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

Edith C.H. Ngai, Student, and Michael R. Lyu, Professor, Fellow, IEEE
Department of Computer Science & Engineering, The Chinese University of Hong Kong,
Shatin, N.T., Hong Kong

Abstract—**A mobile ad hoc network is a wireless communication network which does not rely on a fixed infrastructure and is lack of any centralized control. It is vulnerable to security attacks, so protecting the security of the network is essential. Like many distributed systems, security in ad hoc networks widely relies on the use of key management mechanisms. However, traditional key management systems are not appropriate for them. Our research aims at providing a secure and distributed authentication service in the ad hoc networks. We propose a secure public key authentication service based on a trust model and a network model to prevent nodes from obtaining false public keys of the others when there are malicious nodes in the networks. We perform an overall evaluation of our proposed approach by simulations. The experimental results indicate clear advantages of our approach in providing effective security in mobile ad hoc networks.**

***Index Terms*—Security, Mobile Ad Hoc Network, Trust, Clustering, Authentication.**

I. Introduction

A mobile ad hoc network is a collection of nodes with no infrastructure while its nodes are connected with wireless links. Nodes in the network are able to sense and discover nearby nodes [1]. They communicate with each other by forwarding packets hop by hop in the network [2]. Also, the topology of the ad hoc network is dynamically changing and the nodes of the ad hoc network are often mobile. A major challenge in the design of the mobile ad hoc network is to protect its vulnerability from security attacks. As in many distributed systems, security in ad hoc networks is based on the use of a key management system for authentication. Specific key management systems have to be developed to suit the characteristics of mobile ad hoc networks [3]. In this paper, we propose a new key management scheme with a well-defined trust model and a network model. Our trust model follows the "web of trust" approach proposed in Pretty Good Privacy [4] with several new contributions. Our network model, on the other hand, is based on clustering models [5] 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 in the ad hoc networks.

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. Some trustworthy nodes of the same group introduce new nodes. Nodes in the network monitor the behavior of each other and update their trust tables accordingly. Our public key management mechanism endures the false certificates issued by dishonest users and malicious nodes, and avoids them to be selected as introducing nodes. These features provide a 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 network model and the trust model which lay the foundation for our network design. The system assumptions are also stated. In Section 4, we further propose the security operations on the public key certification and the update of trust tables for the network protection. The new solution is evaluated through simulation and implementation, and the results are presented in Section 5. Finally, we conclude the paper in Section 6.

II. Related Work

Traditional network authentication solutions rely on physically present, trusted third-party servers, or called certificate authorities (CAs). Security requirements for CAs are important with an exploration of a wide range of attacks that can be mounted against CAs [6]. Popular network authentication architectures include X.509 standard [7] and Kerberos [8]. However, ad hoc networks are infrastructure-less, and there is no centralized server for key managements. Hence traditional CA-based solutions do not meet the requirements of the mobile ad hoc networks.

Pretty Good Privacy (PGP) [4, 9] 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 [10]. Nevertheless, the distribution of certificates is based on publicly accessible certificate directories that reside on centrally managed servers, which is not a fully self-organized approach.

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. The paper written by Zhou and Hass [13] proposes a partially-distributed certificate authority that makes use of a *(k,n)* threshold scheme in distributing the services of the certificate authority to a set of specialized server nodes. However, high mobility causes frequent route changes, thus contacting the local CA in a timely fashion is non-trivial. Besides, in ad-hoc networks, the local CA may be multi-hops away and also move. This not only causes complicated dynamic repartitioning of the network, but also stretches the problem of locating and tracking a local CA server. Moreover, every local CA is exposed to single point of compromises or denial of service (DoS) attacks. Similar public key infrastructure service called MOCA (Mobile Certificate Authority) also employs threshold cryptography to distribute the CA functionality over specially selected nodes based on the security and the physical characteristics [14, 15].

Similar to the partially-distributed CA, the fully-distributed certificate authority proposed by Luo and Lu [16] extends the idea of the partially-distributed approach by distributing the certificate services to every node, but it is possible for a node to find insufficient number of neighboring nodes to sign the certificate.

Other solutions include the self-issued certificates proposed by Hubaux et. al. [17]. It is similar to PGP in the sense that public key certificates are issued by the users. However, as opposed to PGP, it does not rely on certificate directories for the distribution of certificates. Instead, in this system, certificates are stored and distributed by the users. This approach considers the signers along a certificate chain to be trustworthy, which is hard to guarantee.

Therefore, our design aims at working in the presence of malicious nodes which may sign incorrect public key certificates. We tend to make better use of the monitoring power, increase the security by selecting trustworthy and multiple nodes as signers, and isolate malicious nodes from the network when they are detected. We divide the network into different clusters to allow nodes in the same cluster to monitor each other naturally. A new node builds up inter-cluster relationship gradually such that it can request certificates of the nodes in other clusters by requesting the trustworthy nodes it knows in these clusters. We prevent incorrect certificates signed on behalf of malicious nodes by requiring multiple introducers for a new certificate. Also, we define a trust value as an authentication metric for a node's reference in selecting introducers. This value will be updated from time to time to increase the security in authentication and to isolate malicious nodes from the network.

III. MODELS

In this section we investigate two major models related to our approach: the network model and the trust model. We survey existing work in these two models and establish the framework of our design for better security in mobile ad hoc networks. We also state the assumptions of our system.

A. Primitives

As an ad hoc network is lack of infrastructure for any centralized control, its operations are usually performed in a fully-distributed manner. This means that nodes in the network play an equal role and share their tasks evenly. From this point of view, we perceive that the "web of trust" approach proposed by Pretty Good Privacy [4, 9] is compatible with the characteristics of the ad hoc network in providing security. The most important feature of our approach is that it can provide network security in the presence of malicious nodes. In addition, it does not rely on any centralized repository to store the certificates, such as PGP does. The certificates are stored and distributed by every node in our approach. A node requests multiple introducers for certification, where PGP only relies on single trust chain for each request. Moreover, it involves a number of security operations, like neighbor monitoring, trust value update, and isolation of malicious nodes.

With our clustering-based network model, behavior monitoring can be conducted in a natural way and availability is ensured for a node to find suitable introducers in the network. We also impose a trust model on the network to increase its security in selecting introducers. Our trust model employs a quantitative trust value to represent the level of trust a node holds. The trust value update operation maintains up-to-date trust values and isolates malicious nodes from the network.

B. The Network Model

Obtaining a hierarchical organization of a network is a well-known and well-studied problem in distributed computing. 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 algorithm is based on independent dominating sets of graphs. Weight-based clustering algorithms, on the other hand, are proposed in [18]. These algorithms define a vertex with optimal weight within its neighborhood as a cluster-head, and the neighborhood of the cluster-head is a cluster. The weighting idea is generalized in [19] 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 cluster-head or is at most *d* hops away from a cluster-head [20]. Weakly-connected dominating set is proposed for clustering ad hoc networks in [21]. A zonal algorithm for clustering ad hoc networks is proposed in [5] to divide the network into different regions. It makes adjustments along the borders of the regions to produce a weakly-connected dominating set of the entire graph. An adaptive method for

maintaining a hierarchical structure in an ad hoc network is proposed in [22], in which the role of nodes and the cluster size can be changed autonomously with the status. Finally, a model of location-aware clustering in ad hoc networks is proposed in [23]. It divides the whole network into a number of geographic zones where each zone forms a logical cluster.

Apart from the view of efficiency, we believe clustering improves the security of a network as well. A mobile ad hoc network lacks a centralized server for management and monitoring purposes. Its security measure relies on individual nodes to monitor each other. However, direct monitoring capability is normally limited to neighboring nodes. Nodes clustered together allow the monitoring work to proceed more naturally, so as to improve the overall network security. In this paper, we propose a trust- and clustering-based public key management approach for the mobile ad hoc network. There are quite a number of existing solutions for clustering in ad hoc networks. Their detailed discussions are beyond the scope of this paper. In our public key management approach, nevertheless, we assume the network has an algorithm to partition the nodes into different clusters with unique IDs. As an example, Figure 1 shows a mobile ad hoc network with four clusters.

C. The Trust Model

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 the trustworthiness of these entities. Many metrics have been proposed to evaluate the confidence afforded by different paths. One of the proposed metric represents a set of trust relationship by a directed graph [24]. It introduces the semantics of direct trust values and recommendation trust values, and shows that different values can be combined to a single value. Moreover, a metric in PGP includes three levels of trust: Complete trust, Marginal trust, and No trust [25]. Another approach explores the use of multiple paths to redundantly authenticate a channel and focuses on two notions of path independence [26]. Besides, a trust management method is proposed in [27] to address the problem of reputation-based trust management. It allows assessing trust by computing agent reputation from its former interactions with other agents, and manages data in a decentralized way with P-Grid [28]. Another reputation system, called EigenTrust [29], employs a distributed and secure method to compute global trust values based on Power iteration.

In our trust model, we define the authentication metric as a continuous value between 0.0 and 1.0. With the consideration in our network model, we define a direct trust relationship as the trust relationship between two nodes in the same group and a recommendation trust as the trust relationship between nodes in different groups. We apply the equations for calculation and combination of trust values from the direct trust and the recommendation trust approach in [24].

The first equation we engage calculates the trust value of a new path. It is a result of the computation of the direct trust values and the semantics of the recommendation values. Direct trust relationship means to believe an entity in its capabilities with respect to the given trust class. Recommendation trust, on the other hand, 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. This is shown in Eq. (1), which derives the new trust relationships from A to C. The symbol V_1 represents the recommendation trust value from A to B, while V_2 represents a direct trust value from B to C.

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

After deriving the direct trust relationship between two entities, it is necessary to classify the trust expressions with respect to the last recommending entity on the recommendation path to get a result which conforms to the semantics of the trust values. This is given in Eq. (2):

$$V_{com} = 1 - \prod_{i=1}^{m} \sqrt[n_i]{\prod_{j=1}^{n_i} (1 - V_{i,j})} \quad (2)$$

Let $Pi\ (i=1...m)$ be defined as the different last entities on the recommendation paths. In Eq. (2), $Vi,j\ (i=1...m,\ j=1\ ...n_i)$, where $Vi,j \neq 0$, are the values of the trust relationships (with n_i denoting the number of relationships having Pi as the last recommending entity). The $Vi,*$ represent values of trust relationships with the same last recommending entity Pi. This equation follows the meaning of a recommendation: There exists an entity which has some experiences with the entity to be recommended. These experiences have been propagated along the recommendation paths, undergoing a reduction corresponding to the values of the recommendation trusts on their way. Since there are not necessarily unique paths from one entity to another, the same experiences may be propagated to an entity several times via different paths with different reductions. Eq. (2) is thus used for drawing a consistent conclusion when there exist several derived trust relationships of the same trust class between two entities.

D. Assumptions

Some assumptions are made in our public key management algorithm in mobile ad hoc networks. They include:

1. Each node keeps exchanging information with other nodes in the group it belongs to.
2. Each node is able to monitor the behavior of its group members and obtain their public keys.
3. Each node keeps a trust table for storing trust values of other nodes.

Basically, we assume that there is an underlying clustering algorithm in the network, so nodes are divided into groups with unique IDs. Nodes are equipped with some local detecting components, like watchdog for monitoring the behavior of neighboring nodes, so they can determine which nodes are trustworthy within the group. Finally, our trust model requires each node to keep a trust table for storing the trust values and public keys of the nodes they know.

IV. SECURITY OPERATIONS

In this section we describe two security operations related to our approach: the public key certification and the trust value update.

A. Public Key Certification

Authentication in our network relies on the public key certificates signed by some trustworthy nodes. Let *s* be the node requesting 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,..., i_n, as shown in Figure 2. Nodes are expected to know their group members by means of their monitoring components and the short distances among them. With the above assumptions, we focus on the public key certifications where *s* and *t* belong to different groups. Nodes in the same group with *t*, which have already built up reliable trust relationship with *s*, can be introducers. The introducers i_1, i_2 ,..., i_n, reply to *s* with the public key and the trust value of *t* upon request. Node *s* will calculate the trust value of *t* by combining the trust value from i_1, i_2, ..., i_n. Each reply message should be signed by the corresponding introducer with its private key for validation purpose.

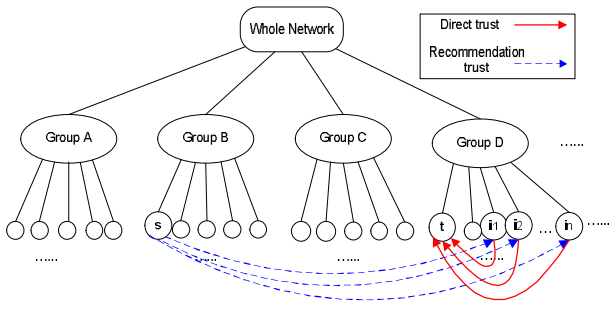

Fig. 2. Public key certification

Table 1 shows the operations of *s* in obtaining public key certificates of *t*. To request the public key of *t*, *s* first looks up the group ID j_t of node *t*. Then, it sorts the trust values of nodes that belong to j_t and selects the nodes with the highest trust values as introducers i_1, i_2 ..., i_n, and sends them request messages. After collecting the reply messages that are 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 selects Pk_t as the one with majority votes. If there is no majority vote, *s* tries to select more introducers and sends the request messages again when it is possible. After that, it reduces the trust values of the nodes which do not agree with that public key, so to avoid selecting these nodes, now deemed dishonest or malicious, as introducers in the future. Finally, *s* calculates and updates the trust value of *t*, V_t.

TABLE I

OPERATIONS OF *s* IN PUBLIC KEY CERTIFIATION

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

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

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

B. Update of Trust Table

Our clustering-based network model and well-defined trust model divide nodes into different groups, and develop both direct and recommendation trust relationships. In our models, a node builds up trust relationships not only with its group members, but also with nodes in other groups. The inter-group trust relationships are established by the recommendations from other nodes or the node's own experiences. Our trust value update mechanism maintains the trust relationships and keeps the trust values up-to-date. With the inter-group trust relationships, a requesting node selects a certain number of trustworthy nodes in the target group as introducers. Each

introducer becomes the intermediate node on an independent trust path from the requesting node to the target node.

In computing Vs,i_k,t at *s*, the trust relationship derived from a trust path is actually combined by a DIRECT trust relationship from the introducer to *t* and the RECOMMENDATION trust relationship from *s* to the introducer. Since the introducer and *t* are in the same group, the introducer can build up DIRECT trust relationship by its monitoring power within a short distance. For *s* and *t*, however, they have not yet establish direct trust relationship with each other due to their long distance. Therefore, node *s* has to reach *t* via the recommendation of an introducer. The value between *s* and the introducer is the recommendation trust. In computing Vs,i_k,t, the direct trust from the introducer to *t* and the recommendation trust from *s* to the introducer are explicitly included, which is shown in Figure 3. In Figure 3, *s* denotes the requesting node, and *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 key of *t*. Vs,i_1, Vs,i_2, ..., Vs,i_n denote trust values from *s* to the introducers $i_1, i_2, \ldots, i_n$; while Vi_1,t, Vi_2,t, ..., Vi_n,t denote trust values from introducers $i_1, i_2, \ldots, i_n$ to *t*. Each $V_{s,i*}$ and $V_{i*,t}$ form a pair to establish a single trust path from *s* to *t*.

To compute the new trust relationship from *s* to *t* on a single path, we apply the following equation:

$$V_{s,ik,t} = V_{s,ik} \Theta V_{ik,t} = 1 - (1 - V_{ik,t})^{Vs,ik} \qquad (3)$$

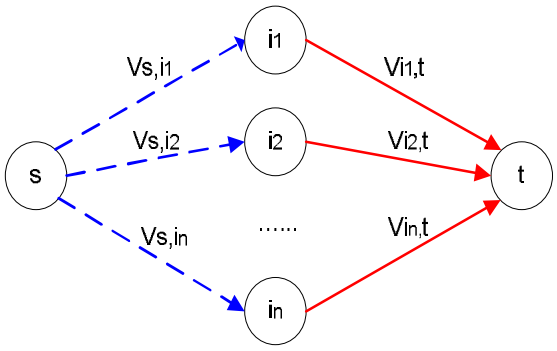

Fig. 3. Trust calculation and combination

The resulting values $V_{s,ik,t}$ from Eq. (3) 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, it can be used as collective information to compute a combined value. The following equation can be applied to combine trust values of the derived trust relationships on different paths:

$$V_t = 1 - \prod_{k=1}^{n} (1 - V_{s,ik,t}) \ , \qquad (4)$$

where *n* denotes the total number of paths. The network model and the trust model build up a hierarchical structure in the ad hoc network and maintain the trust values and trust relationships in a way that Eq. (4), derived from Eq. (2), is more suitable to be applied in our mechanism.

This equation combines trust values $V_{s,ik,t}$ of different paths to give the ultimate trust value V_t of *t*. V_t is the final trust value of *t* in the view of *s* after public key certification. It contains information of trust relationships from *s* to different introducers, and then from the introducers to *t*. Finally, this value will be inserted to the trust table of *s*. If V_t is high, it indicates that *t* is trustworthy and 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.

In our network, each node maintains a repository table for storing trust values of the nodes it knows. These values will be referenced when a node is looking for introducers in public key certifications. Only nodes with the highest trust values in a group will be selected. Normally, the size of a trust table will grow with the network size and time. To make the mechanism more scalable, we can consider the trust table for storing just a certain amount of records. These records may contain nodes with high potential to be introducers; in the meanwhile, they may also contain the list of known malicious nodes in the network. We will also investigate the possibility to remove out-of-date records in our future work.

V. SIMULATION RESULTS

We have implemented our design in the network simulator Glomosim [30]. We evaluate the performance of our system in suppressing false public keys in the replies. We simulate a network that contains 40 nodes which are divided into four groups. Table 2 details the parameters used in our simulations. The network is assigned with a certain percentage *p* of trustworthy nodes at initialization and a certain percentage *m* of malicious nodes. The maximum number of introducers to be selected in each request is three. At least one introducer should give a valid reply in a successful public key certification. The simulation runs for 10000 seconds and totally 800 public key requests are sent out from different nodes.

The values of trust tables are generated randomly at initialization according to the settings of parameters *p* and *m*. Each node initializes its own trust table. In our experiments, a trust value greater than the trust threshold *T* means the node is trustworthy, and vice versa. Each node generates *p* percent of the known values in the trust table to be greater than the threshold. This represents the portion of nodes that are trustworthy to the node itself. Apart from this, nodes in the network have a probability *m* to be malicious. A malicious node usually generates false certificates to harm the network security in these experiments. Since the set of requests on certification are generated randomly, the initial states may impose some

impacts to different sets of requests. However, the interpretation of the experimental results will be the same even with different initial values given that the values are generated according to the parameter *p* and the trust threshold *T*. To get more accurate experimental results, we run each experiment for 10 times and take the average as the result.

TABLE II

SIMULATION PARAMETERS

Network	# of nodes	40
	# of groups	4
	% of trust-worthy nodes at initialization	*p*
	% of malicious nodes	*m*
Public key request	Max # of introducers for each request	3
	Min # of reply for each request	1
Simulation	Time	10000s
	# of query cycles	20
	# of requests per cycle	40
Trust Management	Trust threshold	*T*

A. *Ratings to Percentage of Malicious Nodes*

In this experiment, we evaluate different ratings to the percentage of malicious nodes in the network with the percentage of trustworthy nodes *p* to be fixed at 40% at initialization. The trust threshold *T* is defined as 0.5 during our experiment. It means that 40% of nodes in the trust table will be assigned with a trust value to be greater than 0.5 at initialization. Also, a node is regarded as trustworthy in a certification if its trust value is found to be greater than 0.5. On the other hand, if a node's trust value is less than 0.5 in the view of another node, then that node will not be trusted. Figure 4 depicts the successful rate, failure rate, and unreachable rate on public key certification with the percentages of malicious nodes ranging from 0% to 100%. We find that the successful rate is high in the beginning and it maintains over 50% until the percentage of malicious nodes increases to 80%. The failure rate keeps at a quite low level even the percentage of malicious nodes in the network becomes high. On the other hand, the unreachable rate can be pretty high especially when there are a lot of malicious nodes in the network. The high unreachable rate is because when most of the malicious nodes are identified, the requesting nodes cannot find any introducers to obtain the correct public keys.

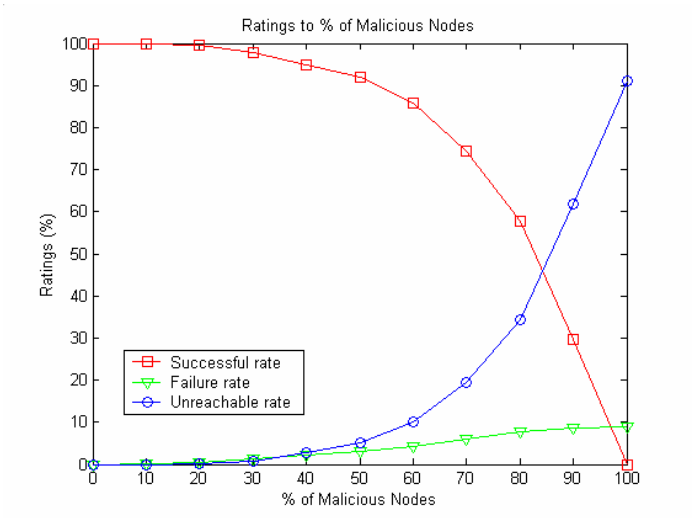

Fig. 4. Ratings to percentage to malicious nodes

B. *Comparisons among Different Mechanisms*

In this experiment, we compare the successful rate and failure rate among the three public key management mechanisms. We fix the number of trustworthy nodes at initialization to be 40% and vary the percentage of malicious nodes from 0% to 100%.

The first mechanism is Pretty Good Privacy [11] with local certificate repositories in individual nodes. A user *s* verifies the public key of user *t* by finding a certificate chain from *s* to *t* in its local certificate repository. The second mechanism, PGP with majority vote, works similarly; but it involves multiple reply messages in a request: Node *s* makes the conclusion on the public key of node *t* by majority voting. The remaining mechanism is the trust- and clustering-based algorithm proposed in this paper.

Figure 5 compares the successful rates among the three mechanisms. It shows that the PGP mechanisms do not achieve a secure system. In these configurations, a node requests for public key certificates of another node by selecting introducers randomly, so their successful rates are low. In our trust- and clustering-based mechanism, on the other hand, each node maintains a trust table and selects introducers with high trust values. Moreover, our public key certificate mechanism can discover and isolate malicious nodes replying with false public key certificates, so it is able to maintain a high successful rate.

Figure 6 compares the failure rate among the above three mechanisms. In the absence of a trustworthy reference for the PGP mechanisms, nodes only select introducers randomly. Malicious nodes thus often succeed in replying false public keys; consequently, the failure rate is very high. With our trust- and clustering-based mechanism, trust values are updated from time to time for maintaining high security in public key authentication. Also, since the dishonest users issuing false certificates are located and isolated, the failure rate is kept relatively low.

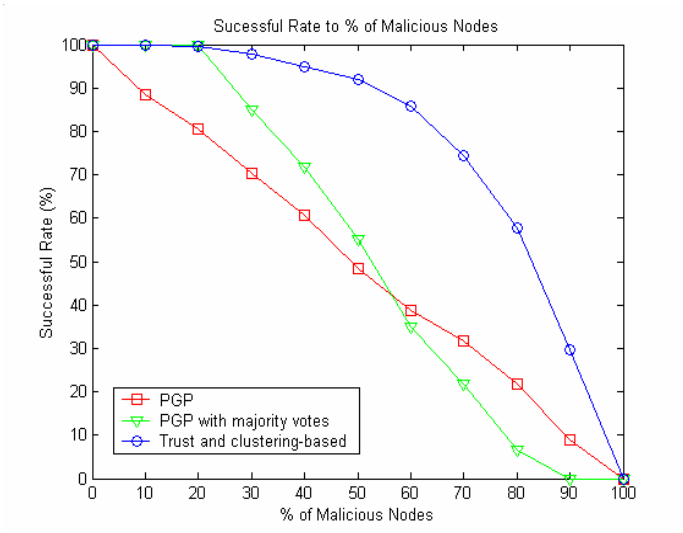

Fig. 5. Comparison on successful rates

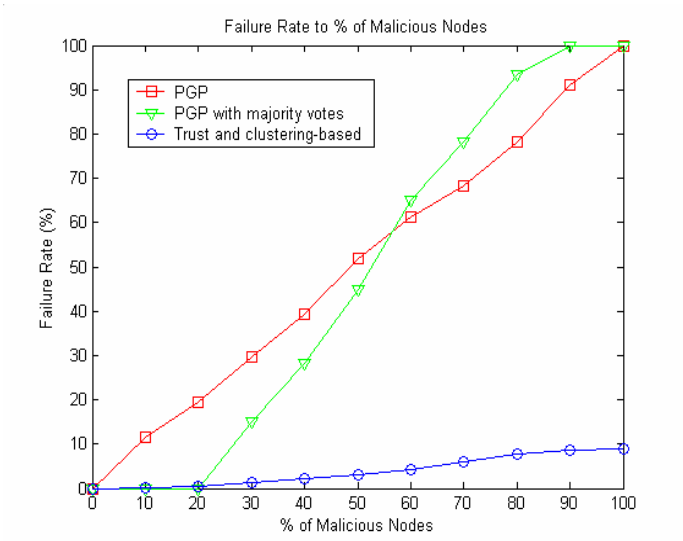

Fig. 6. Comparison on failure rates

VI. CONCLUSION

This paper describes a trust- and clustering-based approach in public key authentication for mobile ad hoc wireless networks. To this end, we propose a trust model that allows nodes to monitor and rate each other with quantitative trust values. We define the network model as clustering-based, such that nodes take advantages of the neighboring monitoring power and short communication distances to their group members. In this work, a trust- and clustering-based public key authentication mechanism is developed. It involves new security operations on public key certification, update of trust table, and discovery and isolation on dishonest users. In addition, we conduct the evaluation of three different approaches in public key authentication to observe their performance and characteristics in providing network security. We compare two PGP-based approaches and the trust- and clustering-based approach we proposed in this paper. With our new mechanism on public key certification, the network endures malicious nodes which issue false certificates. Our approach ensures the security and availability of public key authentication in the inherently insecure and unreliable mobile ad hoc networks.

ACKNOWLEDGMENT

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] 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 4th Annual International Conference on Mobile Computing and Networking (MobiCom'98)*, pp. 85-97, 1998.

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

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

[5] 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, 2003.

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

[7] PKIX Working Group, "Internet X.509 Public Key Infrastructure," draft-ietf-pkix- roadmap-06.txt, 2002.

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

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

[10] "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.

[11] L. Gong, "Increasing Availability and Security of an Authentication Service," *IEEE Journal on Selected Areas in Communications*, vol. 11, no. 5, pp. 657-662, 1993.

[12] T. Wu, M. Malkin, and D. Boneh, "Building Intrusion Tolerant Applications," *Eighth USENIX Security Symposium*, pp. 79-92, 1999.

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

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

[15] 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.

[16] 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, 2001.

[17] 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*, pp. 146-155, 2001.

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

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

[20] 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 the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom'00)*, pp. 32-41, 2000.

[21] T. P. Chen and A. L. Liestman, "Approximating Minimum Size Weakly-connected Dominating Sets for Clustering Mobile Ad Hoc Networks," *The 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computer (MobiHoc '02)*, pp. 164-172, 2002.

[22] T. Ohta, S. Inoue, Y. Kakuda, K. Ishida, and K. Maeda, "An Adaptive Maintenance of Hierarchical Structure in Ad Hoc Networks and its Evaluation," *Proceeding of the 22nd International Conference on Distributed Computing Systems Workshops (ICDCSW '02)*, pp. 7-13, 2002.

[23] J. Li and W. Jia, "Traffic Analysis in Ad Hoc Networks Based on Location-Aware Clustering," *Proceeding of the 23rd International Conference on Distributed Computing Systems Workshops (ICDCSW '03)*, pp.503-507, 2003.

[24] 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.

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

[26] M. K. Reiter and S. G. Stubblebine, "Resilient Authentication using Path Independence," *IEEE Transactions on Computers*, vol. 47, no. 12, pp. 1351-1362, 1998.

[27] K. Aberer and Z. Despotovic, "Managing Trust in a Peer-2-Peer Information System," *Proceedings of the 10th International Conference on Information and Knowledge Management (CIKM01)*, pp. 310-317, 2001.

[28] K. Aberer, "P-Grid: A self-organizing access structure for P2P information systems," *Proceeding of the 9th International Conference on Cooperative Information Systems (CoopIS 2001)*, pp.179-194, 2001.

[29] S. D. Kamvar, M. T. Schlosser, and H. G.-Mollina, "The EigenTrust Algorithm for Reputation Management in P2P Networks," *The 12th International World Wide Web Conference*, pp. 640-651, 2003.

[30] X. Zeng, R. Bagrodia, 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)*, pp. 154-161, 1998.

Edith C.H. Ngai (S'02) received the B.Eng. degree in computer engineering from the Chinese University of Hong Kong in 2002. She is currently an Mphil student in computer science & engineering at the same University. Her research interest includes network security, wireless communication, and multimedia.

Michael R. Lyu (S'84-M'88-SM'97-F'04) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1981, the M.S. degree in computer engineering from University of California, Santa Barbara, in 1985, and the Ph.D. degree in computer science from University of California, Los Angeles, in 1988. He is currently a Professor in the Department of Computer Science and Engineering, the Chinese University of Hong Kong. He was with Jet Propulsion Laboratory, University of Iowa, Bellcore, and Bell Labs. His research interests include software reliability engineering, distributed systems, fault-tolerant computing, wireless communication networks, Web technologies, digital libraries, and E-commerce systems. He has published over 170 refereed journal and conference papers in these areas. He has been an editor of IEEE Transactions on Reliability, IEEE Transactions on Knowledge and Data Engineering, and Journal of Information Science and Engineering. Professor Lyu is a Fellow of the IEEE.