A Group-Based Key Management Protocol for Mobile Ad Hoc Networks

Qing Chen*, Xiaodong Lin†, Sherman Shen‡, Kazuo Hashimoto*, and Nei Kato*
*Graduate School of Information Sciences, Tohoku University, Japan
†Faculty of Business and Information Technology, University of Ontario Institute of Technology, Canada
‡ Department of Electrical and Computer Engineering, University of Waterloo, Canada
E-mail: chen@it.ecei.tohoku.ac.jp

Abstract— **Due to the dynamic topology and non infrastructure, network participants cooperate with their neighbors to route packets. The lack of centralized services allows mobile ad hoc networks to be easily and swiftly deployed, but make it difficult to check others' identities on the other hand. Cryptographic tools have been introduced to secure group communications, such as Private and Public Key Infrastructure. The autonomous and distributed nature of mobile ad hoc network demands a decentralized authentication service, where Public Key Infrastructure is considered a better solution. Public Key Infrastructure can ensure both confidentiality and authenticity, but it is impractical to provide an online trusted third party as Certificate Authority (CA) for mobile ad hoc network. In this paper, we proposed a new key management protocol which utilizes certificate graphs and distributed Certificate Authorities. Certificate graph maintained by each user represents the trust among his neighbors, then the maximum clique of certificate graph is selected to be CAs. Based on the assumption that initial certificate graph building is secure [11], good users have more friends while bad ones have less, thus a reliable group can be constructed. The most trustful subset of these good users – the maximum clique – is elected as the governor of this group, which takes the responsibility of certificate authentication.**

I. Introduction

Wireless mobile ad hoc network is an relative newly developed network architecture for wireless mobile terminals. It consists of mobile communication devices such as laptops, mobile phones and personal digital assistants. Instead of making use of a centralized routing service, mobile devices cooperate with each other to route network packets from source to destination in a multi-hop manner. There is no infrastructure in mobile ad hoc network, each user only knows his neighbors one hop away.

There are several fundamental differences between mobile ad hoc network and traditional wired network: (a) *Dynamic topology*: Network terminals can move freely at certain speed, therefore network topology is always changing and hardly to be predicted; (b) *Resource constraints*: Mobile terminals can be notebooks, PDAs or mobile phones. They all have limited computation power and short battery life; (c) *No infrastructure*: Ad hoc network is meant to be deployed swiftly. Mobile users cooperate to route packets, and thus there is no need of centralized services; and (d) *Limited physical security*: Mobile devices such as notebooks, PDAs and mobile phones don't have strong secure systems due to cost and limited power.

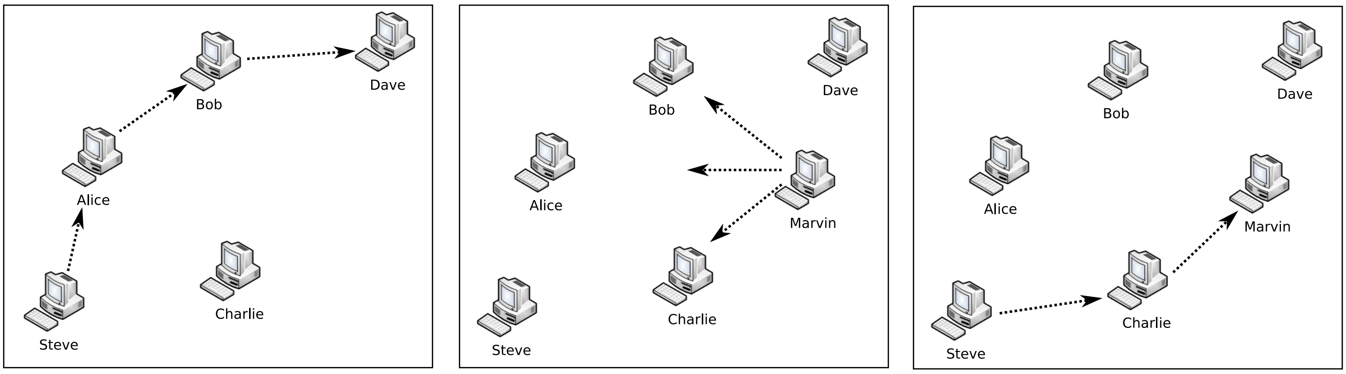
The usability and reliability of mobile ad hoc networks strongly depends on its security. However, due to its openness and lack of centralized services, providing a secure environment is still a challenging task for a mobile ad hoc network. It is not easy to achieve confidentiality and authenticity in such network.

There are a number of different attacks that target the mobile ad hoc network, ranging from simple to sophisticated ones. For example, an attacker can be selfish by unwilling to forward packets for others, or simply discarding data packets received. Malicious routing attacks can disrupt routing discovery or maintenance phase by disobeying the rules defined by the routing protocols. More sophisticated attacks includes blackhole [1], byzantine [3], wormhole [4] and spoofing attack [5] (Fig. 1).

Cryptographic tools have been introduced into mobile ad hoc network to secure group communications. These includes Private [7] [8] [9] and Public Key Infrastructure [10] [11] [12] [13] [14]. Private Key Infrastructure allows two or more users to establish secure communications by sharing a common secret key. However, in a network with numbers of users, such key agreement protocol requires heavy message exchange. Public Key Infrastructure uses a pair of keys to encrypt/decrypt messages. Private key is kept secret while public key can be widely distributed. It reduces the message exchanging overhead as in Private Key Infrastructure, but requires a trusted third party as the server (Certificate Authority) for key generation and authentication. Providing an online CA in a mobile ad hoc network is challenging due to the nature of mobile ad hoc network. Therefore key authentication service should also be decentralized and autonomous.

In this paper, we propose a new key management protocol which employs public key infrastructure. In our protocol, key authentication responsibilities are distributed to several CAs. Instead of choosing CA manually [12], a certificate graph is used to represent the friendship among nodes. Based on the assumption that the trust in such certificate graph can be built by using a secure channel [11] (details on building trust will be described later), then the maximum clique of this graph is chosen as CAs. This is similar to human social network, where good users have more friends while bad ones have less. The most trustful subset of these good users, the maximum clique, is elected as the governor of this group. In order to evaluate our proposal, we implement the key management function as an extension to AODV protocol [6], then demonstrate the performance through simulations.

978-1-4244-4148-8/09/$25.00 ©2009



(a) Normal scenario

(b) Marvin broadcasts fake identity claiming as Dave

(c) Packet sent to fake Dave

Fig. 1. An example of identity spoofing attack

The remainder of this paper is organized as follows. Section II describes several research works based on public key infrastructure. Section III presents the proposed maximum clique based key management protocol. In Section IV, we evaluate the performance of our protocol. Finally, we draw our conclusions in Section V.

II. Related Work

There are basically two kinds of key infrastructures, private and public key infrastructures. The first uses symmetric cryptography to establish shared private keys, such as Diffie-Hellman two-party agreement [7], Group Diffie-Hellman [8] and Asymmetric Group Diffie-Hellman [9]. The basic idea of Diffie-Hellman two-party agreement protocol [7] is generating a strong common key based on weak shared secrets: Alice generates two random numbers a and p, which are common secret to everybody, calculates a^p and sends the result to Bob. Bob also generates a random number q, calculates $a^{p \times q}$ and sends the result back to Alice as the common secret key, by which they can perform message encryption/decryption sent to and received from each other. As the key establishment is based on bidirectional message exchange, there will be higher overhead when the number of nodes increases. Protocols in [8] [9] extend the idea of Diffie-Hellman two-party key agreement to n-party situation.

The second key infrastructure relies on asymmetric cryptography and CA to generate a pair of keys (public/private) for each network participant. Due to the characteristic of mobile ad hoc networks, providing such key infrastructure is a challenging task. Since mobile ad hoc network lacks a centralized service, the key infrastructure should also be decentralized and self-organized. Our work is inspired by [11] and [12], which both utilize public key infrastructure.

Public key infrastructure can assure:

- Confidentiality: message encrypted with Alice's public key cannot be decrypted by anyone except the holder of the corresponding private key – Alice herself, which ensures the confidentiality of the message.
- Authenticity: message signed with Alice's private key can be verified by anyone who has access to Alice's public key, therefore proving that the message comes from Alice. This ensures the authenticity of the message sender.

Several protocols based on Public Key Infrastructure are described in the following.

Threshold Cryptography and Distributed Servers

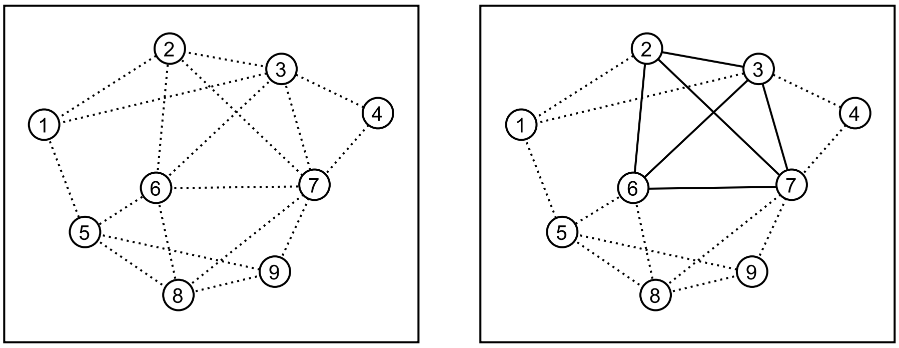
[10] is the first in using public key to secure ad hoc communication. They choose n nodes from the network as servers, allow these servers to share the abilities of CA. In a network with configuration of $(n, k+1)$, when a user wants to sign a certificate, he has to contact $k+1$ servers $(n \geq 3k+1)$ to collect $k+1$ partial signatures in order to compute a full signature. However, contacting with $k+1$ servers introduces both overhead and delay. Besides, no detail on how to choose servers has been mentioned.

Self-Organized Certificate Chain

Self-organized certificate chain [11] is a fully distributed self-organizing public key management system. Similar to PGP (Pretty Good Privacy), users generate and distribute their own public key certificates, issue certificates for others by themselves. Instead of employing online certificate authorities, each node stores a selected number of certificates locally according to predefined rules.

When a user wants to validate another one's public key certificate, for example, Alice wants to verify the authenticity of Bobs public key, she first merges her stored certificates. Then Alice tries to find a certificate chain from Alice to Bob in the merged certificates. The basic operations of self-organized certificate chain are as follows.

- Step 1: Each user generates his own public/private key pair. If Alice believes that a public key K_{Bob} belongs to Bob, then she can issue a public key certificate in which K_{Bob} is bound to Bob by the signature of Alice. There are several reasons for Alice to believe that K_{Bob} belongs to Bob. For example, Alice and Bob may have exchanged their keys through a dedicated secure channel. Every node stores certificates locally, which are a) certificates that issued by itself; b) selected certificates issued by other



(a) Certificate graph

(b) Maximum clique

Fig. 2. Maximum clique searching on certificate graph

nodes, according to certain rules. The result of certificate issuing is a certificate graph G.

- Step 2: Certificate exchange allows users to share their certificates. Each user exchange his certificates periodically with one hop neighbors.
- Step 3: Certificate authentication: When Alice wants to verify Bob's public key K_{Bob}, she asks Bob for his certificates graph G_{Bob}. The requested user sends his stored certificates to Alice, then Alice merges the received certificates with her own G_{Alice} and tries to find a certificate chain from K_{Alice} to K_{Bob} in the merged certificates $G_{Alice} \cup G_{Bob}$.

This solution is fully distributed by eliminating the use of servers. However, it can only guarantee probabilistic certificate requests, since not all certificates are stored by one user.

MObile Certificate Authorities

Similar to threshold cryptography in [10], the MOCA [12] protocol also distributes CA responsibilities over a number of manually selected nodes. It is suggested that mobile nodes are different in power, capabilities, transmission range and physical security, among them the most powerful and secure nodes would be chosen as MOCAs.

As in traditional single CA scheme, MOCAs generate public/private keys and store certificates. However, key authentication is similar to [10]: when Alice wants to authenticate Bob's public key, she sends out certification request (CREQ) packets. Any MOCA which received a CREQ replies with a certification reply (CREP) packet containing its partial signature. Alice waits a certain period of time for k such CREPs. Only when Alice collects k valid CREPs, can she construct a full signature and the certification request succeeds.

How to choose k for networks with different numbers of users is a problem for MOCA, and it pays a lot to reconfigure after k being chosen. Besides, the choice of CA based on physical characteristic may be insufficient for ad hoc network, since topology is the most important factor that significantly affects communication.

III. Proposed Protocol

Inspired by certificates graph in [11] and MOCA, we propose a novel key management protocol utilizing both certificates graph and CAs.

A. Maximum Clique as Certificate Authorities

Similar to the PGP "web of trust" in [11], every node V maintains a friendship graph G, where each edge E represents a trustful relationship. Our basic idea for choosing Certificate Authorities is by running a maximum clique searching algorithm on the graph G. Maximum clique searching problem is a well known problem in graph theory [15] [16]. The problem is defined as follows. Given a graph $G = (V, E)$, the maximum clique is the largest subset S in V, such that for all x, y in S, (x, y) in E. In other words, every pair of vertices v and u in a maximum clique S is connected by an edge in E. Here we use V to represent each network node and E to represent a valid certificate between two nodes. In this context, G can be considered as a friendship graph. (Fig. 2)

The reason we choose the maximum clique as CA is that:

- First, maximum cliques are found by nodes themselves, not manually selected, which ensures a decentralized and autonomous infrastructure.
- Second, in a mobile network with more than one CA, it is obvious that every CA should be familiar with each other by knowing his public key. Besides, certificates stored at different CAs must be coincidence, otherwise conflicts of certificate may occur. In a maximum clique, every member knows each other. There are no two members being strangers, which ensures close cooperation among CAs.

Network consists of good users and bad ones, the later don't play the rules defined by the protocol by being selfish or disrupting packet routing. Similar to human network, good users have more friends while bad ones have less. The most trustful subset of these good users, the maximum clique, is elected as the governor of this group.

B. An Extension to AODV protocol

We extend the AODV protocol to implement key management function. AODV protocol uses Route Request (RREQ) packet and Route Reply (RREP) packet to build route path. When source node Steve wants to send a message to Dave, he

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

broadcasts RREQ packets to the network. Neighboring nodes which get this RREQ keeps forwarding it until it reaches destination Dave. Dave then solicits a RREP and sends it back to Steve. When Steve gets a RREP, an routing entry is created, after which can Steve start talking to Dave. (Fig. 3)

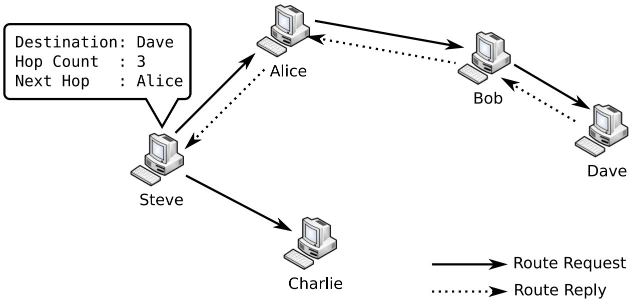


Fig. 3. Route Control Packet in AODV protocol

C. Protocol Implementation

In order to properly implement key management function in AODV protocol, we introduce three additional packets:

- Key packet: public key issued by user himself, which is to be distributed to the network. The corresponding private key is kept secret.
- Certificate chain packet: a chain of certificates, which can also be exchanged with friends.
- Certificate request packet: which includes the required user ID to be sent to CA.

The key management function works as follows:

A. Certificate issuing:

The protocol begins with issuing of certificates. During this phrase, users issue certificates for their trusted neighbors, which is similar to [11]. The issuing of certificates is bidirectional, which means if Alice issues a certificate for Bob, Bob will also issue a certificate for Alice, such that they two becomes friends.

B. Certificate exchanging: These certificates can be exchanged among their friends. This is done by exchanging of certificate chain packets with friends.

C. Maximum clique searching on certificate graph: The purpose of issuing and exchanging certificates is to build a certificate graph. With this graph, user can get a rough knowledge of his neighbors. By running maximum clique searching on this certificate graph, user can find a subset of nodes, which are the maximum clique members. These maximum clique members are announced as Certificate Authorities. (Figure 2)

IV. Performance Evaluation

A. Simulation Setup

We implement our protocol in the QualNet simulator [17]. The objective is to evaluate the performance of proposed key management protocol. In the evaluation, we compare our protocol with the manually chosen Certificate Authorities scheme (the MOCA protocol for a small network, where we set required number of CREP k to 1). Simulation parameters are given in Table I. All simulations have topology as shown in Fig. 4.

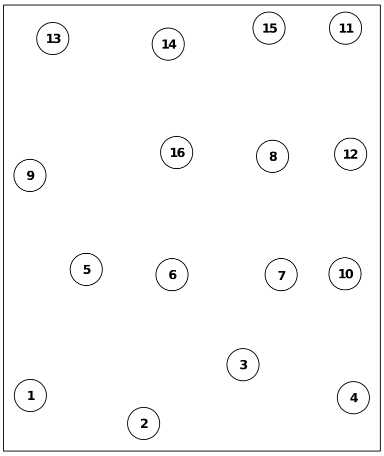


Fig. 4. Network Topology

There are two metrics we would like to evaluate:

- The success rate R of answering certificate request, which measures the effectiveness of key management protocol. In our experiments, we first randomly choose two users Alice and Bob, then let Alice send certificate request to CA asking for Bob's certificate. R is defined as follows
$$R = \frac{Number\ of\ Certificate\ Found}{Number\ of\ Certificate\ Request\ Sent}$$
- Time delay T of answering certificate request. Certificates should be found within a short period, i.e, a small T, in order to make the authentication request applicable. T is calculated by $T = (Time\ of\ finding\ certificate) - (Time\ of\ sending\ certificate\ request)$

Within 600 seconds, certificate requests are sent twice, at 400 and 500 seconds, respectively. For each simulation, a random pair of Alice and Bob are chosen. We run each simulation 50 times, then get the average values of R and T.

TABLE I
Simulation Setup

Simulator	QualNet 4
Routing Protocol	AODV
Scenario Dimension	1000m x 1000m
Simulation Time	600 seconds
Number of Nodes	16
Maximum Clique Size	3
Number of MObile CA	3

B. Simulation Results

Fig. 5 shows the success ratio of answering certificate request. The proposed protocol has higher success ratio in answering certificate requests than the MOCA protocol.

It can be noticed that maximum clique nodes have better connectivity to non-clique nodes in a graph. As a result, selecting maximum clique as CAs can provide better connectivity.

Besides, the choice of maximum clique as CAs is totally made by users themselves, which is suitable for the autonomous nature of ad hoc network.

Fig. 6 shows the delay of answering certificate request. Although the proposed protocol has longer delay than MOCA, it is still acceptable.

Since maximum clique searching can return multiple results (as shown in Table II), they have similar performances. Therefore, we simply choose the first found maximum clique as CAs.

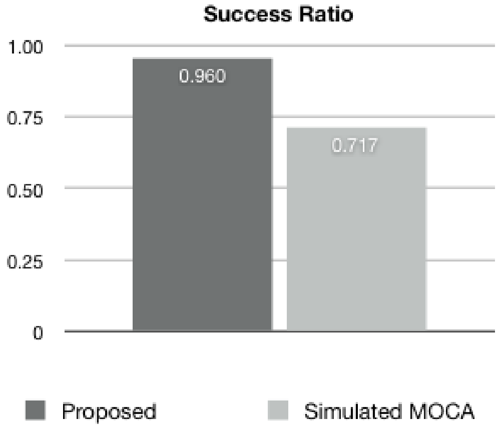


Fig. 5. Success Ratio

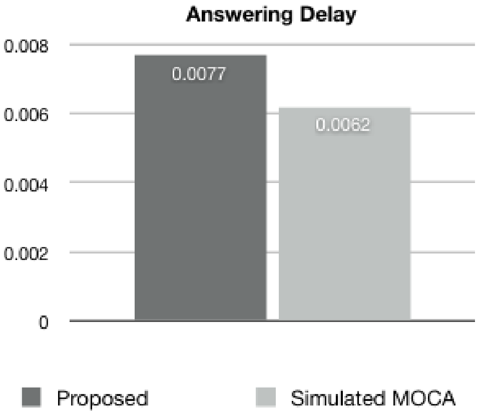


Fig. 6. Answering Delay (seconds)

TABLE II
MAXIMUM CLIQUE SEARCHING RESULTS

Maximum Clique	Success Ratio	Answering Delay (seconds)
1, 5, 13	0.96	0.0072
1, 6, 13	0.959	0.0065
1, 4, 5	0.9	0.0068

V. CONCLUSION

In this paper, we have proposed a new key management protocol for wireless mobile ad hoc networks, which offers distributed authentication service with high certificate request answering ratio, while still maintaining a low answering delay in order to make it applicable. For our future work, we will study a decentralized and autonomous authentication service, which suits mobile ad hoc network better.

ACKNOWLEDGMENTS

This work is supported by the Research and Development of Dynamic Network Technology program of NiCT.

REFERENCES

[1] I. Aad, J.-P. Hubaux, E. W. Knightly, “Impact of Denial of Service Attacks on Ad Hoc Networks”, *IEEE/ACM Transactions on Networking*, 2008.

[2] Y. Hu and A. Perrig, “A Survey of Secure Wireless Ad Hoc Routing”, *IEEE Security & Privacy*, pp. 28–39, 2004.

[3] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens, “An On-demand Secure Routing Protocol Resilient to Byzantine Failures”, *Proceedings of the ACM Workshop on Wireless Security*, pp. 21–30, 2002.

[4] R. Maheshwari, J. Gao, and S. R. Das, “Detecting Wormhole Attacks in Wireless Networks Using Connectivity Information”, In *IEEE Conference on Computer Communications INFOCOM*, 2007.

[5] K. Sanzgiri, B. Dahill, B. Levine, C. Shields, and E. Belding-Royer, “A Secure Routing Protocol for Ad Hoc Networks”, in *Proceedings of IEEE International Conference on Network Protocols*, 2002, pp. 78-87.

[6] C. Perkins and E. Royer, “Ad hoc On-demand Distance Vector (AODV) Routing”, in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, 1999, pp. 90–100.

[7] W. Diffie and M. E. Hellman, “New Directions in Cryptography”, *IEEE Transactions on Information Theory*, vol. IT-22, no. 6, 1976, pp. 644–54.

[8] M. Steiner, G. Tsudik, and M. Waidner, “Diffie Hellman Key Distribution Extended to Group Communication”, *ACM Conference on Computing and Communication Security*, 1996, pp. 31–37.

[9] R. Bhaskar, D. Augot, Cédric Adjih, Paul Mühlethaler, and S. Boudjit, “AGDH (Asymmetric Group Diffie Hellman) An Efficient and Dynamic Group Key Agreement Protocol for Ad hoc Networks”, *New Technologies, Mobility and Security*, 2007.

[10] L. Zhou and Z. J. Haas, “Securing Ad Hoc Networks”, *IEEE Network*, vol. 13, no. 6, 1999, pp. 24–30.

[11] S. Capkun, L. Buttyan, and J.-P. Hubaux, “Self-organized Public Key Management for Mobile Ad Hoc Networks”, *IEEE Transactions on Mobile Computing*, vol. 2, no. 1, Jan. 2003, pp. 52–64.

[12] S. Yi and R. Kravets, “Moca: Mobile Certificate Authority for Wireless Ad Hoc Networks”, *2nd Annual PKI Research Workshop*, Gaithersburg, MD, 2003.

[13] A. Rachedi and A. Benslimane, “A Secure Architecture for Mobile Ad Hoc Networks”, 2nd International Conference on Mobile Ad-hoc and Sensor Networks MSN 2006, *Springer’s LNCS*, December 13-15, Hong Kong, China, pp. 424-435.

[14] A. Rachedi, A. Benslimane, L. Guang and C. Assi, “A Confident Community to Secure Mobile Ad-Hoc Networks”, *ICC 2007*, Glasgow, Scotland, UK.

[15] S. Skiena, “The Algorithm Design Manual”, Springer, 1998.

[16] A. Dharwadker, “The Clique Algorithm”, available at http://www.geocities.com/dharwadker/clique/

[17] QualNet Simulator (version 4), http://www.scalable-networks.com/

978-1-4244-4148-8/09/$25.00 ©2009