# An Authentication Service Based on Trust and Clustering in Mobile Ad Hoc Networks

M.Phil Term 3 paper

Department of Computer Science and Engineering
The Chinese University of Hong Kong

written by Edith C.H. Ngai
advised by Prof. Michael R. Lyu

Fall 2003

## Abstract

Wireless ad hoc network is a collection of mobile nodes dynamically forming a temporary network without a centralized administration. This kind of network has been applied for both civilian and military purposes. However, security in wireless ad hoc networks is hard to achieve due to the vulnerability of the links, the limited physical protection of the nodes, and the absence of a certification authority or centralized management point. Consequently, novel approaches are necessary to address the security problem and to corporate with the properties of wireless ad hoc network. Similar to other distributed systems, security in wireless ad hoc networks usually relies the use of different key management mechanisms. In this paper, we present a public key authentication service to protect security in the network with malicious nodes. Nodes originally trustable in the network may be compromised after the attacks. These malicious nodes can harm the authentication service by signing false certificates, so adequate measure is essential to protect the network security. We develop a novel authentication service based on trust and clustering. It involves a well-defined network model and a trust model. These models allow nodes in the network to monitor and rate each other with an authentication metric. We also propose a new public key certificate operation, and corporate with a trust value update algorithm in public key authentication. Our authentication service is able to discover and isolate malicious nodes in the network. Finally, we perform security evaluation on the proposed solution through simulation. We simulate the network with malicious nodes and measure a number of metrics. In addition, comparison and analysis are made between our approach and the Pretty Good Privacy with distributed certificate repository to demonstrate the effectiveness of the scheme.

# 1 Introduction

Wireless ad hoc network is a collection of mobile devices forming a network without any supporting infrastructure or prior organization. Nodes in the network should be able to sense and discover with nearby nodes [16]. Due to the limited transmission range of wireless network interfaces, multiple network ”hops” may be needed for one node to exchange data with another across the network [10]. There are a number of characteristics in wireless ad-hoc networks, such as the dynamic network topology, roaming of the nodes, limited bandwidth and energy constrain in the network. A crucial difference between ad hoc networks and traditional networks is the lack of central administration or control. This factor leads to a serious problem in network security with the limited physical security on wireless communication. Mobile wireless networks are generally more prone to physical security threats than are fixed-cable nets. The increased possibility of eavesdropping, spoofing, and denial-of-service attacks should be carefully considered [15]. In protecting this vulnerable network from different attacks, the availability of security services is very important [24].

The most common way to protect the network security is done by encryption and decryption of the messages. Public key cryptography has been recognized as one of the most effective mechanism for providing security service like authentication, digital signature, and encryption. Public key cryptography usually relies on the Certificate Authority (CA) to sign and validate digital certificates. Public key infrastructure (PKI) is deployed in wired network and some infrastructure-based wireless network. Security requirements for CAs are important with an exploration of the wide range of attackers that can be mounted against CAs [25]. Popular network authentication architectures include X.509 standard [1] and Kerberos [26]. Another paper suggests make use of interoperation between many small, independent certificate authorities to build a global-scale public-key infrastructure [17]. However, traditional key distribution schemes are not suitable for wireless ad hoc networks due to its network characteristics. Therefore, new security services are necessary to protect the network security in wireless ad hoc network.

Pretty Good Privacy (PGP) [3][18] 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 or trust is established [2]. Its distributed manner in certification is compatible with the characteristics of ad hoc networks. An approach similar to PGP for security in wireless ad hoc networks is proposed in [12][21]. That paper presents the idea of trust graph and the method of finding a certificate chain from one user to another. However, it assumes that users are honest and do not issue false certificates, though it briefly suggests that this assumption could be relaxed by the introduction of some sort of authentication metric. Although an authentication metric represents the assurance that a user can obtain the authentic public key of another, it is hard to be estimated accurately. There is still possibility for a node turns from trustable to malicious in a sudden attack. The ability for detecting such misbehavior and the isolation of malicious nodes are important in public key authentication. In this paper, we provide a secure authentication service that can defend malicious nodes in the network. In addition, we find that it is common to see performance evaluation on new security protocols proposed, but rare to see security evaluation on those works by experiment. Therefore, we carry out a series of simulation to evaluation the security provided by the authentication service we propose. We emulate a network with malicious nodes, which can harm authentication by issuing false certificate. The experiment shows that our authentication service performs well in protecting the authentication even in this hostile environment.

The remaining of this paper is organized as follows: Section 2 discusses the related work on the current key management systems, clustering techniques and trust valuation methods for ad hoc networks. Section 3 formalizes the system architecture, the network model and the trust model which lay the foundation for our design. In Section 4, we further present the security operations on the public key certification and the update of trust tables. The new solution is evaluated through simulation in Section 5. We fix and vary different parameters in the wireless ad hoc network and estimate its security performance in terms of the successful rate, fail rate, unreachable rate, false-positive error rate, and false-negative error rate. We also study the convergence time, effect of mobility, and make comparison of our security scheme with the PGP approach with distributed certificate repositories. Finally, we conclude the paper in Section 6.

# 2 Related Work

Several public key management protocols are proposed for wireless mobile ad hoc network. One of the active research areas is security function sharing [20], including a popular method for threshold secret sharing [34]. The basic idea is distributing the functionality of the centralized CA server among a fixed group of servers. Zhou and Hass proposed a partially distributed certificate authority that makes use of a $(k, n)$ threshold scheme to distribute the services of the certificate authority to a set of specialized server nodes [39]. Another public key infrastructure service called MOCA (Mobile Certificate Authority) was proposed. It employs threshold cryptography to distribute the CA functionality over specially selected nodes based on the security and the physical characteristic of nodes [35][36]. Furthermore, the fully-distributed certificate authority is proposed by Luo and Lu [27] extending the idea of the partially-distributed approach by distributing the certificate services to every node. Other solutions include the self-issued certificates proposed by Hubaux et. al. [21]. It issues certificates by users themselves without the involvement of any certificate authority.

In this paper, we suggest an authentication service that is different from the above protocols. The public key authentication service we propose involves a well-defined trust model and network model. It follows the “web of trust” model proposed in PGP [18] with our own contribution. In addition, it adopts a clustering-based network model in the meantime. One class of existing clustering algorithm in wireless ad hoc network is based on independent dominating sets of graphs. Weighted based clustering algorithms, on the other hand, are proposed in [19]. These algorithms define a vertex with optimal weight within its neighborhood is a clusterhead, and the neighborhood of a clusterhead is a cluster. The weight idea is generalized in [8]such that any meaningful parameter can be used as the weight to best exploit the network properties. Recent work is also performed on cluster formation such that a node is either a clusterhead or is at most d hops away from a clusterhead [7]. Weakly-connected dominating set is proposed for clustering ad hoc networks in [14]. A zonal algorithm for clustering ad hoc networks is proposed in [13] to divide the network into different regions and make adjustments along the borders of the regions to produce a weakly-connected dominating set of the entire graph. Moreover, a Group-based Distance Measurement Service (GDMS) is also proposed. Nodes in GDMS are self-organized into Measurement Groups (Mgroups) to form a hierarchical structure. A set of algorithms is proposed to handle network dynamics and optimize the group organization [29].

Regarding to the authentication in ad hoc network, it generally depends on a trust chain formed by trusted intermediaries. To evaluate the trusts from the recommendation of other reliable entities, the relying node needs to estimate their trustworthiness. It is a well-known technique for authenticating entities in a large-scale system. Some work has extended this technique to include multiple paths to strengthen authentication, but it has to handle intersecting paths, ambiguities in the meaning of certificates, and interdependencies in the use of different keys. A paper develop a set of guiding principles for the design of a satisfactory metric of authentication [31]. Different metrics have been proposed to evaluate the confidence afforded by the paths. A paper proposed a metric that represents a set of trust relationship by a directed graph [9]. It introduces the semantics of direct trust values differ from that of recommendation trust values. It shows that different values can be combined to a single value by considering the opinions from the respective recommending entities. The metric in PGP has three levels of trust, including the Complete trust, Marginal trust, and Notrust [40]. This approach requires one Completely trusted signature or two Marginally trusted signature to established a key as valid [33]. Another paper explores the use of multiple paths to redundantly authenticate a channel and focuses on two notions of path independence. They are the disjoint paths and connective paths that seem to increase assurance in the authentication [32]. Besides, a trust management method is proposed in [5] to address the problem of reputation-based trust management. It allows assessing trust by computing an agents reputation from its former interactions with other agents and manage data in decentralized way with P-Grid [6]. Moreover, a paper presents a distributed and secure method to compute global trust values, based on Power iteration. This algorithm improve the reputation management in P2P networks [23]. The distributed trust model is proposed based on recommendations [4]. In this model, discrete levels of trust are used and it develops an algorithm for calculating trust and using values in recommendations. Furthermore, a distributed scheme for trust inference in peer-to-peer networks. It describes a technique for efficiently storing user reputation information in a completely decentralized manner, and show how this information can be used to efficiently identify non-cooperative users [28]. Finally, a paper solves the problem of users who claim multiple, false identities, or who possess multiple keys, and whose that conflicting certificate information can be exploited to improve trustworthiness [22].

# 3 Models

In this section, we describe the architecture, network model, and trust model of our authentication service the authentication service we propose for wireless ad hoc network.

## 3.1 Architecture of the Authentication Service

The authentication service we propose aims at providing secure public key certification despite the presence of malicious nodes in the network. Malicious nodes in authentication may issue false certificates to the others. To deal with the problem, we propose a novel authentication service which is clustering- and trust-based. The reason is that the clustering-based network model gives advantages on the behavior monitoring among the nodes. The monitoring power of the nodes in wireless mobile ad hoc network is usually limited to its neighboring nodes, so nodes in the same cluster have relatively higher monitoring power with their short distances. With this feature, we assume that any node can monitor and obtain public keys of the nodes in the same group accurately unless they are compromised in a sudden attack. Apart from the clustering model, we define trust value as an authentication metric for indicating the assurance with which a requesting node $s$ can obtain the correct public key of a target node $t$. The chance for obtaining a correct public key certification increases if the node signing the certificate with high trusts value. Simply a clustering model and trust value are not enough in prohibiting dishonest users because a node with high trust value can still be malicious suddenly when it is attacked. Therefore, we design each public key request on new node with multiple replies, so that conclusion can be made with the majority votes. This operation improves the security for obtaining a correct public key and helps to discover dishonest user in the network. Trust value of the dishonest user will be reduced, so malicious nodes will be isolated in our authentication service.

Figure 1 shows the architecture of our authentication service. There are totally $4$ layers in this architecture, including the mobile hosts, network model, trust model, and the security operations. Wireless ad hoc network contains large amount of mobile hosts, each with a transmission range that is small relative to the network size. We divide the network into different region and nodes in the same region form a cluster. A cluster, or we call a group, is a connected sub-network usually with a smaller diameter. We define two kinds of trust relationship in the clustered network, including the trust relationship of two nodes within the same group and the trust relationship of two nodes in different groups. The security operations are performed on top of the lower layers. These operations include public key certification and trust value update, which will be presented in Section 4.

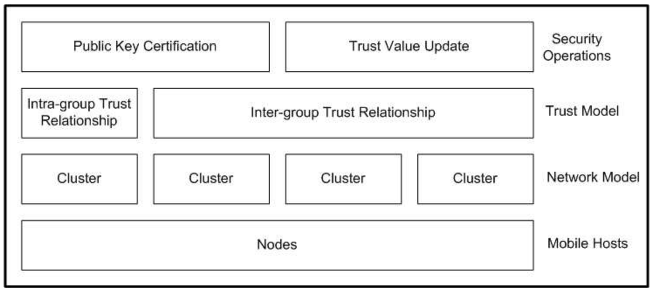


Figure 1: Architecture of Our Authentication Service

## 3.2 The Network Model

Since a wireless ad hoc network is an infrastructureless network that requires only mobile units to form the network, it can involve a large number of mobile units and each with a short transmission range. An important feature in wireless ad hoc networks is multi-hopping, which is the ability of the mobile units to relay packets through radios from one another without the use of base stations. Obtaining a hierarchical organization has been proven effective in minimizing the amount of storage for communication information, and in optimizing the use of network bandwidth.

Apart from the view of efficiency, we believe clustering improves the security of a network. Since wireless ad hoc network lacks of a centralized server for management, security measure relies on individual nodes to monitor each other. However, the monitoring capability of a node is normally limited to its neighboring nodes. On the other hand, nodes clustering together allow the monitoring work to proceed more naturally, so as to improve the overall network security. In this paper, we propose an authentication service in wireless ad hoc network by trust management and clustering techniques.

A number of existing solutions have been proposed for clustering in wireless ad hoc networks. In our design, we divide the network into different regions with similar number of hosts in each of them like in Figure 2. Nodes clustering together in the same region form a group

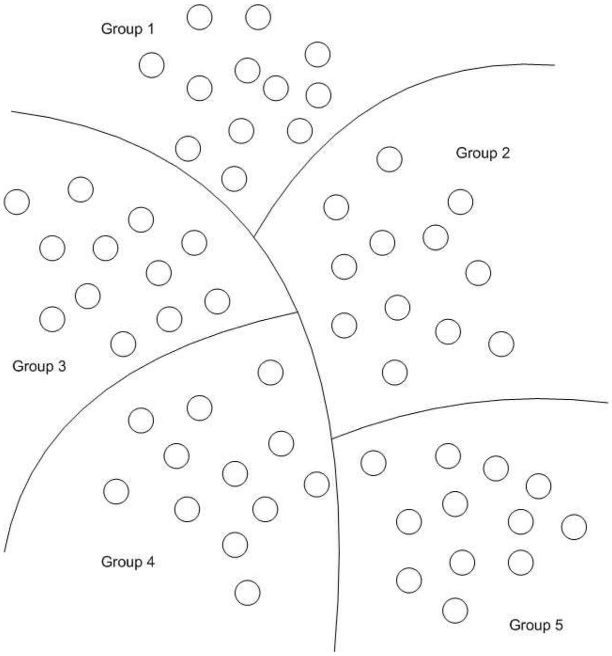


Figure 2: The Network Model

and are assigned with a unique group ID. We adopt the zonal algorithm for clustering ad hoc network [13]in our network model. The zonal distributed algorithm partitions the network into different regions by an asynchronous distributed algorithm for finding minimum spanning tree (MST). The execution of the MST algorithm terminates when the size of components in the tree reaches a value $x$, which is the maximum group size in our network model. Once the network is divided into regions and a spanning tree has been determined for each region. It computes the weakly connected dominating sets of the regions. Finally, it fixes the borders of different regions by including some additional nodes from the borders of the regions. We assume that nodes in the network can know the group, which another belongs to by exchanging messages.

## 3.3 The Trust Model

Authentication in a network usually requires participation of trusted entities. Wireless ad hoc network has no centralized server for trust and key management. We define a fully distributed trust management algorithm to maintain network security. In our trust model, any user can act as a certifying authority. Any node can sign public key certificate of another node in the same group upon request. As mentioned before, we assume a node is able to obtain and store the correct public keys of the same group. Also, a node can observe and give trust value to each of its group members by some monitoring components. We define a trust value as an authentication metric, which represents the assurance with which a requesting node $s$ can obtain the correct public key of a target $t$. We adopts the fully distributed trust management approach, such that each node has a trust table for storing the trust values and public keys of the nodes that they know.

In our authentication service, when a node $s$ wants to obtain the public key of another node $t$. It checks which group node $t$ belongs to. Then, it looks up its trust table to find the first $k$ nodes that belong to the group of node $t$ and with the highest trust values. Node $s$ then selects these $k$ nodes as introducers and sends them request messages on the public key of node $t$. Introducers are the nodes in the same group of the target node $t$ and are trusted by the requesting node $s$. To evaluate the trusts from the recommendation of other reliable entities, relying node should be able to estimate their trustworthiness. Many metrics have been proposed to evaluate the confidence afforded by different paths. In our trust model, we define the authentication metric as a continuous value between 0.0 and 1.0. This authentication metric, or we call trust value is assigned and stored by a node to another in a subjective and localized way. A trust value $V_{i,j}$ represents the level of trust from node $i$ to node $j$. The higher the value represents the more node $i$ trusts node $j$, and vice versa.

Regarding to our network model, we present two types of trust relationships, including the direct trust relationship and recommendation trust relationship as shown in Figure 3. The

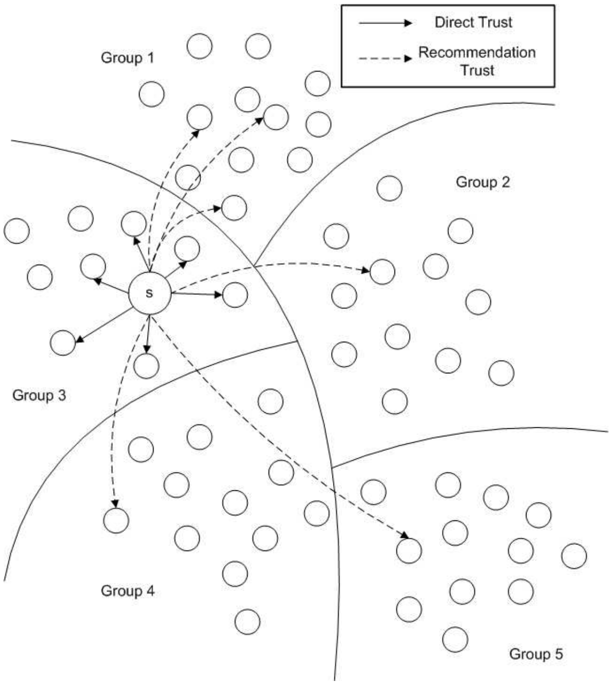


Figure 3: The Trust Model

direct trust relationship is the trust relationship between two nodes in the same group, while the recommendation trust is the trust relationship between nodes of different groups. We apply the formulae for combination of values from the direct trust and recommendation trust approach [9]. From [9], direct trust means to trust an entity directly means to believe in its capabilities with respect to the given trust class. Recommendation trust expresses the belief in the capability of an entity to decide whether another entity is reliable in the given trust class and in its honesty when recommending third entities.

### 3.3.1 Direct Trust

$$\mathrm{P} \xrightarrow{v_d} \mathrm{Q}$$

A direct trust relationship exists if all trust experiences with Q which P knows about are positive experience. It is a value of the trust relationship which is an estimation of the probability that

Q behaves well when being trusted and is based on the number of positive experiences with Q which P knows about. The value vd of these experiences can be computed by:

$$v_d = 1 - \alpha^p \tag{1}$$

It is the probability that Q has a reliability of more than , found on the information P possesses about Q. The reliability is the probability that Q turns out to be reliable when being entrusted with a single task. should be chosen reasonably high to ensure sufficiently safe estimations.

### 3.3.2 Recommendation Trust

$$\text{P} \xrightarrow{v_r} \text{Q}$$

A recommendation trust relationship exists if P is willing to accept reports from Q about experiences with third parties with respect to trust. It represents the portion of offered experiences that P is willing to accept from Q and is based on the experiences P has had with the entities recommended by Q. The recommendation trust value $v_r$ can be computed by:

$$v_r = \begin{cases} 1 - \alpha^{p-n} & if\ p \leq n \\ 0 & else \end{cases} \tag{2}$$

The numbers of positive and negative experiences are represented by p and n, respectively. This value can be regarded as a degree of similarity between P and Q, taking into account that different entities may have different experiences with a third party.

### 3.3.3 Deriving Direct Trust

The first formula computes the trust relationship:

$$V_1 \bigodot V_2 = 1 - (1 - V_2)^{V_1} \tag{3}$$

This formula can be used to calculate value of the new recommendation path. It is a result of the computation of the direct trust values and the semantics of the recommendation values. In our model, a new recommendation path involve a recommendation trust relationship between a relying node and an introducer, and a direct trust relationship between an introducer and a new node. Based on the above relationships, the formula is appropriate for our occasion.

### 3.3.4 Combination of Direct Trust

Another formula combines values of direct trust relationships:

$$V_{com} = 1 - \Pi_{i=1}^{m}(\Pi_{j=1}^{n_i}(1 - V_{i,j}))^{\frac{1}{n_i}} \tag{4}$$

This formula is used for drawing a consistent conclusion when there are several derived trust relationships of same trust class between two entities. This can be applied in our model as well. It is because a relying node asks for multiple introducers, instead of one for signing public key certificates of a new node.

## 4 Security Operations

The authentication protocol we propose takes a certificate-based approach. If a user $i$ believes a given key belongs to a given user $t$, it can issue a public key certificate of $t$. When a node $s$ wants to get the public key of a node $t$, it requests for the public key certificates of node $t$ from some trustable nodes. Node $s$ sends request messages to some nodes that belong to the group of node $t$ and with high trust values in the view of $s$. These nodes which sign the public key certificates of node $t$ are called introducers.

The security operations are divided into two parts, including the public key certification and the trust value update. Figure 4 shows the security operations of a requesting node $s$. When node $s$ wants to obtain the public key of a node $t$, it selects a certain number of nodes that it trusts as introducers. These introducers should be in the same group of node $t$, so they can provide the public key and trust value of node $t$ accurately. Then, node $s$ sends the request of public key certificate to all the selected introducers. After node $s$ collects all the replies, it compares the public key certificates received and concludes the public key of node $t$ with the majority votes. If a malicious introducer providing a false public key certificate of node $t$ is discovered, it will be isolated by reducing its trust value to zero. Finally, trust value of node $t$ will be calculated and inserted into the trust table of node $s$. Details operations on public key certification and trust value update will be presented in the following subsections.

## 4.1 Public Key Certification

Authentication in our network relies on the public key certificates signed by some trustable nodes. Let $s$ be the node requests for public key of a target node $t$. Node $s$ has to ask for public key certificates signed by some introducing nodes, $i_1$, $i_2$, $\ldots$, $i_n$, as shown in Figure 5. Every node is able to request for public key certificates of any other new nodes. However, nodes in the same group are assumed to know each other by means of their monitoring components and the short distances among them. With the above assumptions, we focus on the public key certification where $s$ and $t$ belong to different groups. Nodes which are in the same group with $t$ and have already built up trust relationship with $s$ can be introducers. The requesting node s selects certain number of nodes with the highest trust values as introducers and sends them request messages. The introducers $i_1$, $i_2$ ,$\ldots$, $i_n$, after receiving the messages will reply with the public key of the target node $t$. Apart from the public key of $t$, it includes the trust value of $t$ as well. These values from $i_1$, $i_2$, $\ldots$, $i_n$, will be used for calculating the final trust value of $t$ in $s$ when all the reply messages are received. The reply message should be signed with the introducers' private keys to make the certificate valid.

Table 1 shows the operations of $s$ on obtaining public key certificates of $t$. To request the public key of $t$, $s$ first looks up the group ID $\varphi_t$ of node $t$. Then, it sorts the trust values that belong to $\varphi_t$ and selects the nodes with the highest trust value as introducer $i_1$, $i_2$, $\ldots$, $i_n$ and sends them request messages. After collecting the reply messages encrypted with introducers' secret keys, $s$ decrypts the messages with the corresponding public key. Next, it compares the

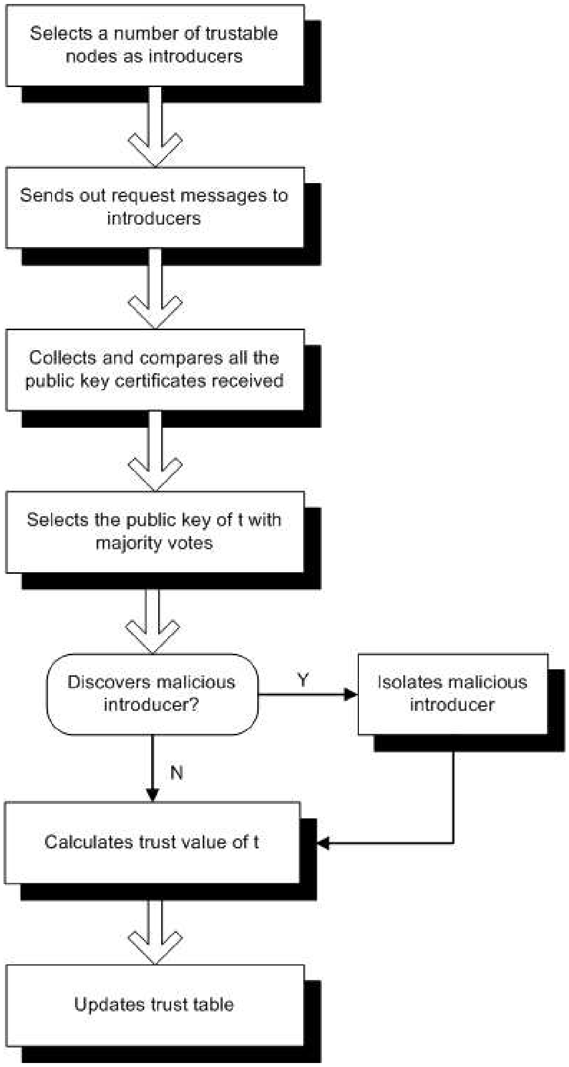

Selects a number of trustable nodes as introducers
Sends out request messages to introducers
Collects and compares all the public key certificates received
Selects the public key of t with majority votes
Discovers malicious introducer?
Y
Isolates malicious introducer
N
Calculates trust value of t
Updates trust table

Figure 4: Security Operations

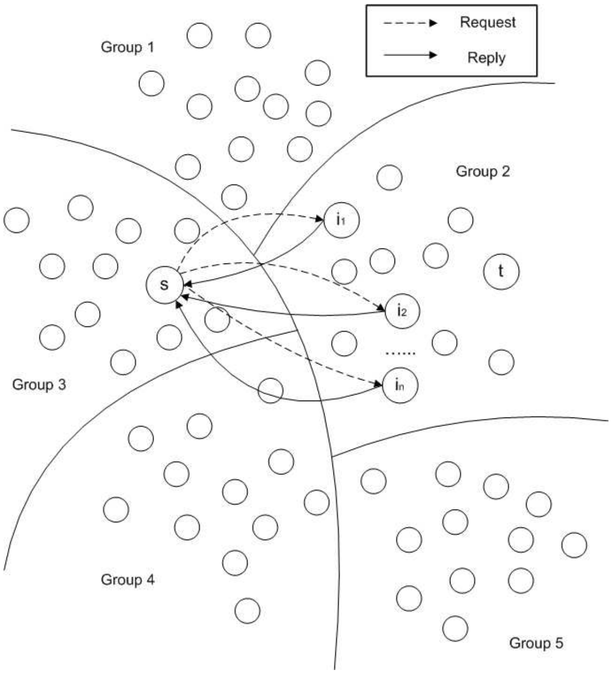


Figure 5: Public Key Certification

public keys obtained from the reply messages and concludes the public key of $t$ as the one with majority votes. It reduces the trust values of the nodes which do not agree with that public key, so to avoid selecting these dishonest nodes as introducers in the future. Finally, $s$ will calculate and update the trust value of $t$, $V_t$.

Table 1: Operations of Node $s$ in Public Key Certification

1. Looks up the group ID of $t$,$\varphi_t$.

2. Sorts the trust values of nodes belonging to group $\varphi_t$ 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 $\varphi_t$.

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 $m$ = $\{Pk_t, V_{i_k,t}, ...\}Sk_{i_k}$. $Pk_t$ 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 concludes 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 this formulae:

$$V_{s,i_k,t} = V_{s,i_k} \bigodot V_{i_k,t} = 1 - (1 - V_{i_k,t})^{V_{s,i_k}} \tag{5}$$

and

$$V_t = 1 - \Pi_{k=1}^{n}(1 - V_{s,i_k,t}) \tag{6}$$

, where $i_k$ denote the nodes in $I_{good}$ and $n$ denotes the number of nodes in $I_{good}$.

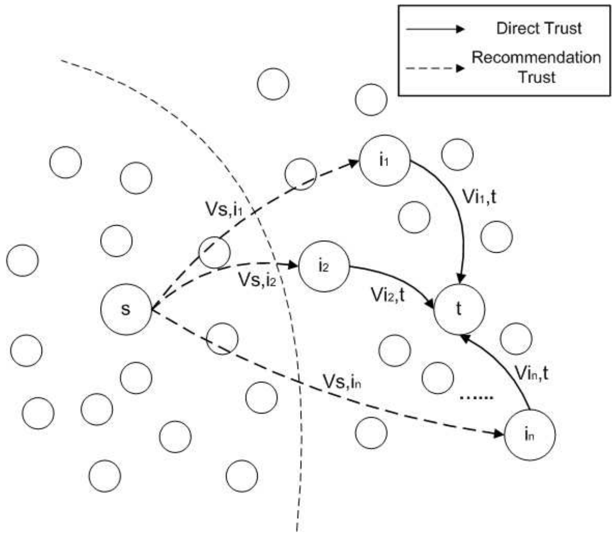


Figure 6: Trust Value Update

## 4.2 Trust Value Update

After collecting and decrypting the reply messages, the relying node obtains the trust values from different introducers $i_k$ to $t$. These values can be used to calculate the ultimate trust value $V_t$ of $t$ in the view of $s$ as shown in Figure 6.

In this figure, $s$ denotes the requesting node; $t$ denotes the target node, whose public key is requested by $s$. Nodes $i_1$, $i_2$,$\ldots$, $i_n$ are the introducers that reply to $s$ with consistent public keys of $t$. $V_{i_1,t}$, $V_{i_2,t}$, $\ldots$,$V_{i_n,t}$ denote trust values from introducers $i_1$, $i_2$,$\ldots$, $i_n$ to $t$; while$V_{i_1,t}$, $V_{i_2,t}$, $\ldots$,$V_{i_n,t}$ denote trust values from $s$ to introducers $i_1$, $i_2$,$\ldots$, $i_n$. Each $V_{s,i*}$ and $V_{i*,t}$ form a pair to make up a single trust path from $s$ to $t$. To compute a new trust relationship from $s$ to $t$ of a single path, we apply the following formula:

$$V_{s,i_k,t} = V_{s,i_k} \bigodot V_{i_k,t} = 1 - (1 - V_{i_k,t})^{V_{s,i_k}} \tag{7}$$

It calculates the new recommendation trust relationship from $s$ to $t$ via an introducer $i_k$. With this formula, we can calculate the three different trust values from $s$ to $t$ via these three introducers on different path separately. The result values are usually different, so one has to find a way to draw a consistent conclusion. Actually, the different values do not imply a contradiction. In contrary, it can be used as collective information to compute a combined value. The following formula can be applied:

$$V_t = 1 - \Pi_{k=1}^{n}(1 - V_{s,i_k,t}) \tag{8}$$

, where $n$ denotes the number of paths.

This formula combines trust values $V_{s,i_k,t}$ of different paths to give the ultimate trust value $V_t$ of $t$. This ultimate trust value $V_t$ represents the trust value of $t$ in the view of $s$ after the public key certification. This value contains information of trust relationships from $s$ to different introducers, and that from introducers to $t$. Finally, this value will be inserted to the trust table of $s$. If $V_t$ is high, it indicates that $t$ can be a possible introducer when $s$ requests for public keys of other nodes that belong to the same group of $t$ in the future.

## 4.3 Special Scenarios

### 4.3.1 Initialization

Initialization When a node first joined into the network, it can only communicate with its neighboring nodes. It broadcasts the joining message to its neighboring nodes and build up intragroup trust relationship with the nodes in the same cluster. Since it is new to the network, it has no experience in communicating with the nodes in other groups. When a new node requests for the public key certificate of nodes in other groups, it collects the information from its group members as intergroup trust relationship has not yet been built up. In the early stage of a node joining into the network, it relies on the intragroup relationship in communicating with the others, including the nodes in different groups. After several communications are made to the nodes in different groups, the new node can build up intergroup trust relationship gradually. Then, the node can rely on intergroup trust relationship for requesting the public key certificates of nodes in different cluster and it use intragroup trust relationship mainly for communication within the local group.

### 4.3.2 Not Enough Introducer

Not enough introducers A node requests for public key certificates of the target nodes that are new to them and in different groups via some introducers. Introducers are the nodes in the same group of the target node and have intergroup trust relationship with the relying node. In some situations, the relying node may find not enough introducers to request for public key certificates of the nodes in other groups. These situations may be at the early stage of a node in the network or a node finds that most of the nodes in another group that it built up intergroup trust relationships become malicious. If there are not enough introducers in the target group, the relying node will choose nodes with high values from its own group to be introducers. It should be noted that a node request public key certificates of the node in another group always find introducers from the target group as the first choice. It finds introducers from its local only if it is unable to find enough number of introducers from the target group. A node chooses introducers from the target group with higher priority than from the local group. It is because nodes in the same group with the target nodes are able to collect more information on the trust of the target node with the relatively shorter distances. In contrary, introducer in the same group with the relying node only provides information of its past communication with the target node and the not up-to-date trust information collected when it requested for the public key certificate of the target node.

## 5 Simulation Results

In this section, we evaluate the performance of the authentication service proposed in terms of security by extensive simulations.

## 5.1 Simulation Set-Up

We implemented our design in network simulator Glomosim [38].Our main objective in the security evaluation is to investigate whether our authentication service provides effective measurement results in public key certifications with the presence of malicious nodes. We imitate the malicious nodes by selecting certain percentage of the nodes in the network randomly and assign them to reply with false public key certificates. A false public key certificate may contain an incorrect public key and trust value of the target node.

The base settings that apply for most of the experiments are summarized in Table 2. The settings represent a wireless ad hoc network with the size of $600mX600m$. It contains $100$ nodes and is divided into $5$ groups. The number of introducers per request is three. A certain percentage of nodes $p$ is regarded as trustable at initialization and certain percentage of nodes $m$ becomes malicious when the simulation begins. We are particularly interested in the successful rate, fail rate, unreachable rate, and type I and type II error rate in our protocol. We vary different parameter in each of the experiment, including the percentage of trustable nodes at initialization, percentage of malicious nodes, and the mobility of the nodes. In the last experiment, we compare the successful rate, fail rate, and unreachable rate between our protocol and the PGP approach with distributed certificate repository. Yet, our experiments indicate that our scheme works well even in a hostile environment.

## 5.2 Evaluation on Ratings to Malicious Nodes

In this experiment, we evaluate the successful rate, fail rate, unreachable rate, false-positive error rate, and false-negative error rate of the authentication service proposed. Successful rate is the percentage of public key requests that lead to a conclusion of the new node’s public key. Fail rate is the percentage of public key requests that are unable to make a conclusion of the new node’s public key or the conclusion drawn is incorrect. Unreachable rate is the percentage of public key requests that are unable to be sent out or the requests have no reply. A request unable

Table 2: Simulation Parameters

| | |
|---|---|
| $Network$ | |
| 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 |

to be sent out may due to no trustable introducer is available, or the request messages cannot reach the introducers. It is also possible that the request messages are sent, but the messages are dropped or unreachable to the requesting node in the reply.

Apart from the successful rate, fail rate, and unreachable rate discussed above, we also carry out the Type I and Type II error tests. We evaluate the false-negative error rate on identifying malicious nodes in the Type I error test and the false-positive error rate in identifying malicious nodes in the Type II error test. In the authentication service we propose, nodes requesting for the public key of a new node compare the public key certificates it received from introducers and try to make a conclusion by the majority votes. If it discovers certain replies of the public key are different from that of the majority, then it suspects the nodes as malicious and lowers their trust values. With this voting algorithm, it is possible for it to incorrectly identify trustable nodes as malicious. We assume that the malicious nodes are not forming malicious peer in the network, so they have low probability to reply with a consistent false public key in the certificates. The following examples illustrate how false-positive and false-negative errors may occur, where “O” indicates a certificate replied by a good node and “X” indicates a certificate replied by a malicious node:

Examples of false-positive error:

**“O X”** Two public key certificates are received from the replies and they are different from each other. The relying node can make no conclusion on the new node’s public key in this case and it concludes that either both or any one of the replies are come from malicious nodes. To put the authentication service in the safest place, it lowers the trust values of both nodes to avoid any malicious node to be selected as introducers in the future. If one of the reply nodes is indeed trustable in this situation, then a false-positive error occurs as it falsely suggests that a node as malicious which it is actually not.

**“O X X”** Similar situation occurs when three different public key certificates are received in the replies. The requesting node can make no conclusion on the new node’s public key again in this case and it concludes that either all or any one of the replies are come from malicious nodes. To keep the network safe, it lowers the trust values of all the nodes to avoid any malicious to be selected as introducers in the future. If one of the introducer is actually a good node, then a false-positive error occurs again in this case.

Not only false-positive errors may occur in the system, but false-negative errors also. The following example shows how false-negative error that may occur in public key certification:

Example of false-negative error:

**“X”** The relying node receives only one reply message, so it has no chance to make comparison and conclude the new node’s public key by majority votes. In this situation, the relying node may believe the reply is trustable as there is no evidence showing inconsistency of the received public key. It may assume this public key certificate is correct to allow its communication with the new node. Unfortunately, if the replying node is indeed malicious, then a false-negative error occurs.

Figure 7 shows the successful rate, fail rate, unreachable rate, false-positive error rate, and false-negative error rate in the authentication service we propose with the percentage of malicious nodes varies from 0% to 100%. The percentage of trustable at initialization is fixed at 40% in Figure 7a and at 70% in Figure 7b respectively. In both figures, the successful rate drops with the percentage of malicious nodes increases. It is because more false public key certificates are received with the increased number of malicious nodes in the network. With the above reason, it is hard for the requesting node to draw a conclusion on the public key of the new node, so the successful rate decreases. The fail rate on the hand increases gradually with the percentage of malicious nodes. It has the same reason as the drop of the successful rate. The unreachable rate increases dramatically with the percentage of malicious nodes. It is due to large amount of nodes initially trustable becomes malicious in the network. These malicious nodes can no longer be introducers upon being discovered and isolated. Some requesting nodes may not be able to contact any introducer as none of them remains trustable on its list, so public key certificate requests cannot be sent.

From the above figures, we can observer that false-positive error rate and false-negative error rate increase with the percentage of malicious nodes as well. The false-positive error rate of both graphs begin from zero and rise gradually from 30% to 70% and then drops to zero gradually afterwards. In our experiment, the number of introducers is three, which means a relying node sends request messages to three introducers in each public key request. The rise and drop of the false-positive rate is related to the probability of having the two cases of false-positive errors (“OX” and “OXX”) from the replies. The false-negative rate rises as there is a higher probability to receive only a single reply from a malicious node when the percentage of malicious nodes increases. The reason is that the higher the percentage of malicious nodes leads to smaller number of trustable introducers left in the network, so a node has a higher chance to find only one introducer to sign valid public key certificate. However, this remaining introducer also has a higher probability to be a malicious node.

In comparing the two figures, we find that Figure 7a has a lower successful rate, higher unreachable rate, lower failure rate, and lower false-positive rate than Figure 7b. The lower successful rate and the higher unreachable rate in Figure 7a are because of the less trustable introducers are available in public key certification with the face that the percentage of trustable node at initialization in Figure 7a is much lower than that of Figure 7b. The lower failure of Figure 7a is due to smaller number of malicious nodes has to be discovered. Since only trustable nodes will be selected as introducers, the higher the percentage of trustable nodes at initialization leads to the greater number of malicious nodes have to be discovered to avoid false public key certification. The malicious node discovering algorithm is based on majority voting in our authentication service. Normally, the more public key certificate request made, the higher number of malicious nodes can be identified. In this experiment, both figures run for 80 cycles and the experiment results are the average of each rating during the whole simulation. It is reasonable that Figure 7b receive more false certificates than Figure 7a, so it has higher failure rate and false-positive rate than Figure 7a.

## 5.3 Evaluation on Ratings to Trustable Nodes at Initialization

Similar to the above experiment, the successful rate, fail rate, unreachable rate, false-positive error rate, and false-negative error rate are evaluated. However, we fix the percentage of malicious nodes and vary the percentage of trustable nodes at initialization in this experiment. We set the percentage of malicious nodes at 40% in Figure 8a and at 70% in Figure 8b, then vary

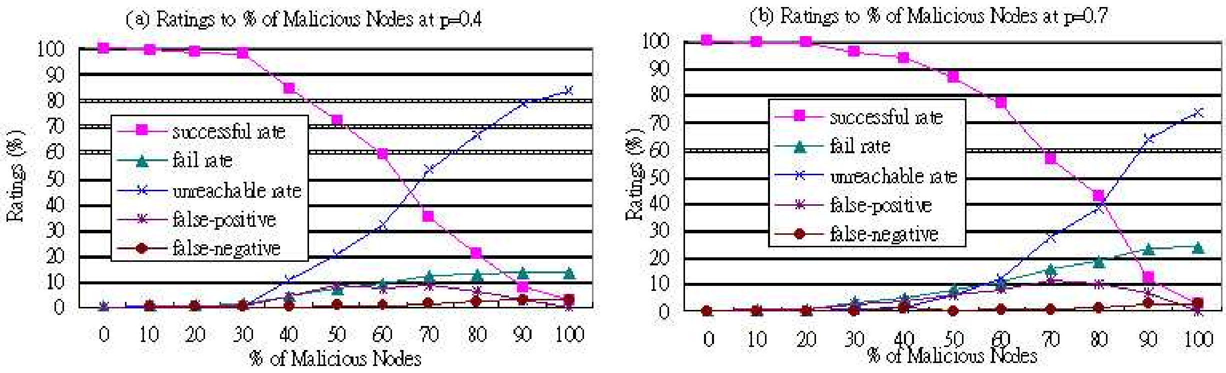


Figure 7: Ratings to Percentage of Malicious Nodes

the percentage of trustable nodes at initialization from 0% to 100%. Both figures show the successful rate increases and the unreachable rate decreases with the increase on the percentage of trustable nodes at initialization. It is because greater number of nodes can be selected as introducers for public key certifications if there is more trustable nodes at initialization. The increase of fail rate is due to more number of malicious nodes need to be discovered as greater number of nodes appear to be trustable initially become malicious later.

In comparing the two figures, Figure 8a has a higher successful rate, lower fail rate, unreachable rate, false-positive rate, and false-negative rate in compare with Figure 8b. The performance in terms of security of Figure 8a is better than that of Figure 8b overall. It is reasonable that a network with lower percentage of malicious nodes to be more secure in public key authentication.

## 5.4 Evaluation on Convergence Time

We investigate the convergence time of our authentication service in this experiment. Again, the same ratings, including the successful rate, fail rate, unreachable rate, false-positive error

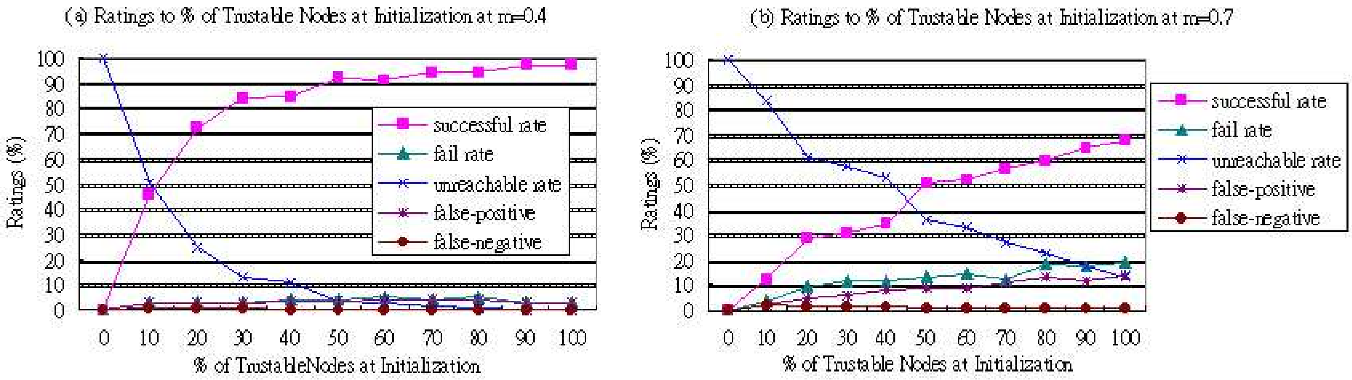


Figure 8: Ratings to Percentages of Trustable Nodes at Initialization

rate, and false-negative error rate are evaluated. We plot different ratings every five cycles in this experiment to get the time of convergence. At the time of convergence, all the ratings are expected to become steady. Since malicious nodes are assigned randomly at the beginning of the experiment, malicious nodes will be discovered gradually and the ratings will vary during this period of time. The convergence time represents the moment that most of the malicious nodes in the network are discovered, so all the ratings become steady upon it.

In Figure 9a, the percentage of malicious nodes and the percentage of trustable nodes at initialization are both fixed at 40%. From the experiment results, we observe that each rating converges to a certain limit value s after certain number of cycles. For example, it shows that the successful rate converges to around 85.4%, fail rate converges to 0.6%, unreachable rate converges to 14%, false-positive rate converges to 0.6%, and false-negative rate converges to 0% in Figure 9a. We define $n$ as the number of cycles, s as the limit value, $x$ as one of the rating at certain cycle. There exists a positive integer $N$ such that when $n > N$, we have $|x_n - s| < \xi$. If we set $\xi$ to be 2% for the successful rate $x_n$, $N$ is equal to 25 in Figure 9a, while $N$ is equal to 30 in Figure 9b.

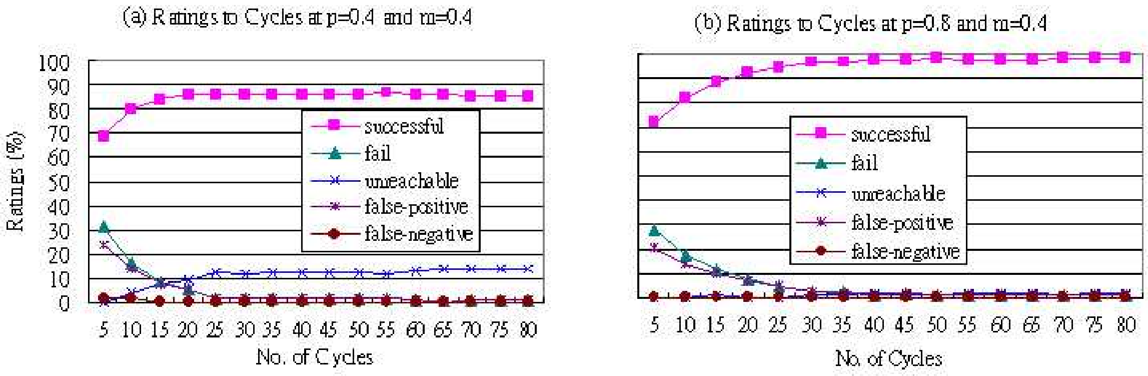


Figure 9: Ratings to No. of Cycles

## 5.5 Evaluation on Ratings to Mobility

In this experiment, we investigate the influence of mobility to the authentication protocol we propose. Throughout all simulations, a relying node sends out public key certificate request to three introducers and gets back their replies. The network size is 600m x 600m with 100 nodes, which allow most of the request and reply messages to reach their destinations. Figure 10 shows the distribution of the ratings under different mobility of nodes. It evaluates the successful rate, fail rate, unreachable rate, false-positive error rate, and false-negative error rate with the percentage of trustable nodes at initialization to be fixed at 40% and the percentage of malicious nodes to be fixed at 60%. We vary the mobility of nodes by setting the maximum speeds of nodes at 0m/s, 5m/s, 10m/s, 15m/s, and 20m/s respectively. The authentication service we propose maintains almost constant distribution under different mobility conditions as shown in Figure 10. Since the network size is not large in compare with the number of nodes, the transmission range of a node normally can cover any of its neighboring nodes. Similar result showing the mobility independent with the successful rate has been appeared in another paper. This paper employed a simple flooding protocol to implement a practical key management framework for ad hoc wireless network [37]. It believes independency of the mobility is because

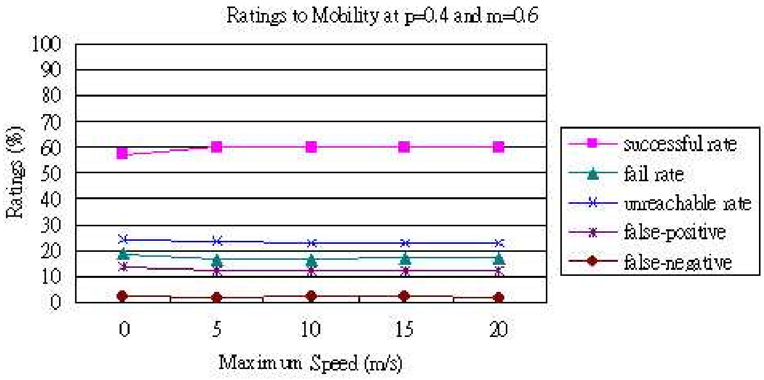


Figure 10: Ratings to Mobility

of the effectiveness of flooding as the reliable data dissemination method.

## 5.6 Comparison with the PGP Approach

In this sub-section, we compare our authentication service with the web of trust model in Pretty Good Privacy (PGP). We judge against their performance in protecting network security during public key certification. A fully self-organizing public key management system using certificate graph, which is similar to PGP, was proposed in ad hoc wireless network [11]. It proposed an algorithm for the construction of the local certificate repositories to help users to find certificate chains to each other in their merged repository. The certificates of this approach are stored and distributed by the nodes and unlike in PGP, where certificates are stored in centralized. It implies that the web of trust model of PGP is applicable to wireless ad hoc networks with certain adjustment.

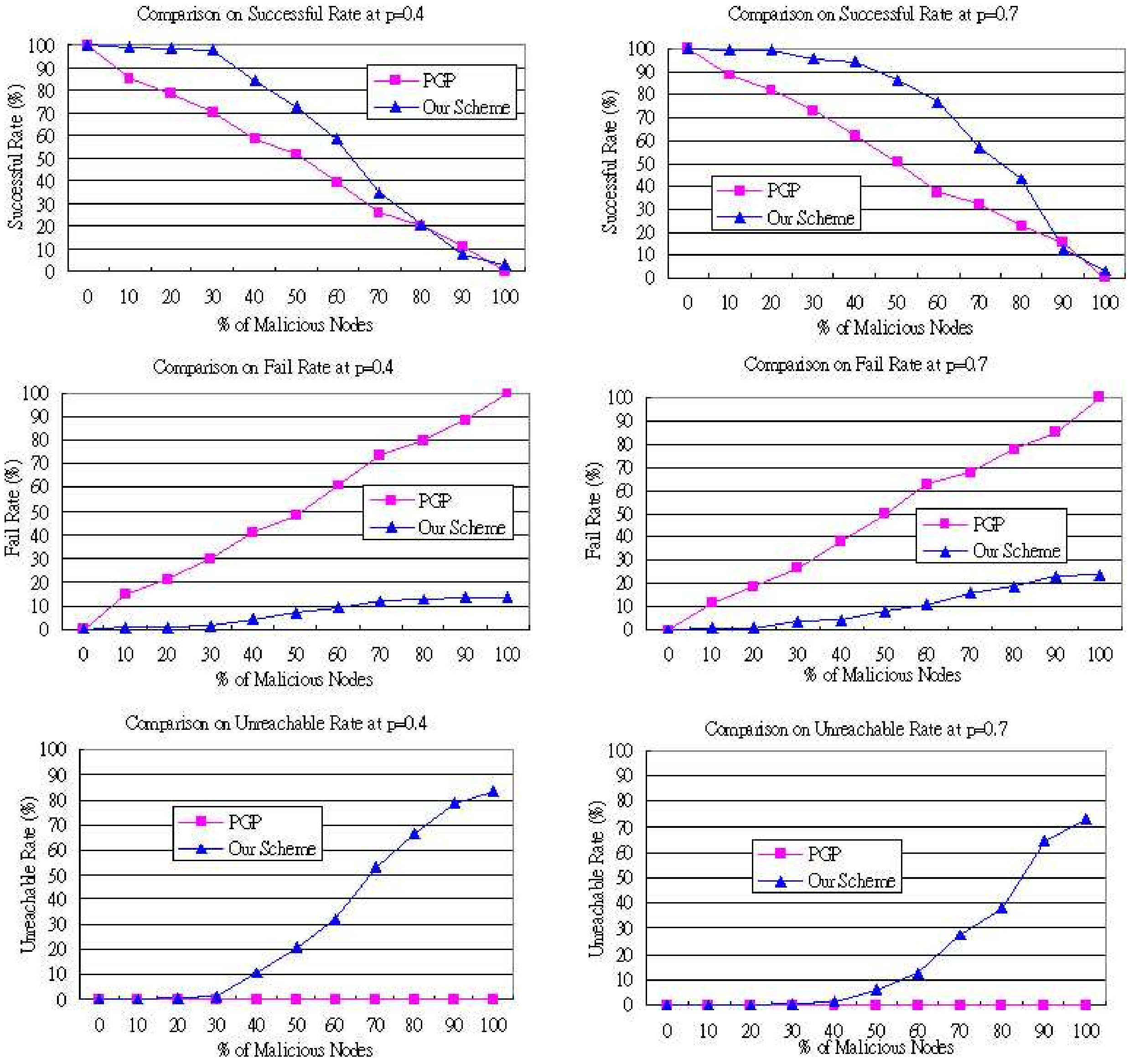
In a PGP environment, any user can act as a certifying authority. A PGP user validates another PGP user's public key certificate if the relying party recognizes the validator as a trusted introducer. Usually, a keyring stores the validity of a particular key and the level of trust it placed on the key that the key's owner can serve as certifier of other's key. There are three levels of validity in PGP, including Valid, Marginally Valid, and Invalid. PGP requires one Completely trusted signatures or two Marginally trusted signature to establish a key as valid. Although PGP involves a trust model with three levels of trust and three levels of validity in public key certification, it does not have any measurement in handling malicious nodes that issue false certificates. It assumes that the public key certificate and the level of trust of a node are valid during its validity period, but this does not reflect the reality. It is because attackers may compromise a node suddenly without being discovered, so it is important to protect authentication against malicious nodes. To deal with the problem of false certificates signed by undiscovered malicious nodes, we propose a novel public key authentication approach based on the trust and clustering techniques.

In comparing our trust- and clustering-based approach with the original PGP approach, our approach is different in distributing repository on certificates among all the nodes. In the original PGP approach, it just defines three levels of trust for a node. In our approach, the trust is defined as a continuous value between 0.0 and 1.0. Therefore, a more accurate trust level can be expressed in our approach than in the original PGP approach. Moreover, the original PGP approach relies on a single trust chain with multiple intermediate nodes to acquire the public key certificate of a new node. In our approach, a trust chain only involves on one intermediate node to reduce the probability for obtaining an invalid trust chain, which involves any malicious nodes. The only intermediate node on a trust chain is in the same cluster as the target node. The close distance between the intermediate node and the target node enhance the performance of the monitoring component on the intermediate node. This increases the correctness for the intermediate node to introduce the target node and estimate its trust value. Also, it relies on multiple trust chains instead of single trust chain in our approach. The public key certificates of the target node signed by different introducers will be compared. Certificates different from the majority votes will be identified and the introducer who signs these suspicious certificates will be isolated gradually. The trust values from different introducers on the target node will be gathered and summarized, and finally be updated to the trust table of the relying node. In summary, our approach makes use the behaviour monitoring advantage and the hierarchical architecture brought by the clustering techniques to develop an authentication procedure that involves multiple trust chains and single intermediate node in each chain. The security is further enhanced by the idea of majority voting and the combination and calculation of continuous trust values among the nodes. It promotes the identification and isolation of malicious nodes, and provides a highly secure public key authentication service in mobile ad hoc network.

The PGP approach we implemented in this experiment distributes certificate repository among all the nodes to fit the characteristics of wireless ad hoc networks. Similar to the fully self-organizing public key management system using certificate graph proposed in [11], a relying node has to look for a certificate chain to perform authentication. It shows from our experiment results that a relying node is able to find a trust chain usually with only one intimidate node. It is probably because the density of nodes in our network is pretty hight. Due to this reason, the PGP approach with distributed certificate repository we implemented is fairly simple as complicate algorithm on finding a trust chain is not required. This experiment focuses on the security evaluation, instead of performance evaluation, between our new authentication protocol and the PGP approach with distributed certificate repository, which is different from the work of the others.

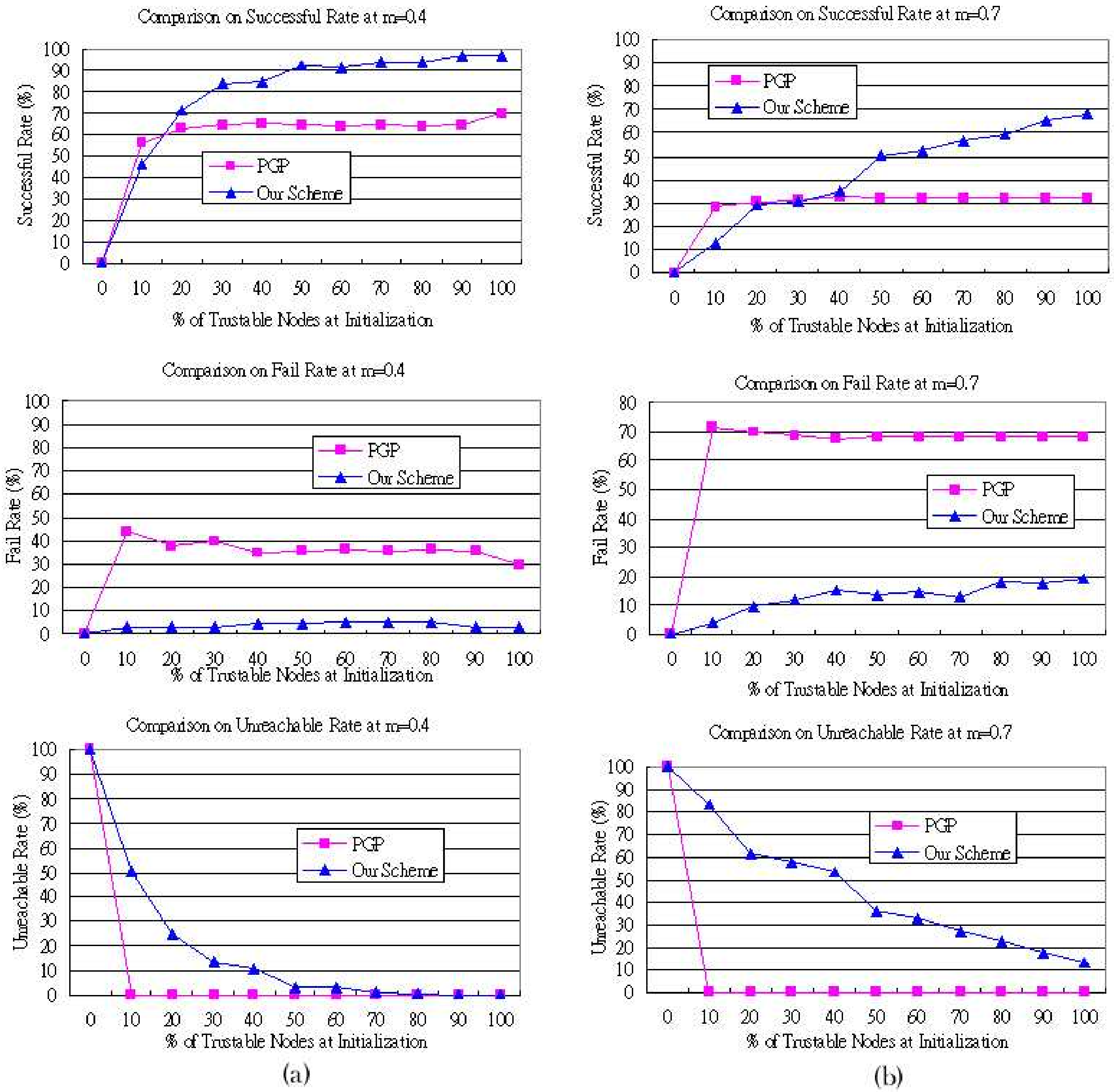
In Figure 11, it shows the successful rate, failure rate, and unreachable rate of our authentication service and the PGP approach. We fix the percentage of trustable nodes at initialization to be 40% and 70% respectively and vary the percentage of malicious nodes $m$ from 0% to 100%. With certain percentage of nodes $p$ is initialized as trustable in the network, a node finds it generally easy to find a valid introducer in PGP. However, there is a probability $m$ for those nodes to become malicious in public key certification. Since there is no mechanism to handling the malicious nodes in PGP, it has a pretty high fail rate in public key certification especially when the percentage of malicious nodes is high. The rise of the fail rate in PGP leads to the drop of its successful rate when the percentage of malicious nodes increases. In contrast, our authentication service has a more sophisticated trust model with a well defined quantitative authentication metric in compare with the PGP approach. Also, its public key certification involves request to multiple introducers, so a relying node is able to identify the malicious nodes by comparing the certificates in the replies. A malicious node in authentication can issue false certificates that are different from the majority. After these malicious are discovered, they will be isolate from public key certification in the future. This leads to the higher successful rate and lower fail rate in our approach than the PGP approach. It should be noted that the unreachable rate of our scheme increase with the percentage of malicious nodes as the increased number of malicious nodes decreases the number of trustable introducers available. However, the unreachable rate keeps zero in the PGP approach as there is no mechanism to detect and isolate malicious nodes during authentication.

Figure 12 shows the same comparison of our approach with the PGP approach as above. The main difference is that it fixes the percentage of malicious nodes instead of the percentage of trustable nodes at initialization in this experiment. The percentage of malicious nodes is fixed at 40% and 70% respectively with the percentage of trustable nodes at initialization varies from 0% to 100%. It shows that our scheme out perform the PGP approach by having a higher successful rate and lower fail rate in average. This is mainly due to the success of our authentication service in identifying and isolating malicious nodes in public key certification as we discussed before. A special phenomenon occurs when the percentage of trustable nodes at initialization $p$ is equal to 10%, we find that the PGP approach performs better than our approach. This may due to the fail rate of the PGP approach keeps at $m$ and its fail rate keeps at $(1 - m)$ constantly upon the percentage of trustable nodes is greater than zero. On the other hand, the malicious introducers are identified in our authentication service, so there may not be enough number of introducers in the network when the percentage of trustable nodes at initialization is only 10%. The increase of unreachable rate leads to the decrease of successful rate in the authentication service we propose subsequently. Though the PGP approach has a higher successful rate when $p$ is equal to 10%, it gives a higher fail rate at the same time that is more harmful than our protocol.



(a) (b)

Figure 11: Comparison Between Our Scheme and PGP with $p$ is Fixed



Comparison on Successful Rate at m=0.4

Comparison on Successful Rate at m=0.7

Comparison on Fail Rate at m=0.4

Comparison on Fail Rate at m=0.7

Comparison on Unreachable Rate at m=0.4

Comparison on Unreachable Rate at m=0.7

(a)

(b)

Figure 12: Comparison Between Our Scheme and PGP with $m$ is Fixed

In this part, we analysis the successful rate and fail rate of the our authentication service and the PGP approach with distributed certificate repository base on the setting of our experiment. In our analysis, the relying nodes under the PGP approach can always find an introducer with Complete trust due to certain percentage of nodes are regarded as trustable at initialization and some of them are assigned with high trust level in our network. We assume that all of the request in the PGP approach are handled by a Complete trust introducer in the following analysis. Let $m_t$ be the percentage of malicious nodes in the set of trustable nodes at certain time $t$. It should be noted that the set size of the trustable nodes may vary with time.

The successful rate of PGP at time $t$ is:

$$1 - m_t \tag{9}$$

The successful rate of the authentication service we propose at time $t$ is:

$$P_1 * (1 - m_t) + P_2 * [C_0^2 * (1 - m_t)^2] + P_3 * [C_0^3 * (1 - m_t)^3 + C_1^3 * m_t * (1 - m_t)^2], \tag{10}$$

where $P_k$ is the probability of receiving $k$ certificate replies, for $1 \leq k \leq 3$.

The fail rate of PGP at time $t$ is:

$$m_t \tag{11}$$

The fail rate of the authentication service we propose at time $t$ is:

$$P_1 * m_t + P_2 * [C_0^2 * m_t^2 + C_1^2 * m_t * (1 - m_t)] + P_3 * [C_3^3 * m_t^3 + C_2^3 * m_t^2 * (1 - m_t)], \tag{12}$$

where $P_k$ is the probability of receiving $k$ certificate replies, for $1 \leq k \leq 3$.

In the PGP approach, this value $m_t$ is equal to the percentage of malicious nodes $m$ that we fix at the beginning of the experiment as it has no algorithm to isolate malicious nodes. However, this value $m_t$ decreases as the number of requests made increases in the authentication service we propose as its security operations help to discover and isolate malicious nodes.

It appears that our authentication service performs better than the PGP approach in protecting network security on public key authentication. Nevertheless, it consumes more network bandwidth and CPU resources than the PGP approach. In the PGP approach, normally only one request and reply message pair are required in the case of involving introducer with Complete trust. Even there is no Complete trust introducer, two Marginally introducers take only two message pairs per request only. In our authentication service, the number of message pairs per request is as same as the number of introducers, $n$. Therefore, it generates more network traffic than PGP.

Message pairs per request in PGP approach
= $P_1 * 1 + P_2 * 2 = O(1)$,
where $P_1$ indicates the probability for having 1 Complete trust introducer and $P_2$ indicates the probability for having 2 Marginally trust introducers.
Message pairs per request in the authentication service we proposed
= $O(n)$

Also, our approach requires the relying node to compares all the certificate replies and conclude with the majority votes, which takes the amount of time $O(n \log n + n) = O(n \log n)$. In addition, the relying node has to calculate the quantitative trust value of the target node and update the trust table, which is $O(n)$. All these operations consume more CPU resources of the relying node than the PGP approach though it seems to be necessary in order to protect the network security.

The CPU cost per request in the authentication service proposed
= $O(n \log n)$

The CPU cost per request in PGP approach
= $O(1)$

Furthermore, the authentication service we propose assume an underlying clustering algorithm in the network. Messages for exchanging grouping information are required among the nodes, which increases the network overhead in the system as well.

# 6 Future Work

In the future, we will have deeper investigation on the clustering techniques in mobile ad hoc networks. We look for better integration between the clustering and the public key authentication mechanisms. Moreover, we will study algorithms for identifying malicious nodes in the network. A trust path only contains one intermediate node in the proposed approach, which can be generalized to involve multiple intermediate nodes. Single intermediate node simplify the process on identifying the malicious node on a trust path. If this restriction is relaxed, more complicated algorithm is needed for identifying the malicious nodes. The identification and isolation on malicious nodes in the network are considered to be essential in providing the security in the network.

# 7 Conclusion

In conclusion, this work aims at providing a secure, scalable and distributed authentication service that assures the correctness of public key certification in wireless ad hoc networks with the presence of malicious nodes. Our system does not rely on any trusted-third party, such that authentication is performed in a distributed manner. New nodes are introduced by other trustable nodes of the same group. Nodes in the network monitor the behavior of each other and update their trust tables accordingly. We suggest a well-defined trust model and a network model to develop our public key authentication services. The trust model allows nodes to monitor and update trust values of each other in a distributed manner. The network model is clustering-based which convenient behavior monitoring and provides high available on public key certification. Based on the above models, we propose a new mechanism to perform public key authentication in wireless ad hoc networks. The security operations proposed include carrying out public key certification and update of trust tables in a novel way. These operations enable a node to discover and isolate malicious nodes who sign false public key certificates. Extensive experiments are completed to evaluate the performance of our authentication protocol in the security perspective. A number of metrics, including the successful rate, fail rate, unreachable rate, type I and type II error rates, and convergence time are evaluated. Parameters like percentage of trustable nodes and percentage of malicious nodes in the network are fixed at different values. In addition, comparison is made between the authentication service we propose and the PGP approach with distributed certificate repository. The experiment results show that our authentication service performs well in protecting the network security in a hostile environment. The approach we propose provides a secure and highly available authentication service in wireless ad hoc network.

# References

[1] "Internet X.509 Public Key Infrastructure," draft-ietf-pkix-roadmap-06.txt, 2002.

[2] "How PGP Works," Chapter 1 of the document Introduction to Cryptography in the PGP 6.5.1 documentation, Copyright ©1990-1999 Network Associates, Inc. and its Affiliated Companies.

[3] A. Abdul-Rahman, "The PGP trust model," *EDI-Forum: the Journal of Electronic Commerce*, April 1997.

[4] A. Abdul-Rahman and S. Halles, "A Distributed Trust Model," *In New Security Paradigms Workshop '97*, pp. 48–60, 1997.

[5] K. Aberer and Z. "Manageing Trust in a Peer-2-Peer Information System," Despotovic Proceedings of the Tenth International Conference on Information and Knowledge Management (CIKM01), 2001.

[6] K. Aberer, “P-Grid:A self-orgainizing access structure for P2P information systems,” Proceeding of the Ninth International Conference on Cooperative Infomation Systems (CoopIS 2001), 2001.

[7] A. D. Amis, R. Prakash, T. H. P. Vuong, and D. T. Huynh, “Max-min D-cluster Formation in Wireless Ad Hoc Network,” *Proceedings of IEEE INFOCOM*, March 2000.

[8] S. Basagni, “Distributed Clustering for Ad Hoc Networks,” *Proceedings of ISPAN’99 International Symposium On Parallel Architectures, Algorithms, and Networks*, pp. 310–315, 1999.

[9] T. Beth, B. Malte, and K. Birgit, “Valuation of Trust in Open Networks,” *Proceedings of the Conference on Computer Security*, Sprintger-Verlag, New York, pp. 3–18, 1994.

[10] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu, and J. Jetcheva, “A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols,” The ACM Annual International Conference on Mobile Computing and Networking, 1998.

[11] S. Capkuny, L. Buttyan, and J.-P. Hubaux, “Self-Organized Public-Key Management for Mobile Ad Hoc Networks,” Swiss Federal Institute of Technology Lausanne (EPFL) Techical Report, June 2002.

[12] S. Capkun, L. Buttyan, and J.-P. Hubaux, “Small Worlds in Security Systems: an Analysis of the PGP Certificate Graph,” Proceedings of the ACM New Security Paradigms Workshop, 2002.

[13] Y. P. Chen and A. L. Liestman, “A Zonal Algorithm for Clustering Ad Hoc Networks,” *International Journal of Foundations of Computer Science*, vol. 14, pp. 305–322, April 2003.

[14] Y. P. Chen and A. L. Liestman, “Approximating Minimum Size Weakly-connected Dominating Sets for Clustering Mobile Ad Hoc Networks,” *The Third ACM International Symposium on Mobile Ad Hoc Networking and Computer (MobiHoc ’02)*, pp. 164–172, June 2002.

[15] S. Corson and J. Macker, “Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations,” Network Working Group rfc 2501, January 1999.

[16] C. Elliott and B. Heile,“Self-Organizing, Self-Healing Wireless Networks,” *Proceedings 2000 IEEE Aerospace Conference*, vol. 1, pp. 149–156, 2000.

[17] W. Ford, “Public-Key Infrastructure Interoperation,” *Proceedings 1998 IEEE Aerospace Conference*, vol. 4, pp. 329–333, 1998.

[18] S. Garfinkel, “PGP: Pretty Good Privacy,” O’Reilly & Associates Inc., USA 1995.

[19] M. Gerla and J. T. C. Tsai, “Multicluster, Mobile, Multimedia Radio Network,” *ACM-Baltzer Journal of Wireless Networks*, vol. 1, no. 3, pp. 255–256, 1995.

[20] L. Gong, “Increasing Availability and Security of an Authentication Service,” *IEEE Journal on Selected Areas in Communications*, vol. 11, no. 5, June 1993.

[21] J-P. Hubaux, L. Buttyan, and S. Capkun, “The Quest for Security in Mobile Ad Hoc Networks,” *Proceedings of the 2001 ACM International Symposium on Mobile ad hoc networking & computing*, Long Beach, CA, USA, pp. 146–155, October 4-5 2001.

[22] Q. Jiang, D. S. Reeves, and P. Ning, “Improving Robustness of PGP Keyrings by Conflict Detection,” RSA-CT Cryptographer’s Track 2004, 2004.

[23] S. D. Kamvar, M, T. Schlosser, and H. G.-Mollina, “The EigenTrust Algorithm for Reputation Management in P2P Networks”, The Twelfth International World Wide Web Conference, Budapest, HUNGARY, 20-24 May 2003.

[24] V. Karpijoki, “Security in Ad Hoc Networks,” Helsinki University of Technology, Tik-110.501 Seminar on Network Security, Telecommunications Software and Multimedia Laboratory, 2000.

[25] S. Kent, “Evaluating Certification Authority Security,” *Proceedings 1998 IEEE Aerospace Conference*, vol. 4, pp. 319–327,1998.

[26] J. Kohl and B. Neuman, “The Kerberos network authentication service (version 5),” RFC-1510, June 1991.

[27] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, “Providing Robust and Ubiquitous Security Support for Mobile Ad-Hoc Networks,” *Proceedings of the 9th International Conference on Network Protocols (ICNP)*, Riverside, California, USA, pp. 251–260, November 11-14 2001.

[28] S. Lee, R. Sherwood, and B. Bhattacharjee, “Cooperative Peer Groups in NICE,” *IEEE Infocom*, April 2003.

[29] J. Liu, X. Zhang, B. Li, Q. Zhang, and W. Zhu, “Distributed Distance Estimation for Large-Scale Networks,” Elsevier Computer Networks, vol. 41, no. 2, pp. 177–193, February 2003.

[30] M. C. Morogan and S. Muftic, “Certificate Management in Ad Hoc Networks”, 2003 Symposium on Applications and the Internet Workshops (SAINT’03 Workshops), 2003 Orlando, Florida January 27 - 31.

[31] M. K. Reiter, “Authentication Metric Analysis and Design,” *ACM Transactions on Information an dSystem Security*, vol. 2, no. 2, May 1999, pp. 138–158.

[32] M. K. Reiter and S. G. Stubblebine, “Resilient Authentication using Path Independence,” *IEEE Transactions on Computers* vol. 47, no. 12, pp. 1351–1362, December 1998.

[33] W. Stallings, “Protect Your Privacy: A Guide for PGP Users,” Prentice-Gall, Inc., Uppder Saddle River, NJ, 1995.

[34] T. Wu, M. Malkin, and D. Boneh, “Building Intrusion Tolerant Applications,” *Eighth USENIX Security Symposium*, pp. 79–92, Washington, D.C., August 23-26 1999.

[35] S. Yi, R. Kravets, “MOCA: Mobile Certificate Authority for Wireless Ad Hoc Networks”, 2nd Annual PKI Research Workshop Program (PKI 03), Gaithersburg, Maryland, April, 2003.

[36] S. Yi and R. Kravets, “Key Management for Heterogeneous Ad Hoc Wireless Networks Department of Computer Science”, University of Illinois, Urbana-Champaign, Technical Report UIUCDCS-R-2002-2290, 2002; Poster Presentation, 10th IEEE International Conference on Network Protocols, 2002.

[37] S. Yi and R. Kravets, “Practical PKI for Ad Hoc Wireless Networks Report No. UIUCDCS-R-2002-2273,” UILU-ENG-2002-1717 August, 2001.

[38] X. Zeng, R. Bagrodia, and M. Gerla, “GloMoSim: a Library for Parallel Simulation of Large-scale Wireless Networks,” *Proceedings of the 12th Workshop on Parallel and Distributed Simulations (PADS ’98)*, Banff, Alberta, Canada, May 26-29 1998.

[39] L. Zhou and Z. J. Hass, “Securing Ad Hoc Networks,” *IEEE Networks Magazine*, vol. 13, issue 6, 1999.

[40] P. Zimmermann, “The Official PGP User’s Guide,” MIT Press, Cambridge, MA, June 1995.