ASRAAM: A SECURED ROUTING ALGORITHM USING ANT AGENTS FOR MOBILE ADHOC NETWORKS

K.R.Ramkumar [1], Dr.C.S.Ravichandran [2]

[1]Associate Professor , Department of Computer Science and Engineering, Sri Venkateswara College of Engineering,Chennai,Tamilnadu(India)

[2]Professor and Head, Department of EEE, Sri Ramakrishna Engineering College,Coimbatore,Tamilnadu(India)

ABSTRACT

The wireless medium is uncovered which is vulnerable to different attacks and open to the elements of snooping. The eaves dropping are copious in wireless networks. The security algorithms that exist for standard wired networks are not applicable to MANET because of many factors like: wireless networks are not stable, topology information may vary from time to time, routing information changes frequently and limited resources availability. The extemporized nature of MANET allows a hacker inside the network since there is no centralized access point to monitor or control traffic. A cost effective, reliable, standard frame work for security is needed. This paper describes an enhanced version of security frame work for MANET. The proposed work leverages the advantages of existing standards like polynomial secret sharing that uses Lagrange's group theory, symmetric and asymmetric keys, security associations and key exchange algorithms. This security frame work prevents all type of men in middle attack, and provides an effective, fast and simple method of payload transaction between source and destination.

KeyWords - ***Secret Sharing, RSA, DES, Certificate, Ant Colony, Signature, MANET***

I. INTRODUCTION

The secured routing for ad hoc networks recently received more attention and a large number of solutions were suggested to provide security against various attacks. The two main types of attack are; active and passive attacks. The first type of attack can execute harmful functions such as packet discarding, routing malfunctioning and payload corruption and passive attacks, mainly can read network functions and collect information about network. Furthermore, malicious nodes [7] can be part of network and can cause attacks by making use of trapped nodes or by disrupting the normal routing operation or it can be an unauthorized node that aims to cause congestion, propagate incorrect routing information, prevent services or shut them down completely. These extortions exist because of intrinsically limited physical security of mobile ad hoc networks. Undeniably, it is easier to interrupt communications and infuse corrupted messages in the wireless communication medium than in an equivalent wired network. A close analysis of "routing security" [1][2] reveals that majority of works are suggesting either centralized dedicated certificate server or asymmetric key based authentication and encryption, which are neither cost effective nor suitable for wireless networks.

II. SECURITY ISSUES

The main instance of integrity outbreak is spoofing, whereby a malicious node overrides an authorized node, which is possible because of the absence of centralized authentication in the current ad hoc routing protocols. The immediate dominance of spoofing attack is the over all crumple in network topology information, trailed by network loops and partitioning of network . It is clear that security for MANET has to be taken into serious account at the beginning stages of design of ad hoc routing protocols. Here is the list of the basic building blocks of secured routing protocol[9][10] 1.Distributed Key Management; 2. IP based Key Generation, 3. Security Association (SA) as the replacement of dedicated Certificate Server (T) and 4. Polynomial Secret Sharing. The certificate-issue [5] to a legitimate node is not a simple process; a set of rules or policies has to be constituted for a successful Certificate issue to a genuine node

1.1 Security Association

The major issues to be taken into account are network traffic, congestion control, key strength and key length. The existing security mechanisms either suggests a dedicated trusted certificate issue server (T) to issue and revoke certificates or asymmetric key based encryption and decryption. But establishing a separate server is against the nature of MANET because practically a physical server cannot be carried to the places where MANETs are formed. Therefore this work requires the use of a Security association "Trust Model"[5] where the Security association (SA) is initially made up of some set of trusted nodes whereby the shared secret key is distributed among the nodes .The members of Security Association are responsible for the overall functioning of key issue and revocation, and it will be a hierarchical model.

1.2. Secret Sharing

The sharing of key is inherited from polynomial secret sharing algorithm which is illustrated in Shamir's secret sharing.When applying such a trust model, an entity is trusted if any k trusted entities approve so. This *k* trusted nodes are typically the neighbouring nodes of the entity. A locally trusted entity is globally accepted and a locally suspected entity is looked upon unreliable all over the network. In the suggested security architecture, each trusted node carries a certificate signed by the shared certificate-signing key P_{SK}, Nodes without valid certificates will be isolated, and their packets will not be forwarded to neighbours. Essentially, any node without a valid certificate is considered a potential intruder.

1.3. Certificate Request

A new node that enters into a MANET needs to get approval from *K* out of *N* members of security association. In turn the *K* value will be decided based upon the security density. The moment it receives *K* out of *N* secret shares, it can regenerate Polynomial Secret Key *(PSK) symmetric* through which it can participate in the network routing.An optimal value can be decided upon the level of threats and noise.

CertificateRequest (C_{req}, K, Cshare)

Input: C_{Req}: CertificateRequest,
nlist: neighbour list, *K*: threshold ,P_X: public key
Output: *Cshare*: Polynomial Secret Share
for n Є nlist do

$C_{req}IP=C_{req}[IPx,P_X]$
// C_{req} : the input variable
//C_{share}: output variable: Certificate share
C_{res}=sendRequest(C_{req}IP:input,C_{share}:output)
if C_{res} == SUCCESS then
CshareArray[S_{index}]=[C_{share}]IP_X-
$S_{index}=S_{index}+1$;
If S_{index} >= K then
P_{SK}= generatePSK(CshareArray)
else
continue
end

A new entity which enters into MANET, first broadcasts key request to all neighbours. If the neighbour is a member of SA, then it can issue or reject key shares, A node cannot participate in network when it is rejected by security association. More over it cannot see routing messages because all routing messages are encrypted by symmetric key.

Send request is a function which issues or denies the Polynomial secret share depending upon the policy profile rules.

---

sendRequest(C_{req},C_{share})

Input : ***Creq:*** **Certificate Request,**
IPx: : **IPaddress of node X**

Output : ***C_{share}*** **: Polynomial share of Certificate**
if isValid(C_{req}, IPx)==TRUE then
if policyFile criteria is satisfied then
C_{share}=[PSK_{Share}]IPx+
Record entry in KeyIssueTable
return SUCCESS
else
C_{share}=NULL
Record failure entry in EntryTable
return FAILURE

The function first verifies the validity of request and then completely checks for policy compatibility with policy File which is made for that particular network. The policyFile is already distributed among the members of Security Association (SA) .The policy file consists of all rules and regulations to issue certificate.The key share is encrypted by the public key of requesting node to avoid eavesdropping. The new legitimate node which asks for a share can decrypt the secret with its private key and can regenerate symmetric key after getting sufficient key shares.

III. FORWARD ANT GENERATION

Forward ant generation is a simple process of creating a route discovery process, general behaviour of forward ant is to find newer paths to a destination and to initialize the routing table. It consists of a unique id, current hop count, maximum hopcount (it can travel), source address, destination address, next hop and a stack which tracks the path it travelled. All values are initialized properly and it carries a certificate. The certificate has IP address of that node, public key, time of creation and expiry time of certificate .The pubic key is distributed to intermediate and destination nodes. This method is reactive one because public key is issued only to the nodes which participate in routing. This limits the possibilities of malicious node behaviour.

Forward ant generation

Input : ***f_{ant}:forward ant attributes, a_f:forward/backward*** *a_p:payload/empty,*
a_{id}: request id *a_{hc}:currenthopcount,* *a_{mhc}:maxhopcount* *a_{src}:sourceaddress*
a_{dst}:destination address,
a_{stack}:ant stack *n_{list}: neighbor list*
Output: $f_{ant}S$: secured forward ant
a_{id} = unique id : Ant Unique Id.
a_{mhc} : maximum hopcount
a_{dst} : destination address
a_f=1 : forward Ant
a_p=0 : nopayload
a_{hc}=0 : Current Hopcount
a_{nhop}=null : Nexthop value
a_{path}=null : Stack to record entries

f_{ant}=fant(a_{id},a_f,a_p,a_{id},a_{hc},a_{mhc},a_{timer},a_{src},a_{nhop},a_{dst}, a_{path})

//Certificate of a node consists
/ IP_X: Ip Address of X*
Px : Public key of node X
toc : Time of Creation
*exp : Expiry time */*
C_{node}=[IP_x,P_x,toc,exp]
$f_{ant}S$=[$f_{ant,}C_{node}$]p_{sk+}
Routediscovery (f_{anSt},n_{list});
end

The forward ant from node X is combined with the certificate of same node and encrypted by the *PSK* (Polynomial Secret key)

A→ *f_{ant}s : [f_{ant} ,C_{node}]P_{SK+} ***(2)***

Here the forward ant is encrypted by PSK ensures that all types of passive attacks on routing are completely evaded. Now route discovery process starts with a secured forward ant.

IV. KEY MANAGEMENT

4.1. Route Discovery

An intermediate node decrypts forward ant with the help of public shared key (PSK) and it checks for destination address. The nodes without *PSK* cannot decrypt forward ants and cannot understand to modify the forward ant attributes. The generation of backward ant algorithm is invoked when forward ant reaches destination and unicast function is invoked.

Routediscovery ($f_{ant}S$,n_{list})

Input :*f_{ant}* :forward Ant, *n_{list}* :neighbor list
Output: Route discovery and table updating
f_{ant}=[$f_{ant}S$]p_{sk-}
if isNew(f_{ant}.a_{id}) then
if f_{ant}.a_{hc}<=f_{ant}.a_{mhc} then
if f_{ant}.a_{dst} == currentNodeID
C_{dst}=[IP_{dst},P_{dst},toc,exp]
Converttobackwardant(f_{ant},C_{dst})

else if $f_{ant}.a_{dst}$***!=currentNodeID and***
$f_{ant}S=[f_{ant}]p_{sk+}$
Routediscovery($f_{ant}S,n_{list}$***)***
else
discard(f_{ant}***)***
end

4.2 Unicast

The Intermediate node decrypts backward ant with the help of *PSK* for performing routing function . The intended source node decrypts backward ant to extract certificate of destination node and stores in to node table for future communication.

---

unicast($b_{ant}S$**)**

Input : $b_{ant}S$: backward ant
Output: *m*:updated path
$b_{ant}=[b_{ant}S]p_{sk-}$
$b_{ant}.a_f=0, b_{ant}.a_r=0, b_{ant}.a_p=0$ ***if*** $b_{ant}.a_{hc}<b_{ant}.a_{mhc}$ ***then***
if $b_{ant}.a_{dst}$***==currentNodeIP then***
KeyIssue(U_{ant}, C_{dst}***)***
else if $b_{ant}.a_{dst}$***!=currentNodeIP then***
pickup next node from $b_{ant}.a_{path}$ ***and***
$b_{ant}.a_{hc}=b_{ant}.a_{hc}+1$
$b_{ant}S=[b_{ant}]p_{sk+}$
unicast($b_{ant}S$***)***
else
discard(b_{ant}***)***
end

4.3. Key Issue

The symmetric key used for payload transaction is first encrypted by private key of source and then by public key of destination to ensure security and to avoid non repudiation. The encrypted symmetric key is post fixed with a unicast forward ant and the entire ant packet is encrypted with the help of PSK.The forward ant used here is to carry the routing message. The intermediate nodes can open forward ant to send it to destionation. Moreover the forward ant depicted here is unicast in nature which follows a preassigned route to reach destination.

KeyIssue(U_{ant}, N_{dst}***)***

Input: U_{ant} *,unicast ant to issue certificate*
Output: Session symmetric key issue.
//start new unicast request from $b_{ant}.a_{src}$ *to* $b_{ant}.a_{dst}$ *for payload transaction*
f_{ant}***=converttoforward(***b_{ant}***)***
$f_{ant}s=[[[SessSymm_{key}]P_{src-}]P_{dst+}, f_{ant}]P_{sk+}$
if keySend($f_{ant}S, N_{dst}$***)==SUCCESS then***
PayloadSec=[[Payload] SessSymm$_{key+}$***,bant]Psk+***
PayloadSend(PayloadSec)
else
invoke exception handling function
end

The intended recipient extracts and decrypts symmetric key with both public key of source and then, private key of recipient.

The key distribution is a vast process. With the help of the above specified algorithms, a seasoned symmetric key which could be used for one transaction is safely given to destination node.

PayloadSend(PayloadSec)

Input : Payload
Output : Payload delivery

b_{ant}=[PayloadSec,b_{ant}]Psk-
if b_{ant}.dst=currentNodeIP then
    Pload=[PayloadSec]S_{key}-
    save Pload
else if b_{ant}.dst!=currentNodeIP then
    unicast(PayloadSec,b_{ant}S)

V. SECURE DATA TRANSACTION

After receiving a symmetric key, both source and destination are becoming partners for that session. They both can communicate by encrypting with their symmetric key. The life time of that key depends upon network conditions. Thus, the above specified algorithms ensure a secured, authenticated and reliable data transmission between source and destination nodes with optimal security mechanisms.

VI. SIMULATION RESULTS

Swans(Scalable Wireless Adhoc Networks Simulator) is used to implement the algorithms, 90 mobile nodes has been plotted in 1000m^2 area using Randomway point mobility model.Totally ten key shares is distributed among ten random nodes. The node movement speed is increased from 1 m/s(meter per second) to 9 m/s(meters per second) ,First, 3 out of 10 shares has to be collected, the Key share identication time ranges from 18 to 20 seconds at different speeds. it could be seen that maximum 160 seconds is taken to collect 6 keys from different nodes. in practice the life time of a MANET is too short so here it is limited with 6 key shares but there is no limit for numbers and could be implemented based upon the security level needed .

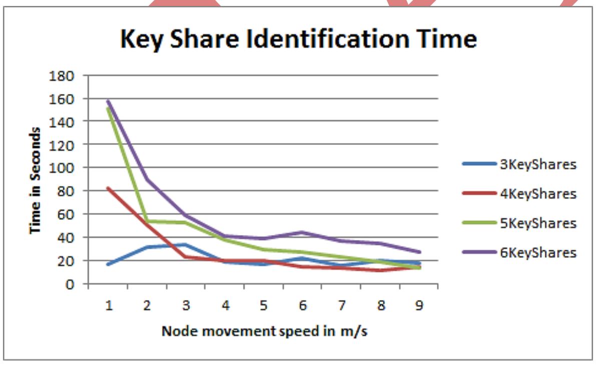
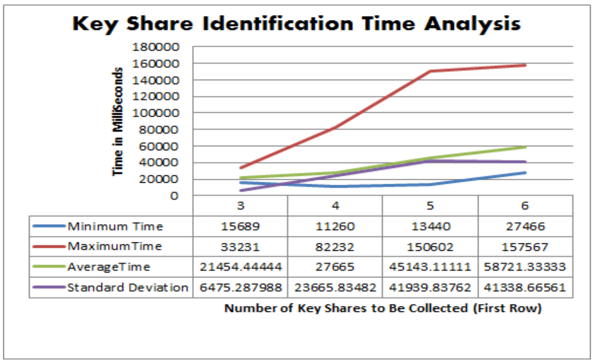


Fig.1. Key Share Time Analysis



	3	4	5	6
Minimum Time	15689	11260	13440	27466
MaximumTime	33231	82232	150602	157567
AverageTime	21454.44444	27665	45143.11111	58721.33333
Standard Deviation	6475.287988	23665.83482	41939.83762	41338.66561

Fig.2. Key Share Identification Time Analysis

The above Figure (1 & 2) says that on an average a new node takes 58 seconds to collect key shares from 6 nodes when it moves on the speed of 9 m/s. The key share algorithm is a polynomial solvable problem, the main perception here is that a separate key server is against the nature of MANETs and it could be replaced by PSK(Polynomial Secret Share algorithm). To test the functionalities of assymetric keys in mobile devices we have used J2ME, netbeans 6.5 and Bluetooth programming , The propogation delay of a Bluetooth device is 1 ms, ie) the hop time from one device to another device is 1 ms.

First the RSA encryption algorithm is implemented in Swans- simulatore for a single hop count network with various key lengths range from 128 bits to 512 bits, The maximum time taken to encrypt a "*Hello*" message is 250 milliseconds,the same has been implemented in a mobile device with 1 GHz speed of (Nokia 500) the maximum time taken is 2357s for 512 bits key. Another low configured mobile device (Nokia C101) has taken 16424 milliseconds for the same program. The Figure 3 shows the time variation in different devices.

The standard assymetric encryption algorithm is ECC that also implemented in all configurations. The maximum time taken is 600 ms which is much lower than RSA ,which is shown in Figure 4.

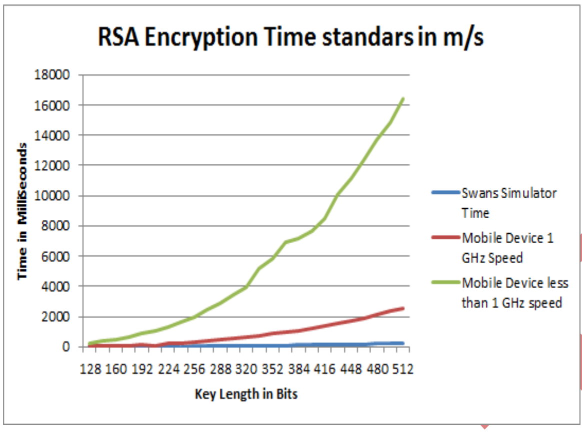


Fig. 3. RSA Encryption Time Standards

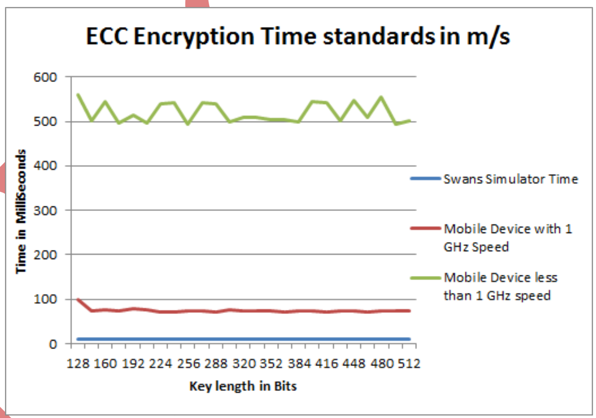


Fig. 4. ECC Encryption time standards

The beacons messages are encrypted with different key sizes and comparsions shows that ECC is suitable for mobile devices in Figure 5 and Figure 6.

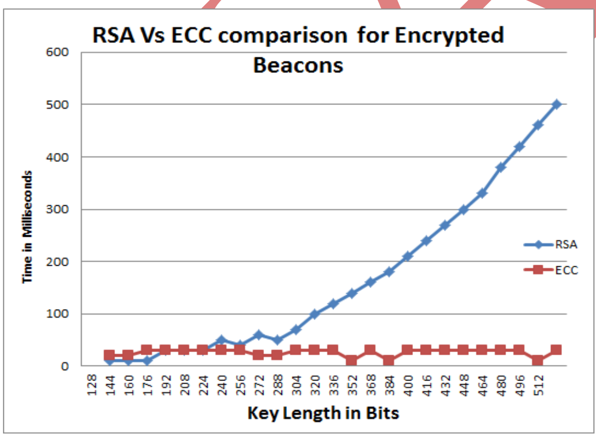


Fig. 5 Break Even Point for Encryption

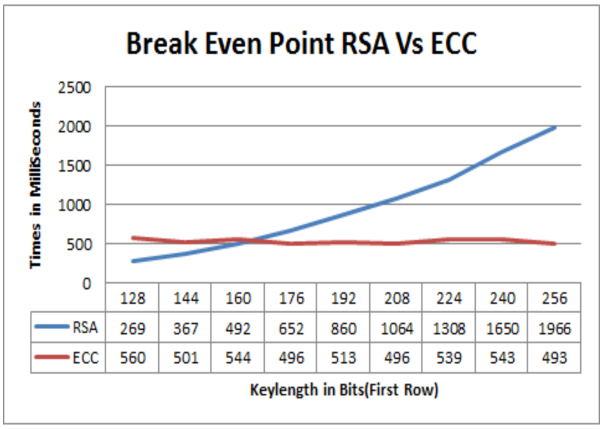


Fig. 6 Break Even Point for Decryption

The break even points are given in following graphs , the first simulation done in Nokia C101(< 1GHz) , it clearly says that RSA outperformce ECC untill 160 Bit key length, there after the time delay grows fast .

The second simulation done in Nokia 500(1GHz) , it says that RSA outperformce ECC untill 208 Bit key length, there after the time delay grows fast , this is depicted in Figure 7

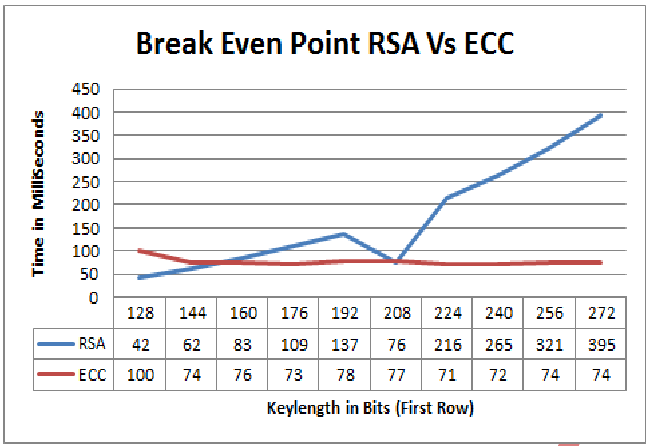


Fig.7. Break Even Point in RealMobile device

The memory space requirement is very high for RSA than ECC .
The conclusions are

i) Polynomial secret sharing e progress is polynomial solvable.

ii) All routing messges are ecrypted by a symmetric key which is generated from PSK.

iii) ECC outperforms RSA which is used to encrypt session symmetric keys.

a) RSA is better than ECC if the key length is less than 208 bits

iv) Session symmetric keys are used to encrypt and decrypt payloads.

The above mentioned standard security framework is optimal which could replace a dedicated key server and leverages all privillage of existing standards.

VII. CONCLUSIONS AND FUTUREWORK

The Security Association (SA) formation requires lot of policies and issues at initial period where the policies for SA yet to be standardized. The process of issuing keys, hand over responsibilities to other trusted nodes ,monitoring the behaviour of individual nodes, certificate issue , revocation, fixing expiry time for certificates and selecting *k* out of *n* nodes are still to be explored to large extend. The other factors like resource utilization, power consumption, key lengths, key strength, and session key based symmetric key generation also should be taken into account while framing a standard security framework.

REFERENCES

[1]Distributed Private Key Generation for Identity Based Cryptosystems in Ad Hoc Networks Chan, A.C.-F. Wireless Communications Letters, IEEE Volume: 1 , Issue: 1 Digital Object Identifier: 0.1109/WCL.2012.120211.110130 Publication Year: 2012 , Page(s): 46 - 48 IEEE Journals & Magazines

[2]Evaluating Trust in Ad Hoc Network Routing by Induction of Decision Trees Sirotheau Serique, L.F.; de Sousa, R.T. Latin America Transactions, IEEE (Revista IEEE America Latina) Volume: 10 , Issue: 1 Digital Object Identifier: 10.1109/TLA.2012.6142481 Publication Year: 2012 , Page(s): 1332 - 1343 IEEE Journals & Magazines

[3] A Survey on Trust Management for Mobile Ad Hoc Networks Jin-Hee Cho, *Member, IEEE*, Ananthram Swami, *Fellow, IEEE*, and Ing-Ray Chen, *Member,* IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 13, NO. 4, FORTH QUARTER 2011

[4]Optimal Combined Intrusion Detection and Biometric-Based Continuous Authentication in High Security Mobile Ad Hoc Networks Jie Liu, F. Richard Yu, *IEEE*, Chung-Horng Lung, and Helen Tang IEEE transactions on wireless communications, vol. 8, no. 2, february 2009

[5] J. H. Cho and A. Swami, "Towards Trust-based Cognitive Networks: A Survey of Trust Management for Mobile Ad Hoc Networks," *14th Int'l Command and Control Research and Technology Symposium*,Washington D.C. 15-17 June 2009.

[6]L. Abusalah, A. Khokhar, and M. Guizani, "A Survey of Secure Mobile Ad Hoc Routing Protocols," *IEEE Commun. Surveys and Tutorials*, vol.19, no. 4, pp.78-93, 2008.

[7]M. A. Ayachi, C. Bidan, T. Abbes, and A. Bouhoula, Misbehavior Detection Using Implicit Trust Relations in the AODV Routing Protocol,"*2009 Int'l Conf. on Computational Science and Engineering*,Vancouver, Canada, vol. 2, 29-31 Aug. 2009, pp. 802-808.

[8]Boukerche and Y. Ren, "A Security Management Scheme using a Novel Computational Reputation Model for Wireless and Mobile AdHoc Networks," *Proc. Int'l Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems*, Vancouver, British Columbia,Canada, pp. 88-95, 2008.

[9]A survey of routing attacks in Mobile ad hoc networks Bounpadith kannhavong,hidehisa nakayama, yoshiaki nemoto, and nei kato, Tohoku university Abbas jamalipour, university of Sydney IEEE Wireless Communications • October 2007

[10]H. Li and M. Singhal, "Trust Management in Distributed Systems," *Computers*, vol. 40, no.2, Feb. 2007, pp. 45-53. E.

[11]Aivaloglou, S. Gritxalis, and C. Skianis, "Trust Establishment in Ad Hoc and Sensor Networks," *Proc. 1st Int'l Workshop on Critical Information Infrastructure Security, Lecture Notes in Computer Science*,vol. 4347, pp. 179-192, Samos, Greece, 31 Aug. – 1 Sep. 2006, Springer.

[12]An Efficient Key Pre distribution Scheme forAd Hoc Network Security Mahalingam Ramkumar and Nasir Memon IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 23, NO. 3, MARCH 2005