DOI: 10.3969/j. issn. 1673-5188. 2016. S0. 004
http://www.cnki.net/kcms/detail/34.1294.TN.20160613.1553.002.html, published online June 13, 2016

A Secure Key Management Scheme for Heterogeneous Secure Vehicular Communication Systems

LEI Ao, Chibueze Ogah, Philip Asuquo, Haitham Cruickshank, and SUN Zhili
(Institute for Communication Systems (ICS), University of Surrey, Guildford, GU2 7XH, United Kingdom)

Abstract

Intelligent transportation system (ITS) is proposed as the most effective way to improve road safety and traffic efficiency. However, the future of ITS for large scale transportation infrastructures deployment highly depends on the security level of vehicular communication systems (VCS). Security applications in VCS are fulfilled through secured group broadcast. Therefore, secure key management schemes are considered as a critical research topic for network security. In this paper, we propose a framework for providing secure key management within heterogeneous network. The security managers (SMs) play a key role in the framework by retrieving 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 Group Key Management (GKM) scheme basing on leaving probability (LP) of vehicles to depart current VCS region. Vehicle´s LP factor is introduced into GKM scheme to achieve a more efficient rekeying scheme and less rekeying costs. The second component of the framework using the blockchain concept to simplify the distributed key management in heterogeneous VCS domains. Extensive simulations and analysis are provided to show the effectiveness and efficiency of the proposed framework: Our GKM results demonstrate that probability-based BR reduces rekeying cost compared to the benchmark scheme, while the blockchain decreases the time cost of key transmission over heterogeneous networks.

Keywords

leaving probability; blockchain; group key management; heterogeneous; vehicular communication systems (VCS)

1 Introduction

Vehicular communication systems (VCS) supports not only message exchange among vehicles, but between cars and infrastructure facilities as well. Infrastructure access points in VCS are called Road Side Units (RSUs) [1]. RSU acts as a base station in VCS and covers a small section on the road. Traditional VCS is comprised of multiple RSU cells and offers a platform among intelligent transportation systems (ITS) for vehicles to exchange different kinds of messages such as safety notification messages. With the help of VCS, ITS can offer a more safe and efficient traffic management, which is the basic function of ITS. Moreover, commercial applications, such as electric vehicle charging [2], can be implemented on a dedicated platform. A recent report from U.S Department of Transport (DoT) shows that 82% of the accidents can be prevented by using ITS systems [3]. 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 important topic for research. High mobility, large volume, frequent handoff of vehicular nodes and heterogeneity networks pose different challenges compared to the traditional mobile networks.

ITS spans across a wide range of applications which are classified into two categories: vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) [4]. VCS security highly relies on the safety for exchange of beacon messages. These beacon messages are usually referred to as Cooperative Awareness Messages (CAMs) for EU [5] or Basic Safety Messages (BSMs) for US [6], 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 BSMs broadcasted by other group members. The trustfulness and legality of BSM information is proved by encrypting safety messages with a pre-agreed 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 group keys among all the communicating participants. Several group key management (GKM) approaches for mobile networks (e.g Logical Key Hierarchy (LKH) and One-way Function Tree) have been presented in recent years. Unfortunately, these approaches are quite inefficient for VCS application due to huge number and high variability in vehicular nodes. Hence, there is need for a novel and more efficient key management scheme

for VCS.

To meet the security requirements, GK has to be refreshed and redistributed (rekeying) securely whenever a group member changes in order to achieve forward and backward secrecy [7]. This approach poses the challenge of rekeying efficiency. Several approaches aim to improve efficiency of managing keys for group nodes, and schemes for individual node rekeying like key tree approaches [8], [9] are developed to ease the problem. Furthermore, Batch Rekeying (BR) [10] is proposed to significantly improve efficiency compared to individual rekeying schemes. But these approaches are not suitable for VCS application as the number of mobility nodes may be huge in VCS. The authors in [11] introduce BR into multiple key trees and select the tree with less rekeying cost upon rekeying. However, nodes in [11] are traditional mobile nodes with irregular trajectory. Paper [12] presents a GKM scheme for Internet of Things applications, including VCS scenarios. Based on the idea in [10], authors introduce their method for VCS but mainly focus on key initialisation and registration stage.

Aside from the aforementioned problem, heterogeneity issues are an inevitable aspect in wireless networks. Heterogeneity in wireless network refers to either the difference on the traffic volumes, or distinct network structures. Heterogeneous traffic volumes are classified as nodes densities or message traffic capabilities [13] while the heterogeneous network structures normally stand for the network managed under different topologies [14], [15] or central managers. These heterogeneities are the major considerations in evaluating the essential requirements of VCS key management scheme. Recently, heterogeneous vehicular communication networks are given more attention. The heterogeneity in terms of different central authorities has become a reality problem as VCS is considered as a worldwide system covering multiple countries. Specifically speaking, two RSUs in different security domains should be able to keep understanding messages from the same car passing through their common border between domains. With this in mind, user cross-domain hand-offs must not be overlooked in VCS.

In this paper, we propose a key management scheme for VCS scenarios, including the group batch rekeying scheme and key transmission between two heterogeneous networks. Different form the previous group batch rekeying schemes [7], [10]–[12], Leaving probability (LP) is introduced into the proposed scheme to further reduce rekeying cost in order to achieve better efficiency. Furthermore, with the help of blockchain, a simplified handshake procedure is achieved for heterogeneous networks. Performance evaluations of this paper demonstrate that LP approach achieves much less rekeying overhead compared to the benchmark BR scheme. The time consumption result of heterogeneous key management approach is compared with that in traditional network structure to prove that the blockchain concept helps to shorten the key transmission handshake time.

The remainder of this paper are organised as follows: Section 2 briefly introduces key management techniques. Model overview and details of our scheme are displayed in section 3. We describe our system model, and then introduce the rest part of our ideas, namely, LP, vehicle initialisation procedures and key transmission between heterogeneous networks. Scenario parameter assumptions, key registration procedures, rekeying costs and blockchain performance are analysed in Section 4. Section 5 concludes the paper and presents some future plans.

2 Related Work

2.1 Key Tree Approach

Key tree approaches include the key graph approach and LKH, which are scalable structures to manage large volume of nodes. Hierarchy tree reduces the processing complexity of each member change request from $O(N)$ to $O(\log_d N)$ [16], where d is the degree of key tree and N is the group size. In key tree structure, GK is placed at the root of the tree. It is called to encrypt messages whenever a member wants to exchange messages with others. This means that all the group members own a copy of GK and these members knows all the details about the GK, so the GK must be a symmetric encryption scheme key, such as Advanced Encryption Standard (AES) [17]. Individual keys (IK) are located at leaf nodes of the tree, they are user nodes in the broadcasting group. The rest of tree nodes are logical key nodes which are used to encrypt parent keys, called Key Encryption Key (KEK) [7]. In wireless network, mobility nodes form the mainstream composition of network, especially in VCS. Therefore BR is a critical method to reduce a large proportion of rekeying messages, which is caused by individual rekeying. To eliminate high rekeying cost in individual rekeying, BR scheme collects all member modification requests within a certain period of time and triggers rekeying broadcasting at end of the period. In this way, key manager aggregates multiple broadcast messages into a single one and achieves much better efficiency, where the period of time is batch interval t_{BR} and end of the period is called batch edge.

GKM algorithm in [10] is the first scheme using batch conception and it was cited by large number of batch rekeying papers. The authors assume there are J vehicles joining the group and L vehicles leaving, respectively. Four situations are classified as follows:

- Case-1: If $J = L$, new joining users replace the previous places of leaving users.
- Case-2: If $J < L$, joining members fill into J minimum-depth tree leaf vacancies of the departing users.
- Case-3: If $J > L$ and $L = 0$, the key manager first finds the shallowest node and remove it, then forms the node and joining users as a new subtree. Finally the key manager inserts

the tree at the deleted point.

- Case-4: If $J > L$ and $L > 0$, the central manager executes steps in case-2 first and operates algorithm in case-3 afterwards.

It is a framework for all mobile networks, but not dedicated for VCS applications. With this in mind, the probability factor in VCS scenarios can be involved in joining member ordering and inserting point selection as well. Details are discussed in section 3.

2.2 Blockchain Applications

A lot of attention has been attracted to the blockchain concept since its parent production, bitcoin, was launched in late 2008 [18]. The core idea of blockchain is that it maintains a distributed and synchronised ledger of transactions. It benefits to accountability function by using block look-up, which is fairly useful since the malicious user must be revoked in time. More importantly, a transaction can be used to transmit information among decentralised network. Even through there is no centralised manager, the key to maintain the information correctness and integrity in blockchain network is that all the blocks are distributed verified by large of network participants (miners) [19].

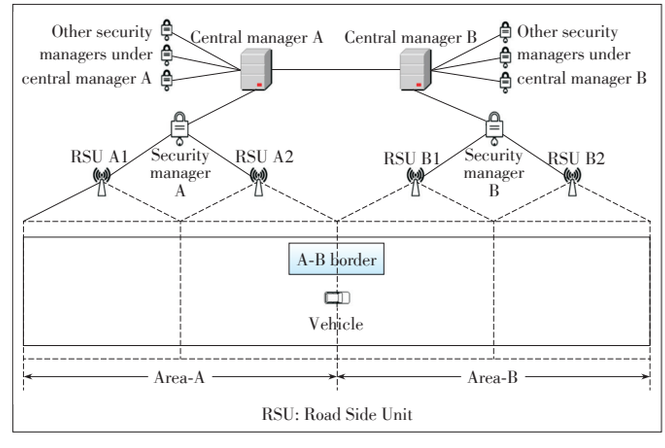
To the best of our knowledge, no previous works have adopted the blockchain mechanism to transmit information for wireless network applications, let alone the VCS applications. In this paper, we utilise the Security Managers (SM) network to transmit and verify vehicle keys in the across border requests, rather than forwarding them to the third party authorities.

3 Proposed Framework

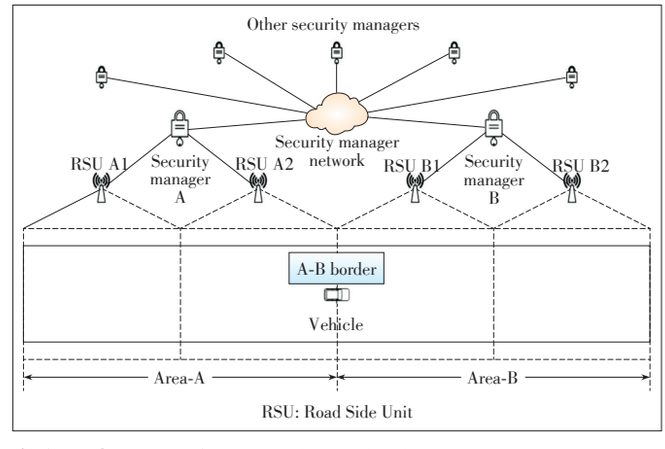
3.1 System Model

We focus exclusively on a system of vehicles each equipping an On Board Unit (OBU) embedded with wireless communication module based on the IEEE 802.11p standard. The OBU enables vehicles to communicate with nearby vehicles and infrastructures (RSU). The RSUs are equipped with the identical wireless standard. Security managers are placed on the upper level of RSUs and their logical coverage area is called security domain. As demonstrated in **Fig. 1**, Area-A is a security domain which is managed by security manager A. This traditional network structure employs central managers (or trusted third party authority) at top of the network to manage cryptography materials, this however makes it an inefficient key exchange, and will require supernumerary handshakes if a car passes from one security domain to another. The key transportation achieved by our approach could thus be simplified by using blockchain mining method, meaning the messages will be verified by SM network but not third party authorities. For instance, let us consider a scenario in which two cars in same security domain apply to depart, each on going in a different direction. The trust authority must send two distinct messages in order to finish key transmission. In our model, on the other hand, simplifies the network structure, specifically the trust third party authorities. Similar to the bitcoin network, the function of blockchain enable nodes to share information without the need for a central party to secure this ledger. The trusted third party authorities only take part in distributing initial keys, while the cryptography issues are computed by SM, which is placed at higher level of the network. As shown in **Fig. 2**, SM is connected with a "cloud" that may link with SMs on other domains and certification entities with a territory.

A key management scheme has three functional components: key initialisation, group key management and key transmission between heterogeneous networks [16], [20]. Our model assumes that the key initialisation is managed by the third party central authorities. We suppose the central authorities have secure communication link with SMs. Therefore, authorities are responsible for generating the permanent vehicle identities only. Vehicles travel on a road and periodically transmit safety messages using OBU, which are collected by RSU that are built along the road at regular intervals. The RSU forwards received messages to the upper level SM to verify the authentici-



▲**Figure 1. Traditional network structure.**



▲**Figure 2. Blockchain based network structure.**

ty of such messages. The aforementioned group key management is executed by SMs. They start their rekeying work by using wireless IEEE 802.11p broadcasting within their own security domain, which is triggered depending on member alteration. The messages are supposed to share with neighbouring SMs to transport keys if they indicate a SM-border-crossing action. Similar to bitcoin applications, the crossing border actions are encapsulated into transactions and a block is formed by multiple transactions within a short period of time. Aside from this, the SMs take the role of miners. 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 list of new joining members is delivered by retrieving the information from a block.

3.2 Probability Based Group Key Management

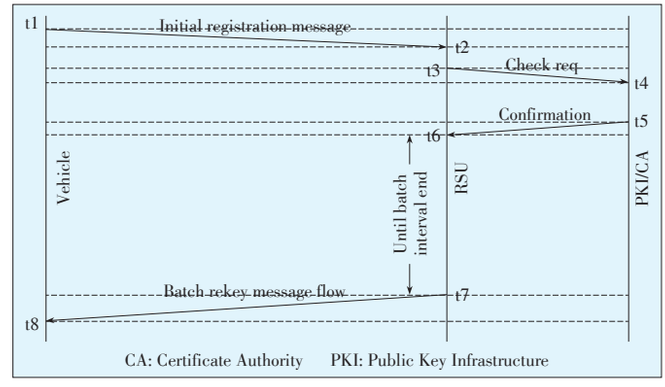
A key management scheme has three functional components: key initialisation, group key management and key transmission between heterogeneous networks [16], [20]. Our model assumes that the key initialisation is managed by the third party central authorities. We suppose the central authorities have secure communication link with SMs. Therefore, authorities are responsible for generating the permanent vehicle identities only.

3.2.1 Joining Handshake

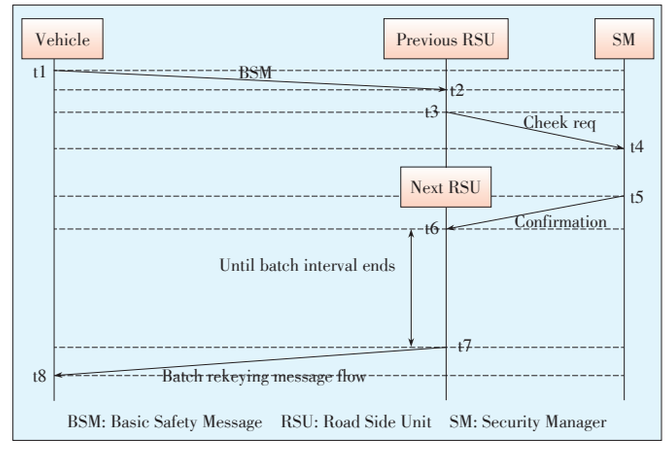
Cryptographic encryption schemes and certificates are introduced to provide security in ITS [21]. Public/private key pairs and certificates are managed by Public Key Infrastructure (PKI) and Certificate Authority (CA), respectively. IEEE1609.2 [21] defines the use of powerful cryptographic schemes such as Elliptic Curve Integrated Encryption (ECIES) [22] for individual encryption (encrypting the rekeying block only for a single user but not for a group), which requires more processing resources. AES is used for group communication, which is considered as a lightweight symmetrical encryption algorithm. In our scheme, all vehicles hold either permanent or temporary certificate in order to complete joining handshake work. A temporary certificate is assigned before vehicle leaves the manufacturer. As shown in **Fig. 3**, new vehicles need to use the temporary certificateto send an Initial Registration Message (IRM) for self-registration at initial participation in ITS environment.

Permanent certificates become effective whenever a vehicle changes to another RSU area under the same security domain. The SM checks the correctness of the safety beacon messages. In this case, a new RSU obtains the region changing information from the verified safety beacon messages. **Fig. 4** illustrates the above procedures. For the above situations, SM and RSUs need to collect vehicle entry and exit information via BSMs or IRMs to achieve batch rekeying.

When a vehicle attempts to move into a new RSU area that



▲**Figure 3. Vehicle initial joining handshake.**



▲**Figure 4. Vehicle RSU changing handshake.**

is under administration of the same SM, it keeps broadcasting BSMs using previous GK: $AES\{Info, GK\} + ECDSA\{Cip, K_{priv}\} + Cert_p$, where $Info$ is the safety information, K_{priv} is private key of vehicle and Cip is ciphertext. Permanent certificate $Cert_p$ includes authorised receipt to prove that the certificate holder possesses a legal digital receipt and public/private key pairs which are authenticated by local SM. The RSU forwards the certificate and signature to the applications layer of SM, after receiving the check request. Digital signature scheme Elliptic Curve Digital Signature Algorithm (ECDSA) [23] is used in our scenario to provide better degree of security. The legality of the vehicle's identity is verified by SM and a confirmation message is then sent back to RSU. The RSU starts to prepare the rekeying message upon "Confirmation" receipt. The rekeying broadcast is sent until the start of next batch interval. We assume that both pervious and new RSU can receive the BSM. Thus, previous RSU knows the leaving activity, while the new RSU obtaining information from the same BSM as RSUs are designed to store GK of its neighbours.

3.2.2 Leaving Probability

LP of mobile node is defined in [24] as an average number of nodes leaving the group within a rekeying interval. For tradi-

tional mobile networks (e.g. 3G, LTE and 5G network), entrance and departure of portable nodes are unpredictable. Hence some key management schemes require nodes subscribing several rekeying intervals in order to calculate leaving probability. Unfortunately, security vulnerabilities appear when system allows users to select their own subscription period: a malicious user eavesdrops critical messages by asking active period longer then its real residence time.

Probability models are much easier to implement for vehicle nodes in VCS since they have predictable moving trajectory. With this in mind, a dedicated LP calculation algorithm is needed for VCS scenarios. According to the traffic survey [25] at a one-directional urban road, speed distribution fits normal distribution function in (1) [26].

$$f(x|\mu,\sigma)=\frac{1}{\sigma\sqrt{2\pi}}e^{\frac{(x-\mu)^2}{2\sigma^2}}, \tag{1}$$

where μ is the mean or expectation of the distribution and σ is the standard deviation. With the help of the speed distribution and vehicle specifications, the central manager is able to compute the possible speed range (PSR) and possible departure speed range (PDSR). The upper boundary U_{PSR} stands for the maximum speed in which vehicle can reach at end of current batch interval (t_{BR}). Similar to U_{PSR} , L_{PSR} is for the minimum speed if vehicle tries to slow down. In addition, U_{PSDR} and L_{PSDR} are the highest and lowest speed for car to leave the current RSU coverage, respectively. We assuming $d_{\mathrm{Re}main}$ is the distance between the vehicle current position and coverage border which is directly ahead of the vehicle. Thus, the leaving probability P_L is calculated as (2):

$$P_L=\frac{\int_{L_{PSDR}}^{U_{PSDR}}f(x|\mu,\sigma)\mathrm{d}x}{\int_{L_{PSR}}^{U_{PSR}}f(x|\mu,\sigma)\mathrm{d}x}. \tag{2}$$

RSU knows vehicle's maximum positive and negative acceleration by listening to the safety beacon messages, thus it is easier to calculate the upper and lower boundary ($V_{\max}$ and $V_{\min}$) of PSR. For PSDR boundaries, $V_{dep-\max}$ stands for the maximum speed for vehicle to depart. There are two different extreme situations:

1) The vehicle keeps speed-up with maximum positive acceleration a_+ until the speed reaches $V_{\max}$. The speed is kept until the end of the batch interval. The overall distance d_{Remain} is covered by the vehicle.
2) The vehicle already has enough speed and d_{Remain} is short enough so that the vehicle is able to leave the region easily. The vehicle speeds up with an acceleration lower than a_+ . It reaches $V_{\max}$ at mid of t_{BR} and keeps the speed $V_{\max}$ until the end of the batch interval.

Similarly, there are two possible situations of $V_{dep-\min}$:

1) The current speed $V_{current}$ is fast enough for vehicle to leave the RSU region, therefore the minimum speed for the vehicle to leave $V_{dep-\min}$ is decided by decreasing speed until the end of t_{BR} under the assumption that the node can travel d_{Remain} .
2) The vehicle has to speed up in order to depart in t_{BR} , therefore the node first improves the current speed from $V_{current}$ to $V_{dep-\min}$, and then keeps it until the end of the batch interval.

According to the possibilities above, the first situation is that the vehicle can leave the region only by driving with current speed:

$$V_{dep-\min}=\frac{2\cdot d_{remain}}{t_{BR}}-V_{current}. \tag{3}$$

Here it is assumed vehicle spends time t_1 speeding up to $V_{dep-\max}$, and $V_{dep-\max}$ are calculated in (4):

$$\begin{cases} V_{dep-\max}=V_{Current}+t_1\cdot a_+ \\ d_{remain}=V_{dep-\max}\cdot(t_{BR}-t_1) \\ +0.5\cdot t_1\cdot(V_{dep-\max}+V_{current}) \end{cases}. \tag{4}$$

Therefore, $V_{dep-\max}$ is computed by a summarised equation:

$$V_{dep-\min}=V_{current}+a_+\cdot t_{BR}+\sqrt{2\cdot a_+(V_{current}\cdot t_{BR}-d_{remain})+a_+^2\cdot t_{BR}^2}. \tag{5}$$

To sum up, LP can be generated by using **Algorithm 1**.

Algorithm 1 Leaving Probability Calculation

Input: Current speed $V_{Current}$, distance to coverage border d_{remain} , vehicle maximum positive and negative acceleration a_+ & a_- , batch interval t_{BR} , maximum speed the vehicle can reach V_{limit}

Output: Leaving Probability (*LP*): P_L

```
Max speed in t_BR: V_max-expect = V_current + a_+ · t_BR
if ( V_min-expect ≥ V_limit ) then
        d_max in t_BR, keep gaining speed until V_limit;
else
        d_max in t_BR, keep gaining speed until V_max-expect;
endif
if ( d_max ≥ d_remain ) then
        V_dep-max = min( V_max-expect , V_limit );
else
        Set LP for this node P_u = 0;
 endif
 MIN speed in t_BR: V_min-expect = V_current - t_BR × a_- ;
 if ( V_current 6 · t_BR ≥ d_remain ) then
        call equation (3) to calculate V_dep-min ;
 else
        call equation (5) to calculate V_dep-min ;
 endif
Calculate maximum and minimum possible speed of
the vehicle, V_max and V_min ;
LP is calculated by employing V_dep-max , V_dep-min , V_max
and V_min into equation (2);
```

20: **End Algorithm**

3.2.3 Leaving Ratio

However, in a VCS scenario, most of the vehicles have no chance to leave the communication group before the next batch edge since it is impossible for them to reach the speed to leave the region border in rekeying interval. For this reason, another parameter Leaving Ratio (LR) is involved to substitute LP. Within the range (0, 1], LR is a ratio of the rekeying interval and time cost for the vehicle leaving the broadcast border. Similar to the definition of LP, LR represents the inverse of the number of rekeying intervals using for a vehicle to leave the group:

$$LR = \min\left(1, \frac{t_{BR}}{t_{out}}\right). \quad (6)$$

The parameter t_{out} is the time cost for a vehicle to leave, which is computed by (7):

$$t_{out} = \frac{d_{remain}}{V_{current}}. \quad (7)$$

3.2.4 Joining User Sequence

According to the batch rekeying scheme in [10], new joining users have two circumstances to be attached to the key tree:

1) New joining users fill into the vacancies caused by departure users.
2) New users joining the subtree form a subtree and the subtree is inserted into the key tree.

Both the circumstances are related to inserting fresh nodes in order of LP and LR values. In our scenario, nodes are arranged according to LP and LR with either positive or negative sequence. LP is considered with higher priority compared to LR during work arrangement. LR is taken into operating if the rest of the nodes are with LP equal to zero. For example, if the joining users are arranged with leaving probabilities from high to low, the sequence should be $LP_{high} > LP_{med} > LP_{low} > LR_{high} > LR_{mid} > LR_{low}$.

3.3 Heterogeneous Key Management

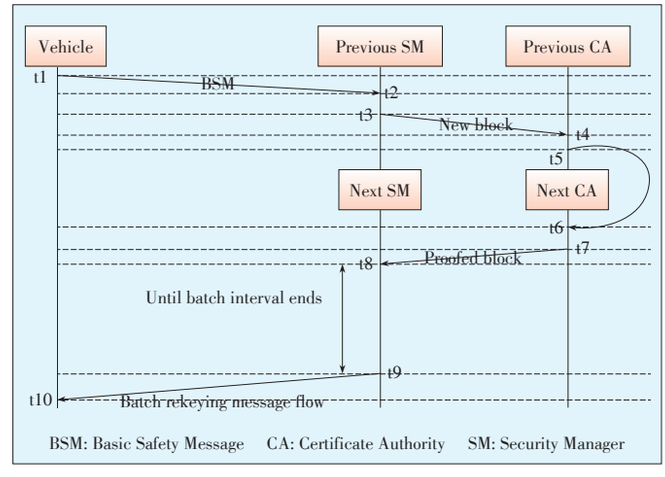
We propose the blockchain concept for heterogeneous key management, which aims to simplify the distributed key management in large heterogeneous security domains. A lightweight and scalable key transmission scheme is implemented in our scheme by using blockchain.

3.3.1 Heterogeneous Key Management

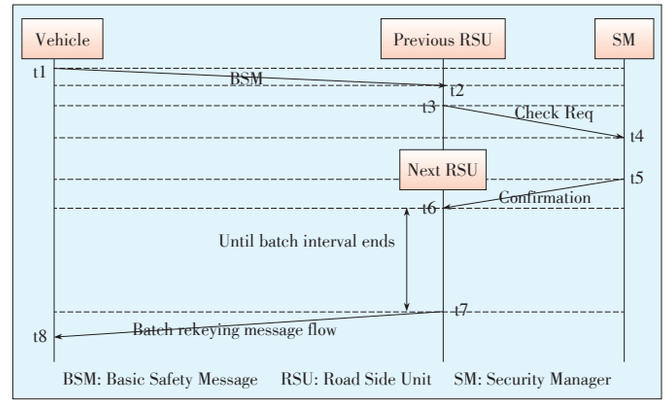
The handshake process in the traditional network is shown in **Fig. 5**. When a vehicle attempts to join a new geography territory in which infrastructures are managed by a new certificate authority (CA), the old CA picks up this border crossing activity from the beacon messages that are sent by the vehicle. Then it generates a border crossing request along with useful information related to the vehicle and forwards all these materials to the next CA. A new group key will be sent to the vehicle after new CA has verified such cryptography materials. This however delays the key transmission between two security domains. The blockchain concept is one approach to facilitate this, because it eliminates the third party authorities and allows decentralised key transmission between networks. We abstract handshake steps of blockchain network (**Fig. 6**). In our model, border crossing requests are gathered into transactions, these transactions are further collected into a candidate block. This candidate block is then distributed into the SM network for other SMs to verify, which follows the mining processes in bitcoin network [19]. The mined block is returned back to SM network after the solution of the proof-of-work has been found [27] and the destination SM retrieves the joining vehicle information from this block.

3.3.2 Transaction Format

Transactions are designed to encapsulate key transmission materials from the source SM to destination SM. Six fields are contained in the transaction of our model (**Table 1**). Hash in



▲**Figure 5. Heterogeneous key management in traditional network.**



▲**Figure 6. Heterogeneous key management in blockchain based network.**

▼**Table 1. The transaction format**

Field	Description
Transaction hash	A Hash of the transaction
Transaction number	Number of this transaction in block
Current security manager	Current security domain
Destination security manager	Next security domain
Vehicle identity (current pseudonym)	Current vehicle pseudonym
Vehicle certificate	Certificate of the pseudonym

the first field aims to simplify computation burden of miners. The destination SM knows the existence of new joining vehicles from the fourth field if the value in this field matches the SM identity. Even more important, the destination SM can encrypt the rekeying message by using the vehicle public key, which is embedded in vehicle certificate in the last field. As one of the most important metric to measure the performance of blockchain, the number of transactions in bitcoin related research is how many transactions are mined in a second. However in VCS scenarios, we use an alternative definition, which is the average number of transactions in a block.

3.3.3 Block Format

The block header is constructed by six fields (**Table 2**), similar to the bitcoin block. The second field links the block to its parent block. All the transactions in the block are embedded into the header using a piece of data content, that is, the merkle tree root [28]. The merkle tree root assures the integrity of transactions as the alteration on transactions causes a totally different value of the merkle root value. Timestamp protects the block from time tampering. Without loss of generality, difficulty is a metric of how difficult it is to successfully find a hash. However, there are two distinct ways of describing the difficulty. The first describes it as the number of zeros at the start of the hash result of the block header, while the second one measures an estimated difficulty target. The target is the number of hash calculations to mine a block. An acceptable block must have a hash below this target level. We propose the same difficulty format as it in the bitcoin block, with the first two hexadecimal bits for the exponent and the remaining part is coefficient. Hence the target difficulty can be computed using (8) [27]:

$$target = coefficient \times 2^{8\times(exponent-3)}. \quad (8)$$

▼**Table 2. The block format**

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 of transactions
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)	
Field	Description
Transaction 1	The first transaction in this block
…	…
Transaction *N*	The last transaction in this block

3.3.4 Mine Proof-Of-Work

In bitcoin, proof-of-work is a digital receipt which is hard to calculate but easy for others to verify [18]. A one-way cryptographic hash function, double SHA256, *dhash*(), is used to calculate the proof-of-work. This function is used in various fields of the bitcoin system [27], including the calculation of the merkle root. The result is calculated by hashing the candidate block header repeatedly, using different nonce value, until the resulting hash value matches the difficulty requirement. More specifically, the block is successfully mined if the hash result starts with the numbers of zeros. The number of zeros is equal to the difficulty.

To mine a block, each time a block candidate is released into SM network, and the hash of the block header is calculated by SMs. At the start of mining, a difficulty target is computed to get the maximum acceptable hash calculation times. An arbitrary number between 0 and the difficulty target is selected as the initial hash attempt number to start mining. As most of the proof-of-works does not appear within a small value of attempts. If it fails to find the proof-of-work within above the value range, the attempt value should starts from 0 to see if there is an answer among small numbers. When the total calculation times exceed the difficulty target, the SM fails to find a proof-of-work basing on this block. Therefore, the transactions must be rearranged and mined again. However, the mining work is aborted when the proof-of-work is found by someone else in SM network. **Algorithm 2** shows a summarised pseudocode of mining procedure.

Algorithm 2 Calculate Nonce (Proof-Of-Work)

Input: Candidate Block Header ***H***
Output: Nonce value nonce

1: Summarise the first five header fields in a basic string ***S***;
2: Calculate the difficult target *tar* using equation (**8**);
3: Initialise the tries number *nonce*, tried string *try*, output *result* from the double hash function *dhash()*;
4: Pick a random number **n** = *Random [0, tar)*;
5: *nonce* = **n**;
6: **while** (result is not found & *nonce* ⩽ *tar* & Not receive Proof-Of-Work from other SM) **do**
7: *result* = *dhash(try* + *nonce)*;
8: *nonce* ++ ;

9: **end while**
10: *nonce* = **0**;
11: **while** (result is not found & *nonce* ≤ **n** & Not receive Proof-Of-Work from other SM) **do**
12: *result* = *dhash(try* + *nonce)*;
13: *nonce* ++ ;
14: **end while**
15: **if** (result is not found & Not receive Proof-Of-Work from other SM) **then**
16: **return** (*nonce* - 1);
17: **else**
18: Generate a new block header hash value by changing the sequence of transactions **then** Repeat the aforementioned steps;
19: **end if**
20: **End Algorithm**

4 Simulation and Evaluation

4.1 Assumptions

The assumed parameters are shown in **Table 3**. Our scenario is set to have each single RSU coverage range with 600 m and the maximum transmit power $P_{t-\max} = 20\,mW$ [29] in vehicles in the network simulation (Veins) [30]. VCS networks need decentralized management by RSU cells due to the fact that ITS application has to be employed in large scale geographical area. Therefore RSU in this scenario acts as the central key manager and a relay between vehicle nodes and the administrator agency. The 2^{10} vehicles pass an 8-row road area. The number of vehicles and rows are considered under a saturated traffic condition. The saturated traffic aims to exam our scheme under the worst case (as well as the heaviest burden of VCS). The vehicle speed follows normal distribution with μ = 46.56 and σ = 6.88 [25] while the departure time follows exponential distribution.

To improve rekeying efficiency, key tree structure of this scenario is based on LKH [8], [9] with binary tree degrees. The higher tree degrees result, the more node individual encryption upon rekeying. Batch rekeying is considered in the model with batch rekeying interval t_{BR} is set to 0.5 s. The benchmark BR scheme [10] is used. This scheme is the basic framework for all mobile networks. Even though there are some incremental schemes based on it, such as [7], but none of them are focus on VCS scenarios. Moreover, recent papers [11], [12] still use [10] as their basic idea.

We assumed that blocks are mined by Digilent Nexys-2 500 k that is considered as one of the lowest cost FPGA mining devices. This device can finish 5 million hash calculations per second. We take an average distance of 5000 m between SMs, while the distance between SM and RSU is set to 1000 m. The average transactions in a block is constrained by 2 and 128, which means the average vehicle departure requests a range by 2^{10} and 2^{7}. The range of the difficulty level is defined by 3 and 5.

▼Table 3. Assumption of scenario parameters

Parameter name	Parameter value
Length of RSU coverage area	600 m
BSM transmit power $P_{t-\max}$	20 mW
Overall vehicle number	2^{10} vehicles
Length of rekeying interval t_{BR}	0.5 s
Distance between SMs	5000 m
Distance between SM and RSUs	1000 m
Range of transaction numbers	2, 4, 8, 16, 32, 64, 128
Range of difficulty (the number of zeros)	3–5
Mining speed	5 million hashes per second

RSU: Road Side Unit SM: Security Manager

4.2 Key Initialisation

Table 4 presents the time cost for a vehicle to register to a RSU when it joins a new broadcast group. Results are generated in OMNeT++ 4.5 [30], [31]. The steps in the table follow the handshake routes in Fig. 3. Step 8 is a unique progress for batch rekeying, the central key manager collects all member list modification requests in this batch period and waits for the start of next batch interval. The rekeying message has complex format which contains information for all group members, therefore the processing time $t_{prepare}$ is much longer than other steps.

The vehicle sends IRM messages without any record about GK, therefore, it has to use its own public key to encrypt moving state information. ECIES with elliptic curve secp160r1 in Crypto++ [32] is selected for the cryptographic schemeECIES, and digital signature scheme ECDSA as well. The cipher block has a length of 75 bytes because ECIES provides much better

▼Table 4. Event timestamps

Step name	Timestamp
1. Vehicle joining	$t_0 = 0\ ms$
2. Registration Msg→RSU	$t_1 = 2.910098956\ ms$
3. RSU receives Msg	$t_2 = 3.040167479\ ms$
4. RSU checks Msg→PKI/CA	$t_3 = 4.350436255\ ms$
5. PKI/CA receives Msg (via router/switch)	$t_4 = 7.350735578\ ms$
6. PKI/CA checks Msg and sends toRSU	$t_5 = 7.351695577\ ms$
7. RSU receives Msg and prepares rekey	$t_6 = 7.372535577\ ms$
8. Send at next batch edge Wait time Rekey Msg preparation time	$t_{send} = t_{BR}$ $t_{wait} = t_{BR} - t_6$ $t_{prepare} = 4.289728099\ ms$
9. Send out rekey Msg	$t_7 = t_{BR} = t_{send}$
10. Vehicle receive rekey Msg	$t_8 = t_{BR} + 0.174698201\ ms$

CA: Certificate Authority PKI: Public Key Infrastructure RSU: Road Side Unit

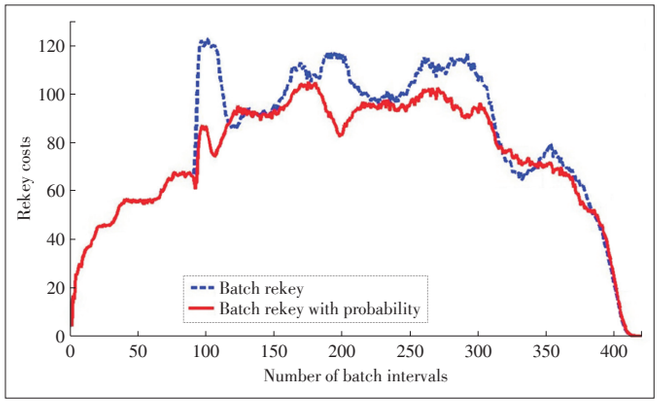
security level. The previous registered vehicle sends the normal BSM to inform RSU about region changing activity. The mobility state in BSM is encrypted by AES-CCM mode [33] by GK. The cipher text of AES has 32 bytes, which provides better efficiency. Digital signatures in both IRM and BSM are generated by ECDSA to demonstrate the authenticity of digital documents. In our scenario, the length of signature is 42 bytes, which provides authentication for messages.

4.3 Rekeying Costs

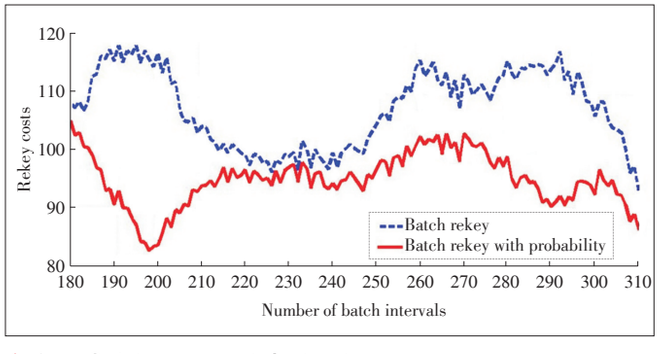
Our scheme is compared with the benchmark scheme from the aspect of rekeying costs, with reference to the batch interval number. To eliminate errors and generate a clear graph, 1000 times Monte Carlos simulations are used.

Fig. 7 demonstrates the rekeying costs of two schemes during a traffic flow of 2^{10} vehicles passing through. From the start to around the 80th batch interval, results are overlap to each other. The results are the same between the two schemes because the probability issue has not yet taken effect at the joining-only situation. Similar results are obtained after 350th interval.

The first node leaving activity happens in the 80th to 120th intervals. We can see that the probability based batch rekeying scheme has much better results when a node leaves suddenly, with approximately 33% less rekeying cost than the benchmark. More details about this region are shown in **Fig. 8**. The



▲**Figure 7. Batch rekey costs for the complete simulation period.**



▲**Figure 8. Batch rekey costs for the stable phase.**

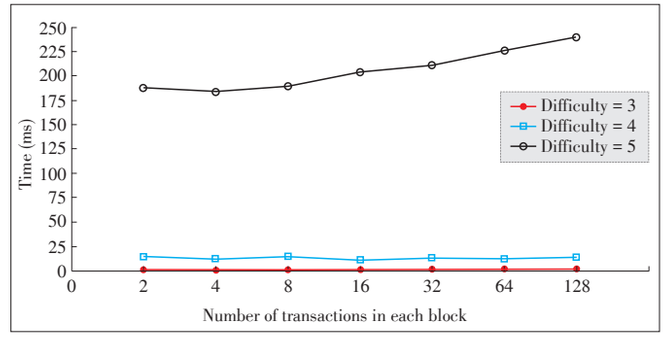
period from the 180th to 310th batch intervals is critical since both joining and leaving appear. Hence Most papers have focused on this period.

Fig. 8 presents more details about the results of the two schemes in the stable phase. The rekey cost for the benchmark algorithm has a sharp increase at about the 185th batch interval. A comparison of our proposed approach to the benchmark scheme shows that our scheme displays a more steady performance which means better robustness. The benchmark scheme shows a significant fluctuation which makes it difficult for the key manager to maintain the required Quality of Service (QoS) through the entire working period. In addition, the overall rekeying cost of our scheme is on average 18% less than that of the benchmark.

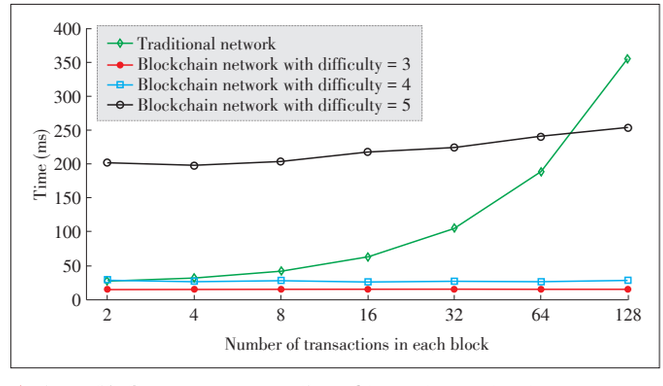
4.4 Key Transmission

Results of the mining time are compared in terms of mining difficulties. **Fig. 9** shows the mining time increases exponentially with the growth of difficulty. Mining runs in a short period of time when the level of difficulty equals to 3. The result of difficulty level 4 costs nearly double the time of difficulty 3 and their curves remain steady. However, difficulty level 5 costs nearly 8 times the time of difficulty 4 and the curve increases linearly.

Performance of key transmission is measured by the block propagation time from the current SM to destination SM. The overall handshake time cost in millisecond is shown in **Fig. 10**



▲**Figure 9. Blockchain mining time.**



▲**Figure 10. Overall handshake time of two network structures.**

in comparison with traditional VCS network structure. The handshake time of the standard network structure increases exponentially with an increasing number of transactions, which is due to the fact that CA must verify each transactions. The handshake time of the traditional network is much more than that of the blockchain network when difficulty is less than 5.

5 Conclusions

In this paper, we propose a key management scheme for group secure communication in heterogeneous VCS networks. Our scheme includes three components: group key management, key registration and key transportation. By simulating a vehicle group passing through different SM areas, our batch rekeying algorithm achieves more efficiency and robustness compared to the benchmark key management scheme. A faster key transmission time between the two security domains is presented with the help of blockchain.

For group key management, probabilities are introduced into the key manager so that the system can decide how to organise key tree properly. A model of vehicle registration is also discussed. The handshake presents the batch rekeying process. Our registration steps combine the registration messages with safety beacon messages that decrease overhead in the network. This procedure acts as foundation to implement further key management schemes. The blockchain concept is used to improve key transportation efficiency. Crossing border activities are formed into transactions and arranged into block. Third party central authorities are set aside since the verification job is delivered by SM network. The simulations show that the time cost for transporting keys is much less than that of standard network structure.

References

[1] P. Papadimitratos, L. Buttyan, T. Holczer, et al., "Secure vehicular communication systems: design and architecture," *IEEE Communications Magazine*, vol. 46, no. 11, pp. 100–109, Nov. 2008. doi: 10.1109/MCOM.2008.4689252.
[2] Y. Cao; N. Wang; G. Kamel; and Y. J. Kim, "An electric vehicle charging management scheme based on publish/subscribe communication framework," *IEEE Systems Journal*, vol. PP, no. 99, pp. 1– 14, 2015. doi: 10.1109/JSYST.2015.2449893.
[3] H. Hartenstein and L. P. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *IEEE Communications Magazine*, vol. 46, no. 6, pp. 164– 171, Jun. 2008. doi: 10.1109/MCOM.2008.4539481.
[4] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, Jul. 2011. doi: 10.1109/JPROC.2011.2132790.
[5] *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Co-Operative Awareness Basic Service*, ETSI 102 637-2, 2010.
[6] *Dedicated Short Range Communications (DSRC) Message Set Dictionary*, SAE J2735, 2009.
[7] W. H. D. Ng, H. Cruickshank, and Z. Sun, "Scalable balanced batch rekeying for secure group communication," *Computers & Security*, vol. 25, no. 4, pp. 265–273, 2006. doi:10.1016/j.cose.2006.02.006.
[8] C. K. Wong, M. Gouda, and S. S. Lam, "Secure group communications using key graphs," *IEEE/ACM Transactions on Networking*, vol. 8, no. 1, pp. 16–30, Feb. 2000. doi: 10.1109/90.836475.
[9] *Logical Key Hierarchy Protocol*, RFC2026, Mar. 1999.
[10] X. S. Li, Y. R. Yang, M. G. Gouda, and S. S. Lam, "Batch rekeying for secure group communications," in *Proc. ACM 10th International Conference on World Wide Web*, New York, USA, 2001, pp. 525–534. doi: 10.1145/371920.372153
[11] O. Zakaria, A. A. Hashim, and W. H. Hassan. "An efficient scalable batch-rekeying scheme for secure multicast communication using multiple logical key trees," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 15, no. 10, pp. 124–129, 2015.
[12] 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 Networks*, vol. 11, no. 8, pp. 2724–2737, Nov. 2013. doi: 10.1016/j.adhoc.2013.05.009.
[13] K. Lu, Y. Qian, M. Guizani, and H. H. Chen, "A framework for a distributed key management scheme in heterogeneous wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 2, pp. 639–647, Feb. 2008. doi: 10.1109/TWC.2008.060603.
[14] Y. Sun, W. Trappe, and K. J. R. Liu, "A scalable multicast key management scheme for heterogeneous wireless networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 4, pp. 653– 666, Aug. 2004. doi: 10.1109/TNET.2004.833129.
[15] Y. Cao, Z. Sun, N. Wang, et al., "Geographic-based spray-and-relay (GSaR): an efficient routing scheme for DTNs," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1548–1564, Apr. 2015. doi: 10.1109/TVT.2014.2331395.
[16] X. B. Zhang, S. Lam, D. Y. Lee, and Y. R. Yang, "Protocol design for scalable and reliable group rekeying," *IEEE/ACM Transactions on Networking*, vol. 11, no. 6, pp. 908–922, Dec. 2003. doi: 10.1109/TNET.2003.820256.
[17] J. Daemen and V. Rijmen, *The Design of Rijndael: AES—the Advanced Encryption Standard*. Secaucus, USA: Springer-Verlag New York, 2002.
[18] S. Nakamoto. (2008, November). *Bitcoin: A peer-to-peer electronic cash system* [Online]. Available: https://bitcoin.org/bitcoin.pdf
[19] C. Decker and R. Wattenhofer, "Information propagation in the Bitcoin network," in *IEEE P2P 2013 Proceedings*, Trento, Italy, Sept. 2013, pp. 1–10. doi: 10.1109/P2P.2013.6688704.
[20] C. K. Wong and S. S. Lam, "Keystone: a group key management service," in *17th International Conference on Telecommunications*, Doha, Qatar, Apr. 2000.
[21] *IEEE Draft Standard for Wireless Access in Vehicular Environments (Wave)—Security Services for Applications and Management Messages*, IEEE P1609.2/D15, May 2012.
[22] D. Hankerson, A. J. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*. New York, USA: Springer-Verlag New York, 2004.
[23] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *International Journal of Information Security*, vol. 1, no. 1, pp. 36–63, Jan. 2001. doi: 10.1007/s102070100002.
[24] Y. H. Park, D. H. Je, M. H. Park, and S. W. Seo, "Efficient rekeying framework for secure multicast with diverse-subscription-period mobile users," *IEEE Transactions on Mobile Computing*, vol. 13, no. 4, pp. 783–796, Apr. 2014. doi: 10.1109/TMC.2013.40.
[25] M. R. Hustim and M. Isran. (2015, Dec 12). *The vehicle speed distribution on heterogeneous traffic: space mean speed analysis of light vehicles and motorcycles in makassar-indonesia* [Online]. Available: http://repository.unhas.ac.id/handle/123456789/16562
[26] W. D. Kelton and A. M. Law, *Simulation Modeling and Analysis*. Boston, USA: McGraw Hill Boston, 2000.
[27] A. M. Antonopoulos, *Mastering Bitcoin: Unlocking Digital Crypto-Currencies*. Sebastopol, USA: O'Reilly Media, Inc., 2014.
[28] R. C. Merkle, "A digital signature based on a conventional encryption function," *Advances in Cryptology—CRYPTO' 87*, vol. 293, no. 6, pp. 369–378, 1987. doi. 10.1007/3-540-48184-2_32.
[29] *DSRC Message Communication Minimum Performance Requirements: Basic Safety Message for Vehicle Safety Applications*, SAE Draft Std. J 2945, Jul. 2007.
[30] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved IVC analysis," in *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3– 15, Jan. 2011. doi: 10.1109/TMC.2010.133.
[31] A. Varga, "The OMNeT++ discrete event simulation system," in *Proc. Europe-*

an Simulation Multiconference (ESM' 2001), Prague, Czech Republic, Jun. 2001, pp. 65.
[32] W. Dai. (2015, July 7). *Crypto++ library 5.6.0* [Online]. Abailable: http:/www.cryptopp.com
[33] *Using Advanced Encryption Standard (AES) CCM Mode with IPsec Encapsulating Security Payload (ESP)*, RFC 4309, Dec. 2015.

Manuscript received: 2016-04-21

Biographies

LEI Ao (a.lei@surrey.ac.uk) received his BEng degree in communication engineering at Harbin Institute of Technology, China and University of Birmingham, UK, in 2013, and the MSc degree in communication engineering at the University of York, UK, in 2014. He is currently working toward the PhD degree in communication engineering in the Institute of Communication Systems at the University of Surrey, UK. His research interests include security and privacy for vehicular networks and privacy protection for location based services. He is currently involved with EU-funded PETRAS Project on security and privacy in smart vehicles and location based services.

Chibueze Ogah (c.anyigorogah@surrey.ac.uk) received the BSc (Hons) in computer science from the Ebonyi State University, Nigeria in 2005. He received the MSc degree (Distinction) in computer network technology from the University of Northumbria at Newcastle, UK in 2011. He is currently a PhD candidate at the Institute for Communication Systems, University of Surrey, UK. He has been a laboratory assistant and lecturer at the Computer Science Department of Ebonyi State University, Nigeria since September 2007 and February 2012. His research interests include security and privacy in vehicular networks, and Cisco routing protocols. He is currently involved with EU-funded PETRAS Project on privacy in smart vehicles.

Philip Asuquo (p.asuquo@surrey.ac.uk) received his Bachelor's degree in computer engineering from University of Uyo, Nigeria and MSc in computer network technology from Northumbria University, UK. He is currently working towards his PhD in electronic engineering at the University of Surrey, UK. His research interest includes cyber security of critical infrastructures, smart grid and smart homes, intelligent transport systems (ITS) and wireless sensor network security. He is currently involved with EU-funded PETRAS Project.

Haitham Cruickshank (h.Cruickshank@surrey.ac.uk) received a BSc degree in electrical engineering from the University of Baghdad, Iraq, in 1980, and MSc in telecommunications from the University of Surrey, UK and a PhD in control systems from Cranfield Institute of Technology, UK, in 1995. He is a senior lecturer at the Institute of Communication Systems (Formerly Centre for Communication Systems Research, CCSR), University of Surrey. He has worked there since January 1996 on several European research projects in the ACTS, ESPRIT, TEN-TELECOM, and IST programmes. His main research interests are network security, satellite network architectures, VoIP, and IP conferencing over satellites. He is a member of the Satellite and Space Communications Committee of the IEEE Communications Society, and is also a Chartered Electrical Engineer and IEE corporate member in the UK. He is active in the ETSI BSM and the IETF MSEC groups.

SUN Zhili (z.sun@surrey.ac.uk) received his BSc in mathematics from Nanjing University, China and PhD from the Department of Computing, Lancaster University, UK, in 1991. He is a professor at the Institute of Communication Systems (Formerly Centre for Communication Systems Research, CCSR), University of Surrey, UK. His research interests include wireless and sensor networks, satellite communications, mobile operating systems, traffic engineering, Internet protocols and architecture, quality of service, multicast, and security. He has been principal investigator and technical coordinator in a number of projects within the European Framework Program including the ESPRIT BISANTE project on evaluation of broadband traffic over satellite using simulation, the TEN-telecom VIPTEN project on QoS of IP telephony over satellite, the GEOCAST project on IP Multicast over satellites and ICEBERGS project on IP based Multimedia Conference over Satellite of IST, the SATELIFE project on Satellite Access Technologies on DVB-S and DVBRCS, and EuroNGI on next generation Internet.