Highly Reliable Trust Establishment Scheme in Ad hoc Networks

Kui Ren[1], Tieyan Li[2], Zhiguo Wan[2],
Feng Bao[2], Robert H. Deng[2], and Kwangjo Kim[1]

[1] *International Center for Information Security, Information and Communication University, Daejeon, Korea 305732*

[2] *Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613*

Abstract

Securing ad hoc networks in a fully self-organized way is effective and light-weight, but fails to accomplish trust initialization in many trust deficient scenarios. To overcome this problem, this paper aims at building well established trust relationships in ad hoc networks without relying on any predefined assumption. We propose a probabilistic solution based on distributed trust model. A secret dealer is introduced only in the system bootstrapping phase to complement the assumption in trust initialization. With it, much shorter and more robust trust chains are able to be constructed with high probability. A fully self-organized trust establishment approach is then adopted to conform to the dynamic membership changes. The simulation results on both static and dynamic performances show that our scheme is highly resilient to dynamic membership changing and scales well. The lack of initial trust establishment mechanisms in most higher level security solutions (e.g. key management schemes, secure routing protocols) for ad hoc networks makes them benefit from our scheme.

Key words: ad hoc networks, security, trust management, distributed trust model

1 Introduction

Although the security objectives of both ad hoc networks and traditional networks are considered the same such as availability, confidentiality, integrity, authentication, and non-repudiation, the security issues involved in ad hoc networks are quite different due to the "mobile" and "ad hoc" constraints, i.e. limited computing and communicating resources, dynamic network topology as well as the roaming nature. These features make it hard for building trust relationships for securing ad hoc network. *The notion of "trust" among entities (e.g., domains, principals, components) is engaged in various protocols as a set of relationships established on the basis of a body of supporting assurance (trust) evidence and required by specified policies (e.g., by administrative procedures, business practice, law)* (7). In traditional networks, most trust evidences are generated via potentially lengthy assurance processes, distributed off-line, and assumed to be valid on a long term and certain at the time when trust relations derived from it are exercised. In contrary, few of these characteristics of trust relations and trust evidence are prevalent in mobile ad hoc networks. Since those solutions developed mainly for the fix wireline networks (3; 12) are not fit in such a scenario, new security solutions are eagerly in demand.

There are many approaches studying ad hoc network security (14; 8; 10; 9; 4; 16; 5). The significant efforts done so far are mainly the adaption from the existing distributed trust model to ad hoc trust model. Among them, one trend is to set up an ad hoc network with the help of Certificate Authority (CA) or Key Distribution Center (KDC) at the bootstrapping phase. A CA/KDC is responsible for setting up the foremost trust relationships among all the nodes by distributing keys or certificates (16). Then, CA's functionality is substituted by t sub-CAs using threshold cryptography, and these t sub-CAs will issue partial certificates afterwards. However, this strategy suffers from difficulty on collecting t certificates efficiently. Notice this limitation, the authors of (5; 9) adopted a different strategy – a fully self-organized way. They developed a PGP-like distributed trust establishment scheme where the social relationships among nodes become the trust evidence for building trust. The scheme is quite effective and scalable for ad hoc networks. Unfortunately, since it is based on the assumption of social relations of ad hoc nodes and social relations are not always profiling the ad hoc feature accurately. Thus, it fails to establish sufficient trust relations in certain distributed trust deficient scenarios.

In this paper, we propose a modified distributed trust establishment approach. A secret dealer is introduced only in system bootstrapping phase to simplify the process of trust initialization. With the help of the secret dealer, sufficient trust relationships are constructed without relying on any specific assumption, such as the assumption of sufficient social relationship among nodes in (5; 9). As for sufficient trust relationships, we mean every two nodes in the ad hoc network can authenticate each other at an almost certain probability via a chain of trust relationships (trust chain) and moreover, the number of independent trust chains should be at least larger than two with same high probability. At the same time, the average length of the trust chains should be as short as possible to provide efficiency and avoid trust dilution. The trust relationships can then be maintained by applying a light-weight node joining and leaving approach. We also provide a probabilistic analysis based on a trust graph to prove the rational of our scheme. The simulation results show our contributions in three aspects clearly:

(1) The average length of the shortest indirect trust paths is decreased significantly (around 2 compared with 4-6 in (6)). Note that the less the intermediate nodes in a trust path, the less the risk involved in trusting between remote nodes.
(2) The average number of the shortest indirect trust paths is guaranteed (at least 2 with same high probability). Thus we assure that the trust relationship can be built robustly and sufficiently.
(3) The scheme is slightly affected by the dynamic joining/leaving of (up to 50%) nodes. Therefore, it is highly resilient to dynamic feature of ad hoc networks.

The remaining of this paper is organized as follows: Section 2 introduces the related works. In Section 3, our trust establishment scheme is elaborated and Section 4 gives the mathematical analysis of our scheme. Section 5 describes the simulation and its results. At last, we conclude the paper and point out the future works.

2 Related Works

This section reviews several security solutions for ad hoc networks (14; 5; 9; 16; 10; 4; 8). Specially, we investigate the typical distributed trust model, an ad hoc trust model based on threshold cryptography and a fully self organized trust model, since these models demonstrate the gradual progress in the literature.

2.1 Distributed Trust Model

Understanding the term "trust an entity", we firstly distinguish the two important concepts of "target of trust" and its "classification". "Target of trust" is the actual entity we trust, while the "classification" describes exactly what the entity is being trusted for. Furthermore, there might be a value of trust which describes how much we trust an entity according to some criteria. Trust for distributed systems should as much as possible be based on trust evidence to prevent irrational trust based on faith. And there can be a hierarchy of trust relationships. Trust can be derived by establishing a direct trust relationship using an indirect path. The Distributed Trust Model (1) is a decentralized approach to trust management and uses a recommendation protocol to exchange trust-related information. This model describes how to establish trust relationships and can be used for various conditions on top of security models like public-key systems. It is applicable to any distributed system and isn't specifically target for ad hoc networks. For instance, PGP system (17) is a typical public key certificate based example.

2.2 Distributed CA Approach Using Threshold Cryptography

L.Zhou and Z. J. Haas (16) first introduced a key management system for ad hoc networks based on threshold cryptography. In their solution, a group of n servers together with a master public/private key pair are firstly deployed by CA. Each server has a share of the master private key and stores the key pairs of all nodes. The shares of master private key are generated using threshold cryptography. Thus, only n servers together can form a whole signature. For any node wanting to join the network, it must first collect all of the n partial signatures. Then the node can compute the whole signature locally and thus get the certificate.

J. Kong, et al (10) gave an extended system. During the network bootstrapping phase, a centralized dealer is needed to issue certificates and private key shares to at least t nodes. A threshold cryptography system is also deployed in order to provide a (t, n) secret sharing service. Any t nodes can form a centralized dealer and can thus issue or revoke certificates. In this scheme, a to-be-member node will collect t partial signatures in its local communication range. This method facilitates the joining and leaving of nodes to some extent, but also increases the risk of private key leaking of the centralized dealer. Any t nodes' being compromised or collusion will break down the whole system.

2.3 Self-organized Certificate-based Trust Model

In order to set up a fully self organized system, S. Capkun and J. P. Hubaux (5; 15) adopted small world model (6) into mobile ad hoc networks. They used a PGP-like mechanism to initialize the system. Trust between nodes is established through secure side channel communication (e.g. physical contact, infrared communication). They assume that the social relationships among mobile ad hoc network members are essentially the same as those of in PGP system, which is the only realistic model in which social relationships in real society result in trust establishment. Thus, the trust relationships formed by mobile ad hoc network members must exhibit the same features as PGP system (6). A public key certificate based approach is also adopted by this scheme. Every node issues certificates to those it trust from its own domain (i.e, signed by its own private key). Due to the small world phenomenon shown in social relationships, nodes can thus authenticate each other with acceptable length of trust relationship chain. They also studied in details on the mechanisms (such as certificate exchange, shortcut hunter algorithm, etc.) to facilitate the initialization and authentication process (9; 5). A to-be-member node can contact with as many nodes as possible (according to its relationship with other nodes) and get their certificates to join the network, and thus can be authenticated by other nodes in the network.

2.4 Trust Establishment Drawbacks in These Schemes

The aforementioned security solutions suffer from many drawbacks. In the distributed CA scheme (10), J. Kong, et al mentioned that the trust between a to-be-member node and t member nodes in its neighborhood can be established by out-of-bound physical proofs, such as human perception or biometrics. However, we can find that this method is far from practical. It is obviously impossible for a node acquiring t nodes to trust it in its local communication range, because the trust evidence should be evaluated and authenticated, or there should exist off-line trust relationships between this node and the t member nodes. In an infrastructureless ad hoc network environment, the evaluation of trust evidence will be very hard. As for off-line trust relationships, the demand for a to-be-member node having off-line trust relationships with t nodes in its neighborhood is also impossible, due to the ad hoc nature. Besides, this kind of scheme suffers from high communication and computation overhead (14).

In the self-organized scheme, trust establishment is solely coming from off-line trust relationships. These off-line trust relationships are generated from general social relationships. The initialization process depends on nodes themselves to issue certificates, and thus forms a network of trust relationships. This process is actually very complex and slow in practical, because every issued certificate will require close contact between two nodes. Further, ad hoc networks are formed haphazardly by member nodes, so the trust relationships among member nodes are much sparse than those of general society. This drawback is fatal to the establishment of sufficient trust relationships in ad hoc networks. On the other hand, even full social trust relationships could be assumed, it still not enough to build sufficient trust relationships in ad hoc networks. In PGP certificate graph (6) of year 2001, the size of largest strongly connected component is only 14000, but the whole graph size is 800,000. This means most nodes can not authenticate each other. Note that PGP system (17) is the only example in the real world which reflects full social trust relationships in society. Also in this scheme, the nodes are assumed to locally store as many certificates as possible. This will cause a big problem due to limited storage capability of most nodes except for few high-end devices (e.g. PDAs, notebooks). As indicated in Section 2, a certificate at the minimum contains a) node identifier and its corresponding public key; b) a signature. In case of

2048-bit RSA keys, it will be at least more than 512 bytes for each certificate. At the same time, even for high-end mobile phones the totally storage capacity could be at most a few mega-bytes, which are used for storing the application data of all kinds and not solely for certificates.

3 Our Trust Establishment Scheme

The goal of our scheme is to establish sufficient trust relationships in ad hoc networks with minimum local storage capacity requirements on the mobile nodes. The scheme should be resilient to nodes' dynamically joining/leaving and scale well at the same time. Without lose of generality, we draw some basic assumptions: Every node has its own private/public key pair and the binding of the node with its public key is known to the chosen secret dealer; There are sufficient trust evidences between nodes and the secret dealer, and the nodes will trust the secret dealer.

3.1 Trust Establishment in the Bootstrapping Phase

In order to simplify the initialization process and establish sufficient trust relationships in the network, a secret dealer is introduced in the bootstrapping phase. The advantage of a centralized secret dealer is that the trust relationships between nodes and secret dealer are out of question, as the trust can be based on some long term trust evidences. The secret dealer could be a telecommunication service provider common to all current member nodes, which has long-term well established trust relationships with current member nodes and properly keeps the bindings of its served nodes and their corresponding public keys.

During the bootstrapping phase, every current member node gets its own secret short list from the secret dealer and stores it locally. The secret short list is pre-computed by the secret dealer and could be distributed to each current member node in parallel to speed up the process. The secret short list includes k entries and the value of k is determined according to the group size n and may vary slightly from node to node. Each entry contains a binding of a node identifier and its corresponding public key: (ID, P_K). After the secret short lists are properly distributed, the following conditions will be met (let SL_i denote the secret short list obtained by node i):

- Every current member node receives a secret short list SL, which contains k semi-randomly selected bindings, i.e., a set of (ID, P_K) pairs with cardinality equals to k;

- The bindings are distributed symmetrically. If node i gets the binding corresponding to node j and its public key in the secret short list, then the binding of node i to its own public key is also included in that of the node j's, that is, $(ID_i, P_{K_i}) \subset SL_j$, if $(ID_j, P_{K_j}) \subset SL_i$.

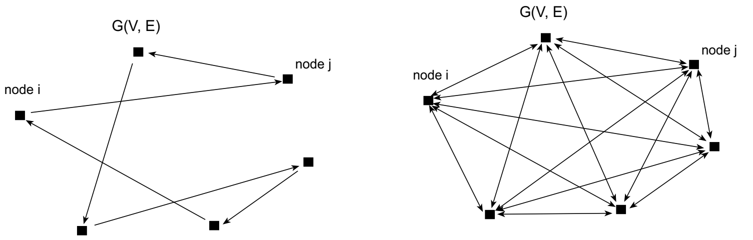

Fig. 1. Trust Chains with different values of k in an ad hoc network

The second part of the bootstrapping phase is a certificate issuing process. After all the current member nodes get their own secret short lists, each of them begins to generate certificates for the received bindings from its own domain and also store it locally. The certificate issued by a node i at the minimum contains the following contents: a) a particular (ID, P_K) binding selected from the obtained secret list; b) valid interval; c) a signature signed on a) and b) using node i's own private key. Therefore, each current member node issues exactly k certificates according to its obtained secret short list. Thus, a network of trust relationships is formed. The value of k is carefully chosen so that there will be sufficient trust relationships in the network. In Figure 1, we show two different situations corresponding to different chosen value of k. When k equals to 1, then the possibility for n nodes to establish a strong connected certificate graph (as defined in Section 4) equals to $(n-1)!/(n-1)^n$, which implies that the chance for any two member nodes to accomplish a mutual authentication process is extremely low. When k equals to $n-1$, we can see all the nodes are fully connected. However, every node is required to store the bindings of all the n current member nodes (including its own), and therefore, the demanding storage capacity, which is limited in mobile nodes, is linear to the network size n. In Section 4 we will show how to determine the value of k and the detail of assigning algorithm. Our assigning scheme guarantees that the probability of any two nodes that can find a trust chain to authenticate each other is "almost certain". At the same time, the value of k is kept to be as small as possible under the given network size n to meet the storage constraints in mobile nodes. A good balance is reached between the required storage capability and the sufficiency of the trust relationships. This is assured by our concrete mathematical analysis in Section 4.

After the system bootstrapping phase is finished, the ad hoc network becomes fully functional, and thus no infrastructures can be expected, i.e., the secret dealer will not exist any more. To address this point, a distributed trust establishment approach is adopted at this phase to accommodate the dynamic membership changing in ad hoc networks. We assume that there are sparse social relationships existed among nodes. By sparse social relationships, we mean that any to-be-member node can establish independent trust relationship with at least two current member nodes.

From this starting point, any to-be-member node should first establish trust relationships between current member nodes and itself, that is, obtaining enough certificates from current member nodes before joining the network. The number of certificates required is at least two. Due to the mobility of the nodes and the sparse social relationship existed among nodes, it is reasonable to require at least two independent certificates to be hold by a to-be-member node. Although we set two as the coarse-grained threshold number, a higher value can be set if necessary and possible (because the number of certificates that could be acquired is determined by actually existed social relationships among nodes). If this is done correctly, then this to-be-member node now becomes a formal network member. In PGP system, getting at least two independence certificates is also assumed as a requirement for a node to be marginally trusted. Note that the required two certificates must be checked to make sure that the signers are neither on the leaving list nor on the revocation list.

An alternate approach to facilitate a to-be-member node to joining process is as follows: A to-be-member node first contacts any current member node within its communication range (one-hop range) to request for joining the network. Any of the current available member nodes can reply the request and handle it. The correspondence member node then communicates with other network member nodes to authenticate the trust evidence provided by this to-be-member node. This operation can be a broadcast, which relays the message (trust evidence information) obtained from the to-be-member node. If the trust evidence is authenticated by a current member node, then the to-be-member node obtains the certificate signed by this node. This process is repeated until the to-be-member node gets at least two independent certificates.

The combination of the two methods can make it efficient to accomplish joining process of a to-be-member node. The certificates issued by the member nodes in this phase is the same as those issued in the bootstrapping phase. The issuing procedure is as follows: A member node issues a certificate that binds the target node's ID and public key together from its own domain once the authentication on the target node is successfully accomplished. Symmetrically, the target node also issues a certificate to that member node from its own domain at the same time and store it locally. After the to-be-member node collects enough certificates and becomes a formal member node, it begins to issue certificates from its own domain according to the available information. It may copy the certificate information stored by its trusted nodes and issue the same certificates but from this own domain. At this point, the node is fully connected to the network.

As for the leaving process, a leaving node may broadcast its leaving information and signs on it so that all other member nodes will revoke all the certificates issued to it after verifying the message. The leaving action of a node may also be detected and reported by any other member node. The local detection leads to a broadcast message sent to all other member nodes. The broadcasted information then will be authenticated by each node independently and if it's true, the certificates issued to this node will be revoked after a reasonable delay. The delay is roughly corresponding to network information convergence time to collect potential disputing message and therefore prevents potential cheating attack. After a node leaves the network, no valid certificate or certificate chain can reach it any more.

4 Mathematical Model and Analysis

In this section, we use a directed graph to evaluate the effect of parameter k, which is defined in subsection 3.1. We assume that a certificate digraph $G(V, E)$ represents the public keys and the certificates of the network system, where V and E stand for the set of vertices and the set of edges, respectively. The vertices of the certificate digraph represent the bindings of public keys and their corresponding IDs and the edges represent the certificates. A directed edge from node i (Public key of node i) to node j (Public key of node j) will exist if there is a certificate signed with the private key of i that binds the the identifier of node j and its public key P_{K_j} (as denoted in Figure 1). A certificate chain from node i to node j is represented by a directed path from vertex i to vertex j in G. So if any two nodes in a certificate digraph is connected, then the trust between the two nodes will be established. The trust chain is thus represented by the certificate chain.

The problem of establishing sufficient trust chains among nodes is now reduced to a connectivity problem in a digraph. We first depict our problem formally: Let p be the probability that a certificate exists between any two nodes, n be the number of network nodes, and $d = p \cdot (n - 1)$ be the expected degree of a node (i.e., the average number of directed edges linking that node with its graph neighbors with direction either in or out), what value should the expected average degree of a node, d, be so that every two nodes in the ad hoc network can authenticate each other at an almost certain probability via a chain of trust relationships (trust chain) and moreover, the number of independent trust chains should be at least larger than two with same high probability? Using the digraph defined above, the first part of the problem is restated as: in a random digraph, what value should the expected average degree of a vertex d, be so that the digraph is strongly connected?

4.1 A Semi-Random Digraph Model for Our Scheme

To answer the above problem, we slightly modify our problem in order to get a simple mathematical model. In a predefined, semi-random digraph, what value should the expected average degree of a vertex d, be so that the digraph is strongly connected?

In order to understand this problem, we first introduce a connectivity model of random graph. A random graph $G(n, p)$ is a graph of n vertices for which the probability that a link exists between two vertices is p. Erdös and Rènyi (11) showed that, for monotone properties, there exists a value of p such that the property moves from "nonexistent" to "certainly true" in a very large random graph. The function defining p is called the threshold function of a property. They showed that if $p = \frac{ln(n)}{n} + \frac{c}{n}$, with c any real constant then

$$\lim_{n \to \infty} Pr[G(n,p) connected] = e^{-e^{-c}}.$$

Therefore, given n we can find p and $d = p \cdot (n-1)$ for which the resulting graph is connected with desired probability $Pr[G(n,p) connected]$. Figure 2 illustrates the plot of the expected degree of a node d as a function of the network size n for various values of $Pr[G(n,p) connected]$. The figure shows that, to increase the probability that a random graph is connected by one order, the expected degree of a node increases only by 2. Moreover, the curves of this plot are almost flat when n is large, indicating that the size of the network has insignificant impact on the expected degree of a node required to have a connected graph. For example, to make sure that the graph is connected with probability 0.999999, the average degree of a node needed is only no more than 24 with n equals to 20000.

Based on the discussion above, we now construct a digraph with same connectivity feature. In our method, each edge in the (undirected) random graph, will be viewed as two directional edges so that the directly connected nodes in (undirected) random graph can still keep mutual directly connection in the

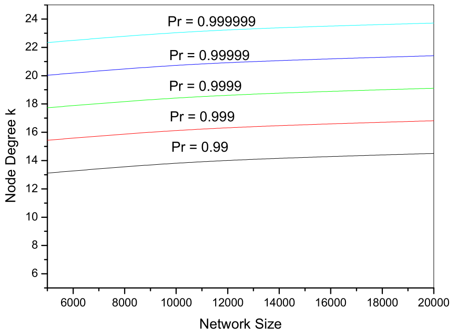

Fig. 2. Expected degree of node vs. network size under various $Pr[G(n,p) connected]$

resulted digraph. Figure 3 illustrates this process. In this way, $dn/2$ edges will be added and totally edges will be dn. Thus the average degree ($\overline{d}$) of the node in the digraph will be twice as that of (undirected) random graph: $\overline{d} = 2d$. At this point, we have obtained a semi-random digraph with average degree of node equals to $2d$ whose connectivity feature is the same as above (undirected) random graph. It is easy to see that the value of k represents the number of out edges of a given node, which equals to $\overline{d}/2 = d$. We can thus decide the value of k and the assigning algorithm by using this semi-random digraph model.

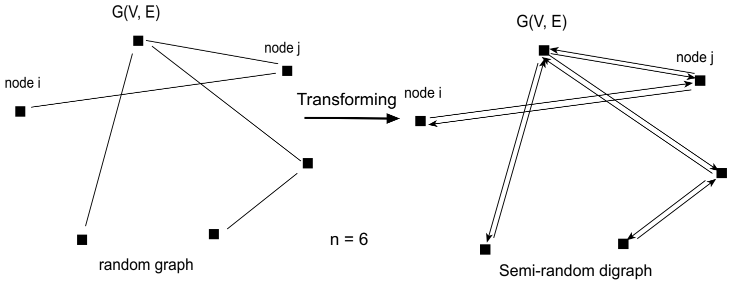

Fig. 3. Transforming a random graph to a semi-random digraph

For an ad hoc network of size 20000, at most 25 semi-random selected (ID, P_K) bindings are needed for every node to assure that any two nodes can establish at least one trust chain at probability 0.999999. So at this point, we have got the value of k, which is a threshold number to make sure that the certificate graph is almost certainly strong connected without relying on any assumption. Therefore, we have answered the first part of the question.

4.2 Analysis of the Sufficiency of Trust Relationships

The second part of the problem is how to provide at least two independent trust chains between every two nodes. We give a probabilistic solution to this problem in this subsection. We first define two parameters. In a certificate digraph G, if node i and j are directly strongly connected, we say there is a direct trust path $path_d$ between node i and j; if node i and j are strongly connected via only one middle node, we say there is a shortest indirect trust path $path_s$ between node i and j.

Given the value of k determined in above subsection, we first estimate the existing probability of $path_d$ and $path_s$, namely, p and p_s in the ad hoc network. Note that in above section, p has been introduced. And because the value of k is very low relative to network size n, the value of p is very low, though the value of $Pr[G(n,p)connected]$ is very large. It is obviously that the value of p is linear to the value of k.

In our semi-digraph model, one shortest indirect path between two nodes means these two nodes hold at least one same (ID, P_K) binding. So the probability p_s can be explained as the probability that two nodes share at least one common (ID, P_K) binding. At the same time, if two nodes hold more than 2 same (ID, P_K) bindings, then there will be at least two independent shortest indirect trust paths between the two nodes. This means the trust between two nodes can be verified through at least two independent shortest indirect trust paths. The probability of existing no less than two shortest trust paths (p_{s2}) can be calculated by following calculation.

$$p_{s2} = 1 - P(2 \text{ nodes share no common bindings}) - \\ P(2 \text{ nodes share only one common binding})$$

$$= 1 - \frac{C_n^k \cdot C_{n-k}^k}{C_n^k \cdot C_n^k} - \frac{C_k^1 \cdot C_n^k \cdot C_{n-k}^{k-1}}{C_n^k \cdot C_n^k} = 1 - \frac{((n-k)!)^2}{(n-2k)!n!} - \frac{k^2((n-k)!)^2}{(n-2k+1)!n!}$$

Using Stirling's approximation: $n! \approx \sqrt{2\pi} n^{n+1/2} e^{-n}$, the simplified equation will be

$$p_{s2} = 1 - \frac{n+(k-1)^2}{n-2k+1} \cdot \frac{(1-k/n)^{2(n-k+1/2)}}{(1-2k/n)^{(n-2k+1/2)}}.$$

Figure 4 presents the curves as the changing of p_{s2} according to k under different network size n. From the figure, we can see that when $k = 100$ ($n = 10000$), the probability of two nodes having at least two shortest indirect trust paths will be around 0.26, i.e. $p_{s2} \approx 0.26$. When $k = 150$ ($n = 10000$), $p_{s2} \approx 0.66$. And When $k = 250$ ($n = 10000$), $p_{s2} \approx 0.988$. So under the network size of 10000, each node only needs to store around 200 certificates locally to establish sufficient trust chains with ideal sufficiency exists in ad hoc networks. So if we set 0.9 as threshold value of p_{s2}, then at most 250 key pairs should be stored by each node under network size 10000. Thus, we have decided the value of k under different network sizes and answered the whole question.

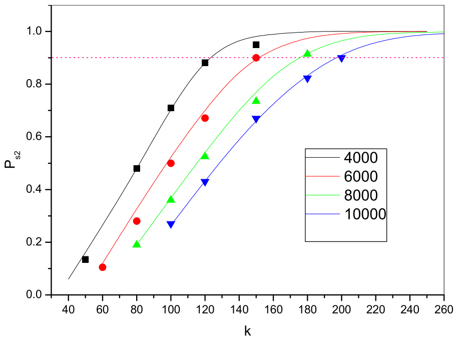

Fig. 4. p_{s2} vs. k under different ad hoc network size n

5 Simulation Results

With the theoretical analysis shown above, we evaluated the performance of our trust establishment scheme and the dynamical characteristic of the scheme. All simulations were implemented in C/C++ with LEDA (Library of Efficient Data Types and Algorithms) (2).

5.1 Performance of Assigning Scheme

We implemented our trust establishment scheme by generating semi-random certificate graph G. The degree k of vertices is chosen according to the mathematical model discussed in Section 4. Different values of k are used to indicate the performance variation of trust sufficiency among nodes. We generated various certificate graphs for different network size to test the performance of trust sufficiency. We repeated the experiment 10 times for each network size. We choose network size of 4000 as our starting point because we believe that the results getting from the larger network size is applicable to the smaller ones, because the underlying storage capacity is always the same and it will not be a problem to handle the storage constraints.

Two parameters are used to evaluate the performance of trust sufficiency. One is $PATH_{leng}$, the average length of the shortest path of digraph G, which is similar to the directed characteristic length defined by (6). The average length of the shortest path of digraph G is defined as the median of the means of the directed shortest path lengths strongly connecting each vertex to all other vertices in G. The other parameter $PATH_{num}$ is defined as the average number of shortest indirect paths of digraph G. The average number of shortest indirect paths of digraph G is defined as the median of the means of the number of indirect shortest trust paths strongly connecting each vertex to all other vertices in G.

The scatter points in Figure 4 denote the actual values of p_{s2} versus k under

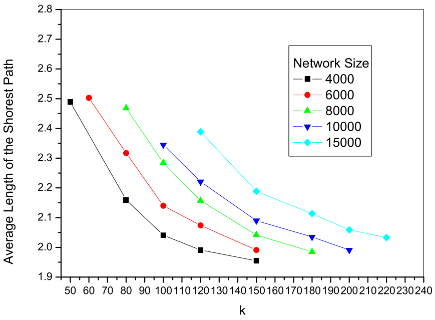

Fig. 5. $PATH_{leng}$ vs. k under different ad hoc network size n

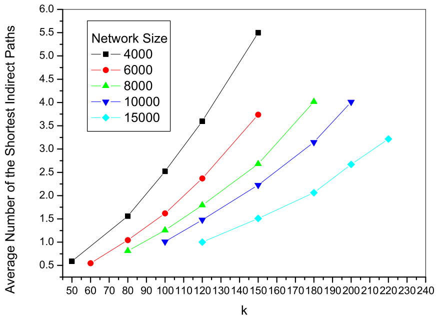

Fig. 6. $PATH_{num}$ vs. k under different ad hoc network size n

different network sizes, obtained from experiment data. The simulation results closely conform to the mathematical model as shown in the same figure (indicated by solid lines with different colors). Figure 5 indicates that the average shortest path between two nodes. It shows that as k increases, the average shortest path between two nodes $PATH_{leng}$ drops to around 2. Compared with the results shown in Figure 11 of (6), $PATH_{leng}$ decreases nearly [1/3–1/2] (from 4-6 to around 2), which is a significant improvement considering ad hoc property.

Figure 6 shows the actual number of shortest indirect trust paths $PATH_{num}$ in graph G. It shows that for two nodes in a certificate digraph G to have larger than 2 shortest indirect trust paths, the needed value of k is relatively small (i.e. $1.2\% - -2.2\%$ for network size from 10000 to 4000). E.g. for a network size of 10000, only 1.5% nodes are needed. The larger the network, the less the ratio of k to network size. Thus, the scheme is quite applicable in large scale of ad hoc network.

5.2 Dynamical Characteristic

We also used two test groups for the examination of the dynamical characteristic of our trust management scheme. Firstly, we examined the joining and leaving process separately. When nodes leaving process is examined, we assume that up to 20 percent members would leave. As for joining process, the joining nodes will be up to 30 percent of the network size. Secondly, we simulated a dynamical joining and leaving process while keeping the network size roughly unchanged, and the performance of the network were examined after each given time interval.

Table 1 shows the experimental results after we removed m =500, 1000, 1500, 2000 nodes randomly. We see that the network still have a good performance even after 20 percent nodes leaving. The simulation is conducted on a network with size $n = 10000$. Compared with the network of size 8000, the performance

m	$PATH_{num}$			$PATH_{leng}$		
	k = 100	k = 120	k = 150	k = 100	k = 120	k =150
0	1.01	1.48	2.23	2.35	2.22	2.09
500	0.94	1.36	2.11	2.37	2.25	2.08
1000	0.90	1.29	2.01	2.38	2.27	2.10
1500	0.84	1.24	1.89	2.42	2.28	2.12
2000	0.81	1.16	1.79	2.44	2.29	2.13
8000*	0.81	1.25	1.79	2.47	2.28	2.16

Table 1
The network performance after nodes leaving (with $n = 10000, m \leq 2000$)

k	$PATH_{num}$		$PATH_{leng}$	
	before joining	after joining	before joining	after joining
80	1.04	1.01	2.32	2.32
100	1.62	1.64	2.14	2.16
120	2.37	2.42	2.07	2.06
150	3.74	3.77	1.99	1.99

Table 2
The network performance after nodes joining (with $n = 6000, m = 2000$)

is basically the same. So the trust sufficiency is tolerant to up to 20% nodes leaving without affecting network performance.

Table 2 shows similar results for nodes' joining. In our scheme, nodes joining does not have significant influence on network performance.

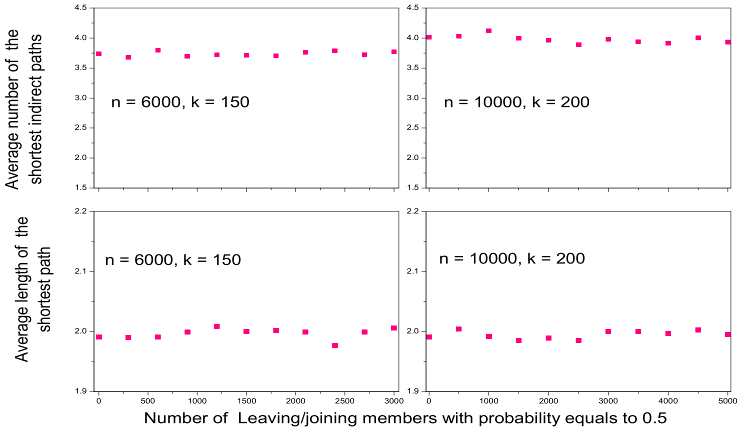

Fig. 7. $PATH_{num}/PATH_{leng}$ vs. dynamically changing k

The dynamic joining and leaving process is simulated by processing random joining or leaving of each individual node. The probability of joining and leaving of a node equals to 0.5. This means that nodes' leaving and joining are of the same chance. We examined the network performance at each 500 interval and get the results as shown in Figure 7. The whole process ended when network has experienced 5000 times joining/leaving actions. We can see that the network performance actually has little change. This result clearly shows that our trust management scheme is very robust in a dynamic environment.

6 Concluding Remarks

In this paper, we proposed a robust trust establishment scheme for securing ad hoc networks. This scheme is actually a modified distributed trust establishment approach based on public-key cryptography. We have shown that with the help of a secret dealer, the trust establishment issue in bootstrapping phase can be solved in a simple way. And the adoption of distributed trust establishment model leads to a light-weight trust establishment solution to node dynamically joining/leaving, which is conform to the very nature of ad hoc networks. To formally analyze the problem, we came out with a probabilistic method based on a trust digraph model. The simulation results on both static and dynamic performances of our trust establishment scheme show that the trust establishment and maintenance in this scheme can be fulfilled very well. Our scheme deals with the fundamental trust establishment problem, it can serve as the building block for higher level security solutions such as key management schemes or secure routing protocols. In the near future, we would like to test our scheme into certain real ad hoc systems like mobile ad hoc networks and analyze the system performances.

References

[1] A. Abdul-Rahman and S. Hailes, "A Distributed Trust Model", New Security Paradigms Workshop 1997, ACM, 1997.

[2] Algorithmic Solutions, "LEDA", www.algorithmic-solutions.com.

[3] D. Balfanz, D. K. Smetters, P. Stewart and H. Chi Wong, "Talking to Strangers: Authentication in Ad-Hoc Wireless Networks", NDSS '02, San Diego, 2002.

[4] L. Buttyan and J. P. Hubaux, "Report on a Working Session on Security in Wireless Ad Hoc Networks", Mobile Computing and Communications Review, Vol. 6, 2002.

[5] S. Capkuny, L. Buttyan and J. Hubaux, "Self-Organized Public-Key Management for Mobile Ad Hoc Networks", EPFL Tech. Report, 2002.

[6] S. Capkuny, L. Buttyan and J. Hubaux, "Small Worlds in Security Sys-

tems: an Analysis of the PGP Certificate Graph", New Security Paradigms Workshop 2002, Norfolk, 2002.

[7] L. Eschenauer, V. D. Gligor and J. Baras, "On Trust Establishment in Mobile Ad-Hoc Networks", Proc. of the Security Protocols Workshop, Cambridge, 2002.

[8] L. Eschenauer, V. D. Gligor, "A Key-Management Scheme for Distributed Sensor Networks", Proc. of the 9th ACM Conference on Computer and Communication Security, Washington D.C., November, 2002.

[9] J. 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, 2001.

[10] J. Kong, P. Zerfos, H. Luo, S. Lu and L. Zhang, "Providing Robust and Ubiquitous Security Support for Mobile Ad-hoc Networks", ICNP'01, 2001.

[11] J. Spencer, "The Strange Logic of Random Graphs", Algorithms and Combinatorics 22, Spring-verlag 2000, ISBN 3-540-41654-4, 2000.

[12] F. Stajano and R. Anderson. "The Resurrecting Duckling: Security Issues for Adhoc Wireless Networks", LNCS 1796, Springer-Verlag, 1999.

[13] J. Undercoffer, S. Avancha, A. Joshi, and J. Pinkston, "Security for Sensor Networks", CADIP Research Symposium, 2002.

[14] A. Weimerskirch and G. Thonet, "A Distributed Light-weight Authentication Model for Ad-Hoc Networks", ICISC 2001, Seoul, December, 2001.

[15] S. Yi and R. Kravets, "Key Management for Heterogeneous Ad Hoc Wireless Networks", 10th IEEE International Conference on Network Protocols, ICNP 2002.

[16] L. Zhou and Z. J. Haas, "Securing Ad Hoc networks", IEEE Networks Special Issue on Network Security, December, 1999.

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