Routing Attacks (Secure Routing)

Panos Papadimitratos
Networked Systems Security Group
www.ee.kth.se/nss

November 18, 2015

Contents

1 Introduction (slides: 1-12)

The definition of routing is often ambiguous. In some texts, it is considered as the forwarding of data - from their source to their destination. Forwarding implies that there is already *communication path*, or a *route* determined by the network for that purpose. Thus, routing, precisely is the *discovery* of routes. Or even the selection of one (in principle, although in some cases more than one) route, from a given source to a given destination. Datagrams (packets) are then forwarded from the source to the destination through a number of intermediate nodes.

The node is an abstraction of any kind of device that is part of the network, such as routers, bridges, gateways, and in the case of flat architectures (e.g., ad hoc networks) laptops, mobile telephones, etc. Nodes are interconnected by direct links, such as actual cables, or they are within wireless range of each other. More generally one can consider virtual links (over multiple physical links or networks, e.g., interconnecting two routers). The connectivity graph is the resultant graph with nodes as the vertexes and links (wire-line or wireless) as the edges. We call a *route*, or *path*, a sequence of connected nodes (simplified view, compared to a sequence of nodes and edges).

The route discovery is the process of figuring out routes from a source to a destination. Depending on the protocol, routes could be expressed explicitly (*explicit route discovery*), or they can be decided in a distributed way (*implicit route discovery*). In the former case, the source node always knows all the intermediate nodes to the destination, while in the latter, the nodes usually know only the next hop for the path to the destination.

The route discovery can be *basic*, that is, return only the connectivity information, the route. But this can be selected/computed in a certain way, to get a preferable route. The simplest such approach is to calculate the *shortest path* in terms of hops, or number of edges or (intermediate) nodes. But we may care about more than that, the qualities of the route, thus run an *augmented* route discovery. In that case, we have a metric, e.g., reliability, delay, cost, current load, etc., for each edge, and the protocol to choose the route that has the best overall metric. E.g., the overall smallest delay, or the cheapest one, or the most reliable one, or the least congested one. There are usually more than one paths connecting a source to a destination, and the route discovery can include a selection of one (or more, as we will see later on) of them.

Usually, nodes in a network are grouped, or to be more precise, the networks are organized in an hierarchical manner. Each network, including a large number of machines and routers is called an Autonomous System (AS): a set

of interconnected networks under common administration, e.g., an Internet Service Provider (ISP), a large organization network. Different ASes may have different routing policies. The routing discovery (and later on, the data forwarding) are done in an hierarchical manner. It is also important to note that for the wired Internet, routing (that is, route discovery) is done *pro-actively*, that is, the routers run periodically the route discovery, and have readily routes that allow them to reach any destination (with the help of other routers of course).

With the route discovery performed and selected the routes, each router stores information that shows exactly how it should route (forward) each data packet it receives. If a packet should routed to network i, then it should forwarded via port j of the router. The information kept indicates the cost for any given destination, and what is the next router, the status of the connection, etc. This is all organized in a table, the *Routing Table*.

Some routing protocols operate only within domains, a collection of hosts and routers that function as a group, and are referred to as *Intra-domain routing* protocols. This is the typical case of an AS. For scalability reasons, an AS can be further organized into areas, with the routing protocol scoped within that area. Other protocols operate within and between domains and are referred to as *Inter-domain routing* protocols. A node that lies over two different areas or domains is usually referred as *border router*. Interior routers and hosts on the perimeter network choose a border router to deliver their traffic towards other areas and domains.

Recall that every machine in the network is accessible via an address, i.e., Internet Protocol (IP) address. IP addresses are logically divided in two parts: the network prefix and host number. The most significant bits of an IP address indicate the network, while the remainder low significant bits show the host number. When a router wants to know the network identifier, it should remove the appropriate number of the least significant bits of the IP address. This is easily done by using a bitwise AND operation on the IP address and a bit string, called *Network Mask*. Through the network number, each router recognizes which (part of the) network the destination machine belongs to and thus how they should forward the packet. Once the subnetwork is recognized and a packet reaches that area, the host number indicates which machine is the exact destination of the packet. (Please revise the CIDR notation for IP addresses if needed.)

2 Routing

2.1 Routing Information Protocol (RIP) (slides: 13-17)

Routing Information Protocol (RIP) is an intra-domain routing protocol. It is an implicit route discovery protocol, i.e., nodes do not know the entire route but only the next hop to the destination. The protocol uses hop count as the metric for paths (shortest path, fewer hops is best) and it calculates paths using the co-called Distance Vector Routing (DVR) algorithm (Bellman-Ford).

Distance Vector routing is one of two major classes of routing protocols[1]. In DVR, each node sends its abbreviated routing information, a vector of distances to all destinations in the network, periodically to the others. Once receiving information from neighbors, each node updates its routing table - essentially, its distance vector along with the next hop neighbor that leads to the specific destination (reachable at the known (shortest) distance. In other words, nodes (routers) do not know (or never learn) the entire topology of the network, only the number of hops to a destination and the next hop to that destination.

One of the most important algorithms that used in DVR for creating the routing table is *Bellman-Ford algorithm*. At the initialization step, each node sets the distance to itself to 0 and the distance to other nodes to ∞. Then, they send information (their distance vector) to their neighbors. After receiving the same from other nodes, each node updates its distance vector, if it found a shorter route to a destination over one of its neighbor (Note: recall, a neighbor is a node that is directly reachable).

2.2 Open Shortest Path First (OSPF) (slides: 18-22)

Open Shortest Path First (OSPF) is another intra-domain routing protocol. It is an explicit route discovery protocol: each node creates a map of the entire network for itself. Besides, it is an augmented protocol, as it considers cost of links in the network. OSPF uses a *Link State Routing (LSR)* algorithm: routing information, Link State Advertisements (LSAs) or Link State Updates (LSUs), announcing the neighbors of the sending router, are broadcasted across the entire network. Each node, after receiving all of the LSAs (or LSUs), constructs an overall view of network topology. Then using the *Dijkstra* algorithm, each node finds the shortest path to all other nodes.

[1] Wikipedia - Distance Vector

2.3 Border Gateway Protocol (BGP) (slides: 23-26

The Border Gateway Protocol (BGP) is the de facto standard inter-domain protocol of the Internet, responsible for the core routing decisions in the Internet. It maintains a table of IP networks or 'prefixes' which designate network reach-ability among ASes[2]. It is explicit, i.e., nodes construct a view of all the network, knowing paths to other networks. In BGP, each link carries a cost determined by commercial agreements between ISPs.

Path Vector routing is the underlying algorithm for BGP where each entry in the routing table contains the destination network, the next router and the path to reach the destination. In Path Vector routing, each node sends an advertisement to all its neighbors and informs them on its available routes, by sending a path vector message that also include the list of the traversed ASes. Each router that receives a path vector message must verify that the advertised path is according to its policy. BGP does not say anything about how a packet will get routed within the AS, nor does it know about the entire network as OSPF does.

2.4 Dynamic Source Routing (DSR) (slides: 27-32)

Dynamic Source Routing (DSR) is an *on-demand* routing protocol for wireless mesh networks or for Mobile Ad-Hoc NETworks (MANETs). On-demand means that a route is created only when two nodes want to communicate with each other, in contrast with what is done for the wireline Internet, where routes are calculated pro-actively. For reactive routing, once a route needs to be discovered, the initiator (source node) sends a packet to its neighbors and ask them about a route to the desired destination. The packet is called Route Request (RREQ).

Once receiving a RREQ, each node appends its address to the packet and forwards the message to all of its own neighbors. Since each node may receive more than one copy of a special request, the initiator includes an id number to the RREQ packet. Each node checks the ID and forwards only one of the (probably) multiple copies of a given RREQ it may receive. It is always possible to cache other requests for route recovery. Once the request reaches to the desired destination, it replies by a message called Route Reply (RREP). This packet includes the selected path from initiator to the target with the purpose to inform the source about the route. To avoid forwarding a request forever (in case that the destination is not accessible or other reasons), a Time To Live (TTL)

[2] Wikipedia - Border Gateway Protocol

number is included in each packet. Each node decreases this value by one when forwarding the packet. This process is called *Route Discovery*.

Once a route is discovered and data are forwarded, an error may occur at a node (source or intermediate), i.e., the node realizes the packet was not received by the next hop node. In this case, a Route Error (RERR) message is generated.

3 Secure Routing (slide: 33)

A variety of attacks relating to routing algorithms (but not only):

Spoofing: A node uses a forged IP address for sending packets to conceal the source of packet or to impersonate another node.

Falsification: This occurs when one node try to send forged data (or other parts of the packet header).

Prefix Hijacking: IP hijacking (sometimes referred to as AS BGP hijacking, prefix hijacking or route hijacking) is the illegitimate takeover of groups of IP addresses by corrupting Internet routing tables[3].

Interference/flooding: A node tries to prevent another node from sending/receiving packets. Generally, a clogging DoS, preventing a system from processing requests drop them.[4].

Man-In-The-Middle (MITM): Consider this as a form of active eavesdropping: the attacker makes independent connections with the victims and relays messages between them, leading them to believe they talk directly to each other (over a private connection), when in fact the entire conversation is controlled by the attacker. The attacker can intercept all messages transferred between the two victims and inject new ones, straightforward in many circumstances[5]; note that injection of cryptographically protected packets implies that the attacker has the relevant cryptographic keys; if not it can only replay packets or parts of.

Black hole: A note can manipulate the routing protocol to create paths that attract all the traffic towards a dead-end node (or an area), thus creating a black hole for the network traffic.

3.1 RIP (slides: 34-40)

In RIP, usually there is no check on the received information. Thus, an attacker can forge the packets and impersonate anyone else in the network. The attacker, by introducing a new route and/or modifying current route advertisements, can create a loop in the network. The original version of RIP has no built-in authentication, and the information provided in a RIP packet is often used without verification. The version 2 of RIP was enhanced with a simple password authentication algorithm, which makes RIP attacks a bit harder.

[3] Wikipedia - IPHijacking
[4] Practical Internet Security
[5] Wikipedia - Man-in-the-Middle Attack

Route filtering is another tool: most routing protocols allow the configuration of route filters that prevent specific routes from being propagated throughout the network. In terms of security, these filters are useful because they help ensuring that only legitimate networks are advertised; while the networks that are not supposed to are never advertised. For example, networks falling within the private address space (RFC 1918) should not be advertised out to the Internet[6].

3.2 OSPF (slides: 41-43)

In OSPF, usually it is easy to falsify a LSA. An attacker can send LSAs on behalf of a compromised node or a router that does not exist at all. There are some defenses against OSPF attacks. OSPF should not be running on the boundary of an AS because it could be targeted from outside the AS perimeter. Authentication can protect the protocol operation. Asymmetric cryptography can easily be used to sign LSAs; a large number of LSAs to be signed and verified can be significant computational overhead. Hash chains can be an efficient secure solution for processing link state information. The idea is as follows:

For two parties, A and B, A generates a secret R and computes a hash chain of length n:

$$H^1(R), \ldots, H^i(R), \ldots H^n(R) \tag{1}$$

Where $H^i(R)$ denotes applying on R a hash function i times. Typically employed hash functions are Message Digest v5 (MD5) or Secure Hash Algorithm (SHA). Initially, A sends to B the values $H^n(R)$ and n by some means and signs them (the two values are not secret and can be sent in plain-text). When A wants to authenticate himself to B, A sends to B the value $H^{n-1}(R)$ and B checks if $H^n(R)$ matches $H(H^{n-1}(R))$. As only A can generate $H^{n-1}(R)$ due to the pre-image resistance[7], B trust that the other party is A. Each value can be used at most once, thus after n authentications the chain must be rebuilt.

3.3 BGP (slides: 44-59)

BGP sustained several attacks in the past years[8]:

1. Pakistan Telecom blocks YouTube: In February 2008, the Pakistan Telecom inadvertently brought down the entire YouTube site worldwide for two hours in the attempt of restricting local access to the site. When

[6] Cisco
[7] See slides from the introductory lectures
[8] Six worst Internet routing attacks

Pakistan Telecom tried to filter the access to YouTube, it sent new routing information via BGP to PCCW, an ISP in Hong Kong that propagated the false routing information across the Internet.

2. ICANN puts root server at risk: In November 2007, the Internet Corporation for Assigned Names and Numbers (ICANN) renumbered the Domain Name System (DNS) root server "L" that it operates. The ICANN failed to notice several unauthorized L root servers operating across the Internet until six months later. By May 2008, ICANN had all the bogus L root servers turned off.

3. Malaysian ISP blocks Yahoo: In May 2004, Yahoo's Santa Clara data-center prefix was hijacked by DataOne, a Malaysian ISP. Network security experts say the incident was malicious, with DataOne intentionally trying to block traffic from Yahoo. The Yahoo attack involved the hijacking of two of its in-use prefixes.

4. Northrop Grumman hit by spammers: In May 2003, a group of spammers hijacked an unused block of IP address space owned by Northrop Grumman and began sending out massive amounts of unwanted e-mail messages. It took two months for the military contractor to reclaim ownership of its IP addresses and get the rogue routing announcements blocked across the Internet. In the meantime, Northrop Grumman's IP addresses ended up on high-profile spam blacklists.

5. Turkish ISP takes over the Internet: On December 24, 2004, TTNet sent out a full table of Internet routes via BGP that routed most Internet traffic through Turkey for several hours that morning. TTNet's routing information claimed that the carrier was the best route to everything on the Internet, according to BGP experts Renesys. The mistake resulted in shifting all traffic from sites such as Amazon, Microsoft, Yahoo and CNN to TTNet.

6. Brazilian carrier leaks BGP table: In November 2008, Brazilian service provider CTBC leaked a full table of routes that could have resulted in an accidental hijacking of other carrier's routes. Thankfully, the BGPMon volunteer service noticed the problem and sent out alerts across the Internet, which minimized the impact of the mistake. Only a few local customers were affected.

This is another hole in BGP: anyone with a BGP router could intercept data headed to a target host, monitor unencrypted Internet traffic anywhere in the world, and even modify it before it reaches its destination[9].

[9] The Internet's Biggest Security Hole

Each AS uses BGP to advertise prefixes that it can deliver traffic to. For example if the network prefix 10.1.0.0/22 is inside AS 4, then that AS will advertise to its provider(s) and/or peer(s) that it can deliver any traffic destined for 10.1.0.0/22.

IP hijacking can occur on purpose or by accident in one of several ways:

- An AS announces it originates a prefix it does not actually originate.
- An AS announces a more specific prefix than what may be announced by the true originating AS.
- An AS announces it can route traffic to the hijacked AS through a shorter route than is already available, regardless of whether or not the route actually exists.

All the above disrupt routing: packets end up being forwarded towards the wrong part of the network and they either enter an endless loop (and discarded) or they are at the mercy of the offending AS. For example, a malicious AS 1 could advertise a more specific network, such as 10.1.0.0/24. In this case, an IP hijacking occurs because AS 1 will be preferred for routing traffic towards this subnetwork.

Typically ISPs filters BGP traffic, allowing BGP advertisements from their downstream networks to contain only valid IP space. However, a history of hijacking incidents shows this is not always the case. For probing further, please refer to: http://safecomputing.umich.edu/events/sumit08/docs/kapela-umich-edited-defcon.ppt

3.3.1 Secure BGP (S-BGP) (slides: 57-68)

Secure BGP (S-BGP) was the first comprehensive routing security solution targeted specifically to BGP [**kent2000secure**]. The S-BGP protocol and its associated architecture are currently under consideration for standardization by the Internet Engineering Task Force (IETF), the organization that provides Internet standards. Implementations of S-BGP exist, and its authors are actively experimenting with its use in operational networks.

A primary element of S-BGP is its use of public key certificates to communicate authentication data. Public key certificates bind cryptographic information to an identity such as an organization. Anyone in possession of the public key certificate can validate information digitally signed with the private key associated with the public key. As the name would imply, the public key is widely distributed, and the private key is kept private [**rivest1978method**].

A Public Key Infrastructure (PKI) is a system for issuing, authenticating and distributing certificates. PKI implements security by validating the data passed between ASes using public key certificates. PKI supports a pair of PKIs used to delegate address space and AS numbers, as well as to associate particular network elements with their parent ASes [**seo2001public**]. One PKI is used to authenticate address allocations through a hierarchy stretching from organizations to the providers and regional registries allocating them space, ultimately leading to ICANN (the ultimate authority for address allocation).

The second PKI is used to bind AS numbers to organizations and organizations to routers in their network. This is accomplished through issued certificates. For example, an organization's AS number is bound to a public key through a certificate.

Statements made by the AS are signed using the associated private key. An entity receiving the signed data verify it came from the AS using the certificate. Because of the properties of the underlying cryptography, no adversary could have generated the signature, and hence it could have only come the signing AS.

All data received by a AS in S-BGP is validated using the certificates in the dual PKI. Address ownership, peer AS identity, path-vectors, policy attributes, and control messages are all signed (and sometimes counter-signed) by the organizations or devices that create them. Because this allows the receiver of the data to unambiguously authenticate the routing information, they can easily detect and remove forged data. However, because of the amount of data and number of possible signers, validation can be extremely costly [**nicol2002challenges**]. These and similar results have raised concerns about the feasibility of S-BGP in the Internet, and led many to seek alternative solutions.

Attestations are digitally signed statements used to assert the authenticity of prefix ownership and advertised routes. Address attestations claim the right to originate a prefix, and are signed and distributed out-of-band. An out-of-band mechanism does not directly use the BGP protocol to transmit information, instead using choose some external interface or service to communicate relevant data. Each address attestation is a signed statement of delegation of address space from one organization or AS to another. The right to originate a prefix is checked through the validation of a delegation chain from ICANN to the advertising AS. Route attestations are distributed within S-BGP in a modified BGP UPDATE message as a new attribute. To simplify, route attestations are signed by each AS as it traverses the network. All signatures on the path sign previously attached signatures (e.g., are nested). Hence, the router can validate not only the path but also that:

a) the path was traversed the ASes in the order indicated in the path, and

b) no intermediate ASes were added or removed by an adversary.

On the other hand, S-BGP introduces a message overhead due to certificate exchanges, Certificate Revocation List (CRL) download, and address attestations. Those informations can instead be stored in internal servers, managed directly by the ISP and, then, accessed through out-of-band channels. Although part of this message overhead can be shrunk, some drawbacks are still in place such as the computational overhead for generating and validating the signatures or the requirement of a PKI.

There are still few cases where an adversary could succeed to attack a network secured by S-BGP. For example, due to missing sequence number in the message exchanged, replay attacks are still possible or updates can be dropped/skipped causing misbehavior of the protocol. More information can be found at http://www.ir.bbn.com/sbgp/

3.3.2 Secure Origin BGP (soBGP) (slides: 69-75)

When considering security in BGP, four goals should be addressed:

- Is the AS originating the destination (prefix) authorized to advertise it? In other words, if a router receives an advertisement for the 10.1.1.0/24 network originating in AS65500, is there any way to verify that AS65500 is supposed to be advertising 10.1.1.0/24?
- Does the AS advertising the destination actually have a path to the destination? In other words, if a router is receiving an advertisement from a BGP peer in AS65501 that it can reach 10.1.1.0/24, is there any way to verify that AS65501 actually has a path to the AS origination 10.1.1.0/24?
- Is the peer advertising the route authorized by the originator, or owner, of the destination, to advertise a path to the destination?
- Does the path advertised by a peer AS fall within the policies the local network administrators have set forward? The most obvious issue is whether or not the AS Path advertised by the peer is an acceptable path to send the traffic along.

However, the second two goals cannot be fully met within an operational internetwork, for many reasons [**white2004considerations**].

Secure Origin BGP (soBGP) seeks flexibility by allowing administrators to trade off security and overhead using protocol parameters. Instead of defining a PKI for authenticating and authorizing entities and organizations in the way of

S-BGP, soBGP uses a sort of Web of Trust. soBGP uses an *EntityCert*, which ties an AS number to a public key (or a set of public keys) corresponding to a private key the AS will be using to sign various other certificates.

The main problem a router faces when accepting an *EntityCert* is knowing whether or not the key carried within the certificate is actually the key of the advertising AS. soBGP resolves this by requiring the *EntityCert* to be signed by a third party, validating that this AS actually belongs with this key. A small number of "root keys" distributed out of band could then be used to validate a set of advertised *EntityCerts*. These are used in turn to build up the database of known good AS/key pairs in the system, allowing even more *EntityCerts* to be validated.

The authorization for an AS to advertise a specific block of addresses is provided through an authorization certificate, or *AuthCert*. An *AuthCert* ties an AS to a block of addresses that the AS may advertise.

Each AS attached to the internetwork builds an *ASPolicyCert*, which contains, primarily, a list of its peers, and signed using the originator's private key. Using this list of transit peers, a map of the internetwork topology may be built. The topology database is used to sanity check received routes: any UPDATE with a path that violates the AS topology is demonstrably bad and dropped. The authors in [**kruegel2003topology**] extend this approach by using other heuristics in detecting anomalous paths (e.g., multiple entrances into core ASes, strange geographic routes, etc.).

Validating signatures is a computationally expensive operation. soBGP tries to mitigate this cost in the presence of limited resources by authenticating long term structural routing elements (such as organization relationships, address ownership, and topology) prior to participating in BGP. Authenticated data is signed, validated, and stored at the routers prior to the establishment of the BGP session, and thus their validation does not introduce significant run-time cost. Transient elements (such as paths) are locally checked for correctness, rather than validated, e.g., adjacent ASes in the path must be reflected in the topology database.

However, soBGP is not fully resilient against integrity, collusion, and replay attacks. For more information on soBGP, refer to [**butler2010survey**] and ftp://ftp-eng.cisco.com/sobgp/index.html

4 Ad Hoc Networks (slide: 76)

Ad-hoc networks are self-organized nodes that communicates without any infrastructure. Those networks need minimal configuration and could be quickly deployed. When one node wants to send a message to another one, all middle nodes cooperate with each other to pass the message through the network. That means each node acts as a router. Ad-hoc network has dynamic topology, since the nodes usually are also mobile. Ad-hoc network was first developed by DARPA in 1970s for military applications. Now, in addition to battlefield applications, they can be used for emergency services, commercial environments, location-aware services, and entertainment. A sensor network is someway related to an ad-hoc network with small differences. It consists of several small devices (called sensors) that senses environmental parameters such as temperature, motion, sound and so on, and cooperate with each other to send their information to a the center. They have tighter constraints on their power consumption, memory size, computational and communicational power.

4.1 DSR (slides: 77-84)

There are some variety of attacks against routing protocols in MANETs:

- One can generate a batch of packets and inject them in the network.
- The attacker can discard routing packets, i.e. RREQ, RREP and RERR; all of them or only a subset of them.
- The attacker can impersonate another node and send a RREQ packet on behalf of that node.
- The attacker can forge a RREP packet and response to the origin of a RREQ packet (impersonate himself as the destination).
- The attacker can modify routing packets en route (insert, delete, or change addresses).
- The attacker can disrupt a link state routing protocol or a distance vector routing protocol. He can disrupt these protocols such that e.g. a longer path appears as a shorter path.

For a routing protocol to be secure, it is necessary to provide the following properties, either for a implicit or an explicit, for a basic or an augmented protocol:

- The protocol should be loop free. It should be guaranteed that for all discovered routes, there is no repetition of nodes in the route.

- All discovered routes should be fresh, i.e. all of the links of the route should be up at least at one point within a time interval (e.g., the route discovery period).

4.2 Secure Routing Protocol (SRP) (slides: 85-92)

Secure Routing Protocol (SRP) Protocol Invocation[10]: A source node (S) initiates a route discovery for a destination node (T) only if no route discovery is under way for the same node T at the time of invocation. Otherwise, a route discovery is performed at a later invocation and only after the conclusion of the ongoing route discovery. The route discovery is triggered when no $S \rightarrow T$ routes are available at S, or it can be triggered by mechanisms independent of the routing protocol.

1. Route Query Generation: S generates a route query or RREQ packet.
   i. The route request includes the querying node S, the sought destination T, a query identifier Q that was not previously used, an authenticator $A = f_K(S, T, Q)$ calculated as a function of the route query fields and a key K, and an empty NodeList.
   ii. The node transmits the route request, i.e., $\mathtt{BcastL}(\mathtt{RREQ})$, and it initializes a ReplyWait timer.
2. Route Query Processing: Each node receiving a RREQ determines if its own identity matches the sought destination. If not, it processes the request either as the querying node or as an intermediate node. Otherwise, it processes the request as the destination.
   (a) Route Query Processing at the Querying Node:
      i. S initializes an empty ForwardList for each RREQ it generates.
      ii. S adds to the ForwardList each neighbor V it overhears relaying RREQ with $\mathtt{NodeList} = \{V\}$.
   (b) Route Query Processing at Intermediate Nodes:
      i. Each V_k node invokes the PreviouslySeen(RREQ) routine to specify if RREQ has been previously processed. If yes, the RREQ is discarded. Otherwise,
      ii. V_k extracts the last entry of the NodeList and verifies this is the address of its precursor V_{k-1}. If not, RREQ is discarded. Otherwise,

[10] How to Specify and How to Prove Correctness of Secure Routing Protocols for MANET

iii. V_k checks the NodeList for duplicate entries; if a loop is detected, RREQ is discarded. Otherwise,

iv. V_k appends its own identity to the RREQ, updating $\mathtt{NodeList} = \{\mathtt{NodeList}, V_k\}$, and BcastL(RREQ).

v. V_k initializes an empty ForwardList for each RREQ it relays. It then adds to the ForwardList each neighbor V it overhears relaying RREQ with $\mathtt{NodeList} = \{\mathtt{NodeList}, V\}$.

(c) Route Query Processing at Destination Node:

i. T invokes the PreviouslySeen(RREQ) routine to check if RREQ has been previously processed. If so, the RREQ is discarded. Otherwise,

ii. T extracts the last entry of the NodeList, verifies that this is the address of its precursor, and discards RREQ if there is a mismatch. Otherwise,

iii. T checks if there is any duplicate entry in NodeList. If a loop is detected, it discards the RREQ; otherwise,

iv. T calculates $f_K(S, T, Q)$ and compares it to A. If they are not equal, RREQ is discarded; otherwise, T generates and returns a route reply to S.

3. Route Reply Generation: T generates a RREP.

i. The RREP packet comprises:

- The querying node S
- The destination T
- The query identifier Q
- A Route list that contains the discovered route and also serves as the information necessary for RREQ to be forwarded across the network towards S. To determine Route, T extracts the identifiers of the intermediate nodes previously accumulated in the RREQ NodeList, namely, $V_1, V_2, \ldots, V_{n-1}$. T stores them in reverse order in the RREP, setting Route = $V_{n-1}, \ldots, V_2, V_1$. And,
- An authenticator $A' = f_K(S, T, Q, \mathtt{Route})$.

ii. The destination transmits the RREQ to the first entry of the Route list: SendL(V_{n-1}, RREP).

4. Route Reply Processing:

i. Each V_k, including S, verifies that its successor V_{k+1} is indeed the node that now forwards the RREP. If not, it discards RREP. Otherwise,

ii. V_k verifies that $V_{k+1} \in$ ForwardList, unless the successor is T. If not, it discards RREP. Otherwise,

iii. V_k checks if there is any duplicate entry in Route; if yes, it discards RREP. Otherwise,

iv. V_k relays the reply to its predecessor, V_{k-1}, i.e., the next entry in the Route list or S; SendL(V_{k-1}, RREP). Once RREP reaches the source,

v. S calculates and compares $f_K(S, T, Q, \mathtt{Route})$ to A'. If there is not a match, S rejects the reply. Otherwise, it accepts the reply, and,

vi. S extracts the Route entries to obtain the $\{V_1, V_2, \ldots, V_{n-1}, T\}$ route.

5. Route Reply Timeout: The ReplyWait timer may expire in either of the following cases:

   (a) no replies from T, in response to the query identified by Q, were accepted by S, or,

   (b) at least one reply from T, in response to the query identified by Q, was accepted by S. In the former case, the route discovery is considered failed, while, in the latter case, the route discovery concludes, and S ignores route replies that are further delayed.

6. Route Discovery Failure: S initiates a new route discovery as in Step 1, using an updated value for the ReplyWait timer (Step 1ii). To calculate this value between ReplyWaitmin and ReplyWaitmax, S invokes an Update(ReplyWait) routine that returns an equal or higher value than the one previously used for the failed route discovery.

7. Route Discovery Conclusion: Upon accepting a RREP from T identified by Q, S considers the discovery concluded after at least ReplyWaitmin seconds elapse from the corresponding query generation, allowing then for a new route discovery, if necessary. If so, the ReplyWait timer is reset, and S invokes Update(ReplyWait) to select ReplyWaitmin as the new route discovery timer value (Step 1ii).

4.3 Secure Link State Protocol (SLSP) (slides: 93-94)

The SLSP is a routing protocol that provides secure proactive topology discovery, which can be multiply beneficial to the network operation. The SLSP can be employed as a stand-alone protocol, or fit naturally into a hybrid routing framework, when combined with a reactive protocol. SLSP nodes disseminate

their link state updates and maintain topological information for the subset of network nodes within R hops, which is termed as their *zone*.

Within each zone, the nodes distribute their public and certified keys through periodical broadcast, so that the receiving nodes validate their subsequent link state updates. Nodes advertise the state of their incident links by broadcasting periodically signed LSUs. The SLSP restricts the propagation of the LSU packets within the zone of their origin node. Receiving nodes validate the updates, suppress duplicates, and relay previously unseen updates that have not already propagated R hops. Link state information acquired from validated LSU packets is accepted only if both nodes incident on each link advertise the same state of the link. The SLSP detailed operations can be found in [**papadimitratos2003secure**].

4.4 Attacks (slides: 95-98)

Even when employing secure protocols, some attacks are still possible. For example an attacker could just relay the messages instead of executing the SRP protocol. In this case, the attacker would behave as a repeater and the protocol will complete its operations. Depending on the metric used, the path that traverses the attacker area could be more convenient and multiple nodes might select it. The attacker, then, would be able to control (part of) the traffic in the network. However, this attempt can be easily defeated by integrating secure neighborhood discovery and hop-by-hop authentication where neighbors nodes authenticate to each other if, and only if, they are in a valid communication range.

Nonetheless, multiple colluding attackers are more difficult to detect since they can use different channels to exchange messages. For example, they could omit the protocol functionality when sending routing request and responses between them, but acting in compliance with the protocol only when communicating with the honest neighbors. This could have the same consequences as in the case where the attacker act as a repeater. However, secure neighbor discovery and hop-by-hop authentication in this case will fail to preserve the correctness of the protocol. Colluding attacker could not even be in the communication range of the radio used for the ad-hoc network, but still able to exchange message through other (e.g., out of bound) channels. For example they could have a direct fiber link between them and *tunnel* the messages over this secondary channel, actually connecting two different parts of the network. The same considerations apply as before, where countermeasures will be ineffective.