Safety Warning! Decentralised and Automated Incentives for Disqualified Drivers Auditing in Ride-Hailing Services

Youshui Lu, *Member, IEEE*, Jingning Zhang, Yong Qi, *Member, IEEE*, Saiyu Qi, *Member, IEEE*, Yue Li, Hongyu Song, and Yuhao Liu

Abstract—Since 2011, the private ride-hailing companies Didi (2019), Uber (2019) and Lyft (2021) have expanded into more and more cities. These ride-hailing services (RHS) bring convenience to our life; however, at the same time they, also raise security concerns for users. For example, several recent news items show that a considerable number of registered drivers whose licenses have been revoked are still taking RHS orders on the respective platforms; this phenomenon directly leads to insecurity on part of its users and the bad reputation of the ride-hailing service provider (SP). The traditional solution to solve this problem is to periodically check the validity of the drivers' licenses; however, it is a considerably time-consuming and costly process since the SPs have to manually interact with the governing authorities. Therefore, in this paper, we have presented an auditable self-sovereign identity system (named AudiSSI), which provides an efficient approach for the SPs to manage their registered drivers' qualifications in a decentralized and automatic manner. Further, using smart contract technology, we propose a safety guarantee insurance in the form of an auditing contract to enable the RHS rider to check their driver's qualifications before the trip starts and get incentives once they detect a disqualified driver. We designed an incentive mechanism and have provided a game theoretical analysis. Finally, we implemented a prototype of AudiSSI and deployed it on Hyperledger Indy and Fabric to show that self-sovereign identity system for RHS driver with qualification auditing is efficient and technically feasible.

Index Terms—Auditing, blockchain, smart contract, ride-hailing, game theory, insurance scheme, decentralized system

1 Introduction

With the emergence of new transport technologies, ride-hailing services have become increasingly popular around the world in recent years. RHS allows a consumer to easily hail a ride and track the driver's location in real-time. Their advantage over traditional taxi services is due to the convenience of hailing services, e.g., ride requests at the touch of a button, price estimation, recommendation of frequent destinations, etc. To make these services available, the registered driver must pass rigorous due diligence and driver's qualification check.

Unfortunately, currently, the driver's qualifications check only occurs when the driver initially registers in the RHS platform, and the SP fails to disqualify the registered drivers with revoked licenses in a timely manner, which results in a large number of disqualified drivers continuing to take RHS orders. Therefore, it leads to security issues to the RHS users and affects the SP's reputation. To put this in perspective, in August 2018, a ride-sharing passenger in the eastern Chinese city of Wenzhou was raped and killed by a driver employed at China's largest ride-hailing firm Didi Chuxing[1]. Just three months later, another incident shocked China after a Didi passenger was killed in a similar circumstance [2]. In another example[3], Uber drivers have been implicated in a number of major scandals, including an attempted kidnapping, and a tragic hit-and-run incident that resulted in the death of a young girl. Following the incident, the government required Didi to suspend its carpooling service until the safety issues were been addressed.

Although governing authorities carry out legislations to regulate the RHS industry, it is challenging for the SPs to detect disqualified drivers when their licenses are revoked. As verifying the validity of one's identity information (e.g., driver's license) implies a time-consuming and costly process involving different centralised governing authorities (e.g., police Station), most SPs fail to regularly check their drivers' qualifications per a short time interval. On the other hand, the RHS is time-sensitive and it cannot tolerate much delay to maintain high-standard service quality. Therefore, an efficient solution is urgently required to combat this challenging issue in the realm of RHS industry.

In response, the nature of distributed ledger (blockchain) technologies (DLT) [15] could build trust in many cases, we propose to use blockchain to build a decentralised identity management system because DLT is particularly useful in business and governance situations that involve multiple

- *Youshui Lu, Jingning Zhang, Yong Qi, Saiyu Qi, Hongyu Song, and Yuhao Liu are with the Department of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China. E-mail: lucienlu@me.com, {jingning, hysong0101, lyuhao}@stu.xjtu.edu.cn, qiy@mail.xjtu.edu.cn, syqi@connect.ust.hk.*
- *Yue Li is with the Department of Economics, Vrije Universiteit Amsterdam, 1081 HV Amsterdam, The Netherlands, and also with the Risk Modelling Department, ABN AMRO Bank N.V, 1082 PP Amsterdam, The Netherlands. E-mail: yue.li@vu.nl.*

Manuscript received 12 Jan. 2021; revised 1 Aug. 2021; accepted 16 Aug. 2021. Date of publication 27 Aug. 2021; date of current version 3 Feb. 2023. (Corresponding authors: Saiyu Qi and Yue Li.)
Digital Object Identifier no. 10.1109/TMC.2021.3108012

distrustful parties to negotiate and execute electronic business transactions [23]. In many cases, such transactions require the ability to establish and validate identities and identity attributes, or to check whether or not they have been revoked [23]. Such a decentralised identity management system allows one's identity information to be synchronised and updated quickly, bringing efficiency and transparency to the verification process.

Although DLT can provide an efficient way to verify the driver's qualification, we still encounter three fundamental problems. First, how can we protect the driver's privacy during the verification and auditing process? The driver's qualification information may be revealed to the rider when executing the auditing contract.

Second, without relying on the third party escrow, how can we automatically check if a registered driver is still qualified ? Manually checking the driver's qualification is usually managed by a centralised third party or simply the SP; further, it is infeasible to process a large number of RHS orders per day especially when interacting with a distributed ledger.

Third, how can we motivate SPs to actively detect the disqualified drivers in the RHS? Riders are motivated by their own safety and security concerns; however, SPs in RHS ecosystem are not motivated to detect disqualified drivers, resulting in a considerable number of disqualified drivers still taking RHS orders online.

In this paper, we propose AudiSSI, a privacy-preserving automated driver's license auditing system. AudiSSI uses a dual-blockchain architecture. One of the dual-chain technologies is the decentralized privacy-preserving identity management chain; by leveraging zero-knowledge proofs, it helps to efficiently manage the registered driver's qualifications. While the other is the decentralised market monitor (DMM) which allows for disqualified drivers to be automatically detected online.

DMM consists of a list of smart contracts, among which, the core is the *Disqualified Driver Auditing Contract (DDAC)*. With *DDAC*, we introduce a novel *Safety Guarantee Insurance* for the RHS rider in the form of *DDAC* to be executed automatically to check the driver's qualification before the trip starts in a trustworthy and transparent way. In addition, to motivate the SP to actively detect the disqualified drivers, we will build an incentive mechanism into *DDAC*, so that it will automatically incentivise the affected rider while punishing the misbehaving SP for allowing disqualified drivers to take RHS orders in the platform at the same time. Meanwhile, since all results regarding detecting drivers' qualifications have been recorded in the ledger of blockchain network, AudiSSI makes the SPs accountable for every registered driver's information. Such intrinsic accountability can mitigate malicious SPs from allowing disqualified drivers effectively, and it will finally benefit the entire RHS ecosystem.

By modeling AudiSSI as a non-cooperative game, we provide the game theoretical analysis of the feasibility of using AudiSSI. Based on the estimated payoff values, we show that as long as AudiSSI is implemented, even if the consumer is not purchasing the insurance, it is not possible for the SP to misbehave by allowing disqualified driver taking RHS order in the platform. We generated driver information datasets sourced from Didi Chuxing GAIA Initiative [14]. With the datasets, we first conducted experiments for comprehensive performance evaluation via a proof-of-concept implementation on Hyperledger Indy [21] and Hyperledger Fabric [16]. The experiment results show the practically affordable performance of our design.

In summary, we have made the following contributions:

- We propose an efficient system to detect disqualified drivers which relies on a dual-chain architecture. We have also carefully designed the incentive mechanism behind the contracts to regulate the behaviors of the involving parties.
- To prevent riders from launching Sybil attacks, we introduce a new concept named *Safety Guarantee Insurance* for the RHS consumer to ensure that a rider who purchased the insurance can have their driver's qualification checked before the trip starts and get compensated according to the proposed mechanism once a disqualified driver is detected.
- We will formalise AudiSSI as a non-cooperation game and use the game theory as a tool to analyse the viability of using AudiSSI in the real world. Based on the payout constraints, we will show that the incentive mechanism of AudiSSI will lead independent, self-interested participants toward outcomes that are desirable from a system-wide perspective.
- We implemented a prototype of AudiSSI and deployed it on Hyperledger Indy and Hyperledger Fabric. We also generated RHS driver's qualification datasets sourced from Didi Chuxing [14], and extensive analysis and evaluations were conducted via a real-life example user case. The results for the same demonstrate the technical feasibility of AudiSSI.

Paper Organisation. In the rest of the paper, we have provided an overview of the background of the driver checking process in RHS, disqualified registered drivers, motivation and the preliminaries in Section 2, and then introduced the problem definition and an adversary model in Section 3. Section 4 provides a detailed explanation of the AudiSSI system design which includes the architecture, events, workflow, Insurance Purchase, Disqualified Driver Auditing and Incentive Mechanism. Next, we have provided an analysis of our proposed model in Section 5. Our experiment has been presented in the evaluation Section 6. Finally, we have surveyed the related work in Section 7 and concluded the paper in Section 8.

2 Background and Preliminaries

2.1 Background

2.1.1 How RHS Driver Checks Work

RHS companies, such as DiDi [11], Uber [32] and Lyft [12], often require applicants to provide their full name, date of birth, Social Security number, a copy of their driver's license, vehicle registration, auto insurance and proof of a completed state vehicle inspection. The RHS companies have to outsource to the external third-party security companies for the checking the drivers' information checking. External third-party security companies search national,

state and local databases for criminal and driving violations over the last seven years, and follow up with courthouse searches at the counties where the applicant lived in that period. Lyft includes an in-person interview as well, among other things, to match the person with the picture on their driver's license .

Who gets disqualified [13] Applicants will be rejected if they have convictions for any of the following offenses within the past seven years: Alcohol- or drug-related driving infractions; fraud or theft; reckless driving; hit-and-run wrecks; violent crimes; sexual offenses; acts of terror; property damage crimes; fatal accidents; resisting or evading arrest; kidnapping, embezzlement or other felonies. Applicants are also rejected if their record includes driving with a suspended, revoked or invalid driver's license or insurance.

2.1.2 Motivation

First, online users usually have to register repeated personal information with different SPs, meanwhile, those SPs hold their private information which raises privacy concerns for the users. Therefore, we are motivated to propose self-sovereign identity(SSI) based privacy-preserving solution which allows the user to self-manage their identities without sacrificing the user's privacy and provides an efficient way for the SPs to check its registered drivers' qualifications.

Moreover, current RHS companies allow a potential driver to register as a qualified driver after a driver's check. However, this check does not often take place again after the driver gets successfully registered since the SPs are not willing to put in so much effort on the time-consuming and costly process. Therefore, it results in a large number of registered drivers who are no longer qualified (e.g., their driver's license is revoked) continuing to take orders on the RHS platform, which will not only lead to serious security problems for the RHS riders and the society but also harm the SP's reputation.

Therefore, in this paper, we are motivated to foster security in RHS settings which not only enables safety and transparency on the user level but also increases the health of the online RHS market ecosystem in general.

2.1.3 Example Use Case

Didi Chuxing [11] and Uber [32], giant technology companies, act as the SP in the RHS, and serve millions of consumers per day to set up rides by their smartphones. Other entities in the ride-hailing service ecosystem include drivers, riders, and governing bodies. To register as a qualified driver, the registering driver has to fulfill the prerequisites including age requirement, identification verification, driver license verification, vehicle registration verification and even police check.

Usually, if a hypothetical user-named Tom for now-wants to register as a driver in Didi Chuxing, he has to upload the aforementioned documents required onto the platform and wait for them to be verified. This verification process is carried out by Didi Chuxing. It could be a short process if the company can simply verify them by itself, and it also could be a long process if each document is to be verified in the corresponding government agency, for example, the vehicle registration verification should be done by the Department of Motor Vehicles. No matter how Didi Chuxing verifies the driver's identity information, the driver's private data is revealed to Didi Chuxing. After some time, Tom wants to register as a driver in Uber, and he has to repeat a similar process as he has did in Didi Chuxing; meanwhile, similarly all his private information will be revealed to Uber as well. Moreover, Tom will start to worry about the risks involved—whether his private information could be reused or manipulated by third parties for their profit or even be leaked to malicious parties.

Suppose that after Tom became a qualified driver in Didi Chuxing or Uber, he committed a crime one day; then, he should ideally be de-registered by Didi Chuxing and Uber as he is not a qualified driver anymore because he would not pass police check in this situation. However, currently, neither SP would update Tom's police check information until Tom updates his information via the platform. Although the SPs can perform the re-verification process towards the drivers' qualification regularly at some time interval, it is quite costly for the SPs to do so. Furthermore, the SPs and the market governing authorities are concerned about the security risks that the online disqualified drivers will bring in.

Currently, if the SPs decide to use AudiSSI to process the driver's identity information verification, for a user who wants to register as a driver, he can manage his own personal information and selectively disclose the same to the SPs. Moreover, for the SPs, they can quickly verify the registration identity information with AudiSSI. Additionally, the market governing authorities who are required to keep a regulated and healthy RHS market can incentivise the RHS rider to detect disqualified drivers online while giving deterrence to enforce the SP to regularly validate their registered drivers' qualifications.

2.2 Preliminaries

2.2.1 Distributed Ledger Technologies (DLT)

Distributed ledger (blockchain) technologies (DLT) are particularly useful in business and governance situations that involve multiple parties which do not necessarily trust one another to negotiate and execute electronic business transactions [23]. In some cases, such transactions require the ability to establish and validate identities and identity attributes, or to check whether or not they have been revoked [23].

Another crucial role of DLTs is to register events by which attestations are revoked. It is easy to envisage the benefits of knowing whether or not a person is still an employee of a company, if a passport or driving licence has been revoked, or a certificate is still valid. The distributed nature of DLTs allow parties to query a single (but redundant) endpoint for revocations of attestations by all parties, rather than having to query a specific endpoint for individual parties [23].

2.2.2 Smart Contract

In the 90s, "Smart contract" was first introduced by N. Szabo [28], originating from the idea that a technological legal framework can help commerce, and reduce disputes

and costs. A smart contract allows users to define and execute contracts on the blockchain [27], and maintains its own data storage and balance, access to both of which is entirely governed by its code (though all contract data and balances can be publicly read on the blockchain). The program code captures the contractual logic clauses between multiple parties and pre-defines the trigger conditions and response actions. The execution of the functions in a smart contract is triggered by times or events, e.g., transactions added to the blockchain. Contracts allow for the creation of autonomous agents whose behaviour is entirely dependent on their code and the transactions sent to them, thus providing functionality comparable to that of a centralised party in a transparent, decentralised manner. With the help of smart contracts, the rules of financial transactions can be enforced without trusted third-parties.

2.2.3 Zero Knowledge Proofs

A zero-knowledge proof [17], [38] is a cryptographic technique where one party can prove to another party that they know a particular value without disclosing the actual value. For example, it can be proved that an accredited university has granted a degree to you without revealing your identity or any other personally identifiable information contained in the degree [38].

The key capabilities introduced by zero-knowledge proof mechanisms are the ability of a holder to:

- Combine multiple verifiable credentials from multiple issuers into a single verifiable presentation without revealing verifiable credentials or subject identifiers to the verifier [38]. This makes it more difficult for the verifier to collude with any of the issuers regarding the issued verifiable credentials.
- Selectively disclose the claims in a verifiable credential to a verifier without requiring the issuance of multiple atomic verifiable credentials [38]. This allows a holder to provide a verifier with precisely the information they need and nothing more.
- Produce a derived verifiable credential that is formatted according to the verifier's data schema instead of the issuer's, without needing to involve the issuer after verifiable credential issuance [38]. This provides great flexibility for holders to use their issued verifiable credentials.

3 Problem Definition and Threat Model

From a macro point of view, the goal of this paper is to regulate the RHS industry by preventing disqualified drivers from taking RHS orders, thus bringing security and safety to the RHS rider and the society. From the SPs' perspective, they must be responsible for guaranteeing that all the registered drivers are qualified at all times. Whereas from the riders' point of view, for safety concerns, they require the qualified driver for their RHS orders. Therefore, the goal of this paper is to efficiently detect disqualified drivers by automatizing the driver's qualification process and incentivising RHS riders to contribute to the detection of disqualified drivers, thus guaranteeing RHS riders' safety and security while achieving a trusted RHS market. In order to achieve this goal, we must design a system which can 1)efficiently process driver's qualifications' check regarding potential disqualified drivers automatically; 2)execute incentives and punishments automatically once disqualified driver was detected;3)act as an incentive mechanism to reward active riders who contribute to the drivers' check and punish misbehaving SPs and de-register detected disqualified drivers.

3.1 Threat Model

The adversary's goal is to allow disqualified drivers to take RHS orders. The adversary can compromise a driver's privacy as well, including a driver's qualification information. We assume that it is the SPs' responsibility to ensure that their registered drivers are qualified, meanwhile, it is also the SPs' misbehaviour for allowing disqualified drivers to take orders in the platform. In addition, we assume that the qualification status is updated seamlessly to AudiSSI. Further, we assume that the security of the standard cryptographic primitives used in AudiSSI cannot be broken, such as finding hashing collisions, forging digital credentials or compromising the private keys of arbitrary entities. We further assume that the communication between different parties is secure, and that the message cannot be compromised.

3.2 Design Requirement

Given the threat model, our design requirements are:

R1 Privacy-Preserving: AudiSSI must guarantee the confidentiality of the driver's qualification information so that the information is not revealed to the external parties.

R2 Verifiable: AudiSSI must enable the entities in the ecosystem to verify the driver's qualifications.

R3 Trustworthiness: the driver's check process should be executed in a trustworthy way.

R4 Automation: the driver's check process should be executed automatically before the trip starts, and the payoffs should be sent automatically once the disqualified driver is detected.

R5 Deterrence: When a disqualified driver is detected, the system must directly de-register the driver in the platform and punish the defaulting SP.

3.3 Design Challenges

We encounter the following technical challenges in the design of AudiSSI:

Challenge 1: Sybil Attack by the RHS Rider. Malicious riders may abuse their rights in checking all the drivers associated with all of their RHS orders which will increase the number of drivers' qualification checking operations, thus prolonging the latency of the system and making the system less efficient or even crash.

Challenge 2: Driver's Privacy Leakage. Driver's qualification information may be revealed to external parties when the rider validates the driver's qualification.

Challenge 3: Spoofing Attack by SP. Malicious SPs may use known qualified driver's qualifications to replace those of the real driver's, so that disqualified driver will never be detected and the rider will never get incentives.

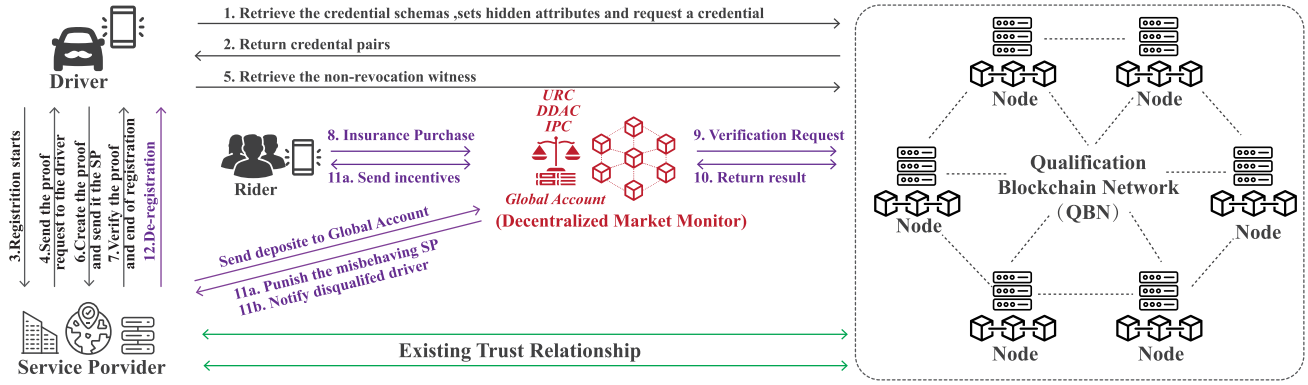


Fig. 1. Overview of AudiSSI.

4 AudiSSI System Design

In this section, we discuss the design of AudiSSI. We first introduce the architecture, then describe the workflow and events in AudiSSI, and so on.

4.1 Architecture

The AudiSSI ecosystem mainly includes five types of parties: Riders, drivers, SPs, Decentralized Market Monitor and Qualification Blockchain Network. We will explain the details based on the use case mentioned in Section 2.1.3.

- *Riders* in AudiSSI can create ride-hailing orders in the platform, and they can also purchase the safety guarantee insurance from the SP to have their driver's qualification checked before the trip starts.
- *Drivers* such as Tom, are responsible for setting the hidden attributes $A_h \subset \{1, 2, \ldots, l\}$ that they do not want to disclose to the verifier according to the predefined credential schema $\mathcal{S}$. And Tom can request a credential from the issuer by sending hidden attributes A_h in a blinded form to the issuer and agreeing on the known attributes $A_k = \{1, 2, \ldots, l\} \backslash A_h$.
- *SPs:* Like Didi Chuxing or Uber, SPs act as verifier that checks that Tom has some credentials provided by the issuer. In this use case, SPs have to verify several types of identity information of Tom including driver's license, age, police check result and so on, only all those credentials have verified, can SPs issue Tom a qualification that proves he is a qualified registered driver in the platform.
- *Decentralized Market Monitor* is made up of lists of smart contracts including *User Registration Contract (URC)*, *Insurance Purchasing Contract(IPC)* and *Disqualified Driver Auditing Contract(DDAC)*. By replacing the centralized monitor, we construct the Decentralized Market Monitor as a response to the core functionalities of the driver's qualification checking process in a trustworthy and transparent manner.
- *Qualification Blockchain Network(QBN)* plays a crucial role in AudiSSI. It consists of several trusted nodes including government authorities among others. It acts as the identity credential issuer that provides a credential to the driver, in addition to determining the credential schema $\mathcal{S}$ and publishing it on the QBN ledger $\mathcal{L}$. Moreover, QBN does include a list of qualification management protocols.

4.2 Workflow

The following steps describe the workflow of the driver registration process on the RHS platforms and the auditing process for disqualified drivers for the use case mentioned in Section 2.1.3, which is illustrated in Fig. 1:

1. Before the driver Tom registers in the SP, he has to request some corresponding credentials $\{C_{Tom}\}$ from the issuer(QBN). Therefore, he first retrieves $\mathcal{S}$ from $\mathcal{L}$, sets and sends the hidden attributes A_h in $\mathcal{S}$ to the issuer, and then agrees on the values of $A_k = \{1, 2, \ldots, l\} \backslash A_h$.
2. After QBN (the issuer) reaches a consensus and verifies the information provide by Tom, they will return a credential pair (C_p, C_{NR}) to Tom. C_p contains the requested l attributes, and C_{NR} asserts the non-revocation status of C_p. Meanwhile, QBN will publish the non-revocation status of the credential on $\mathcal{L}$.
3. After Tom obtains the required credentials from QBN, he now starts to register from the SP (the verifier).
4. The SP sends the proof request ε to Tom. The proof request contains the credential schema S_E and disclosure predicates D.
5. Tom checks that (C_p, C_{NR}) that he holds satisfies the schema S_E. He retrieves the non-revocation witness from $\mathcal{L}$.
6. Then, Tom creates a proof P that his C_{NR} satisfies the proof request ε and sends it to the SP (the verifier).
7. The SP (the verifier) verifies P via QBN (the issuer), and Tom becomes a qualified registered driver.

The rider is also required to register in the system, and the registration procedure is similar. Next, when the rider Jack creates a RHS order, he has an option to purchase the safety guarantee insurance; if he purchases the insurance, it will then execute the auditing contract for the disqualified driver automatically. In this case, we use Tom, as aforementioned in Section 2.1.3, and we assume that Tom committed a crime after he successfully became a registered driver in the RHS platform, but that he did not update his identity information with the SP because he still wanted to be a driver to make a living.

8. Jack's safety concern motivates him to find out whether Tom is a qualified driver or not. Accordingly, Jack purchases the safety insurance and requests for an audit via *DDAC*.
9. The *DDAC* triggers the verification of Tom's credentials corresponding to his qualifications in the SP.
10. The verification process will start by executing the *DDAC* automatically via the Decentralized Market Monitor, and the transaction will be written on the ledger.

    In this example, driver Tom is regarded as a disqualified driver who is still taking RHS orders online.

11a. Tom will be given either pecuniary punishment, or a ban for registering in the platform. Jack will be rewarded for detecting the disqualified drivers.

11b. The SPs will be given pecuniary punishment as well for allowing disqualified drivers in the platform and bringing safety risks to its riders.

12. Then, the SPs will execute the de-registration process for Tom.

4.3 Qualification Blockchain Network

Instead of using a centralised identity management authority, AudiSSI uses qualification blockchain network to act as the issuer in the anonymous credential architecture. In AudiSSI, QBN can use a plug-able consensus protocol, for illustration we use an RBFT consensus protocol [5] to maintain a replicated ledger because RBFT is fault-tolerate and its performance is constant when suffering attacks. The immutable QBN ledger will maintain only pseudonymous identifiers, such as the current accumulator A, the set of non-revoked indices V and the credential schema $\mathcal{S}$, but not store any private data. Moreover, the benefits of QBN are simplified qualification verification process, reduction of intermediaries, reduction of administrative costs and increase in operational efficiency.

QBN does include a list of qualification management protocols as well to support the operations in the system, which will be explained in the in-depth discussion as follows.

Identity Management Protocols. As the core part of AudiSSI, *Identity Management Protocols* consists of *Issuance Protocol*, *Verification Protocol* and *Revocation Protocol*. In the beginning, issuer in AudiSSI first generates the system parameters Table 1 and determines the credential schema $\mathcal{S}$ with l attributes $m_1, m_2, \ldots, m_l$ and the set of hidden attributes $A_h \subset \{1, 2, \ldots, l\}$ which are agreed upon by all entities. In AudiSSI, issuers use CL-signature [7] for primary credentials, and CKS accumulator and signatures [6] to track revocation status of primary credentials. The primary credential contains the requested l attributes(for example the the drivers' open and hidden attributes). The non-revocation credentials assert the non-revocation status of the primary credential. The accumulator is used to store the ids of non-revoked credentials. When issuing new credentials, the issuer will add the non-revocation id to the accumulator, and when revoking certain credentials, the issuer will remove that particular credential's id from the accumulator.

4.3.1 Issuance Protocol

As indicated in Step 1-2 shown in Fig. 1, the issuance process is between the identity owner (e.g., the driver) and the issuer (e.g., QBN). The owner $\mathcal{O}$ will set the process up first by loading the lists of credential schema $\{\mathcal{S}\}$, then setting the hidden attributes $\{m_i\}_{i \in A_h}$ and getting a nonce n_0 from the issuer.

TABLE 1
Setup Procedure

Primary Credential Setup :
1 Issuer generates primes p', q';
2 $p \leftarrow 2p' + 1$;
$q \leftarrow 2q' + 1$;
3 $n \leftarrow pq$;
4 Issuer generates random quadratic residue S modulo n;
5 Issuer generates random $x_Z, x_{R_1}, \ldots, x_{R_l} \in [2; p'q' - 1]$
6 Issuer computes: $Z \leftarrow S^{x_z} (\text{mod } n)$; $\{R_i \leftarrow S^{x_{R_i}} (\text{mod } n)\}_{1 \le i \le l}$;
7 Issuer's public key: $P_k = (n, S, Z, \{R_i\}_{1 \le i \le l})$,
private key : $s_k = (p, q)$.
Non − revocation Credential Setup :
1 Issuer chooses:
Groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ of prime order q;
Type-3 pairing operation $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$;
Generators: g for $\mathbb{G}_1$, g' for $\mathbb{G}_2$.;
2 Issuer generates:
Random $h, h_0, h_1, h_2, \widetilde{h} \in \mathbb{G}_1$;
Random $u, \widehat{h} \in \mathbb{G}_2$;
Random sk, x (mod q);
3 Issuer computes: $p^k \leftarrow g^{sk}$; $y \leftarrow \widehat{h}^x$
4 The revocation public key: $P_R = (h, h_0, h_1, h_2, \widetilde{h}, \widehat{h}, u, pk, y)$,
secret key : (x, sk).

Next, the owner prepares data for both primary and non-revocation credentials and then sends them to the issuer, the process for which is as follows:

1. Load issuer's public key $P_k = (n, S, Z, \{R_i\}_{1 \le i \le l})$ and issuer's private key $P_R = (h, h_0, h_1, h_2, \widetilde{h}, \widehat{h}, u, pk, y)$; Generate random number v', $\{\widetilde{m_i}\}_{i \in A_h}$, $\widetilde{v'}$, nonce n_1 and s'_R mod q;
2. Compute the primary credential by

$$U \leftarrow (S^{v'}) \prod_{i \in A_c} R_i^{m_i} (\text{mod } n);$$
$$\widetilde{U} \leftarrow (S^{\widetilde{v'}}) \prod_{i \in A_c} R_i^{\widetilde{m_i}} (\text{mod } n);$$
$$c \leftarrow H(U || \widetilde{U} || n_0); \widehat{v'} \leftarrow \widetilde{v'} + cv';$$
$$\{\widehat{m_i} \leftarrow \widetilde{m_i} + cm_i\}_{i \in A_h};$$
$$U_R \leftarrow h_2^{s'_R}.$$

3. Send primary credential $\{U, c, \widehat{v'}, \{\widehat{m_i}\}_{i \in A_h}, n_1\}$ and non-revocation credential U_R to the issuer.

After the data preparation, the issuer will verify the correctness of the owner's input, and then prepare for the credentials, for which the process is as follows:

1. Assign index $i < L$ to the owner, which is one of not yet taken indices for the issuer's current accumulator A; Retrieve the current set of non-revoked indices V ; Compute $m_2 \leftarrow H(i||H)$ and store information about the owner and the value i in a local database;

2. Set the values of disclosed attributes with indices from A_k;
3. Generate random number v'', s'', c mod q, random prime e and random $r < p'q'$;
4. Compute

$$Q \leftarrow \frac{Z}{US^{v''}\prod_{i\in A_k} R_i^{m_i} (\text{mod } n)};$$
$$A \leftarrow Q^{e^{-1}(\text{mod } p'q')} (\text{mod } n);$$
$$\widehat{A} \leftarrow Q^{r} (\text{mod } n);$$
$$c' \leftarrow H(Q||A||\widehat{A}||n_1);$$
$$s_e \leftarrow r - c'e^{-1} (\text{mod } p'q');$$
$$\sigma \leftarrow (h_0 h_1^{m_2} \cdot U \cdot g_i \cdot h_2^{s''})^{\frac{1}{x+c}}; \omega \leftarrow \prod_{j\in V} g'_{L+1-j+i'};$$
$$\sigma_i \leftarrow g'^{1/(sk+\gamma^i)}; u_i \leftarrow u^{\gamma^i}$$
$$A \leftarrow A \cdot g'_{L+1-i}; V \leftarrow V \cup \{i\};$$
$$wit_i \leftarrow \{\sigma_i, u_i, g_i, w, V\}$$

5. Send the primary pre-credential $(\{m_i\}_{i\in A_k}, A, e, v'', s_e, c')$ and the non-revocation pre-credential $(I_A, \sigma, c, s'', wit_i, g_i, g'_i, i)$ to the owner; Publish updated V, A on the ledger $\mathcal{L}$.

After verifying the pre-credentials, the owner then computes primary credential $C_p = (\{m_i\}_{i\in C_s}, A, e, v)$ and the non-revocation credential $C_{NR} \leftarrow (I_A, \sigma, c, s'', wit_i, g_i, g'_i, i)$, and the credential issuance process is completed.

4.3.2 Verification Protocol

The verification process is shown in the Steps 3-7 in Fig. 1. We will briefly introduce the procedure in the following paragraphs.

First, the verifier will send a proof request to the owner, which specifies the following: (1) the ordered set of d credential schemas $\{S_1, S_2, \ldots, S_d\}$, so that the owner can provide a set of d credential pairs (C_p, C_{NR}) that correspond to these schemas. (2) the unknown attributes A_h, ϕ which map A_h to $\{1, 2, \ldots, x_i\}$, the disclosed attributes A_v and the set D of predicates. X denotes the total number of attributes in these schemas.

Next, to get the non-revocation proof, the owner will prepare all the credential pairs (C_p, C_{NR}), create empty sets $\mathcal{T}$ and $\mathcal{C}$, then load issuer's public revocation key $p = (h, h_1, h_2, \widetilde{h}, \widehat{h}, u, pk, y)$, the non-revocation credential $C_{NR} \leftarrow (I_A, \sigma, c, s, wit_i, g_i, g'_i, i)$, V, and update C_{NR} by:

$$w \leftarrow w \cdot \frac{\prod_{j\in V\setminus V_{old}} g'_{L+1-j+i}}{\prod_{j\in V_{old}\setminus V} g'_{L+1-j+i}},\ V_{old} \leftarrow V;$$

V_{old} is taken from wit_i; Then compute

$$E \leftarrow h^{\rho}\widetilde{h}^{o}; D \leftarrow g^{r}\widetilde{h}^{o'};$$
$$A \leftarrow \sigma\widetilde{h}^{\rho}; G \leftarrow g_i\widetilde{h}^{r};$$
$$W \leftarrow w\widehat{h}^{r'}; S \leftarrow \sigma_i\widehat{h}^{r''};$$
$$u \leftarrow u_i\widehat{h}^{r'''};$$

and add these values to $\mathcal{C}$.

Next compute:

$$m \leftarrow \rho \cdot c; t \leftarrow o \cdot c;$$
$$m' \leftarrow r \cdot r''; t' \leftarrow o' \cdot r'';$$

Generate random:
$\widetilde{\rho}, \widetilde{o}, \widetilde{o'}, \widetilde{c}, \widetilde{m}, \widetilde{m'}, \widetilde{t}, \widetilde{t'}, \widetilde{m_2}, \widetilde{s}, \widetilde{r}, \widetilde{r'}, \widetilde{r''}, \widetilde{r'''}$ mod q.
Next compute:

$$\overline{T_1} \leftarrow h^{\widetilde{\rho}}\widetilde{h}^{\widetilde{o}};$$

$$\overline{T_2} \leftarrow E^{\widetilde{o}}h^{-\widetilde{m}}\widetilde{h}^{-\widetilde{t}};$$

$$\overline{T_3} \leftarrow e(A, \widehat{h})^{\widetilde{c}} \cdot e(\widetilde{h}, \widehat{h})^{\widetilde{r}} \cdot e(\widetilde{h}, y)^{-\widetilde{\rho}} \cdot e(\widetilde{h}, \widehat{h})^{-\widetilde{m}} \cdot e(h_1, \widehat{h})^{-\widetilde{m_2}} \cdot e(h_2, \widehat{h})^{-\widetilde{s}}$$

$$\overline{T_4} \leftarrow e(\widetilde{h}, acc)^{\widetilde{r}} \cdot e(1/g, \widehat{h})^{\widetilde{r'}};$$

$$\overline{T_5} \leftarrow g^{\widetilde{r}}\widetilde{h}^{\widetilde{o'}}$$

$$\overline{T_6} \leftarrow D^{\widetilde{r''}}g^{-\widetilde{m'}}\widetilde{h}^{-\widetilde{t'}};$$

$$\overline{T_7} \leftarrow e(pk \cdot g, \widehat{h})^{\widetilde{r''}} \cdot e(\widetilde{h}, \widehat{h})^{\widetilde{m'}} \cdot e(\widetilde{h}, S)^{\widetilde{r}}$$

$$\overline{T_8} \leftarrow e(\widetilde{h}, u)^{\widetilde{r}} \cdot e(1/g, \widehat{h})^{\widetilde{r'''}},$$

and add these values to $\mathcal{T}$.

Next, to get the validity proof, the owner will compute several security variables and then add them to $\mathcal{C}$ and $\mathcal{T}$ respectively. And the owner will compute the challenge hash by:

$$c_{\mathcal{O}} \leftarrow H(\mathcal{T}, \mathcal{C}, n_1).$$

Next, the owner will compute the *sub-proofs* for the non-revocation credential C_{NR}, the primary credential C_p, and the predicate $p : \mathcal{X}$, Pr_C, and Pr_p, respectively, and then send ($c_{\mathcal{O}}$, $\mathcal{X}$, Pr_C, Pr_p, $\mathcal{C}$) to the Verifier.

Then the verifier will proceed the non-revocation check and the validity check and get $\widehat{\mathcal{T}}$, we are not going to discuss in details as for a more comprehensive computing process, we refer the reader to the anonymous credential specification [8].

With $\widehat{\mathcal{T}}$, the verifier then computes

$$\widehat{c_{\mathcal{O}}} \leftarrow H(\widehat{\mathcal{T}}, \mathcal{C}, n_1).$$

If $c_{\mathcal{O}} = \widehat{c_{\mathcal{O}}}$, then it means the owner's proof of identity is verified, else it means fail. After this step, it means the verification process is completed.

4.3.3 Revocation Protocol

When the invalid identity information is updated on the trusted entities' end, the issuer will identify a credential to be revoked in the database and retrieve its index i, the accumulator value A, and valid index set V . Then he proceeds as follows:

1. Set $V \leftarrow V \backslash \{i\}$;
2. Compute $A \leftarrow A/g'_{L+1-i}$.
3. Publish $\{V, A\}$.

The *Revocation Protocol* will contribute to updating the invalidity of the credentials; thus the auditing mechanism is able to execute the verification in the later stage.

4.4 Decentralized Market Monitor

To better regulate the RHS market, entities can audit potential disqualified drivers via the decentralised market monitor in a trusted and transparent way. For the riders in the system, they are motivated to detect their drivers for their personal safety and security concerns. AudiSSI further offers its riders a novel type of insurance named safety guarantee insurance which the riders can purchase from their SP via *IPC*. The insuring rider will then execute the *DDAC* automatically before the trip starts. In addition, all entities must register via the *User Registration Contract* before entering the system. A detailed discussion is given in the following paragraphs.

4.4.1 User Registration

We explain by using the example use case. When Tom starts registering as a driver from the SP, different from the traditional way of identity verification where Tom has to upload his identity documents to the SP, what Tom does is only to send the lists of identity proofs required to the SP. The SP will then verify the proofs via the *Verification Protocol*, if verified, the SP will permit Tom as a qualified driver and issue him a digital driver's license in the platform, else reject Tom's request.

The registration process is also recorded as a transaction in the monitoring blockchain network. The detailed registration transaction is as follows.

- Inputs:
  - User's primary credential C_p
  - User's non-revocation credential C_{NR}
  - User's financial account address Pk_u
- Outputs:
  - Registration transaction **rt**

Different entities who wish to use this system must register their identity via *User Registration Contract*. They will generate new and unique public-private key pair (Pk_u, Sk_u) in the system. And the key pair (Pk_u, Sk_u) will be used in the underlying payouts process

$$\mathbf{rt} = (\mathrm{Timestamp}, \mathrm{TxID}, \mathrm{Type}, \mathrm{Info_u}, \mathrm{Pk_u}).$$

TxID is a unique identifier for the transaction which can be hash digest of all the other elements in the **rt** in practice. Type denotes the registering entity's type, it could be a driver, a rider or an SP. $\mathrm{Info_u}$ denotes the registering entity's description, the detailed profile like the driver's identity information, and etc.

4.4.2 Insurance Purchase

As mentioned previously, AudiSSI introduces a novel *Safety Guarantee Insurance* to mitigate *Sybil attack* launched by malicious consumers. A consumer can only have his driver audited before his trip started after purchasing the insurance. To purchase the insurance, the consumer must transfer a fee fee_{Ins} to the *Global Account*. In AudiSSI, a consumer can purchase the insurance through *IPC*, and the detailed transaction is as follows.

- Inputs:
  - Registration transaction **rt**
  - Insurance Fee fee_{Ins}
- Outputs:
  - Insurance payment transaction **ipt**

Each **ipt** includes the sufficient information which can be used for *DDAC* verifying the validity of insurance. Additionally, the insurance contains the fees incurred fee_{Ins} and the information included in the $\mathrm{InsInfo}$. A transaction **ipt** is in the following form:

$$\mathbf{ipt} = (\mathrm{Timestamp}, \mathrm{Pk_u}, \mathrm{TxID}, \mathrm{InsInfo}, fee_{Ins}).$$

4.4.3 Disqualified Driver Auditing

As mentioned in Step 8-12 of the workflow in Section 4.2, the auditing contract will be automatically executed if a rider has purchased the insurance. Note that we have made the assumption in the threat model that the peers in the QBN will always keep the identity information updated; for instance, Tom's police check record will be updated immediately once Tom commits a crime. This is feasible because the police office can be a verifier in the network. The detailed process is provided in Algorithm 1.

Algorithm 1. Disqualified Driver's Audit

Input: 1) Msg_R:Rider's message
2) Msg_{SP}:SP's message
3) Driver's **rt**
4) Rider's **rt**
5) Rider's **ipt**
6) Ledger $\mathcal{L}$

Output: Disqualified driver auditing transaction **ddat**

Procedure Disqualified Driver Auditing Process

Verify the rider's and the driver's registration validity through **rt**;
Verify the rider's insurance validity through **ipt**;
Process ID-Plate-Binding authentication of the different entities' messages:
if *verified* **then**
  $i \leftarrow$get driver's identifier from driver's **rt**;
  $C_p, C_{NR} \leftarrow$ get credential pairs from driver's rt;
  Compute: $\mathcal{T}$, $\widehat{\mathcal{T}}$;
  Generate random number: n_1;
  Compute: $c_i \leftarrow H(\mathcal{T}, \mathcal{C}, n_1)$;
  Compute: $\widehat{c_i} \leftarrow H(\widehat{\mathcal{T}}, \mathcal{C}, n_1)$;
  if $c_i = \widehat{c_i}$ **then**
    Do nothing;
  else
    ReactionPlan.trigger, generate transaction **ddat**.
  end
else
  end process.
end

When auditing a certain driver's qualifications, usually a list of items of identity information is audited; accordingly, if one identity item is detected as disqualified, this driver will be regarded as disqualified driver and he must be de-registered from the SP's database.

ID-Plate-Binding Authentication

During the auditing process for disqualified drivers, the *DDAC* will use the driver's qualification information provided by the SP. However, the malicious SPs can launch *Spoofing attack* by replacing the real driver's information by the known qualified driver's information. To prevent such attack, we propose an ID-Plate-Binding Authentication scheme to guarantee the correctness and immutability of the information used in the auditing process.

Since each driver's vehicle plate number is unique in the RHS system and it must be revealed to the rider, the ID-Plate-Binding Authentication scheme makes the vehicle's plate number in plain text, and this is bound to the driver's other identity information. Therefore, the *DDAC* will record the auditing results together with the driver's plate number in plain text so that the rider can also manually check the correctness of the auditing results.

As all the involved parties will interact with the *DDAC* in the driver auditing process, the SP will transmit Msg_{SP} and the rider will transmit Msg_R through a secure channel to the *DDAC*. Note that the driver's identity information is collected from different sources: The rider's mobile devices and the SP's application (RHS and driver's qualification auditing are two different applications; during the auditing process, we assume that the *DDAC* can not only collect information from the rider's end but also can collect from the SP's RHS application)

$$Msg_R = \{\mathrm{OrderInfo}, \mathrm{PlateNum}, R_{Sign}\}$$

$$Msg_{SP} = \{\mathrm{OrderInfo}', \mathrm{PlateNum}', C_p, C_{NR}, SP_{Sign}\}$$

$$R_{Sign} = Sign(\mathrm{OrderInfo} \parallel \mathrm{PlateNum}, Sk_R),$$

where $\mathrm{OrderInfo}$ denotes the order's information, $\mathrm{PlateNum}$ denotes the driver's plate number collected from the rider's mobile device and R_{Sign} denotes the rider's signature that calculated with rider's secret key Sk_R.

Also, the SP will send the order's information $\mathrm{OrderInfo}'$, driver's plate number $\mathrm{PlateNum}'$ and SP_{Sign} which is the SP's signature signed over $\mathrm{OrderInfo}'$ concatenate to $\mathrm{PlateNum}'$ by the SP's secret key SP_{sk}.

$$SP_{Sign} = Sign(\mathrm{OrderInfo}' \parallel \mathrm{PlateNum}', Sk_{SP}).$$

When the *DDAC* receives messages from the rider and the SP, it first compares the plate number value

$$\mathrm{PlateNum} \stackrel{?}{=} \mathrm{PlateNum}',$$

and if the result is equal, it computes whether $\mathrm{PlateNum}$ is included in A_k which is extracted from driver's primary credential C_p. If yes, it implies that the driver's qualification information is correct and tamper-proof, and then the credential pairs C_p and C_{NR} will be used in the later stage of the auditing process.

4.4.4 Reaction Plan

The reaction plan stands for the insurance claim, it is the financial reaction transaction defined in the *Safety Guarantee Insurance*. The plan includes the payments that occur in response to detect a disqualified driver. As shown in Algorithm 4.4.4, the *DDAC* contains the reaction plan to execute the financial transactions when a disqualified driver is detected.

Algorithm 2. Reaction Plan

Input: 1)$\mathrm{OrderInfo}$;
2) Pre-defined Incentive Mechanism;
3)$\mathrm{Acc_F}$;
Output: Payouts Distribution
Procedure Process of Reaction
1 $\mathrm{Pk_R} \leftarrow$ retrieve rider's account address from $\mathrm{OrderInfo}$;
2 $\mathrm{Pk_{SP}} \leftarrow$ retrieve SP's account address from $\mathrm{OrderInfo}$;
3 $i \leftarrow$ retrieve driver's identity information from $\mathrm{OrderInfo}$;
4 Send $fDet$ from $\mathrm{Acc_F}$ to $\mathrm{Pk_R}$;
5 Send f_{Pun} from $\mathrm{Pk_{SP}}$ to $\mathrm{Acc_F}$;
6 De-register driver i from the platform.
7 **end Procedure**

4.5 Payout Settings and Incentive Mechanism

In this section, we will explain the incentive mechanism that we proposed to incentivise detectors and to punish misbehaving SPs for allowing disqualified drivers taking RHS orders. First, we provide a framework for payout reaction programs, which specifies a set of financial payments to be executed when detecting a disqualified driver.

To guarantee the *Global Fund Account* has enough balance for the pre-defined payouts, when SP first registered in the system, a security deposit $Deposit$ is required and should be transferred to the global fund.

Detector Payouts. The detector payout (f_{Det}) is paid to the rider who purchases the insurance and successfully detects the disqualified driver. The payout compensates the rider for the safety and security risks they incur by having a disqualified driver taking their RHS order and incentivises them for detecting the disqualified driver. At the same time, the punishment fee f_{Pun} will be applied to the misbehaved SP, and we have

$$f_{Det} << f_{Pun}, \tag{1}$$

where f_{Pun} is a fixed large sum of money.

For a reaction plan, AudiSSI presets the payouts f_{Det}, $Deposit$,f_{Pun} and the insurance price f_{Ins}. AudiSSI sets the constraint on the amounts that must hold

$$f_{Ins} < f_{Det} < f_{Pun} < Deposit. \tag{2}$$

5 Analysis

We analyse AusiSSI according to the design requirements.

5.1 Security Analysis

Security Against Sybil Attacks. AudiSSI provides disqualified driver detection in RHS. However, *Sybil attacks* can be launched by malicious riders to increase the blockchain network's computation costs, it will making the entire system inefficient not only when auditing potential disqualified drivers. Through the proposed insurance scheme, AudiSSI actually increases the costs for an audit. Only riders who has purchased insurance can be able to have their drivers audited. Therefore, AudiSSI can mitigate *Sybil attacks*.

Security Against the Malicious SP. AudiSSI adds a new entity named decentralised market monitor(DMM) in the system. Thanks to the decentralised nature, it makes sure that no centralised authority controls the driver auditing process. However, a malicious SP could launch *Spoofing attack* by submitting known qualified driver's information to the *DDAC* during the auditing process. In AudiSSI, we have designed the ID-Plate-Binding authentication scheme which allows *DDAC* to validate the driver's information from the SP's end, as well as the rider's end. The scheme first verifies whether the driver's plate number value $\mathrm{PlateNum}$ from the rider's message matches $\mathrm{PlateNum}'$ from the SP's. If verified, then it verifies whether this $\mathrm{PlateNum}$ is included in the known attributes A_k extracted from the primary credential C_p. AudiSSI will not allow manipulated information to proceed to the next step in the auditing process, it therefore guarantees that the malicious SPs do not succeed.

Security Against Privacy Leakage Attacks. AudiSSI provides resistance against privacy leakage, which can be launched by any entity in the system. AudiSSI leverages zero-knowledge proof technique; the identity information proof will selectively disclose the user's private information in the system. The entity can choose which attributes to hide and append to A_h while only attributes in A_k are revealed to the public. Therefore, the system will not reveal any hidden information to other users even during the auditing process.

5.2 Feasibility Analysis

With the increasing sophistication of the attacks and the complexity of the systems, the protection using traditional methods could be cost-prohibitive[43]. To better understand security from a strategic and decision-making perspective, game theory provides a natural framework to capture the adversarial and defensive interactions between an attacker and a defender. It provides a quantitative assessment of situations of conflict and cooperation, it is appropriate to analyse the acceptance of AudiSSI.

Since the objective of each player in the game is to maximise its own profit, it therefore models the interactions between SPs and consumers in the form of non-cooperative game. The fundamental assumption in the game theory is that all players are rational and try to optimize their individual payoffs. Also, each player knows that other players are using payoffs maximizing decisions. Furthermore, as adopted often in practice, the driver auditing used in this paper is pay-per-use. For easy demonstration and presentation, we summarised the core notations in Table 2. While the fundamental elements in the game are discussed as follows:

TABLE 2
Notation With Nomenclature

Field	Use
$s, \boldsymbol{S}$	SPs;
$r, \boldsymbol{R}$	riders;
f_{Ins}	The cost for purchasing a *Safety Guarantee Insurance*;
f_{Pun}	The penalty paid by an s when detected behaving abnormally;
f_{Det}	The incentives given to the rider s for detecting disqualified driver;
f_{Risk}	losses per order for a rider taking safety risks from riding by a disqualified driver;
$Cost$	cost that the SP have to spend for every driver's checking;
β	$0 \leqslant \beta \leqslant 1$, the probability that r chooses No Insurance strategy, or the share of orders r buys *Safety Guarantee Insurance* within AudiSSI;
θ	$0 \leqslant \theta \leqslant 1$, the probability that s chooses Not Strictly Checking Driver's Qualifications strategy, or the share of orders s uses Not Strictly Checking Driver's Qualifications.

5.2.1 Player

In our RHS system $\mathbb{R} = \{\boldsymbol{S}, \boldsymbol{R}, \boldsymbol{D}, \boldsymbol{M}\}$ with AudiSSI, where $\boldsymbol{S}, \boldsymbol{R}, \boldsymbol{D}$, and $\boldsymbol{M}$ denote the sets of SPs, riders, drivers and market authorities, respectively. Since the authorities' responsibility is monitoring and regulating the market while are not looking for profits, only SPs and riders are considered in our game settings. Also, we regard that the driver defaulting is the SP's misbehaviour.

5.2.2 Strategy

SPs. The SPs have two different strategies to play: Strictly verifying their registered drivers regularly or not strictly verifying their registered drivers. SPs can choose to behave correctly by checking their registered drivers regularly to prevent disqualified drivers from taking RHS orders, or choose to not check their drivers. Some SPs choose the latter option to save huge verification costs; however, if they are caught misbehaving, they would lose a large amount of penalty.

Riders. Since the riders in RHS have safety and security concerns about disqualified drivers taking their orders, it is natural to assume that all the riders want to avoid disqualified drivers. The riders are provided with the option to purchase a *Safety Guarantee Insurance* within AudiSSI. If a rider purchases insurance, the disqualified driver will be surely detected if he is taking this rider's RHS order which means that it is the SP's misbehaviour for allowing disqualified drivers to enter the market. Nevertheless, if a rider does not purchase the insurance, even if disqualified drivers are taking orders, the rider has no efficient way to detect the same and has to take safety risks to use the RHS.

5.2.3 Payoff

Payoff of SPs. In general, if the SP chooses to check its registered drivers' qualifications frequently, we assume that the SP has to spend $Cost$ for every driver's checking. If the rider chooses No Insurance, the payoff of SP s equals the cost incurred in checking the driver's qualifications.

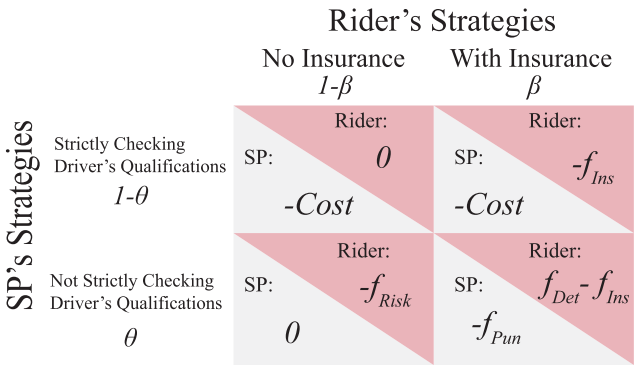


Fig. 2. The payoff bi-matrix in AudiSSI.

When SP does not strictly check its registered drivers' qualifications, once a disqualified driver is detected, the corresponding SP will subsequently be punished by f_{Pun} amount of money per order.

Payoff of Riders. A rider $r \in \boldsymbol{R}$ uses RHS at SP $s \in \boldsymbol{S}$. When a rider does not purchase the *Safety Guarantee Insurance* within AudiSSI, if the SP is not strictly checking its drivers' qualifications, the rider will take safety risks, and we assume this risk will lead to a f_{Risk} loss to the rider for each order.

When r purchases the insurance, and if s is not strictly checking its driver's qualifications and the disqualified driver will be detected, r will be reward by f_{Det} amount of money for each order.

5.2.4 Which Strategy to Choose

The SPs are always pursuing profit maximization under RHS context, while the riders are looking for safety and security in their ride. Fig. 2 shows the game bi-matrix, which demonstrates the possible strategies for both players. Players decide which strategy to choose based on analysing the information in this bi-matrix.

In plain terms, in a pure Nash equilibrium, no player would have anything to gain by changing his/her strategy, given the strategy of the other players unchanged. Considering the payoff bi-matrix shown in Fig. 2, in a single-stage game, there does not exist such a stable combination of strategies in which for both players have no motivation to change. This game does not have a pure strategy Nash Equilibrium.[1] However, if at least one player chooses a randomised strategy, the game will have a mixed strategy Nash equilibrium. Similar to a pure Nash equilibrium, in a mixed strategy Nash equilibrium, both players would not benefit from changing their (randomised) strategy [44].

An SP may not always fail to check its drivers, and we assume that the SP may choose not to regularly check θ proportion of its registered drivers; thus, the proportion of frequently checked driver's is $1-\theta$. Furthermore, a rider may decide whether to buy a *Safety Guarantee Insurance* within AudiSSI. We assume that the rider buys this insurance for β proportion of the orders, and thus the proportion of orders without insurance is $1-\beta$.

Theorem 1. *When SP and rider make payoff-maximizing choices and they both know their opponent is rational, $f_{Pun} \gg Cost$ (the penalty charges for an SP when detected behaving abnormally is much greater than the cost per order of checking driver's qualifications), then in the equilibrium, the incentive of rider to buy the insurance will be exceedingly low.*

Proof. In the definition of a mixed strategy Nash equilibrium, given the strategy of rider, the SP is indifferent between playing Strictly Checking Driver's Qualifications and Not Strictly Checking Driver's Qualifications. This means that the SP's expected payoff of choosing Strictly Checking Driver's Qualifications must be equal to that of playing Not Strictly Checking Driver's Qualifications. We have

$$(1-\beta)\times(-Cost)+\beta\times(-Cost)=(1-\beta)\times 0+\beta\times(-f_{Pun}).$$

We obtain

$$\beta=\frac{Cost}{f_{Pun}}.$$

When $f_{Pun} \gg Cost$ (in practice, SPs always have to pay an extremely large fine when they are detected behaving abnormally, which is much greater than the cost per order of checking driver's qualifications), $\beta \approx 0$. This implies that in the long-run equilibrium, the incentive of rider to buy the insurance would be approximately zero; thus, the theorem is proved. □

Theorem 2. *When SP and rider make payoff-maximizing choices and they both know their opponent is rational, $f_{Det} \gg f_{Ins}$ (the incentives to the affected rider for detecting a disqualified driver is far larger than the the cost of purchasing insurance; then in the equilibrium, the incentive of SP to choose Not Strictly Checking Driver's Qualifications will be exceedingly low.*

Proof. In the definition of a mixed strategy Nash equilibrium, given SP's strategy, the rider is indifferent between choosing No Insurance and With Insurance. The rider's expected payoff when choosing No Insurance must be equal to which of choosing With Insurance. We have

$$(1-\theta)\times 0+\theta\times(-f_{Risk})=(1-\theta)\times(-f_{Ins})+\theta\times(f_{Det}-f_{Ins}).$$

We obtain

$$\theta=\frac{f_{Ins}}{f_{Det}+f_{Risk}}.$$

When $f_{Det} \gg f_{Ins}$ (in reality, the incentives given to the affected rider for detecting disqualified driver is far larger than the cost of purchasing insurance), $\theta \approx 0$. This implies that in the long-run equilibrium, the incentive of SP to apply Not Strictly Checking Driver's Qualifications would be exceedingly low, and the proof of the theorem is done. □

In conclusion, from a long-run point of view, although a rider has a particularly low incentive to buy the insurance, the

1. The payoff matrix in Fig. 2 shows that, if the rider chooses No insurance, SP will choose Not Strictly Checking Driver's Qualifications. When SP chooses Not Strictly Checking Driver's Qualifications, the user will prefer With Insurance. When the user chooses With Insurance, then the SP will choose Strictly Checking Driver's Qualifications. If the SP chooses Strictly Checking Driver's Qualifications, the user chooses No Insurance. Therefore, in this case, there does not exist a pure Nash Equilibrium.

TABLE 3
Time Consumption of Each Phase

Phase	Cost(s)	Accumulated Cost(s)
Setting Trust Nodes	14.56	-
Setup Credential Schemas	2.02	-
Setting Credential Schemas	11.69	28.27
-	-	-
ID Credential Issuance	0.95	-
ID Credential Verification	1.50	2.45
-	-	-
Vehicle Rego Issuance	2.07	-
Vehicle Rego Verification	1.95	4.02

introduction and application of AudiSSI in RHS could prevent against SPs' misbehaviour of not strictly checking their registered driver's qualifications. Meanwhile, the mixed strategy equilibrium (as well as the expected payoff for both side in this equilibrium) is close to that of the social welfare theoretical optimal situation (namely, SP chooses Strictly Checking Driver's Qualifications and rider chooses No Insurance).

6 Experiments and Evaluations

6.1 Experiment 1: Identity Management Procedures

We instantiated AudiSSI prototype by using Hyperledger Indy [21] and Fabric [16] frameworks. Our first experiment was the identity management process. The basic set-up for the experiment is as follows,

- the nodes run in 3.20GHz 6-vCPU Ubuntu VM with 8GB of RAM and HDDs as local disks;
- the nodes run on Indy instrumented performance evaluation through local logging;
- the nodes run in the docker container.
- four nodes and five clients runs in the *indy_pool*, the *indy_pool* depends on *indy_plenum* version 1.9.2 dev871 and *indy_node* version 1.9.2 dev1061.
- the nodes run in a modified RBFT consensus protocol to agree on the order of transactions.

We took the ride-hailing use case as an example to experiment on the time consumption of the different phases in the identity issuance and verification process. We used two different credential schemas in the experiment: The identification information and the vehicle registration information.

6.1.1 Evaluation on Experiment 1

As indicated in Table 3, AudiSSI spends 28.02 seconds during the set-up phases; this is because the phases are only executed once, and it will do the fundamental set-ups for the system such as setting trust nodes, credential schemas and so on. Next, for the two example use cases, as the identification credential schema is simpler than the vehicle registration credential, the total time consumption for the identification credential issuance and verification is 2.45 seconds which is much quicker than 4.02 seconds taken for the vehicle registration credential. In summary, although AudiSSI does take a relatively long time during the set-up phases, it is required to be carried out only once in a lifetime. Further, the duration of the credential issuance and verification phases is in second-level. Therefore, the total time consumption for the identity management process is viable in real life.

6.2 Experiment 2: Auditing Process

In this experiment, we tested the performances of the auditing process. The experiment setup is as follows,

- The version of Hyperledger Fabric is v1.1.0. We instantiated two organisations(orgs) which were embedded on two different physical node, with each org containing one single peer. We also included an extra physical node for the orderer and Certificate Authorities(CAs).
- The system environment of each node is Intel(R) Core^{TM} i5-4460 3.20GHz 4-vCPU VMs running Ubuntu 18.06 with 8GB of RAM, the local disk is 1TB HDD. All nodes are interconnected with 100Mbps networking.
- We used Hyperledger Caliper [10] to measure the performance. (Caliper is a benchmark framework to measurethe performance of multiple blockchain solutions; currently supported performance indicators include throughput and latency.)

During the experiment, the baseline experiment was that these two orgs, the orderer and CAs are all embedded on the same physical node. The first phase of the test was to invoke transactions for three rounds; in each round we submitted 500 transactions, and the send rate increased from 100 transaction per second(tps) to 300 tps. The second phase of the test was to query from the nodes for three rounds; we submitted 10000 transactions per round, and the send rate increased from 500 tps to 1500 tps. In the distributed nodes settings, the two orgs were embedded on different physical nodes, and both orderer and CAs were embedded on another physical node. The experiment procedure was exactly the same as the baseline experiment.

6.2.1 Evaluation on the Experiment 2

According to the results, the throughput of invoking was found to increase as the send rate increased, which is the same as the throughput of querying. When all the logical nodes are settled on the same physical node, the throughput of either querying and invoking is lower than which are settled on the distributed nodes.

As the send rate increases, the throughput increases, but once the send rate reaches its threshold, the throughput will not increase anymore. The threshold is limited by the hardware limitation of the running node.

When the number of physical nodes increases, the threshold increases as well, implying that the performance will increase as well. As is shown in Fig. 3a, when the send rate is 1500 tps, the baseline setting is 1028.3 tps which is far below 1500 tps, which means that it has reached the threshold. A similar pattern can be found in Fig. 3b as well.

As shown in Figs. 4a and 4b, when the throughput approaches the threshold, the latency of each transaction shows gradual improvement.

As the number of physical nodes increases, the changes of latency decreases slowly as the throughput approaches the threshold. Fig. 4a demonstrates that when the send rate increases to 1500 tps, the query throughput can no longer match the send rate. For the baseline settings, the latency

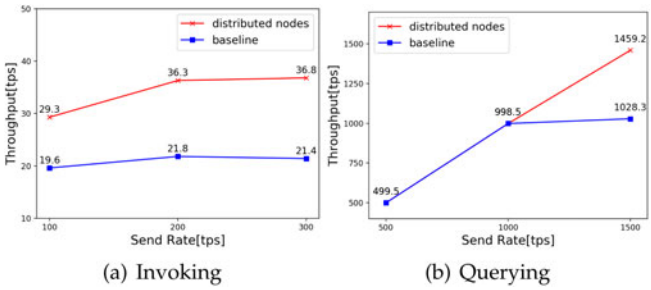


Fig. 3. Throughput of different executions.

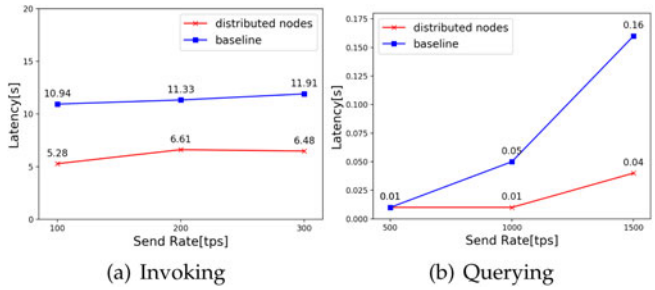


Fig. 4. Latency of different executions.

significantly increases from 0.01s to 0.16s; meanwhile, in a distributed setting, the latency increases from 0.01s to 0.04s.

7 Related Work

7.1 Non-Blockchain Based Self-Sovereign Identity Solutions

Some previous approaches have also sought to solve the identity management problem through other sources. For example, Personal Data Storage (PDS) [4], [35], [37] are environments where the user has full control over the access of other parties. However, PDS does not propose standard schema to store the information in, which makes it difficult for the user to verify the information. Solutions based on attribute-based credential schemes such as IRMA [20], [35] enable users to receive digitally signed attributes from trusted issuers such as the government.

Additionally, IRMA meets the privacy requirement by allowing its user to disclose its information selectively. However, as the private credentials are stored on the user's phone, losing the phone implies losing the identities as well. Therefore, both of the above approaches have not addressed the problem of the cumbersome, lengthy identity verification process.

7.2 Blockchain Based Self-Sovereign Identity Solutions

As blockchain essentially provides a decentralised domain which is not controlled by any single entity, it coincides with some desirable properties of a self-sovereign identity. Many researchers have already exploited this technique to develop self-sovereign identity applications on Blockchain-based platforms[18]. Built on top of the Ethereum [36] platform, uPort [34] is a decentralised identity system which supports the notion of self-sovereign identity [35]. By using smart contracts and Interplanetary File System (IPFS)[22], uPort allows its user to use a mobile application to create, update and share identity information with other users, and to store the corresponding private key on IPFS. Further, uPort also uses the public registry to create a correlation between a uPort identity and its corresponding IPFS data. However, uPort currently only allows creating one identity. Therefore, uPort is not as flexible as our proposed system.

Moreover, as the private key is stored on the smartphone, losing such a device would result in losing the identity, which would also increase the security risks.

Another identity solution, Sora [30], is based upon JSON-LD standardised [24] key-value pairs. Its attribute-based storage enables selective disclosure of the identity information. Although it is easy to use, the identity as the private key is not solely stored on the mobile device but created from a master password which is stored on an accessible centralised server; therefore, it brings some security concerns if the centralised server is compromised.

Different from uPort and Sora, Sovrin [29] network is a public permissioned distributed ledger. Sovrin enables its users to manage its identity themselves while providing opportunities to selectively disclose their identity information by using zero-knowledge proofs. However, they do not seem to provide or require any (verifiable) guarantees concerning the correct functioning of agents in the network [35]. Therefore, Sorvin is not provable.

Although some previous approaches could address particular problems in the identity management area, none of the approaches has solutions towards the time-sensitivity of the identity information. In this regard, AudiSSI does not only allow the user to control their identity in privacy-preserving bases, but also introduces a mechanism towards the invalid time-sensitive identity information.

7.3 Insurance Schemes

Insurance and insurance related schemes are widely used in all walks of life, both in traditional and modern context. Traditional applications such as life insurance and car insurance are everywhere in one's life. Recently, applications such as flight delay insurance, delivery food delay insurance has entered our lives. For example, a food delivery service provider Eleme[40] enables its customers to purchase a delay insurance while placing an order and the consumer will then get compensated if the delivery is delayed. Other types of insurance include electronic insurance policies for service evaluation in distributed systems[41] and authentication metric that follow good design principles [42]. However, both of these proposals offer little accountability and automation. Due to the complexity of the RHS systems' pricing algorithms, the aforementioned insurance scheme can not be directly applied to the price auditing as these methods do not allow auditing publicly.

AudiSSI borrows the idea from Spas [33] which is the first work to introduce such insurance idea into price policies as a means of balancing SP control capability and liability. Spas also utilised smart contract technology to add value and trust to public auditing and automatic execution of transactions. However, the main difference is that the motivation of AudiSSI is raised by SP not carefully checking their registered drivers, cost consumption and privacy concerns in registration process. As we know that current RHS companies allow a potential driver to register as a qualified

driver after a driver's check. However, this check does not often take place again after the driver gets successfully registered since the SPs are not willing to put in so much effort on the time-consuming and costly process. Therefore, it results in a large number of registered drivers who are no longer qualified (e.g., their driver's license is revoked) continuing to take orders on the RHS platform, which will not only lead to serious security problems for the RHS riders and the society but also harm the SP's reputation. Therefore, in this paper, we are motivated to foster security in RHS settings which not only enables safety and transparency on the user level but also increases the health of the online RHS market ecosystem in general.

While the motivation of Spas [33] is to address the price discrimination raised by the SP. They are totally different in this arena.

8 Conclusion

In this paper, we presented an auditable self-sovereign identity system (named AudiSSI), which allows the user to self-manage their identities without sacrificing the user's privacy and provides an efficient way for the SPs to check its registered drivers' qualifications. Further, we proposed a safety guarantee insurance in the form of driver's qualification auditing contract to enable the RHS rider to check his driver's qualifications before the trip start and get incentives once a disqualified driver is detected. We described the workflow of how a driver registers as a qualified driver in the SP and how an insured rider automates the audit of a registered driver. We designed an incentive mechanism and provided a game-theoretical analysis. Finally, we implemented a prototype of AudiSSI and deployed it on Hyperledger Indy and Fabric to show that self-sovereign identity system for RHS driver with qualification auditing is efficient and technically feasible. Our work does not stop all malicious entities, but the deterrence of the incentive mechanism will mitigate this problem and it will contribute to significant huge cost-saving for the corresponding SP in investigating the disqualified drivers.

Acknowledgments

This work was supported by the Blockchain Core Technology Strategic Research Program of Ministry of Education (China) under Grant 2020KJ010801.

References

[1] Globaltimes.cn, "Didi offers 1m yuan to find passenger's killer - Global times," Accessed: Sep. 22, 2020. [Online] Available: https://www.globaltimes.cn/content/1101749.shtml
[2] Techcrunch.com, "Techcrunch is now a part of verizon media," Accessed: Sep. 22, 2020. [Online] Available: https://techcrunch.com/2018/08/26/didi-suspends-carpooling-service/
[3] Uber driver accused of trying to kidnap passenger. Accessed: Mar. 22, 2021. [Online] Available: https://www.abccolumbia.com/2021/03/22/uber-driver-accused-of-trying-to-kidnap-passenger/
[4] S. ă Alboaie and D. Cosovan, "Private data system enabling self-sovereign storage managed by executable choreographies," in *Proc. Distrib. Appl. Interoperable Syst.*, 2017, pp. 83–98.
[5] P. Aublin, S. B. Mokhtar, and V. Qu éma. "RBFT: Redundant byzantine fault tolerance," in *Proc. IEEE 33rd Int. Conf. Distrib. Comput. Syst.*, 2013, 297–306.
[6] J. Camenisch, M. Kohlweiss, and C. Soriente, "An accumulator based on bilinear maps and efficient revocation for anonymous credentials," in *Proc. Int. Workshop Public Key Cryptogr.*, 2009, pp. 481–500.
[7] J. Camenisch and A. Lysyanskaya, "A signature scheme with efficient protocols," in*Proc. Secur. Commun. Netw.*, 2003, 268–289.
[8] J. Camenisch and E. Van Herreweghen, "Design and implementation of theidemix anonymous credential system," in *Proc. 9th ACM Conf. Comput. Commun. Secur.*, 2002, 21–30. [Online]. Available: https://doi.org/10.1145/586110.586114
[9] M. Castro, B. Liskov *et al.*, "Practical byzantine fault tolerance," in *Proc. 3rd Symp. Operating Syst. Des. Implementation*, 1999, pp. 173–186.
[10] Cl, "Hyperledger caliper," Accessed: Sep. 30, 2019. [Online]. Available: https://cn.hyperledger.org/projects/caliper
[11] DiDi, "Didi Chuxing,"Accessed: Jun. 5, 2019. [Online]. Available: https://en.wikipedia.org/wiki/DiDi
[12] Lyft, "Drive with Lyft California," Accessed: Apr. 5, 2021. [Online]. Available:https://www.lyft.com/
[13] Wear, B., "How driver CHECKS work: What Uber and LYFT do, what city would require," Accessed: Apr. 30, 2021. [Online]. Available: https://www.statesman.com/news/20160904/how-driver-checks-work-what-uber-and-lyft-do-what-city-would-require
[14] DiDi, "Brain intelligent driving smart transportation AI labs." Accessed: Aug. 15, 2020. [Online]. Available: https://outreach.didichuxing.com/appEn-vue/Register?path=register
[15] P. Dunphy and F. A. P. Petitcolas, "A first look at identity management schemes on the blockchain," *IEEE Secur. Privacy*, vol. 16, no. 4, pp. 20–29, Jul./Aug. 2018.
[16] Fab, "Hyperledger fabric," Accessed: Aug. 27, 2019. [Online]. Available: https://cn.hyperledger.org/projects/fabric
[17] U. Feige, A. Fiat, and A. Shamir, " Zero-knowledge proofs of identity," *J. Cryptol.*, vol. pp. 77–94, 1988.
[18] M. S. Ferdous, F. Chowdhury, and M. O. Alassaf, "In search of self-sovereign identity leveraging blockchain technology," *IEEE Access*, vol. 7, pp. 103 059–103 079, 2019.
[19] H. Gunasinghe *et al.*, "PrivIdEx: Privacy preserving and secure exchange of digital identity assets," in *Proc. World Wide Web Conf.*, 2019, pp. 594–604. [Online]. Available: https://doi.org/10.1145/3308558.3313574
[20] B. Hampiholi and G. Alpar, "Privacy-preserving webshopping with at-tributes," in *Proc. IEEE Symp. Privacy-Aware Comput.*, 2017, pp. 25–36. [Online]. Available: https://doi.org/10.1109/PAC.2017.34
[21] Ind, "Hyperledger indy," Accessed: Aug. 17, 2019. [Online]. Available: https://cn.hyperledger.org/projects/hyperledger-indy
[22] IPFS, "InterPlanetary file system," Accessed: Jun. 25, 2019. [Online]. Available: https://en.wikipedia.org/wiki/InterPlanetary_File_System
[23] R. Joosten, "Self-Sovereign identity framework and blockchain," *ERCIMNews*, vol. 110, pp. 29–30. [Online]. Available: https://ercim-news.ercim.eu/en110/special/self-sovereign-identity-framework-and-blockchain
[24] M. Lanthaler and C. Gutl, "On using JSON-LD to create evolvable RESTful services," in *Proc. 3rd Int. Workshop RESTful Des.*, 2012, pp. 25–32.
[25] G. Maganis, E. Shi, H. Chen, and D. Song, "OPAAK: Using mobile phones to limit anonymous identities online," in *Proc. 10th Int. Conf. Mobile Syst., Appl., Serv.*, 2012, pp. 295–308. [Online]. Available: https://doi.org/10.1145/2307636.2307664
[26] S. Nakamoto *et al.*, Bitcoin: A peer-to-peer electronic cash system, *Decentralized Bus. Rev.*, pp. 21260, 2008.
[27] M. K. Reiter and S. G. Stubblebine, "Authentication metric analysis and design," *ACM Trans. Inf. Syst. Secur.*, vol. 2, no. 2, pp. 138–158, 1999. [Online]. Available: https://doi.org/10.1145/317087.317088
[28] SMT, "Smart contracts," Accessed: Jul. 25, 2019. [Online]. Available: http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.h tml
[29] SOV, "Sovrin: A protocol and token for self-sovereign identity and de-centralized trust," Accessed: Sep. 27, 2019. [Online]. Available: https://sovrin.org/library/sovrin-protocol-and-token-white-paper/2020–07-27 13:53
[30] M. Takemiya and B. Vanieiev, "Sora identity: Secure, digital identity on the blockchain," in *Proc. IEEE 42nd Annu. Comput. Softw. Appl. Conf.*, 2018, pp. 582–587. [Online]. Available: https://doi.org/10.1109/COMPSAC

[31] K. C. Toth and A. A.-Priddy, "Self-Sovereign digital identity:A paradigm shift for identity," *IEEE Secur. Privacy*, vol. 17, no. 3, pp. 17–27, May/Jun. 2019. [Online]. Available: https://doi.org/10.1109/MSEC.2018.2888782
[32] Uber, "Uber," Accessed: Jun. 10, 2019. [Online]. Available: https://www.uber.com
[33] Y. Lu, Y. Qi, S. Qi, Y. Li, H. Song, and Y. Liu, "Say no to price discrimination: Decentralized and automated incentives for price auditing in ride-hailing services," *IEEE Trans. Mobile Comput.*, early access, Jul. 9, 2020, doi: 10.1109/TMC.2020.3008315.
[34] uPort, "uPort: Tools for decentralized identity and trusted data," Accessed: May 10, 2019. [Online]. Available: https://www.uport.me/
[35] D. van Bokkem, R. Hageman, G. Koning, L. Nguyen, and N. Zarin, "Self-Sovereign identity solutions: The necessity of blockchain technology," 2019, *arXiv:1904.12816v1*.
[36] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, pp. 1–32, 2014.
[37] Z. Yan, G. Gan, and K. Riad, "BC-PDS: Protecting privacy and self-sovereignty through blockchains for openPDS," in *Proc. IEEE Symp. Serv.-Oriented Syst.Eng.* 2017, 138–144.
[38] Zero, "Zero-knowledge proofs," Accessed: Aug. 25, 2019. [Online]. Available: https://w3c.github.io/vc-data-model/CR/2019–07-25/#zero-knowledge-proofs2020-07-27 13:53
[39] Pr. Erich, "Game theory through examples. mathematical association of America," Accessed: Jan. 12, 2020. [Online]. Available: http://www.jstor.org/stable/10.4169/j.ctt6wpwgj
[40] Eleme mobile application. Accessed: Sep. 12, 2020. [Online]. Available: https://www.ele.me/home/
[41] C. Lai, G. Medvinsky, and B. C. Neuman, "Endorsements, licensing, and insurance for distributed system services," in *Proc. 2nd ACM Conf. Comput. Commun. Secur.*, 1994, pp. 170–175.
[42] M. K. Reiter and S. G. Stubblebine, " Authentication metric analysis and design," *ACM Trans. Inf. Syst. Secur.*, vol. 2, no. 2, pp. 138–158, May 1999.
[43] Q. Zhu and S. Rass, " Game theory meets network security: A tutorial," in *Proc. ACM SIGSAC Conf. Comput. Commu. Secur.*, 2018, pp. 2163–2165.
[44] J. Nash, "Non-Cooperative games," *Annals Math.*, vol. 54, no. 2, pp. 286–295, 1951.

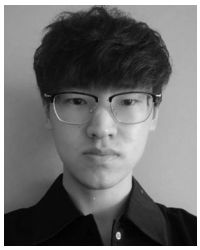
Youshui Lu (Member, IEEE) received the BS degree from The Australian National University, Australia, in 2013 and the MS degree from the University of Sydney, Australia, in 2015. He is currently working toward the PhD degree with the School of Computer Science and Technology, Xi'an Jiaotong University, China. His research interests include blockchain technology, distributed systems, trusted urban computing, and computational social systems.

Jingning Zhang received the BS degree from Shanghai Business School in 2018. He is currently working toward the MS degree with the Faculty of Electronic and Information Engineering, Xi'an Jiaotong University. His research interests include blockchain and applied cryptography.

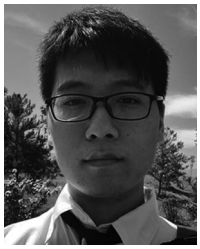
Yong Qi (Member, IEEE) received the PhD degree from Xi'an Jiaotong University, China. He is currently a professor with Xi'an Jiaotong University. His research interests include operating systems, distributed systems, and cloud computing.

Saiyu Qi (Member, IEEE) received the BS degree in computer science and technology from Xi'an Jiaotong University, Xi'an, China, in 2008 and the PhD degree in computer science and engineering from The Hong Kong University of Science and Technology, Hong Kong, in 2014. He is currently an associate professor with the School of Computer Science, Xi'an Jiaotong University, China. His research interests include applied cryptography, cloud security, distributed systems, and pervasive computing.

Yue Li received the BS and MS degrees in econometrics from the University of Groningen, the Netherlands, in 2010 and 2012, respectively. He is currently working toward the PhD degree in economics with the School of Business and Economics, Vrije Universiteit Amsterdam, the Netherlands. He is currently a quantitative risk analyst with the ABN AMRO bank, the Netherlands. His research interests include applied economics, economics models, financial risk models, and macroprudential policies.



Hongyu Song received the Bachelor of Engineering degree in the Internet of Things engineering from Xi'an Jiaotong University, China, in 2019. He is currently working toward the master's degree with the School of Computer Science and Technology, Xi'an Jiaotong University. His research interests include cyber security, Internet of Things, and blockchain.



Yuhao Liu received the BS degree in software engineering from Northwestern Polytechnic University, China, in 2018. He is currently working toward the master's degree with the faculty of Electronic and Information Engineering, Xi'an Jiaotong University. His research interests include cyber security, Internet of Things, and blockchain.