CHAPTER 3

ECC OVER BINARY FIELDS FOR MOBILE AD HOC NETWORKS

3.1 PREAMBLE

A Mobile Ad hoc Network (MANET) is an autonomous system in which all the mobile nodes are connected by wireless links. It does not have a static infrastructure such as base stations, routers and switches for communication. The mobile nodes are moving freely causing the network topology to change dynamically. If a direct path is not available between the source and the destination, the intermediate nodes should forward the traffic. Hence, routing protocols designed for ad hoc networks should be adaptive. Recently Mobile Ad hoc Networks (MANETs) have received tremendous attention in a number of applications which includes emergency medical care and military services where security of data is very important. Security design in ad hoc networking should overcome number of challenges which includes scalability, effective routing, mobility and power management [25]. Consequently, the security solutions for wired networks do not directly apply to MANET domain. Also, the existing wireless security protocols mainly focus on security strength and leave the network performance unaddressed. In fact both dimensions of security strength and network performance are equally important and achieving a good trade-off between two extremes is vital for security design in MANETs. Taking into consideration the above challenges, the security solution for MANETs should provide a security service which implements both authentication and key management along with secure end-to-end routing to mobile users.
3.2 LIMITATIONS OF THE EXISTING KEY MANAGEMENT PROTOCOLS

The following are the limitations imposed by the existing symmetric and asymmetric key [85] management protocols in MANETs:

i) Symmetric key distribution requires a Centralized Authority (CA) for authentication and key management among nodes.

ii) Secret keys have to be stored in key pool.

iii) Frequent key refreshment is needed by the CA.

iv) Authentication process is time consuming and increases communication overhead.

v) For larger networks the average number of hops to the CA increases which means the energy consumed for key requests and replies increases.

vi) Asymmetric key distribution like RSA requires larger key sizes.

vii) Increased computational cost.

viii) Increased power consumption and end-to-end delay.

3.3 PROPOSED PROTOCOL

To overcome the above limitations, this research work proposes a Mutual Authentication and Session Key Management protocol for cluster based MANETs using ECC over binary fields and MAC. Since ECC uses smaller key sizes compared to other asymmetric key distribution techniques like RSA and DH, the computational cost is reduced. Power overhead and successful secure link establishment among the nodes due to high mobility are the other challenges to be analyzed using ECC. The scalability of the network was achieved by using the prediction based hierarchical network clustering model [34]. For secure end-to-end route discovery the AntNet routing mechanism [86] was implemented by estimating the trust values of the authenticated neighbourhood nodes using Trust model [87].
The proposed protocol achieves the following objectives:

- Clustering in MANETs using the prediction based hierarchical clustering network model.
- Mutual Authentication and Session Key Management using ECC and MAC in cluster based ad hoc networks and
- Secure end-to-end route discovery using AntNet routing mechanism among the authenticated neighbourhood nodes by estimating the trust values using Trust model.

The efficiency of the proposed protocol is analyzed by factors like low power overhead, high speed, scalability, stability and successful secure link establishment among the authenticated neighbourhood nodes.

3.4 SCALABLE CLUSTER FORMATION

Figure 3.1 shows the hierarchical structure of clustering model for cluster formation in MANETs.

![Hierarchical Structure of Clustering Model](image)
The scalable structure implemented for MANETs has mobile nodes in the form of clusters based on a hierarchical network model in which the proposed mutual authentication and session key distribution protocol is imposed. As shown in Figure 3.1, in this model the entire network was divided into many clusters. Cluster Head (CH) manages each cluster. All CHs are connected to the Network Manager (NM). CH is an ordinary communication node with additional tasks which includes collecting and processing of data from their cluster members and forwarding the processed data to the NM. The CH in a cluster would change depending upon the node mobility. For the formation of stable clusters in the hierarchical clustering approach, the $P_{sk}, T_{sk}, D_{sk}$ - Proactive clustering model [38] was implemented.

3.5 MUTUAL AUTHENTICATION AND SESSION KEY MANAGEMENT PROTOCOL

Most of the security algorithms for session authentication are based on sharing of secret keys between two parties. They use the session secret keys to authenticate each other and also to encrypt and decrypt the data. Due to the lack of centralized administration, constructing and negotiating these secret keys is a challenging task in MANETs. A new protocol to construct this shared session secret key among host devices has been devised by Multicast link independent authentication and session secret key negotiation algorithm using ECC.

In this proposed protocol, mutual authentication among the nodes is achieved through multicasting a MAC function operating on a secret key $K$ by every node. The protocol also implements different shared session secret keys for different links. As shown in Figure 3.2, node A has a link to node B and also has a link to node C. If node A wants to communicate with node B, it uses the shared session secret key $K_{AB}$. At the same time if it wants to communicate with node C, it makes use of another shared session secret key $K_{AC}$. Thus, security is enhanced by using separate link independent secret keys and also by refreshing the secret key for each session.
As shown in Figure 3.3, both the secret key and the shared session secret key were computed by the session secret key negotiation algorithm using ECC over binary fields.

The notations used in the protocol are explained as follows:

- $K$ is the secret key generated by the session secret key negotiation algorithm
- $K_{AB}$ is the shared session secret key between the node A and node B
- $ID_A$ is the identity of node A and similarly $ID_B$ is the identity of node B
- $N_A$ is the nonce generated by the node A and $N_B$ is the nonce generated by the node B
- Symbol ‘|’ is used for concatenation of two messages and
- $MAC_K(N_A|N_B)$ is the MAC function computed on the data $N_A|N_B$ with the secret key $K$. 

Figure 3.2 Multicast Link Independent Authentication
The proposed Mutual Authentication and Session Key Management protocol is explained as follows:

i) The shared secret key $K$ is computed between the nodes A and B using the secret key negotiation algorithm which implements ECC over binary fields in ECDH algorithm.

ii) Node A broadcasts the message $ID_A[N_A||MAC_K(ID_A[N_A])$ with the computed secret key $K$.

iii) Node B broadcasts the message $ID_B[N_B||MAC_K(ID_B[N_B])$ with the same secret key $K$. 

$K_{AB} = MAC_K(N_A|N_B)$ → Shared Session Secret Key between A and B
iv) Node A computes the MAC function from the broadcast message received from Node B with the secret key K and verifies the authenticity of node B.

v) In a similar fashion, node B computes the MAC function from the broadcast message received from node A with the secret key K and verifies the authenticity of node A. Thus mutual authentication among the nodes is achieved.

vi) For the nodes A and B with the secret key K and the exchanged nonces of $N_A$ and $N_B$, the shared session secret key is computed as $K_{AB} = MAC_K(N_A \mid N_B)$ and

vii) For the next session fresh nonces are exchanged and the shared session key is computed again. In this way the shared session key is refreshed for every session thereby providing greater security in the exchange of data.

3.6 ESTIMATION OF AUTHENTICATION

For secure routing in MANETs, estimation of the authenticated nodes is vital and this is done by using the Trust model. The Trust value obtained for each node is taken as its authentication value. Trust is defined as:

- Trust is a relationship established between two entities for a specific action. In particular, one entity trusts the other entity to perform an action. In this work, the first entity is called the subject; the second entity is called the agent. So, the notation used to describe a trust relationship is \{subject: agent; action\}

Let $T\{subject: agent; action\}$ denote the trust value of the trust relationship \{subject: agent; action\}, and $P\{subject: agent; action\}$ denote the probability that the agent would perform the action in the subject’s point of view. Information theory states that entropy is a natural measure for uncertainty. Thus, the entropy-based trust value is computed using equation (3.1)
\[ T \{\text{trust: agent, action}\} = \begin{cases} 
1 - H(p), & \text{for } 0.5 \leq p \leq 1 \\
H(p) - 1, & \text{for } 0 \leq p < 0.5 
\end{cases} \quad (3.1) \]

where,

\[ H(p) = -p \log_2 p - (1-p) \log_2 (1-p) \]

\[ p = P\{\text{subject: agent: action}\}. \]

In this research work, the trust value is considered to be a continuous real number in the range of \([-1, 1]\). The authenticated values are framed based on the trust values as follows:

- When \( p = 1 \), the subject trusts the agent the most and the trust value is 1
- When \( p = 0 \), the subject distrusts the agent the most and the trust value is \(-1\) and
- When \( p = 0.5 \), the subject has no idea about the agent and the trust value is 0.

In general, trust value is negative for \( 0 \leq p < 0.5 \) and positive for \( 0.5 \leq p \leq 1 \). Trust value is an increasing function with \( p \). It is a one-to-one mapping between \( T \{\text{subject: agent: action}\} \) and \( P \{\text{subject: agent: action}\} \).

Consider an example, that a node A wants to establish the trust relationship with node B as \( \{A: B; \text{act}\} \) based on the previous observation of node A about node B. To begin with, node A requests node B to forward \( N \) number of packets and node B in fact forwarded \( K \) number of packets.

Let \( V(i) \) be the performance action of node \( 'B' \) at the \( i^{th} \) trial, that is if \( V(i) = 1 \), the node B correctly performs the action at the \( i^{th} \) trial else \( V(i) = 0 \). Then, \( n(N) \) is given by equation (3.2):
\[ n(N) = \sum_{i=1}^{N} V_i \]  

(3.2)

where,

\( n(N) \) is the Number of actions successfully performed by 'X' out of totally 'N' trials.

For the 'N' trials of transmission between two nodes, 'k' trials are successful. The probability of success of \((N+1)\)th trial would be predicted by Bayesian theorem and is given by equation (3.3):

\[
P(v(N+1) = 1 | n(N) = k) = \frac{P(v(N+1) = 1, n(N) = k)}{P(n(N) = k)} \]  

(3.3)

The probability of all trials is calculated using Bernoulli's distribution and is given by equation (3.4):

\[
P\{n(N) = K\} = \sum_{k} p^k (1-p)^{N-K} \]  

(3.4)

where,

\( p \) is the average probability of success of each packet.

\( N \) is the total number of packets transmitted by source and

\( K \) is the number of packets transmitted successfully by neighbourhood node.

3.7 SECURE ROUTING MECHANISM

3.7.1 SECURE END-TO-END ROUTE DISCOVERY

To achieve end-to-end route discovery, each MN in the cluster maintains the trust value of its one-hop neighbours. Estimation of the trust value of a node and its neighbourhood node is associated with successful packet forwarding. The trust
relationship of the neighbourhood node is determined by the recommendation of the third party. That is, by observing the trust value of the third party for a particular packet transmission, the trust value of the neighbours can be predicted. Using this trust value, the security of the path would be inferred and established.

The routing problem has to be solved in real time since it is a dynamic optimization problem. The ant-based approach is compatible for the secure route discovery in MANETs. AntNet is an agent based routing algorithm, which is influenced by the unsophisticated and individual ant’s emergent behaviour. That is, MANETs are modeled as biological “networks” or colonies of ants consisting of thousands of ants. During the food gathering operation, each ant spreads pheromone which is ant’s possessive chemical fragrance and successive ants finds the optimum path to the location of the food.

As per the Engineering models and algorithms based on these biological systems, the AntNet routing algorithm considers each ant as an agent based packets like control packets. These packets are used to discover the route between the source and destination. Two types of agent based packets are used to find the end-to-end route by the AntNet algorithm called forward agent packets and backward agent packets. These packets are also considered as software agents. During route discovery process, these packets verify the traffic load, delay and trust value of each node that it visits and updates these parameters in its reserved location of memory.

To discover a secure route, the forward agent packets need to be sent to reach the destination. These forward agent packets, update the trust value of each node which it visits via its path to the destination. After finding the destination, the backward agent packets have to be sent to the source. During the backward flow, the agent packets once again verify the trust value of the discovered route. This verification is performed using the trust values updated by the forward agent packets. Finally, these parameters reach the source. The source would verify the trust value, process the parameters collected by the agent packets and selects the most efficient route.
3.7.2 Secure Route Establishment

The secure end-to-end route is established in the following sequences:

- Each source launches some forward agent packets to destination through multi-hop propagation. The path would be selected randomly based on the current routing table.

![Figure 3.4 Forward Route Path](image)

![Figure 3.5 Reverse Route Path](image)
• As shown in Figure 3.4, the forward agent packets create a stack, pushing in trip times, trust values and traffic intensities of every node it visits during the forward route path.

• As shown in Figure 3.5, when the packet reaches the destination, some backward agent packets would be sent to the source. During reverse route path, the backward agent packets inherit the stack parameters and verify once again the reverse route path and

• The backward agent packets deliver the parameters of trust values, traffic intensities and delays of the discovered routes to the source. Finally the optimum path to the destination is selected by the source.

3.8 RESULTS AND DISCUSSION

3.8.1 Simulation Parameters

The parameters considered for the simulation of the proposed protocol is presented in Table 3.1. The simulation was performed using Network Simulator version 2 (NS-2).

<table>
<thead>
<tr>
<th>Simulation Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of nodes including CH</td>
<td>40</td>
</tr>
<tr>
<td>Number of clusters</td>
<td>4</td>
</tr>
<tr>
<td>Nodes per cluster</td>
<td>10</td>
</tr>
<tr>
<td>Total network area</td>
<td>1 Square Km</td>
</tr>
<tr>
<td>Speed of the node</td>
<td>0 – 50 Metres per second</td>
</tr>
<tr>
<td>Packet size</td>
<td>250 to 500 Bytes</td>
</tr>
<tr>
<td>Data rate and type</td>
<td>100 Kbps and VBR</td>
</tr>
<tr>
<td>Elliptic Curve</td>
<td>Elliptic Curve over binary field</td>
</tr>
<tr>
<td></td>
<td>GF(2^m) for key size of 312 bits</td>
</tr>
<tr>
<td>Power required to transmit one packet</td>
<td>5 Milliwatts</td>
</tr>
</tbody>
</table>

Table 3.1 Simulation Parameters
3.8.2 Analysis of Successful Secure Links among Neighbours

The successful secure links among the neighbouring nodes were analyzed. Figure 3.6 shows the characteristic curves between the percentage of successful links and the number of neighbourhood nodes 'n' for different duration of time 't'.

As shown in Figure 3.6, each characteristic curve shows the percentage of successful links among the adjacent nodes, for every neighbourhood node 'n' changing within a time interval 't'. For example, from the Figure 3.6, the curve for t = 15 minutes shows the percentage of successful link among the adjacent nodes. The time interval t = 15 minutes indicates that due to node mobility, the analysis was done with the assumption that for every 15 minutes, one node among the neighbouring nodes changes, two nodes among the neighbouring nodes changes with the maximum number of node changes being n = 10.

![Figure 3.6 Analysis of Successful Secure Links among Neighbours](image)

The worst case condition is for the number of node changes being n = 10 nodes for t = 1 minute. That is, for every one minute ten neighbouring nodes of a particular central node are considered to change due to mobility. That is almost all the neighbouring nodes are in high speed movement and the percentage of successful link establishment is zero. From the analysis, it is inferred that the
The proposed protocol tolerates a worst case condition of \( n = 5 \) for \( t = 1 \) minute. In this case, the percentage of successful link establishment is 55% which is above 50%.

### 3.8.3 Power Overhead Ratio

The power overhead for the proposed protocol has been simulated and presented in Table 3.2.

**Table 3.2 Power Overhead for Worst Case Condition**

<table>
<thead>
<tr>
<th>Neighbourhood node ( n )</th>
<th>Power over head ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( t = 3 ) minutes</td>
</tr>
<tr>
<td>1</td>
<td>0.050</td>
</tr>
<tr>
<td>2</td>
<td>0.100</td>
</tr>
<tr>
<td>3</td>
<td>0.100</td>
</tr>
<tr>
<td>4</td>
<td>0.125</td>
</tr>
<tr>
<td>5</td>
<td>0.150</td>
</tr>
<tr>
<td>6</td>
<td>0.150</td>
</tr>
<tr>
<td>7</td>
<td>0.150</td>
</tr>
<tr>
<td>8</td>
<td>0.175</td>
</tr>
<tr>
<td>9</td>
<td>0.175</td>
</tr>
<tr>
<td>10</td>
<td>0.200</td>
</tr>
</tbody>
</table>
The Power Overhead Ratio was computed using equation (3.5):

\[
\text{Power Overhead Ratio} = \frac{P_p \times \rho_{kd}}{P_p \times \rho_{kd + \text{data}}} 
\]

where,

- \( P_p \) is the power requirement per packet in milliwatts.
- \( \rho_{kd} \) is the necessary packet rate for key distribution per second and
- \( \rho_{kd + \text{data}} \) is the packet rate including data and key distribution per second.

The performance of the proposed protocol is analyzed for the worst case parameters of power overhead with \( t = 1 \) minute, \( t = 2 \) minutes and \( t = 3 \) minutes for various values of ‘n’. From Table 3.2, it is observed that the worst case condition is for \( t = 1 \) minute simulated for various values of ‘n’. For all the cases, the power overhead is less than the acceptable optimum value of 50%. The worst case of maximum power overhead obtained with the proposed protocol is 45%. This condition is obtained at \( n = 10 \) for \( t = 1 \) minute which is still not above 50%. Thus compared to the existing protocols, the power overhead even for the worst case condition is less in the proposed protocol.

3.8.4 Stability Analysis of the Proposed Protocol in Secure Routing

Figure 3.7 illustrates the stability analysis of the proposed protocol in terms of packet forwarding in the prediction based clustering networks.

The performances have been analyzed with the percentage of malicious nodes in the cluster network. Malicious nodes are the number of dishonest nodes in the cluster. The unmanaged rate denotes the number of packets that cannot reach the destination within the desired time. These packets are stored in the honest nodes and are not dropped.
The failure rate denotes the percentage of packets dropped by the dishonest nodes and these packets are not available to any other nodes. These dropped packets are considered as loss of packets. From the analysis, it is inferred that the protocol tolerates up to 80% of malicious nodes in the cluster with successful rate of 60%. For percentage of malicious nodes above 80%, the failure rate is found to be less than 10%.

3.8.5 Authentication Estimation

Estimation of authentication for four different worst case scenarios is presented in Table 3.3. The trust values of neighbours for the four different worst case scenarios range from 0.1 to 0.4. Normally trust values range between -1 and +1.

The neighbourhood node is considered to be a malicious node if its trust value is negative. On the other hand, if it is positive the neighbourhood node is considered to be an honest node. Finally if it is zero, the node does not have any idea about the neighbourhood.
Table 3.3 Authentication Estimation

<table>
<thead>
<tr>
<th>Authentication value of third party</th>
<th>Neighbourhood Authentication value T = 0.1</th>
<th>Neighbourhood Authentication value T = 0.2</th>
<th>Neighbourhood Authentication value T = 0.3</th>
<th>Neighbourhood Authentication value T = 0.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>T = 0.3</td>
<td>Not Authenticated</td>
<td>Not Authenticated</td>
<td>Not Authenticated</td>
<td>Not Authenticated</td>
</tr>
<tr>
<td>T = 0.4</td>
<td>Not Authenticated</td>
<td>Not Authenticated</td>
<td>Not Authenticated</td>
<td>Not Authenticated</td>
</tr>
<tr>
<td>T = 0.5</td>
<td>Not Authenticated</td>
<td>Not Authenticated</td>
<td>Not Authenticated</td>
<td>Not Authenticated</td>
</tr>
<tr>
<td>T = 0.6</td>
<td>Not Authenticated</td>
<td>Not Authenticated</td>
<td>Authenticated</td>
<td>Authenticated</td>
</tr>
<tr>
<td>I = 0.7</td>
<td>Not Authenticated</td>
<td>Authenticated</td>
<td>Authenticated</td>
<td>Authenticated</td>
</tr>
<tr>
<td>T = 0.8</td>
<td>Authenticated</td>
<td>Authenticated</td>
<td>Authenticated</td>
<td>Authenticated</td>
</tr>
<tr>
<td>T = 0.9</td>
<td>Authenticated</td>
<td>Authenticated</td>
<td>Authenticated</td>
<td>Authenticated</td>
</tr>
<tr>
<td>I = 1.0</td>
<td>Authenticated</td>
<td>Authenticated</td>
<td>Authenticated</td>
<td>Authenticated</td>
</tr>
</tbody>
</table>

In case the trust values are positive and above 0.5 then the trust algorithm considers the node as a fully trusted node. If the trust value of a node is between 0.1 and 0.4 then the algorithm does not trust the node and instead the trust value of the third party has to be considered for authenticating a particular neighbourhood node. Table 3.3 lists the trustworthiness of the neighbourhood nodes with the help of third party.

3.8.6 Packet Rate Analysis with Speed

Figure 3.8 illustrates the performance of necessary end-to-end packet rate under mobility consideration. The analysis is performed by considering four nodes
for a particular central node. The speed of each node is considered to be from 0 to 40 metres per second.

![Graph showing packet rate versus speed](image)

**Figure 3.8 Necessary Packet Rate versus Speed**

### 3.8.7 Authentication Cost Ratio

Authentication is the additional process in the route discovery and is given by equation (3.9):

$$
\text{Authentication cost ratio} = \left( \frac{t_{\text{pro}}}{t_{\text{ecc}}} \right) \times 100 
$$

(3.9)

where,

- $t_{\text{pro}}$ is the processing time at every node during route discovery without authentication.
- $t_{\text{ecc}}$ is the processing time at every node during route discovery with ECC authentication.
Figure 3.9 Authentication Cost Ratio (intra domain)

Figure 3.10 Authentication Cost Ratio (inter domain)

Figure 3.9 and Figure 3.10 illustrates the performances of the authentication cost ratio both for intra domain and inter domain clustering model for MANETs.
The analysis was repeated with the assumption of four nodes considered for a particular central node. Each node was considered to move with a speed of 40 metres per second.

3.9 CONCLUSION

The scalability of MANETs was achieved by considering prediction based hierarchical network model on which the proposed Mutual Authentication and Session Key Management protocol was imposed. The protocol was analyzed in terms of percentage of successful link among neighbourhood nodes by considering the mobility condition for different scenarios with respect to speed. From the simulated results it is inferred that the proposed authentication protocol performs better in terms of power overhead for key management by using ECC and MAC. The power overhead for the worst case scenario was analyzed and improved performance was obtained when compared to the existing schemes.

Efficient end-to-end route discovery was also implemented by using AntNet routing algorithm with estimation of the neighbourhood authentication using the Trust model. Analysis shows that the protocol tolerates up to 80% of malicious nodes in the cluster with successful rate of 60%. Although the percentage of malicious node increases above 80%, the failure rate is very less in percentage which is below 10%. Other performances including trustworthiness of the neighbourhood nodes and the packet rate performances under mobility condition were also analyzed.