Networked System Security

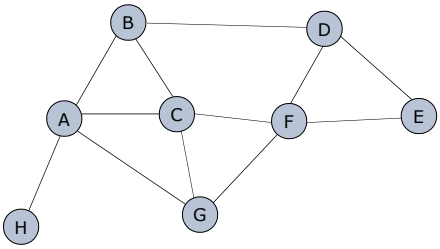
Secure Routing

Panos Papadimitatos
Networked Systems Security Group
www.kth.se/nss

Introduction

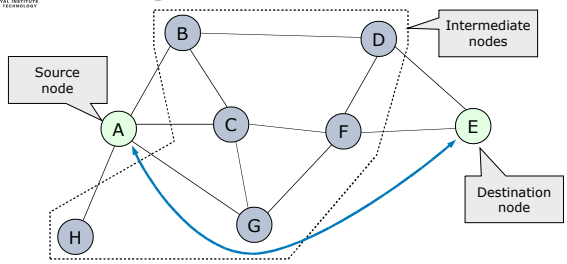
(As far as we know, photo is public domain)

Routing



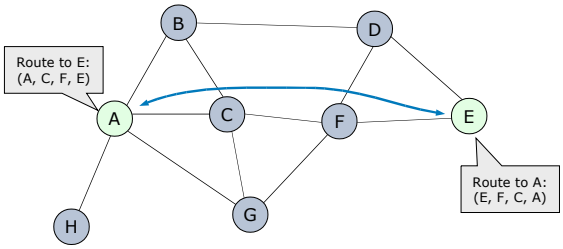
- Connectivity Graph

Routing (cont'd)



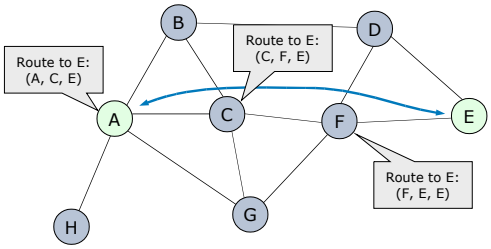
- Route: Sequence of nodes and edges (E.g. {A, G, E})

Routing (cont'd)



- Explicit route discovery
  - Fully, clearly expressed and readily observable route returned by the routing protocol

Routing (cont'd)

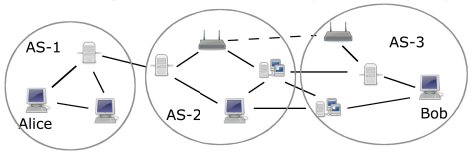


- Implicit route discovery
  - Distributed computation that returns a tuple of the form (current node, relay node, destination node)

Routing (cont'd)

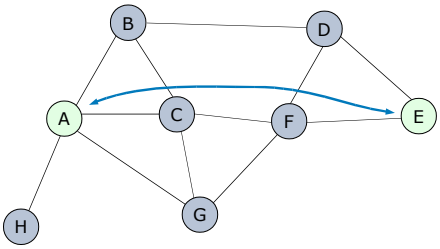
- Basic route discovery
  - Explicit or implicit, providing only the structure of the route
  - Evaluated only on the number of hop to traverse
- Augmented route discovery
  - Need a function that assigns labels to links, denoted as link metrics
  - For a link (V_1,V_2), metric $m_{1,2}$
  - Route metric: a function that is the aggregate of the route link metrics
    - For a route (V_0, V_1, …, V_n), route metric g($m_{0,1}$, $m_{1,2}$,…, $m_{n-1,n}$)

Routing: Autonomous Systems (AS)



- AS: set of interconnected networks under common administration
- Intra-Domain Routing:
  - Send packets within the AS
  - Examples
    - Routing Information Protocol (RIP)
    - Open Shortest Path First (OSPF)
- Inter-Domain Routing:
  - Send packets across multiple ASes
  - Examples
    - Border Gateway Protocol (BGP)

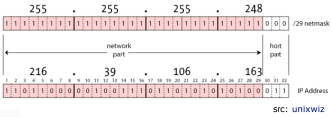
Augmented Routing



- The shortest, in number of hops, is not always the best path

Routing: Subnets

- Network mask
  - 32-bit mask
  - Separate host and network



src: unixwiz

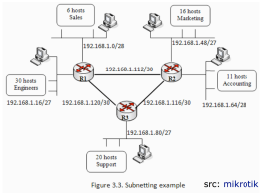
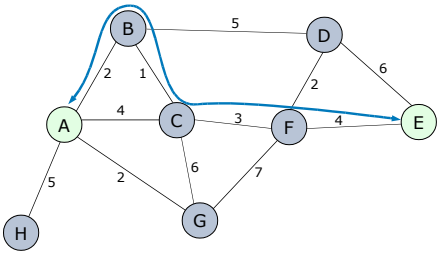


Figure 3.3. Subnetting example src: mikrotik

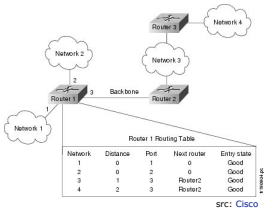
- Subnets
  - Internal hosts have common most-significant bit-group address
  - Hierarchical
  - Can be aggregated
  - Traffic is routed by *routers*

Augmented Routing (cont'd)



- Edge can have metrics (cost, delay, etc.)
- Select route(s) based on those

Routing: Table



src: Cisco

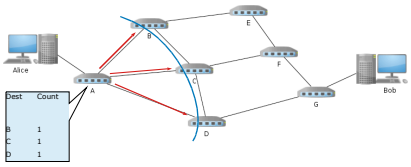
- Information of
  - Destination Network ID
    - Network Destination
    - Netmask
  - Link identification
    - Physical port
    - Next hop / gateway
  - Metric associated to the link
    - Cost
    - Number of hops

Routing Information Protocol (RIP)

- Intra-domain routing
- Implicit
  - No need of full network knowledge for building the table
- Basic
  - Use the hop count as metric (kind-off basic, actually)
  - Distance vector algorithm
- Version 2
  - The node sends its routing table to a multicast address

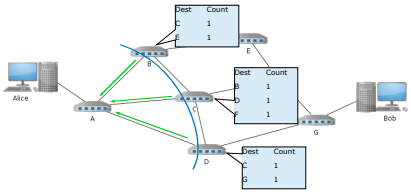
RFC: http://tools.ietf.org/html/rfc2453

Distance Vector Routing



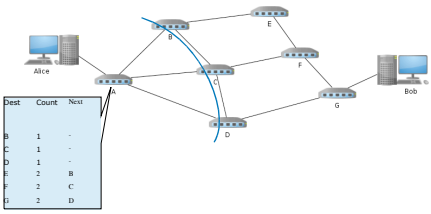
- Routers do not have full knowledge of the path
- Routers exchange *distance vectors*
  - At fixed time / when needed
- View is limited to the next hop (for each destination)

Distance Vector Routing (cont'd)



- Routers do not have full knowledge of the path
  - View is limited to the next hop (for each destination)

Distance Vector Routing (cont'd)



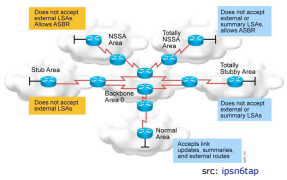
Distance Vector Routing (cont'd)

Bellman-Ford Algorithm

- Initialization
  - Dist(d) = 0
  - Dist(i) = ∞ for all $i \neq d$
- Send the new distance vector to the neighbors
- Receive updates from the neighbors
- Update information for all received vectors $Recv_j$
  - Dist(i) = Min(1 + $Recv_j(i)$, Dist(i))
  - Next(i) = j if the distance is updated

Open Shortest Path First (OSPF)

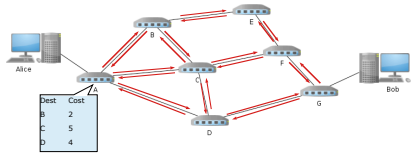
- Intra-domain routing
- Explicit
  - Every node constructs a view of the network topology
  - Paths are evaluated according to the map
- Augmented
  - Link costs
  - Choice of shortest routes
- Network can be divided in multiple areas



src: ipsn6tap

RFC: http://tools.ietf.org/html/rfc2328

Link State Routing



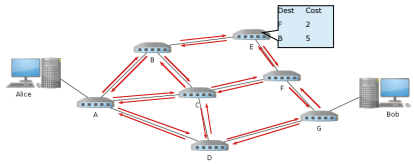
- Link state information is broadcasted to the entire network (flooding)
  - At fixed time / upon detected change
- Each router calculates independently the shortest path to each of the routers

Link State Routing (cont'd)

Dijkstra's algorithm

- Build the network topology with the link state updates received
- Initialization
  - Dist(s) = 0
  - Dist(i) = ∞
  - V ← Ø
- While there are nodes left
  - Find the min(w_{ij}) with $i \in V$ and $j \notin V$
  - V ← V ∪ {j}
  - Dist(j) = Dist(i) + w_{ij}
  - Next(j) = i

Link State Routing (cont'd)



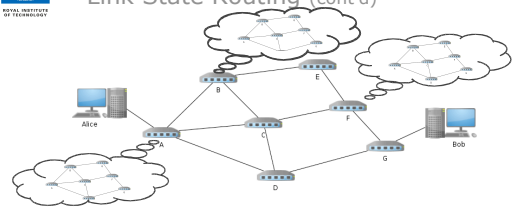
- Link state information is broadcasted to the entire network (flooding)
  - At fixed time / upon detected change
- Each router calculates independently the shortest path to each of the routers

Border Gateway Protocol (BGP)

- Inter-domain routing
- Explicit
  - The routing paths are explicitly provided
- Augmented (kind-off)
  - Routing based on policies, derived from economical agreements between ISP
- Traffic flow vs. Routing information flow
  - Filtering outgoing routing information inhibits traffic flow inbound
  - Filtering inbound routing information inhibits traffic flow outbound

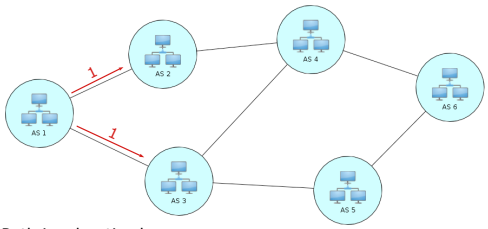
RFC: http://tools.ietf.org/html/rfc4271

Link State Routing (cont'd)



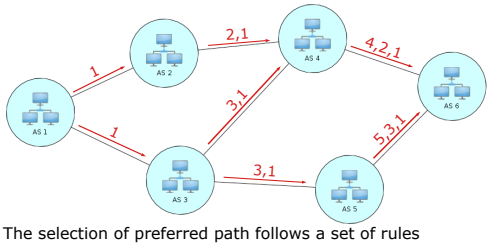
- Link state information is broadcasted to the entire network (flooding)
  - At fixed time / upon detected change
- Each router calculates independently the shortest path to each of the routers

Path Vector Routing



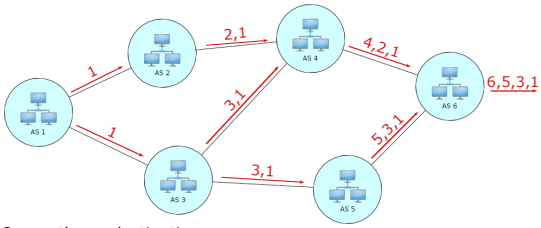
- Path is advertised
  - Each AS advertises the path it prefers to get to AS 1

Path Vector Routing



- The selection of preferred path follows a set of rules
  - E.g., the cost for AS 4 to traverse AS 3 is high

Path Vector Routing



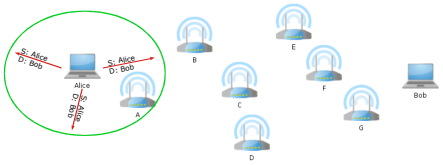
- One path per destination
  - Multiple destinations can be grouped together (address aggregation)

Dynamic Source Routing (DSR)

- Routing in MANET (Mobile Ad hoc Network)
- Explicit
  - The whole path is known by the source and the destination
- Basic
  - Get to know the path structure (of course, you count the links too)
- On demand routing
  - Path is established and maintained only when needed

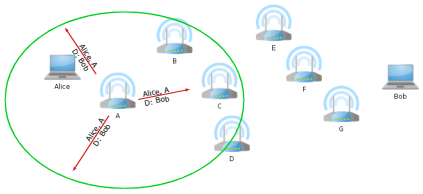
RFC: http://tools.ietf.org/html/rfc4728

DSR (cont'd)



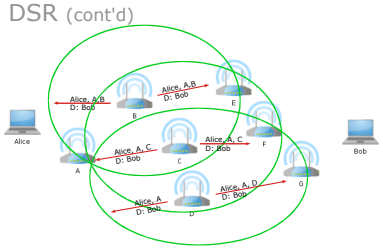
- Route Discovery (RREQ)
  - The source (*initiator*) broadcasts a a route request to its neighbors
  - The initiator advertises the destination (*target*)

DSR (cont'd)



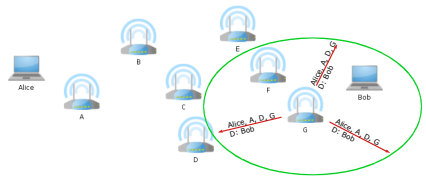
- The node
  - Appends its own address in the route request
  - Propagates it as a local broadcast

DSR (cont'd)



- One node could receive multiple and different copies of the request
  - A request id is included by the source
  - The route request is only propagated once per id
  - Nodes could cache other requests for route recovery

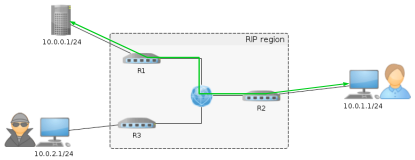
DSR (cont'd)



- The target will answer with a route reply message to the initiator
  - The route response (RREP) will include the discovered path

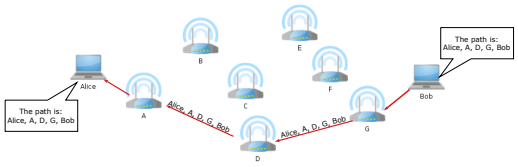
RIP Attacks

- Typically, the received information is unchecked
  - Possible to forge packets and impersonate any host
  - Weak or non-existent authentication



- Introducing new routes or modifying routes

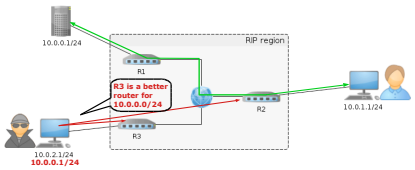
DSR (cont'd)



- The target will answer with a route reply message to the initiator
  - The route response (RREP) will include the discovered path
- Once RREP reaches the originator, the path is established
  - On route error, RERR packets are generated by the last working hop

RIP Attacks (cont'd)

- Typically, the received information is unchecked
  - Possible to forge packets and impersonate any host
  - Weak or non-existent authentication



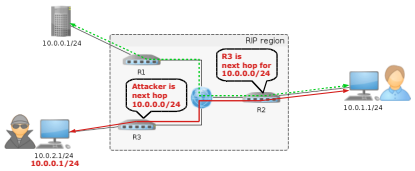
- Introducing new routes or modifying routes

Attacking Routing

- Attacks
  - Spoofing
  - Falsification
  - Prefix Hijacking
  - Usurpation
  - Interference
  - Overload
  - Man-In-The-Middle
  - Black hole

RIP Attacks (cont'd)

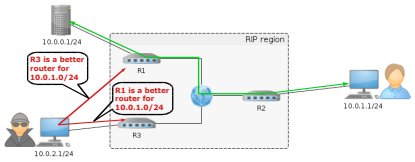
- Typically, the received information is unchecked
  - Possible to forge packets and impersonate any host
  - Weak or non-existent authentication



- Introducing new routes or modifying routes

RIP Attacks (cont'd)

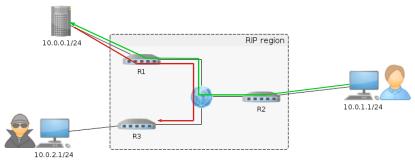
- Typically, the received information is unchecked
  - Possible to forge packets and impersonate any host
  - Weak or non-existent authentication



- Creating routing loops

RIP Attacks (cont'd)

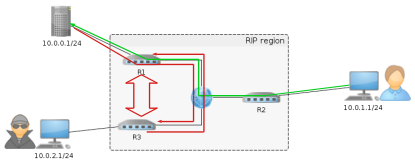
- Typically, the received information is unchecked
  - Possible to forge packets and impersonate any host
  - Weak or non-existent authentication



- Creating routing loops

RIP Attacks

- Typically, the received information is unchecked
  - Possible to forge packets and impersonate any host
  - Weak or non-existent authentication



- Creating routing loops

RIP Defenses

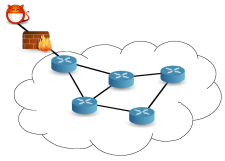
- Authenticate RIP packets
  - The receiver can only authenticate the immediate sender
  - The bogus information could have been injected at the sender side
- Be more skeptical about routes
  - Filter updates (limit IP spoofing)
  - Deny updates for its local networks

OSPF Attacks

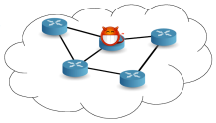
- Link State Advertisement (LSA)
  - Check validity of:
    - Sequence number
    - Check-sum
    - Age
  - Easy to falsify
    - It will trigger the "fight-back" mechanism
    - Attack not always persistent
    - Methods to evade the mechanism
  - Induce topology changes
    - Same problem as RIP

OSPF Defenses

- Attacker is outside the AS boundary
  - Never run OSPF beyond boundaries
  - Deny OSPF packets at the border routers



- Attacker is inside the AS boundary
  - MD5-based authentication with shared keys
    - Maintenance is complex
    - Not much of a defense



OSPF Defenses (cont'd)

- Asymmetric cryptography: sign LSAs
  - Easy to deploy
  - Large number of LSAs
    - Computational overhead
- Hash chain: authenticate LSAs
  - Router uses a precomputed hash chain
  - Distributes the last $H^n(r)$ value with signature
  - The *i-th* update is authenticated if $H^i(u_i) == H^n(r)$

BGP Attacks

- Pakistan Telecom blocks YouTube
- ICANN puts root server at risk
- Malaysian ISP blocks Yahoo
- Northrop Grumman hit by spammers
- Turkish ISP takes over the Internet
- Brazilian carrier leaks BGP table

BGP Attacks (cont'd)

- How these attacks happened
  - Compromised routers
  - Unscrupulous ISPs
  - Configuration error
- Problems
  - Bogus origin of routes
  - Bogus modification of routes

The Internet 'Biggest Security Hole'

src: wired

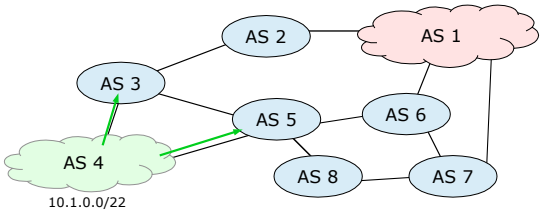
- Anyone with a BGP router could intercept data headed to a target IP address
  - It doesn't exploit any flaw or bug of the BGP
  - It just exploits the natural way BGP works
- Data can be still forwarded to the destination
  - Unnoticed

Security Goals for BGP

- Secure message exchange between neighbors
  - Confidential BGP message exchange
  - Denial of service attack protection
- Validity of the routing information
  - Origin authentication
    - Is the prefix owned bu the AS announcing it?
  - AS path authentication
    - Is AS path the sequence of ASes the BGP update traversed?
  - AS path policy
    - Does the AS path adhere to the routing policies of each AS?
- Correspondence to the data path
  - Does the traffic follow the advertised AS path?

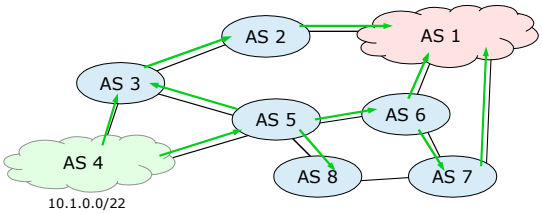
BGP Attacks

- AS 4 (the target) originates 10.1.0.0/22 and sends announcements to its neighbors



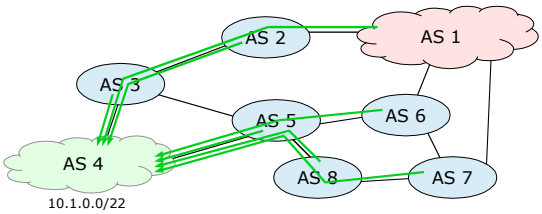
BGP Attacks (cont'd)

- The legitimate announcement propagates



BGP Attacks (cont'd)

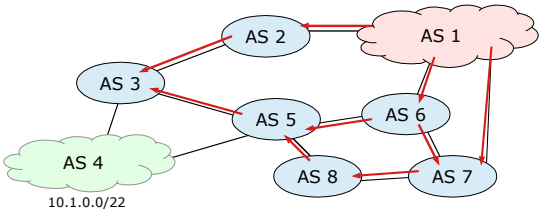
- View of Forwarding Information Base (FIB) for 10.1.0.0/22 after converging



BGP Attacks (cont'd)

Prefix Hijacking

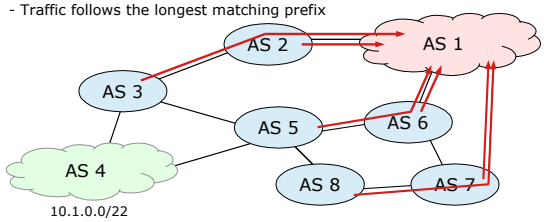
- AS 1 propagates a more specific route for AS 4, announcing 10.1.0.0/24



BGP Attacks (cont'd)

Prefix Hijacking

- Originating the same or a more specific prefix
  - Every AS picks the bogus route for that prefix
  - Traffic follows the longest matching prefix



Hijacking is Hard to Debug

- Real origin AS doesn't see the problem
  - Picks its own route
  - Might not even learn the bogus route
- May not cause loss of connectivity
  - E.g., if the bogus AS snoops and redirects
  - … may only cause performance degradation
- Or, loss of connectivity is isolated
  - E.g., only for sources in parts of the Internet
- Diagnosing prefix hijacking
  - Analyzing updates from many vantage points
  - Launching traceroute from many vantage points

How to Hijack a Prefix

- The hijacking AS has
  - Router with eBGP session(s)
  - Configured to originate the prefix
- Getting access to the router
  - Network operator makes a configuration mistake
  - Disgruntled operator launches an attack
  - Outsider breaks in to the router and reconfigures
- Getting other ASes to believe bogus route
  - Neighbor ASes not filtering the routes
  - … e.g., by allowing only expected prefixes
  - But, specifying filters on peering links is hard

The February 24 Youtube Outage

- YouTube (AS 36561)
  - Web site www.youtube.com
  - Address block 208.65.152.0/22
- Pakistan Telecom (AS 17557)
  - Receives government order to block access to YouTube
  - Starts announcing 208.65.153.0/24 to PCCW (AS 3491)
  - All packets directed to YouTube get dropped
- Mistakes were made
  - AS 17557: announcing to everyone, not just customers
  - AS 3491: not filtering routes announced by AS 17557
- Lasted 100 minutes for some, 2 hours for others

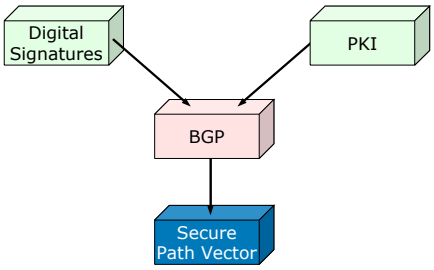
src: renesys

Youtube Outage: UTC Timeline

- **18:47:45** First evidence of hijacked route propagating in Asia, AS path 3491 17557
- **18:48:00** Several big trans-Pacific providers carrying hijacked route (9 ASNs)
- **18:48:30** Several DFZ providers now carrying the bad route (and 47 ASNs)
- **18:49:00** Most of the DFZ now carrying the bad route (and 93 ASNs)
- **18:49:30** All providers who will carry the hijacked route have it (total 97 ASNs)
- **20:07:25** YouTube, AS 36561 advertises the /24 that has been hijacked to its providers
- **20:07:30** Several DFZ providers stop carrying the erroneous route
- **20:08:00** Many downstream providers also drop the bad route
- **20:08:30** And a total of 40 some-odd providers have stopped using the hijacked route
- **20:18:43** And now, two more specific /25 routes are first seen from 36561
- **20:19:37** 25 more providers prefer the /25 routes from 36561
- **20:28:12** Peers of 36561 start seeing the routes that were advertised to transit at 20:07
- **20:50:59** Evidence of attempted prepending, AS path was 3491 17557 17557
- **20:59:39** Hijacked prefix is withdrawn by 3491, who disconnect 17557
- **21:00:00** The world rejoices

src: renesys

Securing BGP



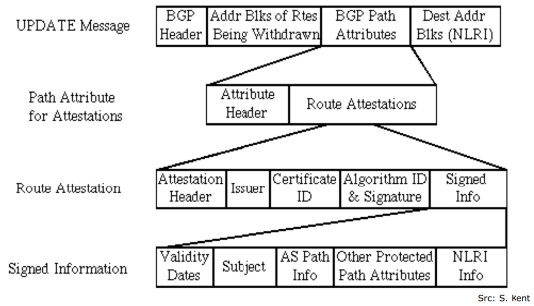
Proposals for Control Plane Security

- **S-BGP**: Secure BGP
  - By BBN
  - Centralized (PKI-based)
  - Signatures on every element of the path
  - (side note: the first comprehensive approach and project)
- **soBGP**: **s**ecure **o**rigin BGP
  - Decentralized
  - Use PKI only for *origin authentication*
  - Topology database for *path authentication*
  - (side note: supported by Cisco)

S-BGP

- Address-based PKI: validate signatures
  - Authentication of:
    - Ownership for IP address block(s)
    - AS number
  - Use existing infrastructure (Internet registries etc.)
  - Routing origination is digitally signed
  - BGP updates are digitally signed
- Address attestation: Validates that the originating AS is authorized to advertize the destination block of addresses
- Route attestation: Validates that the next AS along the update path can use the AS Path, that is, advertize prefixes

Attestations and Update Format



Src: S. Kent

Attestations

- Address attestation(s)
  - May be more than one, depending on how many owners (sub-blocks) exist for the advertized destination (block, or Network Layer Reachability Information (NLRI))
- Route attestations:
  - One per AS that relays the update
  - Identifies the next AS that can use Update
  - Also includes:
    - The id of the AS (its BGP speaker)
    - The destination NLRI
    - An expiration time
    - The signature of the BGP speaker owner

Attestations (cont'd)

- Address attestation
  - To validate, any receiving AS needs:
    - The address allocation certificate of the owner of the advertised address block
- Route attestation:
  - To validate, any receiving AS needs:
    - The certificate of the signing AS
- Route update:
  - To validate, any receiving AS needs:
    - One validated address attestation by each owner of a part of the advertised block
    - One validated route attestation for each AS in the received
- In all cases above, each certificate must be valid (i.e., not revoked)

Reducing Message Overhead

- Problem: How to distribute certificates, certificate revocation lists (CRLs), and even address attestations (AAs)?
  - Redundant data across updates
  - Address attestations do not change frequently
  - UPDATE messages usually 4 kBytes
- Solution: Use servers for these data items and access them out-of-band
  - Replicate for redundancy & scalability
    - At Network Exchange/Access Points; Regional Internet Registry (RIRs); ISP processing centers
  - Download certificates, CRLs, and Aas; possibly pre-processed (by ISPs)
  - ISPs operate repositories and upload updates

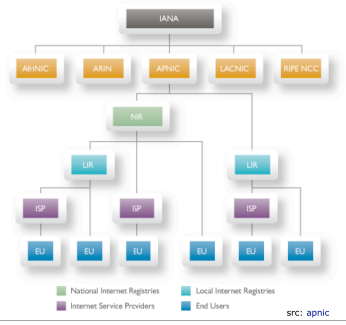
Practical issues with S-BGP

- Requires Public-Key Infrastructures
- Lots of digital signatures per BGP UPDATE messages
  - Transmission overhead
  - Computation overhead
- Secure route withdrawals: what if link or node fails?
- Address ownership data out of date
- Deployment

Public Key Infrastructure (PKI)

- Problem: Key distribution
  - How do you find out an entity's public key?
  - How do you know it isn't another entity's key?
- Root of PKI: Certificate Authority (CA)
  - Bob takes public key and identifies himself to CA
  - CA signs Bob's public key with digital signature to create a certificate
  - Alice can get Bob's key (doesn't matter how) and verify the certificate with the CA
- PKIs are typically organized into hierarchies

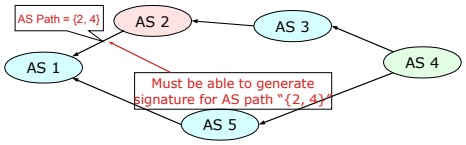
Address Block PKI



src: apnic

Path Shortening Attack

- Adversary AS shortens AS path to divert the traffic
- Not possible with S-BGP



Residual vulnerabilities of S-BGP

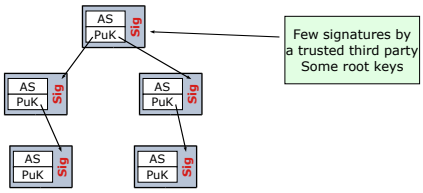
- Message suppression
  - Detecting on a link level, but not if a BGP router chooses not to relay a message; e.g., skip a route withdrawal update
- Connectivity information disclosure
  - Open to eavesdroppers, no confidentiality
- Replay attacks, reassertion of routes
  - Premature re-advertisement of withdrawn routes; yet, within the lifetime a route attestation
  - BGP UPDATE messages do not have sequence numbers
- Incorrect following/application of policies and rules
  - Routers are not strictly bound by the protocol

Secure Origin BGP (soBGP)

- No central authority
  - Web of trust
- Incremental deployment
  - Different options, flexibility
  - No reliance on connection to external data-bases, distribute certificates as part of the routing protocol messaging
- Approach (address / do not address)
  - Is the originating router/AS authorized to advertize a prefix?
  - Does the advertising router/AS have a path to the prefix (destination)?
  - Is the advertising router authorized by the originator to do so?
  - Does an advertised path abide with local policies?

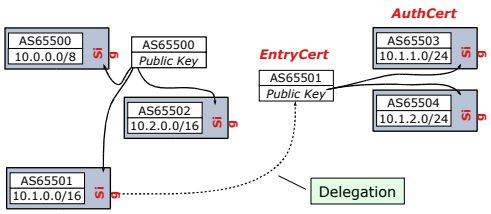
Step 1: AS Identity (EntityCert)

- Each AS creates a public/private key pair
- Basically, a web of trust approach
- Few EntityCerts are signed by a trusted third party



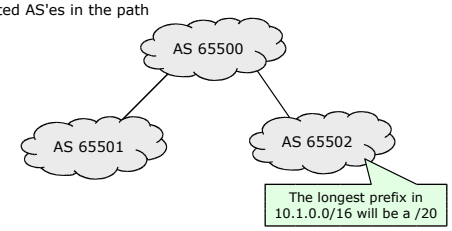
Step 2: Origin ad authorization (AuthCert)

- Signed certificate authorize another AS to advertise a prefix



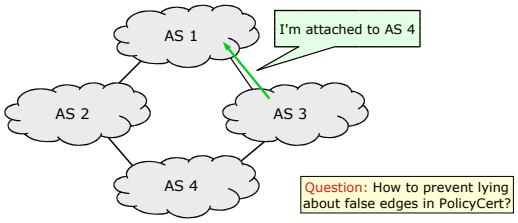
Step 3: Enforcing Policy (PolicyCerts)

- Each AS builds a certificate with contains policy information
  - Maximum prefix length – PrefixCerts, which wrap AuthCerts
  - Unwanted AS'es in the path



Step 4: Validating paths to prefixes (ASPolicyCert)

- Signed ASPolicyCerts
  - Contain a signed list of peers
  - Flooded throughout the network: creating a Topology map



Issues with soBGP

- Integrity: Cannot validate that the update actually traversed the path
- Collusion: Colluding ASes can create false edges
- Replays: ASPolicyCerts do not prevent advertising a path recently withdrawn

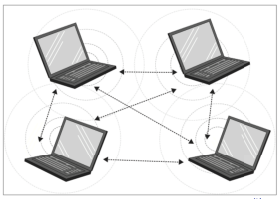
S-BGP vs. soBGP

	soBGP	S-BGP
Does the AS Path exist?	Maybe: PolicyCerts	Yes
Did the received update travel along that path?	No	Yes: Route Attestation + Validity
Was the update authorized to traverse that path by the originator?	Maybe: Depends on how PolicyCerts are written	No

src:N. Feamster

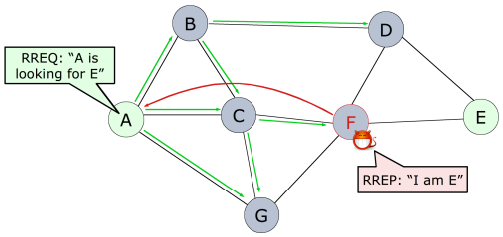
Ad Hoc Networks

- Decentralized
- Multi-hop
- Each node helps to forward data
- Re-think routing protocol
- Mobile Ad-hoc Network (MANET)
  - Frequent route changes due to mol
- Wireless Sensor Network (WSN)
  - Static (or nearly)
  - Energy and hardware constraints



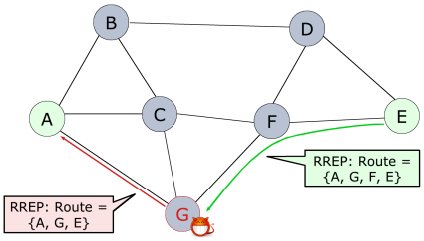
src: itlaw

Attacks: DSR



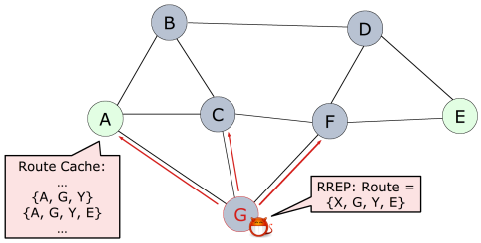
- Impersonation of the destination

Attacks: DSR (cont'd)



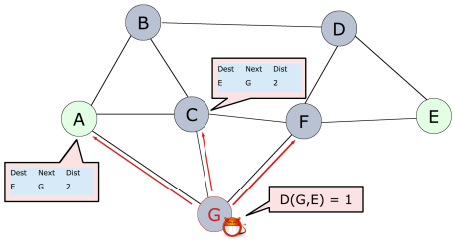
- Modification of the route links

Attacks: DSR (cont'd)



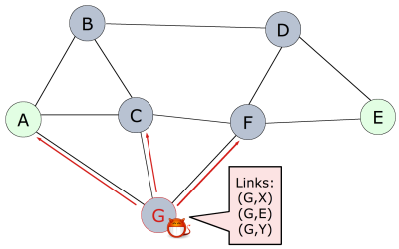
- Abuse of the routing caching mechanism

Attacks: Distance Vector (cont'd)



- Disrupting distance vector routing protocol

Attacks: Link State

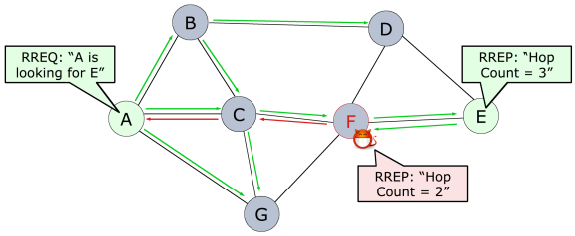


- Disrupting a link state routing protocol

Requirements

- What do we need a secure routing protocol to do?
- Network model
  - Capture the system characteristics
  - For example, dynamically changing topology
- Specification
  - Define the properties of any candidate secure routing protocol **independently** of its functionality

Attacks: Distance Vector



- Disrupting distance vector routing

Requirements (cont'd)

- We are interested in protocols that discover routes with the following two properties:

(1) Loop-freedom
- an (S,T)-route is loop-free when it has no repetitions of nodes

(2) Freshness
- an (S,T)-route is fresh with respect to a (t_1,t_2) interval if each of the route's constituent links is up at some point during the (t_1,t_2)

Loop-freedom and freshness are relevant for both explicit and implicit route discovery, and both basic and augmented protocols

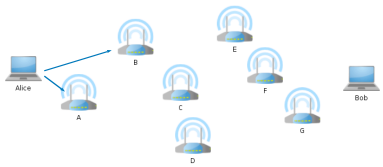
P. Papadimitratos, Z.J. Haas, and J.-P. Hubaux, *"How to Specify and How to Prove Correctness of Secure Routing Protocols for MANET"*, BroadNets'06

Secure Routing Protocol (SRP)

- Explicit basic route discovery
- Observation
  - It is hard to 'know' all nodes in the network, i.e., establish associations with all of them
  - Often infeasible and very costly
  - Especially in 'open' networks
- SRP assumptions
  - Secure neighbor discovery
  - Hop-by-hop authentication of all control traffic
  - End nodes (source, destination) 'know' each other
    - Can set up security associations
    - Shared key $K_{A,B}$

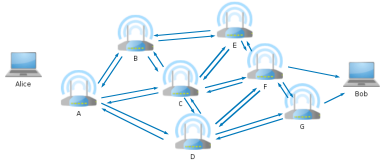
P. Papadimitratos and Z.J. Haas, "*Secure Routing for Mobile Ad Hoc Networks*", CNDS 2002

Secure Routing Protocol (SRP) (cont'd)



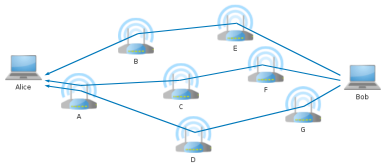
- Route request RREQ
  - Alice, Bob, Q_{SEQ}, Q_{ID}, MIC($K_{Alice,Bob}$, Alice, Bob, Q_{SEQ}, Q_{ID})
  - The addresses of the traversed intermediate nodes are accumulated in the RREQ

Secure Routing Protocol (SRP) (cont'd)



- The intermediate nodes
  - Accumulate the traversed addresses in the RREQ
  - Relay RREQ, so one or more could reach the destination
  - Previously seen RREQ are discarded

Secure Routing Protocol (SRP) (cont'd)

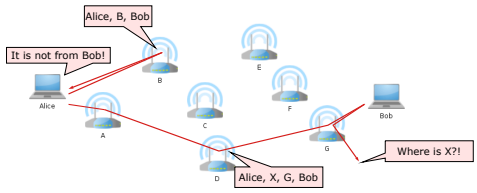


- Route reply RREP (if the MIC is valid)
  - Over the reverse of the route accumulated
  - Q_{ID}, {Bob, F, C, A, Alice}, MIC($K_{Alice,Bob}$, {Bob, F, C, A, Alice}, Q_{SEQ}, Q_{ID})
  - Respond to one or more request packet of the same Q_{SEQ}

Secure Routing Protocol (SRP) (cont'd)

- Route requests verifiably reach destination
  - Intermediate node replies disabled
  - Aggressive caching of routing information disabled
- Route replies must trace back the paths traversed by route requests
- Intermediate nodes are not authenticated at the end nodes
- Dual route request identifier
  - Q_{ID}: random, used by the intermediate nodes
  - Q_{SEQ}: sequence number, used by the destination
  - The adversary cannot launch a "sequence number" attack

Secure Routing Protocol (SRP) (cont'd)



- Intermediate nodes
  - Cannot tamper the path in RREP → MIC will fail
  - Cannot tamper the path in RREQ → No route to host
  - If they drops the RREQ, there will be another node that forwards it
  - If they reply the packets, it will be dropped due to the same Q_{SEQ}

Secure Routing Protocol (SRP) (cont'd)

- Crucial to operate on top a secure neighbor discovery protocol
- Neighbor Look-up Protocol (NLP)
  - Secure neighbor discovery
  - Establish security associations between neighbors
  - Identify control traffic injected by each neighbor
  - Prevent attacks that misuse network addresses
    - IP spoofing
    - Use of multiple identities
    - MAC spoofing
    - DoS protection
- Efficient mechanisms to discard spurious/ corrupted traffic at intermediate nodes
  - Replies relayed only if neighbors had previously forwarded the corresponding request

Secure Routing Protocol (SRP) (cont'd)

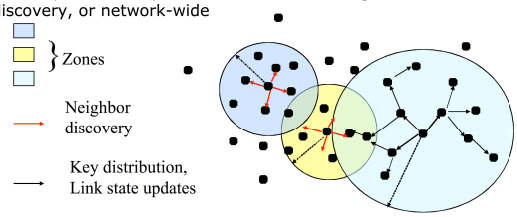
- Routes discovered by SRP in the presence of independent adversaries are fresh
  - t_1 is the point in time at which Alice transmitted a RREQ for Bob, and t_2 is the point at which Alice received the corresponding RREP
- In the presence of colluding adversaries SRP discovers 'weakly fresh' routes
  - A sequence of links, in general different than those in the discovered route were up at some point in (t_1,t_2)

Secure Link State Protocol (SLSP)

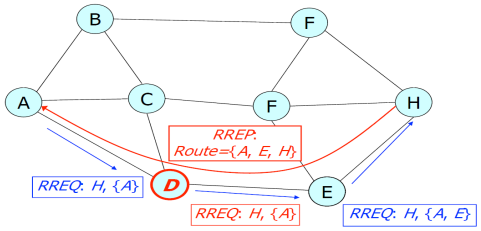
- Secure Neighbor Discovery
  - Correct nodes discover only actual neighbors
- Periodic Link State Update (LSU) advertisements
  - Nodes distribute their discovered neighbors within an extended neighborhood, the zone
- LSUs are signed
- Link state accepted iff reported by both incident nodes
- Nodes distribute their public key throughout the zone
- SLSP can adjust its scope with different zone radii

SLSP (cont'd)

- SLSP can adjust its scope, with different zone radii
- It can operate locally, combined with another global route discovery, or network-wide



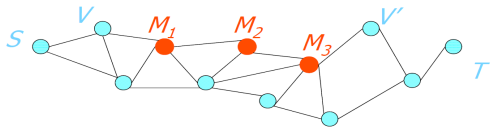
Attacking route discovery (cont'd)



- Adversary acting as a relay, creating Byzantine links
- Secure neighborhood discovery and hop-by-hop authentication defeat this attack

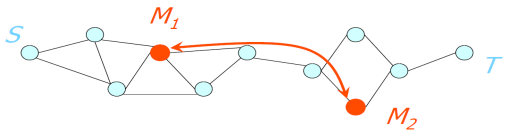
Attacking Routing - Revisited

- Multiple Colluding Attackers
  - M1 and M3 are seemingly correct to their neighbors, but they omit protocol functionality when handling packets from M2
  - Example: M2 relays RREQ and RREP packets without appearing in the route discovery



Attacking Routing – Revisited (cont'd)

- Tunneling Attack
  - Two colluding attackers, M1 and M2
  - M1 encapsulates control traffic and forwards to M2 and vice versa
  - Attackers seemingly follow the protocol with respect to their neighbors



Summary

- Route discovery is vulnerable
- Secure route discovery specification
  - Loop freedom
  - Freshness
  - Accuracy
- Protocols relying on different trust assumptions
- Securing basic and augmented route discovery
- Colluding adversarial nodes can subvert any route discovery protocol, see the tunneling attack