CPS 590.5 Computer Security Lecture 11: IP Traceback and Source Address Authentication

Xiaowei Yang
xwy@cs.duke.edu

Roadmap

- Previous lecture
  - Probabilistic packet marking based IP traceback
- Today
  - Single packet IP traceback
  - Comparison of these two approaches
  - Source Address authentication

Single-Packet IP Traceback

Alex C. Snoeren

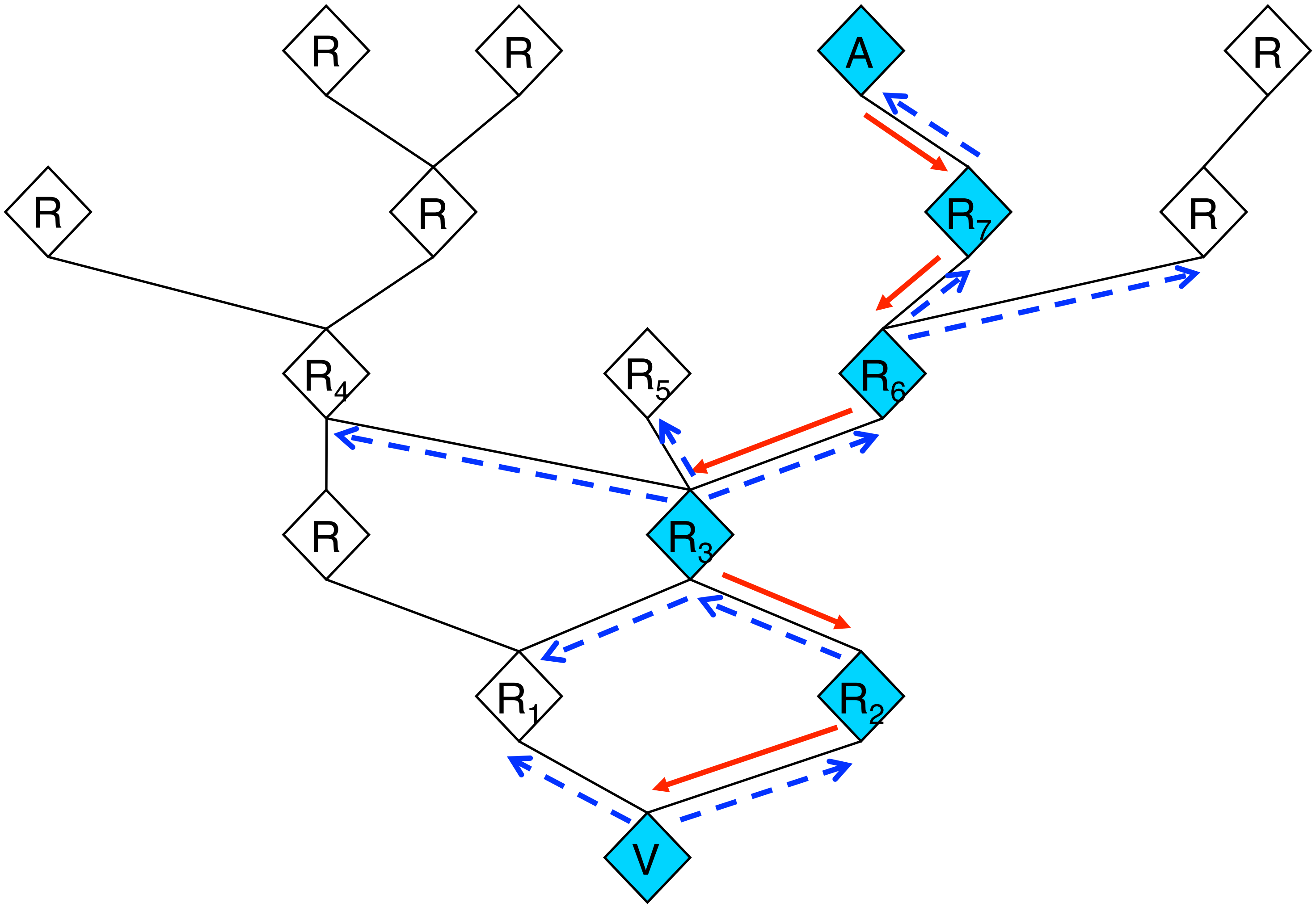
BBN Technologies

(with Craig Partridge, Tim Strayer, Christine Jones,
Fabrice Tchakountio, Beverly Schwartz, Matthew Condell,
Bob Clements, and Steve Kent)

Low-rate attacks

- Not all attacks are large flooding DOS attacks
- Well-placed single packet attacks
- Packets may have spoofed IP addresses
- How to track these attacks and find their origin?

IP Traceback



Logging Challenges

- Attack path reconstruction is difficult
  - Packet may be transformed as it moves through the network
- Full packet storage is problematic
  - Memory requirements are prohibitive at high line speeds (OC-192 is ~10Mpkt/sec)
- Extensive packet logs are a privacy risk
  - Traffic repositories may aid eavesdroppers

Single-Packet Traceback: Goals

- Trace a *single* IP packet back to source
  - Asymmetric attacks (*e.g.*, Fraggle, Teardrop, ping-of-death)
- Minimal cost (resource usage)

One solution: Source Path Isolation Engine (SPIE)

SPIE Architecture

- DGA: Data Generation Agent
  - computes and stores digests of each packet on forwarding path.
  - Deploy 1 DGA per router
- SCAR: SPIE Collection and Reduction agent
  - Long term storage for needed packet digests
  - Assembles attack graph for local topology
- STM: SPIE Traceback Manager
  - Interfaces with IDS
  - Verifies integrity and authenticity of Traceback call
  - Sends requests to SCAR for local graphs
  - Assembles attack graph from SCAR input

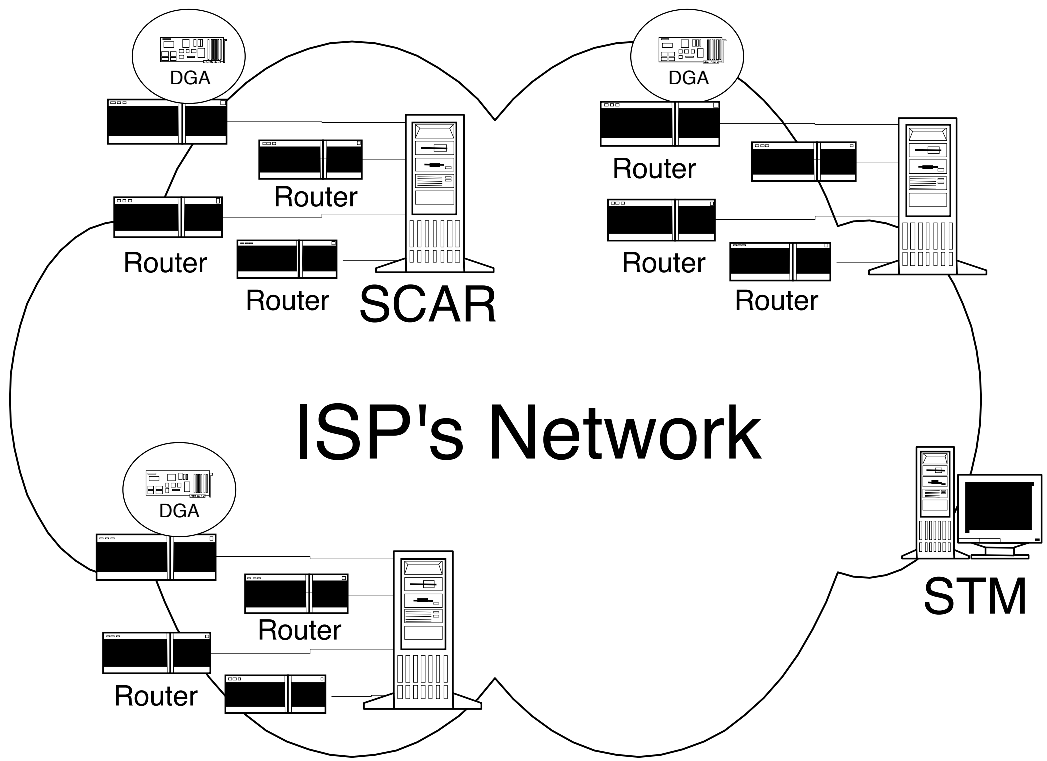


Fig. 4. The SPIE network infrastructure, consisting of Data Generation Agents (DGAs), SPIE Collection and Reduction Agents (SCARs), and a SPIE Traceback Manager (STM).

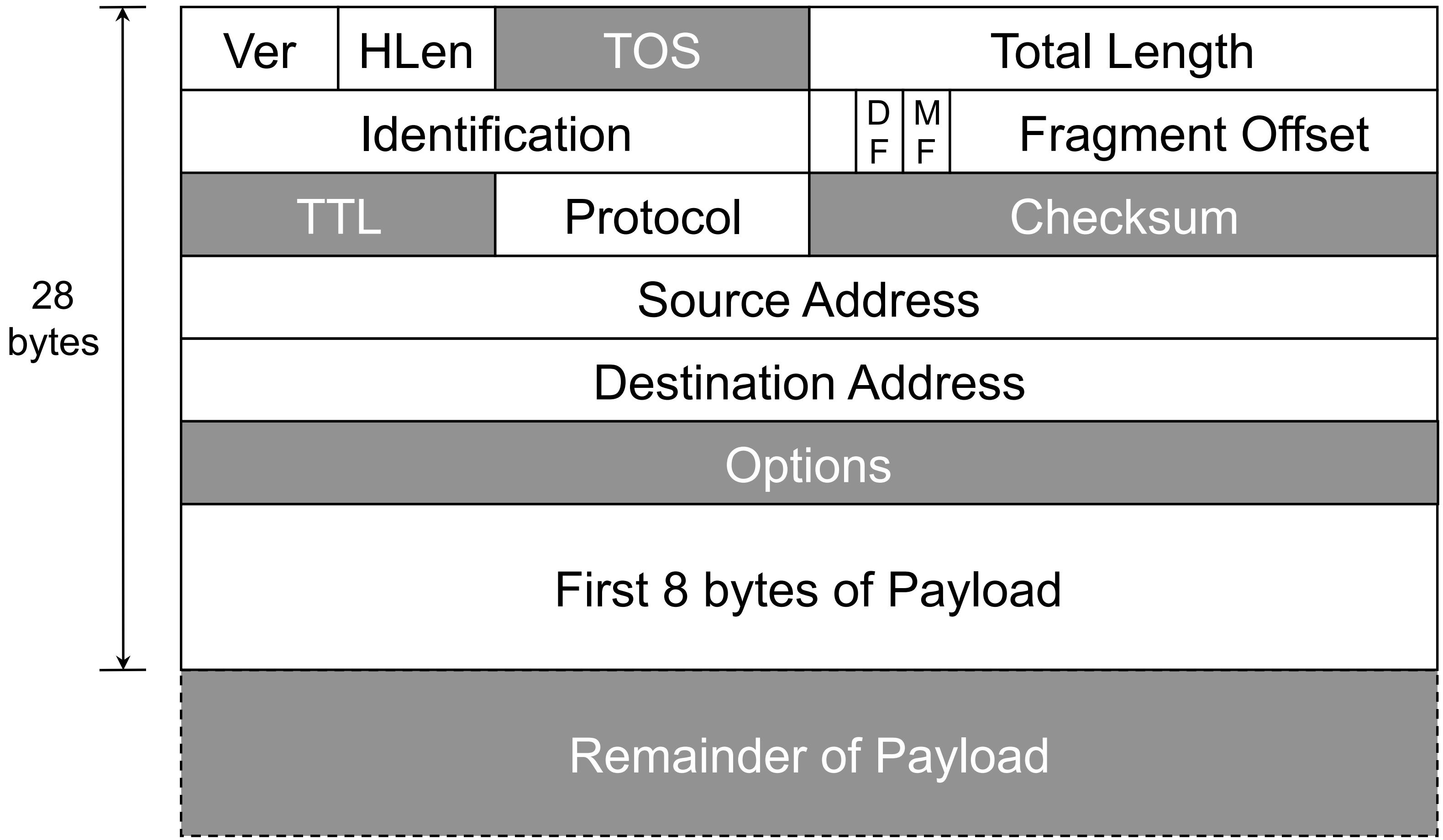
Data Generation Agents

- Compute "packet digest"
- Store in Bloom filter
- Flush filter periodically

Packet Digests

- Compute hash(p)
  - Invariant fields of p only
  - 28 bytes hash input, 0.00092% WAN collision rate
  - Fixed sized hash output, *n*-bits
- Compute *k* independent digests
  - Increased robustness
  - Reduced collisions, reduced false positive rate

Hash input: Invariant Content



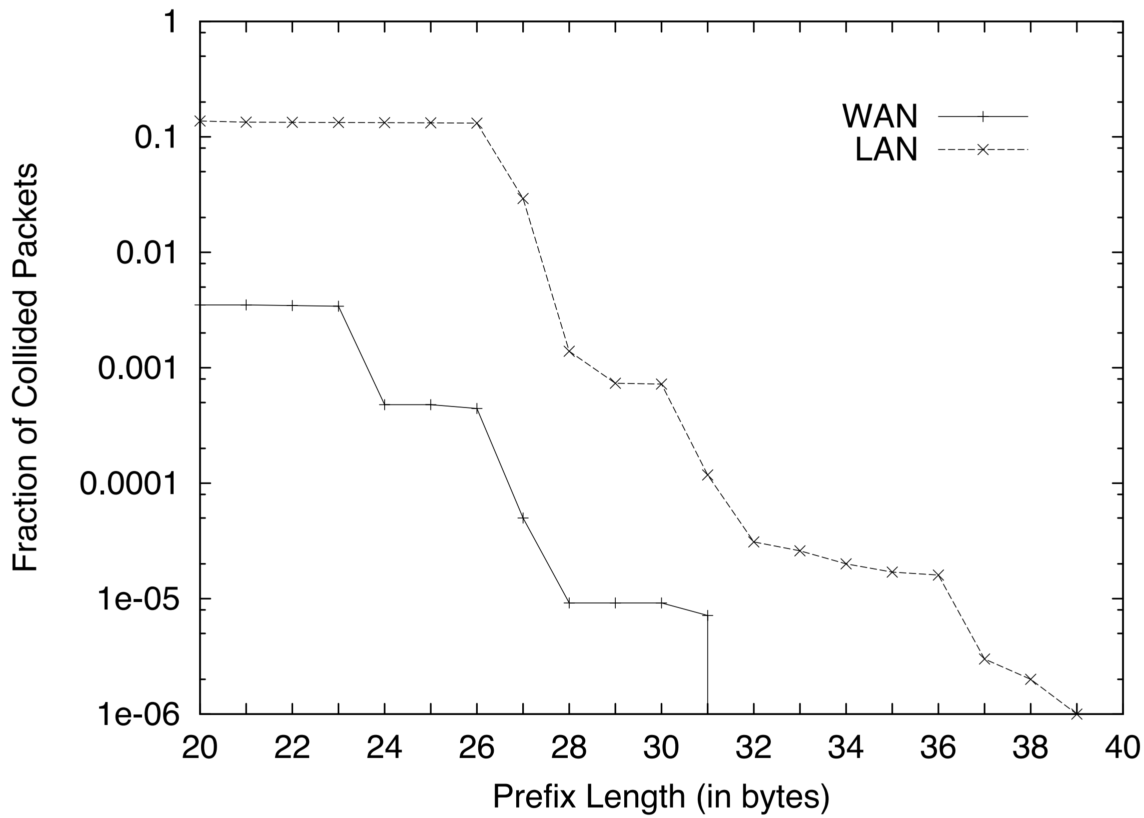


Fig. 2. The fraction of packets that collide (with ToS, TTL, and checksum fields masked out) as a function of prefix length. The WAN trace represents 985,150 packets (with 5,801 duplicates removed) between 6,031 host pairs collected on July 20, 2000 at the University of Florida OC-3 gateway. The LAN trace consists of one million packets (317 duplicates removed) between 2,879 host pairs observed on an Ethernet segment at the MIT Lab for Computer Science.

Hashing Properties

- Each hash function
  - Uniform distribution of input -> output

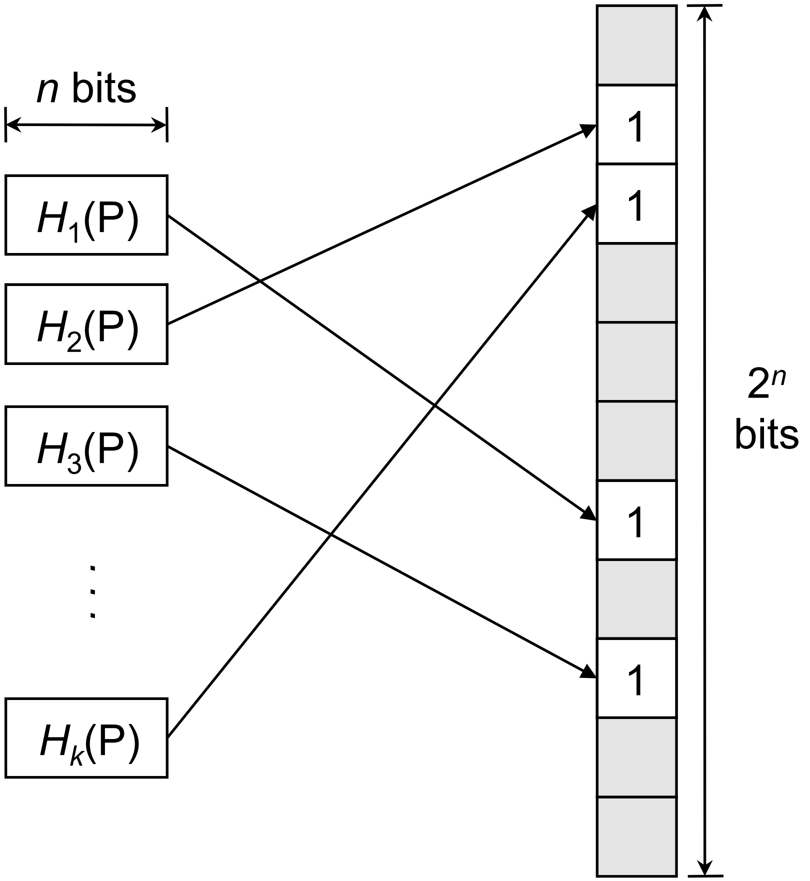
    H1(x) = H1(y) for some x,y -> unlikely

- Use k independent hash functions
  - Collisions among k functions independent
  - H1(x) = H2(y) for some x,y -> unlikely

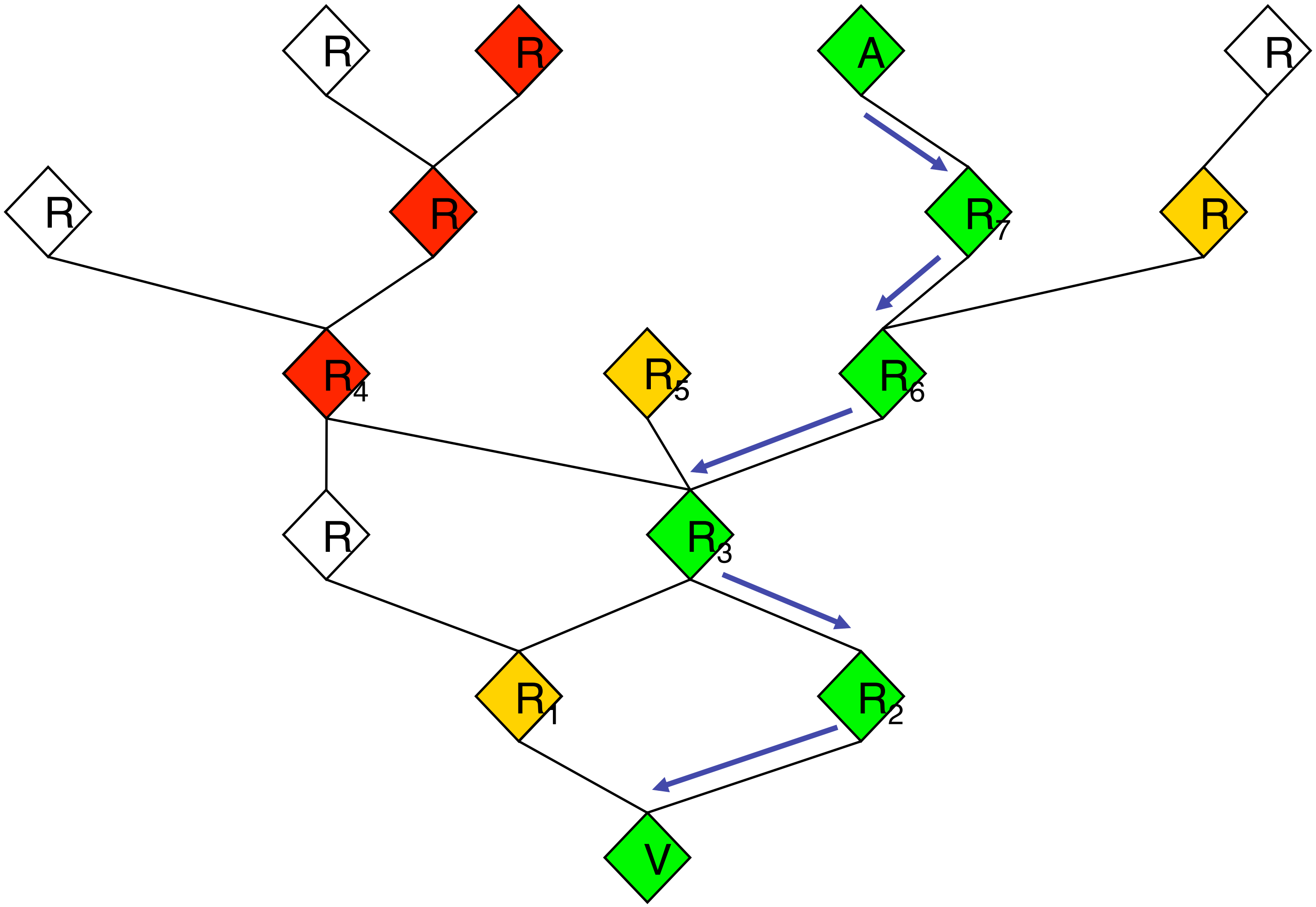
- Cycle k functions every time interval, t

Digest Storage: Bloom Filters

- **Fixed structure size**
  - Uses 2^n bit array
  - Initialized to zeros
- **Insertion**
  - Use n-bit digest as indices into bit array
  - Set to '1'
- **Membership**
  - Compute k digests, d_1, d_2, etc…
  - If (filter[d_i]=1) for all i, router forwarded packet



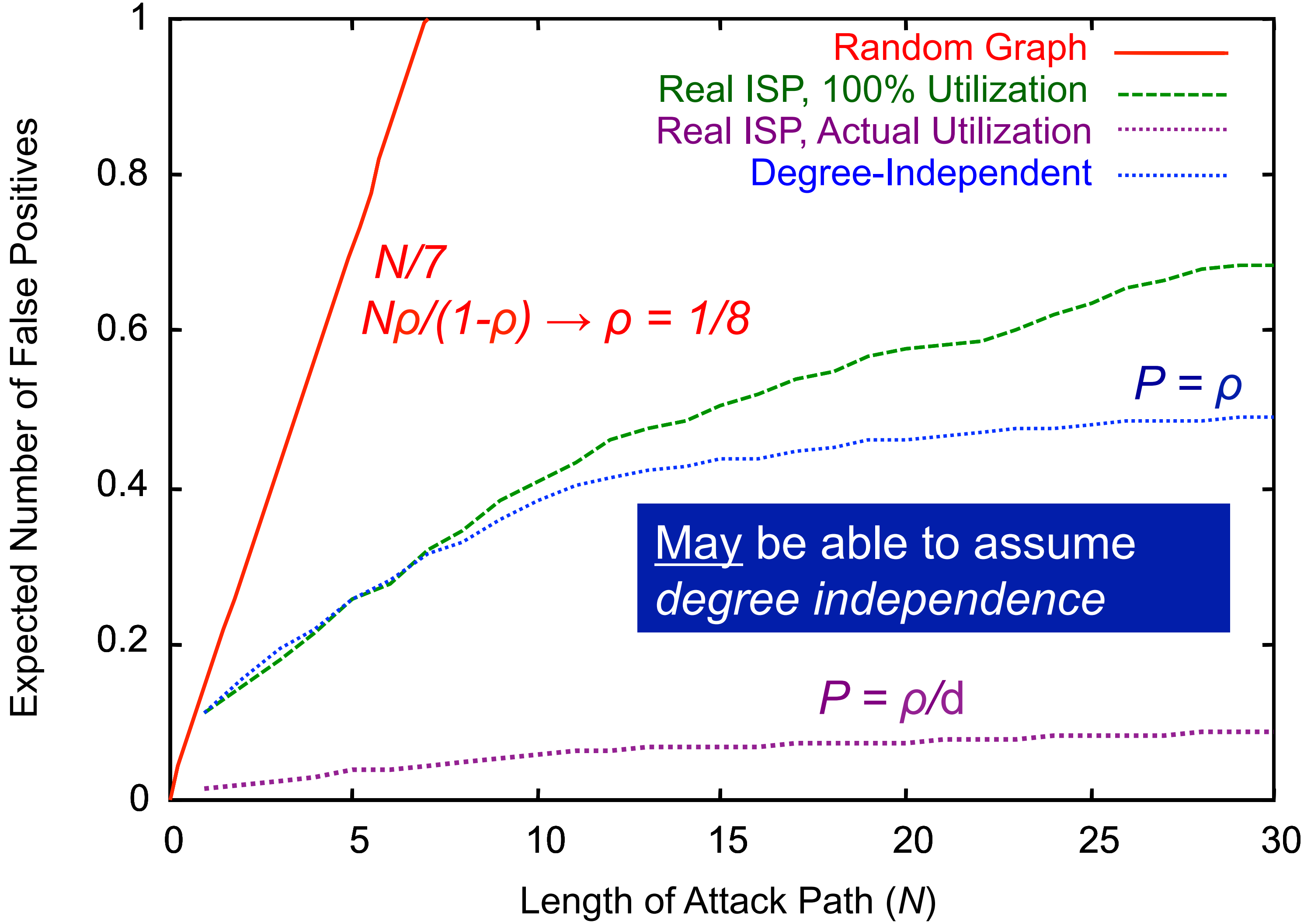
False Positive Distribution



Adjusting Graph Accuracy

- False positives rate depends on:
  - Length of the attack path, N
  - Complexity of network topology, d
  - Capacity of Bloom filters, P
- Bloom filter capacity is easy to adjust
  - Required filter capacity varies with router speed and number of neighbors
  - Appropriate capacity settings achieve linear error growth with path length

Simulation Results



How Big are Digests?

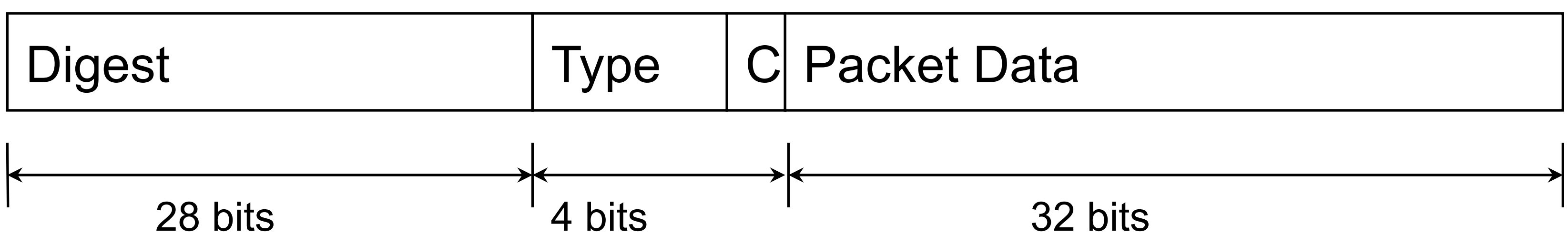
- Quick rule of thumb:
  - $\rho = 1/8$, assuming degree independence
  - Bloom filter $k = 3$, M/n = *5 bits per packet.*
  - Assume packets are ~1000 bits
- Filters require *~0.5%* of link capacity
  - Four OC-3s require 47MB per minute
  - 128 OC-192 links need <100GB per minute
- Access times are equally important
  - Current drives can write >3GB per minute
  - OC-192 needs SRAM access times

Transformations

- Occasionally invariant content changes
  - Network Address Translation (NAT)
  - IP/IPsec Encapsulation, etc.
  - IP Fragmentation
  - ICMP errors/requests
- Routers need to invert these transforms
  - Often requires additional information
  - Can store this information at the router

Transform Lookup Table

- Only need to restore invariant content
  - Often available from the transform (*e.g.*, ICMP)
- Otherwise, save data at transforming router
  - Index required data by transformed packet digest
  - Record transform type and sufficient data to invert
- Bounded by transform performance of router



Prototype Implementation

- Implemented in PC-based routers
  - Both FreeBSD and Linux implementations
    - Packet digesting on kernel forwarding path
  - Zero-copy kernel/user digest tables
    - Digest tables and TLT stored in kernel space
- User-level query-support daemons
  - Supports automatic topology discovery
  - Queries automatically triggered by IDS

Summary

- Hash-based traceback is viable
  - With reasonable memory constraints
  - Supports common packet transforms
  - Timely tracing of *individual* packets
- Publicly available implementations
  - FreeBSD/Linux versions available now
  - SPIEDER-based solution in development

http://www.ir.bbn.com/projects/SPIE

Discussion

- Single-packet v.s. probabilistic marking
  - Goals
  - Assumptions
  - Performance
  - Cost

Accountable source IP addresses

- Traceback does not prevent source address spoofing attacks
- Does not automatically stop the attack
  - Reflector attacks
- Question: can we make the IP source address accountable?

Passport: secure and adoptable source authentication

Xin Liu, Ang Li, Xiaowei Yang

UC Irvine

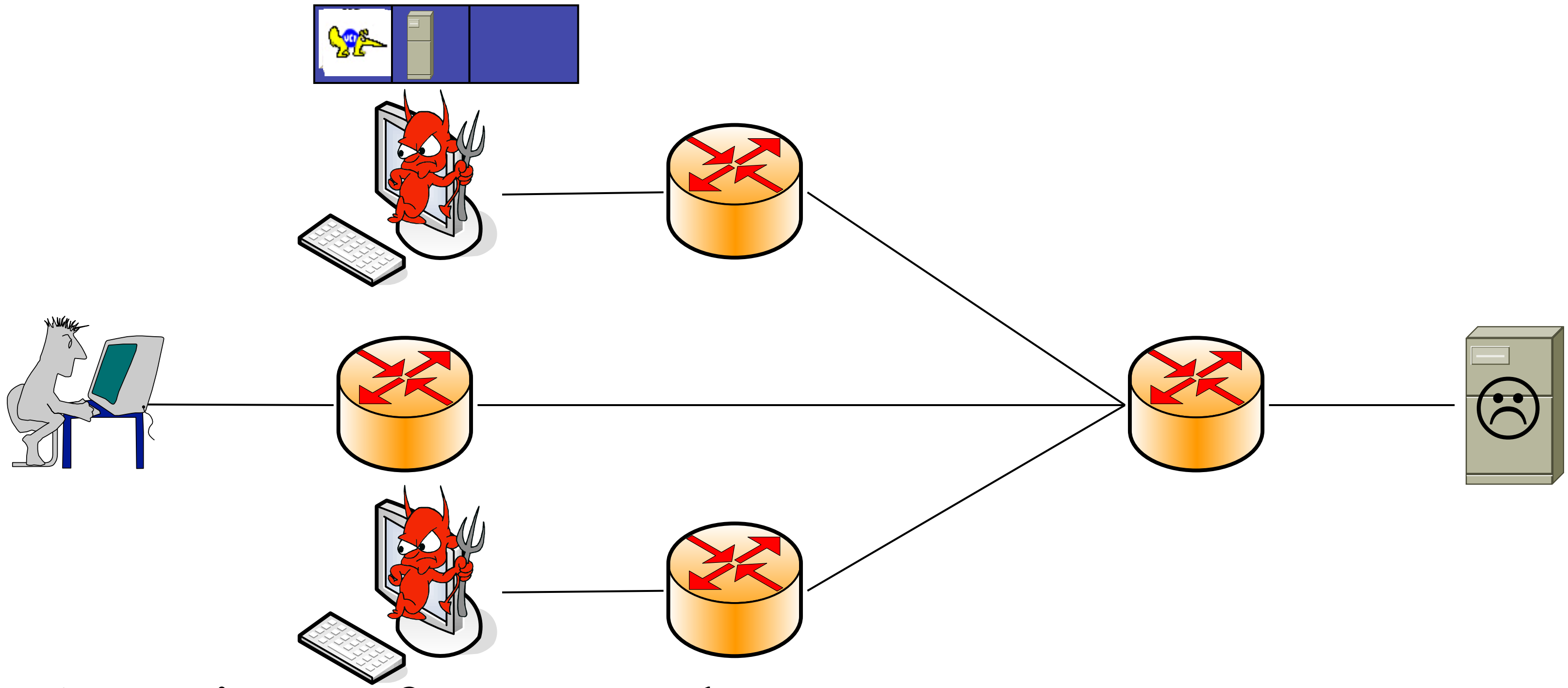
David Wetherall

Univ. of Washington and Intel Research

Outline

- Motivation
  - Source address spoofing weakens DoS defense
- Passport
  - Design
  - Evaluation
  - Applications
- Conclusion
  - Making source addresses trustworthy is feasible and advantageous

Spoofing weakens DoS defense

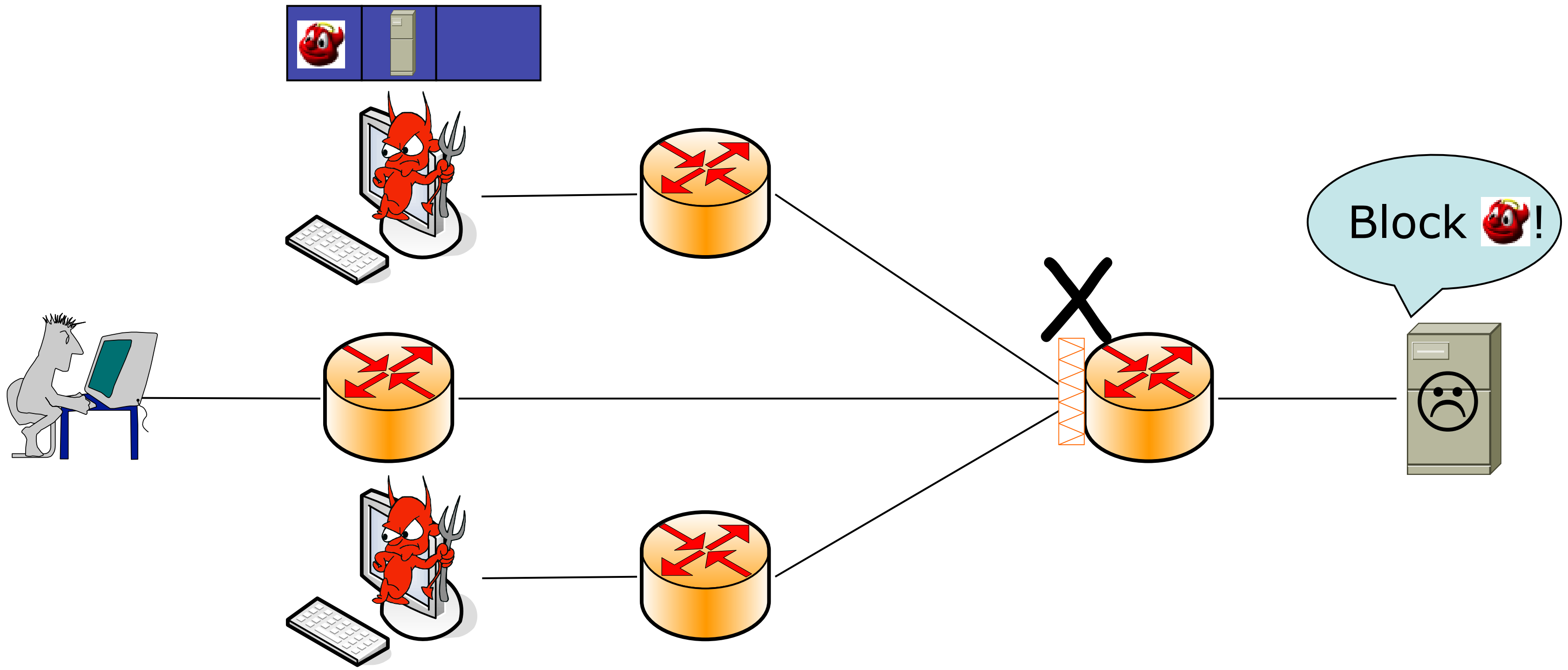


- A variety of proposals
  - Filter-based: AITF, Pushback, CenterTrack, dFence …
  - Capability-based: SIFF, TVA …
  - Overlay-based: SOS, Mayday, i3, Spread Spectrum, Phalanx…

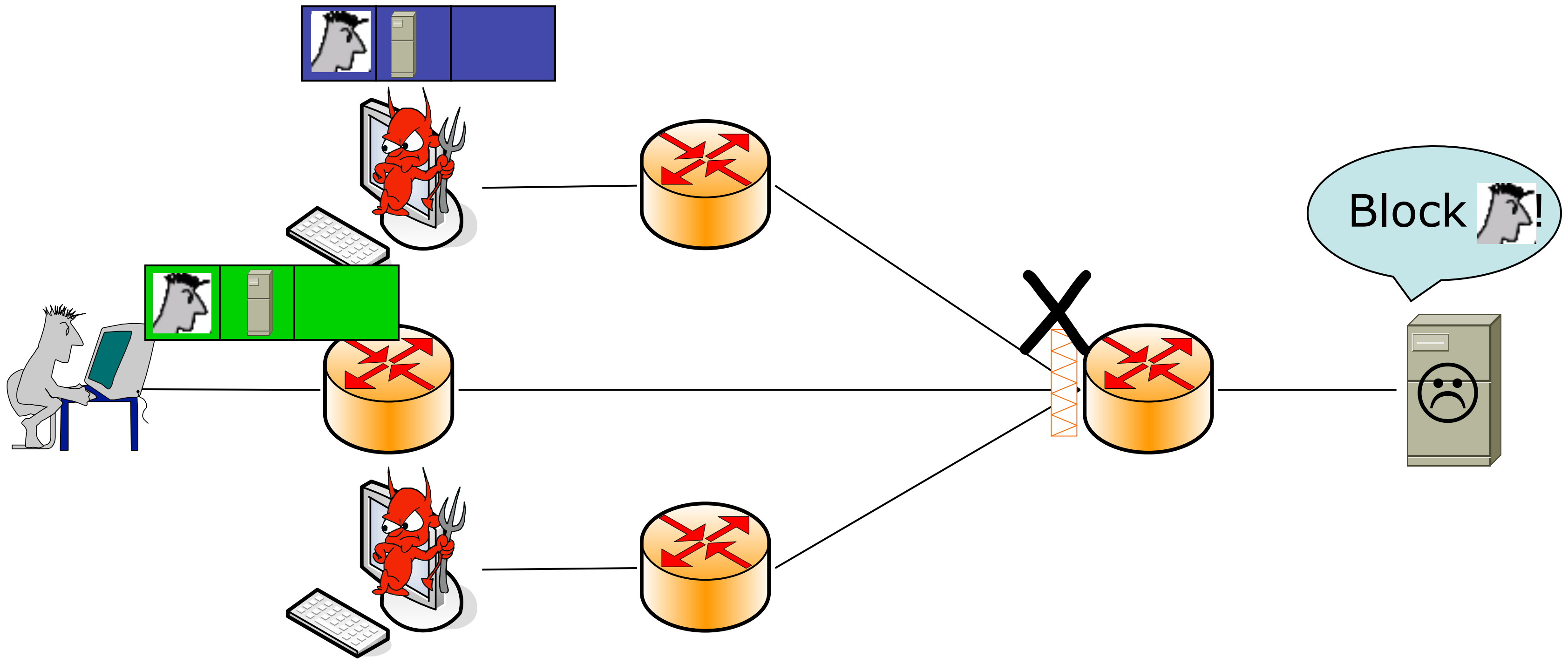
Spoofing weakens DoS defense

- Case Study I: automated filtering
  - Impersonate other hosts
  - Evade filters
  - Reflector attacks
- Case study II: Pushback
  - Hop-by-hop, not directly to source
  - Collateral damage at a legacy router
- Case study III: capability-based systems
  - Can't achieve bandwidth fairness on the request channel

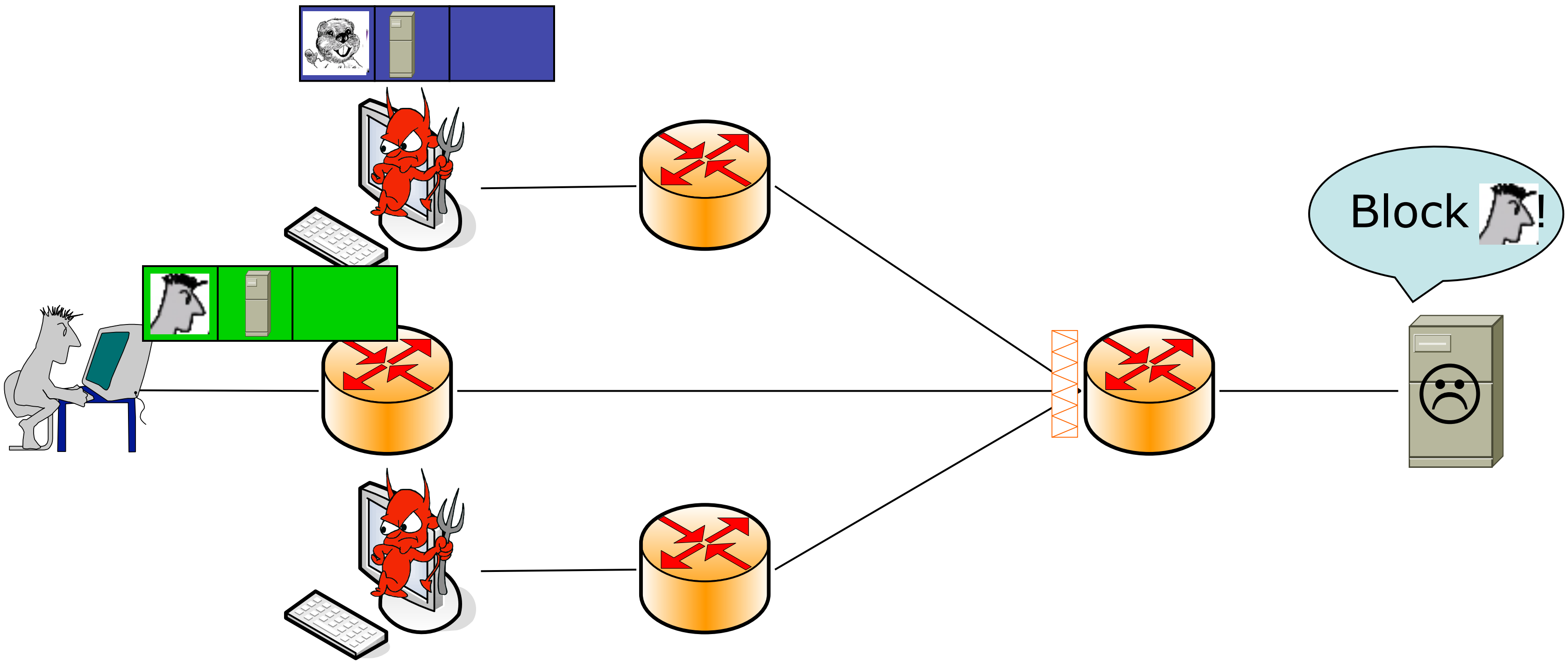
Case study I: automated filtering



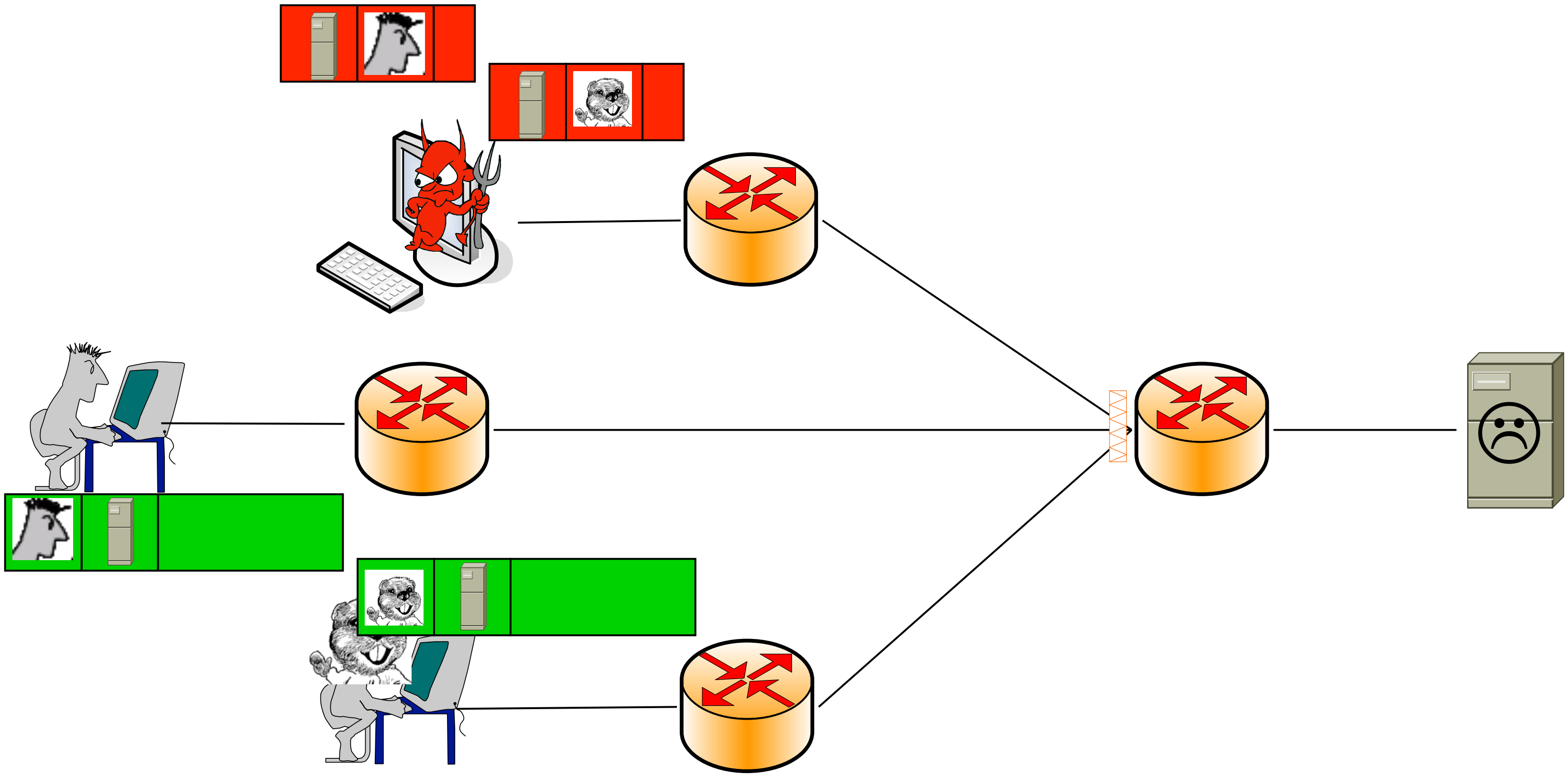
Attackers can impersonate legitimate hosts



Attackers can evade filters



Attackers can launch reflector attacks



- Amplify attack bandwidth
- In early 2006, DNS reflector attacks flooded victims with up to 5Gbps traffic

Spoofing weakens DoS defense

- Case study I: automated filtering
  - Impersonate other hosts
  - Evade filters
  - Reflector attacks

- Case study II: Pushback
  - Hop-by-hop, not directly to source
  - Collateral damage at a legacy router

- Case study III: capability-based systems
  - Can't achieve bandwidth fairness on the request channel

Two steps to combat DoS

1. Make source addresses trustworthy (this talk)
   - Goal of ingress filtering, *Best Current Practice*
2. Build defense systems with trustworthy source addresses
   - “We assume source address spoofing attacks are prevented using systems such as Passport…”
     - Filter-based, capability-based, overlay-based…

Main challenges

	Secure	Lightweight	Adoptable
Ingress filtering	✘	✓	✘
Digital signature	✓	✘	✓
Passport	✓	✓	✓

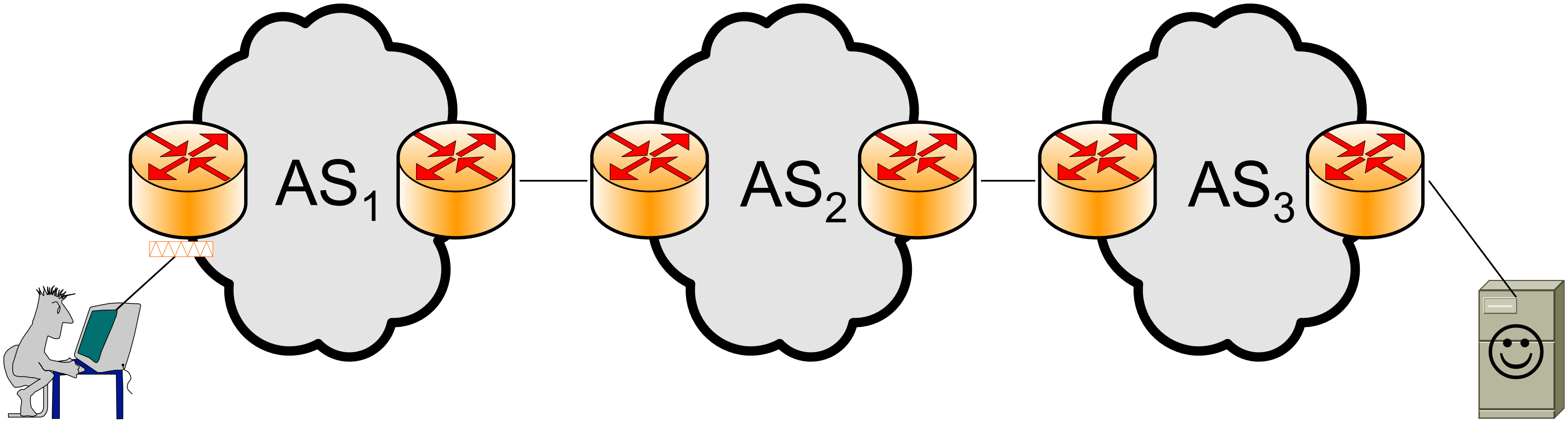
- Ingress filtering
  - One weak link allows spoofing
    - Spoofer shows ~20% of the Internet can spoof
    - Hubble
  - An early adopter can't protect its own address space
- Digital signature
  - PKI, time-consuming to stamp and verify, large header overhead

Passport mechanisms

- Symmetric key cryptography
  – Efficient, secure
- Use routing to distribute keys
  – Bootstrap, efficient, simple
- AS-level (autonomous system) fate sharing
  – Scalable, incentive compatible

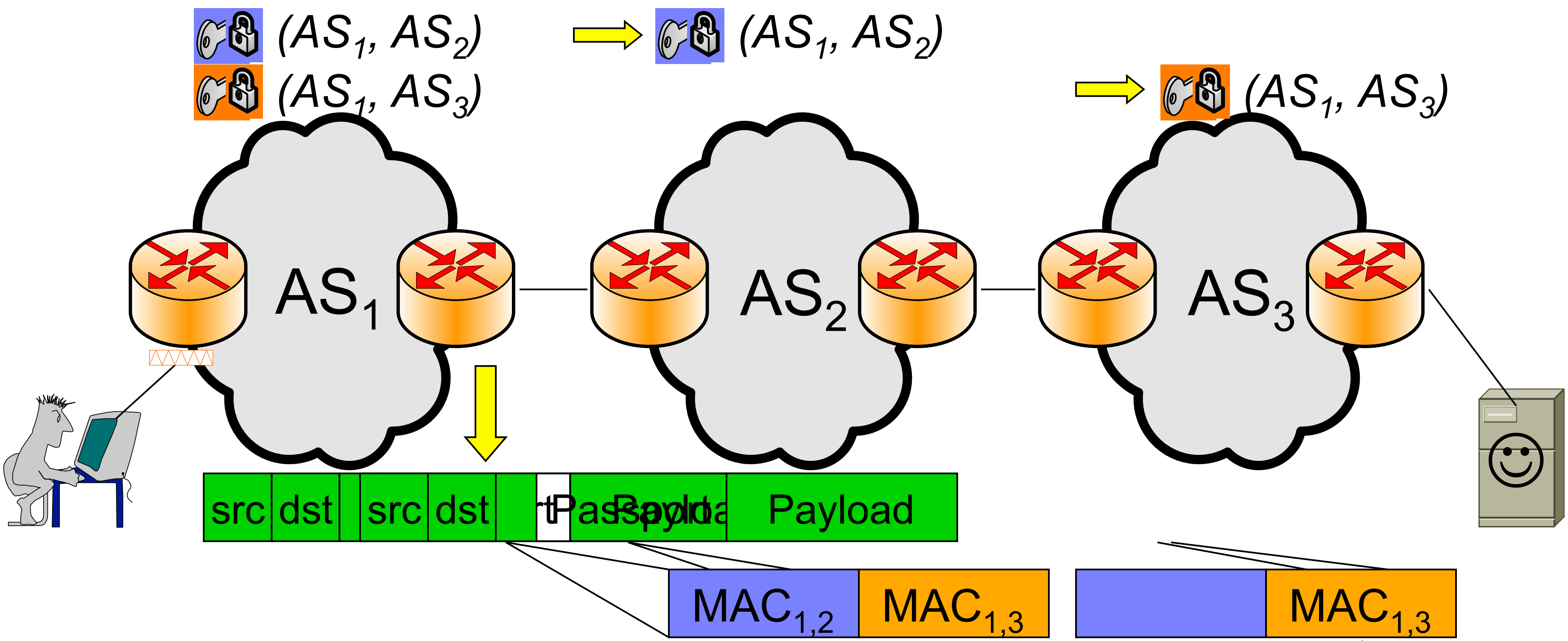
Please refer to our paper for more details.

AS-level fate sharing



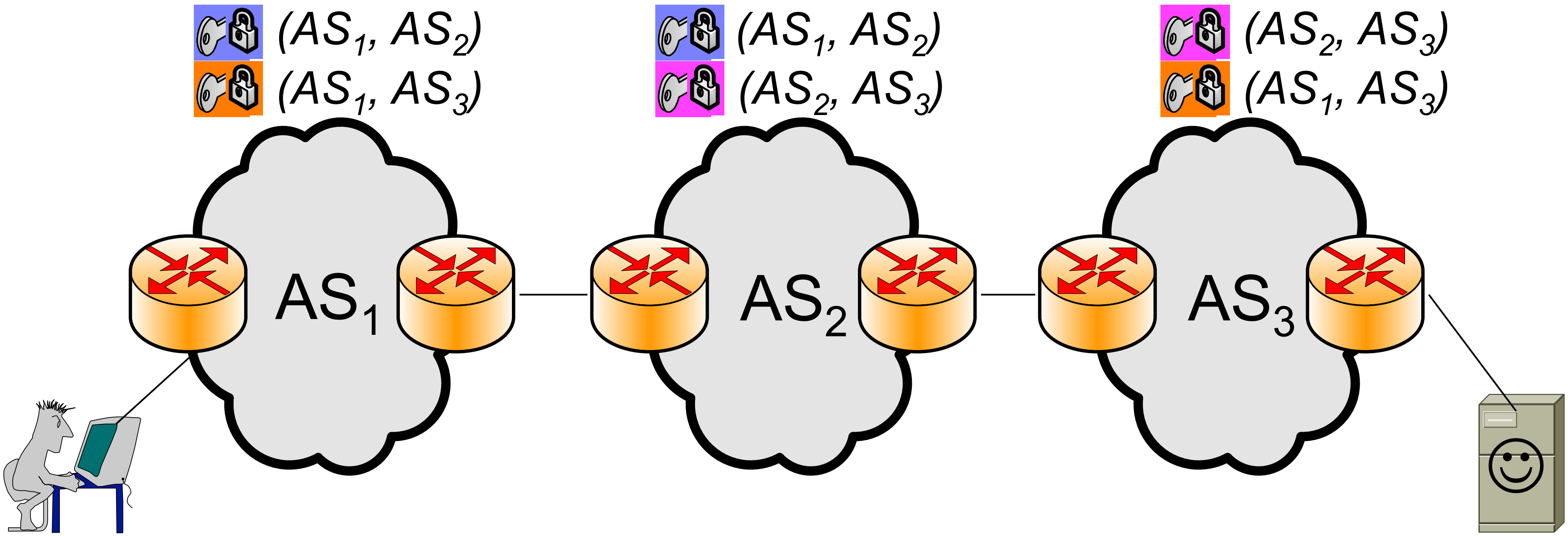
- Passport prevents AS-level spoofing
  - One AS cannot spoof other ASes' addresses
- An AS is responsible to prevent internal spoofing
  - Ingress filters
  - An irresponsible AS only harms its own hosts
- Scalable, incentive compatible

Efficient symmetric key cryptography



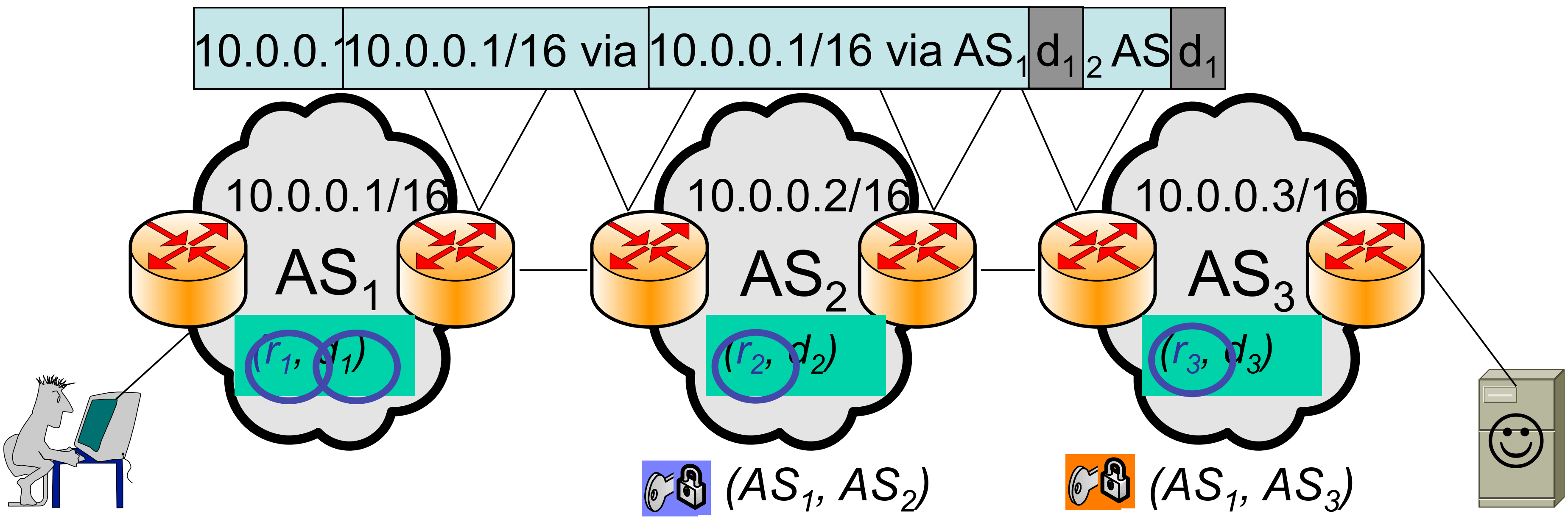
- Source border router stamps Message Authentication Code (MACs) into a Passport header
  - Obtain AS paths from BGP
- Other border routers verify corresponding MACs
  - Demote or discard invalid Passports

How to obtain shared secret keys



- Problems
  - Bootstrap: chicken-and-egg
  - Efficiency: must obtain shared keys with ~30K ASes

A Diffie-Hellman key exchange via routing

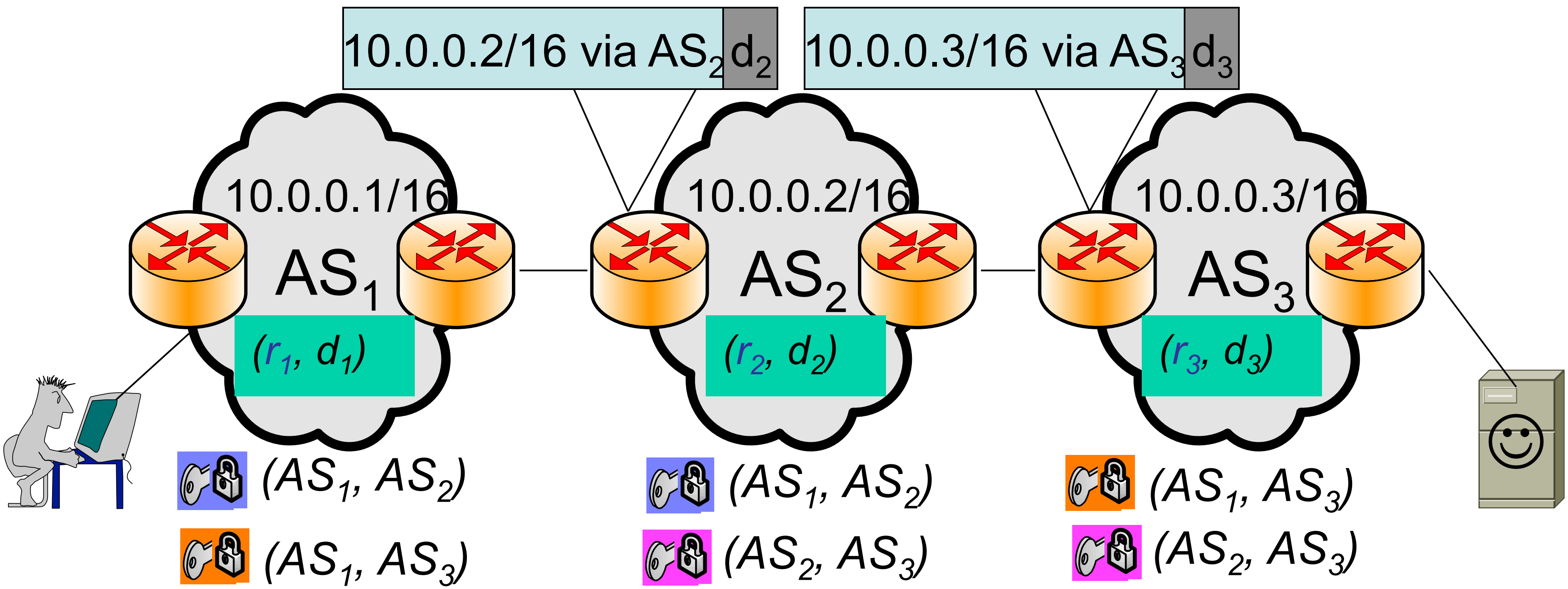


$d_i = g^{r_i} \bmod p$ *g, p* are system-wide parameters

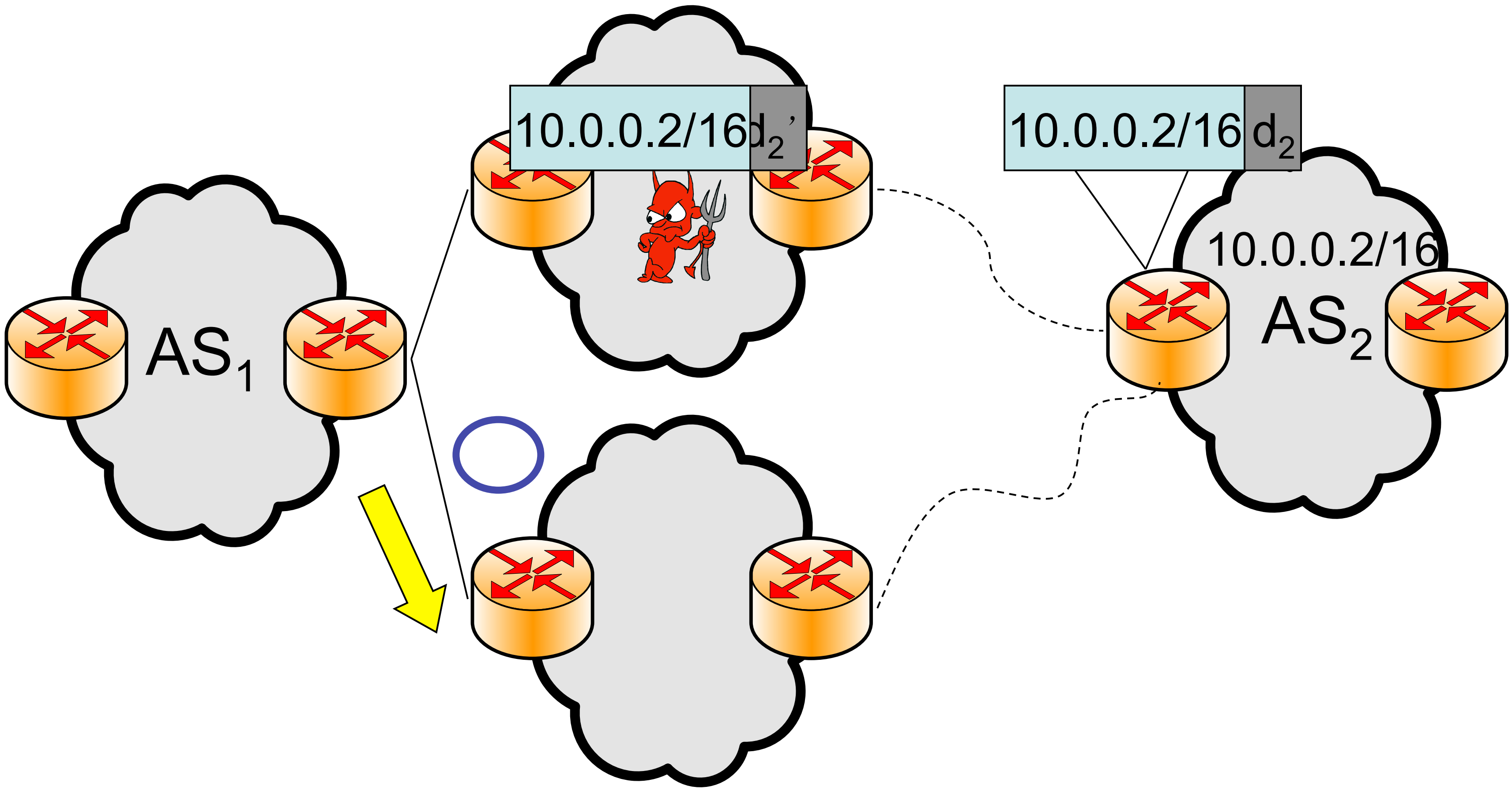
$(AS_1, AS_2) = (d_1)^{r_2} \bmod p = (d_2)^{r_1} \bmod p$

$(AS_1, AS_3) = (d_1)^{r_3} \bmod p = (d_3)^{r_1} \bmod p$

A Diffie-Hellman key exchange via routing



Secure key distribution via routing



- Accept *d* received from the next hop AS
- Secure routing → secure source authentication

Routing helps a lot

- Bootstrap and secure key exchange
- Efficient
  - Send one announcement, establish all pair keys
- DoS-resistant
  - High priority forwarding

Other design issues

- Incremental deployable
  1. Transparent to hosts
  2. Inter-operate with legacy ASes
  3. Downstream legacy ASes can also benefit
  - BGP optional and transitive attributes
  - A shim layer
  - Encapsulation
- Secure under host, monitor, and router attackers
  - Seamless rekey
  - Resistant to sniff-and-replay: bound to a path
- Handle path changes
  - Demote at the intermediate ASes

Evaluation

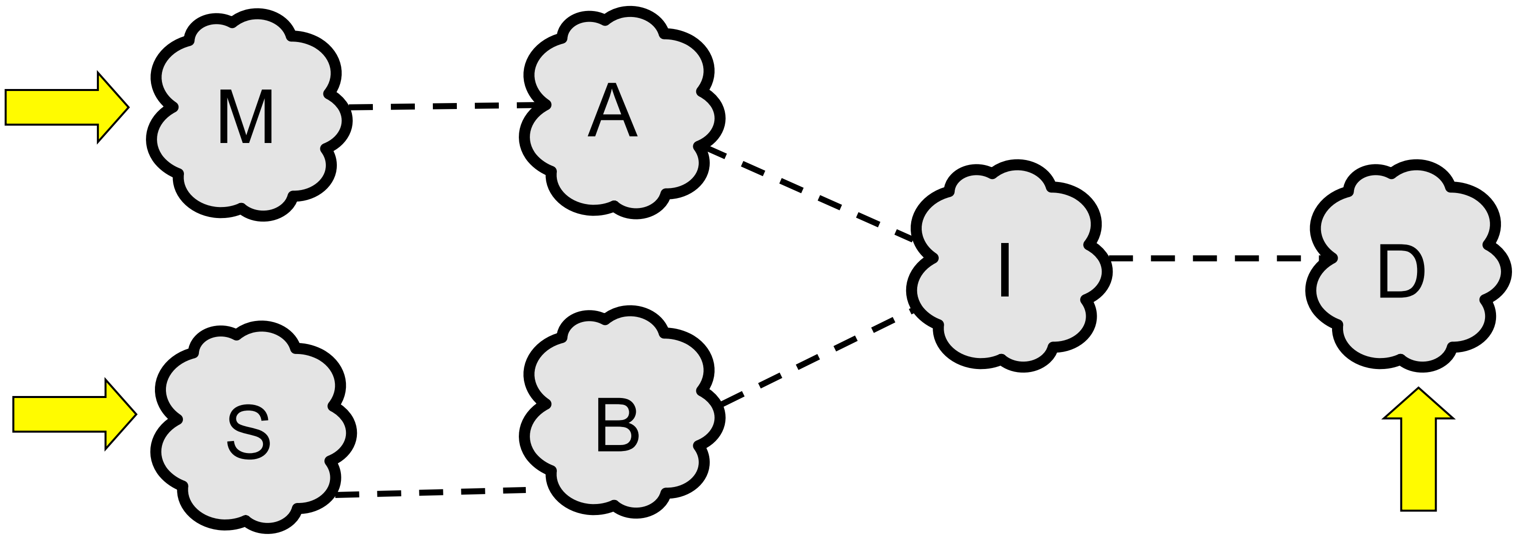
- Challenges: secure, lightweight, and adoptable

- Lightweight
  - Linux-based implementation (Click and XORP)
    - Throughput, processing, header, and memory overhead
    - *Plausible for multi-gigabit implementation*

- Adoptable
  - Model adoptability
    - "Modeling Adoptability of Secure BGP Protocols," Chan et al., SIGCOMM 2006

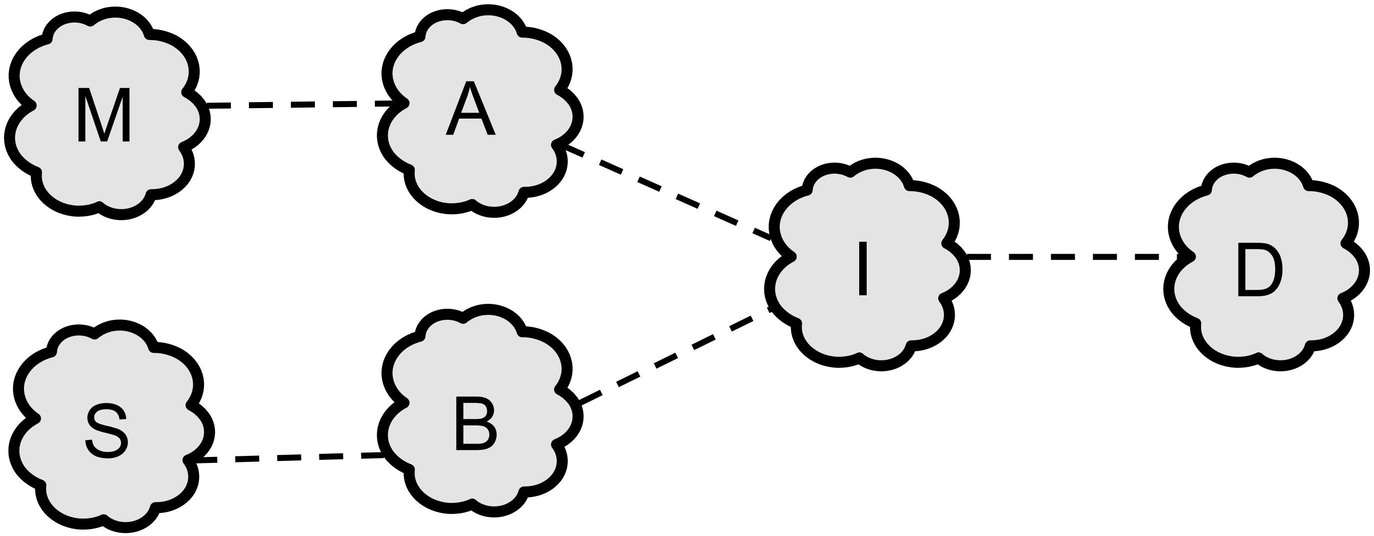
- Security analysis

The adoptability model



- F: the immediate security benefit
- A security indicator: $E(M,D) = \begin{cases} 1, & \text{M cannot spoof S} \\ 0, & \text{M can spoof S} \end{cases}$
- $F = \sum_D w_D \sum_M P(M)\, E(M,D) \,/\, \sum_D w_D$

Simulate the adoption dynamics

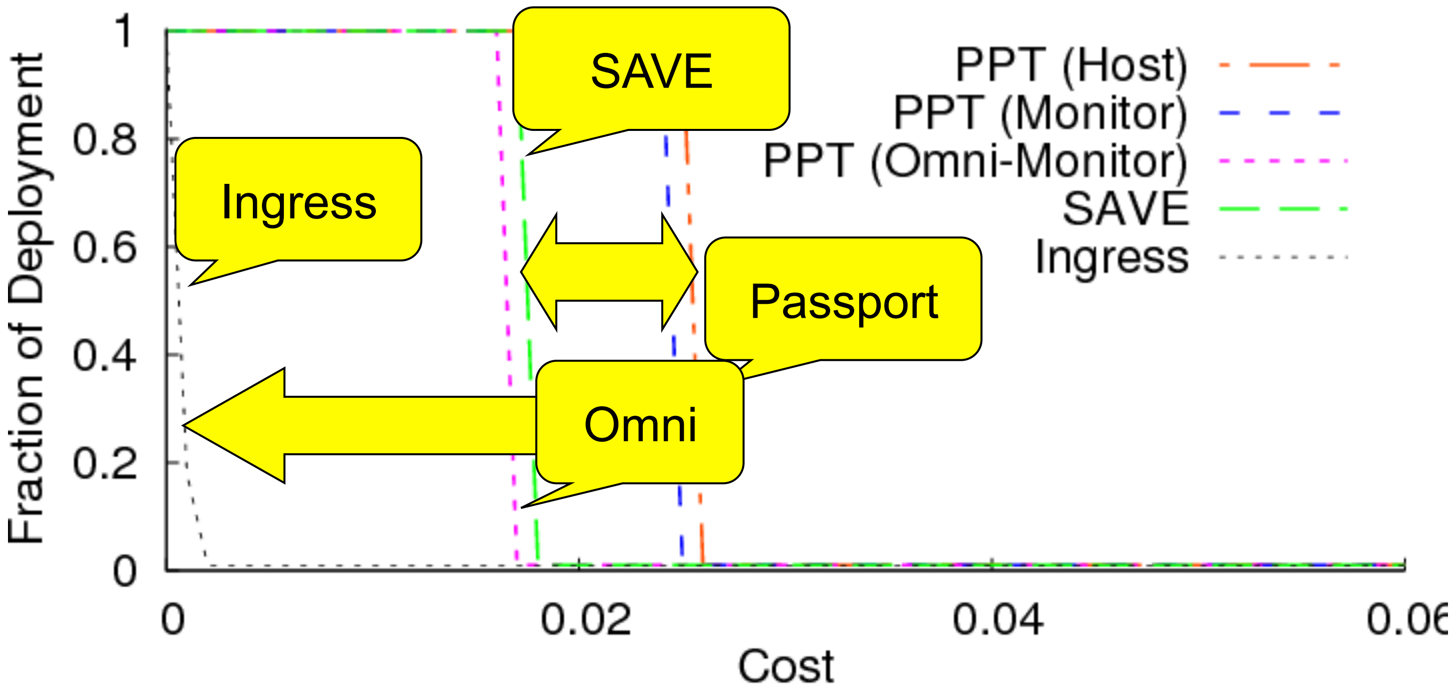


- Δ F > c, S adopts an anti-spoofing mechanism
- Network effect
- Metric: the critical threshold c_{th}
- Higher c_{th}, more adoptable

Comparing different schemes

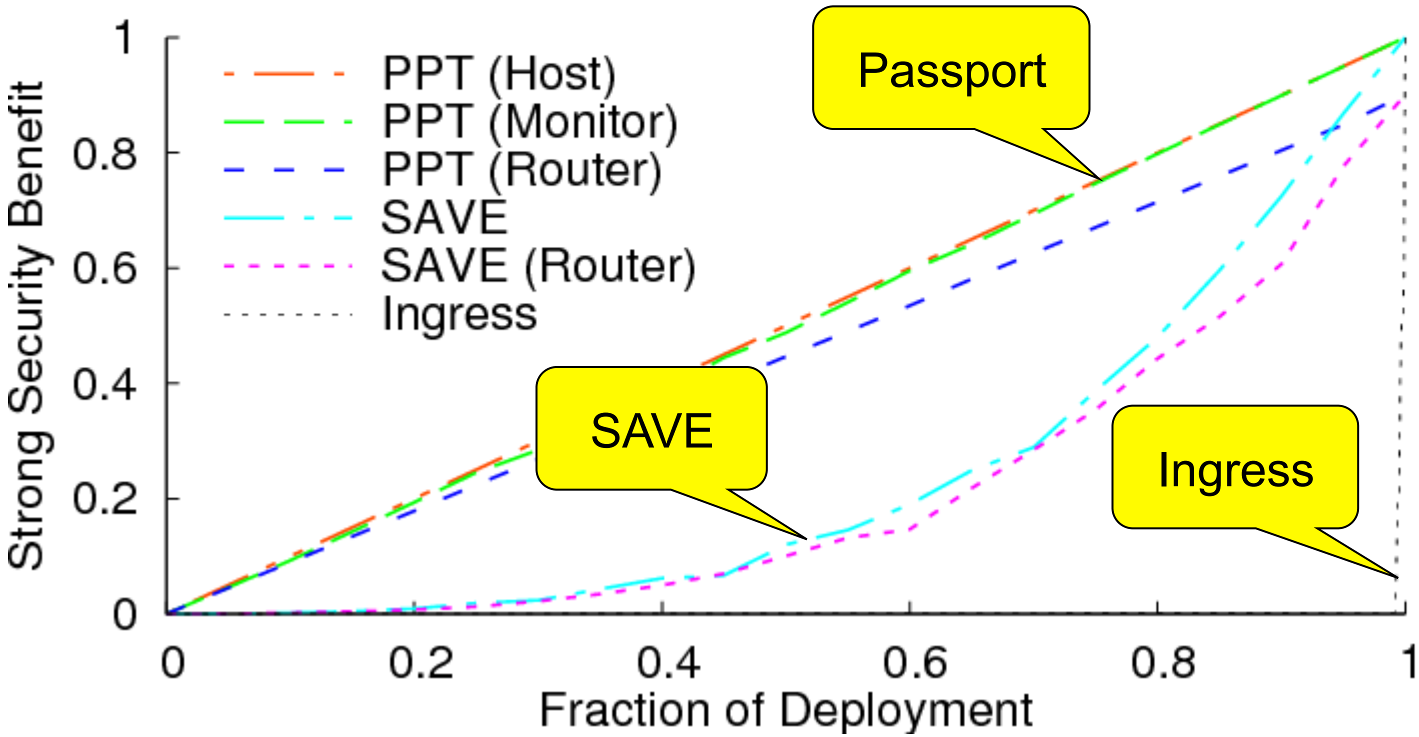
- Passport
- Ingress filtering
- SAVE: a protocol to install route-based filters
  - A router maintains a source address table
  - Best non-cryptographic proposal

Passport is more adoptable than alternatives



- w_D : uniform traffic distribution
- P(M): uniform attacker distribution
- Host, Monitor, and Router attackers

Passport provides stronger security benefit



- F measures probabilistic guarantee
  - $F = \sum_D w_D \sum_M P(M)\, E(M,D) / \sum_D w_D$
- Strong security benefit: fraction of Ds no attacker can spoof S
  - $F_s = \sum_D w_D \sum_M P(M)\, E(M,D) / \sum_D w_D$, s.t. $\sum_M P(M)\, E(M,D) = 1$

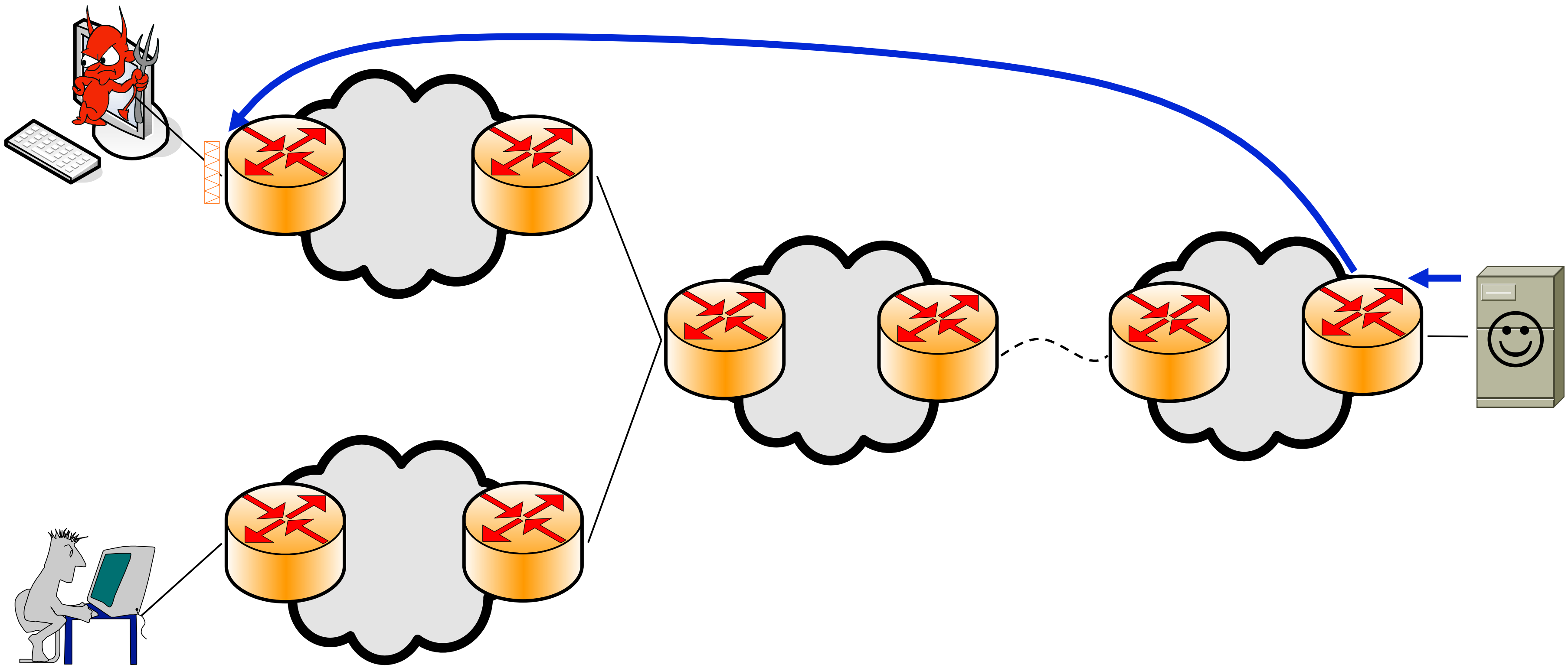
Comparison with related work

- Non-cryptographic approaches: ingress filtering, route-based filtering
  - Less secure and adoptable
- Digital signatures
  - Heavyweight
  - ~2 orders of magnitude slower
- Challenge-response
  - Reflector attacks
  - First-packet attacks
- Path marking: traceback, path identifiers
  - Post-mortem
  - Path prefix spoofing

Applications of Passport

- Prevent reflector attacks
- Strengthen capability-based DoS defense systems
  - Bandwidth fairness on the request channel
- Secure automated filtering systems
- Others
  - Resource allocation
  - Address-based authentication
  - Forensic analysis
  - …

Passport facilitates secure and scalable filtering



- Locate attack sources using source addresses
- Filter based on source addresses

Conclusion

- Passport: trustworthy source addresses
  - Secure, lightweight, adoptable, and incrementally deployable
  - Symmetric key cryptography, use routing to distribute keys

- Applications
  - Prevent reflector attacks
  - Build other DoS defense systems
    - Work-in-progress: filtering, capability-base

More efficient than public key signatures

Passport

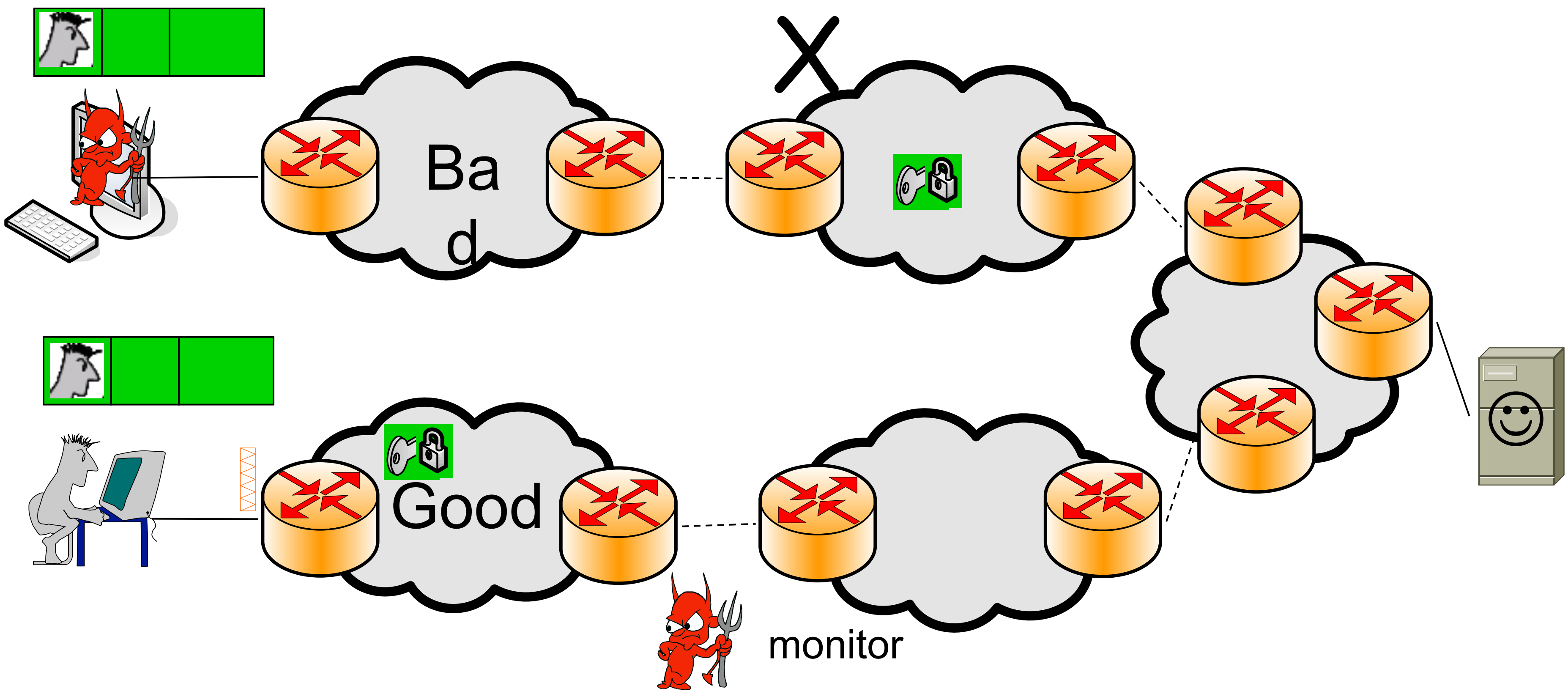
	Operation	Time		
		2-hop	4-hop	8-hop
Per Packet	Passport Stamping	655 ns	1493 ns	3190 ns
	Passport Verification	578 ns	618 ns	631 ns
Re-key	DH value pair (1024-bit)	5.64 ms		
	Symmetric key (128-bit)	5.64 ms		

Digital signatures

	Security	Sig. Size	Signing	Verification
RSA-512	60-bit	64 bytes	512 us	40 us
RSA-1024	72-bit	128 bytes	2214 us	102 us
DSA-512	65-bit	40 bytes	368 us	443 us
ECDSA-160	78-bit	40 bytes	300 us	1400 us

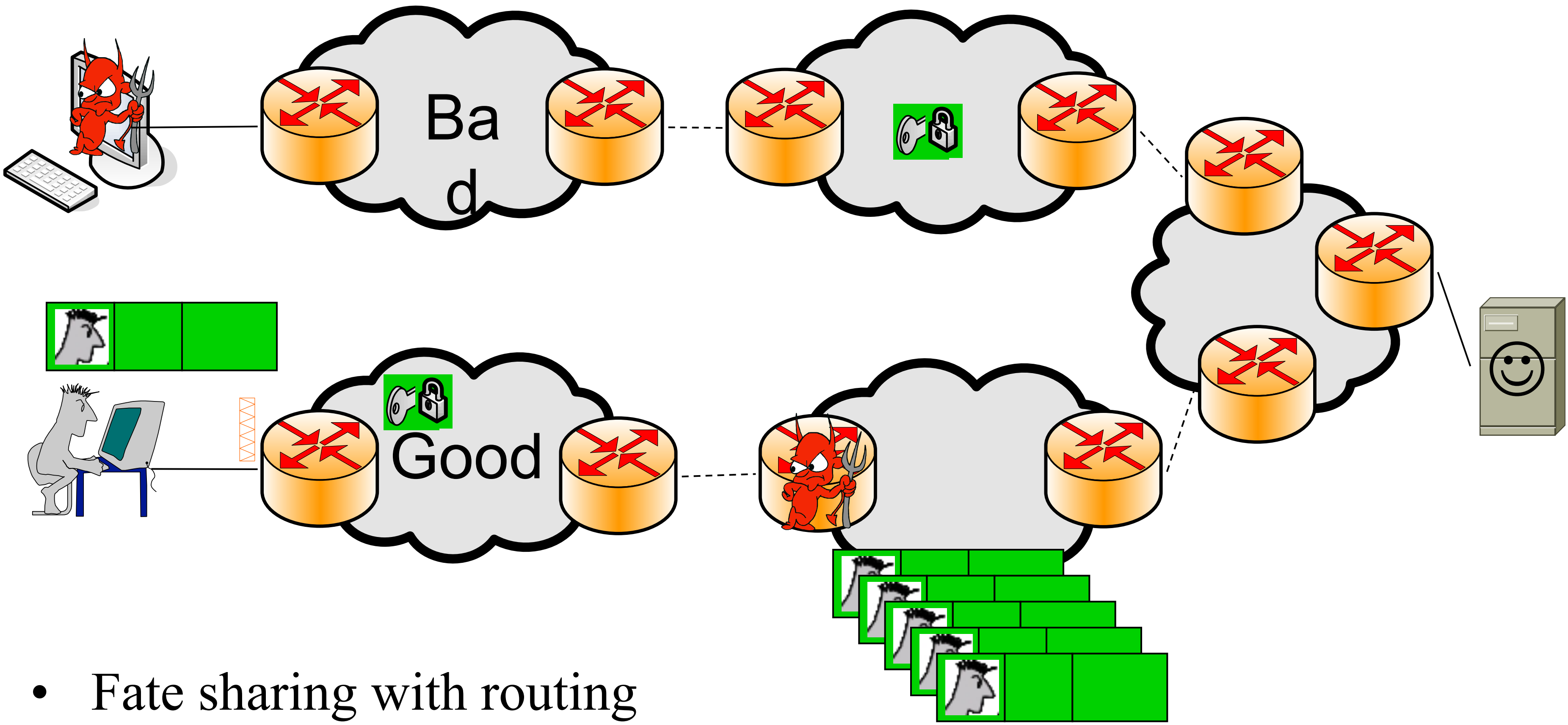
- Differ in magnitude
- Hardware implementation to be faster

Security properties (II)



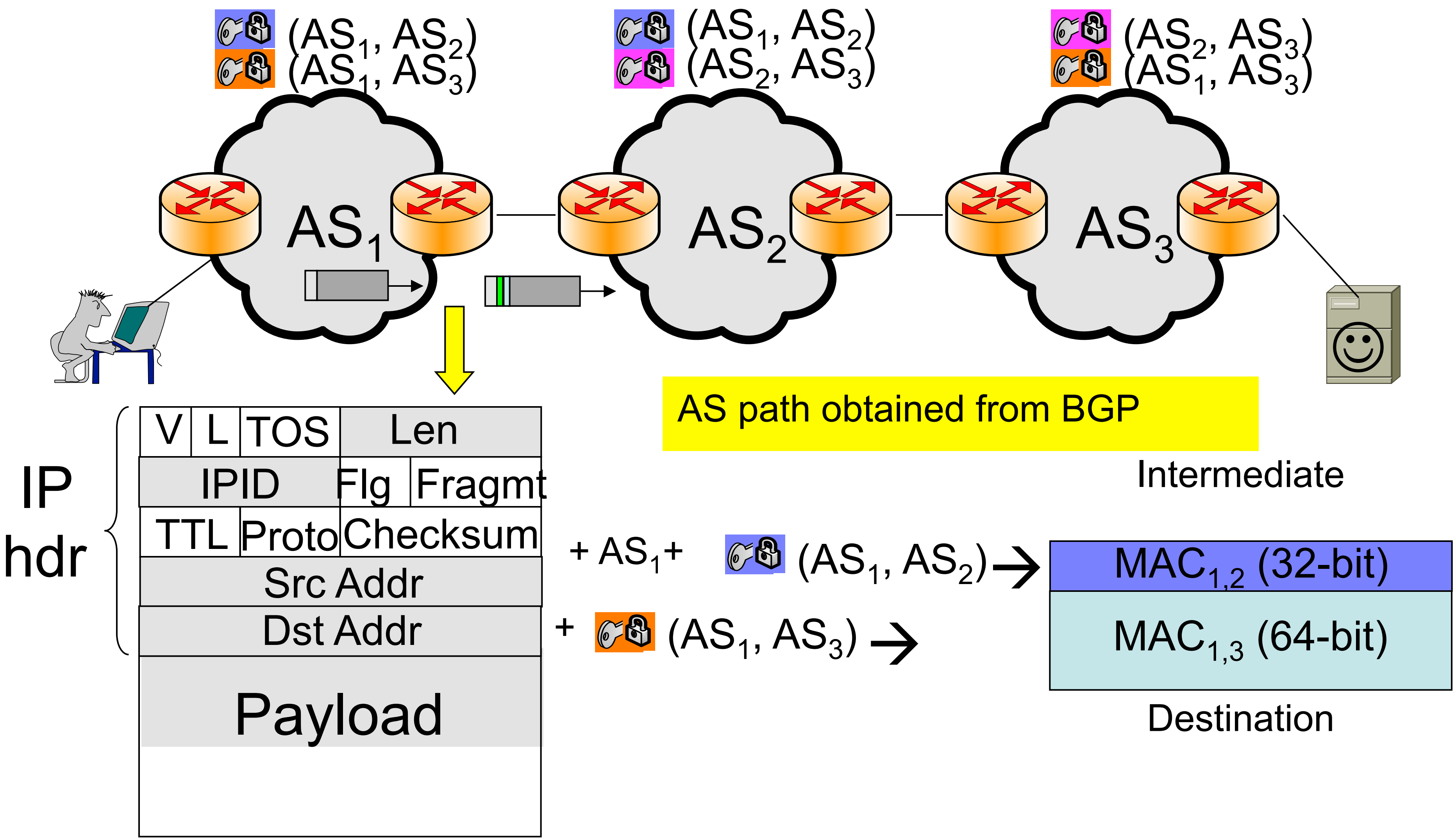
- Resistant to sniff-and-replay attack
  - An intermediate MAC includes path information

Security properties (III)



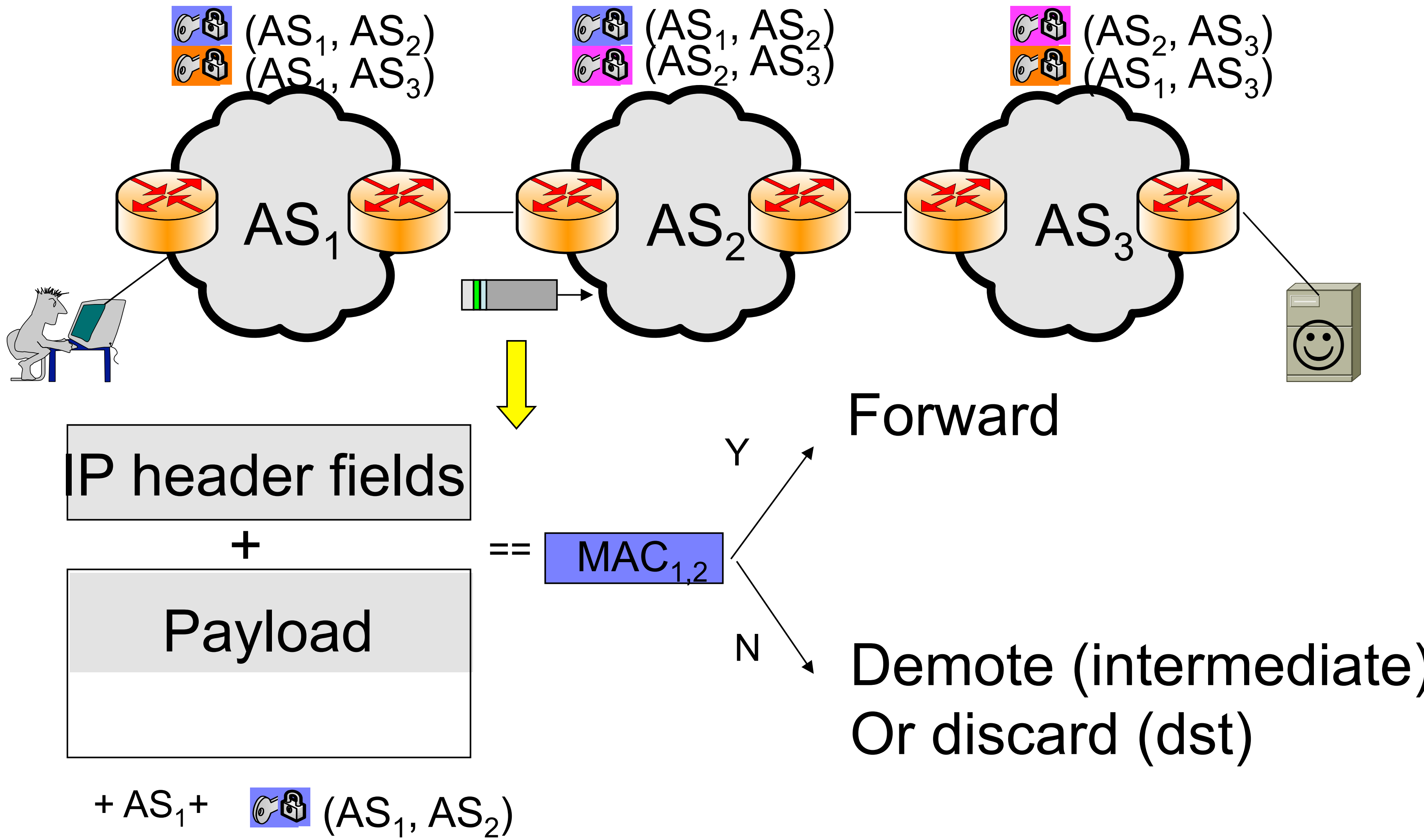
- Fate sharing with routing
  - Switch to a different path
  - Duplicate Passport headers may be detected at a higher cost [SRUTI 06]

Stamping



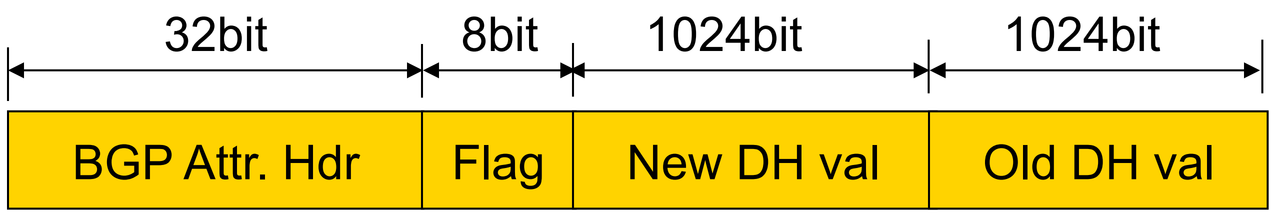
(AS_1, AS_2)
(AS_1, AS_3)
(AS_1, AS_2)
(AS_2, AS_3)
(AS_2, AS_3)
(AS_1, AS_3)
AS_1
AS_2
AS_3
AS path obtained from BGP
IP hdr
V
L
TOS
Len
IPID
Flg
Fragmt
TTL
Proto
Checksum
Src Addr
Dst Addr
Payload
+ AS_1+ (AS_1, AS_2) →
+ (AS_1, AS_3) →
Intermediate
$MAC_{1,2}$ (32-bit)
$MAC_{1,3}$ (64-bit)
Destination

Verification

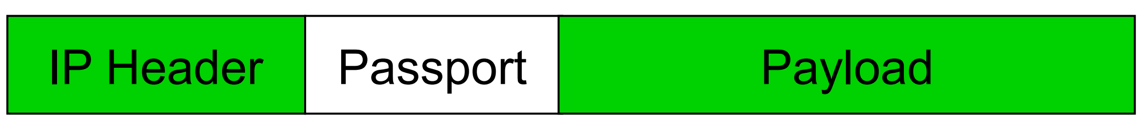


Incremental deployment

- Encapsulate in a BGP optional transitive path attribute

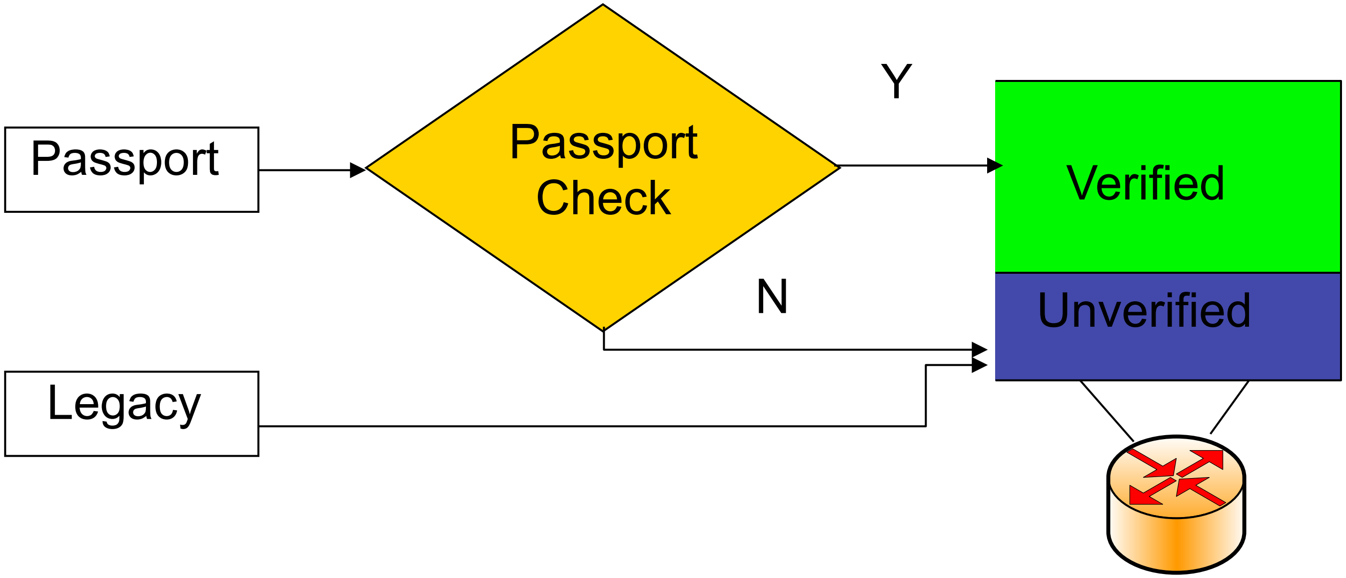


- Packets carry Passport in a shim layer



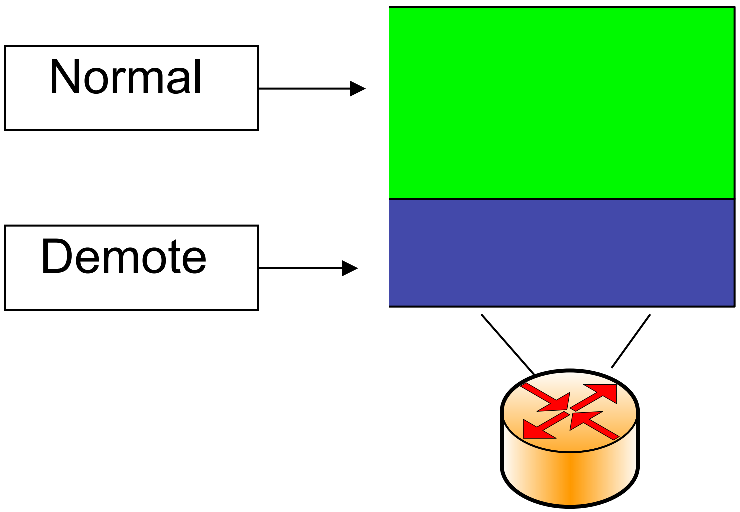
- Hosts need not upgrade
- Downstream non-upgraded ASes can also benefit

Incremental deployment – legacy traffic



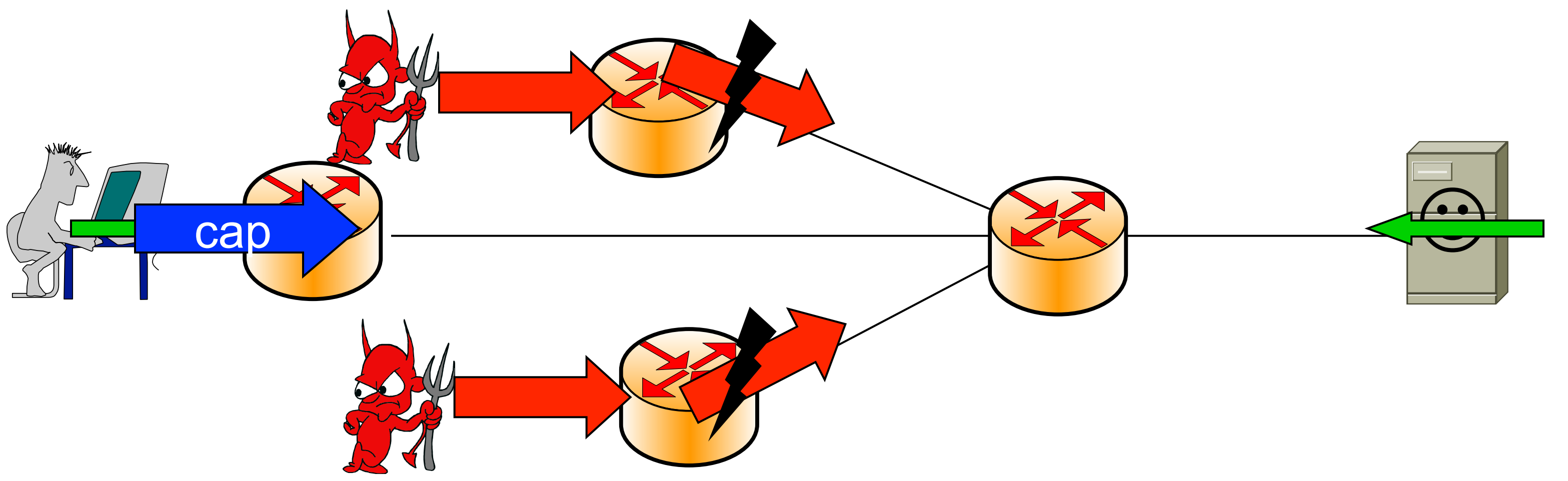
- Legacy traffic is queued separately from Passport verified traffic

Incremental deployment – legacy traffic



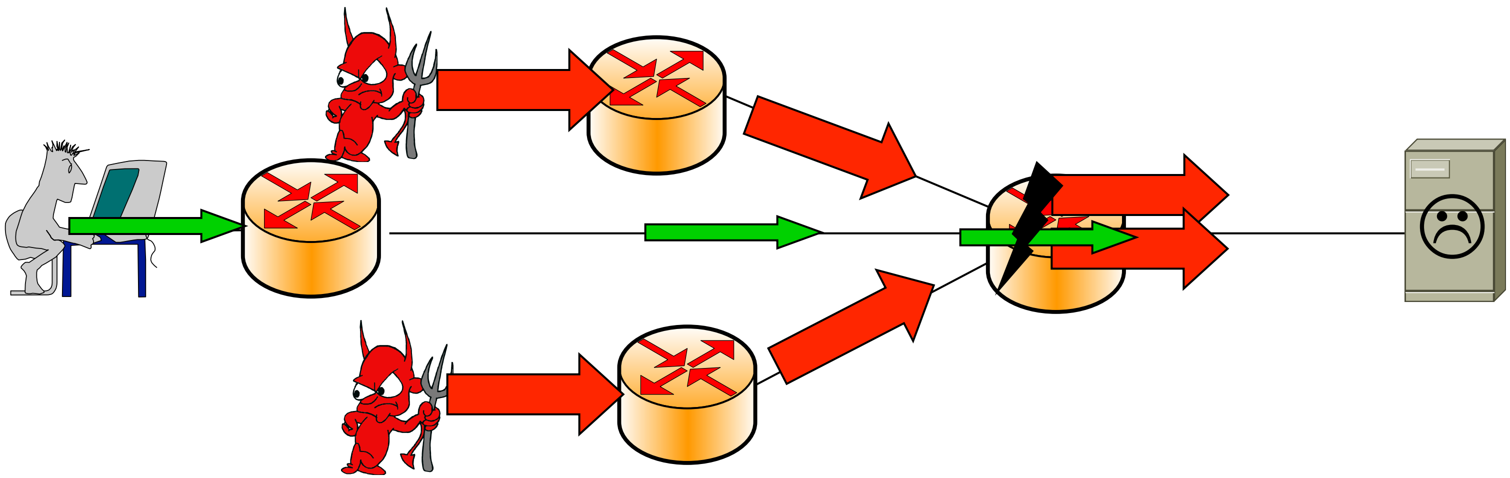
- Non-ungraded AS treats demoted traffic with lower priority
  - Use IP header (DiffServ) to demote

A capability-based architecture TVA



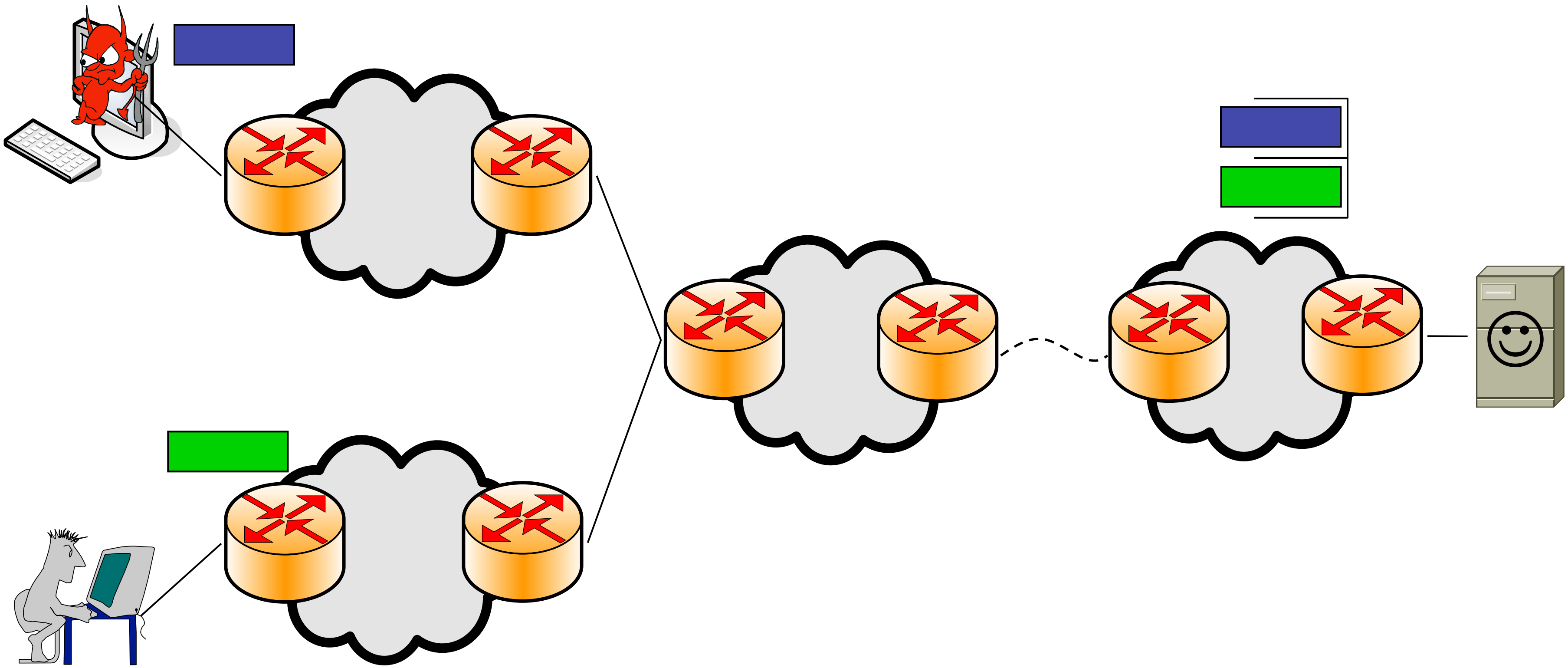
1. Source requests permission to send.
2. Destination authorizes source for limited transfer, e.g, 32KB in 10 secs
   - A capability is the proof of a destination's authorization.
3. Source places capabilities on packets and sends them.
4. Network filters packets based on capabilities. [SIGCOMM 05]

Request channel flooding is the Achilles heel



- Request packets do not carry capabilities
- Denial-of-capabilities

Passport mitigates request flooding attacks

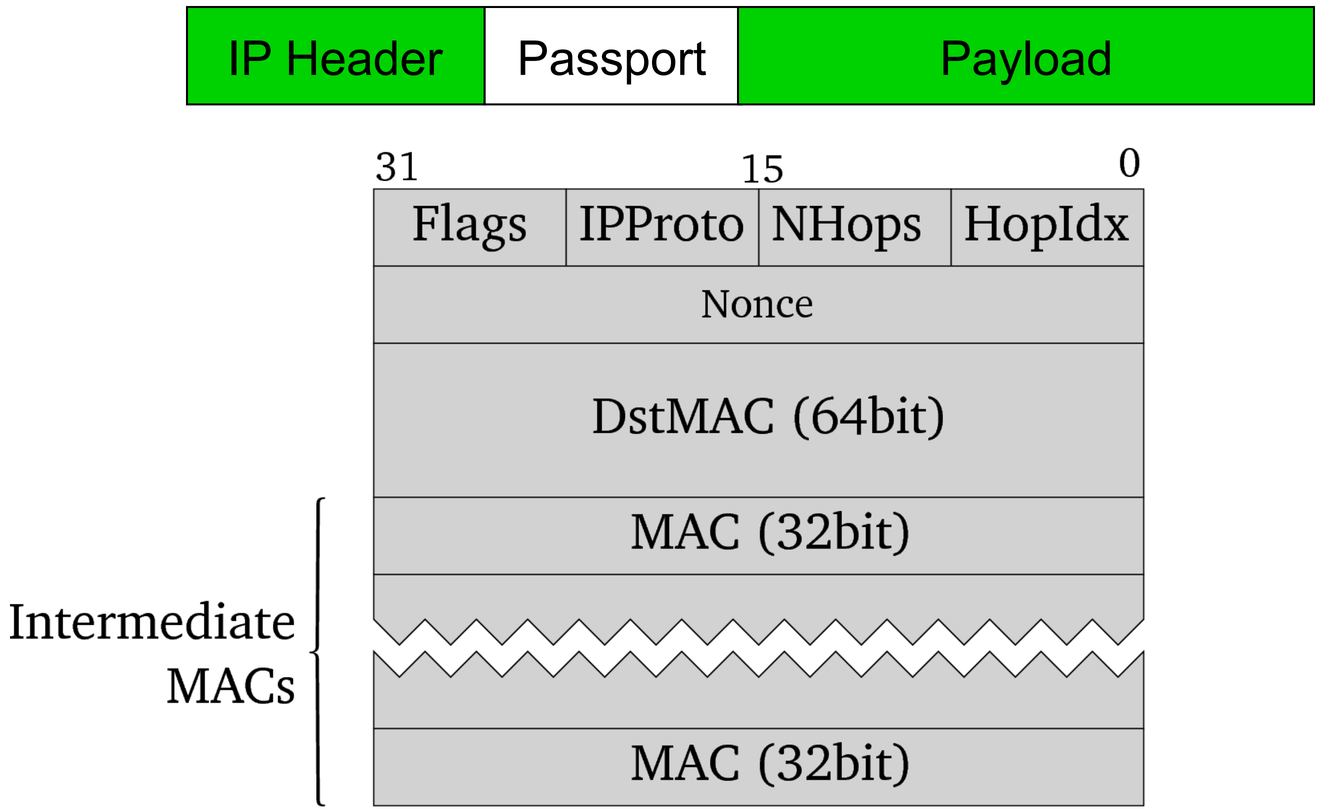


- Request packets can be queued by their source ASes
- Per-network fairness incents improvement on

Other features

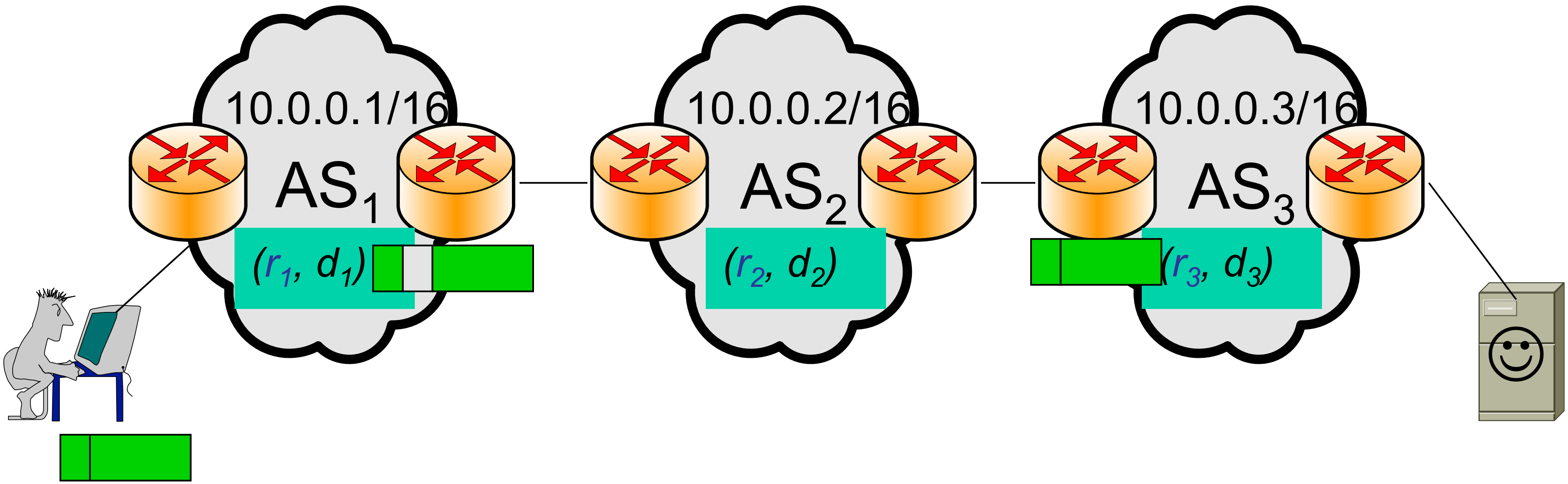
- Incremental deployable
  - Coexist with legacy ASes
  - Hosts need not upgrade
- Seamlessly rekey to improve security
- Downstream non-upgraded ASes can also benefit
  - Demote using IP DiffServ codepoint
  - Encapsulation to inter-operate with non-upgraded destination ASes

Incremental deployment – legacy AS



- Passport header is inserted as a shim layer

Incremental deployment – bump-in-the-wire

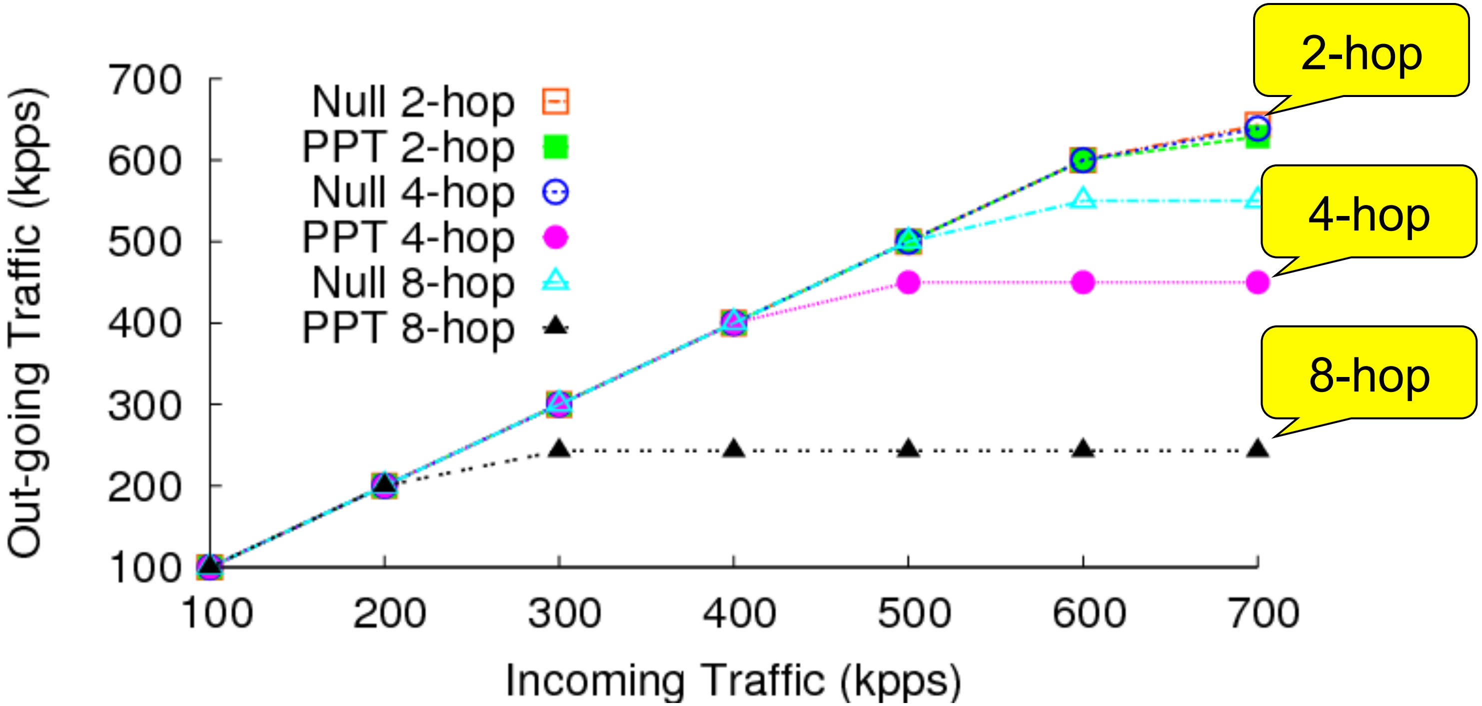


- Hosts need not upgrade

Re-key

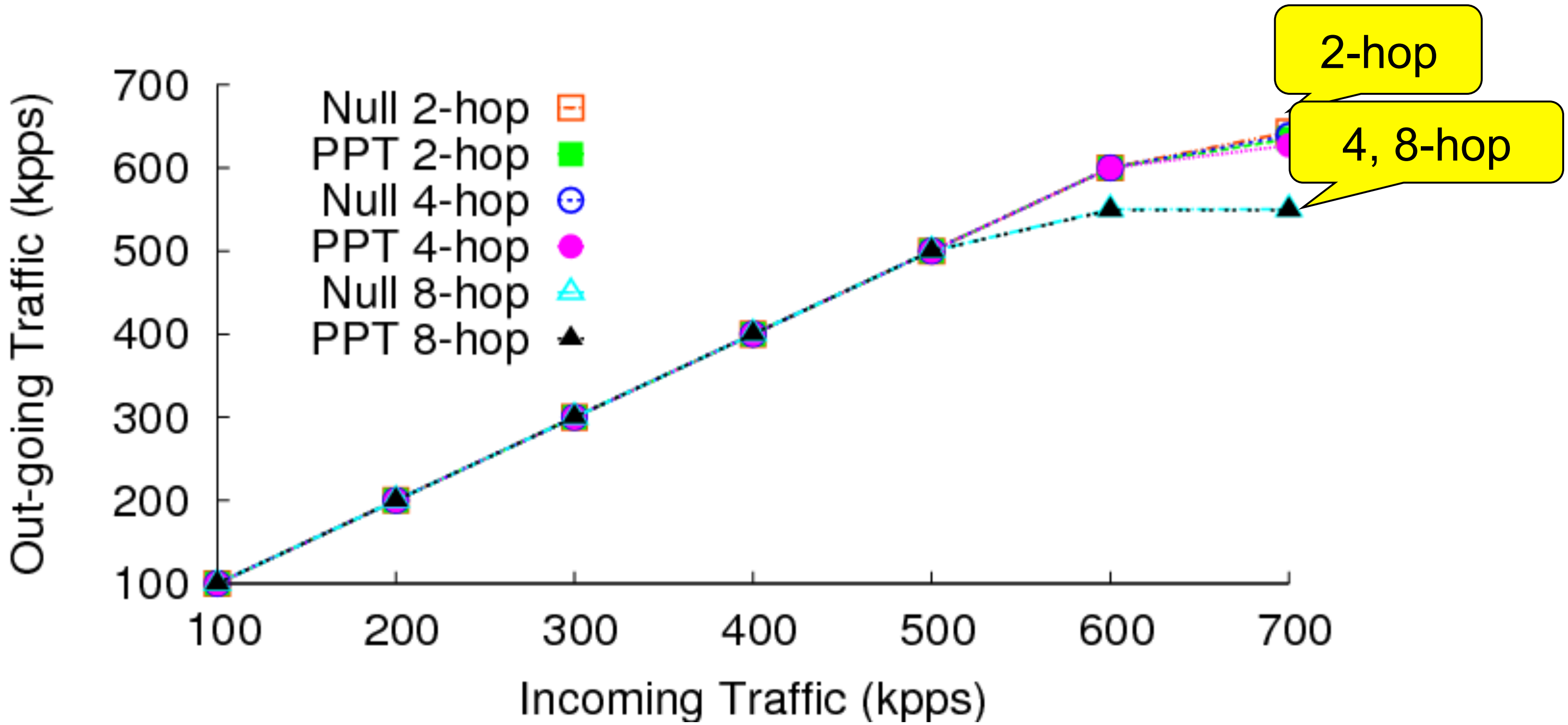
- 1-bit in the Passport header indicates the source AS's Diffie-Hellman value's parity
- 1-bit in each MAC indicates the verifier's Diffie-Hellman value's parity
- 1-bit in BGP attribute's flag field to indicate the parity of a new Diffie-Hellman value

Stamping throughput



- Average AS path length is ~ 4
- Assuming 400-byte average packet size, throughput is 0.9 ~ 2 Gbps
- Only done for traffic an AS originates
- Hardware implementation may achieve 40Gbps AES encryption speed
  - http://www.heliontech.com

Verification throughput

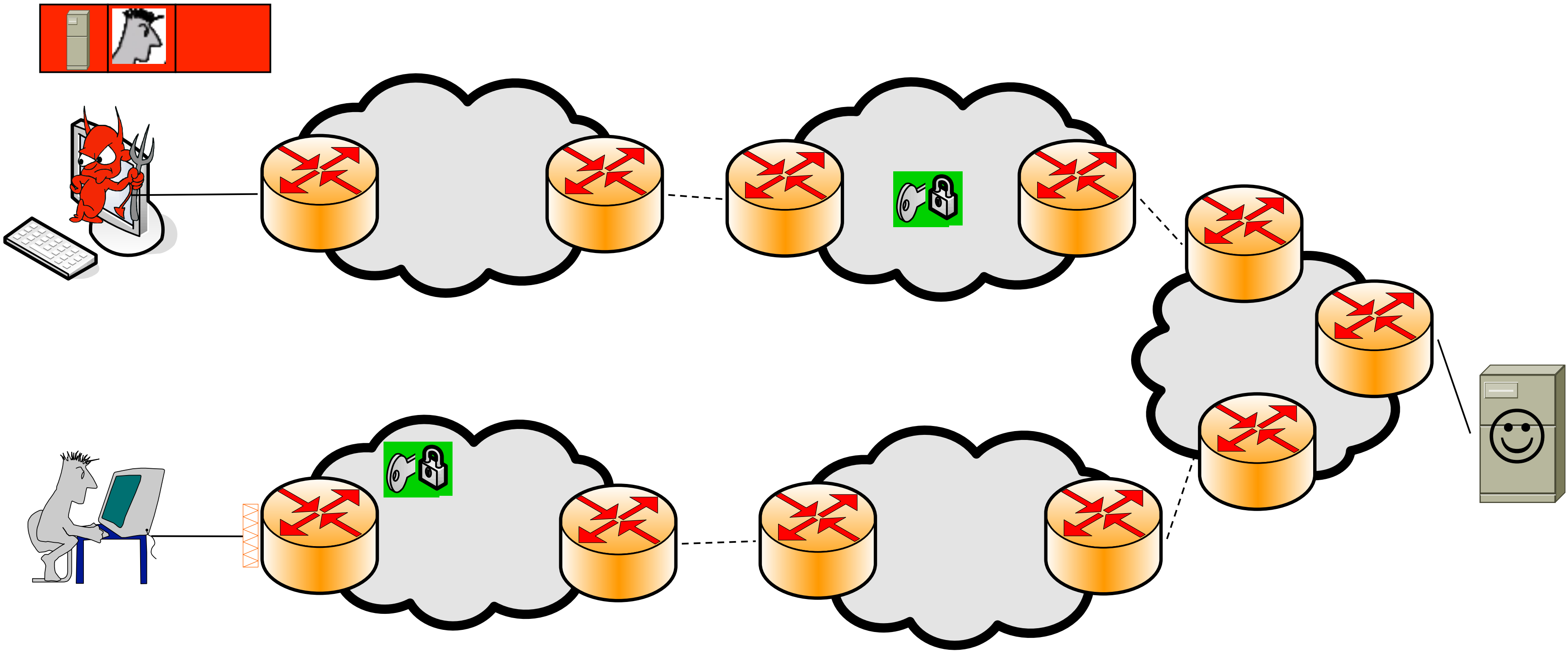


- Assuming 400-byte average packet size, throughput is 2 Gbps

Other overhead

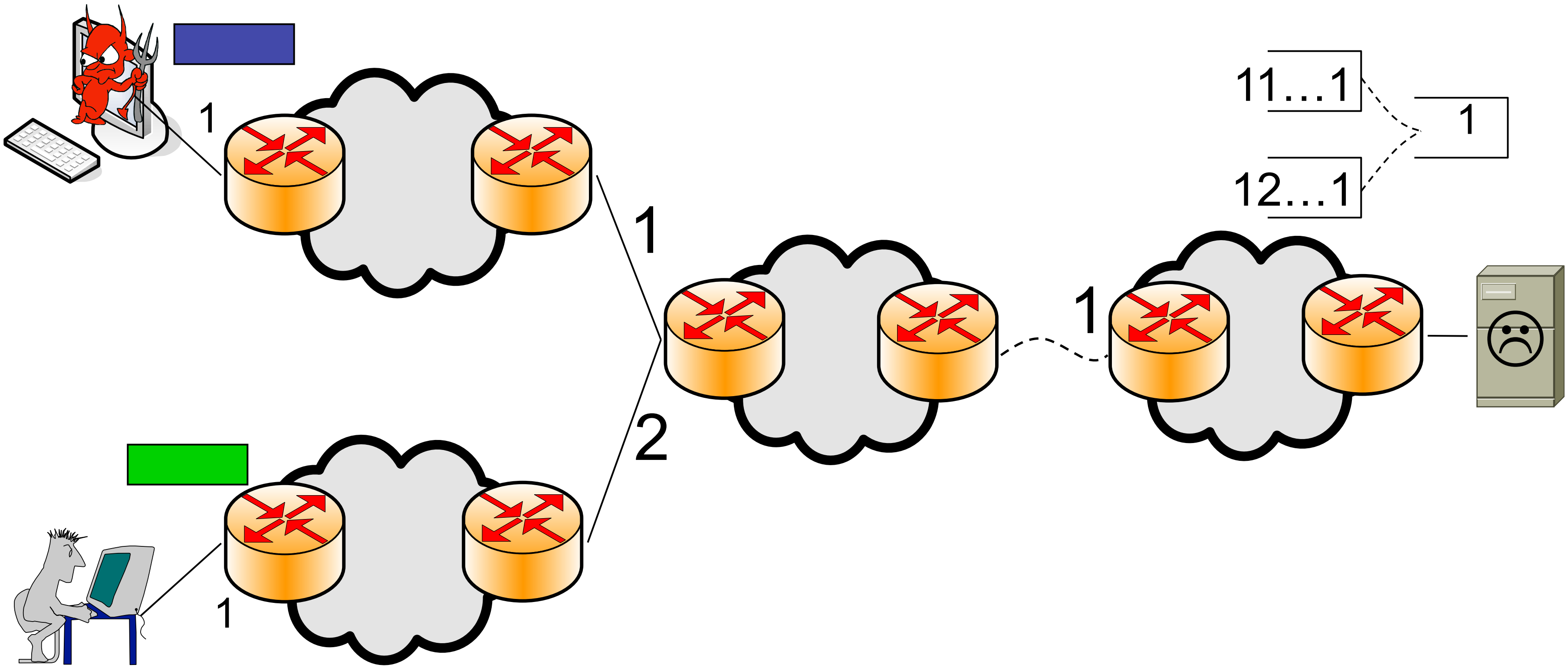
- Header
  - 4-5 AS hops: 24 bytes
  - Four bytes per additional AS hop
  - Can be optimized if combined with capabilities
- Memory
  - 12MB to store 30K shared keys

Mitigate reflector attacks



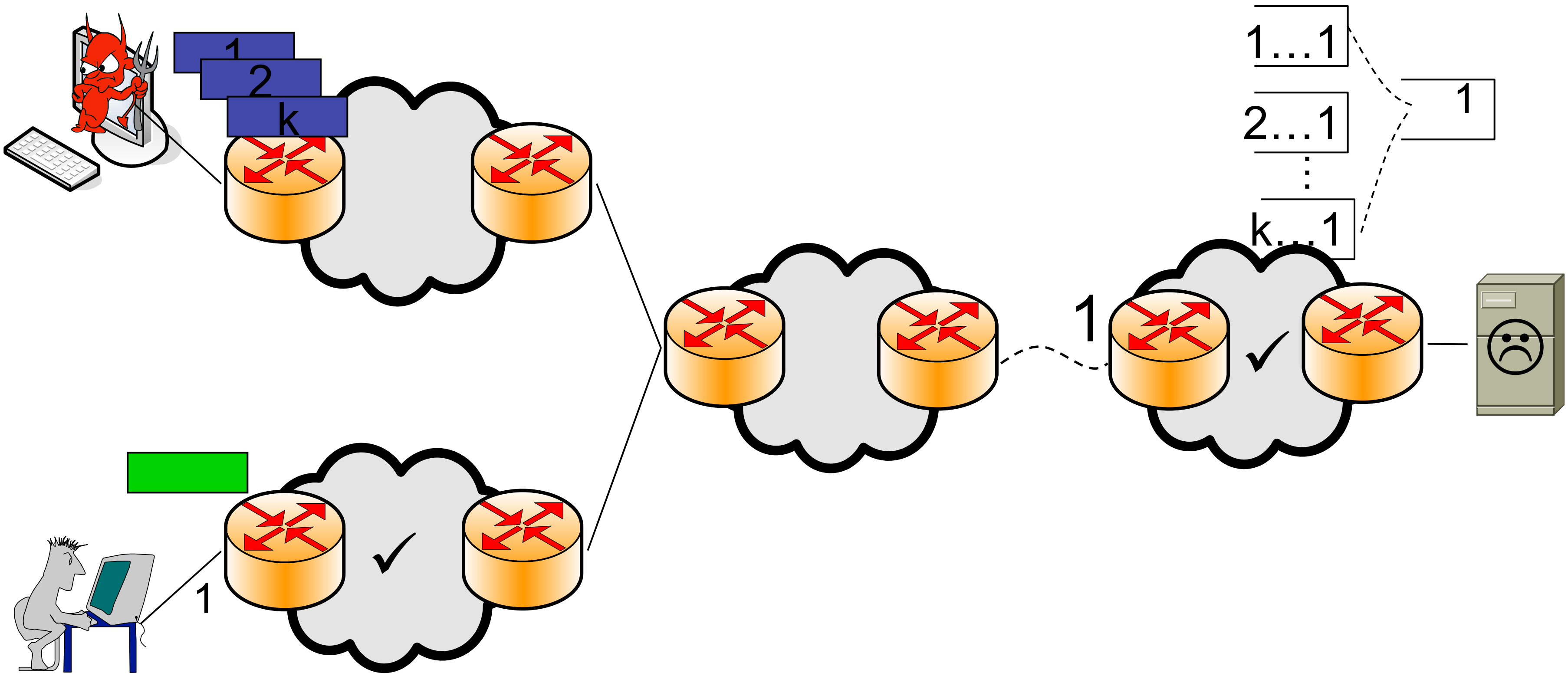
- With Passport, compromised hosts cannot spoof a victim's address

Limited protection using path identifiers



- Deep hierarchy may starve legitimate requests
  - $(1/\text{degree})^L$

Limited protection using path identifiers

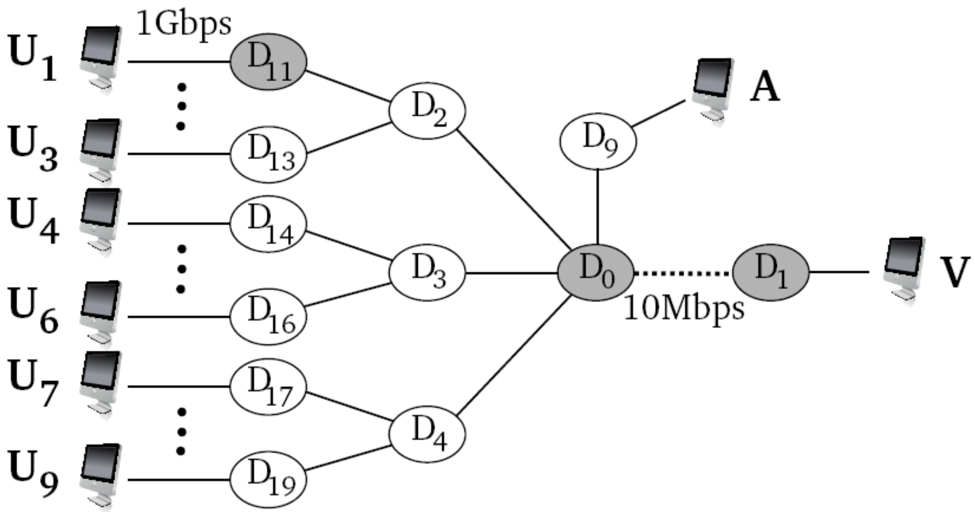


- Deep hierarchy may starve legitimate requests
- Path spoofing is possible in non-deployed regions
- "Denial-of-capabilities"

Experimental validation

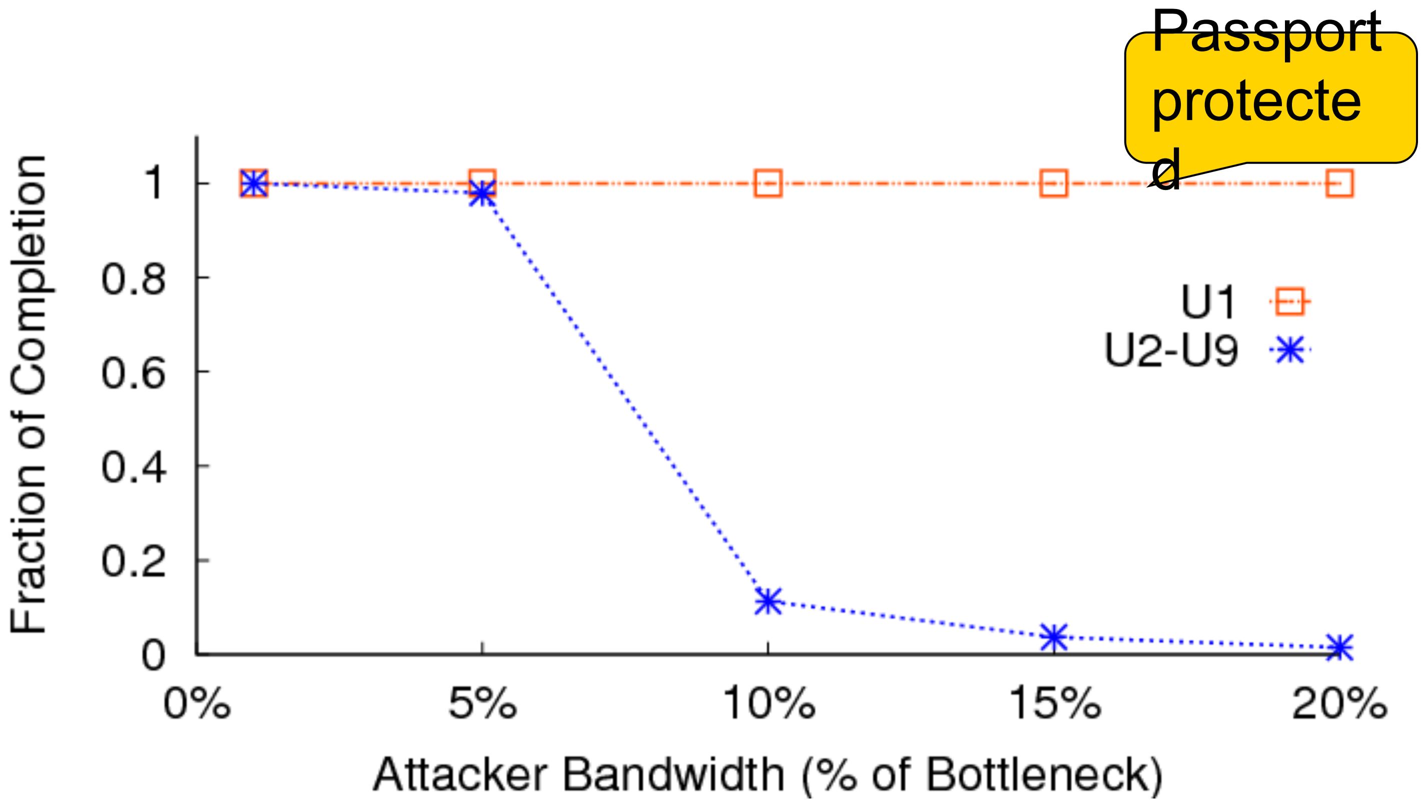
- Mitigate reflector attacks
- Capability-based DoS defense systems
- Secure filtering

Reflector mitigation experiments



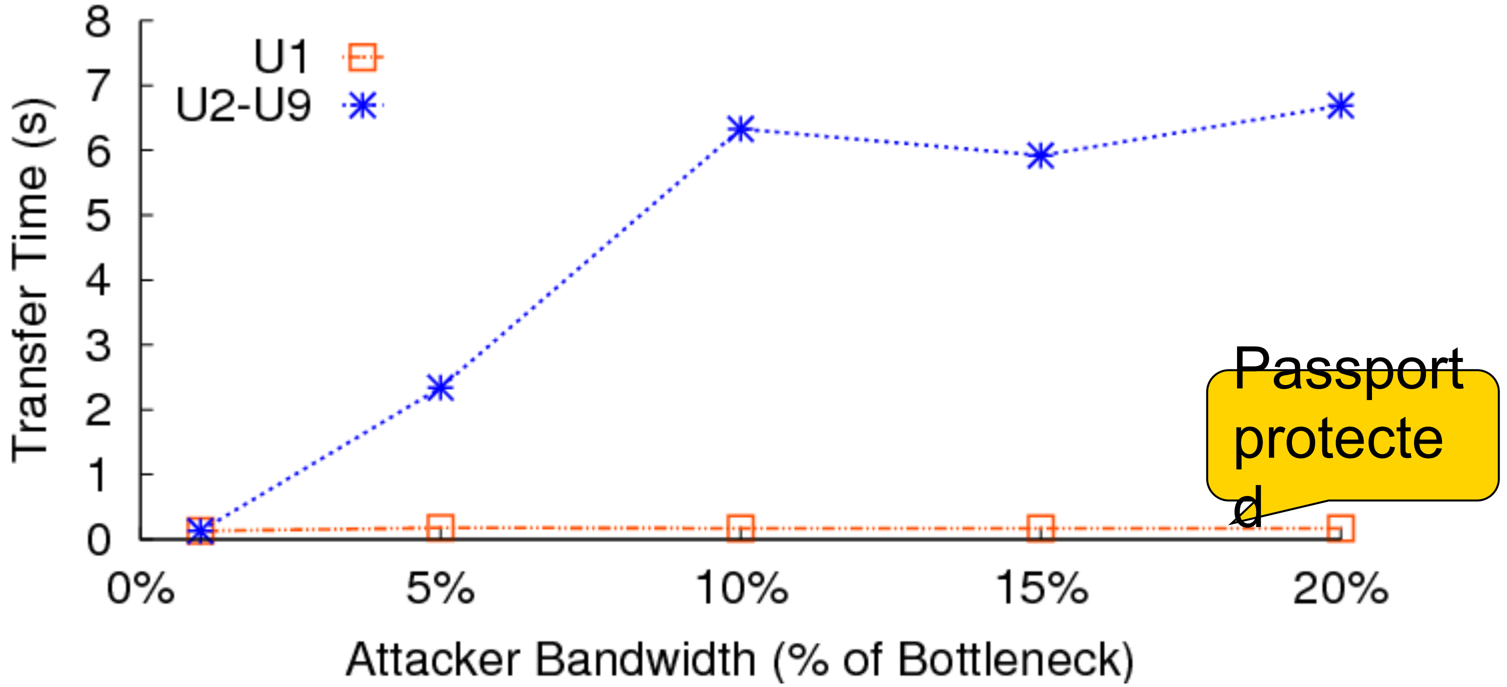
- Shaded circles represent Passport-enabled ASes
- Emulate a DNS reflector attack on a testbed
  - $U_2 \sim U_9$ are reflectors
  - 40 times of traffic amplification
- Metrics
  - TCP transfer times
  - Fraction of completed transfers

Passport mitigates reflector attacks



- 20KB file size
- 60ms round trip time

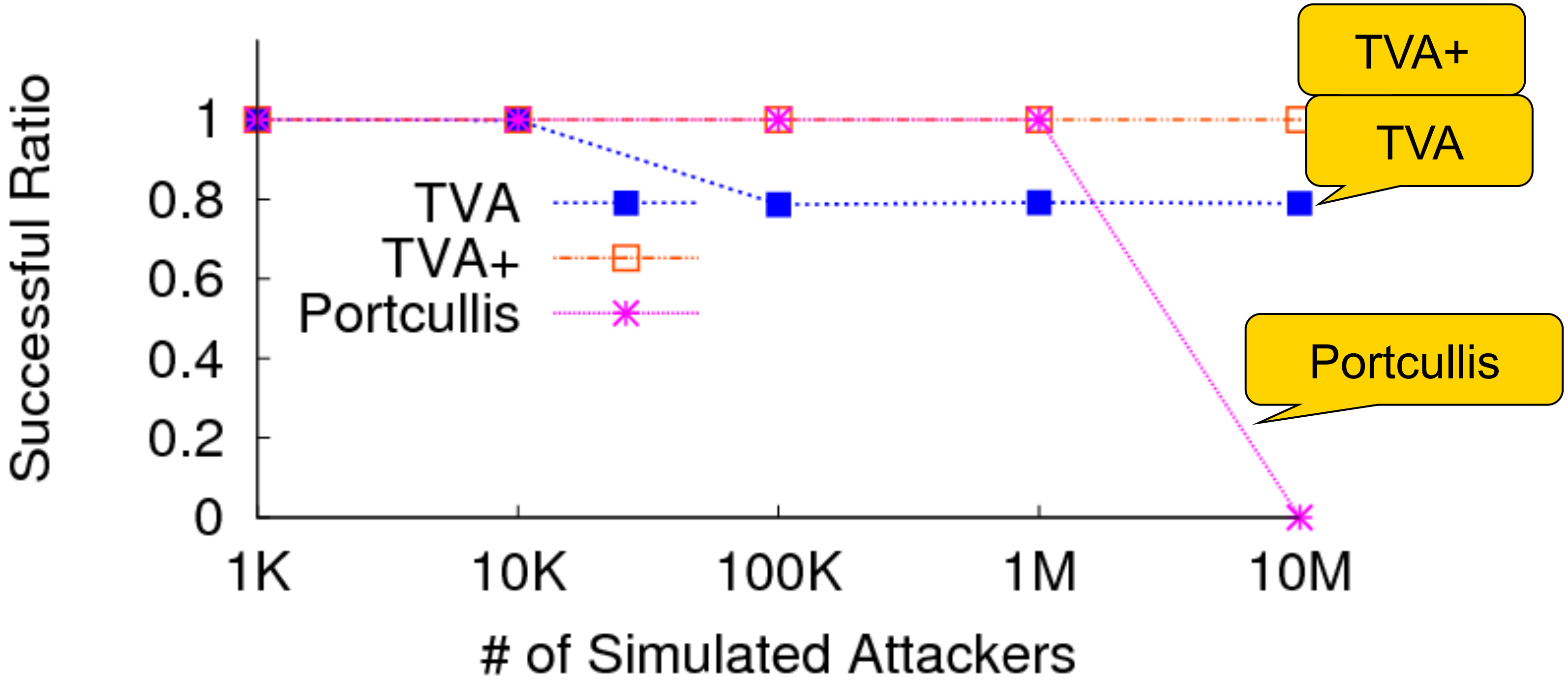
Passport mitigates reflector attacks



Evaluate Passport-enhanced capability-based systems

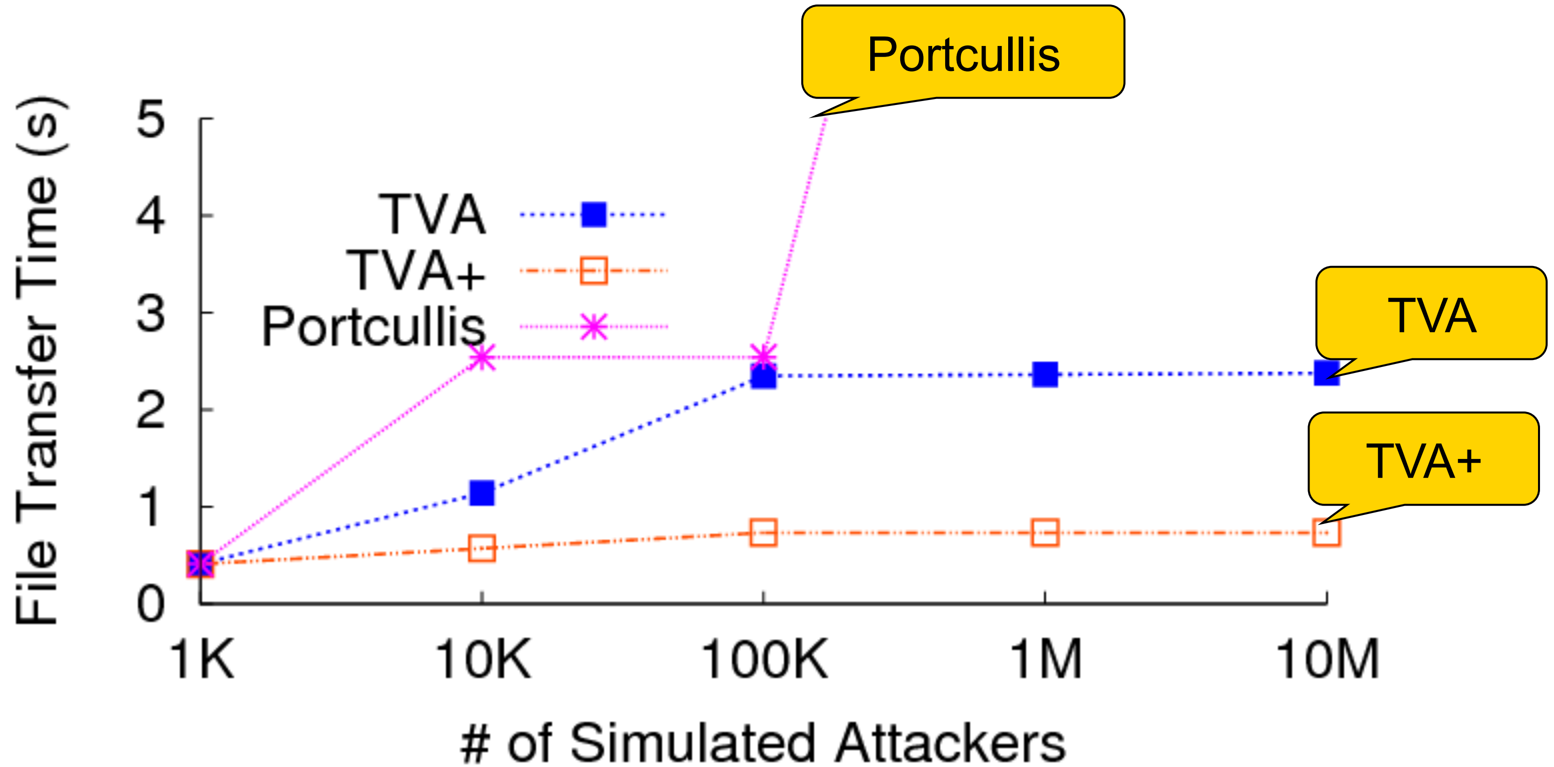
- Realistic Internet topologies from RouteViews
- Simulations on ns-2
  - TVA
  - TVA + Passport
  - TVA + Portcullis (a puzzle-based solution) [Parno 07]
- Metrics
  - TCP transfer times: 20KB files
  - Fraction of completed transfers

Passport improves capability-based systems

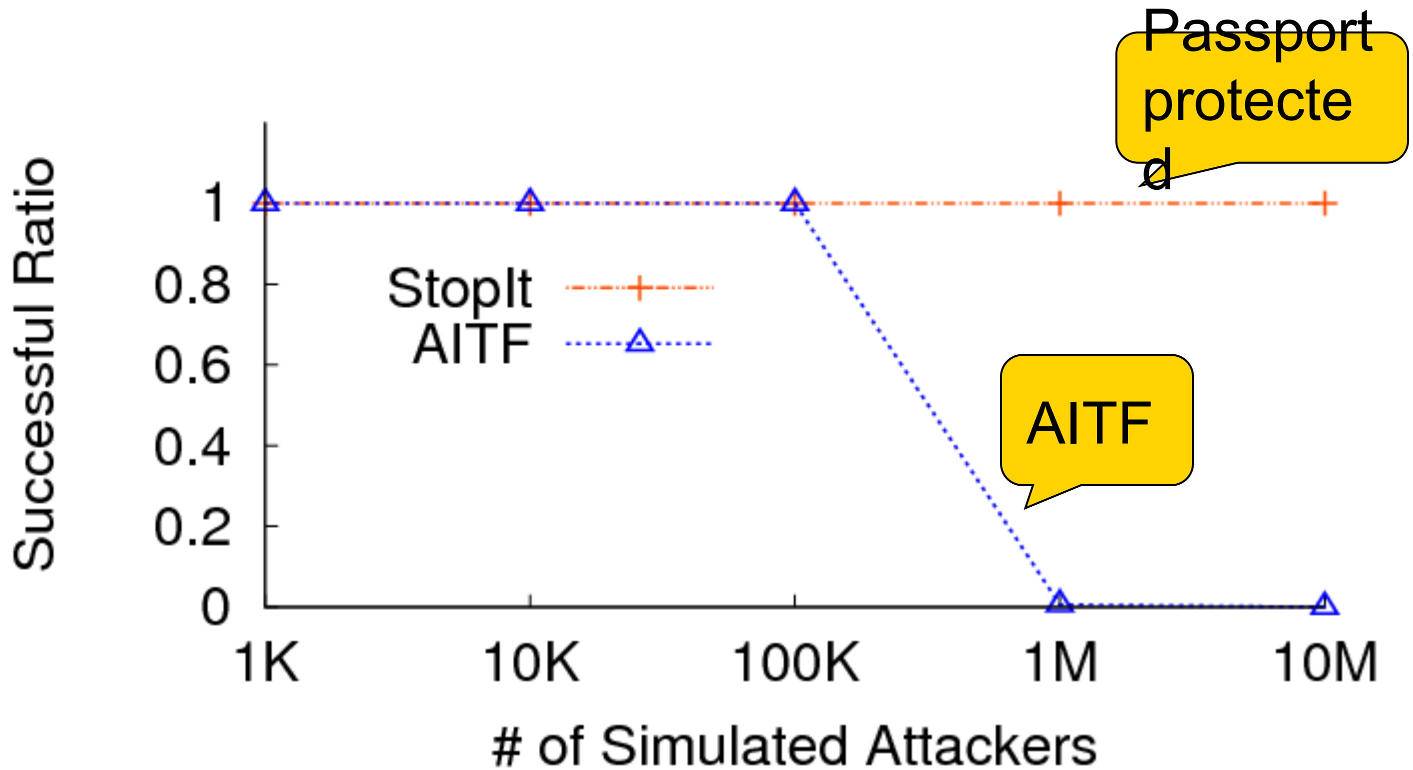


- Full deployment
- Results are users in clean ASes
- Improvement in partial deployment more

Passport-enabled capability-based systems

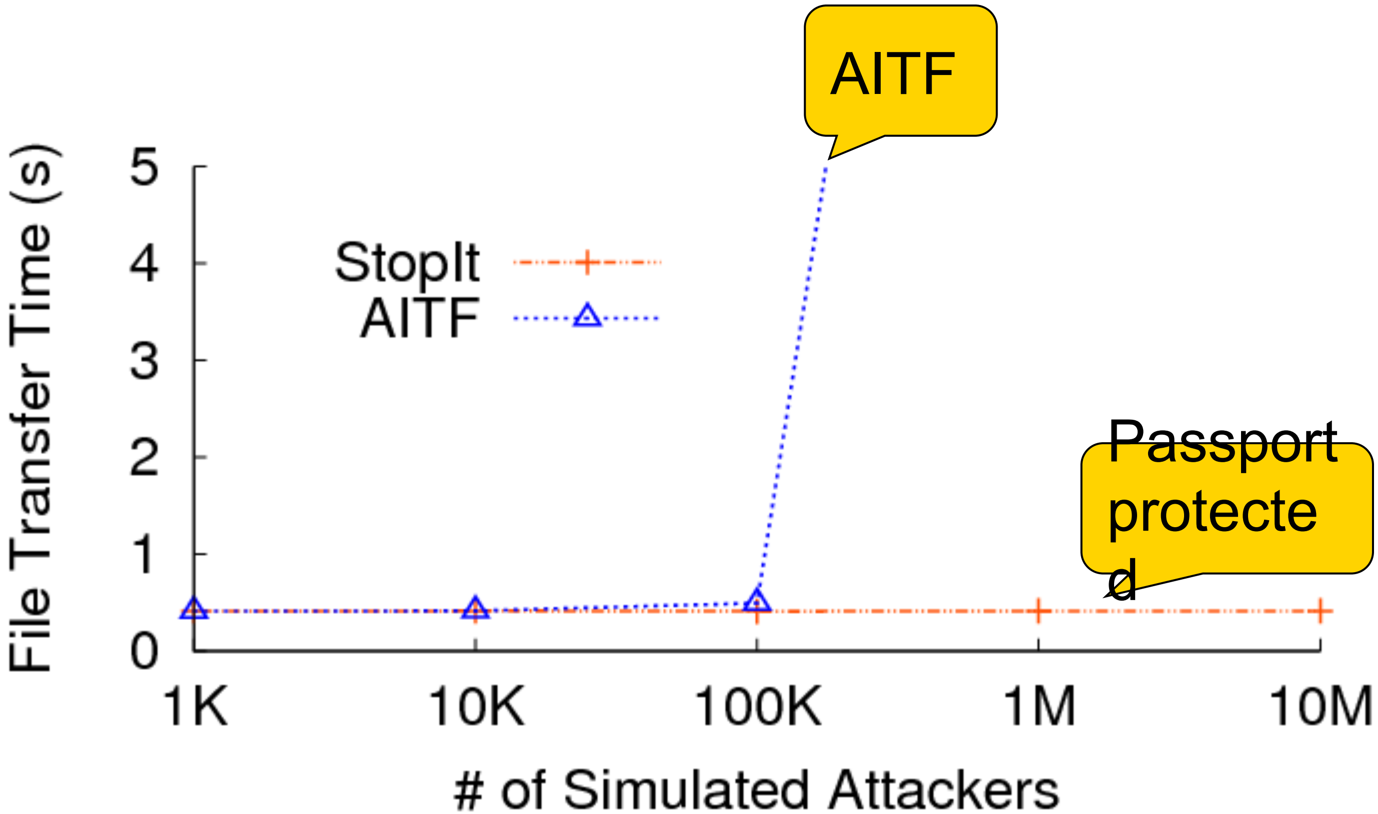


Passport enables secure filtering

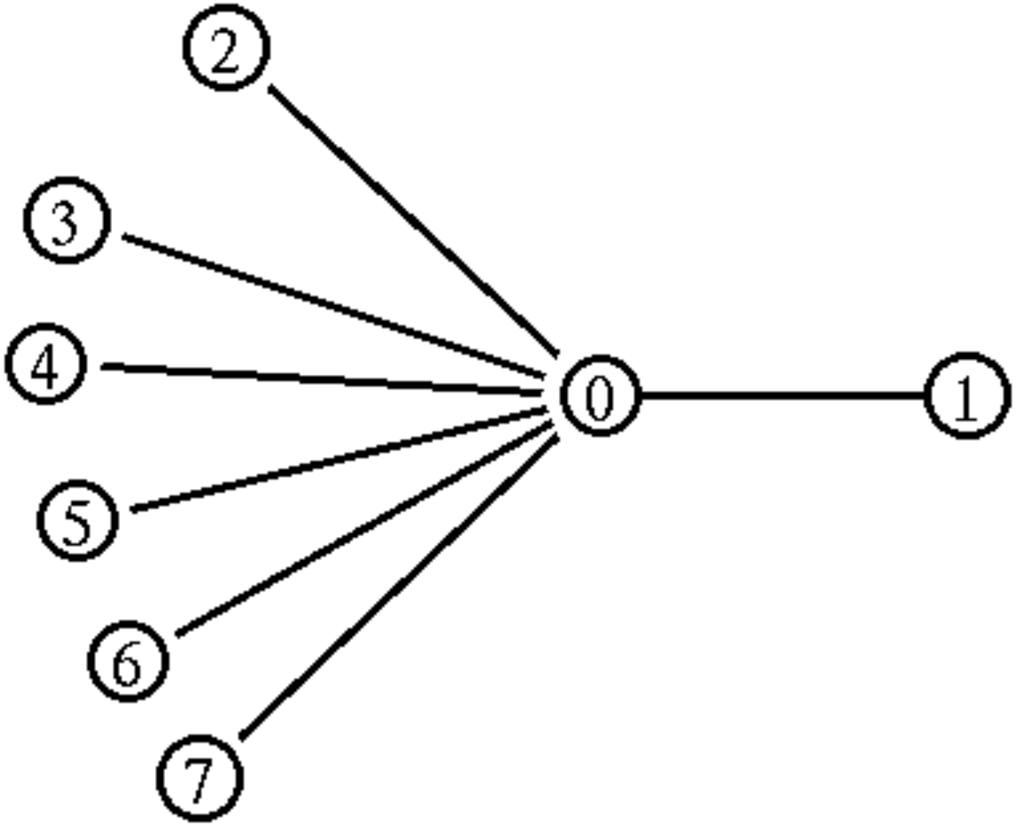


- Comparing with an early filter system Active Internet Traffic Filter [Argyraki05]
  - Path stamping to mitigate spoofing
  - Three-way handshake to verify filter requests

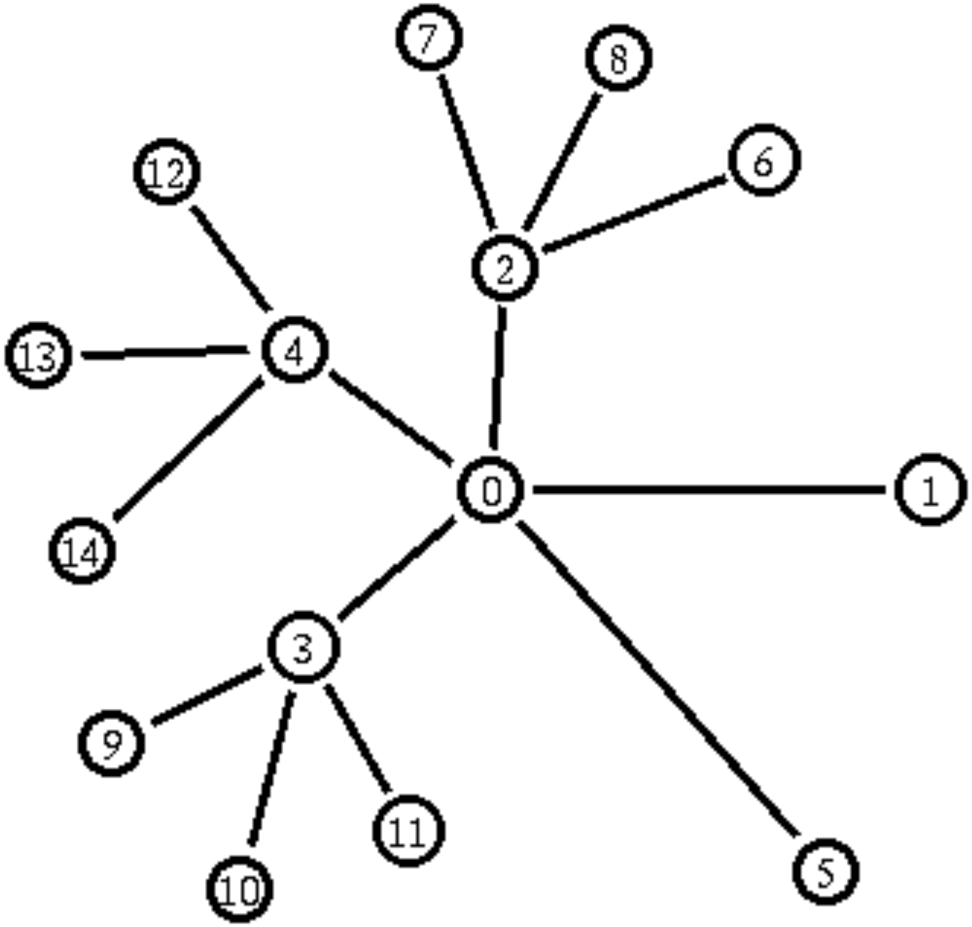
Passport enables secure filtering



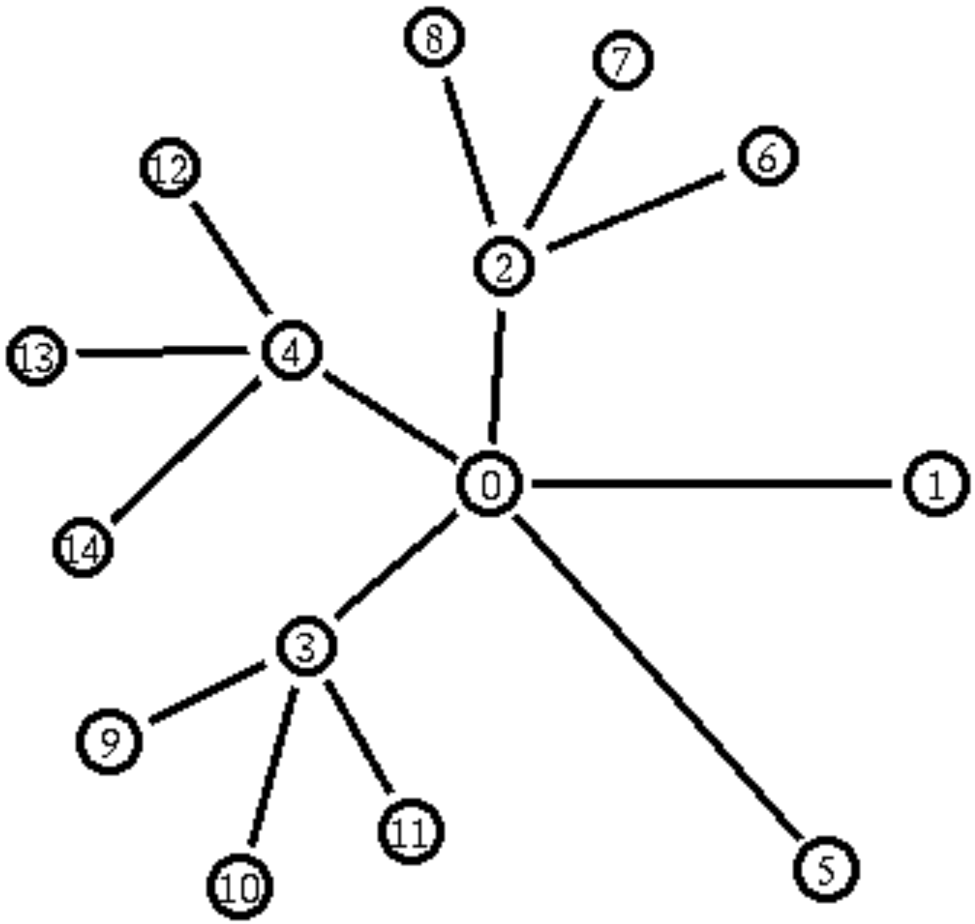
A simulated DoS flooding attack



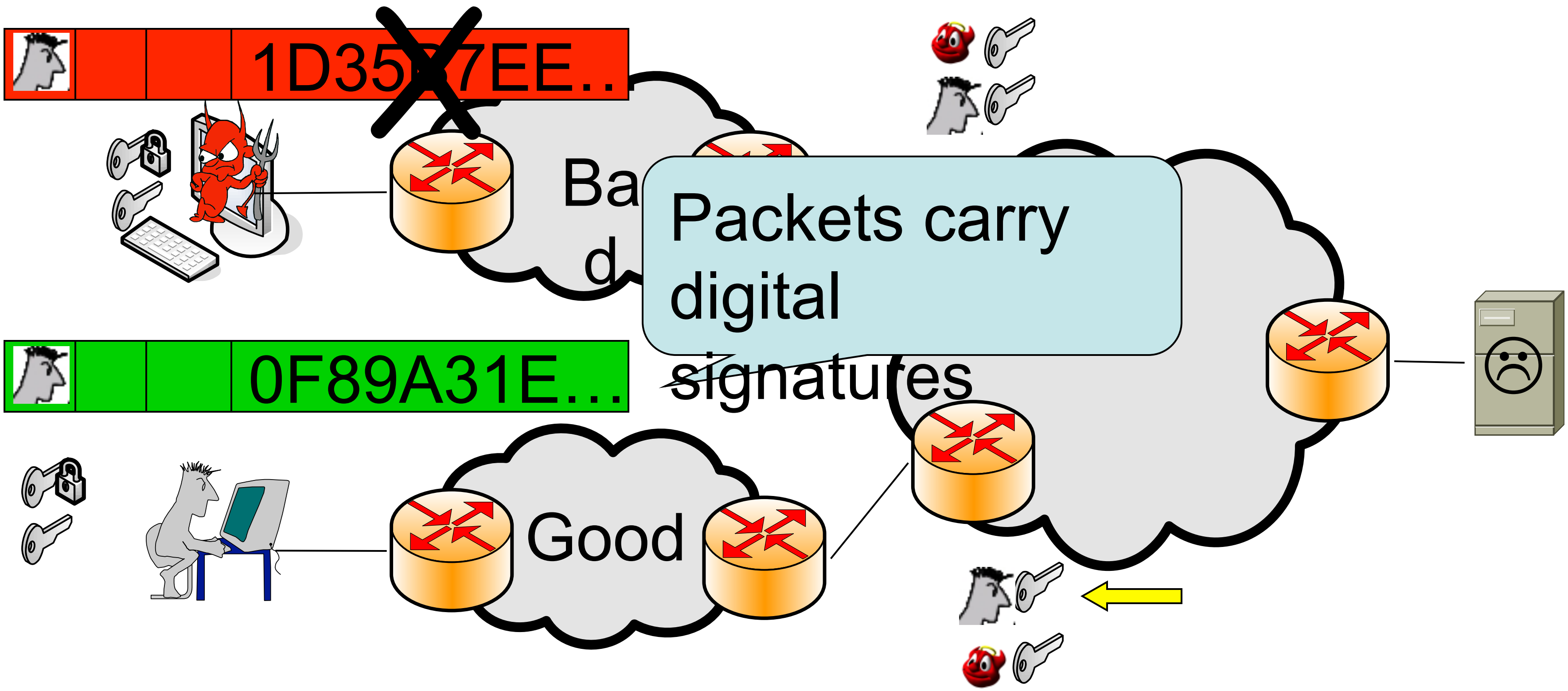
A simulated reflector attack



Passport mitigates reflector attacks

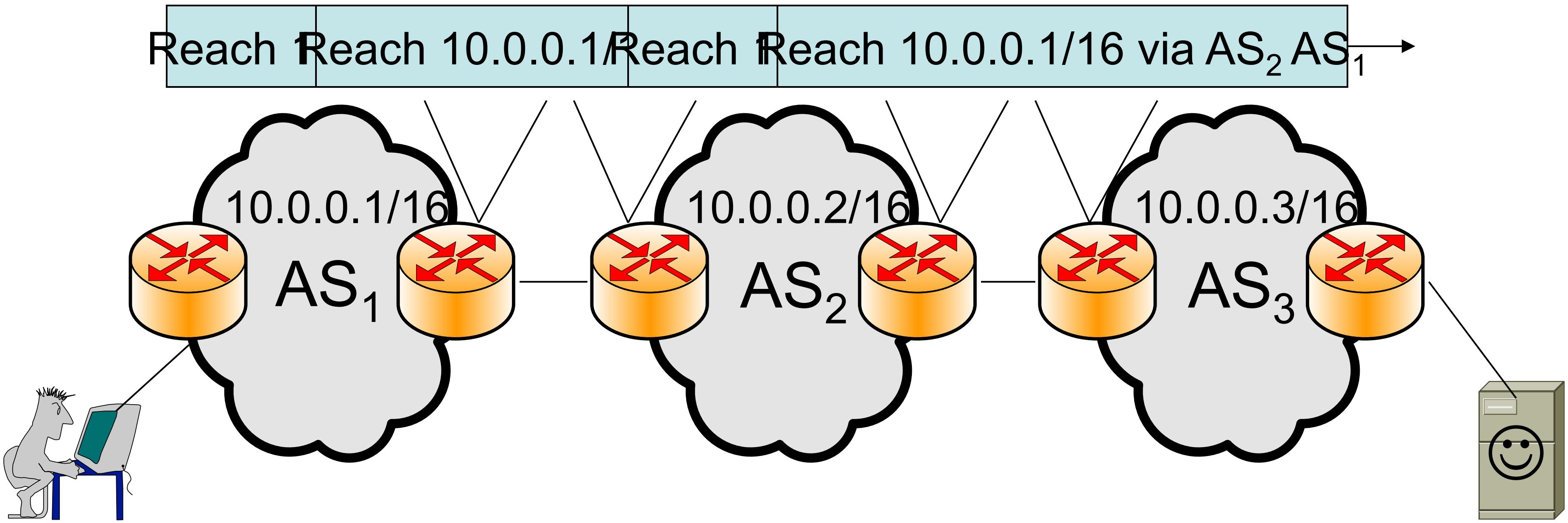


Digital signatures: heavyweight



- Public key infrastructure
- Time-consuming to verify
- High packet header overhead: e.g. RSA ~512

Solution: use routing to distribute keys



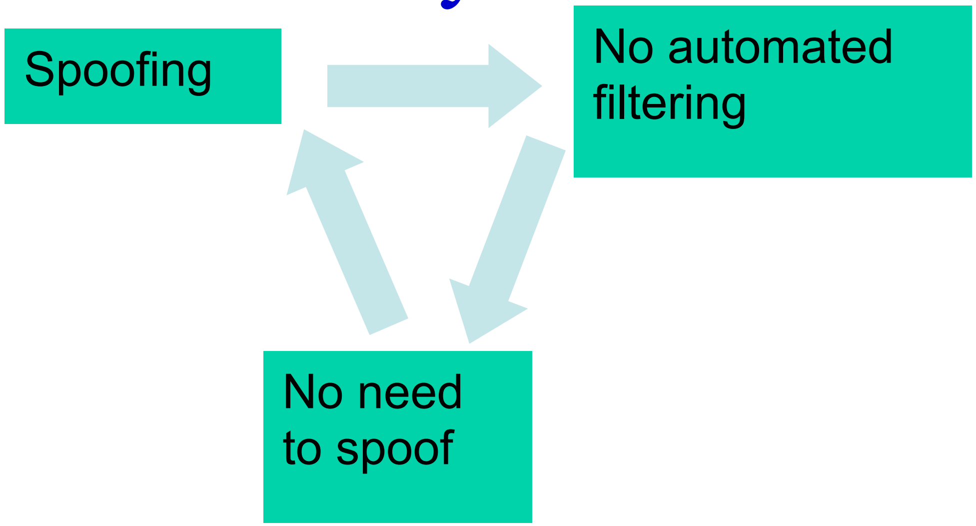
- Routing proceeds packet forwarding
- Routing implements reliable broadcast

Most newsworthy weakness of the Internet

- Nearly 4000 attacks per week [Moore01]

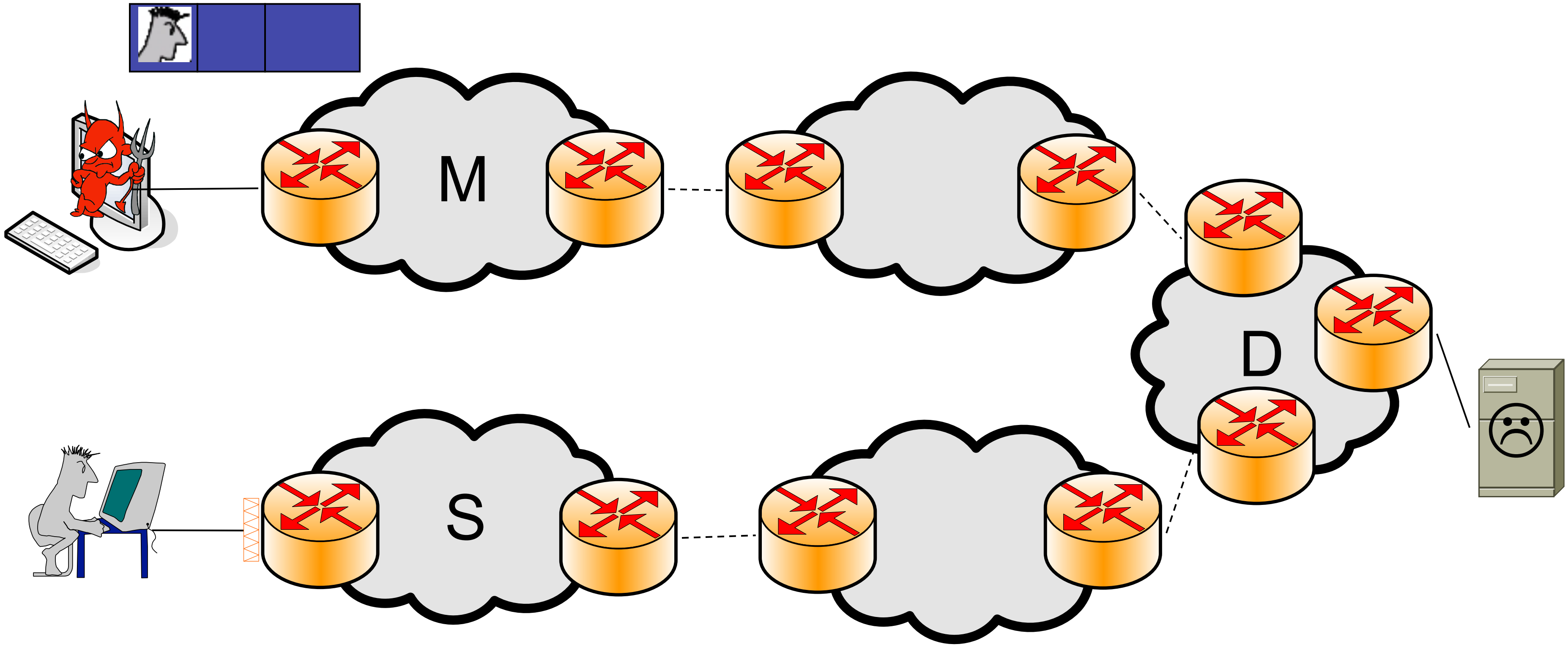
- Data from Prolexic Technologies [Claiborne07]
  - Less than 0.1% of DDoS attacks ending in an arrest in US
  - A major US corporation lost over 2 million in a 2 hour outage
  - An Online payment processor lost 400 thousand in just under 72 hours
  - …

Possibility of spoofing creates a vicious cycle



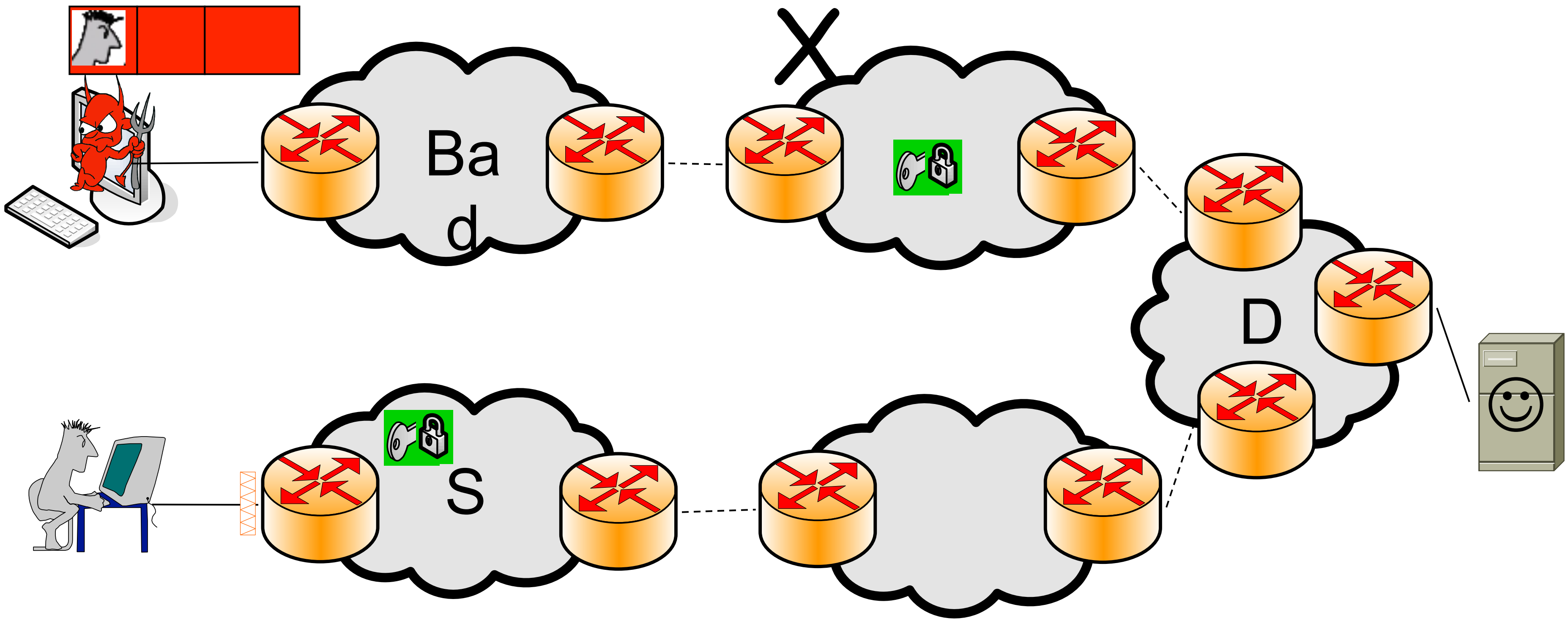
- “Steps towards a DoS-resistant Internet architecture,” *Handley and Greenhalgh, 2005*

Ingress filtering: little incentive



- “Self quarantine”
- Spoofer: ~ 20% of IP addresses or networks still allow spoofing
- You’ve heard Hubble

Passport



- Compromised hosts or networks cannot spoof addresses of other deployed networks