Reducing Communication Overhead For Authentication Using Self Contained Public Key Management Scheme In MANET

Gundala Swathi [#1], Ponugupati Sujala [*2] DR.R.saravanan [*3]
SITE Department, VIT University, Vellore, Tamil Nadu, India
[1] gundalaswathi@vit.ac.in

***Abstract*--- In the past few years, we have seen a rapid increase in important application fields of Mobile ad-hoc networks (MANET). Hence, many industrial and academic researchers have been conducted. Because, these applications are closely related to human beings and their physical environment, the usage of MANET on a large scale depends on whether we can provide proper dependability. Particularly, security is the most important issue in MANET because of the limitation in resources. Management of keys is the most necessary activity for providing security for the network. MANETs have resource limitations. Thus using public-key based solutions is not feasible. Thus, we use symmetric key establishment. In this paper, a key management scheme which is self contained and public is represented. This scheme achieves near zero communication overhead while providing security services. Cryptographic keys in small numbers are inputted at all nodes prior to the deployment in network. Mathematical Combinations of pairs of keys, both public and private is used for better utilization of storage space. This means a combination of more than one key pair is utilized by nodes for the encryption and the decryption of messages. A secure communication algorithm for forwarding the packets in MANET is also proposed.**

Keywords-cryptographic keys, communication overhead, deterministic scheme,Keyword1, network resiliency

I.INTRODUCTION

A mobile ad-hoc network (MANET) has no infrastructure and is a wireless network which has self-configuring mobile devices. In Latin Ad-hoc means “for the purpose”. In MANET, all devices move freely in any direction, and keep changing their links in the network. In their top of a link layer network, they have a routable networking environment. Key management is very important to provide security to promote secure communications. Because there are limitations in storage, energy and computations, public-key based solutions cannot be used. Thus we use Symmetric key establishment in MANETs. As MANET is infrastureless, third party cannot be used to distribute secret keys pair wise between nodes. Hence ,we use key pre-distribution method. Two categories of these schemes exist: deterministic schemes providing secure connectivity totally and probabilistic schemes where there is no certainty of a secure connection. For secure connection, shared keys are required.

To assess the performance of schemes described above, we have five evaluation criteria: resiliency of the network, secure and complete connectivity coverage, overhead in communication, complexity of computation and overhead of storage. Here, we try to increase resiliency of the key management schemes which provides secure connectivity but does not add new communication or storage overheads unlike existing research works. Thus, we propose a key management scheme which is self contained and public, which helps in achieving, for authentication, zero overhead while communicating. In this scheme, Cryptographic keys in small numbers are inputted at all nodes prior to the deployment in network. Mathematical Combinations of pairs of keys, both public and private is used for better utilization of storage space. This means a combination of more than one key pair is utilized by nodes for the encryption and the decryption of messages.

II.RELATED WORKS: KEY MANAGEMENT SCHEMES FOR MANET

Several possible solutions have been provided for managing the key management issues in MANETs. Probabilistic scheme and Deterministic scheme are two categories of symmetric key management schemes.

Because of shared keys existence, connectivity is not always secure in probabilistic schemes. In deterministic schemes, total security is provided in connectivity because of direct secure link.

A. Probabilistic schemes

In this scheme, two of the each neighbouring nodes establish a link which is secure. If this is not possible, then a path which is secure is created.

In [3], Gligor and Eschenauer proposed a Random Key Pre-distribution scheme which is basic called RKP. Here, a ring of keys, rk is selected randomly for every node from large pool Po of keys. Each node 'x' exchanges key identifiers with neighbour y. Thus, nodes x and y can identify the keys they share. A secure link can be established if neighbours share at least one key. Otherwise secure paths are created. But this creates storage overhead. The earliest of key pre distribution schemes involved simple allocation strategies, by Wheeler and Gong[4]. Later Piper and Mitchell[5,6] considered more complex strategies. Du et al. projected in [7] methods in enhancing connectivity which is secure and resiliency of RKP schemes. In [8], Castelluccia and Spognardi basic scheme is applied in multi-stage for wireless sensor networks. Here, to provide connectivity in network, new nodes are deployed from time to time. In [9], RPS(random preloaded subsets) are used.

RPS is defined by parameters k and P. Indexed set of secrets $S=\{k_1,k_2,\ldots,k_p\}$ are used. Each node with unique ID has a subset k of P secrets. Based on a random one way function specific k key are allocated. Thus, a function $F(X)=\{X_1,X_2,..X_k\}$ is assigned to node X. Correspondingly secrets $S_X=\{K_{x1},K_{x2},\ldots,K_{xk}\}$ are provided. $m=k^2/P$ indexes are shared on an average between two nodes. $F(X) \cap F(Y)$ provides the information about shared indexes between these nodes. By hashing together the shared secrets we evaluate k_{xy}, the secret used for the mutual authentication of X and Y. However, an attacker can determine k_{xy} if all secrets from n nodes(say nodes with IDs $M_1,M_2,\ldots M_n$) are known if $F(X) \cap F(Y) \in \{F(M_1) \cup F(M_2) \cup \ldots \cup F(M_n)\}$. The probability p(n) that the attacker can discover k_{xy} is

$p(n)=(1-z(1-z)^n)^k$, where $z=k/P$

and the optional choice of z that minimizes k is $z=1/(n+1)$. Such a scheme which is n-secure with probability 1-p, is referred to as (n,p) secure.

B.Deterministic Schemes

Deterministic schemes help in ensuring that every node establishes a pair-wise key with all its neighbours. Ample number of solutions has been proposed.

Yener and Camtepe, in [10] proposed a pool based deterministic key pre-distribution scheme. Combinatorial design is used instead of selecting keys randomly from a pool of keys. Here, every two key subsets have only one key common to them. Thus total secure connectivity is provided. In [11], authors use the Çamtepe scheme for key management in grid group WSN. The deployment area is divided into square regions. Symmetric Balanced Incomplete Block Design based key pre-distribution is used in each region for intra-region secure communications. Special nodes called agents help in inter-region communications. To avoid key identifier exchanges, all nodes and keys are indexed and a maooing is done between them. In [12,13], Perrig et al proposed a security suite having two blocks: (i)SNEP which authenticates data, provides confidentiality and data freshness and (ii) mTESLA which provides authenticated broadcast.

III. EVALUATION METRICS

In this paper, five metrics are used to evaluate the performance of MANET key management schemes:

A. Network resiliency against node capture(Resilience to break-ins):

Due to the limitation in resources, the nodes can be compromised leaking secret information from its memory. Such attacks will compromise both internal and external links. Thus, network resiliency is essential.

B.Secure Connectivity Coverage:

Since without appropriate security provisions, the Mobile adhoc Networks are subjected to attacks like network traffic, replay transmissions, manipulate packet headers and redirect routing messages. In order to overcome these problems Trust based Packet Forwarding Scheme is suggested by Kartheesan and Srivatsa [14] for mobile ad hoc network system that provides the capability to express network requirements.

C.Computation complexity:

In the approach that we propose no additional computation overheads occur.

D.Communication overhead:

Jitter is applied to control packet transmissions in MANETs.

E.Memory footprint:

Because sensor nodes are small, memory resources are vital. We need to manage the memory needed to store keys.

IV. A SCALABLE METHOD OF CRYPTOGRAPHIC KEY MANAGEMENT SCHEME

We propose a self-contained public key management scheme, where all necessary cryptographic keys are stored at individual nodes before nodes are deployed in the incident area. Thus, there is no exchange of certificates while communicating. Hence, there is nearly zero communication ovehead while authenticating. Only ID of each other needs to be known to call public keys of one another. The required storage space for traditional key

management schemes is of O(n) order. But these days networks are huge. Hence, we present a Scalable Method Of Cryptographic Key management scheme, which has a required storage space of O(log n) order.

We then use the public key cryptography as follows: Each node possesses a unique combination of private keys, and knows all public keys. The private key combination pattern and node ID are linked. Thus, if sender X wants to send a message to receiver Y, X first needs Y's ID to learn about Y's private keys. Then X will encrypt the message with the public key set that corresponds to the private keys owned by Y. This scheme is evaluated based on the above proposed metrics and is proven to work satisfactorily even when small number of nodes are compromised.

V. PROBLEM STATEMENT

Here, let us assume a group of people, who want to communicate securely in pairs. A trusted server maintains the set of private-public key pairs offline. Each key pair consists of two mathematically related keys. The i-th key pair is represented by (k_{ipriv}, k_{ipub}).For secure communication, public keys are known to all and distinct private key sets are provided to each node. Let A(Kpriv) represent a subset of private keys held by A, and A(Kpub) denote A's corresponding public key subset. If B wants to send a secret message to A, he needs to know A(Kpub), where A(Kpriv) is unique. Public keys A(Kpub) is used to encrypt the message to be sent by B to A. The message can be opened only by A, who has the private key set A(Kpriv), but others do not.

Take a group with 10 users for example. Here, we require 5 public-private key pairs. They are ($k_1$priv; $k_1$pub), ($k_2$priv; $k_2$pub), ($k_3$priv; $k_3$pub), ($k_4$priv; $k_4$pub), ($k_5$priv; $k_5$pub). Thus each user has 2 private keys and 5 public keys. The unique private key set for every user is represented as:

K_1priv={$k_1$priv; $k_2$priv}, K_2priv={$k_1$priv; $k_3$priv}, K_3priv={$k_1$priv; $k_4$priv}, K_4priv={$k_1$priv; $k_5$priv}, K_5priv={$k_2$priv; $k_3$priv}, K_6priv={$k_2$priv; $k_4$priv}, K_7priv={$k_2$priv; $k_5$priv}, K_8priv={$k_3$priv; $k_4$priv}, K_9priv={$k_3$priv; $k_5$priv},K_{10}priv={$k_4$priv; $k_5$priv}.

Thus, we know that

#Each person has unique predetermined set of private keys.

#A message is encrypted by multiple public keys, and it can only be read by a user who has the corresponding private keys. For example, if user 3 encrypts a message msg by public keys $k_3$pub and $k_4$pub as Encp(Encp(m; $k_3$pub); $k_4$pub),then only user 8 can decrypt it with private keys $k_3$priv and $k_4$priv.

In traditional public management schemes, each user holds one public-private key pair. Therefore, a user should store n public keys and 1 private. But here a user only stores 7 keys (5 public keys and 2 private keys), which is lesser than 11 keys (10 public keys and 1 private keys) in traditional schemes. Here the total number of keys held by each user is approximately O(log(n)), but it is O(n) under traditional key management schemes.

TABLE I
Terms used in algorithm

K	A Key pool: a set of public-private key pairs
K_p^{priv}	A set of private keys held by user *p*
K_p^{pub}	A set of public keys corresponding to K_p^{priv}
K_p	A set of public-private key pairs held by user *p*.
M	Memory size for key storage
r	Number of distinct key pairs $r = \|K\|$
s	Number of private keys held by each user under isometric key allocation, $s = \|K1\| = \|K2\| = \ldots = \|Kn\|$
$K_c(p)$	Expected number of disclosed keys when *p* nodes are broken in
$K_v(p)$	Maximum number of disclosed keys when *p* nodes are broken in
$V_p(r,s)$	Vulnerability metrics as *p* nodes are broken in.
n	Total number of nodes in the network

A. Definitions

Definition 1: A key pool K ={($k_p$pub,$k_p$priv)|$\forall$ i$\leq$a}, where ($k_p$pub,$k_p$priv) is p^{th} public-private key pair, and r = |K| gives distinct key pairs number. K_{Ypriv} and K_{Ypub} stand for a set of private keys and public keys held by user Y.

Definition 2: A key allocation KA: 2^K ->Y , maps the key pairs in K to a set of users in Y , so that y$\in$V is assigned a subset of key pairs K_p ($K_p \subseteq$ K)). To provide secure communication between nodes p and q, we have

$\forall p \forall q$ Kp$\not\subseteq$ Kq(the same as $K_p$priv$\not\subseteq$ $K_q$priv) and $K_q \not\subseteq K_p$(the same as $K_q$priv$\not\subseteq$ $K_p$priv), iff p $\neq$ q. The key allocation is valid if this condition holds.

Definition 3: If $|K_1|= |K_2|=\ldots= |K_n|= s$, key allocation is isometric; else non-isometric.

Definition 4: User p and q conflict in key assignment, if either $K_p$priv$\subseteq K_q$priv or $K_q$priv$\subseteq K_p$priv.

B. Objectives

To provide good performance of the key management scheme, all the evaluation metrics must be satisfied. Thus, the following objectives neeed to be fulfilled.

Objective 1- Memory Efficiency: Given a network of size n, we need to find a key pool K and a key allocation KA to achieve

min $|K|+\max\{p \in Y\}|K_p\text{priv}|$

where$K_p \not\subseteq K_q$ and $K_p \not\supseteq K_q \forall p \neq q$

This is equation (1) where $|K_p\text{priv}|= |K_p|$ provides the number of private keys stored at node p. |K| is the total number of public keys stored at each node as all public keys are stored in every node. Therefore, $|K|+|K_p\text{priv}|$ is the amount of memory required to store the public keys and private keys for secure communications at node p.

Objective 2- Computational Complexity: To simplify security operation, each person uses few public keys and private keys to encrypt and decrypt messages respectively. Therefore, we represent

min max $\{p \in Y\}|K_p\text{priv}|$

where $K_p \not\subseteq K_q$ and $K_p \not\supseteq K_q$ ($\forall p \neq q$) and $|K| \leq M$

where M is the amount of memory used for storing keys at every node.This is represented as equation (2).

Proposition 1: Based on Objectives 1 and 2, isometric allocation has better performance than non-isometric.

Thus , isometric key allocation is used in this paper.

Objective 3- Resilience Requirement: We denote r =|K| and s = $|K_p|$ = $|K_p\text{priv}|$ in isometric allocation of keys.

Each user will carry only s private keys and r public keys in isometric key allocation, where s << r <<(r + s) << n. If a node is exposed, then all its keys(private and public) are exposed. Thus, when p nodes are broken the averagely $C(k_c(p), s)$ distinct key-sets get affected. In worst-case it is $C(k_v(p), s)$. C(r,s) means r choose s.

We represent vulnerability metric as $V_p(r, s)$, which is percentage of compromised communications when p nodes are exposed or broken. On average $V_p(r, s)$ is $\{C(k_c(p), s)\}/C(r,s)$. In worst case $\{C(k_v(p),s)\}/C(r,s)$. We deduce $k_c(p)$ and $k_v(p)$ in Proposition2. For resilience on breakage of p nodes

$V_p(r, s) = C(\lfloor k_c(p) \rfloor, s)/C(r,s) \leq P$

where P is the upper resilience bound. This is equation (3). The floor of $k_c(p)$ is taken if its not an integer.

Proposition 2: If r is the number of key pairs in MANET and s number of private keys are present for every node and p nodes are broken in, then on an $k_c(p) = r-(r-s)((r-s)/r)^{p-1}$ keys will be disclosed.

C. Key allocation algorithm

Due to Proposition 1, we use isometric key allocation algorithms to fulfill the objectives.

1) Derivation of r and s:

Optimization of the objectives: The encryption and decryption complexity is determined by s. Thus s should be smaller. r can be any integer value . Extreme case is r = n and s = 1,3 where only one key copy is kept by every person. Network size is n here. For effective storage, r/n should be small. But for being resilient r/s should be large. Thus, its conflicting. But the following algorithm solves both these objectives.

Algorithm 1:

(1) Initialize p = 2.

While (C(p,(p/2) <n)

do {p = p + 1};

r = p; s = [p/2];

(2) While (C(r, s - 1) > n)

do {s = s – 1};

(3) While (C(r + 1, s - 1) > n)

do {r = r + 1, s = s – 1};

(4) While (Equation (3) is unsatisfied)

do {

if(C(r + 1; s - 1) > n)

{r = r + 1; s = s – 1}

Else

{r = r + 1}

};

(5) |K| = r and $|K_q|$= s.

Step (1) here calculates the minimum required number of memory slots to store public keys for securely communicating between n nodes. Step (2) helps in increasing memory efficiency. Step (3) decreases computational complexity without changing the memory requirements. Step (4) ensures resiliency. If r and s do not ensure resiliency, then r is increased or simultaneously s is decreased and r is increased. Thus r/n is increased by 1/n and r/s is increased by 1/s or (s+1)/{s(s-1)}.

Meeting key storage constraint: Let MS denote the maximum number of memory slots required to store keys of n nodes. We need to optimize the usage of memory slots and achieve excellent resiliency denoted by Equation (3). Hence, we use Algorithm 2.

Algorithm 2:

(1) Let r = $\lceil 2MS/3 \rceil$, s = $\lfloor MS/3 \rfloor$;

(2) While (C(r + 1, s -1) > n)

do {r = r + 1, s = s – 1};

(3) Then |K| = r and $|K_p|$ = s.

Algorithm 1 is used to achieve resiliency and minimize memory slots usage. In Algorithm 2 for a specified storage space we try to achieve the most feasible resilience.

2) Key Allocation:

For a given network size n, we determined r and s. To support secure communication pair-wise $K_p \nsubseteq K_q$ and $K_p \nsupseteq K_q$, where n = C(r,s). Assume that only one private key can be assigned to at most x nodes, so s*n =r*x. Hence, x =(s/r)n = (r/s)C(r, s). s private keys are randomly assigned where a given key can be assigned to at most (s/r)C(r, s) nodes. Otherwise, the key allocation is invalid. Algorithm 3 provides the procedure to assign a subset private keys to a node. Note that even a very small r and s can support a very large network. E.g., if we do not take into account the resiliency of the network, r = 20, s = 4, can have a network size of 4845.

Algorithm 3:

(1) For the p-th node ($p \leq C(r, s)$), s distinct private keys are randomly selected to create a subset of keys, where either of these s private keys has been assigned more than (s/r) C(r, s) times;

(2) If (the generated key set = an assigned keyset) Adjust key by key in the generated key set to get unassigned key set;

(3) Assign the generated key set to node p.

VI. PACKET FORWARDING SCHEME FOR SECURE COMMUNICATION

For providing security not only to data, but also for routing information, we calculate the trust indexes of the nodes and the route is selected according to the trust value which improves integrity.

We introduce a the scheme for the purpose of data security. We calculate the trust index using the Algorithm 1 and the Route is selected based on the trust Index of all nodes.

A. Trust Index Calculation

Let Z = $\{N_1,N_2,\ldots.N_n\}$ be the network of nodes.

T_i be the trust index of node N_i,

T_{inc} be the value of trust increment,

T_{dec} be the value of trust decrement,

T_{th} be the trust threshold value.

N_k be the node which forwards a data packet P_k.

p be positive constant for trust increment and decrement.

Algorithm 1

1. Initially, a lookup table is maintained by every node that includes source and destination IP addresses, sequence numbers and port numbers, and the next hop address.

2. Node N_i receives the data packet P_k.

3. If P_k is a retransmitted packet, then Node i decrements trust index of N_k by

$T_{dec} = T_{dec} - 2*p$

Compare T_{i-1} with T_{th}.

If $T_0 < T_{i-1} < T_{th}$,

Packet is dropped.

Else

Packet is forwarded to node N_{i+1}.

N_i updates the lookup table with current trust values.

End if

Else

If P_k is an acknowledgement packet, then

If N_k originally forwarded P_k, then

N_i increments trust index of N_k by $T_{inc} = T_{inc} + p$

End if

End if

Explanation: Initially, a lookup table is maintained by every node that includes source and destination IP addresses, sequence numbers and port numbers, and the next hop address. Node N_i receives the data packet P_k. { Node N_i is the receiving node. Node N_k is the node sending packet P_k}. If the packet was retransmitted, then the sender nodes's trust index is decremented. It is then compared to the threshold trust index. If it is smaller than threshold trust index, the packet is dropped. Or else if it is an acknowledgement packet, then trust index of sender node is incremented.

B. Route Selection for Integrity

Let

Ti be the Trust index on the individual neighbour,

Ta be the average of the trust index of all the neighbours that forwarded/generated RREP(Route reply),

Oi be the number of Hops in the route established by the individual node in its RREP,

Oabe the average of all Oi's obtained from individual neighbours,

CRS_A and CRS_B be the Cost of route selection of the node A and node B respectively.

Tr(A) and Tr(B) be the trust index of the nodes A and B respectively which represents the trust index of the individual neighbour for a Route.

N_{hi}^A and N_{hi}^B are the trust index of highest immediate downstream neighbours of the nodes A and B respectively.

Algorithm 2

1. The trust index of all the nodes is calculated and then the source node calculates the Cost of route selection (CRS) for all its available routes to the destination using the formula

$CRS = (Ti / Ta) * (Tr) * (Oa / Oi)$

2. If $CRS_A = CRS_B$, then

If Tr(A) >Tr(B), then

Select route A.

Else if Tr(A) = Tr(B), then

If $N_{hi}^A > N_{hi}^B$

Select N_{hi}^A

Else

Select the shortest route.

End if

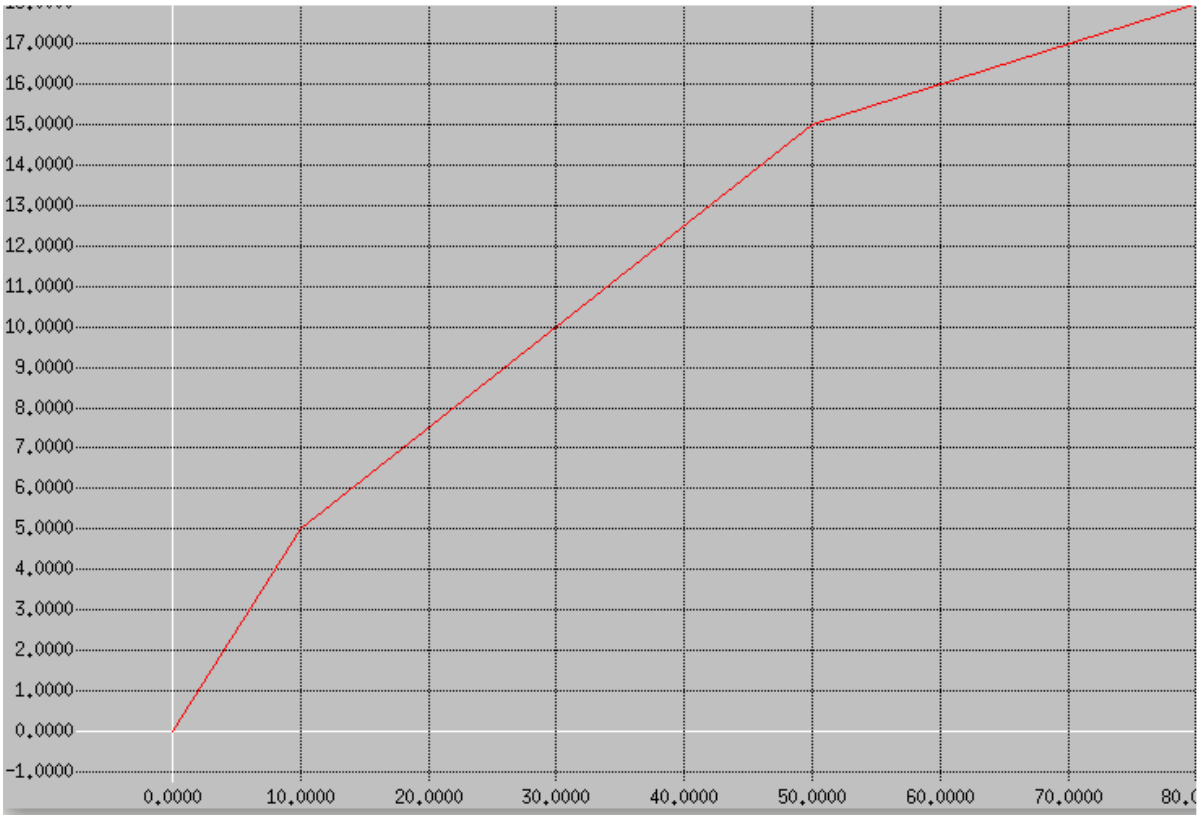
End if

Explanation: The trust index of all the nodes is calculated and then the source node calculates the Cost of route selection (CRS) for all its available routes to the destination using the formula given in the algorithm above. Whichever route has lower CRS is the optimal route. If the CRS of two given nodes A and B is equal, then we check their respective trust indexes. Whichever node has greater trust index is chosen. If both trust indexes are also same then, we check the trust index of highest immediate downstream neighbours of the nodes of A and B. Whichever has highest is chosen.

VII. EVALUATIONS

A. Network resiliency against node capture

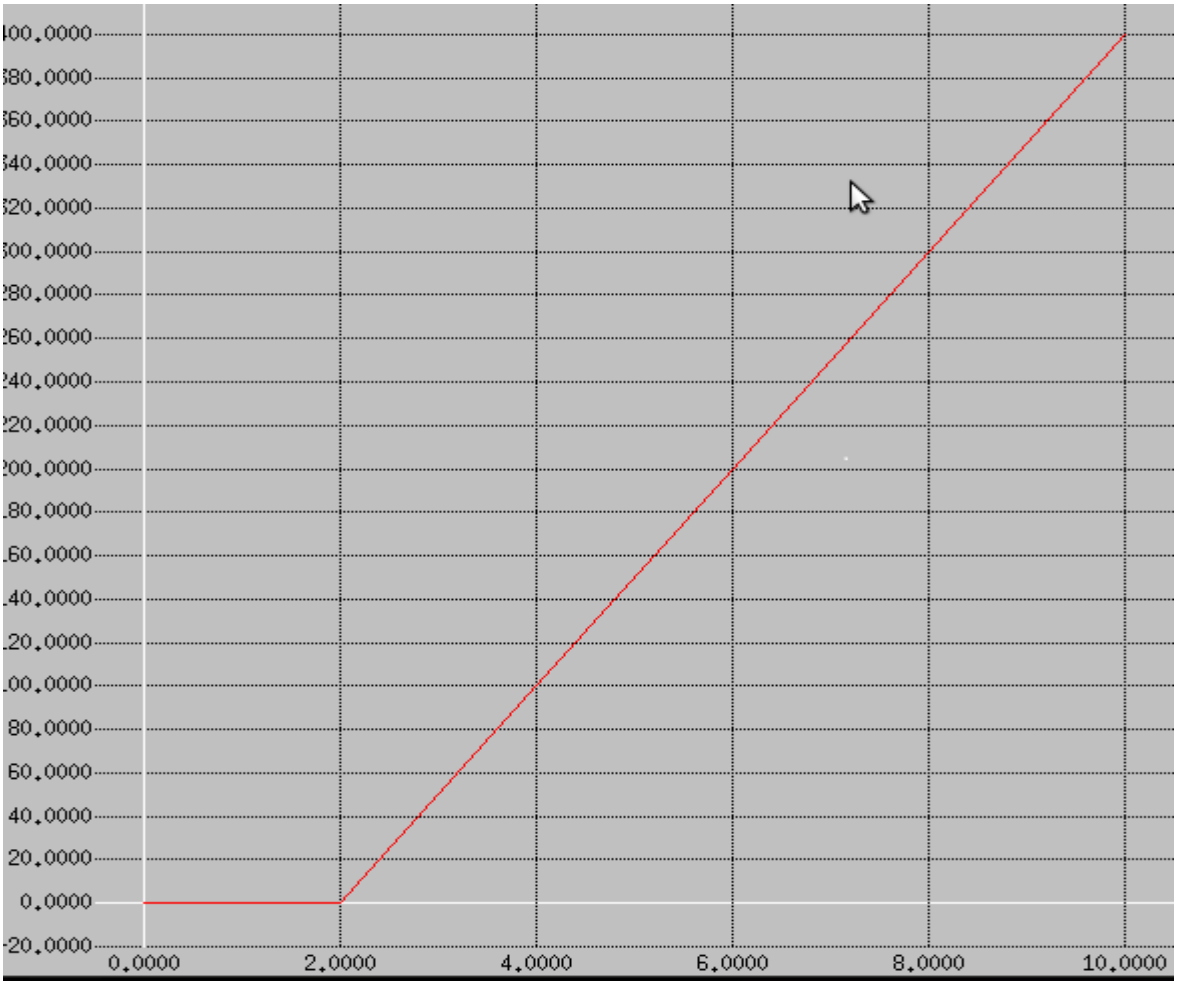
1) Average case analysis: The break-in of any single node by an adversary does not release enough information to the adversary to break secure communication for any pair of nodes. However, break-ins of multiple nodes may compromise a set of other nodes. Assume p nodes are compromised and kc(p) is the expected number of keys disclosed correspondingly. As Proposition 2 shows, $k_c(p) = r-(r-s)((r-s)/r)^{p-1}$. Then C(⌊$k_c(p)$⌋, s)/C(r,s) percentage of nodes will be compromised. This is depicted in the following graph.



x-axis depicts the number of break-ins.

y-axis depicts the number of compromised nodes multiplied by a factor of ten. Here, r=40.

2) Worst case analysis: We find $k_v(p)$ depicting the maximum number of nodes that are compromised. This is depicted in the following graph.



x-axis depicts the number of break-ins.

y-axis depicts the number of compromised nodes multiplied by a factor of 10^5.

Here, r=100, s=5.

B.Secure Connectivity Coverage

Based on trust index, secure communications are possible. The algorithm provided above is used.

C.Computation complexity

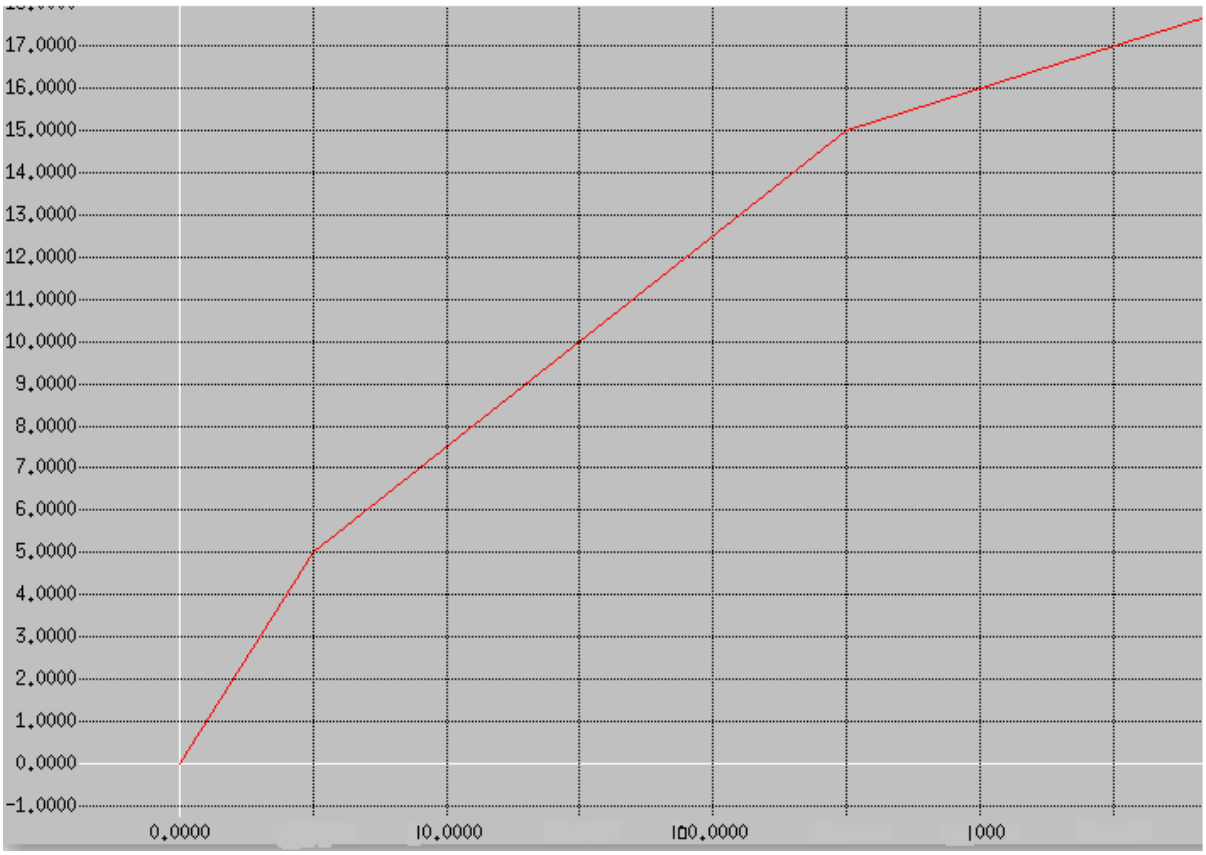
There is no computation complexity in this self-contained scalable key management method.

D.Communication overhead

Since this is a self-contained public-key management scheme, no certicates usage is necessary. If new nodes join the network, then communication is required. Thus, the overhead while communicating is nearly zero.

E. Memory footprint

In oue scheme, only a few pairs of keys are required even for a large network. From Algorithm 1 in section V, we can see that a low as 18 key pairs are sufficient for securely communicating among 1000 nodes without any consideration to resiliency. This can further be depicted by the following graph.



x-axis depicts network size (on logarithmic scale).

y-axis depicts the number of keys.

VIII.CONCLUSION

Because of resource limitations, asymmetric key management schemes are not suited for MANETs. Moreover, it is problematic to assume a trusted third party which can provide pair-wise secret keys to neighbouring nodes because of MANET being infrastructureless. Hence, we use symmetric key pre-distribution schemes for secure transactions in MANETs.

In this paper, a self-contained public key management scheme is represented. This scheme achieves near zero communication overhead while providing security services. Cryptographic keys in small numbers are stored at individual nodes before deployment in network. Combinatorial design of public-private key pairs is used for better utilization of storage space. This means a combination of more than one key pair is utilized by nodes to encrypt and decrypt messages. We also propose a secure communication algorithm for forwarding the packets in MANET.

REFERENCES

[1] WalidBechkit, YacineChallal, AbdelmadjidBouabdallah, A new class of Hash-Chain based key pre-distribution schemes for WSN, Computer Communications 36 (2013) 243–255.

[2] Wenbo He, Ying Huang, RavishankarSathyam, KlaraNahrstedt, WhayChiou Lee: SMOCK: a scalable method of cryptographic key management for mission-critical wireless ad-hoc networks. IEEETransactionsonInformationForensicsandSecurity4 (1): 140-150 (2009)

[3] L. Eschenauer, V.D. Gligor, A key-management scheme for distributed sensor networks, in: ACM CCS '02, 2002, pp. 41–47.

[4] Li Gong, David J. Wheeler, A matrix key distribution system Issue 164 of Technical report (University of Cambridge. Computer Laboratory), University of Cambridge Computer Laboratory.

[5] C.J. Mitchell and F.C. Piper, 'The cost of reducing key-storage requirements in secure networks' (pdf), Computers and Security 6 (1987) 339-341.

[6] C.J. Mitchell and F.C. Piper, 'Key storage in secure networks' (pdf), Discrete Applied Mathematics 21 (1988) 215-228.

[7] W. Du, J. Deng, Y. Han, S. Chen, P. Varshney, A key management scheme for wireless sensor networks using deployment knowledge, in: IEEE INFOCOM, 2004, pp. 586–597.

[8] C. Castelluccia, A. Spognardi, A robust key pre-distribution protocol for multiphase wireless sensor networks, in: IEEE Securecom, 2007, 351–360.

[9] MahalingamRamkumar and NasirMemon, HARPS: HAshed Random Preloaded Subset Key Distribution,Cryptology ePrint Archive: Report 2003/170

[10] S.A. Çamtepe, B. Yener, Combinatorial design of key distribution mechanisms for wireless sensor networks, IEEE/ACM Transactions on Networking 15 (2007) 346–358.
[11] S. Ruj, B. Roy, Key predistribution using combinatorial designs for grid-group deployment scheme in wireless sensor networks, Transactions On Sensor Networks 6 (1) (2009).
[12] A. Perrig, R. Szewczyk, J.D. Tygar, V. Wen, D.E. Culler, Spins: security protocols for sensor networks, Wireless Networks 8 (5) (2002) 521–534.
[13] A. Perrig, R. Szewczyk, V. Wen, D.E. Culler, J.D. Tygar, Spins: security protocols for sensor networks, in: ACM MOBICOM, 2001, pp. 189–199.
[14] Kartheesan. L and S.K. Srivatsa, Trust Based Packet Forwarding Scheme for Data Security in Mobile Ad Hoc Networks, IOSR Journal of Computer Engineering (IOSRJCE) ISSN: 2278-0661 Volume 2, Issue 3 (July-Aug. 2012), PP 40-48