A Secure Global State Routing for Mobile Ad Hoc Networks

Chen Jing[1], Cui Guo Hua[1], and Hong Liang[1]

[1] College of Computer, Huazhong University of Science & Technology Wuhan 430074, China
ever_cs@smail.hust.edu.cn

Abstract. The secure operation of the routing protocol is one of the major challenges to be met for the proliferation of the Mobile Ad Hoc networking (MANET) paradigm. Secure Global State Routing Protocol (SGSR) proposed here defines some rules to ensure secure neighbor discovery. Priority is introduced to prevent denial of service attacks. SGSR also can limit the packet in a certain area. So it can be employed as a stand-alone protocol, or fit naturally into a hybrid routing framework. This paper provides formal analysis to illuminate that SGSR is robust against individual attackers. The simulation result shows that the efficiency and the cost of the protocol are in an acceptable scope after adding the secure mechanisms.

Keywords. Ad Hoc; Network Security; Routing Protocol.

1 Introduction

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

P. Cuenca and L. Orozco-Barbosa (Eds.): PWC 2006, LNCS 4217, pp. 228 – 238, 2006.

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 SRP (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 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.

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].

We assume that network links are bidirectional, which means if node A is able to transmit to node B, then B is also able to transmit to A. we also assume that wireless interfaces supporting promiscuous mode operations. Every node is identified by its IP addresses, which can be assigned by many schemes, e.g., dynamically or even randomly. But after the node enters the network and passes the authentication, IP address becomes unchangeable.

Every node has a neighbor information table as table 1:

Table 1. Neighbor information table

IP_i	K_i	SEQ_i	K_{TC}	$Cert_i$

In order to explain SGSR clearly, we define some symbol as table 2.

Table 2. Symbol definition

IP_i	The IP address of node i	MAC_i	The MAC address of node i
K_i	The public key of node i	K_i^{-1}	The private key of node i
$Cert_i$	The Certification of node i	SEQ_i	The sequence of node i
K_{TC_i}	The single hop broadcast key of node i	$\{X\}K$	Using key K to encrypt or decrypt X
$H(X,K)$	Using key K and X to calculate hashing value	$K_{TC\_i}^{j}$	Node j 's K_{TC_i}

3.2 Neighbor Detecting

Each node submits a pair of its (MAC_n, IP_n), to its neighbors by broadcasting hello messages. If node A considers the hello packet coming from a legal node, it will accept the packet and update the neighbor information table. But if node A finds the packet is initiated by a strange node B, it will launch an authentication process.

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. N_a is a random number created by A. The process is as follow:

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

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

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

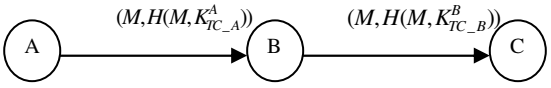

Fig. 1. *A* transmit packets to *C*

Neighbor Detection has the following tasks:

(1) Maintaining the neighbor information table: if neighbor changes IP or uses other 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 which are received from each neighbor, by differentiating the traffic primarily based on MAC addresses, if one neighbor's sending rate is too high, SGSR debases its packets' priority.

3.3 Secure Forwarding Packets

There are three nodes named *A*,*B*,*C* shown in fig.1. *B* is the neighbor of *A* and *C* while *A* and *C* are not neighbors. *A* sends packets to *C*. *M* denotes the packet's content.

(1) $A \rightarrow B : (M, H(M, K^A_{TC\_A}))$

(2) Node *B* uses *M* to calculate $H(M, K^B_{TC\_A})$, if $H(M, K^A_{TC\_A}) == H(M, K^B_{TC\_A})$, goto (3), else drops the packet.

(3) $B \rightarrow C : (M, H(M, K^B_{TC\_B}))$

(4) Node *C* uses *M* to calculate $H(M, K^C_{TC\_B})$, if $H(M, K^B_{TC\_B}) == H(M, K^C_{TC\_B})$, accepts the packet, else drops the packet.

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

3.4 Global State Update and Hops Limitation

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

TYPE stands for the type of packet, R_{HOPS} indicates the number of the hops that the Global State Updates Packet has traveled; RESERVED denotes the field reserved;

HASH_MAXHOPS indicates the hash value[5] of the max hops, *HASH_TRAVERSED* denotes the hash value now, *GSU_SEQUENCE* indicates the sequence of the Global State Updates Packet, NEIGHBOR_TABLE denotes the neighbor information table of the sender, SUMMARY can prevent the malicious node juggling using the method of section 3.3.

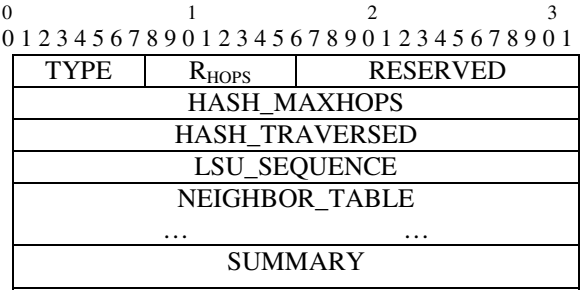

Fig. 2. Global state updates packet (broadcast the state of the node's neighbor)

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

(1) If a node sends the Global State Updates packet, goto (2), and if it forwards the packet, goto (4).

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

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

(4) The node receiving the packet validates the SUMMARY . If it fails, goto (8), else goto (5).

(5) The node uses the *HASH_TRAVERSED* in the received packet to calculate the value of $H^{N-R_{HOPS}}(HASH\_TRAVERSED)$, 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) The node sends or forwards the packet. The process ends.

(8) The node drops the packet. The process ends.

4 Security Analysis

MANET may be suffered from two types of attack. One is active attack. The attackers achieve their illegal aim by modifying, deleting, delaying, inserting the data stream. The other is passive attack[6,7]. The attacker only listens to the information in 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 GSU 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 GSU packets. Every packet sent by the same node has a different sequence, other nodes will store the sequence in their local neighbor

information table. If the sequence is not more than the old one, the packet will be dropped.

(4) Forging attack. Every node must be authenticated by other nodes following the three steps in section 3.2. The vicious node can't get the certification of CA (Certificate Authority), so it will be isolated.

(5) Denial of Service (DoS) attacks. In order to guarantee the responsiveness of the routing protocol, nodes maintain a priority of their neighbors when detecting neighbors. If some nodes send their packets in high frequency, SGSR will reduce their priority. So if 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 that "only the legitimate node can get the key and certificate from authority". So the malicious node can't get the key and certificate, then it can't generate the validate signature which means it 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 by the certificate authority [8]. We can define one or more certificate authorities (CA) to take charge of signing the legitimate node's certificate. The other is by transitive trust and PGP trust graphs [9]. In this way, each node signs certificates for other nodes. A node can search in the network to find a chain of certificates beginning at 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 process 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

$$A \models \xrightarrow{K_{CA}} CA \ , \ B \models \xrightarrow{K_{CA}} CA \ , \ A \models \#(Na) \ , \ B \models \#(Na) \ , \ A \models \phi(\{\xrightarrow{K_B} B\}_{K_{CA}^{-1}}) \ ,$$

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

$$A \ni KCA \ , A \models \xrightarrow{KA} A \ , \quad B \models \xrightarrow{KB} B \ , \quad A \models \#(KTC\_A) \ , A \models \#(KTC\_B) \ ,$$

$$B \models \#(KTC\_A) \ , \quad B \models \#(KTC\_B) \ , B \models CA \models\Rightarrow \xrightarrow{K_A} A \ , \quad A \models CA \models\Rightarrow \xrightarrow{K_B} B$$

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

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

The three processes are as follow:

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

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

(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: $\dfrac{P \models \#(X)}{P \models \#(X,Y), P \models \#(F(X))}$

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

(5) Message Interpretation Rules:

$$\frac{P \models Q \mid\sim X, P \models \#(X)}{P \models Q \ni X}$$

5.4 Analysis

(1) Now from recognizability rules, we can obtain: $\dfrac{B \models \phi(\{\xrightarrow{K_A} A\}_{K_{CA}^{-1}}), B \ni KCA}{B \models \phi(\xrightarrow{K_A} A)}$

B receives Message 1 and gets $\dfrac{B \lhd \{\xrightarrow{K_A} A\}_{K_{CA}^{-1}}, B \ni KCA, B \models \phi(\xrightarrow{K_A} A)}{B \models CA \mid\sim (\phi \xrightarrow{K_A} A)}$

Use the freshness rules: $\dfrac{B \models \#(\{\xrightarrow{K_A} A\}_{K_{CA}^{-1}}), B \ni KCA}{B \models \#(\xrightarrow{K_A} A)}$

From the two previous results, we get: $\dfrac{B \models CA \mid\sim \xrightarrow{K_A} A, B \models \#(\xrightarrow{K_A} A)}{B \models CA \models \xrightarrow{K_A} A}$

Now using the jurisdiction rules, we get: $\dfrac{B \models CA \mid\Rightarrow \xrightarrow{K_A} A, B \models CA \models \xrightarrow{K_A} A}{B \models \xrightarrow{K_A} A}$

which means B believes K_A is public key of A.

(2) When A receives the message 2, similarly A can get $A \models \xrightarrow{K_B} B$

also A will can see $\dfrac{B \lhd \{KTC\_B, Na+1\}_{K_B^{-1}}, A \models \xrightarrow{K_B} B)}{A \models B \mid\sim \{KTC\_B, Na+1\}}$

Use the freshness rules: $\dfrac{A \models \#(Na)}{A \models \#(KTC\_B, Na+1)}$

Use the Message Interpretation Rules

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

$$\frac{A \lhd \{KTC\_B, Na+1\}_{K_B^{-1}}, A \models \xrightarrow{K_B} B, A \models \phi(KTC\_B, Na+1), A \models \#(KTC\_B, Na+1)}{A \models B \ni K_B^{-1}}$$

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

(3) Similarly B can get: $B \models A \ni KTC\_A$, $B \models A \ni K_A^{-1}$.

At last, we get the aim $A \models B \ni KTC\_B, B \models A \ni KTC\_A, A \models B \ni K_B^{-1}, B \models A \ni K_A^{-1}$

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 and the number of the nodes and the connections of the nodes are alterable.

Fig.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.

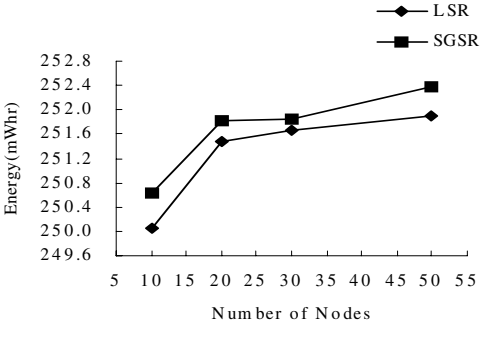

Fig. 3. Consumption of energy

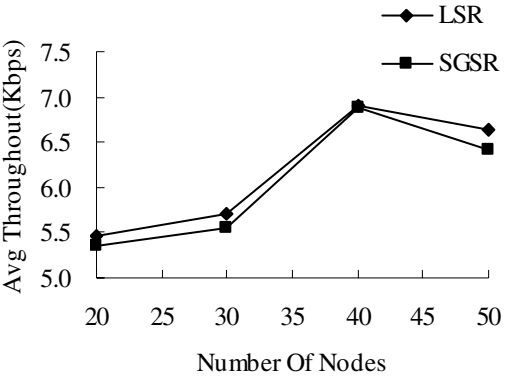

Fig. 4. Average throughput of network in the same rate of the connections

Fig.4 shows that the situation of the throughput with nodes increases when the total network load ratio(the number of connections / the number of nodes in CBR) is changeless. The average throughput rises first and descend later. The reason is that, the throughput will rise with the nodes adding, but when the nodes became more and more dense, the collision will be more and more. The average throughput descends with the collision adding.

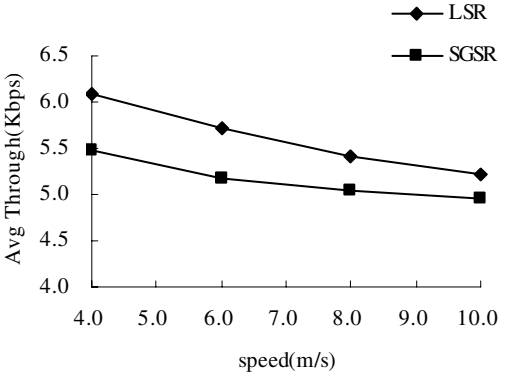

Fig. 5. Average throughput of 50 nodes with increasing speed

Fig.5 shows that when the number of nodes is fixed, the average throughput descends with the nodes' movement rate increasing. The throughput of SGSR is smaller than that of LSR, because with the nodes moving 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 be more serious and the average throughput 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.

7 Conclusions and Future Work

We proposed a Secure Global State Routing Protocol (SGSR) for mobile ad hoc networks. SGSR for mobile ad hoc networks strengthens the security of LSR. The securing of the locally proactive topology discovery process by SGSR can be beneficial for MANET for a number of reasons. The security mechanisms of SGSR can adapt to a wide range of network conditions, and thus retain robustness along with efficiency.

As the next step of our research, we will present a detailed performance evaluation of SGSR, both independently and as part of a hybrid framework (i.e., combine it with a secure reactive protocol), and for various network instances and node processing capabilities.

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 5 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.