Volume 5, Issue 1, January 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Fully Self-Organized Key Management Based on Hash-Chain Scheme for EAACK in MANET

[1]P. Mohanambal, [2]A. Balasubramanian
[1]Department of Computer Science, SNR Sons College, Coimbatore, India
[2]Associate Professor, Department of Information Technology, SNR Sons College, Coimbatore, India

Abstract— The migration to wireless network from wired network has been a global trend in the past few decades. The mobility and scalability brought by wireless network made it possible in many applications. Among all the contemporary wireless networks, Mobile Ad-hoc Network (MANET) is one of the most important and unique applications. This research entitled "Fully Self-Organized Key Management Based On Hash-Chain Scheme For EAACK In MANET" considers the problem in EAACK where the traditional certificate – based key management scheme needs a fixed Certificate Authority or Key Generation Centre to manage certificates for all nodes. As this scheme needs to manage all the nodes, it is less secured from hackers and the attacks, it causes high computational complexity and highly expensive. To overcome this problem, this research proposes a fully self- organized key management scheme in mobile ad-hoc networks which is both certificate less and free from any trusted third party such as Certificate Authority or Key Generation Centre and it is also free from all the certificate operations such as distribution, validation, updation and revocation.

This self-organized key management for EAACK in MANET scheme allows a node to setup the public/private key pair all by itself and use the public key as its identity according to the property of ad-hoc networks. The public/ private key pair is generated by itself through its node ID and the session key is generated for more security while the nodes communicate in transmitting the packets. For the key generation, Diffie Hellman algorithm is used and for encrypting the data, RC4 algorithm is used.

RC4 also known as ARC4 or ARCFOUR meaning Alleged RC4 is the most widely used software stream cipher and is used in popular protocols such as Transport Layer Security (TLS) (to protect Internet traffic) and WEP (to secure wireless networks). Based on the scheme, some applications such as Encryption, Signature, Signcryption algorithms are given. A hash function is a transformation which has a broad application in authentication and signature. These applications show that the key management scheme is self-organized, simple, efficient and practical.

Keywords— Diffie Hellman, RC4, Session key generation, Generating public/private key.

I. INTRODUCTION

MANET is becoming more and more widely implemented in the industry. However, considering the fact that MANET is popular among critical mission applications, network security is of vital importance. Unfortunately, the open medium and remote distribution of MANET make it vulnerable to various types of attacks. For example, due to the nodes' lack of physical protection, malicious attackers can easily capture and compromise nodes to achieve attacks. In particular, considering the fact that most routing protocols in MANETs assume that every node in the network behaves cooperatively with other nodes and presumably not malicious, attackers can easily compromise MANETs by inserting malicious or non cooperative nodes into the network. Furthermore, because of MANET's distributed architecture and changing topology, a traditional centralized monitoring technique is no longer feasible in MANETs. In such case, it is crucial to develop a fully self-organized key management based on Hash chain scheme for EAACK is specially designed for MANETs which is **efficient, simple** and **Self-organized.**

II. PROPOSED SYSTEM

However, the traditional certificate-based key management scheme PKI needs a fixed Certificate Authority to manage certificates for all nodes; moreover, the certificate-based scheme uses a certificate to bind the public key and identity of a node which results in the complication and inefficiency of key management. The identity based key management scheme, although use identity as the public key and is of no certificate, needs a Private Key Generator as the trusted third party. Therefore, neither PKI nor identity based scheme fits for the mobile ad hoc networks, which is characterized by self-organization, distribution and autonomy.

Self – Organized Key Management:

Key management is the management of cryptographic keys in a cryptosystem. This includes dealing with the generation, exchange, storage, use, and replacement of keys. Thus, allowing for the above properties of MANET, the work proposes a fully self-organized key management scheme, which is both certificate less and free for many trusted

third party such as Certificate Authority or Key Generator Centre. The scheme allows a node to set up the public/private key pair all by itself and use the public key as its identity according to the property of ad hoc networks. The proposed public key scheme has many advantages.

(1)Efficient and Simple: Compared to the traditional certificate-based key management schemes, the proposed scheme is free from all the certificate operations such as distribution, validation, update and revocation; in contrast with ID-based schemes, the proposed scheme needs not to communicate with the KGC(Key Generation Centre). That is, the proposed scheme is a light weight one.

(2) Self-organized: On the one hand, the traditional certificate-based key management relies on the support of CA (Certificate Authority) online. On the other hand, although ID-based key management scheme can directly acquire the public key of a node according to its identity information, but the calculation of the private key must rely on KGC (Key Generation Centre). Once the KGC (Key Generation Centre) is compromised, the security of the whole system will collapse. But the proposed scheme does not need any support of TTP (Trusted Third Party) and all key management operations are done by the node itself.

Signencryption Protocol

Sign and encrypt: If **A** needs to send a message **m** to **B** with confidentiality and integrity guaranteed, it computes **c** as the Sign encryption:

$$c = m \oplus H_1(e(x_A Q, PB_{pub}))$$

Verify and decrypt: After receiving **c** from **A**, node **B** resumes **m** as follows:

$$m = c \oplus H_1(e(x_B Q, PA_{pub}))$$

Hash-chain Hn (M) is to do n times hash transformation to the initialized input M, making use of hash function's unidirectivity character. Once getting Hr(M) ($1 \leq r \leq n$),you can easily compute $H^{r+1}(M)$,$H^{r+2}(M)$….$H^{n}(M)$ by hash function, but you can't compute $H^{r-1}(M)$, $H^{r-2}(M)$ … M. Because of the unidirectivity, hash-chain has always been used as the renewable encryption, decryption and authentication.

A hash function H is a transformation that takes a variable-sized input M and returns a fixed-size string, which is called the hash value m (that is, m = H (M)).This transformation relation is many-to-one because the range of output value is much less than the range of input value. Different input may have the same result, so you can't ensure the input value through the result. So hash function has broad application in authentication and signature.

Session key:

Session key is generated by using Diffie – Hellman algorithm. Diffie-Hellman key exchange algorithm enables two users to exchange a key securely over the communication channel. The key can then be used by both parties to encrypt and decrypt data. This algorithm is limited to the exchange of the keys. No KDC or central authentication server is required.

Diffie-Hellman key exchange algorithm enables two users to exchange a key securely over the communication channel. The key can then be used by both parties to encrypt and decrypt data. This algorithm is limited to the exchange of the keys. No KDC or central authentication server is required.

Performance Evaluation

The result of existing and proposed technique is compared with the following parameters like routing overhead and packet delivery ratio.

***Packet Delivery Ratio (PDR)*:** PDR defines the ratio of the number of packets received by the destination node and the number of packets sent by the source node.

***Routing Overhead (RO)*:** RO defines the ratio of the amount of routing-related transmissions (RREQ, RREP, RERR, ACK, S-ACK and MRA).

Advantages of the proposed system:

- Key distribution is an important aspect of conventional algorithm and the entire safety is dependent on the distribution of key using secured channel.
- Using Public Key Cryptography (PKC), nodes can negotiate the session key for secure communication that fulfills the requirement of confidentiality.
- Security analysis results show that protocol establishes a route secure from different kind of attacks such as reply attack, rushing attack, IP spoofing and man in the middle attack.
- Sender compute the multiplication between the coordinates of the key in the encryption algorithm, and the receiver compute the multiplication between the coordinates of the key in the decryption algorithm and this approach is used for forward secrecy.
- In Diffie-Hellman key exchange algorithm no KDC (Key Distribution Centre) or central authentication server is required.
- Less expensive and more secured.

III. ALGORITHM

Diffie- Hellman Algorithm:

Let's assume a party **A** wants to establish a secure session with party **B**. To do this**, A** requires a one-time session key to encrypt the data over the connection. **A** and **KDC** share a secret **key K_A.** Similarly, **KDC** and **B** share secret key **K_B.** The secure channel establishment goes through following steps:

1. **A** issue a request to **KDC** for a session key for its communication with **B**. Request message from **A to KDC** contains: identity of **A** and **B** and a unique identifier **N_1**, for this transaction. This unique identifier is referred to as a nonce. This nonce differs in each request and is generated randomly.
2. The **KDC** responds back with a message encrypted using **K_A**. Since only **A** and **KDC** share this secret, **A** knows that the message is authentic. The message for **A** includes following two items:
   a. The one time session-key **K_s**, to be used for the session with **B**
   b. The original request message, including the nonce, to enable **A** to match this response with the appropriate request.

Thus **A** can verify that its original request was not altered before reaching the **KDC** and because of the nonce, it is not a replay of some previous request.

In addition, the **KDC** message includes two items intended for **B** (sent to **B** by **A** later):

1) The one-time session key, **K_s**, to be used for the session
2) The identifier of **A, ID_A**

These items to be sent to **B** are encrypted using **K_B**, hence **B** will know they have been originated by **KDC**.

3. **B** on getting the message for **A, B** knows that the other party is **A** in the session and session keys are authentic. This means the session keys have been securely exchanged between the two parties.
4. Using the session key **K_s, B** now encrypts and sends a nonce **N_2, to A**.
5. Also knowing **K_s, A** responds with some **$f(N_2)$.**

RC4 Algorithm:

The RC4 algorithm is remarkably simply and quite easy to explain. A variable-length key of from 1 to 256 bytes (8 to 2048 bits) is used to initialize a 256-byte state vector S, with elements S[0], S[1], ..., S[255]. At all times, S contains a permutation of all 8-bit numbers from0 through 255. For encryption and decryption, a byte k is generated from S by selecting one of the 255 entries in a systematic fashion. As each value of k is generated, the entries in S are once again permuted.

The steps for RC4 encryption algorithm is as follows.

1. Get the data to be encrypted and the selected key.
2. Create two string arrays
3. Initiate one array with number from 0 to 255.
4. Fill the other array with the selected key.
5. Randomize the first array depending on the array of the key.
6. Randomize the first array within itself to generate the final key stream.
7. XOR the final key stream with the data to be encrypted to give cipher text

IV. RESULT AND DISCUSSION

Description about network simulation

In this section, the performance of the existing and the proposed system is evaluated. In the existing system, Enhanced Adaptive Acknowledgment (EAACK) specially designed for MANETs. In the proposed system, fully self-organized key management scheme is used. The existing and the proposed method are compared in terms of packet delivery ratio and routing overhead.

Input parameter,

Number of nodes

It is defined as the number of nodes used in the simulation.

Description about output parameters

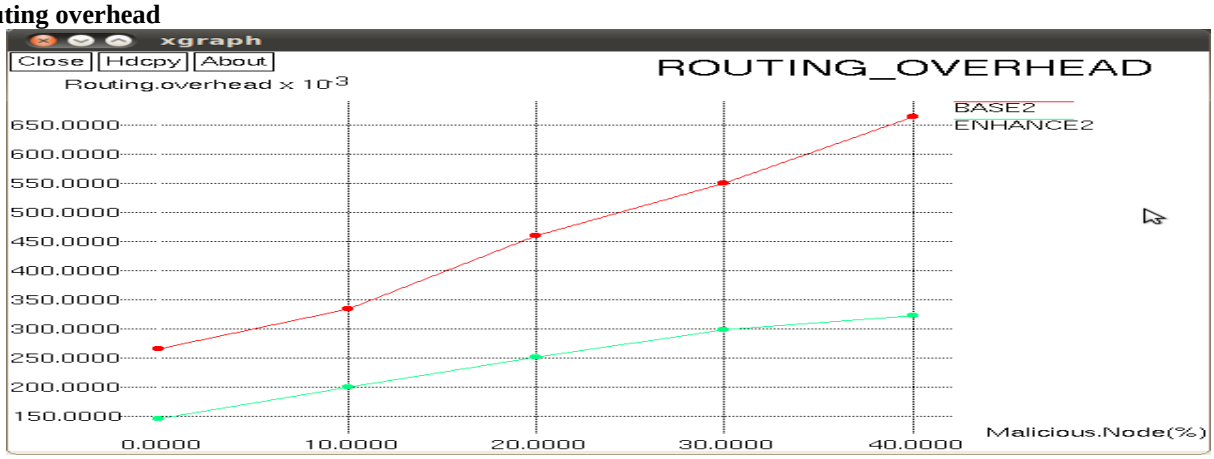
Routing overhead

It is defined as the ratio of the total packet size of control packets to the total packet size of data packets delivered to the destinations. For the control packets sent over multiple hops, each single hop is counted as one transmission.

Packet delivery ratio

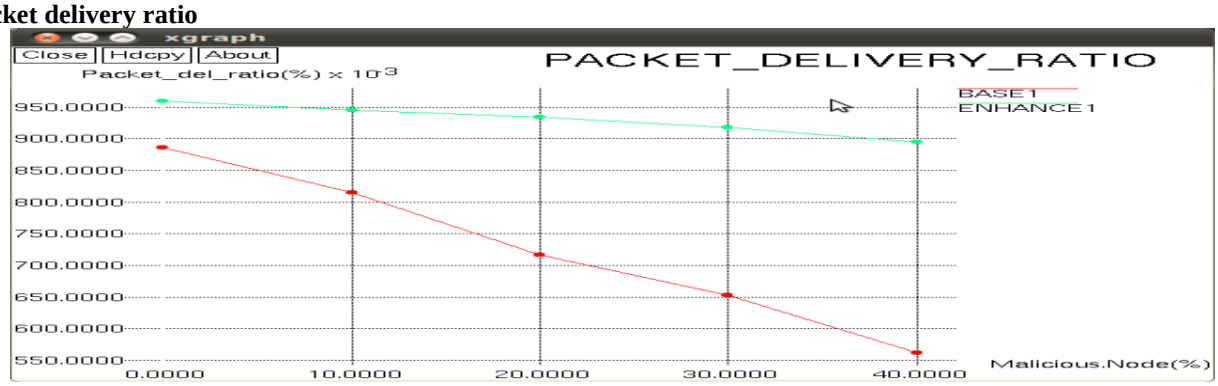
It is defined as the ratio of the number of delivered data packet to the destination. This illustrates the level of delivered data to the destination. ∑ Number of packet received / ∑ Number of packet sends

Routing overhead



The routing overhead is shown in this graph. In the X-axis number of nodes is taken. Y-axis routing overhead is taken. This graph clearly shows that if the number of nodes increases the routing overhead is increased in the existing system. But in the proposed system, routing overhead is decreased.

Packet delivery ratio



The packet delivery ratio is shown in this graph. In the X-axis number of nodes is taken. Y-axis packet delivery ratio is taken. This graph clearly shows that if the number of nodes increases the packet delivery ratio is decreased in the existing system. But in the proposed system, packet delivery ratio is increased.

V. CONCLUSION

Packet- dropping attack has always been a major threat to the security in MANETs. In this research paper, a Fully Self Organized Key Management based on Hash Chain scheme for EAACK in MANET is proposed and compared in different scenarios through simulations. The result demonstrated positive performance against the centralized Authority or Key Generation Centre a fully self- organized key management scheme in mobile ad-hoc networks which is both certificate less and free from any trusted third party such as Certificate Authority or Key Generation Centre and it is also free from all the certificate operations such as distribution, validation, updation and revocation. Malicious nodes are identified and detected, through this approach the routing overhead decreases when compared with the existing system and packet delivery ratio increases. Diffie-Hellman Algorithm is used for generating session key and RC4 algorithm for encrypting the data.

To improve the merits of the proposed system, the following are the future enhancement that can be implemented:

- The possibilities of adopting hybrid cryptography techniques to further reduce the network overhead.
- In mobile ad-hoc network to increase the merits, the performance of EAACK can be tested in real network environment instead of software simulation.

REFERENCE

[1] K. Al Agha, M.-H. Bertin, T. Dang, A. Guitton, P. Minet, T. Val, andJ.-B. Viollet, "Which wireless technology for industrial wireless sensornetworks? The development of OCARI technol," *IEEE Trans. Ind. Electron.*,vol. 56, no. 10, pp. 4266–4278, Oct. 2009.

[2] R. Akbani, T. Korkmaz, and G. V. S. Raju, "Mobile Ad hoc NetworkSecurity," in *Lecture Notes in Electrical Engineering*, vol. 127.New York: Springer-Verlag, 2012, pp. 659–666.

[3] R. H. Akbani, S. Patel, and D. C. Jinwala, "DoS attacks in mobile ad hocnetworks: A survey," in *Proc. 2nd Int. Meeting ACCT*, Rohtak, Haryana,India, 2012, pp. 535–541.

[4] "Evaluating And Comparison Of Intrusion In Mobile Ad Hoc Networks" Zougagh Hicham, Toumanari Ahmed, Latif Rachid and Idboufker Noureddin Watchdog and Pathrater Approach

[5] T. Anantvalee and J. Wu, "A Survey on Intrusion Detection in MobileAd Hoc Networks," in *Wireless/Mobile Security*. New York: Springer-Verlag, 2008*Charlie Obimbo#1, Liliana Maria Arboleda-Cobo*

[6] L. Buttyan and J. P. Hubaux, *Security and Cooperation in WirelessNetworks*. Cambridge, U.K.: Cambridge Univ. Press, Aug. 2007.

[7] D. Dondi, A. Bertacchini, D. Brunelli, L. Larcher, and L. Benini, "Modelingand optimization of a solar energy harvester system for self-powered wireless sensor networks," *IEEE Trans. Ind. Electron.*, vol. 55, no. 7,pp. 2759–2766, Jul. 2008.

[8] V. C. Gungor and G. P. Hancke, "Industrial wireless sensor networks:Challenges, design principles, and technical approach," *IEEE Trans. Ind.Electron.*, vol. 56, no. 10, pp. 4258–4265, Oct. 2009.Cluster Based Intrusion Detection System for MANETS Nisha Dang Pooja Mittal

[9] Y. Hu, D. Johnson, and A. Perrig, "SEAD: Secure efficient distance vectorrouting for mobile wireless ad hoc networks," in *Proc. 4th IEEEWorkshopMobile Comput. Syst. Appl.*, 2002, pp. 3–13.

[10] Y. Hu, A. Perrig, and D. Johnson, "ARIADNE: A secure on-demand routingprotocol for ad hoc networks," in *Proc. 8th ACM Int. Conf. MobiCom*,Atlanta, GA, 2002, pp. 12–23.

[11] G. Jayakumar and G. Gopinath, "*Ad hoc* mobile wireless networks routingprotocol—A review," *J. Comput. Sci.*, vol. 3, no. 8, pp. 574–582,2007.

[12] D. Johnson and D. Maltz, "Dynamic Source Routing in *ad hoc* wirelessnetworks," in *Mobile Computing*. Norwell, MA: Kluwer, 1996, ch. 5,pp. 153–181.

