A Secure Global State Routing for Mobile Ad Hoc Networks

CHEN Jing
College of Computer, Huazhong University of Science & Technology
Wuhan 430074, China

CUI Guo Hua
College of Computer, Huazhong University of Science & Technology
Wuhan 430074, China

HONG Liang
College of Computer, Huazhong University of Science & Technology
Wuhan 430074, China

ever_cs@163.com

Abstract

This paper presents a secure routing protocol for wireless ad hoc networks - Secure Global State Routing Protocol (SGSR). SGSR defines some rules to ensure that nodes can discover neighbor nodes safely and defines priority of neighbor nodes to defend routing against denial of service attacks. It also provides an algorithm to ensure that a packet can't travel more than certain hops. It is capable of adjusting its scope between local and network-wide topology discovery. The paper provides formal analysis to illuminate that SGSR is robust against individual attackers.

1. Introduction

An ad hoc network is a collection of mobile computers (or nodes) that cooperate to forward packets for each other to extend the limited transmission range of each node's wireless network interface. An ad hoc network is often defined as an "infrastructureless" network, meaning a network without the usual routing infrastructure like fixed routers and routing backbones. A routing protocol in such a network finds routes between nodes, allowing a packet to be forwarded through other network nodes towards its destination.

The Mobile Ad Hoc Networking (MANET) has the collaborative, self-organizing environment. It opens the network to numerous security attacks that can actively disrupt the routing protocol and disable communication[1]. Recently, many ad hoc routing protocols have been proposed. Most of the protocols discover the route only when a source node needs to route packets to a destination node; that means, they are reactive routing protocols[2]. But in many situations, proactive discovery of topology performs better. Link State Routing protocol(LSR) is a "proactive" routing scheme. SGSR is based on LSR.

Some vicious nodes may exhibit some malicious behaviors, such as: forgery, replay, corrupting link state updates or Denial of Service (DoS) attacks. This paper provides a scheme to secure the discovery and the distribution of link state information. Section 2 takes a look at related work. Section 3 presents our Secure Global Routing Protocol and the data that nodes need. Section 4 and 5 provide the security and formal analysis. Section 6 shows the result of the simulation. Finally, it concludes with a description related to future work.

2. Related Work

The collaborative, self-organizing environment of the Mobile Ad Hoc Networking technology opens the network to numerous security attacks that can actively disrupt the routing protocol and disable communication. Attacks on ad hoc network routing protocols generally fall into one of two categories: 1) Routing-disruption attacks. The attacker attempts to cause legitimate data packets to be routed in dysfunctional ways. 2) Resource-consumption attacks. The attacker injects packets into the network in an attempt to consume valuable network resources such as bandwidth or to consume node resources such as memory (storage) or computation power.

Recently, a number of protocols have been proposed to secure wireless ad hoc routing. Papadimitratos and Haas proposed the Secure Routing Protocol [6], which we can use with DSR (Dynamic Source Routing Protocol) or the Interzone Routing Protocol in the ZRP (Zone Routing Protocol). They designed SRP (Secure Routing Protocol) as an extension header that is attached to ROUTE REQUEST and ROUTE REPLY packets. SRP doesn't attempt to secure ROUTE ERROR packets but instead delegates the route-maintenance function to the Secure Route Maintenance portion of the Secure Message Transmission protocol. SRP requires that, for every route discovery, source and destination must have a security association between them. Furthermore, the paper does not even mention route error messages. Therefore, they are not protected, and any malicious node can just forge error messages with other nodes as source. Ariadne [12] is a secure on-demand routing protocol based on DSR and TESLA(Timed Efficient Stream Loss-tolerant Authentication), which withstands node compromise and

relies on highly efficient symmetric cryptography and requires clock synchronization. ARAN (Authenticated Routing for Ad hoc Networks) is based on AODV (Ad hoc On-Demand Distance Vector Routing Protocol) and proposed by Dahill In ARAN, each node has a certificate signed by a trusted authority. Every node that forwards a route discovery or a route reply message must also sign it, which is very computing power consuming and causes the size of the routing messages to increase at each hop. Manel Guerrero Zapata and N. Asokan propose Secure AODV (SAODV), another protocol designed to secure AODV.

The idea behind SAODV is to use a signature to authenticate most fields of a route request (RREQ) and route reply (RREP) and to use hash chains to authenticate the hop count. SAODV designs signature extensions to AODV. Network nodes authenticate AODV routing packets with an SAODV signature extension, which prevents certain impersonation attacks.

3. Secure Global State Routing Protocol (SGSR)

The scope of SGSR may range from a secure neighborhood discovery to a network-wide secure link state protocol. SGSR nodes distribute their link state updates and maintain topological information within R hops, which we refer to as zone.

3.1 Node's Equipment

Node i is equipped with a public/private key pair, namely K_i and K_i^{-1}. Key certification can be provided by a coalition of N nodes and the use of threshold cryptography[4].

Each node has a single network interface. It's identified by its IP addresses, which can be assigned by many schemes, e.g., dynamically or even randomly. But after a node enters our region and passes the authentication, it can't be changed. The new node's IP and K_i will be stored in every node.

Besides other nodes' IP and K_i, the SEQ and the single hop broadcast key are very necessary.

3.2 Neighbor Detecting

Each node submits its Medium Access Control (MAC) address and its IP address, the (MAC_n, IP_n) pair, to its neighbors by broadcasting signed hello messages. Receiving nodes validate the signature and retain the information when they find the hello packet coming from a new node; Neighbor Detecting has the following tasks:

1) Maintaining a table of (MAC_n, IP_n) pair of the node's neighbors, if neighbor changes IP or uses others IP, deletes the neighbor from the neighbor table.

2) Judging latent discrepancies, such as a single data-link interface using multiple IP addresses.

3) Measuring the rates at which control packets are received from each neighbor, by differentiating the traffic primarily based on MAC addresses, if one neighbor's sending rate is high, SGSR debases its packets' priority.

3.3 Packets' Transmission

Because the cost of calculating a hash value is smaller than signature, SGSR uses a single hop broadcast key to ensure the authenticity and integrality of the packets. Each node must exchange the single hop broadcast key to its neighbor together with authentication. The process is as follow:

1) $A \rightarrow B : Cert_A, \{N_a\}_{K_A^{-1}}$

2) $B \rightarrow A : \{K_{TC\_B}, Na+1\}_{K_B^{-1}}, Cert_B$

3) $A \rightarrow B : \{K_{TC\_A}, Na+2\}_{K_A^{-1}}$

A and B are two nodes. K_{TC_A} ,K_{TC_B} are the single hop broadcast key of A and B. N_a is a random number created by A.

If there are three nodes named A,B,C as shown in figure 1, B is the neighbor of A and C. But A and C are not neighbors. A sends packets to C. PC indicates the packet's content. $K^j{}_{TC\_i}$ indicates the single hop broadcast key of node i stored in node j.

1) $A \rightarrow B : (PC, (PC, K^A_{TC\_A})_{HASH})$

2) Node B use PC to calculate $(PC, K^B_{TC\_A})_{HASH}$, if $(PC, K^A_{TC\_A})_{HASH} == (PC, K^B_{TC\_A})_{HASH}$, goto 3), else drop the packet.

3) $B \rightarrow C : (PC, (PC, K^B_{TC\_B})_{HASH})$

4) Node C use PC to calculate $(PC, K^C_{TC\_B})_{HASH}$, if $(PC, K^B_{TC\_B})_{HASH} == (PC, K^C_{TC\_B})_{HASH}$, accept the packet, else drop the packet.

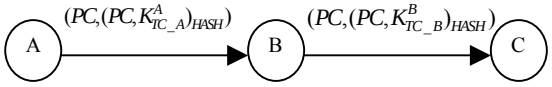

Fig.1 A transmit packets to C

Because the single hop broadcast is created with the process authentication, the malicious node can't get the single hop broadcast key. This method is more effective than using signature and has the same security.

3.4 Hops Limitation

Link State Updates (LSU) are identified by the IP address and the SEQ. The SEQ, a 32-bit sequence number, provides the updates with an address space of four billion. The structure of the Link State Updates is composed of eight parts that are shown in figure 2.

TYPE stands for the type of packet, R_{HOPS} indicates the number of the hops that the Link State Updates Packet has passed; RESERVED indicates the field reserved; HASH_MAXHOPS indicates the hash value[5] when the Link State Updates Packet has passed the max hops, HASH_TRAVERSED indicates the hash value now, LSU_SEQUENCE indicates the sequence of the Link State

Updates Packet, NEIGHBOR_TABLE indicates the neighbor table of the node which sends the packet, SUMMARY can prevent the malicious node juggling.

R_{HOPS}, HASH_MAXHOPS, HASH_TRAVERSED is used for limiting the max hops and avoiding flooding. The arithmetic as follow:

1) If the node wants to send the Link State Updates packet, goto 2), and if transmit the packet, goto 4).

2) The node sending the packet chooses a random value V, and calculates a hash chain, $V_i = H^i(V), i=1,...,N, H^0(V)=V$. N is the max hops of the zone. $H^i(V)$ means the hash value after i times calculating with the parameter V.

3) HASH_TRAVERSED is equal to V_0 and HASH_MAXHOPS is equal to V_N, goto 7).

4) After receiving the packet, the node inspects the SUMMARY . If failed, goto 8), else goto 5).

5) The node uses the HASH_TRAVERSED from the received packet calculating the value of $H^{R-R_{HOPS}}(HASH\_TRAVERSED)$, if the value is equal to HASH_MAXHOPS, then goto 6), else goto 8).

6) The HASH_TRAVERSED is replaced by H(HASH_TRAVERSED), and R_{HOPS} is replaced by $R_{HOPS}+1$.

7) Sending or transmitting the packet. The process ends.

8) Drop the packet. The process ends.

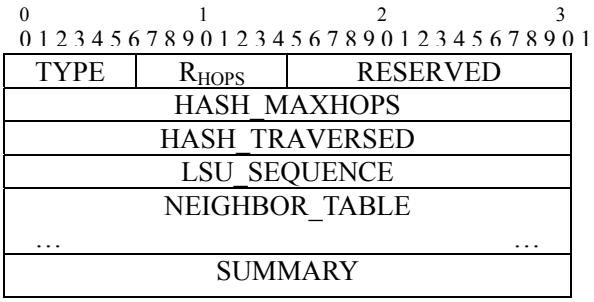

Fig.2 Link State Updates Packet
(broadcast the state of the node's neighbor)

3.5 Public Key Distribution

SGSR uses the LSU packet structure and adds two fields called CERTIFICATE and K_{pub} to distribute public key. This method can save network resource.

Nodes validate the CERTIFICATE of the packets only if the nodes are not yet aware of the originator's public key. Upon validation, K_{pub} and the corresponding source IP address are stored locally, along with the corresponding sequence number. Also, each node can autonomously decide whether to validate the CERTIFICATE and K_{pub} or not. For example, if a node only wants to communicate with the nearby destination, it needn't validate the PKD packets from the remote nodes. Similarly, if the node considers the topology view broad enough, it avoids validating it. When a node moves to a new zone, it can timely acquire and validate other nodes routing information in a timely manner, as do other nodes.

4 Security Analysis

MANET may be suffered from two type attack. One is active attack. The attackers achieve their illegal aim by modifying, deleting, delaying, inserting the data stream. The other one is passive attack[6,7]. The attacker only listens to the information on the network, instead of modifying it. SGSR is effectual when the attack is active.

The attacks which SGSR can resist are as follows:

1) Interrupting attack. Because the LSU packets are sent by broadcasting, the attacker can not interrupt all routes.

2) Juggling attack. The packets have summary. If the packets are changed by illegal nodes, the summary will be wrong.

3) Replaying the old LSU packets. Every packet sent by a same node has only one sequence, other nodes will store the sequence in their local. If the received packet's sequence is not bigger than the one in local, the packet will be dropped.

4) Forging attack. Because other nodes will validate the new node's certificate and Ki, the malicious node must get the correct certificate and a public/private key pair. This is the business of the CA (Certificate Authority).

5) Denial of Service (DoS) attacks. In order to guarantee the responsiveness of the routing protocol, nodes maintain a priority ranking of their neighbors when detecting neighbors. If some nodes send their packets in high frequency, SGSR will reduce their priority. So when malicious nodes broadcast requests at a very high rate, they will be throttled back.

5. Formal Analysis

SGSR's security is based on the assumption "only the legitimate node can get the key certificate from authority". So the malicious node can't get the key certificate, then he can't generate the validate signature which means he can't generate false Topology Message or alter other's routing packets undetectably. And at the same time he also can't pass the identity authentication.

There are two ways for nodes to get its certificates. One is from the certificate authority [8]. We can define one or more certificate authorities (CA) to take charge of signing the legitimate node's certificate. Another is from transitive trust and PGP trust graphs [9]. In this way, each node signs certificates for other nodes. A node can search the network for a chain of certificates leading from the node initiating the query and ending at the node trying to authenticate a message. Of course, such schemes require transitive trust.

Next we present a formal analysis of the identity authentication course and verify that the goals are achieved. The analysis follows the methodology of BAN logic [10]. We follow the notation and inference rules in [11]. The Appendix provides a detail of the notations.

5.1 Initialization Assumption

$$A \mid\equiv \xrightarrow{K_{CA}} CA \quad , \quad B \mid\equiv \xrightarrow{K_{CA}} CA \quad , \quad A \mid\equiv \#(Na) \quad , \quad B \mid\equiv \#(Na)$$

$$A \mid\equiv \phi(\{\xrightarrow{K_B} B\}_{K_{CA}^{-1}}) \ , \ A \mid\equiv \#(\{\xrightarrow{K_B} B\}_{K_{CA}^{-1}}) \ , \ B \ni K_{CA}$$

$B \mid\equiv \phi(\{\xrightarrow{K_A} A\}_{K_{CA}^{-1}})$, $B \mid\equiv \#(\{\xrightarrow{K_A} A\}_{K_{CA}^{-1}})$, $A \ni KCA$

$A \mid\equiv \xrightarrow{KA} A$, $B \mid\equiv \xrightarrow{KB} B$, $A \mid\equiv \#(KTC\_A)$,

$A \mid\equiv \#(KTC\_B)$, $B \mid\equiv \#(KTC\_A)$, $B \mid\equiv \#(KTC\_B)$

$B \mid\equiv CA \mid\Rightarrow \xrightarrow{K_A} A$, $A \mid\equiv CA \mid\Rightarrow \xrightarrow{K_B} B$

5.2 Protocol Idealization

The purpose of the identity authentication is that after three messages exchanged A will believe the message 2's signature is correct and come from B and B believes the signature of message 3 is correct and come from A. In a word, the aims are

$$A \mid\equiv B \ni KTC\_B, B \mid\equiv A \ni KTC\_A, A \mid\equiv B \ni K_B^{-1}, B \mid\equiv A \ni K_A^{-1}$$

The three processes are as follow:

(1) $A \rightarrow B : \{\xrightarrow{K_A} A\}_{K_{CA}^{-1}}, \{N_a\}_{K_A^{-1}}$

(2) $B \rightarrow A : \{KTC\_B, Na+1\}_{K_B^{-1}}, \{\xrightarrow{K_B} B\}_{K_{CA}^{-1}}$

(3) $A \rightarrow B : \{KTC\_A, Na+2\}_{K_A^{-1}}$

5.3 Logical Postulates

(1) Being-Told Rules: $\dfrac{P \lhd (X,Y)}{P \lhd X, P \lhd Y}$

(2) Possession Rules: $\dfrac{P \ni X, P \ni Y}{P \ni (X,Y)}$

(3) Freshness Rules: $\dfrac{P \mid\equiv \#(X)}{P \mid\equiv \#(X,Y), P \mid\equiv \#(F(X))}$

(4) Recognizability Rules: $\dfrac{P \mid\equiv \phi(X)}{P \mid\equiv \phi(X,Y), P \mid\equiv \phi(F(X))}$

(5) Message Interpretation Rules: $\dfrac{P \mid\equiv Q \mid\sim X, P \mid\equiv \#(X)}{P \mid\equiv Q \ni X}$

5.4 Analysis

(1) Now from recognizability rules, we can obtain:

$$\frac{B \mid\equiv \phi(\{\xrightarrow{K_A} A\}_{K_{CA}^{-1}}), B \ni KCA}{B \mid\equiv \phi(\xrightarrow{K_A} A)}$$

B receives Message 1, and then B can get:

$$\frac{B \lhd \{\xrightarrow{K_A} A\}_{K_{CA}^{-1}}, B \ni KCA, B \mid\equiv \phi(\xrightarrow{K_A} A)}{B \mid\equiv CA \mid\sim \phi \xrightarrow{K_A} A}$$

Use the freshness rules:

$$\frac{B \mid\equiv \#(\{\xrightarrow{K_A} A\}_{K_{CA}^{-1}}), B \ni KCA}{B \mid\equiv \#(\xrightarrow{K_A} A)}$$

From the two previous results, we get:

$$\frac{B \mid\equiv CA \mid\sim \xrightarrow{K_A} A, B \mid\equiv \#(\xrightarrow{K_A} A)}{B \mid\equiv CA \mid\equiv \xrightarrow{K_A} A}$$

Now using the jurisdiction rules, we get:

$$\frac{B \mid\equiv CA \mid\Rightarrow \xrightarrow{K_A} A, B \mid\equiv CA \mid\equiv \xrightarrow{K_A} A}{B \mid\equiv \xrightarrow{K_A} A}$$

which means B believes K_A is public key of A.

(2) When A receives the message 2, similarly A can get

$$A \mid\equiv \xrightarrow{K_B} B$$

also A will can see

$$\frac{B \lhd \{KTC\_B, Na+1\}_{K_B^{-1}}, A \mid\equiv \xrightarrow{K_B} B)}{A \mid\equiv B \mid\sim \{KTC\_B, Na+1\}}$$

Use the freshness rules:

$$\frac{A \mid\equiv \#(Na)}{A \mid\equiv \#(KTC\_B, Na+1)}$$

Use the Message Interpretation Rules

$$\frac{A \mid\equiv B \mid\sim \{KTC\_B, Na+1\}, A \mid\equiv \#(KTC\_B, Na+1)}{A \mid\equiv B \ni \{KTC\_B, Na+1\}}$$

$$A \lhd \{KTC\_B, Na+1\}_{K_B^{-1}}, A \mid\equiv \xrightarrow{K_B} B,$$

$$\frac{A \mid\equiv \phi(KTC\_B, Na+1), A \mid\equiv \#(KTC\_B, Na+1)}{A \mid\equiv B \ni K_B^{-1}}$$

so we can say $A \mid\equiv B \ni KTC\_B$, $A \mid\equiv B \ni K_B^{-1}$

(3) When B receives the message 3:

$$\frac{B \lhd \{KTC\_A, Na+2\}_{K_A^{-1}}, B \mid\equiv \xrightarrow{K_A} A)}{B \mid\equiv A \mid\sim \{KTC\_A, Na+2\}}$$

Use the freshness rules:

$$\frac{B \mid\equiv \#(Na)}{B \mid\equiv \#(KTC\_A, Na+2)}$$

Use the Message Interpretation Rules

$$\frac{B \mid\equiv A \mid\sim \{KTC\_A, Na+2\}, A \mid\equiv \#(KTC\_A, Na+2)}{B \mid\equiv A \ni \{KTC\_A, Na+2\}}$$

$$B \lhd \{KTC\_A, Na+2\}_{K_A^{-1}}, B \mid\equiv \xrightarrow{K_A} A,$$

$$\frac{B \mid\equiv \phi(KTC\_A, Na+2), A \mid\equiv \#(KTC\_A, Na+2)}{B \mid\equiv A \ni K_A^{-1}}$$

so we can say. $B \mid\equiv A \ni KTC\_A$, $B \mid\equiv A \ni K_A^{-1}$.

At last, we get the aim

$$A \mid\equiv B \ni KTC\_B, B \mid\equiv A \ni KTC\_A, A \mid\equiv B \ni K_B^{-1}, B \mid\equiv A \ni K_A^{-1}$$

6. Simulation Comparison

To compare the performance between SGSR and LSR, we used GloMoSim to simulate the two routing protocols. GloMoSim is developed by UCLA to simulate the wireless network routing protocol.

The settings of environmental and systemic variable are as follows: The area is 3000 x 3000 m^2, the average speed of the nodes is alterable, the number of the nodes and the connections of the nodes are alterable.

Each node moves randomly. The pause of the waypoint is be set to 5 seconds. In the simulating system, the bandwidth is 2Mpbs and the maximum transport distance is 400m. Each

node's power is the same. MAC layer runs 802.11b and application layer is CBR(Constant Bit Rate).

Figure 3 shows the comparison in consumption of energy between SGSR and LSR. The consumption of the energy doesn't increase notablely in proportion to the number of nodes.

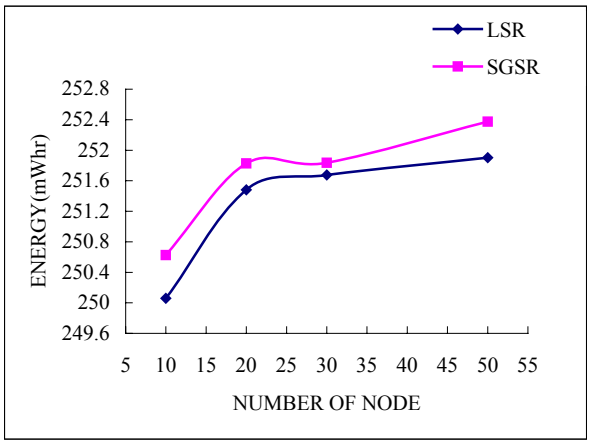

Fig.3 Consumption Of Energy

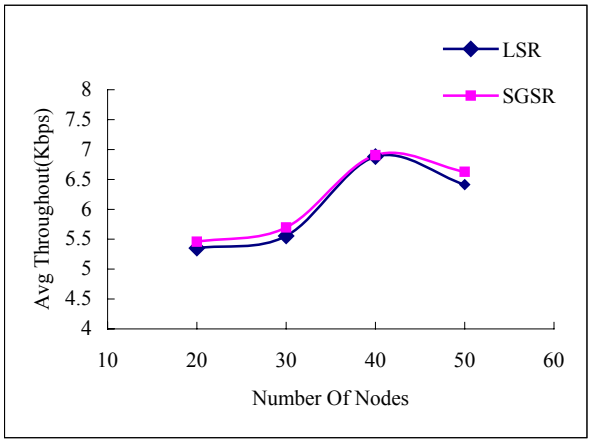

Fig.4 Average throughout of network in the same rate of the connections

Figure 4 shows that the situation of the throughout with nodes increasing when the total network load rate(the number of connections/the number of nodes in CBR)is changeless. The average throughout rises first and descend later. The reason is that, the throughout will rise with the nodes add, but when the node became more and more dense, the collision will be more and more. The average throughout descends with the collision adding. The throughout of SGSR is little bigger than that of LSR.

Figure 5 shows that when the number of nodes is fixed, the average throughout descend with the nodes' movement rate rising. The throughout of SGSR is smaller than that of LSR, because with the nodes move more and more quickly, lose packets rate and collision rate will became bigger and bigger. SGSR adds some fields for authentication or hash link. With the packets' length increasing, the collision will more serious and the average throughout will descend.

As the three pictures show, the efficiency and the cost of the protocol are in an acceptable scope with adding the security mechanisms.

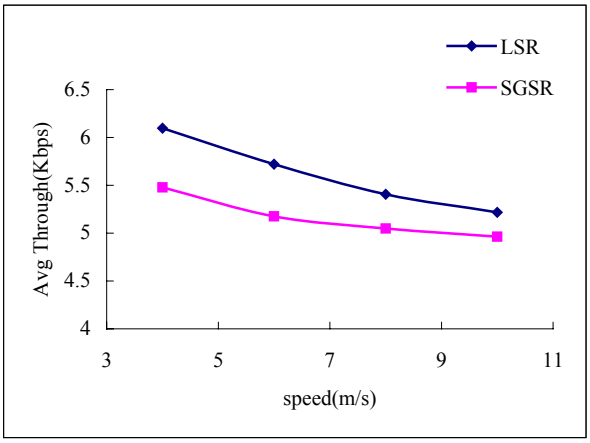

Fig.5 Average Throughput of 50 Nodes with Increasing Speed

7. Conclusions and future work

SGSR for mobile ad hoc networks strengthens the security of LSR. The security mechanisms of SGSR retain robustness along with efficiency.

But SGSR is valid to a single node's attack. As the next step of our research, multi-nodes' attack is our direction.

References

[1] Robertazzi.T.G,Sarachik. Self-organizing communication network[j].IEEE ommunmag,1986, 2~ 5.

[2] Y-C. Hu, A. Perrig, D. B. Johnson. "Ariadne: A Secure On Demand Routing Protocol for Ad Hoc Networks." *MobiCom '02*, Sept. 23-26, Atlanta, GA.

[3] G. Pei, M. Gerla, and T.-W. Chen, "Fisheye State Routing in Mobile Ad Hoc Networks", Proceedings of Workshop on Wireless Networks and Mobile Computing, Taipei, Taiwan, Apr. 2000, 1~ 3.

[4] J. Kong, P. Zerfos, H. Luo, S. Lu and L. Zhang. "Providing Robust and Ubiquitous Security Support for Mobile Ad-Hoc Networks." IEEE ICNP 2001, Riverside, CA, Nov. 2001, 5~ 7.

[5] P. Papadimitratos and Z.J. Haas, "Securing the Internet Routing Infrastructure," *IEEE Communications Magazine*, Vol. 40, No. 10, Oct. 2002.

[6] M. G. Zapata, N. Asokan. "Securing Ad hoc Routing Protocols." *1st ACM WiSe*, Atlanta, GA, Sept. 28, 2002.

[7] P. Papadimitratos and Z.J. Haas. "Secure Routing for Mobile Ad Hoc Networks," *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, San Antonio,TX, January 27-31, 2000

[8] L. Zhou and Z.J. Haas, "Securing Ad Hoc Networks", *IEEE Network Magazin*e, IEEE Press, vol. 13, no. 6, 1999, pp. 24–30.

[9] S. Capkun, L. Buttyan and J.-P. Hubaux, "Self-Organized Public-Key Management for Mobile Ad Hoc Networks", *IEEE Transactions On Mobile Computer*, IEEE Press, vol.2, no.1, 2003,pp. 52–63.

[10] M. Burrows, M. Abadi, and R. Needham, "A Logic of Authentication", *ACM Transactions on Computer System*, vol.8, no. 1, February 1990, pp. 18–36.

[11] L. Gong, R. Needham, and Yahalom, "Reasoning about Belief in Cryptographic Protocols", *Proceeding of the 1990 IEEE Symposium on Research in Security and Privacy,* IEEE Computer Society Press, 1990, pp. 234–248.

[12] Y.-C. Hu, A. Perrig, and D.B. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks", *Proc. 8th Ann. Int'l Conf. Mobile Computing and Networking* (MobiCom2002), ACM Press, 2002, pp.12–23.

Appendix

X and Y are formulas, P and Q are two principals, C is a statement, K/K^{-1} stand for the principal's public and private key. The basic notations used in section 4 are as follows:

- (X, Y): conjunction of two formulas; it is treated as a set with properties of associativity and commutativity.
- H(X): a one-way function of X. It is required that given X it is computationally feasible to compute H(X); given H(X) it is infeasible to compute X; it is infeasible to compute X and X' such that $X \neq X'$ but $H(X) = H(X')$.

Basic Statements

- $P \triangleleft X$: P is *told* formula X.
- $P \ni X$: P *possesses* or is capable of possessing formula X.
- $P \mid\sim X$: P once conveyed formula X.
- $P \mid\equiv \#(X)$: P *believes*, or is entitled to believe, that formula X is *fresh*. That is X has not been used for the same purpose at any time before the current run of the protocol.
- $P \mid\equiv \phi(X)$: P *believes*, or is entitled to believe, that formula X is *recognizable*. That is, P would *recognize* X if P has certain expectations about the contents of X before actually receiving X. P may recognize a particular value (e.g. his own identifier), a particular structure (e.g. the format of a timestamp), or a particular form of redundancy.
- $P \mid\equiv \xrightarrow{K} Q$: P *believes*, or is entitled to believe, that K is a suitable *public key* for Q. The matching *secret key* K^{-1} will never be discovered by any principals except Q or a principal trusted by Q. In this case, however, the trusted principal should not use it to prove identity or to communicate.
- $P \mid\equiv C$: P *believes* or is entitled to believe that C holds.
- $P \Rightarrow C$: P has *jurisdiction* over statement C.

The *horizontal* line separating two statements or conjunctions of statements signifies that the upper statement *implies* the lower one.