An Authentication Service Based on Trust and Clustering in Wireless Ad Hoc Networks: Description and Security Evaluation

Edith C.H. Ngai and Michael R. Lyu
Department of Computer Science and Engineering,
The Chinese University of Hong Kong
{chngai, lyu}@cse.cuhk.edu.hk
(+852) 2609 {8438, 8429}

Abstract

Security in wireless ad hoc networks is hard to achieve due to the vulnerability of its links, limited physical protection, and the absence of a centralized management point. Consequently, novel approaches are necessary to address the security problem without sacrificing the essential properties of the wireless ad hoc network. Similar to other distributed systems, security in wireless ad hoc networks usually relies on the use of key management mechanisms. In this paper, we present a distributed public key authentication service to protect the network containing malicious and colluding nodes. Our solution was built on a clustering-based network model and a trust model. These models allow mobile hosts to monitor and rate each other with an authentication metric. We also propose a new system of public key certification in conjunction with a trust value update algorithm. Our authentication service is able to discover and isolate malicious and colluding nodes in the network. Finally, we perform security evaluation on the proposed solution. We simulate a network containing malicious nodes and measure a number of metrics with various security operations to demonstrate the effectiveness of our scheme.

1. Introduction

Wireless ad hoc networks are under rapid development due to the popularity of wireless devices. However, security has become a primary concern in order to provide protected communication between mobile nodes in a hostile environment [18]. Popular network authentication architectures include X.509 standard [1] and Kerberos [10]. Pretty Good Privacy (PGP) [3, 8] functions by following a web-of-trust authentication model. PGP uses digital signatures as its form of introduction [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 has been proposed by Kapkun et al. [9]. This introduces the idea of a trust graph and a method of finding a certificate chain from one user to another. However, it assumes that the users are honest and do not issue false certificates. In reality, a node may turn from trustworthy to malicious under a sudden attack. The ability to detect 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 the network from malicious nodes.

This paper is organized as follows. Section 2 discusses related work. Section 3 sets out formal definitions of the system architecture, the network model and the trust model. In Section 4, we present security operations for public key certification, identification and isolation of malicious and colluding nodes, and trust value update. Our solution is evaluated through simulation in Section 5. Finally, our conclusions are set out in Section 6.

2. Related Work

Several public key management protocols have been proposed for wireless mobile ad hoc networks. 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 [22]. Another infrastructure called MOCA (Mobile Certificate Authority) distributes the CA (Certificate Authority) functionality over specially selected nodes based on the security and the physical characteristics of the nodes [20]. Furthermore, the fully-distributed certificate authority is proposed by Luo and Lu [11] 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. [9]. In this approach, users issue certificates by themselves without

0-7695-2553-9/06 $20.00 © 2006 IEEE

the involvement of any certificate authority. Another new protocol [14] combines threshold cryptography and routing discovery.

Our solution adopts a clustering-based network model. Amis et al. described a cluster formation approach such that a node is either a clusterhead or is at most D hops away from a clusterhead [4]. A set of algorithms has been described to handle network dynamics and optimize the group organization [13]. We now propose a modified form of Max-Min D-Cluster Formation algorithm [4].

Authentication in ad hoc networks generally depends on a trust chain formed by trusted intermediaries. Different metrics have been proposed to evaluate the confidence afforded by the chain paths. Beth et al. proposed a metric that represents a set of trust relationships by means of a directed graph [5]. They introduced the idea that direct trust values differ from recommendation trust values. The metric proposed in PGP has three levels of trust: complete trust, marginal trust, and no trust [23]. Furthermore, Krukow and Twigg present a theoretical model for trust structures in large-scale distributed systems [12].

In this paper, we define a continuous trust value to represent the trustworthiness of a node in public key certification.

3. Architecture and Models

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

3.1 Architecture of our Authentication Service

The authentication service we propose aims at providing secure public key certification despite the presence of malicious nodes in the network [16, 17]. Figure 1 shows the architecture of our authentication service. There are totally 4 layers in this architecture altogether: mobile hosts, network model, trust model, and security operations. Wireless ad hoc networks contain large number of mobile hosts, each with a transmission range that is small relative to the network size. We divide the network into different regions; and nodes in the same region form a cluster. A cluster is a connected sub-network usually having a relatively small diameter. We define two kinds of trust relationship in the clustered network, namely the trust relationship of two nodes within the same cluster, and the trust relationship of two nodes in different groups. The security operations are performed on top of the lower layers. These operations, including public key certification, identification and isolation of malicious and colluding nodes, and trust value update, will be presented in Section 4.

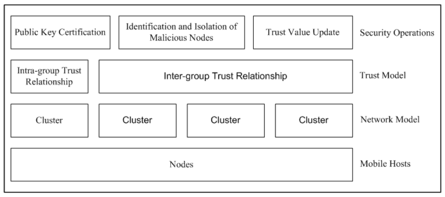


Figure 1. Architecture of Our Authentication Service

3.2 The Network Model

Our clustering-based network model improves the network security by exploiting the monitoring power of individual nodes. However, each node is only capable of monitoring to its neighboring nodes. The monitoring is more efficient when nodes are clustered. In our design, we divide the network into different cluster with similar number of hosts in each of them. Nodes in the same cluster are assigned with a unique cluster ID. We adopt the Max-Min D-Cluster Formation algorithm [4] with some modifications. In the original approach, clusters are formed by diffusing only the node ID along the wireless links. At the end of the algorithm, a node either becomes a clusterhead, or is at most d wireless hops away from its clusterhead. Nodes with higher node ID usually have a higher chance of being a clusterhead. However, node ID actually does not have any special meaning in the protection of the network's security, so we use trust value, instead of node ID, to be the criteria in cluster formation. The clusterheads are usually found to have a high trust value in compared with its cluster members.

3.3 The Trust Model

Authentication in a network requires participation of trusted entities. Wireless ad hoc networks have no centralized server for trust and key management. In our trust model, any node can act as a certifying authority. That is to say, any node can sign the public key certificate of another node in the same cluster upon request. Also, any node can observe its cluster members through certain monitoring activities and give trust values to them. In our trust model, we define the trust value (which is the authentication metric) as a continuous value between 0.0 and 1.0. This authentication metric is assigned 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, the more node i trusts node j. In our network model, we present two types of trust relationships, namely direct trust relationship and recommendation trust relationship. The direct trust relationship is the trust relationship between two nodes in the same cluster, while the recommendation trust relationship is the trust relationship between nodes of different groups. They will be presented with details in Section 4.4.

4. Trust- and Clustering-Based Authentication Service

This section covers the detailed operations of the trust- and clustering-based authentication service proposed. It includes a description of the clustering-based structure maintenance, public key certification, identification and isolation of malicious and colluding nodes, and trust value update.

4.1 Clustering-Based Structure Maintenance

As mentioned in the previous section, the Max-Min D-cluster formation algorithm will be run when a network forms.

Unfortunately, this algorithm forms clusters of widely differing size. It is beneficial to the performance and security of the network if clusters of similar size can be formed. With similar number of members in the clusters, the workload on the clusterheads to maintain their own clusters is evenly distributed. Also, this helps to limit the physical size of the clusters, which enhances the nodes' ability to monitor their neighboring (see Section 4.3). Furthermore, it ensures the cluster can provide enough introducing nodes. However, the clusters formed by this algorithm are not in balance sizes. In the meantime, balance clustering structure benefits to the performance and the security of the network. With similar number of members in the clusters, the clusterheads share almost similar workload to maintain their own clusters. Also, it avoids nodes from crowding in the same cluster, which may reduce the neighboring monitoring power due to the large node distances. Furthermore, it prevents the cluster from not providing enough number of introducing nodes.

The mobile nature of host in ad hoc network has to be handled property. The wireless devices leave one cluster and join another frequently in a highly mobile environment. We handle the changes of membership among the clusters and maintain a balanced clustering structure as follows.

Each node requests the cluster ID and cluster size of its neighboring nodes periodically in order to identify its neighboring clusters. In each cycle, each node broadcasts this request to its neighboring nodes and collects the replies. Algorithm 1 shows that a node joins the smallest neighboring cluster only if it moves out of range of the original cluster or the sizes of the neighboring clusters are not within a certain range. A clusterhead includes the cluster information, such as the cluster size, when it sends control messages to its cluster members. We define two parameters S and L which to represent the minimum and maximum permissible cluster size in the network. If the size of a neighboring cluster is outside this ranges, the node will leave its original cluster and join the smaller neighboring cluster even it still receives the original cluster's ID. Figure 2 shows that this algorithm enables the network to maintain a balanced structure in a network with 40 nodes.

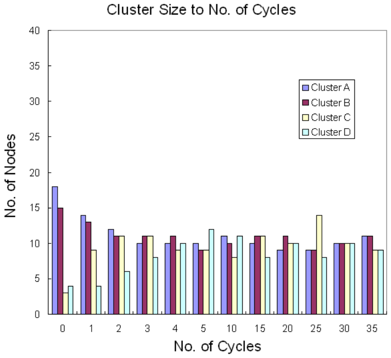


Figure 2. Evolution of Cluster Size

Algorithm 1 Clustering Structure Maintenance

```
for each cycle do
  for each node $n$ do
    $v_n \xrightarrow{b} v_{neighbor_k} : \langle v_n, REQ_{ClusterID} \rangle$;
    $v_{neighbor_k} \rightarrow v_n :$
    $\langle v_n, v_{neighbor_k}, ClusterID_{neighbor_k} \rangle$;
    if $ClusterID_n \neq \forall\ ClusterID_{neighbor_k}$ or $\exists!(S \leq size\,of\ ClusterID_{neighbor_k} \leq L)$ then
      $min_{size} = size\,of\ ClusterID_{neighbor_k}$;
      $min_{cluster} = ClusterID_{neighbor_k}$;
      for $\forall\ ClusterID_{neighbor_k}$ do
        if $min_{size} < size\,of\ ClusterID_{neighbor_k}$ then
          $min_{size} = size\,of\ ClusterID_{neighbor_k}$;
          $min_{cluster} = ClusterID_{neighbor_k}$;
        end if
      end for
      Joins the $min_{cluster}$;
    end if
  end for
end for
```

4.2 Public Key Certification

Authentication in our network relies on the public key certificates signed by some trustable nodes. Let s be the node requesting for the 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 3(a). Every node is able to request for the public key certificates of other new nodes. However, nodes in the same cluster are assumed to know each other by means of their mutual monitoring components. Given the above assumptions, we focus on public key certification in the case that where s and t belong to different groups. Nodes which are in the same cluster as t and have already built up a trust relationship with s can be introducers. The requesting node s selects a certain number n of nodes with the highest trust values as introducers and sends them request messages. The introducers i_1, i_2 ,. . . , i_n, will reply with the public key of the target node t after receiving the messages. Apart from the public key of t, they supply the trust value of t as well. These values from i_1, i_2, . . . , i_n, will be used to calculate 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.

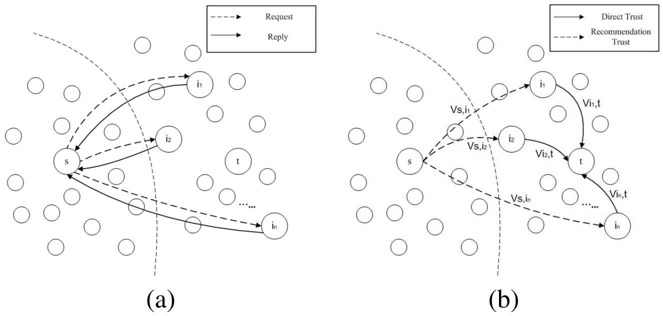


Figure 3. (a)Public Key Certification (b)Trust Value Update

Algorithm 2 shows the procedure for requesting the public key certificates of a target node. In this algorithm, node v_i is requesting the public key certificates of node v_j. Let us assure that node v_i belongs to cluster $CLUST_A$ and node v_j belongs to cluster $CLUST_B$. Before sending out the request message, node v_i first checks whether it is in the same cluster as v_j. If it is, it sends the request message to its neighboring nodes, assuring that some of its neighboring nodes have built up a direct trust relationship with v_j. After receiving the reply messages, v_i stores the public key and updates the trust value by averaging the received values. Nodes are assured to be able to discover any malicious nodes in their own cluster (see Section 4.3), so the neighboring nodes that they are communicating with are always trustworthy. On the other hand, if v_i and v_j are in different clusters, then the problem becomes more complicated. Node v_i has to select some trustworthy nodes in the target cluster to be the introducing nodes, or so-called introducers. They are nodes in the same cluster as v_j for which v_i has high trust values. In a similar way to the previous case, v_i sends the request message to the introducers and waits for the replies. However, it is possible for the introducers to be malicious, and for v_i to have not yet discovered this by direct monitoring due to the long distance between v_i and the introducers. Therefore, a voting procedure will be carried out to conclude the correct public key of the target node by majority vote.

Algorithm 2 Request for Public Key Certificates

```
  Define $v_i$ as a node with the node $ID$, $i$; $V_{k,j}$ as the trust value
  from $v_k$ to $v_j$; and $PK_j$ as the public key of $v_j$. Given $v_i$
  belongs to $CLUST_A$ and $v_j$ belongs to $CLUST_B$. A node $v_i$
  requests for the public key certificate of a node $v_j$:
if $(CLUST_A == CLUST_B)$ then
  $v_i$ sends request to neighbors $v_k$:
  $v_i \xrightarrow{b} v_k : \langle v_i, v_j, REQ_{CERT} \rangle$;
  $v_k \rightarrow v_i : \langle v_j, V_{k,j}, PK_j, ... \rangle_{SK_{v_k}}$;
  $v_i$ updates $PK_j$ and $V_j$;
else
  $v_i$ selects trust-worthy nodes in $CLUST_B$ as introducers $i_k$;
  $v_i \xrightarrow{b} i_k : \langle v_i, v_j, REQ_{CERT} \rangle$;
  $i_k \rightarrow v_i : \langle v_j, V_{k,j}, PK_j, ... \rangle_{SK_{i_k}}$;
  $v_i$ compares the $PK_j$ from the received certificates and up-
  date $PK_j$ in their repository;
  $v_i$ calculates and updates $V_j$;
end if
```

4.3 Identification and Isolation of Malicious Nodes

The first method to identify malicious nodes in the same cluster is by direct monitoring of individual nodes. Nodes in a wireless ad hoc network are able to observe the behavior of their 1-hop neighbors directly. This can be done by listening to the traffic via wireless communications using a monitoring facility such as a watchdog [15]. A number of studies [6, 19] have been carried out on detecting and isolating misbehaving nodes in the network through cooperation among the nodes.

The second method to isolate malicious nodes is by identifying suspicious introducers who provide public key certificates different from the others. In each public key certificate request, a node finds more than one introducer in order to obtain multiple reply messages. After decrypting the public key certificates by using the introducers' public keys, it can read the public key of the target node provided by the introducers. The resulting public key responses should be the same if all the introducers are honestly supplying the correct answer. If some introducers provide a public key of the target node that is different from the others, then these introducers are suspected to be malicious.

In the third method, the requesting node identifies the target node as malicious if the trust values provided from the introducers indicate that. After the requesting node sends

0-7695-2553-9/06 $20.00 © 2006 IEEE

the message asking for the public key of the target node, its introducers reply with the public key certificates. These contain includes not only the ID and public key of the target node, but also the trust value from that particular introducer to the target node. The requesting node can thus summarize the trust value of the target node. If the trust value of the target node is lower than a certain threshold, then the target node is indicated as dishonest.

A malicious node may not only sign an incorrect public key certificate itself but also collude with other malicious nodes to make the false key or trust values more convincing. Colluding nodes are able to sign a false public key of the target node, defame a honest node by providing extremely low trust values, or conspire to introduce a new colluding node by raising its trust values together. To deal with the collusion of nodes, we have some suggestions relating to the process of public key certification.

Given that the maximum number of malicious nodes in a collusion be f and the number of introducers be k, the number of nodes providing a common public key for the target node should be required to be more than f, such that $k >= f + 1$. If $k <= f$, the requesting node has to send request messages to more introducers. After discovering the correct key of the target node, the requesting node will decrease by 0.5 the trust values of introducers who provide a false public key. This prevents them from becoming introducers later. Next, only trust values provided by trustworthy introducers will be analyzed. The requesting node may remove the upper and lower trust value in the range of values supplied by the introducers before calculating the mean and the standard deviation ($S.D.$). Then, it will filter out the values which differ from the mean by more than $2S.D.$. The remaining values will be used to calculate the final trust value of the target node. Finally, the requesting node will increase the trust value of the non-filtered introducers by 0.1, and decrease filtered introducers by 0.3. To make this method more secure, the requesting node may send request to more nodes if the number of non-filtered introducers is less than f. Moreover, isolation messages can be sent if the trust value of a node is found to be less than 0. Nodes that receive isolation messages from a certain number of trustworthy nodes may isolate the suspicious node if it agrees with their evaluation.

4.4 Trust Value Update

After filtering out suspicious introducers, the enquiring node obtains the trust values for t from the remaining introducers i_k. These values can be used to calculate the ultimate trust value V_t of t in the view of s as shown in Figure 3(b).

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 for t. $V_{s,i_1}, V_{s,i_2}, \ldots, V_{s,i_n}$ denote trust values from s to the introducers $i_1, i_2, \ldots, i_n$; while $V_{i_1,t}$, $V_{i_2,t}, \ldots, V_{i_n,t}$ denote trust values from the introducers i_1, $i_2, \ldots, i_n$ 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, $V_{s,i_k,t}$, from s to t via the intermediate node i_k on a single path, we apply the following formula:

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

This value is composed of the trust values from s to i_k and from i_k to t. The differing trust values on various paths can be used as collective information to compute the ultimate trust value from s to t through different introducers. The ultimate value, $V_{s,i_k,t}$, can be obtained by the following formula:

$$V_t = 1 - \Pi_{k=1}^{n}(1 - V_{s,i_k,t}), \qquad (2)$$

where n denotes the number of paths.

This value is now inserted into the trust table of s. If V_t is high, it indicates that t can be a possible introducer in the future. Apart from the trust value of the target, the trust value of the introducers will also be updated. The trust value of the introducers found to be malicious, will be lowered and they may be isolated as mentioned in section 4.3. In contrast, the requesting node will gradually increase the trust values of introducers that provide correct public key certificates for the target nodes. Finally, a node can further adjust the public keys and trust values after gaining real experiences of using the keys and communicating with the nodes.

5. Simulation Results

In this section, we evaluate the security performance of the authentication service proposed in extensive simulation tests.

5.1 Experimental Setup

We implemented our design in the network simulator GloMoSim [21]. Our main objective in the security evaluation is to investigate whether our authentication service provides effective security evaluation results in a public key certification process in the presence of malicious nodes. We imitate the malicious nodes by selecting a certain percentage of the nodes in the network randomly and programming them to reply with false public key certificates. A false public key certificate may contain an incorrect public key or a false trust value for other nodes. Table 1 shows the parameter settings of our experiments. In a cycle, nodes ask

Table 1. Simulation Parameters

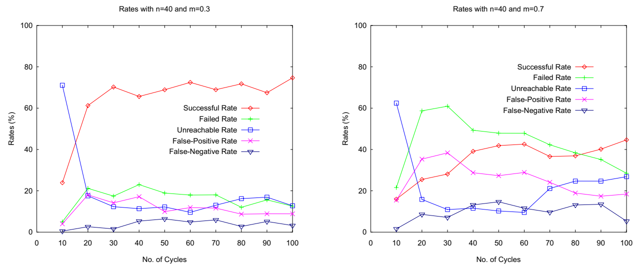
Network	
Network size	1500m x 1500m or 3000m x 3000m
No. of nodes	n
Proportion of malicious nodes	m
Mobility	
Mobility	Random-waypoint
Pause time	20s
Max. speed	10m/s
Clustering	
D-hops	3
Min. cluster size	S
Max. cluster size	L
NeighborMonitoring	
No. of cycles required to identify malicious neighbors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

Table 2. Possible Cases with 3 Introducers

		Outcome				
ID	Cases	Success	Failure	Incom-plete	False+	False-
0	Not enough Introducers			√		
1	OOO	√				
2	OOX	√				
3	OXX		√		√	
4	XXX		√			
5	OO	√				
6	OX		√		√	
7	XX		√			
8	O	√				
9	X		√			√
10	No reply			√		

other nodes which cluster they belong to, and update their own cluster membership. They also request the public key certificates of another node in each cycle. The requesting node concludes the correct public key of the target node by majority vote. At the same time, it may identify suspicious introducers who sign incorrect public key certificates.

Table 2 shows all 11 possible cases of public key certification with 3 introducers. We define a public key certification as successful if its conclusion on the public key is correct, fail if its conclusion on the public key is incorrect or cannot be made, unreachable if it has not enough introducers, false-positive if a trustworthy node is wrongly identified as malicious, and false-negative if a malicious node is not detected. Case 0 represents the situation that there are not enough introducers to support this request, so the request message will not be sent. It results in an increased occurrence of the "unreachable" state. In case 1 to case 10, the request messages are sent and various numbers of public key certificates are received from the introducers. The symbol 'O' indicates that a correct certificate is received for the target node, while the symbol 'X' indicates an incorrect one.



(a) (b)

Figure 4. Rates to No. of Cycles with n=40 and r=100 (a)m=0.3 (b)m=0.7

5.2 Testing the Effectiveness of Neighbor Monitoring

In this experiment, we implement the neighbor monitoring algorithm to identify malicious nodes in the network. When a node stays in the same cluster for a certain period of time, it may be able to detect malicious nodes in its neighborhood. This ability is tested here.

Figure 4(a) shows the experiment result with 40 nodes. 30% of nodes in the network are malicious. Only around 70% of attempts to build a sufficient trust relationship were successful which is as same as the percentage of honest nodes in the network. In addition, nodes in the network do not know each other at the beginning. It takes time to build up the trust relationships among them, so the unreachable rate is high in the first few decades of cycles. The failed rates are improved to 10% according to the two result, which is lower than the proportion of malicious nodes in the network.

Similarly, we simulate a network with 70% of the nodes malicious; this represents a hostile network condition. Figure 4(b) shows that the success rate is around 40%. The failed rate is around 30%, which is lower than the proportion of malicious nodes.

The above experimental results show that the monitoring power of neighboring nodes does not lead to any great improvement in the success rate of public key certifications. This may be because the mobility of the node is too high, so that nodes do not have enough time to detect malicious neighbors. In order to protect the network's security, it is evidently necessary to rely on other security operations, such as the identification of suspicious nodes via public key certification.

5.3 Testing the Effectiveness of Isolation of Malicious Nodes

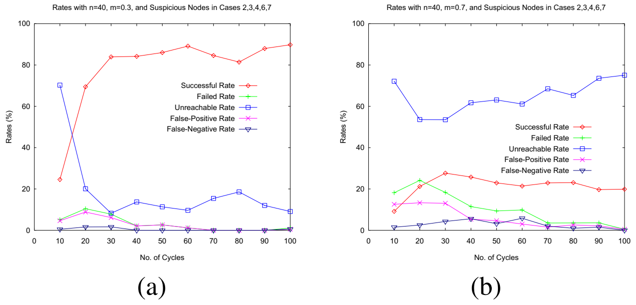
To improve the network's security, we next include the identification of suspicious nodes via public key certification. In this experiment, suspicious nodes will be identified not only by neighbor monitoring, but also by analyzing the public key certificates they supply. Introducers providing certificates different from the majority are identified as suspicious and are excluded from being selected as introducers. There are suspicious nodes in cases 2, 3, 4, 6, and 7 of Table 2. It should be noted that cases 3 and 6 lead to a false positive error, i.e. an honest node may be falsely identified as malicious. Case 9 leads to a false negative result: a single reply cannot be compared with other certificates, so it is always assured to be correct.

Figure 5(a) shows the experiment are result with n=40, m=0.3 and r=100. The success rate is greatly compared with Figure 4(a), and the failed rate is very low. It indicates that the identification and isolation of suspicious nodes in cases 2, 3, 4, 6, and 7 efficiently reduces the number of malicious introducers. The high successful rate and the low failed rate show that satisfactory authentication occurs in the network. Figure 5(b) shows the result of the same experiment with m=0.7. In this case, the success rate is too low, between 20% and 30%, although the failed rate is also low. We notice that the unreachable rate is quite high. This is because many of the honest nodes are falsely identified as suspicious and isolated from taking the role of introducers. Hence, in such a hostile environment, there may easily be not enough introducers. This is the major reason for the high unreachable rate and hence the low success rate.

Unlike the previous approach, the requesting node may choose not to disregard suspicious introducers if it doubts trustworthy introducers may still exist. For example, in cases 3 and 6, the node may opt to keep all the suspicious introducers instead of isolating them. This decision will be based on the trust values of the introducers and their past records in public key certification. If this policy is carried out, the false-positive errors brought in cases 3 and 6 are avoided. The drop in the false-positive error rate may prevent some malicious nodes from being isolated immediately.

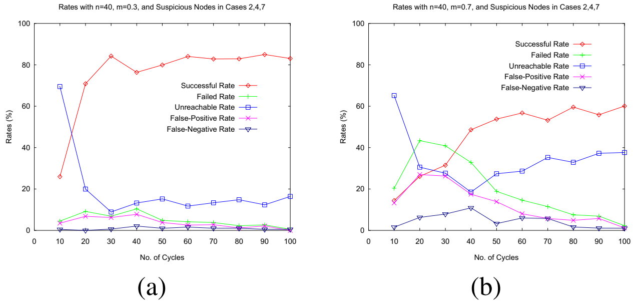
The effect of applying this policy. Now, Figure 6(a) show success rate is quite high and the failed rate almost zero after running for 100 cycles. It indicates that the new policy on the acceptance of suspicious nodes gives a satisfactory result in public key certification. However, the success rate in this figure is not as good as that in Figure 5(a). The main reason is that some of the malicious nodes are not isolated in cases 3 and 6, so it takes longer for a node to discover the malicious nodes in the network. When the malicious nodes are not isolated, they may still be selected as introducers and to provide false certificates, which may decrease the success rate. On the other hand, Figure 6(b) shows that, with a network containing 70% malicious nodes, the success rate is greatly improved with the new strategy compared with Figure 5(b). There is also an extremely low failed rate with this policy. This is because keeping honest introducers in cases 3 and 6 provides more choices of introducers for certifications. It effectively decreases the unreachable rate and increases the success rate in the network.

To sum up, the first approach is able to isolate the malicious nodes quickly. On the other hand, the second approach avoids the false-positive errors in the public key certifications, which effectively reduces the unreachable rate, especially in a hostile environment. We may select the isolation strategy according to the current network condition, so as to give a better performance in authentication.



(a) (b)

Figure 5. Rates with n=40, r=100, and Isolation of Suspicious Nodes in Cases 2,3,4,6,7 (a)m=0.3 (b)m=0.7



(a) (b)

Figure 6. Rates with n=40, r=100, and Isolation of Suspicious Nodes in Cases 2,4,7 (a)m=0.3 (b)m=0.7

0-7695-2553-9/06 $20.00 © 2006 IEEE

6. Conclusion

In conclusion, we propose a secure, scalable and distributed authentication service that enhances the correctness of public key certification in wireless ad hoc networks in the presence of malicious nodes. We suggest a well-defined trust model and a network model to develop our public key authentication service. The trust model allows nodes to monitor and update trust values for each other in a distributed manner. The network model is clustering-based; this facilitates behavior monitoring and provides high availability for public key certification. Our solution provides security operations, including public key certification, identification and isolation of malicious nodes, and trust value update in a novel way. These operations reduce the chance of a node getting false public keys of other nodes. Extensive experiments have been conducted to evaluate the performance of our solution from the security perspective. A number of metrics, including the success rate, failed rate, unreachable rate, and false-positive and false-negative error rates are evaluated. The neighbor monitoring power and different strategies for the identification and isolation of suspicious nodes are evaluated. The experimental results show the effectiveness of our solution in providing a secure authentication service at various levels of node hostility.

7. Acknowledgement

The work described in this paper was fully supported by two grants, RGC Project No. CUHK4205/04E and UGC Project No. AoE/E-01/99, of the Hong Kong Special Administrative Region, China.

References

[1] "Internet X.509 Public Key Infrastructure," January 1999, Available at http://www.ietf.org/rfc/rfc2459.txt.

[2] "How PGP Works," Chapter 1 of "Introduction to Cryptography" in the PGP 6.5.1 documentation, Copyright ©1990-1999 Network Associates, Inc. and its Affiliated Companies, Available at http://www.pgpi.org/doc/pgpintro/.

[3] A. Abdul-Rahman, "The PGP trust model," *EDI-Forum: the Journal of Electronic Commerce*, April 1997, Available at http://www.cs.ucl.ac.uk/staff/F.AbdulRahman/docs/.

[4] A. D. Amis, R. Prakash, T. H. P. Vuong, and D. T. Huynh, "Max-min D-cluster Formation in Wireless Ad Hoc Network," *Proc. of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom'00)*, pp. 32–41, March 2000.

[5] T. Beth, B. Malte, and K. Birgit, "Valuation of Trust in Open Networks," *Proc. of the 3rd European Symposium on Research in Computer Security (ESORICS '94)*, pp. 3–18, Brighton, England, 7-9 November 1994.

[6] J.-Y. L. Boudee and S. Buchegger, "Performance analysis of the confidant protocol (cooperation of nodes: Fairness in dynamic ad-hoc networks)," *Proc. of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02)*, pp. 80–91, Lausanne, Switzerland, June 2002.

[7] S. Capkuny, L. Buttyan, and J.-P. Hubaux, "Self-Organized Public-Key Management for Mobile Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 2, no. 1, March 2003.

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

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

[10] J. Kohl and B. Neuman, "The Kerberos network authentication service (version 5)," RFC-1510, June 1991, Available at http://www.ietf.org/rfc/rfc1510.txt.

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

[12] K. Krukow and A. Twigg, "Distributed Approximation of Fixed-Points in Trust Structures," *Proc. of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*, June 2005, pp. 805 - 814.

[13] 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.

[14] S.-T. Li and X. Wang, "Ad Hoc Network Security with Geographical Aids," *Proc. of the 2004 IEEE International Conference on Networking, Sensing & Control*, pp. 474–479, Taipei, Taiwan, 21-23 March 2004.

[15] S. Marti , T. J. Giuli , K. Lai , M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," *The 6th Annual International Conference on Mobile Computing and Networking*, pp. 255–265, Boston, Massachusetts, United States, 6-11 August 2000.

[16] C. H. Ngai and M. R. Lyu, "Trust- and Clustering-Based Authentication Services in Mobile Ad Hoc Networks," *Proc. 2nd International Workshop on Mobile Distributed Computing (MDC'04)*, pp. 582–587, Tokyo, Japan, 23-26 March 2004.

[17] C. H. Ngai, M. R. Lyu, and R. T. Chin, "An Authentication Service Against Dishonest Users in Mobile Ad Hoc Networks," *Proc. 2004 IEEE Aerospace Conference*, Big Sky, Montana, U.S.A., 6-13 March 2004.

[18] H. Yang, H. Luo, F. Ye, S. Lu, and L. Zhang, "Security in Mobile Ad Hoc Networks: Challenges and Solutions," *IEEE Wireless Communications*, pp. 38–47, February 2004.

[19] H. Yang, X. Meng, and S. Lu, "Self-organised network-layer security in mobile ad hoc networks," In *Proc. of the ACM Workshop on Wireless Security (Wise'02)*, pp. 11–20, Atlanta, GA, USA, 28 September 2002.

[20] S. Yi, R. Kravets, "MOCA: Mobile Certificate Authority for Wireless Ad Hoc Networks," *2nd Annual PKI Research Workshop Program (PKI'03)*, pp. 65–79, Gaithersburg, Maryland, April 2003.

[21] X. Zeng, R. Bagrodia, and M. Gerla, "GloMoSim: a Library for Parallel Simulation of Large-scale Wireless Networks," *Proc. of the 12th Workshop on Parallel and Distributed Simulations (PADS'98)*, pp. 154–161, Banff, Alberta, Canada, 26-29 May 1998.

[22] L. Zhou and Z. J. Hass, "Securing Ad Hoc Networks," *IEEE Networks Magazine*, vol. 13, issue 6, pp. 24–30, 1999.

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