Protocol Security

More TCP Attacks and S-BGP

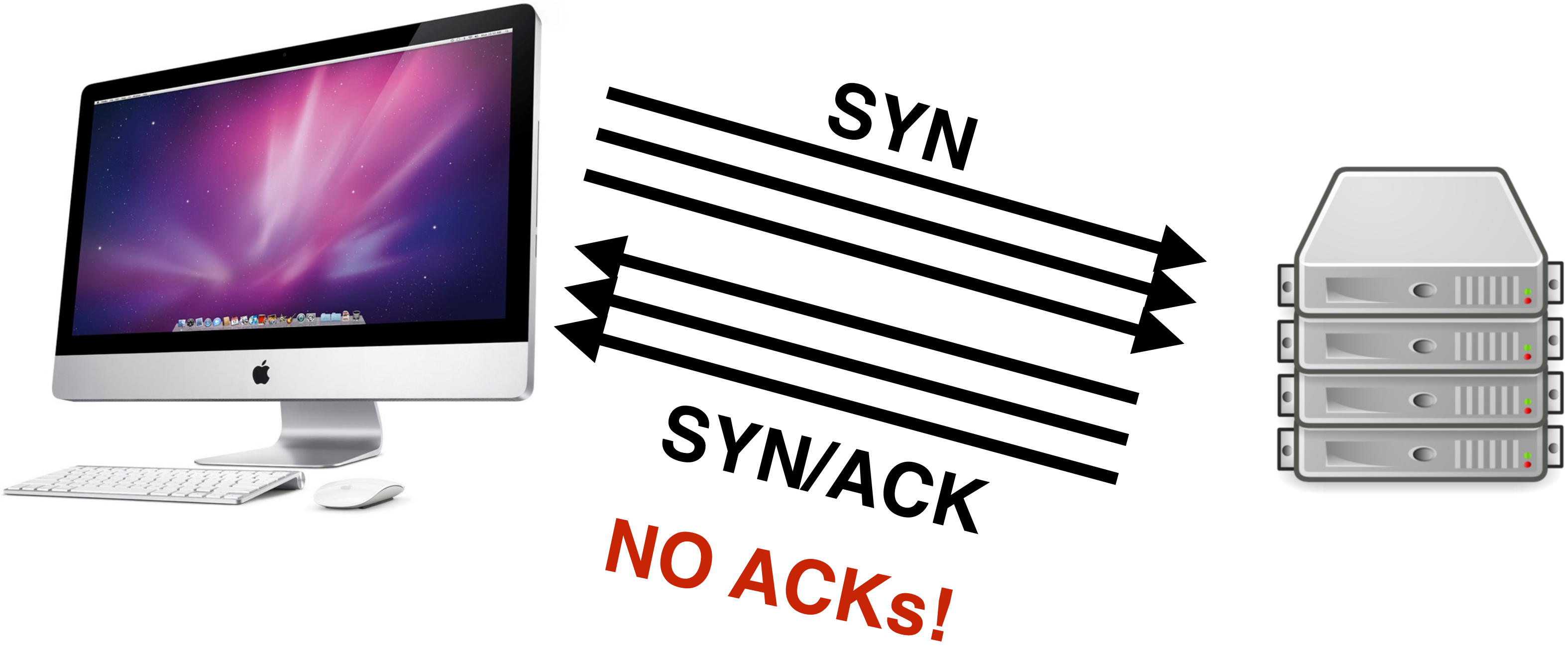
15-441: Computer Networks

Matt Mukerjee
David Naylor
Ben Wasserman

Security!

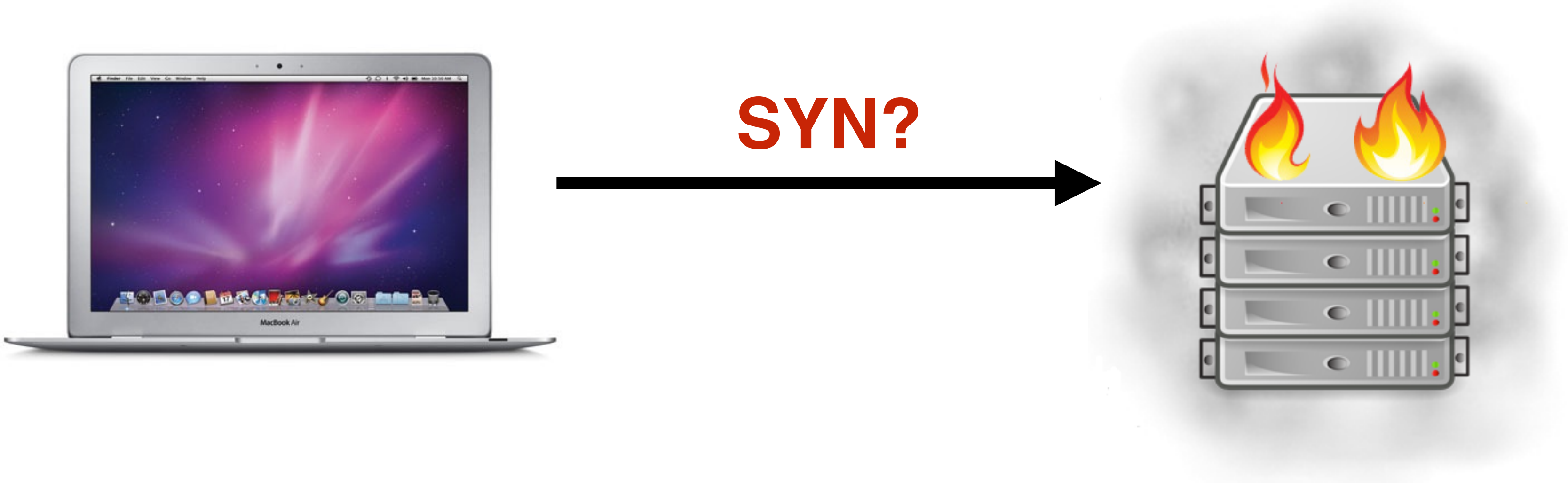
- “Software Security” (Exploiting endhost software)
- “Network Security” (Exploiting infrastructure/proto’s)
  - Attacks at all layers (IP, TCP, Application)
- Today
  - TCP Attacks (and how to fix them)
  - BGP Attacks (and how to fix them)

Remember SYN Floods?



TCP Handshake doesn't complete;
Eats up finite connection queue on server

Remember SYN Floods?



SYN?

Legitimate Hosts can’t connect

SYN Floods

- Solution: Give state to client!
    - Client sends state to server on handshake ACK
    - Problems: How to verify??

TCP SYN Cookies

- Problem 1: How to verify state given by client?
- Solution: Make the state cryptographically secure!
  - Keyed hash of (Src IP, Dst IP, Src Port, Dst Port)

TCP SYN Cookies

- Problem 2: Where do we put this in the packet?
- Solution: Make it the server’s Initial Sequence Number (ISN)!

TCP SYN Cookies

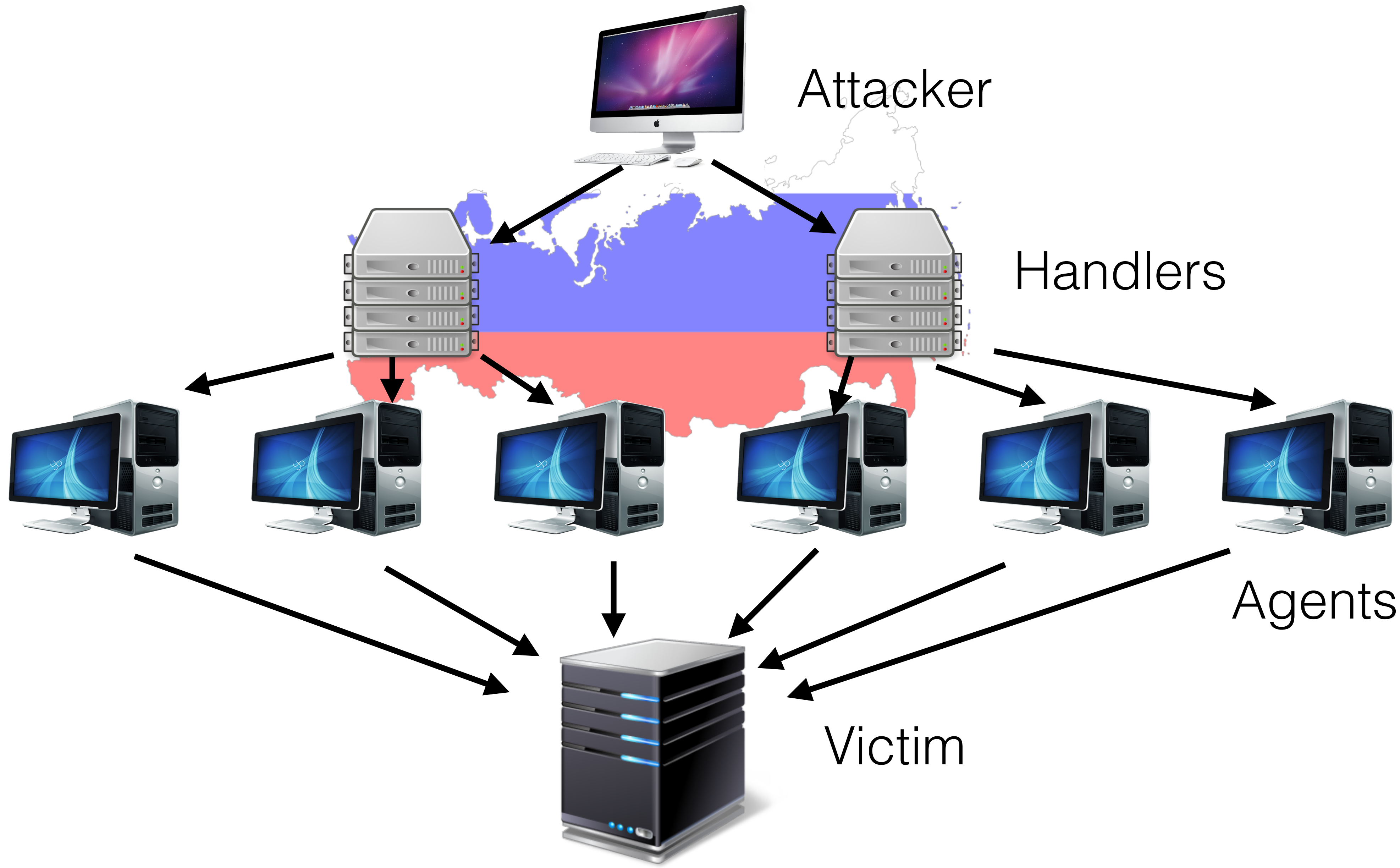
- Problem 3: How to prevent reuse by an attacker?
- Solution: Include a timestamp in the hash!

TCP SYN Cookies

- Problem 4: How to know the timestamp when verifying the hash?

- Solution: Include the timestamp in server’s ISN

Remember DDoS?



Computational Puzzles

- Client must do work before server gives resources
  - Force client invert a hash for a small number
- Must be simple for server to initiate and verify
- Must take client some set amount of time to run
- Minor annoyance for legitimate users; slows DDoS

Computational Puzzles

- Example:
  - Server generates random number *R*
  - Server sends *R* to client
  - Client must find a key *K* for keyed-hash function **H** such that $\mathbf{H}(R)_K$ has 0’s for the first *n* bits. *n* controls the difficulty.
  - Client returns *R*,*K*
  - Server checks first *n* bits of $\mathbf{H}(R)_K$ is 0

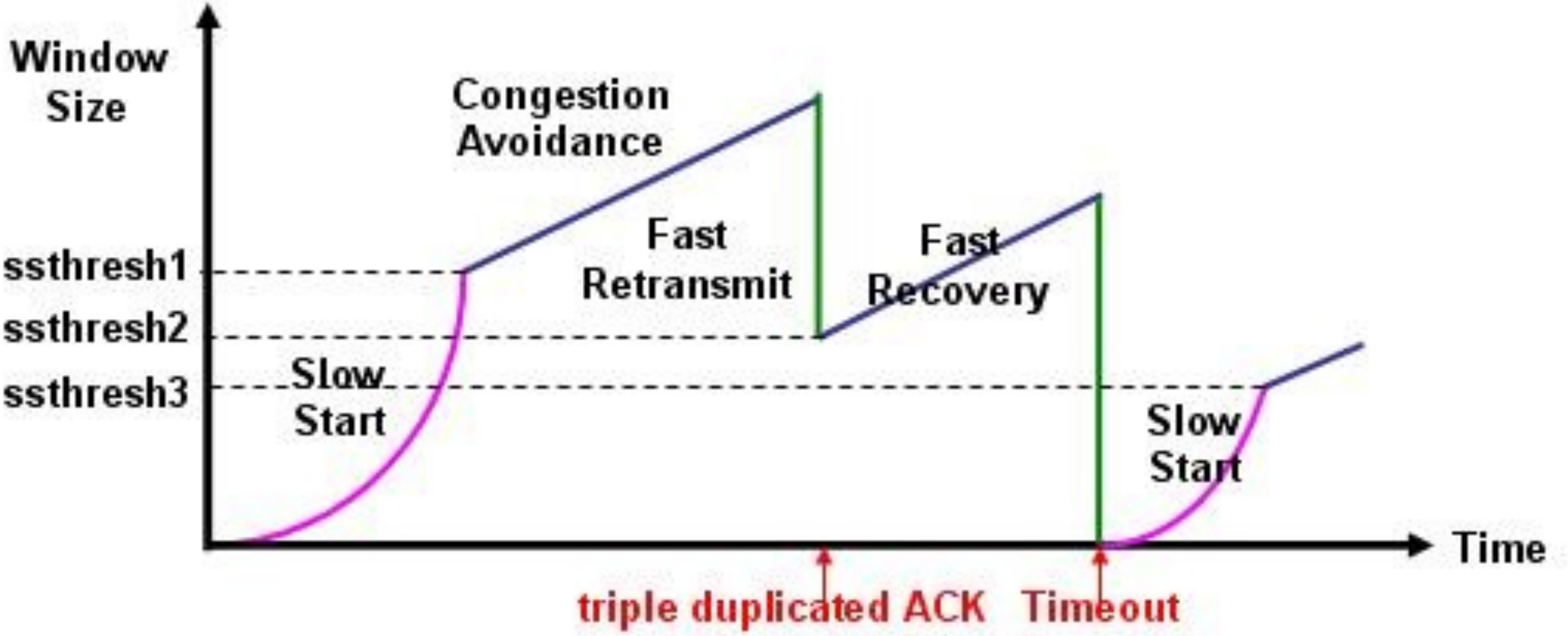
Computational Puzzles

- Problems: Trusting Client's *R*, Liveliness, etc.
- Solutions: Embed data in *R*, provide timestamp, etc

When ACKs Attack!!

- Breaking Congestion Control:
  - Dupe ACKs
  - ACK Division
  - Optimistic ACKs

Remember TCP CC?



Why can’t I just sorta… send a lot of ACKs and get better throughput from a server?

Dupe ACKs

1. Request data from Server
2. Send the same ACK multiple times!
3. ???
4. PROFIT!!! —> (higher throughput!)

Dupe ACKs

- Problem: How to defend? (think about packet loss)
- Solution: Include a nonce in the packet

ACK Division

1. Request data from Server
2. ACK half of a segment at a time
3. ???
4. PROFIT!!! —> (double throughput!)

ACK Division

- Problem: How to Defend?
- Solution: Adjust cwnd based on bytes, not segs

Optimistic ACKs

1. Request Data from Server

2. Send ACKs for Data you haven’t received yet

3. ???

4. PROFIT!!! —> (lower RTT est. == higher tput)

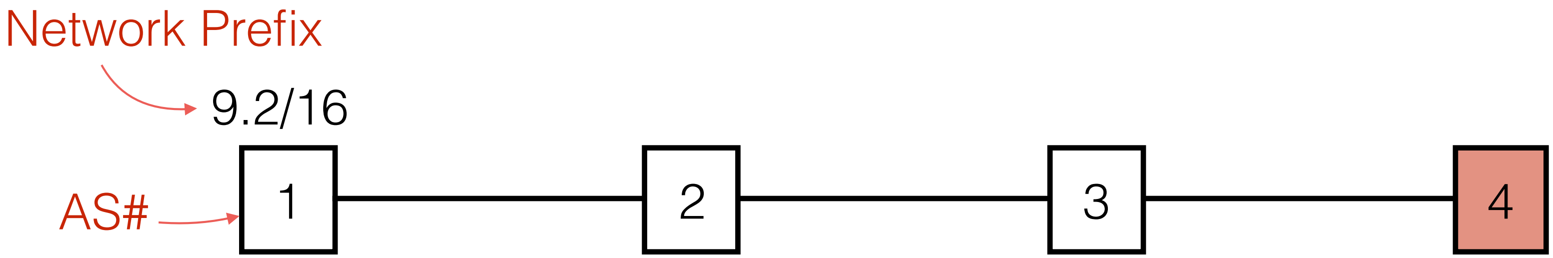
Optimistic ACKs

- Problem: How to Defend?
- Solution: Include a cumulative nonce in the ACKs

TCP Attacks

- SYN Floods + SYN Cookies
- DDoS + Computational Puzzles
- When ACKs Attack!!
    - Dupe ACKs
    - ACK Division
    - Optimistic ACKs

BGP Attacks



AS4 wants to “steal” traffic destined for 9.2/16.

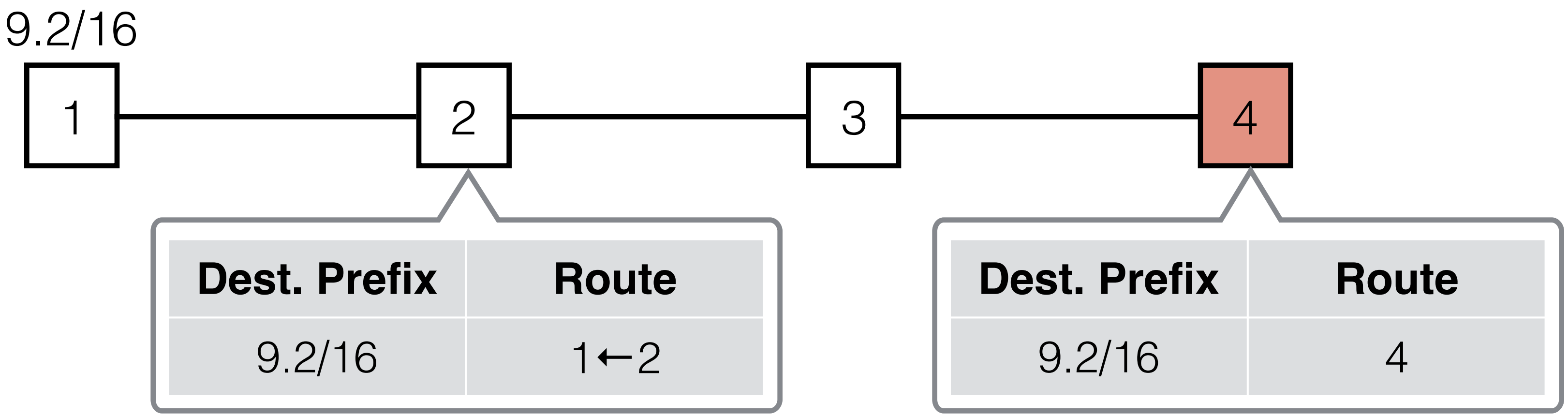
Why?

DoS: Disrupt services running in 9.2/16

Data Interception: AS4 could eventually forward data to 9.2/16…*after it reads/modifies it*

BGP Attacks

Prefix Hijacking



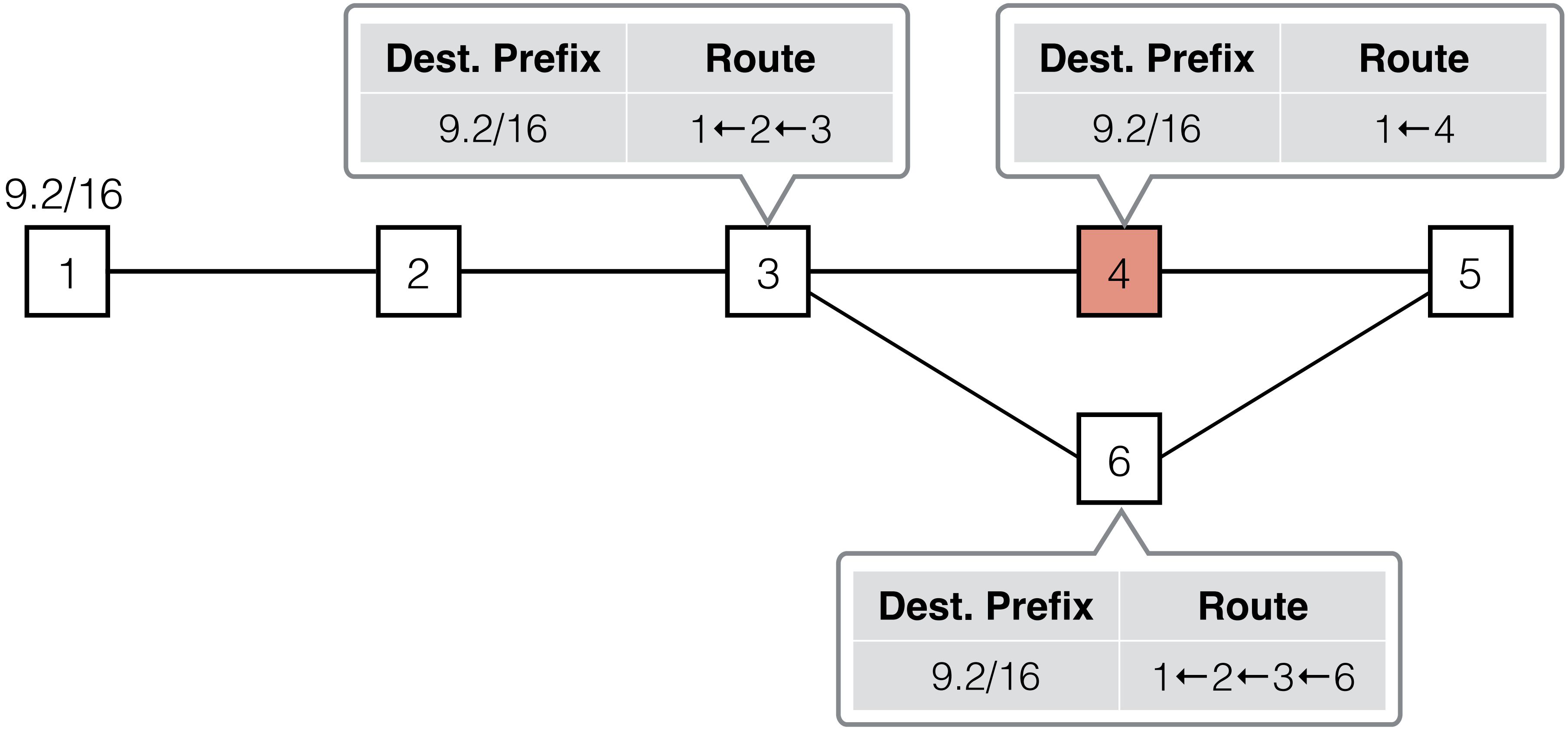
Dest. Prefix	Route
9.2/16	1←2

Dest. Prefix	Route
9.2/16	4

AS3 thinks AS4 has the best route to 9.2/16

BGP Attacks

Path Truncation



Dest. Prefix	Route
9.2/16	1←2←3

Dest. Prefix	Route
9.2/16	1←4

Dest. Prefix	Route
9.2/16	1←2←3←6

AS5 thinks AS4 has the best route to 9.2/16
Works even if 5 knows AS1 owns 9.2/16

S-BGP

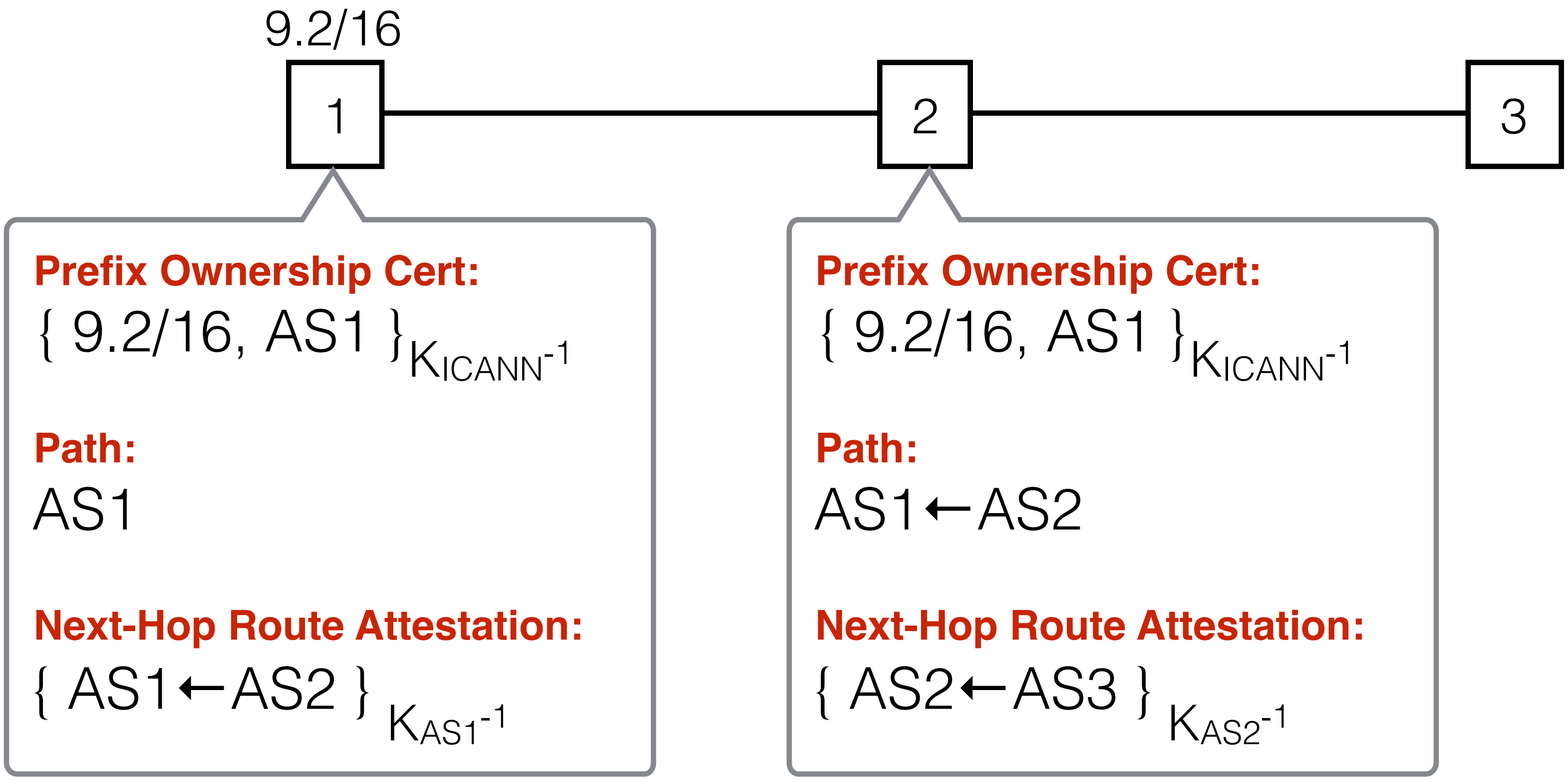
ICANN issues prefix ownership certificates to ASes:

$$\{\ 9.2/16,\ \mathrm{AS1}\ \}_{K_{\mathrm{ICANN}}^{-1}}$$

ASes generate route attestations authorizing next-hop AS to advertise a route:

$$\{\ \mathit{prefix} \,\|\, \mathit{path} \,\|\, \mathrm{AS1} \leftarrow \mathrm{AS2}\ \}_{K_{\mathrm{AS1}}^{-1}}$$

S-BGP



Prefix Ownership Cert:
$\{ 9.2/16, AS1 \}_{K_{ICANN}^{-1}}$

Path:
AS1

Next-Hop Route Attestation:
$\{ AS1 \leftarrow AS2 \}_{K_{AS1}^{-1}}$

Prefix Ownership Cert:
$\{ 9.2/16, AS1 \}_{K_{ICANN}^{-1}}$

Path:
AS1←AS2

Next-Hop Route Attestation:
$\{ AS2 \leftarrow AS3 \}_{K_{AS2}^{-1}}$

Ownership certificate prevents hijacking.
Route attestations prevent path modifications.

Protocol Security

More TCP Attacks and S-BGP

15-441: Computer Networks

Matt Mukerjee
David Naylor
Ben Wasserman