In this chapter probability analysis of RFID Sensor-integrated hierarchical MANET based attacks are performed. For the hierarchical MANET proposed in this work, authentication and distance bounding, ownership transfer and Ad-hoc network based attacks are analyzed for measuring the security levels. Authentication and distance bounding protocols based attacks include distance fraud, mafia fraud, terrorist fraud and distance hijacking attacks. Ownership transfer based attacks include secret disclosure and impersonation attacks. Ad-hoc network based attacks include eclipse, de-synchronization, bad mouthing, on-off, collusion, and whitewashing and traitor attacks. Probability analysis shows that the proposed system provides higher protection from distance bounding attacks as compared to existing systems because of randomness and flexibility to increase the size of challenge or verification bits. Simulation analysis shows that the proposed system detects the ownership and Ad-hoc network based attacks in a stipulated time at minimum cost.

5.1 Type of Nodes

For attack analysis, a node in a network is classified as legitimate member, illegitimate or dishonest and adversary node. These nodes are defined as:

- A node is a legitimate node if it performs the regular activities without revealing its secrets or involved in early reply strategies for getting any benefits.
- A legitimate node turns into an illegitimate or dishonest node if it reveals the secrets or involves in early reply strategies for getting the benefits.
- A node is adversary nodes if it is always involved in either early reply strategy or convincing the other nodes for getting the benefits.

A probability analysis of attacks identifies the chances of attacks over protocols. This work performs the probability analysis of distance related attacks over distance bounding protocol.

5.2 Distance Bounding Protocol based Attacks

In proposed hierarchical MANET, member nodes act as either legitimate members, illegitimate or an adversary node. Legitimate members are the authentic members of a
network with high trust score. Legitimate members have full rights to perform their commitments as expected in a fine grained access control mechanism. Sometimes, legitimate members start behaving badly by providing false information, disclosing the secrets, blocking data paths etc. This behavior provides small gains to attackers in terms of wrong feedback, eavesdrop the secrets, man-in-middle attacks etc. and turns a legitimate node into an illegitimate node. An adversary is a malicious node present in or out of a network to disrupt the network services.

Distance related attacks occur when false information is provided by legitimate members or adversaries. There are four types of distance related attacks possible in RFID networks: distance fraud attack, mafia fraud attack (relay attacks), terrorist fraud attack and distance hijacking attack. Probability analysis of attacks over proposed systems is explained as follows:

### 5.2.1 Distance Fraud Attack

In distance fraud attack, a legitimate member of a network moves out of the local security periphery of a subgroup but still acting as a legitimate member. For example, fig. 5.1 shows a local subgroup where legitimate nodes are part of a subgroup. These legitimate nodes have right to access the secrets of the subgroup or its members. Figure 5.2 shows a scenario where a legitimate member of a subgroup moves out of the subgroup but still having the key to reveal the secrets of its previous subgroup or subgroup members. Distance between two

![Figure 5.1: A group consisting of legitimate members to obtain the secret](image)

![Figure 5.2: A group consisting of legitimate member which behave badly to obtain the secret](image)

![Figure 5.3: Basic structure of timer based distance bounding method](image)
subgroup members can be limited through round-trip timer and signal strength based mechanisms. In this work, a round-trip timer based distance bounding mechanism is integrated with authentication protocol as shown in fig. 5.3. Timer starts when a challenge is generated from verifier and it stops when a response comes back from verifier. This process continues for n-rounds in fast phase of authentication and distance bounding protocol which is equivalent to number of bits in hash function output. Success of distance fraud is possible in following cases:

(a) **Random guessing when bit position is known**

In distance fraud attack, when a member node moves away from the subgroup then new hash and commitment values are generated. Since the left member of the subgroup has information about cryptographic mechanisms therefore it tries to guess the responses in challenge response mechanism. Now, attacker may or may not know the bit to guess. Let the output of hash function be 001100111100011 as shown in fig. 5.4. Here, probability of guessing each bit is \( \frac{1}{2} \). If an attacker knows the bit to guess in ‘n’ rounds then probability of success of attack is given by:

\[
P_{success\_attack} = \left( \frac{1}{2} \right)^n
\]  

(b) **Guessing the challenge in a known direction**

As shown in fig. 5.4, an attacker may not know the target bit to guess. If an attacker does not know the bit to guess then it tries to guess the challenge or response. In proposed authentication and distance bounding protocol, challenge is of two bits. An attacker may try to guess the challenge in one direction (MSB to LSB or LSB to MSB) or in both directions (MSB to LSB and LSB to MSB). Table 5.1 shows the probabilities of guessing the 2-bits challenge in one direction. The success probability of attack is given by equation 5.2.
Table 5.1: Probabilities in guessing the 2-bits challenge (in one direction)

<table>
<thead>
<tr>
<th>Most Significant Bit (MSB)</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Least Significant Bit (LSB)</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Probability</td>
<td>1/4</td>
<td>1/4</td>
<td>1/4</td>
<td>1/4</td>
</tr>
</tbody>
</table>

\[ P_{\text{success, attack}} = P[\text{Adversary correctly guesses the challenge}] = P[\text{Challenge} = 00 \cap \text{Adversary Guess} = 00] \cup P[\text{Challenge} = 11 \cap \text{Adversary Guess} = 11] \cup P[\text{Challenge} = 01 \cap \text{Adversary Guess} = 01] \cup P[\text{Challenge} = 10 \cap \text{Adversary Guess} = 10] = P[\text{Challenge} = 00 \cap \text{Adversary Guess} = 00] \times P[\text{Adversary Guess} = 00] + P[\text{Challenge} = 11 \cap \text{Adversary Guess} = 11] \times P[\text{Adversary Guess} = 11] + P[\text{Challenge} = 01 \cap \text{Adversary Guess} = 01] \times P[\text{Adversary Guess} = 01] + P[\text{Challenge} = 10 \cap \text{Adversary Guess} = 10] \times P[\text{Adversary Guess} = 10] = \left(\frac{1}{4} \times \frac{1}{4} + \frac{1}{4} \times \frac{1}{4} + \frac{1}{4} \times \frac{1}{4} + \frac{1}{4} \times \frac{1}{4}\right) = \frac{1}{4} \right) \]  (5.2)

Probability of success of attack for ‘n’ rounds of successful comparison is: \((1/4)^n\)

In another case, if the attacker has capability to guess the challenge in both directions (MSB to LSB and LSB to MSB) then table 5.2 shows the probabilities of guessing the 2-bits challenge in both directions. The success probability of attack is given by equation 5.3.

Table 5.2: Probabilities in guessing the two bits challenge (in both directions)

<table>
<thead>
<tr>
<th>Most Significant Bit (MSB)</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Least Significant Bit (LSB)</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Probability</td>
<td>1/4</td>
<td>1/2</td>
<td>1/2</td>
<td>1/4</td>
</tr>
</tbody>
</table>

\[ P_{\text{success, attack}} = P[\text{Adversary correctly guesses the challenge}] = P[\text{Challenge} = 00 \cap \text{Adversary Guess} = 00] \cup P[\text{Challenge} = 11 \cap \text{Adversary Guess} = 11] \cup P[\text{Challenge} = 01 \cap \text{Adversary Guess} = 01] \cup P[\text{Challenge} = 10 \cap \text{Adversary Guess} = 10] = \]  
\[ P[\text{Challenge} = 00 \cap \text{Adversary Guess} = 00] \times P[\text{Adversary Guess} = 00] + \]  
\[ P[\text{Challenge} = 11 \cap \text{Adversary Guess} = 11] \times P[\text{Adversary Guess} = 11] + \]  
\[ P[\text{Challenge} = 01 \cap \text{Adversary Guess} = 01] \times P[\text{Adversary Guess} = 01] + \]  
\[ P[\text{Challenge} = 10 \cap \text{Adversary Guess} = 10] + \]  

108
Hence, probability of success of attack for ‘n’ rounds of successful comparison is: \((3/8)^n\).

The probability of success of attack can further be reduced by increasing the number of verification bits. Thus, the fast phase of protocol 4.2 is modified in protocol 5.1.

\[
P[\text{Challenge} = 10 \text{ or } 01 / \text{Adversary Guess} = 10] \times P[\text{Adversary Guess} = 10] = \\
\left( \frac{1}{4} \times \frac{1}{4} + \frac{1}{4} \times \frac{1}{4} + \frac{1}{2} \times \frac{1}{4} + \frac{1}{2} \times \frac{1}{4} \right) = 3/8
\]  

Figure 5.5: Counter example of fast phase of authentication and distance bounding protocol for 3-bits challenge and response

Protocol 5.1: Authentication and Distance Bounding Protocol

Goal: Same as goal of protocol 4.2.

Premises: Same as premises of protocol 4.2.

Step 1 and step 2:- Same as step 1 and step 2 of protocol 4.2.

Step 3: \(SG_{SC_i}^{HL_i}\) and \(SG_{SC_i}^{HL_i}\) compute \(h(N1,N2)\) or \(SG_{SC_i}^{HL_i}\) and \(SM_{(U,K)}^{HL_i}\) compute \(h(N1,N2)\).
h(N1,N2) is divided into direction and verification bits such that (DIR1 (r-bits) ||
DIR2 (r-bits) || DIR3 (r-bits) || DIR4 (r-bits) || DIR5 (r-bits) || DIR6 (r-bits) ||
DIR7 (r-bits) || DIR8 (r-bits) || VER1 (b-bits) || VER2 (b-bits) || VER3 (b-bits) ||
VER4 (b-bits) || VER5 (b-bits) || VER6 (b-bits) || VER7 (b-bits) || VER8 (b-bits) ||
VER9 (b-bits)) = h(N1,N2) and 8r=9b.

Step 4:- Fast Phase: In this phase, bits are compared on both source and destination
sides in challenge and response rounds as:

(a) Challenge Round: Either \(SG_{SC_j}^{HL_i} \rightarrow SG_{SC_i}^{HL_i}\); Q or \(SG_{SC_j}^{HL_i} \rightarrow SM_{(t,k)}^{HL_i}\); Q

(b) Response Round: Either \(SG_{SC_i}^{HL_i} \rightarrow SG_{SC_j}^{HL_i}\); A or \(SM_{(t,k)}^{HL_i} \rightarrow SG_{SC_j}^{HL_i}\); A

Fig. 5.5 shows the counter example of 3-bits challenge and response. In this
example, a challenge for success of attack for an attacker will be of 3-bits. As
discussed in step 3, output of h(N1,N2) is divided into 8 direction (DIR) and 9
verification (VER) blocks. Data in direction blocks are for sending challenges
from source side towards destination side whereas in verification blocks are for
sending challenges from destination side towards source side. Let output of
h(N1,N2) = 110…1 || 100…0 || 001..1 || 101..1 || 000 .. 1 || 101 … 1 ||
101..0 || 111..1 || 011 .. 0 || 010…0 || 110 … 0 || 011..1 || 011..0 || 100 .. 0 ||
111…1 || 101 … 1. Initially source side will initiate the process of sending the
challenge as: Q=DIR1(1st bit)|| DIR2 (1st bit)=11||[0,1]*, i.e. a challenge is
generated by appending the 1st bit from DIR1 and DIR2. The result is further
appended with a random bit. Since, source has sent Q=11, thus it would be
expecting the A= 1st bit of VER7 || 1st bit of VER8 ||MSB of Q=111. Destination
will receive Q and if Q=00 ||[0,1]* then A= 1st bit of VER1 || 1st bit of VER2 ||
MSB of Q else if Q=01 ||[0,1]* then A= 1st bit of VER3 || 1st bit of VER4 ||
MSB of Q else if Q=10 ||[0,1]* then A= 1st bit of VER5 || 1st bit of VER6 ||
MSB of Q else if Q=11 ||[0,1]* then A= 1st bit of VER7 || 1st bit of VER8 ||
MSB of Q. Since Q=11, thus A=111. Source will receive A and if A=00 ||
{0,1}* then Q= 2nd bit of DIR1 || 2nd bit of DIR2 || MSB of A else if A=01 ||
{0,1}* then Q= 1st bit of DIR3 || 1st bit of DIR4 || MSB of A else if A=10 ||
{0,1}* then Q= 1st bit of DIR5 || 1st bit of DIR6 || MSB of A else if A=11 ||
{0,1}* then Q= 1st bit of DIR7 || 1st bit of DIR8 || MSB of A. After receiving A,
destination side will generate new challenge ‘Q’. This process goes on till all
bits are matched. VER9 is used when all bits of VER1 to VER8 are matched. Source side starts the timer 1 while sending Q and stops when A is received. Similarly, destination side starts the timer 2 while sending A and stops when Q is received. These timers ensure that nodes in a subgroup are in close vicinity. Finally, when all bits are matched then an ACK packet is sent from destination side for ending the authentication and distance bounding protocol.

**Step 5:** Same as step 5 of protocol 4.2.

<table>
<thead>
<tr>
<th>Most Significant Bit (MSB)</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st Position Bit</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Least Significant Bit (LSB)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 5.3 shows the probabilities of guessing the 3-bits challenge in one direction. Equation 5.4 gives the success probability of attack for 3-bits challenge. Results show that the success probability of attack is reduced by 50% for 3-bits challenge as compared to 2-bits challenge protocol.

\[
P_{\text{success,attack}} = P[\text{Adversary correctly guesses the challenge}] = P[\text{Challenge} = 000 \land \text{Adversary Guess} = 000] \cup P[\text{Challenge} = 111 \land \text{Adversary Guess} = 111] \cup P[\text{Challenge} = 001 \land \text{Adversary Guess} = 001] \cup P[\text{Challenge} = 010 \land \text{Adversary Guess} = 010] \cup P[\text{Challenge} = 011 \land \text{Adversary Guess} = 011] \cup P[\text{Challenge} = 100 \land \text{Adversary Guess} = 100] \cup P[\text{Challenge} = 101 \land \text{Adversary Guess} = 101] \cup P[\text{Challenge} = 110 \land \text{Adversary Guess} = 110]
\]

\[
= P[\text{Challenge} = 000/\text{Adversary Guess} = 000] \times P[\text{Adversary Guess} = 000] + P[\text{Challenge} = 111/\text{Adversary Guess} = 111] \times P[\text{Adversary Guess} = 111] + P[\text{Challenge} = 001/\text{Adversary Guess} = 001] \times P[\text{Adversary Guess} = 001] + P[\text{Challenge} = 010/\text{Adversary Guess} = 010] \times P[\text{Adversary Guess} = 010] + P[\text{Challenge} = 011/\text{Adversary Guess} = 011] \times P[\text{Adversary Guess} = 011] + P[\text{Challenge} = 100/\text{Adversary Guess} = 100] \times P[\text{Adversary Guess} = 100] + P[\text{Challenge} = 101/\text{Adversary Guess} = 101] \times P[\text{Adversary Guess} = 101] + P[\text{Challenge} = 110/\text{Adversary Guess} = 110] \times P[\text{Adversary Guess} = 110] = \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} = 1/8
\]

(5.4)
Hence, probability of success of attack for ‘n’ rounds of successful comparison is: $(1/8)^n$.

### Table 5.4: Probabilities in guessing the 3-bits challenge (in both directions)

<table>
<thead>
<tr>
<th>Most Significant Bit (MSB)</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1(^{st}) Position Bit</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Least Significant Bit (LSB)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Probability</td>
<td>1/8</td>
<td>1/4</td>
<td>1/4</td>
<td>1/4</td>
<td>1/8</td>
<td>1/4</td>
<td>1/8</td>
</tr>
</tbody>
</table>

If attacker has the capability of guessing the challenge in both directions (LSB to MSB and MSB to LSB) then table 5.4 gives the probabilities of guessing the 3-bits challenge and equation 5.5 gives the chances of success of an attacker. It is observed that with increase in challenge bits the success of attacker is gradually reducing.

$$P_{success\_attack} = \sum \left[ P[\text{Adversary correctly guesses the challenge}] = P[\text{Challenge} = 000 \cap \text{Adversary Guess} = 000] \cup P[\text{Challenge} = 111 \cap \text{Adversary Guess} = 111] \cup P[\text{Challenge} = 001 \cap \text{Adversary Guess} = 001] \cup P[\text{Challenge} = 010 \cap \text{Adversary Guess} = 010] \cup P[\text{Challenge} = 100 \cap \text{Adversary Guess} = 100] \cup P[\text{Challenge} = 101 \cap \text{Adversary Guess} = 101] \cup P[\text{Challenge} = 011 \cap \text{Adversary Guess} = 110] \right]$$

$$= P[\text{Challenge} = 000 / \text{Adversary Guess} = 000] \times P[\text{Adversary Guess} = 000] + P[\text{Challenge} = 111 / \text{Adversary Guess} = 111] \times P[\text{Adversary Guess} = 111] + P[\text{Challenge} = 010 / \text{Adversary Guess} = 010] \times P[\text{Adversary Guess} = 010] + P[\text{Challenge} = 101 / \text{Adversary Guess} = 101] \times P[\text{Adversary Guess} = 101] + P[\text{Challenge} = 001 or 100 / \text{Adversary Guess} = 001] \times P[\text{Adversary Guess} = 001] + P[\text{Challenge} = 011 or 110 / \text{Adversary Guess} = 011] \times P[\text{Adversary Guess} = 011] + P[\text{Challenge} = 100 or 001 / \text{Adversary Guess} = 100] \times P[\text{Adversary Guess} = 100] + P[\text{Challenge} = 110 or 011 / \text{Adversary Guess} = 110] \times P[\text{Adversary Guess} = 110]$$

$$= \left( \frac{1}{8} \times \frac{1}{8} + \frac{1}{8} \times \frac{1}{8} + \frac{1}{8} \times \frac{1}{8} + \frac{1}{8} \times \frac{1}{8} + \frac{1}{4} \times \frac{1}{8} + \frac{1}{4} \times \frac{1}{8} + \frac{1}{4} \times \frac{1}{8} + \frac{1}{4} \times \frac{1}{8} \right) = 3/16 \quad (5.5)$$

Hence, probability of success of attack for ‘n’ rounds of successful comparison is: $(3/16)^n$.

Table 5.5 and table 5.6 show the comparative analysis of chances of success of an attack with increase in number of challenge bits. Table 5.5 shows the comparative analysis for an attacker that has the capability of guessing the challenge bits in one direction (either LSB to MSB or MSB to LSB) only. Results show that with increase in every single challenge bit the
success probability of attack is reduced by 50%. Fig. 5.6 shows that the reduction of chances of success of attack is gradually reducing and if number of challenge bits is 8 then success of attack is almost zero. Similarly, table 5.6 shows the comparative analysis for an attacker that has the capability of guessing the challenge bit in both directions (LSB to MSB and MSB to LSB). Results show that with increase in every single challenge bit the success probability of attack is reducing. There is 50% reduction in chances of success of an attack with increase in every single challenge bit if number of challenge bits is odd. However, reduction of success of an attack is more than 50% if number of challenge bits is even. Fig. 5.7 shows the reduction of chances of success of attack with increase in challenge bits when an attacker has the capability of guessing the challenge in both directions. The reduction in success of chances of attack is lesser for an attacker that has the capability of guessing the challenge in both directions as compared to an attacker that has the capability of guessing the challenge in one direction.

Table 5.5: Comparisons of success probabilities in guessing the challenge (in one direction)

<table>
<thead>
<tr>
<th>Challenge bits</th>
<th>2-bits</th>
<th>3-bits</th>
<th>4-bits</th>
<th>…</th>
<th>N-bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_{success_attack}$</td>
<td>$(1/4)^n$</td>
<td>$(1/8)^n$</td>
<td>$(1/16)^n$</td>
<td>…</td>
<td>$(1/2^N)^n$</td>
</tr>
<tr>
<td>Percentage Improvement</td>
<td>50%</td>
<td>50%</td>
<td>50%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.6: Reduction in chances of success of an attack with increase in challenge bits when an attacker guesses in one direction

Table 5.6: Comparisons of success probabilities in guessing the challenge (in both directions)

<table>
<thead>
<tr>
<th>Challenge bits</th>
<th>2-bits</th>
<th>3-bits</th>
<th>4-bits</th>
<th>…</th>
<th>N-bits</th>
</tr>
</thead>
</table>
| $P_{success\_attack}$ | $(3/8)^n$ | $(3/16)^n$ | $(7/64)^n$ | … | $(2^{N/2} + 2^{N-2N/2})^n$ if $N \in EVEN$  
| | | | | | $(2^{N(N+1)/2} + 2^{N-2(N+1)/2})^n$ if $N \in ODD$ |
| Percentage Improvement | - | 50% | 58.3% |

*N= Number of challenge bits, n= number of fast phase rounds
(c) “Majority Vote” Attack

According to this attack, if two items are placed in two registers then either both boxes will have the same bit value or both will have different bit values. If an adversary has the capability of selecting the value which has majority then it can estimate whether both register will have the same value or different value. Table 5.7 shows all success probabilities for two registers with respect to “Majority Vote” attack. The success probabilities of 00 or 11 are 1 because attacker has the capability of selecting the value which has majority so it can predict whether 0 or 1 has the majority. If it is not able to predict which bit has the majority then it will assume that two registers will have 01 or 10 whose probability of each success is ½. Thus, overall success probability of an adversary is given by:

\[ P_{success,attack} = P[Majority\ Selected] = \frac{(1+\frac{1}{2}+\frac{1}{2}+1)}{4} = \frac{3}{4} \]  \hspace{1cm} (5.6)

For n-rounds in fast phase of authentication and distance bounding protocol, the success probability is \((3/4)^n\).

**Table 5.7:** Majority Vote attack for 2-bits challenge

<table>
<thead>
<tr>
<th>Most Significant Bit (MSB)</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Least Significant Bit (LSB)</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Success Probability</td>
<td>1</td>
<td>1/2</td>
<td>1/2</td>
<td>1</td>
</tr>
</tbody>
</table>

Now, if the challenge bit is increased from two to three bits then eight different combinations are possible as shown in table 5.8. With this attack, an adversary selects registers which have the majority of common bits, that is two or three register values are same. If adversary has the capability of predicting whether three registers have 0 or 1 value with a majority of three then success probability of 000 or 111 is 1. In all other cases, majority wins if two registers have
the same value, thus either both register will have 0 or 1. There are three cases when 0 has the 
majority, i.e. 001, 010 and 100. Similarly, there are three cases when 1 has the majority, i.e. 
011, 101 and 110. Thus, if an attacker successfully predicts the majority bit then success 
probability of each case is 1/3 as shown in table 5.8. In conclusion, the overall success 
probability of an adversary is given by:

\[ P_{\text{success, attack}} = P[\text{Majority Selected}] = \frac{(1+\frac{1}{2}+\frac{1}{2}+\frac{1}{2}+\frac{1}{2}+\frac{1}{2}+\frac{1}{2}+\frac{1}{2})}{8} = \frac{1}{2} \] 

(5.7)

For n-rounds in fast phase of authentication and distance bounding protocol, the success 
probability is \((1/2)^n\).

<table>
<thead>
<tr>
<th>Table 5.8: Majority Vote Attack for 3-bits challenge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Most Significant Bit (MSB)</td>
</tr>
<tr>
<td>1st Position Bit</td>
</tr>
<tr>
<td>Least Significant Bit (LSB)</td>
</tr>
<tr>
<td>Success Probability</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 5.9: Majority Vote Attack for 4-bits challenge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Most Significant Bit (MSB)</td>
</tr>
<tr>
<td>1st Position Bit</td>
</tr>
<tr>
<td>2nd Position Bit</td>
</tr>
<tr>
<td>Least Significant Bit (LSB)</td>
</tr>
<tr>
<td>Success Probability</td>
</tr>
</tbody>
</table>

Further, if challenge bits are increased from three to four bits then sixteen combinations of 
challenge bits are possible as shown in table 5.9. Like other cases, the success probability of 
0000 and 1111 is 1. Now, there are four cases each when 0 or 1 is in majority of three 
registers. The cases where 0 is in majority are: 0001, 0010, 0100 and 1000. Similarly, the 
cases where 1 is in majority are: 0111, 1011, 1101 and 1110. Thus, the success probability of 
each case is \(1/4\). The remaining cases have equal number of 0s and 1s. Hence, the success 
probability of selecting any one case from 0011, 0101, 0110, 1001, 1010 and 1100 is \(1/6\). The 
overall success probability of an adversary is given by:

\[ P_{\text{success, attack}} = P[\text{Majority Selected}] = \frac{(1+\frac{1}{2}+\frac{1}{2}+\frac{1}{2}+\frac{1}{2}+\frac{1}{2}+\frac{1}{2}+\frac{1}{2}+\frac{1}{2}+\frac{1}{2}+\frac{1}{2}+\frac{1}{2}+\frac{1}{2}+\frac{1}{2}+\frac{1}{2}+\frac{1}{2})}{16} = \frac{5}{16} \] 

(5.8)
For n-rounds in fast phase of authentication and distance bounding protocol, the success probability is \((5/16)^n\).

### Table 5.10: Comparisons of success probabilities in Majority Vote attack

<table>
<thead>
<tr>
<th>Challenge bits</th>
<th>2-bits</th>
<th>3-bits</th>
<th>4-bits</th>
<th>5-bits</th>
<th>……</th>
<th>N-bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>(P_{\text{success, attack}})</td>
<td>((3/4)^n)</td>
<td>((1/2)^n)</td>
<td>((5/16)^n)</td>
<td>((3/16)^n)</td>
<td>……</td>
<td>((N+1)/2^N)^n)</td>
</tr>
<tr>
<td>Percentage Improvement</td>
<td>…</td>
<td>66.7%</td>
<td>62.5%</td>
<td>60%</td>
<td>……</td>
<td>50% if (N \approx \infty)</td>
</tr>
</tbody>
</table>

*\(N=\) Number of challenge bits, \(n=\) number of fast phase rounds

Table 5.10 shows a comparative analysis of success probabilities in majority vote attack with increase in number of challenge bits. Results show that a maximum improvement of 66.7% for 3-bits challenge and a minimum improvement of 50% for N-bits challenge (\(N \approx \infty\)) in reduction of chances of attack are observed. Fig. 5.8 shows that the chances of attack gradually reduce with increase in challenge bits. The success probability of attack is almost zero when number of challenge bits is 10.

![Success Probabilities of Attack](image)

**Figure 5.8:** Success probabilities of majority vote attack with increase in number of challenge bits

### (d) Random guessing when bit position is not known

In this case, an attacker tries to draw ‘\(k\)’-sample bits from the set such that the order of drawing each sample is random and repetition is not allowed. Thus, for an unordered sampling without replacement, binomial theorem identifies the chances of success. According to binomial theorem:

\[
(1 + x)^n = C_0 + C_1x + C_2x^2 + C_3x^3 + \cdots \cdots \cdots \cdots + C_nx^n
\]  

(5.9)

Here, \(x\) is the probability of selecting the correct binary bit, thus \(x=1/2\). Now, Multiply eqn. 5.9 with \(x\).
\[ x(1 + x)^n = C_0x + C_1x^2 + C_2x^3 + C_3x^4 + \cdots + C_nx^{n+1} \quad (5.10) \]

where, \( p(k) = C_k \) is the success probability of selecting \( 'k' \)-correct bits out of \( 'n' \)-bits and put \( x=1/2 \) in eqn. 5.10.

\[
\left(\frac{1}{2}\right)\left(1 + \frac{1}{2}\right)^n = C_0\left(\frac{1}{2}\right)^0 + C_1\left(\frac{1}{2}\right)^2 + C_2\left(\frac{1}{2}\right)^3 + C_3\left(\frac{1}{2}\right)^4 + \cdots + C_n\left(\frac{1}{2}\right)^{n+1} = \frac{1}{2}\left(\frac{3}{2}\right)^n
\]

The success probability of selecting \( 'k' \)-correct bits is calculated as:

\[
P_{\text{success,attack}} = p(0)\left(\frac{1}{2}\right) + p(1)\left(\frac{1}{2}\right)^2 + p(2)\left(\frac{1}{2}\right)^3 + p(3)\left(\frac{1}{2}\right)^4 + \cdots + p(n)\left(\frac{1}{2}\right)^{n+1}
\]

For \( 'n' \)-bits string (or \( 'n' \)-rounds), the total possibilities (starting from 0 to \( n-1 \)) are \( 2^{n-1} \). Thus, the success probability is calculated as:

\[
P_{\text{success,attack}} = \left(\frac{1}{2}\right)^{n-1}\left(\frac{3}{2}\right)^n = \left(\frac{3}{4}\right)^n
\]

Since, there are two sides of authentication and distance bounding process, source and destination hence an attacker can guess the challenge bits on either side. \( P_{\text{success,attack}} \) is the probability of success of distance fraud attack on each side.

### 5.2.2 Relay or Mafia Fraud Attack

In relay or mafia fraud attack, attacker member nodes construct a network for providing false distances to legitimate nodes. For example, fig. 5.9 shows three subgroups. Two of these subgroups are attacker’s constructed subgroups for providing false information. Attacker’s subgroups consist of legitimate node, an adversary, and an antenna for reading the information from legitimate and adversary nodes. Here, adversary is authentic member of legitimate subgroup. This adversary and antenna try to mislead the legitimate node for a closer distance with legitimate network. Goal of attacker is to provide false distance in order to reveal the secrets. This is the most dangerous form of distance attacks. This is one form of man-in-the-middle attack.
In this attack, legitimate members of a subgroup are source subgroup controller ($SG_{Sc_d}^{HL_a}$), destination subgroup controller ($SG_{Sc_d}^{HL_a}$) and a member node ($M_{(a,b)}^{(c,d)}$), and adversaries are malicious subgroup controller nearer to legitimate source subgroup controller ($MSG_{Sc_d}^{HL_a}$), malicious subgroup controller nearer to legitimate destination subgroup controller ($MSG_{Sc_e}^{HL_a}$), malicious member node nearer to legitimate source subgroup controller ($MM_{(a,b)}^{(c,d)}$) or malicious member node nearer to legitimate destination subgroup controllers ($MM_{(c,e)}^{(a,b)}$). These malicious entities communicate with original subgroup controller and members and convince them to reveal secret information [249]-[251]. Here, $MSG_{Sc_d}^{HL_a}$, $MSG_{Sc_e}^{HL_a}$ and $MM_{(a,b)}^{(c,d)}$ start Man-in-Middle (MiM) attack by sending nonce value to legitimate source and destination. Figure 5.10 to 5.13 show the possibilities when malicious entities play man-in-the-middle with legitimate members. Malicious subgroup controller ($MSG_{Sc_d}^{HL_a}$ or $MSG_{Sc_e}^{HL_a}$) start MiM attack with legitimate subgroup controller ($SG_{Sc_d}^{HL_a}$ or $SG_{Sc_e}^{HL_a}$) in slow phase of authentication and distance bounding protocol (protocol 4.2) as:

**Figure 5.10:** $MSG_{Sc_d}^{HL_a}$ plays MiM

**Figure 5.11:** $MSG_{Sc_d}^{HL_a}$ plays MiM with the help of $MM_{(a,b)}^{(c,d)}$ nearer to $SG_{Sc_d}^{HL_a}$

**Figure 5.12:** $MSG_{Sc_d}^{HL_a}$ plays MiM with the help of $MM_{(c,e)}^{(a,b)}$ nearer to $SG_{Sc_e}^{HL_a}$

**Figure 5.13:** $MSG_{Sc_d}^{HL_a}$ plays MiM with the help of $MM_{(c,d)}^{(a,b)}$ nearer to both $SG_{Sc_d}^{HL_a}$ and $SG_{Sc_e}^{HL_a}$
Nonce values ($N_{MSG_{SC_d}^{HL_a}}$ or $N_{MSG_{SC_e}^{HL_a}}$) are exchanged with legitimate members through adversaries in different combinations as shown in fig. 5.10 to fig. 5.13. Fig. 5.10 shows a case when $MSG_{SC_d}^{HL_a}$ plays MiM with $SG_{SC_d}^{HL_a}$ or $SG_{SC_e}^{HL_a}$ directly. Here, goal of the attackers is to obtain the nonce values of maximum of nodes such that it can start the eavesdropping of secret information by being an authentic member of a subgroup. In this example, $MSG_{SC_d}^{HL_a}$ is one adversary who is playing MiM attack. There are chances of MiM attack by more than one adversary. Fig. 5.11 to 5.13 show the possibilities of MiM attack between two subgroup controllers ($SG_{SC_d}^{HL_a}$ or $SG_{SC_e}^{HL_a}$) by more than one adversaries. Fig. 5.11 shows an example where $MSG_{SC_d}^{HL_a}$ plays MiM with the help of $MM_{(c,d)}^{(a,b)}$ nearer to $SG_{SC_d}^{HL_a}$. In this example, $MSG_{SC_e}^{HL_a}$ initiates the MiM attack by sending $N_{MSG_{SC_d}^{HL_a}}$ and $N_{MSG_{SC_e}^{HL_a}}$ to $MM_{(c,d)}^{(a,b)}$ and $SG_{SC_e}^{HL_a}$ respectively. $MM_{(c,d)}^{(a,b)}$ forwards $N_{MSG_{SC_d}^{HL_a}}$ to $SG_{SC_d}^{HL_a}$ and receives $N_{SG_{SC_d}^{HL_a}}$ from $SG_{SC_e}^{HL_a}$. $MM_{(c,d)}^{(a,b)}$ forwards $N_{SG_{SC_d}^{HL_a}}$ to $MSG_{SC_e}^{HL_a}$. Finally, $MSG_{SC_e}^{HL_a}$ receives $N_{SG_{SC_d}^{HL_a}}$ and $N_{SG_{SC_e}^{HL_a}}$ which are used to generate $h(N_{SG_{SC_d}^{HL_a}}, N_{SG_{SC_e}^{HL_a}})$. Output of $h(N_{SG_{SC_d}^{HL_a}}, N_{SG_{SC_e}^{HL_a}})$ is used in fast phase of authentication and distance bounding protocol for mutually authenticating each other where an adversary $MSG_{SC_e}^{HL_a}$ get authenticated. Similarly, fig. 5.12 shows the MiM attack with the help of $MSG_{SC_d}^{HL_a}$ and $MM_{(c,e)}^{(a,b)}$. In this case, $MSG_{SC_d}^{HL_a}$ receives $N_{SG_{SC_d}^{HL_a}}$ and $N_{SG_{SC_e}^{HL_a}}$. $MSG_{SC_d}^{HL_a}$ uses output of $h(N_{SG_{SC_d}^{HL_a}}, N_{SG_{SC_e}^{HL_a}})$ in fast phase of authentication and distance bounding protocol for mutually authenticating each other. Here, $MSG_{SC_d}^{HL_a}$ gets authenticated and obtains the secrets. Further, fig. 5.13 shows an example of MiM attack by $MSG_{SC_d}^{HL_a}$, $MSG_{SC_e}^{HL_a}$, $MM_{(c,d)}^{(a,b)}$ and $MM_{(c,e)}^{(a,b)}$. This is an example of false authentication and distance bounding when source and destinations are at maximum distance apart as compare to previous cases. Here, $MSG_{SC_d}^{HL_a}$ or $MSG_{SC_e}^{HL_a}$ initiates the MiM attack with the help of $MM_{(c,d)}^{(a,b)}$ and $MM_{(c,e)}^{(a,b)}$. Finally, $MSG_{SC_d}^{HL_a}$ or $MSG_{SC_e}^{HL_a}$ receives $N_{SG_{SC_d}^{HL_a}}$ and $N_{SG_{SC_e}^{HL_a}}$, and uses output of $h(N_{SG_{SC_d}^{HL_a}}, N_{SG_{SC_e}^{HL_a}})$ in
fast phase of authentication and distance bounding protocol for mutually authenticating each other. Here, $MSG^{H_{la}}_{Sc_d}$ or $MSG^{H_{la}}_{Sc_e}$ gets authenticated and obtains the secrets.

Let fault acceptance rate (FAR) be the chances of acceptance of network nodes in presence of attacker nodes. Fig. 5.14 shows the scenario of two subgroup controllers interacting among themselves for authentication where attack is not detected until $i^{th}$ round of either side. Suppose it is detected in $(i+1)^{th}$ round. Hence, FAR of two subgroup controllers is calculated as:

$$P[FAR] = P[UAND^i_{SG_{Sc_e}} \cap AD^{i+1}_{SG_{Sc_d}}] \cup P[UAND^i_{SG_{Sc_e}} \cap AD^{i+1}_{SG_{Sc_d}}]$$

$$= (\prod_{j=i}^{n} P[UAND^i_{SG_{Sc_e}} \cap AD^{i+1}_{SG_{Sc_d}}]) \cdot P[AD^{i+1}_{SG_{Sc_d}}]$$

where,

- $UAND^i_{SG_{Sc_e}}$: attack not detected by $e^{th}$ subgroup controller until $i^{th}$ round
- $AD^{i+1}_{SG_{Sc_d}}$: attack detected by $d^{th}$ subgroup controller in $(i+1)^{th}$ round

$$P[UAND^i_{SG_{Sc_e}} \cap AD^{i+1}_{SG_{Sc_d}}] = P[AD^{i+1}_{SG_{Sc_d}}]$$

Figure 5.14: MiM attack detection in authentication and distance bounding protocol

In proposed authentication and distance bounding protocol, trust score is also checked. Nodes in the network are considered to be authentic if trust score of nodes is satisfactory. Thus, the conditional probability that attack is not detected until $i^{th}$ rounds at $SG^{H_{la}}_{Sc_d}$ side and detected in $(i+1)^{th}$ round at $SG^{H_{la}}_{Sc_e}$ side can be calculated as:

$$P[UAND^i_{SG_{Sc_e}} \cap AD^{i+1}_{SG_{Sc_d}}] \cdot P[AD^{i+1}_{SG_{Sc_d}}] =$$
\[
\prod_{i=1}^{i-1} P \left[ \text{UAND}^i_{S_G^{HL_a}} \left| \text{AND}^i_{S_G^{HL_a}} \right| \text{trust score of } S_G^{HL_a} \text{ is satisfactory} \right] 
\times P \left[ \text{AD}^{i+1}_{S_G^{HL_a}} \left| \text{trust score of } S_G^{HL_a} \text{ is satisfactory} \right] \right]
\]

Where,

\( \text{AND}^i_{S_G^{HL_a}} \) : attack not detected by \( d^\text{th} \) subgroup controller in \( i^\text{th} \) round

Similarly, the conditional probability that attack is not detected until \( i^\text{th} \) round at \( S_G^{HL_a} \) side and detected in \((i+1)^{th}\) round at \( S_G^{HL_a} \) side can be calculated as:

\[
P \left[ \text{UAND}^i_{S_G^{HL_a}} \left| \text{AD}^{i+1}_{S_G^{HL_a}} \right] \right] \times P \left[ \text{AD}^{i+1}_{S_G^{HL_a}} \right] = 
\prod_{i=1}^{i-1} P \left[ \text{UAND}^i_{S_G^{HL_a}} \left| \text{AND}^i_{S_G^{HL_a}} \right| \text{trust score of } S_G^{HL_a} \text{ is satisfactory} \right] 
\times P \left[ \text{AD}^{i+1}_{S_G^{HL_a}} \right]
\]

In the proposed authentication and distance bounding protocol, trust score is considered to be satisfactory if it is more than a threshold value. As shown in table 5.11, threshold trust score value is calculated using Alloy. As discussed in section 4.4, intruders are inserted into subgroups with variations in trust scores. Here, time is observed for detecting the intruders. In this case, time required for intruder detection gets almost double when trust score varies from 75% to 89% as compared to 90% to 100%, thus a lower limit, i.e. 75% is considered as a threshold trust score value. Let \( p_{\text{threshold}} \) be the probability that trust scores of \( S_G^{HL_a} \) and \( S_G^{HL_a} \) are satisfactory and \( p_{\text{protected}} \) is the probability of moving to protected mode. Protected mode is a state of protocol when source and destination nodes decided to stop the authentication process after detecting the attack. In result, the conditional probability that attack is not detected until and at \( i^\text{th} \) rounds on \( S_G^{HL_a} \) or \( S_G^{HL_a} \) side can be calculated as:

\[
\prod_{i=1}^{i-1} P \left[ \text{UAND}^i_{S_G^{HL_a}} \left| \text{AND}^i_{S_G^{HL_a}} \right| \text{trust score of } S_G^{HL_a} \text{ is satisfactory} \right] 
= \prod_{i=1}^{i-1} P \left[ \text{UAND}^i_{S_G^{HL_a}} \left| \text{AND}^i_{S_G^{HL_a}} \right| \text{trust score of } S_G^{HL_a} \text{ is satisfactory} \right] 
= (1 - p_{\text{protected}})^{i-1} + (p_{\text{threshold}})^{i-1}
\]
Further, there are two types of bits in proposed authentication and distance bounding protocol, direction bits (DIRs) and verification bits (VERs). The process of authentication and distance bounding continues until the direction bits are checked at destination side and verification bits are checked at source side. If output of hash functions is of n-bits then probability that one bit of direction or verification is matched at destination or source side respectively equals 1/n.

Conversely, \( P[\text{Direction or verification bit is not verified}] = (1 - \frac{1}{n}) \) \hspace{1cm} (5.16)

If an attacker has tried ‘k’ out of ‘n’ bits and probability that none of them matches is:

\[
P[\text{no match}] = \left(1 - \frac{1}{n}\right)^k
\]

\hspace{1cm} (5.17)

\[
P[\text{at least one match}] = 1 - \left(1 - \frac{1}{n}\right)^k
\]

\hspace{1cm} (5.18)

Equation 5.18 gives the probability that attacker illegitimate started the authentication and distance bounding protocol with at least one bit match but it results in un-authentication because bit does not matches thereafter. Hence,

\[
P \left[ \text{AD}_{SG_{HC}} \left| \text{trust score of SG}_{SCe} \right. \right. \text{is satisfactory}\right] = \left(1 - \frac{1}{n}\right)^k
\]

\hspace{1cm} (5.19)

Put equation 5.19 and 5.15 in 5.13 and 5.14.

\[
P[\text{UAND}_{SG_{HC}} \left| \text{AD}_{SG_{HC}} \right. ] \times P[\text{AD}_{SG_{HC}}] = P \left[ \text{UAND}_{SG_{HC}} \left| \text{AD}_{SG_{HC}} \right. ] \times \text{Pr} \left[ \text{AD}_{SG_{HC}}\right]
\]

\[
= (1 - p_{\text{protected}})^{i-1} + (P_{\text{threshold}})^{i-1} (1 - \left(1 - \frac{1}{n}\right)^k)
\]

\hspace{1cm} (5.20)

By putting equation 5.20 in 5.12:

---

### Table 5.11: Analysis of threshold trust score

<table>
<thead>
<tr>
<th>Percentage of Trust Score</th>
<th>Intruder Assertion</th>
<th>Proposed Trust Mechanism</th>
</tr>
</thead>
<tbody>
<tr>
<td>%age of Score ≥ 90</td>
<td>1/5/10</td>
<td>20/21/26 (120/226/351)</td>
</tr>
<tr>
<td>90 &gt; %age of Score ≥ 75</td>
<td>1/5/10</td>
<td>35/42/61 (222/350/595)</td>
</tr>
<tr>
<td>75 &gt; %age of Score ≥ 60</td>
<td>1/5/10</td>
<td>41/61/74 (332/530/650)</td>
</tr>
<tr>
<td>60 &gt; %age of Score ≥ 45</td>
<td>1/5/10</td>
<td>52/74/85 (436/626/751)</td>
</tr>
<tr>
<td>% age of Score &lt; 45</td>
<td>1/5/10</td>
<td>62/84/95 (546/726/881)</td>
</tr>
</tbody>
</table>
\[ P[\text{FAR}] = 2 \cdot ((1 - p_{\text{protected}})^{i-1} + p_{\text{threshold}}^{i-1})(1 - (1 - \frac{1}{n})^k) \]  

(5.21)

If authentication and distance bounding protocol do not move to protected mode and trust score of both source and destination is greater than threshold for 96 to 112 bits identification number then plot of equation 5.21 is shown in fig. 5.15. Here, \( k = i \). Plot of equation 5.21 is equivalent to plot of \( 1 - e^{-\frac{i}{96}} \) as shown in fig. 5.15. Here, ‘i’ varies from 2 to 112 bits. Hence

\[ P[\text{FAR}] = 2 \cdot ((1 - p_{\text{protected}})^{i-1} + p_{\text{threshold}}^{i-1})(1 - (1 - \frac{1}{n})^i) \approx (1 - e^{-\frac{i}{96}}) \]  

(5.22)

Figure 5.15: Graph comparisons of \( 2 \cdot ((1 - p_{\text{protected}})^{i-1} + p_{\text{threshold}}^{i-1})(1 - (1 - \frac{1}{n})^i) \) and \( 1 - e^{-\frac{i}{96}} \)

Probability that there are at least 50% chances of fault acceptance can be written as:

\[ P[\text{FAR}] \geq 0.5 \]

\[ P[\text{FAR}] = \left(1 - e^{-\frac{i}{96}}\right) \geq 0.5 \]

\[ e^{-\frac{i}{96}} \leq 0.5 \]

\[ \frac{i}{96} \geq \log_e 2 \]

\[ i \geq 96 \cdot \log_e 2 \]

\[ i \geq 29 \]

According to EPC Global Gen-3, an identifier varies from 96 bits to 112 bits. There is a requirement of a minimum of 29 bits out of 96 bits to be matched for getting 50% chances of fault acceptance when output of hash function in slow phase is of 96 bits. Similarly, there is a requirement of a minimum of 34 bits out of 112 bits to be matched for getting 50% chances of fault acceptance when output of hash function in slow phase is of 112 bits. For matching
maximum bits, a source or destination does not collude with an adversary in this attack but an adversary may launch this attack through anyone of the following ways:

(a) **Random guessing when bit position is known**

In this case, an adversary has to collude with either source or destination for getting the initial challenge bits in fast phase of authentication and distance bounding protocol. Once, an attacker is able to get the challenge bits then either an attacker has to guess every subsequent challenge bit or eavesdrops all challenge bits for answering the last challenge. In case, an attacker is eavesdropping the challenge then it will not be able to run an authentication process with itself and overall, it will be a passive attack only. In an active attack, it has to guess every subsequent challenge bit before it is responded by a legitimate node. This strategy has the same probability as in the random guessing phase of distance fraud attack. Since, probability of guessing each binary bit is \( \frac{1}{2} \), thus if an attacker has to guess the challenge bit for \( n \)-rounds then probability of success of attack is given by:

\[
P_{\text{success, attack}} = \left( \frac{1}{2} \right)^n
\]  

(5.23)

(b) **Guessing the challenge in a known direction for all possible challenge bit combinations**

In this case, an attacker has to perform MiM attack before source and destination start sending the direction or verification bits in authentication and distance bounding protocol. Thus, an attacker initiates the process of sending the correct challenge bits towards destination side before source side forwards the challenge bits. If an attacker guesses the challenge from all possible combinations of challenge bits then:

\[
P_{\text{success, attack}} = P[\text{Adversary randomly and correctly guesses the challenge bits}] \cup
\]
\[
(P[\text{Adversary randomly and incorrectly guesses the challenge bits}] \cap
\]
\[
P[\text{Adversary randomly and correctly responds to the challenge}] =
\]
\[
P[\text{Adversary randomly and correctly guesses the challenge bits}] +
\]
\[
(1 - P[\text{Adversary randomly and correctly guesses the challenge bits}]) \ast
\]
\[
P[\text{Adversary randomly and correctly responds to the challenge}]
\]

Now, if the response to challenge, i.e. verification bit is of size one then:

\[
P_{\text{success, attack}} = P[\text{Adversary randomly and correctly guesses the challenge bits}] +
\]
\[
(1 - P[\text{Adversary randomly and correctly guesses the challenge bits}]) \ast \frac{1}{2}
\]
\[
= (1 + P[\text{Adversary randomly and correctly guesses the challenge bits}]) \ast \frac{1}{2}
\]  

(5.24)
Table 5.12: Comparisons of success probabilities of guessing the challenge for all possible combinations of challenge bits (in one direction)

<table>
<thead>
<tr>
<th>Challenge bits</th>
<th>2-bits</th>
<th>3-bits</th>
<th>4-bits</th>
<th>…</th>
<th>N-bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_{\text{success attack}}$</td>
<td>$\frac{(5/8)^n}{n}$</td>
<td>$\frac{(9/16)^n}{n}$</td>
<td>$\frac{(17/32)^n}{n}$</td>
<td>…</td>
<td>$\frac{((1+2^n)/2^{N+1})^n}{n}$</td>
</tr>
<tr>
<td>Percentage Improvement</td>
<td>-</td>
<td>90%</td>
<td>94.4%</td>
<td>100%</td>
<td></td>
</tr>
</tbody>
</table>

*N= Number of challenge bits, n= number of fast phase rounds

Figure 5.16: Reduction in chances of success of mafia fraud attack with increase in challenge bits when an attacker guesses in one direction

Table 5.12 and fig. 5.16 show the comparative analysis of probabilities of guessing the challenge for all possible combinations of challenge bits. Here, challenge bits vary from 2 to N and guessing is performed either from LSB to MSB or MSB to LSB. Results show that probability of success of guessing the challenge decreases with increase in challenge bits. A minimum improvement of 90% for 3-bits challenge and a maximum improvement of 100% for N-bits challenge is observed, where $N \approx \infty$. Further, table 5.13 and fig. 5.17 show the comparative analysis of success probabilities when attacker is added with capability of guessing the challenge bits in both directions, i.e. LSB to MSB and MSB to LSB. Results show that the success probability of an attacker is reducing with increase in number of challenge bits. A minimum improvement of 86.4% for 3 bits challenge and a maximum of 100% for N-bits challenge is observed, where $N \approx \infty$. It is also observed that if an attacker is having a capability of guessing the bits in one or both directions then a minimum success probability of attack is $\frac{1}{2}$. However, in order to get overall success probability of $\frac{1}{2}$, an attacker has to guess a minimum of 29 challenges.

Table 5.13: Comparisons of success probabilities in guessing the challenge for all possible challenge bits combinations (in both directions)

<table>
<thead>
<tr>
<th>Challenge bits</th>
<th>2-bits</th>
<th>3-bits</th>
<th>4-bits</th>
<th>…</th>
<th>N-bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_{\text{success attack}}$</td>
<td>$\frac{(11/16)^n}{n}$</td>
<td>$\frac{(19/32)^n}{n}$</td>
<td>$\frac{(71/128)^n}{n}$</td>
<td>…</td>
<td>$\begin{cases} \frac{1}{2} + \frac{2^{N/2}}{2^{N+1} \times 2^N} + \frac{2^{N-2^{N/2}}}{2^{N \times 2^N}} \text{ if } N \in EVEN \ \frac{1}{2} + \frac{2^{(N+1)/2}}{2^{N+1} \times 2^N} + \frac{2^{N-2^{(N+1)/2}}}{2^{N \times 2^N}} \text{ if } N \in ODD \end{cases}$</td>
</tr>
</tbody>
</table>

125
<table>
<thead>
<tr>
<th>Percentage Improvement</th>
<th>-</th>
<th>86.4%</th>
<th>93.4%</th>
<th>100%</th>
</tr>
</thead>
</table>

*N= Number of challenge bits, n= number of fast phase rounds

![Graph showing reduction in chances of success of mafia fraud attack with increase in challenge bits.]

**Figure 5.17**: Reduction in chances of success of mafia fraud attack with increase in challenge bits when an attacker guesses in both directions

(c) **Guessing the challenge in a known direction for specific challenge bits combinations**

Like previous cases, an attacker performs MiM attack before source and destination start sending the direction or verification bits in authentication and distance bounding protocol. Instead of selecting and sending the challenge bits from all combinations of challenge bits, an attacker selects from the specific challenge bits. For example: if a challenge is of 2-bits then possible combinations of 2-bits are: 00, 01, 10 and 11. Instead of selecting the challenge from all these four combinations, an attacker selects a challenge from either \{00, 11\} or \{01, 10\}.

\[ P_{\text{success, attack}} = P[\text{Adversary correctly guesses the challenge}] = P[\text{Adversary Guess} = 01 \text{ or } 10] \cap P[\text{Challenge} = 01 \text{ or } 10] \cup P[\text{Adversary Guess} = 00 \text{ or } 11] \cap P[\text{Challenge} = 00 \text{ or } 11] \]

\[ = P[\text{Adversary Guess} = 01 \text{ or } 10/\text{Challenge} = 01 \text{ or } 10] \times P[\text{Challenge} = 01 \text{ or } 10] + P[\text{Adversary Guess} = 00 \text{ or } 11/\text{Challenge} = 00 \text{ or } 11] \times P[\text{Challenge} = 00 \text{ or } 11] \]  

(5.25)

Assume that an adversary has the capability of guessing in one direction, i.e. LSB to MSB or MSB to LSB. There are two sets (\{00, 11\} and \{01, 10\}) and probability of selecting the correct set is \(\frac{1}{2}\). Thus, equation 5.25 become

\[ P_{\text{success, attack}} = \frac{1}{2} \left( \frac{1}{2} + \frac{1}{2} \right) = \frac{1}{4} \]  

(5.26)

If an attacker has to guess the challenge bit for ‘n’-rounds of fast phase in authentication and distance bounding protocol then overall probability of success of attack is: \(\left(\frac{1}{4}\right)^n\).
Further, if an adversary has the capability of guessing in both directions, i.e. LSB to MSB and MSB to LSB then equation 5.25 becomes

$$P_{\text{success\_attack}} = \frac{1}{2}(1 + \frac{1}{4} + \frac{1}{4}) = \frac{3}{8}$$

(5.27)

If an attacker has to guess the challenge bit for ‘n’-rounds of fast phase in authentication and distance bounding protocol then overall probability of success of attack is: $$\left(\frac{3}{8}\right)^n$$.

In this case of attack, an adversary tries to make sets of those combinations which are palindrome in order to increase the probability of success of an attack. In this case, an adversary has an additional capability of observing the patterns of challenge bit combinations in authentication and distance bounding protocol. After observing the patterns, an adversary tries to find those patterns which are most common. For example, the sets formed by adversary after observing pattern in two bit challenge are {01, 10} and {00,11}. In this example there is one set only which is palindrome, i.e. {01, 10}. Here, adversary has to make maximum sets which contain palindrome bits in order to increase the probability of success of an attack.

Table 5.14 shows the comparative analysis of probabilities in guessing the challenge bits for specific challenge bits combinations in one (LSB to MSB or MSB to LSB) as well as in both directions (LSB to MSB and MSB to LSB) for 2 to 5 challenge bits. This comparative analysis is also performed for three cases: (i) without capability of an adversary in palindrome set formation, (ii) with capability of an adversary in palindrome set formation and without considering the palindrome sets selection probability and (iii) with capability of an adversary in palindrome set formation and considering the palindrome sets selection probability. Figure 5.18 shows the analysis of six cases discussed in table 5.14. Results show that case 2 and case 5 are having either no or least reduction in success probabilities of an attack with increase in challenge bits. All other cases show reduction in success probabilities of an attack with increase in challenge bits. Inference drawn from this is that if adversary has the capability of observing the pattern and making the palindrome sets of challenge and verification bits in authentication and distance bounding protocol then chances of success of an attack show no or little reduction with increase in challenge bits as compared to other cases. Further, if an adversary does not have the capability of observing the pattern and making the palindrome sets or have to select the correct palindrome set after its formation then it is analyzed that there is reduction in success probability of attack in every case. With increase in challenge bits, success probabilities of attacks in every case are almost same.
Table 5.14: Comparisons of success probabilities in guessing the challenge for specific challenge bits combinations

<table>
<thead>
<tr>
<th>Case</th>
<th>Challenge bits</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Guessing in one Direction (LSB to MSB or MSB to LSB)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>$P_{\text{success,attack}}$ (Without palindrome sets)</td>
<td>(1/2)$^a$</td>
<td>(1/8)$^b$</td>
<td>(1/16)$^b$</td>
<td>(1/32)$^b$</td>
</tr>
<tr>
<td>2</td>
<td>$P_{\text{success,attack}}$ (Without sets selection probability)</td>
<td>(1/2)$^a$</td>
<td>(1/2)$^b$</td>
<td>(7/16)$^b$</td>
<td>(1/2)$^b$</td>
</tr>
<tr>
<td>3</td>
<td>$P_{\text{success,attack}}$ (With sets selection probability)</td>
<td>(1/4)$^a$</td>
<td>(1/8)$^b$</td>
<td>(1/16)$^b$</td>
<td>(1/26)$^b$</td>
</tr>
<tr>
<td></td>
<td>Guessing in both Directions (LSB to MSB and MSB to LSB)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>$P_{\text{success,attack}}$ (Without palindrome sets)</td>
<td>(3/8)$^a$</td>
<td>(3/16)$^b$</td>
<td>(7/64)$^b$</td>
<td>(7/128)$^b$</td>
</tr>
<tr>
<td>5</td>
<td>$P_{\text{success,attack}}$ (Without sets selection probability)</td>
<td>(3/4)$^a$</td>
<td>(7/8)$^b$</td>
<td>(13/16)$^b$</td>
<td>(25/32)$^b$</td>
</tr>
<tr>
<td>6</td>
<td>$P_{\text{success,attack}}$ (With sets selection probability)</td>
<td>(3/8)$^a$</td>
<td>(7/32)$^b$</td>
<td>(13/112)$^b$</td>
<td>(25/416)$^b$</td>
</tr>
</tbody>
</table>

*N= Number of challenge bits, n= number of fast phase rounds

Figure 5.18: Analysis of variations in success probabilities for specific challenge bits combinations discussed in table 5.14

(d) “Majority Vote” Attack

Table 5.15 shows the comparative analysis of success probabilities in guessing the challenge for “Majority Vote” attack. In this attack, adversary has the capability of comparing and finding those registers which are having maximum common bits. After finding the challenges where maximum bits are common, adversary construct sets. For example, sets for 2-bits challenge are: \{\{00, 11\}\}, \{\{01, 10\}\}, sets for 3-bits challenge are: \{\{000, 111\}, \{001, 010, 100\}, \{011, 101, 110\}\}, sets for 4-bits challenge are: \{\{0000, 1111\}, \{001, 0010, 0100, 1000\}\},
An adversary constructs these sets in a way that it contains challenges with palindrome challenges subset, two same bits subset, three same bits subset etc. For example, a 3-bits challenge contains a palindrome challenges subset: \{000, 111\}, two zero bits subset: \{001, 010, 100\} and one zero bits subset: \{011, 101, 110\}. Thus, the probability of success of this attack is calculated as:

\[
P_{\text{success, attack}} = P[\text{Adversary correctly guesses the challenge}] = P[\text{Adversary Guess} = 001 \text{ or } 010 \text{ or } 100 \cap \text{ Challenge} = 001 \text{ or } 010 \text{ or } 100] \cup P[\text{Adversary Guess} = 011 \text{ or } 010 \text{ or } 100 \cap \text{ Challenge} = 011 \text{ or } 101 \text{ or } 110] \cup P[\text{Adversary Guess} = 000 \text{ or } 111 \cap \text{ Challenge} = 000 \text{ or } 111]
\]

\[
= P[\text{Adversary Guess} = 001 \text{ or } 010 \text{ or } 100 / \text{Challenge} = 001 \text{ or } 010 \text{ or } 100] \times P[\text{Challenge} = 001 \text{ or } 010 \text{ or } 100] + P[\text{Adversary Guess} = 011 \text{ or } 101 \text{ or } 110 \cap \text{ Challenge} = 011 \text{ or } 101 \text{ or } 110] \times P[\text{Challenge} = 011 \text{ or } 101 \text{ or } 110] + P[\text{Adversary Guess} = 000 \text{ or } 111 / \text{Challenge} = 000 \text{ or } 111] \times P[\text{Challenge} = 000 \text{ or } 111]
\]

(5.28)

Assume that an adversary has the capability of guessing in one direction, i.e. LSB to MSB or MSB to LSB. There are three sets \{\{000, 111\}, \{001, 010, 100\}, \{011, 101, 110\}\}, and the overall success probability of attack calculated using equation 5.28 becomes:

\[
P_{\text{success, attack}} = (\frac{1}{2} \cdot \frac{2}{8} + \frac{1}{3} \cdot \frac{3}{8} + \frac{1}{3} \cdot \frac{3}{8}) = \frac{3}{8}
\]

(5.29)

If an attacker has to guess the challenge bit for ‘n’-rounds of fast phase in authentication and distance bounding protocol then overall probability of success of attack is: \(\left(\frac{3}{8}\right)^n\).

Further, if an adversary has the capability of guessing in both directions, i.e. LSB to MSB and MSB to LSB then equation 5.28 become

\[
P_{\text{success, attack}} = (\frac{1}{2} \cdot \frac{2}{8} + \frac{1}{3} \cdot \frac{3}{8} + 1 + \frac{2}{3} = \frac{7}{8})
\]

(5.30)

If an attacker has to guess the challenge bit for ‘n’-rounds of fast phase in authentication and distance bounding protocol then overall probability of success of attack is \(\left(\frac{7}{8}\right)^n\).

Table 5.15 shows the comparative analysis of success probabilities in guessing the challenge for “Majority Vote” attack when adversary has the capability of guessing the challenge in one or both directions. Results show that if the adversary has lesser capability, i.e. guessing the challenge in one direction then success probabilities of this attack reduces with increase in
challenge bits but if it is having capability of guessing in both directions then success probability is increasing with increase in challenge bits.

Table 5.15: Comparison of success probabilities in guessing the challenge for “Majority Vote” attack with increase in challenge bits

<table>
<thead>
<tr>
<th>Challenge bits</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>…</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Guessing in One Direction (LSB to MSB or MSB to LSB)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P\textsubscript{success,attack} (Without sets selection probability)</td>
<td>(1/2)^n</td>
<td>(3/8)^n</td>
<td>(1/4)^n</td>
<td>(5/32)^n</td>
<td>…</td>
<td>(1/N)^n if N is Even (N/2^k)^n if N is Odd</td>
</tr>
<tr>
<td><strong>Guessing in Both Directions (LSB to MSB and MSB to LSB)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P\textsubscript{success,attack} (Without sets selection probability)</td>
<td>(3/4)^n</td>
<td>(7/8)^n</td>
<td>(15/16)^n</td>
<td>(31/32)^n</td>
<td>…</td>
<td>(\left(1 - \frac{1}{2^k}\right)^n)</td>
</tr>
</tbody>
</table>

*N= Number of challenge bits, n= number of fast phase rounds

(e) **Random guessing when bit position is not known**

In this case, it is assumed that an adversary does not have any capability and it has to randomly guess the challenge bits for performing an attack. Thus, if it is randomly guessing the challenge bits then success probability of this attack is \(\left(\frac{3}{4}\right)^n\).

5.2.3 **Terrorist Attack**

In this attack, legitimate member node collaborates with adversary for revealing the secrets [139]-[141]. Adversary tries to convince the legitimate member node for providing the secrets. Once the secrets are disclosed to adversary then it can easily breach the security for providing false distance to other nodes. Probability of terrorist fraud attack is always higher than mafia fraud attack because in terrorist fraud an adversary gets the secrets from a legitimate member node of a subgroup rather than a member which has left the subgroup and distant apart as shown in fig. 5.19.

![Figure 5.19: Terrorist attack](image)

In the proposed mechanism, a legitimate node can not reveal its secret information easily because if it does so then it losses the trust. If trust score deteriorates then its chances of becoming authenticated member of a subgroup also deteriorate. Thus, instead of revealing
complete secret information it will reveal some part of it. For example, a three bits challenge contain following set elements: \{000, 001, 010, 011, 100, 101, 110, 111\}, If an adversary has the capability of forming the sets like: \{\{000, 001\}, \{010, 011\}, \{110, 111\}, \{100, 101\}\} with guessing the bits from LSB to MSB and legitimate node reveals two bits of challenges then success probability of attack is calculated as:

\[
P_{\text{success, attack}} = P[\text{Adversary correctly guesses the challenge}] = \\
P[\text{Adversary Guess} = 000 \text{ or } 001 \cap \text{Challenge} = 000 \text{ or } 001] \cup \\
P[\text{Adversary Guess} = 010 \text{ or } 011 \cap \text{Challenge} = 010 \text{ or } 011 ] \cup \\
P[\text{Adversary Guess} = 110 \text{ or } 111 \cap \text{Challenge} = 110 \text{ or } 111] + \\
P[\text{Adversary Guess} = 100 \text{ or } 101 \cap \text{Challenge} = 100 \text{ or } 101]
\]

\[
= P[\text{Adversary Guess} = 000 \text{ or } 001/\text{Challenge} = 000 \text{ or } 001] \times P[\text{Challenge} = 000 \text{ or } 001] + P[\text{Adversary Guess} = 010 \text{ or } 011 \cap \text{Challenge} = 010 \text{ or } 011] \times \\
P[\text{Challenge} = 010 \text{ or } 011] + P[\text{Adversary Guess} = 110 \text{ or } 111 /\text{Challenge} = \\
110 \text{ or } 111] \times P[\text{Challenge} = 110 \text{ or } 111] + P[\text{Adversary Guess} = 100 \text{ or } 101 / \\
\text{Challenge} = 100 \text{ or } 101] \times P[\text{Challenge} = 100 \text{ or } 101]
\]

\[
P_{\text{success, attack}} = \left(\frac{1}{8}\right)^n \left(\frac{1}{8}\right)^n \left(\frac{1}{8}\right)^n \left(\frac{1}{8}\right)^n = \frac{1}{2^n}
\]

Table 5.16: Comparison of success probabilities in guessing the challenge for terrorist attack

<table>
<thead>
<tr>
<th>Reveal (N-1) bits</th>
<th>Reveal (N-2) bits</th>
<th>Reveal (N-3) bits</th>
<th>...</th>
<th>Reveal (N-m) bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\frac{1}{2^n})</td>
<td>(\frac{1}{4^n})</td>
<td>(\frac{1}{8^n})</td>
<td>(\frac{1}{2^m})</td>
<td></td>
</tr>
</tbody>
</table>

*\(N=\) Number of challenge bits, \(n=\) number of fast phase rounds

If an attacker has to guess the challenge bit for ‘\(n\)’-rounds of fast phase in authentication and distance bounding protocol then overall probability of success of attack for 3-bits challenge is \(\left(\frac{1}{2}\right)^n\). It is also observed that the success probability does not change with increase in number of challenge bits but it changes with increase in number of revealing challenge bits as shown in table 5.16. Chances of success of attack reduce with reduction in number of revealing challenge bits.

5.2.4 Distance Hijacking Attack

This attack is different from distance fraud and terrorist fraud attack. In distance fraud, one or more adversaries collude to convince legitimate members. In terrorist fraud, legitimate
member collude with adversaries but in distance hijacking attack, dishonest legitimate member collude with honest prover and involves them for false distance [138]. Since, nodes in the network are trusted, thus probability of distance hijacking attack is less than distance fraud attack. Although dishonest nodes try to convince honest nodes, for revealing the secrets, but honest nodes do not reveal because if they do so then these nodes lose their trust. Also,

\[ P_{success,\text{attack}} \propto P[\text{honest nodes reveal secret information without being dishonest}] = (1 - p_{\text{threshold}})^n \]

i.e., higher the value of threshold trust score more trusted nodes will be part of network and higher will be the probability of protection from distance hijacking attack. If still higher trusted honest nodes are ready to reveal some secret information at a cost of minute reduction in trust score then probability of success of this attack is same as of terrorist attack. Although this attack is same as MiM attack in mafia fraud and terrorist fraud attacks but complete secret information bits are not revealed. If dishonest member node reveals some bits of secret information then additional capabilities of dishonest legitimate member increases the chances of success of this attack. Although revealing some secret bits of information and capability of dishonest adversary increases the success of this attack but this attack is feasible only if dishonest legitimate member has capability of guessing in one direction, i.e. LSB to MSB or MSB to LSB. Any additional capability does not increase the chances of success of this attack.

### 5.2.5 Comparative Analysis

Table 5.17 shows the comparative analysis of distance bounding protocols based attacks for proposed protocol with existing systems. Results show that the proposed system is strongly protected against attacks as compared to existing systems. The proposed system provides a maximum of \((0.75)^n\) and a minimum of \((0.25)^n\) whereas the existing systems provides a maximum of \((0.875)^n\) and a minimum of \((0.5)^n\) for success probability of distance fraud, mafia fraud and terrorist fraud attacks. Since the worst of proposed system is 50% better than worst of existing system, thus the proposed system is strongly protected against distance based attacks. The best case of success probability of attack is possible only when an adversary has the maximum capabilities of accessing the challenge which is almost impossible because trust mechanism does not allow un-trusted members to become part of the network.
Table 5.17: Comparative analysis of success probabilities of distance bounding protocols based attacks for 2-bits challenge

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Distance Fraud</th>
<th>MiM (Mafia fraud and Terrorist fraud)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brands and Chaum, 1993 [132]</td>
<td>*(0.5)^a to (0.75)^b</td>
<td>*(0.5)^a to (0.75)^b</td>
</tr>
<tr>
<td>Čapkun et al., 2003 [252]</td>
<td>*(0.5)^a to (0.75)^b</td>
<td>*(0.5)^a to (0.75)^b</td>
</tr>
<tr>
<td>Hancke and Kuhn, 2005 [133]</td>
<td>(0.5)^a to (0.75)^b</td>
<td>(0.5)^a to (0.75)^b</td>
</tr>
<tr>
<td>Bussard and Bagga, 2005 [253]</td>
<td>1</td>
<td>*(0.5)^a to (0.75)^b</td>
</tr>
<tr>
<td>Reid et al., 2007 [254]</td>
<td>(0.75)^a</td>
<td>-</td>
</tr>
<tr>
<td>Singelée and Preneel, 2007 [255]</td>
<td>*(0.5)^a to (0.75)^b</td>
<td>*(0.5)^a to (0.75)^b</td>
</tr>
<tr>
<td>Tu and Piramuthu, 2007 [256]</td>
<td>(0.75)^a</td>
<td>1</td>
</tr>
<tr>
<td>Munilla and Peinado, 2008 [135]</td>
<td>(0.75)^a</td>
<td>(0.6)^a to (0.75)^a</td>
</tr>
<tr>
<td>Kim et al., 2008 [136]</td>
<td>-</td>
<td>*(0.5)^a to (0.75)^b</td>
</tr>
<tr>
<td>Avoine and Tchamkerten, 2009 [168]</td>
<td>-</td>
<td>*(0.5)^a to (0.75)^b</td>
</tr>
<tr>
<td>Kim et al., 2009 [257]</td>
<td>(0.875)^a</td>
<td>-</td>
</tr>
<tr>
<td>Durholz et al., 2011 [258]</td>
<td>-</td>
<td>*(0.5)^a to (0.75)^b</td>
</tr>
<tr>
<td>Zhuang et al., 2013 [137]</td>
<td>(0.75)^a</td>
<td>(0.75)^a to (0.875)^b</td>
</tr>
<tr>
<td>The Proposed Protocol</td>
<td>*(0.25)^a to (0.75)^b</td>
<td>*(0.25)^a to (0.75)^b</td>
</tr>
</tbody>
</table>

*For pre-defined challenge

5.3 Tag and Ownership Transfer Protocol based Attacks

In this section, analysis of secret disclosure and impersonation attack is performed for proposed protocol using Alloy. As discussed in distance bounding protocols, a member node does not reveal its secret information to an adversary because if it does so then it losses the trust. In order to be trusted authenticated node, it must not reveal secret information bits. If it reveals some secret information bits then both adversary and trusted node run the matching of challenge or verification bits simultaneously which does not ensure the bit matching in fast phase of authentication and distance bounding protocol. In results, this process ensures the presence of adversary. Other rules in detecting the secret disclosure attack include [142]:

(a) A node should not allow any other un-trusted node’s request of executing any operation.

(b) A node tries to associate with another node without trust score, authentication or integrity check.

(c) An adversary node drop messages or convince the other nodes for dropping messages.
Table 5.18 shows the detection time comparison for secret disclosure and tag impersonation attacks. Results show that average detection time in proposed system is 63.1% better than existing system [142]. If ‘N’ is the total number of challenge bits then overall probability of success of secret disclosure or impersonation attack for ‘n’ rounds is \((1-(1-1/N)^n)\).

Table 5.18: Analysis of tag and ownership transfer protocol based attacks for 2-bits challenge using Alloy

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Secrets Disclosure Attack</td>
<td>160 msec (10000 nodes)</td>
<td>1-(1-1/N)</td>
<td>101 msec (10000 nodes)*</td>
<td>1-(1-1/N)</td>
<td>63.1</td>
</tr>
<tr>
<td>Tag Impersonation Attack</td>
<td>300 msec (10000 nodes)</td>
<td>1-(1-1/N)*</td>
<td>180 msec (10000 nodes)*</td>
<td>1-(1-1/N)*</td>
<td>60.0%</td>
</tr>
</tbody>
</table>

*Average of 5 executions

Tag Impersonation attack is feasible with secret disclosure through simple mathematical operations. In the proposed system, a node would not be interested in revealing the secret information at a cost of losing the trust. Also, a strong one way hash function is used for authentication which protects from forward and backward secrecies. As discussed, if an adversary does not have the secrets or mechanism of backward secrecy then if it tries to do simple operations over the challenge bits for impersonation then chances of success of attacks are very less. The process of impersonation fails at some stage of authentication. Table 5.18 shows the detection time for impersonation for 10000 nodes using Alloy. Results show that with the process of authentication failure the detection time for impersonation is 60% better than existing system.

5.4 Ad-hoc Network based Attacks

5.4.1 Sybil Attack

A Sybil node represents two or more nodes moving together. Aim of Sybil node is to convince other nodes for receiving or transmitting their data. This is possible when it represents itself with multiple addresses as shown in fig. 5.20. In the proposed hierarchical structure, every subgroup has a parent node in parent subgroup which receives, forwards or transmits data or requests packets to/from subgroup. In the proposed Sybil node detection mechanism, every parent node in parent subgroup is facilitated with a functionality of collecting the MAC addresses of nodes which represents multiple logical addresses. Table
5.19 shows the detection time of Sybil attack for 1000 nodes using ns-3. Results show that maximum average detection time for 1000 nodes in proposed detection system is 310 msec. As compared to other systems [259][260], the proposed detection mechanism in proposed hierarchical structure is better because the existing mechanism adds additional data with data packets for detecting the attack whereas in the proposed detection mechanism MAC and logical addresses are used for detection. MAC and logical addresses are part of every data packet transmission, thus there is no additional time required for processing the extra data with data packets. Also, key is refreshed through parent node of parent subgroup which has the capability of detecting the Sybil attack and key is refreshed on regular intervals, thus there is no need of additional detection mechanism inside the subgroup also.

![Figure 5.20: Sybil attack](image)

<table>
<thead>
<tr>
<th>Attack</th>
<th>Detection Time in other Systems</th>
<th>Detection Time in Proposed System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sybil</td>
<td>≤1.0 seconds [259][260]</td>
<td>≤310 msec.*</td>
</tr>
</tbody>
</table>

*1000 nodes, average value of 5 executions

5.4.2 Eclipse Attack

An eclipse attack is possible by separating a network into two or more parts and controlling the traffic through those nodes which are acting as an interface between them. Fig. 5.21 shows an example of 11 node network. In this example, node 11 is acting as an attacker node which colludes with node 9 and node 10 for separating the network into two parts. First part contains node 1 to node 5 and second part contains node 6 to node 8. In this attack, an attacker node colludes with captured nodes for controlling the traffic between two parts. Sybil attack allows the execution of eclipse attack in an unstructured network by capturing the multiple logical addresses. The proposed hierarchical MANET is a structured network and
this attack is possible by controlling the subgroup controllers only. Subgroup controller is the most trusted member node of every subgroup, thus it reduces the chances of this attack. However, the parent node in parent subgroup of every subgroup is having the capability of detecting the multiple logical addresses through MAC addresses which detects this attack with a maximum average detection time of 500 msec. for 1000 nodes using ns-3. As shown in table 5.20, the proposed detection mechanism is better than existing mechanisms [216][261] because of structural routing in hierarchical network, selection of member nodes in its proximity, limits of in-degree and out-degree on each member node and detection mechanism through MAC addresses without any additional processing cost.

<table>
<thead>
<tr>
<th>Attack</th>
<th>Detection Time in other Systems</th>
<th>Detection Time in the Proposed System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eclipse</td>
<td>≤2.3 seconds [216][261]</td>
<td>≤500 msec.*</td>
</tr>
</tbody>
</table>

*1000 nodes, average value of 5 executions

5.4.3 De-synchronization Attack

De-synchronization attack is a MiM attack as shown in fig. 5.22. In this attack, MiM modifies the pre-shared secret between source and destination in fast phase of authentication and distance bounding protocol. If MiM knows the constants A and B then it will receive the data from destination and modify the data for sending it to source. Although source is sending negative acknowledgment (NACK) but MiM replies with verification challenge ‘Y2’. In the proposed authentication and distance bounding protocol nodes are considered un-authentic if hash bits are not verified, thus it easily detects the de-synchronization attack. Table 5.21 shows the analysis of de-synchronization attack. Results show that the proposed system is better than other system [10] because existing system uses secret values verification in attack detection whereas the proposed system uses hash bit verification in fast phase of authentication and distance bounding protocol which saves the detection time.
5.4.4 Bad Mouthing / Slandering and Promoting Attack

A malicious node may give incorrect feedback to undermine the trust management system. Incorrect feedback results in either Bad-mouthing attacks or Ballot-stuffing attacks. These are explained as follows:

a. Bad-mouthing attacks attempt to reduce the trust of a victim node. This is also known as slandering attack.

b. Ballot-stuffing attacks boost trust value of a malicious node. This is also known as promoting attack.

In the proposed trust management, feedback is an infinite sequence of independent values. It can have either positive or negative feedback. Past feedback does not affect the future outcomes. Suppose there are ‘PO’ positive and ‘NE’ negative feedbacks where, ‘PO’ and ‘NE’ are sufficiently large values. Using Bernoulli’s process, the probability of picking a node with a negative feedback is $p(-) = \frac{NE}{NE + PO}$ and a positive feedback is $p(+) = 1 - p(-)$. Now, probability that k-nodes provide positive feedback is calculated as: $p(k) = p(+)p(k-1) + (1-p(+))(1-p(k-1)) = (1+(2p(+)-1)^{k-1})/2$. Here, $p(1) = p(+)$. For example, a node can be selected with 0.5 probability in presence of bad mouthing attack when there are 347 positive feedbacks and 172 negative feedbacks and chances of selecting a node is formed based on opinions of k=5 selective nodes. To analyze the chances of Bad-mouthing or Ballot-stuffing attack, a simulation of 100 nodes for 1000 sec. was performed using ns-3. A node, which is maximum number of time acting as intermediate node, is selected for analysis. Table
5.22 shows the recommendation table of a node that maximum number of time acts as intermediate node. In this analysis, speed of nodes is varied from 2 m/s to 10 m/s. Table 5.22 shows two scenarios of target node. First scenario analyzes the neighbour node recommendation when trust mechanism is integrated in proposed hierarchical model whereas second scenario analyzes without trust mechanism. Results show that in both scenarios the recommendation score of neighbouring nodes decreases with increase in speed. There are more false negatives in the network because nodes get disconnected frequently with increase in speed. Recommendation is highest when speed is lowest. In this analysis, Bernoulli’s process is used for comparing and measuring the exact recommendation value in presence of positive and negative feedback. For the proposed scenario, the recommendation value is 0.7 for target node. Recommendation of nodes is higher and closer to exact value with trust mechanism as compared to without trust mechanism. In conclusion, the proposed trust mechanism in hierarchical MANET protects the network from Bad-mouthing or Ballot-stuffing attacks.

<table>
<thead>
<tr>
<th>Speed (m/s)</th>
<th>Feedback with trust mechanism</th>
<th>Feedback without trust mechanism</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neighboring Nodes</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>0.6</td>
<td>0.4</td>
</tr>
<tr>
<td>2</td>
<td>0.6</td>
<td>0.5</td>
</tr>
<tr>
<td>3</td>
<td>0.5</td>
<td>0.4</td>
</tr>
<tr>
<td>4</td>
<td>0.8</td>
<td>0.6</td>
</tr>
<tr>
<td>5</td>
<td>0.8</td>
<td>0.6</td>
</tr>
<tr>
<td>6</td>
<td>0.7</td>
<td>0.6</td>
</tr>
<tr>
<td>7</td>
<td>0.7</td>
<td>0.6</td>
</tr>
<tr>
<td>8</td>
<td>0.8</td>
<td>0.6</td>
</tr>
<tr>
<td>9</td>
<td>0.7</td>
<td>0.7</td>
</tr>
<tr>
<td>10</td>
<td>0.6</td>
<td>0.6</td>
</tr>
</tbody>
</table>

5.4.5 On-Off Attack

On-off attack works in a square wave as shown in fig. 5.23. In this attack, additional packets are inserted in TCP connection [221]. Unlike collusion attack, no additional traffic is generated by nodes in this attack. Additional packets increase the overhead and reduce the performance by introducing a transient condition. This transient condition adaptation requires network adaptation mechanisms. These adaptation mechanisms force the target node to delay
the packet handling and enter into a state of inactivity. When a network goes into transient state then the attacker node also move to a state of inactivity.

Let $t_{as}$ be the start time of attack in the network, $t_{as,n}$ be the start time of attack on node ‘n’ and $t_{as,n} = t_{as} + J_a$, $J_a$ be the jitter caused by an attack, $L^n_{on}$ and $L^n_{off}$ are the lengths of on and off periods of clock during data processing over node ‘n’, and ‘p’ be the probability that some node is attacked or launches an attack. Let $P(k)$ be the probability that a bottleneck occurs at point ‘k’ during the time of attack. $P(k) = 1 - p(k)$ is the probability that no bottleneck occurs at point ‘k’ during the time of attack.

![Figure 5.23: On-off attack cycle over a node ‘n’.](image)

Now, probability that no bottleneck occurs at ‘N’ points of a network during the time of attack:

$P(k, \text{network}) = \prod_{k=1}^{N} (1 - p(k))$ (5.33)

If $I_1, I_2, ..., I_N$ are the maximum inflows at ‘N’-points of a network during the time of attack then equ. 5.33 becomes:

$P(k, \text{network}) = \prod_{k=1}^{N} (1 - p(k)) = \max_{I_1, I_2, I_3, ..., I_N} \left\{ P(I_1, I_2, I_3, ..., I_N) \right\} = P(I_1(t_{as,n}), I_1(t_{as,n}) + I_2(t_{as,n}), ..., I_1(t_{as,n}) + I_2(t_{as,n}) + ... + I_1(t_{as,n}))$ (5.34)

If ‘$I_{min}$’ is the minimum flow rate at point ‘k’ during the time of attack then according to Kerner’s breakdown minimization (BM) principle [262], a situation with maximum probability of no traffic breakdown at network bottlenecks is selected. According to this principle, traffic control is dynamic and managed in the networks such that probability of breakdown in at least one bottleneck should reach to its minimum value.

i.e. \[ \min_{I_1, I_2, I_3, ..., I_N} \{1 - (1 - p^1(I_1(t_{as,n}))) \cdot (1 - (1 - p^2(I_2(t_{as,n}))) \cdot ... \cdot (1 - (1 - p^N(I_N(t_{as,n})))) \text{ and } I^{BM}_{min} = I_1 + I_2 + ... + I_N. \]
According to Wardrop’s first principle of user equilibrium (UE)[263], journey time of all routes actually used is equal and less than the journey time of single packet on unused route.

\[ TT_1(I_1(t_{a,s,n})) = TT_2(I_2(t_{a,s,n})) = TT_3(I_3(t_{a,s,n})) = \ldots = TT_N(I_N(t_{a,s,n})) \text{ and } I_{min}^{UE} = I_1 + I_2 + \ldots + I_N. \]

Here, \( TT(I_i) \) represents the travel time with inflow rate \( I_i, i \in \{1,2,\ldots, N\} \).

According to Wardrop’s second principle of system optimal (SO)[263], each user cooperates and selects his own best route such that there is an efficient use of network.

\[ \text{i.e. } \min_{I_1,I_2,I_3,\ldots, I_N}(TT_1(I_1(t_{a,s,n})) + TT_2(I_2(t_{a,s,n})) + \ldots + TT_N(I_N(t_{a,s,n}))) \text{, } I_{min}^{SO} = I_1 + I_2 + \ldots + I_N. \]

Randomly selected node to remove congestion which sends signal to neighboring nodes for discarding the recent messages.

Fig. 5.24 shows a scenario to implement three principles for removing the congestion by discarding the recent packets. Here, a random node in the congested path is selected for sending the signals to neighboring nodes for discarding the packets. For example, node N3 is selected to discard the recent packet with packet ID: 3 (from node 3 to node 6). N3 forward this message to N2 and N4. N2 further broadcast it to N1. Thus, the request of sending data from N3 to N6 is removed from congested path from N1 to N4. With this strategy, it is observed that \( I_{min}^{BM} > I_{min}^{SO} > I_{min}^{UE} \), i.e. inflow rate of network using BM-principle is maximum. A maximum inflow rate of 3 pkts/sec. is observed using BM-principle in a network of 100 nodes for 1000 sec. simulation using ns-3.

5.4.6 Collusion Attack

In this attack, two or more nodes collaborate with each other for disturbing the network services. If ‘\( \alpha \)’ and ‘\( \beta \)’ represent the probabilities of transmission and collision of node
respectively then according to Markov chain, probability that a collision is found in n\textsuperscript{th} round of communication is: \( \beta = 1 - (1- \alpha)^{n-1} \) and probability that a node successfully gets a slot whenever it wants to transmit the data is: \( (n.\alpha.(1- \alpha)^{n-1})/(1-(1- \alpha)^n) \). Further, if this model is used for simulation of 1000 nodes for 1000 sec. then it is observed that average throughput of no colluding node varies from 0.55 to 0.85 bytes/sec. and energy consumption per byte for successfully received data varies from 0.015 to 0.11 Jules for connection varies to 5 to 20 per node. If some nodes in a network give lesser throughput or consume more energy then those nodes are considered to be affected with collusion attack.

5.4.7 Whitewashing and Traitor Attack

This is one form of Sybil attack. In this attack, either an attacker cleans its history for removing its malicious activity records or uses all of its identities for capturing the maximum resources of the network. It is already analyzed in Sybil attack detection that capturing multiple identities can be detected in a reasonable time. The other major cause of this attack is through deleting the history and refreshing the identification. A subgroup controller or trusted subgroup member in parent subgroup can change or assign identification. This is refreshed by xor-ing a new random number to the current identification or assigned randomly. Let ‘k’ random number identities are available for use in one session and ‘n.k’ random number identities are available for ‘n’ sessions. For an attacker to get a new unique identity, it should listen to sessions between parties. Let an attacker be listening ‘n-k’ sessions. A subgroup is secure if subgroup controller or trusted subgroup member will not consider those ‘n-k’ sessions for refreshing the identities. Thus, number of refreshes in ‘n’ successive sessions is at least ‘n.(n-k)’, where k<n<2k. If each subgroup has at least two member nodes then it constructs a complete binary tree and requires at least ‘log n’ rounds for identification assignment. Since ‘2n’ messages are exchanged during one round of proposed identification assignment algorithm and algorithm completes with ‘n’ messages informing processes of results, thus message complexity of the algorithm is: ‘2n \log n + n’. Now, probability that one entity is refreshed= \((1/(n-k))^{2n \log n + n}\) and probability that a certain entity is not refreshed is less than \((1-1/(n-k))^{2n \log n + n}\).This equality is possible when one identification is refreshed once in every session and not twice or more. Now, \((1-1/(n-k))^{2n \log n + n} \approx 1/(n-k)^2\). In conclusion, an attacker requires n-1 listening in order to succeed in refreshing or assigning an identity which makes it almost impossible.
5.5 Conclusion

This chapter summarizes the probability and simulation based analysis of authentication and distance bounding, ownership transfer and Ad-hoc network based attacks. Results show that the proposed hierarchical MANET is strongly protected against these attacks. Analysis of authentication and distance bounding protocols based attacks show that the proposed system provides a best of $(0.75)^n$ and a worst of $(0.25)^n$ success probabilities of distance fraud, mafia fraud and terrorist fraud attacks which is 50% better than the existing systems. Tag and ownership transfer protocols based attack analysis shows that the proposed system detects secret disclosure and tag impersonation attacks in 160 msec. and 300 msec. respectively. This is 60% to 63.1% better than the existing systems. Ad-hoc network based attack analysis shows that the system detects the Sybil, eclipse, de-synchronization, bad mouthing, on-off, collusion, whitewashing and traitor attacks in a stipulated time, thus the proposed system is strongly protected against these attacks.