Blockchain-Enabled Identity Verification for Safe Ridesharing Leveraging Zero-Knowledge Proof

Wanxin Li[*], Collin Meese[*], Hao Guo[†], Mark Nejad[*]

[*] Department of Civil and Environmental Engineering, University of Delaware, U.S.A.

[†]School of Software, Northwestern Polytechnical University, Taicang Campus, China

{wanxinli,cmeese,nejad}@udel.edu, haoguo@nwpu.edu.cn

Abstract—The on-demand mobility market, including ridesharing, is becoming increasingly important with e-hailing fares growing at a rate of approximately 130% per annum since 2013. By increasing utilization of existing vehicles and empty seats, ridesharing can provide many benefits including reduced traffic congestion and environmental impact from vehicle usage and production. However, the safety of riders and drivers has become of paramount concern and a method for privacy-preserving identity verification between untrusted parties is essential for protecting users. To this end, we propose a novel privacy-preserving identity verification system, extending zero-knowledge proof (ZKP) and blockchain for use in ridesharing applications. We design a permissioned blockchain network to perform the ZKP verification of a driver's identity, which also acts as an immutable ledger to store ride logs and ZKP records. For the ZKP module, we design a protocol to facilitate user verification without requiring the exchange of any private information. We prototype the proposed system on the Hyperledger Fabric platform, with the Hyperledger Ursa cryptography library, and conduct extensive experimentation. To measure the prototype's performance, we utilize the Hyperledger Caliper benchmark tool to perform extensive analysis and the results show that our system is suitable for use in real-world ridesharing applications.

Index Terms—Blockchain, data privacy, identity verification, ridesharing, zero-knowledge proof.

I. INTRODUCTION

Ridesharing services have received much attention in the past decade as an efficient method for increasing societal access to mobility using existing infrastructure. E-hailing fares have experienced significant growth at a rate of approximately 130% per annum since 2013 [1]. In addition, a recent parking study shows that our vehicles are not in use for an average of 95% of their lifetimes [2], presenting much room for increasing existing vehicle utilization. However, both regulators and users have expressed significant safety concerns with existing ridesharing applications, citing incidents such as driver impersonations, sexual assault and kidnapping. According to a recent report disclosed by Uber, there were nine assault-related deaths as well as over 5,900 cases of non-consensual sexual assault related incidents on their ridesharing platform in 2017 and 2018 combined [3]. As a result, safeguarding the well-being of ridesharing users is a pivotal issue in the on-demand mobility market.

Although some companies have implemented systems to help prevent misidentification and ensure safety, such as the Uber PIN system [4], these schemes are centralized and platform exclusive, making them unsuitable for ensuring safety across the entire ridesharing ecosystem. Additionally, it is challenging to provide an extensive identity verification system while also respecting the privacy of untrusted users. Therefore, the problem of safe and privacy-preserving identity verification in ridesharing systems is of paramount interest and a solution is needed to enable universal and decentralized identity verification.

Recently, blockchain technology has been proposed as a plausible way of enabling efficient and decentralized two-sided sharing economies, such as ridesharing systems [5] [6] [7]. First implemented in Bitcoin: A Peer-to-Peer Electronic Cash System, blockchain technology is an emerging network technology enabling consensus among networked peers on a distributed, immutable digital ledger [8]. The inherent provenance and immutability properties of blockchain make it an ideal candidate for implementing a secure and decentralized identity verification protocol for future ridesharing systems.

However, public blockchain systems present significant privacy concerns when sensitive user information is involved. With their transparency-by-design properties, any networked participant can view the entire contents of the ledger in a permissionless blockchain system such as Bitcoin. On the other hand, zero-knowledge proof (ZKP) is a cryptographic scheme enabling one party to prove to another that they know a secret message without revealing any information other than possession of the secret. As a result, ZKP can be implemented as a module atop blockchain network as a way to protect the privacy of users during the verification process.

In this paper, we propose a permissioned blockchain and zero-knowledge proof [9] inspired approach for safe and privacy-preserving digital identity verification in existing ridesharing systems. This paper makes the following contributions:

- We propose a novel zero-knowledge proof module atop a permissioned blockchain network for identity verification to protect the safety and privacy of riders and drivers without disclosing any sensitive information.
- We prototype the proposed system and perform benchmark tests on Hyperledger platform [10]. The experimental results show that our system is feasible for real-world ridesharing applications.

The rest of this paper is organized as follows. In Section II, we present the requisite background knowledge regarding

zero-knowledge proofs. A detailed overview of the proposed system architecture is described in Section III. To further illustrate the feasibility of our proposed scheme, we perform extensive experiments testing the performance of our identity verification system in Section IV. Also, we discuss the resilience of our proposed system against potential attacks in this section. For Section V, we present previous and related research in blockchain applications and blockchain-based ridesharing systems. Finally, we conclude this study in Section VI.

II. BACKGROUND KNOWLEDGE OF ZERO-KNOWLEDGE PROOF

Zero-knowledge proof (ZKP) was proposed in 1989 by Goldwasser, Micali, and Rackoff [11]. In cryptography, a ZKP protocol is a method by which a prover can convince a verifier that he/she knows a secret message m, without conveying any information, apart from the fact that the prover knows the secret message m [12]. A ZKP of knowledge is a special case when the statement consists only of the fact that the prover possesses the secret information [12]. Based on the frequency of communications between the prover and the verifier, there are two kinds of ZKP schemes: Interactive ZKP and Non-interactive ZKP scheme. A zero-knowledge proof must satisfy three properties:

- Completeness: If the statement is true, the honest verifier (that is, one following the protocol properly) will be convinced of this fact by an honest prover.
- Soundness: There is no such prover can convince an honest verifier if he/she does not compute the results correctly.
- Zero-knowledge: The proof of knowledge can be simulated without revealing any secret information which means that no verifier learns anything other than the fact that the statement is true.

ZKP can be applied to blockchain systems to address privacy issues. To be more specific, ZKP can be used to guarantee that transactions in blockchain are valid without revealing any sensitive information about the sender, the recipient or other transaction details. For instance, Zcash utilizes zk-SNARKs (zero-knowledge succinct non-interactive arguments of knowledge) scheme to allow blockchain peers to reach agreement on the validity of transactions without knowing details about the sender, recipient or amounts being transferred [13].

III. SYSTEM ARCHITECTURE

In this section, we describe the proposed privacy-preserving system architecture for identity verification in ridesharing, which includes a new zero-knowledge proof protocol and permissioned blockchain network. 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 issues identity information (e.g., driver license number) and key pairs to data owners and data verifiers.
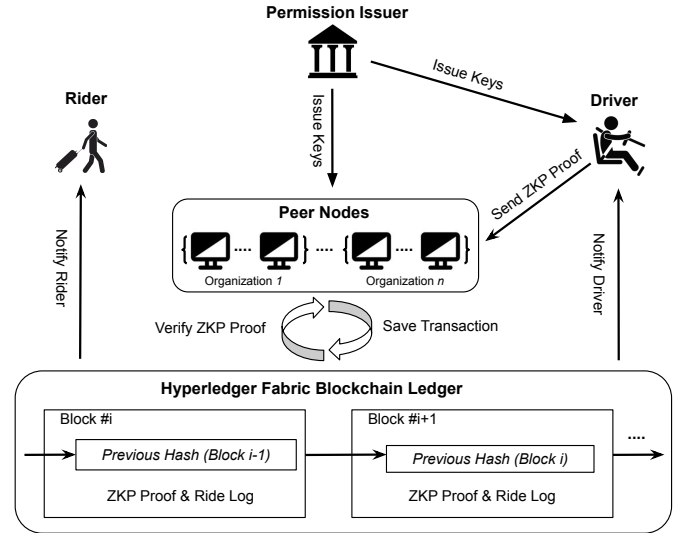

Fig. 1. Proposed System Architecture based on permissioned blockchain.

In practice, an agency such as Department of Motor Vehicles (DMV) can function as the permission issuer in the blockchain network.

- Driver: A driver is a client in blockchain network who owns his/her identity information and wants to prove their identity to riders without revealing the information.
- Rider: A rider is a client in blockchain network who wants to safely verify the identity of driver before trip starts.
- Peer Node: A peer node is the element that hosts the ledger and smart contracts in the permissioned blockchain network and also acts as the data verifier who validates drivers' identities for riders. In our system, peer nodes are deployed and managed by a consortium of multiple organizations creating a decentralized network.
- Permissioned Blockchain: A permissioned blockchain (in our prototype, Hyperledger Fabric) is utilized as the controller of the architecture and serves as the tamper-proof transaction ledger for saving proof records and trip information.

We design and develop the blockchain-based identity verification system for ridesharing on Hyperledger Fabric platform with Hyperledger Ursa cryptographic library [14]. Before each trip starts, the identity of a driver will be validated by the proposed zero-knowledge proof protocol without revealing the sensitive information to other participants in the network. The blockchain network maintains a distributed ledger for recording trip information and proof records. In the remainder of this section, we illustrate the design of zero-knowledge proof protocol, followed by a description of the workflow of identity verification in ridesharing using our proposed system. We discuss the implementation of the blockchain network in Section IV-C.
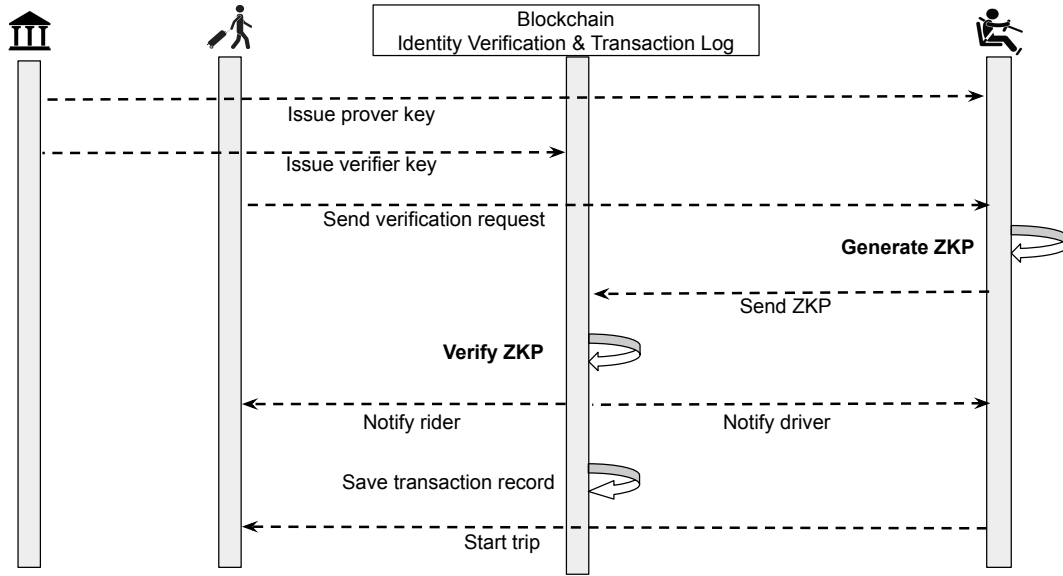
19

Fig. 2. Workflow of privacy-preserving identity verification in ridesharing scenario.

A. Zero-Knowledge Proof Protocol

As shown in Fig. 1, we introduce the zero-knowledge proof protocol for verifying the identity of drivers without revealing sensitive information to other participants. When a ridesharing service matches a driver with a rider, the driver acts as a prover to prove his/her identity to a peer node from blockchain network, acting as a verifier, in ZKP-based encrypted message. After the blockchain peer validates the driver's identity, the result is communicated to both parties.

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 $e(x, y) = e(y, x)$ for all $x, y \in G$ because G is cyclic.*

Based on Theorem 1 [15], we describe how to construct the ZKP protocol in our proposed scheme for identity verification. In addition to the following, we detail the specifics of our proposed ZKP scheme in the context of ridesharing in Section IV-B. In the prototype, we choose BLS scheme [16] to build the generator g and elliptic curve [17] for bilinear pairing e. Our ZKP protocol consists of three key functions:

1) Key Generation: The key generation algorithm selects a random $a \in \mathbb{Z}_p$ and computes $v = g^a$. The prover key is $a \in \mathbb{Z}_p$ and the verifier key is $v \in G$.

2) Proof Generation: Given the prover key a, and the identity information m, the prover computes the hashed m in SHA256 algorithm [18], as $h = H(m)$. Then, the proof is generated as $\delta = h^a \in G$.

3) Proof Verification: Given the proof δ and the verifier key v, the verifier can verify that $e(\delta, g) = e(h, v) = e(H(m), g^a)$ without revealing the identity information m, reject otherwise.

B. Identity Verification Workflow in Ridesharing

The workflow of identity verification for ridesharing is depicted in Fig. 2. In the beginning, all the drivers and riders register themselves in the blockchain network. The permission issuer (e.g., Department of Motor Vehicles) issues prover keys to drivers and verifier keys to peer nodes. The registration and key issuance process only needs to be performed once for each user. When a ridesharing service is matched between a driver and a rider, the rider first sends the verification request to the driver prior to trip start. After receiving the request from rider, the driver uses the prover key to generate zero-knowledge proof for his identity information and send it to a peer node in blockchain network. Next, the peer node uses the verifier key to validate the zero-knowledge proof from the driver. When the verification is complete, the blockchain network will notify both the rider and the driver of the result. Simultaneously, a smart contract is executed to record this transaction with trip details on the ledger. After that, the driver can start the trip.

IV. 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 ZKP module and the blockchain network. The ZKP module is programmed by using the Hyperledger Ursa library [14]. The blockchain network is developed on the Hyperledger Fabric v1.2 and tested using Hyperledger Caliper benchmark tool [19]. For testing, we instantiate 10 clients including 5 drivers and 5 riders, 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.

Fig. 3. Process of the ZKP module.

Fig. 4. Blockchain-based ridesharing network login window.

B. ZKP Module

As illustrated in Fig. 3, the ZKP module performs the functionalities of initial setup, generation and verification of zero-knowledge proofs of drivers' identity information. 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:

1) Phase 1 - Initial Setup: Phase 1 initializes a driver instance to act as the prover. As shown in Fig. 3, the driver has the identity information `driver_license` (value: `9907184`) issued by the Department of Motor Vehicles, and the permission issuer generates a key pair for the identity information it issues. The BLS scheme [16] is used to build the key pair generator, which creates the prover key for the driver and the verifier key for the peer node in blockchain network, as follows:

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

2) Phase 2 - ZKP Generation: In this phase, the driver uses the prover key to generate a one-time zero-knowledge proof for the hashed `driver_license` via SHA256 algorithm [18]. The resulting proof consists of three elements on an elliptic curve. For instance, as shown below:

```
(1, 12AD4E711FFFB24FDCAFAF8E787D725A2A12335DABC3F4B99547D1F9191ADE4E)
(1, 15915C01530805CC3007A48CC8D8D0075C9D95AC76784801884722B01D23CB7C)
(2, 095E45DDF417D05FB10933FFC63D474548B7FFFF7888802F07FFFFFF7D07A8A8)
```

the proof of hashed `9907184` from `driver_license` is a combination of three points on an elliptic curve represented in hexadecimal format. Our experiments show that the average running time for proof generation phase is 32 ms.

3) Phase 3 - ZKP Verification: After proof generation, a peer node from the blockchain network (Section IV-C) verifies the zero-knowledge proof from the driver. The verification function takes the proof,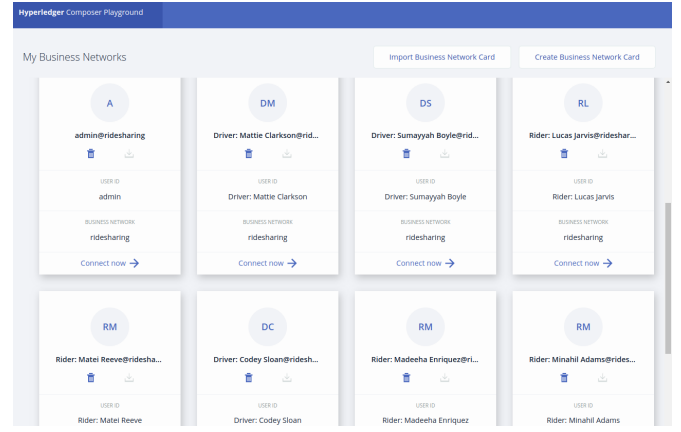 the hashed `driver_license`, the verifier k ey a nd t he c orresponding g enerator a s i nputs, and utilizes elliptic curve bilinear pairing [17] to verify the proof:

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

In our experiments, the average running time for verifying each zero-knowledge proof is around 239 ms. After that, the blockchain system can authenticate the driver's identity anonymously and subsequently communicate the result to both the rider and driver.

C. Blockchain Network

Hyperledger Composer [20] is a framework to build and run applications on top of Hyperledger Fabric. Hyperledger Composer provides four programmable modules: Model File, Script File, Access Control and Query File. We first define all the objects in the Model File including the data structures of participants and transactions. Then, smart contracts are written, using Chaincode [21], in the Script File. The access control policies are coded in the Access Control File to facilitate different access levels to different participants. As for the Query File, it works similar to conventional database query operations for data retrieval. Lastly, these files are packaged up into one Business Network Archive (.bna) file and deployed on the blockchain network.

The proposed blockchain system also provides a web interface for participants interacting with the blockchain network as shown in the Fig. 4. The traffic management authority (e.g., United States Department of Transportation) acts as the administrator to issue access permissions for the other clients including riders and drivers. As a result, each participant has an ID registry for connecting to the blockchain network.

D. Performance Evaluation

In this subsection, we evaluate the performance of both the ZKP module and blockchain network. To measure the performance of the ZKP module, we conduct experiments to analyze the effects of varying the length of the secret message. For
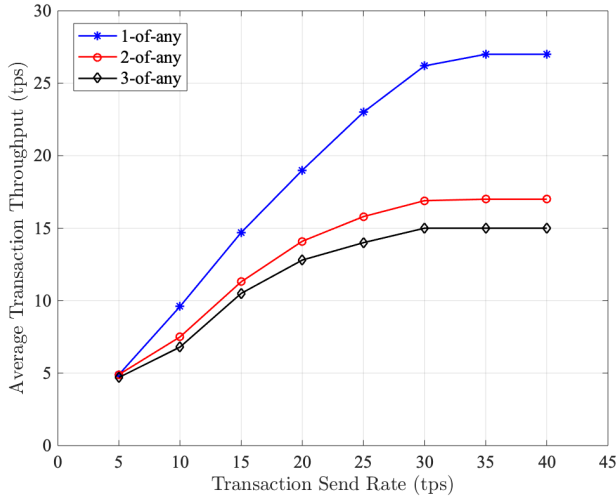
Fig. 5. Average transaction throughput vs. send rate under different Hyperledger Fabric endorsement policies.
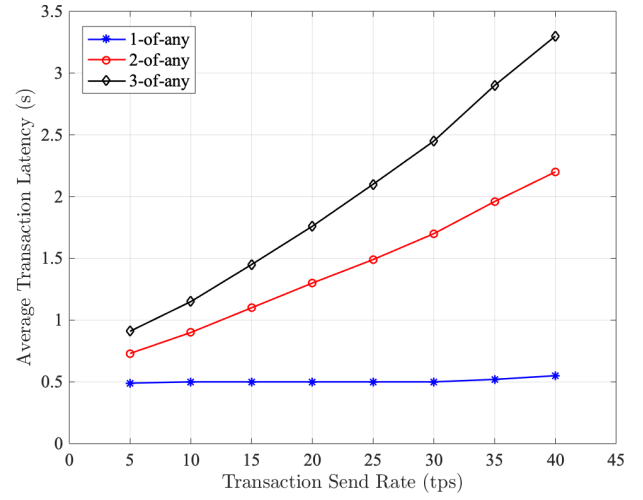

Fig. 6. Average transaction latency vs. send rate under different Hyperledger Fabric endorsement policies.

blockchain network evaluation, we perform benchmark tests using Hyperledger Caliper with varying settings for transaction send rates and endorsement policies. The definitions of these two benchmark factors are as follows:

- Transaction Send Rate: The transaction send rate defines the rate at which transactions are input to the blockchain network system, which is a key factor for stress testing.
- Endorsement Policy: The endorsement policies define the set of peers need to reach agreement on the result of a transaction before it can be committed to the ledger.

1) ZKP Performance: The default length of the secret message (`driver_license`) is 7 characters using String type. In our experiments, we increase the length of secrets from 1 to 10, 100, 1,000 and 10,000 characters and then measure the results. The results show that both proof generation and verification times remain constant regardless of the secret length, with times of approximately 32 ms and 239 ms, respectively. Our ZKP module can offer constant proof generation and verification time because a secret message m is hashed to a fixed length of 256-bit value via SHA256 algorithm [18] before proof generation. As a result, the proof generation and verification time are independent from the length of secret messages. Non-reliance on secret length provides our ZKP scheme with more flexibility for verifying different kinds of secret values (e.g., social security number, taxpayer identification number, etc.) without sacrificing security and efficiency. Compared to the signing phase, the verification phase takes additional time because it requires computing two pairings on the elliptic curve [16].

2) Transaction Throughput: The transaction throughput measures the flow rate of processed transactions through the blockchain network, in units of transactions per second, during the test cycle. As shown in Fig. 5, when increasing the transaction send rate, the average transaction throughput will increase in the beginning and then reach peaks at 27 tps, 17 tps, and 15 tps under 1-of-any, 2-of-any, and 3-of-any endorsement

policies, respectively. The choice of endorsement policy can impact the transaction throughput. For example, with a fixed transaction send rate of 20 tps, when increasing the number of endorsing peers, the average transaction throughput will decrease. This is due to the fact that more endorsing peers increase the complexity of the endorsement process.

3) Transaction Latency: The transaction latency indicates the processing time of a transaction, from submission by the client until it is processed and committed to the ledger. As shown in Fig. 6, when increasing the transaction send rate, the average transaction latency will also increase significantly for 2-of-any and 3-of-any endorsement policies. However, the average transaction latency remains constant at 0.5 second under the 1-of-any endorsement policy, as long as the transaction send rate is lower than 30 tps. When increasing the transaction send rate beyond 30 tps, the average latency begins to increase, but at a significantly slower rate in comparison with the of the 2-of-any or 3-of-any policies. This is because the 1-of-any endorsement policy has a higher transaction throughput threshold due to its reduced complexity. In addition, the choice of endorsement policy can significantly impact the transaction latency. For example, with a fixed transaction send rate of 20 tps, when increasing the number of endorsement peers, the average transaction latency will increase.

4) Success Rate: The success rate measures how many transactions, out of all submitted transactions, have been successfully processed and written on the blockchain during a test cycle. A failed transaction could be due to network time-outs, incorrect network configuration or bugs in smart contract. For all test cycles, regardless of different send rates and endorsement policies, our blockchain network can always achieve 100% success rates.

E. Discussion on Potential Attacks

In this subsection, we discuss the resilience of the proposed privacy-preserving identity verification system against three

```
● ● ●  🏠 wanxinli — wanxinli@wanxinli-ubuntu: ~/Workspaces/zkp_ridesharing — ssh wan...
wanxinli@wanxinli-ubuntu:~/Workspaces/zkp_ridesharing$ cargo run
    Finished dev [unoptimized + debuginfo] target(s) in 0.03s
     Running `target/debug/zkp_ridesharing`

1. Initial Setup:
==> Instantiate a driver with "driver_license" 9907184
==> Instantiate an attacker with "driver_license" 180612
==> Authority issues key pairs for "driver_license"

2. Zero-Knowledge Proof Generation:
==> The attacker generates ZKP based on "driver_license" 180612
        *running time: 32.257642ms

3. Zero-Knowledge Proof Verification:
==> A peer node verifies ZKP for "driver_license" 9907184
        *verification = false
        *running time: 236.800068ms
```

Fig. 7. Response against fake driver attack.

potential attacks.

1) Ride Log Tampering Attack: In the proposed system, a newly generated ride log is stored on the blockchain in the form of a transaction. Our system will reject any attempt to tamper an existing ride log due to the immutable feature of blockchain, ensuring data integrity for all recorded information. Therefore, once the trip information is recorded, it is protected from modification by any entity.

2) Eavesdropping Attack: Our proposed ZKP module can protect ridesharing systems from the eavesdropping attack. In this attack, the malicious attacker intercepts the message between driver (a prover) and peer node (a verifier) from the blockchain network in order to steal or gain access to confidential information. In the worst case, if the sensitive information has been accessed by the attacker, it still cannot reveal the actual information since the identity information is ZKP-encrypted. As a result, no sensitive information is disclosed even in the occurrence of an eavesdropping attack.

3) Fake Driver Attack: A fake driver attack is a situation where the driver wants to spoof the ZKP verification process by providing fake identity information to the verifier. However, the ZKP protocol is resistant to fake identity s poofing by design, protecting against this type of attack. For instance, if the attacker intends to deceive the prover, but actually has the incorrect identity (e.g., `driver_license 180612`), our system will detect and reject the malicious proof. For instance, the output of our protocol against a fake driver attack is shown in Fig. 7.

V. RELATED WORK

In this section we outline the related work on both blockchain in general and blockchain-based ridesharing systems which motivate and compliment our research.

Blockchain distributed network technology has received much research attention since its inception and studies have demonstrated its applicability to a variety of areas including decentralized finance, healthcare, Internet of Things (IoT) and sharing economies. For example, in the area of decentralized finance, Desai et al. [22] present a hybrid blockchain architecture for privacy-enabled and accountable auctions and [23]

details a blockchain-enabled croudsourcing platform that preserves user privacy. Additionally, a platform for initial public offerings on a permissioned blockchain system is outlined in [24]. In the area of healthcare, [25] proposes a novel method for secure access control in electronic health record management using a hybrid blockchain architecture and attribute-based multiple signatures [26]. Regarding blockchain-based IoT systems, Kim et al. [27] introduce SSP: self-sovereign privacy for IoT using a combination of blockchain and Multi-Party-Computations (MCP). Wanxin et al. [28] introduce a blockchain-based approach for defending emerging Intelligent Traffic Signal Systems against malicious attackers and then extend the study in multiple vehicular networks [29]. Hao et al. propose a blockchain-inspired event recording system for autonomous vehicles [30].

There have been various attempts to address the problems of centralized ridesharing, and sharing economies in general, using blockchain-based systems. Baza et al. [31] propose a privacy-preserving and trustless ridesharing application with proof of concept implemented atop the Ethereum blockchain [32]. Their system includes protocols for matching, payment, reputation and identity management utilizing a combination of smart contracts and zero-knowledge-range-proofs. However, their system incorporates a fully decentralized public blockchain which can dramatically increases resource consumption and transaction confirmation time, as well as associated identity verification latency, in comparison to a permissioned blockchain design such as in our system.

[33] and [34] also propose comprehensive ridesharing systems atop a public blockchain which preserves privacy using pseudonymity schemes. These designs motivated our research into a system for identity management and verification for ridesharing which requires no pseudonyms or exchange of private information between either party. Additionally, Semenko et al. [35] propose a decentralized and privacy-preserving ridesharing platform using direct two-party encryption and smart contract based access control. The result, however, is that users must exchange and reveal their sensitive information to other parties if they wish to use the service.

VI. CONCLUSION

In this paper, we present a novel privacy-preserving identity verification system, extending zero-knowledge proof atop blockchain for use in ridesharing applications. Our proposed scheme enables secure verification without requiring exchange of any sensitive information between untrusted parties. We prototype the system and perform extensive experiments to assess its performance and practicality under varying conditions. Our results show the blockchain network, based on Hyperledger Fabric, provides high transaction throughput with low-latency. Meanwhile, the ZKP module is able to perform identity verification within milliseconds, making our design suitable for use in real-world ridesharing applications. In addition, we discuss the resilience of our system against multiple potential attacks.

REFERENCES

[1] T. Baltic, A. Cappy, R. Hensley, and N. Pfaff. How sharing the road is likely to transform american mobility. [Online]. Available: https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/how-sharing-the-road-is-likely-to-transform-american-mobility#

[2] D. Z. Morris, "Today's cars are parked 95% of the time," *Fortune, March*, vol. 13, 2016.

[3] Uber. Uber's united safety report. [Online]. Available: https://www.uber.com/us/en/about/reports/us-safety-report/

[4] ——. For drivers: How pin verification works. [Online]. Available: https://www.uber.com/blog/pin-verification-drivers/

[5] S. E. Chang and C. Chang, "Application of blockchain technology to smart city service: A case of ridesharing," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2018, pp. 664–671.

[6] K. Kato, Y. Yan, and H. Toyoizumi, "Blockchain application for rideshare service," in *2018 8th International Conference on Logistics, Informatics and Service Sciences (LISS)*, 2018, pp. 1–5.

[7] P. Pal and S. Ruj, "Blockv: A blockchain enabled peer-peer ride sharing service," in *2019 IEEE International Conference on Blockchain (Blockchain)*, 2019, pp. 463–468.

[8] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[9] C. Rackoff and D. R. Simon, "Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack," in *Annual International Cryptology Conference*. Springer, 1991, pp. 433–444.

[10] Hyperledger Fabric. [Online]. Available: https://www.hyperledger.org/projects/fabric

[11] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof systems," *SIAM Journal on computing*, vol. 18, no. 1, pp. 186–208, 1989.

[12] Wikipedia contributors, "Zero-knowledge proof — Wikipedia, the free encyclopedia," 2020. [Online]. Available: https://en.wikipedia.org/wiki/Zero-knowledge_proof

[13] D. Hopwood, S. Bowe, T. Hornby, and N. Wilcox, "Zcash protocol specification," *GitHub: San Francisco, CA, USA*, 2016.

[14] Hyperledger Ursa. [Online]. Available: https://www.hyperledger.org/projects/ursa

[15] Cyclic Group Supplement. [Online]. Available: https://www.math.lsu.edu/~adkins/m4200/cyclicgroup.pdf

[16] 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.

[17] 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.

[18] 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.

[19] Hyperledger Caliper. [Online]. Available: https://www.hyperledger.org/use/caliper

[20] Hyperledger Composer. [Online]. Available: "https://www.hyperledger.org/projects/composer"

[21] Chaincode. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.4/chaincode.html

[22] H. Desai, M. Kantarcioglu, and L. Kagal, "A hybrid blockchain architecture for privacy-enabled and accountable auctions," in *2019 IEEE International Conference on Blockchain (Blockchain)*, 2019, pp. 34–43.

[23] S. Zhu, H. Hu, Y. Li, and W. Li, "Hybrid blockchain design for privacy preserving crowdsourcing platform," in *2019 IEEE International Conference on Blockchain (Blockchain)*, 2019, pp. 26–33.

[24] T. Halevi, F. Benhamouda, A. D. Caro, S. Halevi, C. Jutla, Y. Manevich, and Q. Zhang, "Initial public offering (ipo) on permissioned blockchain using secure multiparty computation," in *2019 IEEE International Conference on Blockchain (Blockchain)*, 2019, pp. 91–98.

[25] 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.

[26] 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.

[27] T. H. Kim and J. Lampkins, "Ssp: Self-sovereign privacy for internet of things using blockchain and mpc," in *2019 IEEE International Conference on Blockchain (Blockchain)*, 2019, pp. 411–418.

[28] 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.

[29] 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.

[30] 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.

[31] M. Baza, N. Lasla, M. Mahmoud, G. Srivastava, and M. Abdallah, "B-ride: Ride sharing with privacy-preservation, trust and fair payment atop public blockchain," *IEEE Transactions on Network Science and Engineering*, pp. 1–1, 2019.

[32] F. Vogelsteller, V. Buterin *et al.*, "Ethereum whitepaper," *Ethereum Foundation*, 2014.

[33] Y. Kanza and E. Safra, "Cryptotransport: Blockchain-powered ride hailing while preserving privacy, pseudonymity and trust," in *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. SIGSPATIAL '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 540–543. [Online]. Available: https://doi.org/10.1145/3274895.3274986

[34] M. Baza, M. Mahmoud, G. Srivastava, W. Alasmary, and M. Younis, "A light blockchain-powered privacy-preserving organization scheme for ride sharing services," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–6.

[35] Y. Semenko and D. Saucez, "Distributed privacy preserving platform for ridesharing services," in *Security, Privacy, and Anonymity in Computation, Communication, and Storage*, G. Wang, J. Feng, M. Z. A. Bhuiyan, and R. Lu, Eds. Springer International Publishing, 2019, pp. 1–14.