PrivacySignal: Privacy-Preserving Traffic Signal Control for Intelligent Transportation System

Zuobin Ying, *Member, IEEE*, Shuanglong Cao, *Student Member, IEEE*, Ximeng Liu, *Senior Member, IEEE*, Zhuo Ma, *Member, IEEE*, Jianfeng Ma, *Member, IEEE*, and Robert H. Deng, *Fellow, IEEE*

***Abstract*— A new trend of using deep reinforcement learning for traffic signal control has become a spotlight in the Intelligent Transportation System (ITS). However, the traditional intelligent traffic signal control system always collects and transmits vehicle information (e.g., vehicle location, speed, etc.) in the form of plaintext, which would result in the leakage of commuters' privacy and thus bring unnecessary troubles to users. In this paper, we propose a privacy-preserving traffic signal control for an intelligent transportation system (PrivacySignal). It relies on the existing road facilities to achieve the privacy of commuters, which guarantees the practicality of the system. Real-time decision-making and confidentiality of the system can be achieved simultaneously via the design of a series of secure and efficient interactive protocols, that are based on additive secret sharing, to perform the deep *Q*-network (DQN). Moreover, the security of PrivacySignal is testified, meanwhile, the system effectiveness, and the overall efficiency of PrivacySignal is demonstrated through theoretical analysis and simulation experiments. Compared with the existing privacy-preserving schemes of the intelligent traffic signal, PrivacySignal provides a general DQN based privacy-preserving traffic signal control strategy architecture with high efficiency and low-performance loss.**

***Index Terms*— Secure multiparty computation, privacy-preserving, deep reinforcement learning, intelligent transportation systems, intelligent traffic signal control.**

I. Introduction

ECONOMIC losses caused by traffic congestion are increasing. For example, traffic jams cost Americans more than $88 billion in 2019, with an average loss of $1,377 per person, according to a report by Forbes in 2020.[1] Hence, improving traffic conditions can improve urban transportation efficiency, reduce economic losses, and enhance people's daily-based life satisfaction. An intelligent traffic signal control system is a good fix to address the traffic issue. Nevertheless, the traditional traffic signal is controlled by a pre-defined fixed-time plan, which does not take into account the actual traffic conditions, resulting in low efficiency of traffic signal control. The best way to optimize traffic lights is still open to discussion among researchers. Nevertheless, the use of artificial intelligence technology can be arguably a promising implementation method in achieving the goal.

At present, there are three main existing machine learning paradigms, namely supervised learning, unsupervised learning, and reinforcement learning (RL). Combined with deep learning, Deep Reinforcement Learning (DRL) was coined and recognized as the most advanced machine learning framework in current control systems [1]–[5]. For which RL can solve complex control problems while deep learning helps to approximate highly nonlinear functions from the complex dataset. Recently, with the development of the Internet-of-Vehicles (IoV) and DRL technology, more information about roads (e.g., vehicle speed, vehicle position, and waiting time) can be extracted in real-time via the vehicle network. Then IoV service providers (IVSP) use DRL technology (e.g., Deep *Q*-network) to control traffic signals based on traffic conditions dynamically [3]–[5]. In the standard intelligent traffic signal control architecture, commuters send information such as the vehicle's position and speed to the Road Side Unit (RSU) for better service. IVSP collects road information data through RSU and uses them to train artificial intelligence models, like DQN [4], to automatically formulate traffic signal control strategies, with which efficient road traffic management is achieved. Compared with the traditional scheme for traffic signal control, the average waiting time of vehicle users, which deploys deep *Q*-network (DQN) as traffic signal strategy explorer, can be theoretically reduced by 25.7% [4]. With the number of cars clogging roads around the world expected to double in the coming decades, intelligent traffic signal control technology is widely used in the roads of future cities, according to the BBC report.[2]

Manuscript received 25 January 2021; revised 25 July 2021; accepted 19 January 2022. Date of publication 14 February 2022; date of current version 12 September 2022. This work was supported in part by the Key Project of the Anhui Department of Education under Grant KJ2018A0031 and in part by the National Natural Science Foundation of China under Grant 62072109 and Grant U1804263. The Associate Editor for this article was S. Rathinam. *(Corresponding author: Zuobin Ying.)*

Zuobin Ying is with the Faculty of Data Science, City University of Macau, Macau, China, and also with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: zbying@cityu.mo).

Shuanglong Cao is with the College of Computer Science and Technology, Anhui University, Hefei 230601, China (e-mail: slcao2020@gmail.com).

Ximeng Liu is with the College of Computer and Data Science, Fuzhou University, Fuzhou 350025, China (e-mail: snbnix@gmail.com).

Zhuo Ma is with the School of Cyber Engineering, Xidian University, Xi'an 710071, China (e-mail: mazhuo@mail.xidian.edu.cn).

Jianfeng Ma is with the School of Physical and Information Technology, Anhui University, Hefei 230601, China, and also with the School of Electrical and Electronic Engineering, Xidian University, Xi'an 710071, China (e-mail: jfma@mail.xidian.edu.cn).

Robert H. Deng is with the Secure Mobile Centre, School of Information Systems, Singapore Management University, Singapore 178902 (e-mail: robertdeng@smu.edu.sg).

Digital Object Identifier 10.1109/TITS.2022.3149600

[1] https://www.forbes.com/sites/niallmccarthy/2020/03/10/traffic-congestion-costs-us-cities-billions-of-dollars-every-year-infographic/

[2] https://www.bbc.com/future/article/20181212-can-artificial-intelligence-end-traffic-jams

Additionally, the standard intelligent traffic signal control architecture makes signal control decisions under ultra-low latency and real-time conditions. Therefore, IVSP usually deploys a new paradigm called edge computing to improve service quality. Edge computing is a method of optimizing cloud computing architecture by performing data processing at the edge of the network and close to the data source [6]. This can significantly reduce the bandwidth consumption when delivering large amounts of data to the cloud computing center, as well as the latency between the mobile devices and the cloud center. Nevertheless, as for outsourced intelligent traffic signal systems, one of the most prominent problems is the lack of privacy protection. Private data such as the user's vehicle position are directly displayed to the edge server in plaintext. The edge server may leak the user's vehicle data to a third party without the user's authorization. Leakage of vehicle location privacy may cause many an unnecessary headache [7]. For instance, the personal interests of the commuter can be inferred based on the driving route of the vehicle, or in other cases, the commuter could receive harassment advertisements from surrounding services, etc [8]. It is noted that the data privacy of IoV users is essential for an intelligent traffic control system when DRL and connected car technology are involved to alleviate the crowd traffic.

In recent years, researchers have proposed a variety of technologies to protect vehicle data privacy, such as anonymous technology [9] and homomorphic encryption, in response to the problem of vehicle data privacy protection in machine learning. However, anonymous technology [8], [10], only protecting the users' privacy to a certain extent, could be easy to lose useful information, which compromises the accuracy of its traffic signal control strategy. Moreover, research shows that anonymous technology is not sufficient to resist the re-identification attack of [11]. Additionally, current frameworks based on homomorphic encryption [12], [13] to protect machine learning data privacy are time-consuming, memory-intensive as computational overhead is enormous [14]. This indicates they are not suitable for low-latency scenarios such as traffic signal control. Thus, when constructing the privacy-preserving traffic signal control system, privacy protection must be achieved on the premise of ensuring system accuracy and efficiency.

Aiming at achieving these in the process of intelligent traffic signal control, we design a **Privacy**-preserving Traffic **Signal** Control for Intelligent Transportation System (PrivacySignal). The main contributions of this paper are as follows:

- PrivacySignal allows users to outsource vehicle data to RSU for intelligent traffic signal control without revealing user privacy data. To the best of our knowledge, PrivacySignal is the first DQN-based intelligent transportation system with privacy-preserving.
- Some secret sharing (SS)-based secure multiparty computation (MPC) protocols are designed to realize the sub-operations of PrivacySignal. Compared with the existing protocols based on additive homomorphic encryption, the new protocols not only achieve the corresponding functions securely and accurately but also reduce the computational and communication costs dramatically.

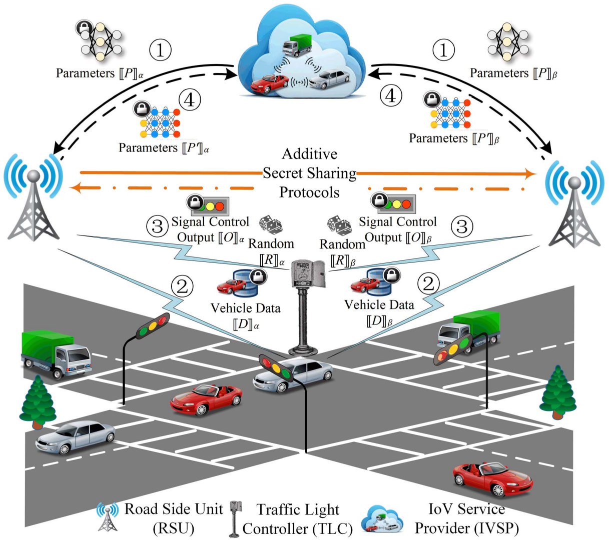

Parameters $[\![P]\!]_\alpha$
①
①
Parameters $[\![P]\!]_\beta$
④
Parameters $[\![P']\!]_\alpha$
④
Parameters $[\![P']\!]_\beta$
Additive
Secret Sharing
Protocols
③ Signal Control Output $[\![O]\!]_\alpha$
Random $[\![R]\!]_\alpha$
Random $[\![R]\!]_\beta$
Signal Control Output $[\![O]\!]_\beta$ ③
②
Vehicle Data $[\![D]\!]_\alpha$
Vehicle Data $[\![D]\!]_\beta$
②
Road Side Unit (RSU)
Traffic Light Controller (TLC)
IoV Service Provider (IVSP)

Fig. 1. System model under consideration.

- A series of interactive protocols are proposed to complete the secure computation of the experience replay, the Q-network Training, and the optimal decision-making of DQN without revealing the original data.
- The security of PrivacySignal is proved through theoretical security analysis. The effectiveness of the PrivacySignal algorithm is verified through simulation experiments, which can meet the needs of most applications of intelligent traffic signal control.

The remaining part of this paper is organized as follows. We present the system model and security model in Section II. In Section III, the primitives about DQN are briefly introduced followed by problem analysis and model presentation. Then the building blocks that support efficient, secret-sharing-based (SS-based) MPC techniques are provided in Section IV. On the basis of that, we propose the details of our system in Section V. Section VI and VII cover the theoretical analysis and experimental results, respectively. Finally, the most related works are stated in Section VIII, and the conclusion is presented in Section IX.

II. System Model and Attack Model

A. Notations

In this section, we formalize the system model and attack model. As a brief notational introduction, we define $[\![x]\!]_\alpha$ and $[\![x]\!]_\beta$ as the sharing of secret value x.[3] Among them, $[\![x]\!]_\alpha$ is the value sent to RSU$_\alpha$, and $[\![x]\!]_\beta$ is the value assigned to RSU$_\beta$. Besides, we also define $[\![x]\!]_*$ as a set of $[\![x]\!]_\alpha$ and $[\![x]\!]_\beta$.

B. System Model

PrivacySignal comprises the following four parties: Vehicle Users (VUs), Road Side Units (RSUs), IoV Service Provider (IVSP), Traffic Signal Controller (TLC) - As shown in Fig. 1.

[3] "$[\![x]\!]$" represents the secret value of x, where $[\![x]\!]_\alpha$ is a random value by the random number generator, and $[\![x]\!]_\beta$ is obtained by subtracting $x - [\![x]\!]_\alpha$.

- **VUs** are vehicle users on the IoV, with real-time vehicle data (e.g., vehicle position, speed, etc.). VUs also have simple data processing capabilities, and they will send the encrypted vehicle data to two fixed RSUs (see ②).
- **RSU**$_\alpha$ and **RSU**$_\beta$ are two RSUs and are responsible for lightweight computing tasks.[4] In PrivacySignal, they accomplish the calculation of all protocols without knowing any vehicle data. The signal control output $[\![O]\!]_\alpha$ and $[\![O]\!]_\beta$ are returned to the TLC (see ③), and the new neural network parameters $[\![P']\!]_\alpha$ and $[\![P']\!]_\beta$ are returned to IVSP after training(see ④).
- **IVSP**, internet of vehicles service providers (e.g., company or IoV research institution with the intelligent traffic signal dynamic control model based on DQN), could provide real-time traffic signal control services. In PrivacySignal, IVSP encrypts its original model parameter and deploys $[\![P]\!]_\alpha$ and $[\![P]\!]_\beta$ to the RSU to provide the traffic signal control service (see ①). At the same time, the IVSP can decrypt the updated model parameters by $[\![P']\!] = [\![P']\!]_\alpha + [\![P']\!]_\beta$, and then update its local DQN model.
- **TLC**, a real-time intelligent traffic signal controller, can decrypt the traffic signals control output through $[\![O]\!] = [\![O]\!]_\alpha + [\![O]\!]_\beta$. Meanwhile, TLC also generates random values for the privacy-preserving protocols operated between RSU$_\alpha$ and RSU$_\beta$ (see ③).

C. Security Model

We use the standard semi-honest attack model [12], [14]–[18], which is also known as passive or honest-but-curious. In this model, each **RSU** completes the protocol precisely as specified. But out of curiosity, they can try to learn as much relevant private information as possible based on stored and processed data. Moreover, the formal definition of adversary $\mathcal{A}$ is as follows.

$\mathcal{A}$ is a legal participant of the communication protocol who will correctly implement the defined protocol. However, $\mathcal{A}$ will try to infer all possible information from the legally received message [19].

It's probably worth noting that we assume that these two **RSU**s cannot collude with each other and cannot be damaged at the same time. Otherwise, adversary or honest but curious participants can directly recover the plaintext by simply adding the secret-sharing together. The data owner **IVSP** and other data generators (**VUs** and **TLC**) are considered honest protocol participants. Furthermore, it is supposed that there is a simulator $\mathcal{S}$. $\mathcal{S}$ that can obtain the real view $\mathcal{V}_1$ and generate random values. With $\mathcal{V}_1$, $\mathcal{S}$ attempts to generate a simulated view $\mathcal{V}_2$ in polynomial time. If adversary $\mathcal{A}$ can find a probabilistic polynomial algorithm to distinguish $\mathcal{V}_1$ and $\mathcal{V}_2$, then $\mathcal{A}$ can successfully initial the attack.[5]

III. Preliminaries

A. Deep Q-Network

The deep Q-network [20] consists of three parts, a Q-network for determining the policy, a target Q-network for generating a target Q value for the loss function, and a replay memory for storing the training samples.

The first core idea of the deep Q-network learning algorithm is to use experience replay. Especially, at each time step t, the agent observes the state s_t by interacting with the environment, selects an action of changing the state to s_{t+1}, and immediately obtains a reward r_t. The experience tuples $e_t \leftarrow (s_t, a_t, r_t, s_{t+1})$ generated by the interaction between the agent and the environment store in the replay memory $\mathbb{D} = \{e_1, e_2, \ldots, e_N\}$. During the training process, some tuples are randomly selected as the supervised samples, and the parameters are trained to reduce the correlation between the samples and obtain stable results.

Second, the DQN contains a Q-network with the current parameter θ and a target Q-network with the previous parameter $\hat{\theta}$. θ is updated multiple times per time step and are copied to $\hat{\theta}$ after every n iterations. $Q(s, a; \theta)$[6] represents the output of the current network for estimating the value function obtained by the agent taking action on a under state s. $\hat{Q}(s, a; \hat{\theta})$ represents the output of the target network. Therefore, at each iteration ζ, the parameter θ is updated to minimize the following loss function,

$$\mathcal{L}(\theta_\zeta)=\sum_{\zeta=1}^{\mathcal{K}}\Big[r_t+\gamma\ \cdot max_{a_{t+1}}\check{Q}(s_{t+1}, a_{t+1}; \hat{\theta})-Q\big(s_t, a_t; \theta\big)\Big]^2.$$

B. Basic Secure Protocols

The following basic secure protocols, proposed in [16], [18], are the basic components to complete the PrivacySignal. Additionally, all protocols mentioned in this article are implemented under the two-party security computing (2PC) [21].

- *Secure Addition Protocol.* The $S_{Add}(\cdot)$ [16] can calculate $f(m, n) = m + n$. Since $m + n = ([\![m]\!]_\alpha + [\![m]\!]_\beta) + ([\![n]\!]_\alpha + [\![n]\!]_\beta) = ([\![m]\!]_\alpha + [\![n]\!]_\alpha) + ([\![m]\!]_\beta + [\![n]\!]_\beta)$, it's easy to see that the protocol can perform secure additions and subtractions locally without the need for interaction between servers. After the computation, each participating party will output $f_* = [\![m]\!]_* + [\![n]\!]_*$. Obviously, we have $[\![f]\!]_\alpha + [\![n]\!]_\beta = m + n$.
- *Secure Multiplication Protocol.* The $S_{Mul}(\cdot)$ protocol [16] is based on the *Beaver's triplet*. Given an input binary group (m, n), the protocol outputs another binary group $([\![f]\!]_\alpha, [\![f]\!]_\beta)$ to the two participants, where $[\![f]\!]_\alpha + [\![f]\!]_\beta = m \cdot n$.
- *Secure Comparison Protocol.* The protocol can be achieved in the comparison of the size of two inputs m and n, while the input of both sides will not be leaked. In this paper, our proposed secure interaction protocol requires two types of secure comparison functions.

[4] We assume that RSU can provide a lightweight edge computing service. And if not, it can also be replaced by other edge nodes.

[5] The above assumptions are commonly used in SS-based privacy-preserving systems [16], [17].

[6] $Q(s, a; \theta)$ is defined as the Q value that takes action a in state s when the Q-network parameter is θ.

```
Input: RSU$_*$ has $\llbracket m \rrbracket_*$, $\llbracket n \rrbracket_*$.
Output: RSU$_*$ outputs $\llbracket f \rrbracket_*$.
  RSU$_*$ computes $\llbracket f \rrbracket_* = \llbracket u \rrbracket_* + \llbracket v \rrbracket_*$ locally.
  RSU$_*$ returns $\llbracket f \rrbracket_*$.
```

Fig. 2. Secure addition protocol.

```
Input: RSU$_*$ has $\llbracket m \rrbracket_*$, $\llbracket n \rrbracket_*$.
Output: RSU$_*$ outputs $\llbracket f \rrbracket_*$.
  TLC generates random numbers $x$, $y$ and computes $z = x \cdot y$.
  TLC splits $a$, $b$ and $c$ into random shares: $x = x_\alpha + x_\beta$, $y = y_\alpha + y_\beta$, $z = z_\alpha + z_\beta$.
  TLC sends $\llbracket x \rrbracket_*$, $\llbracket y \rrbracket_*$ and $\llbracket z \rrbracket_*$ to ES$_*(* = 1, 2)$.
  RSU$_*$ computes $\llbracket g \rrbracket_* \leftarrow \llbracket m \rrbracket_* - \llbracket x \rrbracket_*, \llbracket h \rrbracket_* \leftarrow \llbracket n \rrbracket_* - \llbracket y \rrbracket_*$, and RSU$_\alpha$ sends $g_\alpha$, $h_\alpha$ to RSU$_\beta$, RSU$_2$ sends $g_\beta$, $h_\beta$ to RSU$_\alpha$.
  RSU$_*$ computes $\llbracket g \rrbracket \leftarrow \llbracket m \rrbracket_* - \llbracket x \rrbracket_*, \llbracket h \rrbracket_* \leftarrow \llbracket n \rrbracket_* - \llbracket y \rrbracket_*$, and RSU$_\alpha$ sends $g_\alpha$, $h_\alpha$ to RSU$_\alpha$, RSU$_\beta$ sends $g_\beta$, $h_\beta$ to RSU$_\alpha$.
  RSU$_\alpha$ computes $g \leftarrow g_\alpha + g_\beta, h \leftarrow h_\alpha + h_\beta$, and $f_\alpha \leftarrow z_\alpha + g \cdot y_\alpha + h \cdot x_\alpha$.
  RSU$_\beta$ computes $g \leftarrow g_\alpha + g_\beta, h \leftarrow h_\alpha + h_\beta$, and $f_\beta \leftarrow z_\beta + g \cdot y_\beta + h \cdot x_\beta + g \cdot h$.
  RSU$_*$ returns $\llbracket f \rrbracket_*$.
```

Fig. 3. Secure multiplication protocol.

Type 1 [18], if $m > n$, $S_{Com}(\cdot)$ outputs 1; if $m < n$, it outputs -1; otherwise, it outputs 0. Type 2 [16], if $m < n$, $S^*_{Com}(\cdot)$ outputs 1, otherwise outputs 0.

- *Secure Relu Function Protocol.* Given an input m, and m is split into (m_1, m_2). $S_{Relu}(\cdot)$ [16] can safely implement the function $max(x, 0)$. That is, when $m \leq 0$, S_{Relu} outputs 0, otherwise it outputs m.

The implementation details of two previously proposed additive secret sharing sub-protocols [16], *SecAdd* and *SecMul* are given in Fig. 2 and Fig. 3.

IV. Efficient Sub-Protocol: Secure Computing Based on Additive Secret Sharing

In PrivacySignal, to prevent vehicle data from being leaked to the RSU, all DQN sub-operations must be completed in the form of secret value. Hence, we design a series of lightweight secure computing sub-protocols SS-based MPC technology. First, the **RSU**$_*$ receives its secret value $\llbracket x \rrbracket_*$. The secret value is obtained by randomly splitting x. Then, through the proposed interactive sub-protocol, some functions can be safely calculated. During this period, the two RSUs would not reveal their secret values to each other. Some important variables and the corresponding descriptions are given in TABLE I.

A. Secure Q-Network Protocol

Secure Q-network (S_{Qnet})[7]can calculate the Q value corresponding to different actions in a specific state s_0. In practice, specific state represents the characteristic data of a vehicle, and the action represents the change of traffic signals. Given the specific state ($\llbracket s_0 \rrbracket_\alpha, \llbracket s_0 \rrbracket_\beta$) and Q-network parameters ($\llbracket \theta \rrbracket_\alpha, \llbracket \theta \rrbracket_\beta$), it is worth mentioning that the parameter θ represents the weight $w^{(j)}$ and the bias term $b^{(j)}$, j represents the index of the j-th neural network. Finally, S_{Qnet} can output ($\llbracket q_i \rrbracket_\alpha, \llbracket q_i \rrbracket_\beta$)$(1 \leq i \leq \kappa)$, where $\llbracket q_i \rrbracket_\alpha + \llbracket q_i \rrbracket_\beta = q_i$ and $q_i = Q(s_0, a_i; \theta)$. For simplicity, we only consider that the Q-network has two layers of neurons.[8] The detail of S_{Qnet} is as follows.

TABLE I

Variables and Descriptions

Variables	Descriptions
$\mathbb{S}$	the state set
$\mathbb{A}$	the action set
θ	the Q-network parameters
S_{Add}	secure addition protocol
S_{Mul}	secure multiplication protocol
S_{Com}	type 1 secure comparison protocol
S^*_{Com}	type 2 secure comparison protocol
S_{Relu}	secure Relu function protocol
S_{Qnet}	secure Q-network protocol
S_{Tnet}	secure target Q-network protocol
S_{MaxE}	secure maximum element selection protocol
S_{IV}	secure index value protocol
$S_{\epsilon\text{-}gey}$	secure ϵ-greedy policy protocol
S_{ER}	secure DQN with experience replay
S_{QT}	secure Q-network training
S_{ODM}	secure optimal decision making

(1) **RSU**$_\alpha$ and **RSU**$_\beta$ accept the split state ($\llbracket s_0 \rrbracket_\alpha, \llbracket s_0 \rrbracket_\beta$), as well as the parameters ($\llbracket w^{(\mathrm{I})} \rrbracket_\alpha, \llbracket w^{(\mathrm{I})} \rrbracket_\beta$), ($\llbracket b^{(\mathrm{I})} \rrbracket_\alpha, \llbracket b^{(\mathrm{I})} \rrbracket_\beta$) respectively. For the sake of simplicity, we assume that Q-network has two layers of neurons. For the calculation of the first layer of neurons, **RSU**$_\alpha$ and **RSU**$_\beta$ cooperate to calculate:

$$
\begin{aligned}
(\llbracket \zeta^{(\mathrm{I})} \rrbracket_\alpha, \llbracket \zeta^{(\mathrm{I})} \rrbracket_\beta) &\leftarrow S_{Mul}(\llbracket s_0, w^{(\mathrm{I})} \rrbracket_*),\\
(\llbracket \eta \rrbracket_\alpha, \llbracket \eta \rrbracket_\beta) &\leftarrow S_{Add}(\llbracket \zeta^{(\mathrm{I})}, b^{(\mathrm{I})} \rrbracket_*),\\
(\llbracket \ell \rrbracket_\alpha, \llbracket \ell \rrbracket_\beta) &\leftarrow S_{Relu}(\llbracket \eta \rrbracket_*).
\end{aligned}
$$

At this point, the calculation of the first layer of neurons has ended, and ℓ will be used as the input of the next layer of neurons.

(2) **RSU**$_\alpha$ and **RSU**$_\beta$ receive the output from the upper neurons ($\llbracket \ell \rrbracket_\alpha, \llbracket \ell \rrbracket_\beta$) and the parameters ($\llbracket w^{(\mathrm{II})} \rrbracket_\alpha, \llbracket w^{(\mathrm{II})} \rrbracket_\beta$), ($\llbracket b^{(\mathrm{II})} \rrbracket_\alpha, \llbracket b^{(\mathrm{II})} \rrbracket_\beta$). In the second layer of neurons, they first calculate:

$$
(\llbracket \zeta^{(\mathrm{II})} \rrbracket_\alpha, \llbracket \zeta^{(\mathrm{II})} \rrbracket_\beta) \leftarrow S_{Mul}(\llbracket \ell, w^{(\mathrm{II})} \rrbracket_*),
$$

then, calculate and output:

$$
(\llbracket q_i \rrbracket_\alpha, \llbracket q_i \rrbracket_\beta) \leftarrow S_{Add}(\llbracket \zeta^{(\mathrm{II})}, b^{(\mathrm{II})} \rrbracket_*),
$$

where $\llbracket q_i \rrbracket_\alpha = Q(\llbracket s_0, a_i; \theta \rrbracket_\alpha)$ and $\llbracket q_i \rrbracket_\beta = Q(\llbracket s_0, a_i; \theta \rrbracket_\beta)$.

B. Secure Maximum Element Selection Protocol

Secure maximum element selection protocol (S_{MaxE}) can securely calculate the tuple with the largest value among the

[7] The algorithm process of the secure target Q-network is the same as that of the S_{Qnet}. Due to space limitations, it will not be repeated here. At the same time, we define the safety target Q-network protocol as S_{Tnet}.

[8] To expand a small neural network into a more complex neural network structure, a similar "component combination" construction method can be adopted.

n tuples. Given an encrypted n-tuple $([\![\mathcal{T}]\!]_\alpha, [\![\mathcal{T}]\!]_\beta)$, where $\mathcal{T} = [\![\mathcal{T}]\!]_\alpha + [\![\mathcal{T}]\!]_\beta = t_i (1 < i < n)$. S_{MaxE} outputs $([\![t_{max}]\!]_\alpha, [\![t_{max}]\!]_\beta)$, and $t_{max} = max\{t_1, t_2, \ldots, t_n\}$.

(1) **RSU**$_\alpha$ receives n-tuple $[\![\mathcal{T}]\!]_\alpha$, **RSU**$_\beta$ receives $[\![\mathcal{T}]\!]_\beta$ and then calculates the size of the tuple. Note that although $\mathcal{T}$ is split into $[\![\mathcal{T}]\!]_\alpha$ and $[\![\mathcal{T}]\!]_\beta$, the size of the tuple does not change.

(2) The below algorithm demonstrates the recursive process until $\mathcal{T}$ has only one element left. The algorithm executes as follows:

- If $n = 1$, **RSU**$_\alpha$ and **RSU**$_\beta$ directly output tuple $([\![t_{max}]\!]_\alpha, [\![t_{max}]\!]_\beta)$.

- If $n = 2$, **RSU**$_\alpha$ and **RSU**$_\beta$ compare two elements of a tuple by a secure comparison protocol:

$$([\![\varepsilon]\!]_\alpha, [\![\varepsilon]\!]_\beta) \leftarrow S_{Com}([\![t_1, t_2]\!]_*).$$

Finally, **RSU**$_*$ outputs the maximum value of the two-tuple through the following security calculation:

$$([\![t_{max}]\!]_\alpha, [\![t_{max}]\!]_\beta) \leftarrow S_{Add}(S_{Mul}([\![\varepsilon + 1, t_1]\!]_*), S_{Mul}([\![1 - \varepsilon, t_2]\!]_*)).$$

- If $n > 2$, then **RSU**$_*$ divides the $\mathcal{T}$ tuple into two parts:

$$[\![\mathcal{T}_l]\!]_* \leftarrow ([\![t_1]\!]_*, \ldots, [\![t_{n/2}]\!]_*),$$
$$[\![\mathcal{T}_r]\!]_* \leftarrow ([\![t_{n/2+1}]\!]_*, \ldots, [\![t_n]\!]_*).$$

RSU$_\alpha$ and **RSU**$_\beta$ split n-tuples into n mod 2 tuples through multiple recursions and halving:

$$([\![t_l]\!]_\alpha, [\![t_l]\!]_\beta) \leftarrow S_{MaxE}([\![\mathcal{T}_l]\!]_*),$$
$$([\![t_r]\!]_\alpha, [\![t_r]\!]_\beta) \leftarrow S_{MaxE}([\![\mathcal{T}_r]\!]_*).$$

Next, **RSU**$_*$ only needs to execute the same algorithm as when $n = 2$.

C. Secure Index Value Protocol

The secure index value S_{IV} protocol has three inputs. The first input is a tuple $\mathcal{X}$ of length ℓ, the second input is a value x_j in tuple $\mathcal{X}$, and the last input is a tuple $\mathcal{Y}$ of the same length as $\mathcal{X}$, where $\mathcal{X} = (x_1, x_2, \ldots, x_\ell)$, $\mathcal{Y} = (y_1, y_2, \ldots, y_\ell)$. This protocol realizes by retrieving the index of x_j in tuple $\mathcal{X}$. Via this index, the value of y_i, corresponding to the index of $\mathcal{Y}$, can be retrieved. Furthermore, the inputs $\mathcal{X}$, x_j, and $\mathcal{Y}$ are also randomly divided into $[\![\mathcal{X}]\!]_*$, $[\![x_j]\!]_*$, and $[\![\mathcal{Y}]\!]_*$ and sent to the **RSU**$_*$, respectively. The specific implementation process of the protocol is shown in Fig. 4. First, compare x_j with each element in the tuple $\mathcal{X}$. Here, by invoking the secure comparison protocol, we get an ℓ-tuple Υ (where the index of the Υ tuple in the position of j is 0, and the remaining positions are -1 or 1) (line 3). Second, square each element of tuple Υ (by invoking secure multiplication protocol) to get an ℓ-tuple Φ containing only 0 and 1 (line 4). Third, the value of each element in the Φ tuple takes the opposite number and then adds one to obtain the tuple γ, such that the value of the element with index j in tuple γ is 1 and the remaining elements are 0 (line 5). Finally, multiply the elements of the corresponding index in tuple γ and tuple $\mathcal{Y}$ (by invoking the secure multiplication protocol), so that the element value of the index j in tuple $\mathcal{Y}$ is obtained (lines 6 to 7).

```
Input: RSU$_\alpha$ has $[\![\mathcal{X}, x_j, \mathcal{Y}]\!]_\alpha$; RSU$_\beta$ has $[\![\mathcal{X}, x_j, \mathcal{Y}]\!]_\beta$.
Output: RSU$_\alpha$ outputs $[\![y_0]\!]_\alpha$; RSU$_\beta$ outputs $[\![y_0]\!]_\beta$.
1: RSU$_\alpha$ and RSU$_\beta$ initialize $[\![y_0]\!]_\alpha \leftarrow 0$, $[\![y_0]\!]_\beta \leftarrow 0$.
2: for $i$ = 1, 2, ..., $\ell$ do
3:     $([\![\Upsilon_i]\!]_\alpha, [\![\Upsilon_i]\!]_\beta) \leftarrow S_{Com}([\![x_i, x_j]\!]_*)$.
4:     $([\![\Phi_i]\!]_\alpha, [\![\Phi_i]\!]_\beta) \leftarrow S_{Mul}([\![\Upsilon_i, \Upsilon_i]\!]_*)$.
5:     $[\![\gamma_i]\!]_\alpha \leftarrow 1 - [\![\Phi_i]\!]_\alpha$, $[\![\gamma_i]\!]_\beta \leftarrow -[\![\Phi_i]\!]_\beta$.
6:     $([\![\Psi_i]\!]_\alpha, [\![\Psi_i]\!]_\beta) \leftarrow S_{Mul}([\![\gamma_i, y_i]\!]_*)$.
7:     $[\![y_0]\!]_* \leftarrow [\![y_0]\!]_* + [\![\Psi_i]\!]_*$.
8: end for
9: RSU$_*$ returns $[\![y_0]\!]_*$.
```

Fig. 4. Secure index value protocol.

```
Input: RSU$_\alpha$ has $[\![\mathbb{A}, s_0, \epsilon, \theta]\!]_\alpha$; RSU$_\beta$ has $[\![\mathbb{A}, s_0, \epsilon, \theta]\!]_\beta$.
Output: RSU$_\alpha$ outputs $[\![a_0]\!]_\alpha$; RSU$_\beta$ outputs $[\![a_0]\!]_\beta$.
RSU$_\alpha$ chooses a random number $[\![r]\!]_\alpha$ locally, and
  RSU$_\beta$ selects a random number $[\![r]\!]_\beta$ locally.
$([\![\psi]\!]_\alpha, [\![\psi]\!]_\beta) \leftarrow S^*_{Com}([\![r, \epsilon]\!]_*)$.
if $\psi$ = 1 then
    $([\![q_i]\!]_\alpha, [\![q_i]\!]_\beta) \leftarrow S_{Qnet}([\![s_0, \theta]\!]_*)$.
    $([\![q_0]\!]_\alpha, [\![q_0]\!]_\beta) \leftarrow S_{MaxE}([\![q_1, q_2, ..., q_\kappa]\!]_*)$.
    $([\![a_0]\!]_\alpha, [\![a_0]\!]_\beta) \leftarrow S_{IV}([\![q_i, q_0, \mathbb{A}]\!]_*)$.
else
    RSU$_\alpha$ and RSU$_\beta$ randomly select action
      $([\![a_0]\!]_\alpha, [\![a_0]\!]_\beta)$.
end if
RSU$_*$ returns $[\![a_0]\!]_*$.
```

Fig. 5. Secure ϵ-greedy policy protocol.

D. Secure ϵ-Greedy Policy Protocol

Given the encrypted state $([\![s_0]\!]_\alpha, [\![s_0]\!]_\beta)$ and the Q-network parameters $([\![\theta]\!]_\alpha, [\![\theta]\!]_\beta)$, secure ϵ-greedy policy protocol ($S_{\epsilon\text{-}gey}$) has two different types of output, here a_i represents a randomly selected action:

$$a \leftarrow \begin{cases} argmax_{a'}(s_0, a'; \theta) & 1 \text{ - } \epsilon; \\ a_i & \epsilon. \end{cases}$$

The specific implementation details of the above functions are demonstrated in Fig. 5.

V. Proposed PrivacySignal Framework

In Section IV, we have constructed a series of security sub-protocols with additivity. In this section, to ensure the privacy of vehicle data during the DQN calculation process, we use these sub-protocols to further build the secure DQN with experience replay, secure Q-network training, and secure optimal decision making. These three protocols are used to support the security and efficiency of the entire PrivacySignal system.

A. System Setup and Workflow

Before introducing the implementation details of PrivacySignal, we first outline the workflow of PrivacySignal,

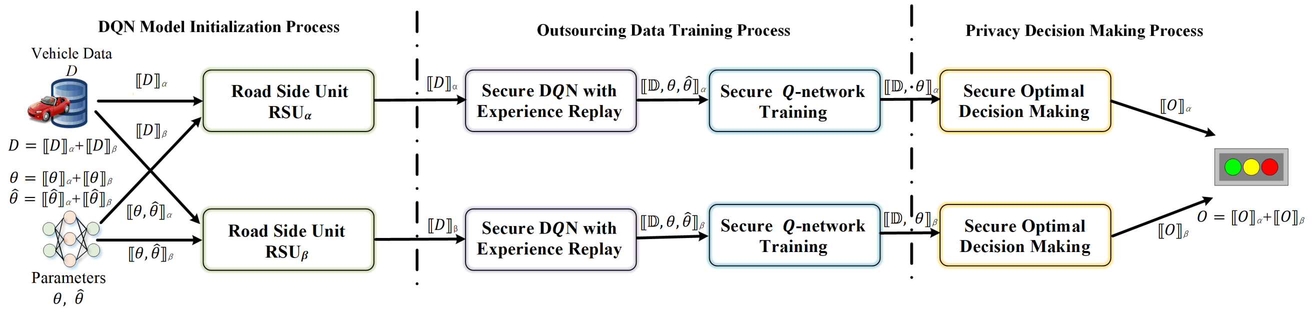

DQN Model Initialization Process
Outsourcing Data Training Process
Privacy Decision Making Process
Vehicle Data
D
$[\![D]\!]_\alpha$
$[\![D]\!]_\beta$
$D = [\![D]\!]_\alpha + [\![D]\!]_\beta$
$\theta = [\![\theta]\!]_\alpha + [\![\theta]\!]_\beta$
$\hat{\theta} = [\![\hat{\theta}]\!]_\alpha + [\![\hat{\theta}]\!]_\beta$
$[\![\theta,\hat{\theta}]\!]_\alpha$
$[\![\theta,\hat{\theta}]\!]_\beta$
Parameters
$\theta,\ \hat{\theta}$
Road Side Unit RSU$_\alpha$
Road Side Unit RSU$_\beta$
$[\![D]\!]_\alpha$
$[\![D]\!]_\beta$
Secure DQN with Experience Replay
Secure DQN with Experience Replay
$[\![\mathbb{D},\theta,\hat{\theta}]\!]_\alpha$
$[\![\mathbb{D},\theta,\hat{\theta}]\!]_\beta$
Secure Q-network Training
Secure Q-network Training
$[\![\mathbb{D},\cdot\theta]\!]_\alpha$
$[\![\mathbb{D},\theta]\!]_\beta$
Secure Optimal Decision Making
Secure Optimal Decision Making
$[\![O]\!]_\alpha$
$[\![O]\!]_\beta$
$O = [\![O]\!]_\alpha + [\![O]\!]_\beta$

Fig. 6. System setup and workflow.

```
Input: RSU$_\alpha$ has $[\![\mathbb{S},\mathbb{D},\theta,\hat{\theta},\gamma,\alpha]\!]_\alpha$ ; RSU$_\beta$ has $[\![\mathbb{S},\mathbb{D},\theta,\hat{\theta},\gamma,\alpha]\!]_\beta$.
Output: TLC outputs fresh memory pool $\mathbb{D}$.
for $\Omega$ = 1, ..., $\mathcal{R}$ do
    RSU$_\alpha$ and RSU$_\beta$ update parameters $([\![\theta,\hat{\theta}]\!]_\alpha,[\![\theta,\hat{\theta}]\!]_\beta) \leftarrow S_{QT}([\![\mathbb{D},\theta,\hat{\theta},\gamma,\alpha]\!]_*)$.
    Set $\varphi \leftarrow \varphi$ + 1.
    VUs selects an observation state $s_{\varphi,1} \in \mathbb{S}$, and sends $([\![s_{\varphi,1}]\!]_\alpha,[\![s_{\varphi,1}]\!]_\beta)$ to RSU$_\alpha$ and RSU$_\beta$.
    for t = 1, ..., $\mathcal{W}$ do
        RSU$_\alpha$ and RSU$_\beta$ compute $([\![a_{\varphi,t}]\!]_\alpha,[\![a_{\varphi,t}]\!]_\beta) \leftarrow \epsilon\text{-}gey_{sec}(s_{\varphi,t},\epsilon,\theta)$. Send $([\![a_{\varphi,t}]\!]_\alpha,[\![a_{\varphi,t}]\!]_\beta)$ to TLC.
        TLC decrypt $a_{\varphi,t} \leftarrow [\![a_{\varphi,t}]\!]_\alpha + [\![a_{\varphi,t}]\!]_\beta$, then use action $a_{\varphi,t}$ to interact with the environment to get new status $s_{\varphi,t+1}$ and instant rewards $r_{\varphi,t}$.
        VUs randomly divides $s_{\varphi,t+1}$, $r_{\varphi,t}$ into $([\![s_{\varphi,t+1}]\!]_\alpha,[\![s_{\varphi,t+1}]\!]_\beta)$, $([\![r_{\varphi,t+1}]\!]_\alpha,[\![r_{\varphi,t}]\!]_\beta)$ by secret sharing, and then sends them to RSU$_\alpha$ and RSU$_\beta$.
        RSU$_\alpha$ and RSU$_\beta$ update $[\![\mathbb{D}]\!]_\alpha$ and $[\![\mathbb{D}]\!]_\beta$ and denote $\mathbb{D}_* \leftarrow \mathbb{D}_* \cup \{[\![s_{\varphi,t}]\!]_*,[\![a_{\varphi,t}]\!]_*,[\![r_{\varphi,t}]\!]_*,[\![s_{\varphi,t+1}]\!]_*\}$.
    end for
end for
RSU$_\alpha$ and RSU$_\beta$ send $[\![\mathbb{D}]\!]_\alpha$ and $[\![\mathbb{D}]\!]_\beta$ to TLC, TLc update memory $\mathbb{D} \leftarrow [\![\mathbb{D}]\!]_\alpha + [\![\mathbb{D}]\!]_\beta$.
TLC returns $\mathbb{D}$.
```

Fig. 7. Secure DQN with experience replay.

as shown in Fig. 6. PrivacySignal consists of three phases: DQN model initialization, outsourcing data training, and privacy decision making. To protect the vehicle user data privacy and the model security of IVSP, the overall goal of PrivacySignal is to make the optimal traffic signal control decision without revealing vehicles' sensitive data to RSUs. To achieve this, RSU in our proposed PrivacySignal system, within a service range of 300 hundred meters, is able to collect the status information of the vehicle (e.g., speed, location, etc.) near the intersection, and then encode it to the status of the intersection. Meanwhile, the vehicle status information, as well as the model parameter, will be split into randomly shared secret values which will be uploaded to the RSUs later. In this way, the commuters' data privacy and the decision model parameters security can be protected. Then, PrivacySignal uses security protocol to perform all subsequent calculations on vehicle data and parameters that are secretly shared. An overview of each phase is as follows.

1) DQN Model Initialization: The decision-making model is a trade secret for IVSP. Once leaked and abused, it will bring economic losses to IVSP. Therefore, to protect the security of the original local model, the IVSP needs to encrypt the model parameters through an SS-based MPC method. The specific process is as follows. The IVSP first defines the state set $\mathbb{S} = \{s_1, s_2, \ldots, s_\lambda\}$ and the action set $\mathbb{A} = \{a_1, a_2, \ldots, a_\kappa\}$. $\mathbb{S}$ represents the state space of vehicle, namely vehicle position and speed. $\mathbb{A}$ indicates the control change of the traffic signals, such as changing the phase of the traffic signals or not making any changes. In addition to defining the state set and action set, the IVSP also needs to initialize the parameter θ of the Q-network, construct a target Q-network of the same structure as the Q-network, and copy the Q-network parameters to the target Q-network parameter $\hat{\theta}$. Before deploying the model to **RSU**$_\alpha$ and **RSU**$_\beta$, $\mathbb{S}$, $\mathbb{A}$, θ, and $\hat{\theta}$ are randomly split into $([\![\mathbb{S}]\!]_\alpha, [\![\mathbb{S}]\!]_\beta)$, $([\![\mathbb{A}]\!]_\alpha, [\![\mathbb{A}]\!]_\beta)$, $([\![\theta]\!]_\alpha, [\![\theta]\!]_\beta)$ and $([\![\hat{\theta}]\!]_\alpha, [\![\hat{\theta}]\!]_\beta)$ by secret sharing. At the same time, other parameters are also randomly split, such as learning rate $([\![\alpha]\!]_\alpha, [\![\alpha]\!]_\beta)$, discount factor $([\![\gamma]\!]_\alpha, [\![\gamma]\!]_\beta)$ and $([\![\epsilon]\!]_\alpha, [\![\epsilon]\!]_\beta)$.

2) Outsourcing Data Training: During the training process of the decision model, some historical roads information, and the original model parameters of the IVSP are needed to train the model. Therefore, to protect the data privacy and model security of vehicle users, in PrivacySignal, VUs collects some historical experience tuples $\mathbb{D}$. It is worth noting that the historical experience at each time stamp t is represented by e_t, which consists of four elements (s_t, a_t, r_t, s_{t+1}). Then, $\mathbb{D}$ is randomly split into shares $[\![\mathbb{D}]\!]_\alpha$ and $[\![\mathbb{D}]\!]_\beta$. When training is required, VUs randomly selects some empirical tuples

$e_t \leftarrow (s_t, a_t, r_t, s_{t+1})$, $1 \leq t \leq \varrho$. and then send them to **RSU**$_\alpha$ and **RSU**$_\beta$. Later, VUs cooperatively calculate the new parameters θ_{new} of the Q-network, and simultaneously update the target Q-network parameters $\hat{\theta}_{new}$ every $\mathcal{C}$ steps. When the entire training is complete, **RSU**$_\alpha$ and **RSU**$_\beta$ return new network parameters to the IVSP.

3) Privacy Decision Making: To protect the privacy of vehicle user data, VU needs to encrypt the VU's most recent state $([\![s_0]\!]_\alpha, [\![s_0]\!]_\beta)$ before making a request decision service, and then send it to the **RSU**$_\alpha$ and **RSU**$_\beta$ for treatment decisions. After **RSU**$_\alpha$ and **RSU**$_\beta$ complete the decision calculation, they will return the optimal decision action $([\![a_0]\!]_\alpha, [\![a_0]\!]_\beta)$ to TLC. TLC decrypts the optimal action by calculating $a_0 = [\![a_0]\!]_\alpha + [\![a_0]\!]_\beta$, where $O = a_0$, $[\![O]\!]_\alpha = [\![a_0]\!]_\alpha$ and $[\![O]\!]_\beta = [\![a_0]\!]_\beta$.

Additionally, the VUs sends a pre-generated random number matrix together with encrypted data to the RSUs. These random numbers are stored on two RSUs. In the process of making optimal traffic signal control decisions, VUs no longer participates in any other calculations. This reduces the communication overhead between VUs and **RSU**$_*$.

B. Secure DQN With Experience Replay

For efficient training of data samples, DQN uses a technique called experience replay. This technique stores the empirical tuple $e_t \leftarrow (s_t, a_t, r_t, s_{t+1})$ obtained by the agent at each time step t in the replay memory $\mathbb{D}$. When the parameter update of the Q-network, the experience randomly extracts from the replay memory $\mathbb{D}$ for training. Through the S_{ER} algorithm, $\mathbb{D}$ will add some new experience sequences. The specific algorithm flow is shown in Fig. 7.

C. Secure Q-Network Training

DQN differs from Q-learning in that DQN uses neural networks instead of Q tables, which reduces the vast overhead of storage and improves efficiency. Note that the structure of the target Q-network is consistent with the structure of the Q-network, and we define the loss function as:

$$\mathcal{L}(\theta_\zeta) = \sum_{\zeta=1}^{\mathcal{K}} \Big[y - Q\big(s_t, a_t; \theta\big) \Big]^2.$$

Deriving the weight function for the loss function yields the following gradient

$$\nabla_{\theta_\zeta} \mathcal{L}(\theta_\zeta) = \sum_{\zeta=1}^{\mathcal{K}} \bigg[\Big(y - Q\big(s_t, a_t; \theta\big) \Big) \nabla_{\theta_\zeta} Q\big(s_t, a_t; \theta\big) \bigg],$$

where, $y = r_t + \gamma \cdot max_{a_{t+1}} \check{Q}(s_{t+1}, a_{t+1}; \hat{\theta})$.

In addition, in the S_{QT} algorithm, $\nabla_{\theta_\zeta} Q(s_t, a_t; \theta)$ represents the gradient set of all parameters. The secure calculation for each gradient is shown as follows:

$$[\![\nabla_{W_\zeta^{(I)}} Q\big(s_t, a_t; \theta\big)]\!]_* \leftarrow [\![W_\zeta^{(II)} \cdot s_t \cdot l \cdot Relu'(\eta)]\!]_*,$$

$$[\![\nabla_{b_\zeta^{(I)}} Q\big(s_t, a_t; \theta\big)]\!]_* \leftarrow [\![W_\zeta^{(II)} \cdot l \cdot Relu'(\eta)]\!]_*,$$

$$[\![\nabla_{W_\zeta^{(II)}} Q\big(s_t, a_t; \theta\big)]\!]_* \leftarrow [\![l]\!]_*,$$

$$[\![\nabla_{b_\zeta^{(II)}} Q\big(s_t, a_t; \theta\big)]\!]_* \leftarrow [\![1]\!]_*.$$

```
Input: RSU$_\alpha$ has $[\![\mathbb{D}, \mathbb{A}, \theta, \hat{\theta}, \gamma, \alpha]\!]_\alpha$ ; RSU$_\beta$ has $[\![\mathbb{D}, \mathbb{A}, \theta, \hat{\theta}, \gamma, \alpha]\!]_\beta$.
Output: RSU$_\alpha$ outputs new parameters $[\![\theta_{new}, \hat{\theta}_{new}]\!]_\alpha$;
    RSU$_\beta$ outputs new parameters $[\![\theta_{new}, \hat{\theta}_{new}]\!]_\beta$.
1:  for $\zeta$ = 1, 2, ..., $\mathcal{K}$ do
2:      RSU$_*$ randomly extract empirical tuple $[\![e_t]\!]_* \leftarrow ([\![s_t]\!]_*, [\![a_t]\!]_*, [\![r_t]\!]_*, [\![s_{t+1}]\!]_*)$ from $[\![\mathbb{D}]\!]_*$.
3:      $([\![\breve{q}_i]\!]_\alpha, [\![\breve{q}_i]\!]_\beta) \leftarrow S_{Tnet}([\![s_{t+1}, \hat{\theta}]\!]_*)$
4:      RSU$_\alpha$ and RSU$_\beta$ compute the maximum $\check{Q}$ value, $([\![\check{Q}]\!]_\alpha, [\![\check{Q}]\!]_\beta) \leftarrow S_{MaxE}([\![q_i]\!]_*)$.
5:      $([\![\mathtt{m}]\!]_\alpha, [\![\mathtt{m}]\!]_\beta) \leftarrow S_{Mul}([\![\gamma, \check{Q}]\!]_*)$.
6:      $[\![\mathtt{n}]\!]_* \leftarrow [\![r_t]\!]_* + [\![\mathtt{m}]\!]_*$.
7:      $([\![q_i]\!]_\alpha, [\![q_i]\!]_\beta) \leftarrow S_{Qnet}([\![s_t, \theta]\!]_*)$.
8:      $([\![q_{a_t}]\!]_\alpha, [\![q_{a_t}]\!]_\beta) \leftarrow S_{IV}([\![a_t, \mathbb{A}, q_i]\!]_*)$.
9:      $[\![\mathtt{g}]\!]_* \leftarrow [\![\mathtt{n}]\!]_* - [\![q_{a_t}]\!]_*$.
10:     $(\nabla_{[\![\theta]\!]_\alpha}, \nabla_{[\![\theta]\!]_\beta}) \leftarrow S_{Mul}(\mathtt{g}, \nabla_\theta Q([\![s_t, a_t; \theta)]\!])$.
11:     RSU$_*$ update parameters $[\![\theta_{new}]\!]_* \leftarrow [\![\theta]\!]_* - [\![\alpha]\!]_* \odot \nabla_{[\![\theta]\!]_*}$.
12:     if $\zeta$ mod $\mathcal{C} = 0$ then
13:         RSU$_*$ resets $[\![\hat{\theta}_{new}]\!]_* \leftarrow [\![\theta_{new}]\!]_*$.
14:     end if
15: end for
16: RSU$_*$ returns $[\![\theta]\!]_*$ and $[\![\hat{\theta}]\!]_*$.
```

Fig. 8. Secure Q-network training.

Finally, we update the parameters of the Q-network as follows:

$$[\![\theta_\zeta Q(s_t, a_t; \theta)_{new}]\!]_* = [\![\theta_\zeta Q(s_t, a_t; \theta)_{old}]\!]_* - [\![\alpha \odot \nabla_{\theta_\zeta} Q(s_t, a_t; \theta)]\!]_*.$$

Detailed S_{QT} algorithms and procedures are shown in Fig. 8. First, VUs randomly extracts the empirical tuples $e_t \leftarrow (s_t, a_t, r_t, s_{t+1})$ in memory $\mathbb{D}$ and distributes them to **RSU**$_\alpha$ and **RSU**$_\beta$ at random (line 2). The following calculations abide by the loss function. Firstly, **RSU**$_\alpha$ and **RSU**$_\beta$ cooperate to security calculate the values of $max_{a_{t+1}} \check{Q}(s_{t+1}, a_{t+1}; \hat{\theta})$ (lines 3 to 4) and complete the secure calculation of the y value (lines 5 to 6). Secondly, **RSU**$_\alpha$ and **RSU**$_\beta$ security calculate $Q(s_t, a_t; \theta)$ (lines 7 to 8). Thirdly, **RSU**$_*$ securely calculates the gradient of each parameter and updates the parameters of each Q-network according to the gradient descent method (lines 9 to 11). Finally, in every $\mathcal{C}$ step, need to copy the parameters of the Q-network to the target Q-network (lines 12 to 14).

D. Secure Optimal Decision Making

After the training optimization of the Q-network is completed, We can provide optimal traffic signal control decision service for VUs. First, to protect the privacy of VUs data, VUs randomly divides the state s_0 of a VU into $[\![s_0]\!]_\alpha$ and $[\![s_0]\!]_\beta$, and then sends it to the RSU. After receiving $[\![s_0]\!]_\alpha$ and $[\![s_0]\!]_\beta$, the RSU calculates by using the SS-based MPC protocol. Finally, **RSU**$_*$ returns the optimal traffic signal control decision $[\![a_0]\!]_*$ to VUs. The specific process of secure optimal decision making (S_{ODM}) algorithm is shown in Fig. 9.

```
Input: RSU_α has $\llbracket s_0 \rrbracket_\alpha$, $\llbracket \theta \rrbracket_\alpha$, $\llbracket \mathbb{A} \rrbracket_\alpha$; RSU_β has $\llbracket s_0 \rrbracket_\beta$, $\llbracket \theta \rrbracket_\beta$, $\llbracket \mathbb{A} \rrbracket_\beta$.
Output: RSU_α outputs $\llbracket a_0 \rrbracket_\alpha$; RSU_β outputs $\llbracket a_0 \rrbracket_\beta$.
$(\llbracket q_i \rrbracket_\alpha, \llbracket q_i \rrbracket_\beta) \leftarrow S_{Qnet}(\llbracket s_0, \theta \rrbracket^*)$.
$(\llbracket q_0 \rrbracket_\alpha, \llbracket q_0 \rrbracket_\beta) \leftarrow S_{MaxE}(\llbracket q_1, q_2, ..., q_\kappa \rrbracket_*)$.
$(\llbracket a_0 \rrbracket_\alpha, \llbracket a_0 \rrbracket_\beta) \leftarrow S_{IV}(\llbracket q_i, q_0, \mathbb{A} \rrbracket_*)$.
RSU_* returns $a_0^*$.
```

Fig. 9. Secure optimal decision making.

E. Feasibility for Multiparty Computation

For a better understanding of the workflow of PrivacySignal, we only discuss the two-party setup (i.e., two RSUs). However, in applications that use PrivacySignal, sometimes more than three parties may be involved. Therefore, we further discuss the feasibility of extending PrivacySignal to multi-party computing (MPC) in this section.

To expand to MPC, we must make adjustments to the PrivacySignal security protocol to accommodate MPC. According to the definition of the above-mentioned interactive algorithm, it is not difficult to find that they are composed of four basic protocols: S_{Add}, S_{Mul}, S_{Com} and S^*_{Com}. Therefore, proving the feasibility of PrivacySignal under the MPC setting is equivalent to proving that all four basic protocols support MPC. In protocols, S_{Add} only contains local addition operations. It is extremely easy to prove that S_{Add} is feasible for MPC. S_{Mul}, S_{Com} and S^*_{Com} are based on the Beavers triplet [22] and most significant bit (MSB) [23] respectively. These three protocols were originally proposed for MPC and their support for MPC have been proven. It can be seen from the above that if more RSUs are involved in practical applications, PrivacySignal can be easily extended to MPC.

VI. Theoretical Analysis

A. Correctness

During the operation of PrivacySignal, the vehicle data D collected through the IoV is split into $D = \llbracket D \rrbracket_\alpha + \llbracket D \rrbracket_\beta$, based on the additive of secret sharing. Then, in the framework of our design, a series of linear and non-linear operations are performed on D. Intuitively, the final output of $\theta = \llbracket \theta \rrbracket_\alpha + \llbracket \theta \rrbracket_\beta$, $\hat{\theta} = \llbracket \hat{\theta} \rrbracket_\alpha + \llbracket \hat{\theta} \rrbracket_\beta$, and $O = \llbracket O \rrbracket_\alpha + \llbracket O \rrbracket_\beta$. Since the calculation is performed on the ciphertext, the final output result may have a loss of accuracy. However, with our framework design, we can ensure that the value of the system output is accurate.

First of all, it has been proved that in all the four basic security protocols proposed in the second section, no matter how many times they are called, the output is still correct. Secondly, in the five sub-protocols we designed, it is composed of invoking these four basic protocols. Therefore, for the input $\mathcal{X} = \llbracket \mathcal{X} \rrbracket_\alpha + \llbracket \mathcal{X} \rrbracket_\beta$, we naturally have the output $\mathcal{Y} = \llbracket \mathcal{Y} \rrbracket_\alpha + \llbracket \mathcal{Y} \rrbracket_\beta$. At the same time, these five sub-protocols maintain the additivity of the algorithm. Secondly, for the S_{ER}, S_{QT} and S_{ODM} protocols, according to the additive secret sharing algorithm, the two protocols can be combined by the above five basic protocols and five sub-protocols. This means that in these two protocols, the additivity of this algorithm is also maintained. Also, for any input $\mathcal{X}$, the output is $\mathcal{Y}$. Finally, we can conclude that for any function F, we have $F = \llbracket F \rrbracket_\alpha + \llbracket F \rrbracket_\beta$, if and only if $F = f(\varsigma_1, \varsigma_2, \ldots)$, where $\varsigma_i (i = 1, 2, \ldots)$ is a random linear mapping function and ς_i can be any of the security functions in this paper. Therefore, based on the inference, we can ensure that $\theta = \llbracket \theta \rrbracket_\alpha + \llbracket \theta \rrbracket_\beta$, $\hat{\theta} = \llbracket \hat{\theta} \rrbracket_\alpha + \llbracket \hat{\theta} \rrbracket_\beta$, and $O = \llbracket O \rrbracket_\alpha + \llbracket O \rrbracket_\beta$ are correct.

B. Security

In this section, we analyze the security of the proposed PrivacySignal system.

Definition 1: If all the sub-protocols of a protocol are fully emulated, then the protocol is fully emulated [24].

Definition 2: We say that a protocol π is secure if there exists a probabilistic polynomial-time simulator $\mathcal{S}$ that can generate a view for the adversary $\mathcal{A}$ in the real world and the view is computationally indistinguishable from its rear view [16].

In addition to the **Definition 1**, the following lemmas are also needed.

Lemma 1: If a random element r is uniformly distributed on $\mathbb{Z}_n$ and independent from any variable $x \in \mathbb{Z}_n$, then $r \pm x$ is also uniformly random and independent from x [24].

Lemma 2: The protocols S_{Add}, S_{Mul}, S_{Com}, S^*_{Com} and S_{Relu} are secure in the semi-honest model [16], [17].

According to **Definition 1**, we only need to verify the security of other protocols.

Theorem 1: The protocols S_{Qnet} and S_{Tnet} are secure in the semi-honest model.

Proof: For S_{Qnet} protocol, the view that $\mathbf{RSU}_\alpha$ will get is $\mathsf{view}_\alpha = \{\llbracket s_0 \rrbracket_\alpha, \llbracket \theta \rrbracket_\alpha, \llbracket q_i \rrbracket_\alpha, \llbracket \zeta^{(j)} \rrbracket_\alpha, \llbracket \eta \rrbracket_\alpha, \llbracket l \rrbracket_\alpha\}$, where $i = 1, 2, \ldots, \kappa$, $j = \mathrm{I}, \mathrm{II}$ and $\{\llbracket w^{(j)} \rrbracket_\alpha, \llbracket b^{(j)} \rrbracket_\alpha\} \in \llbracket \theta \rrbracket_\alpha$. Among them, $\llbracket s_0 \rrbracket_\alpha$ and $\llbracket \theta \rrbracket_\alpha$ are uniformly random as inputs. Since $\llbracket \zeta^{(j)} \rrbracket_\alpha$ is the output S_{Mul}, according to **Lemma 2**, $\llbracket \zeta^{(j)} \rrbracket_\alpha$ is uniformly random. Similarly, $\llbracket l \rrbracket_\alpha$ is the output of S_{Relu}, then $\llbracket l \rrbracket_\alpha$ is also uniformly random. Besides, because of $\llbracket \eta \rrbracket_\alpha = \llbracket \zeta^{\mathrm{I}} \rrbracket_\alpha + \llbracket b^{\mathrm{I}} \rrbracket_\alpha$, so $\llbracket \eta \rrbracket_\alpha$ is uniformly random. Finally, the output of $\mathbf{RSU}_\alpha$ is $\mathsf{output}_\alpha = (\llbracket q_i \rrbracket_\alpha = \llbracket \zeta^{\mathrm{II}} \rrbracket_\alpha + \llbracket b^{\mathrm{II}} \rrbracket_\alpha)$, where $\llbracket q_i \rrbracket_\alpha$ is also uniformly random. Therefore, both view_α and output_α can be simulated by the views of simulators $\mathcal{S}$ and the views of $\mathcal{S}$ and $\mathcal{A}$ is computationally indistinguishable. Using the same method, we can also prove view_β and output_β for $\mathbf{RSU}_\beta$ are computationally indistinguishable. According to **Definition 1**, it can conclude that protocol S_{Qnet} is secure. Correspondingly, we can conclude S_{Qnet} protocol is secure in the semi-honest model. Meanwhile, since the safety certificate of S_{Tnet} is almost the same as that of S_{Qnet}, we won't reiterate them here. □

Theorem 2: The protocols S_{MaxE} is secure in the semi-honest model.

Proof: In S_{MaxE}, the view that $\mathbf{RSU}_*$ will get is $\mathsf{view}_* = \{\llbracket \mathcal{T} \rrbracket_*, \llbracket \mathcal{T}_l \rrbracket_*, \llbracket t_l \rrbracket_*, \llbracket \varepsilon \rrbracket_*\}$. Among them, $\llbracket \mathcal{T} \rrbracket_*$ is uniformly random as inputs. $\llbracket \mathcal{T}_l \rrbracket_*$ is obtained by iteratively cutting half of the uniformly random n-tuples $\llbracket \mathcal{T} \rrbracket_*$, so $\llbracket \mathcal{T}_l \rrbracket_*$ is still uniformly random. Similarly, $\llbracket t_l \rrbracket_*$ is obtained by iteratively intercepting elements in $\llbracket \mathcal{T}_l \rrbracket_*$, and $\llbracket t_l \rrbracket_*$ is uniformly random. Furthermore, $\llbracket \varepsilon \rrbracket_*$ is obtained by agreement S_{Com}. According to **Lemma 2**, it can be inferred that $\llbracket \varepsilon \rrbracket_*$ is uniform and random. The final output of $\mathbf{RSU}_*$ is $\mathsf{output}_* = \{\llbracket t_{max} \rrbracket_* = S_{Add}(S_{Mul}(\llbracket \varepsilon + 1, t_l \rrbracket_*), S_{Mul}(\llbracket 1 - \varepsilon, t_r \rrbracket_*))\}$ since $\llbracket t_{max} \rrbracket_*$ is obtained by protocol S_{Mul} and S_{Add}. According to **Lemma 2**, $\llbracket t_{max} \rrbracket_*$ is still uniformly random. Therefore, both view_* and output_* can be simulated by the views of simulators $\mathcal{S}$ and the views of $\mathcal{S}$ and $\mathcal{A}$ is computationally indistinguishable. From **Definition 1**, we can conclude that protocol S_{MaxE} is secure. □

Theorem 3: The protocols S_{IV} is secure in the semi-honest model.

Proof: For S_{IV}, the view that $\mathbf{RSU}_\alpha$ gets is $\mathsf{view}_\alpha = \{\llbracket \mathcal{X} \rrbracket_\alpha, \llbracket x_j \rrbracket_\alpha, \llbracket \mathcal{Y} \rrbracket_\alpha, \llbracket \Upsilon_i \rrbracket_\alpha, \llbracket \Phi_i \rrbracket_\alpha, \llbracket \gamma_i \rrbracket_\alpha, \llbracket \Psi_i \rrbracket_\alpha\}$. Among them, $\llbracket \mathcal{X} \rrbracket_\alpha$, $\llbracket x_j \rrbracket_\alpha$, and $\llbracket \mathcal{Y} \rrbracket_\alpha$ are uniformly random as inputs. $\llbracket \Upsilon_i \rrbracket_\alpha$ is the output of protocol S_{Com}. According to **Lemma 2**, $\llbracket \Upsilon_i \rrbracket_\alpha$ is uniform and random. Similarly, $\llbracket \Phi_i \rrbracket_\alpha$ and $\llbracket \Psi_i \rrbracket_\alpha$ are both obtained from the protocol S_{Mul}. Therefore, according to **Lemma 2**, it is proved that $\llbracket \Phi_i \rrbracket_\alpha$ and $\llbracket \Psi_i \rrbracket_\alpha$ are also uniformly random. Moreover, $\llbracket \gamma_i \rrbracket_\alpha = 1 - \llbracket \Phi_i \rrbracket_\alpha$, before this, we have proved that $\llbracket \Phi_i \rrbracket_\alpha$ is uniformly random. Simultaneously, $1 \in \mathbb{Z}_n$. Hence, according to **Lemma 1**, it is concluded that γ_i is uniformly random. Finally, the output of $\mathbf{RSU}_\alpha$ is $\mathsf{output}_\alpha = (\llbracket y_0 \rrbracket_\alpha = \llbracket y_0 \rrbracket_\alpha + \llbracket \Psi_i \rrbracket_\alpha)$. Nevertheless, $\llbracket \Psi_i \rrbracket_\alpha$ has been found to be uniformly random, $\llbracket y_0 \rrbracket_\alpha$ is equivalent to adding up these random numbers together. According to **Lemma 1**, $\llbracket y_0 \rrbracket_\alpha$ is also proved to be uniform and random. Therefore, both view_β and output_β can be simulated by the views of simulators $\mathcal{S}$ and the views of $\mathcal{S}$ and $\mathcal{A}$ is computationally indistinguishable. During the implementation of protocol S_{IV}, the view obtained by $\mathbf{RSU}_\beta$ is $\mathsf{view}_\alpha = \{\llbracket \mathcal{X} \rrbracket_\beta, \llbracket x_j \rrbracket_\beta, \llbracket \mathcal{Y} \rrbracket_\beta, \llbracket \Upsilon_i \rrbracket_\beta, \llbracket \Phi_i \rrbracket_\beta, \llbracket \gamma_i \rrbracket_\beta, \llbracket \Psi_i \rrbracket_\beta\}$. And, the final output is $\mathsf{output}_\alpha = (\llbracket y_0 \rrbracket_\beta = \llbracket y_0 \rrbracket_\beta + \llbracket \Psi_i \rrbracket_\beta)$. A similar proof method proves that these values are uniformly random. Then, we also prove view_β and output_β for $\mathbf{RSU}_\beta$ are computationally indistinguishable. According to **Definition 1**, it can be concluded that protocol S_{IV} is secure. □

Theorem 4: The protocols $S_{\epsilon\text{-}gey}$ is secure in the semi-honest model.

Proof: During the running of protocol $S_{\epsilon\text{-}gey}$, the view obtained by $\mathbf{RSU}_\alpha$ is $\mathsf{view}_\alpha = \{\llbracket s_0 \rrbracket_\alpha, \llbracket \theta \rrbracket_\alpha, \llbracket \epsilon \rrbracket_\alpha, \llbracket \psi \rrbracket_\alpha, \llbracket q_i \rrbracket_\alpha, \llbracket r \rrbracket_\alpha\}$. Between them, $\llbracket s_0 \rrbracket_\alpha$, $\llbracket \theta \rrbracket_\alpha$, $\llbracket r \rrbracket_\alpha$ and $\llbracket \epsilon \rrbracket_\alpha$ as inputs are uniformly random. $\llbracket \psi \rrbracket_\alpha$ is the output of protocol S^*_{Com}. According to **Lemma 2**, we can consider $\llbracket \psi \rrbracket_\alpha$ to be uniformly random. Beside, $\llbracket q_i \rrbracket_\alpha$ is the output of the protocol S_{Qnet}, and we have previously proven that the protocol is secure. Therefore, according to **Theorem 1**, it is natural to conclude that $\llbracket q_i \rrbracket_\alpha$ is uniformly random. It is worth noting that there are two types of output in this protocol, when the random number r is within the range of $1-\epsilon$. $\llbracket q_0 \rrbracket_\alpha$ is obtained from protocol S_{MaxE}. According to **Theorem 2**, we can also conclude that $\llbracket q_0 \rrbracket_\alpha$ is uniformly random. In contrast, when the random number r is less than ϵ. Since $\llbracket q_0 \rrbracket_\alpha$ is a value randomly selected in $\llbracket q_i \rrbracket_\alpha$, so $\llbracket q_0 \rrbracket_\alpha$ is uniformly random. Thus, both view_α and output_α can be simulated by the views of simulators $\mathcal{S}$ and the views of $\mathcal{S}$ and $\mathcal{A}$ is computationally indistinguishable. Similarly, view_α and output_α are computationally indistinguishable for $\mathbf{RSU}_\beta$. By **Definition 1**, protocol $S_{\epsilon\text{-}gey}$ is secure in the semi-honest model. □

Theorem 5: The protocols S_{ER}, S_{QT} and S_{ODM} in PrivacySignal are secure in the semi-honest model.

Proof: According to **Lemma 1**, the protocols S_{ER}, S_{QT}, and S_{ODM} composed of protocols S_{Add}, S_{Mul}, S_{Com}, S^*_{Com}, S_{Relu}, S_{Qnet}, S_{Tnet}, S_{MaxE}, S_{IV} and $S_{\epsilon\text{-}gey}$. For the view_* and output_* obtained by $\mathbf{RSU}_*$, we can use the view set $\mathcal{V}_*$ and output set Γ_* to represent. Here, $\mathcal{V}_* = \mathsf{view}_{S_{Add}} \cup \mathsf{view}_{S_{Mul}} \cup \mathsf{view}_{S_{Com}} \cup \mathsf{view}_{S^*_{Com}} \cup \mathsf{view}_{S_{Relu}} \cup \mathsf{view}_{S_{Qnet}} \cup \mathsf{view}_{S_{Tnet}} \cup \mathsf{view}_{S_{MaxE}} \cup \mathsf{view}_{S_{IV}} \cup \mathsf{view}_{S_{\epsilon\text{-}gey}}$ and $\Gamma_* = \mathsf{output}_{S_{Add}} \cup \mathsf{output}_{S_{Mul}} \cup \mathsf{output}_{S_{Com}} \cup \mathsf{output}_{S_{Relu}} \cup \mathsf{output}_{S_{Qnet}} \cup \mathsf{output}_{S_{Tnet}} \cup \mathsf{output}_{S_{MaxE}} \cup \mathsf{output}_{S_{IV}} \cup \mathsf{output}_{S_{\epsilon\text{-}gey}}$. In each protocol, the views of repeatedly called sub-protocols merged into the same subset. The view set of the uncalled sub-protocol, we can set it to null. In **Theorem 1**, **Theorem 2**, **Theorem 3**, and **Theorem 4**, we have shown that the sub-protocols are secure and that they are perfectly simulatable. According to **Lemma 1**, it concluded that $\mathcal{V}_*$ and Γ_* are also fully simulatable. Consequently, $\mathcal{A}$ cannot find a polynomial algorithm to distinguish between $\mathcal{V}_*$ and Γ_* as real view sets of simulated view sets. In conclusion, we can conclude that protocol S_{ER}, S_{QT} and S_{ODM} in PrivacySignal are secure in the semi-honest model. Meanwhile, it is natural to infer that our PrivacySignal is secure. □

C. Efficiency

VUs need to perform encryption before sending data. Although it would bring some computational overhead to VUs, using additive secret sharing in our system will be less complicated than using homomorphic encryption in previous work. On the one hand, VUs only need to generate random numbers and simple subtraction when encrypting vehicle data. The calculation time is only $\mathcal{O}(l)$, which greatly reduces the computational overhead of the VUs' terminal device. On the other hand, the process of making the optimal traffic signal control strategy is performed on the RSU, and the VUs' terminals are only responsible for generating random numbers in the multiplication calculation and do not participate in complicated calculations, which further reduces the users' overhead.

For the efficiency analysis of the two RSUs, we mainly analyze from the aspects of calculation cost and communication overhead. As shown in TABLE II, we first analyze the efficiency of the five sub-protocols[9] and then deduced the efficiency of the S_{ER}, S_{QT}, and S_{ODM} protocols. Here, we assume that the computational cost of the basic protocols S_{Add}, S_{Mul}, S_{Com}, S^*_{Com} and S_{Relu} in Section III.B are $\mathcal{O}$ (S_{Add}), $\mathcal{O}$ (S_{Mul}), $\mathcal{O}$ (S_{Com}), $\mathcal{O}$ (S^*_{Com}) and $\mathcal{O}$ (S_{Relu}). The communication overhead of S_{Mul}, S_{Com}, S^*_{Com} and S_{Relu} is $\mathcal{L}$ (S_{Mul}), $\mathcal{L}$ (S_{Com}), $\mathcal{L}$ (S^*_{Com}) and $\mathcal{L}$ (S_{Relu}).

VII. Performance Evaluation

Our privacy protection DQN framework is implemented in Python 3.7. The package NumPy 1.6 is used as a

[9] The calculation cost and communication overhead of S_{Qnet} and S_{Tnet} are the same. For simplicity, we only analyzed the calculation cost and communication overhead of S_{Qnet}.

TABLE II

SUMMARY OF CALCULATION COST AND COMMUNICATION OVERHEAD OF EACH PROTOCOL

Protocol	**Calculation Cost**	**Communication Overhead**
S_{Qnet}	$2\mathcal{O}(S_{Mul}) + 2\mathcal{O}(S_{Add}) + \mathcal{O}(S_{Relu})$	$2\mathcal{L}(S_{Mul}) + \mathcal{L}(S_{Relu})$
S_{MaxE}	$\log_2 n \cdot \big(2\mathcal{O}(S_{Mul}) + \mathcal{O}(S_{Com}) + \mathcal{O}(S_{Add})\big)$	$\log_2 n \cdot \big(2\mathcal{L}(S_{Mul}) + \mathcal{L}(S_{Com})\big)$
Iv_{sec}	$\ell \cdot \big(2\mathcal{O}(S_{Mul}) + \mathcal{O}(S_{Com}) + 2\mathcal{O}(S_{Add})\big)$	$\ell \cdot \big(2\mathcal{L}(S_{Mul}) + \mathcal{L}(S_{Com})\big)$
$S_{\epsilon\text{-}gey}$	$(1-\epsilon) \cdot \big(\mathcal{O}(S_{Qnet}) + \mathcal{O}(S_{MaxE}) + \mathcal{O}(S_{IV})\big) + \mathcal{O}(S^*_{Com})$	$(1-\epsilon) \cdot \big(\mathcal{L}(S_{Qnet}) + \mathcal{L}(S_{MaxE}) + \mathcal{L}(S_{IV})\big) + \mathcal{L}(S^*_{Com})$
S_{ER}	$\mathcal{R} \cdot \mathcal{W} \cdot \mathcal{O}(S_{QT}) \cdot \mathcal{O}(S_{\epsilon\text{-}gey})$	$\mathcal{R} \cdot \mathcal{W} \cdot \mathcal{L}(S_{QT}) \cdot \mathcal{L}(S_{\epsilon\text{-}gey})$
S_{QT}	$\mathcal{K} \cdot \mathcal{O}(S_{Qnet}) + \mathcal{O}(S_{Tnet}) + \mathcal{O}(S_{MaxE}) + \mathcal{O}(S_{IV})$ $+ 10\mathcal{O}(S_{Mul}) + 6\mathcal{O}(S_{Add}) + 3\mathcal{O}(S_{Relu}) + 2\mathcal{O}(S^*_{Com})$	$\mathcal{K} \cdot \mathcal{L}(S_{Qnet}) + \mathcal{L}(S_{Tnet}) + \mathcal{L}(S_{MaxE}) + \mathcal{L}(S_{IV})$ $+ 10\mathcal{L}(S_{Mul}) + 3\mathcal{L}(S_{Relu}) + 2\mathcal{L}(S^*_{Com})$
S_{ODM}	$\mathcal{O}(S_{Qnet}) + \mathcal{O}(S_{MaxE}) + \mathcal{O}(S_{IV})$	$\mathcal{L}(S_{Qnet}) + \mathcal{L}(S_{MaxE}) + \mathcal{L}(S_{IV})$

TABLE III

PARAMETERS IN THE DRL

Variables	Descriptions
The number of status λ	8
The number of actions κ	8
The number of invokes τ	2
The number of cycles W	2
The length of sequences R	2
The length of tuple l	8
The precision ϵ	0.1
Discount factor γ	0.9
Learning rate α	0.01

multi-dimensional container of the numbers to implement our secret-sharing-based secure computing protocols. The Q-network uses Tensorflow 1.13 for training. Additionally, we use personal computers as vehicle devices with an Intel(R) Core (TM) i7-6700 CPU @3.40GHz and 8.00GB of RAM. The vehicle device sends the encrypted ciphertext to two RSUs for privacy-preserving optimal traffic signal control decision making. Each RSU is equipped with an Intel(R) Core (TM) i7-7700 HQ CPU @2.80GHz and 8.00GB of RAM. To obtain near-real results in the above evaluation environment, we use the SUMO simulator [25] as the environment, which provides real-time traffic simulation in a micro way. It should be emphasized that the default settings of the experiment as shown in TABLE III.

A. PrivacySignal Performance Evaluation

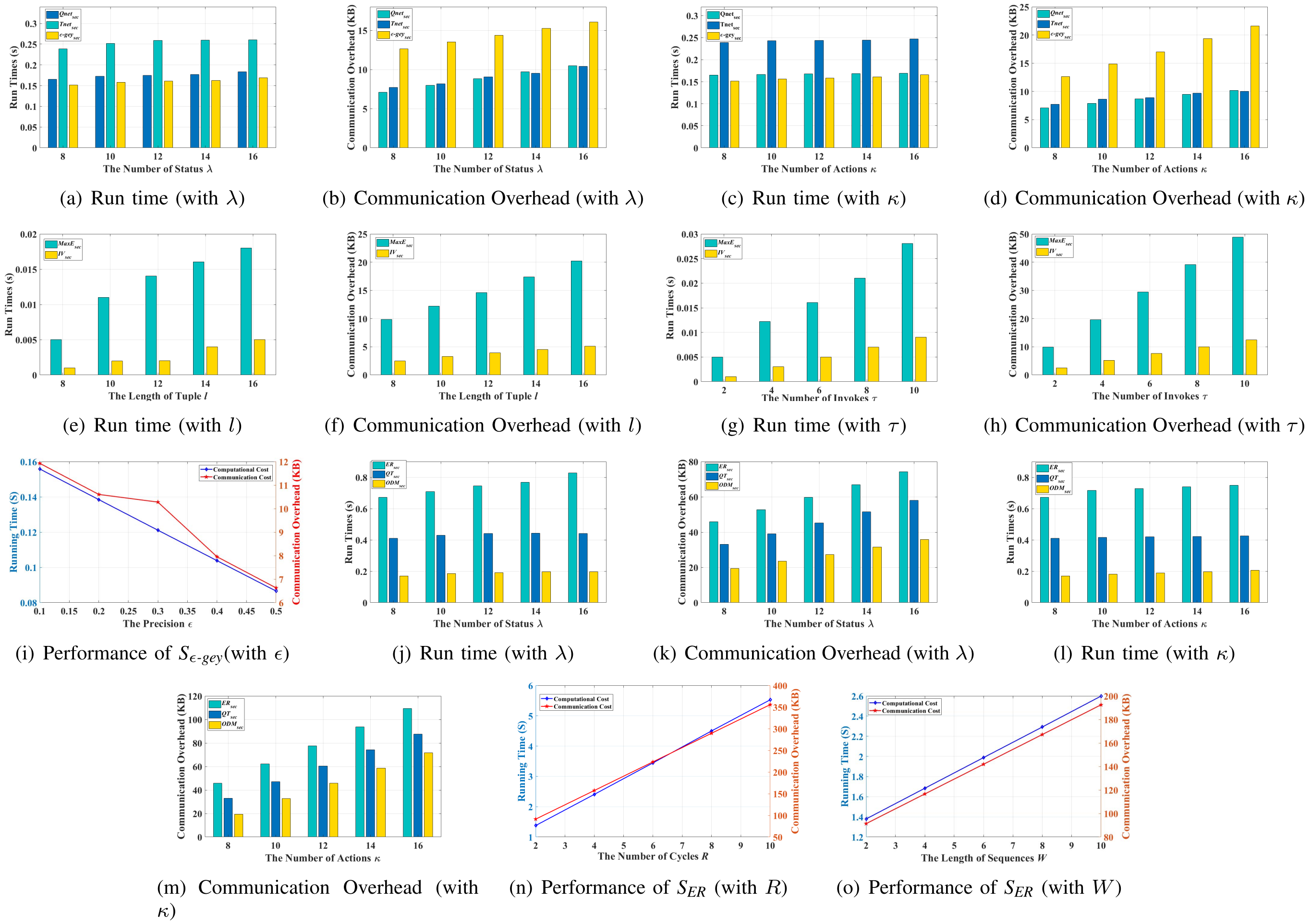
1) Sub-Protocol Performance Evaluation: For the S_{Qnet} and S_{Tnet} sub-protocols, we only need to consider the impact of the number of status λ and actions κ on the performance of the protocol. As shown in Fig. 10(a) to Fig. 10(d), the computational cost and communication overhead [10] of S_{Qnet} and S_{Tnet} increase with the number of λ and κ. The S_{Qnet} and S_{Tnet} protocols invoke S_{Com} and S_{Mul}, and the increase of λ and κ makes the computation of S_{Com} and S_{Mul} increase each time, which leads to an increase in overhead. In the S_{MaxE} and S_{IV} protocols, what depends on their performance is the length of the tuple. As the length of the tuple increases, the number of invokes of S_{Com} and S_{Mul} also increases. Besides, we also consider that in PrivacySignal, S_{MaxE} and S_{IV} are invoked multiple times. Therefore, we evaluate the performance of S_{MaxE} and S_{IV} at different tuple lengths l and number of invokes τ. As shown in Fig. 10(e) to Fig. 10(h), the computational overhead and communication overhead of S_{MaxE} and S_{IV} increase with the length of the tuple and the number of invokes. Finally, for the $S_{\epsilon\text{-}gey}$ protocol, we evaluate the performance of the protocol from two perspectives. In the first evaluation perspective, we consider the impact of different states and actions on the performance of $S_{\epsilon\text{-}gey}$ under the same ϵ. The $S_{\epsilon\text{-}gey}$ protocol invokes the S_{Qnet}, S_{MaxE}, and S_{IV} protocols. Furthermore, the increase in the number of λ and κ means that the tuple length in S_{MaxE} and S_{IV} increases, and we have previously evaluated the impact of the number of λ and κ on S_{Qnet}. Therefore, according to the Fig. 10(a) and Fig. 10(d), the computation and communication overhead of $S_{\epsilon\text{-}gey}$ increases with the number of λ and κ. The second evaluation angle is the performance of $S_{\epsilon\text{-}gey}$ with different ϵ. In theory, if the value of ϵ increases, the probability of invoking the S_{Qnet}, S_{MaxE}, and S_{MaxE} protocols becomes smaller, and the possibility of the RSU randomly selecting actions becomes larger. It means that $S_{\epsilon\text{-}gey}$ overhead reduces accordingly. As shown in Fig. 10(i), as ϵ increases, $S_{\epsilon\text{-}gey}$ calculation and communication overhead decrease.

2) PrivacySignal Performance Evaluation: Since S_{ER}, S_{QT} and S_{ODM} protocols compose of sub-protocols and some basic security protocols, we also consider the performance impact of λ and κ on these protocols. We can see from Fig. 10(j) to Fig. 10(m) that with the increase of λ and κ, the calculation overhead and communication overhead of the S_{ER}, S_{QT} and S_{ODM} protocols increase. It is consistent with the conclusions of the performance of the previously evaluated sub-protocols. Moreover, we also evaluate the impact of the number of cycles $\mathcal{R}$ and the number of sequences $\mathcal{W}$ on its performance. Increasing the number of $\mathcal{R}$ or $\mathcal{W}$ will increase the number of invokes to the sub-protocol, the computing overhead and communication overhead of S_{ER} will also increase accordingly. As shown in Fig. 10(n) and Fig. 10(o), as the $\mathcal{R}$ and $\mathcal{W}$ increase, the calculation and communication costs of the protocol S_{ER} increase. In practice, Q-networks require multiple iterations of training to get the optimal parameters. Therefore, for S_{QT}, we also evaluate the computation and communication overhead of S_{QT} at different training times $\mathcal{K}$.

B. Comparison With Other Schemes

1) Efficiency Comparison With Different Schemes: Prior to this, [14] provided a privacy-protected reinforcement learning scheme based on homomorphic encryption. Thus, to further evaluate the superiority of our system, we compare it

[10] The communication overhead is represented by the size of the message generated when the security protocol is running.



(a) Run time (with λ) (b) Communication Overhead (with λ) (c) Run time (with κ) (d) Communication Overhead (with κ)

(e) Run time (with l) (f) Communication Overhead (with l) (g) Run time (with τ) (h) Communication Overhead (with τ)

(i) Performance of $S_{\epsilon\text{-}gey}$(with ϵ) (j) Run time (with λ) (k) Communication Overhead (with λ) (l) Run time (with κ)

(m) Communication Overhead (with κ) (n) Performance of S_{ER} (with R) (o) Performance of S_{ER} (with W)

Fig. 10. Evaluation findings.

TABLE IV

RUN TIMES AND COMMUNICATION OVERHEAD COMPARISON WITH DIFFERENT SCHEMES

Procotol	Run Times (s)		Communication Overhead(KB)	
	Our scheme	[14]	Our scheme	[14]
S_{Qnet}	0.165	105.522	7.1	11.1
S_{MaxE}	0.003	0.171	4.9	8.9
S_{IV}	0.007	2.747	10.0	16.5
$S_{\epsilon\text{-}gey}$	0.152	56.481	12.6	19.7
S_{ER}	0.672	298.502	45.8	104.2
S_{QT}	0.411	134.668	33.0	61.1
S_{ODM}	0.181	62.652	19.4	40.7

TABLE V

COMPARISON OF AVERAGE WAITING TIME WITH DIFFERENT SCHEMES

	Average Waiting Time(s)	Time Reduction with BL
BL	35	N.A.
[4]	26	25.7%
Ours	27	22.8%

TABLE VI

COMPARISON OF AVERAGE WAITING TIME OF DIFFERENT SCHEMES DURING RUSH HOURS

	Average Waiting Time(s)	Time Reduction with BL
BL	45	N.A.
[4]	33	26.7%
Ours	35	22.2%

with a homomorphic encryption scheme [14]. As shown in TABLE IV, we compare the computational and communication costs of the four protocols from different perspectives. Homomorphic encryption requires complex exponential operations, and PrivacySignal avoids these computationally intensive operations. We can see from the comparison results that our four protocols are better than the [14] in different angles.

2) Comparison of Average Waiting Time With Different Schemes: Additionally, we compared the average waiting time of vehicles of different schemes.[11] The results can be found in Table V. In SUMO, a road traffic simulation experiment was conducted. We first simulated a conventional road scene in which the traffic flow of all lanes was basically the same.[12] BL

[11] Our simulation parameters and DQN parameters are set to the same as [4].

[12] It is assumed here that the vehicle arrival rate of each lane is 1/10 per second.

TABLE VII

COMPARATIVE SUMMARY

Scheme	Privacy-Preserving	Trusted Third Party	High Efficiency	Loss of Accuracy	Semi-honest Model
Liang [4]	×	×	×	−	−
Zang [26]	×	×	×	−	−
Cui [8]	✓	✓	×	✓	−
Liu [14]	✓	✓	×	✓	✓
Bita [27]	✓	×	×	×	✓
Ours	✓	×	✓	×	✓

is a baseline scheme, it usually sets a fixed time to control the traffic signals. Optimal is a DQN scheme based on non-privacy protection [4]. It can be seen from the simulation experiment results that, compared with the traditional traffic signal control scheme (BL), the optimized scheme using DQN can reduce the average waiting time of the vehicle by 25.7%. In particular, although our scheme incorporates privacy protection, it does not introduce calculation errors. The average waiting time of PrivacySignal is almost the same as that of the non-privacy protection optimal scheme. In order to further prove the above conclusions, we also simulated the average waiting time of vehicles during rush hour.[13] As shown in Table VI, compared with BL, the optimization scheme using DQN can reduce the average vehicle waiting time by 26.7%. At the same time, the average waiting time between the PrivacySignal and the non-privacy protection optimization scheme is only 2 seconds.

3) Comparative Summary: In TABLE VII, we summarize some of the features of other schemes and further compare them with that of the PrivacySignal system. For [4] and [26], reinforcement learning techniques are used to implement intelligent traffic signal systems, but user vehicle data privacy is not protected, which makes them at risk of leaking privacy in the application. Cui *et al.* [8] and Liu *et al.* [14] have considered the issue of user data privacy. However, the HE-based privacy protection framework [14] has large computational strength and storage space, and is inefficient in practical applications. Similarly, garbled circuit (GC) [27] also achieves privacy protection, but GC has an expensive computational overhead. As for [8], the anonymous technology provides only limited protection of users' privacy that leads to loss of critical information. In this sense, it is currently not ideal for application.

VIII. Related Work

For data privacy-protecting in machine learning. Liu *et al.* propose a toolkit for efficient and privacy-protected outsourcing calculation under multiple encryption keys (EPOM) [28]. With EPOM, a large number of users can securely outsource data to cloud servers for storage and computation. Later, they also propose an effective and privacy-preserving outsourcing rational number calculation framework (POCR) [29]. With POCR, users can securely outsource storage and processing of rational numbers to cloud servers without compromising the security of user data and calculation results. Recently, they proposed a fast and privacy-protected outsourced computation (LightCom) [30] framework in the cloud. Unlike the existing multi-server outsourcing computing model, users can use LightCom to securely implement data outsourcing storage and fast and secure data processing on a single cloud server. However, the main components of EPOM, POCR, and LightCom are homomorphic encryption. Its large amount of calculation is inappropriate for the real-time IoV environment. Recently, various confusion protocols have been proposed, like Chameleon [31], MiniONN [32], Gazelle [33] and ABY^3 [34]. Nevertheless, it is infeasible for them to expand because they use computationally intensive cryptographic primitives. Furthermore, Yao's protocol is used to implement a common method of privacy calculations. Deepsecure [27] based on GC is proposed for deep learning data privacy protection. Although Bita *et al.* claimed that Deepsecure reduced the running time by at least two orders of magnitude, Saleem *et al.* [35] pointed out that it is still not practical due to serious implementation problems in efficiency and reusability of GC.

Traffic signal control in the field of intelligent transportation systems has always been a hot research topic. Gao *et al.* solved the traffic signal dispatching problem in a heterogeneous traffic network by using a meta-heuristic algorithm [36]. Later, Ye *et al.* introduced the latest developments and research trends of traffic signal control based on model predictive control in transportation network coordination and control [37]. Due to the fixed-time traffic signal control method's inability to adapt to the dynamic traffic road conditions, more recent research attempts to use reinforcement learning algorithms to address signal control problems [3], [5], [26], [38]. Generally, these algorithms take the traffic condition on the road as the state and control the traffic signals as the action. These methods usually show better performance than fixed-time control methods. Methods [39], [40] design road information states as discrete values, such as vehicle position or number of waiting vehicles. However, discrete values require huge storage space, making these methods unsuitable for continuous state values. To tackle the unmanageable large space issue in previous methods, recent research [38], [41] proposed a deep Q-network method using continuous state representation. These studies learned a Q-function (for example, a deep neural network) to map state and action rewards. The status representation of these works is different, including manual functions (for example, vehicle queue length [38], [42], average delay [41], [43]). They are also different in terms of reward design, including average vehicle waiting time [43] and queue length [38].

IX. Conclusion

In this paper, we propose PrivacySignal, a new privacy-preserving traffic signal control for an intelligent transportation

[13] In our experiment, the arrival rate of vehicles in the lane from west to east is 2/10 per second, and the arrival rate of vehicles in other lanes is still 1/10 per second.

system. For experience replay, Q-network training, and optimal decision-making, we specifically design a series of interactive protocols to complete the corresponding computing tasks. To perform PrivacySignal sub-operations efficiently and securely, we also proposed five sub-protocols based on additional secret sharing, namely secure Q-network, secure target Q-network, secure maximum element selection, secure index value, and secure ϵ-greedy policy protocol. With PrivacySignal, the accuracy and efficiency of the decision are improved without leaking crucial data to honest but curious RSUs.

As a future research effort, we intend to further improve the efficiency of our system architecture. At the same time, we will also consider some other application scenarios (e.g., Traffic Flow Prediction [44], [45] and Autonomous Vehicles [46]).

References

[1] H. Wei, G. Zheng, V. Gayah, and Z. Li, "Recent advances in reinforcement learning for traffic signal control: A survey of models and evaluation," *ACM SIGKDD Explor. Newslett.*, vol. 22, no. 2, pp. 12–18, Jan. 2021.

[2] A. Haydari and Y. Yilmaz, "Deep reinforcement learning for intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 11–32, Jan. 2022.

[3] N. Kumar, S. S. Rahman, and N. Dhakad, "Fuzzy inference enabled deep reinforcement learning-based traffic light control for intelligent transportation system," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 4919–4928, Aug. 2021.

[4] X. Liang, X. Du, G. Wang, and Z. Han, "A deep reinforcement learning network for traffic light cycle control," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1243–1253, Feb. 2019.

[5] T. Chu, J. Wang, L. Codecà, and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1086–1095, Mar. 2020.

[6] Z. Yan, J. Xue, and C. W. Chen, "Prius: Hybrid edge cloud and client adaptation for HTTP adaptive streaming in cellular networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 1, pp. 209–222, Jan. 2017.

[7] X. Wang *et al.*, "Privacy-preserving content dissemination for vehicular social networks: Challenges and solutions," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1314–1345, 2nd Quart., 2018.

[8] J. Cui, J. Wen, S. Han, and H. Zhong, "Efficient privacy-preserving scheme for real-time location data in vehicular ad-hoc network," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3491–3498, Oct. 2018.

[9] Z. Tan, C. Wang, C. Yan, M. Zhou, and C. Jiang, "Protecting privacy of location-based services in road networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 10, pp. 6435–6448, Oct. 2021.

[10] P. Asuquo, H. Cruickshank, J. Morley, C. P. A. Ogah, A. Lei, W. Hathal, S. Bao, and Z. Sun, "Security and privacy in location-based services for vehicular and mobile communications: An overview, challenges, and countermeasures," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4778–4802, Dec. 2018.

[11] A. Narayanan and V. Shmatikov, "Myths and fallacies of 'personally identifiable information'," *Commun. ACM*, vol. 53, no. 6, pp. 24–26, 2010.

[12] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 5, pp. 1333–1345, May 2018.

[13] A. Alanwar *et al.*, "PrOLoc: Resilient localization with private observers using partial homomorphic encryption," in *Proc. 16th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, Apr. 2017, pp. 41–52.

[14] Y. Liu, Z. Ma, X. Liu, S. Ma, and K. Ren, "Privacy-preserving object detection for medical images with faster R-CNN," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 69–84, 2022.

[15] T. Araki, J. Furukawa, Y. Lindell, A. Nof, and K. Ohara, "High-throughput semi-honest secure three-party computation with an honest majority," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 805–817.

[16] K. Huang, X. Liu, S. Fu, D. Guo, and M. Xu, "A lightweight privacy-preserving CNN feature extraction framework for mobile sensing," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 3, pp. 1441–1455, May/Jun. 2020.

[17] Z. Ma, Y. Liu, X. Liu, J. Ma, and F. Li, "Privacy-preserving outsourced speech recognition for smart IoT devices," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8406–8420, Oct. 2019.

[18] Z. Ma, Y. Liu, X. Liu, J. Ma, and K. Ren, "Lightweight privacy-preserving ensemble classification for face recognition," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5778–5790, Jun. 2019.

[19] A. Paverd, A. Martin, and I. Brown, "Modelling and automatically analysing privacy properties for honest-but-curious adversaries," Dept. Comput. Sci., Univ. Oxford, Oxford, U.K., Tech. Rep., 2014. Accessed: Oct. 11, 2016. [Online]. Available: https://www.cs.ox.ac.uk/people/andrew.paverd/casper/casper-privacy-report.pdf

[20] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[21] Y. Lindell and B. Riva, "Blazing fast 2PC in the offline/online setting with security for malicious adversaries," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2015, pp. 579–590.

[22] D. Beaver, "Efficient multiparty protocols using circuit randomization," in *Proc. Annu. Int. Cryptol. Conf.* Berlin, Germany: Springer, 1991, pp. 420–432.

[23] I. Damgård, M. Fitzi, E. Kiltz, J. B. Nielsen, and T. Toft, "Unconditionally secure constant-rounds multi-party computation for equality, comparison, bits and exponentiation," in *Proc. Theory Cryptogr. Conf.* Berlin, Germany: Springer, 2006, pp. 285–304.

[24] D. Bogdanov, S. Laur, and J. Willemson, "Sharemind: A framework for fast privacy-preserving computations," in *Proc. Eur. Symp. Res. Comput. Secur.* Berlin, Germany: Springer, 2008, pp. 192–206.

[25] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of sumo-simulation of urban mobility," *Int. J. Adv. Syst. Meas.*, vol. 5, nos. 3–4, pp. 128–138, 2012.

[26] X. Zang, H. Yao, G. Zheng, N. Xu, K. Xu, and Z. Li, "Metalight: Value-based meta-reinforcement learning for traffic signal control," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 1, pp. 1153–1160.

[27] B. D. Rouhani, M. S. Riazi, and F. Koushanfar, "DeepSecure: Scalable provably-secure deep learning," in *Proc. 55th ACM/ESDA/IEEE Des. Autom. Conf. (DAC)*, Jun. 2018, pp. 1–6.

[28] X. Liu, R. H. Deng, K.-K. R. Choo, and J. Weng, "An efficient privacy-preserving outsourced calculation toolkit with multiple keys," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 11, pp. 2401–2414, Nov. 2016.

[29] X. Liu, K.-K. R. Choo, R. H. Deng, R. Lu, and J. Weng, "Efficient and privacy-preserving outsourced calculation of rational numbers," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 1, pp. 27–39, Feb. 2018.

[30] X. Liu, R. H. Deng, P. Wu, and Y. Yang, "Lightning-fast and privacy-preserving outsourced computation in the cloud," 2019, *arXiv:1909.12540*.

[31] M. S. Riazi, C. Weinert, O. Tkachenko, E. M. Songhori, T. Schneider, and F. Koushanfar, "Chameleon: A hybrid secure computation framework for machine learning applications," in *Proc. Asia Conf. Comput. Commun. Secur.*, May 2018, pp. 707–721.

[32] J. Liu, M. Juuti, Y. Lu, and N. Asokan, "Oblivious neural network predictions via MiniONN transformations," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 619–631.

[33] C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan, "GAZELLE: A low latency framework for secure neural network inference," in *Proc. 27th Security Symp.*, 2018, pp. 1651–1669.

[34] P. Mohassel and P. Rindal, "ABY^3: A mixed protocol framework for machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2018, pp. 35–52.

[35] A. Saleem, A. Khan, F. Shahid, M. M. Alam, and M. K. Khan, "Recent advancements in garbled computing: How far have we come towards achieving secure, efficient and reusable garbled circuits," *J. Netw. Comput. Appl.*, vol. 108, pp. 1–19, Apr. 2018.

[36] K. Gao, Y. Zhang, R. Su, F. Yang, P. N. Suganthan, and M. Zhou, "Solving traffic signal scheduling problems in heterogeneous traffic network by using meta-heuristics," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 9, pp. 3272–3282, Sep. 2019.

[37] B.-L. Ye *et al.*, "A survey of model predictive control methods for traffic signal control," *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 3, pp. 623–640, May 2019.

[38] L. Li, Y. Lv, and F.-Y. Wang, "Traffic signal timing via deep reinforcement learning," *IEEE/CAA J. Autom. Sinica*, vol. 3, no. 3, pp. 247–254, Jul. 2016.

[39] M. Abdoos, N. Mozayani, and A. L. C. Bazzan, "Holonic multi-agent system for traffic signals control," *Eng. Appl. Artif. Intell.*, vol. 26, nos. 5–6, pp. 1575–1587, 2013.

[40] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad, "Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): Methodology and large-scale application on downtown toronto," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1140–1150, Sep. 2013.

[41] E. Van der Pol and F. A. Oliehoek, "Coordinated deep reinforcement learners for traffic light control," *Proc. Learn., Inference Control Multi-Agent Syst. (NIPS)*, 2016, pp. 1–8.

[42] P. Mannion, J. Duggan, and E. Howley, "An experimental review of reinforcement learning algorithms for adaptive traffic signal control," in *Autonomic Road Transport Support Systems*. Cham, Switzerland: Springer, 2016, pp. 47–66, doi: 10.1007/978-3-319-25808-9_4.

[43] W. Genders and S. Razavi, "Using a deep reinforcement learning agent for traffic signal control," 2016, *arXiv:1611.01142*.

[44] M. Lv, Z. Hong, L. Chen, T. Chen, T. Zhu, and S. Ji, "Temporal multi-graph convolutional network for traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3337–3348, Jun. 2021.

[45] H. Zheng, F. Lin, X. Feng, and Y. Chen, "A hybrid deep learning model with attention-based conv-LSTM networks for short-term traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 6910–6920, Nov. 2021.

[46] A. Rasouli and J. K. Tsotsos, "Autonomous vehicles that interact with pedestrians: A survey of theory and practice," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 900–918, Mar. 2020.

Zuobin Ying (Member, IEEE) received the Ph.D. degree in computer architecture from Xidian University, Xi'an, China, in 2016. From 2019 to 2021, he was a Research Fellow with the Nanyang Technological University of Singapore. He is currently an Assistant Professor with the Faculty of Data Science, City University of Macau, Macau, China. His research interests include cloud security, applied cryptography, and blockchain.

Shuanglong Cao (Student Member, IEEE) is currently a Research Student with the School of Computer Science and Technology, Anhui University, Hefei, China. His research interests include intelligent transportation systems, applied cryptology, secure multiparty computation, and machine learning.

Ximeng Liu (Senior Member, IEEE) received the B.Sc. degree in electronic engineering and the Ph.D. degree in cryptography from Xidian University, Xi'an, China, in 2010 and 2015, respectively. He is currently a Full Professor with the College of Mathematics and Computer Science, Fuzhou University. He was a Research Fellow with the School of Information System, Singapore Management University, Singapore. He has published more than 250 papers on the topics of cloud security and big data security, including papers in IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, and IEEE INTERNET OF THINGS JOURNAL. His research interests include cloud security, applied cryptography, and big data security. He was awarded the Minjiang Scholars Distinguished Professor, a Qishan Scholars at Fuzhou University, and the ACM SIGSAC China Rising Star Award in 2018.

Zhuo Ma (Member, IEEE) received the Ph.D. degree in computer architecture from Xidian University, Xi'an, China, in 2010. He is currently an Associate Professor with the School of Cyber Engineering, Xidian University. His research interests include cryptography, machine learning in cyber security, and the Internet of Things security.

Jianfeng Ma (Member, IEEE) received the B.S. degree in mathematics from Shaanxi Normal University, China, in 1985, and the M.E. and Ph.D. degrees in computer software and communications engineering from Xidian University, China, in 1988 and 1995, respectively. From 1999 to 2001, he was a Research Fellow with the Nanyang Technological University of Singapore. He is currently a Professor with the School of Computer Science, Xidian University. His current research interests include distributed systems, computer networks, and information and network security.

Robert H. Deng (Fellow, IEEE) is currently the AXA Chair Professor of cybersecurity and the Director of the Secure Mobile Centre, School of Information Systems, Singapore Management University (SMU). His research interests include data security and privacy, cloud security, and the Internet of Things security. He received the Outstanding University Researcher Award from the National University of Singapore, the Lee Kuan Yew Fellowship for Research Excellence from SMU, and the Asia–Pacific Information Security Leadership Achievements Community Service Star from International Information Systems Security Certification Consortium. He is the Steering Committee Chair of the ACM Asia Conference on Computer and Communications Security. His professional contributions include an extensive list of positions in several industry and public services advisory boards, editorial boards, and conference committees. These include the editorial boards of *IEEE Security and Privacy Magazine*, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, and *Journal of Computer Science and Technology*.