Ain Shams Engineering Journal (2015) 6, 161–170

Ain Shams University

Ain Shams Engineering Journal

www.elsevier.com/locate/asej
www.sciencedirect.com

ELECTRICAL ENGINEERING

Self-organized key management with trusted certificate exchange in MANET

Saju P John [a,*], Philip Samuel [b]

[a] Department of Computer Science, Thejus Engineering College, Thrissur 680 584, India
[b] School of Engineering, Cochin University of Science & Technology, 682 022, India

Received 3 October 2013; revised 23 May 2014; accepted 28 September 2014
Available online 15 December 2014

KEYWORDS

Mobile ad hoc networks (MANET);
Trusted third party (TTP);
Distributed trusted authority (DTA);
Public/private key pairs

Abstract In MANET, security is more challenging due to problems related to key exchange. It is necessary to secure the exchanges in MANETs for assuring the development of services in the network. The self-organized MANET is visualized as a key communication technology enabler for application such as network centric warfare, disaster relief operations, emergency situations, and intelligent transportation systems. In this paper, we propose a self-organized key management technique coupled with trusted certificate exchange for mobile ad hoc networks. The proposed architecture consists of one coordinator node, servers and normal mobile nodes. The coordinator acts as a mediator for transmitting the message among the servers and mobile nodes. Each node generates its own public/private key pairs using server-signed public keying technique. Then multi-path certificate exchange technique is employed where public key of the nodes is certified by different nodes. Those nodes that issued the certificates are validated using the Eigen Vector Reputation Centrality. By simulation results, we show that the proposed approach improves security.

1. Introduction

1.1. Mobile ad hoc network (MANET)

An adaptive, self-configurable and self-organizing multi-hop wireless network devoid of infrastructure along with the random topology is termed as MANET. Through this definition, it is revealed that the network can be created, combined or partitioned into detached networks depending on the requirements of the network. Also ad hoc network can be set up quickly without any base station for wireless cellular networks. The routes among the end-users have a multi-hop wireless links in this network. Besides the above features, the ad hoc network has a capacity to move autonomously [1,2].

1.1.1. Self-organized MANET

The entirely self-organized mobile ad hoc network is devoid of any kind of online or offline authority. The end-users generated this network in ad hoc mode. As the relationship among the users is not recognized priorly, the user does not share common keys with their nodes. Thus without depending on the common offline trusted authority TTP, users have to build

[*] Corresponding author. Tel.: +91 8122012375.
E-mail address: shajujohnphd@gmail.com (S.P John).

Peer review under responsibility of Ain Shams University.

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.asej.2014.09.011
2090-4479 © 2014 Production and hosting by Elsevier B.V. on behalf of Ain Shams University.

security relationships among themselves following the formation of network. The authority-based MANET holds up the applications, which insist the utility of offline authority. The nodes related to the authority-based ad hoc networks include priorly established relationships when compared to fully self-organize ad hoc networks. The trusted authority is responsible for offering the cryptographic keying material and set of system parameters for each node before formation of the network. Each node will turn out to be self-authority and further distributes the certificates to the nodes in transmission range following the formation of the network [1].

1.2. Security in MANET

The MANET security is categorized into following types:

(1) The security model depending on TTP in which the certificates are issued using a single authority such as Public Key Infrastructure (PKI).
(2) The completely self-organized model in which the security is not based on trusted authority or fixed server. This is similar to security models based on trust propagation via trust graph like PGP [3,13].

1.2.1. Self-organized key management

Self-organized key management technique is categorized into following two groups:

(1) Virtual CA (Certificate Authority): This technique considers that there exists a certificate authority called TTP. Virtual CA offers high-level guarantees and does not necessitate warm-up time. The main issue concerned with this group is related to selection of Certificate Authority (CA) and overcoming attacks in CA, which is caused by malicious users.
(2) Web-of-trust: This group does not necessitate CA, which reveals that it is more flexible. However, it is affected by recurrent communication and more memory spaces as it should gather public-key certificates in advance. It needs more time to gather all the certificates in the network due to that reason of exchanging the repository among moving users in periodical manner [4].

1.3. Certificate chaining approach

When two nodes desire to interact in secured manner, they exchange public keys with each other using the technique that verifies and signs packet in each hop of the network. This technique is termed as certificate chaining which involves the signing of the key exchange packets by each hop and verification of the signature by the next hop. The merit of this approach is that it permits the transmission of the public keys to the destination in the secured way [5].

1.3.1. Essential functions of certificate chaining approach

1. *Mitigating the Certificate and Private Key Compromise:* Due to the cooperation of the private key/certificate, the malicious attacker utilizes these certificates to initiate man-in-middle attacks. This can be prevented using the certificate chaining approach in which every node guarantees the originality of the certificate.
2. *Setting Model for Future Extension:* For mitigating attacks on availability criteria, the certificate chaining approach plays a major role. For example, the trust management system employs certificate-chaining approach for detecting the flooding attacks [5].

Mainly the certificate chaining approach is appropriate for self-organized MANET that permits the users to generate, accumulate, distribute and revoke their own public keys without the help of trusted authority [6].

1.3.2. Limitations of existing certificate chaining

The current certificate chaining approach exhibits the following limitations:

1. No assurance to the public keys authentication. Certainly, the certificate chaining among two nodes is possibly not established.
2. There is a requirement of extensive time until the web-of-trust is set up among each other.
3. The predicted results in this scheme will not be precise since it is not based on TTP. The nodes act as individual CA and consequently the certificate chain will depend on the nodes honesty concerned with the formation [7].

1.4. Problem identification

The drawbacks of the existing approaches are as follows:

In [8], the proposed technique to cope with misbehaving node does not prevent users from creating virtual identifiers or from stealing the identity of people who do not participate in the network. Also exploration of more sophisticated load-balancing/data management schemes for public-key management is not handled.

In [2], the author has not discussed the authentication parameters.

In [6], the proposed technique lags certificate revocation methodology. Also authentication parameters are discussed in detail.

In [9], they have only assumed that every node in a MANET first generates a public/private key pair.

From the existing works done so far we can come to know that there is a requirement of strong self-certified key generation and certificate exchange mechanisms along with some trusted model.

So, we propose a Self-Organized Key Management with Trusted Management in MANET. Our approach includes three phases, which are as follows:

- Phase 1—Creation of Public/Private Key Pairs.
- Phase 2—Trust Management Mechanism.
- Phase 3—Certificate Exchange Technique.

2. Literature review

Some of the existing self-organization key management approaches are as follows:

Omar et al. [6] have proposed a fully distributed public key management system based on trust graphs and threshold cryptography. Their system is based on a trust graph where they utilized threshold cryptography scheme. This scheme refuses the malicious nodes that offer false public key certificates defrauding the certification service. They utilized certificate-chaining approach for carrying out authentication and permitting the users to issue the public key certificates.

Kawabata et al. [2] have proposed a self-organized key management based on trust relationship list scheme that does not rely on any trusted authority or fixed server in the ad hoc networks. Each node manages the certificate created by them and when the certification is required, node gathers the certificates from other nodes by verifying the list of trust relationship. By using trust relationship list, they minimized the amount of communications and memory used.

van der Merwe et al. [10] have proposed a trustworthy key management for mobile ad hoc networks (AdHocTKM). They utilized threshold cryptography and certificate chaining technique that integrates the self-certified public keys and self-certificates to yield a key management service. They proposed a threshold self-certified public keying technique that allows cooperation among a single entity and a distributed authority for an implicit self-certified public key, without the authority gaining knowledge of the corresponding private key.

Capkun et al. [4] have proposed a fully self-organized public key management scheme for mobile ad hoc networks. In this technique, the users generate the public/private key pairs and issue certificates for executing authentication without considering the network partitions and centralized services. Though the security is performed in a self-organized manner, the key authentication among two users is achieved with local information. Their proposed local repository construction algorithm attains high performance on a wide range of certificate graphs.

Dahshan et al. [7] have proposed an on demand self-organized public key management for MANETs. Each user generates its own public/private key, issues the certificates to the neighbor nodes and executes the public key authentication devoid of any centralized authority. Their scheme is based on the presence of the web-of-trust among mobile nodes forming the network. Also they executed the certificate chain discovery technique with the help of routing process. There scheme greatly reduces the communication cost as the every node limits its search for the certificate chain to its directly trusted nodes only.

van der Merwe et al. [11] have proposed fully self-organized peer-to-peer key management for mobile ad hoc networks. They utilized sub-ordinate public keys and crypto-based identifiers for eradicating any form of trusted third party. They used a localized certificate exchanges on the network for breaking the routing-security interdependence cycle without degrading the performance of the network. Their scheme is more generic as it can be applied over mobile wireless network inclusive of symmetric or asymmetric encryption.

Sen et al. [9] have proposed a robust and efficient key exchange protocol for node authentication in a MANET based on multi-path communication. The proposed key exchange protocol can be integrated with routing protocol. The protocol is based on the multi-path communication and hence it is robust even in the presence of malicious nodes in the network. Moreover, it has a minimal computation and communication overhead.

Ibriq and Mahgoub [14] have presented a hierarchical key establishment scheme called HIKES. In this scheme, the base station acts as the central trust authority and the randomly selected sensors act as local trust authorities authenticating the cluster members and issuing private keys. It uses a partial key escrow scheme that enables any sensor node selected as a cluster head to generate all the cryptographic keys needed to authenticate other sensors within its cluster. It localizes secret key issuance and reduces the communication cost with the base station. It provides an efficient broadcast authentication in which source authentication is achieved in a single transmission and a good defense for the routing mechanism.

Chen et al. [12] have proposed a lightweight and provably secure user authentication scheme with anonymity for the GLOMONET. It uses only symmetric cryptographic and hash operation primitives for secure authentication. Apart from this, it takes only four message exchanges among the user, foreign agent and home agent. They have also demonstrated that the protocol provides the security attributes including prevention of various attacks, single registration, user anonymity, user friendly, no password/verifier table, and use of one-time session key between mobile user and foreign agent.

3. Self-organized key management with trusted certificate exchange

3.1. Overview

The proposed architecture consists of the coordinator node, servers and ordinary mobile nodes. The coordinator node acts as a mediator for transmitting the message among the servers and mobile nodes. Each node generates its own public/private key pairs using server-signed public keying technique. The coordinator node helps in generating the publicly-recoverable public key for any node N_i without the knowledge of the subsequent private key. The coordinator node acts as a distributed trusted authority. It combines the shares of $(t + 1)$ servers for computing signature parameter. The nodes in the network are validated using the trust management mechanism. The trust value is computed using the Eigen Vector Reputation Centrality. This certificate exchange technique helps the nodes to authenticate themselves with the members in the network before they get joined and start accessing the network resources. As a result of multiple independent certifications, the confidence assigned to the certificates is higher.

3.2. Proposed architecture

Our architecture consists of the coordinator node, servers and normal mobile nodes. One coordinator is chosen, as the distributed trusted authority. Fig. 1 shows the proposed architecture of the self-organized key management technique of the MANET. It includes the ordinary nodes $(N_1, N_2, \ldots, N_{10})$, where N_4 is chosen as the coordinator node (N_c). There are 4 servers $\{z_1, z_{2,}, \ldots, z_n\}$. The coordinator acts as a mediator node for transmitting messages from normal mobile nodes to the servers.

The proposed technique includes four phases that are described in the following section.

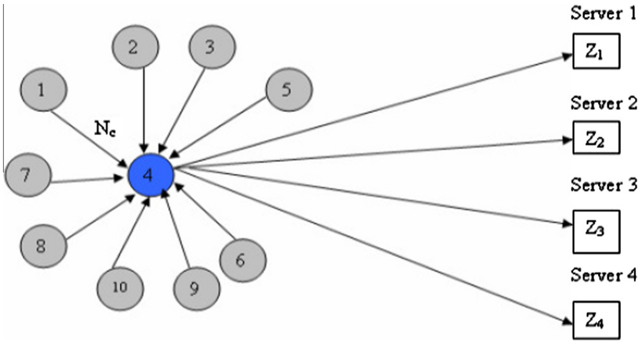

Figure 1 Architecture of the self-organized key management technique.

3.2.1. Phase 1: Creation of public/private key pairs

This phase involves the generation of public/private key pairs (K_{pu}, K_{pr}) using server-signed public keying technique. This technique allows the users to generate their own public/private key pairs. The system consists of n servers ($z_1, z_2, \ldots, z_n$) and a trusted coordinator node. The coordinator node sends a secret value to the corresponding node which requires a public/private key generation. The receiver node sends back a newly generated secret value for getting the signature parameter.

The coordinator node acts as a distributed trusted authority. It combines the shares of (t + 1) servers for computing signature parameter. Consider that coordinator node selects the primes x, λ with $\lambda | x - 1$, a generator d of a multiplicative subgroup of Z_x^* with order λ. Let $h(\cdot)$ denotes a one-way hash function and N_i denotes any node in the network. The coordinator node publishes x, λ, d and h. The steps involve in the generation of public/private key pairs are as follows:

- *Step 1:* After selecting its random number $k'_{\text{Ni}} \in_R Z\lambda^*$, coordinator node computes the secret value (using Eq. (1)).

$$C'_{\text{Ni}} = d^{(k'\text{Ni})} \tag{1}$$

The computed secret value is transmitted to N_i.

- *Step 2:* After selecting its random number $q \in_R Z\lambda^*$, N_i computes the secret value (using Eq. (2)).

$$C_{\text{Ni}} = C'_{\text{Ni}}\, d^q \tag{2}$$

After the above computation of secret value, N_i then transmits its own ID and secret ($\text{ID}_{\text{Ni}}, C_{\text{Ni}}$) again to N_c.

- *Step 3:* The coordinator node forwards the ($\text{ID}_{\text{Ni}}, C_{\text{Ni}}$) to each server.
- *Step 4:* Each server computes the hash function of (ID_{Ni}, C_{Ni}), which is represented as $h[(\text{ID}_{\text{Ni}}, C_{\text{Ni}})]$. Then it computes its threshold signature, i.e. Sign_l $[Z_i, h[(\text{ID}_{\text{Ni}}, C_{\text{Ni}})]]$ to coordinator node.
- *Step 5:* Coordinator node collects all the t + 1 shares from the servers and computes the signature parameter Sign_{Ni} (using Eq. (3)) and forwards both Sign_{Ni} and Sign_l $[h[(\text{ID}_{\text{Ni}}, C_{\text{Ni}})]]$ to N_i.

$$\text{Sign}_{\text{Ni}} = h[(\text{ID}_{\text{Ni}}, C_{\text{Ni}}) + k'_{\text{Ni}} \tag{3}$$

- *Step 6:* Then N_i computes the private key k_{pr} (using Eq. (4))

$$k_{pr} = \text{Sign}_{\text{Ni}} + q \tag{4}$$

The tuple (C_{Ni}, k_{pr}) can be viewed as the signature of the DTA on ID_{Ni}.

- *Step 7:* After verifying the signature, N_i computes the corresponding public key k_{pu} (using Eq. (5)) and publishes C_{Ni} and ID_{Ni}. k_{pu} of N_i is publicly verified by decrypting Sign_l $[h[(\text{ID}_{\text{Ni}}, C_{\text{Ni}})]$ using the public key of the coordinator node, comparing the decrypted hash value to $[h(\text{ID}_{\text{Ni}}, C_{\text{Ni}})]$ and evaluating Eq. (5).

$$k_{pu} = d^{kpr} = d^{(\text{IDNi,CNi})} \cdot C_{\text{Ni}} \tag{5}$$

The above approach assists the users to fully control the security settings of the system.

3.2.2. Phase 2: Trust management mechanism

The trust management mechanism is employed in order to validate the nodes in the network. The trust value is computed using the Eigen Vector Reputation Centrality Mechanism [12]. Each node deployed in the network computes the Eigen vector centrality (EVC_i) of its neighbors (using Eq. (6)) for exhibiting the reputation and level of confidence on each neighbor.

Let n_i and n_j be the adjacent nodes. The centrality for the ith node is proportional to the sum of the scores of all nodes that are linked to it.

$$\text{EVC}_i = \frac{1}{\delta}\sum_{j \in S(i)} \text{EVC}_j = \frac{1}{\delta}\sum_{j=1}^{n} R_{ij}\text{EVC}_j \tag{6}$$

where R_{ij} is the adjacency matrix of the network, $S(i)$ is the set of nodes that are connected to the ith node, n is the total number of nodes and δ is the constant.

R_{ij} is defined in Eigen vector centrality using the following condition:

```
If
  $i$th node is adjacent to the $j$th node
Then
  $R_{ij} = 1$ (In eigen-vector centrality)
Else
  $R_{ij} = 0$
End if
```

A node N computes $F(i,j)$ and $E(i,j)$. $F(i,j)$ is defined as the percentage of packets initiated from n_i which were forwarded by n_j over the total number of packets offered to n_j. $E(i,j)$ is defined as the percentage of packets that were expired over the total number of packets offered to node j. Each node periodically computes its connectivity rating (recent satisfaction index (RSI)) with each of its direct neighbor nodes using the above computed percentages (using Eq. (7)).

$$\text{RSI}_{ij} = F(i,j) - E(i,j) \tag{7}$$

RSI_{ij} is weighted into the direct reputation of node j which is shown in Eq. (8).

$$\text{Tr}_{ij} = \text{Tr}^*_{ij\text{-}pr}\eta + \text{RSI}^*_{ij}(1 - \eta) \tag{8}$$

where $\mathrm{Tr}_{ij\text{-}pr}$ is the trust value in node i for node j prior to reputation value that node i had for node j before inclusion of RSI_{ij}, η is the constant that exhibits level of confidence contained by n_i in the past for n_j.

```
If
  There is no link among $n_i$ and $n_j$,
Then
  $\mathrm{Tr}_{ij}$ is reduced using $\beta$ instead of $\eta$
End if   (where $\beta$ is a constant that reduces the trust value)
```

Tr_{ij} is normalized (using Eq. (9)), by examining it over time t.

$$\mathrm{Tr}_{ij} = \mathrm{EVC}_i^* \frac{\mathrm{Tr}_{ij}}{f(t)_{\max}(\mathrm{Tr}_{ij})} \tag{9}$$

$f(t)_{\max}$ is the function that reports about the maximum Tr_{ij} over time t.

$\mathrm{Tr}_{\min}$ represents the minimum threshold value of trust. The trust value depends on the eigen vector centrality score and the recent satisfaction index. A normal node must have a trust value Tr_{ij}, higher than the threshold minimum $\mathrm{Tr}_{\min}$, of the network (i.e. $\mathrm{Tr}_{ij} > \mathrm{Tr}_{\min}$).

3.2.3. Phase 3: Certificate exchange technique

The certificate exchange technique helps the nodes to authenticate themselves with the members in the network before they get joined and start a new communication. In order to enhance the reliability of certificate exchange protocol, Multi-path Technique is utilized. During the multi-path certificate exchange, the public key of a node is certified by the different nodes. As a result of multiple independent certifications, the confidence assigned to the certificates is higher. Moreover, the authentication is performed mutually.

Let S and D represent the source and destination respectively.
Let N_i represents the intermediate nodes.
Let kpu_d be the public key of D.
Let kpu_s be the public key of S.
Let $T(S)$ be the set of nodes certified for kpu_s.
Let $\mathrm{REQ}_{\mathrm{cert}}$ represents the certificate request message.
Let $\mathrm{REP}_{\mathrm{cert}}$ be the certificate reply message.
Let C_{self} be the self-signed certificate.
Let ID_D be the identity value of D.
Let CL be the certificate list.

When S receives kpu_d, it issues a certificate for that public key. Consequently, D issues a certificate for kpu_s. Each node in $T(S)$ contains its public key certified by S since the authentication is mutual. The steps involved in the certificate exchange process are as follows:

Step 01: S broadcasts $\mathrm{REQ}_{\mathrm{cert}}$ containing ID_D and $T(S)$ for D's certificates.

$$S \xrightarrow{\mathrm{REQ}_{\mathrm{cert}}+\mathrm{ID}_D} \text{Neighbor nodes} \tag{10}$$

This $\mathrm{REQ}_{\mathrm{cert}}$ is sent with a minimum time to live ($\mathrm{TTL}_{\min}$) for minimizing the communication overhead of the protocol.

Step 02: When N_i receives the $\mathrm{REQ}_{\mathrm{cert}}$, it verifies kpu_s and checks its own CL.

```
If
  ($N_i$ has no certificate for $D$) | | ($N_i$ has already replied to the
$\mathrm{REQ}_{\mathrm{cert}}$)
Then
  $N_i$ forwards the $\mathrm{REQ}_{\mathrm{cert}}$ to its neighbor nodes
Else
  $N_i$ feedbacks $\mathrm{REP}_{\mathrm{cert}}$ to $S$ that contains the certificate of $kpu_d$
signed by $N_i$
End if
```

Step 03:

```
If
  $N_i$ is unaware of $S$,
Then
  $N_i$ constructs a $C_{\mathrm{self}}$ and notifies $S$ that it wants to make a
certificate exchange which is performed via a multiple node-disjoint
paths.
End if
```

Step 04:

```
If
  $N_i$ already has a route to $D$ in its cache,
Then
  $N_i$ informs $D$ that $S$ has requested its $kpu_d$.
  $D$ responds to query and requests a certificate for $kpu_s$.
End if
```

Since N_i and D can authenticate each other, the communication among the D and N_i is made secured using N_i's signature. Hence there is no possibility for any node to corrupt the certificate of S which is issued by N_i.

Step 05:

```
If
  $D$ is unaware of adequate number of nodes,
Then
  $D$ replies to $\mathrm{REQ}_{\mathrm{cert}}$ itself.
End if
```

Step 06: S repeats the above process by increasing the TTL value until it obtains the minimum number of certificates for kpu_d.
Step 07: S then calculates the trust value Tr_{ij} of the nodes included in the all offered paths.
Step 08: S considers only those paths, which are free from malicious nodes.

Step 09: Among the obtained paths, source selects a path that is having more certifiers of the destination node D.
Step 10: S then forwards the first packet to D that contains the set of nodes that has offered the certificates for kpu_d.
Step 11: Once they have exchanged their public keys, S and D issue certificates for each other.

Due to multiple independent certifications, the confidence assigned to these certificates is higher. For example, consider Fig. 2. We demonstrate our certificate exchange mechanism by considering N_5. S broadcasts the $\mathrm{REQ_{cert}}$ to its neighbor nodes. When N_5 receives the message, it checks its CL. If N_5 does not know D or it has already sent the $\mathrm{REP_{cert}}$, then it just forwards it to next node N_6. Otherwise, N_5 replies with $\mathrm{REP_{cert}}$ that contains the certificate of kpu_d signed by N_5 to S. When N_5 is not aware of S, then N_5 constructs a C_{self} and notifies S that it wants to make a certificate exchange via multiple node-disjoint paths, i.e. through (N_5–N_1–S) & (N_5–N_4–S) & (N_5–N_8–N_7–S). If N_5 already has a route to D in its cache, then it informs D that S has requested its kpu_d and it responds to query and requests a certificate for kpu_s. If D is unaware of adequate number of nodes, it replies to $\mathrm{REQ_{cert}}$ itself. S repeats the above process by increasing the TTL value until it obtains the minimum number of certificates for kpu_d.

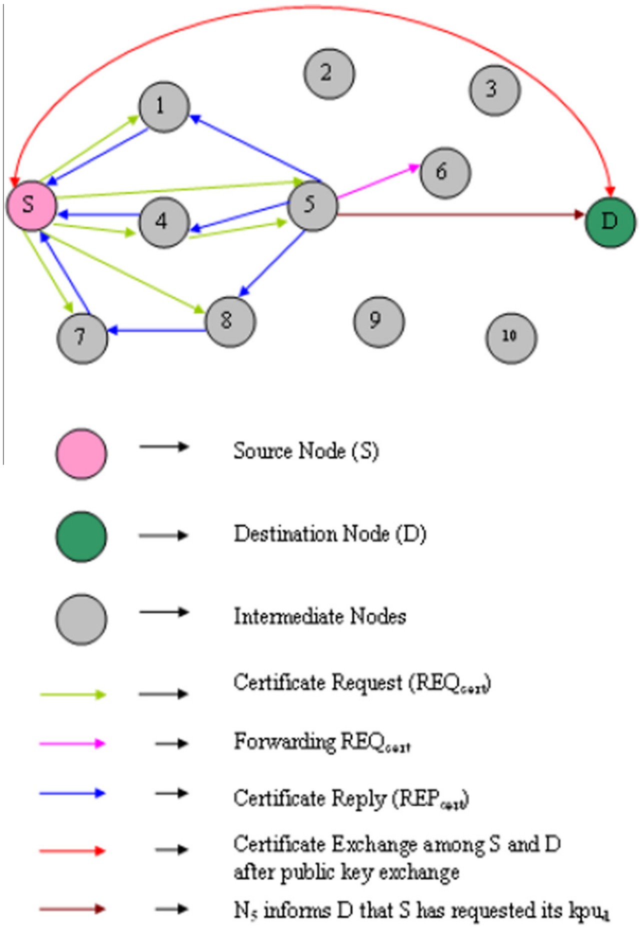
S
1
2
3
4
5
6
7
8
9
10
D
Source Node (S)
Destination Node (D)
Intermediate Nodes
Certificate Request (REQ_{cert})
Forwarding REQcert
Certificate Reply (REPcert)
Certificate Exchange among S and D after public key exchange
N5 informs D that S has requested its kpud

Figure 2 Certificate exchange.

3.3. Overall algorithm

The entire process of the proposed technique is described using the following algorithm.

Step 01—Network Architecture

The architecture for self-organized key management technique is constructed such that it includes a coordinator node, servers and normal mobile nodes. The coordinator acts as a mediator node for transmitting messages from normal mobile nodes to the servers.

Step 02—Public/Private Key Generation

Each mobile node generates its own public/private key pairs using server-signed public keying technique. The coordinator node helps in generating the publicly-recoverable public key for any node $\mathrm{N_i}$ without the knowledge of the subsequent private key. The coordinator node acts as a distributed trusted authority. It combines the shares of (t + 1) servers for computing signature parameter.

Step 03—Trust Calculation

The nodes in the network are validated using the trust management technique named as Eigen Vector Reputation Centrality technique.

Step 04—Multi-Path Certificate Exchange

After the generation of public/private key pairs, multi-path certificate exchange technique is employed where public key of the nodes is certified by different nodes. The authentication is also performed mutually.

Step 05—Malicious Node Detection

Source will collect the certifiers of the destination node and all possible paths to the destination. It then collects the trust value Tr_{ij}, of the nodes included in the all offered paths. $\mathrm{Tr_{min}}$ represents the minimum threshold value of trust. The minimum threshold value depends on the total number of nodes and the adjacency matrix of the network. A normal node must have a trust value Tr_{ij}, higher than the threshold minimum $\mathrm{Tr_{min}}$, of the network (i.e. $\mathrm{Tr}_{ij} > \mathrm{Tr_{min}}$).

Step 06—Path Selection

Among the obtained paths, source selects a path that is having more certifiers of the destination node. After selecting the path, source and destination certify their public keys each other.

4. Simulation results

4.1. Simulation model and parameters

Simulations were performed using Network Simulator (NS-2) [15], particularly popular in the ad hoc networking community. The MAC layer protocol IEEE 802.11 with a data rate of 11 Mbps is used in all simulations. The transmission range is set to 250 m. The propagation model is Two Ray Ground. The total number of nodes is set to 100 nodes in 1000 m × 1000 m network area. In our simulation, the minimal speed is 5 m/s. The source-destination pairs are spread randomly over the network. The ns-2 Constant Bit-Rate (CBR) traffic generator is used to set up the connection patterns with different random seeds. Each node has one CBR traffic connection with a single unique destination. Sources initiation time is uniformly distributed over the first 60 s of the simulation time. We vary the load value as 50, 100, 150, 200 and 250 Kb.

The size of certificates was also set to 512 bytes. The total number of connections in the network was set to 20 connections. The Ad Hoc On-demand Multipath Distance Vector (AOMDV) routing protocol was chosen for the simulations. The simulation results are the average of 10 runs. The proposed technique was easily integrated into the AOMDV protocol's route discovery mechanism.

In the simulation, attacks are simulated where the attacker nodes send spurious certificates to the nodes, which have requested for those certificates. These attacks can be isolated attacks where every attacker certifies a different public key. However, the attackers may also launch a cooperative attack where a group of attackers collude and send certifications for the same public key that is spurious. Both these types of attacks—isolated and collusion—are simulated. The percentage of attacker nodes is fixed as 10% of the total number of nodes in the network (i.e.) 10 attackers. Node initialization at the network bootstrapping phase is also simulated. It is shown that each node has successfully executed the initialization step by exchanging requisite number of certificates with the honest nodes in the network. Initial trust value of 0.75 is assigned to a node that is authenticated during the initialization step, while other nodes are assumed to have a trust value of 0.5. The full trust value is assumed to be 1. The initial trust value is chosen more than half of the full trust value and other nodes trust values are chosen half of the full trust value.

Our simulation settings and parameters are summarized in Table 1.

Table 1 Simulation settings.

No. of nodes	100
Area size	1000 × 1000
Mac	802.11
Radio range	250 m
Simulation time	100–500 s
Routing protocol	AOMDV
Traffic source	CBR
Packet size	512
Speed	5 m/s
Pause time	5 s
Load	1000–5000 Kb
No. of attackers	10

4.2. Performance metrics

We compare the proposed Self-Organized Key Management with Trusted Certificate (SOKMTC) technique with On-demand Self-Organized Public Key Management (SOPKM) scheme of [8]. We select SOPKM among the existing works, since it is the latest work, which deals self-organized key management along with certificate chains and simulates in NS-2.

We evaluate mainly the performance according to the following metrics:

- **Security Cost:** It is the product number of certificates exchanged into certificate size and is represented in Mbits/sec.
- **Certificate Exchange Delay:** It is the interval between the time a request is sent for a certificate and the time when the certificate is accepted as valid.
- **Average end-to-end delay:** The end-to-end-delay is averaged over all surviving data packets from the sources to the destinations.
- **Average Packet Delivery Ratio:** It is the ratio of the number of packets received successfully and the total number of packets transmitted.
- **Fraction of node compromise:** Here we are going to calculate how a node capture affects the rest of network resilience. It is calculated by estimating the fraction of communications compromised by a capture of *x*-nodes.
- **Miss Detection Ratio:** It is the ratio of number of attacks not detected to the total number of attacks.
- **Packet Drop** It is the number of packets dropped during the transmission.

4.2.1. Based on number of connections

The most frequent communication is the acquisition of partial signatures, at least *K* per communicating node, in order to create the complete signature. Consequently, the time needed for the certification process to come to an end should not be very long. This way, the users can receive a satisfactory level of service without having to waste enough time waiting for failed requests for certification.

Figs. 3 and 4 depict the delay and security cost involved in the process of certificate exchange by each pair of source and destinations. The number of connections (source and destination pairs) is varied from 2 to 10, and corresponding delay and cost for the two schemes are measured.

It is observed that the certificate exchange delay and the security cost increase rapidly as the number of connections increases, as more number of intermediate nodes will be involved in certificate exchange. However, for SOKMTC, the delay is less by 43% and cost is reduced by 52%, when compared to SOPKM.

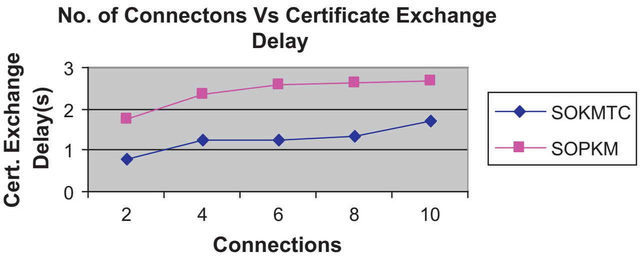

Figure 3 No. of connections vs certificate exchange delay.

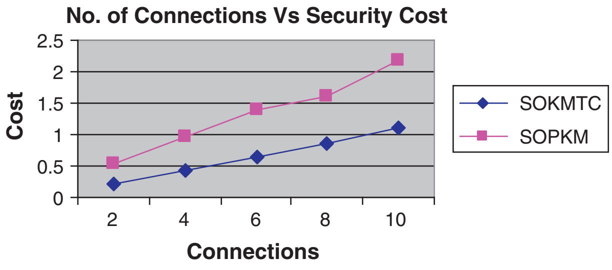

Figure 4 No. of connections vs security cost.

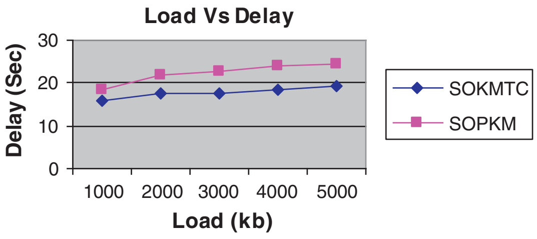

Figure 5 Load vs delay.

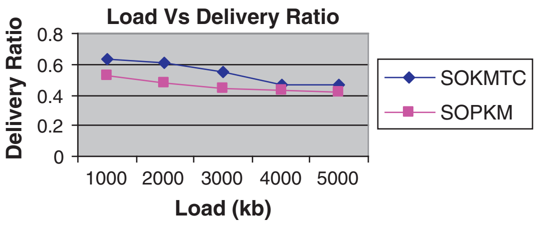

Figure 6 Load vs packet delivery ratio.

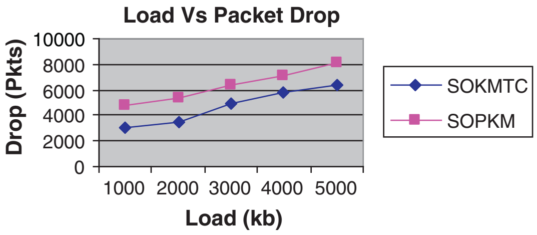

Figure 7 Load vs drop.

4.2.2. *Based on load*

In our first experiment, we vary the load as 50, 100, 150, 200, 250 Kb.

Fig. 5 shows the average end-to-end delay of both the techniques, when the load is increased from 1000 kb to 5000 kb. We can see that the delay increased linearly as the load increases. But the delay of our proposed SOKMTC is less by 20%, than the existing SOPKM scheme.

The CBR data packets dropped due to the attackers and the packet delivery ratio are presented in Figs. 7 and 6, respectively. As the load increases, more data packets flow into the network, causing more packet drops. But SOKMTC has 26% less packet drops when compared to SOPKM. Since the packet drop is linearly increasing, the packet delivery ratio is decreasing, as we can see from Fig. 6. SOKMTC has 15% increases in packet delivery ratio, when compared to SOPKM.

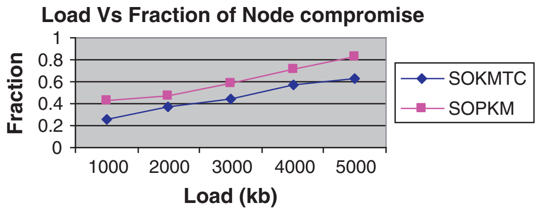

Figure 8 Load vs fraction of node compromise.

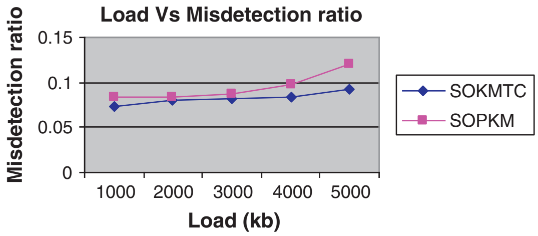

Figure 9 Load vs misdetection ratio.

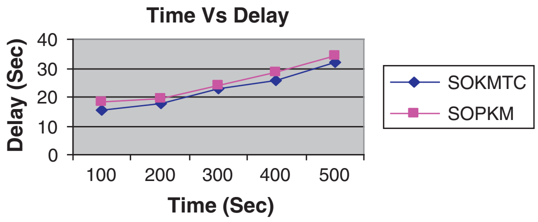

Figure 10 Time vs delay.

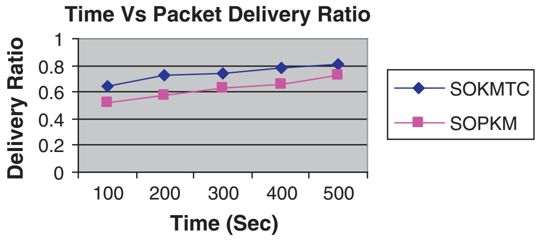

Figure 11 Time vs delivery ratio.

The results of fraction of node compromise against node capture are shown in Fig. 8. Because of the multipath certificate exchange based authentication, the number of compromised nodes is less in SOKMTC. Hence it is 26% lesser for SOKMTC than SOPKM.

The results of miss detection ratio are shown in Fig. 9. Because of the direct trust value estimation based on the centrality measure, the miss detection ratio is less for SOKMTC. Hence the resilience of proposed SOKMTC is 11.9% lesser than SOPKM.

4.2.3. *Based on time*

In our second experiment, we vary the simulation stop time as 100, 200, 300, 400 and 500 s.

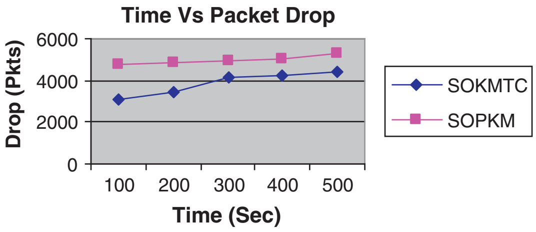

Figure 12 Time vs drop.

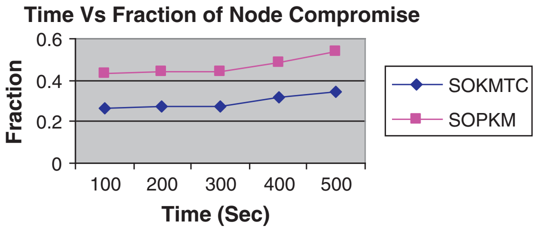

Figure 13 Time vs fraction of node compromise.

From Fig. 10, we can see that the end-to-end delay increases as the time increases from 10 to 50 s. The delay occurred in the proposed SOKMTC is less than 9.3% of the SOPKM scheme.

The results of packets dropped and the packet delivery ratio are presented in Figs. 12 and 11, respectively. As the time increases, more packets are dropped. SOKMTC has 22% less packet drops when compared to SOPKM. We can see from Fig. 12 that SOKMTC has 16% increases in packet delivery ratio, when compared to SOPKM.

The results of fraction of node compromise against node capture are shown in Fig. 13. Because of the multipath certificate exchange based authentication, the number of compromised nodes is less in SOKMTC. Hence the resilience of proposed SOKMTC is 36% lesser than SOPKM.

5. Conclusion

In this paper, we have proposed a self-organized key management technique coupled with trusted certificate exchange technique for mobile ad hoc network. The proposed architecture consists of the coordinator node, servers and ordinary mobile nodes. The coordinator acts as mediator for transmitting the message among the servers and mobile nodes. Each node generates its own public/private key pairs using server-signed public keying technique. The coordinator node helps in generating the publicly-recoverable public key for any node N_i without the knowledge of the subsequent private key. The nodes that issued the certificates are validated using the trust management mechanism. The trust value is computed using the Eigen Vector Reputation Centrality. Then multi-path certificate exchange technique is employed where public key of the nodes is certified by different nodes. As a result of multiple independent certifications, the confidence assigned to the certificates is higher. As a future enhancement we will include the certification revocation mechanism to this multipath technique.

References

[1] Sahadevaiah K, Ramanaiah OBV. Self-organized public key cryptography in mobile ad hoc networks. J Ubiquit Comput Commun.

[2] Kawabata Hideaki, Sueda Yoshiko, Mizuno Osamu, Nishikawa Hiroaki, Ishii Hiroshi. Self-organized key management based on trust relationship list. In: International conference on intelligence in next generation networks (ICIN); 2008.

[3] Ayed Hella Kaffel-Ben, Belkhiri A. Toward a peer-to-peer PKI for mobile ad-hoc networks. Cyber J: Multidisc J Sci Technol, J Select Areas Telecommun (JSAT) 2011.

[4] Capkun Srdjan, Buttya Levente, Hubaux Jean-Pierre. Self-organized public-key management for mobile ad hoc networks. IEEE Trans Mob Comput 2003;2(1).

[5] Chan Aldar C-F. Distributed symmetric key management for mobile ad hoc networks. J Inform Process Lett 2009;109(14).

[6] Omar Mawloud, Challal Yacine, Bouabdallah Abdelmadjid. Fully distributed trust model based on trust graph for mobile ad hoc networks. Comp Sec 2009:199–214.

[7] Dahshan Hisham, Irvine James. On demand self-organized public key management for mobile ad hoc networks. In: IEEE 69th vehicular technology conference. VTC Spring; 2009.

[8] Kitada Y, Takemori K, Watanabe A, Sasase I. On demand distributed public key management without considering routing tables for wireless ad hoc networks. In: 6th Asia–Pacific symposium on information and telecommunication technologies; 2005. p. 375–80.

[9] Sen Jaydip. A robust and efficient node authentication protocol for mobile ad hoc networks. In: Second international conference on computational intelligence, modelling and simulation (CIMSiM); 2010. p. 476–81.

[10] van der Merwe Johann, Dawoud Dawoud, McDonald Stephen. Trustworthy key management for mobile ad hoc networks. In: Proceedings southern African telecommunication networks and applications conference (SATNAC); 2004. p. 6.

[11] van der Merwe Johann, Dawoud Dawoud, McDonald Stephen. Fully self-organized peer-to-peer key management for mobile ad hoc networks. In: Proceedings of the 4th ACM workshop on wireless security (WiSe); 2005.

[12] Chen Chun, He Daojing, Chan Sammy, Bu Jiajun, Gao Yi, Fan Rong. Lightweight and provably secure user authentication with anonymity for the global mobility network. Int J Commun Syst 2011;24(3):347–62. http://dx.doi.org/10.1002/dac.1158.

[13] Zakhary Sameh R, Radenkovic Milena. Reputation-based security protocol for MANETs in highly mobile disconnection-prone environments. In: The seventh international conference on wireless on-demand network systems and services (IEEE/IFIP WONS); 2010.

[14] Ibriq J, Mahgoub I. HIKES: hierarchical key establishment scheme for wireless sensor networks. Int J Commun Syst 2012(8 November). http://dx.doi.org/10.1002/dac.2438.

[15] Network simulator < http://www.isi.edu/nsnam/ns > .

Saju P. John received the B-Tech Degree in Computer Science from Calicut University, the M.E. Degree in Computer science from Anna University and presently doing part time Ph.D. in Cochin University of Science & Technology, India. He is presently working as Associate professor in Computer Science & Engineering Department, Thejus Engineering College, Thrissur, India 680584. His Current research interests include Network security, Mobile Communication, Adhoc networks, etc.

Philip Samuel received the M.Tech. degree in Computer Science from Cochin University and the Ph.D. degree from IIT Khargpur. He is presently working as Associate Professor in Information Technology Division, School of Engineering, Cochin University of Science & Technology, India 682002. His current research interests include, Networking, Object Oriented Modeling & Design, Software Engineering, Mobile Communication, Adhoc Networks, etc.