\}$. For general $r \in R$ and $\mathbf{v} \in \mathbb{L}$, $r\mathbf{v} \in \mathbb{L}$.

Proof: Let $\mathbf{v} = \sum_{i=1}^{k} r_i\mathbf{b}_i$ for $r_i \in R$. Then, one will get $r\mathbf{v} = r\left(\sum_{i=1}^{k} r_i\mathbf{b}_i\right) = \sum_{i=1}^{k}(rr_i)\mathbf{b}_i$. Since $rr_i \in R$, it follows $r\mathbf{v} \in \mathbb{L}$. □

Definition 7: Let us consider a matrix $B = [b_1|b_2|\ldots|b_m] \in \mathbb{R}^{m\times m}$ with rank m, and $b_1|b_2|\ldots|b_m$ are linearly independent [33], then they generate full rank lattice $\mathbb{L}(B) = \{y = Br \in \mathbb{R}^m \mid r \in \mathbb{Z}^m,\ y = Br = \sum_{i=1}^{m} r_i.g_i\}$.

Lemma 8: The extraction algorithm *Extracts*(.) inputs $\xi \in \mathbb{Z}_q^{n\times m}$, basis $Q \in \mathbb{Z}_q^{m\times m}$ of $\mathbb{L}^{\perp}(\xi)$, and $\hat{\xi} \in \mathbb{Z}_q^{n\times \hat{m}}$, and $Extracts(Q,\ \xi' = \xi||\hat{\xi})$ returns a basis $Q' \in \mathbb{L}^{\perp}(\xi') \subseteq \mathbb{Z}^{m+\hat{m}}$ satisfying $||\tilde{Q}'|| = ||\tilde{Q}||$.

Definition 9: **Module Learning with Errors** is a hard assumption, if $\xi \leftarrow U(Q_q^{k\times\ell})$, and $e \leftarrow Q_q^k$ are known, then it is not easy to distinguish between $e \in U(Q_q^k)$, and $e = [\xi|I_k].s$, where $s \leftarrow U(S_{\xi}^{\ell+k})$, and S is distributed uniformly [36], [37].

Definition 10: **Module Short Integer Solution** assumption [36], [37] defined, if $\xi \leftarrow U(Q_q^{k\times\ell})$, then find $e \neq 0 \leftarrow Q_q^{k+\ell}$ such that $[\xi|I_k].s = 0 \in Q_q^k$, and $||e|| < \xi$.

E. TRUCK PLATOONING

Due to congestion by trucks on the road, there is a negative impact on the environment. Trucks also tend to travel in lower lanes than cars, which can reduce highway traffic. Vehicle platooning could be a solution to reduce the number of trucks on the highway. Additionally, vehicle platooning can reduce air resistance, thus reducing fuel consumption and emission levels [38]. The quality of coating cars is expected to increase as the distance between trucks decreases. Therefore, numerous studies have evaluated the concept of interconnecting autonomous vehicles using wireless communications for vehicle wrapping purposes. To create the payment process, trucks first reduce the distance to each other and then adjust their speed to move together on the highway. In the old situation, trucks belonging to one or more trucking companies formed groups before leaving their warehouses.

V. PERMISSIONED BLOCKCHAIN

A permissioned blockchain, also known as a private or consortium blockchain, is a type of blockchain network where access and participation are restricted to a predefined group of entities or nodes. Unlike public blockchains, which are open and decentralized, permissioned blockchains are controlled by a central authority or consortium of entities. There are mainly two categories of blockchain; (1) permissionless and (2) with permission [39]. While using a permissionless system, membership is completely unrestricted and anyone can participate in the networking system and watch all transactions. On the other hand, a permissioned architecture of blockchain serves as a strictly regulated membership networking system. In permissioned blockchain, certificate authority must grant permission for users to validate transactions or access network data. This feature is beneficial for corporations, financial institutions, and organizations that are willing to adhere to regulatory requirements and prioritize maintaining full control over their data. Compared to their permissionless equivalents, permissioned blockchain systems can use more efficient consensus algorithms since they have the capacity to govern membership. Additionally, programmable access restrictions that provide fine-grained control over on-chain data are possible in permissioned blockchain systems. For particular applications that require high transaction throughput while maintaining low latencies, these features make permissioned blockchain (Figure(1)) technology more appealing.

A. PERMISSIONED BLOCKCHAIN WORKING IN THE PROPOSED FRAMEWORK

A permissioned blockchain network restricts access to a specific group of participants, and it is the main difference with respect to public blockchains, where anyone can join. In reference of truck platooning, a permissioned blockchain provides a secure and efficient technique to manage the coordination and communication among the trucks of the

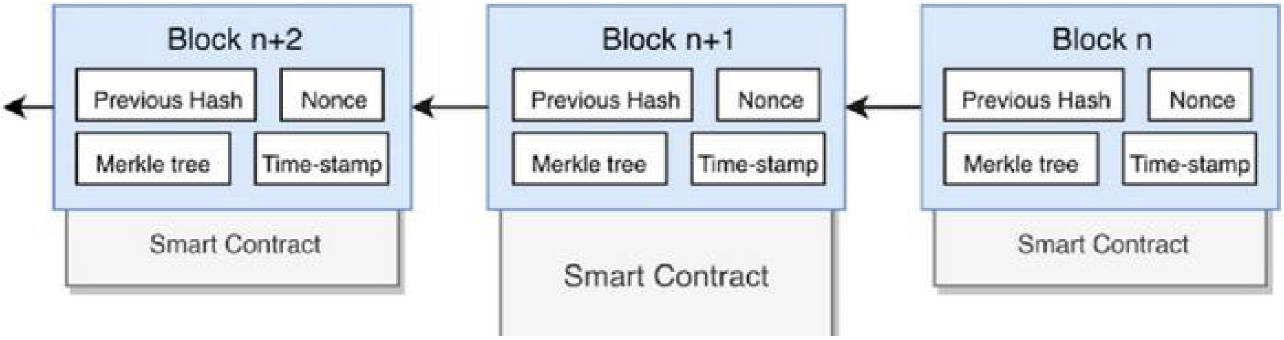


FIGURE 1. A simple architecture for Blockchain of three blocks.

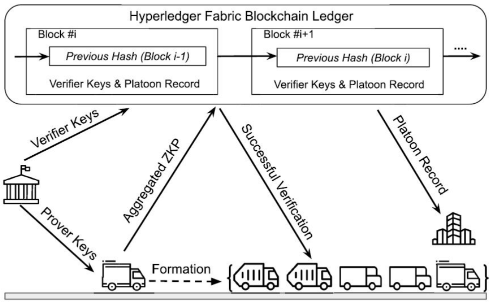


FIGURE 2. An overview of the proposed communication model.

platoon. We have discussed below how a permissioned blockchain will work in truck platooning:

1) **Permissioned Blockchain**: In this type of blockchain network authorized participant joins the network, validates transactions, and maintain the ledger. This framework ensures higher security and privacy compared to public blockchain. Some of well-known platforms are Hyperledger Fabric, Corda, and Quorum.
2) **Truck Platooning**: A leading truck gives guidance to the convoy, with following trucks maintain a close distance for reduced air resistance and improves fuel efficiency. This needs a reliable and secure communication channel for safe coordination.

B. HOW PERMISSIONED BLOCKCHAIN ENHANCES TRUCK PLATOONING

- **Controlled Access**: Permissioned blockchain restricts access to verified trucks and stakeholders participants in the network, and platooning process.
- **Secure and Efficient Communication**: All the transactions related to speed adjustment, and brake alert are recorded on the blockchains, and it ensures they are tamper-proof and verifiable.
- **Smart Contracts for Automation**: Smart contracts help in automating various aspects of platooning, such as constructing a platoon, fuel saving, and reducing human intervention and potential error.
- **Transparency and Trust**: All parties in the network have access to the same data, fostering transparency and trust among the platooning parties, and ensuring accountability and traceability of actions.
- **Decentralized Coordination**: The blockchain helps in coordinating the decentralized platoon, reducing reliance, and ensuring resilience and reliability of the platooning.

C. EXAMPLE: PERMISSIONED BLOCKCHAIN IN TRUCK PLATOONING

We have discussed step-by-step examples of how a permissioned blockchains are implimented in truck platooning:

1) **Network Setup**: Authorized members truck companies sets up a permissioned blockchain networking system using a platform like Hyperledger Fabric. In this environment, all trucks are equipped with blockchain nodes and the required communication hardware.
2) **Joining the Platoon**: Trucks X, Y, and Z want to form a platoon, and then each of the trucks submits a request to join the platoon, which is being recorded transaction on the blockchain. Each smart contract verifies the minimum eligibility based on predefined criteria like truck condition, and driver credentials.
3) **Platooning Operation**: The leader truck (X) is responsible for broadcasting its speed and position, then trucks (Y and Z) try validating the message and adjust their speeds accordingly. All of these transactions containing speed updates, and brake signals are recorded on the blockchain, and they are used to ensure real-time and secure communication.
4) **Exit Protocol**: Suppose truck Z wants to exit the platoon, then it will submit an exit request to the blockchain. The smart contract verifies the request and, if everything seems to be good, then it will approve the exit. The blockchain is also responsible for keeping records for the exit, and the platoon continues with the remaining trucks.
5) **Fuel Savings Distribution**: Finally, after completing the trip, the smart contract computes fuel savings based on the time each truck spent in the platoon. The savings are automatically distributed following the

agreement, and it will be recorded on the blockchain for transparency.

VI. SYSTEM OVERVIEW AND DESIGN GOALS

In this section, we have discussed various components of the given framework (see Figure (1), and Figure (2)).

1) Permissioned Blockchain: Our system is controlled by a permissioned blockchain, which also serves as a validator for new trucks' identities before they can join any mixed fleet platoon.
2) Certificate Authority (CA): In this context, a certificate authority is a company in charge of managing the MAC addresses and other distinctive identities of autonomous vehicles as well as delivering key pairs to data suppliers and data verifiers. The Department of Motor Vehicles (DMV), or another institution with extensive vehicle registration data, may be able to operate as the system's certificate authority.
3) Trucking Firms: In the blockchain network we're putting out, trucking firms act as clients and manage different kinds of trucks. Trucking companies may need to access the platooning records of the trucks they own in order to use them in real-world scenarios. This can involve activities like figuring out the best platoon size and minimizing fuel costs.
4) Autonomous Truck: A self-driving truck participates in mixed fleet platoons and serves as a prover by presenting identification information during the authentication procedure before joining any existing platoon.

A. DESIGN GOALS

Truck platooning, also known as convoy driving, involves a group of two or more trucks traveling closely together in a convoy, with the vehicles electronically linked to synchronize their movements. The design of truck platooning systems aims to achieve various goals related to safety, efficiency, and environmental sustainability. Here are some common design goals for truck platooning:

1) Correctness: Without knowledge of the isolated vehicle beforehand, the blockchain network can precisely verify the proof generated during the platooning formation phase.
2) Maximal Privacy-preserving: A mixed fleet vehicular network cannot allow for the exposure of an autonomous truck's sensitive and private information to anybody other than the trucking company it belongs to and the certificate authority (like the DMV).
3) Efficiency: The verification procedure is established by the use of the aggregated zero-knowledge proof. The aggregated evidence delivers quick and consistent identity verification times, with the option to vary based on the number of proofs, as opposed to sending each proof individually to the blockchain network.
4) Data Ownership: Within the open blockchain ledger, a trucking business re- retains full ownership and control over the registered truck records. This includes the ability to store, read, and analyze data from its own trucks as well as the power to grant other people these access rights using programmable access control policies.

VII. PROPOSED POST QUANTUM SECURE BLOCKCHAIN EMPOWERED AUTHENTICATION FRAMEWORK FOR AUTONOMOUS TRUCK PLATOONING

An authentication framework for autonomous truck platooning is crucial to ensure the security, integrity, and trustworthiness of communication and coordination among platooning trucks. Such a framework should address various security challenges associated with autonomous vehicle technology, including authentication of vehicles, secure communication, and protection against cyber threats. Here's a proposed authentication framework for autonomous truck platooning:

A. PLATOONING FORMATION PROCEDURE

The platooning formation procedure outlines the steps involved in the establishment and operation of a truck platoon, where multiple trucks travel closely together in a convoy while maintaining safe distances and synchronized movements. Here's an overview of the platooning formation procedure. We have used notations, and symbols given in the Table (1) to construct the platooning formation technique. In this subsection, we have discussed an overview of a generic procedure for establishing a platooning formation.

1) Utilizing specialized short-distance communications, a single vehicle has the ability to travel within a range of 300 meters that is already covered by the communication capabilities of an existing platoon. In order to integrate the solitary truck into the current platoon, information is exchanged between vehicles on a one-to-one basis.
2) In the event that a platoon commander determines that the independent truck is a viable candidate for inclusion, they initiate the initial catchup phase. This phase involves accelerating to narrow the distance between the truck and the platoon, while simultaneously initiating transmission of an authentication request to the blockchain network.

Suppose we have a fleet of trucks consisting of $\bar{G} = \{1, \ldots, i, j, \ldots, G\}$. In this fleet, truck-j follows truck-i, while truck-1 acts as the leader/commander of the fleet. The equation (1) represents the temporal relationship between two trucks at time point $\Delta\tau$. Each vehicle undergoes changes in its position relative to one another over time. The equation (2) is utilized to calculate the change of position within the time interval $\Delta\tau$.

$$
\begin{aligned}
&R_{ij}(\tau + \Delta\tau) = R_{ij}(\tau) + [R_i(\tau + \Delta\tau) - R_j(\tau + \Delta\tau)].\\
&R_i(\tau + \Delta\tau) = \vartheta_i(\tau) + \frac{1}{2}\alpha_i(\tau)(\Delta\tau)^2, 0\\
&\le \vartheta_i^{min} \le \vartheta_i(\tau) \le \vartheta_i^{max} \qquad (1)
\end{aligned}
$$

TABLE 1. Variables in the platooning formation procedure.

Variable	Description
G	Denotes counting of trucks in the platoon
P	maximum possible radius for V2V communications
X	The Standalone Truck
$\mathrm{R_i}(\tau)$	GPS of $\mathrm{i^{th}}$ truck with time τ
$\mathrm{R_{ij}}(\tau)$	$\mathrm{i^{th}}$ truck position relative to $\mathrm{j^{th}}$ truck at time τ
$\vartheta_\mathrm{i}(\tau)$	Velocity of $\mathrm{i^{th}}$ truck at time τ
$\vartheta_\mathrm{i}^\mathrm{min}$	minimum possible velocity $\mathrm{i^{th}}$ truck
$\vartheta_\mathrm{i}^\mathrm{max}$	maximum possible velocity of $\mathrm{i^{th}}$ truck
$\alpha_\mathrm{i}(\tau)$	produced acceleration for $\mathrm{i^{th}}$ truck at time τ
$\alpha_\mathrm{i}^\mathrm{min}$	minimum possible acceleration for $\mathrm{i^{th}}$ truck
$\alpha_\mathrm{i}^\mathrm{max}$	maximum possible acceleration for $\mathrm{i^{th}}$ truck
γ	time taken for authentication (1st catch-ups)
t	time of smooth driving
T	time consumed for making platoon

During the 1^{st} catch-up phase, when a single truck X and the overall authentication time γ are considered, the platoon's relative position with respect to the standalone truck will decrease from P to $R_{iX}(\tau + \gamma)\ \forall_i \in \bar{G}$, as per Equation 1. The value of $R_{iX}(\tau + \gamma)$ can be calculated using equation (3).

$$R_{iX}(\tau + \gamma) = P - [R_X(\tau + \gamma) - R_i(\tau + \gamma)], \forall_i \in \bar{G}. \quad (2)$$

After the verification process is finished, the standalone truck will be included in platoon $\bar{G}$ roster for the upcoming stage, which will entail collaborative driving. Essentially, there are three main approaches to forming a platoon; (1) the standalone truck persists in accelerating during the 2^{nd} maneuver to catch up with the group, (2) the platoon slows down to let the lone truck catch up, which is known as the slow-down tactic, and (3) in the hybrid approach, the platoon and the lone truck work together to advance towards a central position. The relationship between equations (1) and (2), along with the three platooning formation tactics, can be utilized to establish the kinematic equations for the period of cooperative driving, as stated in equation (4). The vehicle pair that reaches the goal platoon state at a slower pace will determine the shortest amount of time (t) required.

$$t = \begin{cases} max(\{\dfrac{\Delta R_{1i}}{\Delta\vartheta_i^{max} - \Delta\vartheta_{1i}}\}), & \forall_i \in \bar{G} \quad 2^{nd} catch \\ max(\{\dfrac{\Delta R_{iG}}{-\Delta\vartheta_i^{max} - \Delta\vartheta_{iG}}\}), & \forall_i \in \bar{G} \quad slow \\ max(\{\dfrac{\Delta R_{iG}}{\Delta\vartheta_G^{max} - \Delta\vartheta_i^{max} - \Delta\vartheta_{iG}}\}), & \forall_i \in \bar{G} \quad hybrid. \end{cases} \quad (3)$$

Consider $\Delta R_l X \gg \Delta R_l i$ and $\Delta R_X G \gg \Delta R_i G\ (i \in \bar{G}, X \neq i)$ Equation 4 may be condensed to the following when used in the cooperative driving step:

$$t = \begin{cases} max(\{\dfrac{\Delta R_{1X}}{\Delta\vartheta_X^{max} - \Delta\vartheta_{1X}}\}), & 2^{nd} catch \\ max(\{\dfrac{\Delta R_{XG}}{-\Delta\vartheta_X^{max} - \Delta\vartheta_{XG}}\}), & slow \\ max(\{\dfrac{\Delta R_{XG}}{\Delta\vartheta_G^{max} - \Delta\vartheta_X^{max} - \Delta\vartheta_{XG}}\}), & hybrid \end{cases} \quad (4)$$

$\Delta R_1 X$ and $\Delta R_X G$ are calculated as:

$$\begin{aligned} \Delta R_1 X &= [R_1(\tau) - R_X(\tau)] - [R_1(\tau + \gamma) - R_X(\tau + \gamma)] \\ &= R - [\vartheta_1(\tau) - \vartheta_X(\tau)]\gamma - \frac{1}{2}[\alpha_1(\tau) - \alpha_X(\tau)]\gamma^2 \end{aligned} \quad (5)$$

$$\begin{aligned} \Delta R_X G &= [R_X(\tau) - R_G(\tau)] - [R_X(\tau + \gamma) - R_G(\tau + \gamma)] \\ &= R - [\vartheta_X(\tau) - \vartheta_G(\tau)]\gamma - \frac{1}{2}[\alpha_X(\tau) - \alpha_G(\tau)]\gamma^2 \end{aligned} \quad (6)$$

It can be observed that the complete duration of end-to-end authentication time, denoted as 0, encompasses the entire process starting from generating one-time signature up to verifying aggregated signature. The total time spent during the platooning formation phase is represented by Equation 7, as shown at the bottom of the next page, taken from [44].

B. AGGREGATION OF SIGNATURES FOR BLOCKCHAIN

Aggregation of signatures in the context of blockchain refers to the process of combining multiple individual signatures into a single aggregated signature. This method offers various benefits, including better efficiency, small transaction size, improved scalability, and privacy. In transactions performed on blockchain, multiple nodes often sign one transaction to authorize it. Each of nodes puts signature using its private key rather than including all individual signatures separately, all the signatures are aggregated into one compact signature. Finally, signature aggregation is an advanced technique that improve efficiency, scalability, and privacy of blockchain transactions. The aggregation of signatures uses five algorithms $Key - Gens$, $ProofG$, $Proof - Agg$, $Verifier - Key - Aggregation$, $Verification$, respectively.

1) KEY-GENS

In cryptography, a trapdoor function is a function that is easy to compute in one direction but difficult to invert without special information called the trapdoor. The concept of a trapdoor function is often used in the generation of public-private key pairs, where the trapdoor information allows for efficient computation of the private key from the public key. The key generation process starts with the selection of a trapdoor function that exhibits the properties of being easy to compute in one direction and hard to invert without the trapdoor information.

This inputs n, and q, and uses $TrapG(q,\ n)$ to compute $\xi_0 \leftarrow \mathbb{Z}_q^{n\times m}$, with basis τ_{ξ_0} satisfying $||\tilde{\tau}_{\xi_0}|| \leq O(\sqrt{n\ log(q)})$. It also uses $H_2\ :\ \{0,\ 1\}^* \rightarrow \mathcal{C} = \{\rho \in \mathbb{R}\ :\ ||\rho||_1 = d,\ ||\rho||_\infty = 1\}$ be random oracle with d such that $|\mathcal{C}| > 2^{2.\kappa}$, where κ is the security parameter.

1) Inputs n, and q, then uses $TrapG(q,\ n)$, and it generates $\xi_0 \leftarrow \mathbb{Z}_q^{n\times m}$.
2) It samples τ_{ξ_0} satisfying the condition $||\tilde{\tau}_{\xi_0}|| \leq O(\sqrt{n\ log(q)})$.

This algorithm takes input $\xi_1, \xi_2, \xi_3, \ldots, \xi_n$ for different nodes $(U_1, U_2, U_3, \ldots, U_n)$, and takes help of $Extracts - \xi$, to get $\tau_{\xi_1'} \leftarrow Extracts - \xi(\tau_{\xi_0},\ \xi_1' = \xi_0|\xi_1)$, $\tau_{\xi_2'} \leftarrow Extracts - \xi(\tau_{\xi_0},\ \xi_2' = \xi_0|\xi_2)$, $\ldots$, $\tau_{\xi_n'} \leftarrow Extracts - \xi(\tau_{\xi_0},\ \xi_n' = \xi_0|\xi_n)$, and finally it computes the private/public key $(s\kappa_1 = s_1 \leftarrow \tau_{\xi_1'}, \varrho\kappa_1 = \xi_1'.s_1)$, $(s\kappa_2 = s_2 \leftarrow \tau_{\xi_2'}, \varrho\kappa_2 = \xi_2'.s_2)$, $\ldots$, $(s\kappa_n = s_n \leftarrow \tau_{\xi_n'}, \varrho\kappa_n = \xi_n'.s_n)$ for every user (see algorithm (1)).

Algorithm 1 Key-Gens(n)

```
Input: $\xi_0, \tau_{\xi_0}, \xi_1, \xi_2, \cdots, \xi_n$
Output: $(s\kappa_1,\ \varrho\kappa_1),\ (s\kappa_2,\ \varrho\kappa_2),\ \cdots,\ (s\kappa_n,\ \varrho\kappa_n)$
for $i \leftarrow 1; i \leq n$ do
    $\tau_{\xi_i'} \leftarrow Extracts - \xi(\tau_{\xi_0},\ \xi_i' = \xi_0|\xi_i)$,
    $i \leftarrow i + 1$
end for
for $i \leftarrow 1; i \leq n$ do
    $s\kappa_i = s_i \leftarrow \tau_{\xi_i'}$,
    $\varrho\kappa_i \leftarrow \xi_i'.s_i$,
    $i \leftarrow i + 1$
end for
RETURN $((s\kappa_1,\ \varrho\kappa_1),\ (s\kappa_2,\ \varrho\kappa_2),\ \ldots,\ (s\kappa_n,\ \varrho\kappa_n))$
```

2) PROOFG

The ProofG is run on the inputs $s\kappa_i$, and the message to compute signature σ_i. The algorithm $ProofG(s\kappa_i,\ \mu_i)$ samples $y_i \leftarrow D$, and computes $u_i = [\xi_i'].y_i \in Q_q^k$, $\rho_i = SHA2(u_i,\ \varrho\kappa_i,\ \mu_i)$, and $z_i = s_i.\rho_i + y_i$ with probability $(1 - P_{rej})$, where $P_{rej} = min\{1, \frac{D_{s_i}^{\ell+k}(z_i)}{\mu}.D_{\rho_i.s_i,\ s_i}^{\ell+k}(z_i)\}$. The final signature is $\sigma_i = (u_i,\ z_i)$ for $i \in \{1,\ 2,\ \ldots,\ n\}$ (see algorithm (2)).

3) PROOF-AGG

This algorithm takes $\sigma_1, \sigma_2, \ldots, \sigma_{n-1}, \sigma_n$, and public keys $\varrho\kappa_1,\ \varrho\kappa_2,\ \ldots,\ \varrho\kappa_j$, and messages $\mu_1,\ \mu_2,\ \ldots,\ \mu_n$, and ouputs aggregation (σ) (see Algorithm (3)).

4) VERIFIER-KEY-AGGREGATION

This algorithm needs all the verifier keys to verify the aggregation of signatures. This takes the public key $\varrho\kappa \leftarrow (\varrho\kappa_1,\ \varrho\kappa_2,\ \ldots,\ \varrho\kappa_n)$ to verify the aggregate signature (σ), where $\varrho\kappa_1 = \xi_1'.s_1$, $\varrho\kappa_2 = \xi_2'.s_2$, $\ldots$, $\varrho\kappa_n = \xi_n'.s_n$, respectively.

Algorithm 2 ProofG($s\kappa_i,\ \mu_i$)

```
Input: $s\kappa_i,\ \mu_i$
Output: $\sigma_i = (u_i,\ z_i)$
for $i \leftarrow 1; i \leq n$ do
$y_i \leftarrow D$,
$u_i = [\xi_i'].y_i \in Q_q^k$, $\rho_i = SHA2(u_i,\ \varrho\kappa_i,\ \mu_i)$,
$z_i = s_i.\rho_i + y_i$
end for
if $1 - P_{rej} = 1 - min\{1, \frac{D_{s_i}^{\ell+k}(z_i)}{\mu_i}.D_{\rho_i.s_i,s_i}^{\ell+k}(z_i)\}$ then
    return $\sigma = (u_i,\ z_i)$,
else
    RESTART
end for
```

Algorithm 3 Proof-Agg

```
Input: $\varrho\kappa = \varrho\kappa_j,\ \mu = \mu_j$ for $j \in \{1,\ 2,\ \ldots,\ N\}$
Output: $\sigma_{agg} = ((u_j)_j, z)$
for $j \leftarrow 1; j \leq n$ do
    $\rho_j = SHA2(u_j,\ \varrho\kappa_j,\ \mu_j)$
end for
for $j \leftarrow 1; j \leq n$ do
    for $j \leftarrow 1; j \leq n$ do
        $e_j = SHA2(\rho_1,\ \rho_2,\ \ldots,\ \rho_N,\ j)$
        $z = \sum_j e_j.z_j$ where $||z|| \leq \xi = O(\sqrt{N}B)$
    end for
    RETURN $\sigma = ((u_j)_j, z)$
end for
```

5) VERIFICATION

This algorithm needs verifier key $\varrho\kappa \leftarrow (\varrho\kappa_1,\ \varrho\kappa_2,\ \ldots,\ \varrho\kappa_n)$, messages $\mu = \mu_j$ for all $j \in [N]$, and aggregation σ, and it returns true or false. This algorithm executes $\rho_j = SHA2(u_j,\ \varrho\kappa_j,\ \mu_j)$ for $j \in [N]$, and finds $e_j \leftarrow SHA2(\rho_1,\ \rho_2,\ \ldots,\ \rho_N,\ j)$ for all index $j \in [N]$, and executes $\varrho\kappa.z = \sum_j e_j.(\varrho\kappa_j.\rho_j + u_j)$, and if $||z|| \leq \xi = O(\sqrt{N}B)$, then only returns true (see algorithm (4)).

C. BLOCKCHAIN NETWORK WITH ACCESS CONTROL POLICY

Within our architectural design, the blockchain plays a crucial role as the primary validator and provides a decentralized ledger that contains essential information about truck platooning along with the verifier's keys. The information stored

$$\text{“}T = \gamma + t = \begin{cases} \gamma + \dfrac{P - [\vartheta_1(\tau) - \vartheta_X(\tau)]\gamma - \frac{1}{2}[\alpha_X(\tau) - \alpha_1(\tau)]\gamma^2}{\Delta\vartheta_X^{max} - \Delta\vartheta_{1X}}, & 2^{nd}\ catch. \\ \gamma + \dfrac{P - [\vartheta_G(\tau) - \vartheta_X(\tau)]\gamma - \frac{1}{2}[\alpha_X(\tau) - \alpha_G(\tau)]\gamma^2}{\Delta\vartheta_X^{max} + \Delta\vartheta_{XG}}, & slow. \\ \gamma + \dfrac{P - [\vartheta_G(\tau) - \vartheta_X(\tau)]\gamma - \frac{1}{2}[\alpha_X(\tau) - \alpha_G(\tau)]\gamma^2}{\Delta\vartheta_X^{max} + \Delta\vartheta_{XG} - \Delta\vartheta_G^{max}}, & hybrid''. \end{cases} \tag{7}$$

Algorithm 4 Verification

```
Input: $\varrho\kappa = \varrho\kappa_j, \sigma =< \sigma_1, \sigma_2, \ldots, \sigma_n >$
Output: Accept/Reject
for $i \leftarrow 1; i \leq n$ do
    for $i \leftarrow 1; i \leq n$ do
        $\rho_j = SHA2(u_j, \varrho\kappa_j,\ M_j)$
end for
$e_j = SHA2(\rho_1,\ \rho_2,\ \ldots,\ \rho_N,\ j)$
$\varrho\kappa.z = \sum_j e_j.(\varrho\kappa_j.\rho_j + u_j)$
if then
    $||z|| \leq \xi = O(\sqrt{N}B)$
return ACCEPT
else
    REJECT
end for
```

on the blockchain about platoons could be utilized by a company to measure the advantages. Additionally, platooning provides added benefits in terms of efficiency and safety. Our platform provides customizable access control measures that determine which organizations have permission to access the on-chain information. These regulations are enforced by our blockchain network to protect the sensitive data stored in a specific truck's platoon records. The recommended access control policies are derived from the Hyperledger Fabric's access control lists (ACLs), which link policies to resource accessibility. The fine-grained, attribute-based access control policy language specified in XACML can be seen as similar to ACL. By default, a trucking company is only allowed to view the platoon records for its own trucks. To prevent potential adversaries from conducting a truck profiling attack and reconstructing the driving path of vehicles, the position information has been removed from these records. Our access control system comprises several specific elements:

1) Participant: The access control process is detailed in a list of the organizations involved.
2) Mission: Our blockchain platform facilitates both READ and WRITE operations, with data being fundamentally unalterable.
3) Resource: This demonstrates the scope of the access control policy in terms of ledger data. The on-chain assets within our platform consist of platoon records and verifier keys.
4) Condition: The condition encompasses the conditional statements that apply to multiple variables. Our system has the capability to accommodate combinations of multiple conditional logic statements, enabling the creation of intricate access control policies.
5) Action: After implementing the access control policy, the outcome is displayed as the ultimate decision. This decision can be either DENY or ALLOW.

The planned access control system grants truck companies complete authority over their platoon records and enables them to determine the individuals who can access these records. Through this system, we ensure that each participant's privacy is safeguarded effectively.

VIII. SECURITY ANALYSIS

The signature aggregation paradigm in cryptography is intended to improve the efficiency and scalability of signature systems, particularly multi-signature and threshold signature frameworks. This novel approach combines multiple individual signatures into a single concise signature that encapsulates a group of participants' collective consent or authorization. This eliminates the overhead of processing multiple signatures independently, simplifying verification operations and reducing storage needs. The protocol works as follows:

- The purpose is to efficiently merge N unique signatures to create a single aggregate signature. This aggregated signature is then utilised for verification, which simplifies the entire process.
- The adversary $\mathcal{A}$ has $(N - 1)$ verification keys that it completely controls. This allows $\mathcal{A}$ to maintain and employ these keys as needed inside the protocol.
- When a challenge is provided, $\mathcal{A}$ uses the signature oracle key $\varrho\kappa_N$ to create suitable replies. This relationship with the Oracle key guarantees that the aggregated signature may be correctly checked against the challenge.

By combining several signatures into a single, compact signature, the technique dramatically decreases the computational and transmission cost involved with validating numerous separate signatures. This aggregation approach is especially useful in dispersed networks and contexts where efficiency and scalability are critical.

1) **Setups**: After a challenge key $\varrho\kappa_N^*$ is presented by a forger $\mathcal{A}$, $\mathcal{A}$ can request numerous polynomial-time signatures that can be verified using $\varrho\kappa_N^*$.
2) **Response**: For each $j \in \{1, 2, \ldots, N\}$, the adversary $\mathcal{A}$ computes an aggregation σ and a message $(\mu = \mu_j)$ using a verification key $(\varrho\kappa = \varrho\kappa_j^*)$.
3) **Output**: The adversary succeeds if the aggregate σ is valid for the supplied key-message pair $(\varrho\kappa_j^*, M_j)$ for $j \in [N]$; otherwise, $1 \leftarrow Proof - Agg(\varrho\kappa, M, \sigma_{agg})$. To replicate the framework accurately, $\mathcal{A}$ cannot present a query for an already evaluated pair $(\varrho\kappa_N^*, M_N)$ that is existing in the list.

Theorem 11: The proposed authentication framework, which leverages blockchain technology and is secure against quantum threats, ensures unforgeability in the context of chosen message attacks for autonomous truck platooning.

Proof: This paper examines security through the lens of games $\mathcal{Q}_0$, $\mathcal{Q}_1$, and $\mathcal{Q}_2$. Ultimately, the security proof employs two iterations, referred to as the first fork and the second fork, of the Forking Lemma. A comprehensive analysis of the security proof is provided below.

- $\mathcal{Q}_0$: The $\mathcal{A}$ assumes $N_q = N_{H_1} + N_{H_2} + N_S$, and permitted to send at most N_{H_1} queries to H_1, N_{H_2} queries to H_2, and N_S queries to the signing oracle on $\varrho\kappa_N$, respectively. Suppose C is the space containing possible challenges for signatures. Let us assume $\mathcal{B}$ is simulation for arbitrary $h_i,\ h_j \leftarrow U(C)$ for all $j \in [N_q]$, and it is maintaining list HT_1 for H_1. At the starting, the list HT_1 is empty, and $\mathcal{B}$ takes care of list HT_2 for H_2, and initially HT_1, and HT_2 are empty.

1) **Setup**: Initially, $\mathcal{B}$ generates $(s\kappa_N,\ \varrho\kappa_N)$ using the security parameter κ and the $Key - Gen(1^{\kappa})$ procedure. Finally, $\mathcal{B}$ provides the public key $\varrho\kappa_N$ to $\mathcal{A}$.
2) **Queries on** H_1: When $\mathcal{A}$ sends a query on $g = \langle u,\ \varrho\kappa,\ \mu\rangle, HT_1$ searches for $g = \langle u,\ \varrho\kappa,\ \mu\rangle$ in the existing list. If $HT_1[g]$ is found in the list, it is returned; otherwise, a random value $\rho = HT_1[g] \leftarrow U(C)$ is chosen, saved in the list, and then sent.
3) **Queries on** H_2: When $\mathcal{A}$ sends a query on the tuple $g = \langle\rho_1,\ \rho_2,\ \ldots,\ \rho_N,\ j\rangle$ for $j \in [N]$, HT_1 first checks for its existence in the list. If not found, it selects $\rho = HT_1[g] \leftarrow U(C)$, saves it in the list, and returns it.
4) **Sign Query**: $\mathcal{B}$ faithfully executes the signing process with $s\kappa_N$ for μ.
5) **Forgery**: If $\mathcal{A}$ is capable of computing a forgery $\sigma_{agg} = ((u_j)_j,\ z)$ on $\mu = \mu_j$ for all indices $j \in [N]$, and public keys $\varrho\kappa = \varrho\kappa_j$ for all indices $j \in [N]$, where $c = HT_1$ follows the programming on $g = (u_N,\ \varrho\kappa_N,\ \mu_N)$ with $\rho = h_{ji}$ for $j,\ i$. If this condition does not hold, then $\mathcal{A}$ makes a guess on c_N with probability $(1 - \frac{1}{C})$.

- $\mathcal{Q}_1$: The game $\mathcal{Q}_1$ closely resembles $\mathcal{Q}_0$, with the distinction that the signature process of $\mathcal{B}$ is dishonest. Instead, it simulates messages without employing $s\kappa_N$. During the **Signing** phase, $\mathcal{B}$ selects $c \leftarrow U(C)$, $z \leftarrow D_s^{\ell+k}$, and μ, then computes $u = B_1'.z - t_N.\rho$ and applies programming on $c = HT_1[g]$ with $g = (u,\ t_N,\ \mu)$. Ultimately, it outputs $\sigma = (u,\ z)$ with an advantage of $\frac{1}{\mu}$.
- $\mathcal{Q}_2$: The game $\mathcal{Q}_2$ is nearly identical to $\mathcal{Q}_1$, except for the key generation process of $\mathcal{B}$. $\mathcal{B}$ selects $t_N \leftarrow U(R_q^k)$ with $\varrho\kappa_N = t_N$, and provides the public key $\varrho\kappa_N$ to $\mathcal{A}$. If the signing process can be completed without $s\kappa_N$, then $\mathcal{B}$ substitutes $\varrho\kappa_N$ with a random value. Consequently, we obtain

$$|Pr[\mathcal{Q}_2 - \mathcal{Q}_1]| \le Adv_{MLWE}$$

We will now demonstrate that $Adv - AggSign_{\mathcal{A}} \le Pr[\mathcal{Q}_2] + Adv_{MLWE_{\kappa,\ \ell,\ \xi}} + negl(\kappa)$ by employing the Forking lemma [43] twice on $A' = [A|t_N]$ within $\mathcal{Q}_2$, where A' is randomly selected from a distribution.

1) **Fork (1)**: In order to apply the Forking Lemma [43], we require a simulation $\mathcal{C}$ with input $Gen(1^{\kappa})$ and output denoted as A'. As per [43], we define Acc($\mathcal{B}$) to represent the probability of $\mathcal{B}$ achieving success, and $Fork_{\mathcal{B}}$ denotes the forking advantage of $\mathcal{F}_{\mathcal{B}}$. To obtain the desired outcome, we define the outset as $(j_f,\ VK,\ \mu,\ \sigma_{agg},\ C,\ E)$ and provide inputs $\mathcal{B}$ along with random coins $h_1',\ h_2',\ \ldots,\ h_{N_q}'$. In the event that the fork succeeds for index ctr_{ρ}, the forking process $\mathcal{F}_{\mathcal{B}}$ generates two distinct values $(out,\ \tilde{out}) \neq (\perp,\ \perp)$ with an advantage of $Fork(\mathcal{B})$, where $Acc_{\mathcal{B}} \le |\mathcal{C}|/N_q$. Let's assume $out = (j_f,\ VK,\ \mu,\ \sigma_{agg},\ C,\ E)$ and $\tilde{out} = (\tilde{j_f},\ \tilde{VK},\ \tilde{\mu},\ \tilde{\sigma}_{agg},\ \tilde{C},\ \tilde{E})$. Both simulated algorithms $\mathcal{B}$ and $\mathcal{A}$ lack access to any confidential information and are rewound during the fork. They operate in a similar manner prior to the forking step, with $t_j = \tilde{t}_j,\ u_j = \tilde{u}_j,\ \rho_j = \tilde{\rho}_j$ for all indices $j \in \{1, 2, \ldots, N\}$, and we find $e_j = \tilde{e}_j$ for all indices $j \in \{1, 2, \ldots, N-1\}$, owing to the fact that the arbitrary random choices made by $\mathcal{B}$ result in identical behavior during execution, and H_e is independent of (h_j') for all indices $j \in [N_q]$ except $j \neq N$. Additionally, it yields $\mu = \tilde{\mu}$, enabling verification of forgery in both scenarios, where $||z||_2,\ ||\tilde{z}||_2 < B'$, and $[A|I_k]z = \sum_{j\in[N]} e_j(t_j\rho_j + u_j)$ and $[A|I_k]z = \sum_{j\in[N]} \tilde{e}_j(\tilde{t}_j\tilde{\rho}_j + \tilde{u}_j)$. Consequently, we obtain

$$[A|Ik].(z - \tilde{z}) = (e_N - \tilde{e}_N)(t_N\rho_N + u_N) \qquad (8)$$

However, the above simulation is unable to extract the right vector while meeting the Module's Short Integer Solution assumption, necessitating the use of a second fork.

2) **Fork (2)**: In fork (2), we must simulate algorithm $\mathcal{D}$ based on $\mathcal{C}$, which takes $Gen(1^{\kappa})$ and yields A' in the context of Forking [43]. Let's denote $Acc(\mathcal{C})$ as the advantage of $\mathcal{C}$ and $Fork(\mathcal{C})$ as the advantage of $\mathcal{F}_{\mathcal{C}}$, as outlined in [43]. It's evident that $Acc(\mathcal{C}) = Fork(\mathcal{B})$, and we define $out = (VK,\ \mu,\ \sigma_{agg},\ C,\ E)$, with inputs $\mathcal{C}$ along with random coins $h_1,\ h_2,\ \ldots,\ h_{N_q}$, and counter ctr_{ρ}. Thus, $\mathcal{F}_{\mathcal{C}}$ yields output $(out,\ \tilde{out}) \neq (\perp,\ \perp)$ with $Acc(\mathcal{C}) \le N_q/|\mathcal{C}| + \sqrt{N_q \cdot Fork(\mathcal{C})}$, where $out = (VK,\ \mu,\ \sigma_{agg},\ C,\ E)$, and $\tilde{out} = (\tilde{VK},\ \tilde{\mu},\ \tilde{\sigma}_{agg},\ \tilde{C},\ \tilde{E})$ with $\mu_j = \mu_j',\ u_j = u_j'$, and $t_j = t_j'$ for $j \in [N]$, and $\rho_j = \rho_j'$ for $j \in [N-1]$. Hence, we obtain

$$[A|I_k].(z' - \tilde{z}') = (e_N' - \tilde{e}_N')(t_N\rho_N' + u_N) \qquad (9)$$

Now, by multiplying (1) with $\epsilon' = (e_N' - \tilde{e}_N')$, and (2) by $\epsilon = (e_N - \tilde{e}_N')$, we obtain $[A|I_k].(\epsilon'(z - \tilde{z}) - \epsilon(z' - \tilde{z}')) = \epsilon.\epsilon'(t_N(\rho_N - \rho_N'))$. Therefore, $\mathcal{D}$ computes the tuple $x = (\epsilon'(z - \tilde{z}) - \epsilon(z' - \tilde{z}')) = \epsilon.\epsilon'(t_N(\rho_N - \rho_N'))$, which is a correct vector satisfying the M-SIS assumption for $A' = [A|t_N]$, and its Euclidean norm is $||x||_2 \le 4.||\epsilon.z||_2 + 2.||\epsilon^2.\rho||_2 \le 4.4d.B' + 2.4d^{\frac{3}{2}} = b$.

□

IX. PERFORMANCE

This section contains a detailed description of the experimental setup used to analyse the performance of permissioned blockchains in truck platooning. The analysis mainly keeps track of key performance metrics including communication throughput, and latency. The simulation environment is set up using the following key components and major parameters:

A. BLOCKCHAIN PLATFORM

We have utilized *Hyperledger Fabric* as the permissioned blockchain due to its robustness and outstanding support for smart contracts. The blockchain networking system is comprised multiple nodes, each representing a truck in the platooning of trucks.

B. NETWORKING CONFIGURATION

- **Number of Trucks:** 12
- **Communication Protocol:** Vehicle-to-Vehicle (V2V) using Dedicated Short-Range Communications (DSRC)
- **Simulation Duration:** 50 minutes
- **Simulation Tool:** Simulation of Urban Mobility (SUMO) for traffic simulations, integrated with Hyperledger Fabrics for blockchain operation.

C. BLOCKCHAIN CONFIGURATION

- **Block Interval:** 2-4 seconds
- **Transaction Throughput:** Transactions per second (TPS)
- **Smart Contracts:** Contains platooning formation, exit protocol, throughputs and latencies

In this section, Algorithm (5) is responsible for adding a new platooning provider to the contract, Algorithm (6) verifies existence of provider in the contract, Algorithm (7) asks a truck for providing consent for data sharing, Algorithm (8) verifies if a truck has given consent, Algorithm (9) gives permission to access truck data according to its consent, and Algorithm (10) is consensus/mining to validate transaction and generate new block with Proof of Work consensus algorithm.

```
Algorithm 5 Add Provider
Input: Truck Company's address _provAddress, Provider's name _name, Provider's role _role
Output: None
newProvider ← TruckPlatooningProvider(_name, _role)
providers[_provAddress] ← newProvider
```

```
Algorithm 6 Check Provider Existence
Input: Provider's address _provAdd
Output: True if provider exists, False otherwise
if providers[_provAdd].role is not empty then
    returnTrue
else
    returnFalse
end if
```

```
Algorithm 7 Give Consent
Input: Provider's address _provAdd, Truck's address _truckAddress
Output: None
truckConsent[_truckAddress][_provAdd] ← true
```

```
Algorithm 8 Check Truck Consent
Input: Provider's address _provAdd, Truck's address _truckAddress
Output: True if truck consent exists, False otherwise
if truckConsent[_truckAddress][_provAdd] is true then
    returnTrue
else
    returnFalse
end if
```

In the reference of blockchain based truck platooning, both "time" and "latency" play highly important roles in ensuring efficiency, safety, and reliability of the system. The role of time is important for platooning coordination. Trucks in a platoon should maintain precise distance and speed relative to each other. Accuracy in time synchronization ensures that all trucks can coordinate their activities effectively, and they are able to reduce the risk of collisions and improving overall platoon stability. In a blockchain framework, time synchronization is must for ensuring consensus. Accuracy in timestamps help maintain the correct order of transactions, and it ensures consistency and integrity for the distributed ledger. The time needed to execute transactions on the blockchain affects the timeliness of data update. Quick processing ensures truck positions and speeds is updated in real-time, which is crucial for maintaining the platoon. The number of transactions executed per second sometimes called throughput affects the blockchain's ability to handle large data from multiple trucks. High throughput means more transactions executed per second is necessary in managing the continuous data flow from all participating trucks. Smart contracts automates responses based on specific condition. The execution time of these contracts affects how fast these actions can be executed, which is critical for the safety and

```
Algorithm 9 Access Truck Data
Input: Provider's address _provAdd, Truck's address _truckAddress
Output: Access to truck data
if truckConsent[_truckAddress][msg.sender] is true then
    Access truck data
else
    Output an error: "Truck has not send consent for accessing data"
end if
```

Algorithm 10 Proof-of-Work Mining Consensus

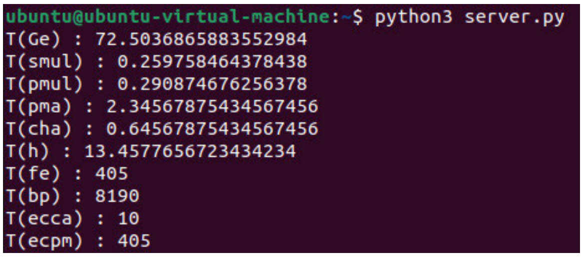
```
Input: Previous Block hash $previousHash$, Data for the block $data$
Output: Nonce $nonce$, Current Hash $currentHash$
$nonce \leftarrow 0$
$difficulty \leftarrow 5$ {Adjusting difficulty as required}
$prefix \leftarrow$ new strings of $difficulty$ zeros
while true do
    $hashInput \leftarrow$ keccak256(abi.encodePacked($previousHash$, $data$, $nonce$))
    $hash\text{-}Bytes \leftarrow$ bytes($hashInput$)
    $validPrefix \leftarrow$ true
    for $i \leftarrow 0$ to $difficulty$ - 1 do
        if $hash\text{-}Bytes[i] \neq prefix[i]$ then
            $validPrefix \leftarrow$ false
            break
        end if
    end for
    if $validPrefix$ then
        $currentHash \leftarrow hInput$
        break
    end if
    $nonce \leftarrow nonce + 1$
end while
```

TABLE 2. Device configuration.

Components	Details
Processor	Intel(R) Core(TM) i7-6700 CPU @ 3.4GHz
System Type	x64-based pc
Graphics	NVIDIA GeForce RTX 2060
RAM	8GB
Clock speed	1.00 GHz
Crypto Libraries	Miracle, Charm-crypto, NumPy, hash lib
Environment	Python 3.9.0
Cores	8
Operating System	Windows 10 Professional

efficiency of the truck platoon. Low latency is required for smooth communication between trucks in a platoon. Delays in communication leads to slower response, and increases the risk of collisions and reducing the overall efficiency of the platoon. An HP laptop with Linux(ubuntu) operating system (Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz and 8GB RAM) will be a server (see Table (2)). We have also included cost of various operations in the Figure (3), and hypothetical key generation time in the Figure (4).

Table (2) contains server configuration and Table (3) contains device configuration respectively. Table (3) shows output for latency, and complexity, and Table (3) shows a good relation between the increment in the number of blocks, and the consumption of time, and growth rates is linear, it is very helpful in practical application which involves blockchain. This paper contains notations for schemes as [28] by [A], [29] by [B], [30] by [C], [31] by [D], [32] by [E], and the introduced framework by [F], respectively.



```
ubuntu@ubuntu-virtual-machine:~$ python3 server.py
T(Ge) : 72.5036865883552984
T(smul) : 0.259758464378438
T(pmul) : 0.290874676256378
T(pma) : 2.34567875434567456
T(cha) : 0.64567875434567456
T(h) : 13.4577656723434234
T(fe) : 405
T(bp) : 8190
T(ecca) : 10
T(ecpm) : 405
```

FIGURE 3. Device computation cost in nanoseconds.

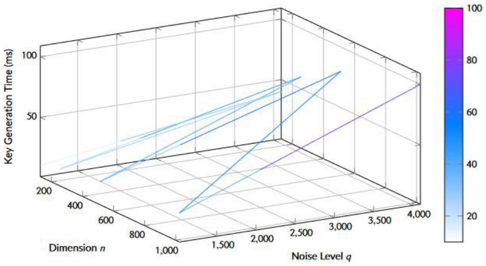


FIGURE 4. Hypothetical key generation time for RLWE: time vs dimension *n* and noise level *q*.

TABLE 3. Proposed design performance.

	Number of blocks										
	10	20	30	40	50	60	70	80	90	100	complexity
Read Latency (seconds)	12.33	14.8	15.56	18.01	24.50	25.33	27.80	28.67	31.88	34.23	$O(\log n^2)$
Transaction Latency (seconds)	46.43	78.70	111.33	173.33	208.23	260.47	317.63	352.03	371.33	410.02	$O((N-1)\log n)$
Space Complexity	$O(nN)$										
n is the number of blocks; N is the number of nodes involved in the transaction of blocks.											

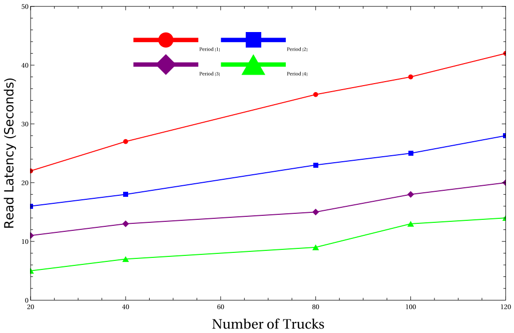


FIGURE 5. Periods vs Read Latency for proposed Frameworks.

TABLE 4. Comparison of the suggested frameworks' computing costs against similar ones.

Designes	Key-Generations	Signatures	Verification
[A] [29]	$2\tau_p + 2\tau_* + 2n\tau_l$	$2\tau_p + 2n\tau_* + 4\tau_h + 2n\tau_a$	$n\tau_p + 4n\tau_* + 2n\tau_h + n\tau_a$
[B] [30]	$4\tau_p + 4\tau_* + 2n\tau_l$	$4\tau_p + 2n\tau_* + 2\tau_h + 4\tau_a$	$n\tau_p + 4n\tau_* + 2n\tau_h + n\tau_a$
[C] [31]	$4\tau_p + 4\tau_* + n\tau_l$	$2\tau_p + 2n\tau_* + 4\tau_h + 2n\tau_a$	$n\tau_p + 4n\tau_* + n\tau_h + 2n\tau_a$
[D] [32]	$2\tau_p + 3\tau_* + \tau_l$	$2\tau_p + 4\tau_* + 2\tau_h + \tau_a + n\tau_l$	$\tau_p + 2\tau_h + (n-1)\tau_l$
[E] [33]	$\tau_p + 3\tau_*$	$2\tau_p + 4\tau_* + 2\tau_h + \tau_a + n\tau_l$	$\tau_p + 2\tau_h + (n-1)\tau_l$

Since all the nodes with number N are new to the network and able to create transactions at the same time, they all require keys, which causes the key generation time to increase. After the nodes have acquired the keys during the initial time interval, there is a reduction in latencies or almost stability in the following time intervals (see Figure(5)).

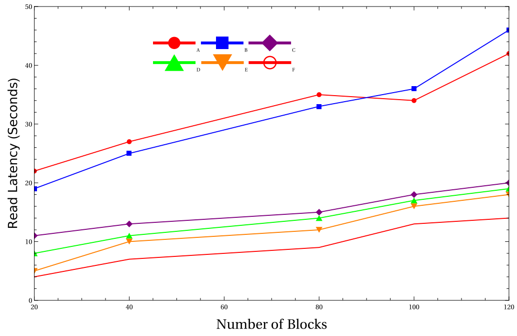


FIGURE 6. Comparison for read latencies for existing and proposed Frameworks.

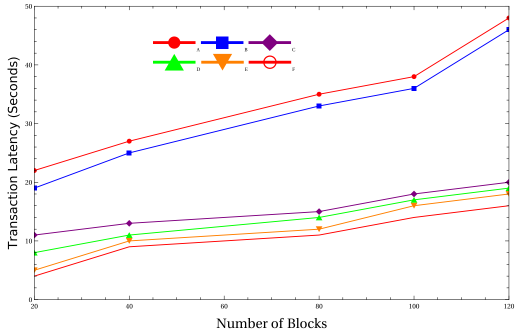


FIGURE 7. Comparison for transactions latencies for existing and proposed frameworks.

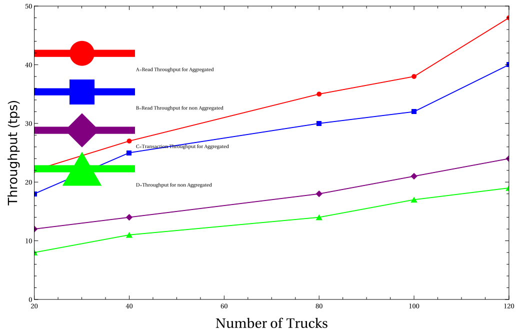


FIGURE 8. Throughputs measurement for read and overall transactions for aggregated vs non aggregated frameworks.

To perform analysis on the model, we have executed more than one transaction iteration, and its average delay initially is 31.7 seconds, whereas average latency in the second time period, with new/old keys, is 21.2 seconds indicates average of latency fall down to 14.2 seconds and 9.1 seconds with overall complexity O(1) respectively. Table (3) discuss about the storage and we found the storage overheads is lesser than the overheads complexity $O(N^n)$ of technique not supporting aggregation technique. In Figure (6), this paper have discussed read latency, and it indicates the reading latency of the proposed framework is lesser than existing techniques discussed in the paper. This method is executed for reading latency and it affects transaction latency (see Figure (7)), and this approach outperforms over existing techniques. We executed transactions, and observed read throughput and transaction throughput in this study (see Figure (8)). It illustrates how the data gathering throughput of the non-aggregated blockchain framework grows from 7 to 57 readings per second by a mean of 21.5 percent. The collective features of the proposed architecture show an important increase in capabilities, with approximately of 40 percentage more readings per second. This indicates that the proposed architecture is superior to the present blockchain structure in general. It also attempts to explore its throughput capabilities considering the processing speed of each individual transaction.

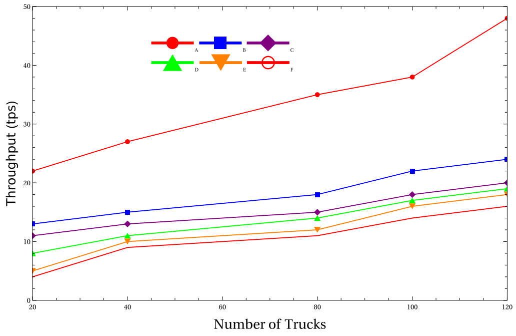


FIGURE 9. Throughputs comparison among existing frameworks.

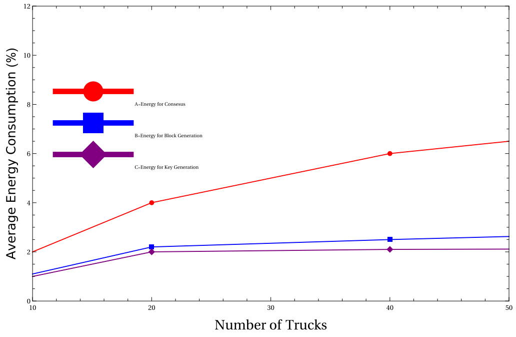


FIGURE 10. Energy consumption for consesus, block, and key generation for proposed Framework.

It can be observed the proposed framework uses more computing throughput by a factor of 62 percentage. We have also done an anlysis of transactions performance with more nodes, and its observation is given in the Figure (9). It is clear that the proposed approach is more efficient in throughput by a factor 36 percentage on average. This happens due to aggregation of multiple signatures, and the multiple nodes can be verified in one step. Most of classic blockchains consume more energy due to tedious key-generation, and blocks-generation using hashing, and consensus. Figure (10) shows a graph of energy consumption for a blockchain supporting usual public key infrastructure. The complete process of key grnrration for s particular node consumes usage of energy by a factor of 1.71 percent of the total energy. Energy usage for blockchain ranges 2.42 percentage to 6.28 percentage at the highest, with an average increment of 0.84 percentage in each step. Table(4) shows the description related phases key-generations, signatures, and verifications processes.

X. CONCLUSION

Truck platooning is a method of organizing vehicles in a convoy to improve efficiency, safety, and fuel economy. In truck platooning, multiple trucks travel closely together in a coordinated manner, with the trucks electronically linked to synchronize their movements. The leading truck in the platoon sets the pace and controls the acceleration, braking, and steering, while the following trucks automatically adjust their speed and maintain a safe following distance. The framework uses quantum safe aggregate signature for authenticity for truck platooning. This framework improves safety and enhances traffic flow. The security of the framework depends upon quantum safe assumptions (1) Module Learning With Error and (2) Module Short Integer Solution. This paper contains an analysis of a framework with high latency and throughput, and its suitability for truck platooning. We have found the cryptographic operations, generally in lattice-based systems, often result in larger key size and ciphertext compared to traditional cryptographic schemes. The increased communication overheads strain the bandwidth of vehicle-to-vehicle (V2V) networking, potentially leading to delay and increased risk of communication bottleneck.

REFERENCES

[1] S. Tsugawa, "Results and issues of an automated truck platoon within the energy ITS project," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2014, pp. 642–647.

[2] B. van Arem, C. J. G. van Driel, and R. Visser, "The impact of cooperative adaptive cruise control on traffic-flow characteristics," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 4, pp. 429–436, Dec. 2006.

[3] J. Ploeg, S. Shladover, H. Nijmeijer, and N. van de Wouw, "Introduction to the special issue on the 2011 grand cooperative driving challenge," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 989–993, Sep. 2012.

[4] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Bus. Rev.*, 2008.

[5] P. K. Sharma and J. H. Park, "Blockchain-based secure mist computing network architecture for intelligent transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5168–5177, Aug. 2021.

[6] C. Meese, H. Chen, S. A. Asif, W. Li, C.-C. Shen, and M. Nejad, "BFRT: Blockchained federated learning for real-time traffic flow prediction," in *Proc. 22nd IEEE Int. Symp. Cluster, Cloud Internet Comput. (CCGrid)*, May 2022, pp. 317–326.

[7] M. Shaygan, C. Meese, W. Li, X. Zhao, and M. Nejad, "Traffic prediction using artificial intelligence: Review of recent advances and emerging opportunities," *Transp. Res. C, Emerg. Technol.*, vol. 145, Dec. 2022, Art. no. 103921.

[8] R. Chen, Y. Li, Y. Yu, H. Li, X. Chen, and W. Susilo, "Blockchain-based dynamic provable data possession for smart cities," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4143–4154, May 2020.

[9] H. Li, L. Pei, D. Liao, G. Sun, and D. Xu, "Blockchain meets VANET: An architecture for identity and location privacy protection in VANET," *Peer-to-Peer Netw. Appl.*, vol. 12, no. 5, pp. 1178–1193, Sep. 2019.

[10] H. Guo, W. Li, and M. Nejad, "A hierarchical and location-aware consensus protocol for IoT-blockchain applications," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 3, pp. 2972–2986, Sep. 2022.

[11] H. Guo, W. Li, M. Nejad, and C.-C. Shen, "A hybrid blockchain-edge architecture for electronic health record management with attribute-based cryptographic mechanisms," *IEEE Trans. Netw. Service Manage.*, vol. 20, no. 2, pp. 1759–1774, Jun. 2022.

[12] A. P. Singh, N. R. Pradhan, A. K. Luhach, S. Agnihotri, N. Z. Jhanjhi, S. Verma, Kavita, U. Ghosh, and D. S. Roy, "A novel patient-centric architectural framework for blockchain-enabled healthcare applications," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5779–5789, Aug. 2021.

[13] H. Guo, W. Li, M. Nejad, and C.-C. Shen, "Access control for electronic health records with hybrid blockchain-edge architecture," in *Proc. IEEE Int. Conf. Blockchain (Blockchain)*, Jul. 2019, pp. 44–51.

[14] J. Partala, T. H. Nguyen, and S. Pirttikangas, "Non-interactive zero-knowledge for blockchain: A survey," *IEEE Access*, vol. 8, pp. 227945–227961, 2020.

[15] W. Li, C. Meese, H. Guo, and M. Nejad, "Blockchain-enabled identity verification for safe ridesharing leveraging zero-knowledge proof," in *Proc. 3rd Int. Conf. Hot Information-Centric Netw. (HotICN)*, Dec. 2020, pp. 18–24.

[16] W. Li, C. Meese, M. Nejad, and H. Guo, "P-CFT: A privacy-preserving and crash fault tolerant consensus algorithm for permissioned blockchains," in *Proc. 4th Int. Conf. Hot Information-Centric Netw. (HotICN)*, Nov. 2021, pp. 26–31.

[17] M. B. Mollah, J. Zhao, D. Niyato, Y. L. Guan, C. Yuen, S. Sun, K.-Y. Lam, and L. H. Koh, "Blockchain for the Internet of Vehicles towards intelligent transportation systems: A survey," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4157–4185, Mar. 2021.

[18] H. Guo, W. Li, M. Nejad, and C.-C. Shen, "Proof-of-event recording system for autonomous vehicles: A blockchain-based solution," *IEEE Access*, vol. 8, pp. 182776–182786, 2020.

[19] A. Lei, H. Cruickshank, Y. Cao, P. Asuquo, C. P. A. Ogah, and Z. Sun, "Blockchain-based dynamic key management for heterogeneous intelligent transportation systems," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1832–1843, Dec. 2017.

[20] W. Li, H. Guo, M. Nejad, and C.-C. Shen, "Privacy-preserving traffic management: A blockchain and zero-knowledge proof inspired approach," *IEEE Access*, vol. 8, pp. 181733–181743, 2020.

[21] C. Lin, D. He, X. Huang, N. Kumar, and K. R. Choo, "BCPPA: A blockchain-based conditional privacy-preserving authentication protocol for vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 12, pp. 7408–7420, Dec. 2021.

[22] Q. Feng, D. He, S. Zeadally, and K. Liang, "BPAS: blockchain-assisted privacy-preserving authentication system for vehicular ad hoc networks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 4146–4155, Jun. 2020.

[23] Y. Yao, X. Chang, J. Mišic, V. B. Mišic, and L. Li, "BLA: blockchain-assisted lightweight anonymous authentication for distributed vehicular fog services," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3775–3784, Apr. 2019.

[24] K. Kaur, S. Garg, G. Kaddoum, F. Gagnon, and S. H. Ahmed, "Blockchain-based lightweight authentication mechanism for vehicular fog infrastructure," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2019, pp. 1–6.

[25] H. Liu, P. Zhang, G. Pu, T. Yang, S. Maharjan, and Y. Zhang, "Blockchain empowered cooperative authentication with data traceability in vehicular edge computing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4221–4232, Apr. 2020.

[26] Z. Ying, L. Yi, and M. Ma, "BEHT: blockchain-based efficient highway toll paradigm for opportunistic autonomous vehicle platoon," *Wireless Commun. Mobile Comput.*, vol. 2020, pp. 1–13, Sep. 2020.

[27] C. Chen, T. Xiao, T. Qiu, N. Lv, and Q. Pei, "Smart-contract-based economical platooning in blockchain-enabled urban Internet of Vehicles," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 4122–4133, Jun. 2020.

[28] E. O. Kiktenko, N. O. Pozhar, M. N. Anufriev, A. S. Trushechkin, R. R. Yunusov, Y. V. Kurochkin, A. I. Lvovsky, and A. K. Fedorov, "Quantum-secured blockchain," *Quantum Sci. Technol.*, vol. 3, no. 3, Jul. 2018, Art. no. 035004.

[29] C.-Y. Li, X.-B. Chen, Y.-L. Chen, Y.-Y. Hou, and J. Li, "A new lattice-based signature scheme in post-quantum blockchain network," *IEEE Access*, vol. 7, pp. 2026–2033, 2019.

[30] C. Li, Y. Tian, X. Chen, and J. Li, "An efficient anti-quantum lattice-based blind signature for blockchain-enabled systems," *Inf. Sci.*, vol. 546, pp. 253–264, Feb. 2021.

[31] R. Saha, G. Kumar, T. Devgun, W. J. Buchanan, R. Thomas, M. Alazab, T. Hoon-Kim, and J. J. P. C. Rodrigues, "A blockchain framework in post-quantum decentralization," *IEEE Trans. Services Comput.*, vol. 16, no. 1, pp. 1–12, Jan. 2023.

[32] D. Chaudhary, M. S. P. Durgarao, D. Mishra, and S. Kumari, “A module lattice based construction of post quantum secure blockchain architecture,” *Trans. Emerg. Telecommun. Technol.*, vol. 35, no. 4, p. e4855, Apr. 2024.

[33] M. Ajtai, “Generating hard instances of lattice problems,” in *Proc. ACM Symp. Theory Comput.*, 1996, pp. 99–108.

[34] M. Ajtai, “Generating hard instances of the short basis problem,” in *Proc. Int. Colloq. Automata, Lang., Program.* Springer, 1999, pp. 1–9.

[35] M. Ajtai and C. Dwork, “A public-key cryptosystem with worst-case/average-case equivalence,” in *Proc. 29th Annu. ACM Symp. Theory Comput.*, 1997, pp. 284–293.

[36] K. Boudgoust and A. Roux-Langlois, “Compressed linear aggregate signatures based on module lattices,” *Cryptol. ePrint Arch.*, 2021.

[37] A. Langlois and D. Stehlé, “Worst-case to average-case reductions for module lattices,” *Designs, Codes Cryptography*, vol. 75, no. 3, pp. 565–599, Jun. 2015.

[38] C. Bonnet and H. Fritz, “Fuel consumption reduction in a platoon: Experimental results with two electronically coupled trucks at close spacing,” SAE, Warrendale, PA, USA, Tech. Rep., 2000.

[39] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell, “Bulletproofs: Short proofs for confidential transactions and more,” in *Proc. IEEE Symp. Secur. privacy (SP)*, 2018, pp. 315–334.

[40] J. Cui, L. Wei, J. Zhang, Y. Xu, and H. Zhong, “An efficient message-authentication scheme based on edge computing for vehicular ad hoc networks,” *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 5, pp. 1621–1632, May 2019.

[41] K. Hashimoto and W. Ogata, “Unrestricted and compact certificateless aggregate signature scheme,” *Inf. Sci.*, vol. 487, pp. 97–114, Jun. 2019.

[42] G. Wu, F. Zhang, L. Shen, F. Guo, and W. Susilo, “Certificateless aggregate signature scheme secure against fully chosen-key attacks,” *Inf. Sci.*, vol. 514, pp. 288–301, Apr. 2020.

[43] M. Bellare and G. Neven, “Multi-signatures in the plain public-key model and a general forking lemma,” in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, Oct. 2006, pp. 390–399.

[44] W. Li, C. Meese, H. Guo, and M. Nejad, “Aggregated zero-knowledge proof and blockchain-empowered authentication for autonomous truck platooning,” *IEEE Trans. Intell. Transp. Syst.*, pp. 1–15, 2023.

DHARMINDER CHAUDHARY (Member, IEEE) received the Ph.D. degree in cryptography and network security. He is currently an Assistant Professor (Senior Grade) with the Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Chennai, India. He has published 30 SCI/Scopus-indexed articles in the areas of cryptography and network security, Internet of Drones security, and vehicle security.

P. SANTHI (Member, IEEE) received the Ph.D. degree in networking systems. She is currently a Professor with the Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala Research and Development Institute of Science and Technology, Chennai, Tamil Nadu, India. She has published 12 SCI/Scopus-indexed articles in the areas of cryptography and network security, Internet of Drones security, and vehicle security.

M. S. P. DURGARAO (Member, IEEE) is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Chennai, India. He has published 2 SCI/Scopus-indexed articles in the areas of cryptography and network security, Internet of Drones security, and vehicle security.

A. PADMAVATHI (Member, IEEE) received the Ph.D. degree in cryptography and network security. She is currently an Assistant Professor (Selection Grade) with the Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Chennai, India. She has published 12 SCI/Scopus-indexed articles in the areas of cryptography and network security, Internet of Drones security, and vehicle security.

MOHAMMAD MEHEDI HASSAN (Senior Member, IEEE) received the Ph.D. degree in computer engineering from Kyung Hee University, South Korea, in February 2011. He is currently a Professor with the Department of Information Systems, College of Computer and Information Sciences, King Saud University (KSU), Riyadh, Saudi Arabia. He has authored or coauthored more than 340 publications, including refereed IEEE/Springer/Elsevier journals, conference papers, books, and book chapters. His research interests include edge/cloud computing, the Internet of Things, cyber security, deep learning, artificial intelligence, body sensor networks, 5G networks, and social networks.

BADER FAHAD ALKHAMEES (Member, IEEE) received the bachelor’s degree in computer science from King Saud University, Riyadh, Saudi Arabia, in 2003, the master’s degree in software systems from Heriot-Watt University, U.K., in 2008, and the Ph.D. degree in biomedical informatics from Rutgers University, NJ, USA. He is currently an Assistant Professor with the Information System Department, College of Computer and Information Sciences, King Saud University. His research interests include biomedical informatics, medical imaging and diagnosis, machine learning, fuzzy systems, cloud and edge computing, the Internet of Things, and computer networks.