1-2024

PIAS: Privacy-Preserving Incentive Announcement System based on Blockchain for Internet of Vehicles

Yonghua ZHAN

Yang YANG

Hongju CHENG

Xiangyang LUO

Zhuangshuang GUAN

See next page for additional authors

PIAS: Privacy-preserving Incentive Announcement System based on Blockchain for Internet of Vehicles

Yonghua Zhan, Yang Yang* (*Senior Member, IEEE*), Hongju Cheng, Xiangyang Luo, Zhangshuang Guan, Robert H. Deng (*Fellow, IEEE*)

Abstract—More vehicles are connecting to the Internet of Things (IoT), transforming Vehicle Ad hoc Networks (VANETs) into the Internet of Vehicles (IoV), providing a more environmentally friendly and safer driving experience. Vehicular announcement networks show promise in vehicular communication applications. However, two major issues arise when establishing such a system. Firstly, user privacy cannot be guaranteed when messages are forwarded anonymously, thus the reliability of these messages is in question. Secondly, users often lack interest in responding to announcements. To address these problems, we introduce a Blockchain-based incentive announcement system called PIAS. This system enables anonymous message commitment in a semi-trusted environment and encourages witnesses to respond to requests for traffic information. Additionally, PIAS uses blockchain accounts as identities to participate in the system with incentives, ensuring privacy in anonymous announcements. PIAS successfully protects the privacy of participants and motivates witnesses to respond to requests. Furthermore, our assessment of security and compatibility shows that PIAS can maintain privacy and incentivization while being compatible with both the Bitcoin and Ethereum blockchains. Further evaluation has confirmed the system's efficiency in terms of performance.

Index Terms—Internet of Vehicles, Blockchain, Incentive Mechanism, Fair Payment, Privacy Preservation.

1 Introduction

According to recent research [1], an increasing number of devices will be connected to the Internet, with vehicles making up a significant portion. As a result, applications for vehicular communication are becoming increasingly important. The Internet of Things (IoT) is leading to more vehicles becoming connected to it, leading to the transformation of traditional Vehicle Ad-hoc Networks into the Internet of Vehicles (IoV). This evolution is a result of the new era of the IoT. With the rapid advancement of computing and 5G technologies, IoV has gained significant commercial and research interest. It focuses on the exchange of information among vehicles, humans, and roadside units (RSU). Furthermore, IoV improves vehicle safety, promotes eco-friendliness by reducing driving risks, and enhances transportation efficiency, ultimately resulting in decreased public resource expenditure.

Attention to privacy concerns related to data has increased among the public, leading to two prominent issues in the vehicular announcement network. Firstly, ideally, most messages sent through the network should be forwarded anonymously to protect the personal information of users, such as their vehicle numbers and identities. However, anonymous forwarding does not guarantee the reliability of the messages. Secondly, users often lack the motivation to respond to these messages. Without incentives to reply, their willingness to do so decreases. We consider a hypothetical scenario where a requester wants information about an unfamiliar location without compromising their privacy. Some witness vehicles near the location have seen an accident on the road and plan to inform other drivers through an announcement, expecting rewards in return.

Prior research has utilized both threshold authentication and group signature methods to tackle the initial challenge. However, these methods are constrained by significant workload and a lack of incentives for message responses from users. Recently, there has been a growing interest in blockchain technologies due to their decentralized nature. Blockchain technology is based on a ledger-based decentralized system that records transaction histories on a shared ledger, maintained by a distributed network of mutually untrusting nodes. Two widely used blockchain systems are Bitcoin and Ethereum. With blockchain technology, it is possible to transfer messages anonymously and directly between participating parties without the need for third-party intermediaries, through a transaction or smart contract.

This paper proposes a novel scheme for addressing the two primary challenges associated with establishing an anonymous vehicular announcement system. We present a privacy-preserving incentive announcement system, called PIAS, which utilizes Blockchain technology to create an effective incentive mechanism while ensuring privacy preser-

Y. Zhan is with College of Computer and Data Science, Fuzhou University, Fuzhou, China, 350116. (email: zhanyonghua@126.com) Y. Yang is with College of Computer and Data Science, Fuzhou University, Fuzhou, China, 350116, and School of Computing and Information Systems, Singapore Management University, Singapore 188065. (email: yang.yang.research@gmail.com) H. Cheng is with College of Computer and Data Science, Fuzhou University, Fuzhou, China, 350116. (email: cscheng@fzu.edu.cn). X. Luo is with State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou. (email: luoxy_ieu@sina.com) Z. Guan is with the School of Computer Science and Technology, Zhejiang University, Hangzhou, 310027, China (e-mail: aguanzs@163.com) R. Deng is with School of Computing and Information Systems, Singapore Management University, Singapore 188065. (email: robertdeng@smu.edu.sg)
Corresponding author: Yang Yang.

vation. Our proposed system ensures that even in the presence of malicious users, compensation will be given to honest participants.

1.1 Our Contributions

We present PIAS, a privacy-preserving incentive announcement system based on blockchain for Internet of Vehicles, with the aim of achieving anonymity and ensuring fairness of payment against malicious users. PIAS eliminates the need for a third party to achieve fair payment, which may or may not be trusted. The paper makes three main contributions:

1) The paper presents the system model, system definition, adversary model, and design goals of PIAS. The design details are also described. Our method guarantees the reliability of anonymous messages in PIAS by requiring more than one witness to provide the information. We prove that PIAS achieves our goals, such as privacy-preserving and robust fairness, based on the collision-resistance of hash functions and the unforgeability of the ECDSA. In PIAS, fairness implies that each party is forced to behave honestly. Our system ensures that the participant witness either earns the message fee or loses his guarantee based on whether his message is the honest majority or not.
2) The paper presents a compatibility analysis indicating that PIAS is compatible with both the Bitcoin blockchain and the Ethereum blockchain, which are the most widely used blockchains. The implementation of PIAS can gather traffic information and process payments using only basic scripts, without requiring more powerful scripts such as string concatenation scripts supported by the Bitcoin blockchain.
3) We conducted a performance analysis of the scheme and evaluated the expenses on the Ethereum test network.

1.2 Related Work

In recent years, vehicular communication networks have gained significant attention from both academia and industry due to the increasing number of vehicles connected to the Internet of Things (IoT). These networks have the potential to enhance driving experiences by providing increased safety, efficiency, and comfort [2], [3]. The advancements in computing and 5G technologies have transformed traditional Vehicle Adhoc Networks (VANETs) into the Internet of Vehicles (IoV), with the ultimate goal of integrating humans, vehicles, objects, and the environment to reduce social costs and improve transportation efficiency [4].

However, the increasing concerns regarding data privacy [5]-[7] have led to privacy issues surrounding the development of a reliable vehicular communication network. Since user messages often include sensitive information like vehicle numbers and identities, it is crucial for these messages to be forwarded anonymously. Nevertheless, maintaining anonymity poses challenges in ensuring the credibility of the messages. Moreover, if a message discloses a user's true identity, it could potentially compromise both their location and identity privacy.

To address challenges related to anonymity, prior research has utilized threshold authentication and group signatures. In threshold authentication protocols, a message is only accepted by the receiver if it has been validated by a threshold number of vehicles [8]. However, anonymous forwarding of messages can result in issues such as Sybil and DDoS attacks [9]-[12]. The threshold value can be fixed system-wide [8] or user-controlled [17]. Shao et al. [17] proposed a customizable group signature scheme for achieving threshold authentication in various scenarios. Wu et al. [13] utilized context-aware threshold authentication and message-linkable group signatures to detect malicious users. Chen et al. [14] introduced a threshold anonymous announcement scheme based on direct anonymous attestation and one-time anonymous authentication. Zhang et al. [15] proposed a one-time identity-based aggregate signature to address linkability. Gao et al. [16] suggested a threshold-based authentication method using group signatures to ensure the security and privacy of vehicles while enhancing verification speed and preventing DoS attacks. Azees et al. [18] introduced an anonymous authentication scheme to block malicious vehicles from entering the system. Cui et al. [19] developed a vehicular network framework for multi-cloud scenarios. To tackle the high computation and communication costs in anonymous authentication, Cheng et al. [20] recommended an efficient method for anonymous authentication using a privacy-preserving reliability evaluation algorithm to improve mutual authentication efficiency and ensure the reliability of sensing vehicles. However, these solutions do not offer incentives for participants to respond and forward messages, which could potentially lead to a lack of motivation to do so.

However, in the aforementioned scenarios, either incentives are neglected or an incentive structure is utilized that requires a reliable third party to address the problem of motivation. To address these issues, blockchain technologies have been taken into consideration [21]. Blockchain-based solutions offer significant advantages over traditional payment technologies, primarily decentralization and anonymity. Andrychowicz et al. [22] introduced a secure multiparty protocol based on the Bitcoin blockchain, using a timed commitment scheme based on Bitcoin to design a fair protocol. Similarly, Zhang et al. [23] proposed a fair protocol based on similar principles. Li et al. [24] developed an incentive announcement network based on blockchain to motivate users to share traffic information, but the system requires a trusted third-party to trace malicious users. In addition, blockchain technology has been used in other applications [25]-[28].

Honest majority is a commonly used assumption in distributed systems and cryptography, used to describe a situation where the majority of participants in the system are honest and trustworthy. Rabin et al. [30] proposed a verifiable secret sharing protocol based on honest majority. Araki et al. [31] proposed a secure three-party computation protocol with honest majority. Dalskov et al. [32] proposed a four-party secure computation protocol based on honest majority, which has active security and is simpler due to not relying on function-dependent preprocessing. Chida et al. [33] introduced a multiparty computation protocol based on honest majority for secure computation of any function

represented by arithmetic circuits, with very high computation speed. In our paper, we propose a privacy-preserving incentive announcement system that utilizes blockchain technology. Users can anonymously send messages and incentivize others to share traffic information without the need for a trusted third-party, utilizing either the Bitcoin or Ethereum blockchain.

1.3 Organization

The remaining part of the article is structured as follows. Section 2 provides some preliminary information. Our system model, definitions, adversary model, and design objectives are outlined in Section 3. The details of the construction are presented in Section 5. Section 6 presents security and performance analysis. Finally, Section 7 concludes the paper.

2 PRELIMINARIES

Table 1 presents the notations used in this paper. The paper also gives a brief explanation of the concepts of blockchain transactions and Bitcoin-based timed commitments.

TABLE 1: Notations

R	Requester	r_R	The secret of R
W_i	Witness	t	A time lock
$RSUs$	Roadside units	$Majority(a_i, A)$	Check if a_i is a major element in set A
H	Hash function		
h_S	The hash value $H(r_S)$	$Reveal(MSG)$	Check if all elements in set MSG are revealed
pk_A/sk_A	ECDSA key pair of A		
msg_i	Witness message of W_i		Maximal delay between broadcasting a transaction and including it on the blockchain
MSG	Witness message set	max	
Tx	Transaction in Blockchain		
σ_{Tx}	Signature of Tx	₿	Indicates the Bitcoin currency

2.1 Blockchain and Transaction

Blockchain is the core technology of cryptocurrencies, where the longest chain serves as the valid blockchain documenting transaction history on a shared ledger managed by a decentralized network of untrusted nodes. In the following, we use the Bitcoin currency system to describe the blockchain.

The address, which is a hash of a public key of an ECDSA signature pk, plays a crucial role in the Bitcoin system. For simplicity, we refer to pk as the address and assume that a user A possesses a public-secret key pair (pk_A, sk_A). Let $\sigma = \mathsf{sig}_A(m)$ represent the ECDSA signature on a message m, and let $\mathsf{ver}_A(m, \sigma)$ be the output (true or false) of the verification of the ECDSA signature σ on the message m relative to pk_A. A Bitcoin transaction, denoted as Tx_x, consists of an input script and an output script. The time-lock, represented by t, specifies the time at which the transaction Tx_x becomes valid. The body of Tx_x, denoted as $[\mathsf{Tx}_x]$ (i.e., Tx_x excluding the input script), must satisfy the condition that when time t is reached, every output script evaluates to true for Tx_x to be valid. We use the Bitcoin scripting language to compose scripts, which is a non-Turing complete, stack-based language. In Figure 1, user A aims to transfer d ₿ from Tx_2 to user B after time t, and the output script involves ECDSA signature verification. To simplify matters, we assume that transaction fees are negligible.

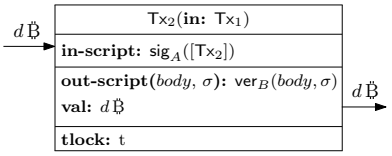


Fig. 1: A Transaction Example

2.2 Bitcoin-based Timed Commitment

Andrychowicz [22] proposed a bitcoin-based timed commitment scheme, denoted by $\mathsf{CS}(R, W, d, t, s)$, to counter malicious behaviors. The scheme requires R, the requester, to act as a committer and send a deposit of value d ₿ to W, the witness. R commits to a secret s, which must be revealed before a specific time t to redeem the deposit. Otherwise, W will redeem the deposit after time t. The scheme comprises the commitment phase $\mathsf{CS.Commit}(R, W, d, t, s)$, the opening phase $\mathsf{CS.Open}(R, W, d, t, s)$, and the punishment phase $\mathsf{CS.Fine}(R, W, d, t, s)$. If the honest committer reveals the commitment before time t, the punishment phase will not occur. Figure 2 illustrates the timed commitment transactions, with omitted arguments denoted by $\perp$, and H being a hash function.

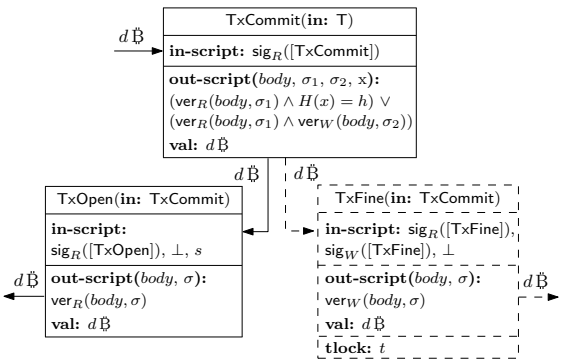


Fig. 2: Bitcoin-based Timed Commitment Transactions

3 SYSTEM MODEL AND SECURITY MODEL

3.1 System Model

Privacy-preserving **i**ncentive **a**nnouncement **s**ystem (PIAS) contains roadside units (RSUs), vehicles and blockchain, where the system scenario is depicted in Fig. 3. RSUs are communication devices placed along the road that offer connectivity support and information to vehicles as they pass by. During a particular incentive announcement scenario, vehicles are classified as either Requesters (R) or Witnesses (W), and their roles alternate for different requests.

- ***Requester*** (R) is willing to pay remuneration to get the traffic information in the corresponding area. After requesting witness messages, R depends on RSUs to collect them. Once R receives authentic traffic information, it compensates the RSUs for their services and pays the witnesses for their messages if they are genuine.
- ***Witness*** (W) is a driver who is present in the vicinity of the requested location and provides a witness message to R, for which they receive a message fee if it is

deemed authentic. However, W will be penalized if found to be cheating. To guarantee the credibility of witness messages, the system mandates that a group of witnesses $(W_1, \cdots, W_n)$ participate in the request and the majority of witness messages are considered genuine (where n is an odd number).

- **Roadside units** (RSUs) broadcast R's request and earn the service fee by gathering the witness messages of W.
- **Blockchain** records the transactions, and ensures the fairness of the incentive announcement.

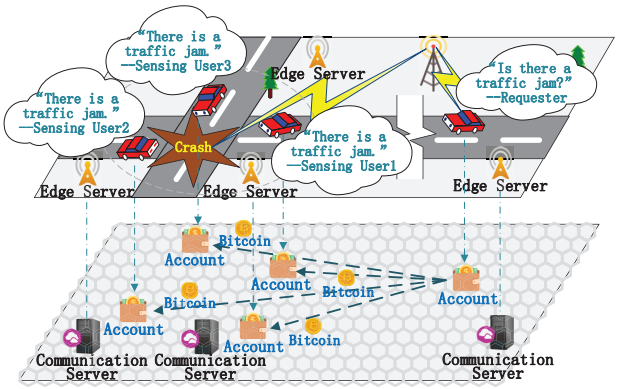


Fig. 3: System Scenario

3.2 Security Model

Assuming that Roadside Units (RSUs) faithfully broadcast requests, collect, and forward messages, they are expected not to collude with any other parties, including requesters and witnesses.

The system operates under the assumption of an honest majority, where most witnesses are expected to act truthfully, adhering to the protocol and refraining from intentionally transmitting incorrect messages or engaging in destructive behaviors. This assumption necessitates that witnesses avoid collusion with more than half of the total witnesses.

Moreover, there is a mutual distrust between requesters and witnesses, with both parties potentially being malicious. Specifically, a malicious requester, denoted as R, seeks to exploit real-time traffic information without remunerating the service or message fees, while a malicious witness, denoted as W, endeavors to receive message fees from R without providing genuine traffic information.

3.3 Attack Model

Furthermore, PIAS does not necessitate the use of private channels. Therefore, potential security threats such as sybil attacks, denial-of-service (DoS) attacks, replay and forgery attacks, man-in-the-middle (MitM) attacks, eavesdropping, and malleability attacks must be considered. These attacks aim to compromise the availability, privacy, and fairness of the PIAS system.

1) *Sybil attack*: Malicious requesters may forge many normal requesters to forward requests.[11]
2) *Denial of service attack*: The system can be subjected to a denial of service attack by malicious requesters who persistently send requests to RSUs.
3) *Replay attack*: An adversary may replay existing legitimate messages received from other requesters.
4) *Forgery attack*: An adversary may forge a request of other requesters.
5) *Man-in-the-middle attack*: The request sent from the requester to the RSUs can be subject to a man-in-the-middle attack by an adversary, who can alter the request's contents, including remuneration and destination values.
6) *Eavesdropping attack*: It is possible for an adversary to listen in on the public channel in order to obtain transaction information before it is recorded on the blockchain.
7) *Malleability attack*: Without changing the semantics, an adversary may attempt to invalidate certain transactions by altering their hash values.

3.4 Design Goals

1) *Decentralization*: Payment for PIAS is conducted in a decentralized environment without any involvement of third parties, whether they are trustworthy or not.
2) *Enthusiasm*: Witnesses are incentivized by PIAS to respond to requests.
3) *Reliability*: Transactions are secure and cannot be tampered with or forged by adversaries.
4) *Privacy preserving*: Requesters and witnesses' identities are not disclosed in requests and transactions.
5) *Soundness*: Assuming truthfulness from both requester and witnesses, the requester can access traffic information while witnesses receive messaging fees.
6) *Robust fairness*: The system ensures fairness for the requester, witnesses, and RSUs. Dishonest witnesses cannot receive messaging fees without providing genuine messages, and deceitful requesters face challenges in accessing traffic information without paying both service and messaging fees.
7) *Compatibility*: Our system is designed to work with blockchains like Bitcoin and Ethereum. Given the expressive nature of the scripts used on these blockchains, our system's compatibility with Bitcoin implies compatibility with Ethereum.

4 PIAS Overview

PIAS has six phases, which are described in Fig. 4. The first five phase are included as regular operations, and the *claim phase* is invoked when some witness message is not revealed or some witness is discovered cheating.

In ***setup phase***, entities generate their public and secret key pairs, and all participants create unredeemed transactions for subsequent procedures.

In ***request phase***, The traffic request for the relevant area is sent to RSUs by R and, at the same time, R creates a blockchain transaction $\mathsf{PutMoney}_0^R$ containing the service fee for RSUs. Upon examining both the request and $\mathsf{PutMoney}_0^R$, RSUs disseminate the request within the corresponding area.

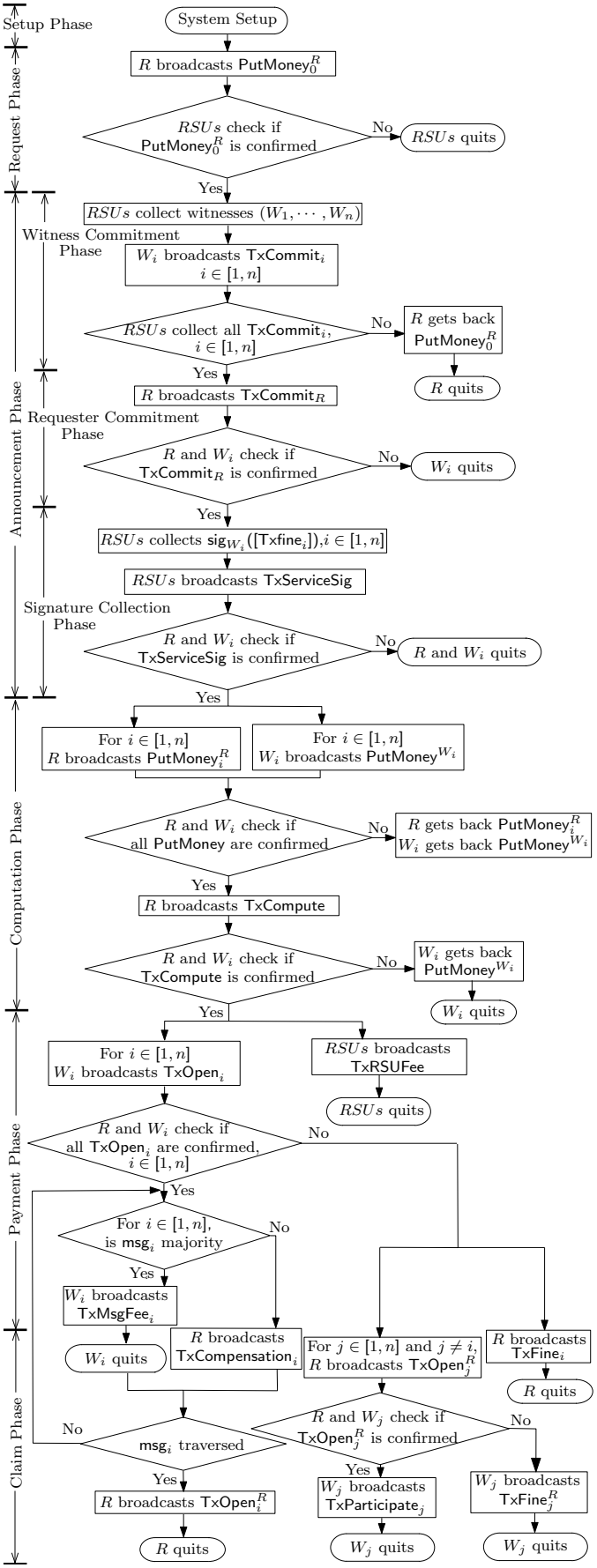


Fig. 4: System Workflow

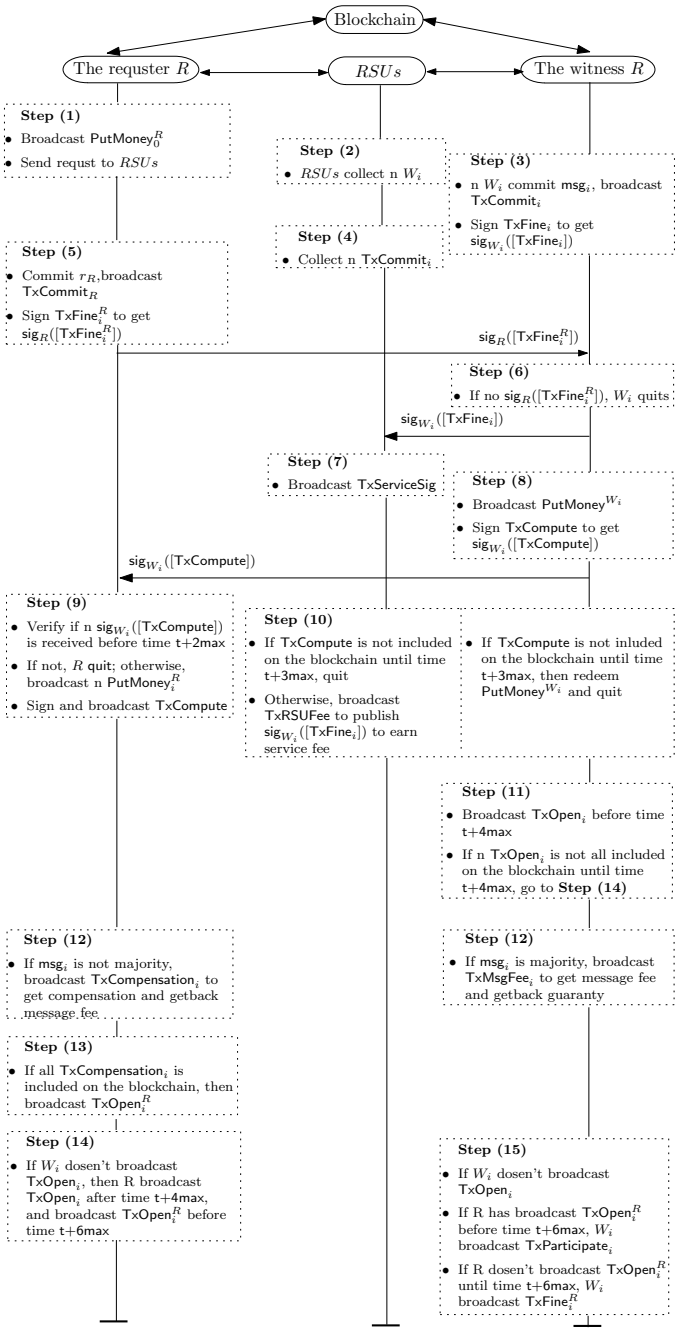


Fig. 5: Operations in Time Sequence

Announcement phase (for witness messages collection) composes three procedures: *witness commitment*, *requester commitment* and *signature collection*.

1) *Witness Commitment Phase*: To commit to the authenticity of the witness message, each witness W_i creates a commitment transaction $\mathsf{TxCommit}_i$ and broadcasts it on the blockchain. RSUs should collect all the commitments from witnesses $(W_1, \cdots, W_n)$ within a period of time.
2) *Requester Commitment Phase*: Once the witness message commitments are gathered by RSUs, R makes a commitment $\mathsf{TxCommit}_R$. If R refuses to pay the message fee to the honest witnesses and behaves maliciously,

then each honest W_i can claim the compensation by initiating the TxFine_i^R transaction after a specified period.

3) *Signature Collection Phase*: Each witness W_i signs the penalty transaction TxFine_i^R and sends the signature $Sig_{W_i}(\mathsf{TxFine}_i^R)$ to RSUs. Once all the signatures are collected, RSUs store the hash of signatures on the blockchain.

In the ***computation phase***, requester R and witnesses $(W_1, \cdots, W_n)$ jointly establish a transaction $\mathsf{TxCompute}$, which temporarily freezes the message fees from R, and guarantees from witnesses. To ensure fair payment, we design a series of contracts to determine the computation rules for different scenarios. It guarantees that: 1) RSUs obtain the service fee no matter if R or W_i misbehaves in the following procedures; 2) If some malicious witness refuses to acknowledge the witness message, R can either acquire authentic traffic information during the *payment phase* or receive adequate reimbursement in the *claim phase*; 3) honest witnesses can get remuneration for providing true information; 4) malicious witnesses will lose money as punishment.

In the ***payment phase***, honest witnesses earn the message fee, and RSUs earn the service fee.

- *Service fee payment*: RSUs earn the service fee by providing all the signatures ($Sig_{W_i}(\mathsf{TxFine}_i^R)$) of witnesses in the *signature collection phase*.
- *Message fee payment*: To get the deposit back, all witnesses reveal their commitments before a specific time. If the committed traffic message is real (computed as the majority of all messages), the witness will earn the message fee.

PIAS proceeds to ***claim phase*** if at least one witness does not reveal the traffic message or provides false message. The different scenarios are discussed below.

- *All committed traffic messages are revealed*. In this scenario, all the witnesses reveal their messages, and partial dishonest witnesses provides false message. 1) During the *payment phase*, the honest witnesses will receive the message fee of R in addition to having their deposit returned. 2) The dishonest witnesses cannot get the service fee, and lose the deposit as penalty in *claim phase*. 3) R will get the deposit of dishonest witnesses as compensation. Meanwhile, R gets back his own deposit.
- *Partial committed traffic messages are revealed*. If any witness doesn't reveal the message, the computation transaction cannot calculate the majority message. Then, the malicious witness will pay for the other witnesses and R in *claim phase*. 1) The witness who reveals the traffic message will get money from R as participate fee in *claim phase*. 2) The malicious witnesses will lose the deposit as penalty in *claim phase*. 3) R will get the deposit of dishonest witnesses as compensation. Meanwhile, R gets back his own deposit.

5 PIAS: Concrete Construction

A description of the concrete construction of PIAS is provided in this section. The system's workflow can be viewed in Fig. 4 and clearly depicted in Fig.5 and 6.

5.1 System Setup Phase

The ECDSA key pairs (pk_R, sk_R) and $(pk_{W_1}, sk_{W_1}), \cdots, (pk_{W_n}, sk_{W_n})$ are chosen independently by R and W, respectively. R prepares some unredeemed transactions $\mathsf{Tx}_0^R, \cdots, \mathsf{Tx}_n^R$ and Tx_{sig}^R, each W_i prepares some unredeemed transactions $\mathsf{Tx}_{sig}^{W_i}$ and Tx_{W_i}, RSUs prepare an unredeemed transactions Tx_{sig}^{RSUs}.

5.2 Request Phase

The service fee source for transaction $\mathsf{PutMoney}_0^R$ of value d_0 ₿ is broadcasted by R who then sends the request to RSUs. R specifies the number n of witnesses, the value of remuneration, and the corresponding area. n is an odd integer to ensure that two messages are not the same, and the witness must choose between two conditions msg_A or msg_B based on whether the area has a traffic jam or not. Upon verifying $\mathsf{PutMoney}_0^R$ on the blockchain, RSUs disseminate the request among the corresponding area for R. Figure 7 displays the details of $\mathsf{PutMoney}_0^R$.

5.3 Announcement Phase

In this phase, witnesses are collected based on three sequential sub-phase: the *Witness Commitment Phase*, the *Requester Commitment Phase* and the *Signature Collection Phase*.

5.3.1 Witness Commitment Phase

The witnesses respond to the request, and RSUs collect $W_1, \cdots, W_n$. In case $\mathsf{TxCommit}_i$ are insufficiently collected, R will redeem $\mathsf{PutMoney}_0^R$ and exit. Each W_i executes $\mathsf{CS.Commit}(W_i, R, d_1, t + 4max, msg_i)$, where t is the time when all $\mathsf{TxCommit}_i$ are confirmed on the blockchain, and max represents the greatest delay between broadcasting and including a transaction on the blockchain. W_i broadcasts a $\mathsf{TxCommit}_i$ worth d_1 ₿ on the blockchain which commits to their witness message msg_i. If a malicious W_i fails to disclose msg_i before time $t + 4max$, R can obtain a penalty of d_1 ₿ from W_i during the *Claim Phase*. As the same information has the same hash value, malicious W_i can directly commit the majority if they discover that a specific message is the majority based on the messages already included on the blockchain. Moreover, we should prevent R from obtaining the messages from the blockchain directly. So the msg_i should be signed by W_i first and store the hash of signature on the blockchain, denoted as $h_{msg_i} = H(\sigma_{msg_i})$, where $\sigma_{msg_i} = msg_i || Sig(SK_{W_i}, msg_i)$, and "||" is the concatenation notation. The details of $\mathsf{TxCommit}_i$ are shown in Figure 8.

Note*: The required number of witnesses can be adaptively defined based on various scenarios. For example, during emergency situations such as accidents, where an urgent message needs to be transmitted, the requester may reduce the threshold number of witnesses, especially if there are fewer vehicles available for message transmission. In critical situations, the number of witnesses can even be set as low as 1. Conversely, in regular scenarios, the number of witnesses can be adjusted upward accordingly.*

1.System Setup Phase: (pk_R, sk_R) and $(pk_{W_1}, sk_{W_1}), \cdots, (pk_{W_n}, sk_{W_n})$ are the ECDSA public-secret key pairs chosen by R and W, respectively. Then, R and W prepare some unredeemed transactions.
2.Request Phase: Once R broadcasts $\mathsf{PutMoney}_0^R$ and sends it to RSUs, the latter distribute the request within the relevant area only if $\mathsf{PutMoney}_0^R$ is validated; if not, RSUs quit.
3.Announcement Phase:

- *Witness Commitment Phase*: RSUs collect $W_1, \cdots, W_n$. For $i \in [1, n]$, W_i broadcasts a deposit transaction $\mathsf{TxCommit}_i$ to commit his witness message msg_i. If RSUs don't collect enough $\mathsf{TxCommit}_i$, R gets back $\mathsf{PutMoney}_0^R$ and quits.
- *Requester Commitment Phase*: R broadcasts a deposit transaction $\mathsf{TxCommit}_R$ to commit a random value r_R used in the *Claim Phase*. W checks if $\mathsf{TxCommit}_R$ is confirmed or not. If not, all W quit.
- *Signature Collection Phase*: RSUs collect the signature $\mathsf{sig}_{W_i}([\mathsf{TxFine}_i])$ signed by W_i. Then RSUs broadcast the transaction $\mathsf{TxServiceSig}$ in which stores the hash of all W's signatures. R and W check if $\mathsf{TxServiceSig}$ is confirmed or not. If not, R and W quit.

4.Computation Phase: For $i \in [1, n]$, R broadcasts the transaction $\mathsf{PutMoney}_i^R$ as the message fee and each W_i broadcasts the transaction $\mathsf{PutMoney}^{W_i}$ as the guaranty. When all $\mathsf{PutMoney}$ are confirmed, R broadcasts the joint transaction $\mathsf{TxCompute}$. All W check if $\mathsf{TxCompute}$ is confirmed. If not, all W get $\mathsf{PutMoney}^{W_i}$ back and quit.
5.Payment Phase: One way RSUs broadcast the $\mathsf{TxRSUFee}$ transaction is by providing all $\mathsf{sig}_{W_i}([\mathsf{TxFine}_i])$ required to redeem the service fee. Conversely, for $i \in [1, n]$, W_i, W_i broadcasts the transaction TxOpen_i to reveal the witness message. If all TxOpen_i are confirmed and the witness message msg_i is the majority, W_i broadcasts the transaction $\mathsf{TxMsgFee}_i$ to redeem the message fee.
6. Claim Phase: If all the TxOpen_i transactions are confirmed and the witness message msg_i is not in the majority, R will send out the transaction $\mathsf{TxCompensation}_i$ to reimburse W_i for their guarantee. In another scenario, if any W_i fails to reveal the message msg_i before time $t + 4max$, R will broadcast the transaction TxFine_i to redeem W_i's deposit. Subsequently, R will reveal the random value r_R before time $t + 6max$. For $j \in [1, n], j \neq i$, W_j will broadcast the transaction $\mathsf{Txparticipate}_j$ using r_R as the participation fee. However, if R does not reveal r_R, W_j will broadcast the transaction TxFine_j^R to redeem R's deposit as the compensation.

Fig. 6: The PIAS protocol

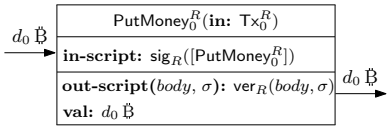


Fig. 7: The Transaction PutMoney

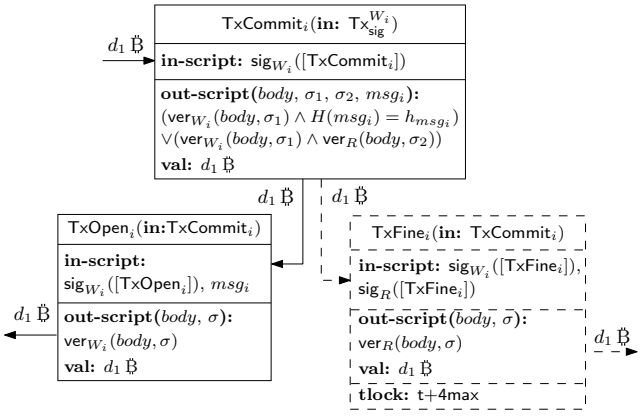


Fig. 8: The Transaction of Witness Commit

5.3.2 Requester Commitment Phase

After RSUs collect n witnesses $W_1, \cdots, W_n$, R performs $\mathsf{CS.Commit}(R, W_i, d_2^{R,i} + d_2^{W_i}, t + 6max, r_R)$ who posts a deposit transaction $\mathsf{TxCommit}_R$ of value $n(d_2^{R,i} + d_2^{W_i})$ ₿ to commit a random value r_R. If a dishonest requester R prevents an honest witness from claiming the participation fee during the *Claim Phase*, then each W_i can demand a penalty of $d_2^{R,i} + d_2^{W_i}$ ₿ from R's deposit during the same phase. Figure 9 displays the specifics of $\mathsf{TxCommit}_R$.

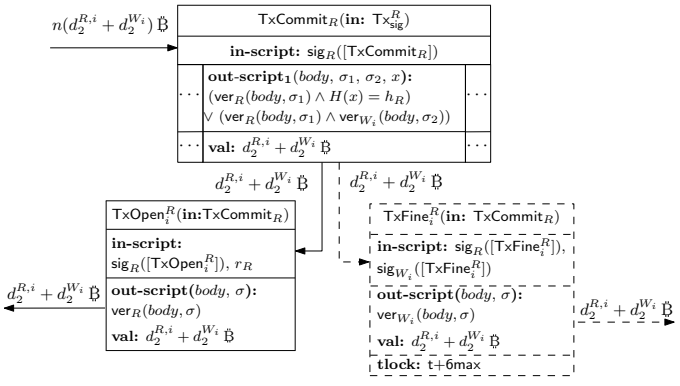


Fig. 9: The Transaction of Requester Commit

5.3.3 Signature Collection Phase

Upon verifying and confirming the transaction $\mathsf{TxCommit}_R$, W_i generates the content of the punishment transaction TxFine_i, signs it, and transfers the signatures $\mathsf{sig}_{W_i}([\mathsf{TxFine}_i])$ to the RSUs. The RSUs collect all signatures and broadcast the transaction $\mathsf{TxServiceSig}$ to store the hash of all signatures as shown in Equation (1) on the blockchain.

$$\sigma_{TxFine} = (\sigma^1_{TxFine} = H(\mathsf{sig}_{W_1}([\mathsf{TxFine}_1])), \cdots, \sigma^n_{TxFine} = H(\mathsf{sig}_{W_n}([\mathsf{TxFine}_n]))) \quad (1)$$

In the *Payment Phase*, RSUs can provide these signatures to earn the service fee. The opcode OP_RETURN is used by TxServiceSig to publicly output σ_{TxFine}. The details of TxServiceSig are shown in Figure 10.

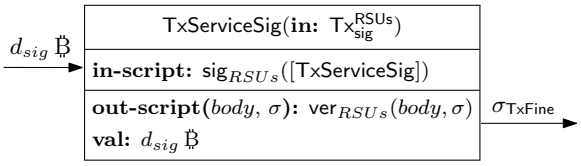


Fig. 10: The Transaction of TxServiceSig

5.4 Computation Phase

If the transaction TxServiceSig is included on the blockchain with enough confirmations, each W_i computes the hash of his signature and checks if RSUs tamper his signature $\mathsf{sig}_{W_i}([\mathsf{TxFine}_i])$ or not. Then, for $i \in [1, n]$, R broadcasts the transaction $\mathsf{PutMoney}_i^R$ of value $d_2^{R,i}$ ₿ as the message fee and W_i broadcasts the transaction $\mathsf{PutMoney}^{W_i}$ of value $d_2^{W_i}$ as the guaranty. When all PutMoney are confirmed, R and W make a joint transaction TxCompute, in which the computation rules of service and message fees are determined.

Figure 11 provides details of TxCompute. The transaction is posted on the blockchain by R before time $t + 3max$ and its input consists entirely of PutMoney transactions. The output of TxCompute determines a series of contracts containing a service fee for RSUs and the message fee and guaranty for witnesses. σ_{TxFine} is the set of all signature hash and MSG is the set of all msg_i, denoted as $\mathsf{MSG} = (msg_1, \cdots, msg_n)$. The function $\mathsf{Majority}(\cdot)$ as a script operation are based on stack using opcode OP_TOALTSTACK to push the msg_i into stack. The number of elements msg_i in stack is calculated by opcode OP_DEPTH and compared with $n/2$ using opcode OP_GREATERTHAN.If the number exceeds $n/2$, then msg_i would represent the majority of the messages. To illustrate, if there are two possible road conditions, namely msg_1 and msg_2, with msg_1 being the majority, we can indicate this as $\mathsf{Majority}(msg_1, \mathsf{MSG}) = \mathsf{True}$ and $\mathsf{Majority}(msg_2, \mathsf{MSG}) = \mathsf{False}$. During the *Payment Phase*, W_i can earn the message fee and recover the collateral if msg_i represents the majority, and we have $\mathsf{Majority}(msg_i, \mathsf{MSG}) = \mathsf{True}$. Conversely, if msg_i does not constitute the majority, and the result is $\mathsf{Majority}(msg_i, MSG) = \mathsf{False}$, W_i will forfeit his collateral. If there is any W_i who doesn't reveal the msg_i, the number of messages doesn't meet n. So the majority cannot be gotten, denoted as $\mathsf{Reveal}(\mathsf{MSG}) = false$. In this case, R can obtain the deposit of W_i and other witnesses can earn the participation fee. More details can be described in the *Payment Phase* and *Claim Phase*.

5.5 Payment Phase

During this stage, RSUs are eligible to receive the service fee, and an honest witness has the ability to earn the message fee.

- *Service fee payment*: RSUs broadcast the transaction TxRSUFee by providing all witnesses' signatures $\mathsf{sig}_{W_1}([\mathsf{TxFine}_1]), \cdots, \mathsf{sig}_{W_n}([\mathsf{TxFine}_n])$ as input script which are stored in the *Signature Collection Phase* to get the service fee of value d_0 ₿.
- *Message fee payment*: For $i \in [1, n]$, if all W_i perform $\mathsf{CS.Open}(W_i, R, d_1, t + 4max, msg_i)$ to reveal the message msg_i by broadcasting the transaction TxOpen_i before time $t + 4max$ as shown in Figure 8 and the message msg_i is the majority, W_i is able to receive a message fee of value $d_2^{R,i}$ ₿ and get the guaranty of value $d_2^{W_i}$ ₿ back by broadcasting the transaction $\mathsf{TxMsgFee}_i$ as shown in Figure 11.

5.6 Claim Phase

Only in the case that msg_i isn't the majority or partial messages aren't revealed, PIAS comes to the *Claim Phase*. We discuss these in different scenarios.

- *Requester claim*:
  - *All committed messages are revealed*: When all witnesses reveal their message, if the message msg_i is not the majority, R can claim the guarantee of value $d_2^{W_i}$ ₿ as a penalty and retrieve the message fee by broadcasting the transaction $\mathsf{TxCompensation}_i$ as depicted in Figure 11. After claiming all penalties, R performs $\mathsf{CS.Open}(R, W_i, d_2^{R,i} + d_2^{W_i}, t + 6max, r_R)$ to reveal the random value r_R by broadcasting the transaction TxOpen_i^R to receive the deposit back before time $t + 6max$ as shown in Figure 9.
  - *Partial Committed messages are revealed*: In the event that W_i discovers that his message is not in the majority, he may choose to withhold payment to R, resulting in msg_i not being disclosed. In such cases, R may recover W_i's initial deposit made during the *Witness Commitment Phase*. R performs $\mathsf{CS.Fine}(W_i, R, d_1, t+4max, msg_i)$ to obtain W_i's deposit by broadcasting the transaction TxFine_i which is shown in Figure 8. To ensure that W_i discloses the witness message prior to a designated time, let $d_1 \geq n(d_2^{R,i} + d_2^{W_i})$. Then R reveals r_R by broadcasting the transaction TxOpen_i^R before time $t + 6max$.
- *Witness claim*:
  - *All Committed messages are revealed*: Honest witness W_i earns the message fee of value $d_2^{R,i}$ ₿ and get the guaranty of value $d_2^{W_i}$ ₿ back by broadcasting the transaction $\mathsf{TxMsgFee}_i$.
  - *Partial Committed messages are revealed*: If W_i chooses not to disclose their msg_i with malicious intent, other witness W_j who has revealed the msg_j can earn the participation fee of value $d_2^{R,j} + d_2^{W_j}$ ₿ by broadcasting the transaction $\mathsf{TxParticipate}_j$. Otherwise, if R doesn't reveal r_R, the rest of witness W_j can redeem R's deposit by broadcasting the transaction TxFine_j^R as shown in Figure 9.

Figure 12 illustrates the interconnectedness of different transactions in PIAS, taking into account all of its technical aspects from a holistic perspective.

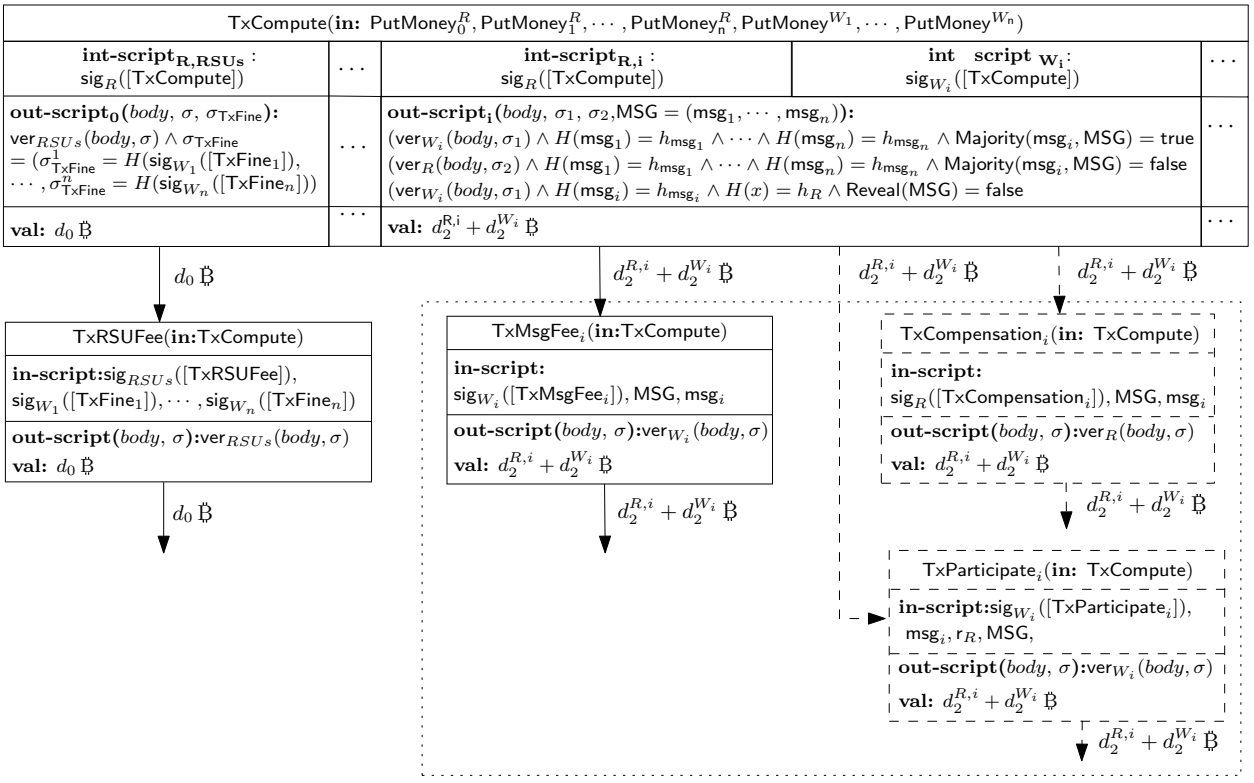


Fig. 11: The details of transaction TxCompute

6 Analysis and Evaluation Of PIAS

6.1 Security Analysis

The security results for PIAS are presented in compliance with the security requirement.

1) *Sybil-Resistance*: Before sending the request to RSUs, R needs to broadcast a transaction PutMoney. If a malicious user forges many normal requesters to forward requests, he needs to prepare many PutMoney transactions, each of which requires a deposit.
2) *Prevention of Denial of service attack*: A malicious requester may initiate a denial of service attack by continuously sending requests to RSUs. However, before sending the request to RSUs, R needs to broadcast the transaction PutMoney. The process efficiently prevents the adversary from initiating an excessive number of requests.
3) *Prevention of Replay attack*: An adversary can hinder other participants by replaying existing legitimate messages eavesdropped from other requesters. However, RSUs will check the transaction PutMoney to determine if it has been redeemed and the deadline of the request.
4) *Prevention of Forgery*: The request is signed by R with his private key. The signature is difficult to tamper with if ECDSA is unforgeable. During the verification process, the request for forgery is detected.
5) *Prevention of Man-in-the-middle attack*: If a valid signature cannot be forged on the tampered message, the adversary attempting a man-in-the-middle attack on the request sent from the requester to the RSUs will not succeed, even if they modify the request contents such as remuneration or destination.
6) *Prevention of Eavesdropping attack*: An adversary may eavesdrop on the public channel to get the commitment transaction of other witnesses. However, the witness message is a hash value on the blockchain. The adversary can not get the majority.
7) *Prevention of Malleability attack*: Adversaries can launch malleability attacks by monitoring transactions on the public channel. However, as the transactions used in PIAS are published in an orderly fashion on the blockchain, these attacks are insignificant.

For a detailed security analysis of the protocol, please refer to the **Supplemental Material A**.

6.2 Performance Analysis

In this section, we evaluate the computation and communication overheads of PIAS and compare it with related work.

TABLE 2: The Notations of Performance

Notation	Description
T_{KP}	the average computation time for ECDSA key pair
T_{sig}	the average computation time for signature generation
T_{ver}	the average computation time for signature verification
T_{HG}	The average computation time for hash to G
T_M	The average computation time for scalar multiplication
T_P	The average computation time for bilinear pairing
$\|Z_p\|$	The size of element in Z_p
$\|G\|$	The size of element in group G
$\|T\|$	The size of timestamp
$\|M\|$	The size of a message for announcement

6.2.1 Computation and Communication Overhead

We compare the performance of PIAS with a secure and trustworthy announcement dissemination scheme intro-

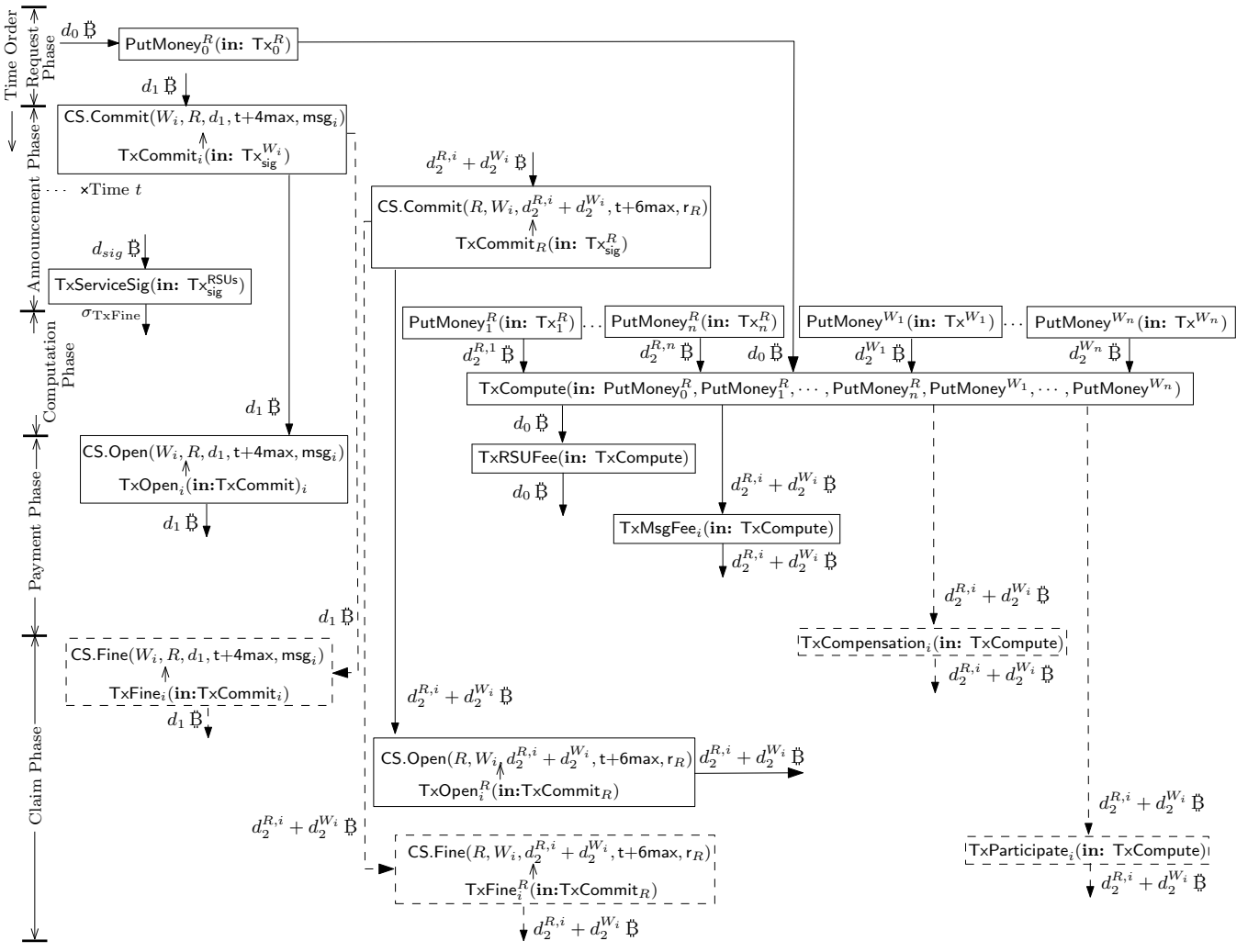


Fig. 12: Relationship of Transactions in Time Order

duced in [29], and evaluate their computational and communication overheads.

TABLE 3: Comparison of Computation Overhead

Scheme	Computation cost
Li et al. [29]	$n(2t+2)T_P + n(1+5t)T_M + ntT_{HG}$
Ours	$nT_{KP} + (13n-10)T_S$

t represents the threshold of threshold signatures.
n represents the number of participants, our scheme's number of participants is 1 requester and $n-1$ witnesses.

We conduct a theoretical analysis of the computational costs associated with the schemes outlined in Table 3. Scheme [29] relies on public key encryption and utilizes blockchain as an auxiliary. When assessing computational costs, we exclude general hash algorithms and scalar addition due to their minimal impact. For initiating n announcements and verifying them, scheme [29] requires $n(2t+2)$ bilinear pairing calculations, $n(1+5t)$ scalar multiplications, and nt hash mappings to G. In contrast, PIAS is designed based on blockchain protocol and only requires hash, signature, and verification calculations. During the initialization phase of PIAS, n pairs of ECDSA keys are generated. In the request phase, a single signature operation is performed. During the announcement phase, $9n-8$ signing operations occur. In the computation phase, $2n-1$ signings are performed, and in the payment phase, $2(n-1)$ signings are executed. In total, $nT_{KP} + (13n-10)T_S$ operations are required in PIAS.

TABLE 4: Comparison of Communication Overhead

Scheme	Communication overhead
Li et al. [29]	$n(t+5)\|G\| + n(5t+11)\|Z_p\| + n(t+3)\|T\|$
Ours	$(17n-5)\|S\| + 5(n-1)\|H\| + 2\|M\|$

We conduct a theoretical analysis and comparision of the communication overhead in Table 4. The the primary communication cost of scheme [29] is introduced by the algorithms of public key encryption. On the contrary, PIAS is entirely blockchain-based, with the communication cost being determined by the size of the blockchain data blocks. In scheme [29], the communication cost encompasses $n(t+5)$ elements of G, $n(5t+11)$ elements of Z_p, and $n(t+3)$ timestamps T, resulting in a total communication cost of $n(t+5)|G|+n(5t+11)|Z_p|+n(t+3)|T|$. In PIAS, the communication cost in the request initiation phase is 2 elements of $|S|$, in the announcement phase is $(14n-12)|S|+3(n-1)|H|$,

in the computation phase is $(2n+3)|S|+2(n-1)|H|$, and in the payment phase is $(n+2)|S|+2|M|$. Therefore, the total communication cost is $(17n-5)|S|+5(n-1)|H|+2|M|$.

We conduct experiments on a desktop computer running 64-bit Windows 10 operating system. The computer is equipped with Intel(R) Core(TM) i7-9700 CPU @ 3.00 GHz and 16.00 GB RAM. We use ECDSA and Hashlib libraries in Python 3.12, select the SECP256k1 curve, and the Miracl library for integer and rational arithmetic to test the performance of our scheme and scheme [29]. The order of the group G in scheme [29] is represented by q. The bit length of q is 256 bits, and the bit length of elements in G is 512 bits. We choose pairing $e : G \times G \to G_T$ to evaluate the performance of scheme [29].

TABLE 5: Computational Cost of Vehicles (ms)

The Number of participants (n)	3	5	7	9
Li et al. [29]	140.505	234.175	327.845	421.515
Ours	66.540	118.072	169.603	221.135

Next, we analyze the computational efficiency. Table 5 presents a comparison of the computational costs for the vehicles. In Table 3, we examined the computational costs through theoretical analysis. Scheme [29] has computation overhead of 140.505 ms for $n = 3$. When the number n increased from 5 to 9, the time increases from 234.175 ms to 421.515 ms. Thus, in Table 5, the computation time of Scheme [29] appears stable. In PIAS, the computation time for $n = 3$ is 66.54 ms. With the witness number n increasing from 5 to 9, the time increases from 118.072 ms to 221.135 ms. In summary, our system maintains relatively low computational costs.

TABLE 6: Transmission Cost of Vehicles (kb)

The Number of participants (n)	3	5	7	9
Li et al. [29]	26.719	44.531	62.344	80.156
Ours	27.438	48.001	68.563	89.125

The transmission costs of vehicles are detailed in Table 6. Our comparison focuses on the total transmission overheads of vehicles, considering varying reference participant numbers from 3 to 9. In scheme [29], the transmission overhead is 26.719 kb for $n = 3$. As n increases from 5, 7, to 9, the transmission costs of vehicles are 44.531 kb, 62.344 kb, and 80.156 kb, respectively. In PIAS, the transmission cost is 27.438 kb for $n = 3$. With n increasing from 5 to 9, the incurred communication costs are 48.001 kb, 68.563 kb, and 89.125 kb, respectively. While our scheme relies entirely on blockchain, resulting in higher communication loads compared to the public-key based scheme [29], a comparative analysis reveals that our solution's communication costs are only slightly higher, with both solutions showing similar communication costs.

TABLE 7: The transmission speed of data

The Number of participants (n)	3	5	7	9
Data Size (kb)	27.438	48.001	68.563	89.125
Transmission Time (s)	0.004	0.008	0.011	0.015

Based on the 802.11p protocol [34], Table 7 calculates the communication time of PIAS at a network speed of 6 Mbps. For $n = 3$, the communication cost is 27.438 kb, and the transmission time is 4 ms. When the participant number n increases from 5 to 9, the transmission time increases from 8 ms to 15 ms. It can be seen that PIAS is suitable for the actual VANETs environment.

6.2.2 Number Of Transactions

In the *Request Phase* of PIAS, only a single transaction, $\mathsf{PutMoney}_0^R$, is necessary. In the *Announcement Phase*, n transactions $\mathsf{TxCommit}_i$, one transaction $\mathsf{TxCommit}_R$ and one transaction $\mathsf{TxServiceSig}$ are involved. In the *Computation Phase*, n transactions $\mathsf{PutMoney}_i^R$ which can be replaced with one transaction, n transactions $\mathsf{PutMoney}^{W_i}$ and one transaction $\mathsf{TxCompute}$ are involved. In the *Payment Phase*, if all W are honest, n transactions TxOpen_i are required. One transaction $\mathsf{TxRSUFee}$ and n transactions $\mathsf{TxMsgFee}_i$ are involved. Then n transactions TxOpen_i^R are involved which can be replaced with one transaction because they only need the same input. So if all participants are honest, it needs $7 + 4n$ transactions in total.

If there is any malicious participant in the *Claim Phase* when *all announcement messages are revealed* and there are t messages of witnesses that are not in the majority, then $n - t$ transactions $\mathsf{TxMsgFee}_i$ and t transactions $\mathsf{TxCompensation}_i$ are involved. It also requires a total of $7+4n$ transactions. Therefore, if all announcement messages are revealed, the number of involved transactions is only related to n. When *partial announcement messages are revealed*, and t messages are not revealed, $n-t$ transactions TxOpen_i and t transactions TxFine_i are involved. If R reveals r_R, then n transactions TxOpen_i^R are involved, which can be replaced with one transaction because they have the same input. Additionally, $n - t$ transactions $\mathsf{TxParticipate}_i$ are involved, resulting in a total of $7 + 4n - t$ transactions. If R doesn't reveal r_R, then n transactions TxFine_i^R are involved, necessitating a total of $6 + 4n$ transactions. The data presented in Table 8 indicate a positive correlation between the quantity of witnesses and the number of transactions involved in PIAS.

TABLE 8: The number of transactions in PIAS

Scenario / *Witness*	Honest	All revealed	Partial not revealed				
			1	3	5	7	r_R not revealed
3	19	19	17	–	–	–	18
5	27	27	23	25	–	–	26
7	35	35	29	31	33	–	34
9	43	43	35	37	39	41	42

6.2.3 Blockchain Cost

Based on the compatibility analysis, PIAS has been found to be compatible with both the Bitcoin and Ethereum blockchains. In blockchain technology, the block time signifies the average duration for an additional block to be generated on the blockchain network. Bitcoin's block time has been set to 10 minutes, whereas Ethereum's block time ranges between 14 and 15 seconds. For PIAS, it requires a minimum of 6 maximum delays to insert a transaction into the Bitcoin blockchain after broadcasting it. So it needs more than 60 minutes which isn't efficient for the time overhead for the Bitcoin blockchain.

To assess the expenses of PIAS, we conducted a simulation of the system scheme on the Ethereum official test network Ropsten and analyzed it by deploying smart contracts. Our implementation utilized the SHA-256 hash function and was written in the Solidity programming language. The smart contract is capable of recording the addresses of participants and messages. Additionally, it stores the deposit and triggers a payment after computation. In our Ethereum experiment, we do not require R to commit a random value. Furthermore, the transaction $\mathsf{TxParticipate}_i$ can be combined with the transaction TxMsgFee, and the transaction TxFine_i can be combined with the transaction TxCompensation. Throughout the experiment, we set a gas price of 15 $Gwei$ and established that 1 Ether was equivalent to 1367 USD, where $1Gwei = 10^9 wei = 10^{-9} ether$. Ethereum gas is a unit that measures the computational effort needed to execute specific operations. The gas overhead of the smart contracts, as measured in the experiments, is as follows. It is essential to acknowledge that the value of cryptocurrency is highly volatile and should solely be utilized as a reference value.

***Remark**: Due to the volatile nature of Ether and Bitcoin prices, which are currently at relatively high levels, customers might be more inclined to pay higher fees in emergency situations, such as when facing danger on unfamiliar roads or requiring urgent information in advance. When deploying the system, it's advisable to consider utilizing a consortium blockchain or other blockchain systems with lower fees.*

TABLE 9: The constant smart contract costs (gasprice=15 $Gwei$, 1 Ether=1367 USD)

Function	Gas Used	Actual Cost(ether)	USD
Deployment	1071122	0.01606683	21.9634
TxCommit	70179	0.001052685	1.4390
TxRSUFee	35071	0.000526065	0.7191

TABLE 10: Smart contract costs under different number of all honest witnesses(gasprice=15 $Gwei$, 1 Ether=1367 USD)

The number of W	Function	Gas Used	Actual Cost(ether)	USD
3	TxCompute	68904	0.00103356	1.4129
	TxMsgFee	72064	0.00108096	1.4777
5	TxCompute	83074	0.00124611	1.7034
	TxMsgFee	101650	0.00152475	2.0843
7	TxCompute	97244	0.00145866	1.9940
	TxMsgFee	131236	0.00196854	2.6910
9	TxCompute	111414	0.00167121	2.2845
	TxMsgFee	160822	0.00241233	3.2977

Table 9 displays the costs associated with deployment of contract, performing TxCommit and TxRSUFee. The expenses incurred for these operations remain relatively constant. Deploying the matching contract only requires a single execution of the create operation, which costs 1071122 gas = 21.9634 USD. Each witness must pay 70179 gas = 1.4390 USD to commit the message using TxCommit, and 35071 gas = 0.7191 USD is required for transferring the service fee through TxRSUFee.

The cost of TxCompute, TxMsgFee, and TxCompensation increases proportionally to the increase in the number of witnesses. The process TxCompute, TxRSUFee, TxMsgFee, TxCompensation only need to be performed once. Then we build the contracts for four different numbers of witnesses, respectively. As shown in Table 10. Assuming the honesty of all witnesses, the expenditure for TxCompute amounts to roughly 83074 gas = 1.7034 USD with 5 witnesses. The cost of TxMsgFee is approximately 101650 gas = 2.0843 USD with 5 witnesses. Because there is no malicious witness, TxCompensation will not be performed.

When there are malicious witnesses, the costs of TxCompute, TxMsgFee, and TxCompensation are also correlated with the number of witnesses. As shown in Table 11, the cost of TxMsgFee is less when there are no malicious witnesses. However, there is a cost associated with TxCompensation. When there is only one malicious witness, the cost of performing TxCompute is approximately 77149 gas = 1.5819 USD with 5 witnesses. The cost of TxMsgFee is around 87908 gas = 1.8026 USD with 5 witnesses. Similarly, the cost of TxCompensation is nearly 64275 gas = 1.3180 USD with five witnesses. In our experiment, the execution of an operation that uses up to 160822 gas takes no more than one second; therefore, Ethereum blockchain processing time is more efficient and tolerable compared to the Bitcoin blockchain.

TABLE 11: Smart contract costs under different number of witnesses with one malicious witness(gasprice=15 $Gwei$,1 Ether=1367 USD)

The number of W	Function	Gas Used	Actual Cost(ether)	USD
3	TxCompute	62979	0.000944685	1.2914
	TxMsgFee	85322	0.00127983	1.7495
	TxCompensation	50355	0.000755325	1.0325
5	TxCompute	77149	0.001157235	1.5819
	TxMsgFee	87908	0.00131862	1.8026
	TxCompensation	64275	0.000964125	1.3180
7	TxCompute	91319	0.001369785	1.8725
	TxMsgFee	117494	0.00176241	2.4092
	TxCompensation	78195	0.001172925	1.6034
9	TxCompute	105489	0.001582335	2.1631
	TxMsgFee	147080	0.0022062	3.0159
	TxCompensation	92115	0.001381725	1.8888

7 Conclusion

This paper introduces PIAS, a blockchain-based privacy-preserving incentive announcement system designed for the Internet of Vehicles. The paper presents the system model, definition, adversary model, and security requirements, as well as the specific design details of PIAS. Witnesses are required in our system to provide traffic information, and they will be compensated with a valid message. Our system guarantees that the witness either receives the compensation or loses their deposit. Our security analysis indicates that PIAS achieves privacy preservation and incentive as long as the hash function is collision-resistant and ECDSA is unforgeable. Through our experimental results, we demonstrate that PIAS demonstrates efficiency with regard to computation costs on the Ethereum blockchain.

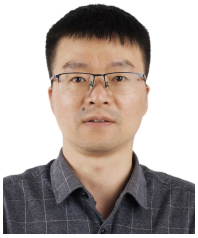
Acknowledgments

This work are supported by National Natural Science Foundation of China under Grant No. 62372110, Fujian Provincial

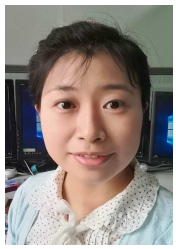
Natural Science of Foundation under Grant No. 2023J02008 and AXA Research Fund. Hongju Cheng is supported by National Natural Science Foundation of China under Grant No.62372111. X. Luo is supported in part by the National Key Research and Development Program of China (No.2022YFB3102900), the National Natural Science Foundation of China (No. U23A20305, 62172435).

REFERENCES

[1] Taslimasa H, Dadkhah S, Neto E C P, et al. Security issues in Internet of Vehicles (IoV): A comprehensive survey[J]. Internet of Things, 2023: 100809.
[2] Grover J. Security of Vehicular Ad Hoc Networks using blockchain: A comprehensive review[J]. Vehicular Communications, 2022: 100458.
[3] Zavvos E, Gerding E H, Yazdanpanah V, et al. Privacy and Trust in the Internet of Vehicles[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(8): 10126-10141.
[4] Yang, F., Wang, S., Li, J., Liu, Z. and Sun, Q, An overview of internet of vehicles, China communications, 2014, 11(10), 1-15.
[5] Sakiz, F., and Sen, S., A survey of attacks and detection mechanisms on intelligent transportation systems: VANETs and IoV, Ad Hoc Networks, 2017, 61, 33-50.
[6] Qureshi K N, Din S, Jeon G, et al. Internet of vehicles: Key technologies, network model, solutions and challenges with future aspects[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 22(3): 1777-1786.
[7] Sheikh M S, Liang J, Wang W. Security and privacy in vehicular ad hoc network and vehicle cloud computing: a survey[J]. Wireless Communications and Mobile Computing, 2020, 2020: 1-25.
[8] M. Raya, A. Aziz, and J.-P. Hubaux, "Efficient secure aggregation in VANETs," in Proc. 3rd Int. Workshop Veh. Ad Hoc Netw., Sep. 2006, pp. 67–75.
[9] Miller, Charlie, and Chris Valasek. "Remote exploitation of an unaltered passenger vehicle." Black Hat USA 2015 (2015): 91.
[10] Engoulou, Richard Gilles , et al. "VANET security surveys." Computer Communications 44(2014):1-13.
[11] Yu, Bo , C. Z. Xu , and B. Xiao . "Detecting Sybil attacks in VANETs." Journal of Parallel and Distributed Computing 73.6(2013):746-756.
[12] Petit, Jonathan , and S. E. Shladover . "Potential Cyberattacks on Automated Vehicles." IEEE Transactions on Intelligent Transportation Systems 16.2(2014):1-11.
[13] Wu, Qianhong , J. Domingo-Ferrer , and U. Gonzalez-Nicolas . "Balanced Trustworthiness, Safety, and Privacy in Vehicle-to-Vehicle Communications." IEEE Transactions on Vehicular Technology 59.2(2010):559-573.
[14] Chen, Liqun , S. L. Ng , and G. Wang . "Threshold Anonymous Announcement in VANETs." IEEE Journal on Selected Areas in Communications 29.3(2011):605-615.
[15] Zhang, Lei , et al. "APPA: Aggregate Privacy-Preserving Authentication in Vehicular Ad Hoc Networks." International Conference on Information Security Springer, Berlin, Heidelberg, 2011.
[16] Gao W, Wang M, Zhu L, et al. Threshold-based secure and privacy-preserving message verification in VANETs[C]//2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications. IEEE, 2014: 795-802.
[17] Shao, Jun , et al. "A Threshold Anonymous Authentication Protocol for VANETs." IEEE Transactions on Vehicular Technology 65.3(2016):1711-1720.
[18] Azees, Maria , P. Vijayakumar , and L. J. Deboarh . "EAAP: Efficient Anonymous Authentication With Conditional Privacy-Preserving Scheme for Vehicular Ad Hoc Networks." IEEE Transactions on Intelligent Transportation Systems (2017):1-10.
[19] Cui J, Zhang X, Zhong H, et al. Extensible conditional privacy protection authentication scheme for secure vehicular networks in a multi-cloud environment[J]. IEEE Transactions on Information Forensics and Security, 2020, 15: 1654-1667.
[20] Cheng Y, Ma J, Liu Z, et al. Efficient Anonymous Authentication and Privacy-Preserving Reliability Evaluation for Mobile Crowdsensing in Vehicular Networks[J]. IEEE Internet of Things Journal, 2023.
[21] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.[Online]. Available: http://bitcoin.org/bitcoin.pdf
[22] M. Andrychowicz, S. Dziembowski, D. Malinowski, and L. Mazurek, "Secure multiparty computations on bitcoin," in Security and Privacy (SP), 2014 IEEE Symposium on. IEEE, 2014, pp. 443–458.
[23] Zhang, Yinghui, et al. "Outsourcing service fair payment based on blockchain and its applications in cloud computing." IEEE Transactions on Services Computing (2018).
[24] Li, Lun, et al. "CreditCoin: A privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles." IEEE Transactions on Intelligent Transportation Systems 19.7 (2018): 2204-2220.
[25] H. Huang, X. Chen, Q. Wu, X. Huang, and J. Shen, "Bitcoin-based fair payments for outsourcing computations of fog devices," Future Generation Computer Systems, vol. 78, pp. 850–858, 2018.
[26] Miyachi K, Mackey T K. hOCBS: A privacy-preserving blockchain framework for healthcare data leveraging an on-chain and off-chain system design[J]. Information processing & management, 2021, 58(3): 102535.
[27] Yeh L Y, Shen N X, Hwang R H. Blockchain-based privacy-preserving and sustainable data query service over 5g-vanets[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(9): 15909-15921.
[28] Fernandes C P, Montez C, Adriano D D, et al. A blockchain-based reputation system for trusted VANET nodes[J]. Ad Hoc Networks, 2023, 140: 103071.
[29] Li, Xincheng, Xinchun Yin, and Jianting Ning. "Trustworthy Announcement Dissemination Scheme With Blockchain-Assisted Vehicular Cloud." IEEE Transactions on Intelligent Transportation Systems 24.2 (2022): 1786-1800.
[30] Rabin, Tal, and Michael Ben-Or. "Verifiable secret sharing and multiparty protocols with honest majority." Proceedings of the twenty-first annual ACM symposium on Theory of computing. 1989.
[31] Araki, Toshinori, et al. "High-throughput semi-honest secure three-party computation with an honest majority." Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. 2016.
[32] Dalskov, Anders, Daniel Escudero, and Marcel Keller. "Fantastic four:Honest-MajorityFour-Party secure computation with malicious security." 30th USENIX Security Symposium (USENIX Security 21). 2021.
[33] Chida, Koji, et al. "Fast large-scale honest-majority MPC for malicious adversaries." Advances in Cryptology–CRYPTO 2018: 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19–23, 2018, Proceedings, Part III 38. Springer International Publishing, 2018.
[34] Abdelgader, Abdeldime MS, and Wu Lenan. "The physical layer of the IEEE 802.11 p WAVE communication standard: The specifications and challenges." Proceedings of the world congress on engineering and computer science. Vol. 2. 2014.



Yonghua Zhan received the master degree from Fuzhou University, Fuzhou, China, in 2017. Now, he is pursing the PhD degree under Supervision of Prof. Yang Yang in College of Computer and Data Science, Fuzhou University, Fuzhou, China. His research interests are in the area of privacy protection and IoT.



Yang Yang received the B.Sc. degree from Xidian University, Xi'an, China, in 2006 and Ph.D. degrees from Xidian University, China, in 2011. She is a senior research scientist in School of Computing and Information System, Singapore Management University. She has been a full professor with College of Computer and Data Science, Fuzhou University. Her research interests are in the area of information security, privacy protection and Blockchain. She is Senior Member of IEEE.

Hongju Cheng received the PhD in Computer Science from Wuhan University in 2007. Now, he is a professor with the College of Computer and Data Science, Fuzhou University, Fuzhou, China. He is serving as editor / guest editor for more than 10 international journals. His interests include internet of things, mobile ad hoc networks, wireless sensor networks, and wireless mesh networks.

Xiangyang Luo received the B.S., M.S., and Ph.D degrees from State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, China, in 2001, 2004, and 2010, respectively. He has published more than 100 international journal and conference papers. He is a full professor of State Key Laboratory of Mathematical Engineering and Advanced Computing. His research interests are network and information security.

Zhangshuang Guan is currently pursuing a Ph.D. degree in the School of Computer Science and Technology, Zhejiang University, China. He received the B.Sc. degree from Qingdao University, China, in 2017 and M.Sc. degree from Shandong University, China, in 2020. His research interests include applied cryptography, intelligent computing, blockchain privacy and security.

Robert H. Deng is AXA Chair Professor of Cybersecurity, Director of the Secure Mobile Centre, School of Computing and Information Systems, Singapore Management University. His research interests include applied cryptography, data security and privacy, and network security. He has served/is serving on the editorial boards of many international journals in security, such as IEEE TIFS, IEEE TDSC, etc. He is Fellow of IEEE.