Trust- and Clustering-Based Authentication Service in Mobile Ad Hoc Networks

Presented by Edith Ngai

Supervised by Prof. Michael R. Lyu

Mphil thesis defense

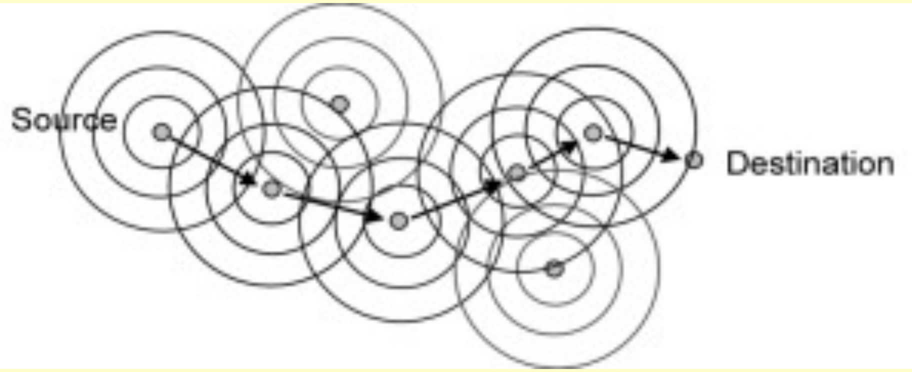
18 June 2004

Outline

- Introduction
- Related Work
- Architecture and Models
- Trust- and Clustering-Based Authentication Service
- Simulation Results
- Conclusion

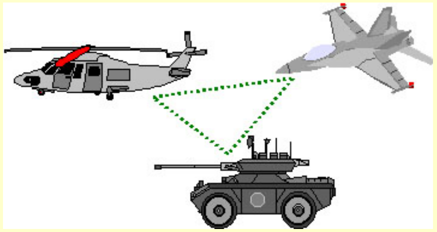
Mobile Ad Hoc Network

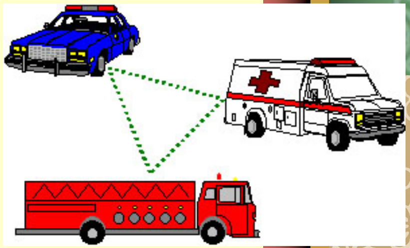
- An ad-hoc network (of wireless nodes) is a temporarily formed network, created, operated and managed by the nodes themselves.
- It is also often termed an infrastructure-less, self-organized, or spontaneous network.



Characteristics

- Connected with wireless communication
- Dynamic Topology
- Nodes are often mobile
- Vulnerable to security attacks
- Applications
  - Military: for tactical communications
  - Rescue missions : in times of natural disaster
  - Commercial use: for sales presentations or meetings





Vulnerabilities

- Unlike conventional networks, nodes of ad hoc networks cannot be secured in locked cabinets
- Risk in being captured and compromised
- Wireless communications are vulnerable to eavesdropping and active interference
- Adversary who hijacks an ad hoc node could paralyze the entire network by disseminating false routing information

Security Mechanisms

- Fundamental security mechanisms rely on the use of appropriate cryptographic keys
- Confidentiality, authentication, integrity, non-repudiation, access control and availability are considered as the main services of a security system
- Authentication service establishes the valid identities of communicating nodes
- The compromise of the authentication service breaks down the whole security system
- We focus on public key authentication service in our work

Trust and Security

- Trust in wired networks based on trusted certification agencies and authentication servers
- Trust in mobile ad hoc networks is still an open and challenging field
- Ad-hoc networks are based on naive “trust-your neighbour” relationships
- Non-presence of a central trust authority

Related Work

- Partially-distributed certificate authority makes use of a *(k,n)* threshold scheme to distribute the services of the certificate authority to a set of specialized server nodes.
- Fully-distributed certificate authority extends the idea of the partially-distributed approach by distributing the certificate services to every node.

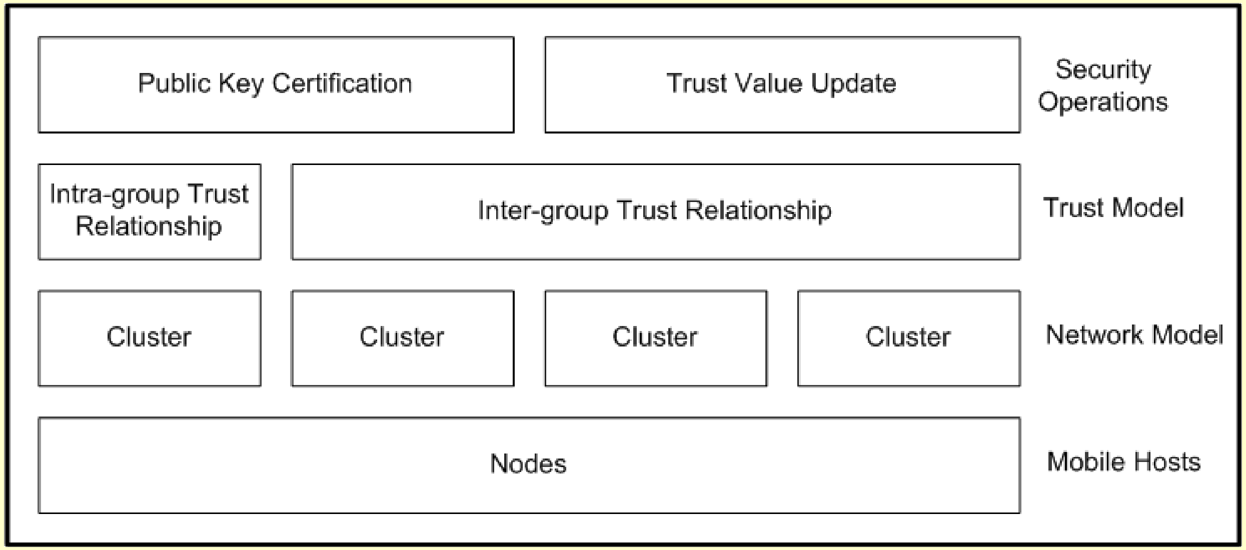
Related Work

- Pretty Good Privacy (PGP) is proposed by following a web-of-trust authentication model. PGP uses digital signatures as its form of introduction. When any user signs for another user's key, he or she becomes an introducer of that key. As this process goes on, a web of trust is established.
- Self-issued certificates distribute certificates by users themselves without the involvement of any certificate authority.

Our Work

- Propose a secure public key authentication service in mobile ad hoc networks with malicious nodes
- Prevent nodes from obtaining false public keys of the others
- Engage a network model and a trust model
- Design security operations including public key certification, identification of malicious nodes, and trust value update

Architecture

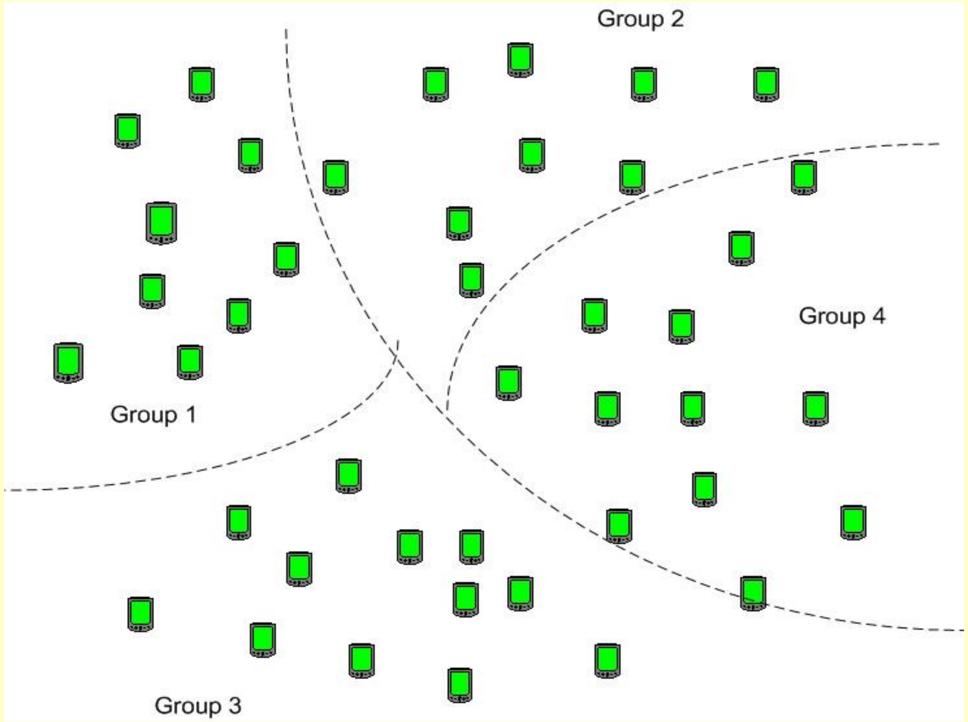


The Network Model

- Obtain a hierarchical organization of a network
- Minimize the amount of storage for communication information
- Optimize the use of network bandwidth
- Limit direct monitoring capability to neighboring nodes
- Allow monitoring work to proceed more naturally
- Improve network security

The Network Model

- Unique cluster ID
- Balance cluster sizes



The Trust Model

- Define a fully-distributed trust management algorithm that is based on the web-of-trust model, in which any user can act as a certifying authority
- This model uses digital signatures as its form of introduction. Any node signs another's public key with its own private key to establish a web of trust
- Our trust model does not have any trust root certificate. It just relies on direct trust and groups of introducers in certification

The Trust Model

- Define the authentication metric as a continuous value between 0.0 and 1.0
- Define a direct trust relationship as the trust relationship between two nodes in the same group and a recommendation trust as the trust relationship between nodes of different groups.
- The first formula calculates the trust value of a new recommendation path:

$$V_1 \Theta V_2 = 1 - (1 - V_2)^{V_1}$$

- The second formula draws a consistent conclusion when there are several derived trust relationships between two entities:

$$V_{com} = 1 - \prod_{i=1}^{m} \sqrt[ni]{\prod_{j=1}^{ni} (1 - V_{i,j})}$$

Clustering Structure Formation

Algorithm 1 Clustering structure formation

for each node n **do**
  Obtain trust values $t_{j,n}$ from its neighboring nodes j:
  $v_n \xrightarrow{\delta} v_{neighbor_k} :< v_n, REQ_{TRUST} >$;
  $v_{neighbor_k} \rightarrow v_n :< v_n, v_{neighbor_k,t} >$;
  Calculates its trust value by averaging the values $t_{j,n}$ received:

$$t_n = \frac{\sum_{j=1}^{k} t_{i,j}}{k} \quad (1.1)$$

  Initializes the winning pair $\langle WINNER_{TRUST}, WINNER_{ID} \rangle$ to be its trust value t_n and node ID;
end for
for each node n **do**
  Broadcasts its winning pair $< t_n, id >$ to its 1-hop neighbors for d-rounds in this Floodmax mechanism:
  for $i = 1$ to d **do**
    $v_n \xrightarrow{\delta} v_{neighbor_k} : \langle v_n, WINNER_{ID}, WINNER_{TRUST} \rangle$;
    $v_{neighbor_k} \rightarrow v_n : \langle v_j, WINNER_{ID}, WINNER_{TRUST} \rangle$;
    Updates the winner pair by selecting the one with maximum trust value;
  end for
end for
for each node n **do**
  Broadcasts its winning pair $< t_n, id >$ to its 1-hop neighbors for d-rounds in this Floodmin mechanism:
  for $i = 1$ to d **do**
    $v_n \xrightarrow{\delta} v_{neighbor_k} : \langle v_n, WINNER_{ID}, WINNER_{TRUST} \rangle$
    $v_{neighbor_k} \rightarrow v_n : \langle v_j, WINNER_{ID}, WINNER_{TRUST} \rangle$;
    Updates the winner pair by selecting the one with minimum trust value;
  end for
end for
for each node n **do**
  if $WINNER_{ID} == ID$ **then**
    Declares itself as a clusterhead;
  else
    Joins the clusterhead whose node ID occurs at least once as a winning pair in both the Floodmax and Floodmin rounds of flooding;
  end if
end for

1. Obtain its trust value from neighboring nodes
2. Run the FloodMax algorithm for d rounds
3. Run the FloodMin algorithm for d rounds
4. Select clusterhead

Clustering Structure Maintenance

- Maintain a balance clustering structure for supporting our trust model and security operations
- Adapt to the mobility of nodes
- Handle leave and join from one cluster to another
- Each node requests for the cluster ID of its neighboring nodes periodically
- In each cycle, a node collects this information and updates its cluster ID in different approaches

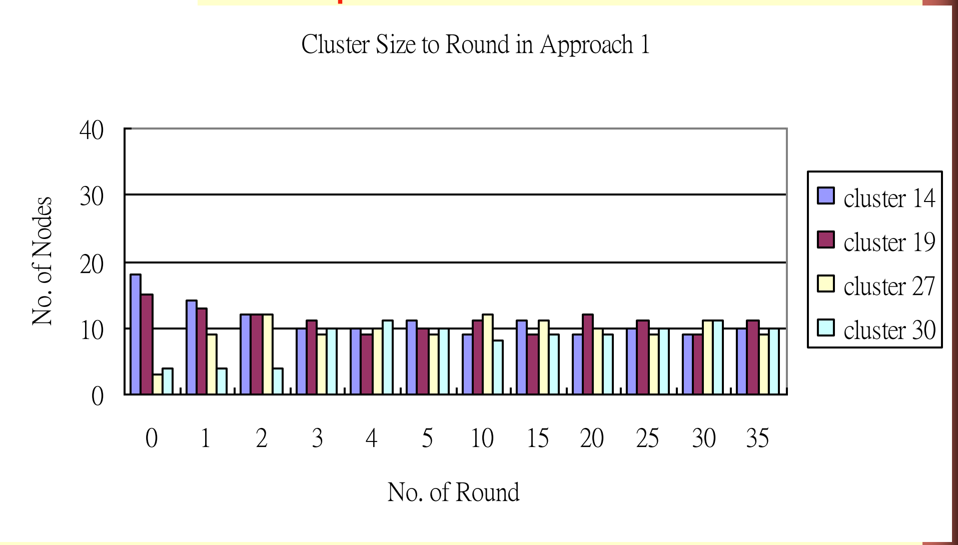
Approach 1

- A node updates its cluster ID by joining the neighboring cluster with minimum size in each cycle

Algorithm 5 Clustering Structure Maintaining - Approach 1

for each cycle **do**
  for each node n **do**
    $v_n \xrightarrow{b} v_{neighbor_k} :< v_n, REQ_{ClusterID} >;$
    $v_{neighbor_k} \rightarrow v_n :< v_n, v_j, ClusterID_j >;$
    $min_{size} = size\ of\ ClusterID_j;$
    $min_{cluster} = ClusterID_j;$
    for $\boxed{\forall\ ClusterID_j}$ **do**
      if $min_{size} < size\ of\ ClusterID_j$ **then**
        $min_{size} = size\ of\ ClusterID_j;$
        $min_{cluster} = ClusterID_j;$
      end if
    end for
    Joins the $min_{cluster}$;
  end for
end for

- 40 nodes in the network
- Keeps balance cluster sizes



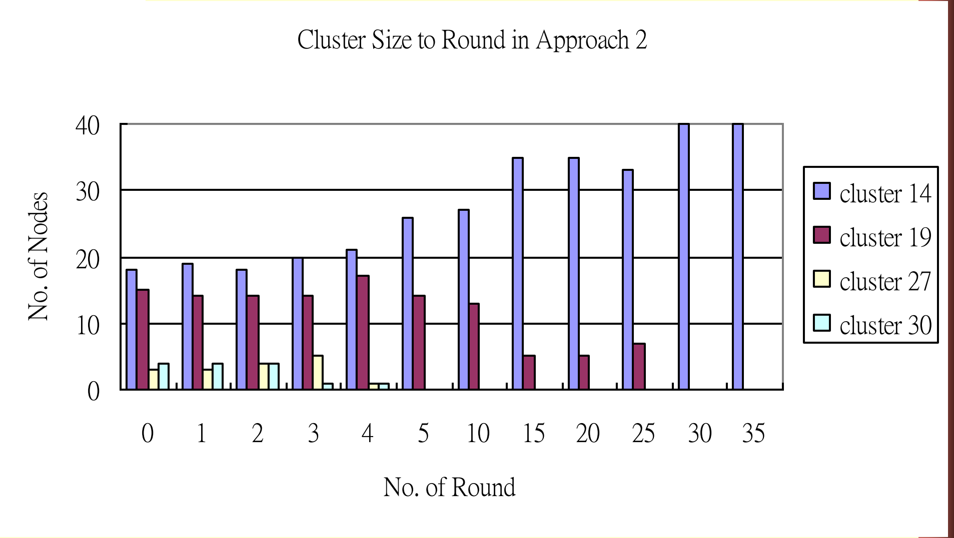
Approach 2

- A node joins the neighboring cluster with minimum size only if it leaves the original cluster

Algorithm 6 Clustering Structure Maintaining - Approach 2

for each cycle **do**
  for each node n **do**
    $v_n \xrightarrow{b} v_{neighbor_k} :< v_n, REQ_{ClusterID} >$;
    $v_{neighbor_j} \to v_n :< v_n, v_j, ClusterID_j >$;
    if $\boxed{ClusterID_n \neq \forall ClusterID_j}$ **then**
      $min_{size} = size\ of\ ClusterID_j$;
      $min_{cluster} = ClusterID_j$;
      for $\forall ClusterID_j$ **do**
        if $min_{size} < size\ of\ ClusterID_j$ **then**
          $min_{size} = size\ of\ ClusterID_j$;
          $min_{cluster} = ClusterID_j$;
        end if
      end for
    end if
    Joins the $min_{cluster}$;
  end for
end for

- Converge to one cluster
- Due to the imbalance cluster sizes after cluster formation?



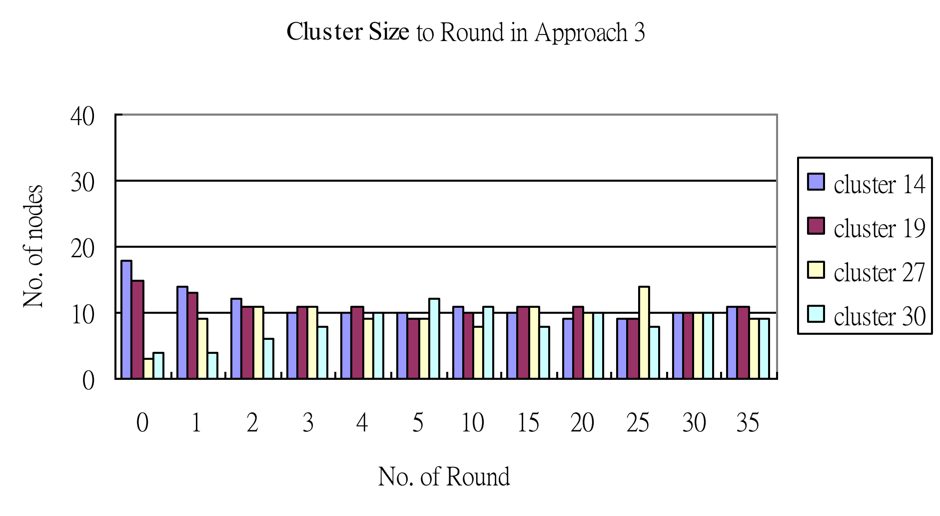
Approach 3

- A node joins the neighboring cluster with minimum size only if it leaves the original cluster or the sizes of the neighboring clusters are not within certain range

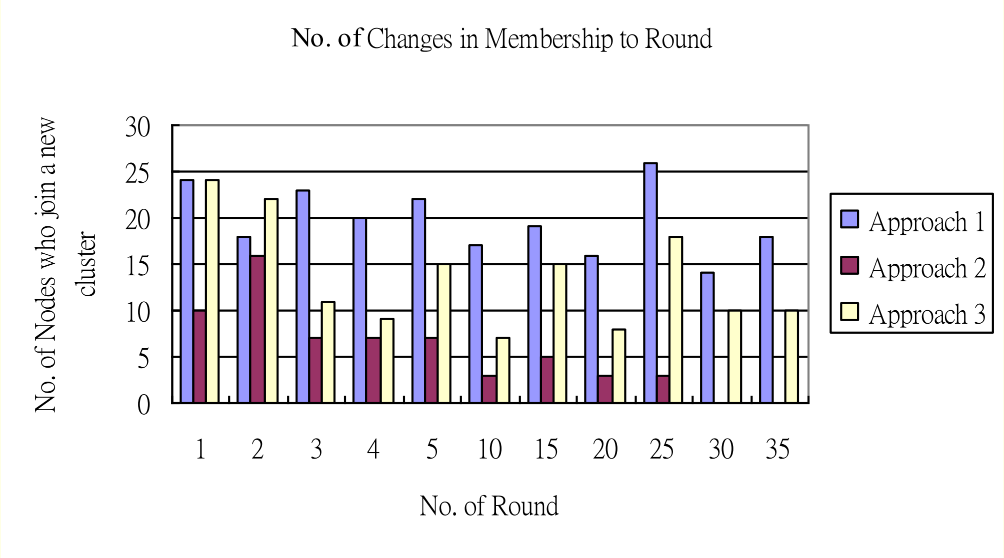
Algorithm 7 Clustering Structure Maintaining - Approach 3

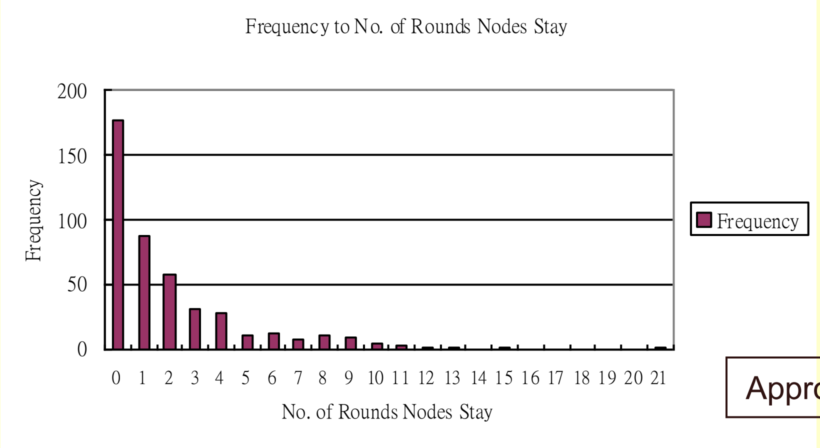
for each cycle **do**
  for each node n **do**
    $v_n \xrightarrow{\delta} v_{neighbor_k} :< v_n, REQ_{ClusterID} >$;
    $v_{neighbor_k} \to v_n :< v_n, v_j, ClusterID_j >$;
    if $\boxed{ClusterID_n \neq \forall ClusterID_j \text{ or } \exists!(S \leq size\ of\ ClusterID_j \leq L)}$ **then**
      $min_{size} = size\ of\ ClusterID_j$;
      $min_{cluster} = ClusterID_j$;
      for $\forall ClusterID_j$ **do**
        if $min_{size} < size\ of\ ClusterID_j$ **then**
          $min_{size} = size\ of\ ClusterID_j$;
          $min_{cluster} = ClusterID_j$;
        end if
      end for
    end if
    Joins the $min_{cluster}$;
  end for
end for

- Keeps balance cluster sizes



Changes in Membership

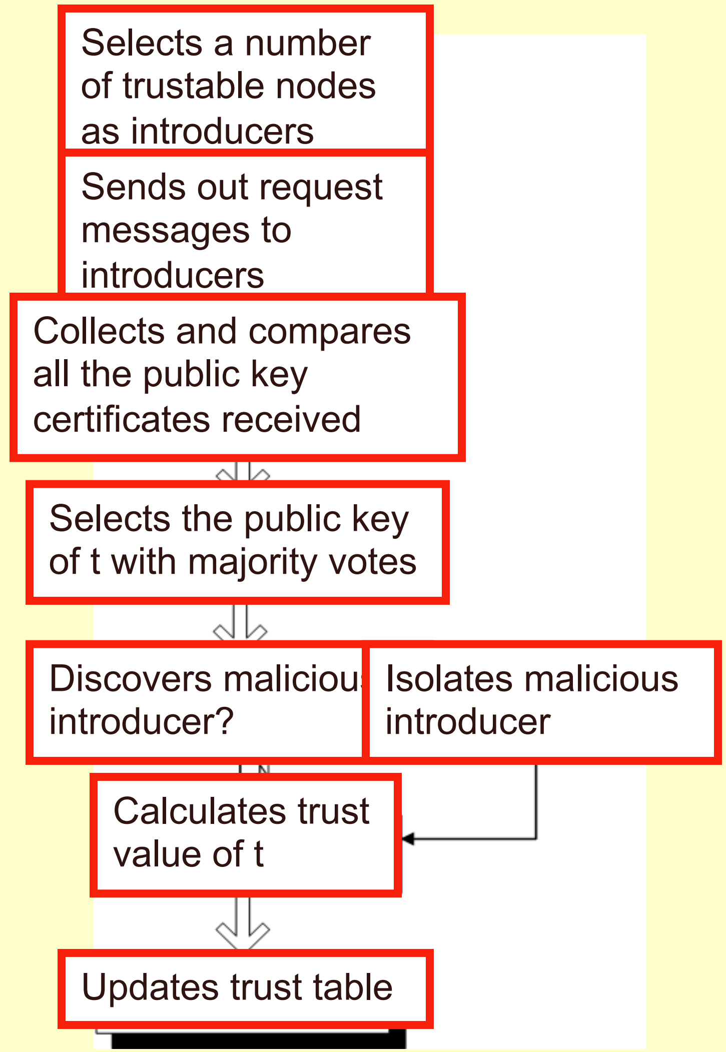




Approach 3

Authentication Service

- Public key certification
- Identification of Malicious Nodes
- Trust value update



Authentication Service

1. Looks up the group ID of t, φ_t.
2. Sorts the trust values of nodes belonging to group φ_t in the trust table. Let $i_1, i_2, \ldots, i_n \in I$, where $i_1, i_2, \ldots, i_n$ denote nodes with the highest trust values in group φ_t.
3. Sends request messages to nodes in I.
4. Collects the reply messages $m \in M$ from $i_1, i_2, \ldots, i_n$, where $m = \{Pk_t, V_{i_k,t}, \ldots\}_{Sk_{i_k}}$. Pk_t denotes the public key of node t, $V_{i_k,t}$ denotes the trust value from i_k to t, and Sk_{i_k} denotes the secret key of i_k. The reply message is signed by the secret key of i_k, Sk_{i_k}.
5. Compares the public keys received and selects Pk_t with the majority votes. Let $i_{good} \in I_{good}$ and $i_{bad} \in I_{bad}$, where i_{good} are the nodes that thought to be honest (agree on Pk_t with the majority) and i_{bad} are the remaining nodes that thought to be dishonest.
6. Reduces the trust values of i_{bad} to zero. Computes and updates the trust value of t, V_t, with the following formulae:

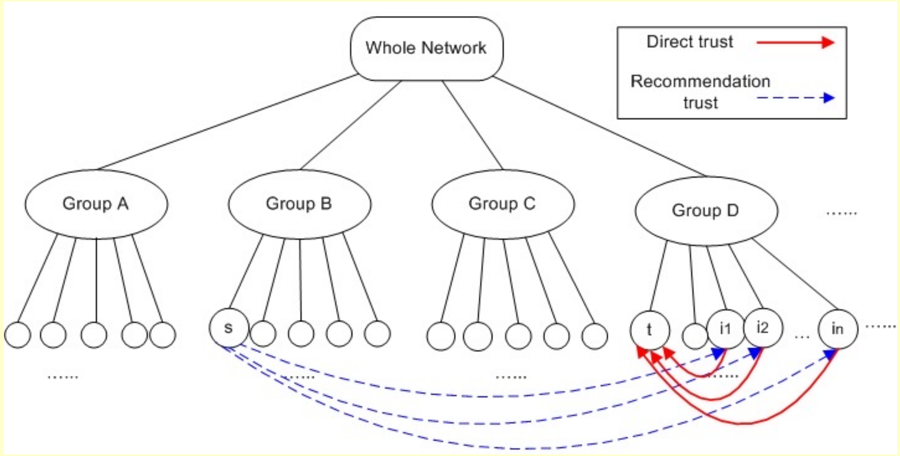
$$V_{s,i_k,t} = V_{s,i_k} \Theta V_{i_k,t} = 1 - (1 - V_{i_k,t})^{V_{s,i_k}}$$ and

$$V_t = 1 - \prod_{k=1}^{n} (1 - V_{s,i_k,t})$$ where i_k denote the nodes in I_{good} and n denotes the number of nodes in I_{good}.

1. Request for public key certificates
2. Identify malicious nodes
3. Update trust values

Public Key Certification

- Authentication in our network relies on the public key certificates signed by some trust-worthy nodes.
- Nodes in the same group always know each other better by means of their monitoring components and the short distances among them



Public Key Certification

Algorithm 4 Request for public key certificates

Given v_i belongs to $CLUST_A$ and v_j belongs to $CLUST_B$. A node v_i requests for the public key certificate of node v_j:

if $(CLUST_A == CLUST_B)$ **then**

  v_i sends request to neighbors v_k:

  $v_i \xrightarrow{b} v_k : \langle v_i, v_j, REQ_{CERT} \rangle;$

  $v_k \rightarrow v_i : \langle v_j, T_{v_k \rightarrow j}, PK_j, \ldots \rangle_{SK_{v_k}};$

  v_i updates PK_j and T_j;

else

  v_i selects trust-worthy nodes in $CLUST_B$ as introducers i_k;

  $v_i \xrightarrow{b} i_k : \langle v_i, v_j, REQ_{CERT} \rangle;$

  $i_k \rightarrow v_i : \langle v_j, T_{i_k \rightarrow j}, PK_j, \ldots \rangle_{SK_{i_k}};$

  v_i compares the PK_j from the received certificates and update PK_j in their repository;

  v_i calculates and updates T_j;

end if

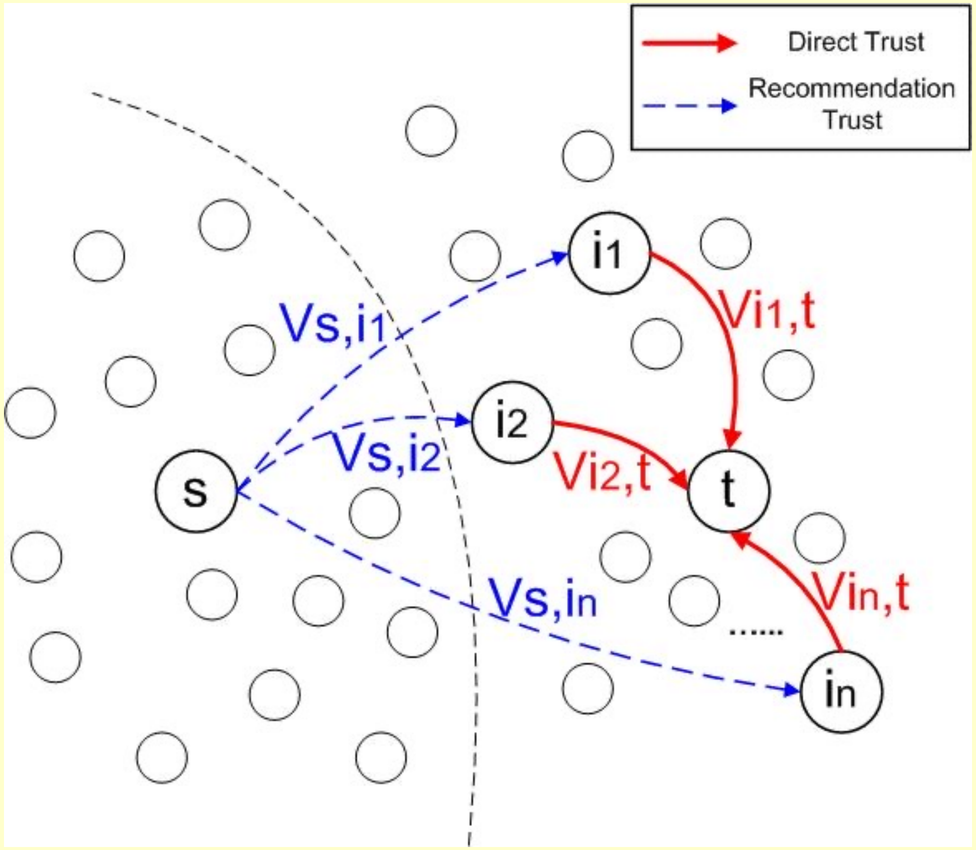
Send request to neighbors if target node in same cluster

Send request to introducers if target node in different cluster

Identification of Malicious Nodes

- Identify malicious neighboring nodes by monitoring their behaviors
- Identify introducers who provide public key certificates different from the others
- Identify target node as malicious if the trust values provided from the introducers indicate that

Trust Value Update



$$V_{s,i_k,t} = V_{s,i_k} \Theta V_{i_k,t} = 1 - (1 - V_{i_k,t})^{V_{s,i_k}}$$

$$V_t = 1 - \prod_{k=1}^{n} (1 - V_{s,i_k,t})$$

Parameters Setting

- Network simulator Glomosim
- Evaluate the effectiveness in providing secure public key authentication in the presence of malicious nodes

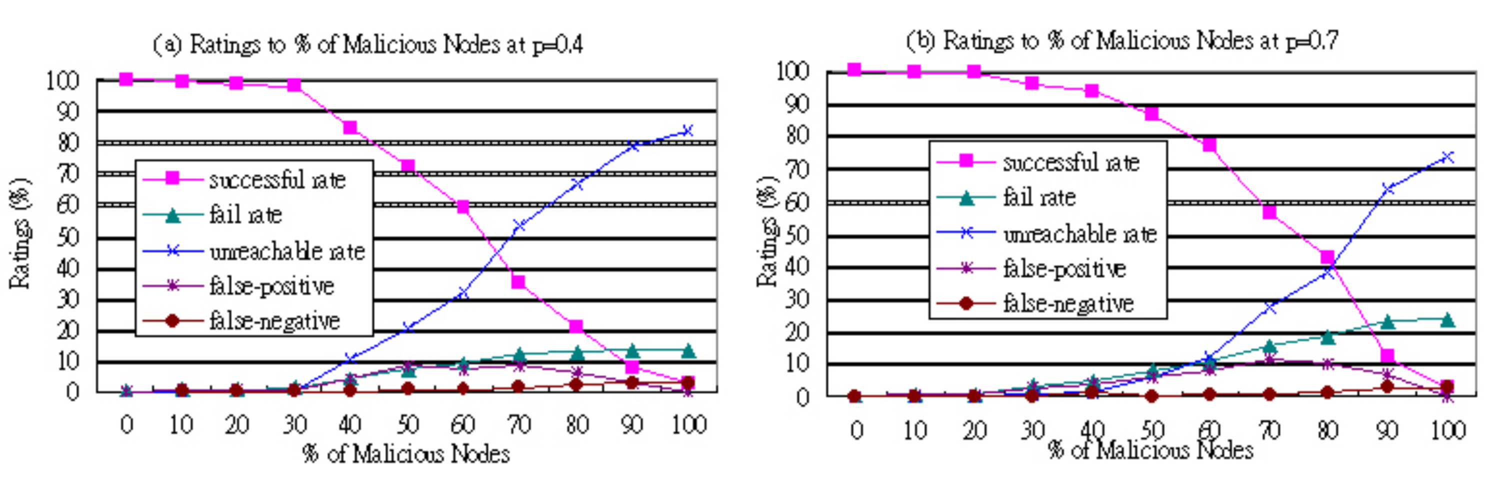
Network	
Network size	600m x 600m
No. of nodes	100
No. of groups	5
% of trustable nodes at initialization	p
% of malicious nodes	m
Mobility	
Mobility	Random-Waypoint
Pause Time	20s
Maximum speed	10m/s
PublicKeyCertification	
Max. no. of introducers for each request	3
Min. no. of reply for each request	1
No. of query cycles	80
No. of requests per cycles	100
Simulation Time	100000s

Simulation Metrics

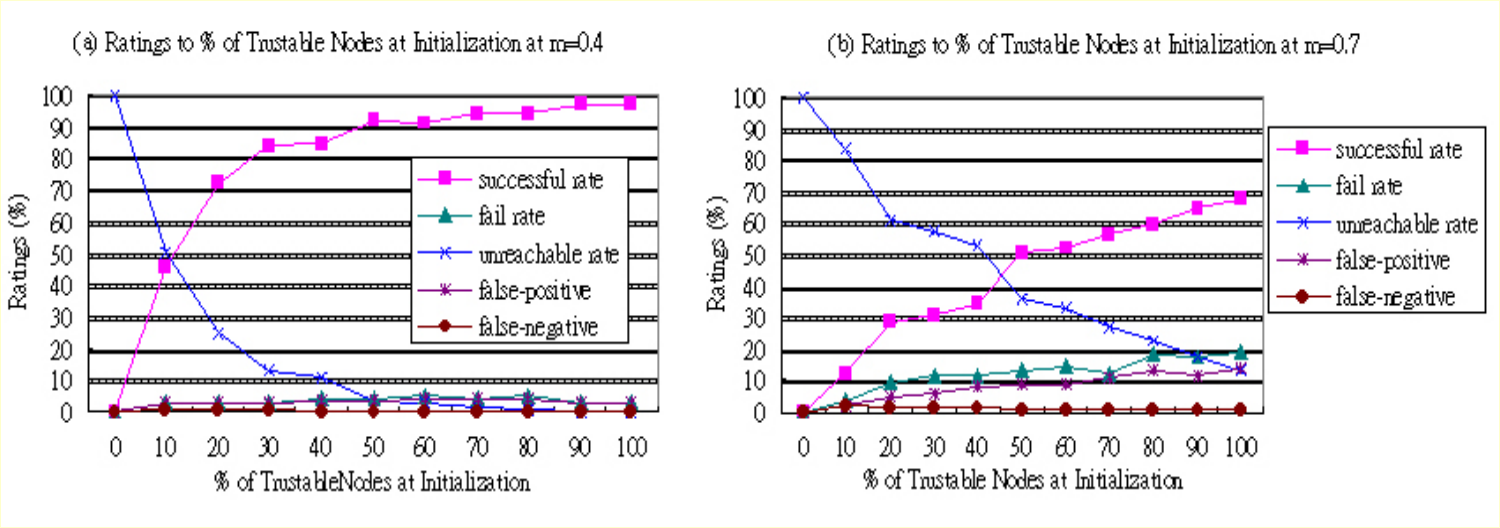
- Successful rate
- Fail rate
- Unreachable rate
- False-positive error rate
- False-negative error rate

ID	Cases	Successful Rate	Fail Rate	Unreachable Rate	False + Rate	False – Rate
0	Not enough Introducers			√		
1	OOO	√				
2	OOX	√				
3	OXX		√		√	
4	XXX		√			
5	OO	√				
6	OX		√		√	
7	XX		√			
8	O	√				
9	X		√			√
10	No Reply			√		

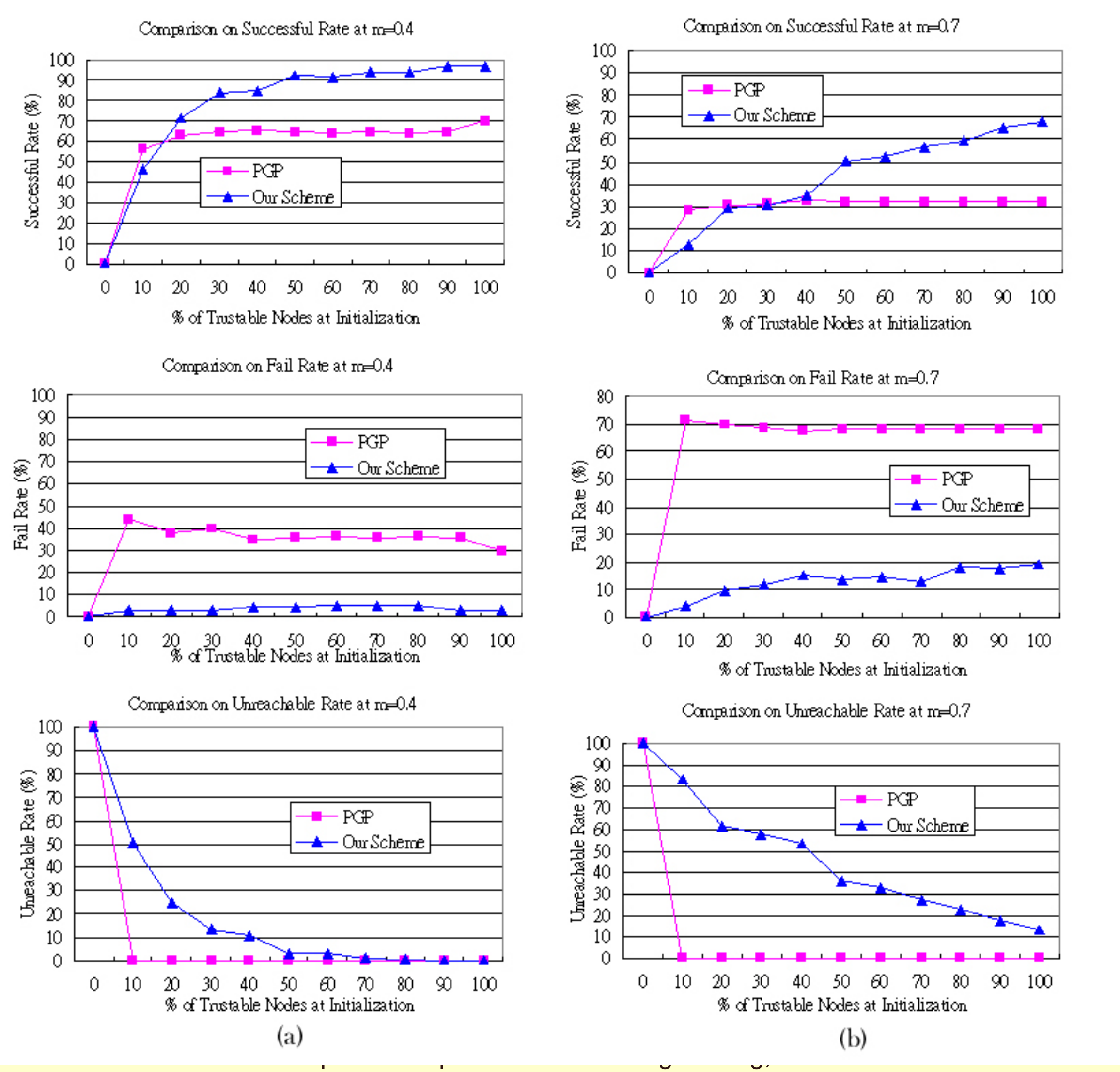
Ratings to Malicious Nodes



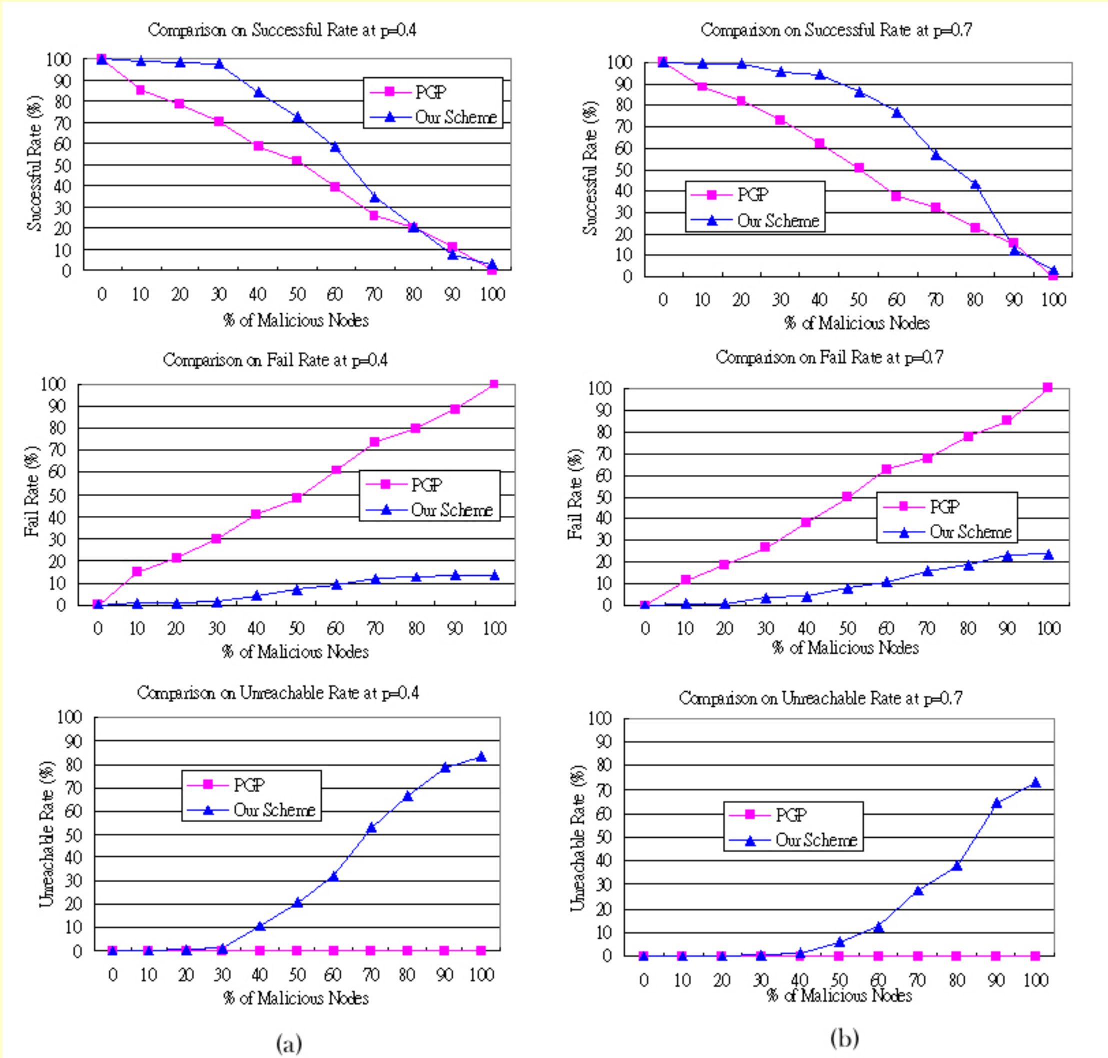
Ratings to Trustable Nodes at Initialization



Comparison to PGP with fixed *m*



Comparison to PGP with fixed *p*

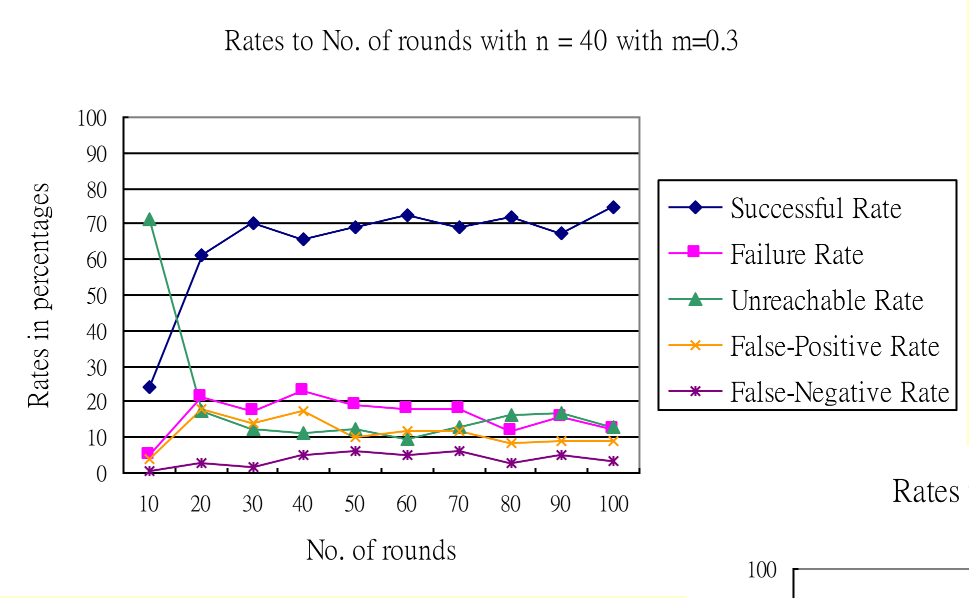


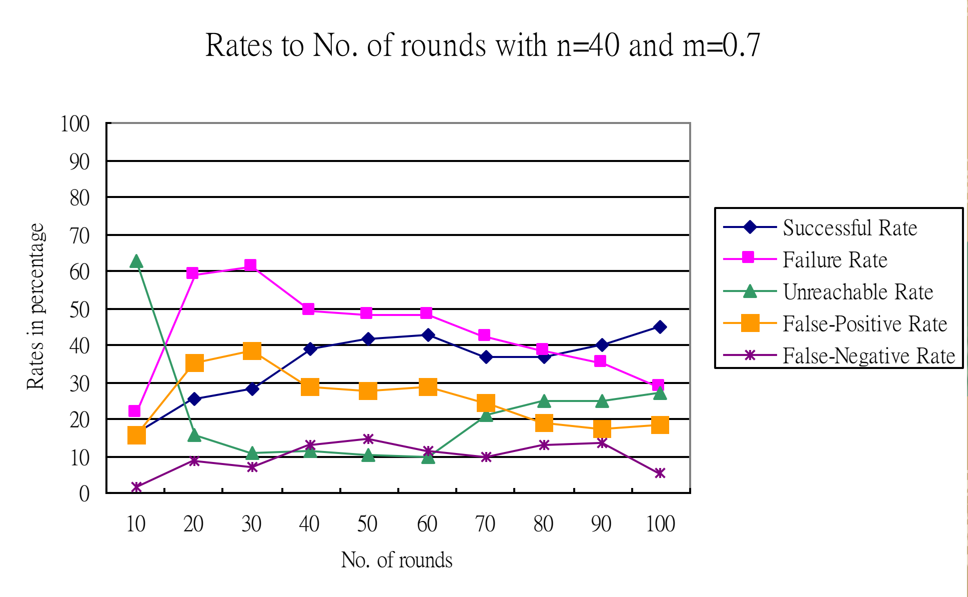
Parameters Setting

- This experiment includes the neighbor monitoring, clustering formation and maintenance algorithm

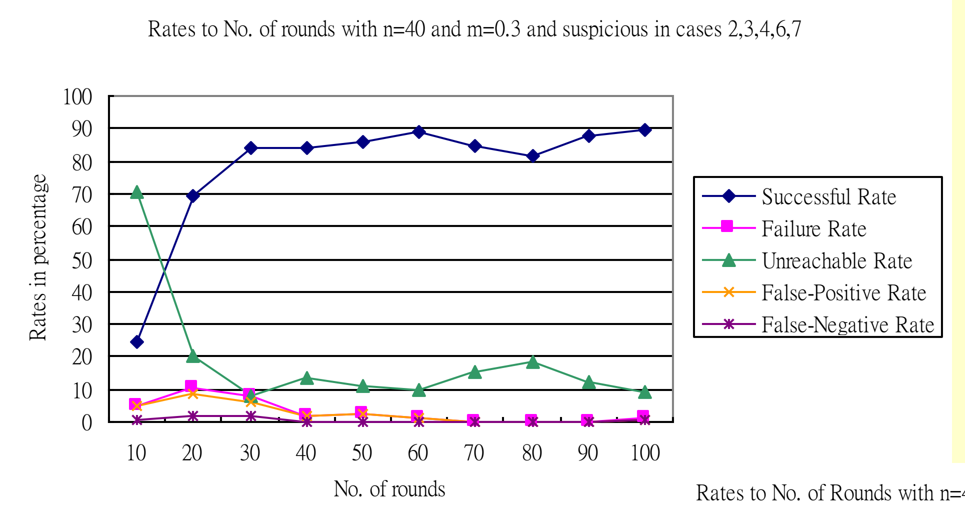
Network	
Network size	1500m x 1500m or 3000m x 3000m
No. of nodes	n
% of malicious nodes	m
Mobility	
Mobility	Random-Waypoint
Pause Time	20s
Max. speed	10m/s
Clustering	
D-hops	3
Min. cluster size	S
Max. cluster size	L
NeighborMonitoring	
No. of cycles required to identify malicious neighbors	2
PublicKeyCertification	
Max. no. of introducers for each request	3
Min. no. of reply for each request	1
No. of cycles	r
Simulation Time per cycle	110-120s

Neighbor Monitoring

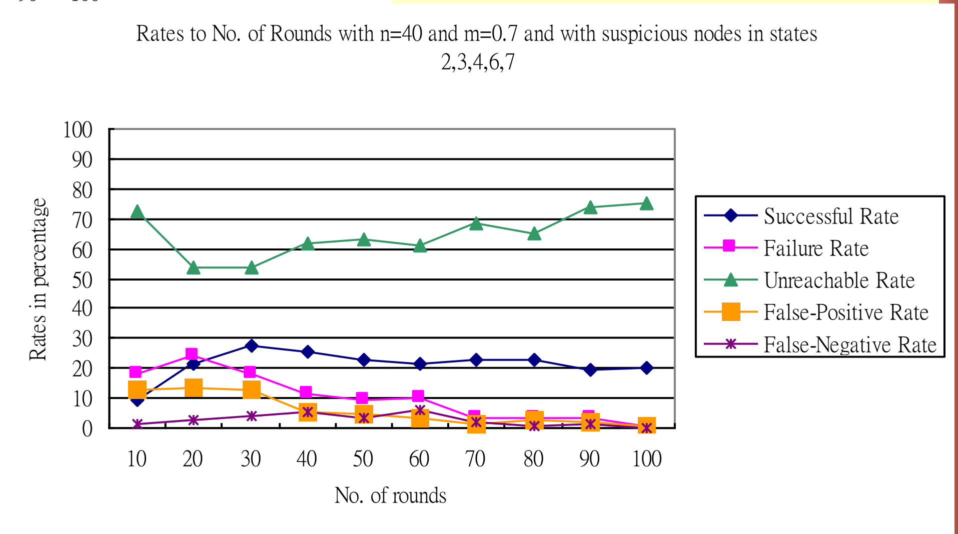




Identify Suspicious Nodes in cases 2,3,4,6,7

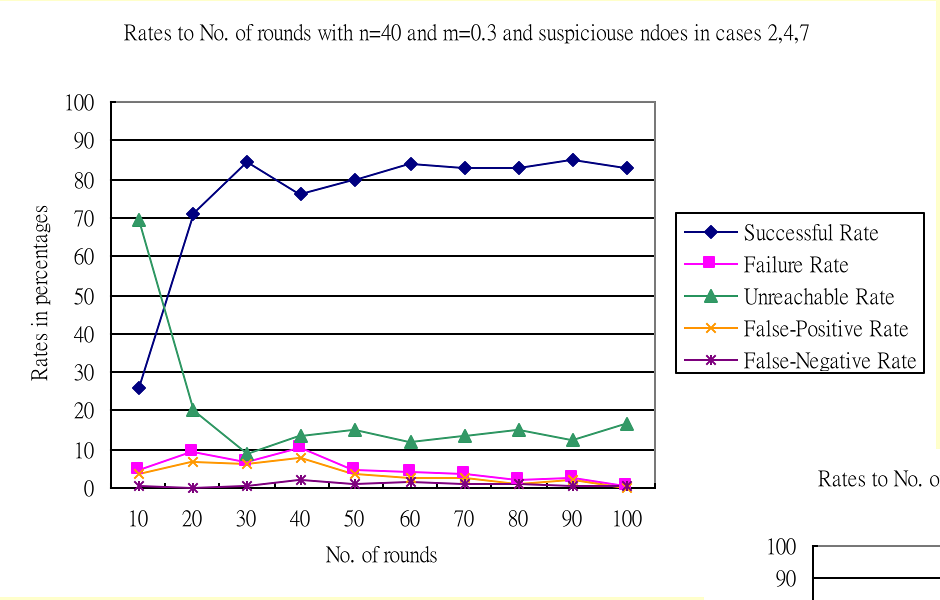


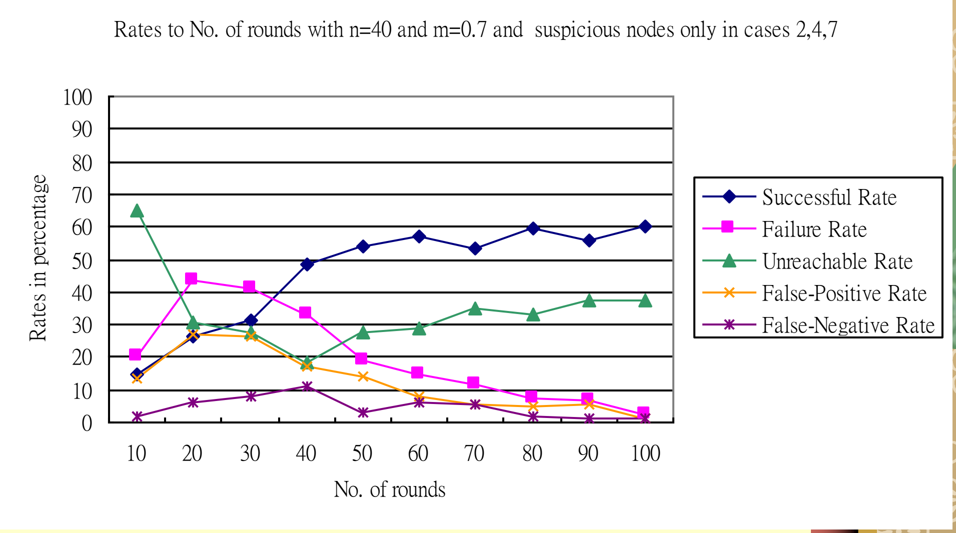
ID	Cases
0	Not enough Introducers
1	OOO
2	OOX
3	OXX
4	XXX
5	OO
6	OX
7	XX
8	O
9	X
10	No Reply



Identify Malicious Nodes in cases 2,4,7

ID	Cases
0	Not enough Introducers
1	OOO
2	OOX
3	OXX
4	XXX
5	OO
6	OX
7	XX
8	O
9	X
10	No Reply





Future Work

- Colluding Nodes
  - Revise trust values of nodes after real experiences with the public keys
- Trust Values Combination
  - New equations for trust values update
- Overhead
  - Evaluate the costs of the proposed scheme
- Address the problem of multiple identities

Conclusions

- We developed a trust- and clustering-based public key authentication mechanism
- We defined the network model as clustering-based and with a balance structure
- We defined a trust model that allows nodes to monitor and rate each other with quantitative trust values
- The authentication protocol proposed involves new security operations on public key certification, update of trust table, discovery and isolation of malicious nodes
- We conducted security evaluation
- We compared our approach with the PGP approach to demonstrate the effectiveness of our scheme