Journal of Computer Science 8 (6): 908-912, 2012
ISSN 1549-3636

Secure Packet Encryption and Key Exchange System in Mobile Ad hoc Nerwork

[1]Sudhakar Sengan and [2]S. Chenthur Pandian
[1]Department of Computer Science and Engineering,
Tagore Institute of Engineering and Technology, Salem, India
[2]Principal, Mahalingam College of Engineering and Technology, Pollachi, India

Abstract: Problem statement: Mobile Ad-hoc Network is infrastructureless network supported by no fixed trusted infrastructure. The packets had a chance to drop or hacked by eavesdropper during transmission. So, encryption method is required for sending and receiving packet in secret manner. **Approach:** In this approach, the block and key size had been increased by 256 bits. When compared to Rijndael algorithm, it was more secure and effective. To attain security goals like: authentication, integrity, non- repudation, privacy, a secret key was necessary to be shared between the sender and receiver. For data communication, we use MAC address for exchanging packet with encrypted key exchange system. **Results:** For encryption, In Rijndael algorithm the whole data had to be run twice but our proposed algorithm would encrypt the whole data and run once. The encryption was done with neighborhood key and with message specific key for the enhancement of security. **Conclusion:** In our algorithm, the time required to break an encryption scheme is excessive as the key size is larger. Here the security is focusing the application level. The forward and backward security is ensured with neighborhood and with message specific key for the route discovery.

Key words: Block cipher, symmetric encryption, random mobility, MANET neighborhood key, message specific key, key exchange, MANET routing

INTRODUCTION

Mobile Adhoc Network is basically defined as the infrastructureless network that make use of multiple hop radio relaying and it also capable to work in the absence of fixed infrastructure. Its nodes are accomplished to communicate directly to other nodes in the wireless channels. In this network, packet security is primarily concerned because channels are openly available and data propagate through the air (Chatterjee *et al*., 2011; Coppersmith, 1994; Islam *et al*., 2008; Kaosar *et al*., 2006; Liebeherr and Dong, 2007; Perkins and Bhagwat, 1994; Ruangchaijatupon and Krishnamurthy, 2001; Wang and Hu, 2009). All the nodes coordinate to enable communication among them as a group for routing and resource management in distributed manner. Every node in adhoc network act as a network host for transmitting and receiving data and as a network router for routing packets from other nodes. Since it significantly differs in many aspects, it needs an environment-specific and efficient–key management system.

In this network, the nodes made a mutual agreement on a exchanged secret key to secure nodes against the third parties. During key exchange mechanism, only Secret key is shared with authenticated neighbors to eliminate the re-key operations. In this proposed system, it enhances protection and handle unknown routing interms of the network security. The ID is added with the encrypted data before sending it in network. So the destination receives authorized packets from the source. If the ID matches then it will undergoe decryption by using neighborhood key as well as message key.

MATERIALS AND METHODS

Packet Encryption: For the purpose of security, this approach is to increase the block and key size into 256 bits, which can represent as a 8 by 4 matrix construction of byte. The block size will be increased by adding one column at a time.

Definition: The matrix construction represents the array of bytes which is known as state is shown in Fig 1. The array has four rows; the number of columns is indicated as Nb and is equal to the block length divided by 32. In this proposed system, the input and output are made as one- dimentional arrary of 8-bit bytes from 0 to 8*Nb-1. And also consider the cipher key as a one-dimensional array of 8-bit bytes numbered upwards from 0 to 8*Nk-1.

Corresponding Author: Sudhakar Sengan, Department of Computer Science and Engineering,
Tagore Institute and Engineering and Technology, Salem, India

a0, 0	a0, 1	a0, 2	a0, 3	a0, 4	a0, 5	a0, 6	a0, 7
a1, 0	a1, 1	a1, 2	a1, 3	a1, 4	a1, 5	a1, 6	a1, 7
a2, 0	a2, 1	a2, 2	a2, 3	a2, 4	a2, 5	a2, 6	a2, 7
a3, 0	a3, 1	a3, 2	a3, 3	a3, 4	a3, 5	a3, 6	a3, 7

(a)

$k_{0,0}$	$k_{0,1}$	$k_{0,2}$	k0, 3	k0, 4	k0, 5	k0, 6	k0, 7
k1, 0	k1, 1	k1, 2	k1, 3	k1, 4	k1, 5	k1, 6	k1, 7
k2, 0	k2, 1	k2, 2	k2, 3	k2, 4	k2, 5	k2, 6	k2, 7
k3, 0	k3, 1	k3, 2	k3, 3	k3, 4	k3, 5	k3, 6	k3, 7

(b)

Fig. 1: Matrix construction for Block size and Key size

The cipher input bytes are mapped onto the state bytes in the order a0,0, a1,0, a2,0, a3,0, a4,0, a0,1, a1,1, a2,1, a3,1, a4,1 ... and the bytes of the Cipher Key are mapped onto the array in the order k0,0, k1,0, k2,0, k3,0, k4,0, k0,1, k1,1, k2,1, k3,1, k4,1 ... At the end of the cipher operation, the cipher output is extracted from the state by taking the state bytes in the same order. Hence the one-dimensional index of a byte within a block is n and the two dimensional index is (i, j), we have: i= n mod 8; j=n/8;n= i+8* j.

The round transformation: The round transformation is consisting of four different transformations which is similar in Rijndael algorithm. The following pseudocode is representing the round transformation:

```
Round (State, RoundKey)
{
ByteSub (State);
ShiftRow (State);
MixColumn (State);
AddRoundKey (State, RoundKey);
}
```

The final round of the cipher is slightly different.

```
FinalRound (State, RoundKey)
{
ByteSub (State);
ShiftRow (State);
AddRoundKey (State, RoundKey);
}
```

In this code,the "functions" (Round, ByteSub, ShiftRow) operate on arrays to which pointers (State,RoundKey) are provided. In final round the mixcolumn phase is eliminated. The component transformations are specified in the following subsections.

The ByteSub transformation: The bytesub transformation is similar to Rijndaelbytesub transformation. For increasing the efficiency we have to use the Rijndael S-box.

The ShiftRow transformation: For encryption the first row is remain unchanged, 2nd Row is shifted 1 byte to the left, 3rd is 2 byte to the left 4th is 3 byte to the left. For decryption the operation is similar to that for encryption but in reverse direction.

The MixColumn transformation: In MixColumn, the columns of the State are considered as polynomials over GF(2^8) and multiplied modulo $x^8 + 1$ with a fixed polynomial c (x), given by $c(x) = 07x^7+06x^6 +05x^5 +04x^4 +03x^3 +01x^2 +01x + 02$. Polynomial is coprime to $x^8 + 1$ and therefore invertible.Let b (x) = c(x) ⊗ a (x):

$$\begin{pmatrix} b0 \\ b1 \\ b2 \\ b3 \end{pmatrix} = \begin{pmatrix} 02 & 07 & 06 & 05 & 04 & 03 & 01 & 01 \\ 01 & 02 & 07 & 06 & 05 & 04 & 03 & 01 \\ 01 & 01 & 02 & 07 & 06 & 05 & 04 & 03 \\ 03 & 01 & 01 & 02 & 07 & 06 & 05 & 04 \end{pmatrix} \begin{pmatrix} a0 \\ a1 \\ a2 \\ a2 \end{pmatrix}$$

The MixColumns ( ) transformation operates on the State column-by-column, treating each column as a eight-term polynomial.

The round key addition: In this operation, a Round Key is added to the State by a simple bitwise XOR. The Round Key is nothing but the Cipher Key by means of the key schedule. The transformation that consists of XORing a Round Key to the State is denoted by:

```
Add round key (state, round key)
l = round * Nb.
```

AddRoundKey( ) XORs each column of the State with a word from the key schedule.

Key schedule: The Round Keys are derived from the Cipher Key by means of the key schedule. This consists of two components: the Key Expansion and the Round Key Selection. The basic principle is as follows:The total number of Round Key bits is equal to the block size multiplied by the number of rounds plus 1. The Cipher Key is expanded into an Expanded Key. Round Keys are taken from this Expanded Key in the following way: the first Round.

Key expansion: The Expanded Key is a linear array of 8-byte. In c code this is given follow:

```
Keyexpansion (unsigned short int *key, unsigned
        short int *expandkey)
{
        unsigned short int temp[5],*temp1;
        inti,j;
        for (i = 0;i<8;i++){
        fo r(j = 0;j<8;j++){
        expandkey [i*8+j]=key[i*8+j];
        }
        }
        for (i=0;i<8;i++){
        for (j=0;j<8;j++)
        temp[j]= expandkey [(i-1)*8+j];
        if (i%8==0){
        temp1=subword (rotbyte (temp));
        for (j=0;j<8;j++)
        temp [j]=temp1[j];
        temp [0]=temp[0] ^ Rcon [i/8-1];
        }
        for (j=0;j<8;j++)
        expandkey [i*8+j]= expandkey [(i-
        8)*8+j]^temp[j];
        } }
```

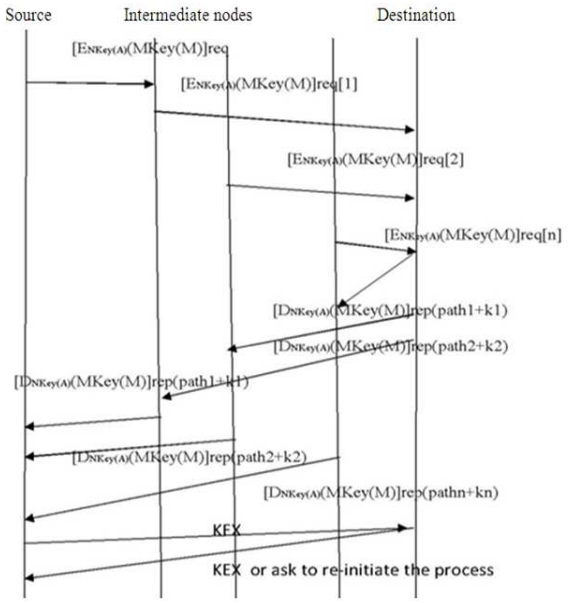

Fig. 2: Key generation and Key exchange Procedure

Table 1: Routing Table

Source		Destination	
Path	Key	Path	Key
P1	K1	P1	K1
P2	K2	P2	K2
P3	K3	P3	K3
…	…	…	…
P_N	K_N	P_n	K_n

Minimizing the number of rounds which is useful for power savings but it is not good for security purpose. In this algorithm key expanded upto14 rounds. More than seven rounds gives high security and could be used to save energy in some cases. The rotbyte and subbyte stand for rotation of byte in a single vector and substitute a byte using S-box.

Key generation and exchange: The source node generate RREQ which is used to find the destination path by using broadcasting as shown in Fig. 2. The packet from the source node is re-broadcasted in the neighboring node until it reaches the destination. A RREQ packet contains <Source_ID, Destination_ID, Path, TTL, other information>. Whenever the destination receives the RREQ packets it replies as RREP to the source. By this way it discovers a path by the source.Based upon the environment, different category of packets are generated according to the availability of route.Then the key is shared between source and destination.If it encounters the unavaliability of packet then the source will generate and broadcast KRREQ which is supported to construct the route and key.Thus the destination will generate keys and send it to the source as many times as it receive the RKREQ/KREQ packet.The path and key information is updated in the route table periodically is shown in Table 1:

$P_i \rightarrow$ = Is the path from source to destination found in ith RREP packet

$K_i \rightarrow$ = Is the key provided by the destination with ith RREP packet

$\int \rightarrow$ = Cryptographic key generation function

$E_{ID} \rightarrow$ = Ncryption of Source_ID using K_{SD}

Next the source will combine all paths and keys to generate a secret key:

$$K_{SD}\int(P_1,P_2,P_3,\ldots\ldots\ldots P_N;K_1,K_2,K_3,\ldots\ldots K_N)$$

$$E_{ID} = E_{KSD}(\text{SourceID})$$

Destination will perform following operations:

$$\int(p_1,p_2,p_3,\ldots\ldots p_n;k_1,k_2,k_3,\ldots\ldots k_n)$$

D_{ID} = D_{KSD}i(destination ID)

Pi→ = Is the path from source to destination found in ith RREP packet

Ki→ = Is the key provided by the destination with ith RREP packet

∫ - > = Cryptographic key generation function
DID- > = Encryption of Source_ID using KSD

Once the key is shared, it can be used for long period of time agreed by both party.

Encryption and decryption for key exchange system: Key Exchange System for the encryption and decryption algorithm is provided to secure the key.

Encryption algorithm: Each node has its own symmetric key called neighborhood key which is encrypted. Source Node A creates a Message Specific Key[MKey(M)]. Message is encrypted with Message Specific Key $[E_{Mkey(M)}(M)]$. Further the Message Specific Key is encrypted with A's neighborhood key $[E_{Nkey(A)}(Mkey(M)]$.Then the ID is appended with that encrypted message Node ID(destination). $[([E_{Nkey(A)}(Mkey(M)\ E_{Mkey(M)}(M)\ )$ Node ID(B)].

The following diagram shows how the message is encrypted and decrypted:

Encrypted Message at node B	$E_{Nkey(A)}$ (Mkey(M))	$E_{Mkey(M)}(M)$	B
If B is the intended recipt decrypt with A's neighborhood Key	$D_{Nkey(A)}(E_{Nkey(A)}$ (Mkey(M)))	$E_{Mkey(M)}(M)$	
Decrypt message with the obtaind	Mkey(M)	DNKey(M) $(E_{Mkey(M)}(M))$	

If the node is not intended node it will swicthover to another node:

If b is not the intended recipient, ID	$E_{Nkey(A)}$ (Mkey(M)	$E_{Nkey(M)}(M)$	C
Decrypt the message key with A's neighborhoodkey	$D_{Nkey(A)}$ $(E_{Nkey(A)}$ (Mkey(M)))	$E_{Mkey(M)}(M)$	C
Re-encrypt the message with B's neighborhoodkey	$D_{NKEY(B)}$ (Mkey(M)	$E_{Mkey(M)}(M)$	C

Neighborhood key exchange procedure: Key exchange with only neighborhood nodes is mainly to reduce the overhead occurance during sending and receiving data and to minimize crypto functions. This neighbhourhood scheme is easy and is carried out at handshake process between any pair of neighbors. Handshaking is the process of exchanging key between nodes and neighbhour. After that each node shared the secret key and HELLO messages are sent frequently to the node in the group.To send and receive the data between source and destination RREQ and RREP messages are used.

Decryption algorithm: At receiver side ID is used to verify the destination ie whether it matches or not,if it matches, then decryption will start and provide the plaintext messages.otherwise reencryption takes place with neighborhood key and transmits to its aunthenticated neighbor nodes.decyrption is the reverse process of encryption in which decrpt the message key as well as the node keys.the following specifies the decryption:

$[(D_{Nkey(A)}\ (Mkey(M)\ D_{Mkey\ (M)}(M))$ Node ID (source)].

RESULTS

The linear and differential cryptanalysis require more time than Rijndael to break our proposed cipher. The security is enhanced by increasing the key size as well as the block size. The encryption time is increased for the whole packets, if the number of bits in a packet increases.

For encryption in Rijndael algorithm the whole data has to be run twice but our proposed algorithm will encrypt the whole data and run once.The encryption is done with neighborhood key and with message specific key for the enhancement of security.

A comparison of various types of Encryption and Decryption algorithm is shown in Fig. 3. The simulation time of our algorithm is added to the various delay parameters in a random mobility model and the performance will become more efficient with the increase of block size.

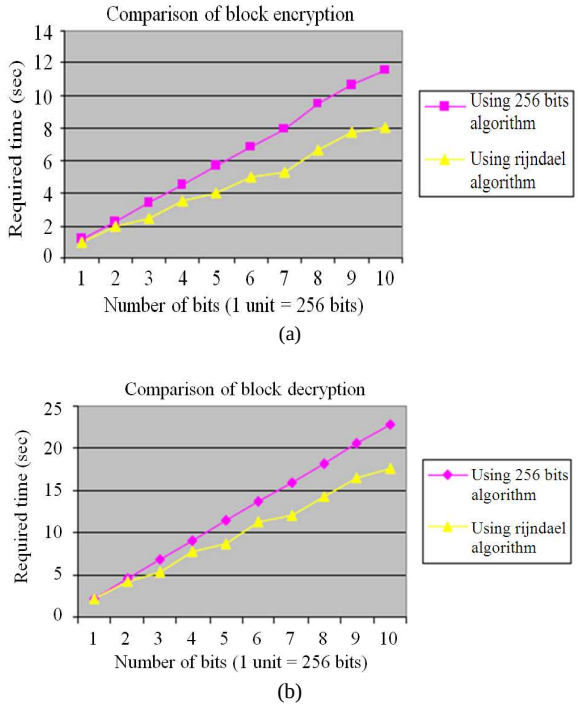

Fig. 3: Comparasion of block encryption and decryption

DISCUSSION

The secure communication is improved with secure key exchange system by using this algorithm. Here the block size as well as the key size is increased by 256 bits for the secure packet transmission.The result is compared with existing Rijndael algorithm for packet encryption and decryption. In this security scheme, the nodes will exchange data only with its authenticated neighbors.

CONCLUSION

In this proposed algorithm, the secure packet transmission has been done which is of most important. In our algorithm, the time required to break an encryption scheme is excessive as the key size is larger. Here the security is focused on the application level. The forward and backward security is ensured with neighborhood and with message specific key for the route discovery. During the route discovery for key sharing, less overhead is occurred. In MANET, the shared key encryption is done faster for the exchange of data.

REFERENCES

Chatterjee, D., J. Nath, S. Dasgupta and A. Nath, 2011. A new symmetric key cryptography algorithm using extended MSA method: DJSA symmetric key algorithm. Proceedings of the 2011 International Conference on Communication Systems and Network Technologies (CSNT), Jun. 3-5, IEEE Xplore Press, Katra, Jammu, pp: 89-94. DOI: 10.1109/CSNT.2011.25

Coppersmith, D., 1994. The Data Encryption Standard (DES) and its strength against attacks. IBM J. Res. Dev., 38: 243-250. DOI: 10.1147/rd.383.0243

Islam, M.N., M. Mia, M. Chowdhury and M.A. Matin, 2008. Effect of security increment to symmetric data encryption through AES methodology. Proceedings of the 9th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, Aug. 6-8, IEEE Xplore Press, Phuket, pp: 291-294. DOI: 10.1109/SNPD.2008.101

Kaosar, G., A.S.H. Mahmoud and T.R. Sheltami, 2006. Performance improvement of dynamic source routing protocol the mobility effect of nodes in cache management. Proceedings of the 2006 IFIP International Conference on Wireless and Optical Communications Networks, (WOCN' 06), IEEE Xplore Press, Bangalore, pp: 5-5. DOI: 10.1109/WOCN.2006.1666532

Liebeherr, J. and G. Dong, 2007. An overlay approach to data security in ad-hoc networks. Ad Hoc Netw., 5: 1055-1072. DOI: 10.1016/j.adhoc.2006.05.017

Perkins, C.E. and P. Bhagwat, 1994. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. ACM SIGCOMM Comput. Commun. Rev., 24: 234-244. DOI: 10.1145/190809.190336

Ruangchaijatupon, N. and P. Krishnamurthy, 2001. Encryption and power consumption in wireless LANs. University of Pittsburgh, Pittsburgh, PA.

Wang, Y. and M. Hu, 2009. Timing evaluation of the known cryptographic algorithms. Proceedings of the International Conference on Computational Intelligence and Security, Dec. 11-14, IEEE Xplore Press, Beijing, pp: 233-237. DOI 10.1109/CIS.2009.81