Wireless Network Security
Spring 2016

Patrick Tague

Class #12 – Routing Security;
Forwarding Security

SoW

- SoW Presentation
  - Thursday in class
  - I'll post a template
  - Each team gets ~5 minutes

- Written SoW
  - Due Thursday
  - Use IEEE 2-column format

- **Questions?**

Class #12

- Examples of approaches for control-plane security

- Data-plane attacks and defenses

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?

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]

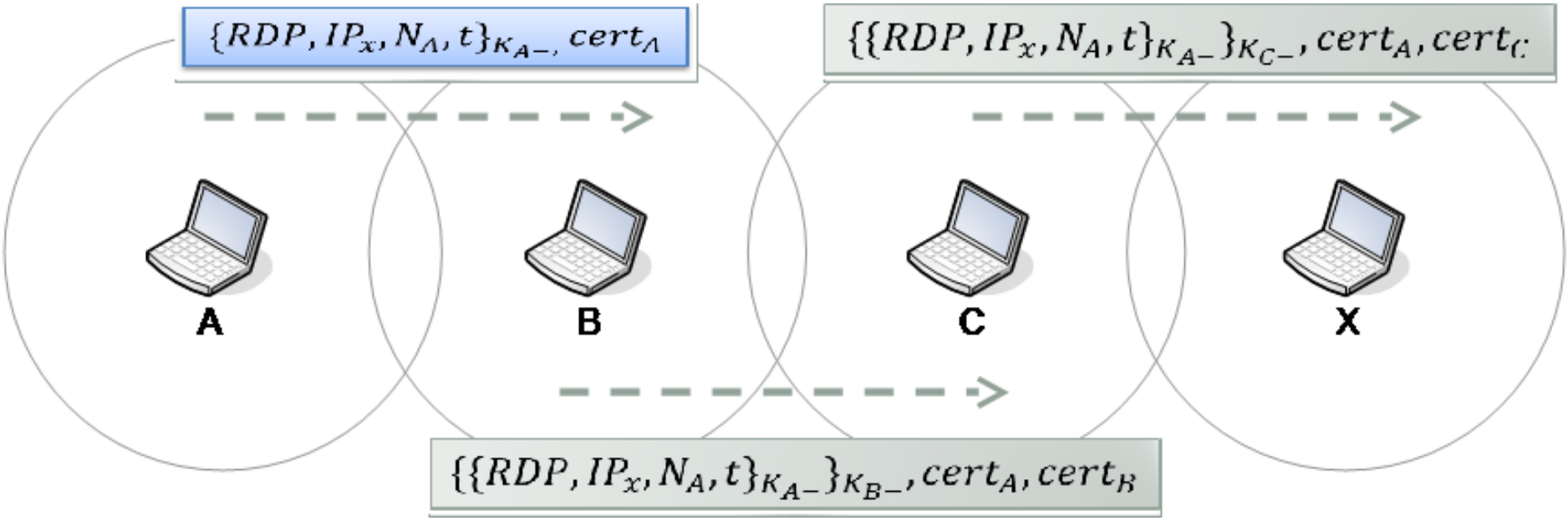
- Secure AODV introduces signatures into the AODV routing protocol to authenticate various message fields
  - RREQ and RREP messages are signed, hop counts are authenticated using hash chains

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

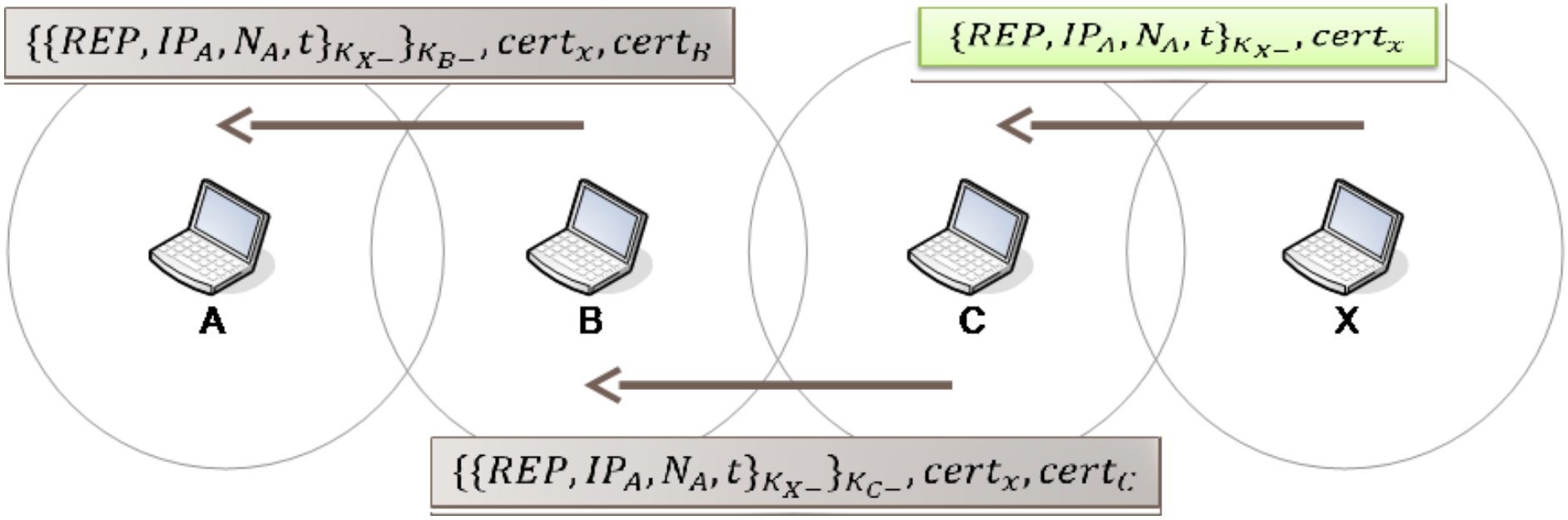
Auth. Route Discovery



Broadcast Message

Unicast Message

Auth. Route Setup

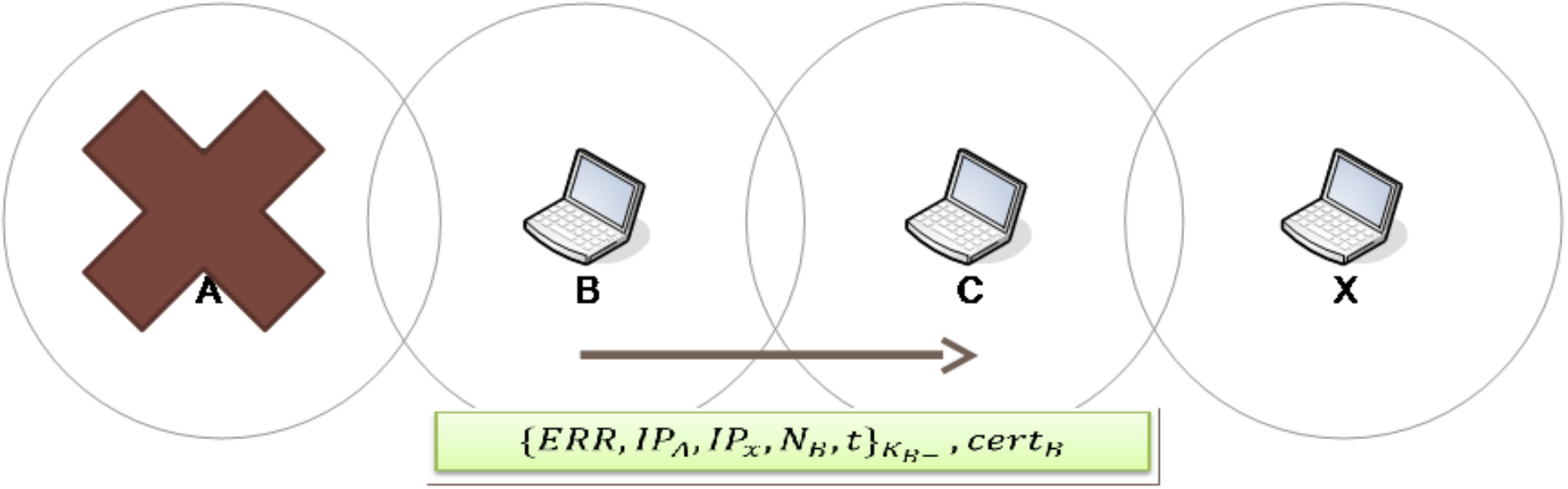


Broadcast Message

Unicast Message

Route Maintenance

- Send ERR message to deactivate route



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.

Ariadne

[Hu, Perrig, & Johnson, 2004]

- Ariadne is a secure on-demand routing protocol built on DSR and Tesla
  - DSR: Dynamic Source Routing, Tesla: Timed Efficient Stream Loss-tolerant Authentication (broadcast auth)
  - Route request and reply messages are authenticated

S: $h_0 = \text{MAC}_{K_{SD}}(\text{REQUEST}, S, D, id, ti)$

$S \rightarrow *$: $\langle \text{REQUEST}, S, D, id, ti, h_0, (), () \rangle$

Ariadne is vulnerable to malicious forwarding by attackers on the selected routing path – requires an additional mechanism to feed back path loss/quality

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

Per-Node Delivery Measures

- Suppose packet delivery is measured per node
  - Verification at finer granularity may require more overhead (e.g., MAC per node)
  - Quicker retransmission requests can be issued by intermediate routers, but malicious routers can also request retransmissions
  - Routers are forced to do more computation and reporting
  - Neighbors may be required to “overhear” behavior

ShortMAC

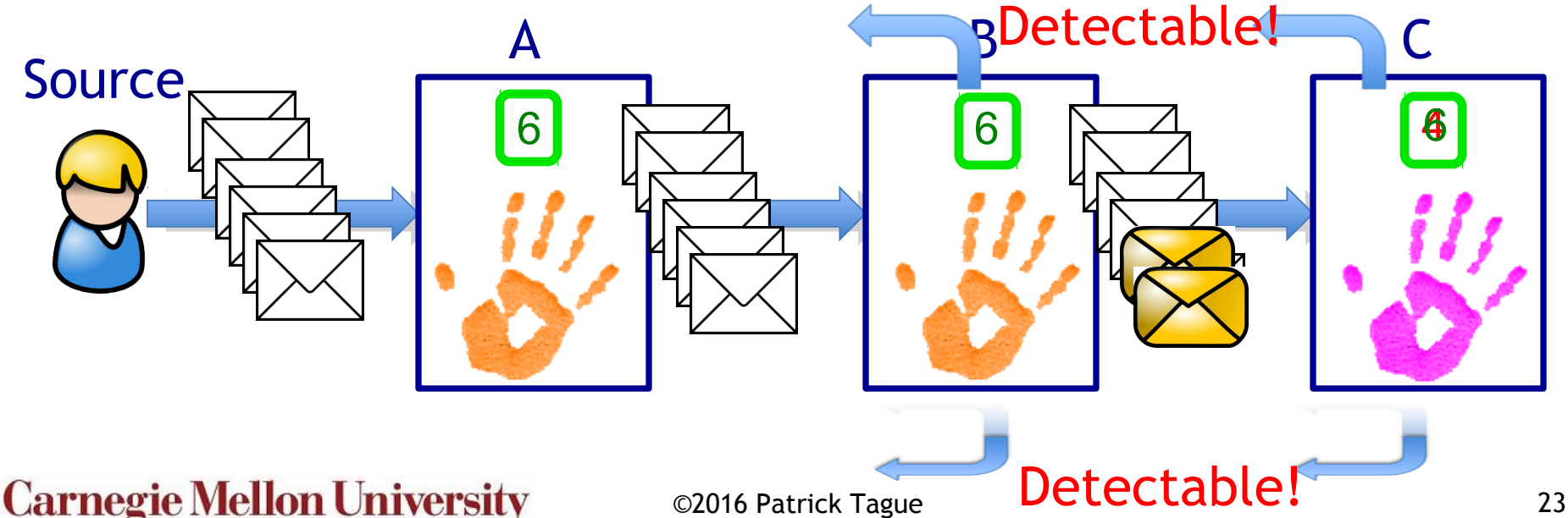
[Zhang et al., NDSS 2012]

- Instead of reacting to poor performance, highly efficient monitoring can enable continuous monitoring with minimal overhead

- A few key design insights allow for significant efficiency gains by making seemingly-significant tradeoffs with detectability

ShortMAC Counters

- Fault Localization → Packet authentication
  - Fault localization → monitor packet *count*, *content*
  - W/ pkt authentication, content → count
  - Counters-only approach yields small state and low communication overhead

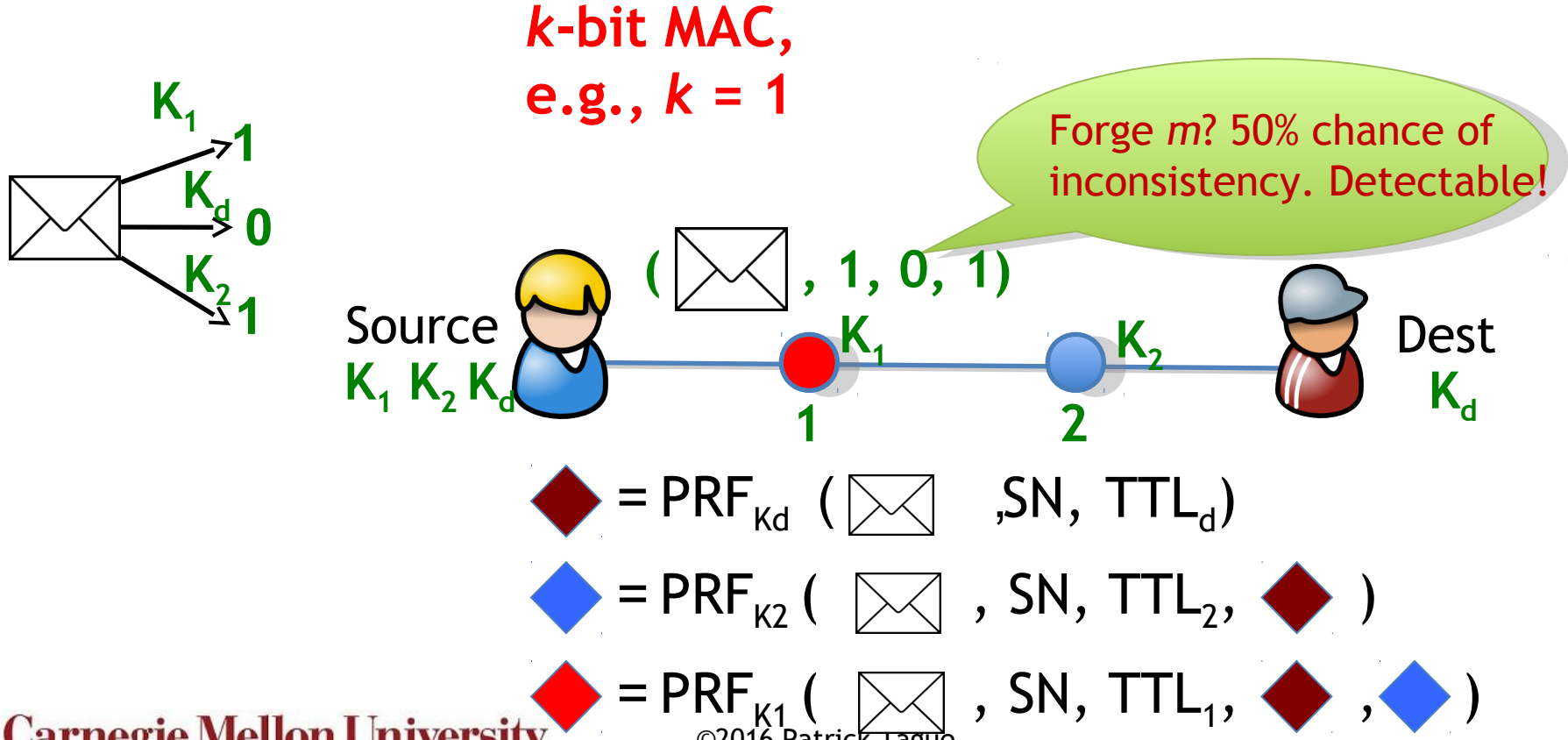


Limiting the Attacker

- 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

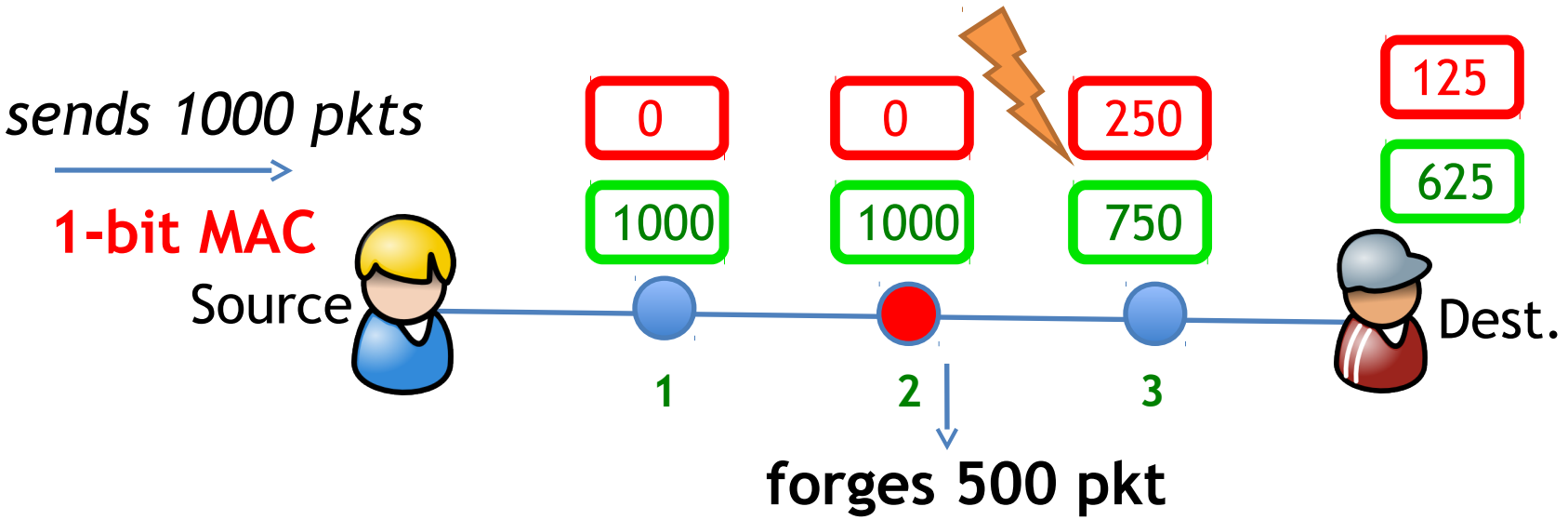
ShortMAC

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



Detection using Counters

- High-level steps
  - Each node maintains two counters (*counter only*!)
  - *Secure* reporting (details in paper)
  - Threshold-based detection (details in paper) robust to *natural errors*



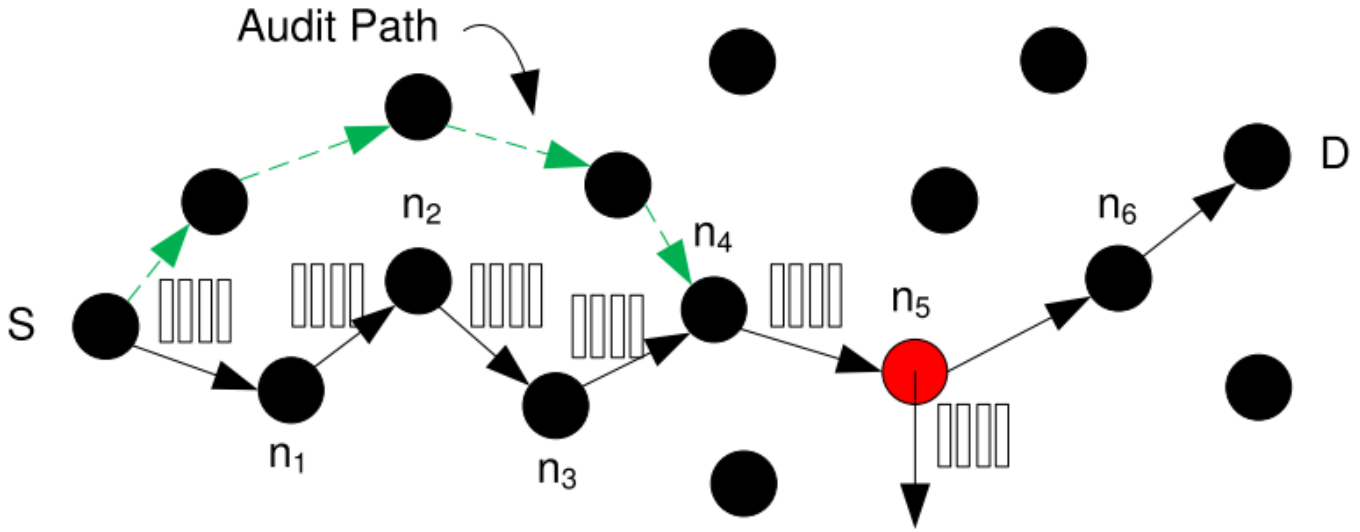
Limitations

- 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

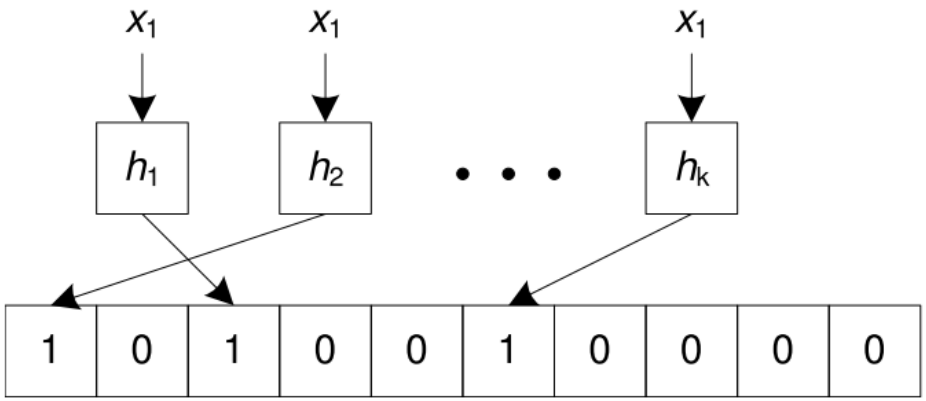
[Kozma & Lazos, WiSec 2009]

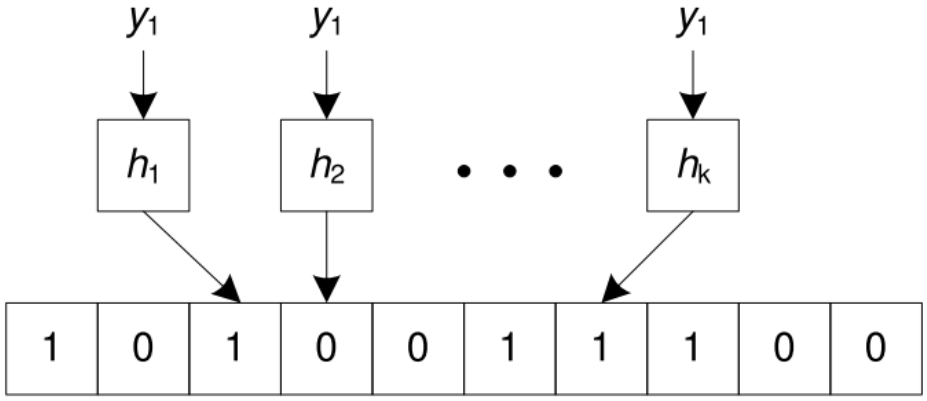
- Instead of constantly monitoring every node's forwarding behavior, only perform path audits when end-to-end performance degrades
- To audit a path, the source constructs a disjoint audit path to a node on the path and uses this path to carry audit request/response



Efficient Auditing

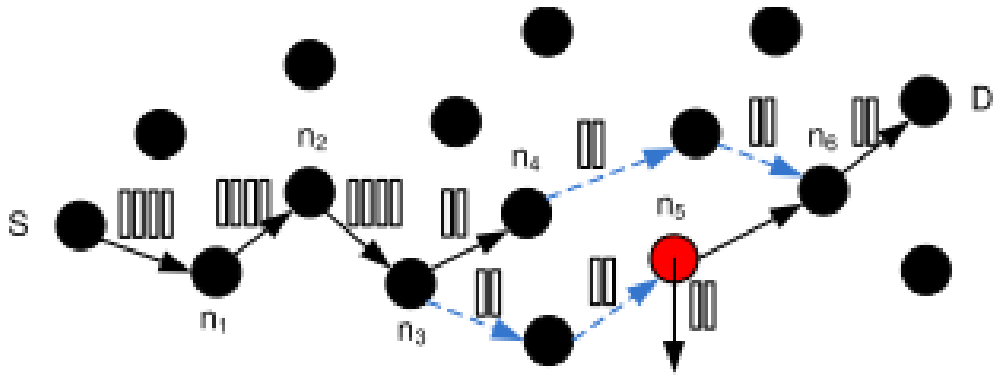
- Upon request, a node generates a proof of which packets it has seen
  - Reporting a list of all packets is inefficient, so compression is required
  - Bloom Filter does lossy packet list compression:
    - A 2^n-bit vector can be indexed by an n-bit hash function
    - Each of k such hash functions 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

Feb 25:
SoW Presentations;
Network Privacy & Anonymity