

# **Secure Routing in Wired Networks and Wireless Ad Hoc Networks**

Huaizhi Li      Zhenliu Chen      Xiangyang Qin

Chengdong Li      Hui Tan

Department of Computer Science  
University of Kentucky

{hli3, zchen2, xqin0, cli4, htan2}@uky.edu

April, 2002

## Table of Contents

Abstract .....	4
1. Introduction .....	4
2. Securing Routing in Wired Networks .....	5
2.1 Routing Model, Possible Attacks, and Requirements .....	5
2.1.1 Routing Model .....	5
2.1.2 Threats to Routing Protocols .....	6
2.1.3 Requirements for Secure Routing Protocols .....	7
2.2 Securing Distance Vector Routing Protocol .....	7
2.2.1 Add sequence information to updates .....	7
2.2.2 Add predecessor information to updates .....	8
2.2.3 Digitally Signed Updates .....	8
2.3 Secure Link State Routing .....	9
2.3.1 Secure OSPF Routing Protocol .....	9
2.3.1.1 OSPF Background .....	9
2.3.1.2 Security Strong Points of OSPF [Wang98] .....	10
2.3.1.3 Authentication Mechanisms in OSPF Version 2 .....	12
2.3.1.4 Digital Signature Protection of OSPF Routing .....	12
2.3.2 Hashing Chain for Protection of Link State Routing .....	14
2.3.2.1 Hauser's Protocol .....	14
2.3.2.2 Cheung's Protocol .....	17
2.3.2.3 Zhang's Protocols .....	17
2.4 Secure Border Gateway Routing Protocol .....	19
2.4.1 BGP Components .....	19
2.4.2 BGP Threats and Vulnerabilities .....	20
2.4.3 Secure BGP .....	21
3. Secure Routing in Wireless Ad Hoc Networks .....	21
3.1 Attacks to Ad Hoc Routing .....	22
3.1.1 Passive Attacks .....	22
3.1.2 Active Attacks .....	22
3.2 Secure Routing Protocol (SRP) .....	24
3.2.1 Assumptions: .....	24
3.2.2 Basic Idea of SRP .....	24
3.2.3 Detailed PROTOCOL DESCRIPTION .....	24
3.2.4 Scenarios of Possible Security Attacks .....	26
3.3 Authenticated Routing for Ad hoc Networks .....	27
3.3.1 Protocol Description .....	28
3.3.2 Drawbacks .....	29

3.4 Security-Aware Ad-Hoc Routing (SAR) .....	29
3.4.1 Basic Idea of AODV [Perkins] .....	29
3.4.2 Basic Assumptions of SRP .....	30
3.4.3 Description of SRP .....	30
3.4.4 Implementation of SRP .....	30
3.4.5 Remaining Problems .....	31
3.5 Other Protocol .....	31
3.5.1 Watchdog .....	31
3.5.2 Pathrater .....	32
3.5.3 Weaknesses .....	32
4. Conclusion .....	32
References .....	33

# Secure Routing in Wired Networks and Wireless Ad Hoc Networks

## Abstract

*This paper identifies the threats to routing protocols of wired networks and wireless Ad Hoc networks. We discuss the existing secure routing protocols, and point out their drawbacks and vulnerabilities. Finally we conclude with our remarks.*

## 1 Introduction

Routing protocol supports the delivery of packets. It is the fundamental part of network infrastructure. Today network security has attracted more attention than before but the security concern for routing protocols has not been fully aware by the public.

For wired networks, generally the network is partitioned into two levels: intra-domain and inter-domain. Intra-domain routing handles routing procedures within an Autonomous System (AS) or routing domain. And the commonly deployed routing protocols are distance-vector routing protocol or link state routing protocol. Inter-domain routing handles routing procedures that span multiple Autonomous Systems. Nowadays Border Gateway Routing Protocol is used widely. These current routing protocols are mostly designed to deal with simple network failures (e.g., links going up and down, nodes crashing) and can have many vulnerabilities facing malicious intruders. The compromise of routing function can lead to the denial of network service, the disclosure or modification of sensitive routing information, the disclosure of network traffic, or the inaccurate accounting of network resource usage.

For wireless Ad Hoc networks, the situation is even worse. Ad Hoc networks have no pre-deployed infrastructure available for routing packets end-to-end in a network. Nodes communicate with each other without the intervention of centralized access points or base stations, so each node acts both as a router and as a host. Securing Ad Hoc routing presents difficulties not present in traditional network: neither centrally administrated secure routers nor strict policy exist in an Ad Hoc network; the nodes in the networks can be highly mobile, thus rapidly changing the node constellation and the presence or absence of links. So the routing in ad hoc networks is an especially hard task to accomplish securely, robustly and efficiently.

The main objective of this paper is to discuss routing security in wired networks and wireless Ad Hoc networks.

The rest of this paper is organized as follows: Section 2 analyzes secure routing in wired networks, which includes four parts: (1) definition of general routing model, possible attacks to the routing protocols, and requirements for secure routing protocols. (2) Secure distance-vector routing protocol. (3) Secure link state routing protocols, and (4) secure Border Gateway Routing Protocol. Security of Ad Hoc routing protocol is discussed in Section 3: first, possible attacks to Ad Hoc routing protocols are analyzed, then four existing secure Ad Hoc routing protocols are discussed respectively. Section 4 offers concluding remarks.

## 2 Securing Routing in Wired Networks

### 2.1 Routing Model, Possible Attacks, and Requirements

#### 2.1.1 Routing Model

Wang defined a general routing framework [Wang97]. In his model the basic unit of routing protocol is intermediate system (IS) or router.

Forwarding network-layer protocol data units (PDUs) is a major purpose of connection-less network-layer protocol. ISs make forwarding decision based on two sources of information: PDU header (e.g. destination address) and a forwarding table, called the forwarding information base (FIB). An IS builds its FIB using routing information by participating routing protocols. And routing protocol maintains its own routing information base (RIB).

The paper made two important distinctions, which could be easily neglected. One is the difference between forwarding and routing. Forwarding consists of taking a packet, looking at its destination address, consulting the FIB table, and sending the packet in a direction determined by that table. FIB is built from RIB. While routing is the process by which routing protocol determines what goes into RIB. The other one is the distinction between RIB and FIB. RIB is maintained by routing protocol entity, while FIB is maintained by network layer. The figure below illustrates this information flow model.

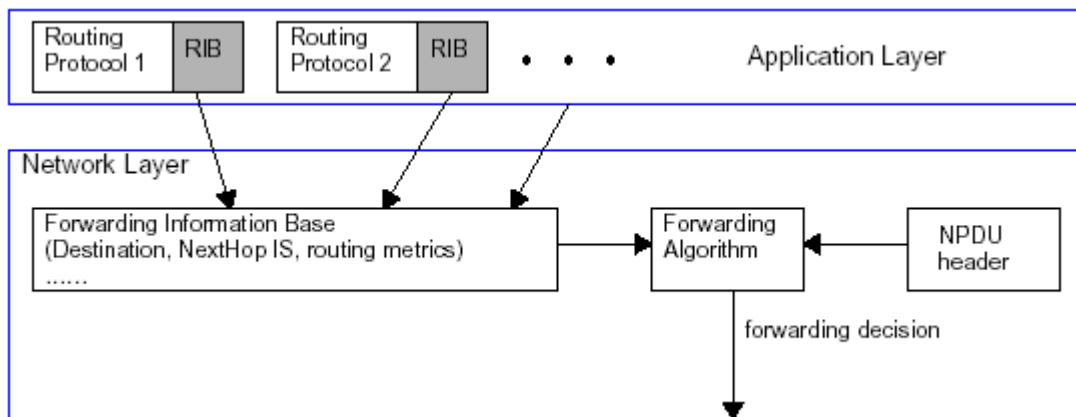


Figure 2.1.1 Routing information flow model inside a IS

This paper defines a simplified model which reflects the core procedures of a routing protocol and it includes five components: (1) Neighbor Acquisition (2) Neighbor Reachability (3) Routing Information Exchange (4) Route Generation and Selection (5) Neighbor Relation Termination.

Here we mainly discuss Route Generation and Selection. A route selection algorithm will determine what goes into the FIB based on the local RIB. The commonly used routing protocols are distance vector routing protocol and link state routing protocol. In distance

vector routing protocol, each IS communicates only with its directly connected neighbors. In link state routing protocol, each node communicates the states of its directly connected links to all the nodes. In practice, if a node possessed complete topology information, Dijkstra's Algorithm is preferred. If only partial information, Bellman-Ford Algorithm is preferred.

This routing protocol framework reflects the essential routing components and in the next section we will discuss threats to routing protocols based on this framework.

## 2.1.2 Threats to Routing Protocols

Our main concern is those attacks, which are trying to improperly modify data, gain authentication, or gain authorization by inserting false packets or by modifying packets. Broadly speaking, threats to routing protocols come mainly from two sources, external and internal. External threats come from outside intruders who are non-participants in the protocol. Internal threats come from compromised protocol participants.

### I. Threats From Outside Intruder

An outside intruder could attack a routing domain in various ways. Specific threats include [Wang97]:

1. *Breaking the neighbor relationship*: An intelligent filter placed by an intruder on a communication link between two ISs could modify or change information in the routing updates or even intercept traffic belonging to any data session. For example, if KEEPALIVE message are filtered out, then the neighbor relationship is terminated.
2. *Replay attack*: An intruder could passively collect routing information. Later, the intruder could retransmit "obsolete" routing information messages. If obsolete information is accepted and disseminated, a normal IS could make incorrect routing decision.
3. *Masquerading*: During the neighbor acquisition process, a outside intruder could masquerade an nonexistent or existing IS by attaching itself to communication link and illegally joining in the routing protocol domain by compromising authentication system. The threat of masquerading almost the same as that of a compromised IS.
4. *Passive Listening and traffic analysis*: The intruder could passively gather exposed routing information. Such a attack can not effect the operation of routing protocol, but it is a breach of user trust to routing the protocol. Thus, sensitive routing information should be protected. However, the confidentiality of user data is not the responsibility of routing protocol.

### II. Threats From A Subverted IS

If an IS has been subverted, all information inside the IS is exposed and at risk. The forward information base (FIB) [RFC3222] could be directly manipulated system commands or kernel interfaces and disrupt the network layer decisions and causing misroute or reroute. By seizing the control of an IS, an intruder could add a route entry to FIB which will reroute data traffic to a particular destination. An intruder could also randomly modify FIB to make router misroute, which is a kind of denial of service attack. The compromised IS problem has not received much attention to date, for several reasons.[Wang97] First, there are usually much fewer ISs in a routing domain than hosts, and they are usually under tight control and monitoring of network administrators. Therefore, the ISs have a larger defense perimeter than the ordinary hosts. In other words, the trust of them from humans is high. Second, routing distributed and cooperative in nature (i.e. ISs in a routing domain must coordinate or cooperate to meet their protocol requirement) and thus there is a tradition of trust in routing protocol

design. If a IS was compromised, the trust relationship would be broken. But with the growing size of internet, IS security should be considered as a big issue in the future protocol design.

### **2.1.3 Requirements for Secure Routing Protocols**

Generally speaking, to secure a routing protocol requires that important routing information be authenticated between neighboring routers. Those various kinds of attacks actually take advantage of the lack of authenticity, integrity or confidentiality. Here are the definitions of these services in the context of secure routing protocols [Wang97]:

1. Authentication services are primarily concerned with the providing assurances about the identity of an entity. In a routing protocol context, when a router sends out a routing message, the identity of the originator of the information should be able to be validated.
2. Integrity services ensure that the data being transmitted is consistent with the data being received.
3. No-repudiation services provide irrefutable evidence that a certain event took place.
4. Confidentiality service provides privacy of routing message, which uses encryption to prevent others from knowing what the routing message is. No current routing protocol supports it.

## **2.2 Securing Distance Vector Routing Protocol**

In a distance-vector algorithm, a router knows the length of the shortest-path from each of its neighbors to every destination in the network, and uses this information to compute its own distance and next router (successor) to each destination. Each update message sent by a router to its neighbors contains a vector with one or more entries, each of which specifies as a minimum, the distance to a given destination. So each router sends only summarized information, which is computational result based on reachability information from its neighbors. Well-known examples of routing protocols based on distance-vector algorithms, is the routing information protocol (RIP). Distance vector routing has its inherent problems, for example the routing to infinity” problem. And the speed of convergence is one of the key advantages of its competitor, link state routing [Peterson], which we will discuss later.

The nature of distance vector routing makes it more vulnerable to internal attacks. So security measures must be combined with the routing protocol to ensure correct operation. Smith et al[Smith97a and 97b] developed a securing distance vector routing protocol. The protocol assumes that each router has a public/private key pair, and knows all other routers' public keys. The protocol adopts the following measures to protect routing messages:

### **2.2.1 Add sequence information to updates**

Sequence information, which can be sequence number or timestamp is added to each update to prevent the replay of old routing information. New sequence information is generated for each routing message. An update with old sequence information is dropped. A sequence information must be valid for the life of a given router id. The primary challenge posed by this requirement of a long life is how to prevent sequence information from wrapping around. The primary advantage of sequence numbers compared to timestamps is their significantly longer life. A sequence number can be relatively small and still provide reasonable assurance of not cycling.

## 2.2.2 Add predecessor information to updates

A routing table update of a distance-vector routing protocol consists of one or multiple entries, each specifying a destination and a distance to the destination. By including the information about the second-to-last hop (predecessor) in the path to a destination, the validity and integrity of the entire path from the router to the destination can be verified iteratively using information reported by the routers directly adjacent to the destination and routers immediately adjacent to each intermediate hop in the path. At all time, all the entries of a correct routing table have the property that the length of a path from a router to a destination equals to the distance from the router to the predecessor of the destination plus the length of the link between the destination and its predecessor.

The figure [smith] above illustrates the path traversal using predecessor information.

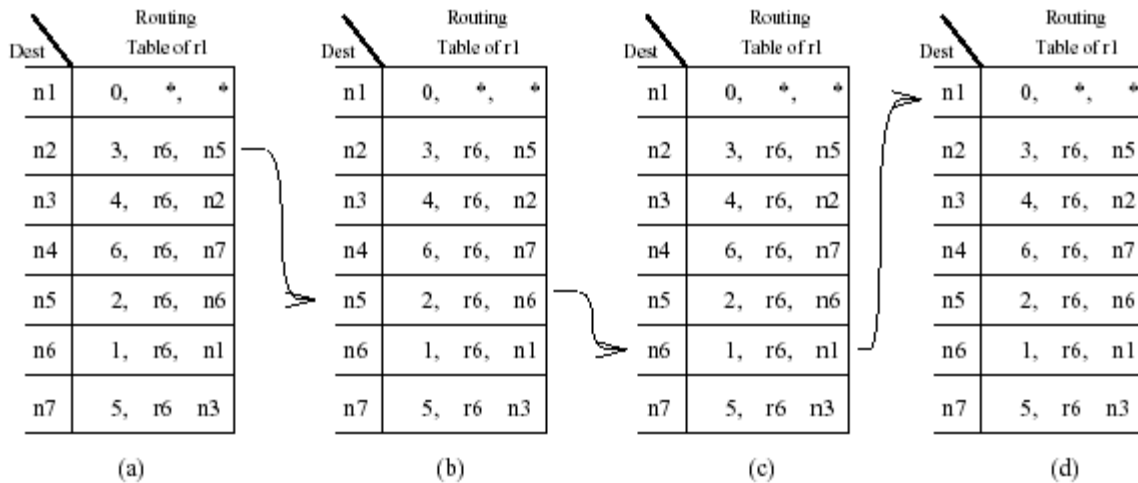


Figure 2.2.1 Path traversal using predecessor information

Let r1 and r6 be the routers, and n1-n7 be the networks. The figure shows the routing table entries at node n1. Each entry specifying the destination, current shortest distance, successor and the predecessor. Infinite distance is represented as ¥ and null path by \*. Node r1 want to determine if a network n7 is in the shortest path to destination n2. Node r1 starts the traversal from the entry for destination n2((a)) and finds that the predecessor to n2 is network n5. Subsequently, n1 walks through the predecessors of its path to n5 and n6 until it reaches the directly connected network n1((d)). From this, node r1 determines n7 is not in the path from r1 to n2. The sequence of predecessors encountered during such a trace represents a path from n1 to n2. This is the implicit path extracted from the predecessor network information.

The addition of predecessor information to each update provides a means of validating a link in the network, which can then be used with the routing table to authenticate the implicit path of a route.

## 2.2.3 Digitally Signed Updates

To ensure the authenticity and integrity of the routing information exchanged between different routers, the originating router digitally signs each update it generates. In addition, to allow receiving routers to validate the signature, an IP address of the originating router must be added to each update. These signatures are used to validate a candidate path to a destination before that path is selected for use.

There are still a few vulnerabilities with this securing routing protocol. A subverted router is still able to forge destination information, delete routing updates, and disclose routing information. There are some complements to the protocol, for example it could be required that a routing authority sign destination information with the IP of the originating router to allow recipients to verify that the originating router is connected to the destination. Vulnerability to the deletion and disclosure of routing updates is inherent in the requirements of routing protocols to trust routers in their handling of routing updates. In some part of network with the high degree of connectivity, it is possible that the deletion of routing updates will not cut off access to destinations, because there are alternative paths. And this kind of attacks is detectable through the correlation of received routing information.

## **2.3 Secure Link State Routing**

Link state routing is another major intradomain routing protocol. The basic assumption for link state routing is similar to those for distance vector routing. That is, each node is able to find out the state of the link to its neighbors and the cost of each link. Each router constructs link state information that describes the link status of the router with its directly connected neighboring routers. This information is disseminated to the entire network by a process called flooding, whereby a router sends link state information to all its neighbors, who in turn forward the same message to their neighbors et cetera. Then after a certain period of time, each router can establish a complete view of the network topology. Based on the received link state information, a router applies a shortest path algorithm to select the best route to all possible destinations. The difference between link state routing and distance vector routing is sending information about one's neighbors to the whole network vs. sending information about the whole network to one's neighbors. Because all routers perform the computation independently on the same set of information, link state routing converges quickly than distance vector routing. The amount of protection that cryptographic algorithms provide is different for the two routing techniques. In both situations, cryptography can be used to protect the information exchanged between neighboring routers from external attackers, for example, digital signature, or keyed-MD5. Internal attacker or internal sources of false routing information is more difficult to prevent. For distance vector routing protocol, each router processes the information received from its neighbors and send back the aggregate information. The result is that it is very hard to validate the received information. And the originator of the information is obscured. It is difficult to protect the authenticity of the source of the information by using cryptographic algorithms, if the source to be authenticated cannot be decided. For link state routing, each router floods the same information to the whole network. The source of the information is still the originator. So the source authenticity and integrity can be protected by cryptography.

In the following sections, we will present several protocols for the protection of link state routing.

### **2.3.1 Secure OSPF Routing Protocol**

#### **2.3.1.1 OSPF Background**

OSPF is a link state routing protocol used within one autonomous system (AS) or routing domain. It creates a global network topology in three phases [Vetter]:

**Phase I: Neighbor and Adjacency Establishment**

A router broadcasts periodically a Hello packet to discover its neighboring routers. After the neighboring routers establish connections, they synchronize their databases with each other through a Database Exchange Process.

**Phase II: Information Exchange by LSA Flooding**

A router assembles the link state information about its local neighborhood into a Link State Advertisement (LSA) and floods it to the whole network.

**Phase III: Calculate Shortest Route using Link State Database**

After a router collects all the link state information, it calculates a shortest path tree with itself as the root by using Dijkstra algorithm and forms a complete structure of routing in the network.

OSPF divides an AS into groups of routers called *areas*. A two level hierarchy among these areas is established, with the top level defined as the backbone area and the second level consisting of many areas attached to the backbone. Routers belonging to a single area are called *internal routers*. Routers that belong to more than one area are called Area Border Routers (ABR). All ABRs belong to the backbone. Various of the routers, within an area or within the backbone, which exchange information with an external autonomous system are known as Autonomous System Boundary Routers (ASBR). The figure below shows an OSPF autonomous system.

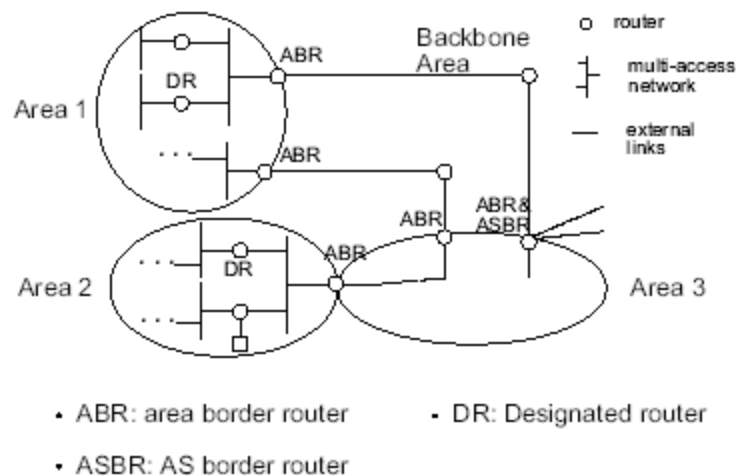


Figure 2.3.1 OSPF Structure

Within each area and within the backbone, a separate copy of the link state algorithm runs. And the topological details of one area are concealed from the rest of the AS. Between the areas and the backbone, OSPF operates more like a distance vector algorithm.

Within each area and within the backbone, a separate copy of the link state algorithm runs. And the topological details of one area are concealed from the rest of the AS. Between the areas and the backbone, OSPF operates more like a distance vector algorithm.

OSPF defines five types of link state advertisement (LSA), which contain the routing information.

Type 1: Each router advertises a Router Links LSAs for its area, which describes the state of each of the router's interfaces in the area.

Type 2: Each multi-access network selects a Designated Router to reduce traffic on the network. The Designated Router generates a Network Links LSA, describing the list of routers connected to the network.

Type 3: Each ABR generates a Summary Link LSA, describing routes to networks outside its attached area (but within the autonomous system).

Type 4: An ABR generates a Summary Link LSA, describing routes to ASBR's outside that area.

Type 5: Each ASBR generates AS External LSAs, describing routes to a destination external to the AS.

Of the five type of LSA, only AS external LSAs are flooded throughout the AS, all others are only flooded within a single area.

A link state update OSPF packet carries one or more LSA instance describing the status of one or more network links. The header of a LSA is shown in the figure below:

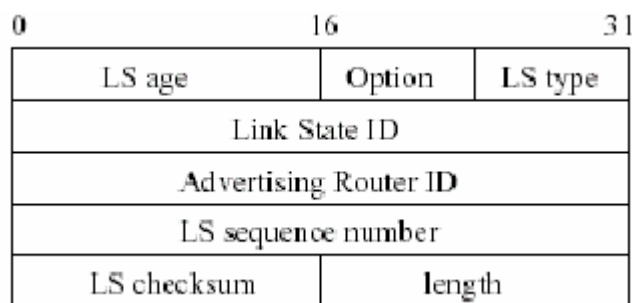


Figure 2.3.2 LSA header

The LS age field is used to keep track of how long a LSA stays in the system. The LS age is set to zero by the originator and increased by every intermediate router that floods it. The age of a LSA is also increased when a router holds it in its topology database. After the LS age reaches the MaxAge, the LSA is purged from the database of all the routers. The sequence number and checksum fields are well known fields. With the age, the sequence number and checksum, a router determines which LSA is more recent.

### 2.3.1.2 Security Strong Points of OSPF [Wang98]

As a routing protocol, some inherent properties of OSPF make it very robust to failures and some attacks.

#### (1) Flooding and information least dependency

As we mentioned above, OSPF uses flooding for the dissemination of LSAs. This makes sure that within the same *area* all the routers have the identical topological database. Even if a router goes down, other routers can still exchange their link state information provided that an alternate path exists.

Furthermore the link state information propagated in the network is the raw message generated by the original router instead of the summarized information from neighbors, which is the situation for distance vector routing. This makes it easy to protect the authenticity of the information.

## **(2) Hierarchy routing and information hiding**

OSPF is a two level routing protocol: intra-area routing and inter-area routing. ABRs connect to backbone and exchange summarized area information. Since intra-area routing depends only on information from within that area, it is not vulnerable to problems out of the area. And problems in one area will not influence the intra-area routing of other areas and inter-area routing among other areas. So hierarchy routing has security advantage.

### **2.3.1.3 Authentication Mechanisms in OSPF Version 2**

The current OSPF specification is OSPF Version 2. It contains two authentication methods. The first one is a simple password scheme. The OSPF header carries a plaintext password so that the routers within the routing domain can share a secret for authentication. It is obvious that it is not secure since the password is transmitted in the clear.

Another much stronger authentication algorithm is cryptographic message digest, e.g. keyed MD5 with assumption that routers on a common network share a secret key. This is a symmetric cryptographic scheme. There are two cases here. If all the routers share the same secret key, then security level is low. If each pair of routers share a secret key, it requires a  $O(N^2)$  set of secret keys. So the key distribution process will be very complex.

Murphy and Badger [Murphy] proposed a digital signature scheme to protect the OSPF routing protocol. Since digital signature is a public key scheme, the number of keys is in the order of  $O(N)$ . Below we will discuss this algorithm.

### **2.3.1.4 Digital Signature Protection of OSPF Routing**

The basic idea of this scheme is to add digital signature to OSPF LSA packet, and use message digest (like keyed MD5) to protect all exchanged messages. The originator of the LSA will sign the message and the signature will stay with the data during the OSPF flooding process. This will protect the message integrity and provide authentication for LSA data. The key management and distribution also make use of a type of signed LSA.

#### **1. Authenticated LSA and Processing**

The content of an authenticated LSA is:

- Normal LSA Header  
Contains fields about: link ID, router ID, Ls age, LSA type, sequence number and checksum (refer to Figure 2.3.2).
- Signature Information  
Information about the signature algorithm, hash algorithm, key size and key identifier.
- Link State Data  
Variable with different types of LSA.
- Signature  
Signature of the originator of the LSA.

The link state data carries important link information, e.g. metric, and is fully protected by the signature. All the fields of the LSA header are protected except the LS age field, because the age field is modified by all routers which propagate the LSA.

When a router receives a routing information LSA, it verifies the signature using the current public key of the originator. Distribution of public keys will be discussed later. If the router does not have the public key for the originator, the signed LSA is discarded. If the verification fails, the LSA is discarded. If the signature verifies, the router stores the LSA in its database for route calculation.

## 2. Key Management and Distribution

The digital signature scheme relies on Public Key Cryptography. It assumes that there exists a Trust Entity in the autonomous system. The Trust Entity has a private/public key pair. Each router is configured with its own pair of private/public key and the public key of the Trust Entity. It obtains a copy of the Trust Entity's certification of the binding between the router's ID and its public key from the Trust Entity. The Trust Entity verifies the binding between a router's ID and its public key according to the policy of the AS. To distribute its public key, an authenticated router disseminates a public key LSA containing its certified public key through OSPF flooding. The content of the public key LSA is:

- Normal LSA Header
- Signature Information
  - Contains the same information as those of authenticated routing information LSA. It also includes the length of the certification field and the length of the signature field.
- Certified Information
  - The information that has been certified by the Trust Entity: the router's ID, the router's role (internal router, ABR or ASBR), the router's public key, key identifier, and the expiration time.
- Certification
  - Signature produced by the Trust Entity of the certified information
- Signature
  - The router's signature of the LSA, excluding the age field.

After receiving this public key LSA, all routers verify the certification by using the Trust Entity's public key and store the public key of the advertising router. They use the public key to verify the authenticated routing information LSA of the advertising router and this public key LSA.

The key identifier is required to be strictly increasing. If a router receives more than one public key, only the one with the greatest key identifier should be accepted for the verification of incoming LSAs. Periodically, keys will be changed, and a new public key of a router will be certified by the Trust Entity. The old keys are revoked implicitly by the public key LSAs—the key identifiers are superseded.

## 3. Remaining Vulnerabilities

The digital signature scheme can prevent external attackers. Since external attackers can not generate correct signature for LSAs, if they intercept the LSA and modify it or inject some malicious information into the system, they can be detected. But some vulnerabilities still remain.

### (1) MaxAge Problem

The age field is the only element of LSA, which is not protected by digital signature. The attacker can modify the age field to MaxAge. This will cause the LSA to be purged prematurely. To cope with this problem, the protocol requires that only the originator of a LSA can flood a LSA which reaches the MAXAge so that this matured LSA can be purged from the databases of all routers, and other routers can only purge a matured LSA from its own database and are not permitted to flood a LSA which reaches MaxAge. But this can only relieve the problem a little bit, because an attacker might modify

the age field so that it is not MaxAge but close to Maxage, e.g. MaxAge-1. So the LSA will still be discarded prematurely.

### **(2) Area Border Routers**

ABRs runs a distance vector routing like protocol. Even with this protocol, the ABRs can generate false information in the summary LSAs about their attached area and inject into the backbone. They can also inject false information about the backbone into their attached areas. One solution to this problem is that an area may have several ABRs, and these ABRs can confirm with each other about the LSAs. If some inconsistencies happen, a warning message can be flooded into the network. But this means additional overhead.

### **(3) Autonomous System Boundary Routers**

The ASBRs can generate false routing information. It is impossible to double check the information as the ABRs do.

### **(4) Internal Routers**

Internal routers can generate incorrect routing information because of faulty configuration or bugs. If an internal router is compromised, then the attacker can control the router. This kind of faulty information and attacks are more difficult to prevent, because the digital signature is correctly generated.

An internal attacker can also generate bogus information, for example, announcing a nonexistent links. The OSPF Dijkstra computation will not consider a link unless the database contains a corresponding LSA from another router at the other end of the connection. If the announced nonexistent connection is to a transit network, no damage will happen without the cooperation of another router. If the announced connection is to a nonexistent stub network or a host, the Dijkstra computation cannot check by the LSA from the other end, because the other end does not exist.

One drawback of the algorithm is that public key cryptography is very expensive and it will slow performance of the router, which should be fast. Hauser [Hauser], Cheung [Cheung], and Zhang [Zhang] proposed three approaches by using one-way hashing to reduce the cost of securing link state routing. We will discuss them in the following parts.

## **2.3.2 Hashing Chain for Protection of Link State Routing**

### **2.3.2.1 Hauser'Protocol**

Hauser et al. [Hauser] proposed a technique for efficient and secure processing of link state information. This approach is based on Lamport'authentication algorithm using hashing chain. The basic idea of Lamport'algorithm is that:

There are two parties: A and B. A generates a secret R and computes a hash chain of length n:

$H^1(R), \dots, H^i(R), \dots, H^n(R)$

Where  $H^i(R) = H(H^{i-1}(R))$ ,  $0 < i < n$ , and the hash function can be MD5 or SHA.

Initially, A sends B the value  $H^n(R)$  and n by some means, for example by mail (The two values are not secret and can be sent in plaintext). When A wants to authenticate himself to B, A sends B  $H^{n-1}(R)$  and B just check if  $H^n(R)$  matches  $H(H^{n-1}(R))$ . Since only A can generate  $H^{n-1}(R)$ , B believes that the other party is A. This one-time authentication can be used n times. The most important feature of this algorithm is that the two parties do not need to share any secret before authentication.

Hauser'algorithm assumes that each router a public/private key pair and its public key has been distributes to all other routers. Each router has k links and the states of each link are UP or DOWN.

For each link two hash functions H and G are used: H for UP state and G for DOWN state. G and H must be distinct but need to have the same security properties.

The secret values for hash chains are  $R_j$  ( $1 < j < k$ ), and each link is assigned a secret value. The values need to be unique. They can all be generated from a single random  $R$ , e.g.,  $R_j = F(j; \text{nodeID}; R)$  where  $F$  is a function and  $[j; \text{nodeID}]$  uniquely identifies the link/node pair. The definition of the two hashing chains is:

$$H^i(R_j) = H(H^{i-1}(R_j)), \text{ and } G^i(R_j) = G(G^{i-1}(R_j)), 0 < i < n, 1 < j < k$$

The hash table of a router is illustrated in the table below:

	$L_1$		...	$L_j$		...	$L_k$	
	up	down	...	up	down	...	up	down
1	$\mathcal{H}^1(R_1)$	$\mathcal{G}^1(R_1)$	...	$\mathcal{H}^1(R_j)$	$\mathcal{G}^1(R_j)$	...	$\mathcal{H}^1(R_k)$	$\mathcal{G}^1(R_k)$
.	.	.	...	.	.	...	.	.
.	.	.	...	.	.	...	.	.
.	.	.	...	.	.	...	.	.
i	$\mathcal{H}^i(R_1)$	$\mathcal{G}^i(R_1)$	...	$\mathcal{H}^i(R_j)$	$\mathcal{G}^i(R_j)$	...	$\mathcal{H}^i(R_k)$	$\mathcal{G}^i(R_k)$
.	.	.	...	.	.	...	.	.
.	.	.	...	.	.	...	.	.
.	.	.	...	.	.	...	.	.
n	$\mathcal{H}^n(R_1)$	$\mathcal{G}^n(R_1)$	...	$\mathcal{H}^n(R_j)$	$\mathcal{G}^n(R_j)$	...	$\mathcal{H}^n(R_k)$	$\mathcal{G}^n(R_k)$

So we can see that each router generates  $2 \times K$  hashing chains.

Initially, each router composes an anchor LSA (ALSA). It contains a set of signed anchor values taken from the bottommost row of the hash table:

$$DS_K[\text{nodeID}; T_n; H^n(R_1); G^n(R_1); \dots; H^n(R_j); G^n(R_j); \dots; H^n(R_k); G^n(R_k)]$$

Where,  $DS_K[ ]$  means digital signed message.  $T_n$  is the timestamp to ensure the timeliness of the message, and will be discussed later.

Upon receipt of an ALSA, a router first verifies the signature over the anchor values. If the signature is correct and the timestamp  $T_n$  is considered fresh, the entire ALSA is stored.

When it is time for an update (either because of time or a change in some link's state), the originating node composes a chained LSA (CLSA)<sub>i</sub> (i means the i<sup>th</sup> CLSA).

For each link  $L_j$  ( $1 < j < k$ ) and for each CLSA<sub>i</sub> ( $1 < i < n$ ), link state flags (LSF<sub>i</sub>) is defined as:

$$LSF_i = [LF_i(1); \dots; LF_i(k)]$$

Where,

$$LF_i(j) = \begin{cases} 1, & \text{if } L_j \text{ is UP} \\ 0, & \text{if } L_j \text{ is DOWN} \end{cases}$$

Link state vector (LSV<sub>i</sub>) is defined as:

$$LSV_i = [LS_i(1); \dots; LS_i(k)]$$

Where,

$$LS_i(j) = \begin{cases} H^{n-1}(R_j), & \text{if } LF_i(j) = 1 \\ g^{n-1}(R_j), & \text{if } LF_i(j) = 0 \end{cases}$$

The i<sup>th</sup> CLSA following the ALSA is  $[\text{nodeID}; i; T_i; LSF_i; LSV_i]$ . Every receiving router is assumed to store an earlier CLSA, CLSA<sub>p</sub>. In most cases  $p = i - 1$ , which means that the receiving node has not missed any CLSAs since CLSA<sub>p</sub>. The case of  $i - p > 1$  indicates that the router missed  $(i-p-1)$  CLSAs and, if  $i = p$ , CLSA<sub>p</sub> is a duplicate.

A router processes a received CLSA<sub>i</sub> in the following steps:

1. Looks up the current entry for nodeID
2. Validates T<sub>i</sub> and i:  
Checks that T<sub>i</sub> is reasonably close to current time,  $i > p$  and  $T_i > T_p$  (last stored timestamp of CLSA<sub>p</sub>.)

3. For each link L<sub>i</sub> reflected in CLSA<sub>i</sub> ( $0 < j < k$ ):

- a) if state unchanged ( $LF_i(j) = LF_p(j)$ ), compute:

$$g^{i-p}(LS_i(j)) \quad \text{if } LF_i(j) = 0$$

$$H^{i-p}(LS_i(j)) \quad \text{if } LF_i(j) = 1$$

and compare to LF<sub>p</sub>(j); reject if mismatch.

- b) if state changed ( $LF_i(j) \neq LF_p(j)$ ), compute:

$$g^i(g^{n-i}(R_j)) \quad \text{if } LF_i(j) = 0$$

$$H^i(H^{n-i}(R_j)) \quad \text{if } LF_i(j) = 1$$

and compare to LF<sub>0</sub>(j); reject if mismatch.

After the entire LSV<sub>i</sub> is verified, the previous link state vector (LSV<sub>p</sub>) is replaced by LSV<sub>i</sub>.

### Clock Synchronization

Loose clock synchronization is necessary for the protocol. Assume  $t$  is the interval between successive CLSAs. Below gives an analysis

1. At T<sub>i</sub>, CLSA<sub>i</sub> contains:  $LF_i(j) = 1, H^{n-i}(R_j)$

2. At T<sub>i+1</sub>, CLSA<sub>i+1</sub> indicates a status change:  $LF_{i+1}(j) = 1, g^{n-i-1}(R_j)$

3. At T<sub>i+2</sub>, CLSA<sub>i+2</sub> contains:  $LF_{i+2}(j) = 1, H^{n-i-2}(R_j)$

So the state history of link L<sub>j</sub> is: UP, DOWN, UP. If a malicious node records all three CLSAs, he can calculate  $g^{n-i}(R_j) = g(g^{n-i-1}(R_j))$  after step 2, and distribute:

$$LF_i(j) = 0, g^{n-i}(R_j)$$

If a router does not synchronize well with the originator, for example with a clock drift of more than ( $3' t$ ), he will accept the forged CLSA as fresh and authentic. This will cause the link L<sub>j</sub> recorded as DOWN while it is actually UP.

So the protocol can only permit clock drift of no more than ( $2' t$ ).

### Other Limitations

Very frequent state changes:

If the state of links and node change very often, the protocol becomes unworkable, because the routers must be very tightly synchronized.

Multiple-valued link state:

The protocol assumes that the link state is binary: UP or DOWN. But sometimes routing protocols express link state as available bandwidth, for example as a percentage of the link's current capacity. It can be a value between 0 and 100%. The protocol cannot be used in this situation.

### 2.3.2.2 Cheung'Protocol

Cheung [Cheung] improved Hauser'algorithm. He developed an optimistic link state verification algorithm (OLSV). The assumption of this algorithm is:

1. There exists a secure public key distribution protocol. Every router has a public/private key pair and knows the public keys of all other routers.
2. The clocks of all routers are loosely synchronized. The maximum clock drift of any two routers is bounded by  $\epsilon$ (a small value).
3. There exists a one-way hash function, which will be used to generate hashing chain. It can be MD5 or SHA.
4. A secure MAC scheme is used, with  $MAC_k(m)$  denoting the MAC generated by using a key  $k$  on a message  $m$ .

The sender process generates a secret  $r$  and constructs a hash chain of length  $l$  using  $r$  and a hash function  $H$  with the similar process in Lamport'algorithm. Then the sender composes a key-chain anchor (KCA), which consists of the router id, the current time  $T$ , and  $H^l(r)$  and signed with the private key of the router. The signed KCA message  $(id, T, H^l(r), DS_k(id, T, H^l(r)))$  is disseminated to other routers by flooding.

The quantities  $H^{l-i}(r)$ ,  $1 < i < l$ , are used as keys to generate MAC for LSAs. The signed LSA message is  $(LSA, i, MAC_{H^{l-i}(r)}(LSA, i))$ , and it is flooded to other routers before time  $(T + iD - t)$ , where  $D$  is the duration of the time intervals between consecutive key releases and  $t$  is value and  $t > ae$ ,  $a$  is a factor. A *hash-chained key* (HCK) message  $(id; i; H^{l-i}(r))$  is released to other routers at time  $(T + iD)$ . So the signed LSA is distributed to all the routers before the key.

When a router receives a KCA with a digital signature  $DS_k(id, T, H^l(r))$ , it verifies the authenticity of the KCA using the public key of the originator. A verified KCA with  $T$  reasonably close to the current clock value of the originator is accepted and stored. When a router receives a signed LSA  $(LSA, i, MAC_{H^{l-i}(r)}(LSA, i))$ , it optimistically accepts it, if the receiving time is less than  $T+iD-\epsilon$ . When a HCK message  $(id; i; H^{l-i}(r))$  is received, the authenticity of the HCK is verified by applying the hash chain. A verified HCK message  $(id; i; H^{l-i}(r))$  is then used to verify the authenticity of LSA. If the LSA can be verified, it will be stored in the database of the router for the calculation of shortest path later.

The OLSV scheme can handle multiple-value link state, so it has advantage over Hauser's scheme. But it still requires clock synchronization among the routers. The signed LSA is flooded before  $(T + i\Delta - \tau)$ , and the key is released at  $(T + i\Delta)$  in plaintext. If an attack records the messages, it can know the key after  $(T + i\Delta)$ . So if the clock drift is larger than  $\tau$ , the attacker can modify the signed LSA and the router with slow clock will accept it as authentic.

### 2.3.2.3 Zhang'Protocols

Zhang [Zhang] proposed two approaches for the protection of routing protocol by using one-way hashing. One method needs loose clock synchronization, and the other one does not require this condition. The basic idea is that: for a message  $m$  and a one-way function  $f(\cdot)$ , assume  $m' = f(m)$  and  $m'$  is of  $l$  bits long. Some bits of  $m'$  are 1s, and some bits are 0s. If a counter is used to count the number of 1s in  $m'$ , the length of the counter should be  $(\lfloor \log_2 l \rfloor + 1)$ . Concatenate  $m'$  with the count field and assume the total length is  $n$  ( $n = l + \lfloor \log_2 l \rfloor + 1$ ). Sign each bit of the  $n$  bits long information with a hash value and



- Sender
  1. Generates one-time secret components  $x_j$ ,  $j = 1, \dots, n$  and computes public key  $P' = h(h(x_1)|\dots|h(x_n))$ .
  2. Generate a  $n$ -bit binary string  $g$  by concatenating  $f(\text{Mi}|P')$  with a count field.
  3. Form one-time signature  $S$  by concatenating signature  $s_j$ ,  $j = 1, \dots, n$ , where,

$$s_j = \begin{cases} h(x_j), & \text{if } g_j = 0 \\ 0, & \text{if } g_j = 1 \end{cases}, \text{ } g_j \text{ is the } j\text{th bit of string } g.$$

4. Send out  $(\text{Mi}|P')$  with one-time signature  $S$ .
  5. Update  $x_j$  with  $x_j'$
- Receiver
    1. Generate a  $n$ -bit binary string  $g$  of the received message  $\text{Mi}$  by concatenating  $f(\text{Mi}|P')$  with a count field.
    2. Compute  $V = h(v_1|\dots|v_n)$ , where,

$$v_j = \begin{cases} r_j, & \text{if } g_j = 0 \\ h(r_j), & \text{if } g_j = 1 \end{cases}, j = 1, \dots, n$$

where,  $r_j$  is the received  $j$ th signature component and  $g_j$  is  $j$ th bit of string  $g$ .

3. If  $V = P'$ , accept the message and update  $P$  with  $P'$ .

The advantage of COSP is that if a router misses some messages from other routers, it can easily catch up, since it can authenticate the received signature by using the anchor values. While in IOSPP if a router misses a message from another router, they have to re-setup. So COSP is more robust, but it requires clock synchronization.

We have discussed security problems of two intra-domain routing protocols: distance-vector routing protocol and link state routing protocol. In the next part, we will analyze an inter-domain routing protocol: **Border Gateway Routing Protocol**.

## 2.4 Secure Border Gateway Routing Protocol

Inter-domain routing protocols are designed to perform policy-based routing among Autonomous Systems (AS), which consists of many routers grouped into management domains.

### 2.4.1 BGP Components

There are four basic components in a BGP system: speakers, peers, links, and border routers. A BGP speaker is a host in an AS, which is essentially a spokesperson for the AS. BGP peers are two BGP speakers that form a connection and engage in a BGP dialog. A BGP peer is either an internal or external peer, depending on whether it is in the same or a different AS as the reference BGP speaker. The connections between BGP peers are called links, with internal and external links being defined similarly to internal and external peers. BGP links are formed using a reliable transport protocol such as TCP. This eliminates the need to implement transport services such as retransmissions, acknowledgments, and sequence numbers in the routing protocol. A border router is a router with an interface to a physical network shared with border routers in other autonomous systems. Similar to BGP speakers,

border routers are either internal or external. Note that BGP speakers need not be border routers (or even routers of any kind). It is possible that a non-routing host could serve as the BGP speaker, gathering routing information from internal or other external routing protocols, and advertising that information to internal and neighboring external border routers.

BGP speakers exchange UPDATE messages to advertise route changes within each AS. The format of UPDATE is shown in the figure below [Smith96]:

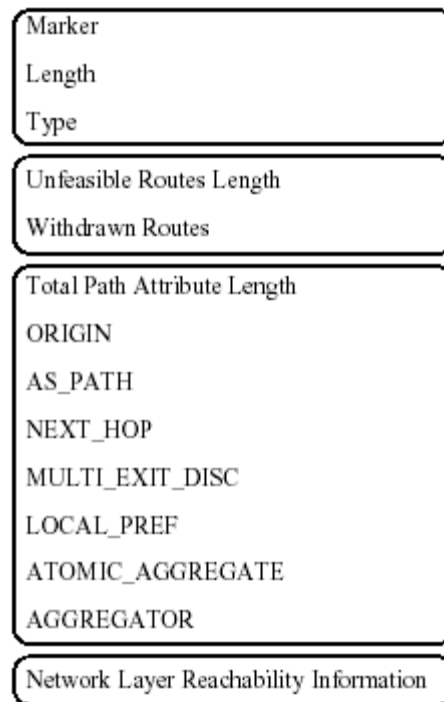


Figure 2.4.1 BGP UPDATE format

BGP also uses KEEPALIVE and NOTIFICATION messages for maintenance of link status. For example, on detection of corrupted information of a link, the link is terminated using a NOTIFICATION message.

## 2.4.2 BGP Threats and Vulnerabilities

BGP is a critical component of the Internet's routing infrastructure. However, it is highly vulnerable to a variety of attacks due to the lack of a scalable means of verifying the authenticity and authorization of BGP control traffic.

We assume that an intruder can be located at any point in the network through which all traffic of interest flows, and that the intruder has the capability to fabricate, replay, monitor, modify, or delete any of this traffic. Interpreting this description for a BGP environment, we identify the following four general classes of intruders: subverted BGP speakers, unauthorized BGP speakers, masquerading BGP speakers, and subverted links.

A subverted BGP speaker occurs when an authorized BGP speaker is caused to violate the BGP protocols, or to inappropriately claim authority for network resources. This typically occurs due to bugs in the BGP software, mistakes in the speaker's configuration, or by causing a BGP speaker to load unauthorized software or configuration information, which can be achieved by many means, depending on the design and configuration of the BGP speaker.

An unauthorized BGP speaker exists when a node that is not authorized as a BGP speaker manages to circumvent any access control mechanisms in place, and establish a BGP link with an authorized BGP speaker. How this is achieved depends on the design and configuration of existing access control mechanisms.

A masquerading BGP speaker occurs when a node successfully forges an authorized BGP speaker's identity. This can be accomplished using the IP spoofing or source routing attacks.

There are a number of forms that a subverted link can take. One is to gain access to the physical medium (e.g. copper or fiber optic cable-plant) in a manner that allows some

### 2.4.3 Secure BGP

Smith proposed the following security countermeasures for BGP [Smith96]:

1. Peer-to-Peer Encryption

Upon establishment of each BGP link, a session key is exchanged by the peers for use in encrypting each BGP message transmitted over that link. The purpose of this encryption is to provide confidentiality of the messages and to provide authenticity and integrity of the KEEPALIVE and NOTIFICATION messages, and of some of the path attributes carried in UPDATE messages. A number of path attributes carried in UPDATE messages are modified in each AS they transit, which include the NEXT\_HOP, MULTI\_EXIT\_DISC, and LOCAL\_PREF attributes. Peer-to-Peer encryption protects authenticity and integrity of these path attributes.

2. Message Sequence Number

A sequence number is added to each message; it is initialized to zero on establishment of a BGP link, and is incremented with each message. On detection of a skipped or repeated sequence number, the BGP link is terminated with a NOTIFICATION message.

3. UPDATE Sequence Number

A sequence number is added to each UPDATE message to protect against replay. An UPDATE message with a sequence number equal to or less than that of a previously received UPDATE message from the same BGP speaker is defined as invalid and dropped.

4. UPDATE Message Digital Signature

Here assumes that each speaker has a public/private key pair and the public key has been distributed to other speakers. To ensure the integrity and authenticity, the unchanging fields of UPDATE messages are digitally signed by the originating BGP speaker. The digital SIGNATURE is calculated over the following fields: UPDATE sequence number, Unfeasible Route Length, Withdrawn Routes, ORIGIN, ATOMIC-AGGREGATE, AGGREGATOR and the NLRI.

The above countermeasures can effectively protect BGP from external attacks. But they cannot prevent internal attacks. An internal attacker can generate legitimate signature. So it is difficult to detect it.

## 3 Secure Routing in Wireless Ad Hoc Networks

Ad Hoc network is a set of wireless mobile nodes forming a dynamic autonomous network through a fully mobile infrastructure. Nodes communicate with each other without the intervention of centralized access points or base stations, so each node acts both as a router and as a host.

In the traditional Internet, routers within the central parts of the network are owned by a few well-known operators and are therefore assumed to be somewhat trustworthy. This assumption no longer holds in an Ad Hoc network since all nodes entering the network are expected to take part in routing. Also, because the links are usually wireless, any security that was gained because of the difficulty of tapping into a network is lost. Furthermore, because the topology in such a network can be highly dynamic, traditional routing protocols can no longer be used. Thus Ad Hoc network has much harder security requirements than the traditional network and the routing in Ad Hoc networks is an especially hard task to accomplish securely, robustly and efficiently.

Several Ad Hoc routing protocols have been proposed, which include AODV, DSR, ZRP, TORA, DSDV, STAR, and others. But all these protocols have security vulnerabilities and exposures, and can easily be attacked. The purpose of this section is to analyze the vulnerabilities of Ad Hoc routing and discuss the existing secure routing protocols.

## **3.1 Attacks to Ad Hoc Routing**

Similar to wired network routing, there are two kinds of attacks toward Ad Hoc routing protocols: passive attacks, and active attacks [Lundberg]:

### **3.1.1 Passive Attacks**

Passive attacks typically involve unauthorized "listening" to the routing packets. That is, the attacker does not disrupt the operation of a routing protocol but only attempts to discover valuable information by listening to the routing traffic.

The major advantage for the attacker in passive attacks is that in a wireless environment the attack is usually impossible to detect. This also makes defending against such attacks difficult. Furthermore, routing information can reveal relationships between nodes or disclose their addresses. If a route to a particular node is requested more often than to other nodes, the attacker might expect that the node is important for the functioning of the network, and disabling it could bring the entire network down.

Other interesting information that is disclosed by routing data is the location of nodes. Even when it might not be possible to pinpoint the exact location of a node, one may be able to discover information about the network topology.

### **3.1.2 Active Attacks**

To perform an active attack the attacker must be able to inject arbitrary packets into the network. The goal may be to attract packets destined to other nodes to the attacker for analysis or just to disable the network. A major difference in comparison with passive attacks is that an active attack can sometimes be detected. This makes active attacks a less inviting option for most attackers.

Next we describe some types of active attacks that can usually be easily performed against an Ad Hoc network.

## 1. Black Hole

A malicious node uses the routing protocol to advertise itself as having the shortest path to nodes whose packets it wants to intercept. In a flooding based protocol such as AODV, the attacker listens to requests for routes. When the attacker receives a request for a route to the target node, the attacker creates a reply where an extremely short route is advertised. If the malicious reply reaches the requesting node before the reply from the actual node, a forged route has been created. Once the malicious device has been able to insert itself between the communicating nodes, it is able to do anything with the packets passing between them. It can choose to drop the packets to perform a denial-of-service attack, or alternatively use its place on the route as the first step in a man-in-the-middle attack.

## 2. Routing Table Overflow

In a routing table overflow attack the attacker attempts to create routes to nonexistent nodes. The goal is to create enough routes to prevent new routes from being created or to overwhelm the protocol implementation. Proactive routing algorithms attempt to discover routing information even before it is needed while a reactive algorithm creates a route only once it is needed. This property appears to make proactive algorithms more vulnerable to table overflow attacks. An attacker can simply send excessive route advertisements to the routers in a network. Reactive protocols, such as AODV on the other hand, do not collect routing data in advance.

## 3. Sleep Deprivation Torture

Usually, attack is practical only in Ad Hoc networks, where battery life is a critical parameter. Battery powered devices try to conserve energy by transmitting only when absolutely necessary. An attacker can attempt to consume batteries by requesting routes, or by forwarding unnecessary packets to the node using, for example, a black hole attack. This attack is especially suitable against devices that do not offer any services to the network or offer services only to those who have some special credentials. Regardless of the properties of the services, a node must participate in the routing process unless it is willing to risk becoming unreachable to the network.

## 4. Location Disclosure

A location disclosure attack can reveal something about the locations of nodes or the structure of the network. The information gained might reveal which other nodes are adjacent to the target, or the physical location of a node. The attack can be as simple as using an equivalent of the trace route command on Unix systems. Routing messages are sent with inadequate hop-limit values and the addresses of the devices sending the ICMP error-messages are recorded. In the end, the attacker knows which nodes are situated on the route to the target node. If the locations of some of the intermediary nodes are known, one can gain information about the location of the target as well.

Papadimitratos [Papadimitratos], Dahill [Dahill], Marti [Marti], and Yi [Yi] developed their own secure Ad Hoc routing protocols separately. We will discuss them respectively in the rest of Section 4.

## 3.2 Secure Routing Protocol (SRP)

Papadimitratos [Papadimitratos] proposed a Secure Routing Protocol (SRP) based on Dynamic Source Routing (DSR).

### 3.2.1 Assumptions:

1. There is a security association (SA) between the source node S and the destination. By using the SA, the principles that participated in the exchange can verify each other.
2. The source and destination share a secret key  $K_{S,T}$ , which is negotiated by the SA.
3. An attack is mounted in this protocol by only two colluding nodes during a single route discovery.

### 3.2.2 Basic Idea of SRP

MAC (Message Authentication Codes) plays an important role in SRP. The source node S sets up the route discovery and constructs a route request packet by a pair of identifiers: a query sequence number and a random query identifier. The source and destination and the unique query identifiers are the input for the calculation of the MAC, along with a shared key  $K_{S,T}$ .

When receiving a route request, if it is a fresh one, the intermediate nodes adds its IP address to the route request and relay the request, so that when query packets arrive at the destination, only a limited amount of state information are needed to be maintained regarding the relayed queries, thus previously seen route requests are discarded at the destination.

When route requests reach the destination T, T verifies the request. Then T constructs a route replies and calculates a MAC covering the route reply contents and returns the packet to S over the reverse of the route accumulated in the respective request packet.

### 3.2.3 Detailed Protocol Description

#### (1) SRP Message Structure

IP Header	
Basic Routing Protocol Packet	
Type	Reserved
Query Identifier	
Query Sequence Number	
SRP MAC	

SRP query includes IP header, basic routing protocol packet and SRP header.

SRP header fields are:

- Type field: 1-byte length, it's used to distinguish the types of SRP messages such as query message or reply message.

- Query Sequence Number (Qseq): 32-bit sequence number increases monotonically, it's used for each destination T to perform secure communications and detect outdated route requests. The sequence number is initialized at the establishment of the SA and it is not allowed to wrap around
- Query Identifier ( $Q_{ID}$ ): 32-bit random number, which is used by intermediate nodes to identify the each outgoing route request.  $Q_{ID}$  is the output of a secure pseudorandom number generator, so its output is statistically indistinguishable and unpredictable by an adversary with limited computational power.
- Message Authentication Code (MAC): a 96-bit long field, which is generated by a keyed hash algorithm, which calculates the truncated output of a one-way or hash function (e.g., SHA-1 or MD5). The one-way function input is the entire IP header, the basis protocol route request packet and most importantly, the shared key  $K_{S,T}$ .

## (2) Route Request

The source node S initiates the route discovery, by constructing a route request packet identified by a pair of identifiers: a query sequence number and a random query identifier. The source and destination and the unique query identifiers are the input for the calculation of the Message Authentication Code (MAC) along with  $K_{S,T}$ , where  $K_{S,T}$  is a secret key only known between S and T.

## (3) Query Handling / Propagation

The intermediate nodes extract the  $Q_{ID}$ , and also extract the source and the destination addresses in order to create an entry in the query table. Queries with  $Q_{ID}$  matching one of the table entries for the same pair of end nodes are discarded. Otherwise, the intermediate nodes re-broadcast the route request. Intermediate nodes measure the frequency of queries received from their neighbors, in order to regulate the query propagation process. On one hand, all nodes self-regulate generation of new route requests, in order to maintain the control traffic overhead low. On the other hand, malicious nodes probably act selfishly and avoid backing off before generating a new route query, or generate queries at the highest possible rate, consuming network resources and degrading the routing protocol performance.

## (4) Route Reply

T first checks the received route request packet to see if it has originated from a node with which it has a security binding. Secondly, T compares Qseq with  $S_{max}$ , which is the maximum query sequence number received from S, within the lifetime of the SA. If  $Qseq \leq S_{max}$ , then the request is discarded as outdated or replayed. Otherwise, T calculates the keyed hash of the request fields. If the output matches the SRP header MAC, the integrity of this request is verified, along with the authenticity of its origin. The destination generates a number of replies to valid requests, at most as many as the number of its neighbors, in order to disallow a possibly malicious neighbor to control multiple replies. The MAC covers the basis protocol route reply and the rest of the SRP header, protects the integrity of the reply on its way to the source and offers evidence to S that the request has indeed reached the destination.

## (5) Route Reply Validation

When source node S receives a Route Reply, S checks the source and destination addresses,  $Q_{ID}$  and Qseq and discards the Route Reply if it does not correspond to the currently pending query. Otherwise, it compares the reply IP source-route with the reverse of the route carried in the reply payload. If the two routes match, S calculates the MAC using the replied route, the SRP header fields and  $K_{S,T}$ . Upon successful verification, S is assured that the request and that the reply are not compromised during on the networks, Thus, the connectivity information is genuine.

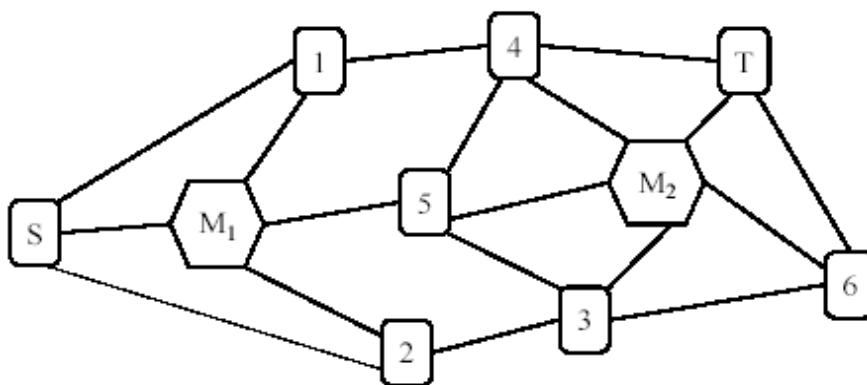
### (6) Intermediate Node Replies

A malicious node can fabricate data packets or route replies and when such routes are used or provided as replies, more unsuspecting nodes cache such invalid routes and may use them in the future, so in order to achieve the required robustness, route caching is not encouraged in general and intermediate nodes are not required to provide route replies. If an intermediate node N has an active route to destination T and a SA exists between source S and N, N can generate a route reply. And this is the only situation that a route request does not reach the destination.

### 3.2.4 Scenarios of Possible Security Attacks

The paper also presents that SRP can prevent common attacks to Ad Hoc routing protocol. Below gives a brief discussion.

Example Topology: S wishes to discover a route to T in the presence of two malicious nodes,  $M_1$  and  $M_2$ , as the figure shown below.



S and T: S queries the network to discover one or more routes to T.

$M_1$  and  $M_2$ : malicious intermediate nodes on the networks

Query request: a list of  $\{Q_{S,T}; n_1, n_2, \dots, n_k\}$

Where,  $Q_{S,T}$  denoting the SRP header for a query searching for T and initiated by S. and  $n_i, i \neq \{1, k\}$ , are the IP addresses of the traversed intermediate nodes and  $n_1=S, n_k=T$ .

Route reply: a list of  $\{RS, T; n_1, n_2, \dots, n_k\}$ .

The following is a number of scenarios of possible security attacks by the two malicious nodes.

**Scenario 1:**  $M_1$  receives  $\{Q_{S,T}; S\}$ , and it attempts to mislead S by generating a reply  $\{R_{S,T}; n_1, n_2, \dots, n_k\}$ . SRP can prevent such attack. First SRP regulates that the request reaches the destination disallows any intermediate node to provide a reply in this manner, secondly, since  $M_1$  doesn't know  $K_{S,T}$  and then can not generate the valid MAC, so the false reply packet is discarded.

**Scenario 2:** Assume that  $M_1$  appropriately relays  $\{Q_{S,T}; S, 1, M_1\}$ ; upon arrival of  $\{Q_{S,T}; S, 1, M_1, 5, 4\}$  at T, the reply is generated and routed over the reverse path. When  $M_1$  receives  $\{R_{S,T}; S, 1, M_1, 5, 4, T\}$ , it alters the content and relays  $\{R_{S,T}; S, 1, M_1, Y, T\}$ , where Y is any invented sequence of nodes. But such a reply made by  $M_1$  is discarded by S, due to the integrity protection provided by the MAC.

**Scenario 3:** When  $M_2$  receives  $\{Q_{S,T}; S, 2, 3\}$ , it corrupts the accumulated route and relays  $\{Q_{S,T}; S, X, 3, M_2\}$  to its neighbors, where X is a false invented IP address or, any sequence of IP addresses. This request arrives at T, which constructs the reply and routes it over  $\{T, M_2, 3, X, S\}$  towards S. When node 3 receives the reply, it cannot forward it any further, since X is not its neighbor, and the reply is dropped.

**Scenario 4:** Node  $M_1$  attempts to forward  $\{Q_{S,T}; S, M^*\}$ ; that is, it spoofs an IP address.  $M^*$  forwards the message by using other node's IP address, and such an query is possible and would propagate

through the network and reach T. So S would accept  $\{R_{S,T}; S, M^*, 1, 4, T\}$  as a route. It is apparent that the connectivity information conveyed by such a reply is correct. However, when T sends back a reply message, this message would not reach  $M_1$  since the query message does not include the IP address of  $M_1$ , so such an attack is temporary and the malicious node would not achieve anything more than its placement on a potential  $S \rightarrow T$  route, which would have been possible in the first place, without any IP spoofing.

**Scenario 5:** Assume that  $M_1$  attempts to return a number of replies, each with a different spoofed IP address, namely,  $M_1, M_{i+1}, \dots, M_{i+j}$ , i.e., an "extension" of Scenario 7. This would lead S to believe that a multitude of possible routes to T exist, although, in reality, all of these routes are controlled by  $M_1$ . To against to such attacks,  $M_1$  is not allowed to generate replies,  $M_1$ 's neighbors relay only one route request, with specific source and target nodes and query identifier. For example, nodes 1,3 and 5 will relay the first of such queries and drop subsequent packets as previously seen requests, thanks to the broadcast channel. If  $M_1$  modified the query identifier, the forged query would be forwarded, but T would detect the alteration, due to the MAC, and drop the request.

**The only possible attack** against the protocol would be if nodes colluded during the two phases of a single route discovery. In such a case, they would manage to make the source node to accept partially false routing information. For example, when  $M_1$  receives the route request, it can tunnel it to  $M_2$ ; i.e. discover a route to  $M_2$  and send the request encapsulated in a data packet. Then,  $M_2$  broadcasts a request with the route segment between  $M_1$  and  $M_2$  falsified, e.g.  $\{Q_{S,T}; S, M_1, Z, M_2\}$ . T receives the request and constructs a reply, which is routed over  $\{T, M_2, Z, M_1, S\}$ .  $M_2$  receives the reply and tunnels it back to  $M_1$ , which, then, returns it to S. As a result, the connectivity information is only partially correct (in this example, only the first and last link). However, one pair of colluding nodes can convince S of only a single false path that will include the two nodes. The reason is that  $M_2$  cannot forward a number of requests towards T using spoofed IP addresses, as explained above.

Other drawbacks of SRP are: (1) SRP cannot handle the attack if more than nodes collude during the phases of a single route discovery. (2) Each SRP query can only discover one route, while diverse routes should be set up to ensure robustness.

### 3.3 Authenticated Routing for Ad hoc Networks

Dahill [Dahill] proposed a protocol: Authenticated Routing for Ad hoc Networks (ARAN) which could detect and protect against malicious actions by third parties and peers in the ad hoc environment. ARAN introduces authentication, message integrity, and non-repudiation to an ad hoc environment. The security design for this protocol is intended to manage the special circumstances common to ad hoc networks: quickly changing topologies, low-power devices, and heterogeneous policy requirements of applications. Accordingly, ARAN is composed of two distinct stages. The first stage is simple and requires little extra work from peers beyond traditional ad hoc protocols. Nodes that perform the optional second stage increase the security of their route, but incur additional cost for their ad hoc peers who may not comply (e.g., if they are low on battery resources). ARAN makes use of cryptographic certificates for the purposes of authentication and non-repudiation. It consists of a preliminary certification process, a mandatory end-to-end authentication stage, and an optional second stage that provides secure shortest paths. The optional stage is considerably more expensive than providing end-to-end authentication.

The table below gives variables and notation:

$K_{A+}$	Public-key of node A.
$K_{A-}$	Private-key of node A.
$\{d\}K_{A+}$	Encryption of data d with key $K_{A+}$ .
$cert_A$	Certificate belonging to node A.
t	Nonce issued by node A.
e	Certificate expiration time.
$N_a$	Nonce issued by node A.
$IP_A$	IP address of node A.
RDP	Route Discovery Packet identifier.
REP	REPLY packet identifier
SPC	Shortest Path Confirmation packet identifier.
RSP	Recorded Shortest Path packet identifier.
ERR	ERRor packet identifier.

### 3.3.1 Protocol Description

There are totally twelve steps to implement ARAN:

- 1) T  $\rightarrow$  A:  $cert_A = [IP_A, K_{A+}, t, e]K_T$ .
- 2) A  $\rightarrow$  broadcast:  $[RDP, IP_X, cert_A, N_A, t]K_{A-}$ .
- 3) B  $\rightarrow$  broadcast:  $[[RDP, IP_X, cert_A, N_A, t]K_{B-}, cert_B]$
- 4) C  $\rightarrow$  broadcast:  $[[RDP, IP_X, cert_A, N_A, t]K_{A-}]K_{C-}, cert_C$
- 5) X  $\rightarrow$  D:  $[REP, IP_a, cert_x, N_A, t]K_{X-}$ .
- 6) D  $\rightarrow$  C:  $[[REP, IP_a, cert_x, N_A, t]K_{X-}]K_{D-}, cert_C$
- 7) C  $\rightarrow$  B:  $[[REP, IP_a, cert_x, N_A, t]K_{X-}]K_{C-}, cert_c$
- 8) A  $\rightarrow$  broadcast:  $SPC, IP_X, cert_x, \{[IP_X, cert_A, N_A, t]K_{A-}\}K_{X+}$
- 9) B  $\rightarrow$  broadcast:  $IP_X, cert_x, SPC, IP_X, cert_x, \{[\{[IP_X, cert_A, N_A, t]K_{A-}\}K_{X+}]K_{B-}, cert_B\}K_{X+}$
- 10) X  $\rightarrow$  D:  $[RSP, IP_A, cert_x, N_A, route]K_{X-}$ .
- 11) B  $\rightarrow$  C:  $[ERR, IP_A, IP_X, cert_c, N_b, t]K_B$
- 12) T  $\rightarrow$  broadcast:  $[revoke, cert_t]K_T$ .

During STEP-1 ARAN requires the use of a trusted certificate server T. Before entering the ad hoc network, each node requestes a certificate from T. Please reference Figure 2 for notation.

#### 1. First Stage

STEP-2 to STEP-7 constitute the first stage of APAN which is End-to-End authentication. The goal of this stage is for the source to verify that th intended destination was reached. STEP-2: A source node, A, begins route instantiation to a destination X by broadcasting to its neighbors a route discovery packet (RDP). Each node records the neighbor from which it received the message.(STEP-3) It then forwards the message to each of its neighbors, signing the contents of the message. This signature prevents spoofing attacks. STEP-4: Upon receiving the broadcast, B's neighbor C validates the sianature with the given certificate. C then rebroadcasts the RDP to its neighbors, first removing B's signature.Eventually, the message is received by the destination, X , who replies to the first RDP that it

receives for a source and a given nonce. There is no guarantee that the first RDP received traveled along the shortest path from the source. The RDP that travels along the shortest path will not be received first only if it encounters congestion. In this case, however, a non-congested, non-shortest path is likely to be preferred to a congested shortest path because of the reduction in delay.

The destination unicasts a Reply (REP) packet back along the reverse path to the source. Let the first node that receives the RDP sent by X be node D. (STEP-5) Nodes that receive the REP forward the packet back to the predecessor from which they received the original RDP. All REPs are signed by the sender. Let D's next hop to the source be node C. (STEP-6) C validates D's signature, removes the signature, and then signs the contents of the message before unicasting the RDP to B. (STEP-7)

## **2. Second Stage**

The optional second stage of the protocol ensures shortest paths but is very costly. Sources must first perform the first stage of ARAN in order to learn the certificate of the destination. However, route instantiation has already taken place at the end of stage one and data transfer may begin while stage 2 occurs. The source begins by broadcasting a Shortest Path Confirmation (SPC) message to its neighbors (the same variables are used as in stage 1.) (STEP-8) A neighbor that receives the message, B, rebroadcasts the message after including its own cryptographic credentials. B signs the encrypted portion of the received SPC, includes its own certificate, and re-encrypts with the public key of X. (STEP-9) Once the destination X receives the SPC, it checks that all the signatures are valid. X replies to the first SPC it receives and also any SPC with a shorter recorded path. X sends a Recorded Shortest Path (RSP) message to the source through its predecessor D. (STEP-10)

ARAN is an on-demand protocol. Nodes keep track of whether routes are active. When no traffic has occurred on an existing route for that route's lifetime, the route is simply de-activated in the route table. Data received on an unactive route causes nodes to generate an Error (ERR) message that travels the reverse path towards the source. Nodes also use ERR messages to report links in active routes that are broken due to node movement. All ERR message must be signed. For a route between source A and destination X, a node B generates the ERR message for its neighbor C as the STEP-11.

STEP-12: In the event that a certificate needs to be revoked, the trusted certificate server, T, sends a broadcast message to the ad hoc group that announces the revocation.

### **3.3.2 Drawbacks**

The protocol uses public key cryptography. It is too expensive, especially for Ad Hoc network, which is resource poor. The distribution of public key is a problem, because in Ad Hoc network all nodes are moving, how to select a trusted Certificate Authority (CA) is not clear.

## **3.4 Security-Aware Ad-Hoc Routing (SAR)**

Yi [Yi] developed a generalized SAR protocol for quantifiable secure route discovery and propagation with trust levels and security attributes as metrics. The protocol is implemented over Ad Hoc On-demand Distance Vector Routing (AODV).

### **3.4.1 Basic Idea of AODV [Perkins]**

AODV is a reactive distance vector routing protocol. A route is discovered only when necessary. To

request a route, source node broadcasts a Route Request message (RREQ), which has a unique sequence number. When the RREQ message reaches either the destination or an intermediate node that has a valid route to the destination, a Route Reply message (RREP) is created and unicasted back to the source node. As the RREP propagates back to the source, intermediate nodes receiving the RREP update their routing tables with a route to the destination.

### 3.4.2 Basic Assumptions of SRP

1. The nodes in an Ad hoc network have different security attributes and are classified into different trust levels. The trust level can be decided by an internal hierarchy of privileges in an organization. For example, in battlefield situation the military ranks of the user of the Ad Hoc nodes form an order of trust level.
2. The nodes of same trust level share a secret key.
3. Routing is to find the nodes that match particular security attributes and trust levels.

### 3.4.3 Description of SRP

Two security metrics RQ\_SEC\_REQUIREMENT and RQ\_SEC\_GURANTEE are embedded into the RREQ packet, and change the forwarding behavior of the protocol with respect to RREQs. All RREQs and RREPs are encrypted by the keys shared in the same level. Only nodes that provide the required level of security can generate or propagate route requests, updates, or replies.

#### **Source node:**

Broadcasts a RREQ packet to its neighbors. The RQ\_SEC\_REQUIREMENT field specifies the required security level in the trust hierarchy for the route the source wishes to discover.

#### **Intermediate nodes:**

After receiving an RREQ packet, the protocol first check if the node can satisfy the security requirement. If it does, it adds itself to the intermediate nodes in the current path and modifies the RQ\_SEC\_GURANTEE field to specify the highest security can be achieved currently. Then it either propagates the RREQ packet to its neighbors if it does not know the path to the destination node or replies with a RREP packet, using the reverse path to the source.

#### **Destination node:**

If there is a route in the ad hoc network from source node to destination node, finally, the RREQ packet will reach the destination node. Then the destination node will set the RQ\_SEC\_GURANTEE field of RREP and replies using the reverse path to the source. A route from source to destination nodes is established.

### 3.4.4 Implementation of SRP

Changes to RREQ:

Add a new field RQ\_SEC\_REQUIREMENT, which is set by the source node and indicates the desired level of trust for the path to the destination.

Add a second field to the RREQ packet RQ\_SEC\_GURANTEE, which indicates the maximum level of security afforded by all discovered path. It is updated by every hop during the

route discovery phase.

Changes to RREP:

Add a new field RP\_SEC\_GUARANTEE, which is set by the destination node. The destination node just simply copy RQ\_SEC\_GURANTEEE to it. This field can be sued to determine the security level over the whole path by the source node.

### 3.4.5 Remaining Problems

1. Is the trust level fixed or can be changed?
2. How to distribute key within the same trust level?
3. The protocol can protect against external attacks, since the routing message is encrypted. But it cannot prevent internal attacks.

## 3.5 Other Protocol

Marti [marti] proposed a method to improve the throughput of an Ad Hoc network in the presence of misbehaving node. The solution is two extensions to DSR routing protocol: The first is Watchdog method. It Keeps track of misbehaving nodes; The second is Pathrater method. It avoids routing through such misbehaving nodes.

### 3.5.1 Watchdog

Watchdog is misbehaving node locator running on every node. It is implemented by maintaining a buffer of recently sent packets and comparing each overheard packet with the packet in the buffer to see if there is a match. If so, the packet has been forwarded and the watchdog removes the packet from the buffer. If a packet has remained in the buffer for longer than a certain value, the watchdog increments a failure count for the node responsible for forwarding on the packet. If the count exceeds a certain threshold value, it determines that the node is misbehaving and sends a message to the source notifying it of the misbehaving node.

### 3.5.2 Pathrater

Pathrater run by each node maintains a rating for every other node that it knows in the network. In order to pick the route most likely to be reliable, it computes a path metric by averaging the node ratings on the paths known from DSR, then chooses the path with the highest metric. Node rating calculations is according to the following algorithm: initially the rating equals to 0.5, increment the ratings of all nodes on all actively used paths by 0.01 for each 200ms of time and decrement a node rating by 0.05 on detecting a link break to that node Misbehaving node's rating equals to -100. Further note that instead of using a path with lower number of hops, it use a path with higher reliability rating (out of all paths returned by DSR route request).

### 3.5.3 Weaknesses

Ambiguous collision problem. It is shown in the figure below:

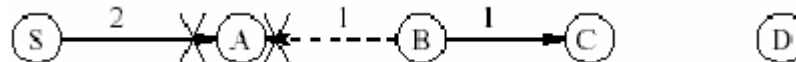


Figure 3.5.3.1: Node A does not hear B forward packet 1 to C, because B's transmission collides at A with packet 2 from the source S.

A packet collision can occur at A while it is listening for B to forward on a packet. There are two possibilities: the collision was caused by B forwarding a packet as it should; Or if B did not forward the packet and the collision was caused by other nodes in A's neighborhood. A cannot distinguish them.

- Receiver collision problem. Node A can only tell whether B sends the packet to C, but it cannot tell if C receives it as shown in the figure below:

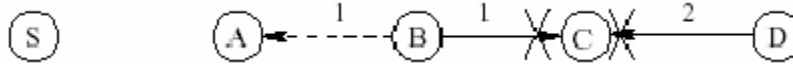


Figure 3.5.3.1: Node A believes that B has forwarded packet 1 on to C, though C never received the packet due to a collision with packet 2.

- False report. A malicious node could attempt to partition the network by reporting that some nodes following it in the path are misbehaving.

A misbehaving node can control its transmission power to evade the watchdog. A node could control its transmission power such that the signal is strong enough to be overheard by the previous node but too weak to be received by the true recipient.

- Multiple nodes can collude to mount a more sophisticated attack. For example, B and C from the Figure 3.5.2 could collude to cause misbehavior: B forwards a packet to C but does not report to A when C drops the packet.

## 4 Conclusion

Above all, we have discussed the secure routing problems of wired network and wireless Ad Hoc network. Routing is the heart of network infrastructure. It controls and manages the "flow" of messages in the network. To set up connection and maintain updated network topology, routers keep exchanging messages about link state, cost and metric. It is possible that attackers can eavesdrop, intercept, or modify these messages or even inject harmful messages into routing infrastructure. Attacker also can comprise a router and fully control it to destroy the network from inside. This kind of inside attack is more destructive. In all, the compromise of routing infrastructure can lead to the denial of service, the disclosure or modification of sensitive routing information, the revelation of network traffic, or the inaccurate accounting of network resource usage.

The general method to fight against these attacks is digital signature: the originator of a routing message signs the message, and the recipients verify the signature so that the authenticity and integrity of the message can be protected. For wired network, Smith [Smith96,97a,97b] and Murphy [Murphy] proposed secure routing protocols by using RSA digital signature. The advantage of RSA signature is that it is simple and easy to be implemented. The disadvantage is that it is too expensive. To reduce latency, routing should be a fast process. The verification of RSA signature costs too many CPU cycles. So there come the one-time signature schemes developed by Hauser [Hauser], Cheung [Cheung], and Zhang [Zhang] respectively, which are much faster than RSA digital signature. The drawbacks of these one-time signature methods are that the signature can only be used once and the clocks of routers need to be synchronized. For wireless Ad Hoc network, the lack of fixed infrastructure support and the frequent changes to network topology make the secure routing problem more difficult. Dahill [Dahill] proposed a signature protocol based on public key cryptography. But Ad Hoc network is resource poor. Public key cryptography is too expensive to be acceptable. Papadimitratos's protocol [Papadimitratos] uses MAC to protect the authenticity and integrity of the message. It is fast. The problem is how a SA between a source and a destination node can be established before a route is set up between them.

Yi [Yi] extended AODV by adding two security attributes to RREQ and RREP and assigning a trust level to each node. Yet how secure and how reliable it is, is not clear. Marti [Marti] proposed a different scheme: based on DSR, each node is assigned a rating and runs a Watchdog and Pathrater, which monitor the operation of other nodes and select the route. This method has some weaknesses. If combined with intrusion detection techniques, the result maybe better.

Routing infrastructure is a very important component of network and has vulnerabilities. Some researchers have developed several schemes to protect it. Yet new cryptographic methods are still needed to ensure its secure operation.

## References

- [Cheung] S. Cheung. An efficient message authentication scheme for link state routing. In 13th Annual Computer Security Applications Conference, 1997.
- [Dahill] B. Dahill, B.N. Levine, C. Shields, and E. Royer, A Secure Routing Protocol for Ad Hoc Networks, Submitted for publication. August 2001. UMass Tech Report 01-37.
- [Hauser] R. Hauser, T. Przygienda, and G. Tsudik. Reducing the cost of security in link-state routing. In Proceedings of the Symposium on Network and Distributed System Security (SNDSS'97), pages 93–99, February 1997.
- [Lundberg] J Lundberg. Routing Security in Ad Hoc Networks, Proceedings of the Helsinki University of Technology, 2000.
- [Marti] S. Marti, T.J. Giuli, Kevin Lai and Mary Baker, Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. Proceedings of MOBICOM 2000, August 2000.
- [Murphy] S. Murphy and M. Badger. Digital signature protection of the OSPF routing protocol. In Proceedings of the Symposium on Network and Distributed System Security (SNDSS'96),.pages 93–102, February 1996.
- [Papadimitra] 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, 2002.
- [Perkins] C. E. Perkins and E. M. Royer. Ad hoc on-demand distance vector routing protocol. In 2nd IEEE Workshop on Mobile Computing Systems and Applications, February 1999.
- [Peterson] L. Peterson. Computer Network: A System Approach, Morgan Kaufmann Publisher, 2000.
- [Smith96] B. Smith and J. Garcia-Luna-Aceves. Securing the border gateway routing protocol. In Proceedings of Global Internet. 1996, November 1996.
- [Smith97a] B. Smith, S. Murthy, and J. Garcia-Luna-Aceves. Securing distance-vector routing protocols. In Proceedings of the Symposium on Network and Distributed System Security (SNDSS' 97), pages 85–92, February 1997.

- [Smith97b] B. Smith, Secure Distance-Vector Routing Protocols, Master Thesis, University of California Santa Cruz, 1997.
- [Vetter] B. Vetter, Feiyi Wang, An Experimental Study of Insider Attacks for the OSPF Routing Protocol, published in 5th IEEE International Conference on Network Protocols, Atlanta, GA, Oct 28-31, 1997.
- [Wang97] F. Wang, B. Vetter, S.F. Wu. Secure Routing Protocols: Theory and Practice. May 1997.
- [Wang98] F. Wang, On the Vulnerability and Protection of OSPF Routing Protocol, IEEE Seventh International Conference on Computer Communications and Networks, Lafayette, LA, Oct 12-15, 1998.
- [Yi] S. Yi, Prasad Naldurg and Robin Kravets. Security-Aware Ad hoc Routing for Wireless Networks, UIUCDCS-R-2001-2241 August 2001
- [Zhang] K. Zhang. Efficient protocols for signing routing messages. In Proceedings of the 1998 Internet Society (ISOC) Symposium on Network and Distributed System Security, San Diego, California, March 1998.