Trust-group-based authentication services for mobile ad hoc networks

Chih-Peng Chang
Department of CSIE
National Taiwan University
Taipei, Taiwan, 106
Email: r92076@csie.ntu.edu.tw

Jen-Chiun Lin
Department of EE
National Taiwan University
Taipei, Taiwan, 106
Email: simon@orchid.ee.ntu.edu.tw

Feipei Lai
Department of EE & CSIE
National Taiwan University
Taipei, Taiwan, 106
Email: flai@ntu.edu.tw

***Abstract*— In recent years, mobile ad hoc networks have received more attention, because of their easy deployment. However, the characteristics of mobile ad hoc networks are more prone to physical security threats than the wired network environments. Therefore, it has become a primary concern of securing mobile ad hoc networks. In this paper, we address the problem of authentication in mobile ad hoc networks. Public-key based mechanisms are ideal to provide the authentication services. Although this is already mature in a fixed network, providing public key based authentication is still very challenging in mobile ad-hoc networks because of shared wireless medium, energy constrains, dynamic network topology. Here, we present a more efficient public key management system, like Self-Organized scheme[1], and include the concept of trust group for mobile ad hoc networks.**

I. Introduction

In recent years, mobile ad hoc networks have received more attention, because of their easy deployment. However, the characteristics of mobile ad hoc networks are more prone to physical security threats than the wired network environments, which include infrastructureless, dynamic topologies, energy-constrained, a shared wireless channel. Therefore, it has become a primary concern of securing mobile ad hoc networks.

In this paper, we address how to provide authentication services for mobile ad hoc networks. Public-key based systems can be used to provide authentication services. However, the main problem of public-key based security systems is to make each user's public key available to others in such a way that its authenticity is verifiable. In a fixed network, this problem is often solved by the usage of public key infrastructures. Each user has to prove his identity and his public key to a certification authority and then receives a digitally signed public-key certificate by the certification authority. However, from the security aspect, the certification authority will be exposed to single point of failure due to system faults, compromises and denial-of-service attacks. Therefore, the traditional public-key management solutions are not suitable for the mobile ad hoc networks.

In [1], the authors propose a fully Self-Organized public-key management system that allows users to generate their private-public key pairs, to issue certificates for others, and to perform authentication via a chain of Public-key certificates regardless of the network partitions and without any centralized services. Here, we present a more efficient public key management system, like Self-Organized [1], and include the concept of trust group for mobile ad hoc networks.

The rest of the paper is organized as follows: Section 2 presents other solutions that provide authentication services for mobile ad hoc networks. Section 3 describes the basic operations of our proposed scheme. Section 4 discusses the simulation of our proposed scheme and analysis of the simulation result. Section 5 gives the conclusion and our futher work.

II. Related works

Two similar security solutions based on distributed trust for mobile ad hoc networks have been suggested in [2] [3]. The distributed trust approachs are assumed that all nodes in the system know the public-key K and trust any certificates signed using the corresponding private key k. The private key k is divided into n shares using an $(n, t+1)$ threshold cryptography scheme [4], and the shares are assigned to n arbitrarily chosen nodes. And then, multiple nodes can act as servers to sign Public-key certificates for other nodes. However, this approach assumes that some nodes must be initialized by a trusted authority.

In [1], the authors propose a fully Self-Organized public-key management system that allows users to generate their private-public key pairs, to issue certificates for others, and to perform authentication via a chain of Public-key certificates regardless of the network partitions and without any centralized services. Furthermore, this approach does not require any trusted authority, not even in the system initialization phase.

III. Our proposed scheme

In the following, we will describe the basic operations of our scheme. First, the public key and the corresponding private key of each node are created locally by the node itself. After that, each node will issue public-key certificates for its neighboring nodes based on its knowledge about their public keys through location-limited channel [5]. That is, if node u believes that a given public key belongs to a given node v, then u can issue a public-key certificate which is bound to v by the signature of u. Certificates are issued with a limited validity period and each certificate contains its issuing and expiration

0-7803-9410-0/06/$20.00

times. When a certificate expires and its issuer believes that the certificate is still valid, the issuer will issue a new updated version of the same certificate with a new issuing time. After each node issues public-key certificates for its neighbors, the whole certificate graph will be created.

Similar to Self-Organized, each node maintains two caches to store certificates: updated certificate cache and nonupdated certificate cache. The nonupdated certificate cache of a node contains expired certificates that it does not keep updated, and the updated certificate cache contains certificate that it keeps updated. Certificates are periodically exchanged among neighboring nodes. The received certificates are stored in the nonupdated certificate cache of node.

For simplicity, the public keys and the certificates are modeled as a directed graph $G(V, E)$, where V and E stand for the set of vertices and the set of edges, respectively. We call this graph the certificate graph. The public/private key pair of node u are denoted by KU_u and KR_u. The vertices of the certificate graph represent public keys and the edges represent certificates.

The following are the detail of operations about trust group, constructing updated certificate cache, key authentication.

A. Trust group

1) Construct trust group: Imagine the following situation; if two nodes are good friends and they trust each other, in Self-Organized scheme, they still need to find a chain of certificates between them. While in the real situation, there must be some trust relations among nodes. Therefore, in our approach when some nodes have trust relations among them, they can establish a trust group.

We assume that there are trust relations among some nodes and the trust relations have transitive properties. When a node has trust relations with some nodes, they can use these relations to form a trust group. And then, a randomly selected trust group member is responsible to generate a public/private key pair that represents the group's private-public key pair KU_G, KR_G and send this key pair to other trust group members. After receiving this key pair, each trust group member will use KR_G to sign public-key certificates for its neighbors, and its original private key is not used. Moreover, when its neighbors want to issue public-key certificate to it, it will use KU_G not the original public key it owns, to represent its public key.

Each edge in the certificate graph has a weight. The weight of certificate that is signed to or issued by the trust group member is larger than that of normal nodes.

2) The requirements of trust group: To satisfy the requirements of a trust group, we define the trust relations among the trust group members should satisfy the requirements of an equivalence relation. Since the trust relations in our scheme have the transitive property, the trust graph of the trust group will form a bidirectional connected graph. In other words, if the trust relations of some nodes form a bidirectional connected graph, these nodes can form a trust group.

3) Break up trust group: In order to break up trust group, each member received KR_G and KU_G will sign a public-key certificate for KU_G with issuing and expiration time by its own private key. These certificates are issued with an issuing time and an expiration time, and then after the expiration time T, the group will be automatically broken.

B. Constructing the updated certificate cache

We construct the updated certificate cache of each node by Maximum Weight-Degree algorithm, similar to Maximum Degree algorithm in [1]. It selects a set of edges from the nonupdated certificate cache to the updated certificate cache of each node. The selection of edges are based on the weights of the edges and the degrees of the destination vertices of these edges. More precisely, the edge which weight is the largest and the destination vertex of the highest degree is selected. Finally, the updated certificate cache of each node is constructed.

C. Key Authentication

Here, key authentication is performed via chains of public-key certificates. For instance, when node u wants to verify the authenticity of public key of node v, they will merge their certificate caches. And then, u has to find a certificates chain form u to v in their merge certificate cache. To authenticate public key of node v, node u needs to check whether the certificates on the chain have been revoked and the user-key bindings in the certificates are correct. If the check fails, node u aborts the authentication.

IV. Simulation

The purpose of the simulation is to show the improvement of authentication services in ad hoc networks due to the trust group, and the performance of the Maximum Weight-Degree algorithm. In the following section, we will describe the simulation metrics and provide the simulation results.

A. Simulation metrics

we define the average friend ratio $AVGfriend(G)$ of the certificate graph G as the ratio between the number of key pairs(KU_u, KU_v) where there is a directed path from K_u to K_v in the certificate graph G, and the number of key pairs (KU_u, KU_v) among all nodes. Formally, the average friend ratio is defined as follows:

$$AVGfriend(G) = \frac{|\{(KU_u, KU_v) \in VxV : KU_u \rightarrow_G KU_v\}|}{|\{(KU_u, KU_v) \in VxV\}|}.$$

In a similar way, we define the average shortest path $AVGsp(G)$ of the certificate graph G as follows:

$$AVGsp(G) = \frac{1}{|W|} \sum_{(KU_u, KU_v)} sp(KU_u, KU_v, G),$$

where $W = \{(KU_u, KU_v) \in VxV : KU_u \rightarrow_G KU_v\}$ are the total number of key pairs (KU_u, KU_v) where there is a directed path from KU_u to KU_v in the certificate graph G and $sp(KU_u, KU_v, G)$ is the length of the shortest path between KU_u and KU_v. We also denote the total number of certificates of G by $Certificate(G)$.

In order to evaluate the performance of Maximum Weight-Degree algorithm, we define the performance of friend ratio

TABLE I

BASIC PARAMETERS IN OUR SIMULATION

Simulation area	1000 x 1000m^2
Trust group members	5 nodes
Average velocity	9m/sec
Pause time	3sec
Transmission power	8000-12000$pWatt$
Received threshold	1$pwatt$
Bandwidth	11Mb/sec
Movement model	Random way point

TABLE II

COMPARSION OF SELF-ORGAINZED AND GROUP

Certificate graph	$AVGfriend$	$Certificate$
Self-Organized(n=50,m=250)	90%	250
GROUP(n=50,m=239)	90.65%	239
Self-Organized(n=100,m=500)	91.49%	500
GROUP(n=100,m=496)	91.49%	496
Self-Organized(n=50,m=350)	90.24%	350
GROUP(n=50,m=310)	92.24%	310
Self-Organized(n=100,m=700)	99%	700
GROUP(n=100,m=677)	99%	677

$Pfriend(S,G)$ of this algorithm with updated certificate cache size S in the certificate graph G as the ratio between the number of key pairs (KU_u, KU_v) where there is a directed path from KU_u to KU_v in their merged graph $G_u \cup G_v$(their merge updated certificate cache), and the number of key pairs (KU_u, KU_v) where there is a directed path from KU_u to KU_v in the certificate graph G

$$Pfriend(S,G) = \frac{\{(KU_u, KU_v) \in VxV : KU_u \rightarrow_{Gu \cup G\ v} KU_v\}}{\{(KU_u, KU_v) \in VxV : KU_u \rightarrow_G KU_v\}}.$$

B. Simulation environment

We have simulated our approach using an object-oriented modular discrete event simulator OMNET++ [7] [8]. Simulation parameters for mobile ad hoc networks are given in Table 1. We assume there are five members in a trust group in our simulation. The network for simulation runs consists of 50 nodes or 100 nodes in a 1000 x 1000m^2 square area, and the nodes move following with the random way-point mobility model.

C. Result and analysis

As shown in Table 2, we generate lower and higher connective certificate graphs in our simulation. The first four certificate graphs are lower connective certificate graphs, and the others are higher connective certificate graphs. Two of the lower connective certificate graphs are generated by the Self-Organized scheme and denoted by Self-Organized. The other lower connective certificate graphs are generated by our scheme and denoted by GROUP. Moreover, there exists a trust group in each certificate graph GROUP. Like the lower certificate graphs, the higher connective certificate graphs also consist of Self-Organized and GROUP. For simplicity, the number of vertices and the number of edges of a certificate graph are denoted by $n = |V|$ and $m = |E|$, respectively.

Here, we show the effects of trust group in the whole certificate graphs. Table 2 shows the comparison of Self-Organized and GROUP. We observe that the trust group not only improves $AVGfriend$, but also reduces $Certificate$. Figure 1 shows the comparison of $AVGsp$ ratios of Self-Organized and GROUP. In Figure 1, the $AVGsp$ ratio of first pair is reduced by 24.64% and that of second pair is reduced by 18.49%. The third and fourth pairs are reduced by 17.3% and 10.3%. Therefore, the $AVGsp$ ratios can be reduced by establishing the trust group. However, the improvement is smaller, as the number of nodes increases. For instant, the reduced ratios of $AVGsp$ and $Certificate$ will decrease as the number of nodes increases.

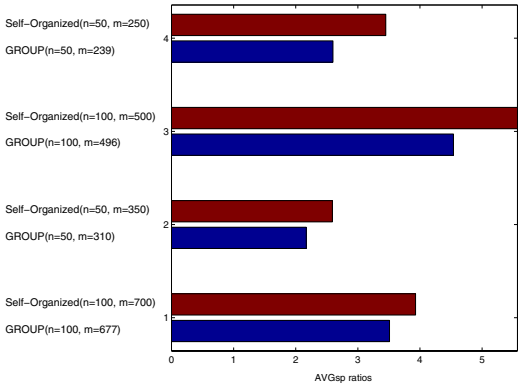

Fig. 1. $AVGsp$ ratios of Self-Organized and GROUP

In the following, we show the performance of Maximum Weight-Degree algorithm. Figure 2 and Figure 3 show the $Pfriend$ ratios of Self-Organized and GROUP in the merged updated certificate caches $(G_u \cup G_v)$. We observe that on all types of graph, Maximum Weight-Degree algorithm exhibits high performance, even if the size of the updated certificate cache is small compared to the number of nodes and the total number of certificates in the certificate graph. Moreover, for the same cache size, the $Pfriend$ ratios of GROUP grow faster than that of Self-Organized, especially when the cache size is small.

In Figure 4 and Figure 5, we show the $AVGsp$ ratios of Self-Organized and GROUP in the merged updated certificate caches $(G_u \cup G_v)$. We observe that the lengths of the shortest paths in the merged updated caches are not significantly longer than those in the whole graph. On all types of graph, the $AVGsp$ ratios of GROUP are smaller than that of Self-Organized. In addition, the $AVGsp$ ratios in Figure 4 and Figure 5 are very close to that of Self-Organized and GROUP in Figure 1, because of using the Maximum Weight-Degree algorithm.

V. CONCLUSION

In this paper, we propose efficient authentication services for mobile ad hoc networks. The main contributions of our work are summarized as follows: 1. We introduce the concept of trust group into the public key management system and define the requirements of a trust group clearly. The simulation

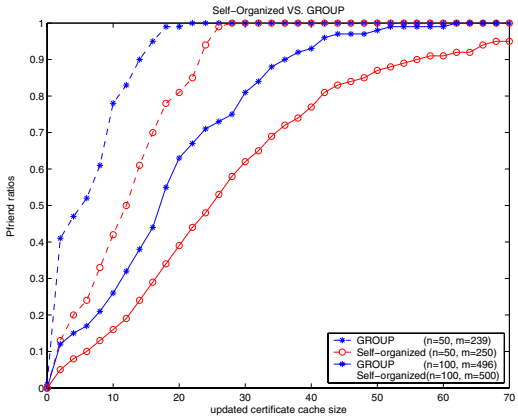

Fig. 2. $Pfriend$ ratios of lower connective Self-Organized and GROUP

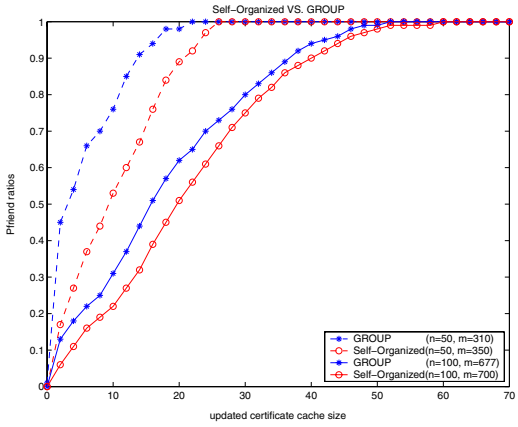

Fig. 3. $Pfriend$ ratios of higher connective Self-Organized and GROUP

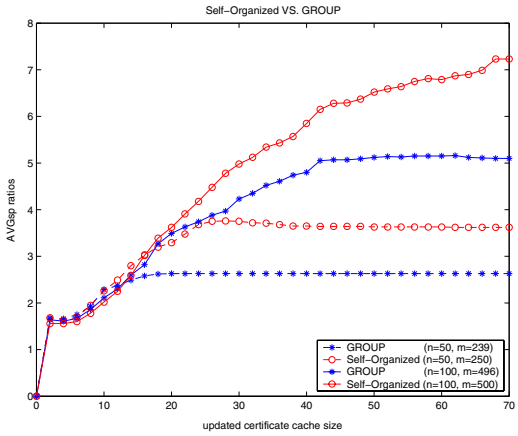

Fig. 4. $AVGsp$ ratios of lower connective Self-Organized and GROUP

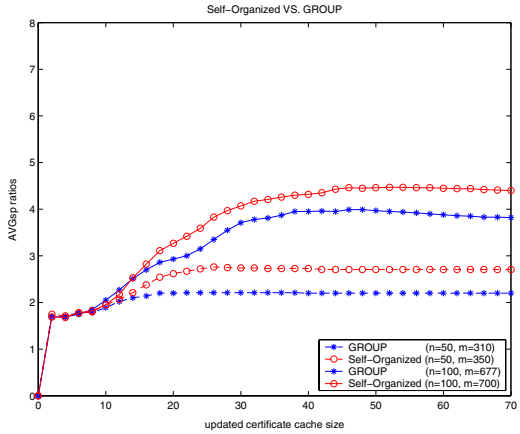

Fig. 5. $AVGsp$ ratios of higher connective Self-Organized and GROUP

results show the advantages of establishing a trust group: The trust group not only reduces $AVGsp$ and $Certificate$, but also improves $AVGfriend$. 2. The Maximum Weight-Degree algorithm exhibits high performance, even if the size of the updated certificate cache is small compared to the total number of nodes and certificates in the certificate graph, especially when a trust group exists in the network. In particular, when each node applies the Maximum Weight-Degree algorithm, the estimated $AVGsp$ ratios are very close to the $AVGsp$ ratios of the global certificate graph. In the future, we will study the evaluation of the fuzzy trust relations among users to make the definition of the trust group more robust.

REFERENCES

[1] S. Capkun, L.i Buttyan, and J.P. Hubaux, "Self-Organized public-key management for mobile ad hoc network" IEEE Transactions on Mobile Computing, vol. 2 Jan/Mar 2003, pp. 52-64.
[2] L. Zhou and Z. J. Hass, "Securing Ad Hoc Networks," IEEE Network vol. 13, no. 6, Novemember/Decemeber 1999, pp. 24-30.
[3] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, "Providing robust and ubiquitous security support for mobile ad-hoc networks," The 9th International Conference on Network Protocols, November 2001, pp. 251-260.
[4] A. Shamir, How to share a secret, Communications of ACM 1979.
[5] D. Balfanz, D. K. Smetters, P. Stewart and H. Chi. Wong, "Talking to Strangers: Authentication in Ad-Hoc Wireless Networks," In Symposium on Network and Distributed Systems Security, 2002.
[6] P. Zimmermann, The Official PGP User's Guide. MIT Press, 1995.
[7] "OMNET++ Community Site", http://www.omnetpp.org/index.php.
[8] N. Concer, "Ad Hoc Sim version 1.1".