A novel key management protocol for vehicular cloud security

Nayana Hegde*[1], Sunilkumar S. Manvi[2]
[1]ECE Department, Sri Krishna Institute of Technology, Bengaluru, India
[2]School of Computing and Information Technology, Reva University, Bengaluru, India
*Corresponding author, e-mail: nayana.srikanth@gmail.com[1], sunil.manvi@revainstitution.org[2]

Abstract

Vehicular cloud computing (VCC) is a new hybrid technology which has become an outstanding area of research. VCC combines salient features of cloud computing and wireless communication technology to help drivers in network connectivity, storage space availability and applications. VCC is formed by dynamic cloud formation by moving vehicles. Security plays an important role in VCC communication. Key management is one of the important tasks for security of VCC. This paper proposes a novel key management protocol for VCC security. Proposed scheme is based on Elliptical Curve Cryptography (ECC). The simulation results demonstrated that the proposed protocol is efficient compared to existing key management algorithms in terms of key generation time, memory usage and cpu utilization.

***Keywords**: ECDSA, key management, security analysis, vehicular cloud*

1. Introduction

Vehicular Cloud Computing (VCC) technology is a very new promising area of research. It uses internet to keep the nodes (vehicles) connected while they are moving [1]. VCC is a framework in which the vehicle user can use the computing power, services and infrastructure of VCC without investing in new resources [2, 3]. VCC is gaining popularity due to various applications like: vehicle maintenance, traffic management, road condition sharing, accident alerts at intersections, safety applications, intelligent parking management, planned evacuation, entertainment, data storage facility [4-6]. VCC needs strong security model for implementation because of the decentralized cloud environment, dynamic physical infrastructure, wireless communication and dissimilar operating system (OS) [7, 8]. Cryptographic techniques can be employed in VCC to achieve security. Key management is one of the important issues in cryptography. Challenging factors in key management are software complexity, cost and resource optimization. In VCC, key management is owned by Certification Authority (CA) and not with either user (vehicle) or cloud [9].

There are different approaches for key administration and we can broadly classify them into three categories as centralized, decentralized and distributed key management systems [10]. Centralized key management system is better due to following advantages. The single point of control for generating and distributing keys to users increases trust among cloud providers and cloud users. Rekeying cost is minimized when users frequently join and leave VCC. Evicted users will not be able to share their individual piece of information to regain access. Collusion attacks are prevented as distribution of keys is not shared among the Road Side Unit (RSU). With reference to recent research work in centralized key management area, for VCC and VANET, some of the research gaps are identified which are presented as follows: 1) Most of the centralized key management systems discussed for VANET are group key management systems. In vehicular cloud, change of the group is very frequent. So group leader taking charge of key production and distribution is not optimum in real time scenario. 2) Computational complexity will be more in cases of binary search and bloom filter algorithms employed for key distribution. 3) Rekeying increases time complexity and computational complexity.

In this paper we propose a novel key management protocol for VCC to facilitate Storage as a Service (SaaS) model. The protocol works in following steps: 1) CA generates individual unique key pair for each registered vehicle. 2) Secure and reliable key distribution protocol,

Received March 15, 2018; Revised December 2, 2018; Accepted January 25, 2019

based on Public Key Infrastructure (PKI) and digital certificates is used by CA. 3) Keys are securely stored at CA repository using key wrapping constructors/hash functions. 4) CA utilizes global revocation approach based on PKI Certificate Revocation List (CRL) to delete keys and certificates from compromised vehicles.

Our major contributions in this work are as follows. 1) Use ECDSA algorithm for key pair generation 2) Use secure communication protocol for dynamic key distribution 3) Keys stored with encryption. The structure of our paper is organized as follows: Section 2 presents related work on security of VCC. Section 3 presents the problem statement, assumptions. Section 4 gives details of protocol proposed. Section 5 deals with the implementation of key management protocol. Section 6 discusses the results. Section 7 concludes our contributions and directs for future research.

2. Related Work

In this section we summarize some of the related works. Authors in [11] have proposed a scheme based on ECC with signature hash function in WBAN. The scheme employs hash chain based key signature technique to achieve efficient, secure transmission. The work given in [12] presents efficient conditional privacy preservation (ECPP) protocol in vehicular adhoc networks to address the issue on anonymous authentication for safety messages with traceability. Authors in [13] propose a dual authentication based security management scheme for VANET. They rely on tamper proof device to store keys and use fingerprint for user identification. Work in [14] proposed authentication protocol for verification of users in cloud computing. Authors have focused on interaction between cloud and smart phone users in their proposed work. Approach in [15] explains an implementation of SHA-1 and ECDSA for secure communication in VANET.

The results show that the algorithm don’t increase any delay in time for message transmission. Authors in [16] proposed an ECDH based efficient key agreement protocol to secure user identity for lower computing environments. Authors in [17] have analysed ECC based security schemes for cloud based applications in comparison to RSA based schemes. Results show that ECC based schemes are more efficient. Authors have compared the results for secured and non-secured communication [18]. Various numbers of mobile nodes and digital signatures are considered for simulation. Approach in [19] proposed an efficient protocol by analysing two variants of ECC-based wireless authentication protocol, namely, Aydos-Savas-Koc’s wireless authentication protocol (ASK-WAP) and user authentication protocol (UAP) from various security aspects and communication concerns. Work given in [20] explained a reconfigurable hardware based on prime fields. Bit length up to 256 is considered. Complete ECDSA signature processing system is used. Work in [21] designed a set of network and information security mechanisms in line with the requirements of confidentiality, authentication, non-repudiation, conditional anonymity, and conditional intractability. Work in [22] explains a centralized key management scheme based on Chinese remainder theorem to support secure communication. Work in [23] has proposed Elgamal algorithm based Group Key Management scheme. Based on these observations, we propose an ECDSA based a novel key management protocol for VCC to facilitate SaaS, considering the efficiency of key generation, distribution, storage, and revocation.

3. System Model and Problem Definition

In this section, we discuss environment, assumptions and problem statement.

3.1. Environment

We consider VCC architecture in which vehicles are willing to share their unutilized resources. Vehicles should be in close proximity, should have relatively small speed difference between vehicles and travel in same direction. A broker is chosen randomly among the users. Broker owns responsibility of creation, maintenance and deletion of vehicular cloud. Permission to host VCC is obtained from CA by broker and he invites other users join VCC. Cloud is automatically formed by combining and resources form available number of vehicles like sensors, storage space and computational power. Vehicles involved in VCC have a basic trust relationship between them. Figure 1 illustrates the system model, which consists of network entities like: CA, RSU, and On Board Unit (OBU) equipped moving vehicle. All entities communicate with Dedicated Short Range Communication (DSRC) identified as IEEE 802.11p.

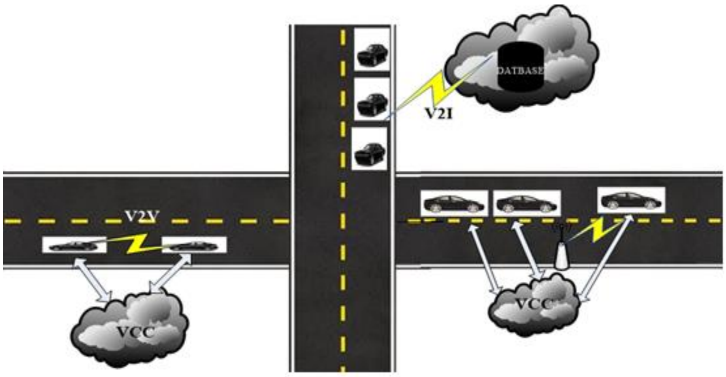

Figure 1. System Model of Vehicular Cloud

3.2. Assumptions

The following assumptions are made in deployment of the proposed protocol:

a. All vehicles are equipped with wireless communication devices, Global Positioning System (GPS), digital maps and optional sensors for reporting vehicle conditions.
b. CA is trusted by all entities involved in VCC, has powerful firewalls and is not compromised.
c. Every RSU deployed on the road or highway has a unique id (RSU_{ID}) and is registered to a CA.
d. VCC users approach physically to CA, provide all essential data like vehicle id, name, phone number, email id, unique identity number and get registered with CA.
e. OBU, RSU and CA communicate to each other via high bandwidth, low bit error and low delay links [13, 24, 25].
f. Data messages are encrypted at OBU before uploading it to vehicular cloud. This will reduce computation burden to VCC server.
g. Decryption process of data messages is carried out at receivers OBU.

VCC stores the encrypted data. The system thus separates storage of keys, encryption/decryption schemes from data which provides good security.

3.3. Problem Statement

The objective of the work is to design and develop an efficient key management protocol with resource optimization to secure communication in VCC environment to facilitate storage as a service model. The objective is achieved by 1) Key generation 2) Secure key distribution 3) Encrypted key storage 4) Compromised key revocation.

4. Proposed Scheme

The proposed key management protocol consists of four parts: key generation, key distribution, key storage and revocation. Table 1 illustrates notations used in this paper.

Table 1. Notations and Descriptions

Notation	Description	Notation	Description
CA	Certificate Authority	APc	Authorization permission code
n	Large random prime number	t	Time stamp of communication
CA_{ID}	Certificate authority identity	T_0	Time stamp of registration with CA
CA_{PU}	Public key of CA	V_{IDi}	Vehicle identifier
CA_{PR}	Secrete seed used by CA, 1 < CAPR < n	||	Concatenation operation
Loc	Location identification of VCC user	RSU_{IDj}	RSU id
G	A base point over $E(F_p)$	RSU_{PU}	Public key of RSU
Hc	Hash code	RSU_{PR}	Private key of RSU
V_{PR}	Vehicles private key	E_E	Re-encryption
V_{PB}	Vehicles private key	B_{ID}	Vehicular cloud broker id

4.1. Key Management

This section discusses various phases like: generation, distribution, storage and revocation for cryptographic key used for VCC security. Before key generation for any entity like vehicle user or RSU, registration with CA is necessary. Registration process is carried out by Registration Authority (RA), which is part of CA.

4.2. Key generation

During registration procedure user interacts with registration authority. Initially vehicle user submits all required data to registration authority. CA verifies complete information of user requested for registration. Registration process results in binding keying material to information associated to an entity. This binding is a cryptographic process. CA generates unique id for vehicles and RSU (V_{ID})/(RSU_{ID}) for vehicle or RSU and the same is communicated. All the communication and verification needs this number. Public and private keys for vehicle/RSU are generated and the same are mapped to that particular vehicle id (V_{ID})/(RSU_{ID}). Algorithm 1 describes key generation procedure. CA chooses the system secret V_{PR}, where $1<V_{PR}<n$, and computes {$V_{PB}=(V_{PR})$ G}. CA maps key pair generated with unique identification number V_{ID}. After successful registration and key generation, CA provides user with a smart card which contains private key, CA certificate. CA certificate contains public key of CA. All these details are stored in vehicles OBU using smart card.

```
Algorithm1--- Key Generation
//Inputs: Domain parameters, $V_{ID}$, $RSU_{ID}$
//Outputs: Output: Public key $V_{PBi}$/$RSU_{PUj}$ and private key $V_{PRi}$/$RSU_{PRj}$
    1. loop
    2. CA receives request for key generation by user.
    3. Registration status of user is verified by CA by contacting RA.
    4. If (not registered )
    5. send message to user to register at RA
    6.  return // If  not registered, stop
    7. end if
    8. if( user_type==Vehicle user)
    9. then check for authenticity and obtain $V_{ID}$ from RA
    10. CA generates private key $V_{PRi}$ --using $1<V_{PR}<n$
    11. CA generates public key $V_{PBi}$-- using $V_{PBi}=( V_{PRi})G$
    12. else if (user_type==RSU)
    13. then generate key pairs $RSU_{PUj\ and}$ $RSU_{PRj}$
    14. go to loop
    15. close
```

4.3. Key distribution

A certificate binds an identity to a public key. CA generates a public/private key pair V_{PBi}/V_{PRi} and transmits a message to user consisting of public key V_{PB}, time of registration (T_0) and a unique identification number V_{IDi}. Before sending signed certificate, CA and user should set up a secure communication channel. Key distribution process is explained in algorithm 2. User requests for public key, sending an encrypted message. Message is encrypted using public key of CA which is available with authenticated user. Message consists of user's unique id and time of registration with RA. This request is decrypted by CA using its private key. CA verifies the user unique id; registration details. Post verification, a response message is sent to user. This message is encrypted with CA private key.

The response message consists time of verification. User now sends a request for signed certificate from CA. User can send this certificate to VCC broker for secure communication. Certificate issued by CA to users will have limited expiry time and a unique serial number. This helps to protect from malicious users storing certificate illegally. Public key of a user is available to other users with this certificate. Received certificate will have timestamp and expiry details. The same certificate cannot be cached by other users for malpractice or by the same user again.

```
Algorithm 2--Key Distribution
//Inputs: Request
//Outputs: Certificate/Reject
```

```
1. loop
2. User sends encrypted request message to CA=> Req{$V_{ID}$||$T_0$}
3. E[$CA_{PU,}$ Req{$V_{ID}$||$T_0$}||t]
4. CA verifies the message and confirms the identity of the user by response.
5. E[$CA_{PR,}$ Res{t}]
6. User requests for signed certificate after successful authentication
7. CA sends signed certificate=> E[$CA_{PR}$, ($V_{IDi}$||$T_0$||$V_{PBi}$)]
8. User decrypts certificate using public key of CA=> D[Cert] = D[E[$CA_{PU}$,($V_{IDi}$||$T_0$||$V_{PBi}$)]
9. User send acknowledgement to CA =>E[$CA_{PU}$,(Ackj[$V_{IDi}$])
10. CA stores the Ack with Cert and its expiry time=> D[E[$CA_{PR}$, (Ack||$V_{IDi}$)]]
11. Close
```

4.4. Key storage

CA stores details of VCC user and generated keys securely as follows in different databases:

a. *Encryption Key Database (EKDB*): All public and private keys of users, RSUs and CA are stored in this database. Key wrap constructors are used to encrypt the keys. When keys are updated periodically they are encrypted again. Keys are divided in blocks of n 64-bit blocks like P_1, P_2, P_n. Then keys are encrypted cipher text, (n+1) 64 bit values C_1, C_2, ..., C_n. Key unwrapping is obtained by (n+1) 64 bit blocks of cipher text consisting of previously wrapped key K. It gives plaintext n 64 bit values P_1, P_2,… P_n.

b. *Vehicle User Database (VUDB):* Details of all users are available in this database. Vehicle details, owner details and date of registration etc. are encrypted and stored.

c. *Key Revocation Database (KRDB):* Once the key of any vehicle are compromised, keys are revoked from user. Revoked key list is maintained in this database. Key storage process in explained in algorithm 3.

```
Algorithm 3--- Key Storage
//Inputs: key pair
//Outputs: encrypted keys
    1. loop
    2. All public and private keys are divided into blocks of 64 bit , $P_1$, $P_2$ ..$P_n$
    3. Keys are encrypted to cipher text of 64 –Kb $C_1$,$C_2$…$C_n$
    4. Encrypted keys are stored in Encryption Key Database(EKDB)
    5. Vehicle details, owner details and date of registration etc. are encrypted
    6. Encrypted personal details are stored in Vehicle User Database(VUDB)
    7. Keys are revoked from compromised users.
    8. Revoked key list is maintained in Key Revocation Database (KRDB)
    9. Close
```

4.5. Key revocation

On proving the identity disclosure, a CA may decide revoke the permission for the vehicle in order to prevent future attacks on the vehicular cloud. CA must maintain a list consisting of all revoked but not expired certificates, known as the Certificate Revocation List (CRL). Each entry consists of the serial number of a certificate and revocation date for that certificate. Because serial numbers are unique within CA, the serial number is sufficient to identify the certificate. In revocation process, CA first develops a message using CA_{ID}, V_{IDi}, RSU_{IDj}, vehicles public key, timestamp and encrypts this message using public key of CA. User after receiving this revocation message deletes all the keys sends an acknowledgement to CA. Key revocation procedure is explained in algorithm 4.

```
Algorithm 4--Key Revocation
//Inputs: Revocation message
//Outputs: CRL
    1. loop
    2. If (vehicle compromised)
    3. CA constructs revocation message=>E[CAPU ,(CAID ||RSUIDj ||VAIDi ||VPB||t )]
    4. CA sends revocation message to RSU,Vehicle
```

5. RSU decrypts the message and appends it with new timestamp and re-encrypts it.
6. $E_E[CA_{PU}\ ,(CA_{ID}\ ||RSU_{IDj}\ ||V_{IDi}\ ||V_{PB}||t\ )]$
7. RSU updates revocation list and sends message to vehicles in cloud.
8. Vehicle deletes all keys and sends acknowledge to CA and RSU.
9. CA updates CRL.
10. close

5. Implementation

In this section we explain the implementation of the work. We have used Java programming for ECDSA based key management protocol implementation. In the Java architecture, the Security API is one of the main interfaces of the language. It has been implemented on a computer with Intel core i5-4200 M CPU 2.50 GHz and 4GB RAM. We focused on the ECDSA digital signatures with key length 160,192,224, 256 bits. The obtained results have been plotted for memory consumption and time performance.

6. Results and Discussion

In this section we analyse the computational complexity, time complexity and space complexity of our proposed novel key management protocol. ECDSA signature key generation is compared with DSA, RSA and ElGamal cryptographic technique based key generation schemes.

6.1. Computational Complexity

DSA and RSA key generation algorithms require prime number testing, which is not cost effective. Besides, prime tests are probabilistic, which results into varying execution time. ECC cryptography based key generation algorithms involve only scalar point multiplication. In Figure 2 CPU utilization for key generation of different cryptographic algorithm are compared. Results show that ECDSA scheme is computationally more effective as greater security is achieved with smaller key size in comparison to RSA, Elgamal and DSA based key generation algorithms.

6.2. Memory Space Complexity

Different types of public key algorithms are compared for key generation. They are DSA, RSA, and ElGamal. The proposed protocol consumes less memory for key generation. Figure 3 shows the memory usage of the ECDSA key generation.

6.3. Time Complexity

Major factor for the key generation is the time consumption. Efficient system should consume less time for key generation. Table 2 shows the time for key generation for different key sizes. The time of generation is represented in seconds. The ECDSA can create keys in superior speed compared to other cryptographic algorithms of same key lengths. We can see that key generation time for RSA algorithm for lower key bit size is less. As key size increases, time taken for the key generation of increases. DSA and Elgamal algorithm have comparatively less time to RSA algorithm. ECDSA algorithm has less time compared to all algorithms. This results in faster processing times, and lower demands on memory and bandwidth which is well suited for VCC environment.

Table 2. Key Generation Time Comparison

Key Size(in bits)	192	224	256	384	512
RSA(time in sec)	1.262	5.976	10.877	237.715	1513.897
DSA time in sec)	0.067	0.1920	0.427	0.591	0.71
ECDSA(time in sec)	0.08	0.079	0.081	0.082	0.0894
ElGamal(time in sec)	7.021	7.461	7.5260	8.1662	11.544

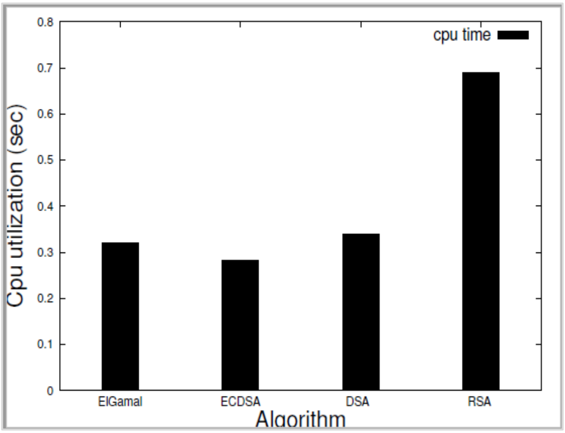

Figure 2. CPU Utilization Time

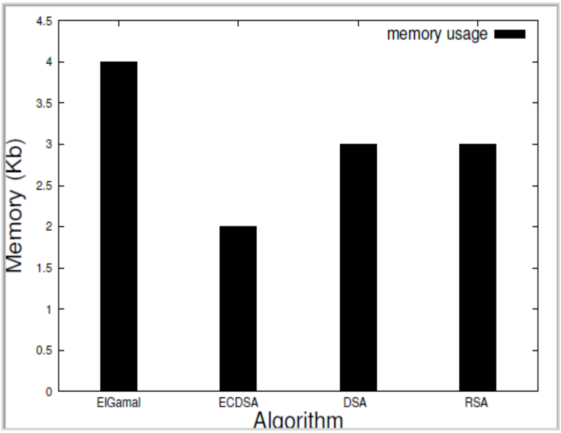

Figure 3. Memory Utilization

6.4. Performance Analysis

The graphical results shown in Figure 4 are used to compare the key computation time at CA for our proposed key management protocol with the existing methods. It compares the results obtained from our proposed work with DAKM [13], CRGK [22], and NTRU [23]. From Figure 4, it is observed that when the key is 512 bits, the group key computation time of CA is found to be 14 msec in our proposed approach, which is better in comparison with the other existing schemes.

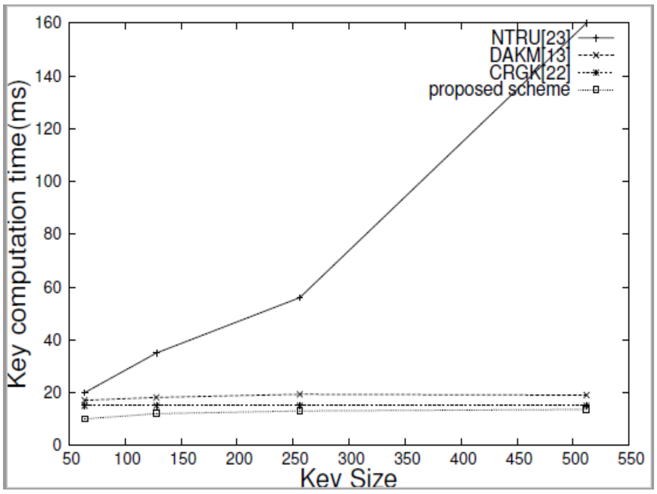

Figure 4. Key computation time comparison

Table 3 shows computation and storage complexities. Among the existing key management works that we have considered, the Number Theory Research Unit (NTRU) [23] uses convolution product for calculating multiplication for key generation. It includes addition (A), division (D) and finding inverse using extended Euclidian Algorithm (EEA). It takes more computations. CRGK [22] and DAKM [13] perform only 1 subtraction(S) or addition (A) operation. Proposed key generation based on ECDSA is the simplest with performs scalar multiplication (M) and squaring (Sq). Our scheme being centralized does not need key generation every time a new user joins or leaves the group. This provides a less calculations among the scheme compared. In all other schemes key secrete is owned by one of the group member, but in proposed scheme key secrete is maintained by CA. N is total number of users.

Proposed scheme is very secure in distribution of the keys. Every time a secure channel is set up between CA and user for key distribution. Key storage is very simple and easy in the proposed scheme because they are encrypted with block ciphers and stored at CA secure data base. Therefore only N (number of users) keys will not have multiple storage complexity in proposed method. Communication complexity is more in proposed scheme compared to CRGK [22] and DAKM [13] as every user needs individual, secure key distribution.

Table 3. Computation, Storage and Communication Complexities

Parameter	CRGK [22]	DAKM [13]	NTRU [23]	Proposed scheme
Computations at CA	(A or S)	(A or S)	(A+D+EEA)	(M +Sq)
Storage Complexity	4N+3	4N+3	2N+7	N
Communication Complexity	1 broadcast	1 broadcast	N broadcast	N broadcast

7. Conclusion and Future Work

In this paper we presented a novel key management system for VCC. Focus is given to key generation, distribution, storage and revocation. Extensive simulations are presented to evaluate the performance of the proposed techniques. Our proposed protocol provides greater security and more efficient performance than the existing key management techniques. Algorithm design supports dynamic nature of VCC. Scheme improved response time without affecting the communication time. The future work aims to measure the parameters like maximum storage in VCC, storage time available, cost per storage etc.in VCC environment.

References

[1] G Yan, O Stephan, M. Weigle. Security challenges in vehicular cloud computing.*IEEE Transactions on Intelligent Transportation Systems*. 2013; 14(1): 284–294.

[2] Whaiduzzaman, M Sookhak, M Gani, A B. Rajkumar. A survey on vehicular cloud computing. *Journal of Network and ComputerApplication*.2014; 40(1): 325–344.

[3] S Arif, S Olariu, J. Wang, G Yan, W Yang, Khalil. Data center at the airport: reasoning about time-dependent parking lot occupancy.*IEEE Transactions on Parallel Distributed Systems*. 2012; 23(11): 2067–2080.

[4] M Eltoweissy, S Olariu, M Younis.Towards autonomous vehicular clouds. *AdhocNets*. 2011; 11(9): 1–11.

[5] Vrun chand H, Karthikeyan J.Survey on the Role of IoT in Intelligent Transportation System. *Indonesian Journal of Electrical Engineering and Computer Science*. 2018; 11(3): 936-941

[6] S.Olariu, I.Khalil, M. Abuelela. Taking vanet to the clouds. *I J Pervasive Computer Communication*.2011; 7(1): 7–21.

[7] Haifaa Jssim Muhasin, Rodzia Atan, Marzanah A.Jabar, Salfarina Abdullah. The factors affecting On Managing Sensitive Data in Cloud Computing. *Indonesian Journal of Electrical Engineering and Computer Science*. 2018; 1(3): 01-02

[8] S Olariu, T Hristov, G Yan. The next paradigm shift: From vehicular networks to vehicular clouds. *Wiley and Sons*. 2012: 645–703.

[9] R Barskar, M Chawla. A survey on efficient group key management schemes in wireless networks.*Indian Journal of Science andTechnology*.2016; 9(14): 1-16

[10] B Cui, ZiyueWang, B Zhao, X Liang. Enhanced key management protocols for wireless sensor networks.*Hindawi Publishing CorporationMobile Information Systems*, 2015: 1–10.

[11] G Sridevi Devasena, S.Kanmani. Robust Security for Health Information by ECC with Signature Hash Function in WBAN. *IJEECS*. 2018; 11(1): 256-262.

[12] R Lu, X Lin, H Zhu, P-H. Ho, Xuemin.ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications. *IEEE Conference on Computer Communications Phoenix, USA*, 2008: 1903–1911.

[13] P Vijayakumar, M Azees, A kannan, L J. Deborah. Dual authentication and key management techniques for secure data transmissionin vehicular adhoc networks. *IEEE Transactions on IntelligentTransportation Systems*.2015; 17(4): 1015–1028.

[14] H-T Wu,G-J. Horng. Vehicular cloud network and information security mechanisms. *International Conference on Network and CommunicationTechnologies Taiwa;* 2016:196–199.

[15] S Nacy, T Oh,J Leone. Implementation of sha-1 and ecdsafor vehicular ad-hoc network using ns-3. *Conference on Research inInformation Technology Orlando Florida, USA*; 2013: 83–88.

[16] R-C. Wang,W-S.Juang, C-L. Lei. Provably secure and efficient identification and key agreement protocol with user anonymity. *Journal of Computer and System Sciences*. 2011; 77(4): 790-798.

[17] A Tripathi, P Yadav. Enhancing security of cloud computingusing elliptic curve cryptography. *International Journal of Computer Applications*. 2012; 57(1): 26–30.

[18] J Durech, M Frankova, P Holecko, . Bubenikova. Modelling of security principles within car-to-car communications in modern cooperative inteligent transportation systems. *Information and safety related systems*. 2016; 14(1): 40–57.

[19] B Glas, O Sander, V Stuckert, K D.Muller-Glaser, J Becker. Prime field ecdsa signature processing for reconfigurable embedded systems. *International Journal of Reconfigurable Computing*. 2011; 1(5): 1–12.

[20] V Waziri, O Adebayo, H Danladi, A Isah, A Magaji, M B.Abdullahi. Network security in cloud computing with elliptic curve cryptography. *Network and Communication Technologies.* 2013; 2(2): 43–58.

[21] L Tripathy, N R. Paul. An efficient and secure key managementscheme for hierarchical access control based on ecc. *International Journal Communication and Network Security*.2011; 1(2): 50–55

[22] P.Vijayakumar, S Bose, A. Kannan. Chinese remainder theorem based centralized group key management for secure multicast communication. *IET Inf. Security*. 2014; 8(3): 179–187.

[23] X Lv, H Li,B Wang.Group key agreement for secure group communication in dynamic peer systems. *J. Parallel Distributed Computation*. 2012; 72(10): 1195–1200.

[24] A Das, D Roychoudhury, D Bhattacharya, R Srinivasan, R Shorey. Thomas. Authentication schemes for vanets: a survey. *International Journal of Vehicle Information and Communication Systems*. 2013; 3(1): 48-55.

[25] Euisin Lee, Eun-Kyu Lee, Mario Gerla. Vehicular Cloud Networking: Architecture and design principles. *IEEE Communications Magazine.* 2014*;*52(2): 148-155