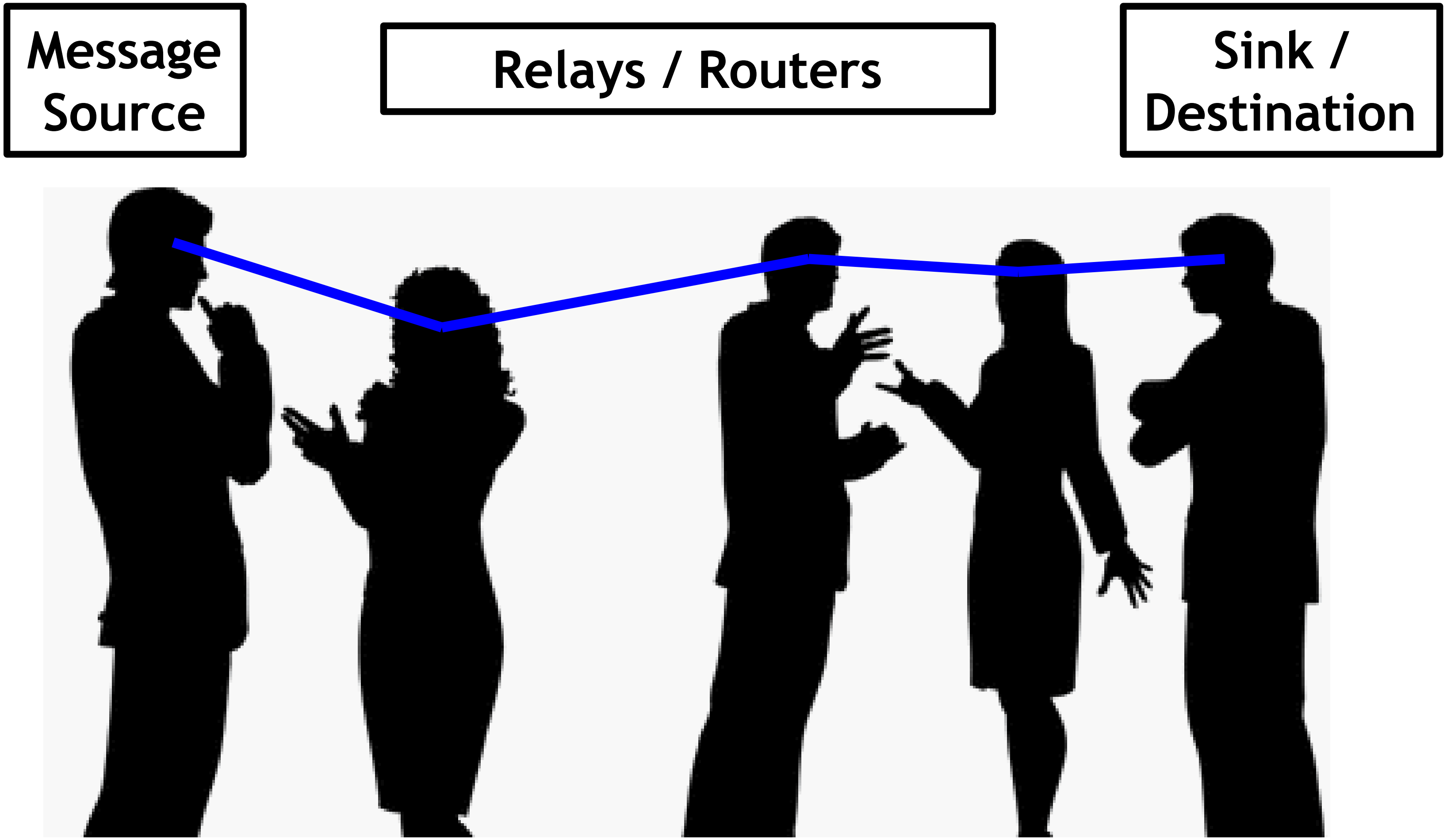
Wireless Network Security and Privacy

Autumn 2023

Xiaoyu Ji

Network Layer Threats

Wireless Networking



Network Layer Functionality

- The network layer is primarily responsible for establishing end-to-end paths and delivering packets over them

- Includes several fundamental services:
  - Addressing: network ID management
  - Routing: finding/establishing paths
  - Forwarding: delivering packets

  - Interactions with Transport layer and Link/MAC layer

Addressing

- Before routing can be performed, nodes need some sort of ID or address
  - **ISP**: Address/ID types range from local to global, just like in the postal system (streets up to ZIP codes)
  - **Hierarchical**: In very large-scale systems (e.g. Internet), addresses must have some sort of structure
    - IP addresses follow a specific hierarchy and are reused within each domain
  - Within a domain and in small-scale systems (e.g. MANET/WSN), addresses are typically unstructured or random
    - Address management needed within a domain to prevent duplication and other failure scenarios

Addressing Threats

- Addresses can be changed arbitrarily
  - Allows for address spoofing
    - Masquerading as other node(s)
    - Potential for a large number of attacks
  - Changing identity to prevent detection/punishment
- Attackers can infiltrate address management protocols (ARP, DHCP) to cause problems
  - Inducing address duplication
  - Forcing frequent address changes
  - Manipulating forwarding schemes

Routing

- **Routing = path management**
  - Routing does not involve actual sending of packets from source to destination(s), only sets up the path
  - Lives in the “control plane”
  - Involves path setup/discovery, maintenance, and tear-down
- **Challenges in MANET/WSN environments**
  - Route using multiple untrusted relay nodes
  - Resource and capability limitations
  - No centralized authority or monitor
  - Secure routing often relies on existing key mgmt.

Routing Threats

- Just as with other types of misbehavior, routers can be greedy, non-cooperative, or malicious
  - Greedy routers can refuse route discovery requests in order to save their own resources
  - Non-cooperative routers can choose to selectively accept route requests to specific sources/dests
  - Malicious routers can persuade route discovery protocols so paths pass through them, avoid them, or take unnecessary detours

Path Attraction

- **Black-hole attack:**
  - A malicious router broadcasts false claims of being “close” to the destination in order to attract all traffic and drop it
- **Gray-hole attack:**
  - Similar to black-hole attack, except it only drops some packets selectively
    - Ex: forward all routing control packets but drop all data
- **Worm-hole attack:**
  - Colluding routers create a low-latency long-distance out-of-band channel to attract routing paths and control data flow

Path Manipulation

- Detours:
  - A malicious router can modify/inject control packets to force selection of sub-optimal routes

- “Gratuitous detours”:
  - Greedy routers can avoid being on a selected route by advertising long delays or creating “virtual nodes”
    - Could be considered a form of Sybil attack, where all “personalities” are on the routing path

Route Subversion

- Targeted blacklisting:
  - In any routing protocols using blacklisting, attackers can accuse/slander/blame others to force them onto the blacklist → DoS
- Rushing attacks:
  - Attackers can quickly disseminate forged requests, causing later valid requests to be dropped

Forwarding

- **Forwarding = point-to-point data management**
  - Forwarding involves actual sending of packets from source to destination(s) on given routing paths
  - Lives in the "data plane"
  - Correct forwarding involves
    - Sending the correct packets
    - Maintaining packet order
    - Respecting headers and rules
    - Relaying in a timely manner
    - Respecting rate control mechanisms

Forwarding Threats

- Misbehavior in the forwarding mechanism (often called Byzantine forwarding) includes various ways of going against forwarding rules
  - Dropping packets
  - Modifying packet contents or header information
  - Injecting bogus packets on source's behalf
  - Forwarding to the wrong next hop
  - Disrespecting rate control (flooding or throttling)

Network Privacy Threats

- Routing protocols inherently reveal information to curious/malicious eavesdroppers
  - An attacker can listen to route discovery interactions and learn (1) locations of source and destination nodes, (2) type of interactions between nodes, (3) commonly used paths, (4) network events, or (5) data

  - These are all issues of location privacy, network privacy, and data privacy due solely to the routing process

Let's go through these different threats in some detail, starting with addressing

Agenda

- Identity threats and countermeasures
- Basics of routing in ad hoc networks
- Control-plane attacks and defenses

Addressing

- In traditional networking, each device (radio) has two identities, in the form of addresses
  - MAC address: hardware address of the radio needed for link layer communication (e.g., 802.3, 802.11)
    - Hard-coded into the NIC
    - In theory, unique and static
  - IP address: network layer address used for routing and some other higher layer services
    - Virtual software address

MAC Addresses

- MAC addresses in the Internet
  - Ethernet and WiFi use MAC addresses for link layer communication
  - Independent of any higher-layer functionality
  - Link layer frames carry source and destination MAC addresses (6B each)

- MAC addresses in other systems
  - Not typically used in sensor networks due to overhead
  - Not needed if other addressing is available

IP Addresses

- IP addresses in the Internet
  - Network layer and above use IP addresses for some identity purposes
  - Independent of whatever is below the network layer
  - IP addresses must be unique
- IP addresses in other systems
  - To support common applications, most designers are aiming to support IP addressing (to some extent)

IP Address Resolution

- In most Internet domains, IP addresses are assigned centrally using DHCP and bound to MAC addresses using ARP
  - DHCP = Dynamic Host Configuration Protocol: host asks server for IP address, which it keeps until expiry
  - ARP = Address Resolution Protocol: host asks other hosts for MAC address corresponding to an IP address

Limitations

- MAC addresses are no longer hardware-bound
  - Most Linux-like systems allow software to change MAC address used, despite hard-coded MAC address
  - Many devices don't have (unique) MAC addresses
- DHCP is impractical for distributed systems
  - Requires centralization
  - High overhead in dynamic systems
- ARP has high overhead in distributed systems
  - Requires request flooding

Distributed Addressing

- **Problem**: How should IP addresses (or other suitable identities) be determined in a distributed system such that:
  - Addresses are compact(-able) for low-overhead communication in sensors or embedded devices
  - Network overhead is (relatively) low
  - Addresses are (sufficiently) unique
  - Systems can split and join
    - Duplicate addresses can be detected and fixed
    - Address space is large enough and dynamic

A Few Approaches

- Random selection with duplicate address detection (DAD)
  - Send a query to the selected address; if no response, the address probably isn't in conflict
  - Requires flooding a query through the entire network
  - Merging existing networks is difficult
- MANETconf
  - Configured "initiator" nodes act like a server that can assign addresses to "requesters" who arrive later
  - Configured node floods notification and assigns address if no nodes respond negatively
  - Merging existing networks is difficult

Security Issues

- Those approaches were not designed with malicious behaviors in mind
- Threats [Wang et al., 2005]:
  - Address spoofing – attacker spoofs the IP address of a victim and hijacks its traffic
  - False address conflict – attacker injects conflict messages (or events) to a target victim, e.g., cconflict notice
  - Address exhaustion – attacker claims many addresses to deny service or prevent nodes from joining
  - Negative reply – in cases where approval is needed to join, attacker can prevent nodes from joining

Secure MANET Auto-Conf

[Wang et al., 2005]

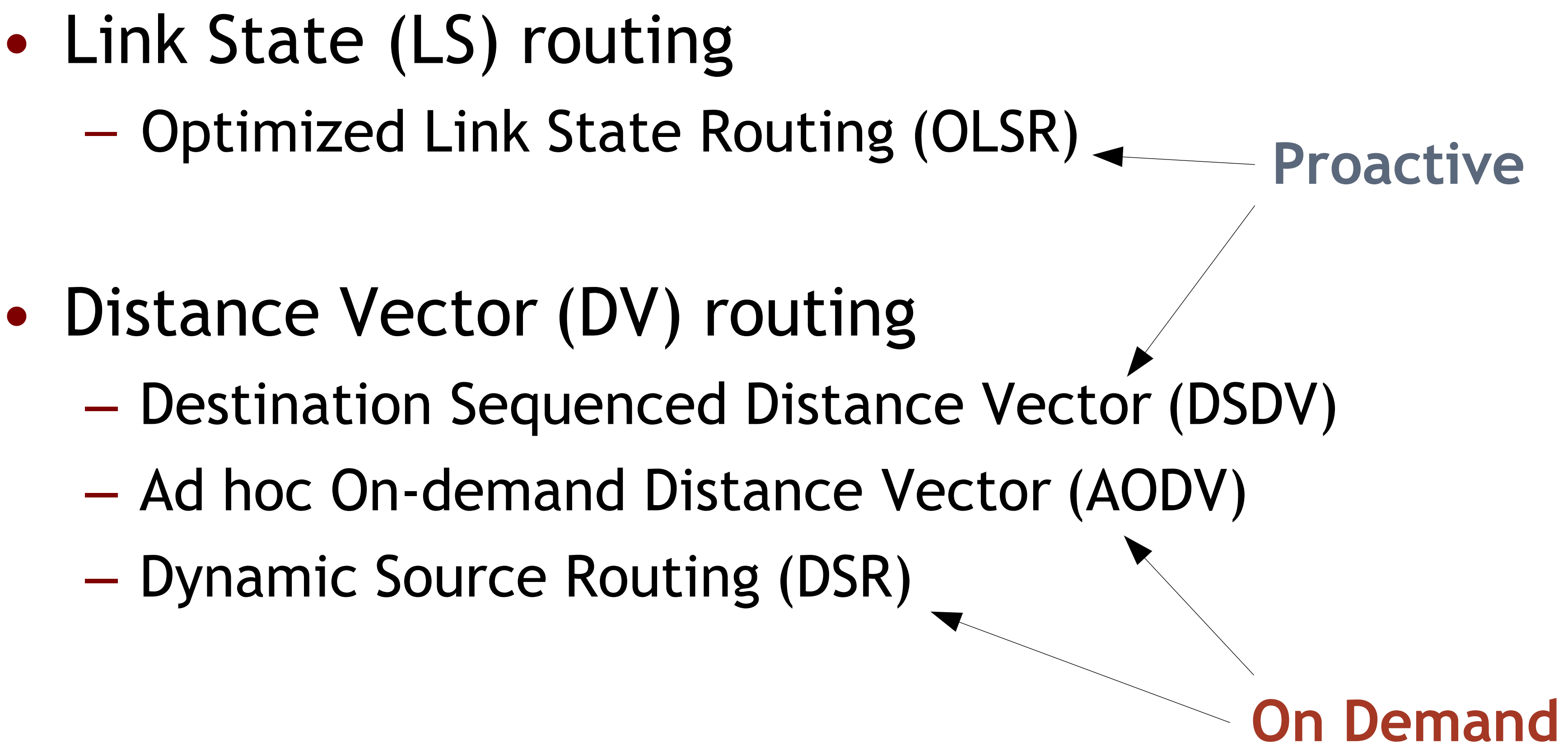
- Bind the IP address to a public key to authenticate auto-configuration processes
  - New node A chooses an IP address as the hash of its public key
  - A sends a query to the network for the IP address using a signed, time-stamped Duplicate Address Probe
    - If a receiving node B has an IP conflict, it checks signatures (authenticity, replay prevention, etc.) and conditionally replies with a signed, time-stamped Address Conflict Notice
    - If A receives ACN from B, it checks signatures and conditionally starts over with a new key pair
    - If no reply within a fixed time period, A joins the network using the generated IP address

Benefits of the Approach

- Forces the attacker to find a public key that hashes to a victim's IP address before launching the attack
  - Even with relatively small address space, computation/storage overhead is prohibitive
  - Detailed analysis in the paper

On to routing security – let's start with some basics of MANET routing

Popular Routing Protocols



- Link State (LS) routing
  - Optimized Link State Routing (OLSR)
- Distance Vector (DV) routing
  - Destination Sequenced Distance Vector (DSDV)
  - Ad hoc On-demand Distance Vector (AODV)
  - Dynamic Source Routing (DSR)

Proactive

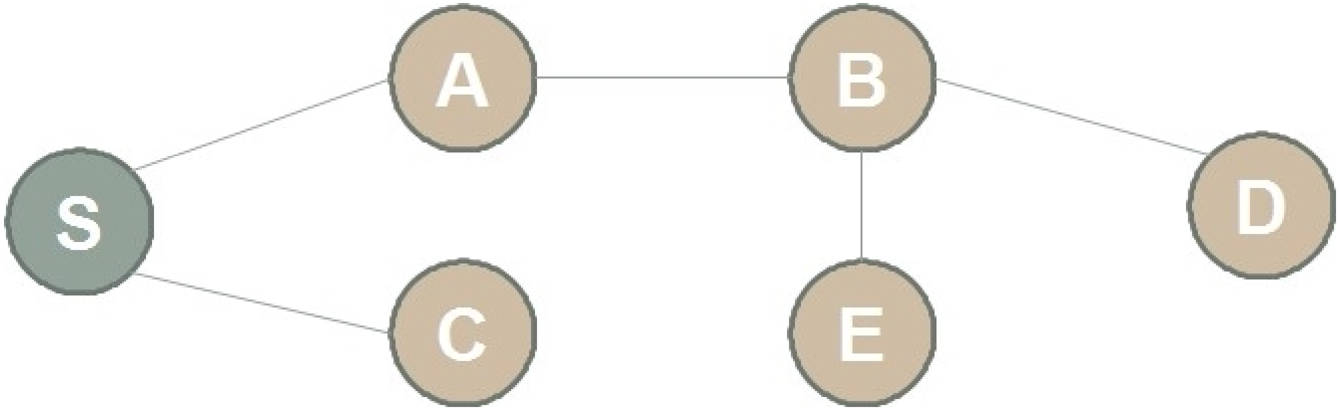
On Demand

On-Demand Routing

- On-demand routing has several advantages and disadvantages in MANETs
  - Efficiency:
    - (+) Routing information isn't constantly collected and updated, only when needed
    - (-) One-time cost of info collection can be higher
  - Security:
    - (+) Source nodes are aware of the entire path, unlike fully distributed algorithms that just focus on next hop
    - (-) Long-term information typically isn't available
  - Overall, advantages outweigh the disadvantages, so on-demand routing (esp. source routing) is popular

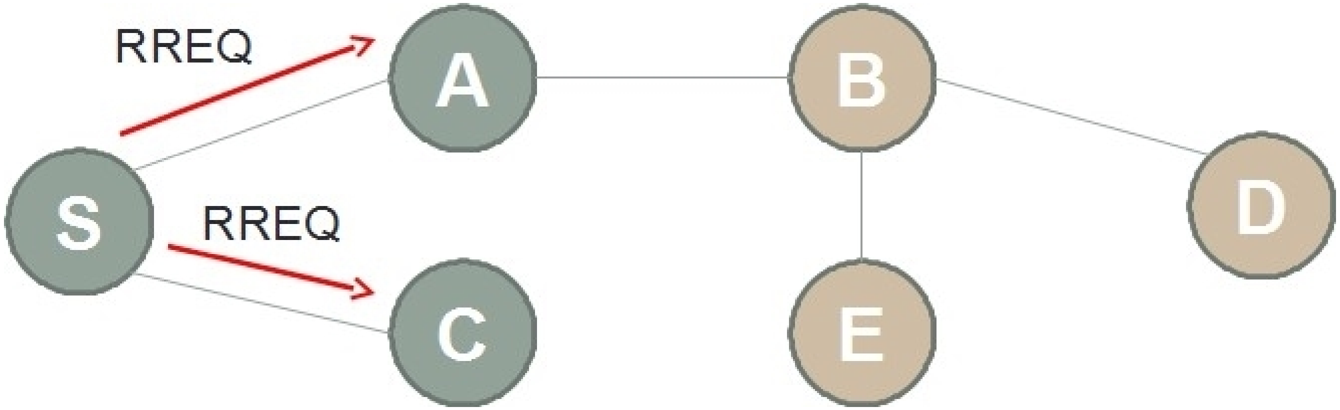
Route Discovery

- Source *S* and neighboring nodes use control message exchanges to discover a route from *S* to destination *D*



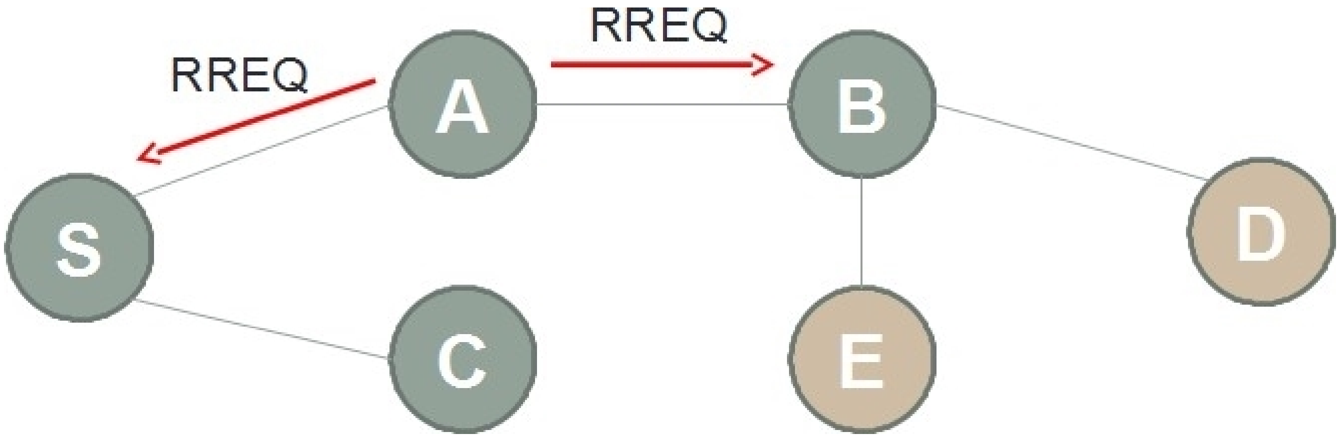
Route Discovery

- Route request flooding:
    - Source *S* broadcasts a Route Request (RREQ) packet to its neighbors



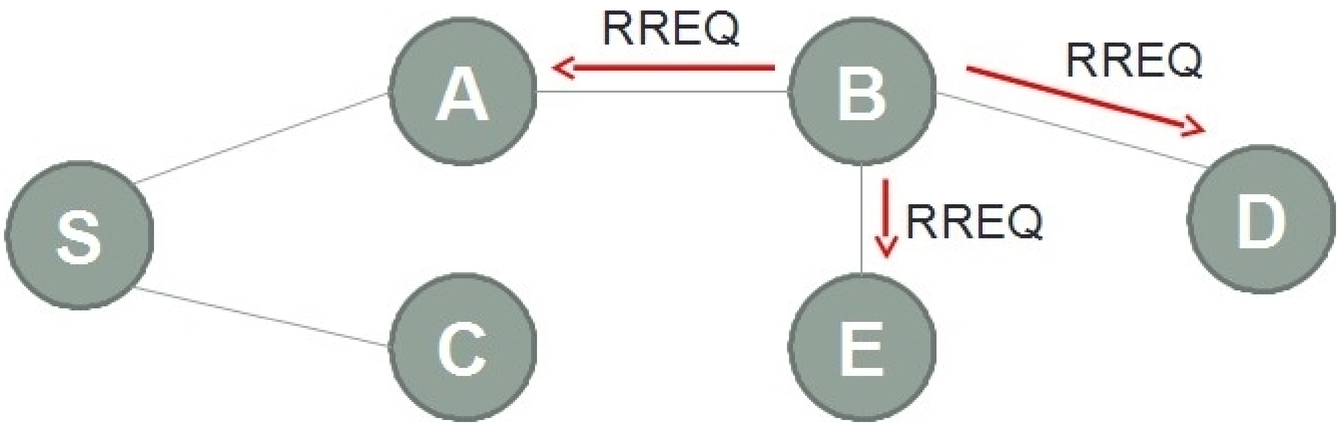
Route Discovery

- RREQ forwarding:
  - If the neighbor has no prior relationship with the destination, it will further broadcast the RREQ



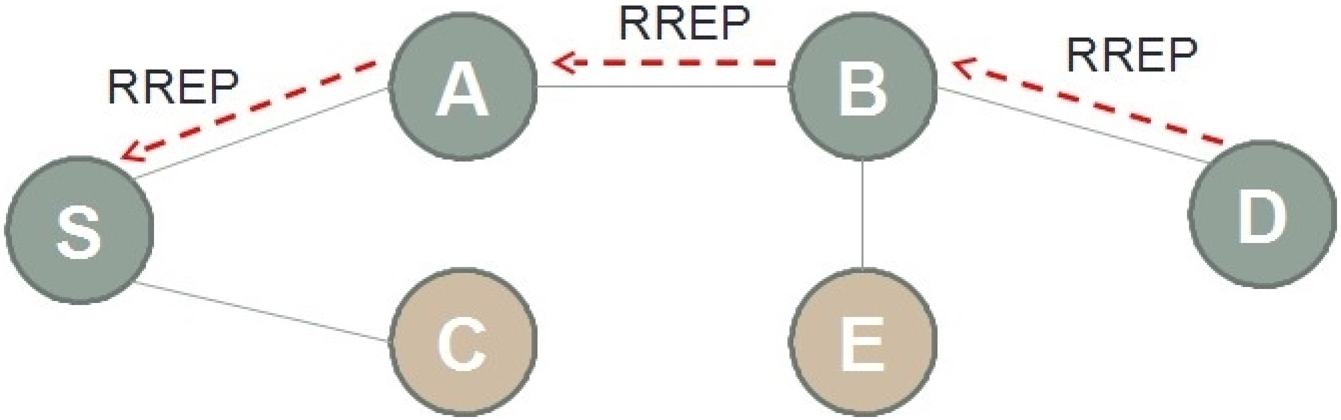
Route Discovery

- Flooding of control packets to discover routes
  - Once the RREQ packet reaches the destination, or a node that knows the destination, the node will unicast a RREP packet to the source via the routed path



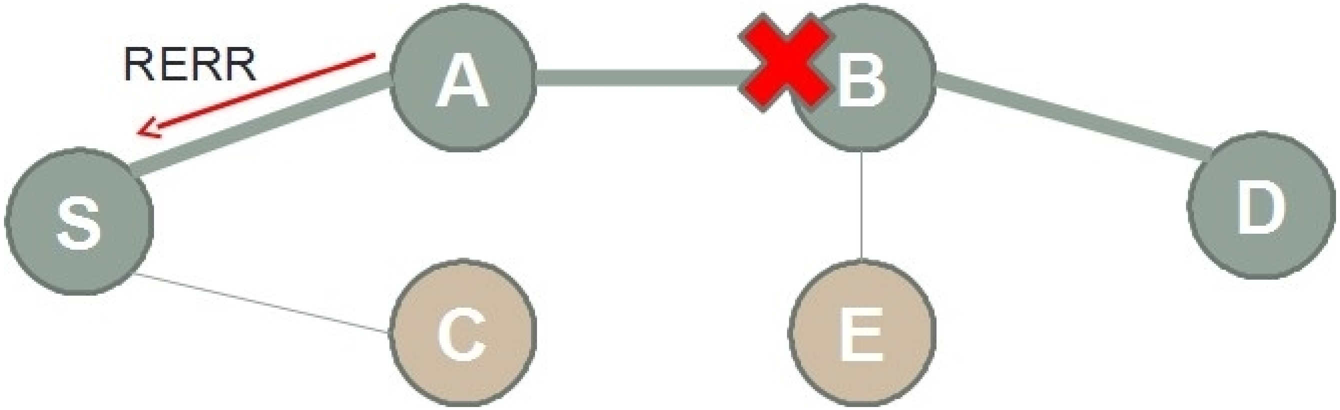
Route Discovery

- Upon receiving the RREQ, *D* (or another node that knows *D*) will unicast a Route Reply (RREP) back to *S* along the found path



Route Maintenance

- If a node can no longer reach the next hop
  - Sends Route Error (RERR) control packet to inform upstream neighbors
  - Route cache alternative (DSR) or rediscovery



AODV vs. DSR

AODV	DSR
Routing tables • one route per destination	Routing caches • multiple routes per destination
Always chooses fresher routes • Sequence numbers	Does not have explicit mechanism to expire stale routes
More frequent discovery flood to ensure freshness	Source Routing • Intermediate nodes learn routes in 1 discovery cycle

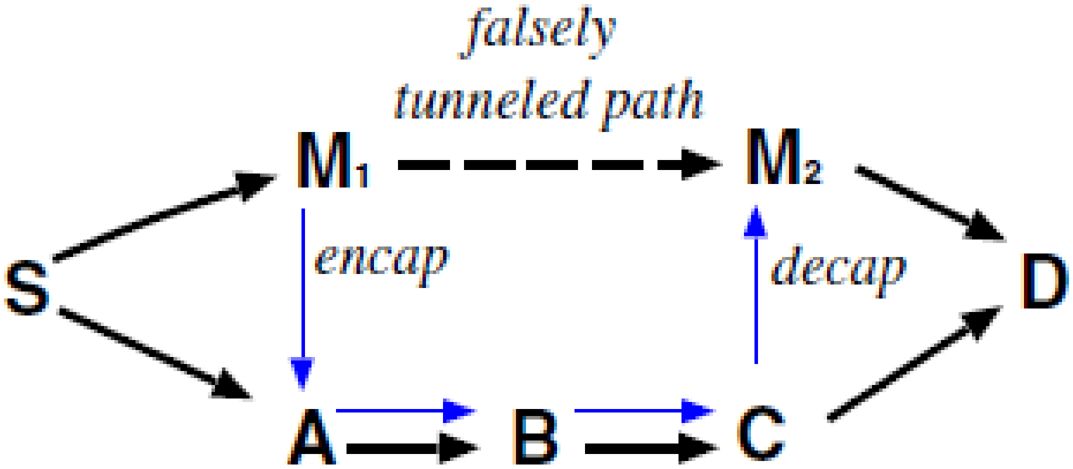
Now, how could an attacker interfere with or manipulate MANET routing?

Modification Attacks

- AODV seq# modification
  - AODV uses seq# as a timestamp (high seq# → fresh)
  - Attacker can raise seq# to make its path attractive
- DSR hop count modification
  - DSR uses #hops for efficiency (low #hops → cheap)
  - Attacker can lower/raise #hops to attract/repel

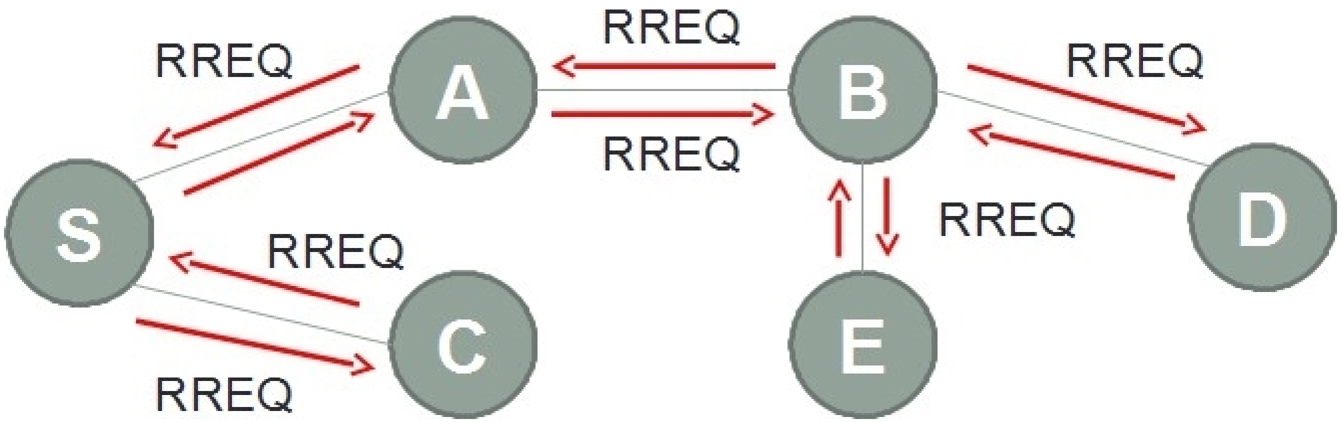
Modification Attacks

- DSR route modification
  - Non-existent route (DoS)
  - Loops (resource exhaustion, DoS)
  - No control to prevent loops after route discovery (more of a data plane attack, we'll get there later)
- Tunneling



RREQ Flooding

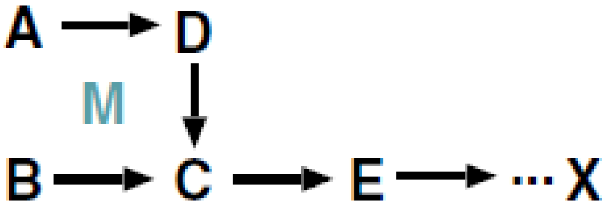
- Flood the network with RREQs to an unreachable destination address



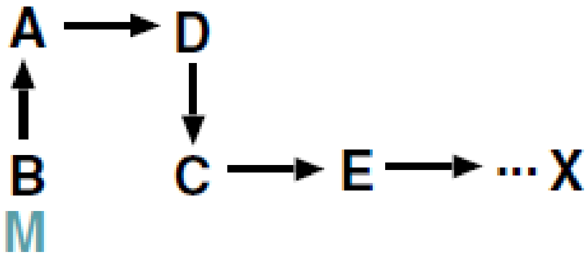
Example : S continuously send RREQ packet to destination X

AODV/DSR Spoofing

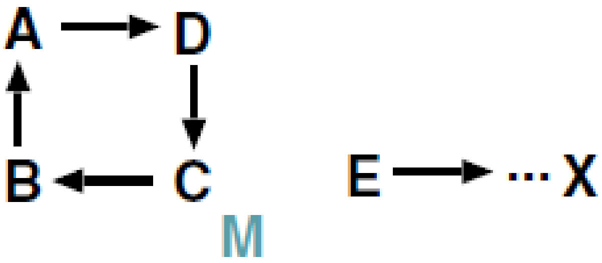
- Attacker listens for RREQ/RREP from neighbors



- Send an "attractive" RREP with spoofed ID

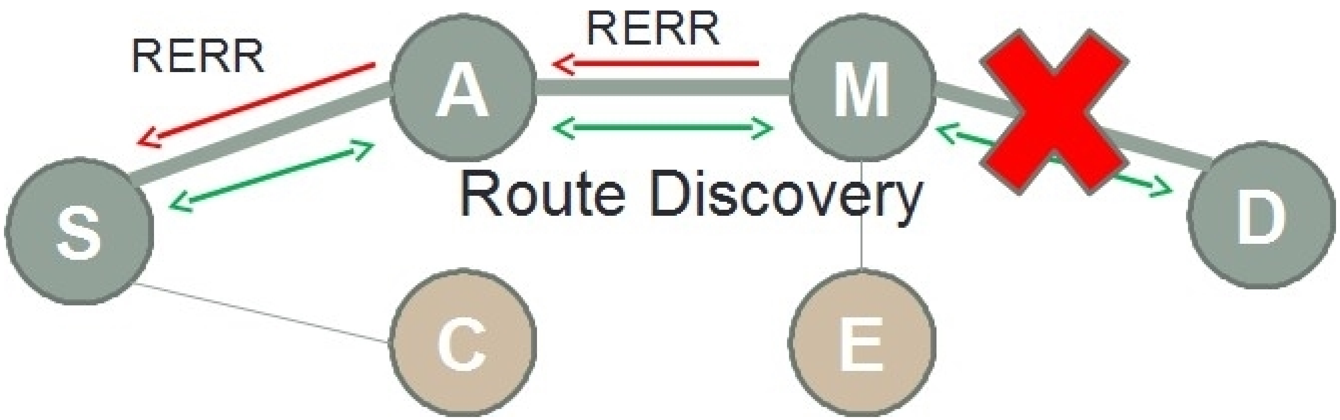


- Spoof more IDs with interesting results



Fabrication Attacks

- DoS against AODV/DSR by falsifying route errors



Control-Plane Security

- How to guarantee that an established path can be efficient (e.g., short) and/or reliable?
- How to prevent attackers from manipulating path discovery/construction?
- What metrics can be used to quantify the value of a path?
  - Length? Latency? Trust?

Agenda

- Examples of approaches for control-plane security
- Data-plane attacks and defenses

Securing DV Routing

- Distance vector (DV) routing is one of the classical approaches to network routing

- SEAD: Secure Efficient Ad hoc DV routing
  - [Hu et al., Ad Hoc Networks 2003]
  - Based on DSDV protocol using sequence numbers to prevent routing loops and async. update issues
  - Uses hash chains to authenticate routing updates
  - Relies on existing mechanisms to distribute authentic hash chain end-elements

Securing LS Routing

- Link-state (LS) routing is another classical approach to network routing

- SLSP: Secure Link-State Protocol
  - [Papadimitratos and Haas, WSAAN 2003]
  - MAC address / IP address pairs are bound using digital signatures
  - Allows for detection of address re-use and change
  - Link state updates are signed and propagated only in a limited zone, with the hop count authenticated by a hash chain

Secure Routing Protocol

[Papadimitratos & Haas, 2002]

- SRP authenticates single-hop exchanges in DSR request and reply messages
  - Since protection is hop-by-hop, SRP over DSR is vulnerable to path (or other parameter) modification

SAODV

[Guerrero Zapata & Asokan, 2002]

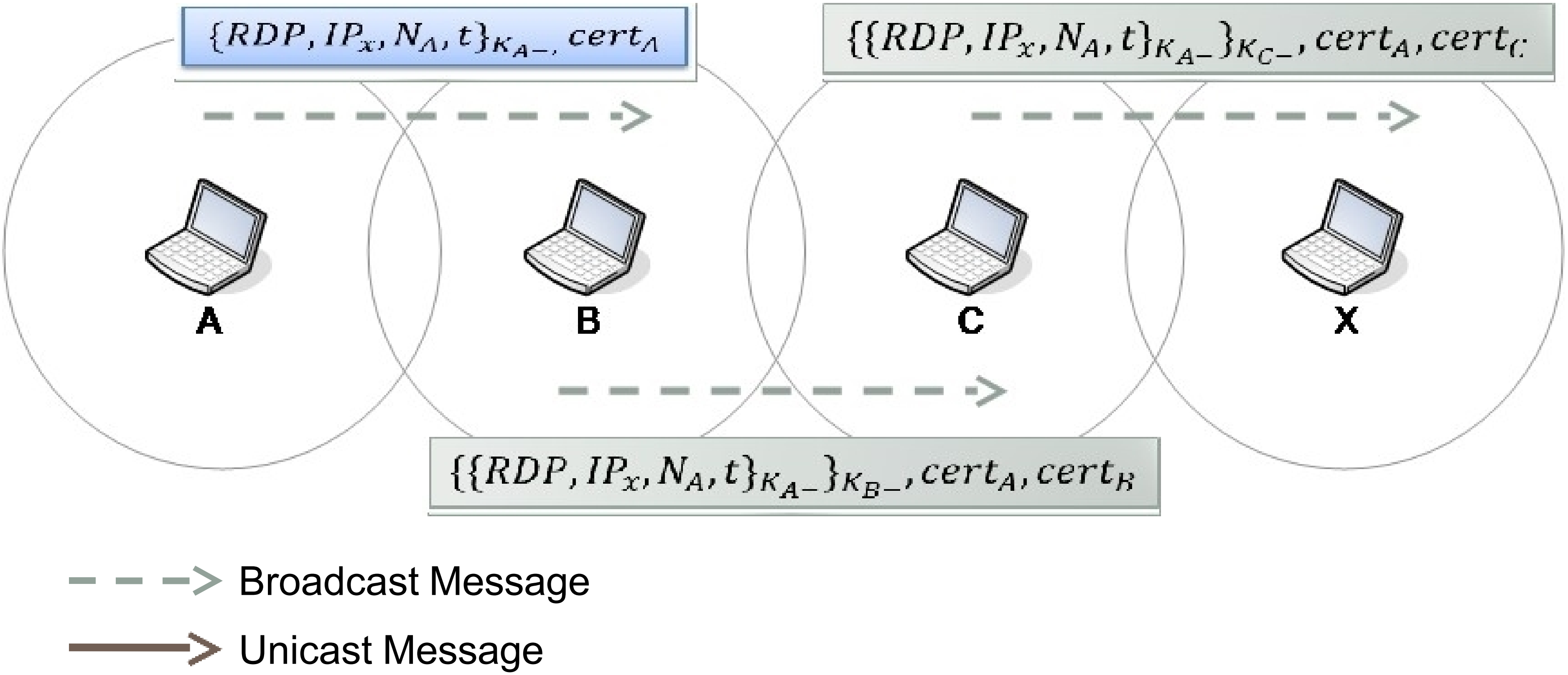
- Secure AODV introduces signatures into the AODV routing protocol to authenticate various message fields
  - RREQ and RREP messages are signed, hop counts are authenticated using hash chains

ARAN

[Sanzgiri et al., ICNP 2002]

- ARAN: Authenticated Routing for Ad hoc Networks (based on AODV)
  - Make use of cryptographic certificates and asymmetric key to achieve authentication, message integrity and non-repudiation
  - Need preliminary certification process before a route instantiation process
  - Routing messages are authenticated at each hop from source to destination and vice versa

Auth. Route Discovery

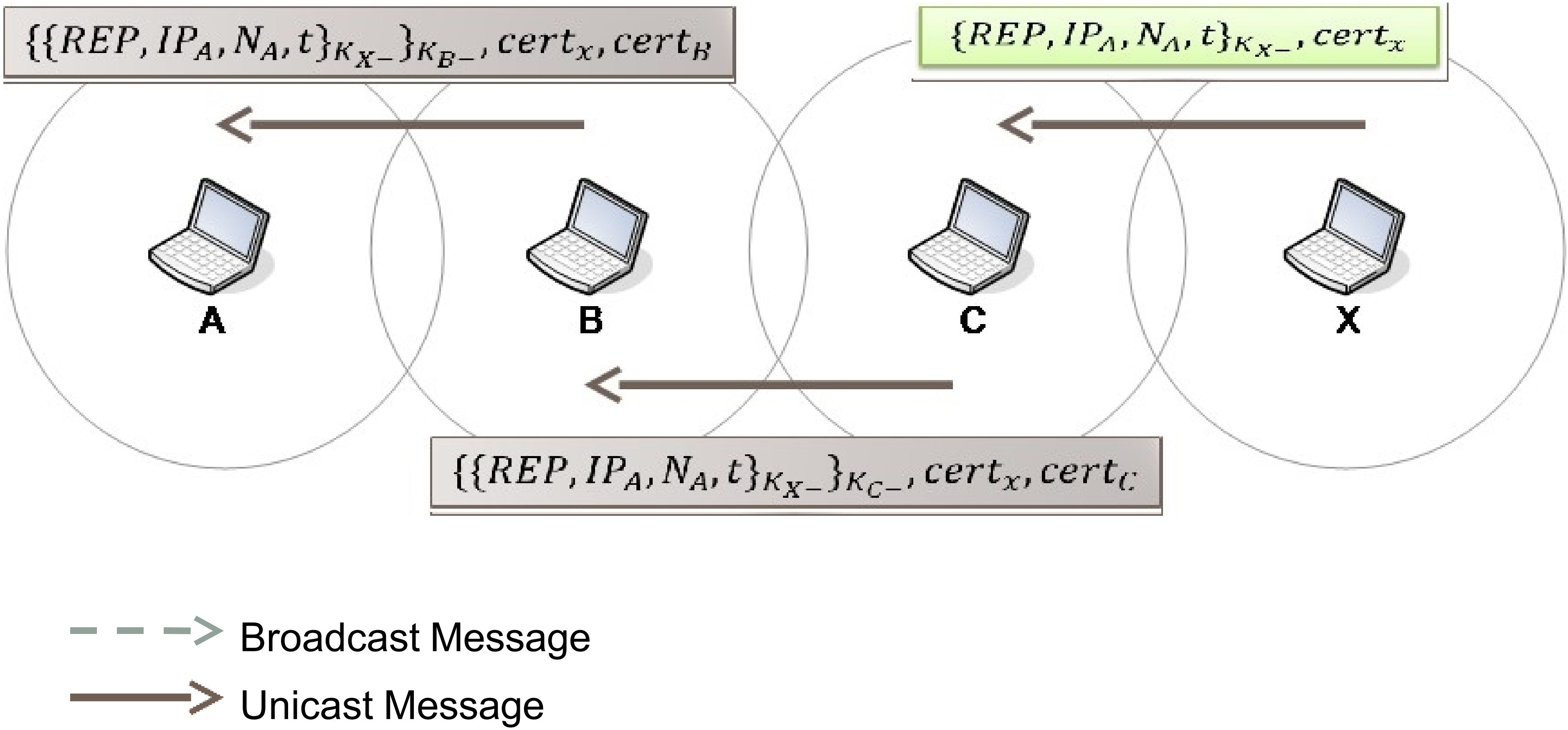


$\{RDP, IP_x, N_A, t\}_{K_{A-}}, cert_A$

$\{\{RDP, IP_x, N_A, t\}_{K_{A-}}\}_{K_{C-}}, cert_A, cert_C$

$\{\{RDP, IP_x, N_A, t\}_{K_{A-}}\}_{K_{B-}}, cert_A, cert_B$

Auth. Route Setup



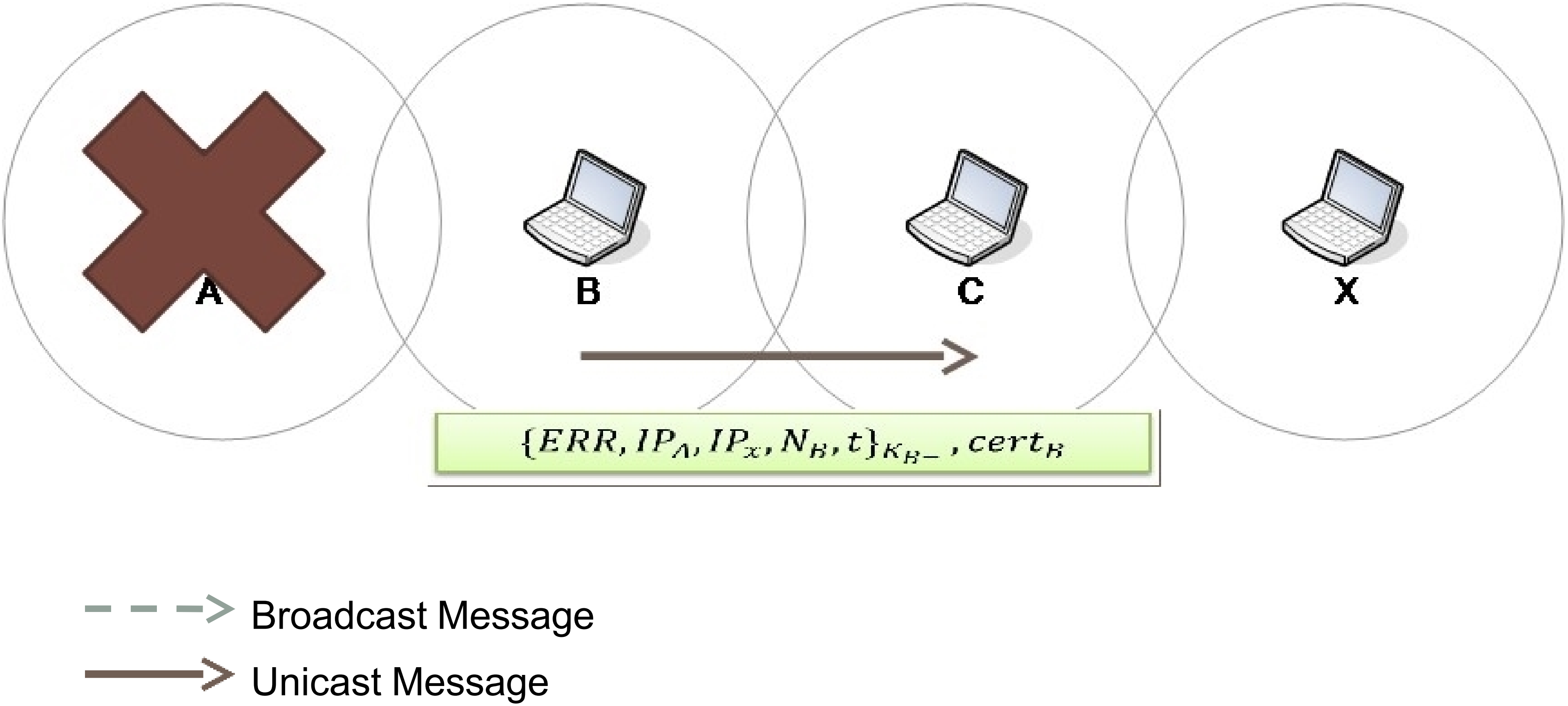
$$\{\{REP, IP_A, N_A, t\}_{K_{X-}}\}_{K_{B-}}, cert_X, cert_B$$

$$\{REP, IP_A, N_A, t\}_{K_{X-}}, cert_X$$

$$\{\{REP, IP_A, N_A, t\}_{K_{X-}}\}_{K_{C-}}, cert_X, cert_C$$

Route Maintenance

- Send ERR message to deactivate route



$\{ERR, IP_A, IP_x, N_B, t\}_{K_{B-}}, cert_B$

Broadcast Message

Unicast Message

ARAN Security

- Modification attacks
  - Prevents redirection using seq# or #hops
  - Prevents DoS with modified source routes
  - Prevents tunneling attacks

- Impersonation attacks
  - Prevents loop-forming by spoofing

- Fabrication attacks
  - Prevents route error falsification

ARAN Limitations

- ARAN relies on an underlying PKI
  - Requires a trusted third-party / infrastructure
  - Requires either:
    - Significant communication overhead to interact with the TTP for near-term updates/revocation
    - Long delays in certificate updates, revocation lists, etc.

What about forwarding security at the data plane?

Data Plane Security

- Injecting and modifying packets are issues of packet/data **integrity**, can be solved using cryptographic techniques
  - Though not efficiently solved...more in a moment
- Forwarding to the wrong next hop is an issue of **protocol compliance**, but can be checked and reported similar to packet/data integrity
- Packet dropping is an issue of **compliance** and **availability**

Data Plane Availability

- Cryptographic primitives alone cannot solve availability problems at the data plane
  - Cannot provide any sort of guarantee about delivering data through routers that misbehave
  - In general, crypto alone cannot solve DoS problems
  - Data plane availability is partially due to compliant behavior of routing nodes and partly due to natural non-deterministic faults, errors, and failures

E2E Delivery Measures

- Suppose packet delivery is measured end-to-end using signatures or MACs
  - Every message carries overhead for packet authentication, but message authentication is already desirable for many other reasons
  - Packet drop induces end-to-end retransmission
    - With high delay if the ACK is also dropped/modified
  - Packet modification forces routers to carry bogus message all the way to the destination node

Limitations

- Paths can only be changed after a large number of end-to-end transactions, i.e., after enough data is available to make a decision
- Path-based detection only identifies a bad path, not a bad node
  - Good nodes may be excluded from networking
  - May have to search a large number of paths to find one with good performance
    - In fact, exponential in #attackers

Limiting the Attacker

- Limiting attacks instead of perfect detection
  - Detect every misbehavior? Costly! Error-prone!
  - Absorb low-impact attack: tolerance threshold
  - Trap the attacker into a ***dilemma***
  - Enable probabilistic algorithms with provable bounds

Attack more?
Will get caught!

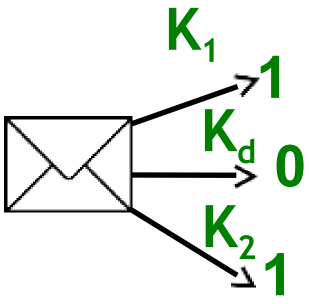
Stay under the threshold?
Damage is bounded!

Source

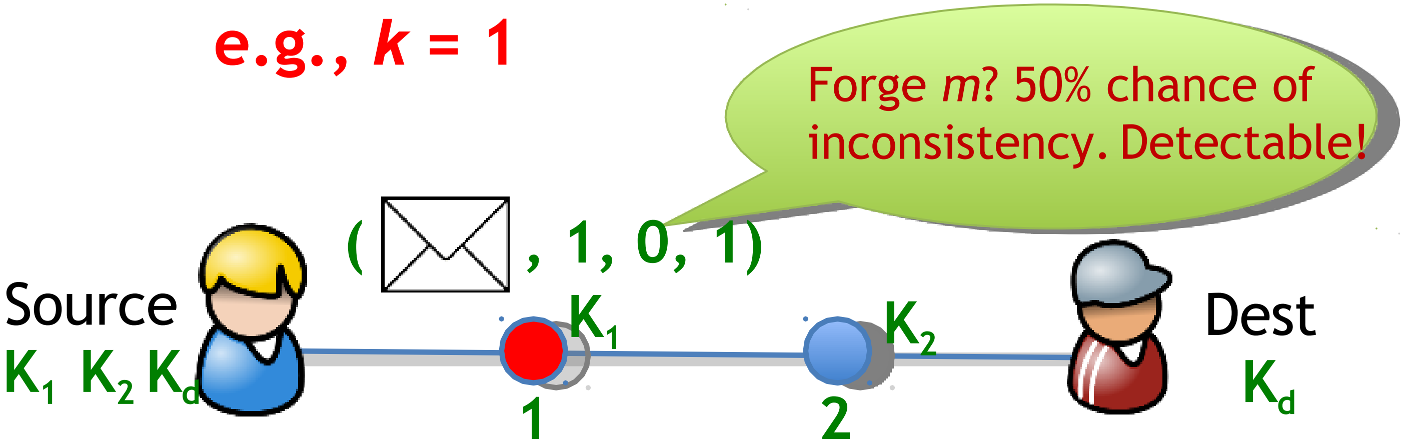
Dest.

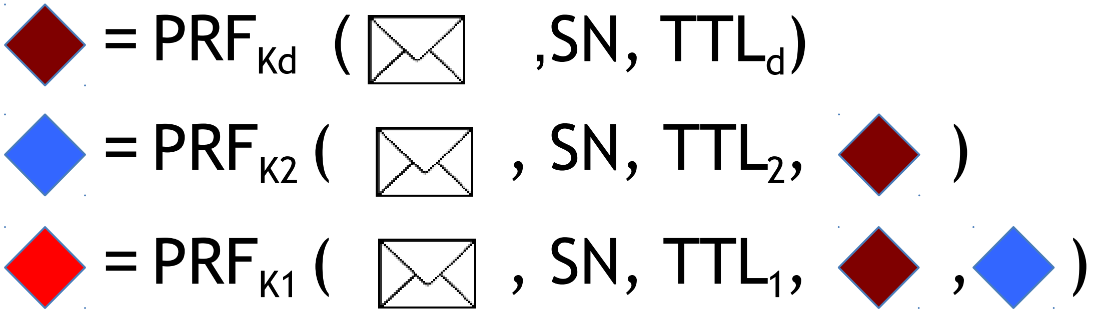
ShortMAC

- ShortMAC packet marking
  - Limiting instead of perfectly detecting fake packets
  - Source marks each packet with k bits (w/ keyed PRF)



k-bit MAC,
e.g., *k* = 1


Forge *m*? 50% chance of inconsistency. Detectable!


◆ = PRF_{Kd} (✉ ,SN, TTL_d)

◆ = PRF_{K2} (✉ , SN, TTL_2, ◆)

◆ = PRF_{K1} (✉ , SN, TTL_1, ◆ , ◆)

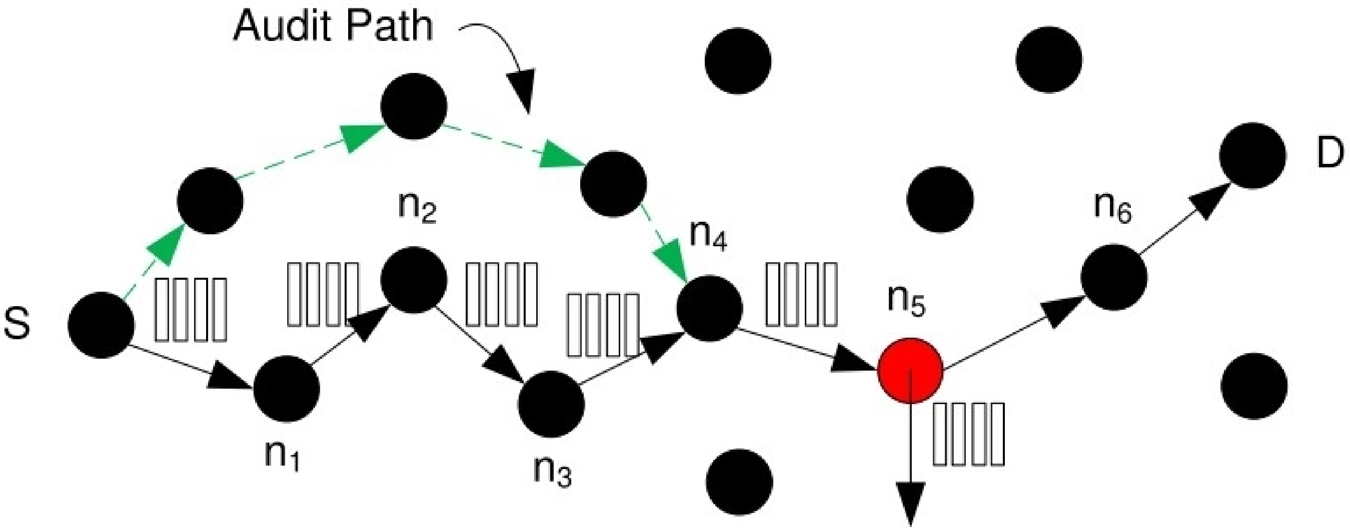
Limitations

- ShortMAC was designed for the Internet and has some implicit assumptions that limit its use in wireless domains
  - Detection is based on a threshold value much higher than a natural packet loss threshold – in wireless, natural packet loss can be high
  - Source must share pairwise symmetric key with every node along the path

Random Audits in MANETs

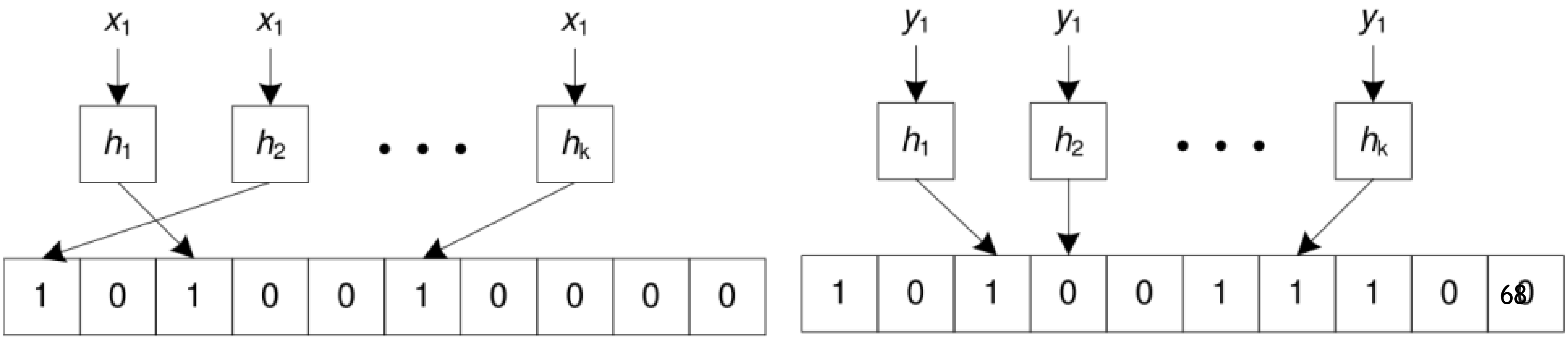
[Kozma & Lazos, WiSec 2009]

- Instead of constantly monitoring every node's forwarding behavior, only perform path audits when end-to-end performance degrades
- To audit a path, the source constructs a disjoint audit path to a node on the path and uses this path to carry audit request/response



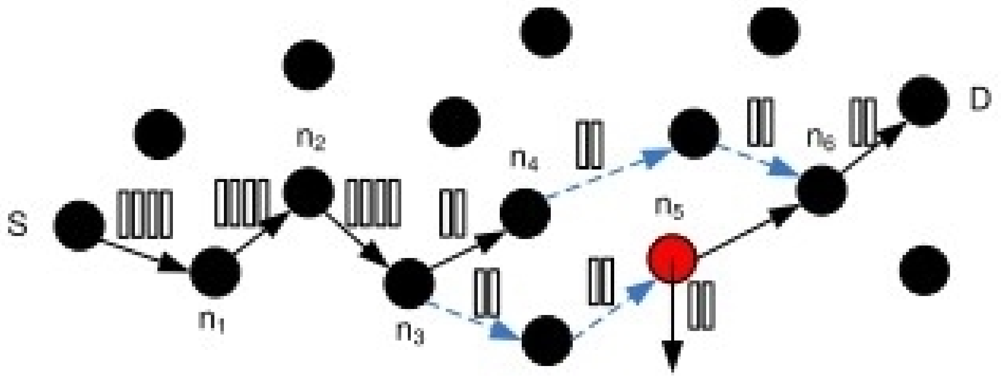
Efficient Auditing

- Upon request, a node generates a proof of which packets it has seen
  - Reporting a list of all packets is inefficient, so compression is required
  - Bloom Filter does lossy packet list compression:
    - A 2^n-bit vector can be indexed by an n-bit hash function
    - Each of k such hash functions maps a packet to a bit
    - Any “0”: the corresponding packet was not received
    - All k “1”s: corresponding packet was probably received



Random Audits

- REAct = Resource Efficient ACcounTability
  - Audits are triggered by performance degradation
  - Source S audits a node N on the path
  - If the returned Bloom filter from N is sufficiently close to that of S, then audit a node downstream
  - Else, audit a node upstream of N
  - Eventually, search will converge to the lossy link
  - Source can change route around the lossy link to identify which node is misbehaving



Limitations

- REAct assumes that attackers have a static attack strategy
  - Dropping packets only when not being audited will work, but it will allow detection in other ways

- REAct assumes that multiple attackers do not collude
  - Colluding attackers can trade duties when being audited, thereby throwing off the search process