An Authentication Service Against Dishonest Users in Mobile Ad Hoc Networks

Edith C. H. Ngai and Michael R. Lyu
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, NT, Hong Kong
{chngai, lyu}@cse.cuhk.edu.hk

Roland T. Chin
Department of Computer Science
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
roland@ust.hk

Abstract— A mobile ad hoc network is a collection of wireless mobile nodes dynamically forming a temporary network without the use of any existing network infrastructure or centralized administration. It is an emerging technology for civilian and military applications. However, security in mobile 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. Similar to other distributed systems, security in mobile ad hoc networks usually relies on the use of different key management mechanisms. In this paper, we exploit characteristics of an ad hoc network and present our authentication service to protect network security in the presence of dishonest users. Nodes originally trustable in the network may become malicious due to sudden attacks, so an adequate security support for authentication to deal with dishonest users who issue false public key certificates is crucial. We describe a new authentication service with 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 novel public key certificate operation, incorporating with a trust value update algorithm in public key authentication. The authentication service we propose is able to discover and isolate dishonest users in the network. Finally, we evaluate the proposed solution through simulation to demonstrate the effectiveness of the scheme.

TABLE OF CONTENTS

1. INTRODUCTION

A mobile ad hoc network is a collection of nodes that do not rely on a fixed infrastructure. Every node in such a network has sufficient intelligence to continuously sense and discover nearby nodes, and to dynamically determine the optimal path for forwarding data packets from itself hop by hop through the network links to any other nodes in the network [1]. There are a number of differences between mobile ad hoc networks and traditional networks. An ad hoc network relies on wireless communication to keep the network connected. Also, the topology of the ad hoc network is dynamically changing and the nodes of the ad hoc network are often mobile. Due to the above characteristics, a major challenge in the design of mobile ad hoc networks is to protect their vulnerability from security attacks. The security issues must be thoroughly addressed to provide any successful applications [2].

Since an ad hoc network is a network without any infrastructure and centralized control, its operations are usually performed in a fully distributed manner. This means every node is carrying out an equal role and sharing its jobs evenly. From this point of view, we perceive that the "web of trust" approach proposed by Pretty Good Privacy [3][4] is compatible with the characteristics of ad hoc networks in providing security. An approach similar to PGP for security in mobile ad hoc networks is proposed in [5]. That paper presents the idea of the 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 with which a user can obtain the authentic public key of another, it is a metric which is hard to estimate in practice. A node originally trustable to the others may become malicious or dishonest all of a sudden due to the invasion of hackers. The ability to detect such misbehavior and the isolation of malicious nodes are important in public key authentication, because malicious nodes may give false certificates, consequently harming the security of the network. In this paper, we provide a secure authentication service that can defend against dishonest users in the network. The proposed authentication service adopts the certificate-based approach which is able to discover and isolate dishonest users who sign false public key certificates.

We suggest a well-defined trust model and a network model to develop our public key authentication service. Our trust model follows the web of trust model proposed in Pretty Good Privacy [3] and we make several new contributions. Our network model is based on some clustering models [6]

0-7803-8155-6/04/$17.00/
IEEEAC paper #1536

in mobile ad hoc networks, upon which we propose a new mechanism to perform authentication. The work aims at providing a secure, scalable, and distributed authentication service that assures the correctness of public key certification from the attacks of dishonest users in an ad hoc network. The key features of our design are as follows. The system does not rely on any trusted third party. Authentication can be performed in a distributed manner, and new nodes can be introduced by any trustable nodes of the same group. Nodes in the network monitor each other's behavior and update their trust tables accordingly. Our public key management mechanism identifies dishonest users and malicious nodes that issue false certificates, and prevents them from introducing nodes later. These features provide secure and highly available authentication service in the ad hoc network, which is demonstrated through our experimentation.

The remaining of this paper is organized as follows: Section 2 discusses the related work on the current key management systems developed 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 propose the security operations on the public key certification and the update of trust tables. Our new solution is evaluated through simulation and implementation, and the results are presented in Section 5. Finally, we conclude the paper in Section 6.

2. Related Work

Public key certificates employed by applications are created by Certificate Authorities (CAs) that vouch for the binding of various attributes to a public key. Security requirements for CAs are important with an exploration of the wide range of attackers that can be mounted against CAs [7]. Popular network authentication architectures include X.509 standard [8] and Kerberos [9]. Another paper suggests making use of interoperation between many small, independent certificate authorities to build a global-scale public-key infrastructure [10]. However, ad hoc networks are infrastructure-less, and there is no centralized server for key managements. Hence, traditional solutions do not meet the requirements of mobile ad hoc networks. On the other hand, Pretty Good Privacy (PGP) [3][4] 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. Another active research area is security function sharing [11], including a popular method for threshold secret sharing [12]. The basic idea is distributing the functionality of the centralized CA server among a fixed group of servers. Zhou and Hass [13] propose 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. Similar to the partially-distributed CA, the fully-distributed certificate authority proposed by Luo and Lu [14] extends 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. [5]. It issues certificates by users themselves without the involvement of any certificate authority.

As mentioned before, the authentication service we propose is based on a network model and a trust model. A mobile ad hoc network is an infrastructureless network that contains mobile units with a limited transmission range. Mobile hosts communicate with each other by relaying packets from one node to other nodes when their distance is long in comparison with the transmission range. Hierarchical organization of a network is a well-studied subject in many distributed systems. In ad hoc networks, partitioning the nodes into groups or clusters is a similar function. Clustering has been proven effective in minimizing the amount of storage for communication information, and in optimizing the use of network bandwidth. One class of existing clustering algorithms is based on independent dominating sets of graphs. Weight based clustering algorithms, on the other hand, are proposed in [15]. These algorithms define a vertex with an optimal weight within its neighborhood as a clusterhead, and the neighborhood of the clusterhead as a cluster. The weight idea is generalized in [16], 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 [17]. A weakly-connected dominating set approach is proposed for clustering ad hoc networks in [18]. Finally, a zonal algorithm for clustering ad hoc networks is proposed in [6] to divide the network into different regions and make adjustments along the borders of the regions, producing a weakly-connected dominating set of the entire graph.

Our trust model follows a web of trust approach [19], in which any user can act as a certifying authority. The web of trust model is a cumulative trust model such that certificates may be trusted directly, through back-tracking a chain to a directly trusted root certificate, or by a group of introducers. Since our trust model does not have any trusted root certificate, it relies on direct trust and groups of introducers in the certification. This model uses digital signatures as its form of introduction. Any node can sign another node's public key with its own private key to establish a web of trust. Authentication in an ad hoc network without centralized certificate authorities generally depends on a path of trusted intermediaries. To evaluate the trusts from the recommendation of other reliable entities, the relying node should be able to estimate their trustworthiness. Many metrics have been proposed to evaluate the confidence afforded by different paths. One of the proposed metrics represents a set of trust relationship by a directed graph [20]. It introduces the semantics of direct trust values different from those of recommendation trust values. It shows that different values can be combined into a single value by considering the opinions from the respective recommending entities. Another metric is used in PGP, which has three levels of trust, including Complete trust, Marginal trust,

and Notrust [21]. This approach requires one Completely trusted signature or two Marginally trusted signatures to establish a valid key [22]. Another paper explores the use of multiple paths to redundantly authenticate a channel. It focuses on two notions of path independence, the disjoint paths and the connective paths, which seem to increase assurance in the authentication [23]. Finally, a distributed trust model is proposed based on recommendations in [24]. This model uses discrete levels of trust, developing an algorithm for calculating trust and using values in recommendations.

3. MODELS

In this section, we describe the system model of our authentication service in mobile ad hoc networks. We present the architecture of our authentication service, and discuss its network model and trust model in detail.

Architecture

Our authentication service aims at providing secure public key certification despite the presence of dishonest users in the network. Dishonest users can be a node which issues false certificates to the other nodes. To deal with the problem, we design our authentication service as clustering- and trust-based. The clustering-based network model gives advantages on the behavior monitoring among the nodes. The monitoring power of the nodes in mobile ad hoc networks is usually limited to their 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 a trust value as an authentication metric for indicating assurance. The chance for obtaining a correct public key certification increases if the node signs the certificate with a high trust value. The clustering model and the trust value alone are not enough to prohibit dishonest users because a node with a high trust value can still suddenly become malicious when it is attacked. Therefore, we design each public key request on a new node with multiple replies, so that conclusion can be made on the basis of a majority votes. This operation improves the security for obtaining a correct public key and helps to discover dishonest users in the network. The 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. Altogether there are four layers in this architecture, including the mobile hosts, the network model, the trust model, and the security operations. A mobile ad hoc network contains a large amount of mobile hosts, each with a transmission range that is relatively small to the network size. We divide the network into different regions, with nodes in the same region forming a cluster. A cluster, or as we call it, 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 at the highest layer. These operations include public key certification and trust value update, which will be presented in Section 4.

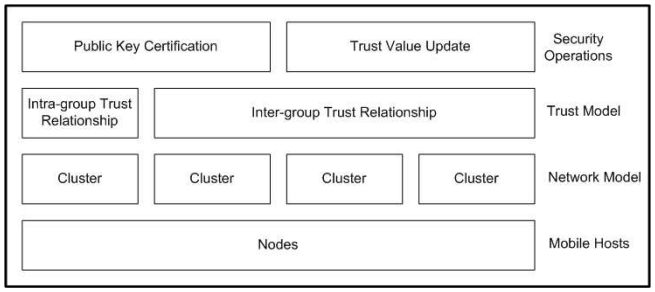

Figure 1. Architecture of Our Authentication Service

The Network Model

As a mobile ad hoc network is an infrastructureless wireless communication network that requires only mobile units to form the network; it can involve a great number of mobile units, each with a transmission range. Each mobile unit can only communicate directly with other subscriber units in the same range. An important feature in the mobile ad hoc network 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 of a network is a well-known and well-studied problem in distributed computing. In the case of ad hoc networks, partitioning the nodes into groups or clusters is equally important. Clustering has been proven effective in minimizing the amount of storage for communication information and in optimizing the use of network bandwidth.

In addition to efficiency, we believe clustering improves the security of a network. In a mobile ad hoc network which lacks a centralized server for management and monitoring, security measure relies on individual nodes to monitor each other. However, the direct monitoring capability is normally limited to the 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 the mobile ad hoc network with the use of trust and clustering techniques.

There are a number of existing solutions for clustering in ad hoc networks. In our design, we divide the network into different regions with a similar number of hosts in each region, as shown in Figure 2. Nodes clustered together in the same region form a group and are assigned a unique group ID. We adopt the zonal algorithm for clustering ad hoc networks [6] in our network model. The zonal distributed algorithm partitions the network into different regions by an asynchronous distributed algorithm for finding a 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 is 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 another node belongs to by exchanging messages.

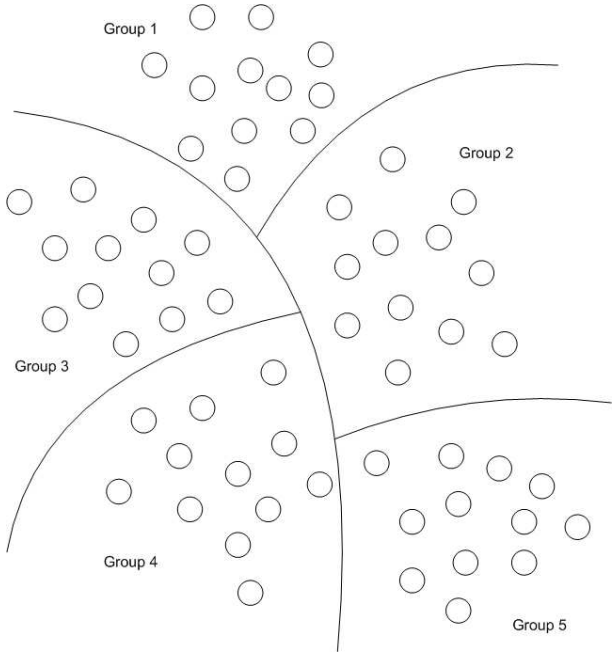

Figure 2. The Network Model

The Trust Model

Authentication in a network usually requires participation of trusted entities. Since a mobile 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 the public key certificate of another node in the same group upon request. As mentioned in the last sub-section, we assume a node can obtain and store the correct public key of the nodes in the same group. Also, a node can observe and give a 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 node t. Each node in the network should have a trust table for storing the trust values and public keys of the nodes that they know in the network.

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 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, a 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 trust value, from one node to another is assigned and stored 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.

With consideration to our network model, we define two types of trust relationship, including the direct trust relationship and the recommendation trust relationship in our trust model, as shown in Figure 3. The direct trust relationship represents the trust relationship between two nodes in the same group, while the recommendation trust represents the trust relationship between nodes of different groups. We apply the formula for combination of values from the direct trust and recommendation trust approach [20].

The first formula computes the trust relationship:

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

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 trust values. In our model, a new recommendation path involves a recommendation trust relationship between a relying node and an introducer, and a direct trust relationship between the introducer and the new node. Based on the above relationships, the formula is appropriate for our occasion.

Another formula combines values of different trust relationships:

$$V_{com} = 1 - \Pi_{i=1}^{m}(\Pi_{j=1}^{n_i}(1 - V_{i,j}))^{\frac{1}{n_i}} \quad (2)$$

This formula is used for drawing a consistent conclusion when there are several derived trust relationships of the 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 only one, for signing public key certificates of a new node.

4. Security Operations

Our authentication service takes a certificate-based approach. If user i believes a given key belongs to user t, it can issue a public key certificate t. When node s wants to get the public key of node t, it requests the public key certification of node t from some trustworthy nodes. Node s sends a request messages to some nodes that belong to the group of node t with high trust values in s's view. These nodes which sign the public key certificates of node t are called introducers.

The security operations are divided into two parts, including

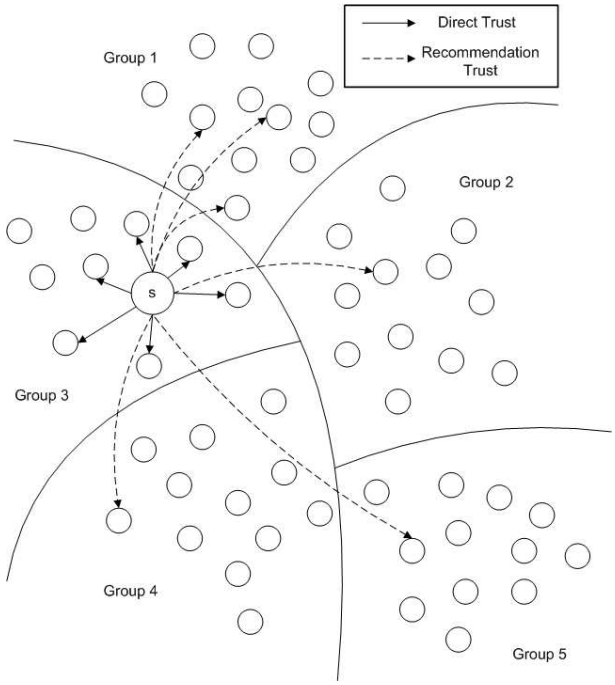

Figure 3. The Trust Model

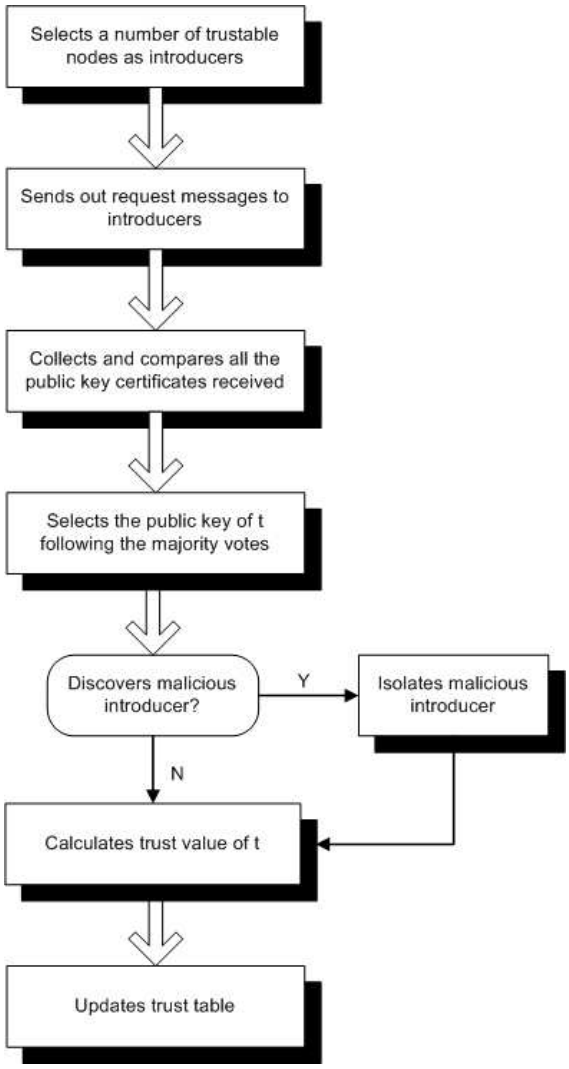

Figure 4. Security Operations

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 node t, it selects a certain number of nodes that it trusts as the introducers. These introducers should be in the same group of node t, so they can provide the public key and the trust value of node t accurately. Then, node s sends the request for the public key certificate to all the selected introducers. After node s collects all the replies, it compares the received public key certificates and establishes the public key of node t following the majority votes. If a malicious introducer providing a false public key certificate of node t is discovered, it will be isolated by the reduction of its trust value to zero. Finally, the trust value of node t will be calculated and inserted into the trust table of node s. Detailed operations on public key certification and trust value update will be presented in the following subsections.

Public Key Certification

Authentication in our network relies on the public key certificates signed by some trustable nodes. Let s be the node requesting a 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, ..., i_n, as shown in Figure 5. Every node is able to request 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 a trust relationship with s can be the introducers. The requesting node s selects a certain number of nodes with the highest trust values as introducers and sends them request messages. The introducers i_1, i_2 ,..., i_n, after receiving the messages, will reply with the public key of the target node t. In addition to the public key of t, the trust values of t are included as well. These values from i_1, i_2, ..., i_n will be used for calculating the final trust value of t in s when all the reply messages are received. The reply messages should be signed with the introducers' private keys to make the certificates 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 φ_t of node t. Then, it sorts the trust values that belong to φ_t and selects the nodes with the highest trust values as the introducers i_1, i_2, ..., i_n and sends them request messages. After collecting the reply messages encrypted by introducers' secret keys, s decrypts the messages with the corresponding public keys. Next, it compares the public keys obtained from the reply messages and concludes the public key of t as the one with the majority votes. It reduces the trust

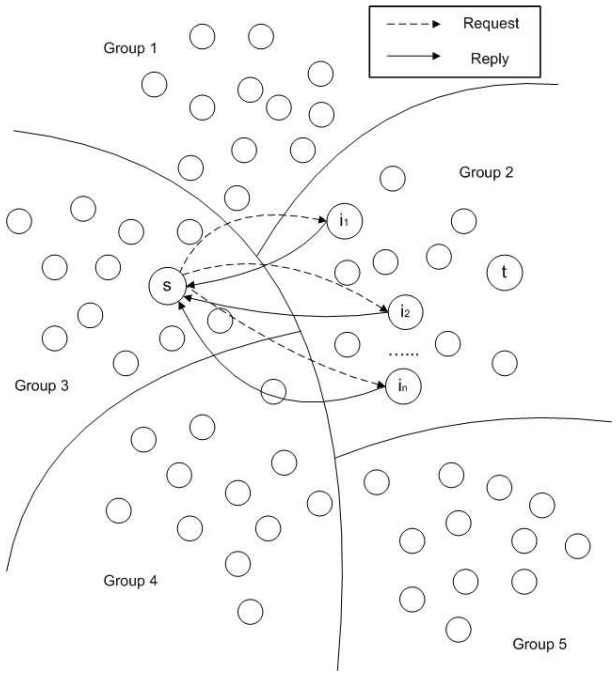

Figure 5. Public Key Certification

values of the nodes which do not agree with that public key, so as to avoid selecting these assumed dishonest nodes as introducers in the future. Finally, s calculates and updates the trust value of t, V_t.

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.

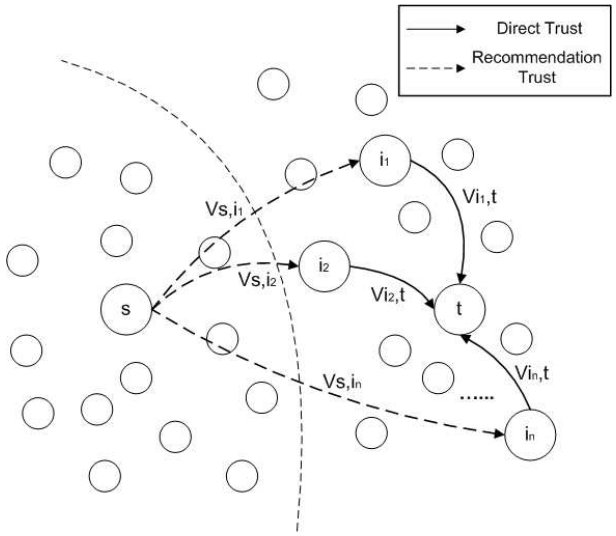

Figure 6. Trust Value Update

In this figure, s denotes the requesting node, and t denotes the target node whose public key is requested by s. Nodes i_1, i_2, ..., i_n are the introducers that reply to s with consistent public keys of t. V_{s,i_1}, V_{s,i_2}, ..., V_{s,i_n} denote trust values from s to the introducers i_1, i_2, ..., i_n, while $V_{i_1,t}$, $V_{i_2,t}$, ..., $V_{i_n,t}$ denote trust values from the introducers i_1, i_2, ..., i_n to t. 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:

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

This calculates the new recommendation trust relationship from s to t via the introducer i_k. With this formula, we can calculate the n different trust values from s to t via these n introducers on different paths. 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. On the contrary, they can be used as collective information to compute a combined value. The following formula can thus be applied:

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

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 from

Table 1. Operations of Node s in Public Key Certification

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 $m = \{Pk_t, V_{i_k,t}, ...\}Sk_{i_k}$. 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 follows the majority votes. Let $i_{good} \in I_{good}$ and $i_{bad} \in I_{bad}$, where i_{good} are the nodes thought to be honest (that agree on Pk_t with the majority) and i_{bad} are the remaining nodes that are 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 these formulae: $V_{s,i_k,t} = V_{s,i_k} \bigodot V_{i_k,t} = 1 - (1 - V_{i_k,t})^{V_{s,i_k}}$ (3) and $V_t = 1 - \Pi_{k=1}^{n}(1 - V_{s,i_k,t})$, (4) where i_k denotes the nodes in I_{good} and n denotes the number of nodes in I_{good}.

these 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 public keys for other nodes that belong to the same group as t in the future.

5. Simulation Results

We implemented our design in the network simulator Glomosim [25]. We evaluate the performance of our system in suppressing false public keys in the replies. The simulation is used to evaluate the successful rate, failure rate and unreachable rate on the requests of public key certificates. The successful rate is the percentage of public key requests that are made in which the relying nodes are able to establish correct public keys from the replies. The failure rate is the percentage of public key requests that are made where the relying nodes establish false public keys from the replies. The unreachable rate is the percentage of requests that cannot be made successfully because no introducer can be found in the network.

We simulate a network that contains 100 nodes divided into 5 groups. Table 2 details the parameters used in our simulations. At initialization, the network is assigned a certain percentage p of trustable nodes and a certain percentage m of malicious nodes. The maximum number of introducers selected in each request is 3. At least one introducer should give a valid reply in a successful public key certification. The simulation runs for 45000 sec. and a total of 4000 public key requests are sent out from different nodes.

Table 2. Simulation Parameters

Network	
No. of nodes	100
No. of groups	5
% of trustable nodes at initialization	p
% of malicious nodes	m
Mobility	0-10m/s
Public Key Request	
Max. no. of introducers for each request	3
Min. no. of replies for each request	1
Simulation	
Time	45000s
No. of query cycles	40
No. of requests per cycle	100

Evaluation on Ratings to Periods of Time

Figure 7 shows the successful rate, failure rate, and unreachable rate on requests for public keys with the percentage of malicious nodes at initialization fixed at 70% and the percentage of trustable nodes at initialization fixed at 40%. The ratings are shown in four different periods of time to demonstrate the changes. Each period of time involves 1000 public key certificate requests. After the first 1000 requests, the successful rate is the lowest and the failure rate is the highest among the four periods. This is mainly because 70% of nodes become malicious in the network initially, replying with false public key certificates, but they are not yet discovered. After a certain number of requests, more nodes discover their misbehavior and lower their trust values. These malicious nodes have a much lower chance of being selected as introducers afterwards, so the successful rate rises and the failure rate drops in the following periods. Finally, all the ratings keep steady after most of the malicious nodes are discovered.

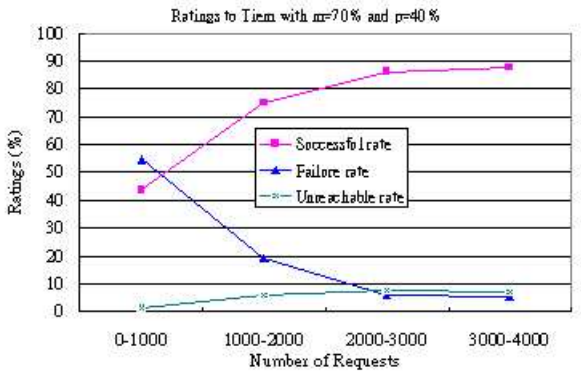

Figure 7. Ratings to Number of Requests with p=40% and m=70%

Evaluation on Ratings to Malicious Nodes

In these experiments, we evaluate different ratings relative to the percentage of malicious nodes in the network with the percentage of trustable nodes at initialization fixed at 30% and 60% respectively. Figure 8 and Figure 9 show the successful rate, failure rate, and unreachable rate in the network with the percentage of malicious nodes varies from 0% to 100%.

We find that the successful rate is high at the beginning and remains at over 50% until the percentage of malicious nodes reaches 80%. The failure rate keeps at a quite low level even when the percentage of malicious nodes in the network is high. However, the unreachable rate can be pretty high especially when there are a lot of malicious nodes in the network. The high unreachable rate is due to the fact that most of the malicious nodes are identified, so the requesting nodes cannot find any reliable introducers from which to request public key certificates. In comparing the two figures, the performance of Figure 9 is better than that of Figure 8, showing a relatively higher successful rate and a lower unreachable rate. This is because a larger set of introducers can be selected with a higher number of trustable nodes at initialization in Figure 9. However, the failure rate in Figure 9 is higher than that of Figure 8.

Evaluation on Ratings to Trustable Nodes at Initialization

In this experiment, we compare the relationship among different ratings to the percentage of trustable nodes at initializa-

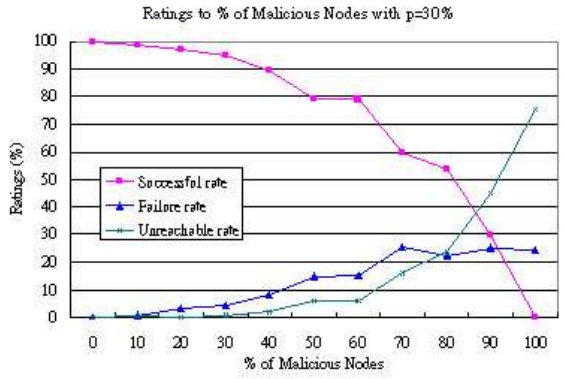

Figure 8. Ratings to % of Malicious Nodes with p=30%

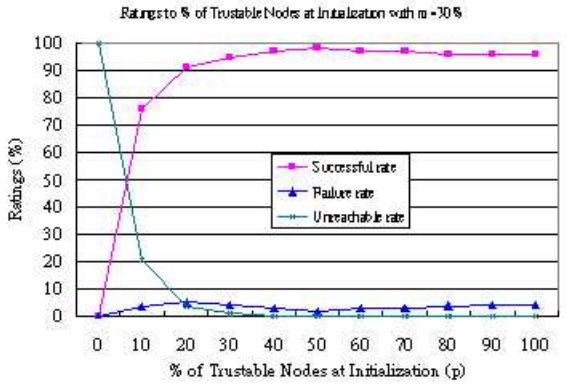

Figure 10. Ratings to % of Trustable Nodes with m=30%

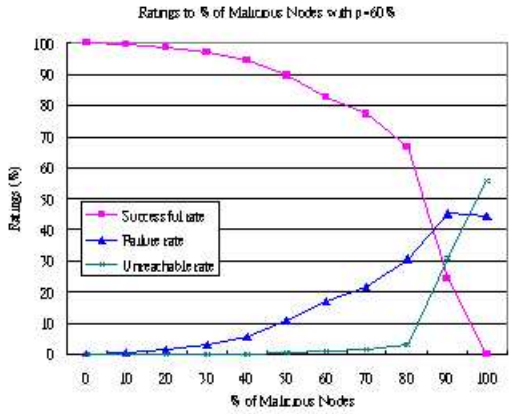

Figure 9. Ratings to % of Malicious Nodes with p=60%

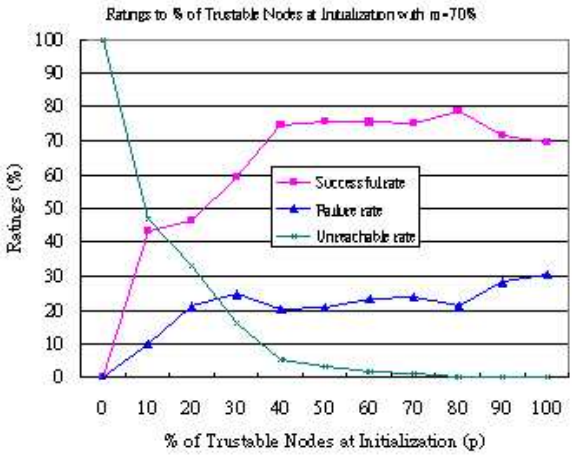

Figure 11. Ratings to % of Trustable Nodes with m=70%

tion with the percentage of malicious nodes fixed at 30% and 70% respectively. Figure 10 and Figure 11 show the successful rate, failure rate, and unreachable rate with the percentage of trustable nodes varying from 0% to 100% in the network. From these figures, we observe that the successful rate rises with the increase in the percentage of trustable nodes at initialization. The increase in the successful rate is mainly due to the increased number of trustable nodes that can be selected as introducers. Similarly the unreachable rate drops with the increase in the percentage of trustable nodes at initialization. Both figures show that the successful rate remains steady after the percentage of trustable nodes at initialization reaches 40%. This implies that nodes are able to find enough number of trustable introducers to request public key certificates after p reaches a certain value.

The successful rate in Figure 10 is higher than that in Figure 11 because it contains a lower percentage of malicious nodes. The network has a higher chance of receiving false public key certificates from dishonest introducers with a high percentage of malicious nodes. This also explains the failure rate in Figure 11 is higher than that in Figure 10.

Comparison with the PGP Approach

In this experiment, we compare different ratings of the authentication service we proposed with the Pretty Good Privacy (PGP) approach. Again, these ratings include the successful rate, failure rate, and unreachable rate. We fix the number of trustable nodes at initialization at 60% and vary the percentage of malicious nodes from 0% to 100%. We try to compare our authentication service with the approach applying Pretty Good Privacy [22] in the ad hoc network. In the PGP approach, a user u verifies the public key of another user v by finding a certificate chain from u to v in their local certificate repository.

Figure 12 compares the successful rates between the two mechanisms. In the PGP approach, a node finds a trust path for public key certification. If any nodes on the trust path are dishonest, the relying node will get a false public key certificate in the reply. If our authentication service is applied, a node selects introducers only if the nodes are with high trust values. These nodes usually give high assurance on the public key and trust value of the target node. Also, dishonest

users who issue false public key certificates can always be discovered and assigned with low trust values by our security operations, and they are rarely selected as introducers. Our trust- and clustering-based authentication service reduces the chance for selecting malicious nodes as introducers in the future. In Figure 12, we find that our authentication service has a much higher successful rate than that of the PGP approach in facing dishonest users.

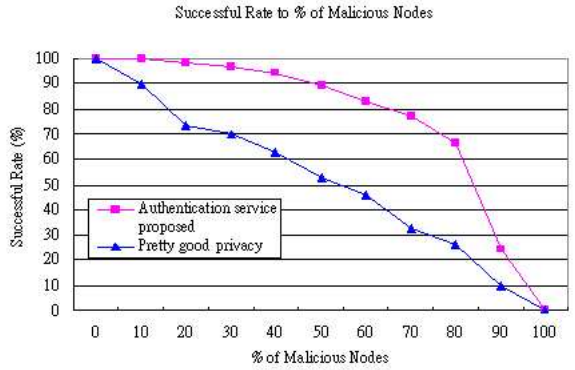

Figure 12. Comparison of Successful Rate to % of Malicious Nodes

Figure 13 compares the failure rate between the two mechanisms as above. In the PGP approach, nodes only select introducers by random, and malicious nodes may succeed in replying false public key certificates. The failure rate is very high with the random algorithm. With the introducer selection algorithm in our authentication service, trust values are updated from time to time to maintain high security in public key authentication. Also, dishonest users reply with false certificates are usually discovered and recorded, so they will not be selected as introducers again. Consequently, the failure rate in our approach is relatively low.

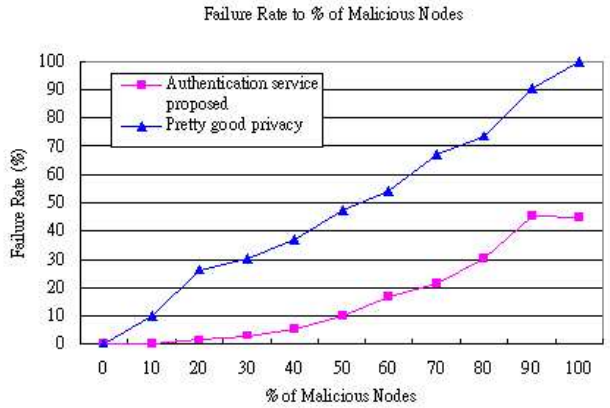

Figure 13. Comparison of Failure Rate to % of Malicious Nodes

Figure 14 compares the unreachable rate among the two mechanisms as above. In the PGP approach, nodes select other nodes as introducers randomly, so the probability for not finding any or enough introducers is extremely low. However, in our authentication mechanisms, there is a probability that all the possible introducers are found to be malicious. In this case, the request messages will not be sent, so the target node is regarded as unreachable.

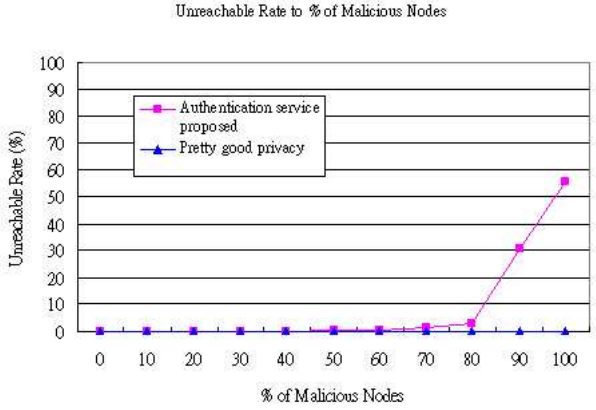

Figure 14. Comparison of Unreachable Rate to % of Malicious Nodes

6. CONCLUSIONS

In this paper, we describe a solution to provide a secure authentication service in mobile ad hoc networks. Our design is motivated by the fact that honest nodes can become malicious due to sudden attacks. Therefore, our design has to work in the presence of dishonest users who sign false public key certificates. The solution has to be secure and fully distributed to operate in a mobile ad hoc network whose nodes are easy to be compromised. To this end, we propose an authentication service based on a trust model with quantitative authentication metrics and a clustering-based network model. Our trust model follows the "web of trust" model with our own contribution. Our clustering-based network model enhances the monitoring power among nodes, so it ensures the correctness in obtaining the public keys and trust values within a cluster. The authentication service that we propose is secure and fully distributed. It adopts the certification approach in public key authentication. We devise a new mechanism in public key certification that is able to discover and isolate dishonest users who signed false certificates. We conduct evaluation to compare our approach with the Pretty Good Privacy approach in defending against malicious nodes in the public key authentication. Our authentication service is shown to be effective in protecting network security from dishonest users in the inherently insecure and unreliable network. It is a secure, fully-distributed, and highly available solution in the mobile ad hoc network.

7. ACKNOWLEDGMENTS

The work described in this paper was fully supported by two grants, RGC Project No. CUHK4182/03E and UGC Project

No. AoE/E-01/99, of the Hong Kong Special Administrative Region, China.

REFERENCES

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

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

[3] S. Garfinkel, "PGP: Pretty Good Privacy," O'Reilly & Associates Inc., USA, 1995.

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

[5] 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.

[6] 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.

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

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

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

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

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

[12] 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.

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

[14] 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.

[15] 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.

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

[17] 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.

[18] T. 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.

[19] "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.

[20] 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.

[21] P. Zimmermann, "The Official PGP User's Guide," MIT Press, Cambridge, MA, June 1995.

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

[23] 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.

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

[25] 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.

Edith Ngai *received her bachelor's degree in Computer Science & Engineering from the Chinese University of Hong Kong in 2002. She is currently an MPhil student in the same department. Her research interest is in the security issues of mobile ad hoc networks.*

Dr. Michael R. Lyu *received the B.S. (1981) in electrical engineering from National Taiwan University, the M.S. (1985) in computer engineering from University of California, Santa Barbara, and the Ph.D. (1988) in computer science from University of California, Los Angeles. He is a Professor in the Computer Science and Engineering Department of the Chinese University of Hong Kong. He worked at the Jet Propulsion Laboratory, Bellcore, and Bell Labs, and taught at the University of Iowa. His research interests include software reliability engineering, software fault tolerance, distributed systems, image and video processing, web technologies, multimedia systems, and wireless communications. He has published over 150 papers in these areas. He initiated Internatinal Symponsium on Software Reliability Engineering (ISSRE), and was Program Chair for ISSRE'1996, Program Co-Chair for WWW10, and General Chair for ISSRE'2001. He also received Best Paper Awards in ISSRE'98 and in ISSRE'2002. He is the editor for two book volumes: Software Fault Tolerance, published by Wiley in 1995, and the Handbook of Software Reliability Engineering, published by IEEE and McGraw-Hill in 1996. He has been an associated editor of IEEE Transactions on Reliability, IEEE Transactions on Knowledge and Data Engineering, and Journal of Information Science and Engineering. Dr. Lyu is a fellow of IEEE.*

Professor Chin *received the B.S. and the Ph.D. in electrical engineering from the University of Missouri, Columbia. From 1979 to 1981, he was a researcher at NASA Goddard Space Flight Center, Maryland. He was on the faculty of Electrical & Computer Engineering at the University of Wisconsin, Madison from 1981 to 1995; became a full Professor in 1989 and served as Associate Department Chair from 1986 to 1990. Since 1992, he has been a professor of Computer Science of the Hong Kong University of Science & Technology (HKUST), and was Department Head from 1996 to 2001. Subsequently, he served as Vice President of Information Technology at Hong Kong Government's Applied Science and Technology Research Institute (ASTRI) before rejoining HKUST in 2003 as Vice President for Research & Development. Professor Chin had served on the editorial board of various journals including the IEEE Transactions on Image Processing, and the IEEE Transactions on Pattern Analysis and Machine Intelligence. His research interest is in computer vision, pattern recognition and digital signal processing.*