Key-Management Systems in Vehicular Ad-Hoc Networks

Sasikumar P
Assistant professor
School of Electronics Engineering
VIT University

Vivek C
Assistant professor
School of Electronics Engineering
VIT University

Jayakrishnan P
Assistant professor
School of Electronics Engineering
VIT University

ABSTRACT

Networking in vehicles is a promising approach to facilitate road safety, traffic management, and infotainment dissemination for drivers and travelers. Hence it becomes essential to provide security services such as authentication, non-repudiation, confidentiality, access control, integrity, and availability. The possible types of attacks include eavesdropping, denial of service and replay attacks. The security feature being implemented has to be tailor-made to suit the resource constraints imposed by the mobile nodes. In this paper I propose an approach to distribute the key management activities among the nodes using the concept of the RSA algorithm, the D-H algorithm and the RC4 algorithm. The message is encrypted using a public key and the corresponding private key is shared among the participating parties in all of these cases. The efficiency of each of these cases is demonstrated using the Network Simulator tool.

General Terms

Vehicular networks, Roadside-to-Vehicle Communication, Key-Management Systems, DH algorithm, RSA algorithm, RC4 algorithm

1. INTRODUCTION

The invasion wireless communication technologies have revolutionized human lifestyles in providing the most convenience and flexibility over accessing Internet services and reliable services offered for privacy. Now days the car manufacturers and telecommunication companies have been gearing up to equip each car with technology that allows drivers and travelers to communicate with each other, and presently due to high traffic levels of the road and some critical sections of the road needs special attention to improve the driving experience and make driving safer. [1] For example, Microsoft Corp.'s MSN TV and KVH Industries, Inc. have introduced an automotive vehicle Internet access system called TracNet, which can bring Internet service to any in-car video screen. It also turns the entire vehicle into an IEEE 802.11-based Wi-Fi hotspot, so passengers can use their wireless-enabled laptops to go online.

The onboard units (OBUs) are used in the vehicles to communicate with one other directly or through roadside units (RSUs) located at various points on the road. A self-organized network is formed by connecting the vehicles and RSUs, normally called a vehicular ad hoc network (VANET). All the RSUs in the defined architecture are connected to one another by using a backbone network for centralized services. Increasing interest has been raised recently in the applications of roadside-to-vehicle communications (RVCs) and inter-vehicle communications (IVCs), aiming to improve driving safety and traffic management.

1.1 Motivation

The challenges in VANET are located especially in the aspects of security and privacy. This is due to the inheritance of mobile ad hoc networks (MANETs), a VANET inherits all the known and unknown security weaknesses associated with MANETs, and could be subject to many security and privacy threats. It is obvious that any malicious behavior of users, such as a modification and replay attack with respect to the disseminated messages, could be fatal to the other users. In addition, the issues in VANET security become more challenging due to the unique features of networks, such as the high-speed mobility of the network entity (or vehicle) and the extremely large amount of network entities. Furthermore, conditional privacy preservation must be achieved in the sense that user related privacy information, including the driver's name, license plate, speed, position, and traveling routes along with their relationships, has to be protected; while the authorities should be able to reveal the identities of message senders in case of dispute such as a crime/car accident scene investigation, which can be used to look for witnesses. Therefore, it is critical to develop a suite of elaborate and carefully designed security mechanisms for achieving security and conditional privacy preservation in a VANET.

1.2 Problem description

The Vehicular ad hoc networks (VANETs) are dynamically formed by autonomous mobile nodes using multi-hop communication. Its highly dynamic topology and the shared wireless medium leads to new challenges in providing network security. Hence it becomes essential to provide security services such as authentication, confidentiality, integrity, availability, etc. It is hence necessary to implement a distributed key management system to manage the secure communication among the nodes. The security scheme implemented should be efficient and should not bring down the performance of the network.

2. RELATED WORK

A mobile ad hoc network MANET is a group of mobile nodes independent from any centralized administration. Each mobile node is able to communicate by radio waves with other nodes within its transmission range and relays on other nodes to communicate with mobile nodes outside its transmission range. The shared wireless medium, the highly dynamic topology, and the lack of any centralized network management make MANETs vulnerable to infiltration, eavesdropping, interference, and so on. The need for security in MANET is an essential component to supply the network with the basic functions such as routing and packet forwarding. Security in mobile ad hoc network is considered to be more difficult than traditional networks due to the lack of infrastructure.

Many security solutions rely on public key cryptography, the deployment of which requires the effective management of digital certificates. A certificate is a statement issued by trusted party saying that it verifies that the public key belongs to the user. In the mobility environment of MANETs, only distributive key management schemes can work efficiently.

There has been a rich literature on public-key management in MANETs. Some schemes depend on certificate-based cryptography [1], [2], [3], [4], and [5] in which public-key certificates are used to authenticate public keys by binding public keys to the users' identities. Identity-based (ID-based) key management schemes have a simple key management process and reduced memory storage cost compared to other methods. In ID-based schemes the node or user identity, such as an IP address, is used to derive its public key, while the private key is generally provided by an external entity.

The main concern with this approach is the need for public key certificate distribution. This approach suffers from lack of scalability with increasing the network size. Another approach is providing keying material through a web of trust [6], [7]. In these schemes, each node generates the public/private key pair by the node itself, issues certificates to its neighboring nodes and holds these certificates in its certificate repository. Key authentication is performed via chains of certificates. However, this scheme suffers from the delay and the large amount of traffic needed to collect certificates.

Distributed key generation by distributing trust among a group of nodes provides a promising solution for the above problems. In the distributed key generation, a set of n servers jointly generate a pair of public and private keys in a way that the public key is known to all nodes in the network while the private key is divided between the n servers via a threshold secret sharing scheme such as Shamir's (t, n) threshold cryptography [8]. Later, in order for a new node to join the network, at least t nodes (among n nodes) need to cooperate and sign a certificate for the new node. Blom proposed a symmetric key generation system (SKGS) based on secret sharing systems. In SKGS, nodes are supplied with a relatively small amount of secret data that is used to derive all the node's keys.

A central server (trusted authority) generates a global matrix G of size $k \times n$ that is known to all the nodes in the network, and a symmetric secret matrix D of size $k \times n$. The central node calculates the key matrix for the network as $K = (D \cdot G)T \cdot G$. Because D is symmetric, K will be also symmetric, for rows i and j in K, we have $Ki,j = Kj,i$. So, Ki,j is common between the rows i, j. If row i is the key chain for node i, and row j is the key chain for node j, the element Ki,j will be the symmetric key between them.

Because G is known by the entire participant in the networks, while external nodes (malicious nodes) do not know this matrix G, the central server delivers the i^{th} row of

$(D \cdot G)\ T$ to node i. Upon reception, node I will calculates its key chain $k_i = i^{th}$ row of $(D \cdot G)T \cdot G$. This division of key generation into more than one step adds more secrecy to their key chains and makes it harder for the intruder to retrieve any information about other nodes. Blom showed in his paper that by using SKGS scheme, at least k users have to co-operate to get any information about keys they do not have. Due to node mobility and geometric constrains, the connectivity of the network graph is hard to achieve in the web of trust key management schemes. Consequently, the process of public key authentication fails. In addition, in some situations (such as in the battlefield), there is a need for a level of trust between the network entities higher than that of the web of trust approach. In [9], Hisham Dahshan and James Irvine propose a robust scheme which combines the web of trust model and threshold cryptography. Certificate signing and certificate revocation is done by a group of trusted nodes.

3. CRYPTOGRAPHY MODELS

In classical cryptography models, Alice and Bob secretly choose a key. This key enables both the encryption and decryption of the message to be sent through the use of publicly known encryption and decryption algorithms. The main drawback of this technique is that it requires prior communication of the key via a secure channel, which is often unavailable.

The idea of a public-key cryptosystem was put forward by Diffie and Hellman in 1976. A public-key cryptosystem is based on the assumption that it might be possible to find a system where it is computationally infeasible to determine the decryption rule given its encryption rule. If so, the encryption rule is a public key that can be published in a directory whereas the decryption rule is the private key that is known solely by the recipient.

Encryption is the act of encoding text so that others not privy to the decryption mechanism (the "key") cannot understand the content of the text. Encryption has long been the domain of spies and diplomats, but recently it has moved into the public eye with the concern of the protection of electronic transmissions and digitally stored data. Standard encryption methods usually have two basic flaws (1) A secure channel must be established at some point so that the sender may exchange the decoding key with the receiver; and (2) There is no guarantee that who sent a given message. Public key encryption has rapidly grown in popularity because it offers a very secure encryption method that addresses these concerns.

In a public-key cryptosystem, the sender encrypts a message with the recipient's public key. This key is usually posted in a directory similar to a phone book. Upon receiving the message, the recipient uses his/her own private key to decrypt the message. For example, Alice encrypts a message using Bob's public key and sends it to him over an insecure channel. Bob then decrypts the message with a private key that is known only to him.

With this project, I am to make a comparison between 3 of the most popular key management schemes namely, RSA, Diffie-Hellman and RC4 algorithms. Cryptography achieves the security needs such as confidentiality and integrity against malicious nodes. It also provides data integrity and availability in a hostile environment and can also employ verification of the correct data sharing. All this is achieved without revealing the secret key.

RC4 (also known as ARC4 or ARCFOUR meaning Alleged RC4, see below) is the most widely-used software stream cipher and is used in popular protocols such as Secure Sockets Layer (SSL) (to protect Internet traffic) and WEP (to secure wireless networks). While remarkable for its simplicity and speed in software, RC4 has weaknesses that argue against its use in new systems. It is especially vulnerable when the beginning of the output keystream is not discarded, non random or related keys are used, or a single keystream is used twice; some ways of using RC4 can lead to very insecure cryptosystems such as WEP.

RSA is a public-key cryptosystem that supports both encryption and digital signatures (authentication).Like all public-key cryptography models, the RSA cryptosystem encrypts and decrypts a message using a pair of keys known as public key and private key. Its security is based on the difficulty of factoring large integers. Presently, most implementations of the RSA algorithm employ the use of 512-bit numbers. Cracking such a system requires the ability to factor the product of two 512-bit prime numbers. Factoring a number of this size is well beyond the capability of the best current factoring algorithms.

4. ANALYSIS AND DESIGN

4.1 RSA Algorithm

Encryption

Alice transmits her public key *(n,e)* to Bob and keeps the private key secret. Bob then wishes to send message M to Alice.

He first turns *M* into an integer 0 < m < n by using an agreed-upon reversible protocol known as a padding scheme. He then computes the cipher text *c* corresponding to:

$$c = m^e \bmod n$$

This can be done quickly using the method of exponentiation by squaring. Bob then transmits c to Alice.

Decryption

Alice can recover m from *c* by using her private key exponent *d* by the following computation:

$$m = c^d \bmod n$$

Given *m*, she can recover the original message *M* by reversing the padding scheme.

4.2 Diffie Hellman Algorithm

- For P and G are both publicly available numbers

  P is at least 512 bits

  Users pick private values a and b

  Compute public values:

  $$x = g^a \bmod p$$

  $$y = g^b \bmod p$$

  Public values x and y are exchanged

- Compute shared, private key

  $k_a = y^a \bmod p$

  $k_b = x^b \bmod p$

- Algebraically it can be shown that $k_a = k_b$

  Users now have a symmetric secret key to encrypt

4.3 RC4 Algorithm

RC4 generates a pseudorandom stream of bits (a keystream) which, for encryption, is combined with the plaintext using bit-wise exclusive-or; decryption is performed the same way (since exclusive-or is a symmetric operation). (This is similar to the Vernam cipher except that generated pseudorandom bits, rather than a prepared stream, are used.)

To generate the keystream, the cipher makes use of a secret internal state which consists of two parts:

1. A permutation of all 256 possible bytes (denoted "S" below).

2. Two 8-bit index-pointers (denoted "*i*" and "*j*").

The permutation is initialized with a variable length key, typically between 40 and 256 bits, using the key-scheduling algorithm (KSA). Once this has been completed, the stream of bits is generated using the pseudo-random generation algorithm (PRGA).

The key-scheduling algorithm is used to initialize the permutation in the array "*S*". "*keylength*" is defined as the number of bytes in the key and can be in the range *1 ≤ keylength ≤ 256*, typically between 5 and 16, corresponding to a key length of 40 – 128 bits. First, the array "S" is initialized to the identity permutation. S is then processed for 256 iterations in a similar way to the main PRGA, but also mixes in bytes of the key at the same time

For as much iterations are needed, the PRGA modifies the state and output - a byte of the keystream. In each iteration, the PRGA increments i, adds the value of S pointed to by *i* to *j*, exchanges the values of *S[i]* and *S[j]*, and then outputs the value of *S* at the location *S[i] + S[j]* (modulo 256). Each value of *S* is swapped at least once in every 256 iterations.

Hence all the three algorithms are implemented as shown above and are implemented using C coding for simulation purpose

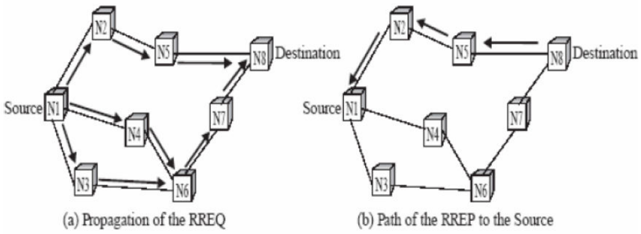
5. IMPLEMENTATION

5.1 Tools Used

NS or the network simulator (also popularly called ns-2, in reference to its current generation) is a discrete event network simulator. ns is popularly used in the simulation of routing and multicast protocols, among others, and is heavily used in ad-hoc networking research. ns supports an array of popular network protocols, offering simulation results for wired and wireless networks alike. It can be also used as limited-functionality network emulator. Emulation refers to the ability to introduce the simulator into a live network. Special objects within the simulator are capable of introducing live traffic into the simulator and injecting traffic from the simulator into the live network. It is popular in academia for its extensibility (due to its open source model) and plentiful online documentation. NS was built in C++ and provides a simulation interface through OTcl, an object-oriented dialect of Tcl. The user describes a network topology by writing OTcl scripts, and then the main NS program simulates that topology with specified parameters. It implements network protocols such as TCP and UPD, traffic source behavior such as FTP, Telnet, Web, CBR and VBR, router queue management mechanism such as Drop Tail, RED and CBQ, routing algorithms such as Dijkstra, and more. NS also implements multicasting and some of the MAC layer protocols for LAN simulations. Ns make use of flat earth model in which it assumes that the environment is flat without any elevations or depressions. The measurements of the network performance are made using a script that analyzes the trace file output provided by the ns2. Simulations in ns2 can be logged into trace files that can be used by network animator. The network animator is a visualization tool to see the network running.

5.2 Routing Protocol

AODV [10] is built upon DSDV routing protocol. DSDV is required to maintain a complete list of routes, whereas AODV creates routes on an on-demand basis; i.e., only when desired. This approach considerably reduces the number of required broadcast messages. When a source node desires to send data to a destination node, it checks if it already has a route to that particular destination node. If no valid route is present, it initiates a route discovery process to locate the other node. The source node sends out a Route Request (RREQ) to its neighbors, which then is forwarded to its neighbors until the destination node is reached or an intermediate node with a route to the destination is found. Figure 7 shows the propagation of the RREQ packet within the network.



(a) Propagation of the RREQ (b) Path of the RREP to the Source

Figure 1. AODV route discovery

All intermediate nodes receive the RREQ packet update or record in their routing tables the address of the neighbor from which the packet was received. Once the destination route is found or destination route is reached, a Route Reply packet (RREP) is routed back to the source node along the reverse path. Once the source node receives the RREP packet, it can start sending data using the new found route.

AODV also supports route maintenance. If a mobile node moves away, it reinitiates the route discovery process to find a new route to the destination. When any intermediate node moves, the upstream node notices the move and broadcasts a route failure notification message. All the nodes receiving the route failure notification message forward the message up to the source node. The source node then chooses to re-initiate route discovery if the route is still desired.

5.3 Data Flow Diagram

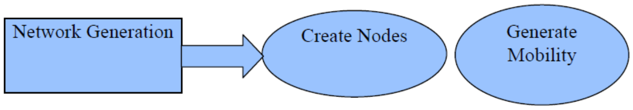


Figure 2. Creating Network

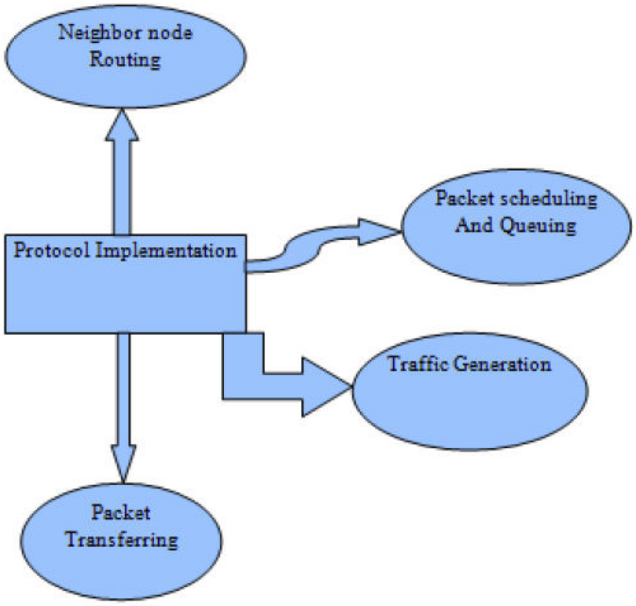


Figure 3. Implementing Protocol

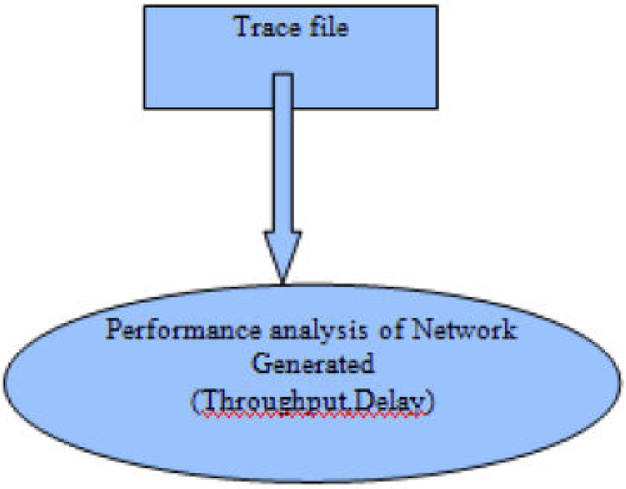


Figure 4. Generating Graph for analysis

6. RESULTS AND DISCUSSION

6.1 Simulation parameters

The Network simulator 2 was used to perform simulations. The key management scheme was implemented based on the existing Ad Hoc On-demand Distance Vector (AODV) routing protocol. The MAC layer protocol IEEE 802.11 is used. A total of 100 mobile nodes were created. Packet size is 500 bytes. The constant bit-rate (CBR) generator is used to set connection patterns with a traffic loading speed of 1 CBR packet/sec. The sending rate is 100 Mbps. The node mobility is varied from 2 m/sec to 10 m/sec. Figure 5, Three metrics were chosen to evaluate the performance: Packet Delivery Ratio, Average End-To-end Delay and number of packets dropped. Delivery ratio is the ratio number of data packets received at the destinations to those that are sent by the sources. Average End-to-End Delay is the time taken by a data packet to travel from the source to the destination. The number of attackers is varied from 10 to 50. The simulation graphs contrast the performance with and without the key management technique deployed.

Table 1. Table of Simulation Parameters

Simulator	Ns-2 (version 2.31)
Simulation time	20(s)
MAC Layer protocol	802.11
Number of mobile nodes	15
Topology	800m x 800m
Traffic loading speed	1 CBR packet/s
Routing protocol	AODV
Maximum bandwidth	100Mbps
Traffic	Constant Bit Rate
Maximum speed	2-10 m/s
Packet size	500 bytes

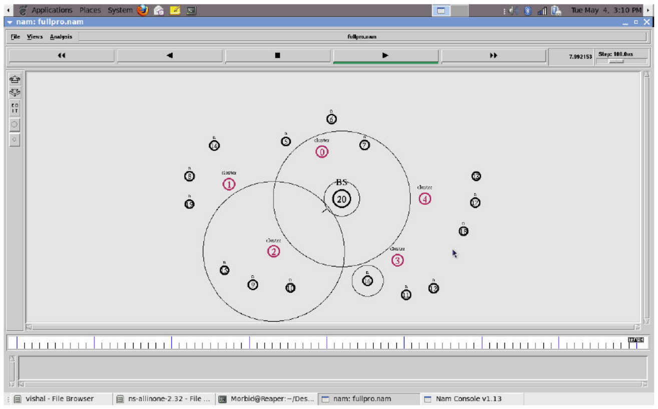


Figure 5. Showing Simulation Environment

6.2 Performance Metrics

We have selected the success ratio of the delivery of the packet with and without the scheme applied as the metric for this simulation. This is the ratio of packets delivered to the destination to the packets sent by the sender.

During the simulation, data is being transferred from the Base Station – BS, to the cluster heads, that represent RSU (pink) and the nodes – cars.

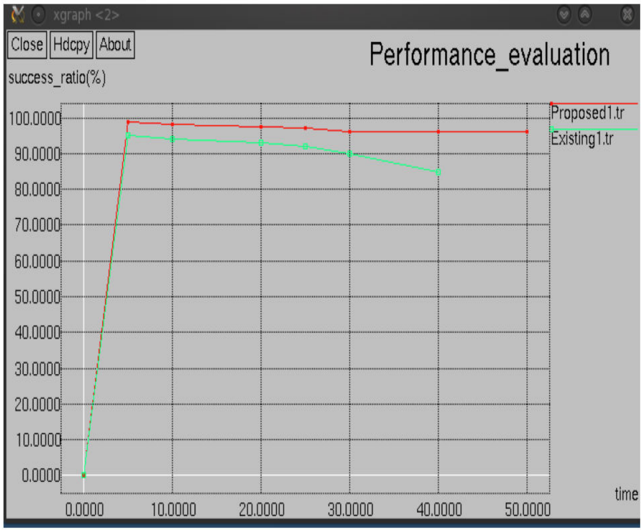


Figure 6. Showing performance of RSA algorithm

From the above shown figure, the red line represents the efficiency of the network by implementing the RSA algorithm scheme and the green without shows the performance when there is no security scheme implemented, which is prone to attacks

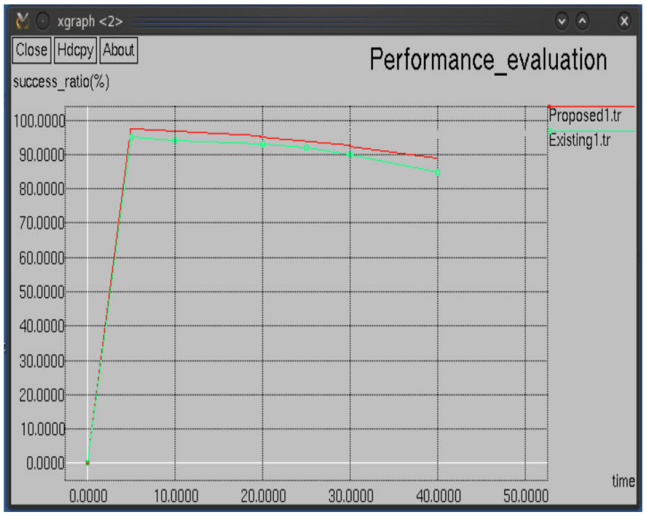


Figure 7. Showing performance of D-H algorithm

The red line represents the efficiency with D-H scheme implementation and the green without. As observed, the efficiency is less compared to that of the RSA algorithm

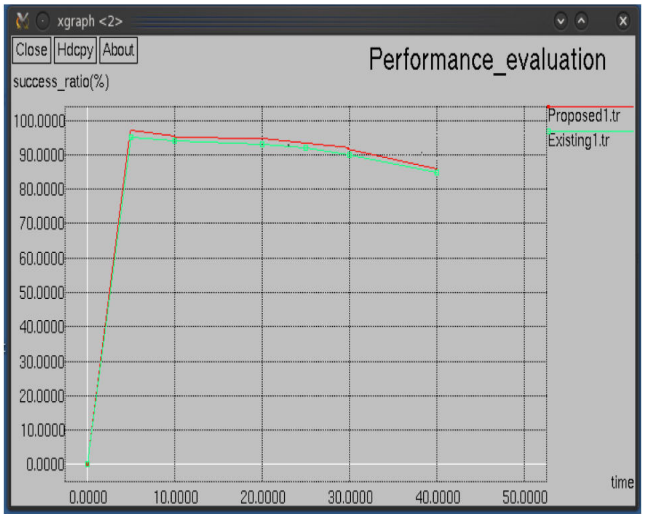


Figure 8. Showing performance of RC4 algorithm

This is the output for the RC4 scheme. It is observed that it's the least effective out of the three schemes employed. But this does not mean it is a bad scheme. RC4 is more effective in wired networks.

7. CONCLUSION

Hence, a comparison of the efficiencies of three key-management schemes for a VANET has been performed. From the results it has been shown that there is an increase in the efficiency of the system when there is a scheme in place. There is a considerable improvement in the data communication between the nodes after key management techniques have been employed. Out of the three schemes, it is found that the RSA algorithm is found to be the most efficient out of the three with the RC4 scheme being the least efficient for the model used. All this has been proved on by simulation on the Network Simulator 2 tool.

At critical security areas which are prone to attacks a key management technique is absolutely compulsory. Without it the delivery ratio becomes so less that there is no meaningful data communication possible. This technique can be used in security-sensitive applications like police and government agencies where VANETs are increasingly being used.

8. ACKNOWLEDGMENTS

Our thanks to the Vishal K. Thomas, UG Scholar VIT University, and all the faculty colleagues who participated in discussions to improve the work and our VIT University for their continuous encouragement.

9. REFERENCES

[1] X. Lin, R. Lu, C. Zhang, H. Zhu, P.H. Ho, and X. Shen, "Security in Vehicular Ad Hoc Networks", IEEE Communications Magazine, Vol. 46, No. 4, 88-95, 2008

[2] Nai-Wei Lo and Hsiao-Chien Tsai, "Illusion Attack on VANET Applications", IEEE Globecom Workshops, pp. 1–8 (2007)

[3] IEEE Std. 1609.2-2006, "IEEE Trial-Use Standard for Wireless access in Vehicular Environments-Security Services for Applications and Management Messages," 2006.

[4] P. Wohlmacher, "Digital Certificates: A Survey of Revocation Methods," Proc. ACM Wksp. Multimedia, Los Angeles, CA, Oct. 2000, pp. 111–14.

[5] M. Raya and J.-P. Hubaux, "Securing Vehicular Ad Hoc Networks," J. Computer Security, Special Issue on Security, Ad Hoc and Sensor Networks, vol. 15, no. 1, 2007,pp. 39–68.

[6] Pradeep B, Manohara Pai M.M and M. Boussedjra, J. Mouzna, Global Public Key Algorithm for secure location service in VANET, IEEE 2009

[7] P. S. L. M. Barreto et al., "Efficient and Provably-Secure Identity-Based Signatures and Signcryption from Bilinear Maps," Proc. Advances in Cryptology — ASIACRYPT 2005, Taj Coromandel, Chennai, India, Dec. 2005, pp. 515–32

[8] Xiaodong Lin, Student Member, IEEE, Xiaoting Sun, Pin-Han Ho Member, IEEE, and Xuemin (Sherman) Shen, Senior Member, IEEE "GSIS: A Secure and Privacy Preserving Protocol for Vehicular Communications", IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 56, NO. 6, NOVEMBER 2007

[9] A Stampoulis and Z Chai, Yale University "A survey of security in Vehicular Networks".

[10] Abedi, O, Fathy, M, and Taghiloo, J "Enhancing AODV routing protocol using mobility parameters in VANET" Computer Systems and Applications, 2008.