2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC) | 978-1-6654-3578-9/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICBC51069.2021.9461116

Location-aware Verification for Autonomous Truck Platooning Based on Blockchain and Zero-knowledge Proof

Wanxin Li* Collin Meese* Zijia (Gary) Zhong† Hao Guo‡ Mark Nejad*
* Department of Civil and Environmental Engineering, University of Delaware, U.S.A.
†John A. Reif, Jr. Department of Civil and Environmental Engineering, New Jersey Institute of Technology, U.S.A.
‡School of Software, Northwestern Polytechnical University, Taicang Campus, China.
{wanxinli,cmeese,nejad}@udel.edu, zijia.zhong@njit.edu, haoguo@nwpu.edu.cn

***Abstract*—Platooning technologies enable trucks to drive cooperatively and automatically, which bring benefits including less fuel consumption, more road capacity and safety. In order to establish trust during dynamic platoon formation, ensure vehicular data integrity, and guard platoons against potential attackers, it is pivotal to verify any given vehicle's identity information before granting it access to join a platoon. To address this concern in dynamic truck platooning, we present a novel location-aware and privacy-preserving verification protocol based on zero-knowledge proof and permissioned blockchain. By performing the verification process within the spatially-local area defined by a given platoon, our system can provide lower latency and communication overhead compared to a location-agnostic blockchain system. We prototype the proposed system and perform benchmark tests on the Hyperledger platform. The experimental results show that our system is suitable for real-world truck platooning.**

***Index Terms*—Autonomous truck, blockchain, data privacy, identity verification, location-aware, platoon, zero-knowledge proof.**

I. Introduction

Truck platooning involves linking two or more trucks together in a convoy with a short following headway with wireless connectivity and vehicle automation. As a by-product of the short following headway, fuel efficiency is expected to improve. According to Japan ITS Energy project, 15% of fuel can be saved with a 4.7-m intra-platoon following gap at 80 km/h [1]. Truck platooning also allows the driver to disengage from driving tasks. Human error was estimated to be responsible for 94% of the traffic accident in the U.S. [2]. Compared to human drivers, automated driving systems could achieve a much shorter response time and more accurate assessment of the dynamic traffic conditions.

The security of the platooning systems, which protects them from unauthorized access to proprietary information about a specific vehicle or fleet specifications, under mixed fleet scenario has not been extensively studied thus far, though awareness of such aspect has gained increasing attention [3]–[5]. To ensure the security of the system, it is crucial to be able to verify a given vehicle's identity information prior to granting it access to join a platoon in order to establish trust, ensure the platoon integrity and guard against potential attackers.

Over the past decade, research in blockchain technology has highlighted it as a promising technology for supporting a myriad of decentralized applications between both trusted and anonymous peers [6]–[11]. In relation to dynamic truck platooning, blockchain presents some desirable properties for creating a robust, dynamic and decentralized system.

In this paper, we propose and prototype a system for identity verification in the context of dynamic truck platooning, motivated by permissioned blockchain technology and zero-knowledge proofs.

II. System Architecture

By referring to Fig. 1, we first define the following entities that take part in the proposed architecture:

- Permission Issuer: A permission issuer is a trusted entity who manages identifiers of autonomous vehicles (e.g., MAC address) and issue key pairs to data owners and data verifiers. In practice, an agency such as the Department of Motor Vehicles (DMV) can function as the permission issuer in the proposed system.
- Permissioned Blockchain: A permissioned blockchain (in our prototype using Hyperledger Fabric) is utilized as the

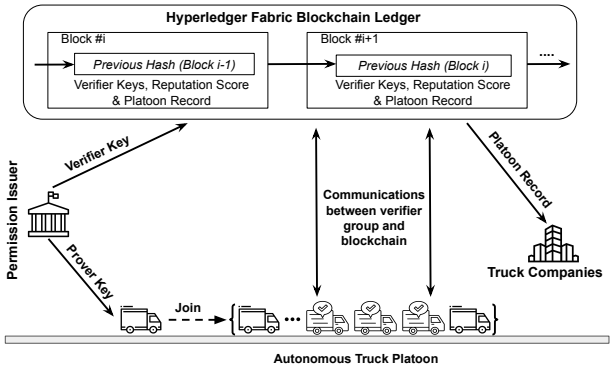

Figure 1. Proposed System Architecture based on permissioned blockchain.

controller of the architecture and serves as the tamper-proof transaction ledger for recording verifier keys, reputation scores and platoon records.

- Verifier Group: A verifier group is a subset of the platoon that includes the most trusted autonomous trucks. It validates the identities of new trucks before joining the platoon, and this group is dynamically updated based on the reputation scores of trucks in the current platoon.
- Autonomous Truck: An autonomous truck is a participant of dynamic platoons and also a candidate for the spatially-local verifier group in the blockchain network.
- Truck Company: A truck company is a client in the blockchain network who may need to retrieve its owned trucks' platooning histories. Trucking companies can use this information for practical applications, such as determining the optimal platoon size on each route to reduce fuel consumption or to quantify platooning benefits such as efficiency and safety improvements.

A. Location-aware Verification Protocol

We introduce the location-aware verification protocol that validates the identity of the autonomous truck in a privacy-preserving manner. When an autonomous truck needs to join a platoon, it acts as the prover to prove its identity to the verifier group in ZKP-based proof. Trusted participants within the platoon form a spatially-local verifier group to validate the proof without revealing the information. The verifier group members are dynamically updated based on the sorting of their reputation scores.

Theorem 1. *Let G be a multiplicative cyclic group of prime order p with generator g. Let $e : G \times G \to G_T$ be a computable, bilinear and non-degenerate pairing into the group G_T. Then, we have $e(x^a, y^b) = e(x, y)^{ab}$ for all $x, y \in G$ and $a, b \in \mathbb{Z}_p$ because G is cyclic.*

Based on Theorem 1 [12], we describe how to construct the location-aware verification protocol as shown in Algorithm 1. The event-driven algorithm mainly consists of four parts: lines 2-6 represent the phase of key generation; lines 7-16 start the joining request by the autonomous truck (prover) and show how zero-knowledge proof is generated; lines 17-31 verify the proof by the verifier group; and lines 32-41 update the verifier group after the autonomous truck joins the platoon. In addition to the following, Section III-B details the prototype of our proposed ZKP scheme in the context of truck platooning. In the prototype, we choose BLS scheme [13] to build the generator g and elliptic curve [14] for bilinear pairing e.

Algorithm 1 Location-aware Verification Protocol

1: **OUTPUT:** The identity verification result r_i for prover i;
2: **KEY GENERATION**
3: The permission issuer selects a random $a_i \in \mathbb{Z}_p$ and computes $v_i = g^{a_i} \in G$;
4: Prover key a_i; ▷ Assign to prover i
5: Verifier key $v_i = g^{a_i}$; ▷ Save on blockchain
6: **END KEY GENERATION**
7: **START UP** *The prover i (autonomous truck) starts a $request$ to join the platoon:*
8: The prover computes its hashed identity information m_i in SHA256 algorithm [15], as $h_i = H(m_i)$;
9: One-time zero-knowledge proof $\delta_i = {h_i}^{a_i} \in G$;
10: The prover sends δ_i to the verifier group;
11: The prover waits for r_i in a time period T;
12: The prover gets r_i in a time period T;
13: **if** no r_i within time T **then**
14: Restart $request$;
15: **end if**
16: **END START UP**
17: **UPON EVENT** *The verifier group receives the one-time zero-knowledge proof δ_i:*
18: **for** each verifier j **do**
19: **if** $e(\delta_i, g) = e(h_i, v_i)$ **then** ▷ Theorem 1
20: $r_{i,j}$ = TRUE;
21: **else**
22: $r_{i,j}$ = FALSE;
23: **end if**
24: **end for**
25: The verifier group returns the r_i based on voting;
26: **if** $r_i ==$ TRUE **then**
27: Approve $request$;
28: **else**
29: Reject $request$;
30: **end if**
31: **END UPON EVENT**
32: **UPON EVENT** *The prover i joins the platoon:*
33: **for** each verifier j **do**
34: **if** $r_{i,j} == r_i$ **then**
35: rs_j++; ▷ Increase reputation score
36: **else**
37: rs_j--; ▷ Decrease reputation score
38: **end if**
39: **end for**
40: Update verifier members based on sorting of each participant's reputation score;
41: **END UPON EVENT**

B. Blockchain Network

Our permissioned blockchain system is prototyped using the Hyperledger Fabric platform. In our design, the blockchain functions as a distributed ledger which stores verifier keys, reputation scores, and truck platoon records. Data is maintained on-chain to guarantee immutability and integrity. By storing the platoon history records on-chain, we provide practical benefits to the fleet companies operating on our platform. For example, a trucking company can retrieve and analyze the platoon records for their vehicles in order to determine the optimal platoon size on each of their routes based on historical data. Furthermore, platooning provides additional benefits of efficiency and safety, and the platoon records stored on the blockchain can be leveraged to help a company quantify the benefits.

III. Experiments and Evaluation

A. Experimental Setup

We prototype the proposed identity verification system and conduct a series of experiments to evaluate its performance. The system consists of two primary portions that interact seamlessly: the verification module based on ZKP and the blockchain network. The ZKP scheme is programmed by using the Hyperledger Ursa library [16]. The blockchain network is developed on the Hyperledger Fabric v1.2 and tested using the Hyperledger Caliper benchmark tool [17]. For testing, we instantiate 10 participants, including 8 autonomous trucks and 2 companies, in the blockchain network. The prototype and experiments are deployed and conducted on multiple Fabric peers in Docker containers locally on Ubuntu 18.04 operating system with 2.8 GHz Intel i5-8400 processor and 8GB DDR4 memory.

B. Verification Module Based on ZKP Scheme

As illustrated in Fig. 2, the ZKP scheme performs the functionalities of initial setup, generation and verification of zero-knowledge proofs of autonomous trucks' identifiers. These functionalities are programmed by using Hyperledger Ursa, a cryptographic library for Hyperledger applications. Hyperledger Ursa is programmed using the Rust language and provides APIs for various cryptographic schemes. Our ZKP module operates in the following three phases:

Phase 1 - Initial Setup: Phase 1 initializes an autonomous truck instance to act as the prover. As shown in Fig. 2, the truck has the identifier information `mac_address` (value: `00:A0:C9:14:C8:29`), and the permission issuer generates a key pair for the identity information. The BLS scheme [13] is used to build the key pair generator, which creates the prover key for the truck and the verifier key on the blockchain ledger, as follows:

```
let generator = Generator::new().unwrap();
let prover_key = SignKey::new().unwrap();
let verifier_key = VerKey::new(&generator,
                   &sign_key).unwrap();
```

Phase 2 - ZKP Generation: In this phase, the autonomous truck uses the prover key to generate a one-time zero-knowledge proof for the hashed `mac_address` via SHA256 algorithm [15]. The resulting proof consists of three elements on an elliptic curve. For instance, as shown below:

```
(1, 0DA6294C26738DCB05F0E660E960C3EAB0BC98E8BF5D6DACE4105BF4BCEEA572)
(1, 1AA3729F7F95269A914B55BDE149BBFFB9304651D86B0FCA311E49CE9B15375B)
(2, 095E45DDF417D05FB10933FFC63D474548B7FFFF7888802F07FFFFFF7D07A8A8)
```

the proof of hashed `00:A0:C9:14:C8:29` from `mac_address` is a combination of three points on an elliptic curve represented in hexadecimal format. Our experiments show that the average running time for the proof generation phase is 29 ms.

Phase 3 - ZKP Verification: After the proof generation, the verifier group from the platoon validate the zero-knowledge proof from the truck. The verification function takes the proof, the hashed `mac_address`, the verifier key and the corresponding generator as inputs, and utilizes elliptic curve bilinear pairing [14] to verify the proof:

```
let result = Bls::verify($proof,
             mac_address.as_slice(),
             $verifier_key, $generator)
             .unwrap();
```

In our experiments, the average running time for verifying each zero-knowledge proof is around 210 ms. After that, the platoon can authenticate the vehicle's identifier anonymously and, subsequently, communicate the result to both the truck and blockchain network.

C. Blockchain Network

Hyperledger Fabric is an open-source and modular permissioned blockchain framework [18]. The four programmable modules used in our system are: model file (.cto) which is used to define all of the data structures in the network; script file (.js) where smart contracts are written; access control list (.acl) for deploying access control policies; and the query file (.qry)

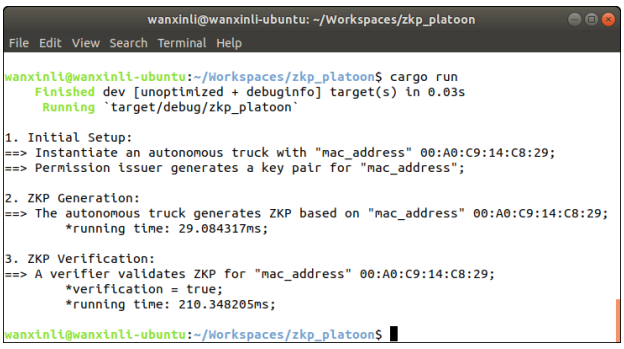

Figure 2. Process of the ZKP scheme on Hyperledger Ursa.

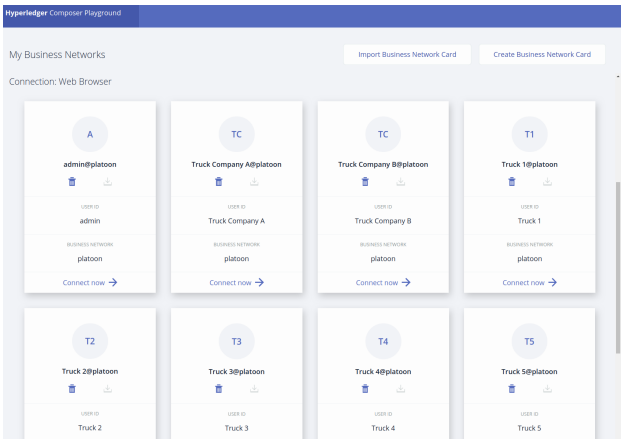

Figure 3. Blockchain network login window for companies and autonomous trucks.

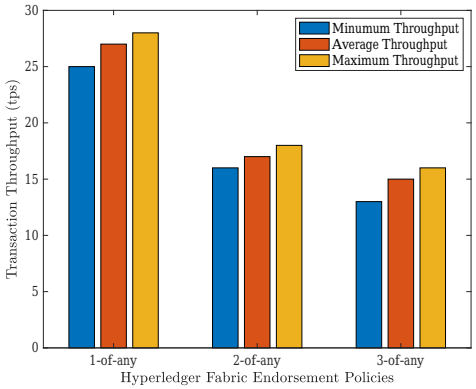

Figure 4. Minimum, average and maximum transaction throughput vs. Hyperledger Fabric endorsement policy.

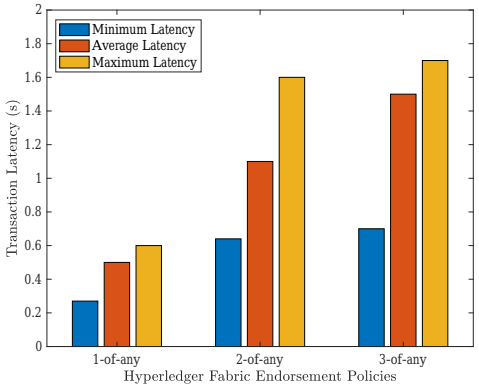

Figure 5. Minimum, average and maximum transaction latency vs. Hyperledger Fabric endorsement policy.

which defines the query operations similarly to a traditional database system.

In our prototyped blockchain system, we provide a web portal for the network participants (autonomous trucks and companies), which can be used to interact with the blockchain network. An example can be seen in Fig. 3, where each participant has a registered ID for connecting to the blockchain. The trusted entity operating as the permission issuer in our system (e.g., DMV) also acts as the blockchain administrator.

D. Performance Evaluation

1) Transaction Throughput: We first measure the transaction throughput of our blockchain network prototype. Transaction throughput for a blockchain network quantifies the rate at which transactions are processed through the network over a given time cycle in units of transactions per second. We tested the throughput under different endorsement policies, and the results for 1-of-any, 2-of-any, and 3-of-any policies are summarized in Fig. 4. Our results show that the number of endorsing peers has an inverse relationship to the transaction throughput of the network, and the transaction throughput peaks at 27 tps, 17 tps, and 15 tps under 1-of-any, 2-of-any, and 3-of-any policies respectively.

The endorsement policy has a strong impact on transaction throughput. However, this makes sense because increasing the number of peers required to validate a transaction also increases the complexity of the endorsement process. That being said, our results from multiple rounds of testing show that for a given endorsement policy, the performance is relatively stable and the difference between the minimum, maximum and average cases is minor.

2) Transaction Latency: We also perform experiments to measure and quantify the transaction latency of our prototyped blockchain network. Transaction latency measures the end-to-end processing time for a transaction in the blockchain network, from initial client submission to the time when the transaction is committed to the ledger. We perform multiple experiment rounds with varying endorsement policies and compiled our results in Fig. 5. The relationship between the transaction latency and endorsement policy is readily apparent: as the number of endorsing peers increases, we see an increase in both average and maximum transaction latency's. However, it is worth noting that as the number of endorsing peers increases, we also see an increase in variability between the minimum, average and maximum cases. That being said, the difference in latency between 2-of-any and 3-of-any endorsement policies is significantly lower than the difference between 1-of-any and 2-of-any cases.

IV. Conclusion

This paper introduces a novel location-aware and privacy-preserving verification protocol focused on the application of dynamic platooning for autonomous trucks. Our proposed system integrates zero-knowledge proof with permissioned blockchain technology. By performing the ZKP-based identity verification within the spatially-local area defined by a given platoon, our system can provide lower latency verification with less communication overhead compared to a location-agnostic system. To analyze the system performance, we prototype our design on Hyperledger platform and perform various experiments. Initial results highlight our system's real-world feasibility for providing both low-latency identity verification and transaction processing on the order of milliseconds.

References

[1] S. Tsugawa, "Results and issues of an automated truck platoon within the energy its project," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE, 2014, pp. 642–647.

[2] S. Singh, "Critical reasons for crashes investigated in the national motor vehicle crash causation survey," National Highway Traffic Safety Administration, Tech. Rep., 2015.

[3] Truckinginfo, "Vtti to study autonomous trucks in mixed fleets," accessed: 2020-10-17. [Online]. Available: https://www.truckinginfo.com/355628/vtti-to-study-autonomous-trucks-in-mixed-fleets

[4] C. Chen, T. Xiao, T. Qiu, N. Lv, and Q. Pei, "Smart-contract-based economical platooning in blockchain-enabled urban internet of vehicles," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4122–4133, 2020.

[5] P. K. Singh, R. Singh, S. K. Nandi, and S. Nandi, "Integrating blockchain with cacc for trust and platoon management," *Cryptocurrencies and Blockchain Technology Applications*, pp. 77–97, 2020.

[6] W. Li, M. Nejad, and R. Zhang, "A blockchain-based architecture for traffic signal control systems," in *2019 IEEE International Congress on Internet of Things (ICIOT)*. IEEE, 2019, pp. 33–40.
[7] W. Li, C. Meese, H. Guo, and M. Nejad, "Blockchain-enabled identity verification for safe ridesharing leveraging zero-knowledge proof," in *IEEE International Conference on Hot Information-Centric Networking (HotICN)*. IEEE, 2020.
[8] H. Guo, W. Li, M. Nejad, and C. Shen, "Access control for electronic health records with hybrid blockchain-edge architecture," in *2019 IEEE International Conference on Blockchain (Blockchain)*, 2019, pp. 44–51.
[9] H. Guo, W. Li, E. Meamari, C. C. Shen, and M. Nejad, "Attribute-based multi-signature and encryption for ehr management: A blockchain-based solution," in *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2020, pp. 1–5.
[10] 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. 181 733–181 743, 2020.
[11] 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. 182 776–182 786, 2020.
[12] "Cyclic Group Supplement," accessed: 2020-11-05. [Online]. Available: https://www.math.lsu.edu/~adkins/m4200/cyclicgroup.pdf
[13] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in *International conference on the theory and application of cryptology and information security*. Springer, 2001, pp. 514–532.
[14] G. Frey, M. Muller, and H.-G. Ruck, "The tate pairing and the discrete logarithm applied to elliptic curve cryptosystems," *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1717–1719, 1999.
[15] D. Rachmawati, J. Tarigan, and A. Ginting, "A comparative study of message digest 5 (md5) and sha256 algorithm," in *Journal of Physics: Conference Series*, vol. 978, no. 1, 2018, p. 012116.
[16] "Hyperledger ursa," accessed: 2020-11-03. [Online]. Available: https://www.hyperledger.org/projects/ursa
[17] "Hyperledger caliper," accessed: 2020-11-15. [Online]. Available: https://www.hyperledger.org/use/caliper
[18] "Hyperledger fabric," accessed: 2020-10-21. [Online]. Available: https://www.hyperledger.org/projects/fabric