Distributed Key Management Framework in Vehicular Adhoc Networks

[a]Dr. Nalini N., Prof and HOD of CS&E, NMIT, Bangalore.
[b]Chandrika C. N., PG Student, NMIT, Bangalore.

Abstract : The purpose of this work is to study how to provide security in vehicular adhoc networks using distributed key management framework which is based on group signature to provision privacy . Distributed key management is expected to facilitate the revocation of malicious vehicles, maintenance of the system, and heterogeneous security policies, compared with the centralized key management assumed by the existing group signature schemes. In this framework, each road side unit (RSU) acts as the key distributor for the group, where a new issue incurred is that the semi-trust RSUs may be compromised. Thus, the security protocols are developed for the scheme which is able to detect compromised RSUs and their colluding malicious vehicles. The distributed key management framework also addresses the issue of large computation overhead due to the group signature implementation. A practical cooperative message authentication protocol is thus proposed to alleviate the verification burden, where each vehicle just needs to verify a small amount of messages.

Introduction : THE Vehicular ad hoc networks (VANETs) have attracted a lot of attentions due to their interesting and promising functionalities including vehicular safety, traffic congestion avoidance, and location based services [1]. In this paper, the main focus is on safety driving application, where each vehicle periodically broadcasts messages including its current position, direction and velocity, as well as road information. Privacy is an important issue in VANETs [2]. As the wireless communication channel is a shared medium, exchanging messages without any security protection over the air can easily leak the information that users may want to keep private. Pseudonym based schemes [3]–[5] have been proposed to preserve the location privacy of vehicles. However, those schemes require the vehicles to store a large number of pseudonyms and certifications, and do not support some important secure functionality such as authentication and integrity.

The *group signature* [6] is a promising security scheme to provide privacy in VANETs. But many of the existing group signature schemes in VANETs [7]–[9] are based on *centralized key management* which preloads keys to vehicles off-line. The centralized key management has some disadvantages. For instance, the system maintenance is not flexible. Another issue regarding the centralized key management is that many existing schemes assume a tamper-proof device [1] being installed in each vehicle. The tamper-proof device normally costs several thousand dollars, such as IBM 4764 card [10]. The framework to be developed in this paper does not require the expensive tamper-proof device. In this paper, a secure *distributed key management* framework is proposed and developed. According to this framework, the road side units (RSUs) [11] are responsible for secure group private keys distribution in a localized manner. When a vehicle approaches an RSU, it gets the group private key from the RSU dynamically. All vehicles which get the group private key from the same RSU form a *group*. A new issue induced by the distributed key management framework is that compromised RSUs may misbehave in the key distribution procedure. For example, a compromised RSU may deliver other vehicles' group private keys to its accomplice. Then, the accomplice can send messages under the name of other vehicles. Therefore, we develop security protocols for the distributed key management framework, which are capable of detecting the compromised RSUs and their collusion with the malicious vehicles if any. Computation overhead is another critical issue in VANETs. To avoid this cooperative message authentication protocol (CMAP) is developed.

System Model : In this paper entities are classified into three categories: authorities, road side infrastructure, and nodes.

Authorities are responsible for key generation and malicious vehicle judgment. Authorities have powerful firewalls and other security protections. Therefore, they have the highest security level. We assume that they can not be compromised.

Road side infrastructure consists of RSUs deployed at the road sides which are in charge of key management in this framework. Traffic lights or road signs can be used as RSUs after renovation. RSUs communicate with authorities through wired network. We assume a trusted platform module is equipped in each RSU. It can resist software attacks but not sophisticated hardware tampering. The cost of a trusted platform module is only a few tens of dollars which is affordable [1]. RSUs are semi-trust with the medium security level [5].

Nodes are ordinary vehicles on the road that can communicate with each other and RSUs through radio. Here an assumption is that each vehicle is equipped with a GPS receiver using DGPS [12] with an accuracy on the order of centimeters and an on board unit (OBU) which is in charge of all communication and computation tasks. Nodes have the lowest security level.

Group Signature Based Privacy System

In this distributed key management framework, the communications can be divided into the *key distribution phase* and the *regular broadcast phase*. Vehicles get keys dynamically in the key distribution phase and then start to broadcast their geographic and road condition messages periodically in the regular broadcast phase. The group signature scheme is used for privacy provision. With group signature, members of a group sign messages under the name of the group. In a group, there are one group public key and many corresponding group private keys. A message that is signed by any group private keys can be verified with the unique group public key, and the signer's identifier will not be revealed. However, authorities hold a tracing key which can be used to retrieve the group private key from the signature[13]. If one group private key is assigned to only one user, the signer can be identified after authorities get its group private key.

DISTRIBUTED KEY MANAGEMENT

A. Short Group Signature : In short group signature, one group public key is associated with multiple group private keys. The members of a group sign messages under the name of the group. A message that is signed by any group private keys is verified with the unique group public key. For key generation short group signature is adopted because it has smaller communication overhead than other group signature schemes[13]. Meanwhile, in the short group signature protocol, there is a group private key generator which can be assigned to key distributors without revealing other secrets. The existence of the generator makes the third party possible to be key distributors. Another attractive feature of the short group signature is that it has a tracing key which can retrieve group private keys from signatures. The short group signature works as following:

Tracing Key : To find the dispute vehicle the tracing key is used. To generate tracing key first generate random number and check whether this random number is present in the array if it is present then again generate random number otherwise add in the array. Then check for the size, if the key size is 7 bytes then exit otherwise again generate random numbers.

Private Key : To generate private key first generate random number and check whether this random number is present in the array if it is present then again generate random number otherwise add in the array. Then check for the size, if the key size is 30 bytes then exit otherwise again generate random numbers

Public Key : To generate public keys set SequenceNumber = 000 and generate random 256 chars and store in "RandomChar" char array. Then SequenceKey = "RSU0" + SequenceNumber. If RandomChar[TracingKey[i]] = SequenceKey[i] then increment SequenceNumber. Make this "RandomChar" to string, this is "public key"

Identity Keys : Both RSU's and vehicles have their own pair of I-keys(Identity keys) which are unique and these keys are helpful to find an malicious vehicle in registration phase. The pseudocode to generate I-keys is shown below:

string[3] $V_{ipublicKey}$, string[3] $V_{iprivateKey}$, string[2] $R_{ipublicKey}$, string[2] $R_{iprivateKey}$, int BitStrength = 256 * 2;

```
public void Ikeys_Generate()
    { Int i=0;for (i = 0; i < 3; i++)
      {RSACryptoServiceProvider RSAProvider = new RSACryptoServiceProvider(BitStrength+i*8);
```

---------- }

for (i = 4; i < 6; i++) {RSACryptoServiceProvider RSAProvider = new RSACryptoServiceProvider(BitStrength+i*8);

---------- }}

Encryption and Decryption: In the reference paper the author uses elliptical curve integrated encryption scheme (ECIES) as the encryption protocol. But in our framework during encryption each vehicle have GPS value, road condition(RC) value and we are assume that value is always true and traffic status(TC) value and is Boolean i.e. true or false. Along with that for tracing the malicious vehicle and RSU's instead of SHA1 we are using MD5 hashing technique because SHA1 take 132 bytes where MD5 take only 16 bytes. In encryption first vehicle has to register with RSU's and have to except the Group keys if RSU is not revoked and then they have to do encrypt publickey[privatekey[i]]=message[i]. In decryption the vehicle first have to receive broadcasted message and then they have to decrypt using the function Decrypted Message+=Encrypted Message[privatekey[i]]OString().

Compared with existing schemes which preload keys into the vehicle off-line, the proposed key distribution framework has the following advantages:

(1) The revocation is more efficient. In this scheme, the revocation list is stored in RSUs. However, in preload schemes, revocation list has to be transmitted to every vehicle through wireless channels. Due to the large number of vehicles, the revocation list must be changed quickly. Meanwhile, both adding and deleting an item in the revocation list that distributes in so many vehicles is resource and time consuming.

(2) The system maintenance is easier and more flexible. The number of vehicles that are affected by group-key updating is much smaller than that in the preload scheme.

(3) Heterogeneous security policies can be implemented in this scheme. While, in preload schemes, the policy is difficult to be changed after it is deployed.

B. Secure Key Distribution Protocol Design

In this section, a procedure to detect compromised RSUs and their accomplices which is a brand new security issue induced by the distributed key management framework is implemented. A misbehaved RSU will let authorities fail to identify malicious vehicles[13]. The proposed protocol allows vehicles to be authenticated with their real identifier under protection and guarantees authorities to find compromised RSUs and identities of malicious vehicles if there is a dispute. The proposed protocol defines message types in registration, messages broadcasting and accusation. Authorities make decisions according to the registration information that vehicles provide. Hereby, the registration procedure is the most important part. Here assumption is that each vehicle and RSU is preloaded with a global, long term public/private key pair with key size of 224 bits and a corresponding certificate of the public key signed by the certification authority (CA). A pair of identity keys (I-keys) are defined . The group public key and group private keys are local, short term keys in this scheme and they called as group keys (G-keys). Both I-keys and G-keys are unique. Thus they are considered as identifiers of vehicles and RSUs. CA's public key size is 256 bits. Furthermore, a hash function h(x), such as MD5, is known by authorities, RSUs and all vehicles.

1. Registration: The procedure and workflow of registration is shown in Figure 1[13] and its notations are shown in Table 1[13]..

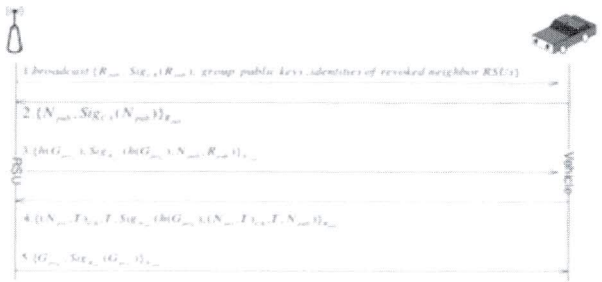

Figure 1 : Registration Message Flow

Table 1 : Notation and Description

Notations	Descriptions
R_{pub} / R_{pri}	RSU's public / private key pari (I-key)
N_{pub} / N_{pri}	Node (Vehicle)'s Public / Private key pair (I-key)
Sig_A (M)	Signature of message M signed by A's private key
$(M)_k$	Message M is encrypted by k or k's public key
$Gpub_k$ / $Gpri_k$	Group public / private key pair (G-key) for user k
T	Timestamp
h(.)	A one-way hash function such as SHA - I

Message 1 : RSUs broadcast I-public keys, G-public keys of themselves and their neighbor RSUs with certificates and identities of revoked RSUs in their neighborhoods regularly. Authorities employ benign RSUs around compromised RSUs to implement revocation by regular broadcasting those compromised RSUs' identities.

Message 2 : When a vehicle detects the hello message, it starts registration by sending its I-public key and the certificate to the RSU if the RSU is not revoked. Normally, a public key should not be encrypted. However, in our system model, each vehicle's I-public key is unique, so it is also an identifier of the vehicle. We encrypt it to protect vehicle's privacy.

Message 3 : The RSU sends the hash value of the G-private key which plans to be assigned to the vehicle and the signature of the hash value, vehicle's I-public key and RSU's I-public key to the vehicle. RSU's I-public key is also unique. The vehicle can identify the RSU's legitimacy after it verifies this message because the RSU uses its I-private key in the message.

Message 4 : The vehicle encrypts its N pri and the timestamp by using authorities' public key. Then, it sends the encryption data with the timestamp and the signature of corresponding information to the RSU. The encryption of its N and the timestamp is a commitment. We will use it to detect illegitimate users. Meanwhile, the signature signed by the vehicle binds vehicle's information pri and the assigned G-private key. Then, the RSU cannot remap them because the RSU does not have vehicle's I-private key.

Message 5 : The RSU sends the G-private key to the vehicle. The vehicle finishes registration procedure after it gets a valid G-private key.

2. Messages Broadcasting : Vehicles can broadcast messages under the name of the group after they get G-private keys from the RSU. In the broadcast message format, the "Grp ID" is the group ID which is used to identify a group. We add a hash value of vehicle's I-private key and the timestamp in the message. The vehicle signs the first five items in this message using the vehicle's G-private key, resulting in the signature item.

3. Accusation : When a vehicle finds that other vehicles send false messages, it will report to authorities. For example, a vehicle may maliciously detour traffic by claiming a traffic jam at a certain place but there is not in fact. Other vehicles at that place will report such claim as a false message. "Grp ID" is the accuser's group identifier.

The "Msg." field copies the whole message that the accusor considers false. "h(N ,T)" is the hash value of accuser's I-private key and the timestamp. The accuser signs the first six items in this message by using its G-private key. The entire message should be encrypted by CA's public key so that the accusation messages cannot be read by others. pri After receiving an accusation, authorities verify the signature in the accusation message by using G . Then, authorities perform key retrieve operations to get the accuser's and the accused's G-private keys[13]. Where after, authorities contact RSUs which assign G-private keys to the accuser and the accused according to group IDs. RSUs will send corresponding information back to authorities after they receive the requests from authorities. After that, authorities will calculate accuser's and accused's h(N pri pub ,T) by using vehicles' I-private keys and timestamps which are obtained from the accusation message and the broadcast message respectively. If the value that authorities calculate is the same with the value they get from the report, the user will be considered as legitimate. If both of them are authorized users, authorities will start the evaluation mechanism to decide which user tells the truth. To show how efficient our framework compared to many cryptographic algorithm we have implemented Diffie-Hellman key exchange algorithm. In other cryptographic algorithm including Diffie-Hellman if any nodes are compromised they exchange wrong information to all nodes. Therefore in our framework in every step authenticity is checked by encrypting and decrypting the message. The Diffie-Hellman algorithm is as follows:

Diffie-Hellman Key Exchange Algorithm

Step 1 : Let "q" be prime number and "a" is "a<q" and it is primitive root of "q".

Step 2 : User "A" key generation, Select X_A such that $X_A<q$, Calculate Y_A such that $Y_A=a^{X_A} \bmod q$

Step 3 : User "B" key generation, Select X_B such that $X_B<q$, Calculate Y_B such that $Y_B=a^{X_B} \bmod q$

Step 4 : Generation of Secret key by user A, $K=(Y_B)^{X_A} \bmod q$

Step 5 : Generation of Secret key by user B, $K=(Y_A)^{X_B} \bmod q$

Disadvantage of Diffie-Hellman Algorithm : For two users who want to share a secret, we can use the original algorithm. But if four or five or one million users want to share a secret, it requires that every user will be connected with every other user. This creates a complete graph and if one of the nodes is not secure, the rest are not secure too.

Cooperative Message Authentication Protocol : In this section, a cooperative message authentication protocol is discussed , which augments the basic short group signature protocol by mitigating the computation overhead in the regular broadcast phase. The workflow of cooperative message authentication protocol is shown in Figure 2[13].

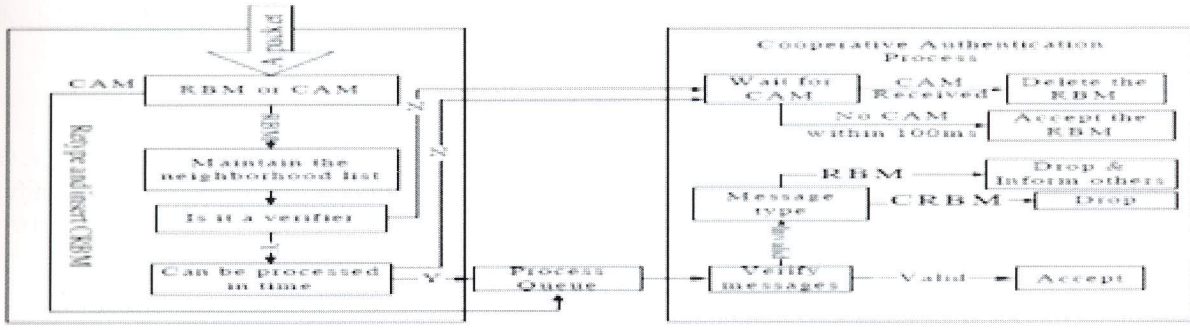

Figure 2: Work flow of the cooperative message authentication protocol

Each vehicle maintains two processes which are verifiers selection process and cooperative authentication process, a neighborhood list, a process queue and a buffer[13]. The verifiers selection process is in charge of selecting verifiers, neighborhood list and process queue maintenance. The cooperative authentication process controls message authentication and warning message sending. In other words, verifiers selection process fills the process queue while cooperative authentication process clears it up after verifications. The neighborhood list contains neighbor vehicles geographic information. Messages which will not be processed are stored in the buffer. When a vehicle receives a regular broadcast message (RBM), it extracts information of the location, speed, direction and acceleration of the sending vehicle and decides whether to verify the message or not according to geographic information. If a verifier finds an invalid RBM, it will broadcast one-hop warning information, which is termed as cooperative authentication messages (CAM), to inform others. A non-verifier resorts to the CAM broadcasted by other vehicles to authenticate RBM.

Result Analysis

1. Table and Graph for RBM message type

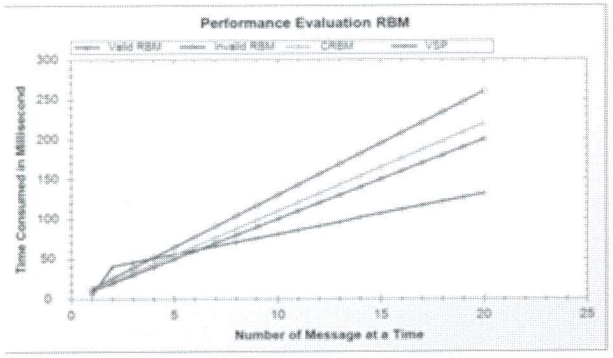

Figure 3 : Performance analysis of Regular Broadcast Message

Table 2 : Table for RBM message type

Type of Message	Time Taken in milliseconds
Valid RBM	197ms
Invalid RBM	260ms
CRBM	218ms

The Figure 3 and Table 2 gives a performance analysis of RBM(regular broadcast message) based on the number of messages to the total amount of time taken to process that message. Here total number of messages is 20. If the message type is valid RBM it will take less time to process. If it is invalid it will take more time because it has to be checked first and it should be dropped and it is informed to others. For CRBM message type the protocol will take around 218 milliseconds to process it.

2. Table and Graph for CAM message type

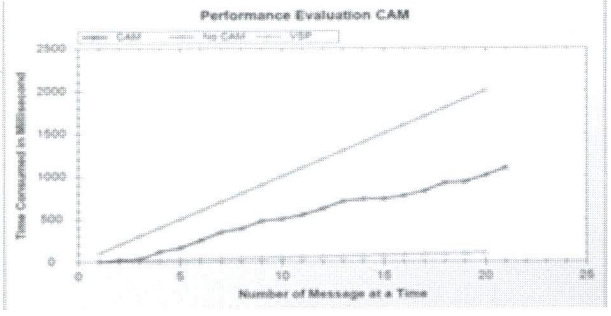

Figure 4 : Performance analysis of Cooperative authentication Message

Table 3 : Table for CAM message type

Type of Message	Time Taken in milliseconds
CAM	1100ms
No CAM	2000ms

The Figure 4 and Table 3 gives a performance analysis of CAM(cooperative authentication message) based on the number of messages to the total amount of time taken to process that message. Here total number of messages is 20. CAM is a warning message and if it is found it will be informed to other vehicle by verifier vehicle. No Cam is nothing but the RBM message type and it will take around 1100 milliseconds to process it while CAM takes around 2000 milliseconds.

3. Graph of Road Side Unit

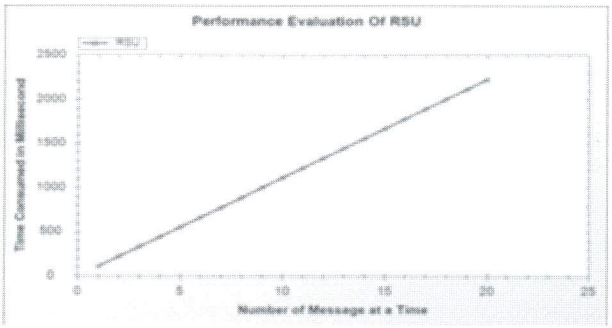

Figure 5: Performance analysis of Road Side Units

The Figure 5 gives a performance analysis of RSU's (Road Side Unit) based on the number of messages to the total amount of time taken to process that message. The total number of messages is 20. Here both the RSU will take the same time in order to broadcast the message because for every 30 seconds they are getting new keys from the authority. The time taken by the RSU's are around 2200 milliseconds.

4. Table and Graph for Process queue based on number of messages

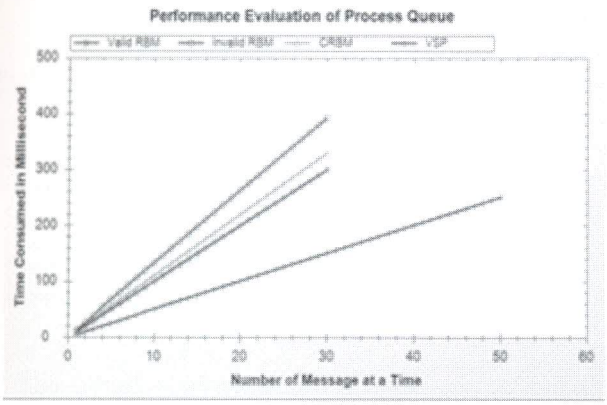

Figure 6 : Process Queue performance analysis

Table 4 : Table for Process Queue

Type of Message	Time Taken in milliseconds	Number of messages
Valid RBM	296ms	30
Invalid RBM	392ms	30
CRBM	320ms	30
VSP	240ms	50

The Figure 6 and Table4 show the processing of three types of messages in a process queue. Here the total number of passing messages is 50 to process queue. If the number of messages exceeds 30 the CMAP will process up to 30 messages and other messages will be discarded and thereby it will avoid computation overhead. Here VSP is verifier selection process and single vehicle is considered as verifier.

5. Table and Graph for Diffie-Hellman vs Shortgroup signature

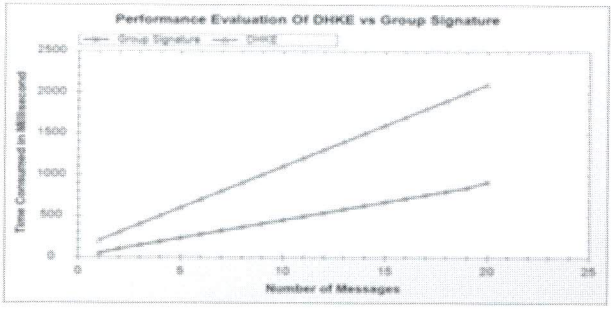

Figure 7 : Performance evaluation of Diffe-hellman algorithm with Shortgroup signature

Table 5 : Table for CAM message type

Type of Method (Algorithm)	Time Taken in milliseconds
Short Group Signature	900ms
Diffie-Hellman	2100ms

The Figure 7 and Table 5 shows the performance evaluation of short group signature with Diffe-hellman algorithm. The total number of messages to the process is 20.The short group signature is more secure compared to Diffie-Hellman algorithm because in each step of message passing every vehicle and RSU's are authenticated.

Conclusion : The proposed distributed key management scheme based on the short group signature is developed to provision privacy in the VANETs. The distributed key management is further enhanced with a cooperative message authentication protocol to alleviate the heavy computation overhead. The challenging issue that semi-trust RSUs may be compromised, and compromised RSUs may even collude with malicious vehicles is investigated. A security protocol to prevent compromised RSUs and malicious vehicles from attacking is designed. This design guarantees that RSUs distribute keys fairly and provide some mechanisms to detect compromised RSUs and malicious vehicles. Moreover, by a cooperative message authentication protocol, a vehicle only needs to verify a small amount of messages, and the computation burden of vehicles is reduced greatly

References :

[1] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks,"*Journal of Computer Security*, vol. 15, no. 1, pp. 39-68, 2007.

[2] R. Lu, X. Lin and X. Shen, "SPRING: A social-based privacy-preserving packet forwarding protocol for vehicular delay tolerant networks", in *Proc. IEEE INFOCOM*, San Diego, California, 2010.

[3] J. Freudiger, M. Raya, M. Feleghhazi,P. Papadimitratos and J.P.Hubaux., "Mix zones for location privacy in vehicular networks,"in *Proc. International Workshop on Wireless Networking for Intelligent Transportation Systems*, Vancouver, British Columbia, Aug., 2007.

[4] L. Huang, K. Matsuura, H. Yamane, and K. Sezaki, "Enhancing wireless location privacy using silent period," in *Proc. IEEE WCNC*, pp. 11871192,2005.

[5] K.Sampigethava, L.Huang, M.Li, R.Poovendran, K.Matsuura and K.Sezaki, "AMOEBA: Robust location privacy scheme for VANET," in *IEEE J. Sel. Areas Commun.*, vol. 25, no. 8, pp.1569-1589, 2007.

[6] D. Chaum and E. van Heyst, "Group signatures," in *Proc. Advances in Cryptology - Eurocrypt*, vol. 547, pp. 257-265, 1991.

[7] J. Guo, J.-P. Baugh and S. Wang, "A group signature based secure and privacy-preserving vehicular communication framework," in *Proc. IEEE INFOCOM*, Anchorage, Alaska, May 2007.

[8] X. Lin, X. Sun, P.-H. Ho and X. Shen, "GSIS: a secure and privacy preserving protocol for vehicular communications," *IEEE Trans. Veh. Technol.*, vol. 56, no. 6, pp. 3442-3456, 2007.

[9] G. Calandriello, P. Papadimitratos, A. Lloy, and J.-P. Hubaux, "Efficient and robust pseudonymous authentication in VANET," in *Proc. ACM Mobicom*, pp. 19-28, QC, Canada, Sept. 2007.

[10] IBM 4764 PCI-X Cryptographic Coprocessor. http://www-03.ibm.com/security/cryptocards/pcixcc/order4764.shtml.

[11] N. Banerjee, M.D. Corner, D. Towsley and B.N. Levine, "Relays, base station and meshes: enhancing mobile networks with infrastructure," in *Proc. ACM Mobicom*, San francisco, California, Sep. 2008.

[12] P. Enge, "Retooling the global positioning system," *ScientificAmerican*, May 2004.

[13]."A Distributed Key Management Framework with message authentication in VANETs"IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 29,NO. 3, MARCH 2011 Yong Hao, *Student Member, IEEE, Yu Cheng, Senior Member, IEEE,* Chi Zhou, *Senior Member, IEEE, and WeiSong.*

Dr. Nalini N. is presently working as Professor and Head, Department of Computer Science and Engineering at Nitte Meenakshi Institute of Technology, Yelahanka, Bangalore. She completed her B.E from Kuvempu University in the year 1996 and MS from BITS, Pilani in the year 1999 and her PhD from Visvesvaraya Technological University in the year 2007. She has more than 16 years of teaching and 10 years of research experience. Her research interests are Cryptography and Network Security, Wireless Sensor Networks, Security issues in Cloud Environment. She has more than 12 International Journal papers and 25 National/International Conference publications to her credit. She has received "Bharath Jyothi Award "by Dr.Bhishma Narain Singh, Former Governor of Tamilnadu and Assam, given by India International Friendship Society on 20-DEC-2012 at New Delhi. She is a lifetime member of ISTE, CSI, ACEEE and IIFS.

Chandrika C N, Completed her MTech in Computer Science and Engineering from Nitte Meenakshi Institute of Technology, Bangalore in the year 2013.