Blockchain-Based Dynamic Key Management for Heterogeneous Intelligent Transportation Systems

Article · August 2017

DOI: 10.1109/JIOT.2017.2740569

6 authors, including:

Ao Lei
University of Surrey
12 PUBLICATIONS **43** CITATIONS
SEE PROFILE

Philip Asuquo
University of Surrey
13 PUBLICATIONS **52** CITATIONS
SEE PROFILE

Z. Sun
University of Surrey
246 PUBLICATIONS **1,571** CITATIONS
SEE PROFILE

Blockchain-Based Dynamic Key Management for Heterogeneous Intelligent Transportation Systems

Ao Lei, Haitham Cruickshank, *Member, IEEE*, Yue Cao, *Member, IEEE*, Philip Asuquo, Chibueze P. Anyigor Ogah, and Zhili Sun, *Senior Member, IEEE*

***Abstract*—As modern vehicle and communication technologies advanced apace, people begin to believe that the Intelligent Transportation System (ITS) would be achievable in one decade. ITS introduces information technology to the transportation infrastructures and aims to improve road safety and traffic efficiency. However, security is still a main concern in vehicular communication systems (VCSs). This can be addressed through secured group broadcast. Therefore, secure key management schemes are considered as a critical technique for network security. In this paper, we propose a framework for providing secure key management within the heterogeneous network. The security managers (SMs) play a key role in the framework by capturing the vehicle departure information, encapsulating block to transport keys and then executing rekeying to vehicles within the same security domain. The first part of this framework is a novel network topology based on a decentralized blockchain structure. The blockchain concept is proposed to simplify the distributed key management in heterogeneous VCS domains. The second part of the framework uses the dynamic transaction collection period to further reduce the key transfer time during vehicles handover. Extensive simulations and analysis show the effectiveness and efficiency of the proposed framework, in which the blockchain structure performs better in term of key transfer time than the structure with a central manager, while the dynamic scheme allows SMs to flexibly fit various traffic levels.**

***Index Terms*—Blockchain, dynamic key management, handover, Intelligent Transportation System (ITS), vehicular communication systems (VCSs).**

I. Introduction

CYBER-PHYSICAL system (CPS) is considered as one of the most potential techniques to bring a better life to human beings. One of the most attractive CPS scenarios is the Intelligent Transportation Systems (ITSs). Vehicles and ITS infrastructures play the role of physical units, while the vehicular communication systems (VCSs) is the network platform of ITS. VCS supports not only message exchange among vehicles, but also between vehicles and infrastructures as well. Infrastructure access points (APs) in VCS are called road side units (RSUs) [1]. RSUs act as a base station in VCS and covers a dedicated section of the road. Traditional VCS is composed of multiple RSU cells and offers a platform among ITS for vehicles to exchange various kinds of messages, such as safety notification message. With the help of VCS, ITS can offer safer and efficient traffic management. Moreover, commercial applications, such as electric vehicle charging [2], image recognition for license plates, location-based service information, and dynamic scene to assist vehicle navigation [3], can be implemented on a dedicated platform. A recent report from U.S. Department of Transport shows that 82% of the accidents can be prevented by using ITS systems [4]. Even though significant developments have taken place over the past few years in the area of VCS, security issues, especially key management schemes are still an open topic for research [5], [6]. High mobility, large volume, frequent handoffs of vehicular nodes and heterogeneity networks pose different challenges compared to the traditional mobile networks.

VCS applications are classified into vehicle-to-vehicle and vehicle-to-infrastructure [7] and its security highly relies on the exchange of safety beacon messages. These beacon messages are usually referred to as cooperative awareness messages in Europe [8] or basic safety messages (BSMs) for U.S. [9], as they enable other vehicles to be aware of their surroundings. Vehicles located in the same RSU cell form a group and the current traffic situation is generated based on the summary of BSM broadcast from other group members [10]. The trustfulness and legality of BSM information are proved by encrypting safety messages with a preagreed group key (GK). For this reason, the problem of providing ITS security can be mapped into the problem of how to reliably distribute or update GKs among all the communicating participants. Several approaches were developed to improve the efficiency of managing keys for groups. Key tree approaches [11], [12] were developed to ease the problem. Furthermore, batch rekeying [13]–[15] was proposed to significantly improve efficiency compared to individual rekeying schemes. But these approaches are not suitable for VCS application as the number of mobile nodes (MNs) may be very large in VCS.

Aside from the aforementioned problem, it is critical to make sure the cryptographic materials can be timely delivered to the security manager (SM) in a new security domain. Moreover, GK has to be refreshed and redistributed (rekeying) securely whenever group member changes in order to achieve forward and backward secrecy [16]. This approach

Manuscript received November 30, 2016; revised May 27, 2017 and July 24, 2017; accepted August 11, 2017. Date of publication August 15, 2017; date of current version December 11, 2017. *(Corresponding author: Yue Cao.)*

A. Lei, H. Cruickshank, P. Asuquo, C. P. A. Ogah, and Z. Sun are with the Institute of Communication Systems, University of Surrey, Guildford GU2 7XH, U.K. (e-mail: a.lei@surrey.ac.uk; h.cruickshank@surrey.ac.uk; p.asuquo@surrey.ac.uk; c.anyigorogah@surrey.ac.uk; z.sun@surrey.ac.uk).

Y. Cao is with the Department of Computer and Information Sciences, Northumbria University, Newcastle-upon-Tyne NE1 8ST, U.K. (e-mail: yue.cao@northumbria.ac.uk).

Digital Object Identifier 10.1109/JIOT.2017.2740569

poses challenges of rekeying efficiency, especially in the heterogeneous network. Heterogeneity in wireless network refers to either the difference on the traffic volumes, or distinct network structures [17]. The heterogeneous networks structures normally stand for the networks managed under different topologies or central managers [18], [19]. Recently, heterogeneous VCSs are given more attention. The heterogeneity in terms of different central managers has become a real problem as VCS is considered as a worldwide system covering multiple countries. Specifically speaking, SM should timely deliver a vehicle's cryptographic materials to the neighbor SM when the car passes the cross-domain border.

With this in mind, blockchain [20] is considered as a feasible tool to achieve the goal. Blockchain is a synchronized and distributed ledger which stores a list of blocks. Blocks record user information and a receipt to link to the previous block. Central managers are removed from the blockchain structure and the public ledger is maintained by all the network participants instead. Messages are broadcasted into the network for nodes to authenticate. A new block is attached to the ledger if the messages pass the authentication process. With the help with this simplified structure, information propagation between security domains can be accelerated since the information is directly sent to the destination rather than passing the messages through central managers. Moreover, the distributed structure of blockchain network performs better robustness under the single point of failure.

In this paper, we propose a key management scheme for VCS scenario, including the key transfer between two heterogeneous networks and the dynamic key management scheme to decrease the key transfer time. A novel blockchain concept is introduced into the proposed scheme to simplify the key transfer handshake procedure in order to achieve better efficiency. In the blockchain-based scheme, we removed the third-party authorities (central managers) and the key transfer processes are verified and authenticated by the SM network. The record of these processes (mined blocks) is shared within the network for SMs to create public ledgers. Furthermore, the transaction collection period is able to dynamically change with respect to various traffic levels. The time consumption result of heterogeneous key management is compared with that in the traditional network structure to evaluate the performances of our blockchain-based scheme.

The remainder of this paper is organized as follows. Section II briefly introduces key management techniques. Model overview and details of our scheme are discussed in Section III. We describe our system model, including blockchain algorithms, key transfer between heterogeneous networks, and dynamic transaction collection periods. Scenario is set up for performance evaluation in Section IV. Section V concludes this paper and presents some future plans.

II. Related Work

In this section, we present the overview of the characteristics of any related schemes in this section, a brief literature review about CPS, bitcoin, blockchain applications, and VCS key management is introduced afterwards.

A. CPS

In CPS, components are classified into physical part and software part [21]. Physical components include infrastructures, network sensors, and computation devices. Software components contain programme, software operation systems and the Internet of Things (IoT) environment. CPS has various use cases, including ITS, smart grid, smart meters, smart medical systems, smart cities, etc. These use cases assist living, improve safety, and release traffic jam. However, challenges hide in the positive impact of CPS. Major challenges about CPS have been conducted in enhancing the security and privacy, as well as network efficiency [22], [23]. For instance, wireless sensor network is a well known CPS use case. It requires security scheme to maintain both efficient secret key distribution and low energy consumption [24]. A cutting-edge CPS scenario is described in paper [25]. The paper proposes a solution in vehicular fog-computing services (vehicular CPS). The fog-computing follows the distribution structure and distributes the heavy computation tasks to the infrastructures, instead of central manager. Paper [25] enables a smart resource management to optimize the communication-plus-computing energy efficiency in order to achieve the best QoS requirement. A more applicable fog-computing-based CPS system is discussed in [26]. This paper developed a framework to optimize TCP/IP virtualized data centers, the dynamic scheduler and the dynamic queue system are taken into consideration. The dynamic approach not only maximize the average workload admitted by the data center, but also minimize the resulting network-plus-computing average energy consumption. However, both the above schemes only cover the network efficiency issue, but not consider the security and privacy vulnerabilities.

B. Blockchain and Security Analysis

A lot of attention has been attracted to the blockchain concept since its parent production, bitcoin, was launched in late 2008 [20]. The core idea of blockchain is that it maintains a distributed, authenticated, and synchronized ledger of transactions. Without the administration from the central manager, network nodes denote their processing power to proofread transactions. The authenticated transactions are written into the public ledger in the form of blocks. Accountability function is benefited by using block look-up, which helps to timely revoke the cryptographic materials of malicious users. Another issue of blockchain approach is the use of transactions which conveys information among the distributed network and can hence send messages using peer-to-peer mode [27]. More importantly, network participants (miners) contribute their processing power to verify information correctness and integrity in blockchain network [27].

Two characteristics are always mentioned along with blockchain: 1) distributed and 2) decentralized. The distributed characteristic means that the network structure follows mesh or P2P topologies. Decentralization mainly refers to the management mode of blockchain network. However, the core principle in blockchain is decentralization. The centralized network depends on a network manager to prevent malicious behaviors.

As a result, centralized managers take too much communication and computation burden. Furthermore, the whole network suffers from disconnection if the central manager is under attack. Decentralization management networks, on the other hand, distribute the responsibility and control permissions between the user nodes. Security and privacy of the network are based on the proof-of-work [28]. Due to the nature of distributed computation, the network has better robustness to against network failure caused by nodes disconnection. Although the 51% attack [28] still plays a potential problem for blockchain applications, holding a majority of the total network's processing power is highly unlikely. It is uneconomic for individual attackers to employ a powerful system like ITS. Paper [29] precisely analyzed the security threats to blockchain system.

C. Blockchain Applications

Most of the contributions using blockchain are devoted to optimize decentralized currency models. Danezis and Meiklejohn [30] presented a new cryptocurrency. Basing on the concept of bitcoin, the new cryptocurrency improves scalability and flexibility of cryptocurrencies.

Despite the fact that there is no other work about using blockchain in VCS, blockchain architectures in some contributions are still valuable for reference. Paper [31] proposed to use blockchain to build a decentralized system to manage personal data. Authors design two types of un-financial transactions which are assumed to access data in blockchain and write data into the ledger, respectively. The access control of personal data is monitored by blockchain. An access transaction is sent when a user tries to access the database for storage or retrieval of data. After the access transaction is approved, the user needs to describe their requirements in data transaction in order to finish the communication between blockchain. However, Zyskind *et al.* [31] did not consider overhead and efficiency problems.

Since IoT aims to seamlessly fit into CPS, for maximizing adoption by users and infrastructures. It is critical to compress the overhead and efficiently manage the increasing number of node identity materials [32]. Aitzhan and Svetinovic [33] focused on a cutting-edge secure transaction exchange system using blockchain for decentralized energy trading in another CPS, smart grids. They address the scalability, security and privacy problems of the centralized system. The security issues are analyzed with reference to the processing time to different cryptographic schemes. However, the analysis is not based on the network performance. Contribution in [33] involves another blockchain-CPS research focusing on smart medical systems. Wireless nodes and sensors play the role of blockchain miners. Miners in this approach can get access to anonymized medical data as rewards, in return for their mining work to maintain the blockchain. Both patients and the health care staffs are given accessible and credible electronic medical records.

D. Our Contributions

To the best of our knowledge, our previous work [34] is the first time the technology has been used in VCS applications.

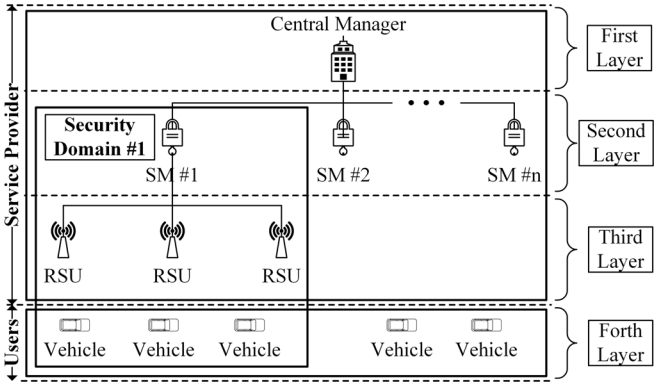


Fig. 1. Traditional network structure [35].

In [34], the SM network was used to transfer and verify vehicle keys in the across border requests, rather than forwarding them to the third party authorities. The time consumption result of heterogeneous key management is compared with that in the traditional network structure to prove that the blockchain concept helps to shorten the key transfer time. However, it did not take the information proofreading time into consideration. Moreover, it was apparent that our previous result did not consider dynamic elements in VCS. For example, the transaction collection periods should keep the same pace with various vehicle rekeying periods.

III. Proposed Framework

A. System Model

We focus exclusively on a system of ITS infrastructures each equipped with a device embedded with wireless communication module based on the IEEE 802.11p standard. Meanwhile, vehicles are required to have a built-in on board unit (OBU) to support the IEEE 802.11p standard. Vehicles travel on a road and periodically transmit safety messages using the transmitter in the OBU, which are collected by infrastructures that are built along the road at regular intervals. Safety message includes movement information, such as speed, orientation, position, and vehicle size. The infrastructures relay messages between vehicles and SMs which are placed on the upper level of VCS. Each SM has their own logical coverage area which is called security domain.

1) Network Hierarchy: VCS networks normally have four layers. Three layers on the side of service providers, while the user side occupies a single layer [35]. As shown in Fig. 1, layers on the service providers' side, namely, RSUs, SMs, and central managers. RSUs act as IEEE 802.11p AP which offer an interface to route messages from vehicles to upper-level managers. RSUs are built along the road at regular intervals in order to provide maximum network coverage. SMs are placed at the second layer which manages cryptography materials of different security domains. It is proposed to install SMs in a geographically sparse manner, one for each security domain. Central managers rule the network on the first(top) layer, they are also known as certificate authorities (CAs). Vehicles' permanent identities, certificates, pseudonyms are calculated and authenticated at CA to issue legal identities using in VCS.

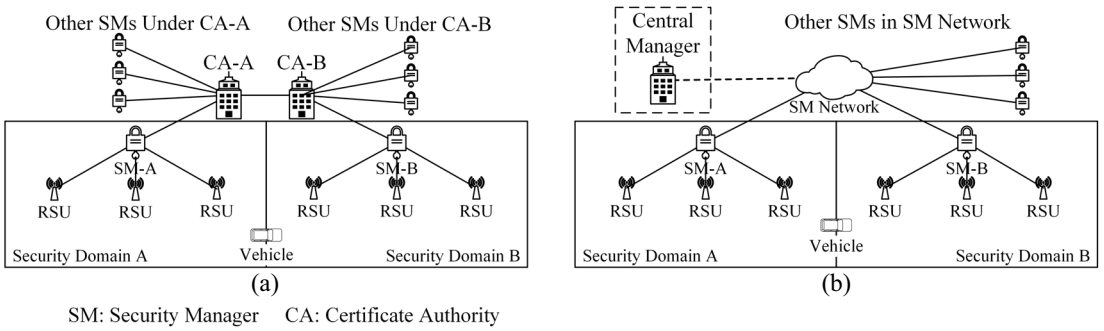


Fig. 2. Network structures. (a) Traditional structure with the third-party authorities. (b) Blockchain structure without the third-party authorities.

2) Traditional Structure: Traditional structure strictly follows the aforementioned hierarchy. As shown in Fig. 2(a), security domain A is an area which is managed by SM-A. CAs take the role of central managers at the top level. Several SMs are managed by CA. This traditional network structure employs CA or trusted third party authority at the top of the network to manage cryptography materials, this however, makes it an inefficient key exchange, and will require several handshakes if a car passes from one security domain to another. When a vehicle attempts to join a new geographic region in which infrastructures are managed by a new CA, the previous SM-A (previous SM) picks up this border crossing activity from the beacon messages that are sent by the vehicle. Then it generates a border crossing request along with useful information related to the vehicle and forwards all these materials to the CA-A (previous CA). The request will be forwarded to the CA-B (new CA) if it has passed the verification steps. CA-B will inform SM-B (new SM) about the border crossing activity with necessary cryptography materials, before it checks the correctness of the request. Rekeying procedures will be triggered in the new area after new SM has received such cryptography materials.

3) Blockchain-Based Structure: The above procedures in traditional network delay the key transfer between two security domains. Different to the traditional structure, the functions of the central manager on the top level are merged into SM in blockchain-based structure. In this case, SM takes over the role of the network manager. As presents in Fig. 2(b), central manager is placed in an isolated environment, acting as a facility to store and generate vehicle cryptographic materials. Cryptographic materials, such as vehicle identities, pseudonyms, and pseudonym certificates, are supposed to be kept in a dedicated facility to cope with the privacy and security purposes [36]. Thus, central managers are accessed under the following three situations.

1) *Initial Registration:* New vehicles need to apply for the initial registration when they leave the manufacturer and first participate in a new security domain.
2) *Change the Identity-Related Information:* Vehicles must to periodically change their pseudonym set, as well as all the cryptographic materials related to this pseudonym. Thus, they need to contact the central manager to generate a new set of cryptographic identity for them.
3) *Adversary Revocation:* In the blockchain-based structure, malicious behaviors are recognized by using blockchain look-up. Identity (including pseudonyms) of the adversary is publicized once the malicious behaviors have been confirmed.

Similar to the bitcoin network, the function of blockchain enables nodes to share information without the need for a central party to secure this ledger. SM is connected with an SM group that may link with SMs on other domains and certification entities with a domain. Similar to the bitcoin applications, the information in safety messages are encapsulated into transactions if they indicate an SM-border-crossing action. Transactions are shared with neighbor SMs to transport keys. Aside from this, the SMs take the role of miners which forms transaction within a period of time into a block. As a reward, miners are allowed to get their block authenticated by the SM network. Our proposal is to transport keys by mining blocks so that a blockchain can be maintained for heterogeneous key management purpose, at least within a local SM domain. As a result, the keys of new joining members is delivered by retrieving the information from the block.

B. Heterogeneous Key Management

We introduce the blockchain concept in this section, which aims to simplify the distributed key management in large heterogeneous security domains. Blockchain helps to achieve a lightweight and scalable key transfer scheme. The conventional multiserver handover steps are illustrated before the blockchain idea is demonstrated.

1) Key Transmission Handshake: The cross domains handshake steps are discussed in [37] and [38] to guarantee efficiency and security. The handover steps were studied under mobile multiserver networks. The scenario involves five entities: 1) MN; 2) home agent (HA); 3) foreign agent (FA); 4) authentication server (AS); and 5) RSUs. The schemes assume that all the MU had registered with AS to obtain the cryptographic materials. The MU sends a request message to sign into FA upon joining the foreign domain. The request is received by RSU, verified and forwarded to FA afterwards. The handover is triggered by FA. Before building connection with MU, FA must authenticate the unfamiliar node through HA. Two handshakes will be established between FA and HA before MU is proved as a legal user of HA. Finally, FA will issue a new set of cryptographic materials to MU.

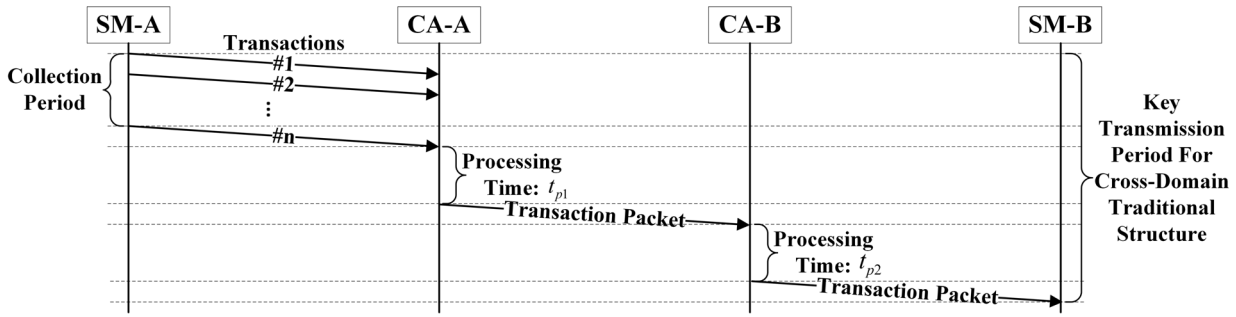


Fig. 3. Key transfer handshake procedures in cross-domain traditional structure.

The typical handover steps in [37] and [38] are designed basing on the unpredictable node movement trajectories. However, vehicle movement trajectories are easily predicted due to the fact that the current RSU knows all the driving trends of vehicles under its coverage area. For this reason, the conventional handover steps can be triggered by the HA instead of FA. In the VCS traditional structure, we assume the SMs take the job of HA and FA and RSUs only for improving coverage area. Additionally, malicious behaviors in VCS can easily endanger human life, it requires a top level security to deliver trusted service. The requirement is fulfilled by equipping server who supervises user data. Thus, the handshakes between SMs are checked by CA in a mandatory manner.

The cross domains handshake process in the traditional network is shown in Fig. 3. The network set a collection period based on the traffic level. SM-A (previous SM) picks up all the border crossing activities from beacon messages within this transaction period. These border crossing activities are formed into individual transactions. SM-A sends these transactions one by one to CA-A (previous CA) to proof. To ensure security, proof work should verify the signature to check authentication and message integrity. The ciphertext will be decrypted using CA-A's private key and re-encrypted using CA-B's public key. That is because the original ciphertext is secured using CA-A's public key and CA-B does not have the corresponding key to decrypt. During the proofreading, the proved transactions are translated into a new version which is readable by CA-B (next CA). CA-B repeats the proof steps after receiving the transaction packet and convert them into SM-B readable version. Finally, all the cross-border requests arrive at SM-B, packing in transaction packet. A handshake message flow is shown below with details. Where En$\{*\}$ stands for the encryption activities using elliptic curve integrated encryption scheme (ECIES) scheme [39], Sig$\{*\}$ is the signing conducts using elliptic curve digital signature algorithm (ECDSA) scheme [40]. PK_* and SK_* are elliptic curve-based public and private key pairs, respectively.

1) SM-A sends transactions to CA-A

$$\mathrm{En}\{\mathrm{info}\}_{\mathrm{PK_{CA\text{-}A}}} + \mathrm{dest_{SM}} + \mathrm{Sig}\{\mathrm{Cipher} + \mathrm{dest_{SM}}\}_{\mathrm{SK_{SM\text{-}A}}}.$$

2) CA-A forwards the transaction packet to CA-B

$$\mathrm{En}\{\mathrm{info}\}_{\mathrm{PK_{CA\text{-}B}}} + \mathrm{dest_{SM}} + \mathrm{Sig}\{\mathrm{Cipher} + \mathrm{dest_{SM}}\}_{\mathrm{SK_{CA\text{-}A}}}.$$

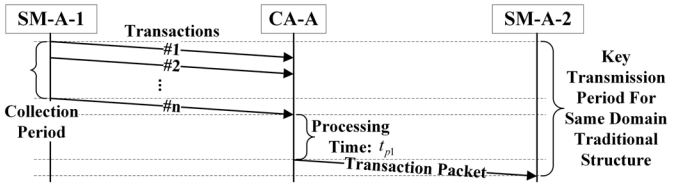


Fig. 4. Key transfer handshake procedures in same-domain traditional structure.

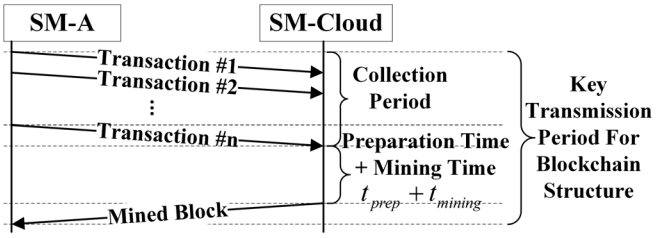


Fig. 5. Key transfer handshake procedures in blockchain structure.

3) CA-B forwards the transaction packet to SM-B

$$\mathrm{En}\{\mathrm{info}\}_{\mathrm{PK_{SM\text{-}B}}} + \mathrm{dest_{SM}} + \mathrm{Sig}\{\mathrm{Cipher} + \mathrm{dest_{SM}}\}_{\mathrm{SK_{CA\text{-}B}}}.$$

The handshake steps are reduced if SMs are located in the same security domain. SM-A-1 forwards transactions to CA-A to proof. Under this structure, both SMs are under managed by the same CA. Therefore, there is no need to translate transactions into another version which is dedicated to other CAs. Similar to the cross domain version, SM-A-2 receives transactions from transaction packet in the end. The aforementioned steps are presented in Fig. 4.

1) SM-A-1 sends transactions to CA-A

$$\mathrm{En}\{\mathrm{info}\}_{\mathrm{PK_{CA\text{-}A}}} + \mathrm{dest_{SM}} + \mathrm{Sig}\{\mathrm{Cipher} + \mathrm{dest_{SM}}\}_{\mathrm{SK_{SM\text{-}A}}}.$$

2) CA-A forwards the transaction packet to SM-A-2

$$\mathrm{En}\{\mathrm{info}\}_{\mathrm{PK_{SM\text{-}A\text{-}2}}} + \mathrm{dest_{SM}} + \mathrm{Sig}\{\mathrm{Cipher} + \mathrm{dest_{SM}}\}_{\mathrm{SK_{CA\text{-}A}}}.$$

The key transportation handshake could thus be simplified by using blockchain mining method, meaning the messages will be verified by SM network but not third party authorities. The blockchain structure removes CAs and nodes supervising each other by mining the blocks and broadcasting the results. A simplified handshake graph is shown in Fig. 5. Collection

TABLE I
FORMAT OF TRANSACTION

Transaction Header
Hashed result of the transaction
Number of this transaction in block
Current security domain number *SM-this*
Destination security domain number *SM-dest*
Vehicle identity materials including the encrypted vehicle pseudonym and certificate
Signature of this transaction to ensure integrity and authentication $Sig\{Cipher + dest_{SM}\}_{SK_{SM\text{-}this}}$
Payload: (Encrypted Transaction Information) $Cipher = En\{info\}_{PK_{SM\text{-}dest}}$

TABLE II
FORMAT OF BLOCK

Block Header	
Field	Description
Version	Block Version Number
Previous Block Hash	Hash of the previous block in the chain
Merkle Tree Root	Hash of the merkle tree root $Root_M$
Timestamp	Creation time of this block
Targeted Difficulty	The Proof-Of-Work difficulty target
Nonce	A counter for the Proof-Of-Work
Block Payload (Transactions)	
Transaction No.1 $\cdots$ Transaction No.n	

period allows several transactions to be broadcasted into SM network and picked up by SMs in the network. Signatures in transactions are processed to verify if the information in transactions is trustworthy. Ciphertext in transactions is kept from decryption until they reach the destination SM since the ciphertext is encrypted using the public key of the destination SM. According to the nature of blockchain mining and transactions are inserted into the block in random order. Last but not least, the above block will be mined using mining algorithm and the mined block will be broadcasted back to the network. The above procedures are presented as follows.

1) SM-A sends transactions to SM-Cloud

$$\mathrm{En\{info\}_{PK_{SM\text{-}dest}}} + \mathrm{dest_{SM}} + \mathrm{Sig\{Cipher + dest_{SM}\}_{SK_{SM\text{-}A}}}.$$

2) SM-Cloud returns the mined block to SM-A.

2) Transaction Format: Transactions are designed to encapsulate key transfer materials from the source SM to destination SM. Six fields are contained in the transaction header of our model (Table I) [34]. The transaction number shows the position of this transaction in the transaction packet. Current and destination SM number are equivalent to bitcoin input and output, respectively, [20]. The identity materials including the current vehicle pseudonym and certificate, are encrypted using the public key of the destination SM. The signature occupies the last position of the transaction to maintain the authentication, integrity, and nonrepudiation of key transfer information.

Here, info is the identity materials in the transaction. Privacy-related information is encrypted into ciphertext $\mathrm{En\{info\}_{PK\text{-}dest}}$ using destination SM's public key $\mathrm{PK_{dest}}$. Signature is computed using both ciphertext and the number of destination SM, signed using source SM's private key $\mathrm{SK_{this}}$.

To keep the confidentiality of the information in transactions, identity materials are encrypted using destination SM's public key. As a result, the information stays unreadable to the SM network expect the destination SM. Encrypted privacy related information combined with digitally signed transaction contents ensure that an adversary cannot act as a normal node, or amend and eavesdrop cross-domain requests, as that would require the adversary to forge a signature. Simultaneously, other SMs are able to exam if this transaction is legitimate or not. Similarly, a malicious user cannot read anything from the encrypted message, as only the destination SM has the key to decrypt the message.

3) Block Format: The block header is constructed by six fields (Table II), similar to the bitcoin block [34]. The second field links the block to its parent block. This field helps blocks linking to each and creating a chain structure. All the transactions in the block are merged into the Merkle tree root [41]. Merkle tree root assures the integrity of transactions as the alteration on transactions causes a totally different value of Merkle root value. Time tampering is prevented by checking the timestamp field. A target mining solution is a 256-bit number with number of zeros n_{zeros} at the start of the hash result of the block header [33]. The number n_{zeros} is the targeted difficulty. SM collects all transactions within a certain period (transaction collection period) of time and inserts these transactions in arbitrary order into a block. In this way, blocks are able to aggregate multiple cross-border requests.

The payload of a block is composed of transactions that SMs collect within the transaction collection period, denoted by t_{CP}. These transactions are packed into the same block. The theoretical number of transactions is decided by t_{CP} and the number of passing vehicles in each hour (n_H). An expression of n_T is shown in the following equation:

$$\text{number of transactions} = \frac{n_H}{3600\ \mathrm{s/h}} \times t_{\mathrm{CP}}. \tag{1}$$

4) Mining and Proof Algorithm: In blockchain, proof-of-work is a digital receipt which is hard to calculate but easy for others to verify [20]. A one-way cryptographic hash function, double SHA256, dhash(), is used to calculate the proof-of-work. Where the double SHA256 is calculated as follows:

$$\text{hashed result} = \text{dhash(input)} = \text{SHA256(SHA256(input))}.$$

The hash of the block header is calculated by SMs. This candidate block header is hashed repeatedly using different nonce value until the resulting hash value starts with the numbers of zeros (matches the difficulty requirement). As we mentioned in the block format section, the transactions are placed in block payload in random order. The reason that we using this approach is because it forces different SMs having different mining time. The random order provides an extent of stochasticity, leading to distinct header hash results. To be more precisely, different header hash results make SMs mining same sets of the transaction in different time lengths. Therefore, block conflicts are prevented in advance. Algorithm 1 shows a summarized pseudocode of mining procedure. Algorithm 2 gives a detailed overview of necessary procedures for proofing a mined block.

TABLE III
TIME ELEMENTS OF PROCESSING PROCEDURES

Parent Field	Description of Parent Field	Child Field	Description of Child Field
t_{prep}	The time cost to prepare block which will be mined later	t_{rand}	Calculation time to generate random transaction sequence
		t_{fill}	Time cost to insert transactions into the block message
		t_{merkle}	Calculation time to get Merkle Tree Root
		t_{header}	Processing time to prepare block header
$t_{transfer}$	Transmission time cost in SM network including CSMA back-off time	t_{BO}	Average CSMA back-off time
		t_P	Propagation time in network cable
$t_{processing}$	Processing time for message **Encryption**, **Decryption**, **Signing** and **Verification**	t_E	Processing time to encrypt plain text (ECIES)
		t_D	Processing time to decrypt cipher text (ECIES)
		t_S	Processing time to sign messages (ECDSA)
		t_V	Processing time to verify signature (ECDSA)

Algorithm 1 Calculate Nonce (Proof-of-Work)

Input: : Information to create Candidate Block Header **H**: Block Version V_B, Previous Block Hash $Hash_{prev}$, Timestamp t_{now}, difficulty number d and transactions $Trans = [T_1, T_2...T_n]$
Output: : Nonce value *nonce*

Initialise bool variable $gotAns$ = **FALSE**;
while (**NOR** *gotAns*) **do**
  The transaction order $a_{rand} = randPerm(n)$ by permuting integers within range [1, n];
  Calculate Merkle tree root $Root_M$ basing on a_{rand};
  Create the hashed block header H_{temp} Where $H_{temp} = V_B||Hash_{prev}||Root_M||t_{now}||d$;
  Initialise tries number $nonce = 0$; Hash output *result*;
  while (**NOR** *gotAns* **& NOT** got Proof-Of-Work from network) **do**
    $result = dhash(H_{temp}||nonce)$;
    nonce ++ ;
    if (*result* has at least d padding zeros in front **& NOT** got Proof-Of-Work from network) **then**
      Write (nonce - 1) into nonce field;
      gotAns = ***TRUE***;
      return (nonce - 1);
    else if (receive Proof-Of-Work from Network) **then**
      gotAns = ***TRUE***;
      return NULL;
    end if
  end while
end while
End Algorithm

Algorithm 2 Proof the Block

Input: : Mined Block Header $\mathbf{H}_{mined}$; Block payload $B_{payload}$
Output: : Bool variable *isCorrect*

Extract nonce value: $nonce = getNonce(\ \mathbf{H}_{mined}\ )$;
Calculate Merkle Tree Root $Root_M$ basing on the transactions in $B_{payload}$;
Create header: $H_{verify} = V_B||Hash_{prev}||Root_M||t_{now}||d$;
The string to verify: $Input_{verify} = H_{verify}||nonce$;
Calculate the hashed value of the string: $result = dhash(\ Input_{verify}\ )$;
if (*result* has at least d padding zeros in front) **then**
  isCorrect = **TRUE**;
else
  isCorrect = **FALSE**;
end if
return isCorrect;
End Algorithm

5) Time Composition: Table III shows all the time elements that composes the key transfer time. For traditional structure, all the time variables in $t_{\text{processing}}$ are taken into account, while t_V is the only one to be considered in blockchain structure. Message transfer time t_{transfer} including the information propagation time in cable, as well as the random back-off time in the CSMA protocol. The variable t_{prep} is dedicated to blockchain applications, containing time cost variables to create a new block.

As describes above, processing time for three situations are summarized in (2)–(4). Where n_T is the average number of transactions among a single collection period. Variable t_{TC}, t_{TS}, and t_B are processing time of key transfer procedures in cross-domain traditional structure, same-domain traditional structure, and blockchain structure, respectively,

$$t_{\text{TC}} = n_T \times (t_V + t_D + t_E + t_S) \times 2 + (t_{\text{BO}} + t_P) \times 3 \quad (2)$$

$$t_{\text{TS}} = n_T \times (t_V + t_D + t_E + t_S) + (t_{\text{BO}} + t_P) \times 2. \quad (3)$$

Equations (2) and (3) describe the time components in the traditional structure. Due to the fact that CAs in the traditional structure must verify and translate transactions to the neighbor CAs or SMs. Both situations take all the elements in $t_{\text{processing}}$ into calculation. For the cross-domain scenario, the above processes are designed to be implemented twice

$$t_B = n_T \times t_V + (t_{\text{BO}} + t_P) \times 2 + t_{\text{prep}} + t_M. \quad (4)$$

Equation (4) expresses that only signature verification is required in transaction checking. However, mining time t_M and block preparation steps are attached into overall processing time in order to extend the blockchain.

6) Dynamic Key Management: Our dynamic key management is achieved by using dynamic transaction collection periods. To decrease the side effect of variables, the method of control variable is employed in our scheme. We use one second as the standard metric to measure the performances of various collection periods. Thus, $n_{T-\text{All}}$ is a sum up number of transactions in all the roads. t_{B-1} is the average processing time in one second under various collection periods. Basing on (1) and (4), we can derive the number of transactions on n_R roads and t_{B-1} as follows:

$$n_{T-\text{All}} = \frac{\text{traffic amount}}{3600\ \text{s/h}} \times t_{\text{CP}} \times n_R \quad (5)$$

$$t_{B\text{-}1} = \left[n_{T-\text{All}} \times t_V + (t_{\text{BO}} + t_P) \times 2 + t_{\text{prep}} + t_M\right] \div t_{\text{CP}}. \quad (6)$$

Estimated key transfer time is calculated using various collection periods as inputs. The optimized transaction collection time is selected according to the minimum key transfer time

$$\underset{t_{\text{CP}}}{\text{argmin}}\ t_{B\text{-}1}\ \text{ subject to: }\ t_{\text{CP}} \in \left[t_{\text{CP}}^{\mathbf{1}},\ t_{\text{CP}}^{\mathbf{n}}\right].$$

Algorithm 3 Optimize the Transaction Collection Period

Input: : Traffic amount on each road n_H, n optional transaction collection periods ($t_{CP}^{\mathbf{1}} \cdots t_{CP}^{\mathbf{n}}$)
Output: : Optimised transaction collection period $t_{CP}^{\mathbf{m}}$

1: Initialise a data sink $\mathbf{t}_{B\text{-}1} = [t_{CP}^{\mathbf{1}} \cdots t_{CP}^{\mathbf{n}}]$
2: **for** $(i = 1; i \leqslant n; i{+}{+})$ **do**
3: Call **Equation**(6), calculate $t_{B\text{-}1}^{\mathbf{i}}$ when $t_{CP} = t_{CP}^{\mathbf{i}}$ and traffic amount on each road is equal to n_H;
4: $\mathbf{t}_{B\text{-}1}[i] \leftarrow t_{B\text{-}1}^{\mathbf{i}}$, record $t_{B\text{-}1}^{\mathbf{i}}$ into the result sink;
5: **end for**
6: $t_{CP}^{\mathbf{m}} = \min(\mathbf{t}_{B\text{-}1})$, Find the minimum key transfer timeïijŻ
7: **return** $t_{CP}^{\mathbf{m}}$;
8: **End Algorithm**

To sum up, a transaction collection period optimization algorithm is demonstrated using pseudo-algorithm in Algorithm 3.

IV. Performance Evaluation

The performance evaluation of blockchain-based key management scheme was carried out using simulations. Performance evaluation is broken into three parts. The first part studies the processing time components, namely encryption, decryption, signing, verification, block mining, and block preparation. The comparison of processing time results between the blockchain structure and the traditional structure is demonstrated in the second part. The last part further studies the processing time in blockchain network against different transaction collection periods. This section starts with the simulation assumptions.

A. Simulation Assumptions

Our result is generated using OMNeT++ 4.5 [42], [43] with the dedicated network simulation (Veins) packet [43]. ECIES [39] with elliptic curve secp160r1 in Crypto++ [44] is selected not only for cryptographic scheme ECIES, but ECDSA as well. Cipher block has a length of 75 bytes which is because ECIES provides much better security level. 20 bytes are used to store the cross-border information in transactions. The difficulty of each block is set to 3 to maintain efficiency and security. This security level is enough to secure the network which is due to the fact that SMs are trusted entities. We simulated that blocks are mined by our laptop with Intel Core i5 and 8 GB RAM and display card GeForce 920 M. This device can complete 250 K hash calculations per second. The performances of first two parts of simulations focus on the processing time in terms of transactions. The results depend only on the overall number of transactions. Thus, the simulation setup is compromised the following steps.

1) At end of each t_{CP}, a certain number of transactions flooding into the SM network. The movement of vehicles are not considered in these two parts.
2) Each SM records the processing time results of cryptography schemes and block preparation. The results are records by averaging the results from SMs.
3) Transactions ranging from 0 to 200 is set for test cryptographic schemes to get a zoom-in view of results.

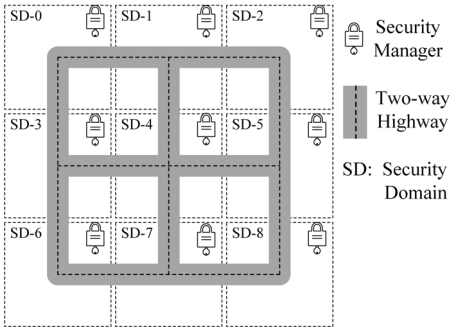


Fig. 6. Assumed topology for blockchain network.

Up to 1000 transactions are introduced in block preparation simulations so that the exponential growth of results can be demonstrated. Maximum 2000 transactions are simulated to compare the key transmission time which aims to test the time value differences between blockchain and tradition structures.

The third simulation aims to test key transmission time under different traffic levels and transaction collection periods. Here, we assume that the system calculates the overall number of cross-border activities at end of the collection periods. The vehicle cross-border activities follow the exponential distribution. The cross-border events occur rate follows the quantile function of exponential distribution [45]:

$$t_i = -\frac{\ln(1 - P_i)}{\lambda} \qquad \lambda = \frac{1}{\mu}$$

where i is the event number, λ is the rate of expected events, t_i is the expected events occurrence time, P_i is the probability following the normal distribution and μ is the mean value of the exponential distribution. The upper and the lower amount of vehicle traffic are considered under a saturated traffic condition and off-peak traffic of Beijing, which is considered as one of the most crowded cities in the world. The off-peak time has 3000 vehicles per hour, while the saturated traffic is set to have 15 000 vehicles passing a road in an hour, aiming to examine our scheme under the worst case as well as the heaviest burden of VCS. The topology of scenario in the third part results are assumed in Beijing. There are eight urban districts, therefore we assume a 3×3 topology. As shown in Fig. 6, each urban district is managed by one SM. Security domains are connected to each other via two-way highways. Here, we assume each common edge has five two-way highways to connect to the neighbor security domain. Thus, there are overall 120 highways basing on this topology. For each SM, t_{CP} is ranged from 0.5 s to 1 s in order to test the performance regarding different transaction collection length.

B. Processing Time of Cryptographic Schemes

We first study the processing time cost for cryptographic schemes. It aims to obtain the data of each elements in Table III and further complete the result of (2)–(4).

Since the key transfer time needs to consider the computation time of cryptographic schemes. Therefore, we simulated the time cost for different schemes. Fig. 7 shows the performance of different cryptographic schemes, which are used in

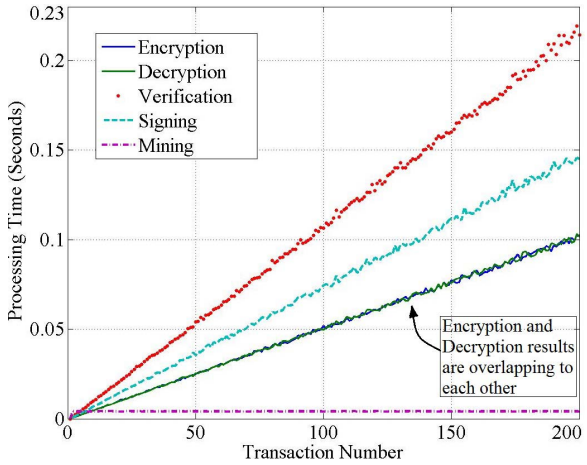


Fig. 7. Computation time of cryptographic schemes over transaction number.

TABLE IV
AVERAGE CRYPTOGRAPHY PROCESSING TIME

Cryptography Scheme	Processing Time (Milliseconds)
ECIES Encryption	0.51027
ECIES Decryption	0.73996
ECDSA Signing	0.51011
ECDSA Verifing	1.10171
Block Mining	4.11046

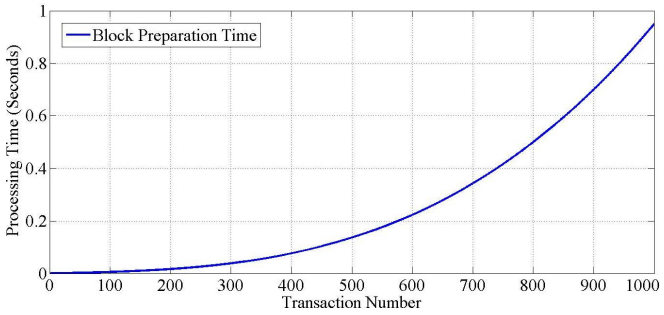


Fig. 8. Block preparation time with respect to the transaction number.

key transfer procedures. Except for the mining time cost, the processing time increases linearly with the growth of transaction number. The mining algorithm always mines a single header due to the fact that block header is able to contain multiple transactions. The mining processing time is an average value of multiple simulations. the practice value is likely less than this average value due to the network only accept the fastest mined block. The encryption and decryption schemes cost similar processing time. Signature verification costs the longest computation time among schemes. According to (2)–(4), signature verification plays a key component in key transfer time. Table IV records the average processing time for each cryptographic schemes.

Fig. 8 plots the block preparation time in terms of transaction number. The preparation time increases exponentially with the growth of transaction number. The processing time slowly increases before 300 transactions. Processing time over 0.1 s when transaction bigger than 400. Finally, preparation time reaches 0.95 s when there are 1000 transactions. The nonlinear curve is caused by exponentially increasing of t_{rand}, while

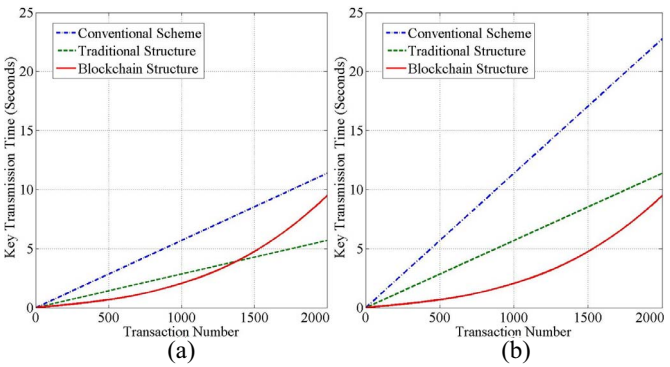


Fig. 9. Processing time comparison between structures and schemes. (a) Time cost values when hand over within same security domain. (b) Time cost values when hand over across different security domains.

rest of the preparation time components increase linearly in proportion to the transaction number.

C. Comparison of Blockchain and Traditional Structures

To evaluate the performance of the novel blockchain and traditional structures, we compare them with the conventional handover schemes in [37] and [38]. We conduct experiments under two situations.

1) Key transfer between two security domains which are within the same security division of the district.
2) Key transfer between two security domains which separate in different security divisions of the district. Here, the divisions mean geographical districts monitored by different central managers.

Fig. 9 depicts the key transfer performances of blockchain scheme with respect to varying the number of transactions. The results of schemes in [37] and [38] are used as benchmark of the simulation which aims to show the performance improvement by using our scheme. Comparison of situation 1) is shown in Fig. 9(a). All the results have zero processing time when border across actions does not appear in the network. It takes approximately 0.8 s to finish transfer 500 transactions, while nearly double the time is cost to handle the same amount of transactions in the traditional structure. The conventional scheme costs more than triple the key handover time of blockchain structure. However, two curves have an intersection at around 1500 transactions due to the exponential increase of blockchain key transfer time. Although our scheme cost more processing time due to the growing number of transactions, our schemes provide better scalability against the traditional structure when transaction number less than 1500. Additionally, our blockchain-based scheme saves nearly half of the processing time at transaction number equalling to 1500 and the time results always below the benchmark when transactions no less than 2000. Similar contradistinction is demonstrated in Fig. 9(b) to show the result of situation 2). CA translates messages from one security domain to another in the conventional scheme. For the traditional structure in this situation, two CAs need to communicate with each other in order to finish key transfer. Thus, extra handshakes between CAs cause tedious key transfer time in the traditional structure. Handover

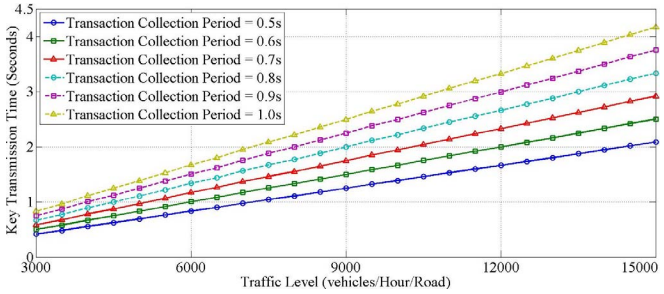


Fig. 10. Average transaction number under different traffic levels.

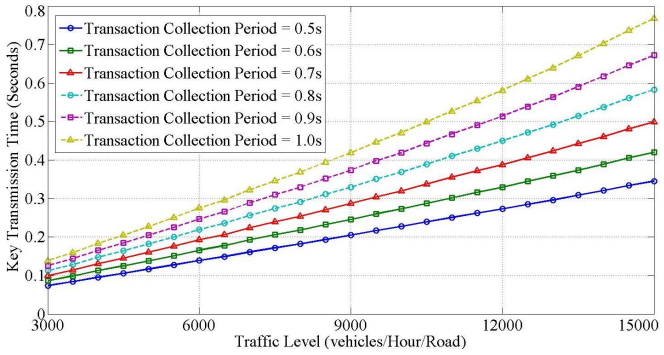


Fig. 11. Key transfer time under transaction collection periods.

time cost of the conventional scheme and traditional structure exceed 10 s when transaction number more than 800 and 1750, respectively. The blockchain scheme costs much less time. To summarize, blockchain structure has better scalability performance against the traditional structure in situation 2) due to less processing time cost.

D. Blockchain Performance Evaluation

The transaction collection period provides a window to allow SMs to pick received transactions. Therefore, different period lengths decide the amount of transactions flooding into SM network. Double-direction highways are considered based on the assumptions in Fig. 6, leading to two traffic flows on each highway. Therefore, we take a single traffic flow as a standard metric unit and simulate the average transactions in a single traffic flow. Fig. 10 plots the average transactions as a function of traffic levels and transaction collection periods. We observe that the transaction number is directly proportional to the traffic level. Moreover, the longer t_{CP}, the more transactions are caught by SMs. The average number of transactions per t_{CP} per traffic flow is calculated as follows:

$$\lambda = n_{T/\mathrm{CP}} = \frac{n_H}{3600} \times t_{\mathrm{CP}}$$

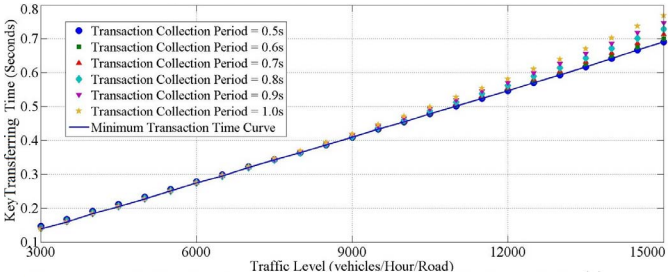
where $n_{T/\mathrm{CP}}$ is the average number of cross-border actions (transactions) within each t_{CP} and n_H is the average number of vehicles (traffic level) passing a road in each hour. A parameter n_R is multiplied by the $n_{T/\mathrm{CP}}$ to get the average transaction number in all the roads, here n_R is the amount of roads that are taken into calculation.

Fig. 11 illustrates the key transfer performances under various collection periods. It can be seen that for each

	Traffic Level											
	4500	*5000*	*5500*	*6000*	*6500*	*7000*	*7500*	*8000*	*8500*	*9000*	*9500*	*10000*
t_{CP} = 0.5s	0.211	0.233	0.255	0.279	0.298	0.323	0.344	0.365	0.387	0.409	0.433	0.455
t_{CP} = 0.6s	0.209	0.231	0.253	0.276	0.296	0.321	0.343	0.364	0.387	0.409	0.434	0.455
t_{CP} = 0.7s	0.207	0.229	0.252	0.275	0.295	0.320	0.342	0.364	0.387	0.410	0.435	0.457
t_{CP} = 0.8s	0.206	0.228	0.251	0.275	0.295	0.320	0.343	0.365	0.389	0.412	0.438	0.461
t_{CP} = 0.9s	0.205	0.228	0.251	0.275	0.296	0.321	0.344	0.366	0.391	0.415	0.442	0.466
t_{CP} = 1.0s	0.205	0.227	0.251	0.275	0.296	0.322	0.346	0.369	0.394	0.419	0.447	0.471

Minimum Processing Time

(a)



(b)

Fig. 12. Key transfer time results measured in one second. (a) Key transferring time from traffic level of 4500 to 10 000 vehicles/hour/road. (b) Key transferring time between fixed and collection period schemes.

value of collection period, there exists a marked rise trends when collection period longer than 0.8 s. The results indicate that longer collection period lets SMs to accept much more transactions, leading to heavy processing burden and tedious computation time. According to the results in the previous figure, for instance, average 4.2 transactions are captured using 1 s collection period when traffic level is equal to 15 000 vehicles/h/road, while 3.3 transactions are captured under traffic level of 12 000 vehicles/h/road. This causes $120 \times (4.2 - 3.3) = 108$ transactions' difference, resulting in huge difference of key transfer time. In addition, the result of 0.5 to 0.7 s increases steadily when other results increase exponentially.

E. Dynamic Transaction Collection Period

To measure the effect more accurately, we use 1 s as a standard metric to make sure that every transaction collection periods have equal running time. In order to confirm the effectiveness of the dynamic transaction collection period, we have carried out a simulation experiment to investigate the average processing time of key transfer in 1 s.

The evaluation is done using simulation. The modeling and parameter settings are discussed in the assumption section. The running time of simulation is set to be 1 h, multiple key transfer procedures under various collection periods are recorded and divided by 3600 s. Part of the results of above description are shown in Fig. 12(a). Along with the growth of traffic level, the minimum time results occur under different t_{CP} values. Longer transaction collection period provides shorter key transfer time under mild traffic conditions. However, rapid collection frequency and shorter collection interval performance better under heavy traffic burden. Fig. 12(b) shows the curve change under our dynamic collection period scheme. From the figure, it can be seen that the dynamic scheme always occupies the minimum key transfer time among results. This is because the optimal choice of collection periods are computed using

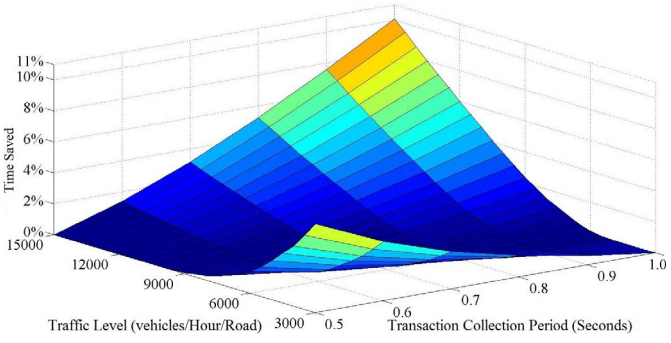


Fig. 13. Decreased key transfer time in percentage.

Algorithm 3. The algorithm forces SMs to select a t_{CP} which forces the system to transfer keys with the minimum time cost.

We further studied the time-saving performance of the dynamic scheme. Fig. 13 plots the average decreased time as a function of various traffic levels and t_{CP} ranging from 0.5 to 1 s. We observe that under the heavier traffic level, the more frequently transaction collection, the lower proportion of decreased time. In contrast, infrequent transaction collection guides to a larger proportion of decreased time at off-peak traffic level. Albeit fewer handshakes, longer collection period takes more than 10% of time cost to finish key transfer at peak traffic level. Thus, for higher traffic levels, using a shorter t_{CP} becomes an economic selection to release the computation burden and improve system efficiency. Shorter collection period, on the other hand, consumes more time to transfer transactions at low traffic situations.

V. Conclusion

In this paper, we propose a novel key management scheme for key transfer among SMs in heterogeneous VCS networks. Our scheme introduces blockchain concept and optimizes the performance using dynamic transaction collection periods. The proposed blockchain structure allows key transfer securely within the decentralized SM network. We developed an effective and flexible transaction collection period selection method to shrink the key transfer time of blockchain scheme. Two components are discussed: 1) blockchain-based key management scheme and 2) dynamic transaction collection scheme. We first studied cryptographic schemes' processing time which composes the key transfer time. Second, by simulating a range of 0 to 2000 transactions transfer from one security domain to another, our blockchain structure achieves more efficiency and robustness compared to the traditional structure. Finally, dynamic transaction collection period further optimize the key transfer time cost. With the help of our mathematical model, SMs are able to decide how to use different transaction collection periods. This paper focuses to further take privacy issues into consideration, including the investigation of a system which provides both security and privacy. In the future, the extension of our work aims at pseudonym management using blockchain basing on the current system. Our future work aims at pseudonym management using blockchain basing on the current system. Moreover, users are able to decide the tradeoff between security and privacy.

References

[1] P. Papadimitratos *et al.*, "Secure vehicular communication systems: Design and architecture," *IEEE Commun. Mag.*, vol. 46, no. 11, pp. 100–109, Nov. 2008.

[2] Y. Cao, N. Wang, G. Kamel, and Y.-J. Kim, "An electric vehicle charging management scheme based on publish/subscribe communication framework," *IEEE Syst. J.*, to be published, doi: 10.1109/JSYST.2015.2449893.

[3] J. Yang *et al.*, "Local stereo matching based on support weight with motion flow for dynamic scene," *IEEE Access*, vol. 4, pp. 4840–4847, 2016.

[4] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," *Proc. IEEE*, vol. 99, no. 7, pp. 1162–1182, Jul. 2011.

[5] J. Lin *et al.*, "A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet Things J.*, to be published, doi: 10.1109/JIOT.2017.2683200.

[6] N. Ekedebe, C. Lu, and W. Yu, "Towards experimental evaluation of intelligent transportation system safety and traffic efficiency," in *Proc. IEEE Int. Conf. Commun. (ICC)*, London, U.K., Jun. 2015, pp. 3757–3762.

[7] H. Hartenstein and L. P. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *IEEE Commun. Mag.*, vol. 46, no. 6, pp. 164–171, Jun. 2008.

[8] "Intelligent Transport Systems (ITS); vehicular communications; basic set of applications; part 2: Specification of co-operative awareness basic service," ETSI, Sophia Antipolis, France, Tech. Rep. 102 637-2, 2010.

[9] *Dedicated Short Range Communications (DSRC) Message Set Dictionary*, SAE Standard J2735, 2009.

[10] B. Zhou, Q. Chen, P. Xiao, and L. Zhao, "On the spatial error propagation characteristics of cooperative localization in wireless networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 2, pp. 1647–1658, Feb. 2017.

[11] C. K. Wong, M. Gouda, and S. S. Lam, "Secure group communications using key graphs," *IEEE/ACM Trans. Netw.*, vol. 8, no. 1, pp. 16–30, Feb. 2000.

[12] H. Harney and E. Harder, "Logical key hierarchy protocol," IETF Internet-Draft draft-harney-sparta-lkhp-sec-00, SPARTA Inc., Lake Forest, CA, USA, Mar. 1999.

[13] X. S. Li, Y. R. Yang, M. G. Gouda, and S. S. Lam, "Batch rekeying for secure group communications," in *Proc. ACM 10th Int. Conf. World Wide Web*, Hong Kong, 2001, pp. 525–534.

[14] O. Zakaria, A.-H. A. Hashim, and W. H. Hassan, "An efficient scalable batch-rekeying scheme for secure multicast communication using multiple logical key trees," *Int. J. Comput. Sci. Netw. Security*, vol. 14, no. 11, pp. 35–40, 2014.

[15] L. Veltri, S. Cirani, S. Busanelli, and G. Ferrari, "A novel batch-based group key management protocol applied to the Internet of Things," *Ad Hoc Netw.*, vol. 11, no. 8, pp. 2724–2737, 2013.

[16] W. H. D. Ng, H. Cruickshank, and Z. Sun, "Scalable balanced batch rekeying for secure group communication," *Comput. Security*, vol. 25, no. 4, pp. 265–273, 2006.

[17] K. Lu, Y. Qian, M. Guizani, and H.-H. Chen, "A framework for a distributed key management scheme in heterogeneous wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 2, pp. 639–647, Feb. 2008.

[18] Y. Sun, W. Trappe, and K. J. R. Liu, "A scalable multicast key management scheme for heterogeneous wireless networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 4, pp. 653–666, Aug. 2004.

[19] Y. Cao *et al.*, "Geographic-based spray-and-relay (GSaR): An efficient routing scheme for DTNs," *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1548–1564, Apr. 2015.

[20] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. Accessed on Apr. 8, 2017. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[21] J. Shi, J. Wan, H. Yan, and H. Suo, "A survey of cyber-physical systems," in *Proc. Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Nanjing, China, 2011, pp. 1–6.

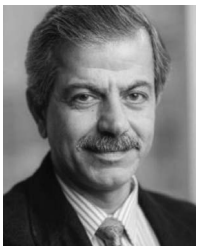
[22] E. A. Lee, "Cyber physical systems: Design challenges," in *Proc. 11th IEEE Int. Symp. Object Component Oriented Real Time Distrib. Comput. (ISORC)*, Orlando, FL, USA, May 2008, pp. 363–369.

[23] R. R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: The next computing revolution," in *Proc. ACM 47th Design Autom. Conf. (DAC)*, Anaheim, CA, USA, 2010, pp. 731–736. [Online]. Available: http://doi.acm.org/10.1145/1837274.1837461

[24] A. Mehmood, M. M. Umar, and H. Song, "ICMDS: Secure inter-cluster multiple-key distribution scheme for wireless sensor networks," *Ad Hoc Netw.*, vol. 55, pp. 97–106, Feb. 2017.
[25] M. Shojafar, N. Cordeschi, and E. Baccarelli, "Energy-efficient adaptive resource management for real-time vehicular cloud services," *IEEE Trans. Cloud Comput.*, to be published, doi: 10.1109/TCC.2016.2551747.
[26] E. Baccarelli, P. G. V. Naranjo, M. Shojafar, and M. Scarpiniti, "Q*: Energy and delay-efficient dynamic queue management in TCP/IP virtualized data centers," *Comput. Commun.*, vol. 102, pp. 89–106, Apr. 2017.
[27] C. Decker and R. Wattenhofer, "Information propagation in the Bitcoin network," in *Proc. IEEE P2P*, Trento, Italy, Sep. 2013, pp. 1–10.
[28] A. M. Antonopoulos, *Mastering Bitcoin: Unlocking Digital Crypto-Currencies*, 1st ed. Cambridge, U.K.: O'Reilly Media, 2014.
[29] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "MedRec: Using blockchain for medical data access and permission management," in *Proc. 2nd Int. Conf. Open Big Data (OBD)*, Vienna, Austria, Aug. 2016, pp. 25–30.
[30] G. Danezis and S. Meiklejohn, "Centrally banked cryptocurrencies," in *Proc. NDSS Symp.*, San Diego, CA, USA, May 2016, pp. 1–14.
[31] G. Zyskind, O. Nathan, and A. S. Pentland, "Decentralizing privacy: Using blockchain to protect personal data," in *Proc. IEEE Security Privacy Workshops (SPW)*, San Jose, CA, USA, May 2015, pp. 180–184.
[32] X. Huang *et al.*, "Software defined networking with pseudonym systems for secure vehicular clouds," *IEEE Access*, vol. 4, pp. 3522–3534, 2016.
[33] N. Z. Aitzhan and D. Svetinovic, "Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams," *IEEE Trans. Depend. Secure Comput.*, to be published, doi: 10.1109/TDSC.2016.2616861.
[34] A. Lei, C. Ogah, P. Asuquo, H. Cruickshank, and Z. Sun, "A secure key management scheme for heterogeneous secure vehicular communication systems," *ZTE Commun.*, vol. 21, no. 3, p. 1, 2016.
[35] M. Raya, P. Papadimitratos, and J. P. Hubaux, "Securing vehicular communications," *IEEE Wireless Commun.*, vol. 13, no. 5, pp. 8–15, Oct. 2006.
[36] R. Schlegel, C.-Y. Chow, Q. Huang, and D. S. Wong, "User-defined privacy grid system for continuous location-based services," *IEEE Trans. Mobile Comput.*, vol. 14, no. 10, pp. 2158–2172, Oct. 2015.
[37] D. He, C. Chen, S. Chan, and J. Bu, "Secure and efficient handover authentication based on bilinear pairing functions," *IEEE Trans. Wireless Commun.*, vol. 11, no. 1, pp. 48–53, Jan. 2012.
[38] D. He, S. Chan, and M. Guizani, "Handover authentication for mobile networks: Security and efficiency aspects," *IEEE Netw.*, vol. 29, no. 3, pp. 96–103, May/Jun. 2015.
[39] D. Hankerson, A. J. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*. New York, NY, USA: Springer, 2006, doi: 10.1007/b97644.
[40] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *Int. J. Inf. Security*, vol. 1, no. 1, pp. 36–63, 2001.
[41] R. C. Merkle, "A digital signature based on a conventional encryption function," in *A Conference on the Theory and Applications of Cryptographic Techniques on Advances in Cryptology.* London, U.K.: Springer-Verlag, 1988, pp. 369–378.
[42] A. Varga, "The OMNeT++ discrete event simulation system," in *Proc. Eur. Simulat. Multiconf. (ESM)*, vol. 9. 2001, p. 65.
[43] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved IVC analysis," *IEEE Trans. Mobile Comput.*, vol. 10, no. 1, pp. 3–15, Jan. 2011.
[44] W. Dai. (2009). *Crypto++ Library 5.6.0*. [Online]. Available: http://www.cryptopp.com
[45] J. A. Rice, *Mathematical Statistics and Data Analysis* (Wadsworth & Brooks/Cole Statistics/Probability Series). Pacific Grove, CA, USA: Brooks/Cole, 1988.

Ao Lei received the B.Eng. degree in communication engineering from the Harbin Institute of Technology, Harbin, China, and the University of Birmingham, Birmingham, U.K., in 2013, and the M.Sc. degree in communication engineering from the University of York, York, U.K., in 2014. He is currently pursuing the Ph.D. degree in communication engineering at the Institute of Communication Systems, University of Surrey, Guildford, U.K.

He is currently involved with two EU-funded projects on security and privacy (PETRAS and B-IoT). His current research interests include security and privacy for vehicular networks and privacy protection for location-based services.



Haitham Cruickshank (M'99) received the B.Sc. degree in electrical engineering from the University of Baghdad, Baghdad, Iraq, in 1980, the M.Sc. degree in telecommunications from the University of Surrey, Guildford, U.K., and the Ph.D. degree in control systems from the Cranfield Institute of Technology, Cranfield, U.K., in 1995.

He is a Senior Lecturer with the Institute of Communication Systems, University of Surrey. He has been involved with research on several European research projects in the ACTS, ESPRIT, TENTELECOM, and IST programs. His current research interests include network security and privacy and satellite network architectures.

Dr. Cruickshank is a member of the Satellite and Space Communications Committee, IEEE Communications Society. He is also a Chartered Electrical Engineer and IEE Corporate.

Yue Cao (M'16) received the Ph.D. degree from the Institute for Communication Systems (ICS), University of Surrey, Guildford, U.K., in 2013.

He was a Research Fellow with the ICS, in 2016, and a Lecturer with the Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne, U.K., in 2017, and has been a Senior Lecturer since 2017. His current research interests include DTNs, e-mobility, and QoS/QoE in 5G.

Philip Asuquo received the B.Sc. degree in computer engineering from the University of Uyo, Uyo, Nigeria, and the M.Sc. degree in computer network technology from Northumbria University, Newcastle upon Tyne, U.K. He is currently pursuing the Ph.D. degree in electronic engineering at the University of Surrey, Guildford, U.K.

His current research interest includes cyber security of critical infrastructures, smart grids and smart homes, intelligent transport systems, and wireless sensor network security.

Chibueze P. Anyigor Ogah received the B.Sc. degree in computer science from Ebonyi State University, Abakaliki, Nigeria, in 2005, and the M.Sc. degree (with Distinction) in computer network technology from the University of Northumbria, Newcastle upon Tyne, U.K., in 2011. He is currently pursuing the Ph.D. degree at the Institute for Communication Systems, University of Surrey, Guildford, U.K.

His current research interests include security and privacy in vehicular networks and Cisco routing protocols.

Zhili Sun (M'99–SM'15) received the B.Sc. degree in mathematics from Nanjing University, Nanjing, China, and the Ph.D. degree from the Department of Computing, Lancaster University, Lancashire, U.K., in 1991.

He is a Professor with the Institute of Communication Systems, University of Surrey, Guildford, U.K. He has been a Principal Investigator and a Technical Coordinator on a number of projects within the European Framework Program including ESPRIT BISANTE, TENtelecom VIPTEN, GEOCAST, ICEBERGS, SATELIFE, and EuroNGI. His current research interests include wireless and sensor networks, satellite communications, mobile operating systems, traffic engineering, Internet protocols and architecture, quality of service, multicast, and security.