IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier

Privacy-Preserving Traffic Management: A Blockchain and Zero-Knowledge Proof Inspired Approach

WANXIN LI[1], HAO GUO[2], MARK NEJAD[1], and CHIEN-CHUNG SHEN[2]
[1]Department of Civil and Environmental Engineering, University of Delaware, Newark, DE 19716, USA
[2]Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716, USA

Corresponding author: Mark Nejad (e-mail: nejad@udel.edu)

ABSTRACT Incorporation of connected vehicle (CV) data into real-time traffic management systems presents a host of new challenges resulting from the current lack of data integrity and data privacy in traffic networks. Over the past few years, blockchain technologies have been inspiring extensive innovations in the transportation field. However, due to the transparency property, sensitive data stored on the blockchain would be accessible to anyone, resulting in a lack of privacy. In this paper, we propose a decentralized and location-aware architecture to address the data integrity along with the privacy-preserving issues in blockchain-based traffic management systems. Our proposed architecture integrates with permissioned and modular blockchain network and non-interactive zero-knowledge range proof (ZKRP) protocol. We develop the prototype system on the Hyperledger Fabric platform and Hyperledger Ursa cryptographic library. The performance results show that our approach is effective and feasible for real-time traffic management while preserving the data privacy requirements.

INDEX TERMS Blockchain, connected vehicle, data integrity, data privacy, traffic management, vehicular network, zero-knowledge range proof.

I. INTRODUCTION

Modern traffic management systems utilize a large amount of vehicular data (such as vehicles' identification number, location, trajectory, etc.) for real-time decision making. The ever-growing incorporation of real-time traffic data from connected vehicles into these traffic management systems brings further data security and privacy challenges. Therefore, assurance of the integrity and privacy of traffic data over its entire life-cycle is a critical aspect of the design, implementation, and operation of such traffic management systems.

For integrity, centralized traffic management systems and their data centers can be attacked by processing malicious messages containing false traffic and vehicular data sent from connected vehicles in the vehicular networks [1]. These malicious messages can include false information about vehicle identification number, location, trajectory, etc. Without an effective defense mechanism, malicious data could lead to severe consequences in a transportation such as congestions [1] and collisions [2].

For privacy, commuters' fear of leaking personal information and the regulatory requirements for compliance in protecting data privacy are the primary concern when designing traffic management systems. For instance, there are regulatory requirements on data privacy such as the General Data Protection Regulation and the California Consumer Privacy Act with implementation dates in 2018 and 2020, respectively [3] [4]. In addition, real-time traffic feeds from passing vehicles into traffic management systems may be exploited by a successful attacks to extract sensitive information about the commuters. The problem becomes worse when the raw data stream contains commuters' private information that is not needed for the operation the traffic management systems.

In addition to the integrity and privacy issues of traffic data, centralized traffic management systems suffer from a single point of failure. In this paper, using the permissioned blockchain platform of Hyperledger Fabric [5], we propose a blockchain-based, decentralized traffic management architecture that leverages the vehicular networks of connected vehicles (CV) and edge nodes (roadside units and toll stations). As depicted in Fig. 1, over a large geographic

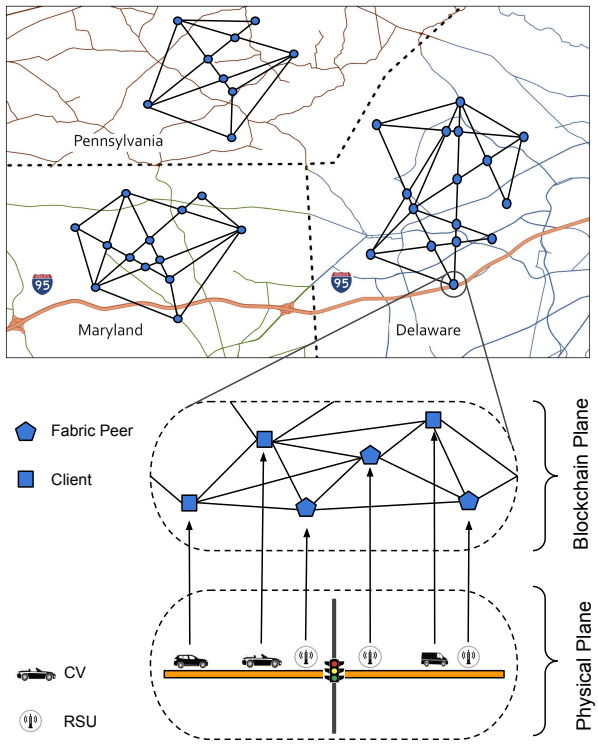

FIGURE 1. Physical and blockchain planes of connected vehicular networks.

area, there exist multiple CV-based vehicular networks. For instance, the transportation authority of each municipality, county, and province or state may be responsible for the traffic management over its respective area of jurisdiction. Also depicted in Fig. 1, the traffic management architecture of each transportation authority is composed of the physical plane and the blockchain plane. Vehicles and edge nodes form a vehicular network in the physical plane. In contrast, in the blockchain plane, traffic management functions are implemented in smart contracts. Connected vehicles, acting as clients, send vehicular information in transactions to the edge nodes that act as Hyperledger Fabric peers to execute transactions, order transactions vi a consensus protocol, and validate transactions before committing them into the blockchain, an immutable transaction ledger maintained within a distributed network of peers. In addition, the tamper-proof nature of the blockchain ensures the integrity of the traffic data.

To preserve privacy for the connected vehicles within each Hyperledger Fabric blockchain network (*intra*-blockchain privacy), mechanisms such as channels, private transactions, access control polices, and zero-knowledge proof (ZKP) based schemes of Identity Mixer and Zero-Knowledge Asset Transfer (ZKAT) have been adopted by or proposed for Hyperledger Fabric [6]. However, it is not obvious to facilitate *inter*-blockchain privacy when vehicles traverse across the boundary between two areas under different jurisdiction and need to switch from one traffic management system into another in a privacy-preserving manner.

We address the problem of data integrity and privacy for CV-based traffic management systems in multiple vehicular networks. This paper makes the following contributions:

- We present a decentralized and location-aware traffic data management architecture for multiple blockchain-based connected vehicular networks scenario. The architecture provides a novel design of transforming centralized traffic data management systems into decentralized blockchain-based networks and maintained vehicular data as digital records
- We propose the concept of *gateway* that resides between two adjacent blockchain-based traffic management systems to switch traveling vehicles from one blockchain into another. The gateway is responsible for verifying the information of an incoming vehicle, preventing spoofing attacks from malicious vehicles, and logging into the 'entering' traffic management system on behalf of the traveling vehicle. Specifically, we articulate the design of the gateway by developing a non-interactive zero-knowledge range proof (ZKRP) scheme, where a traveling vehicle (acting as a prover) sends ZKRP-encrypted message to the gateway (acting as a verifier) to validate the information of the vehicle without revealing any sensitive data.
- We prototype the proposed architecture and gateway on Hyperledger Fabric with Hyperledger Ursa [7] cryptographic library. In the proof-of-concept experiments, we successfully develop our blockchain-based traffic management system to protect traffic data of connected vehicles against potential attacks while preserving their privacy. To measure the system performance, we analyze transaction latency, throughput and success rate based on Hyperledger Caliper benchmark tool. The results show that our system is effective and feasible for decentralized traffic management.

The remainder of the paper is organized as follows. Section II reviews the background knowledge of cryptographic commitment and zero-knowledge proof. In Section III, we describe the architecture design of gateway based on ZKRP and depict its operational workflow with two blockchain networks. In Section IV, we perform extensive experiments to evaluate the performance of the prototyped blockchain network and the gateway validation process. Also, we discuss the robustness of the proposed architecture against potential attacks. We review related work in Section V and conclude the paper in Section VI.

II. BACKGROUND KNOWLEDGE

A. CRYPTOGRAPHIC COMMITMENT

A cryptographic commitment scheme allows a prover to compute a value that hides some secret without ambiguity, in the sense that no one later will be able to argue that this value corresponds to a different secret. In other words, given the impossibility to change the hidden secret, we say that the prover commits to that secret. A commitment scheme has the following two properties:

1) **Binding.** Given a commitment y, it is hard to compute a different pair of secret δ and random number γ whose commitment is also y. This property guarantees that there is no ambiguity in the commitment scheme. Thus, after y is published, it is hard to open it to a different value.
2) **Hiding.** It is hard to compute any information about the secret δ given the commitment y.

Formally, a commitment scheme is defined by algorithms *Commit* and *Open* as follows:

1) Given secret δ and random value γ, $Commit(\delta, \gamma)$ computes a commitment y as the output that hides the actual information δ such that it is hard to compute secret δ' and random value γ' that satisfies $Commit(\delta', \gamma') = Commit(\delta, \gamma)$. In particular, it is hard to invert function $Commit$ to find δ or γ.
2) Given the commitment y, secret δ and random value γ, $Open(y, \delta, \gamma)$ returns true if and only if $y = Commit(\delta, \gamma)$.

Commitment schemes are used in zero-knowledge proofs. Specifically, we propose to extend the Pedersen commitment [8]. Given group $\mathbb{Z}_p$ of prime order p, elements g and h, and random value γ, the commitment for secret δ is computed as follows:

$$y = Commit(\delta, \gamma) = g^{\delta} h^{\gamma}. \tag{1}$$

B. ZERO-KNOWLEDGE PROOF

Zero-knowledge proof was proposed in 1989 by Goldwasser, Micali, and Rackoff [9]. In the context of cryptography, a ZKP protocol is a method by which one party, termed prover, can prove, through a cryptographic commitment scheme, to another party, termed verifier, that they know a secret δ, without conveying any information apart from the fact they know the secret δ [10]. A zero-knowledge set membership (ZKSM) proof enables a prover to prove that a secret δ lies in a given set $[u, v]$. We describe ZKSM based on the notation by Camenisch et al. [11]:

$$PK\{(\delta, \gamma) : y = g^{\delta} h^{\gamma} \wedge (u \leq \delta \leq v)\}\,, \tag{2}$$

where $y = g^{\delta} h^{\gamma}$ is a commitment of the secret $\delta \in [u, v]$ using the random value γ. In other words, the above proof will convince the verifier that the secret in the commitment y lies in the set $[u, v]$.

In this paper, we focus on a particular kind of ZKP, called zero-knowledge range proof (ZKRP), which is closely related to ZKSM protocols. The first schemes of ZKRP protocols were proposed in 1995 by Damgård [12] and in 1997 by Fujisaki et al. [13]. However, these schemes were not efficient to be used in practice. The first practical ZKRP scheme was proposed by Boudot in 2000 [14] and followed by the work accomplished by Schoenmakers in his presentations [15] and [16]. The main difference is that ZKRP works with numeric intervals instead of generic sets used in ZKSM, which makes ZKRP a special case of ZKSM. ZKRP allows the blockchain network to validate that a secret number is within a known range without disclosing the secret number. For example, in the context of payment systems, it is possible to validate that a payment-amount is positive without disclosing the amount, which is done by Monero [17]. Moreover, ING Bank described how to implement ZKRP protocol in Ethereum [18]. Therefore, ZKRP can be applied to many kinds of decentralized applications that have numeric intervals along with other requirements, such as e-voting systems [19] [20] and e-auction systems [21] [22].

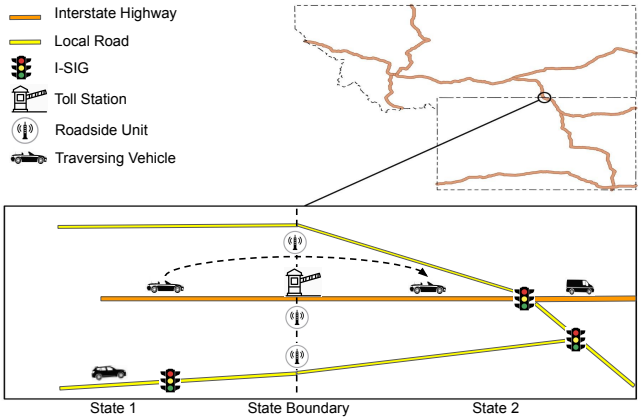

FIGURE 2. Components of gateway and its operation scenario.

III. ARCHITECTURE OF THE GATEWAY

In this section, we describe the proposed gateway architecture in two subsections. In Subsection A, we describe the detailed steps for gateway design with the proposed ZKRP protocol. In Subsection B, we explain the workflow of gateway. By referring to Fig. 2, we first describe the following components which take part in the proposed architecture:

- I-SIG: Intelligent Traffic Signal System. It takes arrival vehicle information as input and generates optimal signal plans at intersections. I-SIG system has been deployed in New York City, City of Tampa, and State of Wyoming since 2016 [23].
- Gateway: The gateways act as verifiers for validating traversing vehicles and consist of RSUs at state boundaries.
- Traversing Vehicle: The traversing vehicle is the commuter who wants to switch the vehicular network without revealing sensitive information to gateways (verifiers).
- Blockchain: Blockchain (in our prototype, Hyperledger Fabric) is utilized as a distributed ledger for the architecture, which manages vehicular data and serves as a tamper-proof log for intelligent traffic management systems.

Our architecture brings a novel design approach to facilitate inter-network operations while preserving data privacy. We use the scenario depicted in Fig. 2 to illustrate an instance when a vehicle traverses cross the boundary of two states over a physical vehicular network and switches

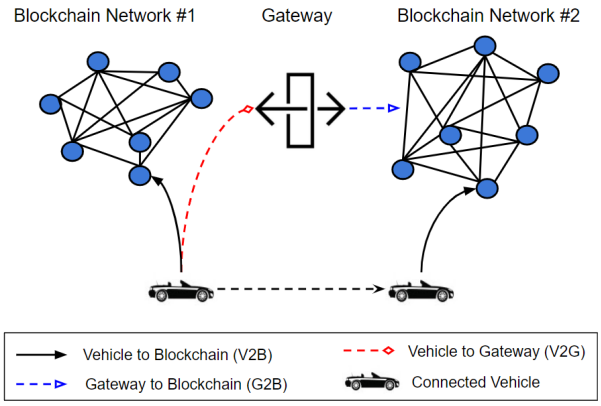

FIGURE 3. Function of gateway mechanism.

between two blockchain networks. The ZKRP protocol will preserves the privacy and integrity of CVs traversing across states or localities. Within the communication range of a gateway, a traversing connected vehicle acts as a prover to prove its vehicular information to the gateway, a verifier, with a ZKRP-based encrypted message. Therefore, it can pass the boundary and switch to a different blockchain network without revealing any sensitive information.

For intra-network, we design and develop blockchain-based vehicular networks on Hyperledger Fabric platform. Each blockchain network maintains a regional (e.g., statewide) distributed ledger for recording and sharing vehicular data as input for traffic management systems. After registration, connected vehicles can broadcast their vehicular data to blockchain network by submitting transaction requests. Transactions will be validated by Hyperledger Fabric peers and recorded permanently on the ledger.

A. GATEWAY DESIGN WITH ZKRP SCHEME

As shown in Fig. 3, we introduce gateways that are deployed on the boundaries between blockchain-based vehicular networks for seamlessly switching from one blockchain network into another. To preserve the privacy of the traveling vehicle, we describe how to construct the ZKRP protocol step by step for the gateway module. The traversing vehicle acts as a prover to prove its vehicular information (e.g., location) to a gateway, acting as a verifier, in a ZKRP-based encrypted message. As a result, the vehicle passes the boundary and switch the blockchain network without revealing any sensitive information to the other parties.

1) Interactive Zero-Knowledge Proof

Most available ZKP protocols described in the literature are interactive. In general, the prover must answer the challenge message sent by the verifier in order to convince him/her that the proof is valid. Scheme Version 1 describes an interactive ZKP in the context of gateway.

Scheme Version 1 Interactive ZKP

1) The traversing CV wants to prove to the gateway that it comes from the location δ: compute the commitment based on the discrete logarithm [24] of $y = g^{\delta}$ to the base g.
2) The traversing CV picks a random $v \in \mathbb{Z}_p$, computes $t = g^v$ and sends t to the gateway.
3) The gateway picks a random $c \in \mathbb{Z}_p$, and send it back as a challenge message to the traversing CV.
4) The traversing CV computes $r = v - c\delta$ and returns r to the gateway.
5) The gateway checks whether $g^r y^c \equiv t$. This holds because $g^r y^c = g^{v-c\delta} g^{\delta c} = g^v = t$.

2) Non-Interactive Zero-Knowledge Proof

Interactive ZKP is not suitable for the gateway since it would increase the communication overhead for verification and cannot meet the gateway's real-time operation requirements. Fiat-Shamir heuristic [25] is a generic technique that converts interactive ZKP schemes into non-interactive protocols. It allows replacing the interactive step 3) in Scheme Version 1 with a non-interactive random oracle function. In practice, we can use a cryptographic hash function [26] instead. The non-interactive ZKP is shown in Scheme Version 2 [27].

Scheme Version 2 Non-interactive ZKP

1) The traversing CV wants to prove that it comes from the location δ: compute the commitment based on the discrete logarithm of $y = g^{\delta}$ to the base g.
2) The traversing CV picks a random $v \in \mathbb{Z}_p$, computes $t = g^v$.
3) The traversing CV computes $c = H(g, y, t)$, where H is a cryptographic hash function [26].
4) The traversing CV computes $r = v - c\delta$. The resulting proof is the pair (t, r). As r is an exponent of g, it is calculated modulo $q - 1$.
5) The gateway checks whether $g^r y^c \equiv t$.

3) Non-Interactive Zero-Knowledge Range Proof

The secret δ can be decomposed into $\delta_j u^j$ $(0 \leq j \leq l)$ to obtain ZKRP [28] as follows:

$$\delta = \sum_{j=0}^{l} \delta_j u^j. \tag{3}$$

Therefore, if each δ_j belongs to the interval $[0, u)$, we have $\delta \in [0, u^l)$. Scheme Version 2 can be transformed into Scheme Version 3.

Scheme Version 3 works for the range $[0, u^l)$. In order to obtain ZKRP on an arbitrary range $[a, b]$, we propose to apply an improvement of a folklore reduction described by Schoenmakers in [15] and [16]. Instead of trying to prove range proof through a square decomposition, the folklore reduction is a more efficient method based on bit decomposition [29].

Scheme Version 3 Non-interactive ZKRP for interval $[0, u^l)$

1) The traversing CV wants to prove that it comes from the location δ: compute the commitment based on the discrete logarithm of $y = g^\delta$ to the base g and $\delta \in [0, u^l)$.
2) The traversing CV picks a random $v_j \in \mathbb{Z}_p$ for every $j \in \mathbb{Z}_l$, and computes $t_j = g^{v_j}$.
3) The traversing CV computes $c = H(g, y, t)$, where H is a cryptographic hash function [26].
4) The traversing CV computes $r_j = v_j - c\delta$ for every $j \in \mathbb{Z}_l$. The resulting proof is the pair (t_j, r_j). As r_j is an exponent of g, it is calculated modulo $q - 1$.
5) The gateway checks whether $g^{r_j} y^c \equiv t_j$ for every $j \in \mathbb{Z}_l$.

In the context of our range proof construction, suppose that $u^{l-1} < b < u^l$. To prove $\delta \in [a, b]$, it suffices to show that:

$$\delta \in [a, a + u^l] \quad and \quad \delta \in [b - u^l, b]. \tag{4}$$

As illustrated in Fig. 4, proving that secret δ lies in above subsets can be derived from the previous proof that $\delta \in [0, u^l)$, respectively:

$$\delta \in [a, a + u^l) \iff \delta - a \in [0, u^l), \tag{5}$$

$$\delta \in [b - u^l, b) \iff \delta - b + u^l \in [0, u^l). \tag{6}$$

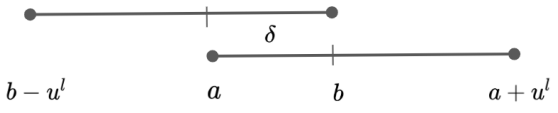

FIGURE 4. Two overlapping ranges illustrating secret δ lies in the range $[a, b]$.

As a result, the range proof can be extended for an arbitrary range $[a, b]$. The final non-interactive range proof construction process is shown in Scheme Version 4.

Scheme Version 4 Non-interactive ZKRP for interval $[a, b]$

1) The traversing CV wants to prove that it comes from the location δ: compute the commitment based on the discrete logarithm of $y = g^\delta$ to the base g and $\delta \in [a, b]$, namely $\delta \in [a, a + u^l] \cap \delta \in [b - u^l, b]$.
2) The traversing CV picks a random $v_j \in \mathbb{Z}_p$ for every $j \in \mathbb{Z}_l$, and computes $t_j = g^{v_j}$.
3) The traversing CV computes $c = H(g, y, t)$, where H is a cryptographic hash function [26].
4) The traversing CV computes $r_j = v_j - c\delta$ for every $j \in \mathbb{Z}_l$. The resulting proof is the pair (t_j, r_j). As r_j is an exponent of g, it is calculated modulo $q - 1$.
5) The gateway checks whether $g^{r_j} y^c \equiv t_j$ for every $j \in \mathbb{Z}_l$.

B. WORKFLOW OF THE GATEWAY

The workflow of the proposed gateway that resides between two adjacent blockchain-based traffic management systems is depicted in Fig. 5. In the beginning, a CV riding over the area covered by blockchain network #1 provides vehicular data to the corresponding traffic management systems (e.g., I-SIG). When the CV (a prover) wants to cross the boundary into another vehicular network, it first encrypts its vehicular information using the proposed ZKRP protocol and then broadcasts a proof request to a crossing gateway (a verifier). Then, the gateway validates the ZKRP encrypted information. If the ZKRP encrypted information can be verified, the gateway confirms the request with the traversing CV and switches the vehicle into blockchain network #2. After switching the network, the CV starts sharing vehicular data directly with blockchain network #2.

IV. EXPERIMENTS AND EVALUATION

A. EXPERIMENTAL SETUP

We conduct extensive experiments to evaluate the performance of the prototyped blockchain-based traffic management systems and the ZKRP gateway. The prototype is implemented in three interworking modules: i) Blockchain Network; ii) Reverse Geocoding; iii) Gateway Validation. The Blockchain Networks are developed on Hyperledger Fabric v1.2. The Reverse Geocoding is developed on JSFiddle [30] with the Google Maps Geocoding API [31]. The Gateway Validation is developed by using the Hyperledger Ursa cryptographic library. To measure the performance of blockchain-based system, we run benchmark tests using Hyperledger Caliper [32]. 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. MODULE 1: BLOCKCHAIN NETWORK

1) Development

Hyperledger Composer [33] is a framework and toolset to build and run applications on top of Hyperledger Fabric, which provides four programmable portions: model file (.cto), script file (.js), access control list (.acl) and query file (.qry). The model file defines all the objects in the network while smart contracts are written in the script file. Hyperledger Fabric provides access control list to facilitate access polices for different participants. As for query file, it works similar to conventional database query operations. These files are finally packaged into one business network archive (.bna) file and deployed into Hyperledger Fabric blockchain network.

As shown in Fig. 6, we develop the blockchain-based traffic management system on Hyperledger Fabric using Composer, which maintains regional ledgers for recording and sharing vehicular data containing vehicle identity number, location, trajectory and timestamp. Data structures of participants and vehicular data are defined in the model file. Smart

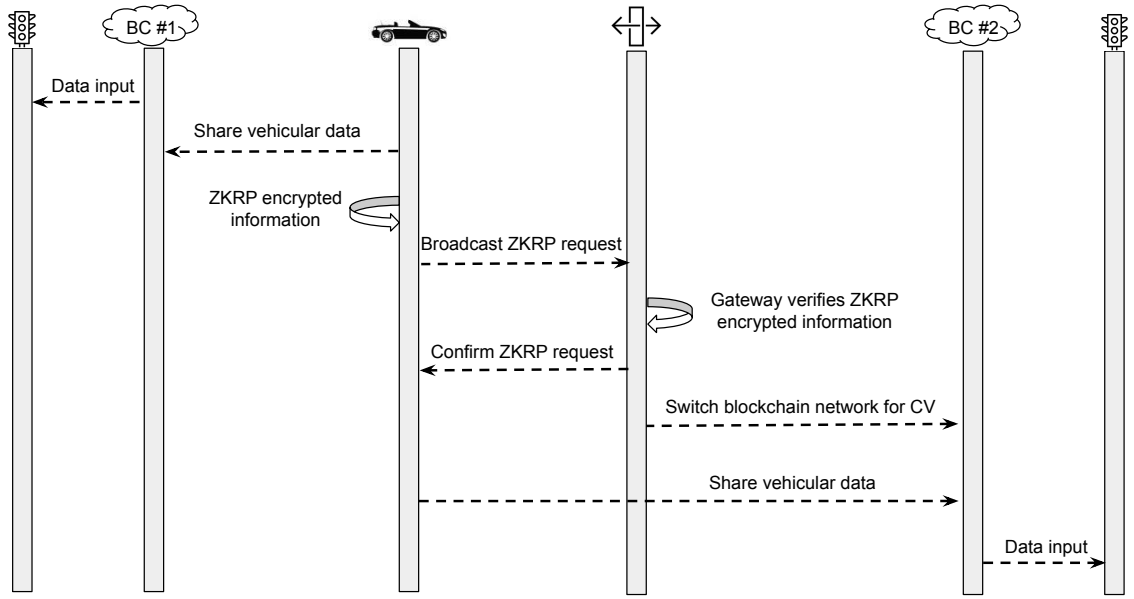

FIGURE 5. Workflow of gateway between two blockchain-based vehicular networks.

contracts including functionalities of information recording and retrieval are coded in the script file.

By utilizing access control policy, we enable clients to own their generated vehicular data and enforce the system to determine which participants are allowed to read, write, and update data. We define the access control policy for our system with the follows components:

- Participant: It indicates the participants involved in the access control procedure.
- Operation: It defines the actions governed by the access control policy. Three actions are supported in our system: READ, WRITE, and UPDATE.
- Resource: It indicates the vehicular data which the access control policy applies to.
- Condition: It defines the conditional statements over multiple variables. Our system can support combinations of multiple conditional statements to serve complex access control design.
- Action: It indicates the final decision after executing the access control policy. It can be either ALLOW or DENY.

For instance, the policy below states it allows a client to only READ his/her own vehicular data from the ledger:

```
rule Client_Can_Read_Vehicular_Data {
  description: "Client can only read
        his/her own vehicular data."
  participant(p): "org.bvn.prototype
        .Client"
  operation: READ
  resource(r): "org.bvn.prototype
        .CV_Data"
  condition: "r.owner.getIdentifier()
        === p.getIdentifier()"
  action: ALLOW
}
```

Queries defined in the query file (.qry) contain WHERE clause to define the criteria by which vehicular data or participants are selected. In our design, the query language can return specific results from the ledger if the given condition is satisfied. For instance, the query below can filter out speeding connected vehicles that are faster than 70 mph:

```
query Select_Speeding_Vehicles {
  description: "Select speeding vehicles
        that are faster than 70 mph."
  statement:
        SELECT org.bvn.prototype.CV_Data
          WHERE (Trajectory.speed > 70)
}
```

Hyperledger Composer also provides a web interface for interacting with the blockchain network. Each participant has an ID registry for connecting to the blockchain network as shown in Fig. 7. A traffic management authority (e.g., US Department of Transportation) acts as the administrator to issue access permissions for the other participants including connected vehicles, RSUs, and traffic management systems (e.g., I-SIG).

2) Performance of Blockchain Network

To evaluate the performance of the blockchain-based traffic management system, we conduct benchmark tests using Hyperledger Caliper benchmark tool with different endorsement policies. These endorsement policies define the set of peers need to agree on the results of a transaction before it can be committed to the ledger. The latency measures the time of a transaction from submission by the client until it is

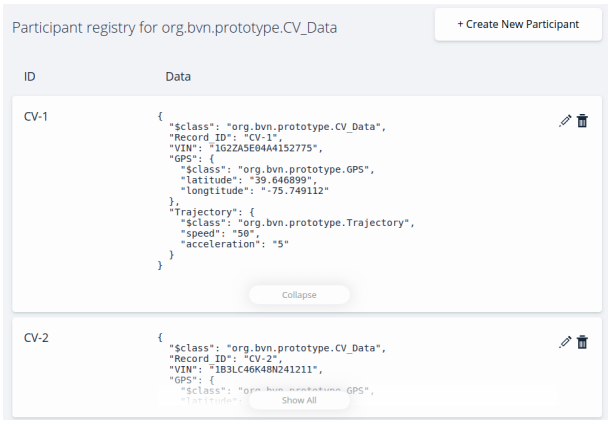

FIGURE 6. Vehicular data in blockchain network.

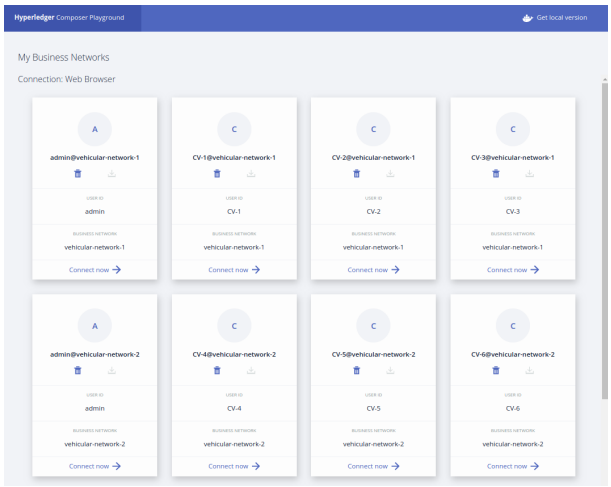

FIGURE 7. Blockchain network login window.

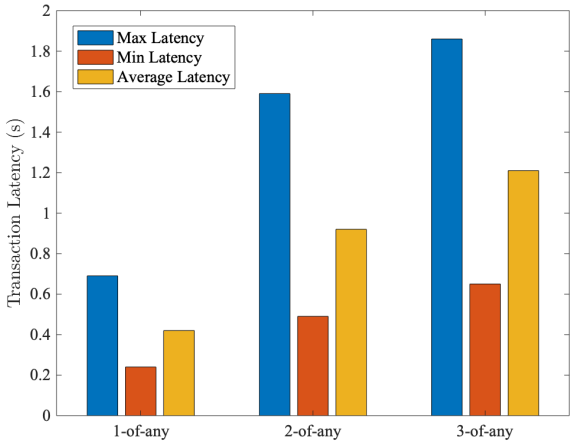

FIGURE 8. Transaction latency vs. Hyperledger Fabric endorsement policies.

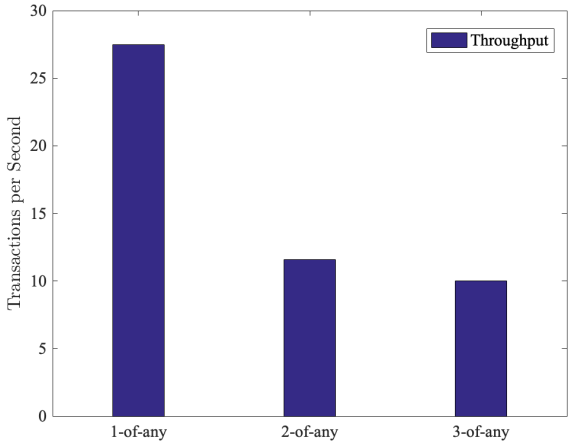

FIGURE 9. Transaction throughput vs. Hyperledger Fabric endorsement policies.

processed and written into the ledger. Maximum, minimum and average latency for the test cycles are shown in Fig. 8. With the increasing number of peers, the transaction latency increases. The throughput measures the flow rate of processed transactions through the blockchain network, in the unit of transactions per second, during the test cycle. As shown in Fig. 9, the transaction throughput decreases with the increasing number of peers. The choice of endorsement policy can impact transaction latency and throughput because more endorsing peers increase the complexity of the endorsing process.

The success rate measures how many transactions out of the submitted transactions have been successfully processed and written into the blockchain during a test cycle. A failed transaction could be due to the time-outs, wrong network configuration or bugs in smart contracts. For all the test cycles with different endorsement policies, our blockchain network can always achieve 100% success rates.

C. MODULE 2: REVERSE GEOCODING

The Blockchain Network module records geographic information in GPS coordinates to serve traffic management systems (e.g., I-SIG). Specifically, the Gateway Validation module is designed to take the ZIP Code as the secret for ZKRP. For this reason, we add the Reverse Geocoding module, which serves as a critical step to convert geographic information from GPS coordinates (latitude and longitude) into integer values of the ZIP Code for the Gateway Validation module.

Reverse geocoding services are available through APIs and other web services as well as mobile phone applications [34]. In our study, we use Google Maps Geocoding APIs to enable the Reverse Geocoding function on JSFiddle, which is an online integrated development environment (IDE) for developing and testing user-created HTML, CSS and JavaScript codes. For instance, if a traversing vehicle's current GPS coordinate is (45.091466, -107.349952), the Reverse Geocoding module converts the geographic information into ZIP Code 59089, which is used to generate a ZKRP-encrypted message for Gateway Validation in next subsection.

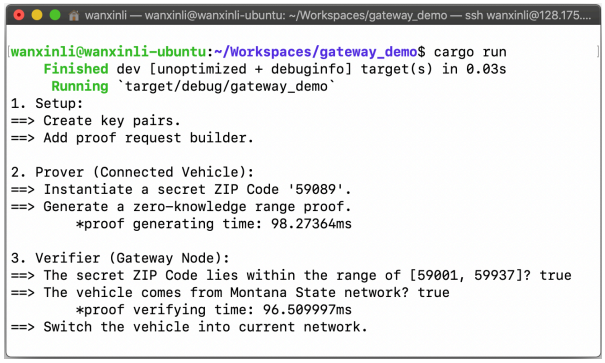

FIGURE 10. Running process of Gateway Validation module.

D. MODULE 3: GATEWAY VALIDATION

In the design of Gateway Validation, ZKRP is a method by which one traversing vehicle proves to the gateway (verifier) that it comes from a specific blockchain network $N(\delta)$ covering location δ, without conveying any information apart from the fact that it comes from the location δ. The proposed ZKRP scheme protects the privacy of vehicular information in the process of switching blockchain networks.

We develop the Gateway Validation module using Hyperledger Ursa, which is an active incubating project for providing trusted cryptographic libraries for distributed systems. Hyperledger Ursa provides Rust APIs for constructing the ZKRP scheme. As shown in Fig. 10, the Gateway Validation module runs in three processes of setup, prover action, and verifier action.

1) Process Clarification

In the setup process, the Gateway Validation module first generates private and public key pairs for all participants, which works similarly to attribute-based credentials [35]. Then, the module executes `sub_proof_request_builder` to invoke the `new_sub_proof_request_builder` function of the Ursa Verifier library to set the interval for range proof. In this example, we use the range [59001, 59937] to represent the ZIP Code range for the state of Montana:

```
let mut sub_proof_request_builder =
  Verifier::new_sub_proof_request
  _builder().unwrap();
sub_proof_request_builder
  .add_predicate("ZIPCode", "GE", 59001)
  .unwrap();
sub_proof_request_builder
  .add_predicate("ZIPCode", "LE", 59937)
  .unwrap();
let sub_proof_request =
  sub_proof_request_builder
  .finalize().unwrap();
```

On the prover side, we instantiate a secret value, e.g., 59089, as the ZIP Code for the traversing vehicle's location, which is converted by the reverse geocoding module. The traversing vehicle, known as the prover, generates a ZKRP-based proof for this secret value by executing `proof_builder` to invoke `new_proof_builder` function of the Ursa Prover library. The running time for proof generation is, on average, 98 ms in our experimental setting. The `proof_builder` works as follows:

```
let mut proof_builder = Prover:
   :new_proof_builder().unwrap();
proof_builder
   .add_common_attribute("ZIPCode")
   .unwrap();
proof_builder.add_sub_proof_request(
    &sub_proof_request,
    &credential_schema,
    &non_credential_schema,
    &cred_signature,
    &cred_values,
    &cred_pub_key,
    None,
    None,
  ).unwrap();
```

On the verifier side, upon receiving the proof, the gateway runs `proof_verifier` to verify if the secret value lies within the range of [59001, 59937] without revealing the actual information. The `proof_verifier` invokes `new_proof_verifier` function from the Ursa Verifier library. If the response is positive, the gateway validates and switches the network for the traversing vehicle. The running time for verification is, on average, 97 ms in our experimental setting. The `proof_verifier` is shown as follows:

```
let mut proof_verifier = Verifier:
   :new_proof_verifier().unwrap();
proof_verifier.add_sub_proof_request(
   &sub_proof_request,
   &credential_schema,
   &non_credential_schema,
   &cred_pub_key,
   None,
   None,
  ).unwrap();
let is_valid = proof_verifier
  .verify(&proof, &proof_request_nonce)
  .unwrap();
```

2) Performance of ZKRP Scheme

To evaluate the performance of the ZKRP scheme, we conduct extensive experiments to analyze the effect of varying the size of secrets and the number of secrets. The size of secrets in ZKRP is measured by its set range and length. The default secret range is 936 from the interval [59001, 59937] defined in Section IV-D1. We first changed the range of secrets from 2 to 2^5, 2^{10}, 2^{15} and 2^{31}, and the results showed that both proof generating time and verifying time

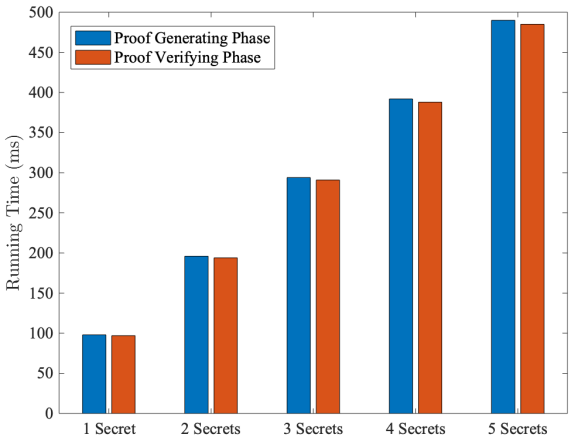

FIGURE 11. Running time of Ursa ZKRP vs. the number of secrets.

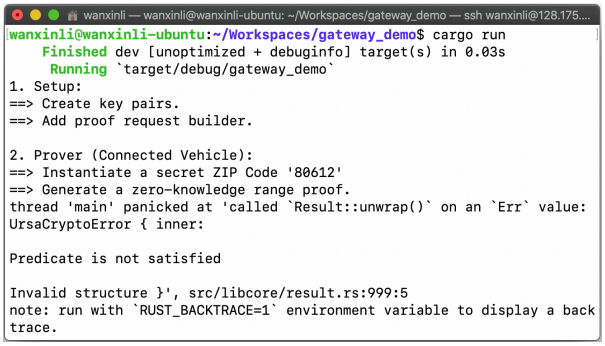

FIGURE 12. Response against gateway spoofing attack.

are constant regardless of the secret range, and the time remains around 98 ms and 97 ms, respectively. The default length of the secret instance is 5 digit (ZIP Code). We then changed the length of secrets from 1 to 3, 5, 7 and 9 digits, and the results showed that both proof generating time and verifying time are also constant regardless of the secret length, and the time remains around 98 ms and 97 ms, respectively, for each secret.

Our ZKRP scheme can offer constant proof generating and verifying time because the commitment of a secret is computed by a hash function [26] in Scheme Version 4 (Section III-A). In the experiments, we invoke `new_proof_builder` and `new_proof_verifier` functions from the Ursa library, which utilize a HashMap to compute and verify the commitment. As a result, proof generating and verifying time are independent from the size of secret values. This allows our ZKRP scheme to have more flexibility for verifying different numerical secret values (e.g., ID number, credit card number, etc,.) without sacrificing security and efficiency. We then increase the number of secrets from 1 to 2, 3, 4 and 5 for each client, and measure the running time of the proof generating and verifying phases. Because each secret is processed sequentially, both proof generating and verifying running time showed a linear growth with the increasing number of secrets in Fig. 11. Given our scenario where the secrets are independent, in theory, multiple secrets can be proved in parallel to achieve constant time. However, the linear growth in running time cannot be avoided in this work because the current Hyperledger Ursa does not have parallel computing capability.

E. DISCUSSION ON POTENTIAL ATTACKS

In this subsection, we discuss the robustness of the proposed blockchain and ZKP inspired architecture for traffic management against potential attacks.

1) Vehicular Data Attack

Conventional traffic management systems can be attacked by tampering their centralized database [1]. Our proposed system can reject tampering existing vehicular data due to the immutable feature of blockchain which ensures data integrity by recording data on a distributed ledger. Once the information of an arriving vehicle is recorded on the distributed ledger, it does not allow any participant to tamper it. If there is any attempt to tamper the ledger, our system will immediately reject this action.

2) Gateway Spoofing Attack

A gateway spoofing attack happens when an attacker belonging to one blockchain network $N(\epsilon)$ spoofs the gateway that it belongs to another blockchain network $N(\delta)$. In this scenario, validating vehicular information and switching networks become challenging tasks if the attacker successfully injects false login status among multiple blockchain networks. For instance, if the attacker intends to spoof that it is coming from the Montana State network, but it is actually coming from the Colorado State network (e.g., ZIP Code 80612). The Gateway Validation module is resistant to gateway spoofing attacks by incorporating the ZKRP protocol, which is shown in Fig. 12.

3) Eavesdropping Attack

Our proposed gateway mechanism can protect the system from eavesdropping attacks. In such attacks, the malicious attacker intercepts the message between the data sender and data receiver in order to recover the secret information or gain access to the sensitive information. In the worst case, if the message has been accessed by the attacker, it cannot reveal the actual information since the vehicular information is ZKRP-encrypted. As a result, no sensitive information is disclosed if the eavesdropping attack happens.

V. RELATED WORK

A. BLOCKCHAIN RESEARCH IN TRANSPORTATION

In recent years, investigating the blockchain paradigm in the general transportation field has attracted a great deal of attention [36] [37] [38] [39]. Two main applications of the blockchain technology to the transportation industry are freight tracking and supply chain management. For instance, IBM has been working with Walmart to develop an efficient blockchain-based tracking system for the food supply

chain [40], which involves the transportation of merchants. In such a scenario, the blockchain technology helps to reduce tracking time for goods from weeks to seconds. Blockchain technology is also a candidate solution in forensic investigation. Hossain et al. [41] proposed FIF-IoT, which is a forensic investigation framework using a public blockchain to find facts in criminal incidents in IoT-based systems. Besides, Guo et al. [42] proposed a blockchain-inspired "proof of event" mechanism for accident event recording in CV networks. A decentralized trust management system for vehicular networks was proposed in [43]. In this paper, vehicles are able to query the trust values of neighbors and then assess the credibilities of received messages using blockchain technologies.

B. SPOOFING ATTACKS IN VEHICULAR NETWORKS

Vehicular networks are vulnerable to cyberattacks. Amoozadeh et al. [44] presented the spoofing effects of security attacks on the communication channel as well as sensor tampering. Dominic et al. [45] proposed a risk assessment framework for CV applications consisting of an automated driving reference architecture and threat model. In a recent study, Chen et al. [1] showed that the I-SIG system is vulnerable at the signal control algorithm level. Due to limited computation power, the signal controller cannot handle data validation in real-time processing requirement, 5-7 seconds. They conducted their V2I attacking strategy by spoofing arrival vehicular information, which caused congestion at intersections.

C. ZERO-KNOWLEDGE PROOF FOR BLOCKCHAIN

Zero-knowledge proof enables one party to prove the knowledge to another party without conveying any information about the knowledge. Zcash implements the zero-knowledge succinct non-interactive arguments of knowledge (ZK-SNARK) to protect the transaction privacy in cryptocurrency network [46]. Koens et al. [18] proposed an efficient zero-knowledge range proof in Ethereum without the interactive communications between participants. In addition, Bulletproofs are proposed for efficient range proofs on committed values, which are short non-interactive zero-knowledge proofs without a trusted setup process [47].

VI. CONCLUSION

In this paper, we propose a decentralized and location-aware traffic management system to protect data integrity and privacy in a scenario of multiple blockchain-based connected vehicular networks. Our system innovatively incorporates zero-knowledge range proof into a gateway mechanism to verify connected vehicles traversing between adjacent blockchain networks without revealing any sensitive information. We develop the Blockchain Network module on the Hyperledger Fabric platform, the Reverse Geocoding module on the Google Maps APIs, and the Gateway Validation module on the Hyperledger Ursa cryptographic library. For the blockchain network, we measure the benchmarks including transaction latency, throughput and success rate using the Hyperledger Caliper benchmark tool. For the ZKRP scheme, we measure the proof generating and verifying time under different settings. The results demonstrate that our proposed system is effective and feasible for decentralized traffic management.

REFERENCES

[1] Q. A. Chen, Y. Yin, Y. Feng, Z. M. Mao, and H. X. Liu, "Exposing congestion attack on emerging connected vehicle based traffic signal control." in NDSS, 2018.
[2] J. Blum and A. Eskandarian, "The threat of intelligent collisions," IT Professional, vol. 6, no. 1, pp. 24–29, 2004.
[3] Wikipedia contributors, "General data protection regulation," 2020. [Online]. Available: https://en.wikipedia.org/wiki/General_Data_Protection_Regulation
[4] Wikipedia contributors, "California consumer privacy act," 2020. [Online]. Available: https://en.wikipedia.org/wiki/California_Consumer_Privacy_Act
[5] "Hyperledger Fabric." [Online]. Available: https://www.hyperledger.org/projects/fabric
[6] E. Androulaki, S. Cocco, and C. Ferris, "Private and confidential transactions with hyperledger fabric," IBM Developer, 2018. [Online]. Available: https://developer.ibm.com/tutorials/cl-blockchain-private-confidential-transactions-hyperledger-fabric-zero-knowledge-proof
[7] "Hyperledger Ursa." [Online]. Available: https://www.hyperledger.org/projects/ursa
[8] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in Annual International Cryptology Conference. Springer, 1991, pp. 129–140.
[9] 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.
[10] Wikipedia contributors, "Zero-knowledge proof," 2020. [Online]. Available: https://en.wikipedia.org/wiki/Zero-knowledge_proof
[11] J. Camenisch, R. Chaabouni et al., "Efficient protocols for set membership and range proofs," in International Conference on the Theory and Application of Cryptology and Information Security. Springer, 2008, pp. 234–252.
[12] I. B. Damgård, "Practical and provably secure release of a secret and exchange of signatures," Journal of cryptology, vol. 8, no. 4, pp. 201–222, 1995.
[13] E. Fujisaki and T. Okamoto, "Statistical zero knowledge protocols to prove modular polynomial relations," in Annual International Cryptology Conference. Springer, 1997, pp. 16–30.
[14] F. Boudot, "Efficient proofs that a committed number lies in an interval," in International Conference on the Theory and Applications of Cryptographic Techniques. Springer, 2000, pp. 431–444.
[15] B. Schoenmakers, "Some efficient zeroknowledge proof techniques," in Workshop on Cryptographic Protocols, 2001.
[16] B. Schoenmakers, "Interval proofs revisited," in International Workshop on Frontiers in Electronic Elections, 2005.
[17] K. Li, R. Yang, M. H. Au, and Q. Xu, "Practical range proof for cryptocurrency monero with provable security," in International Conference on Information and Communications Security. Springer, 2017, pp. 255–262.
[18] T. Koens, C. Ramaekers, and C. Van Wijk, "Efficient zero-knowledge range proofs in ethereum," ING, blockchain@ ing. com, 2018.
[19] P. McCorry, S. F. Shahandashti, and F. Hao, "A smart contract for boardroom voting with maximum voter privacy," in International Conference on Financial Cryptography and Data Security. Springer, 2017, pp. 357–375.
[20] I. Damgård, M. Jurik, and J. B. Nielsen, "A generalization of paillier's public-key system with applications to electronic voting," International Journal of Information Security, vol. 9, no. 6, pp. 371–385, 2010.
[21] I. Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: Anonymous distributed e-cash from bitcoin," in 2013 IEEE Symposium on Security and Privacy. IEEE, 2013, pp. 397–411.
[22] M. O. Rabin, Y. Mansour, S. Muthukrishnan, and M. Yung, "Strictly-black-box zero-knowledge and efficient validation of financial transactions," in International Colloquium on Automata, Languages, and Programming. Springer, 2012, pp. 738–749.

[23] "CV Pilot Deployment Program." [Online]. Available: https://www.its.dot.gov/pilots
[24] J. Camenisch and M. Stadler, "Proof systems for general statements about discrete logarithms," Technical report/Dept. of Computer Science, ETH Zürich, vol. 260, 1997.
[25] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in Conference on the Theory and Application of Cryptographic Techniques. Springer, 1986, pp. 186–194.
[26] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in Proceedings of the 1st ACM conference on Computer and communications security. ACM, 1993, pp. 62–73.
[27] Wikipedia contributors, "Fiat–shamir heuristic," 2020. [Online]. Available: https://en.wikipedia.org/wiki/Fiat-Shamir_heuristic
[28] E. Morais, C. van Wijk, and T. Koens, "Zero knowledge set membership," 2018.
[29] R. Chaabouni, "Efficient protocols for set membership and range proofs," 2007, supervisors: Dr. Jan Camenisch (IBM ZRL), Prof. Abhi Shelat (IBM ZRL, University of Virginia), Prof. Serge Vaudenay (EPFL LASEC) Asiacrypt 2008 publication: http://infoscience.epfl.ch/record/128718. [Online]. Available: http://infoscience.epfl.ch/record/113794
[30] "JSFiddle Editor." [Online]. Available: https://jsfiddle.net
[31] "Google Maps Geocoding API." [Online]. Available: https://developers.google.com/maps/documentation/geocoding/start
[32] H. Caliper, "Hyperledger caliper architecture," Electronic Article. url: https://hyperledger. github. io/caliper/docs/2_Architecture. html (visited on 03/10/2019), 2019.
[33] "Hyperledger Composer." [Online]. Available: https://www.hyperledger.org/projects/composer
[34] S. Kumar, M. A. Qadeer, and A. Gupta, "Location based services using android (lbsoid)," in 2009 IEEE international conference on internet multimedia services architecture and applications (IMSAA). IEEE, 2009, pp. 1–5.
[35] M. Koning, P. Korenhof, G. Alpár, and J.-H. Hoepman, "The abc of abc: An analysis of attribute-based credentials in the light of data protection, privacy and identity," 2014.
[36] Y. Yuan and F.-Y. Wang, "Towards blockchain-based intelligent transportation systems," in 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2016, pp. 2663–2668.
[37] 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.
[38] 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, 2019.
[39] M. Baza, M. Nabil, N. Lasla, K. Fidan, M. Mahmoud, and M. Abdallah, "Blockchain-based firmware update scheme tailored for autonomous vehicles," in 2019 IEEE Wireless Communications and Networking Conference (WCNC). IEEE, 2019, pp. 1–7.
[40] "IBM Food Trust," https://www.ibm.com/blockchain/solutions/food-trust.
[41] M. Hossain, Y. Karim, and R. Hasan, "Fif-iot: A forensic investigation framework for iot using a public digital ledger," in 2018 IEEE International Congress on Internet of Things (ICIOT). IEEE, 2018, pp. 33–40.
[42] H. Guo, E. Meamari, and C.-C. Shen, "Blockchain-inspired event recording system for autonomous vehicles," arXiv preprint arXiv:1809.04732, 2018.
[43] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. Leung, "Blockchain-based decentralized trust management in vehicular networks," IEEE Internet of Things Journal, vol. 6, no. 2, pp. 1495–1505, 2018.
[44] M. Amoozadeh, A. Raghuramu, C.-N. Chuah, D. Ghosal, H. M. Zhang, J. Rowe, and K. Levitt, "Security vulnerabilities of connected vehicle streams and their impact on cooperative driving," IEEE Communications Magazine, vol. 53, no. 6, pp. 126–132, 2015.
[45] D. Dominic, S. Chhawri, R. M. Eustice, D. Ma, and A. Weimerskirch, "Risk assessment for cooperative automated driving," in Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy. ACM, 2016, pp. 47–58.
[46] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in 2014 IEEE Symposium on Security and Privacy. IEEE, 2014, pp. 459–474.
[47] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell, "Bulletproofs: Short proofs for confidential transactions and more," in 2018 IEEE Symposium on Security and Privacy (SP). IEEE, 2018, pp. 315–334.

WANXIN LI received the BSc and MSc degrees in computer science from the Chongqing University, Chongqing, China in 2015, and the University of Delaware, United States in 2017, respectively. He is currently working toward the Ph.D. degree at the University of Delaware. His research interests are in the area of blockchain, intelligent transportation systems (ITS), connected and autonomous vehicles, and Internet of Things (IoT). He is a member of IEEE.

HAO GUO received the B.S. and M.S. degrees from the Northwest University, Xi'an, China in 2012, and the Illinois Institute of Technology, Chicago, United States in 2014, and his Ph.D. degree from the University of Delaware, Newark, United States in 2020, all in computer science. His research interests include blockchain and distributed ledger technology, data privacy and security, cybersecurity, cryptography technology, and Internet of Things (IoT). He is a member of both ACM and IEEE.

MARK NEJAD is an Assistant Professor in the Department of Civil and Environmental Engineering at the University of Delaware. His research interests include connected and automated vehicles, network optimization, parallel and distributed computing, blockchain, and game theory. He has published more than thirty peer-reviewed papers in venues such as Transportation Science, IEEE Transactions on Parallel and Distributed Systems, and IEEE Transactions on Computers. He received several publication awards including the 2016 best doctoral dissertation award of the Institute of Industrial and Systems Engineers (IISE) and the 2019 CAVS best paper award from the IEEE VTS. He is a member of the IEEE and INFORMS.

CHIEN-CHUNG SHEN received his B.S. and M.S. degrees from National Chiao Tung University, Taiwan, and his Ph.D. degree from UCLA, all in computer science. He was a research scientist at Bellcore Applied Research working on control and management of broadband networks. He is now a Professor in the Department of Computer and Information Sciences of the University of Delaware. His research interests include blockchain, Wi-Fi, SDN and NFV, ad hoc and sensor networks, dynamic spectrum management, cybersecurity, distributed computing, and simulation. He is a recipient of NSF CAREER Award and a member of both ACM and IEEE.