SCTSC: A Semi-centralized Traffic Signal Control Mode with Attribute-based Blockchain in IoVs

Lichen Cheng, Jiqiang Liu, Guangquan Xu, Zonghua Zhang, Hao Wang, Hong-Ning Dai, Yulei Wu, and Wei Wang

***Abstract*—Assisting traffic control is one of the most important applications on Internet of Vehicles (IoVs). Traffic information provided by vehicles is desired since drivers or vehicle sensors are sensitive in perceiving or detecting nuances on roads. However, the availability and privacy-preservation of this information are critical while conflicted with each other in vehicular communication. In this paper, we propose a semi-centralized mode with attribute-based blockchain in IoVs to balance the trade-off between the availability and the privacy-preservation. In this mode, a method of control-by-vehicles is used to control signals of traffic lights to increase traffic efficiency. Users are grouped their attributes like locations and directions before starting the communication. The users reach an agreement on determining a temporary signal timing by interacting with each other without leaking privacy. Final decisions are verifiable to all users, even if they have no a priori agreement and processes of consensus. The mode not only achieves the aim of privacy-preservation but also supports responsibility investigation for historical agreements via ciphertext-policy attribute-based encryption and blockchain technology. Extensive experimental results demonstrated that our mode is efficient and practical.**

***Index Terms*—attribute-based encryption, blockchain, privacy preserving, internet of vehicles.**

I. Introduction

INTERNET of vehicles (IoVs) has become one of the measures to ease traffic congestion in cities. Traffic information is automatically collected, disposed and broadcasted through IoVs for traffic condition prediction, traffic accident detection, and efficient service deliverie.

Current projects [1], [2], [3], [4], [5] relied on the capture of vehicle information to deal with traffic problems, instead of the information provided by vehicles. Let vehicles generate and broadcast messages about traffic information; this brings advantages to intelligent transportation since drivers are able to perceive tiny but essential traffic information compared with infrastructures like detection devices. However, in the process of capturing vehicle information, preserving the privacy of road users is less considered, such as driving habits and vehicles trajectories.

Researchers in academia preferred to use announcement protocols in vehicle ad-hoc networks (VANETs), inter-vehicle communication (IVC) or internet of vehicles (IoVs) to realize an interaction among vehicles, aiming at bypassing congestion roads and avoiding accidents. In [6], three necessary standards of messages are given, including message integrity, legitimate generation, and reliability measurement.

In this work, we introduced a semi-centralized traffic signal control mode (SCTSC mode) with attribute-based blockchain in IoVs for signalized intersections. Different from pre-timed modes and actuated modes, our mode dynamically modifies signal timing based on inputs received from vehicles and an attribute-based blockchain, which is not affected by environmental conditions and strict device installation requirements. Vehicles are divided into groups implicitly and dynamically by their dynamic attributes (e.g., locations and directions). Each vehicle votes to reach a temporary agreement of signal timing change encrypted by ciphertext-policy attribute-based encryption in its group. The temporary agreement and messages of agreement rounds are recorded on an attribute-based blockchain as records. Traffic signal controllers and participants (e.g., users in other groups and bystanders) are able to get and verify final decisions of temporary agreements without the leakage of drivers' privacy.

In summary, we make the following contributions:

- We proposed a semi-centralized traffic signal control mode (SCTSC mode) with attribute based blockchain in IoVs. The mode realizes an efficient dynamic traffic signal control from a novel method of control by vehicles.
- To the best of our knowledge, the attribute-based blockchain that we constructed in our mode is the first blockchain structure that supports fine-grained non-interactive access control on traffic data. The blockchain is tamper resistance. Data is generated and recorded in groups. Some part of data is transparent while others is only readable for those who have access (a proper attribute set).
- SCTSC mode achieves a balance between privacy-preservation and availability of information. Users are

The work reported in this paper was supported in part by National Key R&D Program of China, under grant 2017YFB0802805, in part by Natural Science Foundation of China, under Grant U1736114 and 61672092, and in part by the Fundamental Research Funds for the Central Universities of China under Grants 2018JBZ103.

L. Cheng, J. Liu and W. Wang are with Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, Beijing 100044, China (e-mail: lccheng@bjtu.edu.cn; jqliu@bjtu.edu.cn; wangwei1@bjtu.edu.cn).

G. Xu is with Tianjin Key Laboratory of Advanced Networking (TANK), College of Intelligence and Computing, Tianjin University, Tianjin, 300350, China (e-mail: losin@tju.edu.cn).

Z. Zhang is with IMT Lille Douai, Institut Mines-Tlcom, 59650, Villeneuve-dAscq, France (e-mail: zonghua.zhang@imt-lille-douai.fr).

H. Wang is with Department of Computer Science, Norwegian University of Science and Technology, Norway (e-mail: hawa@ntnu.no).

H. Dai is with Faculty of Information Technology, Macau University of Science and Technology, Macau (e-mail: hndai@ieee.org).

Y. Wu is with Department of Computer Science, College of Engineering, Mathematics and Physical Sciences, University of Exeter, Harrison Building, North Park Road, Exeter, EX4 4QF, United Kingdom (e-mail: y.l.wu@exeter.ac.uk).

W. Wang is the corresponding author.

anonymous in the mode. Temporary agreements and agreement rounds' messages in a group are recorded on an attribute-based blockchain. Contents of the agreements and messages are unreadable by other users or groups. However, final decisions of signal timing change are readable and verifiable to traffic signal controllers and all users. Moreover, Authentication Centers and Trace Managers are used to authenticate users' real identities beforehand, trace malicious users, and investigate malicious users' accountability. Standards of message integrity, legitimate generation, and reliability measurement are also required.
- The SCTSC mode is more efficient for a non-traffic-heavy road compared with the pre-timed modes, since it allows vehicles to pass quickly. The novel mode is more stable than actuated modes, since it is less likely to be affected by weather or overweight trucks.
- Each phase of SCTSC mode was simulated. Experimental results showed that the mode is efficient and practical in a real situation.

The rest of this paper is organized as follows. Section II introduces related works. In Section III, a framework of SCTSC mode is briefly described from problem description, attribute-based blockchain, and roles. Each phase of the mode is introduced in Section IV, while a concrete instantiation of the mode with complex formulas is given in Appendices A. Section V analyzes security and simulation results of the mode. Section VI concludes the paper.

II. Related work

A. Announcement Protocol

Announcement protocol allows infrastructures and vehicles to generate and broadcast messages in VANETs. With attaching importance to privacy[7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], novel protocols devoted to finding a trade-off between availability and privacy preservation.

Some researchers used threshold method to satisfy the standards. Ref. [18] achieved targets of threshold authentication by using group signature scheme. Based on identity-based aggregate signature, [19] and [20] proposed efficient secure and privacy-preserving authentication scheme in VANETs. However, they did not consider the case of long-term responsibility investigation. If an effect caused by a fake message appears far after the communication stage, because of a negative recording of message in vehicles and instruments, the investigation scheme is not usable.

Trust-based method and reputation-based method were also used in research. Ref. [21] designed Dempster-Shafter theory, reputation algorithm, and message forwarding criterion to evaluate the reliability of messages broadcasted in VANETs. Ref. [22] used Dempster-Shafer theory to evaluate a trust level of location findings. Based on Bayesian filter, a robust distributed reputation model was proposed in [23]. ARS[24] used pseudonyms and reputation levels to construct a centralized reputation system for VANETs. Hidden markov model (HMM) was used in [25] to build a reputation computation mechanism. However, trust-based and reputation-based methods have difficulty in dealing with Sybil attack, if an adversary has pretended to enhance his or her reputation. Creditcoin [26] combined credit with transaction and coin damping. In this way, Sybil attack is prevented. But, transaction records can be read by anyone, consequently leading to a risk of privacy leakage.

B. Blockchain technology

In 2008, Satoshi used a hash chain and proof of work in Bitcoin [27]. The technology that is drawn from Bitcoin is known as Blockchain Technology, or Distributed Ledger Technology (DLT).

A blockchain is a tamper-resistant data chain ordered by data blocks. Block contains a data area and a pointer to a previous block. The data area is used to store data (e.g., transactions). The pointer is used to guarantee the order among blocks and tamper-resistance. A variety of works about the blockchain network has been done in [28], [29], [30], [31].

Zerocoin [32] and Zerocash [33] are decentralized anonymous payment systems from Bitcoin. These payment systems used zero-knowledge proof to protect users' privacy. Ripple improved the consensus algorithm and proposed RPCA based on UNLs. Constructions in [34] and [35] realized Public Key Infrastructure (PKI) on blockchain to trace operations and change public keys. However, in present projects, data stored on the blockchain was either public to anyone or only available to the owner. Fine-Grained access control on data was missing.

C. Attribute-based encryption

Sahai and Waters proposed fuzzy identity based encryption [36] on the basic of identity based encryption (IBE)[37][38] in 2005. Goyal gave a concept and definition of attribute-based encryption [39]. Attribute-based encryption supports fine-grained non-interactive access control inherently. Only the user whose attribute set conforms to an access control policy has the right to access the decrypt data and get plaintext. In attribute-based encryption, enciphers do not need to focus on identities or the number of ciphertext receivers. In this way, the cost of encryption is decreased, and a flexible access control policy is provided.

Due to the differences in access control policies, attribute-based encryption consists of ciphertext policy attribute-based encryption (CP-ABE) [40] and key policy attribute-based encryption (KP-ABE) [41]. Access control policies in CP-ABE are related to ciphertexts, while keys are related to attribute sets. However, for KP-ABE, access control policies are keys correlation, while attributes set are ciphertexts correlation.

The algorithm in [42] realized access control policies in AND gate. Algorithms in [43] and [44] improved the efficiency of [42], and achieved hidden policies. The length of ciphertext stayed constant in [44]. However, policies realized by AND gate only support an AND operation between attributes. Thus, [45] and [46] constructed access policy with a tree structure in a more flexible way. The algorithm in [40] used LSSS access structure to represent access policies, but the time cost of encryption and decryption was increased linearly with the complexity of access structure.

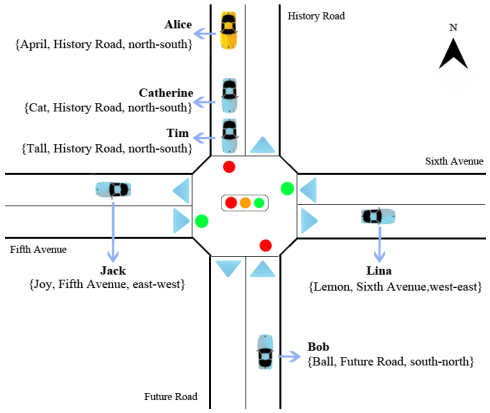

Fig. 1. Problem Description.

III. Problem Definition

This section introduces essential elements of the proposed SCTSC mode. Construction of the proposed method is provided in next section.

A. Problem Description

Figure 1 gives a simple example of the problem. The road is an example of driving on the right.

Alice drives into the History Road from north to south. The traffic light in front of her is red. However, no vehicles are driving in the approach lane of Fifth Avenue and Six Avenue. So, all vehicles waiting at approach lanes of History Road and Future Road waste their time. The traffic efficiency has vast potential to be increased. Moreover, if Alice is in a hurry (e.g., she has to drive to the hospital with her sick child), it is better for her to pass through the low traffic flow intersection as soon as possible.

A possible solution is to devolution the power of signal control to vehicles. Giving vehicles rights to determine the next direction and order of traffic is a good choice. Thus, a method of control-by-vehicles is used.

B. Attribute based Blockchain

This part discusses a novel blockchain structure, which attaches the goal of fine-grained access control on data.

1) Block and Chain Structure: As is used in Bitcoin and other similar projects, block structure is divided into two parts, a block header and a main block. A block header contains a unique index to distinguish each block, in which a hash function is commonly used. A main block is used to record some significant information (e.g., transactions) of a project.

For a block structure of an attribute-based blockchain, we retain some necessary fields described above, and a new structure is shown in Table I.

A chain consists of a series of ordered and deterministic blocks. Based on the block structure above, each new block is related to its previous block. Any modification to a previous block leads to a change of its hash value. All following blocks are influenced as well. Thus, the blockchain is tamper resistant.

TABLE I
Block Structure

Parts	Fields	Areas
Block Header	a block index a previous block hash a timestamp a signature	
Main Block	several message fields	message content related information

2) Nodes and Attributes: In general, users or infrastructures can be seen as nodes. Drivers draft temporary agreements to save their time. Infrastructures (e.g., traffic signal controllers) are also able to draft temporary agreements to improve traffic efficiency. There are two kinds of nodes in the blockchain, simple nodes and consensus nodes. Simple nodes have rights to read and generate new messages. However, they have no right to write messages into a blockchain directly. Consensus nodes take part in the process of consensus and have the right to read and write messages into a blockchain. However, they have no right to generate new messages. It is important to be aware that, in some cases, a node plays a role of both simple node and consensus node.

A node in attribute-based blockchain is identified not only by its anonymous identity (e.g., pseudonym) but also by a set of fuzzy identities (e.g., locations). Both anonymous identities and fuzzy identities are attributes.

As shown in Figure 1, each driver is a node. An attribute set of a node contains a pseudonym, a current location and a direction, for example, {*April, History Road, north-south*} for Alice. If Alice wants to control the traffic light signal only for the drivers in the same lane, she needs to encrypt and broadcast a message with an access policy {*History Road and north-south*}, instead of sending different encrypted messages to different drivers separately. Moreover, the privacy of a broadcast message and the related users should be protected.

Using attributes as essential characteristics give the blockchain a dynamic fine-grained access control on data. Nodes whose attribute set satisfied the same access policy are grouped (e.g., Alice, Catherine and Tim are in the same group). Interactions among them are recorded on the blockchain along with other groups' interactions through CP-ABE. However, nodes that are outside the group are not able to decrypt the encrypted interaction messages of this group since the access policy is not satisfied (e.g., Bob cannot read encrypted messages in Alice's group). Thus, the privacy is preserved in a group.

If the strict supervision is necessary for blockchain, messages that are not readable to consensus nodes will be discarded. If privacy needs more consideration than supervision, consensus nodes can input all messages into consensus algorithm blindly without knowing anything about contents.

From another point of view, attributes are able to divide into two groups.

- Static attributes: Entities' inherent features are static attributes. For example, a vehicle's engine number is a static attribute. Moreover, a person's name is used as a static attribute as well, since it is less likely to be changed

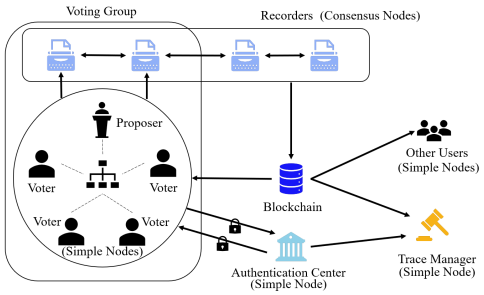

Fig. 2. General information stream in SCTSC mode.

in a short period. However, if privacy is more concerned, we should be cautious of using such static attributes. For CP-ABE, some parts of a private key that are related to static attributes do not need to be frequently updated. Thus, key distribution costs are reduced.

- Dynamic attributes: Frequently modified attributes are dynamic attributes. For example, the location information is a dynamic attribute, since it will change with moving car. For CP-ABE, some parts of a private key that are related to dynamic attributes need to be frequently updated.

C. Roles

Different roles used in SCTSC mode are introduced in this part. Figure 2 shows a general information stream in SCTSC mode.

Proposer: A proposer is an initiator of an agreement, who is responsible for drafting an agreement and organizing a proper voting group for the agreement. When voting is finished, a proposer is in charge of vote counting. A user is allowed to acts as different proposers in different voting groups. A proposer is a simple node in the attribute-based blockchain.

Voter: A voter replies to an agreement and votes for it. A user is allowed to acts as different voters in different voting groups. A voter is a simple node in the attribute-based blockchain.

Voting Group: Each agreement is related to a specific voting group. A voting group contains a proposer and several voters who are selected by the proposer with the help of access policies (a combination of attributes).

Recorder: A recorder has the right to read and write a blockchain. Besides, a recorder does not know real identities of proposers and voters, except for communication channels. In a voting group, a role of recorder must be different from proposers and voters. A recorder is a consensus node in an attribute-based blockchain. For example, drivers are proposers and voters, while infrastructures are recorders.

User: Proposers, voters, and recorders are called users. A user is allowed to read messages from a blockchain at any time. When joining SCTSC mode, a new user must request an Authentication Center to perform an authentication operation. A user can be a driver, a vehicle sensor or an infrastructure.

Authentication Center: An Authentication Center verifies users' real identities, gives attribute sets, distributes key pairs for signature, and distributes private keys for decryption to users. However, a center does not reject a user for any other reasons, such as the limitation on the number of participants. So, the proposed method is considered to be semi-centralized. Meanwhile, a user's pseudonym and signature public key are published to all users. A user is allowed to apply new signature key pair and the private encryption key with different pseudonyms, to guarantee long-term anonymous.

Trace Manager: A Trace Manager takes charge of tracing malicious participants or fraudulent messages. A Trace Manager gets fraudulent records from a blockchain and gets a malicious user's real identity by querying a pseudonym in an Authentication Center. The Trace Manager is secure and reliable.

System: A system is an environment in which the SCTSC mode is operated. It is generally considered as trustworthy and secure.

IV. Methodology

The novel mode is based on the method of control-by-vehicles. Vehicles driven in the same road and the same direction are grouped together. They get their keys from the Authority Center for the new group. The proposer sends a signal control message to voters in the group without the knowledge of the identities of the voters. Voters reply to the message to show their standpoints with pseudonyms and the fresh nonce in that message. The proposer collects the voters reply and make the final decision. All the messages, replies, and decisions are recorded on the blockchain. Vehicles do not communicate with each other directly while through the blockchain.

A. Basic Method

Security parameter: λ is a general security parameter, and κ is the security parameter used in attributed based encryption (briefly, bilinear group size).

Hash function: Choosing a collision resistance hash function for SCTSC mode with the form $CRH : \{0,1\}^* \rightarrow \{0,1\}^{O(\lambda)}$.

Statistically hiding commitment: Choosing a statistically hiding commitment for SCTSC mode with the form $\{COMM_s : \{0,1\}^* \rightarrow \{0,1\}^{O(\lambda)}\}_s$, in which s is the secret value of the commitment.

Digital Signature: Choosing a digital signature algorithm as follows,

$Sig = (Setup_{sig}, Keygen_{sig}, Sign_{sig}, Verify_{sig})$:

- $Setup_{sig}(1^\lambda) \rightarrow pp_{sig}$: Giving a security parameter λ, $Setup_{sig}$ generates a public parameter pp_{sig}.
- $Keygen_{sig}(pp_{sig}) \rightarrow (pk_{sig}, sk_{sig})$: Giving a public parameter pp_{sig}, $Keygen_{sig}$ generates a pair of keys (pk_{sig}, sk_{sig}) used to sign a message.
- $Sign_{sig}(sk_{sig}, m) \rightarrow \sigma$: Giving secret key sk_{sig} and message m, $Sign_{sig}$ generates a signature σ to message m.
- $Verify_{sig}(pk_{sig}, m, \sigma) \rightarrow b$: Giving a public key pk_{sig}, a message m and a signature σ, $Verify_{sig}$ verifies the relationship between message m and signature σ. If σ is the signature of the message m, $b = 1$. Or else, $b = 0$.

Ciphertext Policy Attribute based Encryption: Choosing a CP-ABE as follows,

$Att = (Setup_{att}, Keygen_{att}, Enc_{att}, Dec_{att})$:

- $Setup_{att}(1^\lambda) \rightarrow (PK, MK)$: Giving a security parameter λ, $Setup_{att}$ generates a public key PK and a master secret key MK.
- $Keygen_{att}(MK, S) \rightarrow SK$: Giving a master secret key MK and an attribute set S, $Keygen_{att}$ generates a set of secret key SK based on attributes S.
- $Enc_{att}(PK, M, T) \rightarrow CT$: Giving a public key PK, message m and an access tree T, Enc_{att} generates a ciphertext CT.
- $Dec_{att}(CT, SK) \rightarrow M$: Giving a ciphertext CT and the corresponding secret key SK, Dec_{att} decrypts the ciphertext and gets the plaintext M.

B. General Method

A setup phase is operated at the beginning of SCTSC mode. Then, users communicate with each other according to different phases. As shown in Figure. 3, there are four kinds of phases for a temporary agreement round. For a whole system, the agreement rounds are executed concurrently while the consistency is controlled by recorders through consensus algorithm. Processes in full line are operated in a sequential execution. The first group of processes in full line represents a drafting phase. The second group of processes in full line represents a reply phase. The third group of processes in full line represents a decision phase. Each group of processes in dashed line represents verification phase, which can be operated at any time.

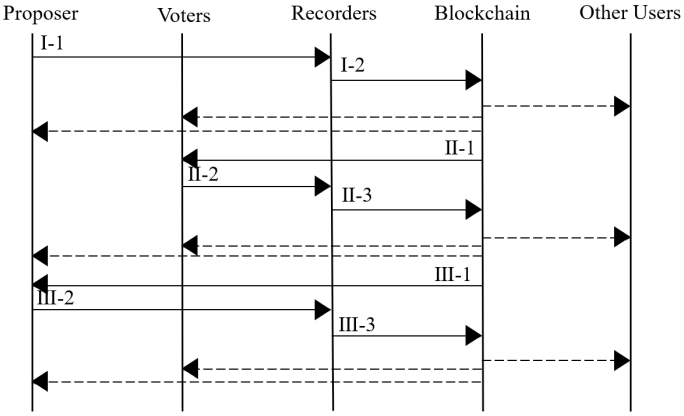

Fig. 3. Processes of SCTSC mode for a single agreement round.

1) Setup Phase: To install a SCTSC mode in a real situation, public parameters are generated in setup phase as shown in Algorithm 1. However, in an experimental environment, tasks of attribute distribution and key distribution are also taken during the setup phase, which is shown in Algorithm 2. Verification of a user's identity is also done in this phase. A secret key for CP-ABE and a key pair for signature are generated and distributed. Moreover, an attribute set of the system is updated due to new attributes of a new user.

Although the system's new attribute set is frequently updated in real time, users do not need to download the set in each time, since users are able to get attributes about location information from outside environments (e.g., road names and directions). Pseudonyms are useless in the construction of access policies during encryption.

Algorithm 1 Setup Phase in real.

Input: security parameter λ; security parameter κ; a set of attributes about road $att[\,]$;

Output: public key for CP-ABE PK, master secret key for CP-ABE MK, system's initial attribute set S_{system}, public parameter pp_{sig}

function SETUPPHASEREAL(λ, κ)
  $(PK, MK) \leftarrow Setup_{att}(1^\lambda)$
  $S_{system} \leftarrow att[\,]$
  $pp_{sig} \leftarrow Setup_{sig}(1^\kappa)$
  return $PK, MK, S_{system}, pp_{sig}$
end function

Algorithm 2 Setup Phase for each user.

Input: master secret key for CP-ABE, MK; system's initial attribute set, S_{system}; user's real identity, id; pseudonym, pse; user's attribute set, $S_{user}[\,]$; public parameter, pp_{sig}; using $*$ to represent all inputs above;

Output: system's new attribute set S_{system}, secret key for CP-ABE SK, keys for signature (pk_{sig}, sk_{sig})

function SETUPPHASEUSER($*$)
  $S_{system} \leftarrow S_{system} + S_{user}[\,]$
  $SK \leftarrow Keygen_{att}(MK, S_{system})$
  if id is verified **and** pse is unique **then**
    $(pk_{sig}, sk_{sig}) \leftarrow Keygen_{sig}(pp_{sig}, pse)$
  end if
  return $S_{system}, SK, pk_{sig}, sk_{sig}$
end function

2) drafting phase: In drafting phase, a proposer P drafts a new agreement for a group of voters, as the process of I-1 and I-2 in Figure. 3.

Choosing a proper voting group is needed to be considered first. A proposer does not choose voters by their identities or their pseudonyms, instead by attributes. So, a proposer does not need to communicate with others beforehand; this decreases the communication cost and keeps voters anonymous in this phase. A proposer chooses enough attributes to construct an access policy which is only satisfied by potential voters. The access policy is organized in the tree structure (e.g., access tree in [45]).

Each agreement needs a fixed number (i.e., threshold value) of voters to give affirmative votes. The threshold value should be decided at the drafting phase and stay unchangeable in the following phases. Otherwise, a malicious proposer is able to change the threshold value into a different value, which may disturb the mode. So, a commitment to the threshold value is required. A proposer chooses a secret value s for threshold value t (called affirmative votes amount) and computes $COMM_s(t)$ as a commitment to t. The commitment is published in this phase, while secret value s and threshold value t are kept without leakage until the decision phase completes. In decision phase, both of values are disclosed.

Algorithm 3 Drafting Phase.

Input: public key for CP-ABE, PK; system's attribute set, S_{system}; a new agreement, $agreement$; proposer's secret key of signature, $sk_{sig,P}$; the number of affirmative vote, t; using $*$ to represent all inputs above;

Output: a new agreement record $AgreementRecord$;

function DRAFTINGPHASE($*$)
  choose $S_{dra} \in S_{system}$ or choose S_{dra} from outside environment
  $T \leftarrow S_{dra}$
  $nonce \leftarrow Random()$
  $protext \leftarrow agreement + nonce$
  $CT_{protext} \leftarrow Enc_{att}(PK, protext, T)$
  $s \leftarrow Random()$
  $COMM_s(t) \leftarrow COMM(t, s)$
  $timestamp \leftarrow System.time()$
  $h_{AR} \leftarrow HASH(CT_{protext} + COMM_s(t) + timestamp)$
  $\sigma_P \leftarrow Sign_{sig}(sk_{sig,P}, h_{AR})$
  $AgreementRecord \leftarrow h_{AR} + CT_{protext} + COMM_s(t) + timestamp + \sigma_P$
  return $AgreementRecord$;
end function

Users are able to verify the two value after the decision phase, to make sure that the threshold value is the initial one committed in the drafting phase. But no one is able to get the threshold value before the decision phase, except for the proposer.

Another important issue is the method to distinguish valid voters and pretended voters. Obviously, only valid voters in a voting group have the right to vote on the related agreement. A voter has to give an evidence to a proposer, which indicates the fact that the voter's attribute set satisfies the access policy. So, a fresh nonce encrypted with the access policy is needed. The nonce is generated for each agreement randomly and uniquely. So, only valid voters are able to get a valid nonce. Pretended voters are difficult to predict or guess the nonce.

A timestamp is needed. An index is given to each agreement to identify an agreement more simply. Moreover, a signature of proposer is used for responsibility investigation. As shown in Algorithm 3, this phase outputs a new agreement record.

A proposer sends AgreementRecord to a recorder. Recorder packs the record into a new block and adds the block into the blockchain via a cooperation consensus algorithm among recorders.

3) Reply Phase: In reply phase, a voter V reads agreements from the blockchain, as the process of II-1 in Figure. 3. An access tree of each agreement is checked by the voter. If the access tree is satisfied by the voter's attribute set, the voter needs to vote on the agreement, as the process of III-2 in Figure. 3.

For such an agreement, the voter verifies signatures of the related block and the AgreementRecord. The content of the agreement and nonce are gotten after decrypting the ciphertext in AgreementRecord. Then, the voter decides whether to vote. An action of voting represents an affirmative vote. An action of ignoring represents a dissenting vote. If the voter is for the agreement, a reply is needed. Otherwise, the voter does nothing.

Algorithm 4 Reply Phase.

Input: public key for CP-ABE, PK; an agreement record, $AgreementRecord$; voter's secret key of signature, $sk_{sig,V}$; proposer's public key of signature, $pk_{sig,P}$; voter's secret key for CP-ABE, SK_V; system's attribute set, S_{system}; voter's pseudonym, pse_V; using $*$ to represent all inputs above;

Output: a new reply record $ReplyRecord$;

function REPLYPHASE($*$)
  $(h_{AR}, CT_{protext}, COMM_s(t), timestamp, \sigma_P) \leftarrow AgreementRecord$
  $h \leftarrow HASH(CT_{protext} + COMM_s(t) + timestamp)$
  if $h == h_{AR}$ **then**
    $b \leftarrow Verify_{sig}(pk_{sig,P}, h, \sigma_P)$
    if b is true **then**
      $M \leftarrow Dec_{att}(CT_{protext}, SK_V)$
      $(agreement, nonce) \leftarrow M$
      choose $S_{rep} \in S_{system}$ or choose S_{rep} from outside environment
      $T_{rep} \leftarrow S_{rep}$
      $reptext = nonce + pse_V$
      $CT_{reptext} \leftarrow Enc_{att}(PK, reptext, T_{rep})$
      if agree with the $agreement$ **then**
        $vote \leftarrow$ Random()
        $timestamp \leftarrow System.time()$
        $h_{PR} \leftarrow HASH(h_{AR} + CT_{reptext} + vote + timestamp)$
        $\sigma_V \leftarrow Sign_{sig}(sk_{sig,V}, h_{PR})$
        $ReplyRecord \leftarrow h_{PR} + h_{AR} + CT_{reptext} + vote + timestamp + \sigma_V$
        return $ReplyRecord$
      end if
    end if
  end if
end function

A ciphertext of a concatenation string ($reptext := \{nonce||pse\}$) is used in reply. Thus, a pretended voter (i.e., malicious voter) cannot disturb the agreement round with a correct nonce unless the nonce is leaked by voters. A timestamp, signature and hash index are also needed in this phase. The reply phase is showed as Algorithm 4.

A voter sends his or her ReplyRecord to a recorder. Recorder packets the record and adds the new block into the blockchain by cooperation consensus algorithm, as the process of III-3 in Figure. 3.

4) Decision Phase: In decision phase, a proposer collects all ReplyRecords related to the agreement, as the process of III-1 in Figure. 3. The signatures are verified first. Then, the proposer decrypts ciphertext of each ReplyRecord and gets a set of $reptext$. For each ReplyRecord, the proposer checks the pseudonym in $reptext$ and the pseudonym in signature. If these pseudonyms are one-to-one correspondence, the proposer checks uniqueness of pseudonyms among all Re-

Algorithm 5 Decision Phase.

Input: public key for CP-ABE, PK; a set of reply records, $ReplyRecord[\ ]$; proposer's secret key of signature, $sk_{sig,P}$; related voters' public key of ssignature, $pk_{sig,V}[\,]$; proposer's secret key for CP-ABE, SK_P; the nonce of the related agreement, $nonce$; the secret value of the related agreement, s; the threshold value of the related agreement, t; using $*$ to represent all inputs above;

Output: a new decision record $DecisionRecord$;

function DECISIONPHASE($*$)
  for $i = 0 \rightarrow ReplyRecord[\ ].size - 1$ **do**
    $(h_{PR,i}, h_{AR,i}, CT_{reptext,i}, vote_i, timestamp_i, \sigma_{V,i}) \leftarrow ReplyRecord[i]$
    $h_i \leftarrow HASH(h_{AR,i} + CT_{reptext,i} + vote_i + timestamp_i)$
    if $h_i == h_{PR,i}$ **then**
      $b_i \leftarrow Verify_{sig}(pk_{sig,V,i}, h_i, \sigma_{V,i})$
      if b_i is true **then**
        $M_i \leftarrow Dec_{att}(CT_{reptext,i}, SK_P)$
        $(pse_{V,i}, nonce_i) \leftarrow M_i$
        if $nonce_i == nonce$ **and** $pse_{V,i}$ equals to the pse in $pk_{sig,V,i}$ **and** $pse_{V,i}$ is not repetitive **then**
          $list_{RP} += ReplyRecord[i]$
        end if
      end if
    end if
  end for
  $t^{'} \leftarrow list_{RP}.size()$
  $timestamp \leftarrow System.time()$
  $h_{DR} \leftarrow HASH(h_{AR} + list_{RP} + s + t + t^{'} + timestamp)$
  $\sigma_P \leftarrow Sign_{sig}(sk_{sig,P}, h_{DR})$
  $DecisionRecord = h_{DR} + h_{AR} + list_{RP} + s + t + t^{'} + timestamp + \sigma_P$
  return $DecisionRecord$
end function

plyRecords. The proposer discards all repetitive ReplyRecords and reserves the first one. So, the selected set of ReplyRecords contains affirmative votes from different voters without repetition.

The proposer counts the number of affirmative votes. If the number does not reach the expected threshold value t, the proposer keeps on collecting until the goal is achieved or time is run out. Otherwise, the proposer writes all indexes of affirmative votes into a new list $list_{RP}$. Meanwhile, the proposer publishes the secret value s and threshold value t. Hash indexes, number of affirmative votes $t^{'}$, a timestamp, and a signature are also needed in this phase. The decision phase is shown as Algorithm 5.

A proposer sends DecisionRecord to a recorder, as the process of III-2 in Figure. 3. Recorder packets the record into a new block and adds the block into the blockchain by cooperation consensus algorithm, as the process of III-3 in Figure. 3.

5) Verification Phase: In verification phase, users verify the process from drafting phase to decision phase for an agreement. Different records are related by h_{AR}, h_{RR}, h_{DR} and $list_{RP}$. So, users are able to find all corresponding records easily.

6) Signal Control: A passed agreement with the most affirmative votes is considered first in a signalized intersection. And the related lane has priority of starting.

However, the real situation is more complicated because of different types of traffic lanes. A signalized intersection with only straight lane and four approach lanes is the simplest case. Let us explain how our proposed method works in general scenarios based on this simple case.

If there are more than two passed agreements for the same road and the same direction, the agreement with a larger number of affirmative votes is accepted. Assuming an intersection has four approach lanes, so there will be up to four kinds of passed agreements at the same time with affirmative votes number of $t^{'}_{north-south}$, $t^{'}_{south-north}$, $t^{'}_{east-west}$, and $t^{'}_{west-east}$. These four passed agreements are divided into two groups, a NS-group $\{t^{'}_{north-south}, t^{'}_{south-north}\}$ and a WE-group $\{t^{'}_{east-west}, t^{'}_{west-east}\}$, based on the conflict approach lanes. Thus, new affirmative votes number of NS-group and WE-group are $t^{'}_{ns} = t^{'}_{north-south} + t^{'}_{south-north}$ and $t^{'}_{we} = t^{'}_{east-west} + t^{'}_{west-east}$. The group that has larger affirmative votes number has the priority of starting. If both the two values are equal, traffic lights follow the current fixed cycle length of signal control.

V. SECURITY ANALYSIS AND SIMULATION

In this section, we first discuss several important security issues in SCTSC mode. Then, time cost consideration is analyzed by game theory method. Finally, the experiment result is given.

A. Security Analysis

Anonymity: Records saved on the attribute-based blockchain do not contain users' identities. Thus, group members and the public do not know the others' identities. Users' real identities are only known to the Authentication Center. Publishing a user's identity to the Authentication Center is necessary since the Authentication Center needs to distribute proper encryption and signature key pairs to users. The Authentication Center gives a user some attributes by a user's real identity. Although a signature key pair is related to a user's pseudonym, a user does not need to worry about anonymity too much. A user is able to apply a new key pair at any time with a different pseudonym. However, for the whole scheme, each pseudonym should be unique. If a user changes his or her pseudonym frequently, it is tough for an adversary to trace the user's records and relate the pseudonym with a real person.

Untraceability: Assuming a malicious user Malice has already found a record a of a real person Alice on the blockchain, Malice wants to find another record of Alice on the blockchain. Since Malice knows the record a, he is able to get the corresponding pseudonym in the record. Records signed with the same pseudonym on the blockchain are easy to find. However, records signed by Alice but with a different

pseudonym are not able to be recognized. Because for all data recorded on the blockchain, there is no evidence to support the relationship between two pseudonyms unless Alice gives her pseudonyms of some relations. If Alice changes her pseudonym after each record she sends, Malice can get nothing except the current record.

Man-in-Middle Attack Resistance: Assuming a malicious user Malice is listening between a proposer Alice and a voter Bob, he tries to tamper the records sent by Alice and Bob. If Malice modifies any field of records, the hash value will be changed, and the corresponding signature will not be correct anymore. So, Malice needs to forge a signature for the modification. However, the private key of signature is only known to the user and the Authentication Center which is considered to be trusted. Malice cannot create a new valid signature of Alice or Bob for his modification. Any modification without a valid signature can be easily identified by both Alice and Bob, even other users who do not know the real contents.

Reply Attack Resistance: Records published on the blockchain have two fields, hash index, and timestamp. Hash index is computed from the whole record except the signature. Different records have different hash index values generally since we have already known that hash collision appears with low probability. A timestamp is a field used to store the creation time of a record. A recently created record has an absolute fresh timestamp. Records are able to be divided into many agreement rounds. A reply attack to those agreement rounds that have already been finished has less meaning since the decision has already been made and no records will be considered by the voting group anymore. So, if a malicious user Malice tries to launch a reply attack in the scheme with potential malicious effect, he should attack those agreement rounds that are running in reply phase, or tries to start a new agreement round with an old AgreementRecord. Assuming Alice is a proposer of a voting group, and Bob is a voter in the same voting group. For the first kind of attack, Malice has to copy the ReplyRecord of Bob and send to a recorder. Since repetitive ReplyRcords are rejected by proposers, a reply attack on ReplyRecord does not work. For the second attack, Malice has to copy the AgreementRecord of Alice and send to a recorder. Voters, like Bob, will not reply to the agreement since he has already reply an agreement with the same hash index.

Fine-grained Access Control: Only a few of data recorded on the blockchain is readable to all users. Sensitive data is encrypted with CP-ABE. Only a user whose attribute set satisfies the access policy of the ciphertext is able to decrypt it. So, fine-grained access control is inherently realized. Users record their data on the blockchain together, but only the data belongs to his or her is able to be visited.

Non-repudiation: A signature is contained in each record and a block. In the consensus process, recorders check the signature of each record. Those records with invalid signature are denied by recorders. The records saved on the blockchain cannot be modified, since the tamper-resistant feature of the blockchain. Each record of an agreement round is recorded honestly and integrated on the blockchain. A whole process of an agreement round is able to be reconstructed based on these records. Thus, SCTSC mode is non-repudiation.

Sybil Attack Detectable: Among all anonymous voting schemes, it is difficult to trace real identities in anonymous phases, which leads to the chances of launching Sybil attacks. With the help of blockchain and the inherent feature of unmodifiable, all operations of anonymous identities and the corresponding records are saved on the blockchain and stayed unchangeable. If a malicious voter votes twice for a single agreement with different anonymous identities gotten from the Authentication Center, a Trace Manager are able to trace the malicious voter with the assistance of the Authentication Center. A severe punishment is excepted. Several restrictions or modifications on SCTSC mode are also able to deal with the Sybil attack in a similar way. For example, restricting the Authentication Center to authenticate only ten pseudonyms for each real identity at the same time, and let the later authentication apply covers the old ones. So, the anonymous identities of each person are limited. If the limitation on the minimum number of affirmative votes (larger or much larger than ten) is also set, a single or a small group of malicious voters are difficult to launch a successful Sybil attack.

B. Time Cost Consideration

An essential link in supporting the SCTSC mode is actively voting. That is, the final result should have almost the same percentage of affirmative votes as the voters' real thought. The most common way of dealing with this problem is through incentives. An incentive mechanism encourages participants by giving rewards. In SCTSC mode, an incentive mechanism is contained inherently, called travel time cost. We analysis the mechanism in a game theory model.

Assuming Alice and Bob are two voters in the same voting group in SCTSC mode, they agree with the same agreement. In reply phase, both of them have two choices. One is to reply honestly. That is, Alice or Bob votes an affirmative vote for the agreement. We called this behavior as *to vote*. Another one is to reply negatively. In other words, Alice or Bob stays in silence. We called this behavior as *not to vote*. A vote action leads to the cost of interaction time since a voter need to receive and send messages. A passed agreement saves the cost of travel time to voters, since the traffic light turns green as long as voters arrive at the intersection. We define interaction time as $\mu(\mu > 0)$ and travel time as $\omega(\omega > \mu > 0)$.

Alice and Bob do not know each other's choice while they vote for an agreement. Assuming that if both Alice and Bob vote affirmative votes, the agreement is passed. The time cost is $\mu - \omega$, since they save the cost of travel time by paying out an interaction time. If Alice and Bob do not choose to vote, the agreement is failed. The time cost is 0. If one of them gives an affirmative vote, the agreement is passed. The time cost is $\mu - \omega$ and $-\omega$ respectively.

Base on the above assumptions, we have the description in Table II. The horizontal line of *to vote* and *not to vote* are Bob's choices, and the vertical line of *to vote* and *not to vote* are Alice's choices. In each grid, the first equation is Alice's time cost, and the second equation is Bob's time cost.

TABLE II
TIME COST CONSIDERATION

	to vote	not to vote
to vote	$\mu-\omega, \mu-\omega$	$\mu-\omega, -\omega$
not to vote	$-\omega, \mu-\omega$	$0, 0$

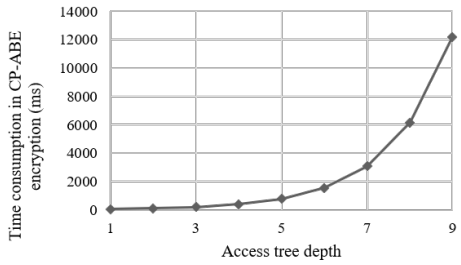

Fig. 4. Time consumption in CP-ABE encryption with different access tree depth.

There is no dominant strategy in Table II. If Alice thinks that Bob will choose *to vote*, Alice will choose *not to vote*. If Alice thinks Bob will choose *not to vote*, Alice will choose *to vote*. So, Alice's choice is different based on Bob's choice.

However, as we have assumed, Alice does not know Bob's choice. Her guesses may lead her into a worse situation. For example, Alice guesses Bob will choose *to vote*, and she chooses *not to vote*. However, Bob has the same train of thought with Alice, and he chooses *not to vote*. In the end, both Alice and Bob will get nothing. So, when considering the rewards of both choices, *to vote* is the best strategy for Alice. Because no matter what Bob chooses, Alice does not need to worry about getting 0. Also, the similar scenario applies to Bob.

One efficient way to enhance the best strategy is to widening the gap between interaction cost μ and travel cost ω. If interaction cost μ is far lower than travel cost ω, there is no difference between choosing $\mu-\omega$ and $-\omega$. However, the risk of choosing *not to vote* becomes more considerable.

C. Experiment Analysis

We use the elliptic curve in our program to construct bilinear pairs. We develop local operations of each phase with the help of library bcprov-jdk15on-158, commons-codec-1.7, jpbc-api-1.2.1, jpbc-plaf-1.2.1 and libbswabe-0.9 on Java Runtime Environment 1.8 with an Intel Core i5-2400 CPU.

Since the encryption of an agreement is related to an access tree, the relationship between encryption time and access policy complexity needs to be considered. Let each parent node have two child nodes, the relationship between encryption time and access tree depth is shown in Figure 4. For example, the access tree depth of a policy $\{$*Sixth Avenue or Fifth Avenue*$\}$ is two, and the access tree depth of a policy $\{$*(Sixth Avenue and west-east) or (Fifth Avenue and east-west)*$\}$ is three. From Figure 4, we find that the encryption time increases with the depth of the access tree. Let the access tree depth be four, the relationship between encryption time and attribute complexity is shown in Figure 5. We define attribute complexity as the number of child nodes for each non-leaf node. But, if each non-leaf node only has one child node, an access tree with depth of four is meaningless. From Figure 5, we know that encryption time increases with attribute complexity.

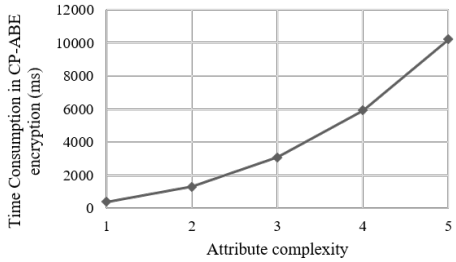

Fig. 5. Time consumption in CP-ABE encryption with different attribute complexity.

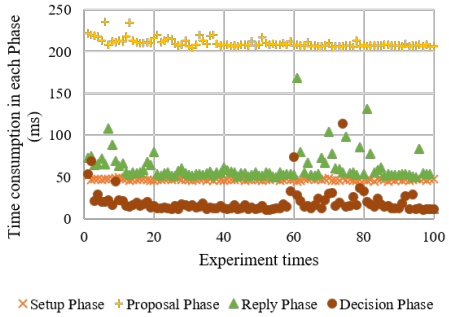

Fig. 6. Time consumption for each phase in SCTSC mode.

TABLE III
AVERAGE TIME CONSUMPTION IN SCTSC MODE.

Phase	Time Consumption (ms)
Setup Phase	46.89
Drafting phase	210.30
Reply Phase	60.93
Decision Phase	19.47

The SCTSC mode is controlled by vehicles. As discussed in the problem description, the novel mode is efficient when the traffic of the road is not heavy. If the traffic of the road is heavy and there are too many cars driven in different directions, this kind of controlled-by-vehicles mode may cause conflict and disorder. If the road is in a rush hour, the signal controller is suggested to turn back to a pre-timed mode or other similar fixed modes, since the mode change operation exists in most of the modern traffic signal controllers. Thus, in order to simulate a non traffic-heavy road, we choose to set a small number of vehicles.

To simplify the experiment, we assume each voting group has five voters. The result of the experiment does not contain consensus and message transmission time since the time is depended mainly on the network environment.

The average time of setup phase is 46.89ms, as shown in Table III and Figure 6. In setup phase, the system generates public key and master secret key for CP-ABE, registers new

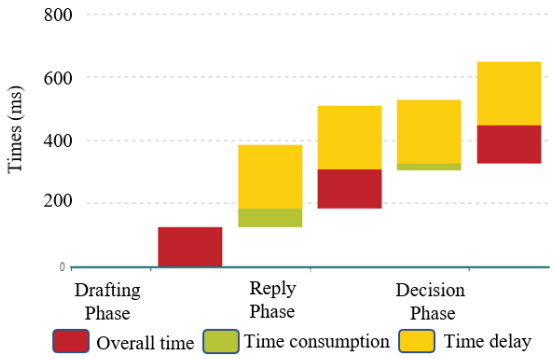

Fig. 7. Idea time cost in real situation.

users and distributes secret key for CP-ABE and key pairs for signature. In the experiment, we run the above processes step by step and record the operation time. However, in a real situation, the above processes are operated separately. For example, the system generates public key and master secret key for CP-ABE at the beginning. When a new user applies for registration, the next two steps are operated. When an old user wants to use a new pseudonym or change attributes, the system runs some part of the last two steps. So, in a real situation, setup phase is able to have less time cost. And the operations are able to become more dispersion.

The average time of drafting phase is 210.30ms, as shown in Table III and Figure 6. In drafting phase, encryption occupies a large portion of time cost, which depends on the complexity of an access tree. An access tree with deeper depth and wider breadth leads to more time cost of encryption. However, in a real situation, drafting phase is able to be operated previously, since this phase does not contain any interaction. A proposer is able to finish all computing locally.

The average time of reply phase is 60.93ms, as shown in Table III and Figure 6. In reply phase, five voters reply to the same agreement. However, because of time delay and other reasons of the network environment, in a real situation, the time cost of reply phase should be more than the experiment average time cost.

The average time of decision phase is 19.47ms, as shown in Table III and Figure 6. In decision phase, a proposer collects affirmative votes and makes a decision. In our experiment, the operations start from the beginning of this phase. However, in a real situation, some of the operations are able to start as long as a ReplyRecord is received. For example, a proposer decrypts the first vote while waiting for the second vote.

Considering the real situation described above, Figure 7 shows the idea time cost in a real situation. In Figure 7, there are three types of time. Time consumption is the experiment result shown above. The time delay is a total of consensus time (92.4ms, as simulated in [26]) and a desired average delay of the network (30ms). Overall time represents other kinds of delay, such as a delay caused by operation exception.

D. Applicable Scene

In the existing pre-timed mode, the signal cycle consists of a set of fixed value. If the traffic has some sudden change, the pre-timed mode needs human control to change the signal. Obviously, for a modern metropolis, it is difficult to reach the goal without delay. Comparing with the pre-timed mode, the SCTSC mode could adjust the signal cycle dynamically. And the novel mode reduces the burden of the management center.

In the existing sensor-based mode, signal control relies on sensor sensitivity. Photoelectric sensors are susceptible to the weather, while pressure sensors are easily damaged by overloaded vehicles. Comparing with the sensor-based mode, the SCTSC mode is not affected by the weather, and the maintenance was spread out among the vehicles, which is more efficient.

The limitations are not severe to the role assignment or amounts, since all kinds of vehicles may have an emergency. But the signal controller must be contained in the groups around the intersection, since the signal controller must know the decision and executes the signal change. The public infrastructure (e.g., signal controllers or smart parking charge units, the same as the notion of roadside units in VANET) controlled by the traffic management department is suggested to be set as recorders, since the growth of the blockchain should be stable.

As discussed in the section above, the SCTSC mode is especially efficient for a not heavy road. For a road with heavy traffic, this kind of controlled-by-vehicles mode may generate frequent orders to the controller. So, it is difficult for the controllers to change the signal. In this case, the signal controller is suggested to turn back to a pre-timed mode or other similar fixed modes, since the average waiting time stays almost the same.

For a pretty long road, some vehicles (e.g., a vehicle has just driven into the road) may not pass through the intersection in time. There are three kinds of solutions to solve the problem. The first one is to propose the signal change agreement in a proper time (e.g., when the vehicle is driven into the middle of the road). The second one is to propose a new signal change agreement to extend the time of passing if the vehicle is in an emergency. The third one is to change the passing time of the intersection into a more proper value in order to give vehicles enough time to pass through the road and the intersection.

Generally, in main urban areas of cities, drivers cannot drive their vehicles fast due to the reason of traffic congestion and speed limitations (e.g., 50 kilometers per hour in China and 15 miles per hour in New York City). Let the ideal time cost of an agreement be 600 ms. A decision is able to reach during vehicle travel of fewer than 10 meters. We randomly measured the length of more than 200 roads with signalized intersections in the main urban area of Beijing. The result shows that the average road length between two signalized intersections is around 383 meters. A decision is reachable on the average road length. So, the proposed SCTSC mode is efficient and practical in a real situation.

VI. Conclusion

In this work, we proposed a novel semi-centralized traffic signal control mode (SCTSC mode) for signalized intersection with attribute based blockchain in IoVs. Combining CP-ABE

with blockchain technology, the protection of messages and users' identities are achieved while ensuring public verification and responsibility investigation. Through extensive experiments, the total time cost of local operations in SCTSC mode is 290.70 ms. However, with a pre-execution of drafting phase, the local operation time decreases to 80.40 ms. To conclude, SCTSC mode is practical for signalized intersection in the scenario of IoVs.

In future work, we plan to decrease the interactions and encryption cost in SCTSC mode. Designing more effective modes or protocols is also being investigated.

APPENDIX A
A CONCRETE INSTANTIATION OF SCTSC MODE

A. COMM and CRH from Hash

We instantiate COMM and CRH via SHA256. That is, $COMM_s(t) = SHA256(s||t)$, and CRH as $SHA256(*)$ for $* \in \{0,1\}^{512}$.

B. Sig from SM2

For digital signature, we use SM2 to realize a secure and efficient signature based on ECC.

C. CP-ABE from BSWABE

On the construction of SCTSC mode, the attribute-based encryption method in [45] is used, since the original method is applied to most of demands of SCTSC mode.

System initializes an attribute set S for all users' attributes. Then, system chooses bilinear group G_0, G_1, and bilinear mapping $e : G_0 \times G_0 \to G_1$, while G_0 has an order p and a generator g.

$Att = (Setup_{att}, Keygen_{att}, Enc_{att}, Dec_{att})$:

- $Setup_{att}(1^\lambda) \to (PK, MK)$: Choosing $\alpha, \beta \in Z_p$ randomly to get both $PK = (G_0, g, h = g^\beta, e(g,g)^\alpha)$ and $MK = (\beta, g^\alpha).PK$ is published. MK is kept secret.
- $Keygen_{att}(MK, S) \to SK$: Let S be an attribute set of a user. Choosing $\xi \in Z_p$ and $\xi_j \in Z_p (j \in S)$ randomly, $SK = (D = g^{(\alpha+\xi)/\beta}, D_j = g^\xi H(j)^{\xi_j}, D'_j = g^{\xi_j}, j \in S)$. SK is distributed to a related user.
- $Enc_{att}(PK, M, T) \to CT$: Let Y represents the leaf node set of the access tree T, $CT = (T, \tilde{C} = Me(g,g)^{\alpha\rho}, C = h^\rho, C_y = g^{f_y(0)}, C'_y = H(att(y))^{f_y(0)}, y \in Y)$, while $att(y)$ is an attribute related to a node y.
- $Dec_{att}(CT, SK) \to M$: Let ζ be a node in the access tree T. And,

$$DecryptNode(CT, SK, \zeta) = \begin{cases} \dfrac{e(D_i, C_\zeta)}{e(D'_i, C'_\zeta)} = e(g,g)^{\xi f_\zeta(0)}, & i \in S_V \\ \perp, & i \notin S \end{cases} \quad (1)$$

For each child node z of a non-leaf node ζ, let $F_z = DecryptNode(CT, SK, z)$. Let S_ζ be a set with size k_ζ at node ζ, and have $F_z \neq \perp$ for each child node z. If any set described above does not exist, the process of decryption is stopped. Or else,

$$F_\zeta = \prod_{z \in S_\zeta} F_z^{\triangle_{i,S'_\zeta}(0)} = e(g,g)^{\xi f_\zeta(0)} \quad (2)$$

where,

$$i = index(z) \quad (3)$$

$$S'_\zeta = \{index(z) : z \in S_\zeta\} \quad (4)$$

$$\triangle_{i,S'_\zeta}(x) = \prod_{j \in S'_\zeta, j \neq i} \frac{x-j}{i-j} \quad (5)$$

If a user's attribute set S satisfy access tree T, the user is able to get M as follows,

$$\begin{aligned} A &= DecryptNode(CT, SK, \varrho) \\ &= e(g,g)^{\xi f_\varrho(0)} = e(g,g)^{\xi\rho} \end{aligned} \quad (6)$$

$$M = \frac{\widetilde{C}}{e(C,D)/A} \quad (7)$$

REFERENCES

[1] R. Timothy, "A smart city in China tracks every citizen and yours could too — New Scientist," 2017.

[2] S. Schaefer, C. Harrison, N. Lamba, and V. Srikanth, "Smarter Cities Series: Understanding the IBM Approach to Traffic Management." [Online]. Available: http://files.cnblogs.com/files/menghe/Intelligent_Transportation_redp4737.pdf

[3] Siemens, "Intelligent Transportation," 2017. [Online]. Available: https://www.siemens.com/us/en/home/company/topic-areas/intelligent-transportation.html

[4] S. Annie, "Seattle to Install Siemens Software That Intelligently Syncs and Manages City's Traffic System to Increase Visibility and Reduce Congestion — Siemens USA Newsroom," 2016. [Online]. Available: http://news.usa.siemens.biz/press-release/mobility/seattle-install-siemens-software-intelligently-syncs-and-manages-citys-traffi

[5] Toronto, "Toronto Road Restrictions," 2012. [Online]. Available: https://www1.toronto.ca/wps/portal/contentonly?vgnextoid=c5e6e69ae554e410VgnVCM

[6] Q. Li, A. Malip, K. M. Martin, S. L. Ng, and J. Zhang, "A reputation-based announcement scheme for VANETs," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 9, pp. 4095–4108, 2012.

[7] B. Parno and A. Perrig, "Challenges in Securing Vehicular Networks," in *Proceeding Workshop on hot topics in networks*, Meryland,USA, 2005, pp. 1–6. [Online]. Available: http://conferences.sigcomm.org/hotnets/2005/papers/parno.pdf

[8] F. Dötzer, "Privacy Issues in Vehicular Ad Hoc Networks," in *Proceeding International Workshop on Privacy Enhancing Technologies*, 2005, pp. 197–209.

[9] H. Hartenstein and K. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *IEEE Communications Magazine*, vol. 46, no. 6, pp. 164–171, jun 2008.

[10] H. Zhu, C. Fang, Y. Liu, C. Chen, M. Li, and X. S. Shen, "You Can Jam But You Cannot Hide: Defending Against Jamming Attacks for Geo-Location Database Driven Spectrum Sharing," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 10, pp. 2723–2737, oct 2016.

[11] H. Hamssa, S. Abed, B. Carole, and L. Anis, "VANet security challenges and solutions: A survey," *Vehicular Communications*, vol. 7, pp. 7–20, jan 2017.

[12] H. Li, H. Zhu, and D. Ma, "Demographic Information Inference through Meta-Data Analysis of Wi-Fi Traffic," *IEEE Transactions on Mobile Computing*, vol. 17, no. 5, pp. 1033–1047, may 2018.

[13] L. Zhou, S. Du, H. Zhu, C. Chen, K. Ota, and M. Dong, "Location Privacy in Usage-Based Automotive Insurance: Attacks and Countermeasures," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 1, pp. 196–211, jan 2019.

[14] X. Zeng, G. Xu, X. Zheng, Y. Xiang, and W. Zhou, "E-AUA: An Efficient Anonymous User Authentication Protocol for Mobile IoT," *IEEE Internet of Things Journal*, pp. 1–1, 2018.

[15] G. Xu, Y. Zhang, A. K. Sangaiah, X. Li, A. Castiglione, and X. Zheng, "CSP-E2: An abuse-free contract signing protocol with low-storage TTP for energy-efficient electronic transaction ecosystems," *Information Sciences*, vol. 476, pp. 505–515, feb 2019.

[16] G. Xu, Y. Cao, Y. Ren, X. Li, and Z. Feng, "Network Security Situation Awareness Based on Semantic Ontology and User-Defined Rules for Internet of Things," *IEEE Access*, vol. 5, pp. 21 046–21 056, 2017.

[17] G. Xu, J. Liu, Y. Lu, X. Zeng, Y. Zhang, and X. Li, "A novel efficient MAKA protocol with desynchronization for anonymous roaming service in Global Mobility Networks," *Journal of Network and Computer Applications*, vol. 107, pp. 83–92, apr 2018.

[18] J. Shao, X. Lin, R. Lu, and C. Zuo, "A Threshold Anonymous Authentication Protocol for VANETs," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1711–1720, mar 2016.

[19] L. Zhang, C. Hu, Q. Wu, J. Domingo-Ferrer, and B. Qin, "Privacy-Preserving Vehicular Communication Authentication with Hierarchical Aggregation and Fast Response," *IEEE Transactions on Computers*, vol. 65, no. 8, pp. 2562–2574, aug 2016.

[20] L. Zhang, Q. Wu, J. Domingo-Ferrer, B. Qin, and C. Hu, "Distributed Aggregate Privacy-Preserving Authentication in VANETs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 3, pp. 516–526, mar 2017.

[21] Z. Cao, Q. Li, H. W. Lim, and J. Zhang, "A multi-hop reputation announcement scheme for VANETs," in *Proceedings of 2014 IEEE International Conference on Service Operations and Logistics, and Informatics*. IEEE, oct 2014, pp. 238–243.

[22] K. Sharma and B. K. Chaurasia, "Trust Based Location Finding Mechanism in VANET Using DST," in *2015 Fifth International Conference on Communication Systems and Network Technologies*. IEEE, apr 2015, pp. 763–766.

[23] Y. Begriche, R. Khatoun, L. Khoukhi, and C. Xiuzhen, "Bayesian-based model for a reputation system in vehicular networks," in *2015 International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC)*. IEEE, aug 2015, pp. 1–6.

[24] L. M. S. Jaimes, K. Ullah, and E. dos Santos Moreira, "ARS: Anonymous reputation system for vehicular ad hoc networks," in *2016 8th IEEE Latin-American Conference on Communications (LATINCOM)*. IEEE, nov 2016, pp. 1–6.

[25] A. Shrivastava, K. Sharma, and B. K. Chaurasia, "HMM for reputation computation in VANET," in *2016 International Conference on Computing, Communication and Automation (ICCCA)*. IEEE, apr 2016, pp. 667–670.

[26] L. Li, J. Liu, L. Cheng, S. Qiu, W. Wang, and X. Zhang, "CreditCoin:A Privacy-Preserving Blockchain-based Incentive Announcement Network for Communications of Smart Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, pp. 2204–2220, 2018.

[27] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," *Www.Bitcoin.Org*, 2008.

[28] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," in *2017 IEEE International Congress on Big Data (BigData Congress)*. IEEE, jun 2017, pp. 557–564.

[29] Z. Zheng, S. Xie, H. N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: a survey," *International Journal of Web and Grid Services*, vol. 14, no. 4, p. 352, 2018.

[30] Z. Li, W. Wang, G. Liu, L. Liu, J. He, and G. Huang, "Toward open manufacturing: A Cross-Enterprises Knowledge and Services Exchange Framework based on Blockchain and Edge Computing," *Industrial Management & Data Systems*, vol. 118, no. 1, pp. 303–320, feb 2018.

[31] Z. Li, A. V. Barenji, and G. Q. Huang, "Toward a blockchain cloud manufacturing system as a peer to peer distributed network platform," *Robotics and Computer-Integrated Manufacturing*, vol. 54, pp. 133–144, dec 2018.

[32] I. Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: Anonymous distributed e-cash from bitcoin," in *Proceedings - IEEE Symposium on Security and Privacy*. IEEE, may 2013, pp. 397–411.

[33] E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in *Proceedings - IEEE Symposium on Security and Privacy*. IEEE, may 2014, pp. 459–474.

[34] C. Fromknecht and D. Velicanu, "A Decentralized Public Key Infrastructure with Identity Retention," *Cryptology ePrint Archive*, pp. 1–16, 2014. [Online]. Available: https://eprint.iacr.org/2014/803.pdf

[35] L. Axon, "Privacy-awareness in blockchain-based PKI," 2015.

[36] A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption." Springer, Berlin, Heidelberg, 2005, pp. 457–473.

[37] A. Shamir, "Identity-Based Cryptosystems and Signature Schemes," in *Advances in Cryptology*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1984, pp. 47–53.

[38] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing." Springer, Berlin, Heidelberg, 2001, pp. 213–229.

[39] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security - CCS '06*. New York, New York, USA: ACM Press, 2006, p. 89.

[40] W. Brent, "Ciphertext-Policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Proceediings of the 14th international conference on Practice and theory in public key cryptography conference on Public key cryptography*. Taormina, Italy: Springer, 2011, pp. 53–70.

[41] H. Zhu, L. Wang, H. Ahmad, and X. Niu, "Key-Policy Attribute-Based Encryption with Equality Test in Cloud Computing," *IEEE Access*, vol. 3536, no. c, pp. 1–1, 2017.

[42] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in *Proceedings of the 14th ACM conference on Computer and communications security - CCS '07*. New York, New York, USA: ACM Press, 2007, p. 456.

[43] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-Based Encryption with Partially Hidden Encryptor-Specified Access Structures," in *Applied Cryptography and Network Security*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 111–129.

[44] K. Emura, A. Miyaji, A. Nomura, K. Omote, and M. Soshi, "A Ciphertext-Policy Attribute-Based Encryption Scheme with Constant Ciphertext Length," in *Proceedings of the 5th International Conference on Information Security Practice and Experience*. Springer-Verlag, 2009, pp. 13–23.

[45] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," in *Proceeding of the 2007 IEEE Symposium on Security and Privacy*. Washington: IEEE Computer Society, 2007, pp. 321–334.

[46] L. Ibraimi, Q. Tang, P. Hartel, and W. Jonker, "Efficient and Provable Secure Ciphertext-Policy Attribute-Based Encryption Schemes," in *Proceedings of the 5th International Conference on Information Security Practice and Experience*. Springer-Verlag, 2009, pp. 1–12.