Blockchain-assisted Personalized Car Insurance with Privacy Preservation and Fraud Resistance

Cheng Huang, *Member, IEEE,* Wei Wang, *Member, IEEE,* Dongxiao Liu, *Member, IEEE,* Rongxing Lu, *Fellow, IEEE,* Xuemin (Sherman) Shen, *Fellow, IEEE*

***Abstract*—It is well known that auto insurance companies (ICs) use personalized car insurance (PCI) to continuously track drivers' behavior to determine their auto premiums. However, drivers inevitably have concerns about the transparency of data collection/processing and the potential privacy leakage. In this paper, we propose a new PCI scheme to achieve privacy preservation and transparency with the assistance of a consortium blockchain. Specifically, a blockchain is first established by a group of consortium members, and each IC can deploy insurance contracts on the blockchain to support public verification of data collection/processing and thus fulfill the transparency requirement. Then a verifiable and privacy-preserving driving behavior evaluation protocol is designed by tailoring partially homomorphic encryption and zero-knowledge proof techniques. Drivers can use the protocol to interact with ICs through the contracts, and ICs can learn drivers' behavior and set corresponding auto premiums by analyzing encrypted driving data. Furthermore, a third-party auditor (TPA) is authorized by drivers and ICs to audit encrypted driving data on the contracts and resist fraud attacks. We model the contract-based auditing as a recursive inspection game where TPA can minimize the number of audits to detect data fraud and penalize malicious drivers according to Nash equilibrium. Therefore, the proposed PCI scheme can guarantee that most of the collected driving data are not biased. Formal simulation-based security analysis is given to prove the security of the proposed scheme, and a proof-of-concept prototype is also developed on an open-source blockchain to demonstrate the feasibility.**

***Index Terms*—Personalized car insurance, blockchain, privacy preservation, fraud resistance, data auditing.**

I. Introduction

Personalized Car Insurance (PCI, a.k.a. usage-based insurance and pay-as-you-go insurance) has been widely adopted by many auto insurance companies (ICs) in recent years, as a more flexible auto insurance scheme compared with traditional schemes [1]. By continuously tracking the driving data of drivers such as daily trajectories, ICs can analyze their behavior based on particular driving safety evaluation models, and adaptively adjust their auto premiums based on the evaluation results. With the rapid advancement in mobile sensing technologies, PCI can be easily and conveniently deployed. Drivers only need to install and run applications published by ICs on their smartphones, and the applications can automatically collect and upload telematics data in the background. Apparently, PCI brings mutual advantages to both drivers and ICs, since it not only motivates drivers to drive safely with less insurance fee, but also helps companies reduce accident claim costs.

In spite of the aforementioned advantages, PCI still faces several barriers, and one of which is the lack of transparency, i.e., the rules of PCI are opaque and how drivers' auto premiums are calculated based on their behavior is unknown. According to a recent report, some PCI schemes used by ICs are not fair and may penalize people who are a group disproportionately made up of low-income people driving to or from a late shift [2]. To achieve the transparency requirement, consortium blockchain has been introduced into vehicular applications such as PCI [3]–[5]. The blockchain-assisted PCI is a more robust PCI scheme, as a group of ICs can offer decentralized trusts for drivers, compared with conventional centralized schemes. Based on the decentralization, two extensive properties, immutability and transparency, are guaranteed such that drivers' behavior and their auto premiums can be evaluated unbiasedly and also be verifiable with the assistance of smart contracts deployed in the blockchain. Moreover, the blockchain-assisted PCI is believed to be a promising scheme for sharing driving records and behavior, which is easier for drivers to switch ICs [6].

However, it becomes very challenging for not only drivers but also ICs to achieve privacy preservation in such a blockchain-assisted scheme, to meet the requirements of privacy regulations such as GDPR [7]. In light of the characteristics of the blockchain, drivers' sensitive driving data are stored and analyzed distributedly, making it hard for drivers to manage and protect their data. From another perspective, the private and valuable parameters of the driving safety evaluation models trained by different ICs must be exposed for distributed model evaluation, which also violates model privacy [8]. Aiming at these points, a straightforward method is to apply homomorphic encryption [9]: driving data and model parameters are encrypted using a shared public key generated by a group of consortium companies, and driving behavior can be analyzed in ciphertext to determine personalized auto premiums. Yet, the approach has two drawbacks: 1) ICs must work together to decrypt final evaluation results, which costs additional computational and communication resources; and 2) the model parameters can still be extracted under the model extraction attacks with the inputs of the evaluation results,

This work was supported in part by the National Natural Science Foundation of China No. 62001220, the Natural Science Foundation of Jiangsu Province BK20200440, the research grants from Huawei Technologies Canada, and the Natural Sciences and Engineering Research Council (NSERC) of Canada.

C. Huang, D. Liu, and X. Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, N2L 3G1, Canada.

W. Wang is with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, 210016, China.

R. Lu is with the Faculty of Computer Science, University of New Brunswick, E3B 5A3, Canada.

which cannot guarantee the model privacy [10].

To address the issues, we propose a new verifiable and privacy-preserving driving behavior evaluation protocol between a driver and an insurance company. The protocol is designed based on Paillier cryptosystem [11], which allows both a driver and an insurance company to cooperatively evaluate the driver's behavior atop a carefully-designed insurance contract deployed in the blockchain. The evaluation results only reveal whether the driver has a safe trip or not, since the encrypted evaluation results are randomized by both sides before disclosing them. Furthermore, to ensure the correctness and verifiability of auto premiums received by drivers, we design several efficient zero-knowledge proof protocols, and seamlessly integrate them into the proposed protocol to resist malicious adversaries that aim to bias the evaluation results.

It is noted that the encryption of driving data also raises another security challenge, i.e., how to detect intentioned data fraud in ciphertext [12]. To audit the cheats in the encrypted driving data, ICs can authorize a trusted third-party auditor such as government to decrypt the encrypted data and conduct a detection algorithm on every driving record. Nevertheless, the involvement of a trusted party increases the risk of unintentional privacy exposure and also becomes a bottleneck. To achieve fraud resistance while minimizing the participation of the trusted party, we propose a contract-based auditing method where the auditing process is modeled as a recursive inspection game between ICs and drivers on top of smart contracts. In the game, ICs set an auditing strategy based on probability and audit part of the data uploaded by drivers. Malicious drivers can also set their cheating strategies based on probability and upload fabricated driving data accordingly. With the Nash equilibrium, the parameters of an optimal auditing strategy can be calculated from the perspective of ICs, which can maximize the chance of detecting the cheating behavior of malicious drivers with minimized auditing attempts. In this way, a rationally malicious driver will not upload fabricated driving data since there is no benefit. Our method can significantly reduce the number of audits performed by the auditor to minimize the risk of privacy leakage, as a Nash equilibrium can always be found. For example, for a total number of 300 uploaded trips, although the auditor is only allowed to audit at most 3 trips (auditing 1% of all reported driving data), the possibility that a driver never cheats is 97%.

In summary, we propose a blockchain-assisted PCI scheme that simultaneously achieves privacy preservation and fraud resistance under a rationally malicious model, and the main contributions are four-folds:

- We propose a consortium-blockchain-assisted PCI scheme that achieves transparent driving behavior evaluation and auto premium calculations on the top of a well-designed smart contract of car insurance.
- We propose a verifiable and privacy-preserving driving behavior evaluation protocol, which allows ICs to correctly analyze drivers' behavior without leaking either drivers' sensitive data or companies' model parameters.
- We propose a contract-based auditing method that significantly reduces the number of audits needed to achieve fraud resistance and minimize the risk of privacy leakage due to auditing.
- We formally analyze the security of the proposed scheme based on a simulation-based approach and develop a proof-of-concept prototype to demonstrate the feasibility of our scheme using an open-source consortium blockchain and a real-world dataset.

The remainder of this paper is organized as follows. In Section II, we introduce the system model, define the adversarial model, and identify the design goals. Then, we propose our blockchain-assisted personalized car insurance scheme that achieves privacy preservation and fraud resistance in Section IV. Subsequently, security analysis and performance evaluation are presented in Sections V and VI, respectively. Finally, Section VII reviews some related works and Section VIII draws the conclusion of this work.

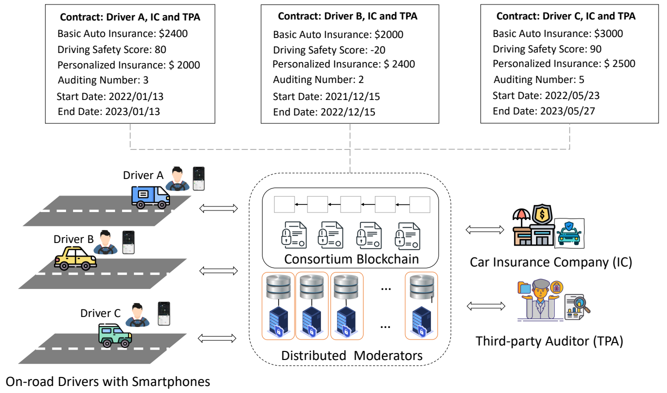
II. Models and Design Goals

In this section, we formalize the system model and adversarial model, and also present our design goals for achieving transparent, verifiable, and privacy-preserving personalized car insurance.

A. System Model

As shown in Figure 1, our system model consists of four entities: drivers, an auto insurance company (IC), distributed moderators (DMs), and a third-party auditor (TPA).

- Drivers are users of an auto insurance company who are required to upload their driving data to IC over a long period of time (e.g., six months) using a mobile App published by IC. Based on collected driving data, IC can determine a driver's safety scores and accordingly give different discounts on drivers' auto insurance premiums.
- IC builds a driving safety evaluation model for measuring drivers' behavior. By monitoring a driver's uploaded driving data trip by trip, IC can calculate the driver's auto premiums based on his/her overall safety rating which is decided by the aggregation of all driving safety scores. Drivers with higher ratings can enjoy more discounts on auto premiums.
- DMs are responsible for establishing a consortium blockchain to record each driver's driving data and auto premiums transparently as a distributed ledger. Smart contracts can be deployed in the blockchain and can be triggered by drivers, IC, and TPA to reach an agreement about auto premiums without trusted intermediates. In the real world, they can be a group of consortium auto insurance companies.
- TPA is maintained by the government. A driver and IC can authorize TPA to audit the driver's data. TPA generally owns other data sources, e.g., deployed roadside cameras, and is able to identify the manipulated driving data uploaded by a driver who wants to cheat IC.



Contract: Driver A, IC and TPA
Basic Auto Insurance: $2400
Driving Safety Score: 80
Personalized Insurance: $ 2000
Auditing Number: 3
Start Date: 2022/01/13
End Date: 2023/01/13

Contract: Driver B, IC and TPA
Basic Auto Insurance: $2000
Driving Safety Score: -20
Personalized Insurance: $ 2400
Auditing Number: 2
Start Date: 2021/12/15
End Date: 2022/12/15

Contract: Driver C, IC and TPA
Basic Auto Insurance: $3000
Driving Safety Score: 90
Personalized Insurance: $ 2500
Auditing Number: 5
Start Date: 2022/05/23
End Date: 2023/05/27

Driver A
Driver B
Driver C
On-road Drivers with Smartphones
Consortium Blockchain
...
...
Distributed Moderators
Car Insurance Company (IC)
Third-party Auditor (TPA)

Fig. 1. System model under consideration

B. Adversarial Model

Based on the system model, adversaries are considered to be rationally malicious. Compared with fully malicious adversaries, rational adversaries only perform maliciously to maximize their profits. Their malicious behaviors are outcome-driven, i.e., adversaries will take the action that results in the best outcome. Concretely, IC may behave maliciously to infer sensitive information from collected drivers' data. Although various privacy laws are enforced to regulate IC's privacy policies, they cannot technically prevent internal adversaries from compromising data privacy as internal attacks are hard to be detected. To maximize its payoff, IC may choose to decrease the safety score of a driver who drives safely, by deviating from the scheme. We do not consider the situation that IC deliberately modifies their model parameters, since the attack can be easily handled by the technique proposed in [13], where IC is required to prove the model accuracy of a public dataset while preserving model privacy. From another perspective, drivers may behave maliciously to cheat IC by uploading fake driving data. By doing so, they can obtain more discounts on their auto insurance premium. In addition, drivers may be curious about the model parameters trained by IC and behave maliciously to compromise the model privacy.

Following the literature that is designed based on a consortium blockchain [14], [15], our scheme's security is established on an assumption that only a minority of DMs can be compromised by malicious drivers and IC, and the majority of them are honest (e.g., more than $\frac{2}{3}$ DMs are honest if Practical Byzantine Fault Tolerance protocol (PBFT) is chosen as the consensus protocol used in the consortium blockchain) [16], [17]. As a government department whose purpose is to identify data fraud in our scheme, TPA is conditionally trusted. TPA is normally trusted to some extent, but to make the model more realistic, less auditing information should be shared with TPA, since the information shared with TPA may be leaked unintentionally.

Strong collusion attacks are also considered in our scheme, where malicious drivers can collude with each other to perform attacks on model parameters. Malicious drivers can also collude with IC to infer other drivers' reported data. Note that we mainly focus on the application-layer attacks and thus other attacks are beyond the scope of the paper.

C. Design Goals

Based on the system model and adversarial model, we have the following objectives:

- Driver Privacy: The driving data collected from a driver should not be exposed to IC, other drivers, or external adversaries to achieve privacy preservation in the system.
- Model Privacy: The model parameters hold by ICs for evaluating a driver's safety score should not be obtained by any internal or external adversaries.
- Fraud Resistance: The scheme should prevent drivers from reporting fabricated driving data as much as possible, with the help of TPA.
- Transparency of Auto Insurance: How ICs collect drivers' driving data and calculate their insurance fees are transparent. To achieve the objective, the following two subgoals should be guaranteed:
  - Public Verifiability: Each driver's safety score can be verified by any entity to ensure that the score is calculated following the evaluation model defined by ICs.
  - Permission-based Audit: Every audit is performed based on the permission of drivers, i.e., TPA and ICs cannot audit the data of a driver without the permission of the driver.

III. Preliminaries

In this section, we present some preliminaries including cryptographic building blocks, a driving safety evaluation model for calculating personalized auto premiums, the basic knowledge about game theory, and consortium blockchain.

A. Building Blocks

1) Paillier Cryptosystem: Paillier Cryptosystem is a partially homomorphic cryptosystem that supports additions of ciphertexts and multiplications of plaintexts [11]. It achieves semantic security against chosen-plaintext attacks (*IND-CPA*) and consists of five algorithms:

- *KeyGen:* The algorithm generates an RSA modulus $N = pq$ ($\varphi(N) = (p-1)(q-1)$), where p and q are safe primes, and outputs public key $\mathtt{pk} = N$ and private key $\mathtt{sk} = (q, p, \lambda = \mathsf{Lcm}(p-1, q-1))$, where $\mathsf{Lcm}()$ is an algorithm for calculating least common multiple;
- *Encryption:* The algorithm inputs a message $m \in \mathbb{Z}_N$ and $\mathtt{pk}$, chooses a random number $\Gamma \in \mathbb{Z}_N^*$, and outputs $c = \mathsf{Enc}(m) = (1+N)^m \Gamma^N \bmod N^2$;
- *Decryption:* The algorithm inputs a ciphertext c and $\mathtt{sk}$, and outputs $m = \frac{L(c^\lambda \bmod N^2)}{L((1+N)^\lambda \bmod N^2)}$, where $L(x) = \frac{x-1}{N}$;
- *Addition:* The algorithm inputs $c_1 = \mathsf{Enc}(m_1)$ and $c_2 = \mathsf{Enc}(m_2)$, and outputs $c_1 \cdot c_2 = c_3 = \mathsf{Enc}(m_1 + m_2)$;
- *Multiplication:* The algorithm inputs $c_1 = \mathsf{Enc}(m_1)$ and $m_2 \in \mathbb{Z}_N$, and outputs $c_2 = c_1^{m_2} = \mathsf{Enc}(m_1 \cdot m_2)$.

2) Fujisaki-Okamoto Commitment: Given an RSA modulus N, let g and h be two elements of large orders in $\mathbb{Z}_N^*$, and the discrete logarithms of g base h or vice versa are unknown, a Fujisaki-Okamoto (FO) commitment to an integer m can be defined as $C = g^m h^r \bmod N$, where r is a random number chosen from $[0, 2^\tau N]$, where τ is a security parameter [18]. The commitment has two properties:

- *Statistically Hiding:* C statistically reveals no information, i.e., there exists a simulator $\mathcal{S}$ that can output simulated commitments which are statistically indistinguishable from the real one;
- *Computationally Binding:* A polynomial-time adversary cannot generate two commitments $C_1 = g^{m_1}h^{r_1}$ and $C_2 = g^{m_2}h^{r_2}$ on (m_1, r_1) and (m_2, r_2), where $C_1 = C_2$ but $m_1 \neq m_2$.

3) Zero-knowledge Proof: A zero-knowledge proof of knowledge (ZkPoK) protocol involves two parties: a prover and a verifier. The prover can prove to the verifier that $(x, w) \in \mathcal{R}$, i.e., some NP relation $\mathcal{R}$ is correct about a statement x without revealing a witness w. Similar to [19], we denote a proof as $\pi \leftarrow \mathsf{ZkPoK}\{(w) : (x, w) \in \mathcal{R}\}$, which has three properties:

- *Completeness:* The verifier will always accept π if $(x, w) \in \mathcal{R}$;
- *Soundness:* The verifier will accept π with negligible probability if $(x, w) \notin \mathcal{R}$, i.e, there exist a knowledge extractor who can extract the witness, w, by using rewinding techniques.
- *Zero-Knowledge:* A simulator $\mathcal{S}$ can be constructed, who accesses the verifier as a black box and can generate a transcript for a true statement x. The generated transcript is computationally indistinguishable from the real transcript received by the verifier.

B. Driving Safety Evaluation Model

IC uses a logistic regression (LR) model to evaluate the driving safety of a driver in a trip as $y = \vec{w} \cdot \vec{x} + \epsilon$, where $(\vec{w} = (w_1, w_2, ...), \epsilon)$ are the model parameters and the model intercept, $\vec{x} = (x_1, x_2, ...)$ are the driving features extracted from the detailed driving data of the driver's one trip, and $\mathsf{sigmoid}(y) \in [0, 1]$ is the model classification result where sigmoid is the Sigmoid function. $\mathsf{sigmoid}(y) \geq 0.5$ means the driving behavior in the trip is measured to be safe (the probability that the driver drives safely is equal to or larger than 0.5) and $\mathsf{sigmoid}(y) < 0.5$ means the opposite. The LR model is believed to be an effective and popular model for calculating the driving safety score based on drivers' behavior in existing research works [20]–[22].

Without loss of generality, Extract is a feature extraction function used by IC, we have $\vec{x} = \mathsf{Extract}(\mathrm{info})$, where info is the time-series driving data, including the telematics data of a trip and other information. The feature dimension is set as n, i.e., the lengths of $\vec{w}$ and $\vec{x}$ are n. We denote ψ as a pre-defined predicate of the model parameters, i.e., $\psi(\vec{w}, \epsilon) = 1$, which means $w_i \in [-2^{l_w}+1, 2^{l_w}-1]$ for $i = 1$ to n and $\epsilon \in [-2^{l_\epsilon} + 1, 2^{l_\epsilon}-1]$. We denote ϕ as a pre-defined predicate of the driving features, i.e., $\phi(\vec{x}) = 1$, which means $x_i \in [-2^{l_x}+1, 2^{l_x}-1]$ for $i = 1$ to n. With such predicates, y locates in a range of $[-2^\kappa + 1, 2^\kappa - 1]$ $(\kappa \geq \max(l_w + l_x + 5, l_\epsilon + 5))$. To obtain the PCI premium, the total number of trips that need to be uploaded by a driver is $\mathcal{N}$, i.e. the driver's total uploaded driving features are $(\vec{x}_1, \vec{x}_2, ..., \vec{x}_\mathcal{N})$ and the driver obtains $\mathcal{N}$ driving safety evaluation results $(y_1, y_2, ..., y_\mathcal{N})$. The initial rating of the driver behavior is set as $R = 0$ and for trip i, if $y_i \geq 0$ $(\mathsf{sigmoid}(y_i) \geq 0.5)$, the trip is classified to be safe, $R = R + 1$; otherwise, $\mathsf{sigmoid}(y_i) < 0.5$, and $R = R - 1$. Supposing that the basic auto insurance quote of the driver is Q, if $R < 0$, $\hat{Q} = (1 + 20\% * \frac{|R|}{\mathcal{N}}) * Q$ or if $R \geq 0$, $\hat{Q} = (1 - 25\% * \frac{|R|}{\mathcal{N}}) * Q$, where $\hat{Q}$ is the PCI premium[1].

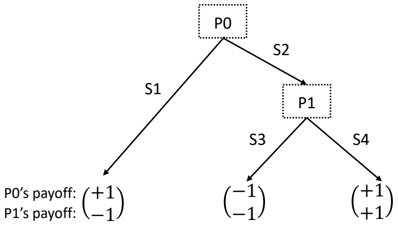


Fig. 2. A typical game with two players $P0$ and $P1$

C. Games & Nash Equilibrium

We describe a non-cooperative game of multiple players as a game tree in this paper. Each node in the game tree represents a player, and routes from the root node to one leaf node denote different strategies adopted by participating players. Each leaf node points to the players' payoffs which are decided by how they choose strategies. As shown in Figure 2, two players, $P0$ and $P1$, play the game and do not cooperate. The game starts from $P0$. If $P0$ chooses the strategy $S1$, the game ends, and $P0$ obtains the payoff $+1$ while $P1$ gets the payoff -1. If $P1$ chooses the strategy $S2$, $P1$ can choose different routes with different outcomes. In most of the realistic cases, a game runs in a situation where players have imperfect information, i.e., a player does not know the actions taken by other players.

One of the most common decision-making theorems to locate an optimal solution of a non-cooperative game is the Nash equilibrium. With the theorem, one route of a game tree can be found, which guarantees that each player's payoff is best considering other players' strategies. When a Nash equilibrium is reached, no one can gain more by changing strategies if other players do not change their strategies. In a standard multiplayer non-cooperative game, if the number of players and the strategies of each player are finite, at least one Nash equilibrium can be found, which may contain mixed strategies. More detailed and formal definitions of game trees and Nash equilibrium can be found in [23].

D. Consortium Blockchain

A consortium blockchain is a distributed service maintained by a group of pre-selected entities (they are named moderators

[1] The discount percentages follow https://www.desjardinsgeneralinsurance.com/auto-insurance/ajusto

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2022.3215811

in our system). The blockchain mainly offer two types of trusted services: storage service and computation service. Data can be sent to the storage service and stored by the blockchain to guarantee data integrity. A computer program that involves many functions can also be outsourced to the computation service, and the program is named "smart contract" in the blockchain. Each function of a contract can be triggered and executed by the blockchain, and each execution of a function may result in some state changes of the contract, and the changes are also updated and stored by the blockchain.

For auto insurance companies, they can easily build a consortium blockchain for the business of car insurance by using open-source consortium blockchains such as Hyperledger fabric. Compared with public blockchains, the consortium blockchain has low transaction confirmation delay and high transaction throughput, which makes it more suitable for real-world mobile applications.

IV. Proposed Blockchain-assisted Personalized Car Insurance Scheme

In this section, we propose a blockchain-assisted scheme for achieving secure personalized car insurance (PCI). The scheme mainly involves three high-level ideas and their detailed constructions: 1) designing the smart contract for achieving PCI atop blockchain and initializing the blockchain; 2) verifiable and privacy-preserving evaluation of a driver's overall safety rating based on a series of trip-based driving safety scores; and 3) a contract-based auditing method for detecting malicious drivers.

A. Contract Design & System Setup

The core of a blockchain-assisted scheme is the design of the contract for PCI among a driver, IC, and TPA. The whole procedures of our scheme center around the contract that incorporates nine functions, as shown in Figure 3.

Function ***Init*** defines a driver's unique insurance number (DIN), a basic auto insurance premium (Q), committed model parameters ($\vec{C}$), encrypted model parameters ($\vec{E}$), the public keys of IC and TPA (pk and $\widehat{\texttt{pk}}$, respectively), the required deposit (DP) and the number of uploaded trips ($\mathcal{N}$), and the permitted number of audits($\mathcal{M}$). Also, some parameters related to the driver are initialized, including the serial number of updated trips (NUM), the initial rating of the driver (R), the personalized insurance premium ($\hat{Q}$), and the state of the driver's PCI (STATE). There are 8 states indicating different stages of the driver's car insurance, and finally the states will be transitioned to 6, 7, or 8, that implies the end, as shown in Figure 4. IC deploys the contract into the blockchain to trigger ***Init***, which implies the beginning of the evaluation of the driver's auto premiums.

```
Init:       Initialize (DIN, $Q$, $\vec{C}$, $\vec{E}$, pk, $\widehat{\texttt{pk}}$, DP, $\mathcal{N}$, $\mathcal{M}$);
            Set NUM $= 0$, $R = 0$, $\hat{Q} = Q$, and STATE $= 0$;
Deposit:    Upon receiving (DP) from Driver
              Assert STATE $= 0$;
              Deposit(DP);
              STATE $= 1$;
Record:     Upon receiving ($\texttt{DD}_i, \pi_{\texttt{DD}_i}, \texttt{DA}_i$) from Driver
              Assert NUM $< \mathcal{N}$;
              Assert STATE $= 1$;
              Assert True $=$ $\mathsf{Verify}_1(\pi_{\texttt{DD}_i}, \texttt{DD}_i)$;
              Store(NUM, $\texttt{DD}_i, \texttt{DA}_i$);
              NUM $=$ NUM $+ 1$;
              If $\mathcal{N} =$ NUM then STATE $= 2$;
Evaluate:   Upon receiving ($\{\texttt{DS}_i\}_{i=1}^{\mathcal{N}}, \{\pi_{\texttt{DS}_i}\}_{i=1}^{\mathcal{N}}$) from IC
              Assert STATE $= 2$;
              For $i = 1$ to $\mathcal{N}$
                If $1 =$ Check($\texttt{DS}_i$) then $R = R + 1$;
                Else If True $=$ $\mathsf{Verify}_2(\texttt{DD}_i, \pi_{\texttt{DS}_i}, \texttt{DS}_i)$:
                  Then $R = R - 1$;
              If $R \geq 0$ then $\hat{Q} = (1 - 25\% * \frac{|R|}{\mathcal{N}}) * Q$;
              Else $\hat{Q} = (1 + 20\% * \frac{|R|}{\mathcal{N}}) * Q$;
              STATE $= 3$;
Audit:      Upon receiving ($\{\overline{\texttt{NUM}}_i\}_{i=1}^{\mathcal{M}}$) from IC
              Assert STATE $= 3$;
              Assert $\{\overline{\texttt{NUM}}_i \in [0, \mathcal{N} - 1]\}_{i=1}^{\mathcal{M}}$;
              AuditEvent($\{\overline{\texttt{NUM}}_i\}_{i=1}^{\mathcal{M}}$);
              STATE $= 4$;
Authorize:  Upon receiving ($\{\overline{\texttt{NUM}}'_i\}_{i=1}^{\mathcal{M}}, \{\texttt{DK}_i\}_{i=1}^{\mathcal{M}}, \pi_{\texttt{DK}}$) from Driver
              Assert STATE $= 4$;
              Assert True $=$ $\mathsf{Verify}_3(\pi_{\texttt{DK}}, \{\texttt{DK}_i\}_{i=1}^{\mathcal{M}})$;
              Assert $\{\overline{\texttt{NUM}}_i = \overline{\texttt{NUM}}'_i\}_{i=1}^{\mathcal{M}}$;
              AuthEvent($\{\overline{\texttt{NUM}}'_i\}_{i=1}^{\mathcal{M}}, \{\texttt{DK}_i\}_{i=1}^{\mathcal{M}},$);
              STATE $= 5$;
Inspect:    Upon receiving (RES) from TPA
              Assert STATE $= 5$;
              If True $=$ RES then Return(DP);
              Else Transfer(DP, IC);
              STATE $= 6$;
Confirm:    Upon receiving ('confirm') from IC
              Assert STATE $= 3$;
              Return(DP);
              STATE $= 7$;
Quit:       Upon receiving ('quit') from Driver
              Assert STATE $< 4$;
              Return(DP);
              STATE $= 8$;
```

Fig. 3. The contract for personalized car insurance

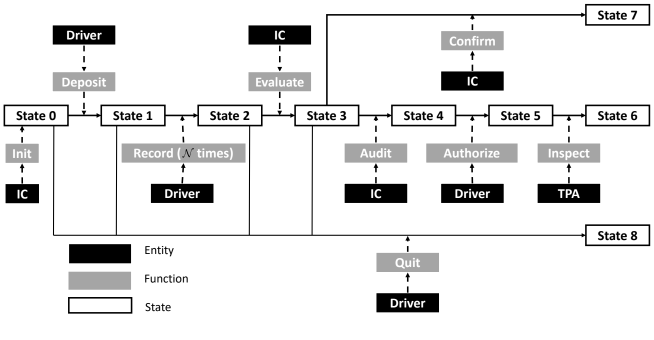


Fig. 4. The state transitions in the contract

In the contract, function ***Deposit*** needs to be triggered firstly by the driver to deposit DP into the contract. When the deposit is completed, each time the driver drives a trip i, a transaction consisting of the encrypted driving features ($\texttt{DD}_i$), the proof of the driving features ($\pi_{\texttt{DD}_i}$), the encrypted raw driving data ($\texttt{DA}_i$) is sent to the contract, and function ***Record*** is triggered by the driver to verify and store $\texttt{DD}_i$ and $\texttt{DA}_i$ (we use Store to denote the procedures of storing the data in the contract). After the encrypted driving features of $\mathcal{N}$ trips are documented, function ***Evaluate*** can be triggered by IC to generate the corresponding overall rating of the driver (R) and the personalized auto insurance premium ($\hat{Q}$). To trigger the function, IC needs to send the transaction involving the private results of driving safety scores of $\mathcal{N}$ trips and the proof of the private results, $\{\texttt{DS}_i\}_{i=1}^{\mathcal{N}}$ and $\{\pi_{\texttt{DS}_i}\}_{i=1}^{\mathcal{N}}$, to the contract.

The premium ($\hat{Q}$) is not confirmed yet. IC will follow

TABLE I
THE FUNCTIONS DEFINED IN THE CONTRACT

Function	Description
Init	IC initializes the car insurance contract (STATE 0)
Deposit	Driver deposits the money (STATE $0 \to 1$)
Record	Driver reports driving data (STATE $1 \to 2$)
Evaluate	IC evaluates the car insurance (STATE $2 \to 3$)
Audit	IC starts the process of auditing (STATE $3 \to 4$)
Authorize	Driver authorizes the auditing (STATE $4 \to 5$)
Inspect	TPA ends the contract with auditing (STATE $5 \to 6$)
Confirm	IC ends the contract without auditing (STATE $3 \to 7$)
Quit	Driver ends the contract through quitting (STATE $0 \sim 3 \to 7$)

the contract-based auditing method defined in Section IV-C, and can choose to either trigger function ***Confirm*** to confirm the premium without auditing or trigger function ***Audit*** to initialize an auditing event (we use AuditEvent to represent the procedures of starting an audit event in the contract) to audit uploaded driving data of up to $\mathcal{M}$ trips. If IC would like to audit the uploaded data by triggering ***Audit***, IC can send the serial numbers of the chosen $\mathcal{M}$ trips, $\{\overline{\texttt{NUM}}_i\}_{i=1}^{\mathcal{M}}$ to the contract. If the audit event is received, the driver should also trigger function ***Authorize*** to authorize the auditing event posted by IC. Similar to the transaction sent by IC, the driver also needs to send the serial numbers of the chosen $\mathcal{M}$ trips, $\{\overline{\texttt{NUM}}_i'\}_{i=1}^{\mathcal{M}}$, the authorization keys, $\{\texttt{DK}_i\}_{i=1}^{\mathcal{M}}$, and the proof of authorization keys, $\pi_{\texttt{DK}}$, to the contract and trigger an authorization event (we use AuthEvent to represent the procedures of starting an authorization event in the contract). After receiving the audit event and the authorization event, by using the authorization keys and its private key, TPA can recover the raw driving data of up to $\mathcal{M}$ trips and detect whether the trips are real or fabricated, and trigger function ***Inspect*** to report the result and end the contract. Without auditing, IC can directly trigger function ***Confirm*** by sending 'confirm' and end the contract. According to the contract, when no malicious behavior is detected, $\hat{Q}$ is confirmed and the deposit will be return (Return); otherwise, the driver will be punished by losing the deposit. The driver can also choose to quit the contract by sending 'quit' to trigger function ***Quit*** before the auditing happens. We summarize the descriptions of the functions in TABLE I.

In addition to these functions, there are other redundant functions such as making written data accessible and readable and allowing the driver or IC to end the contract in case of timeout. Since these functions are trivial, we omit them for brevity. In the contract, we use Verify_1 and Verify_2 to denote two verification algorithms that verify the zero-knowledge proofs of the correctness of encrypted driving features and the driver's driving safety scores, and use Check to denote the checking algorithm that rates a trip-based driving safety score as high or low. How to encrypt the driving features and the raw driving data, the proofs, the verifications algorithms (a.k.a. zero-knowledge proofs), and the checking algorithm are not discussed here and will be given in Section IV-B and IV-C.

One-time Setup: Before IC deploys the contract for drivers, some public parameters need to be negotiated once among drivers, IC, and TPA, including a security parameter τ, an RSA modulus $N = pq$ where p and q are two safe primes, two elements of large orders in $\mathbb{Z}_N^*$, $g \in \mathbb{Z}_N^*$ and $h \in \mathbb{Z}_N^*$, a symmetric-key encryption/decryption algorithm $\mathsf{SEnc}(\cdot)/\mathsf{SDec}(\cdot)$, and a public key encryption/encryption algorithm $\mathsf{PEnc}(\cdot)/\mathsf{PDec}(\cdot)$.

$$\begin{aligned}\pi_{\text{Params}} \leftarrow \mathsf{ZkPoK}\Big\{&(\vec{w}, \epsilon, \{r_i\}_{i=1}^{n+1}, \{\Gamma_i\}_{i=1}^{n+1}) : \psi(\vec{w}, \epsilon) = 1\\ &\wedge \{C_i = g^{w_i} h^{r_i}\}_{i=1}^{n} \wedge C_{n+1} = g^{\epsilon} h^{r_{n+1}}\\ &\wedge \{E_i = (1+N)^{w_i} \Gamma_i^N\}_{i=1}^{n} \wedge E_{n+1} = (1+N)^{\epsilon} \Gamma_{n+1}^N\Big\}.\end{aligned}$$

Fig. 5. Proof of the correctness of committed and encrypted model parameters

Note that, the discrete logarithms of g base h or vice versa are unknown to drivers, IC, and TPA. The factorization of N should be only known to IC and unknown to drivers and TPA. For this purpose, our scheme can rely on DMs to generate and publish N, g, and h, and shares two safe primes, p and q, with IC in a decentralized setting [24], [25]. The bit length of N is ζ. Based on the parameters, IC can generate and publish its public key, $\texttt{pk} = N$, and keep its private key $\texttt{sk} = (q, p, \lambda = \mathsf{Lcm}(p-1, q-1))$. IC additionally chooses $2(n+1)$ random numbers $\{r_i\}_{i=1}^{n+1}$ and $\{\Gamma_i\}_{i=1}^{n+1}$, where $r_i \in [0, 2^{\tau} N)$ and $\Gamma_i \in (0, N)$. IC then generate the committed model parameters, $\vec{C} = (C_1, C_2, ..., C_n, C_{n+1})$, and the encrypted model parameters, $\vec{E} = (E_1, E_2, ..., E_n, E_{n+1})$, where $C_i = g^{w_i} h^{r_i} \bmod N$ and $E_i = (1+N)^{w_i} \Gamma_i^N \bmod N^2$ for $i = 1, 2, ..., n$ and $C_{n+1} = g^{\epsilon} h^{r_{n+1}} \bmod N$ and $E_{n+1} = (1+N)^{\epsilon} \Gamma_{n+1}^N \bmod N^2$. A corresponding zero knowledge proof, π_{Params} can be constructed and verified publicly, as shown in Figure 5, whose objective is to certify the model parameters are correct formed and consistent with ψ. TPA also publishes its public key $\widehat{\texttt{pk}}$, and keep its private key $\widehat{\texttt{sk}}$.

B. Privacy-preserving Evaluation of Drivers' Behavior

A driver's overall rating (R) is determined by $\mathcal{N}$ driving safety scores received by the driver ($\{y_i\}_{i=1}^{\mathcal{N}}$). Hence, privacy-preserving evaluation of R means secure calculations of $\{y_i\}_{i=1}^{\mathcal{N}}$ without exposing $\vec{w}$, ϵ, and $\{\vec{x_i}\}_{i=1}^{\mathcal{N}}$, where $\{\vec{x_i}\}_{i=1}^{\mathcal{N}}$ are the driver's driving features of $\mathcal{N}$ trips. A straightforward idea is to utilize Paillier homomorphic encryption to achieve secure calculations of $\{y_i\}_{i=1}^{\mathcal{N}}$. However, as the relationship among y_i, $\vec{w}$, $\vec{x_i}$, and ϵ is linear, a malicious driver can easily extract $\vec{w}$ and ϵ since y_i is deterministic based on $\vec{x_i}$, if $n+1$ driving safety scores $\{y_i\}_{i=1}^{n+1}$ is revealed based on Gaussian elimination. To hide the relationship among y_i, $\vec{w}$, $\vec{x_i}$, and ϵ, our idea is to disclose a randomized value of y_i, $\delta_i = \alpha_i(a_i y_i + b_i) + \beta_i$, where α_i and β_i are random numbers chosen by IC and a_i and b_i are random numbers chosen by the driver. By carefully selecting the bit lengths of $(\alpha_i, \beta_i, a_i, b_i)$, δ_i only shows whether $y_i \geq 0$ or $y_i < 0$ for trip i, which is sufficient to determine the updates of R. To achieve the goal, we design a privacy-preserving driving behavior evaluation protocol between the driver and IC, as shown in Protocol 1. In the protocol, a driver will report $\texttt{DD}_i$ and $\texttt{DA}_i$ for trip i, and IC will monitor the contract until enough data are collected, and evaluate the encrypted driving features to obtain the driver's overall safety rating and auto premium.

Protocol 1 Privacy-preserving Evaluation of Driving Behavior

```
1: (Driver) Reporting Driving Data (Trip $i$ for $i = 1$ to $\mathcal{N}$):
    1) chooses a random number $r_i$ whose bit length is $l_r$ ($l_r > \kappa$);
    2) calculates $\mathcal{E}_i$, the Paillier ciphertext of $\vec{w} \cdot \vec{x_i} + \epsilon + r_i$:
       $$\mathcal{E}_i = \prod_{j=1}^{n} E_j^{x_{i,j}} \cdot E_{n+1} \cdot (1+N)^{r_i} \bmod N^2;$$
    3) chooses two random numbers $a_i$ and $b_i$ whose bit lengths are $l_a$ and $l_b$, respectively ($l_b > \kappa$);
    4) calculates $r'_i = a_i r_i$, and computes $\mathcal{E}'_i$, the Paillier ciphertext of $a_i(\vec{w} \cdot \vec{x_i} + \epsilon + r_i) - a_i r_i + b_i = a_i y_i + b_i$:
       $$\mathcal{E}'_i = \mathcal{E}_i^{a_i} \cdot (1+N)^{-r'_i} \cdot (1+N)^{b_i} \bmod N^2;$$
    5) chooses a random number $v_i \in [0, 2^{\tau} N)$, calculates $v'_i = a_i v_i$, and commits to $r_i$ and $r'_i$ as $\mathtt{com}_i = g^{r_i} h^{v_i}$ and $\mathtt{com}'_i = g^{r'_i} h^{v'_i}$;
    6) sets $\mathtt{DD}_i = (\mathcal{E}_i, \mathcal{E}'_i, \mathtt{com}_i, \mathtt{com}'_i)$ and generates the proof $\pi_{\mathtt{DD}_i}$;
    7) chooses a symmetric key $k_i$, and encrypts $\vec{x_i}||\text{info}_i$ as $\mathtt{ct}_i = \mathsf{SEnc}(k_i, (\vec{x_i}||\text{info}_i))$, where $\text{info}_i$ is the raw driving data of the trip;
    8) sets $\mathtt{DA}_i = \mathtt{ct}_i$, and sends $(\mathtt{DD}_i, \pi_{\mathtt{DD}_i}, \mathtt{DA}_i)$ to trigger function Record.
2: (IC) Evaluating Driving Safety (All $\mathcal{N}$ trips):
    1) downloads $\{i, \mathcal{E}'_i\}_{i=1}^{\mathcal{N}}$ for $i = 1$ to $\mathcal{N}$;
    2) chooses $2\mathcal{N}$ random numbers $\{\alpha_i, \beta_i\}_{i=1}^{\mathcal{N}}$, and the bit lengths of $\alpha_i$ and $\beta_i$ (for $i = 1$ to $\mathcal{N}$) are $l_\alpha$ and $l_\beta$ ($l_\beta > l_a + \kappa$);
    3) calculates $\mathfrak{E}_i = (\mathcal{E}'_i)^{\alpha_i}(1+N)^{\beta_i}$ for $i = 1$ to $\mathcal{N}$;
    4) decrypts $\mathfrak{E}_i$ using $\lambda$ to obtain $m_i = \alpha_i(a_i y_i + b_i) + \beta_i = \alpha_i a_i y_i + \alpha_i b_i + \beta_i$ for $i = 1$ to $\mathcal{N}$;
    5) chooses a random number, $\Upsilon_i \in (0, N)$, and encrypts $m_i$ as $\mathcal{U}_i = (1+N)^{m_i} \Upsilon_i^N \bmod N^2$ for $i = 1$ to $\mathcal{N}$;
    6) calculates $\mathcal{D}_i = \mathfrak{E}_i \cdot \mathcal{U}_i^{-1} = Z_i^N \bmod N^2$ for $i = 1$ to $\mathcal{N}$;
    7) extracts $Z_i$ from $\mathcal{D}_i$, i.e., $Z_i = \mathcal{D}_i^{N^{-1} \bmod \varphi(N)} \bmod N$ for $i = 1$ to $\mathcal{N}$, where $\varphi(\cdot)$ is the Euler function;
    8) sets $\mathtt{DS}_i = (m_i, \mathfrak{E}_i, \mathcal{U}_i, \mathcal{D}_i)$ and generates the proof $\pi_{\mathtt{DS}_i}$ for $i = 1$ to $\mathcal{N}$;
    9) sends $(\{\mathtt{DS}_i\}_{i=1}^{\mathcal{N}}, \{\pi_{\mathtt{DS}_i}\}_{i=1}^{\mathcal{N}})$ to trigger function Evaluate.
```

$$
\begin{aligned}
&\pi_{\mathtt{DD}_i} \leftarrow \mathsf{ZkPoK}\Big\{(\vec{x_i}, r_i, r'_i, a_i, b_i, v_i, v'_i, \widehat{\Gamma}_i, \overline{\Gamma}_i) : \phi(\vec{x_i}) = 1 \\
&\wedge \mathtt{com}_i = g^{r_i} h^{v_i} \wedge \mathtt{com}'_i = \mathtt{com}_i^{a_i} = g^{r'_i} h^{v'_i} \\
&\wedge \mathcal{E}_i = \prod_{j=1}^{n} E_j^{x_{i,j}} \cdot E_{n+1} \cdot (1+N)^{r_i} \\
&\wedge \mathcal{E}'_i = \mathcal{E}_i^{a_i} \cdot (1+N)^{-r'_i} \cdot (1+N)^{b_i} \wedge r_i \in [2^{l_r - 1}, 2^{l_r} - 1] \\
&\wedge a_i \in (2^{l_a - 1}, 2^{l_a} - 1] \wedge b_i \in (2^{l_b - 1}, 2^{l_b} - 1]\Big\}.
\end{aligned}
$$

Fig. 6. Proof of the correctness of encrypted driving features that can be verified by Verify_1 algorithm

$$
\begin{aligned}
&\pi_{\mathtt{DS}_i} \leftarrow \mathsf{ZkPoK}\Big\{(\alpha_i, \beta_i, \Upsilon_i, Z_i) : \mathcal{D}_i = Z_i^N \\
&\wedge \mathfrak{E}_i = (\mathcal{E}'_i)^{\alpha_i}(1+N)^{\beta_i} \wedge \mathcal{U}_i = (1+N)^{m_i} \Upsilon_i^N \\
&\wedge \alpha_i \in (2^{l_\alpha - 1}, 2^{l_\alpha} - 1] \wedge \beta_i \in (2^{l_\beta - 1}, 2^{l_\beta} - 1]\Big\}.
\end{aligned}
$$

Fig. 7. Proof of the correctness of driving safety scores that can be verified by Verify_2 algorithm

Considering that the driver and IC may behave maliciously in the protocol, zero-knowledge proof is applied to guarantee that the driver and IC do not deviate from the protocol. From the driver side, a zero-knowledge proof protocol is designed, $\pi_{\mathtt{DD}_i}$, as shown in Figure 6, which can be verified to ensure that $\mathcal{E}_i$ is a valid Paillier ciphertext that is calculated correctly based on the driver's logistic regression model ($y_i = \vec{w} \cdot \vec{x_i} + \epsilon$) and $\mathcal{E}'_i$ can be decrypted by IC to obtain $a_i y_i + b_i$ for trip i. As IC holds the private key, $\mathtt{sk}$, Figure 6 is not straightforward. Our idea is to introduce a random number, r_i, into the proof and use r_i to hide the private driving safety score y_i of trip i. Our design first enables the driver to prove $\mathcal{E}_i$ is a valid Paillier ciphertext of $y_i + r_i$, and then allows the driver to prove that $\mathcal{E}'_i$ is another valid Paillier ciphertext of $a_i y_i + b_i$ that eliminates r_i from $\mathcal{E}_i$ and adds two addition random numbers, (a_i, b_i). By doing so, IC can only decrypt $\mathcal{E}_i$ and $\mathcal{E}'_i$ to obtain the random values, $y_i + r_i$ and $a_i y_i + b_i$.

Another zero-knowledge proof is designed for IC, $\pi_{\mathtt{DS}_i}$, as shown in Figure 7, which can be verified to show that $\mathfrak{E}_i$ is a valid Paillier ciphertext that encrypts $\alpha_i a_i y_i + \alpha_i b_i + \beta_i$ and $m_i = \alpha_i a_i y_i + \alpha_i b_i + \beta_i$. To prove that m_i is the exact plaintext without exposing IC's private key, two additional ciphertexts, $\mathcal{U}_i$ and $\mathcal{D}_i$, are introduced, where $\mathcal{U}$ is a new Paillier ciphertext that encrypts m_i and $\mathcal{D}_i$ is the multiplication of $\mathfrak{E}_i$ and $\mathcal{U}^{-1}$. Based on Paillier's homomorphic property, if $m_i = \alpha_i a_i y_i + \alpha_i b_i + \beta_i$, we have $\mathcal{D}_i = Z_i^N \bmod N^2$, where Z_i is a random number located in $(0, N)$. Using the private key, IC can extract Z_i, and conduct the proof, $\pi_{\mathtt{DS}_i}$. The random value, $m_i = \alpha_i a_i y_i + \alpha_i b_i + \beta_i$, can be used for testing whether $y_i \geq 0$ or $y_i < 0$ if α_i and β_i follows some settings. When $y_i \geq 0$, $m_i \bmod N$ is a random value whose bit length is much smaller than ζ, and the checking algorithm, Check, output 1; otherwise the value's bit length is close to ζ, and Check outputs 0.

Instantiations of π_{Params}, $\pi_{\mathtt{DD}_i}$, $\pi_{\mathtt{DS}_i}$: The proposed three zero-knowledge proofs can be efficiently instantiated using the Sigma-protocol and can be securely transformed to the non-interactive zero-knowledge proofs using Fiat-Shamir heuristic in the random oracle model. We can achieve efficient range proofs for $\vec{w}$, ϵ, $\{\vec{x_i}\}_{i=1}^{\mathcal{N}}$, $\{a_i\}_{i=1}^{\mathcal{N}}$, $\{b_i\}_{i=1}^{\mathcal{N}}$, $\{\alpha_i\}_{i=1}^{\mathcal{N}}$, and $\{\beta_i\}_{i=1}^{\mathcal{N}}$, by using the techniques proposed in [26].

Correctness Analysis: The proposed protocol is correct when parameters are selected properly. Since our protocol is constructed based on Paillier cryptosystem, the inputs and outputs of the calculations in the protocol should be in the field $\mathbb{Z}_N^*$. Therefore, the following conditions should be satisfied: $l_r << \zeta$, $\kappa << \zeta$, $l_a + \kappa << \zeta$, $l_b << \zeta$, $l_\alpha + l_a + \kappa << \zeta$, $l_\alpha + l_b << \zeta$, and $l_\beta << \zeta$. Moreover, to guarantee that the final result of the evaluation model, $m_i = \alpha_i a_i y_i + \alpha_i b_i + \beta_i$, can be correctly decrypted from the Paillier ciphertext, we need to set $l_a > l_b$, and $l_\alpha > l_\beta$. Under the circumstance, we have $\mathsf{Len}(m_i) \approx \zeta$ if $y_i < 0$ and $\mathsf{Len}(m_i) << \zeta$ if $y_i \geq 0$, where $\mathsf{Len}(m_i)$ denotes the bit length of m_i. In addition, the proposed protocol correctly calculates y_i in ciphertext by using the homomorphic property of Paillier cryptosystem, e.g., $\mathcal{E}_i$ is the ciphertext of $y_i + r_i$, which can be easily verified.

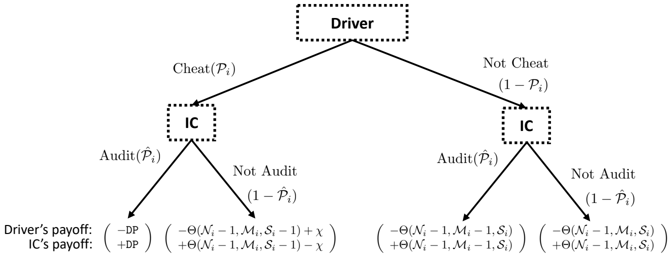

Driver
Cheat($\mathcal{P}_i$)
Not Cheat
($1-\mathcal{P}_i$)
IC
IC
Audit($\hat{\mathcal{P}}_i$)
Not Audit
($1-\hat{\mathcal{P}}_i$)
Audit($\hat{\mathcal{P}}_i$)
Not Audit
($1-\hat{\mathcal{P}}_i$)
Driver's payoff:
IC's payoff:
$\begin{pmatrix} -\mathtt{DP} \\ +\mathtt{DP} \end{pmatrix}$
$\begin{pmatrix} -\Theta(\mathcal{N}_i-1,\mathcal{M}_i,\mathcal{S}_i-1)+\chi \\ +\Theta(\mathcal{N}_i-1,\mathcal{M}_i,\mathcal{S}_i-1)-\chi \end{pmatrix}$
$\begin{pmatrix} -\Theta(\mathcal{N}_i-1,\mathcal{M}_i-1,\mathcal{S}_i) \\ +\Theta(\mathcal{N}_i-1,\mathcal{M}_i-1,\mathcal{S}_i) \end{pmatrix}$
$\begin{pmatrix} -\Theta(\mathcal{N}_i-1,\mathcal{M}_i,\mathcal{S}_i) \\ +\Theta(\mathcal{N}_i-1,\mathcal{M}_i,\mathcal{S}_i) \end{pmatrix}$

Fig. 8. The sequential game of determining whether to cheat (Driver) or audit (IC) on trip i (for $i = 1$ to $\mathcal{N}$)

C. Contract-based Auditing

After a driver's personalized auto premium is calculated, IC should determine whether to audit the driving data uploaded by the driver, and we design a new auditing method based on a recursive inspection game [27]. The auditing process can be viewed as a sequential game between a driver and IC. Specifically, since the number of total uploaded trips is $\mathcal{N}$, we require IC can only audit $\mathcal{M}$ trips and try to minimize $\mathcal{M}$ to achieve our goal.

We can describe the game between IC and a driver in extensive form with imperfect information, which is a game tree as shown in Figure 8. The game is a non-cooperative zero-sum sequential game that involves $\mathcal{N}$ stages (corresponding to $\mathcal{N}$ trips uploaded by the driver, $i = 1$ to $\mathcal{N}$), and the game tree shows the actions of IC and the driver and the payoffs of IC and the driver. For trip i, the driver may cheat with the probability, $\mathcal{P}_i \in [0, 1]$, and may not cheat with $(1 - \mathcal{P}_i)$. IC may choose to audit the trip with the probability, $\hat{\mathcal{P}}_i \in [0, 1]$, and may not audit with $(1 - \hat{\mathcal{P}}_i)$. $\mathcal{N}_i$, $\mathcal{M}_i$, and $\mathcal{S}_i$ are three variables that indicates the number of needed trips, permitted auditing actions, and the number of intended cheats before trip i's data is uploaded, i.e., $\mathcal{N}_1 = \mathcal{N}$, $\mathcal{M}_1 = \mathcal{M}$, and $\mathcal{S}_1 = \mathcal{S} \in [0, \mathcal{N} - \mathcal{M}]$. For the most cautious (malicious) drivers, $\mathcal{S} = 1$ and for the most aggressive (malicious) drivers, $\mathcal{S} = \mathcal{N} - \mathcal{M}$. The payoff of IC before the driver uploads the driving data of trip i can be denoted as a function, $\Theta(\mathcal{N}_i, \mathcal{M}_i, \mathcal{S}_i)$. If trip i is cheated and is audited, according to Figure 4, the game will end immediately and the deposit of the driver, $\mathtt{DP}$, will be transferred to IC. If the driver successfully makes a cheat without being audited at trip i, the payoff of IC is updated as $\Theta(\mathcal{N}_i - 1, \mathcal{M}_i, \mathcal{S}_i - 1) - \chi$, where χ is the reward of each cheat behavior. In our model, $\chi = 25\% * \frac{1}{\mathcal{N}} * Q + 20\% * \frac{1}{\mathcal{N}} * Q$ and $\mathtt{DP} = \eta \cdot \chi$, where η is a positive number determined by IC. When the driver does not cheat at trip i, if IC makes an audit, the payoff is $\Theta(\mathcal{N}_i - 1, \mathcal{M}_i - 1, \mathcal{S}_i)$; otherwise, the payoff is $\Theta(\mathcal{N}_i - 1, \mathcal{M}_i, \mathcal{S}_i)$. Since the game is zero-sum, the payoff of the driver is the opposite. For concise expression of equations in the following calculations, we denote $\Theta(\mathcal{N}_i - 1, \mathcal{M}_i - 1, \mathcal{S}_i)$ by $\mu_{i,1}$, denote $\Theta(\mathcal{N}_i - 1, \mathcal{M}_i, \mathcal{S}_i - 1)$ by $\mu_{i,2}$, and denote $\Theta(\mathcal{N}_i - 1, \mathcal{M}_i, \mathcal{S}_i)$ by $\mu_{i,3}$.

Based on the adversarial model, the game can lead to a unique mixed Nash equilibrium where both players choose their actions with fixed positive probabilities. For the stage that the driver uploads the driving data of trip i, the equilibrium choice of $\hat{\mathcal{P}}_i$ is to make the driver's payoff indifferent no matter whether the driver chooses to cheat or not, i.e.,

$$\begin{aligned} &\hat{\mathcal{P}}_i \cdot (-\mu_{i,1}) + (1 - \hat{\mathcal{P}}_i) \cdot (-\mu_{i,3}) \\ &\quad = \hat{\mathcal{P}}_i \cdot (-\mathtt{DP}) + (1 - \hat{\mathcal{P}}_i) \cdot (-\mu_{i,2} + \chi); \end{aligned} \tag{1}$$

similarly, the equilibrium choice of $\mathcal{P}_i$ is to make IC's payoff indifferent no matter whether IC chooses to audit or not, that is,

$$\begin{aligned} &\mathcal{P}_i \cdot (\mu_{i,2} - \chi) + (1 - \mathcal{P}_i) \cdot \mu_{i,3} \\ &\quad = \mathcal{P}_i \cdot \mathtt{DP} + (1 - \mathcal{P}_i) \cdot \mu_{i,1}. \end{aligned} \tag{2}$$

Based on two equations above, we can calculate the probability of auditing as

$$\begin{aligned} \hat{\mathcal{P}}_i &= \frac{\mu_{i,3} - \mu_{i,2} + \chi}{\mu_{i,3} - \mu_{i,2} + \chi + \mathtt{DP} - \mu_{i,1}} \\ &= f'(\mathcal{N}_i, \mathcal{M}_i, \mathcal{S}_i, \chi, \mathtt{DP}), \end{aligned} \tag{3}$$

and

$$\begin{aligned} \mathcal{P}_i &= \frac{\mu_{i,3} - \mu_{i,1}}{\mu_{i,3} - \mu_{i,2} + \chi + \mathtt{DP} - \mu_{i,1}} \\ &= f(\mathcal{N}_i, \mathcal{M}_i, \mathcal{S}_i, \chi, \mathtt{DP}). \end{aligned} \tag{4}$$

Both sides of Equation 2 denote IC's payoff $\Theta(\mathcal{N}_i, \mathcal{M}_i, \mathcal{S}_i)$, and we substitute $\mathcal{P}_i$ to obtain

$$\Theta(\mathcal{N}_i, \mathcal{M}_i, \mathcal{S}_i) = \frac{\mathtt{DP} \cdot \mu_{i,3} + \chi \cdot \mu_{i,1} - \mu_{i,1}\mu_{i,2}}{\mu_{i,3} - \mu_{i,2} + \chi + \mathtt{DP} - \mu_{i,1}}. \tag{5}$$

Based on Theorem 1 in [27], for $\mathcal{N}_i > \mathcal{M}_i > 0$ and $\mathcal{S}_i > 0$, we can get

$$\Theta(\mathcal{N}_i, \mathcal{M}_i, \mathcal{S}_i) = \frac{-\sum_{j=1}^{\mathcal{S}_i} \chi \cdot \binom{\mathcal{N}_i - j}{\mathcal{M}_i}}{\sum_{j=0}^{\mathcal{M}_i} \mathtt{DP}^{\mathcal{M}_i - j} \cdot \binom{\mathcal{N}_i}{j}}. \tag{6}$$

The objective of our auditing method is to adjust $\mathcal{M}$ and $\mathtt{DP}$ and find the appropriate $\mathcal{M}$ and $\mathtt{DP}$ that minimizes $\mathcal{P}_i$. Equation 4 shows that $\mathcal{P}_i$ is function $f()$ of $\mathcal{N}_i$, $\mathcal{M}_i$, $\mathcal{S}_i$, χ, and $\mathtt{DP}$, i.e., $\mathcal{P}_1 = f(\mathcal{N}_1, \mathcal{M}_1, \mathcal{S}_1, \chi, \mathtt{DP})$, if we substitute $\mu_{i,1}$, $\mu_{i,2}$, and $\mu_{i,3}$ with $\Theta(\mathcal{N}_i - 1, \mathcal{M}_i - 1, \mathcal{S}_i)$, $\Theta(\mathcal{N}_i - 1, \mathcal{M}_i, \mathcal{S}_i - 1)$, and $\Theta(\mathcal{N}_i - 1, \mathcal{M}_i, \mathcal{S}_i)$ based on Equation 6. Obviously, the function is monotonic with $\mathcal{M}_i$ and $\mathtt{DP}$, and thus the minimum of $\mathcal{P}_i$ means the maximum of $\mathcal{M}_i$ and $\mathtt{DP}$.

$\hat{\mathcal{P}}_i$ is a function $f'()$ of $\mathcal{N}_i$, $\mathcal{M}_i$, $\mathcal{S}_i$, χ, and $\mathtt{DP}$ based on Equation 3 ($\hat{\mathcal{P}}_i = 0$ if $\mathcal{M}_i = 0$), i.e., $\hat{\mathcal{P}}_i = f'(\mathcal{N}_i, \mathcal{M}_i, \mathcal{S}_i, \chi, \mathtt{DP})$, and IC can create a randomness algorithm $\mathsf{Random}_i(\hat{\mathcal{P}}_i)$ that returns 1 with $\hat{\mathcal{P}}_i$ and 0 otherwise. IC then follows Algorithm 1 to generate its auditing strategies based on probabilities. Although IC does not have the information about the left number of the malicious driver's attempts, $\mathcal{S}$, to report fabricated driving data, IC can make a reasonable assumption that a malicious driver will set $\mathcal{S}_i \in [0, \mathcal{N}_i - \mathcal{M}_i]$, because the malicious behavior will be detected if $\mathcal{S}_i > \mathcal{N}_i - \mathcal{M}_i$. Therefore, IC can set $\mathcal{S} = \mathcal{N} - \mathcal{M}$ in the real implementation. The algorithm finally outputs $\{\overline{\mathtt{NUM}}_j\}_{j=1}^{\mathcal{M}}$. If $\overline{\mathtt{NUM}}_j = 0$ for $j = 1$ to $\mathcal{M}$, IC sends 'confirm' to trigger function ***Confirm*** without auditing; otherwise, IC sends $\{\overline{\mathtt{NUM}}_j\}_{j=1}^{\mathcal{M}}$ to trigger function ***Audit***. According to the contract, the driver needs to authorize the auditing requests by sending $(\{\overline{\mathtt{NUM}}'_j\}_{j=1}^{\mathcal{M}}, \{\mathtt{DK}_j = \mathsf{PEnc}(\widehat{\mathtt{pk}}, k_{\overline{\mathtt{NUM}}'_j})\}_{j=1}^{\mathcal{M}})$, and $\pi_{\mathtt{DK}}$,

Algorithm 1 AuditSelection

```
Input: $\mathcal{N}$, $\mathcal{M}$, $\mathcal{S}$, $\chi$, $\mathtt{DP}$;
Output: $\{\overline{\mathtt{NUM}}_j\}_{j=1}^{\mathcal{M}}$;
$\mathcal{N}_1 = \mathcal{N}$, $\mathcal{M}_1 = \mathcal{M}$, $\mathcal{S}_1 = \mathcal{S}$;
for $j = 1 \rightarrow \mathcal{M}$ do
    $\overline{\mathtt{NUM}}_j = 0$;
end for
$j = 1$;
for $i = 1 \rightarrow \mathcal{N}$ do
    $\hat{\mathcal{P}}_1 = f'(\mathcal{N}_i, \mathcal{M}_i, \mathcal{S}_i, \chi, \mathtt{DP})$;
    if $1 == \mathsf{Random}_i(\hat{\mathcal{P}}_i)$ then
        $\overline{\mathtt{NUM}}_j = i$, $j = j + 1$, $\mathcal{M}_{i+1} = \mathcal{M}_i - 1$;
    else if $0 == \mathsf{Random}_i(\hat{\mathcal{P}}_i)$ then
        $\mathcal{M}_{i+1} = \mathcal{M}_i$;
    end if
    $\mathcal{N}_{i+1} = \mathcal{N}_i - 1$;
    if $\mathcal{S}_i > \mathcal{N}_{i+1} - \mathcal{M}_{i+1}$ then
        $\mathcal{S}_{i+1} = \mathcal{N}_{i+1} - \mathcal{M}_{i+1}$
    else if $\mathcal{S}_i \leq \mathcal{N}_{i+1} - \mathcal{M}_{i+1}$ then
        $\mathcal{S}_{i+1} = \mathcal{S}_i$
    end if
    if $\mathcal{M} == j$ then
        break;
    end if
end for
Output $\{\overline{\mathtt{NUM}}_j\}_{j=1}^{\mathcal{M}}$;
```

$$\pi_{\mathtt{DK}} \leftarrow \mathsf{ZkPoK}\Big\{(\{k_{\overline{\mathtt{NUM}}_j}\}_{j=1}^{\mathcal{M}}) : \{\mathtt{DK}_j = \mathsf{PEnc}(\widehat{\mathtt{pk}}, k_{\overline{\mathtt{NUM}}_j})\}_{j=1}^{\mathcal{M}}\Big\}.$$

Fig. 9. Proof of the correctness of encrypted authorization keys that can be verified by Verify_3 algorithm

as shown in Figure 9, by triggering ***Authorize***. Then, TPA can decrypt $\{\mathtt{ct}_{\overline{\mathtt{NUM}}_j}\}_{j=1}^{\mathcal{M}}$ to obtain detailed driving data of trips, audit the data, and report the auditing result to end the contract by triggering ***Inspect***. The whole auditing procedures are public and authorized by drivers and IC such that permission-based audit can be achieved.

V. Security Analysis

A. Simulation-based Security Analysis

We use a simulation-based approach to capture the security notions of our proposed scheme, including driver privacy, model privacy, and verifiability, and transparency [28]. The idea is to prove that, a real world where drivers, DMs, IC, and TPA execute the proposed scheme, is computationally indistinguishable from an ideal world where the same players achieve the same functions relying on a trusted party, $\mathcal{T}$. In other words, the ideal world is built that fulfills all desirable security properties, and we also construct a simulator as an ideal-world adversary, $\mathcal{S}$, who externally communicates with $\mathcal{T}$ and honest players in the ideal world, and also pretends to be honest players and internally executes the proposed scheme with a real-world adversary, $\mathcal{A}$. If $\mathcal{A}$ cannot computationally distinguish the inputs and the outputs of the "simulated" world and the real world, it indicates that $\mathcal{A}$ obtains no information except the information leaked in the ideal world. In our proof, we consider a static model where malicious drivers, IC, and DMs are fixed.

Four main phases in our proposed scheme require private inputs from drivers and IC according to Figure 4, and these inputs affect the outputs. Thus, we focus on constructing the ideal functionalities of these four phases in the ideal world by relying on $\mathcal{T}$: 1) setup phase; 2) recording phase; 3) evaluation phase; and 4) authorization phase:

- Phase-1: *(Real world)* IC generates and publishes ($\vec{E}$, $\vec{C}$, π_{params}). *(Ideal world)* IC needs to submit $\vec{w}$ and ϵ to $\mathcal{T}$, and $\mathcal{T}$ verifies the consistency of the model parameters;
- Phase-2: *(Real world)* Drivers upload $(\mathtt{DD}_i, \pi_{\mathtt{DD}_i}, \mathtt{DA}_i)$ to the blockchain and the blockchain accepts or rejects the report. (Ideal world) Drivers submit $\vec{x_i}$ to $\mathcal{T}$, and $\mathcal{T}$ can judge the correctness of $\vec{x_i}$ with acceptance or rejection.
- Phase-3: *(Real world)* IC submits $(\{\mathtt{DS}_i\}_{i=1}^{\mathcal{N}}, \{\pi_{\mathtt{DS}_i}\}_{i=1}^{\mathcal{N}})$ to the blockchain and the blockchain calculates and publishes R and $\hat{Q}$. *(Ideal world)* As $\mathcal{T}$ knows all information, IC only needs to trigger $\mathcal{T}$ and $\mathcal{T}$ calculates R and $\hat{Q}$ in plaintext and publish how R and $\hat{Q}$ are calculated for each trip.
- Phase-4: *(Real world)* Drivers submit $\{\mathtt{DK}_j\}_{i=1}^{\mathcal{M}}$ and $\pi_{\mathtt{DK}}$ to the blockchain, and TPA uses $\{\mathtt{DK}_i\}_{i=1}^{\mathcal{M}}$ to audit the driving data and submit the auditing result to the blockchain. *(Ideal World)* Drivers submits $\{\text{info}_{\overline{\mathtt{NUM}}'_j}\}_{j=1}^{\mathcal{M}}$ to $\mathcal{T}$ who can audit and publish the auditing result.

According to the ideal functionalities, we can easily conclude that the ideal world provides all desirable security properties. Assuming that D is a probabilistic polynomial-time (PPT) algorithm used for distinguishing the ideal world and the real world controlled by $\mathcal{A}$ (D outputs 1 when running in the real world), $\mathtt{Real}_{\mathcal{A}}$ and $\mathtt{Ideal}_{\mathcal{S}}$ are the views (inputs and outputs) of the real execution of our proposed scheme and the "simulated" execution, respectively, we can have the following definition of security for our scheme:

Definition 1 (Security): Our proposed scheme is secure if for all PPT algorithms D, the following expression holds:

$$|Pr[D(\mathtt{Real}_{\mathcal{A}}) = 1] - Pr[D(\mathtt{Ideal}_{\mathcal{S}}) = 1]| = \mathtt{negl}(\tau, \zeta),$$

where $Pr[\cdot]$ is the probability function and $\mathtt{negl}(\tau, \zeta)$ denotes a negligible function in security parameters (τ, ζ).

We analyze the security of our proposed scheme by proving the above equation holds when the following conditions are satisfied: 1) ZkPoK is simulation-extractable zero-knowledge proof of knowledge in the random oracle model; 2) FO commitment is statistically hiding and computationally binding; 3) Paillier cryptosystem is semantic secure; 4) $\mathsf{PEnc}(\cdot)/\mathsf{PDec}(\cdot)$ is a semantic secure public-key cryptosystem; 5) $\mathsf{SEnc}(\cdot)/\mathsf{SDec}(\cdot)$ is a secure symmetric-key cryptosystem; and 6) $l_r > \kappa$, $l_b > \kappa$, and $l_\beta > l_a + \kappa$.

The security proofs are categorized into two cases since different players are controlled by $\mathcal{A}$. In case-1, $\mathcal{A}$ controls drivers and a subset of DMs, and intend to compromise model privacy. In case-2, $\mathcal{A}$ controls IC and a subset of DMs, whose target is driver privacy. We sketch how $\mathcal{S}$ can be constructed in two cases.

(***Case-1***) $\mathcal{S}$ internally simulates the proposed scheme by representing honest IC to $\mathcal{A}$, and externally communicates with $\mathcal{T}$ in the ideal world as malicious drivers:

- Phase-1: $\mathcal{S}$ randomly chooses $n+1$ model parameters $(\vec{w}' = (w_1', w_2', ..., w_n'), \epsilon')$ which satisfies $\psi(\vec{w}', \epsilon') = 1$, and encrypts $\vec{w}'$ and ϵ' as $\vec{E}' = (E_1', E_2', ..., E_n', E_{n+1}')$ where E_i' is a valid Paillier ciphertext using N. Similarly, $\mathcal{S}$ can construct $\vec{C}' = (C_1', C_2', ..., C_n', C_{n+1}')$ where C_i' is a valid FO commitment. $\mathcal{S}$ can use the zero-knowledge simulator to simulate the proof, π'_{Params}, and publishes $(\vec{E}', \vec{C}', \pi'_{\text{Params}})$ internally to $\mathcal{A}$;
- Phase-2: By interacting $\mathcal{A}$ internally, $\mathcal{S}$ stores $\mathtt{ct}_i$ and extracts $\vec{x_i}$ using the knowledge extractor of the proof $\pi_{\mathtt{DD}_i}$, and submits $\vec{x_i}$ to $\mathcal{T}$ for $i = 1$ to $\mathcal{N}$ in the ideal world;
- Phase-3: $\mathcal{S}$ retrieves the evaluation results of personalized auto insurance premium $(R', \hat{Q}')$ from $\mathcal{T}$ in the ideal world, and accordingly construct $\mathtt{DS}_i'$. For trip i, if $y_i \geq 0$, $\mathcal{S}$ chooses a random number γ_i whose bit length is chosen uniformly between $(l_\alpha + l_a, l_\alpha + l_a + \kappa)$ and sets $m_i' = \gamma_i$; otherwise, $m_i' = -\gamma_i \bmod N$. $\mathcal{S}$ then randomly chooses $\alpha_i' \in (2^{l_\alpha - 1}, 2^{l_\alpha} - 1]$, $\beta_i' \in (2^{l_\beta - 1}, 2^{l_\beta} - 1]$, and $\Gamma_i' \in (0, N)$, and generates Paillier ciphertexts $\mathfrak{E}_i', \mathcal{U}_i', \mathcal{D}_i'$ based on $(\alpha_i', \beta_i', m_i', \Gamma_i')$. $\mathcal{S}$ sets $\mathtt{DS}_i' = (m_i', \mathfrak{E}_i', \mathcal{U}_i', \mathcal{D}_i')$ and uses the zero-knowledge simulator to simulate the proof, $\pi'_{\mathtt{DS}_i}$ for $i = 1$ to $\mathcal{N}$;
- Phase-4: $\mathcal{S}$ extracts k_i using the knowledge extractor of the proof $\pi_{\mathtt{DK}}$, and use $k_{\overline{\mathtt{NUM}'_j}}$ to decrypt $\mathtt{ct}_{\overline{\mathtt{NUM}'_j}}$ to recover $\text{info}_{\overline{\mathtt{NUM}'_j}}$ for $j = 1$ to $\mathcal{M}$.

Analysis of Case-1: For phase-1, if D can distinguish $\mathtt{Real}_{\mathcal{A}} = (\vec{E}, \vec{C}, \pi_{\text{Params}})$ and $\mathtt{Ideal}_{\mathcal{S}} = (\vec{E}', \vec{C}', \pi'_{\text{Params}})$, it means D can be used to break the semantic security of Paillier cryptosystem, the statistically hiding property of the FO commitment, and the zero-knowledge property of ZkPoK, which happens with negligible probability. For phase-2 and phase-3, if D can distinguish $\mathtt{Real}_{\mathcal{A}} = (\{\mathtt{DS}_i\}_{i=1}^{\mathcal{N}}, \{\pi_{\mathtt{DS}_i}\}_{i=1}^{\mathcal{N}}, R, \hat{Q})$ and $\mathtt{Ideal}_{\mathcal{S}} = (\{\mathtt{DS}_i'\}_{i=1}^{\mathcal{N}}, \{\pi'_{\mathtt{DS}_i}\}_{i=1}^{\mathcal{N}}, R', \hat{Q}')$, it means $\mathcal{S}$ not only breaks the semantic security of Paillier cryptosystem but also fails to extract $\vec{x_i}$ and simulate $\{\pi'_{\mathtt{DS}_i}\}_{i=1}^{\mathcal{N}}$, which happens with negligible probability under the soundness property and the zero-knowledge property of ZkPoK. For phase-4, the simulation of $\mathcal{S}$ is perfect as long as the soundness property is achieved in the proof, $\pi_{\mathtt{DK}}$, since the auditing result, $\mathtt{RES}$ is fully determined by $\{\text{info}_{\overline{\mathtt{NUM}'_j}}\}_{j=1}^{\mathcal{M}}$ when $\{k_{\overline{\mathtt{NUM}'_j}}\}_{j=1}^{\mathcal{M}}$ can be correctly extracted.

(***Case-2***) $\mathcal{S}$ internally simulates the proposed scheme by representing honest drivers to $\mathcal{A}$, and externally communicates with $\mathcal{T}$ in the ideal world as malicious IC:

- Phase-1: $\mathcal{S}$ uses the knowledge extractor of the proof, π_{Params}, to extract $\vec{w}$ and ϵ by interacting $\mathcal{A}$ internally, and submits $\vec{w}$ and ϵ to $\mathcal{T}$ in the ideal world;
- Phase-2: In the ideal world, $\mathcal{S}$ triggers $\mathcal{T}$ and retrieves the personalized auto insurance premiums $(\tilde{R}, \tilde{Q})$ from $\mathcal{T}$, and then generates $\vec{x_i'} = (x_{i,1}', x_{i,2}', ..., x_{i,n}')$ according to $y_i = \vec{w} \cdot \vec{x_i'} + \epsilon$. Specifically, if $y_i \geq 0$, y_i is chosen uniformly from $[0, 2^\kappa - 1]$; otherwise, y_i is chosen from $[-2^\kappa + 1, 0)$, and $\mathcal{S}$ can calculate $\vec{x_i'}$ accordingly. $\mathcal{S}$ then chooses random numbers $\tilde{r}_i \in (2^{l_r - 1}, 2^{l_r} - 1]$, $\tilde{a}_i \in (2^{l_a - 1}, 2^{l_a} - 1]$, $\tilde{b}_i \in (2^{l_b - 1}, 2^{l_b} - 1]$, $\tilde{v}_i \in [0, 2^\tau N)$ and sets $\tilde{r}_i' = \tilde{a}_i \tilde{r}_i$ and $\tilde{v}_i' = \tilde{a}_i \tilde{v}_i$. Based on the inputs of $\mathcal{A}$ $(\vec{E})$, $\mathcal{S}$ can create Paillier ciphertexts and FO commitments, $\tilde{\mathtt{DD}}_i = (\tilde{\mathcal{E}}_i, \tilde{\mathcal{E}}_i', \tilde{\mathtt{com}}_i, \tilde{\mathtt{com}}_i')$, as follows:

$$\tilde{\mathcal{E}}_i = \prod_{j=1}^{n} E_j^{x'_{i,j}} \cdot E_{n+1} \cdot (1+N)^{\tilde{r}_i},\ \tilde{\mathtt{com}}_i = g^{\tilde{r}_i} h^{\tilde{v}_i},$$
$$\tilde{\mathcal{E}}_i' = \tilde{\mathcal{E}}_i^{a_i} \cdot (1+N)^{-r_i'} \cdot (1+N)^{b_i},\ \tilde{\mathtt{com}}_i' = g^{\tilde{r}_i'} h^{\tilde{v}_i'}.$$

  Based on $\tilde{\mathtt{DD}}_i$, $\mathcal{S}$ uses the zero-knowledge simulator to simulate the proof, $\tilde{\pi}_{\tilde{\mathtt{DD}}_i}$, chooses a random permutation, $\tilde{ct}_i$, and sets $\tilde{\mathtt{DA}}_i = \tilde{ct}_i$;
- Phase-3: There is no need for $\mathcal{S}$ to simulate the phase since $\mathcal{S}$ has no input nor output in this phase in the real world;
- Phase-4: $\mathcal{S}$ randomly chooses symmetric keys $\{\tilde{k}_{\overline{\mathtt{NUM}'_j}}\}_{j=1}^{\mathcal{M}}$, and uses the public key cryptosystem to generate $\{\tilde{\mathtt{DK}}_j = \mathsf{PEnc}(\tilde{k}_{pk, \overline{\mathtt{NUM}'_j}})\}_{j=1}^{\mathcal{M}}$. Next, $\mathcal{S}$ uses the zero-knowledge simulator to simulate the proof, $\tilde{\pi}_{\tilde{\mathtt{DK}}}$.

Analysis of Case-2: For phase-1, the probability that $\mathcal{S}$ fails to extract $\vec{w}$ and ϵ is negligible according to the soundness property of ZkPoK. For phase-2, if D can distinguish $\mathtt{Real}_{\mathcal{A}} = (\mathtt{DD}_i, \pi_{\mathtt{DD}_i}, \mathtt{DA}_i, R, \hat{Q})$ and $\mathtt{Ideal}_{\mathcal{S}} = (\tilde{\mathtt{DD}}_i, \tilde{\pi}_{\tilde{\mathtt{DD}}_i}, \tilde{\mathtt{DA}}_i, \tilde{R}, \tilde{Q})$, it means that the semantic security of Paillier cryptosystem can be broken, the FO commitments are not statistically hiding, and ZkPoK is the zero-knowledge, which happens with negligible probability. For phase-4, if D can distinguish $\mathtt{Real}_{\mathcal{A}} = (\{k_{\overline{\mathtt{NUM}'_j}}\}_{j=1}^{\mathcal{M}}, \{\mathtt{DK}_j\}_{j=1}^{\mathcal{M}}, \pi_{\mathtt{DK}})$ and $\mathtt{Ideal}_{\mathcal{S}} = (\{\tilde{k}_{\overline{\mathtt{NUM}'_j}}\}_{j=1}^{\mathcal{M}}, \{\tilde{\mathtt{DK}}_j\}_{j=1}^{\mathcal{M}}, \tilde{\pi}_{\tilde{\mathtt{DK}}})$, we can use D to break the semantic security of the public key cryptosystem and the symmetric key cryptosystem used in our proposed scheme, and also the probability that $\mathcal{S}$ fails to simulate $\tilde{\pi}_{\tilde{\mathtt{DK}}}$ is negligible according to the zero-knowledge property of ZkPoK.

As a consequence, we can conclude that the views of the ideal world, $\mathtt{Ideal}_{\mathcal{S}}$, and the views of the real world, $\mathtt{Real}_{\mathcal{A}}$ are computationally indistinguishable in two cases, and completes the security proof.

B. Analysis of Fraud Resistance

Whether a driver chooses to cheat or not depends on the payoff received after cheating. According to Equation 6, if IC increases $\mathtt{DP}$ and $\mathcal{M}$, the average payoff is reduced. From Figure 10, we can see that when $\mathtt{DP} = 1500$ and $\mathcal{M} = 3$, the payoff $\Theta \approx 0$ indicating the driver can gain approximate zero from the cheating and has no motivation to cheat in our proposed scheme.

We also analyze the situation from a different angle: how to ensure a driver does not cheat in all $\mathcal{N}$ reports based on Equation 4. As the driver only knows $\mathcal{M}_1 = \mathcal{M}$ and has no information about $\mathcal{M}_i (1 < i \leq \mathcal{N})$, the driver is believed to set $\mathcal{S}_i = \mathcal{N}_i - \mathcal{M}$ (as the most aggressive attacker), and we

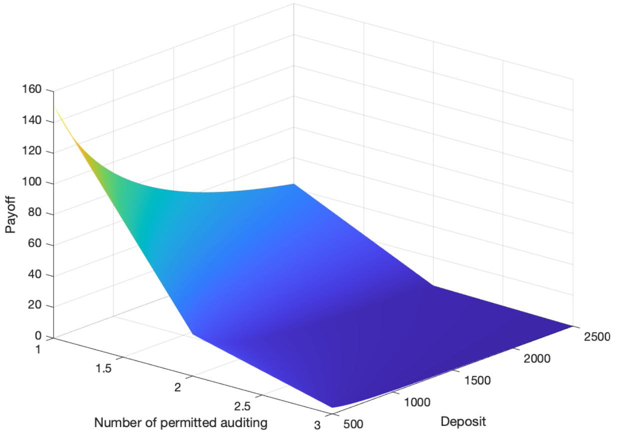


Fig. 10. The payoff of a malicious driver ($\mathcal{N} = 300$, $\mathcal{S} = \mathcal{N} - \mathcal{M}$, $Q = 1800$, and $\chi = 2.7$)

can calculate the probability, $\tilde{\mathcal{P}}$, that the driver never chooses to cheat in all $\mathcal{N}$ reports as follows:

$$\tilde{\mathcal{P}} = \prod_{k=0}^{\mathcal{N}-\mathcal{M}} (1 - f(\mathcal{N} - k, \mathcal{M}, \mathcal{N} - k - \mathcal{M}, \chi, \mathtt{DP})) \qquad (7)$$

The probability, $\tilde{\mathcal{P}}$, is affected by $\mathcal{N}$, $\mathcal{M}$, χ, and DP. The number of permitted auditing, $\mathcal{M}$, should be a relatively small value compared to $\mathcal{N}$ and DP should be a reasonable value that is not much larger than the driver's basic auto insurance premium, Q. Under the circumstance, $\tilde{\mathcal{P}}$-fraud resistance can be defined as follows:

Definition 2 ($\tilde{\mathcal{P}}$-fraud resistance): Given $\mathcal{N}$, $\mathcal{M}$, χ, and DP*, the probability that a driver does not cheat in all $\mathcal{N}$ reports of driving data is equal to $\tilde{\mathcal{P}}$.*

For easy understanding, an instance is set where $\mathcal{N} = 300$, $\mathcal{M} = 1, 2, 3$, $S = \mathcal{N} - \mathcal{M}$, and $Q = 2400$. We set the different deposits $\mathtt{DP} = (50\% * Q, 100\% * Q, 100\% * Q)$, and draw the varies of the probability that the driver does not cheat at trip $i = 1$ to $\mathcal{N}$, the varies of the probability that the driver does not cheat until trip $i = 1$ to $\mathcal{N}$, and the mean number of cheats in Figure 11 (a), (b), and (c), respectively. From Figure 11 (a) and (b), we can see that appropriate parameter selections are extremely important, with a few deposits and permitted auditing numbers, the chance that a driver chooses to cheat is largely increased. From Figure 11 (c), we can see that the permitted auditing number, $\mathcal{M}$, is a dominant parameter that influences a malicious driver's behavior compared to the deposit, DP. It can be found that when $\mathcal{N} = 150\% * Q$ and $\mathcal{M} = 3$, $\tilde{\mathcal{P}} \geq 97\%$, which is a reasonable setting, i.e., the driver never cheats with the probability of 97% in uploading the driving data of all 300 trips. In other words, our proposed scheme only requires TPA to audit at most 1% driving data to ensure that drivers will behave honestly with the probability 97% when uploading their driving data.

VI. Performance Evaluation

A. Real-world Dataset

We use a public real-world dataset published by Grab[2] for measuring driving safety and personalized car insurance, which includes around $20,000$ real-world trips. Each trip record has the telematics attributes such as GPS accuracy, GPS bearing, acceleration data, gyroscope data, timestamp, speed, and a label indicating whether the trip is safe or not.

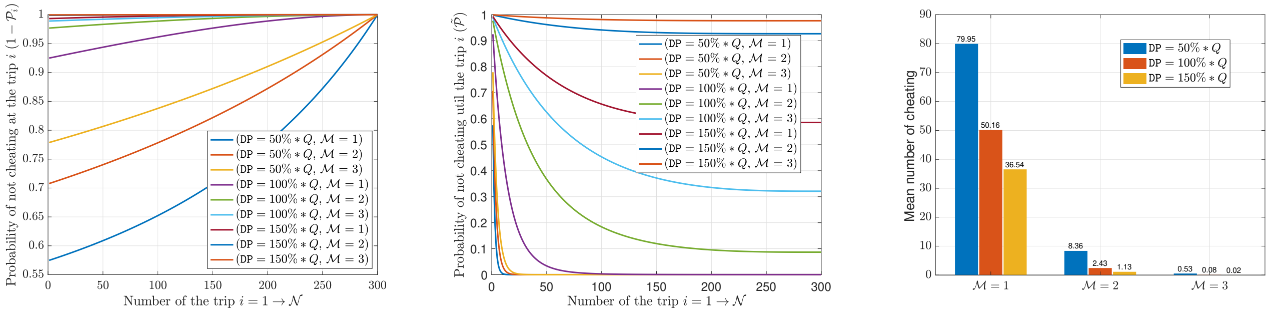
We perform some features engineering techniques on the raw data. Specifically, we clean the dataset by removing the unrealistic records where the speed is larger than 400km/h, and extract additional features of a trip including the mean, max, and min values of acceleration, speed, and gyroscope, etc. We choose mean, max, and min values because these values capture the driving behavior of a driver, e.g., preferring a sudden brake. After feature engineering, we obtain a new dataset of extracted features, and the dimension of the features is 34 ($n = 34$).

B. Proof-of-concept Implementation and Experiments

In order to evaluate the performance of the proposed scheme, we implement a proof-of-concept prototype using Java and built our implementation on the testnet of an open-source consortium blockchain, Hyperledger Fabric [29]. We have a smartphone (OS: Android, CPU: Krin 980, RAM: 6G) as the client and a laptop (OS: MAC OS, CPU: Dual-Core Intel Core i7, RAM: 16GB) as the server which simulates the testnet of the consortium blockchain and simulates the server of the auto insurance company. The blockchain adopts the default setting of Hyperledger Fabric v2.1: two peer nodes belonging to two organizations and one order node with the RAFT consensus protocol, and they are connected in a local network. We develop a Java library which realizes function modules defined in our proposed scheme, including the driving feature extraction module, the encryption/decryption module, the evaluation results proof/verification module, etc. The developed Java library is integrated with Hyperledger Fabric SDKs (Java) to construct the smartphone client, the server of IC, and the contract deployed in the blockchain. The source codes can be found at https://github.com/EnderCheng/PCI.

Since the applications of personalized car insurance do not require to be real-time, the requirements of the computational and the communication efficiency can be relaxed to some extent, and we put more efforts on testing the feasibility of the proposed scheme. Although there are some existing works related to privacy-preserving personalized car insurance as discussed in Section VII, they do not have the same driving safety evaluation model as ours (the LR model) or do not design their schemes under a malicious security model as shown in Table III. That is the reason that we do not compare our scheme with their schemes in terms of computational costs and communication overheads, as the comparison cannot be performed fairly. It is explicit that our scheme achieves a more rigorous security model with flexible functions by sacrificing

[2] https://www.grab.com/sg/aiforsea/aiforseachallenges/



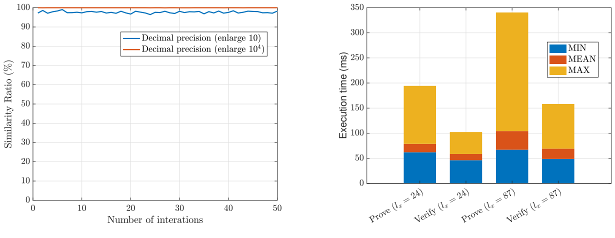
(a) (b) (c)

Fig. 11. Under the setting DP $= (50\% * Q, 100\% * Q, 150\% * Q)$ and $\mathcal{M} = (1, 2, 3)$, we have (a) the probability that the driver does not cheat at trip $i = 1 \to 300$; (b) the probability that the driver does not cheat until trip $i = 1 \to 300$; and (c) the mean number of cheats in all 300 trips

a little efficiency. Therefore, instead of comparing with existing works, we compare the scheme with a plaintext-based personalized car insurance scheme to measure the efficiency loss caused by applying the privacy-preserving techniques. The plaintext-based scheme implies that the driver calculates $y_i = \vec{w} \cdot \vec{x_i} + \epsilon$ and uploads y_i in plaintext to the blockchain and only the detailed driving data are encrypted using a symmetric key cryptosystem.

In our experiments, the security parameters are chosen to fulfill essential security requirements, i.e., $\tau = 256$, $\zeta = 2048$, $l_r = l_a = 300$, $l_b = 250$, $l_\alpha = 600$, and $l_\beta = 350$. The symmetric key cryptosystem is AES-CBC-128, and the public key cryptosystem is EC-Elgamal with BN254 curve. The configuration of the consortium blockchain also affects the throughput and the latency of the system, and we follow the default setting of Hyperledger Fabric v2.1 testnet where a block is generated every 2 seconds and at most 10 transactions are allowed in one block.

System Initialization: It takes around 37 seconds to set up the consortium blockchain on the laptop. Two DMs, one IC, and one TPA, take fewer than 13 seconds to generate public parameters and public/private key pairs. Since the initialization only runs once, it can fulfill the requirements of real-world applications.



(a) (b)

Fig. 12. (a) Similarity ratio of driving safety classification results (with 1 and 4 decimal precision) and (b) computational costs of one range proof ($l_x = 24$ and $l_x = 87$)

TABLE II
THE OFF-CHAIN COSTS FOR DRIVERS AND IC ($n = 34$)

Entity	Driver (Android)	IC (Laptop)
Time-Rep (Plaintext)	112 ms	N/A
Time-Rep (This paper)	3708 ms	N/A
Time-Eval (Plaintext)	N/A	84 ms
Time-Eval (This paper)	N/A	1995 ms

Accuracy: Since the inputs fields of Paillier cryptosystem is $\mathbb{Z}_N$ while the weights, $\vec{w}$, the intercept, ϵ, and a driver's features $\vec{x}$ are floating numbers, we need to magnify the number with the decimal places and transform the number into $\mathbb{Z}_N$ and determine the corresponding l_w, l_ϵ, l_x, and κ, e.g., when one of the weights is 1.043235465, we need to transform the number to $1,043,235,465$ by magnifying 10^9 times if we want to keep the full precision. The times of magnifying also affect the choices of l_w, l_ϵ, l_x, and κ, and the efficiency of the proposed scheme since the range proofs are affected by the bit lengths. For the real-world dataset, the features extracted from the datasets need to be magnified 10^{23} times ($l_x = 87$) to guarantee the full precision, which leads to large computational burdens on the client side. To reduce the costs, we approximate the model parameters and the driver's features and notice that the classification accuracy is hardly affected. As shown in Figure 12 (a), when we maintain the whole integer precision and keep the decimal precision to only 4, the classification results of the original model and the model with approximated parameters and features are slightly different, and around 99.9% results are same. Under the circumstance, the features only need to be enlarged 10^4 times, and we have $2^{l_w} \approx 10^5 (l_w = 17)$, $2^{l_\epsilon} \approx 10^5 (l_\epsilon = 17)$, and $2^{l_x} \approx 10^7 (l_x = 24)$. When the decimal precision is set to 1 ($l_w = 7, l_\epsilon = 7, l_x = 14$), we can still guarantee that around 97% classification results are same. Also, we run a range proof 100 times, and chooses different l_x to evaluate the computational efficiency. From Figure 12 (b), the execution time of a single range proof is less when l_x (similar to l_w and l_ϵ) is smaller, and the trade-off between accuracy and efficiency can be obtained by adjusting l_w, l_ϵ, and l_x. In the following evaluations, we choose $l_w = l_\epsilon = 17$ and $l_x = 24$ to guarantee both efficiency and accuracy.

Computational Costs: The computational costs involve two parts: 1) the off-chain costs at the client side (the driver's smartphone and IC's server) and 2) the on-chain costs. For the off-chain costs, we mainly evaluate the execution time spent by a driver to generate one driving report (Time-Rep), and the

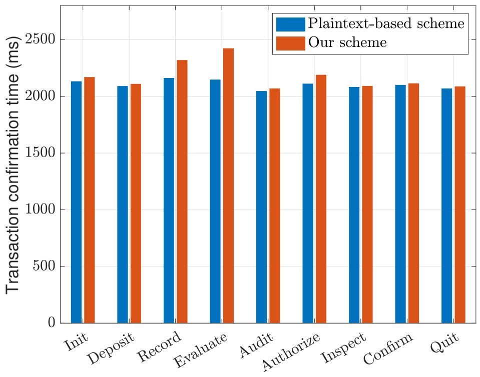


Fig. 13. The on-chain costs of core functions defined in the personalized auto insurance contract

execution time spent by the server to generate one evaluation result based on the report (Time-Eval). The experiment results are the average results with 20 trials and are shown in Table II. Our proposed scheme leads to second-level delays due to the ciphertext operations, compared with the plaintext-based scheme. However, the Android insurance App can be run in the background as services, which does not affect user experience, since the driving report is only generated when the driver is driving. In the meantime, IC's server can be more powerful cloud server with multiple threads to support more drivers. For the on-chain costs, we evaluate the core functions defined in Figure 3. The average experiment results with 20 trials are shown in Figure 13. The y-axis of Figure 13 denotes the transaction confirmation delay of each core function when the function is triggered, and it can be seen that the transaction confirmation delay of the proposed scheme and the plaintext-based scheme are almost the same. The exception is that our proposed scheme has obvious longer confirmation delay when triggering functions *Record* and *Evaluate*. According to our setting, this situation happens because the consortium blockchain needs to wait at least 2 seconds to package a confirmation transaction into a block. With the setting, no matter how efficient the on-chain computations can be, e.g., just taking 5 ms to run on-chain operations, a transaction (that triggers a function) still needs to wait 2 seconds to be confirmed. It is noticed that except for functions *Record* and *Evaluate*, other on-chain operations defined in our scheme can be done in around 2 seconds, which makes the transaction delay is around 2 seconds for these functions (similar to the plaintext-based scheme). Functions *Record* and *Evaluate* require more than 2.5 seconds to run, and thus they have longer transaction confirmation delay. From another perspective, the block generation/confirmation delay in the blockchain cannot be arbitrarily shortened since a short delay may seriously degrade the blockchain throughput. These reasons together lead to the result: our proposed scheme has almost the same on-chain delay as the plaintext-based scheme.

Communication & Storage Overheads: We also evaluate the communication and on-chain storage overheads. We can store the hash values of the data in the blockchain while maintaining the data contents in an off-chain storage such as IPFS. In our experiments, the size of personalized auto insurance contract is 6.5MB (since we package several open-source java libraries). The data uploaded by the driver vary according to the size of the driving data (how many trips), and the data stored in the blockchain also vary accordingly. For a real trip with around 1500 records, the data size uploaded by a driver is around 327KB, and the on-chain storage is hundreds of bytes (i.e., hash values).

VII. Related Works

In this section, we review existing works on securing personalized car insurance (PCI). Popa et al. [30] proposed an anonymous scheme for protecting driver privacy in PCI. In their scheme, drivers' reports are anonymized by using random tags generated from homomorphic commitments, and the data such as locations are bound with random tags. Only a pair of the driver's reports can be linked for counting the number of speed violations to measure the driving safety. To prevent data fraud, an auditable tamper-evident transponders are installed at drivers' vehicles [31]. The similar idea is also adopted by a following work proposed by Troncoso et al., where a black box is deployed at the driver's vehicle [32]. The black box includes sensors such as GPS modules to collect the necessary information and generate statistical data for calculating a driver's insurance premium. Moreover, third parties named policyholders can use USB sticks to physically extract detailed data from the black box, and detect the cheating behavior of the driver.

Instead of being concentrated on privacy protection, Zhou et al. proposed a privacy attack on PCI where drivers' trajectories are recovered based on collected telematics data [33]. They also proposed a privacy-preserving speed data aggregation protocol in which the driving safety can be analyzed based on the sum of the speed data that exceeds a threshold. To prevent data forgery, they assume that there are two kinds of data collected from a vehicle: one from the vehicle's on-board unit and the other from the driver's smartphone, and at least one of the data are unhampered. Based on the assumption, they proposed an auditing method to compare these two data to detect potential adversaries. The above-mentioned three schemes use aggregation methods for calculating driving safety scores, while Rizzo et al. proposed a privacy-preserving driver style recognition protocol based on secure multi-computations, which can evaluate driving safety privately under a decision-tree model [34]. There exists another similar work that applies the K-means model using secure multi-party computations, but since these protocols are designed based on a semi-honest adversarial model, they do not consider the data fraud attack performed by malicious drivers [35].

Recently, blockchain has been introduced into PCI, as blockchain can provide many additional properties for PCI, such as immutable, distributed, and transparent data exchange and management, which brings many advantages [36]–[40]. Accordingly, Wan et al. proposed a blockchain-assisted usage-based insurance scheme which relies on hidden-vector encryption [41]. In their scheme, driving time, speed, acceleration

TABLE III
COMPARISON BETWEEN OUR SCHEME AND EXISTING WORKS

Scheme	Architecture	Driver Privacy	Model Privacy	Adversarial Model	Public Verifiability	Fraud Resistance	Driving Safety Evaluation Model	Permission-based Audit
[30]	Centralized	Data Anonymity	N/A	Malicious	N/A	Tamper-evident Device	Statistic-based Model	N/A
[32]	Centralized	Data Privacy	N/A	Malicious	N/A	Tamper-evident Device	Statistic-based Model	N/A
[33]	Centralized	Data Privacy	N/A	Untrusted[a]	N/A	Probabilistic Data Auditing	Statistic-based Model	N/A
[34]	Centralized	Data Privacy	Parameter Privacy	Semi-honest	N/A	N/A	Decision Tree Model	N/A
[35]	Centralized	Data Privacy	Parameter Privacy	Semi-honest	N/A	N/A	K-means Model	N/A
[41]	Blockchain	Data Privacy	N/A	Untrusted	Support	Accident-based Detection[b]	Statistic-based Model	N/A
[42]	Blockchain	Data Privacy	N/A	Untrusted	Support	Accident-based Detection	Statistic-based Model	N/A
This paper	Blockchain	Data Privacy	Parameter Privacy	Rationally Malicious	Support	Contract-based Fraud Resistance	Logistic Regression Model	Support

[a] An untrusted driver means the driver will follow the scheme but also choose to cheat to reduce insurance fee.
[b] There is an assumption that the data fraud can be easily detected if a traffic accident happens.

data are encoded using binaries and encrypted using the hidden-vector encryption, and the ciphertext are published in the blockchain that supports privacy-preserving comparison, which can be used to measure the driving statistics including the number of speed violations and sudden accelerations/brakes. To reduce the large storage costs due to the hidden-vector encryption and binary vectors, Qi et al. built another design based on compact cryptographic commitments where the binaries are hidden in the commitments [42]. Average speed and acceleration for a time period can be extracted from the commitments with zero-knowledge proof in the blockchain, which can be used for calculating driving safety scores and insurance premiums. These two schemes both assume that the driver will faithfully follow the protocol and the data fraud behavior can be easily detected as long as the traffic accidents happen.

Table III summarizes the difference between our proposed scheme and existing works, in terms of: (a) architecture, (b) driver privacy, (c) model privacy, (d) adversarial model, (e) public verifiability, (f) fraud resistance, (g) driving safety evaluation model, and (h) permission-based audit. Note that, although most of the existing works also consider data privacy, their data privacy definitions are different from our scheme. In the existing works, the statistics of the driving data such as the aggregated average speed data will be exposed for evaluating driving safety, while our work only reveals whether a trip of a driver is safe or not without leaking any driving data.

VIII. CONCLUSION

We have proposed a blockchain-assisted personalized car insurance scheme, which not only allows auto insurance companies to analyze drivers' behavior in a privacy-preserving manner but also resists fraud attacks launched by malicious drivers. Since our scheme is designed on a consortium blockchain, drivers can verify that auto insurance companies indeed use publicly recognized models to customize their auto premiums based on long-term driving data, which is more transparent than centralized schemes. We have conducted formal security analysis and implemented a proof-of-concept prototype based on an open-source consortium blockchain to demonstrate our scheme's feasibility. For the future work, we will investigate a new blockchain-assisted privacy-preserving scheme that enables auto insurance companies to transparently and privately train driving safety evaluation models through federated learning, and explore more advanced blockchain-assisted vehicular applications.

REFERENCES

[1] J.-L. Yin and B.-H. Chen, "An advanced driver risk measurement system for usage-based insurance on big driving data," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 4, pp. 585–594, 2018.

[2] K. Waddell, "What you're giving up when you let your car insurer track you in exchange for discounts," https://www.consumerreports.org/car-insurance/how-car-insurance-telematics-discounts-really-work-a1549580662/, 2021, [Online; accessed 28-Feb-2022].

[3] L. Wang, X. Lin, E. Zima, and C. Ma, "Towards airbnb-like privacy-enhanced private parking spot sharing based on blockchain," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 2411–2423, 2020.

[4] C. Zhang, L. Zhu, and C. Xu, "Bpaf: Blockchain-enabled reliable and privacy-preserving authentication for fog-based iot devices," *IEEE Consumer Electronics Magazine*, vol. 11, no. 2, pp. 88–96, 2021.

[5] Q. Kong, R. Lu, F. Yin, and S. Cui, "Blockchain-based privacy-preserving driver monitoring for maas in the vehicular iot," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 4, pp. 3788–3799, 2021.

[6] X. Shen, C. Huang, D. Liu, L. Xue, W. Zhuang, R. Sun, and B. Ying, "Data management for future wireless networks: Architecture, privacy preservation, and regulation," *IEEE Network*, vol. 35, no. 1, pp. 8–15, 2021.

[7] D. Chulerttiyawong and A. Jamalipour, "A blockchain assisted vehicular pseudonym issuance and management system for conditional privacy enhancement," *IEEE Access*, vol. 9, pp. 127 305–127 319, 2021.

[8] S. Gyawali, Y. Qian, and R. Q. Hu, "Machine learning and reputation based misbehavior detection in vehicular communication networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8871–8885, 2020.

[9] A. Viand, P. Jattke, and A. Hithnawi, "SoK: Fully homomorphic encryption compilers," in *Proc. of IEEE S&P*, 2021, pp. 1092–1108.

[10] J. Hou, H. Liu, Y. Liu, Y. Wang, P.-J. Wan, and X.-Y. Li, "Model Protection: Real-time privacy-preserving inference service for model privacy at the edge," *IEEE Transactions on Dependable and Secure Computing*, to appear.

[11] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. of Eurocrypt*, 1999, pp. 223–238.

[12] C. Zhang, L. Zhu, J. Ni, C. Huang, and X. Shen, "Verifiable and privacy-preserving traffic flow statistics for advanced traffic management systems," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 10 336–10 347, 2020.

[13] T. Liu, X. Xie, and Y. Zhang, "ZkCNN: Zero knowledge proofs for convolutional neural network predictions and accuracy," in *Proc. of ACM CCS*, 2021, pp. 2968–2985.

[14] Z. Xiong, S. Feng, W. Wang, D. Niyato, P. Wang, and Z. Han, "Cloud/fog computing resource management and pricing for blockchain networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4585–4600, 2018.

[15] Z. Xiong, J. Kang, D. Niyato, P. Wang, and H. V. Poor, "Cloud/edge computing service management in blockchain networks: Multi-leader multi-follower game-based admm for pricing," *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 356–367, 2019.

[16] J. Kang, Z. Xiong, X. Li, Y. Zhang, D. Niyato, C. Leung, and C. Miao, "Optimizing task assignment for reliable blockchain-empowered federated edge learning," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 2, pp. 1910–1923, 2021.

[17] M. Baza, A. Sherif, M. M. Mahmoud, S. Bakiras, W. Alasmary, M. Abdallah, and X. Lin, "Privacy-preserving blockchain-based energy trading schemes for electric vehicles," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 9369–9384, 2021.

[18] I. Damgård and E. Fujisaki, "A statistically-hiding integer commitment scheme based on groups with hidden order," in *Proc. of Asiacrypt*, 2002, pp. 125–142.

[19] C. Huang, R. Lu, J. Ni, and X. Shen, "DAPA: A decentralized, accountable, and privacy-preserving architecture for car sharing services," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 4869–4882, 2020.

[20] Z. Fang, G. Yang, D. Zhang, X. Xie, G. Wang, Y. Yang, F. Zhang, and D. Zhang, "MoCha: Large-scale driving pattern characterization for usage-based insurance," in *Proc. of SIGKDD*, 2021, pp. 2849–2857.

[21] Y. Bian, C. Yang, J. L. Zhao, and L. Liang, "Good drivers pay less: A study of usage-based vehicle insurance models," *Transportation research part A: policy and practice*, vol. 107, pp. 20–34, 2018.

[22] M. Winlaw, S. H. Steiner, R. J. MacKay, and A. R. Hilal, "Using telematics data to find risky driver behaviour," *Accident Analysis & Prevention*, vol. 131, pp. 131–136, 2019.

[23] C. Dong, Y. Wang, A. Aldweesh, P. McCorry, and A. van Moorsel, "Betrayal, distrust, and rationality: Smart counter-collusion contracts for verifiable cloud computing," in *Proc. of ACM CCS*, 2017, pp. 211–227.

[24] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Secure distributed key generation for discrete-log based cryptosystems," *Journal of Cryptology*, vol. 20, no. 1, pp. 51–83, 2007.

[25] C. Hazay, G. L. Mikkelsen, T. Rabin, T. Toft, and A. A. Nicolosi, "Efficient rsa key generation and threshold paillier in the two-party setting," *Journal of Cryptology*, vol. 32, no. 2, pp. 265–323, 2019.

[26] G. Couteau, T. Peters, and D. Pointcheval, "Removing the strong rsa assumption from arguments over the integers," in *Proc. of Eurocrypt*, 2017, pp. 321–350.

[27] B. von Stengel, "Recursive inspection games," *Mathematics of Operations Research*, vol. 41, no. 3, pp. 935–952, 2016.

[28] Y. Lindell, "How to simulate it–a tutorial on the simulation proof technique," *Tutorials on the Foundations of Cryptography*, pp. 277–346, 2017.

[29] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proc. of EuroSys*, 2018, pp. 1–15.

[30] R. A. Popa, H. Balakrishnan, and A. J. Blumberg, "VPriv: Protecting privacy in location-based vehicular services," in *Proc. of Usenix*, 2009, pp. 335–350.

[31] K. Fu and W. Xu, "Risks of trusting the physics of sensors," *Communications of the ACM*, vol. 61, no. 2, pp. 20–23, 2018.

[32] C. Troncoso, G. Danezis *et al.*, "Pripayd: Privacy-friendly pay-as-you-drive insurance," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 5, pp. 742–755, 2010.

[33] L. Zhou, S. Du, H. Zhu, C. Chen, K. Ota, and M. Dong, "Location privacy in usage-based automotive insurance: Attacks and countermeasures," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 1, pp. 196–211, 2018.

[34] N. Rizzo, E. Sprissler, Y. Hong, and S. Goel, "Privacy preserving driving style recognition," in *Proc. of ICCVE*, 2015, pp. 232–237.

[35] O. E. Omri, A. Boudguiga, M. Izabachene, and W. Klaudel, "Privacy-preserving k-means clustering: an application to driving style recognition," in *Proc. of NSS*, 2019, pp. 685–696.

[36] A. Dorri, M. Steger, S. S. Kanhere, and R. Jurdak, "Blockchain: A distributed solution to automotive security and privacy," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 119–125, 2017.

[37] L. Campanile, M. Iacono, A. H. Levis, F. Marulli, and M. Mastroianni, "Privacy regulations, smart roads, blockchain, and liability insurance: putting technologies to work," *IEEE Security & Privacy*, vol. 19, no. 1, pp. 34–43, 2020.

[38] Y. Hui, Y. Huang, Z. Su, T. H. Luan, N. Cheng, X. Xiao, and G. Ding, "BCC: blockchain-based collaborative crowdsensing in autonomous vehicular networks," *IEEE Internet of Things Journal*, vol. 9, no. 6, pp. 4518–4532, 2021.

[39] X. Shen, D. Liu, C. Huang, L. Xue, H. Yin, W. Zhuang, R. Sun, and B. Ying, "Blockchain for transparent data management toward 6g," *Engineering*, vol. 8, pp. 74–85, 2022.

[40] Y. Tian, T. Li, J. Xiong, M. Z. A. Bhuiyan, J. Ma, and C. Peng, "A blockchain-based machine learning framework for edge services in iiot," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1918–1929, 2021.

[41] Z. Wan, Z. Guan, and X. Cheng, "Pride: A private and decentralized usage-based insurance using blockchain," in *Proc. of Cybermatics*, 2018, pp. 1349–1354.

[42] H. Qi, Z. Wan, Z. Guan, and X. Cheng, "Scalable decentralized privacy-preserving usage-based insurance for vehicles," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4472–4484, 2020.

Cheng Huang (Member, IEEE) received his B.Eng and M.Eng in information security from Xidian University, China, in 2013 and 2016 respectively, and received the Ph.D. degree in Electrical and Computer Engineering, University of Waterloo, ON, Canada in 2020. He is currently a Postdoctoral Research Fellow with the Department of Electrical and Computer Engineering, University of Waterloo. His research interests are in the areas of applied cryptography, cyber security and privacy in the mobile network.

Wei Wang (Member, IEEE) received the B.Eng. degree in Information Countermeasure Technology and the M.Eng. degree in Signal and Information Processing from Xidian University in 2011 and 2014, respectively, and the PhD degree in Electrical and Electronic Engineering from Nanyang Technological University (NTU), Singapore, in 2018. From Sep. 2018 to Aug. 2019, he was a postdoctoral fellow at the Department of Electrical and Computer Engineering, University of Waterloo, Canada. Currently, he is a Professor at Nanjing University of Aeronautics and Astronautics. His research interests include wireless communications, space-air-ground integrated networks, wireless security, and blockchain. He was awarded the Chinese government award for outstanding self-financed students abroad in 2018, and the Young Elite Scientist Sponsorship Program, China Association for Science and Technology in 2021.

Dongxiao Liu (Member, IEEE) is a Postdoctoral Research Fellow with the Department of Electrical and Computer Engineering, University of Waterloo. He received the PhD degree at the Department of Electrical and Computer Engineering, University of Waterloo, Canada in 2020. His research interests include security and privacy in intelligent transportation systems, blockchain, and mobile networks.

Rongxing Lu (Fellow, IEEE) received the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, Canada, in 2012. He worked as a Post-Doctoral Fellow with the University of Waterloo from May 2012 to April 2013. He is currently a Mastercard IoT Research Chair, a University Research Scholar, and an Associate Professor with the Faculty of Computer Science (FCS), University of New Brunswick (UNB), Canada. Before that, he worked as an Assistant Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore, from April 2013 to August 2016. He has published extensively in his areas of expertise. His research interests include applied cryptography, privacy enhancing technologies, and IoT-big data security, and privacy. He was a recipient of nine best (student) paper awards from some reputable journals and conferences. He was awarded the most prestigious "Governor General's Gold Medal," when he received his Ph.D. degree. He has won the Eighth IEEE Communications Society (ComSoc) Asia Pacific (AP) Outstanding Young Researcher Award in 2013. He also serves as the Chair of IEEE Communications and Information Security Technical Committee (ComSoc CIS- TC), and the Founding Co-Chair of IEEE TEMS Blockchain and Distributed Ledgers Technologies Technical Committee (BDLT-TC). He is the Winner of 2016–2017 Excellence in Teaching Award, FCS, UNB.

Xuemin (Sherman) Shen (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990. He is a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research focuses on network resource management, wireless network security, Internet of Things, 5G and beyond, and vehicular ad hoc and sensor networks.

Dr. Shen is a registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, a Chinese Academy of Engineering Foreign Member, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society. Dr. Shen received the Canadian Award for Telecommunications Research from the Canadian Society of Information Theory (CSIT) in 2021, the R.A. Fessenden Award in 2019 from IEEE, Canada, Award of Merit from the Federation of Chinese Canadian Professionals (Ontario) in 2019, James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, Joseph LoCicero Award in 2015 and Education Award in 2017 from the IEEE Communications Society, and Technical Recognition Award from Wireless Communications Technical Committee (2019) and AHSN Technical Committee (2013). He has also received the Excellent Graduate Supervision Award in 2006 from the University of Waterloo and the Premier's Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada. He served as the Technical Program Committee Chair/Co-Chair for IEEE Globecom' 16, IEEE Infocom'14, IEEE VTC'10 Fall, IEEE Globecom'07, and the Chair for the IEEE Communications Society Technical Committee on Wireless Communications. Dr. Shen is the President of the IEEE Communications Society. He was the Vice President for Technical & Educational Activities, Vice President for Publications, Member-at-Large on the Board of Governors, Chair of the Distinguished Lecturer Selection Committee, Member of IEEE Fellow Selection Committee of the ComSoc. Dr. Shen served as the Editor-in-Chief of the IEEE IoT JOURNAL, IEEE Network, and IET Communications.