Dahshan, Hisham M.K. and Irvine, J. (2010) *A robust self-organized public key management for mobile ad hoc networks.* Security and Communication Networks, 3 (1). pp. 16-30. ISSN 1939-0114

A Robust Self-Organized Public Key Management for Mobile Ad Hoc Networks

Hisham Dahshan and James Irvine
Department of Electronic & Electrical Engineering
University of Strathclyde
Email: hisham.dahshan@strath.ac.uk, j.m.irvine@strath.ac.uk

March 2, 2009

Abstract

A mobile ad hoc network MANET is a self-organized wireless network where mobile nodes can communicate with each other without the use of any existing network infrastructure or centralized administration. Trust establishment and management are essential for any security framework of MANETs. However, traditional solutions to key management through accessing trusted authorities or centralized servers are infeasible for MANETs due to the absence of infrastructure, frequent mobility, and wireless link instability. In this paper we propose a robust self-organized, public key management for MANETs. The proposed scheme relies on establishing a small number of trust relations between neighboring nodes during the network initialization phase. Experiences gained as a result of successful communications and node mobility through the network enhance the formation of a web of trust between mobile nodes. The proposed scheme allows each user to create its public key and the corresponding private key, to issue certificates to neighboring nodes, and to perform public key authentication through at least two independent certificate chains without relying on any centralized authority. A measure of the communications cost of the key distribution process has been proposed. Simulation results show that the proposed scheme is robust and efficient in the mobility environment of MANET and against malicious node attacks.

keywords: Key Management, Mobile Ad Hoc Network

1 Introduction

A mobile ad hoc network MANET is a self-organized wireless network where mobile nodes can communicate with each other without reliance on a centralized authority. Each node is able to communicate with other nodes within its transmission range and relies on other nodes to communicate with nodes outside its transmission range. In cellular networks, communications between two mobile nodes completely rely on the wired backbone and the fixed base stations. In a MANET, no such infrastructure exists and the network topology may dynamically change in an unpredictable manner since nodes are free to move [1]. The absence of centralized administration and the infrastructureless nature make MANETs good for emergency, military and fast deployment communications.

The security of most conventional networks relies on the existence of a specialized network administration that defines the security policy and provides the infrastructure for implementing it. The lack of any centralized network management or certification authority makes MANET very vulnerable to infiltration, eavesdropping, interference, and so on. Security in MANET is an essential component to supply the network with the basic functions such as routing and packet forwarding. If a priori trust relations exist between the nodes of an ad hoc network, entity authentication can be sufficient to assure the correct execution of critical network functions. A priori trust can only exist in a few special scenarios like military networks and corporate networks, where a common, trusted authority manages the network [1]. With the lack of a priori trust, classical network security mechanisms can not be used in MANETs. The trust relationships established between network nodes could be used for the provision of higher level security solutions, such as key management. Key management is a basic part of any secure communication. Secure network communications normally involve a key distribution procedure between communication parties, in which the key may be transmitted through insecure channels. A framework of trust relationships needs to be built for authentication of key ownership in the key distribution procedure.

In MANET, key management can be classified into two kinds; the first one is based on a centralized or distributed trusted third party (TTP). The TTP is responsible for issuing, revoking, renewing, and providing keying material to nodes participating in the network such as [2], [3], and [4] where the key management process is performed using threshold cryptography [5]. In the (n,t) threshold cryptography, a secret key is divided into n shares according to a random polynomial and kept by n legitimate nodes, which we call share holders. Later, a new node needs to collect t shares from the response of t nodes (among n nodes) based on Lagrange interpolation and generates the original secret key as a legitimate node.

The second kind of key management is the self-organized key management schemes, such as [6], and [7]. Self-organized schemes allow nodes to generate their own keying material, issue public-key certificates to other nodes in the network based on their knowledge. Certificates are stored and distributed by the nodes. Each node maintains a local certificate repository that contains a limited number of certificates selected by the node according to an appropriate algorithm. Public-key authentication is performed via chains of certificates.

In this paper, we propose a robust self-organized public key management for MANETs. The proposed scheme is based on the existence of a web of trust between mobile nodes forming the network. Since the use of random graph theory in MANET is limited to quasi-static networks due to constraints and nodes mobility [8], random graph theory is used in the proposed scheme to represent the public keys and certificates of the system in the initialization phase only. The proposed scheme allows each user to create its public key and the corresponding private key, to issue certificates to neighboring nodes, and to perform public key authentication without relying on any centralized authority. Each node in the network has a trust table to store the public key certificates and the corresponding trust values. A trust value represents a node's belief that another node is trustworthy. The certificate chain discovery will be performed with the aid of the routing process. A measure of the communication cost is proposed. A comprehensive analysis of the proposed scheme in the mobility environment of MANETs will be performed. The rest of the paper is organized as follows: Section 2 reviews related work. Section 3 provides the system description and trust model of our proposed scheme. The simulation environment and performance metrics are described in Section 4 and then the results are presented in Section 5. Finally Section 6 concludes the paper.

2 Related Work

In this section a review of key management schemes for MANETs will be presented. In [2], and [9], threshold cryptography has been proposed to provide a reliable, distributive key management for MANET by exploiting some nodes as trust anchors for the rest of the network. In these schemes, threshold cryptography involves additional computationally intensive modular exponentiation compared to the underlined asymmetric-key cryptographic protocols. Most low-powered wireless nodes do not have the resources to handle such computationally intensive operations. Capkun et al. in [4] proposed a self-organized public key management scheme in which each node issues certificates independently and manages them at its repository. In this scheme, certificates are stored and distributed by the nodes and each node maintains a local certificate repository that contains a limited number of certificates selected by the node according to an appropriate algorithm. Key authentication is performed via chains of certificates. However, this scheme suffers from the delay and the large amount of traffic required to collect certificates. Kitada et al. in [10], and [11] considered the problem of certificate chain discovery by introducing the Ad hoc Simultaneous Nodes Search protocol (ASNS) to find a certificate chain. In the proposed scheme by Kitada et al., each node holds in its local repository only certificates issued to it in order to reduce the memory size and collects certificates by broadcasting search packets to chained nodes. The scheme suffers from high communication cost because of broadcasting packets with certificates. Li et al in [12] proposed a public key management scheme performed by generating a public/private key pair by the node itself, issuing certificates to neighboring nodes, holding these certificates in its certificate repository. This scheme considers only the updated certificate repository to reduce the number of certificates stored in its certificate repository. Ren et al. in [13] proposed a distributed trust model based on introducing a secret dealer to accomplish trust initialization in the system bootstrapping phase to overcome the problem of delayed trust establishment as in [4]. In this scheme a secret dealer provides each node with a secret short list includes a number of entries, and the number of entries is determined according to the group size n and may vary slightly from node to node. Each entry contains a binding of node identifier and its corresponding public key: (ID, P_k). After receiving the short list, the following conditions should be met:

- Every node in the network receives a secret short list SL, which contains n semi-randomly selected (ID,PK) pairs;
- The (ID,PK) pairs are distributed symmetrically. If node i gets the (ID,PK) pair of node j, then the (ID,PK) pair of node i is also included in the secret short list

SL of node j's, that is, $(ID_i, PK_i) \subset SL_j$, if $(ID_j, PK_j) \subset SL_i$, where SL_i is the short list obtained by node i.

After that, each node starts to issue certificates for the received bindings and store them locally. The existence of the secret dealer makes the scheme prone to the centralized administration problems. For example, it has a single point of attack because if the secret dealer is compromised during the bootstrapping phase, the security of the whole system will be at risk. The performance of the scheme in the mobility environment of MANETs is low as will be shown in section 5 because the certificate graph, which is used to model this web of trust relationship, may not be strongly connected.

3 System Description And Trust Model

In this section we present an overview of the trust model and the system description of our proposed scheme.

3.1 Trust Model

3.1.1 Certificate Chain

The trust model of our proposed scheme is based on the existence of public-key certificates as bindings of the public keys and the corresponding user identities IDs. The certificate should also contain the node's identity/network address, sequence number, trust value, certificate generation and validity dates.

We denote $Cert_{A \to B}$ as the certificate signed by node A's private key SK_A to represent its assurance in the binding of node B and its public key PK_B. For simplicity we denote Sig_i as the digital signature of node i. A certificate graph $G(V,E)$, represents the public keys and certificates of the system in the initialization phase, where V and E stand for the set of vertices and the set of edges, respectively. The vertices of the certificate graph represent public keys and the edges represent certificates. A directed edge from node A to node B will exist if there is a certificate signed with the private key of node A that binds node B's identity ID_B and its public key PK_B. In order for a node A to authenticate the public-key of another node D as shown in Figure 1, it has to acquire a chain of valid certificates from node A to node D. The first certificate in the chain is a certificate issued by node A, so that it will be verified by node A by using its public key PK_A. Each remaining certificate in the chain will be verified using the public key of the previous certificate in the chain. The last certificate in the chain holds the public key of the target node D. The certificate chain from node A to node D in this example is $\{Cert_{A \to B}, Cert_{B \to C}, Cert_{C \to D}\}$, and the certificate chain from node D to node A is $\{Cert_{D \to C}, Cert_{C \to B}, Cert_{B \to A}\}$.

It is assumed that there exist sparse trust relationships among the nodes so that any node that wishes to join the network can establish independent trust relationship with some of the existing member nodes in the network. For example, a node that wishes to join the network contacts one of the existing network members through secure side channels and provides its trust evidence. If the existing network member believes that the requesting node is trustworthy according to its trust evidence, they can sign and exchange certificates. The process is repeated until the joining node gets a sufficient number of certificates.

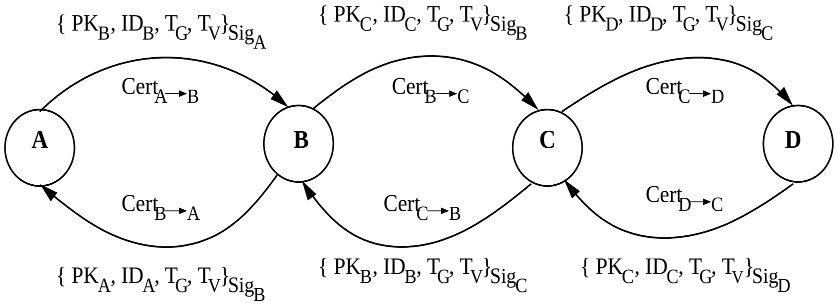


Figure 1: Certificate Chain

3.1.2 Trust Evaluation

We define a trust value as an authentication metric. This trust value represents the assurance with which a requesting node can obtain the correct public key of a target node. However, the same assurance in the reverse direction need not exist at the same time. In other words, the trust relationship is unidirectional. Each node in the network should have a trust table as shown in table 1 to store the public-key certificates and the corresponding trust values of the nodes it trusts in the network. There are many trust metrics have been proposed to evaluate the trust values, some assume discrete trust values as in PGP [14]. Others assume continuous values for trust [3]. In our trust model, we define the trust value as a continuous value between 0 and 1. A trust value $T_{i,j}$ represents node i's belief that node j is trustworthy. The higher the value of $T_{i,j}$, the more node i trusts node j, and vice versa. Any node in the network can calculate the value of trust $T_{i,j}$ in another node's public key if there exist a certificate chain between the two nodes using the following formula:

$$T_{i,j} = \prod_{k=1}^{k=h} T_k \tag{1}$$

where T_k is the value of trust between two directly trusted nodes along the certificate chain from node i to node j, and h is the number of hops between node i and node j as shown in Figure 2.

It is obvious from Figure 2 that the value that a node trusts in another node's public key fades along the path of recommendation. So, if there is more than one certificate chain between a pair of nodes, the source node should choose the path with higher trust value, which is more probably the path with minimum number of hops.

3.1.3 Independent Trusts

The previous work on independent trust evaluation presented in is extended in our proposed scheme. Dependence between certificate chains leads to inaccurate trust values.

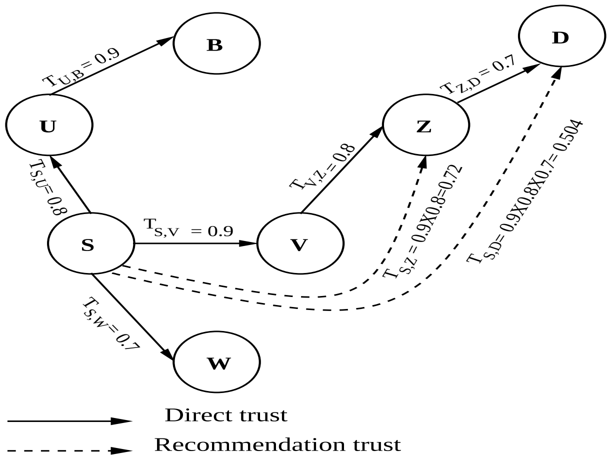


Figure 2: The Trust Model

For example, if a malicious node is a part of dependent certificate chains, the trust computation result will be completely incorrect. In order for a source node to authenticate the destination node's public key in the adversary environment of MANET, our proposed key management scheme insists on obtaining the destination node's public key through at least two independent certificate chains. In order to introduce the independent certificate chains in computing the trust value of the target node, formula 1 will be modified as follows:

$$T_{i,j} = \bigcup_{l=1}^{l=n} (\prod_{k=1}^{k=h} T_k) \tag{2}$$

Where n is the number of independent certificate chains between node i and node j.

From the inclusion-exclusion principle, the probability of the union of two events can be computed by taking the sum of their probabilities, and subtracting the probabilities resulting from intersecting as shown in formula

$$p(A \cup B) = p(A) + p(B) - p(A).p(B) \tag{3}$$

In our proposed scheme, A and B are the trust values computed from the first and the second independent certificate chains respectively.

The presence of independent certificate chains in the authentication process results in an increase in the computed trust value of the target node as shown in the example in Figure 3.

The values that the source node S trusts in the destination node D from the first and the second certificate chains are $T_{S,D,1} = 0.504$, and $T_{S,D,2} = 0.703$ respectively. According to formulas 2, and 3, the overall trust value will be computed as follows:

$T_{S,D} = 0.504 + 0.403 - 0.504 \, . \, 0.403 = 0.703$

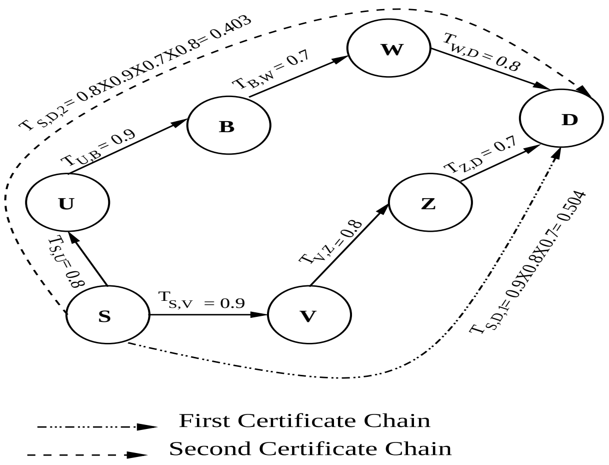


Figure 3: Independent Certificate Chains

It is obvious from the above example that the presence of independent certificate chains has increased the trust value more than the trust values computed from a single certificate chain, which is an advantage of our proposed key management scheme.

3.2 System Description

The proposed robust self-organized public key management scheme involves four processes as shown in Figure 4: public key and public-key certificate generation, certificate chain discovery, certificate verification, and certificate revocation. We describe our proposed scheme in detail according to these four processes.

1. Public Key and Public-Key Certificate Generation

In our proposed scheme, each node generates its public key and the corresponding private key locally before joining the network by the node itself. Public-key certificates are issued by the nodes based on node information about other nodes in the network. If a node u believes that a public key PK_i belongs to a certain user i, it has to issue a certificate to node i signed by its private key PK_u representing its assurance of the binding of the user's identity ID_i and its corresponding public key PK_i. Issued certificates to or from the node should be stored in its trust table with a validity time.

2. Certificate Chain Discovery

In this process, the certificate chain discovery is performed by exploiting the routing infrastructure. We assume that a certain number of direct trust relations have been established between each node and its neighbors during the network initialization. Direct trust relations are usually obtained off-line by visual identification, audio exchange through side channels, and physical contact, but can also be obtained on-line. The number of directly trusted nodes per node is assumed to be uniformly distributed to enable

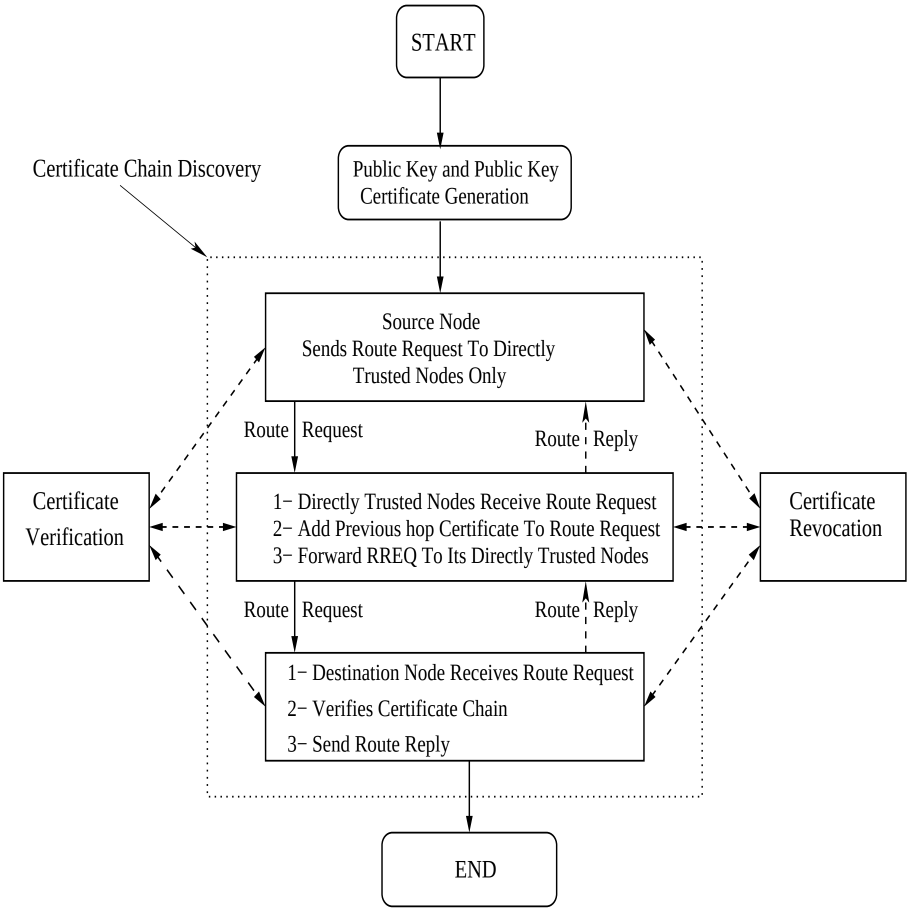

START
Public Key and Public Key Certificate Generation
Certificate Chain Discovery
Source Node Sends Route Request To Directly Trusted Nodes Only
Route Request
Route Reply
Certificate Verification
1− Directly Trusted Nodes Receive Route Request
2− Add Previous hop Certificate To Route Request
3− Forward RREQ To Its Directly Trusted Nodes
Certificate Revocation
Route Request
Route Reply
1− Destination Node Receives Route Request
2− Verifies Certificate Chain
3− Send Route Reply
END

Figure 4: Scheme Description

every node in the network to carry out its role in the certificate chain discovery process with equal probability. We assume that a random graph $G(n, p)$ is a graph of n vertices and the probability that a connection exists between any two vertices is p. In [15], Erdos and Renyi showed that there exists, for monotone properties, a value of p such that the property transfers from ''nonexistent'' to ''certainly true'' in a random graph that has a very large number of vertices. However, Frank and Martel have shown by simulation in [16] that these properties are also valid in graphs of moderate size (between 30 to 480 vertices).

Erdos and Renyi showed that if $p = ln(n)/n + c/n$, where c is a real constant then

$$\lim_{n \to \infty} Pr[G(n, p)\, connected] = e^{-e^{-c}}.$$

Therefore, given the network size n we can find p and the average degree of a node (number of trusted neighboring nodes) $d = p.(n-1)$ for which the resulting graph is connected with the desired probability $Pr[G(n, p)\, connected]$. Figure 5 illustrates the plot of the expected degree of a node d as a function of the network size n for various values of $Pr[G(n, p)\, connected]$. For example if the desired probability to get a connected graph for 50 nodes network is 0.99, then the average number of neighboring nodes d should be 6, and for 100 nodes network d should be equal to 10 for the same probability. The proposed robust self-organized public key management scheme works in support of an ad hoc on demand routing protocol (such as AODV [17]) after doing the necessary modifications. The RREQ and the RREP messages format of the AODV routing protocol have been modified as shown in Figures 6, and 9 respectively in order to implement the proposed key management scheme. The certificate chain discovery process can be explained as follows:

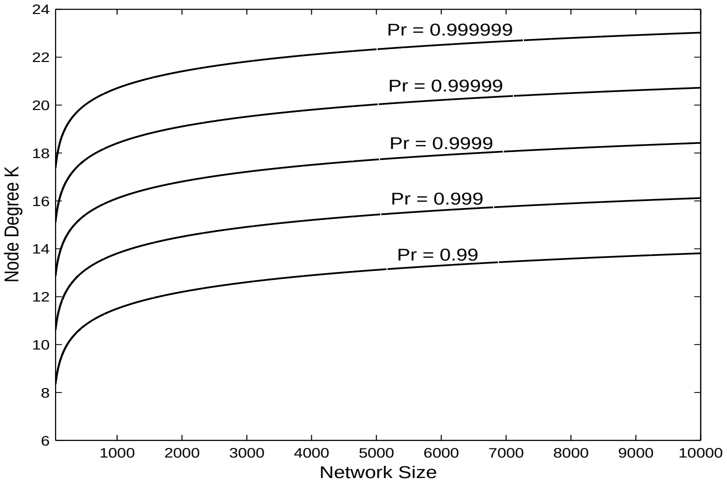


Figure 5: Degree of node d vs. network size for various $Pr[G(n, p)\, connected]$

- The source node sends a route request packet to nodes that the source node directly trusts.

- When a directly trusted node by the source node receives the route request packet from the source node, it searches for the certificate of the source node signed by this node in its trust table and adds it to the route request packet before forwarding the route request packet to nodes it directly trusts.

- Each node has a trust reply table as shown in table 3.

- When an intermediate node receives the route request packet from a node it trusts, it sends a route reply to the source node if it has a fresh route to the destination node.

- If the intermediate node does not have a fresh route to the destination node, it searches for the certificate of the sender node in its trust table, and adds it to the route request packet.

- Before forwarding the route request packet to nodes it directly trusts, the intermediate node inserts the ID of the source node, the ID of the destination node, the RREQ ID, and the ID of the sender node in its trust reply table.

- When the destination node receives a route request packet from a node it directly trusts, it verifies the certificates included in the route request packet and inserts the ID of the source node, the RREQ ID, and the ID of the sender node in its trust reply table.

- The destination node waits until it receives at least another route request, repeats the previous step and makes sure that the certificate chain included in the second route request is completely independent from the certificate chain included in the first received one (i.e. there is no intersecting intermediate node in both routes).

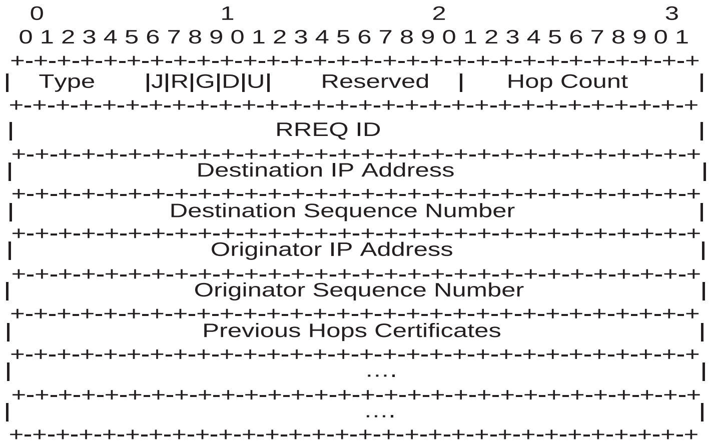
- The destination node sends a route reply to the senders of this route request according to its trust reply table.

- When an intermediate node receives the route reply, it adds the certificate of the node it received the route reply from to the route reply and forwards the route reply to the sender nodes of the corresponding route request according to its trust reply table.

- The process continues until the route reply reaches the source node.

- When the source node receives the route reply it verifies the certificate chain; computes the trust value of the chain; inserts the RREQ Id, certificate chain entities ID's, and the computed trust value of the chain in the route request table as shown in table 2.

- The source node waits until it receives a route reply from at least another independent certificate chain, and according to the chain trust value the source node chooses a chain to start data transmission to the destination through it. The



```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |     Type      |J|R|G|D|U|   Reserved    |   Hop Count   |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                            RREQ ID                            |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                    Destination IP Address                     |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                  Destination Sequence Number                  |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                    Originator IP Address                      |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                  Originator Sequence Number                   |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                  Previous Hops Certificates                   |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                             ....                              |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                             ....                              |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: RREQ Message Format

source node examines the dependency of the received certificate chain by searching for any intersecting node in the chains stored in its route request table.

Successful communication result in increasing trust between certificate chain entities and signing more certificates for each others. Node mobility through the network results in new trust relations and more certificates are added to the node's trust table.

At the end of the route reply process, the source node S receives two independent certificate chains as shown in the example in Figures 8, and 9 for the route request and the route reply processes respectively. The first certificate chain contains certificates of nodes, B, E, and Q in addition to the destination node's certificate. The second certificate chain contains certificates of nodes, C, F, and K in addition to the destination node's certificate. According to the computed trust values for each certificate chain, the source node S chooses a chain to be used in transmitting data packets to the destination node D. After receiving the second route reply the route request table of the source node S will be as shown in table 4.

3. Public-Key Certificate Verification

The verification of the public-key certificate is performed by checking the validity time T_v in the certificates forming the certificate chain. After verifying the certificates of the certificate chain the authenticity of the public keys of both the source and the destination nodes is performed by the certificate chain from the source node to the destination node and vice versa.

4. Public-Key Certificate Revocation

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Type       |R|A|  Reserved   |Prefix Sz|     Hop Count       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination IP Address                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Destination Sequence Number                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Originator IP Address                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Lifetime                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Previous Hops Certificates                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               ....                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               ....                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: RREP Message Format

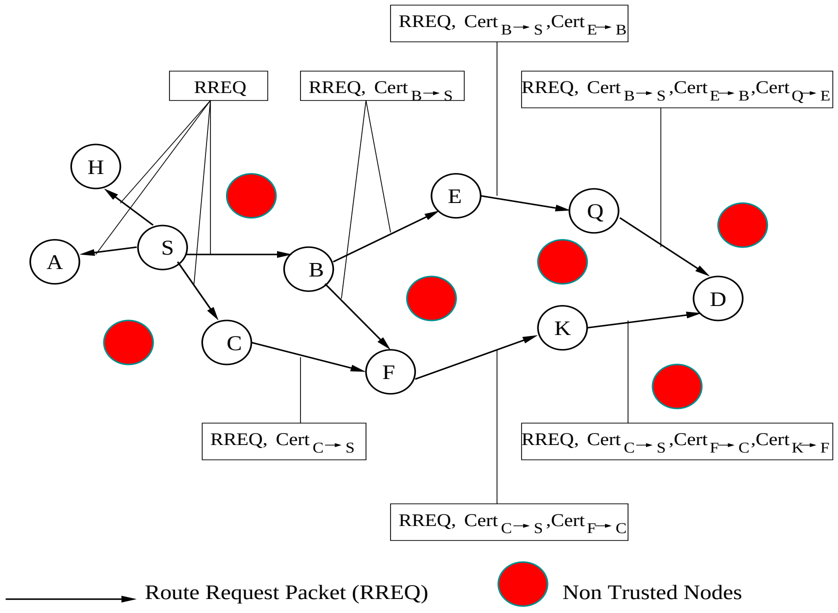

RREQ, Cert$_{B \to S}$,Cert$_{E \to B}$
RREQ
RREQ, Cert$_{B \to S}$
RREQ, Cert$_{B \to S}$,Cert$_{E \to B}$,Cert$_{Q \to E}$
H
E
Q
A
S
B
D
K
C
F
RREQ, Cert$_{C \to S}$
RREQ, Cert$_{C \to S}$,Cert$_{F \to C}$,Cert$_{K \to F}$
RREQ, Cert$_{C \to S}$,Cert$_{F \to C}$
Route Request Packet (RREQ)
Non Trusted Nodes

Figure 8: Route Request Process

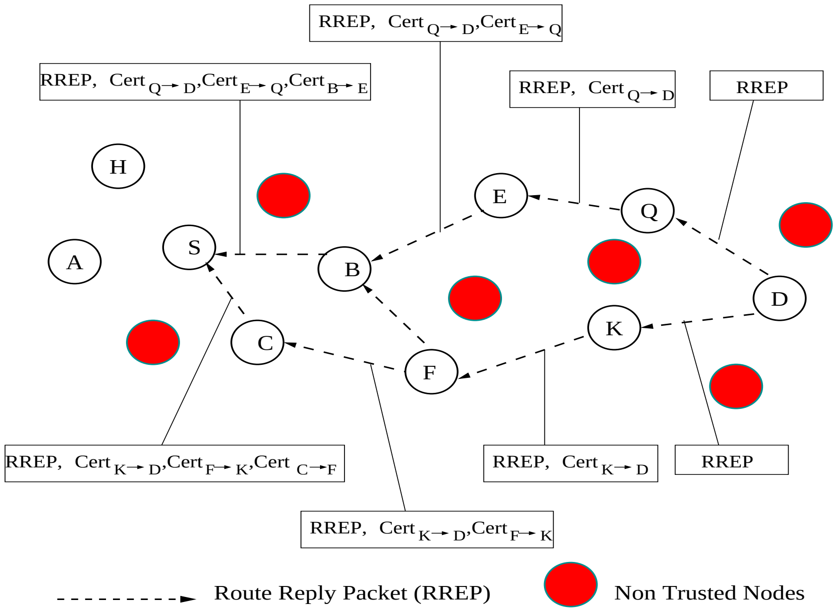


Figure 9: Route Reply Process

Each node can revoke a certificate it issued if it believes that the binding between the public key and the node's identity is no longer valid or the trust value of the target node is below the trust threshold. Each node can revoke its own certificate if it believes that its private key is compromised. A node can revoke its own certificate by broadcasting a certificate revocation message signed by its private key to nodes it directly trusts includes the revoked certificate and the new one. When a node receives the certificate revocation message, it verifies the revocation message and replaces the revoked certificate by the new one. A node can revoke a certificate it issued by broadcasting a certificate revocation message signed by its private key to nodes it directly trusts. When a node receives a certificate revocation message, it verifies the signature of the revocation message, searches in its trust table for the trust values of the sending node and that of the node its certificate is claimed to be revoked. According to the search results, it will do the following:

- If the search results yields that the trust value of the sending node is greater than the trust threshold (i.e. the sending node is trustworthy), it multiplies the trust value of the sending node by -1.
- Adds the calculated value from the previous step to the trust value of the node its certificate is claimed to be revoked.
- If the sum from the previous step is below the trust threshold, it deletes the revoked certificate from its trust table. Otherwise if the sum is still greater than the trust threshold, the accused certificate is still to be used.
- Otherwise, if the search results yields that the trust value of the sending node is

below the trust threshold (i.e. the sending node is not trustworthy), it ignores the revocation message.

3.3 Communication Cost

The communication cost of our proposed scheme consists of two phases; the route request phase and the route reply phase. In the route request phase, the source node sends the route request packet to nodes it directly trusts without inserting its own certificate in the route request packet as this certificate is stored in the trust table of nodes it directly trusts which saves bandwidth in the first hop. The communication cost of the certificate chain discovery during the route request phase can be calculated as follows:

$$C_{req}(h) = \sum_{i=2}^{i=h} d^i \times (i-1) \times cert\_size$$

Where h is the average number of hops between the source and the destination, d is the average number of trusted nodes per node, and *cert_size* is the size of the certificate.

In the route reply phase, the destination node does not insert its own certificate in the first hop because this certificate is stored in the trust table on nodes it directly trusts which saves bandwidth in the first hop of the route reply. The destination node sends a route reply message to all sender nodes it has received the route request from according to its trust reply table as shown in table 3.

The upper limit of the communication cost in this phase can be calculated as follows:

$$C_{rep}(h) = \sum_{i=2}^{i=h} (i-1) \times d \times cert\_size$$

The total communication cost of the certificate chain discovery process will be calculated as follows:

$$C_{total}(h) \quad = \quad C_{req}(h) + C_{rep}(h)$$

$$C_{total}(h) = \sum_{i=2}^{i=h} d \times (1 + d^{i-1}) \times (i-1) \times cer\_size$$

4 Simulation Environment and Performance Metrics

4.1 Simulation Environment

Simulations were performed using Network Simulator (NS-2) [18], particularly popular in the ad hoc networking community. on a desktop with an Intel cor 2 Duo 2.6 GHz processor and 1 GB memory. The OpenSSL library (version 0.9.8h) [19] is used for generating certificates, and digital signatures [20]. In order to complete a certificate chain with probability equal to 0.999, the average number of trusted nodes per node need not to be less than 6 as discussed in subsection 3.2. The MAC layer protocol

IEEE 802.11 is used in all simulations. 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 and can generate at most 10,000 packets. Sources initiation time is uniformly distributed over the first 90 seconds of the simulation time. Every simulation run is 1000 seconds long. The mobgenss [21] mobility scenario generator was used to produce random waypoint mobility patterns [22]. The random waypoint mobility model is widely used model when evaluating MANETs [23, 24, 25, 26] and hence is considered in this paper. The pause time is set to zero. The Ad Hoc On-demand Distance Vector (AODV) routing protocol [17] was chosen for the simulations. The simulation results are the average of 10 runs. The rest of the simulation parameters are summarized in table 5. These scenarios are the worst case scenarios because all nodes in the network are transmitting and receiving packets starting from the time of joining the network till the end of the simulation. It is generally accepted that data encrypted using a key size of 2048-bits should be safe. Therefore, the public key would require 256 bytes, a signature consists of 128 bytes value, node's identity is of 16 bytes, network address is of 16 byte, sequence number is of 16 bytes, trust value is of 16 bytes, and the certificate expiration time can be encoded in 2 bytes. In the simulation, a certificate < IDA ; Network Add; *PK*; Sequence #; Expiration time; trust value; signature; 62 bytes reserved> will be a total of 512 bytes in length.

4.2 Performance Metrics

We have selected the packet delivery ratio, average end-to-end delay, the number of certificates delivered through the network, certificate chain completion ratio, average hop length, and the routing overhead as metrics during the simulation in order to evaluate the performance of the proposed robust self-organized public key management for mobile ad hoc networks.
Packet delivery ratio (PDR): The ratio of data packets delivered to destinations to those generated by sources.
Average end-to-end delay: This includes all possible delays caused by buffering during route discovery latency, queuing at the interface queue, retransmission delays at the MAC, and propagation and transfer times.
Number of certificates delivered through the network: the total number of certificates delivered through the network during the certificate chain discovery processes.
Certificate chain completion ratio: is the ratio of the average number of independent certificate chains received by a node to the number of transmitted route requests.
Routing Overhead: The ratio of the total number of routing packets to data packets transmitted during the simulation.

5 Performance Evaluation

In this section, the simulation results will be presented. The performance evaluation of the proposed scheme will be performed in two steps, in the first step, a comparison of the performance of the proposed scheme, the performance of Ren et al. web of trust scheme [13], and the CBR reference which is the simulation of the AODV [17] without any modifications are presented. The performances of the three mentioned schemes are measured for different numbers of trusted nodes per node. In the second step the dynamical characteristic of the proposed scheme will be presented by measuring the performance of the proposed scheme in a realistic environment for different numbers of nodes joining and leaving the network.

5.1 Performance of the Proposed Scheme

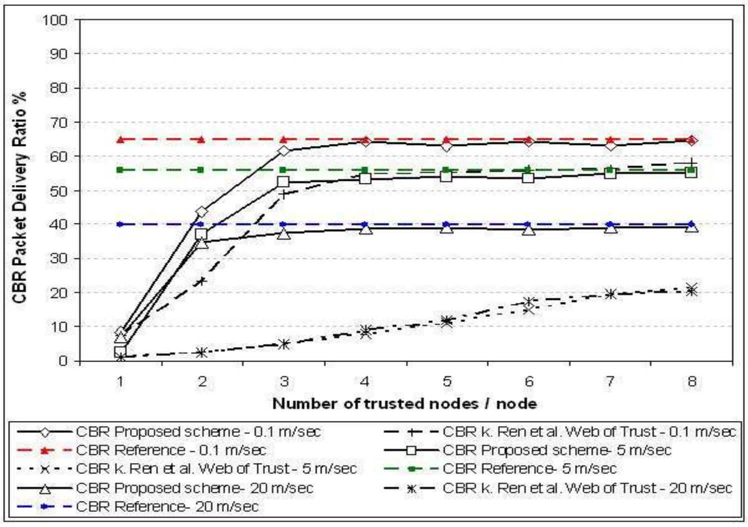


Figure 10: Packet Delivery Ratio PDR % vs Number of trusted nodes / node

Figure 10 shows that the packet delivery ratio increases with increasing the number of trusted nodes per node d. In our proposed scheme, discovering a certificate chain is performed through trusted nodes only. However, increasing the number of trusted neighbors d increases the probability for a node to discover a certificate chain which results in an increase in the packet delivery ratio. Increasing the number of trusted nodes per node d to 6 at the network initialization phase (as expected in our setting) makes the packet delivery ratio approaches that of the AODV reference. It shows also that Ren et al. scheme is not robust in the mobility environment as increasing the node's speed from 0.1 m/ sec to 5 m/sec and 20 m/ sec degrades the packet delivery ratio too much.

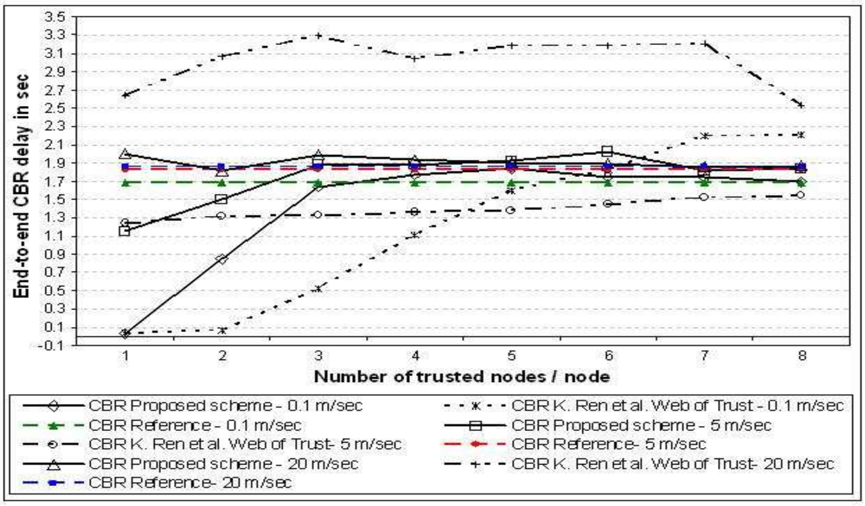


Figure 11: End-to-end Delay in sec vs Number of trusted nodes / node

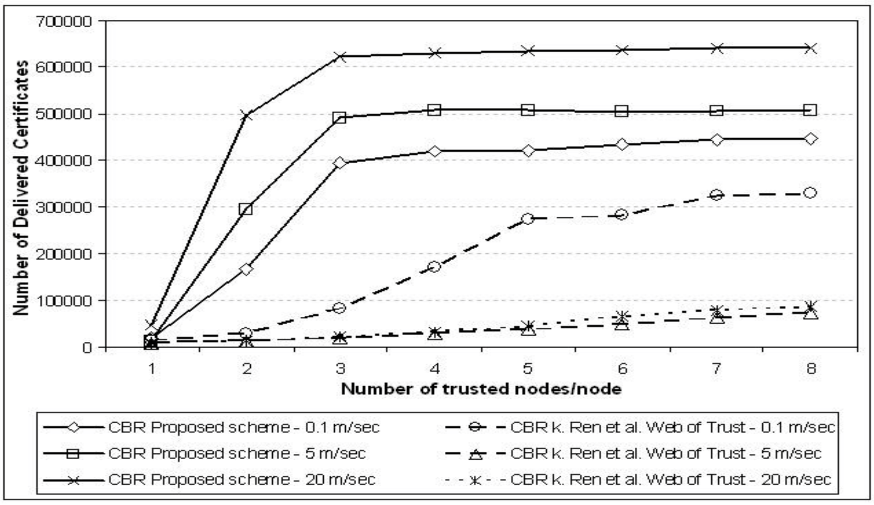


Figure 12: Number of certificates delivered through the network

Figure 11 shows that the end-to-end delay of our proposed scheme for d equal to 6 approaches the end-to-end delay of AODV reference, and the end-to-end delay of Ren et al. is far away from the end-to-end delay of AODV reference because of the successive failures to find certificate chains. Figure 11 shows also that the end-to-end delay in our proposed scheme increases with increasing the number of trusted nodes per node d. In order to understand why the end-to-end delay increases with increasing d we should put Figure 13 into context which shows the average hop length (AHL) as a function of the number of trusted nodes per node. It is obvious that the AHL increases with increasing d. By increasing d, data and control packets traverse higher number

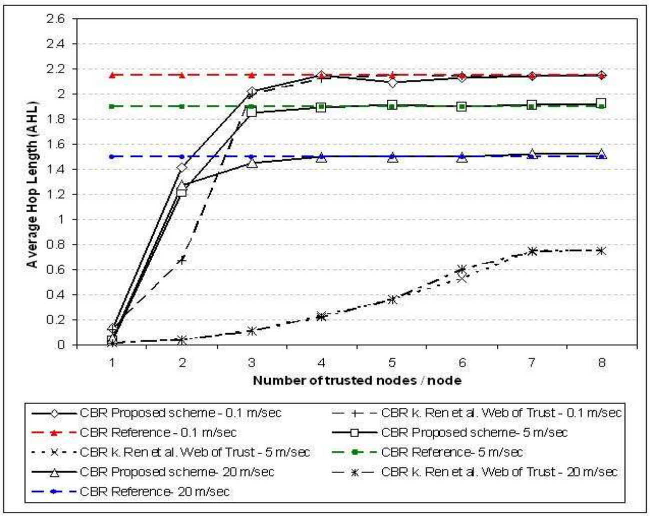


Figure 13: Average Hop Length

of hops than that in case of lower *d* which result in higher end-to-end-delay. Figure 12 shows that the number of delivered certificates through the network increases as *d* increases as expected. It also shows that the number of delivered certificates of Ren et al. scheme is lower than that of our proposed scheme.

Figure 13 shows the certificate chain completion probability as a function of the number of trusted nodes per node. Increasing the value of *d* increases the certificate chain completion probability. For *d* equal to 8, nodes can certainly complete a certificate chain. Completing a certificate chain in Ren et al. scheme for 0.1 m/ sec node mobility does not exceed 0.68 and becomes worse with increasing nodes mobility. It is obvious from Figures 10, 11, 12, and 13 that when *d* is less than three it is very difficult for a node to get two independent certificate chains as a reply to its request which causes a degradation on the network performance.

5.2 The Dynamic Characteristics of the Proposed Scheme

In this subsection, the performance of the proposed scheme is evaluated with regard to different numbers of nodes joining and leaving the network which is known as the dynamical characteristics of the network. Nodes joining and leaving the network have been simulated in ns-2 [18] by changing the initial energy of the Mobile node in its energy model.

The processes of joining and leaving the network occur simultaneously while keeping the network size unchanged. The performance of the network is measured after each pair of nodes leaves or joins the network. In order to simulate a realistic network, the simulation environment used in subsection 4.1 will be the same except that the number of connections is set to 25 rather than 50.

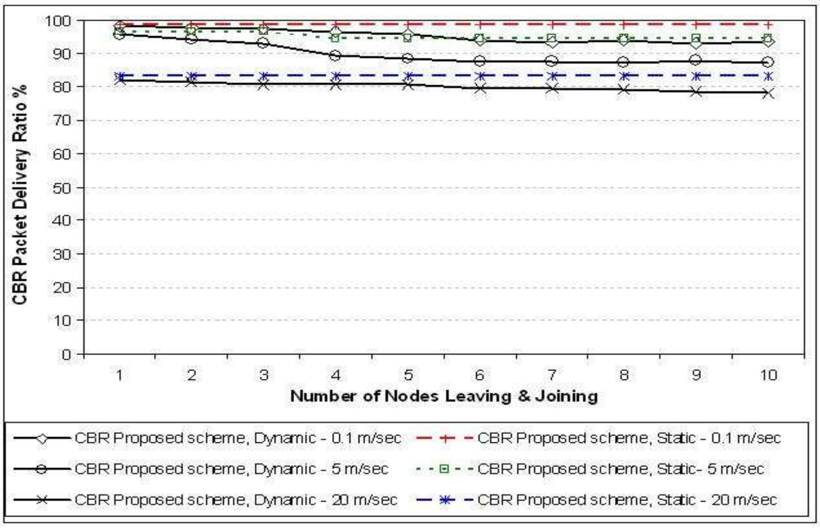


Figure 14: Packet Delivery Ratio PDR % vs Number of Nodes Leaving and Joining

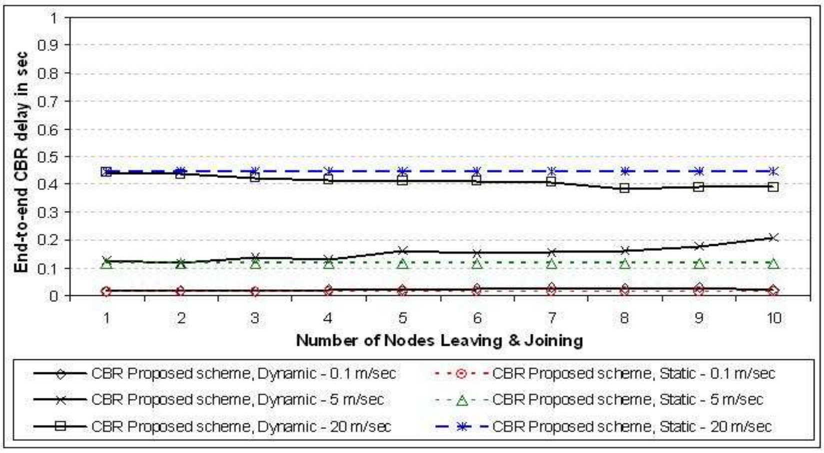


Figure 15: End-to-end Delay in sec vs Number of Nodes Leaving and Joining

Figure 14 shows that the packet delivery ratio of our proposed scheme has changed little even after the number of nodes leaving and joining the network approaches 10, i.e. 20 % of the network size. The packet delivery ratio in this scenario is higher than the packet delivery ratio in Figure 10 due to decreasing the number of connections to 25 rather than 50. Decreasing the number of connections result in an increase in the packet delivery ratio as a result of decreasing the radio interference and collisions between nodes caused by the hidden/exposed terminals problems. The change in the end-to-end delay of our proposed scheme with increasing the number of nodes leaving and joining the network was low even for 20 % of nodes leaving and joining the network as shown in figure 15.

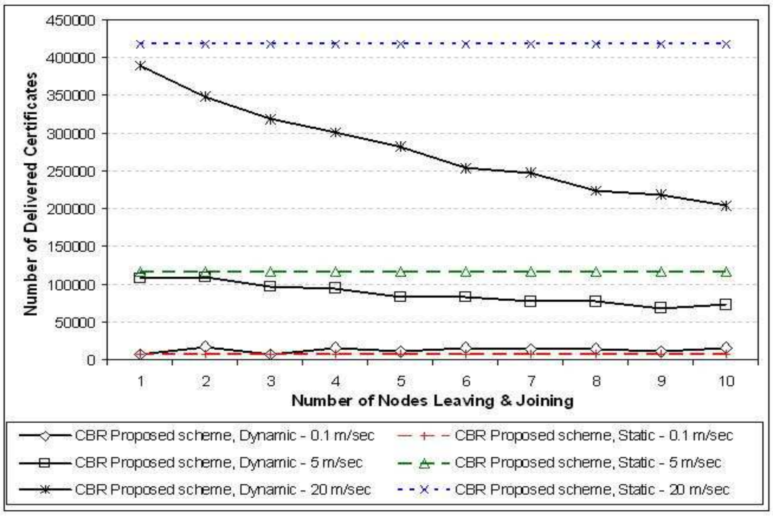


Figure 16: Number of certificates delivered through the network vs Number of Nodes Leaving and Joining

The number of certificates delivered through the network decreases with increasing the number of nodes leaving and joining the network as shown in figure 16.

5.3 Security Analysis

In this subsection, we study the security of the proposed key management scheme. In the network initialization phase, an attacker can compromise a mobile node and try to deceive other nodes by making them believe in false certificates. First, the malicious node can insert false certificates in the certificate chain during the certificate chain discovery process. This can be done by binding a public key PK_A to a user B instead of to a user A. It can also be done by binding a false key $PK_A^{'}$ to a user A. Second the malicious node can delete certificates from the certificate chain to thwart the certificate chain discovery process. Our proposed key management scheme prevents these attacks by allowing mobile nodes to detect false certificates and to perform the public key authentication process with high probability. If a malicious node inserted false certificates in the certificate chain, or deleted correct certificates from the certificate chain during the certificate chain discovery process, the authentication process will not be affected

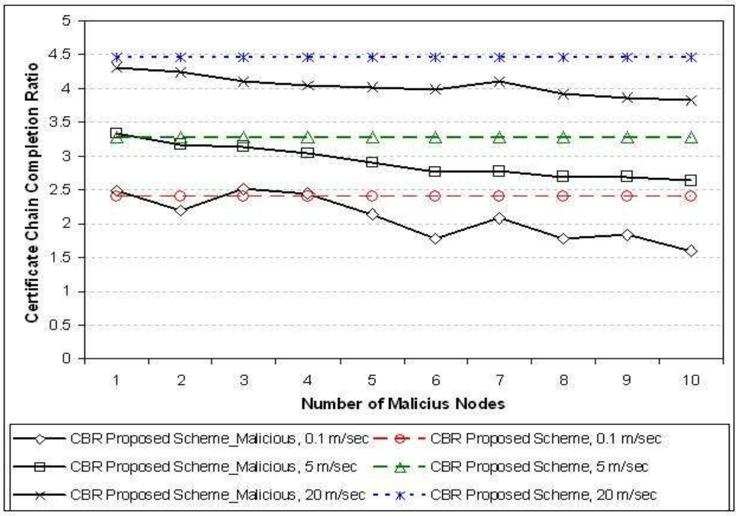


Figure 17: Certificate Chain Completion Ratio

because the source node waits until it receives two independent certificate chains without any intersecting node to perform the authentication process. If a certificate received by a mobile node that binds a node ID A to a public key PK_A and later on received a certificate that binds the node ID A to a public key $PK_A^{'}$, it can easily detect the duplication. The duplication can be resolved by authenticating the (ID, PK_A) pair through another two independent certificate chains. The existence of the trust value along with the certificate enhances the security of our proposed key management scheme by enabling mobile nodes to choose the certificate chain that has the highest trust value. Figures 17 and 18 justified the robustness of the proposed key management scheme against malicious nodes attacks. In Figure 17, the certificate chain completion ratio is calculated for different number of malicious nodes in the network. It shows that the certificate chain completion ratio increases with increasing node's mobility due to decreasing the average hop length with increasing the node's mobility as shown in Figure 13, i.e. the number of certificates in the certificate chain is lower during higher mobility than that in lower mobility which makes the certificate chain discovery process easier. Figure 17 shows also that the certificate chain completion ratio decreases with increasing the number of malicious nodes in the network as a result of inserting false certificates or deleting correct ones. It is obvious from Figure 17 that mobile nodes can perform the certificate chain discovery process and subsequently the public key authentication with high probability even the number of malicious nodes in a 50-nodes network reaches 10, i.e. the number of malicious nodes is 20 % of the network size. Figure 18 shows that the induced routing overhead due to increasing the number of malicious nodes in the network which results in increasing failures in the certificate chain discovery process and an increase in broadcasting certificate revocation messages through the network is negligible for different node's mobility.

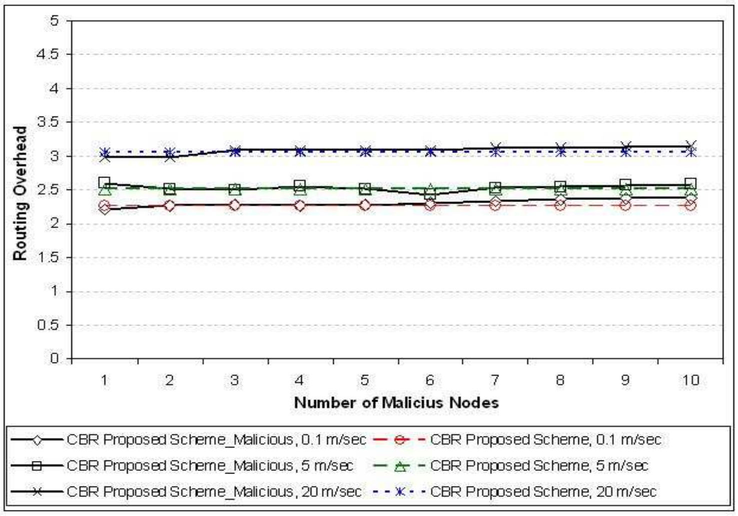


Figure 18: Routing Overhead

6 Conclusions

In this paper, we presented a robust self-organized public key management scheme. The proposed scheme exploits the routing infrastructure to discover a certificate chain through a web of trust. A measure of the communication cost of the certificate chain discovery process has been proposed. The first contribution of this paper is that our proposed scheme has low communication cost because each node limits its search for the certificate chain to its directly trusted nodes only. The second contribution is that the use of trust values along with the public key certificates in at least two independent certificate chains enhances the authentication of the proposed key management scheme. The third contribution is concluded from the performance evaluation of Ren et al. scheme that shows that random graph theory is not suitable for managing trust in the mobility environment of MANET. Random graph theory is suitable for trust management during the network initialization phase or for stationary networks only. The results have shown that our proposed scheme has negligible impact on the network performance and the scheme is suitable for stationary networks and networks with low to high mobility. Simulation results show also that our proposed scheme is robust in both the dynamic and the static networks and against malicious nodes attacks.

References

[1] C. de Morais Cordeiro and D. P. Agrawal, *AD HOC AND SENSOR NETWORKS Theory and Applications*, World Scientific, 2006.

[2] L. Zhou and Z. J. Hass, Securing ad hoc network, *IEEE network*, Vol. 13, no. 6, pp. 24–30, 1999.

[3] E. Ngai, M. Lyu, and R. Chin, An authentication service against dishonest users in *Proceedings of the IEEE Aerospace Conference*, vol. 2, 2004, pp. 1275–1285.

[4] S. Capkun, L. Buttyan, and J.-P. Hubaux, Self-organized public-key management for mobile ad hoc networks, *IEEE Transactions on Mobile Computing*, vol. 2, no. 1, pp. 52–64, 2003.

[5] Y. T. Ronald L. Rivest, Adi Shamir, How to share a secret, *Communication of the ACM*, vol. 22, pp. 612–613, 1979.

[6] S. Capkun, L. Buttyan, and J.-P. Hubaux, Mobility helps peer-to-peer security, *IEEE Transactions on Mobile Computing*, vol. 5, pp. 43–51, 2006.

[7] M. G. Zapata, Key management and delayed verification for ad hoc networks, *Journal of High Speed Networks*, vol. 15, pp. 93–109, 2006.

[8] J. Härri, C. Bonnet, and F. Filali, Kinetic mobility management applied to vehicular ad hoc network protocols, *Computer Communications*, vol. 31, no. 12, pp. 2907–2924, July 2008.

[9] S. Yi and R. Kravets, Moca: Mobile certificate authority for wireless ad hoc networks, *in Proceedings of the 2nd Annual PKI Research Workshop (PKI 2003)*, 2003.

[10] Y. Kitada, Y. Arakawa, On demand distributed public key management using routing information for wirelss ad hoc netwoks, *IEICE Transactions on Information and Systems*, vol. J88 D1, no. 10, pp. 1571–1583, October 2005.

[11] Y. Kitada, A. Watanabe, K. Takemori, and I. Sasase, On demand distributed public key management for wireless ad hoc networks, *in IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PacRim)*, 2005.

[12] H. K. R. Li, J. Li and P. Liu, Localized public key management for mobile ad hoc networks, *in IEEE Global Telecommunications Conference (Globecom)*, 2004, pp. 1284–1289.

[13] K. Ren, T. Lib, Z. Wanb, F. Baob, R. H. Dengb, and K. Kima, Highly reliable trust establishment scheme in ad hoc networks, *The International Journal of Computer and Telecommunications Networking, ElSEVIER*, vol. 45, no. 6, pp. 687–699, 2004.

[14] P. Zimmermann, The Official PGP User's Guide, *M. Press*, Ed. Cambridge, Massachusetts, 1995.

[15] J. Spencer, The Strange Logic of Random Graphs, Algorithms and Combinatorics. *Springer*, Berlin, 2002.

[16] J. Frank and C. U. Martel, Phase transitions in the properties of random graphs, Principles and Practice of Constraint Programming (CP-95), Cassis, France, 1995.

[17] C. E. Perkins and E. M. Royer, Ad-hoc on-demand distance vector routing, *in 2nd IEEE Workshop on Selected Areas in Communication*, New Orleans, LA, 1999, pp. 24–30.

[18] K. Fall and K. Vardhan, The network simulator (ns-2), *Available at: http://www.isi.edu/nsnam/ns.*

[19] OpenSSL cryptography library. *Available online at www.openssl.org.*

[20] A. S. R. L. Rivest and L. M. Adleman., A method for obtaining digital signatures and public-key cryptosystems, Communications of the ACM, vol. 21, no. 2, pp. 120–126, February 1978.

[21] W. Navidi and T. Camp, "Stationary distributions for the random waypoint mobility model", *IEEE Transactions on Mobile Computing*, vol. 3, no. 1, pp. 99–108, 2004.

[22] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva, A performance comparison of multi-hop wireless ad-hoc network routing protocols, in the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking. ACM, Dallas, TX. ACM, October 1998, pp. 85–97.

[23] C. Perkins, Ad Hoc Networking. *Addison-Wesley*, 2001.

[24] E. Royer and C. Toh, A review of current routing protocols for ad-hoc mobile wireless networks, UCSB, Tech. Rep., 1999.

[25] S. R. Das, C. E. Perkins, and E. M. Royer, Performance comparison of two ondemand routing protocols for ad hoc networks, in INFOCOM 2000, Tel-Aviv, Israel, March 2000.

[26] I. A. H. Zafar, D. Harle and Y. Khawaja, Performance evaluation of shortest multipath source routing scheme, IET Communications, 2009.

Table 1: Trust Table

Node ID	Certificate	Trust Value

Table 2: Route Request Table

Chain #	RREQ ID	Destination ID	1st Hop Node ID	2nd Hop Node ID	...	Last Hop Node ID	Trust Value
1st Chain							
2nd chain							
...							

Table 3: Trust Reply Table

Source ID	Destination ID	RREQ ID	Sender 1	Sender 2	...	Sender n
			Node ID	Node ID	Node ID	Node ID

Table 4: Route Request Table Of the Source Node *S*

Chain #	RREQ ID	Destination ID	1st Hop Node ID	2nd Hop Node ID	Last Hop Node ID	Trust Value
1st Chain	1	D	B	E	Q	Value 1
2nd chain	1	D	C	F	K	Value 2

Table 5: Simulation Parameters

Parameter	Value
No. of Nodes	50
Area (m2)	1000x1000
Transmission range	250m
Mobility Model	Random waypoint
Propagation Model	TwoRayGround
Mean speeds (m/s)	0, 5, 20
Data Rate	11 Mbps
Load	4 packet/s
CBR connections	50
Data packet size	512 bytes