

# Practical Network Support for IP Traceback

---

Stefan Savage  
University of Washington/  
University of California, San Diego

David Wetherall, Anna Karlin and Tom Anderson  
University of Washington, Seattle

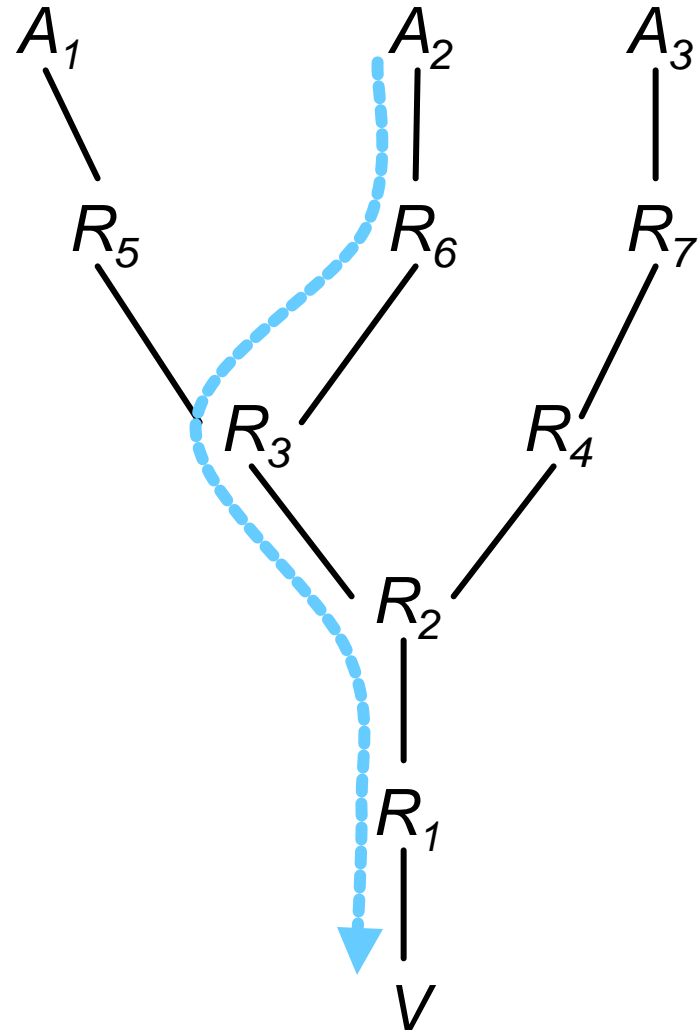
# Motivation

---

- Denial-of-service via packet flooding
  - Attackers overwhelm network and CPU resources with spurious requests
- Internet architecture permits anonymity
  - Use of IP source address is **voluntary**
  - Attackers use “false” source address
  - Network is stateless (no audit trails)
- Difficult to respond to attack without knowing what network path(s) it traverses

# Traceback problem

- Basic elements
  - Set of attackers  $A_i$
  - Set of routers  $R_i$
  - Victim  $V$
- Attack path for  $A_i$ 
  - Ordered list of routers between  $A_i$  and  $V$
  - e.g.  $\{R_6, R_3, R_2, R_1\}$
- Goal
  - Determine attack path for each attacker



# State of the practice

---

- Router traffic stats + manual guesswork
- Input debugging
  - Install filter to detect attack traffic, determine upstream router, repeat
- Serious practical limitations
  - High management overhead, per-ISP only
  - Attack must be in progress

# Design constraints (self-imposed)

---

- **Assumptions about environment**
  - Attacker can generate any packet
  - Multiple attackers may conspire (DDoS)
  - Attackers send many packets
  - Routers resource limited (no *per-flow* state)
  - Routers not malicious
- **Goals**
  - Real-time and *post-mortem* trace capability
  - Incrementally deployable
  - Low management and network overhead

# Key idea

---

- **Routers store path state in packets**
- Naïve approach
  - Routers append their address to each packet
  - Can be expensive to implement
  - No assurance of “enough” space in packet
- Spread state across packets?

# Edge sampling: basic approach

---

- Augment packets with additional fields
  - **Edge**: two adjacent router addresses (start & end)
  - **Distance**: # edges traversed since marked
- Probabilistically mark packets in routers
  - Marked packets contain “sample” of path
- Victim, or victim’s ISP, reconstructs path after receiving enough packets

# Marking procedure at router

---

- Marking procedure for packets forwarded by  $R$ :
  - with probability  $p$ ,
    - write  $R$  into *start* field
    - write 0 into *distance* field
  - else
    - if *distance* == 0 then
      - write  $R$  into *end* field
    - increment *distance* field

# Path reconstruction at victim

---

- Extract identifiers from attack packets
- Build graph rooted at victim
  - Each  $(start, end, distance)$  tuple is an edge
  - Eliminate edges with inconsistent distance
  - Traverse edges from root to find attack paths
- # packets needed to reconstruct path

$$E(X) < \frac{\ln(d)}{p(1-p)^{d-1}}$$

$p$ : marking probability  
 $d$ : length of path

# Robustness

---

- Random marking decisions
  - Can't be anticipated by attacker
  - Can't be controlled by attacker
- Attackers can “lie” and invent edges
  - But not for *distances* less than their own
- Offline validation of “valid suffix”

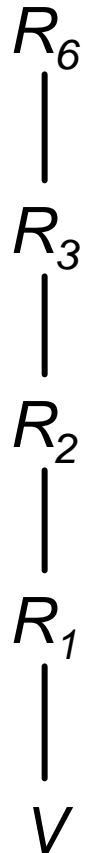
# Practical problems

---

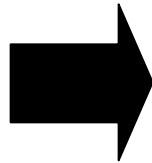
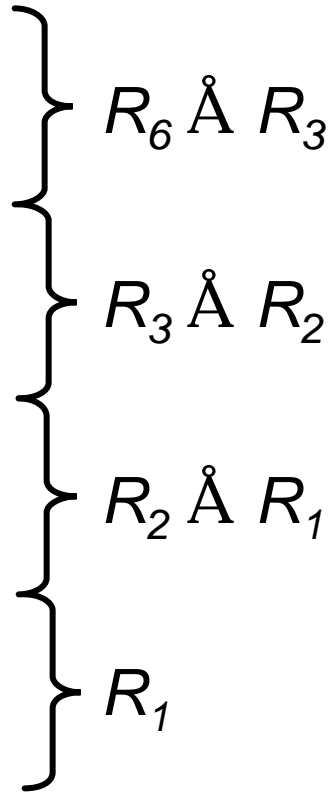
- Deployment issues
  - How to fit edge/distance data into IP header
  - Current scheme requires too much space
    - 32 bits for each router address, 8 for distance
    - Need lower per-packet overhead
- One hybrid approach
  - Trade space for reconstruction/robustness
  - Overload existing packet header field

# Compressing edges to edge-ids using XOR

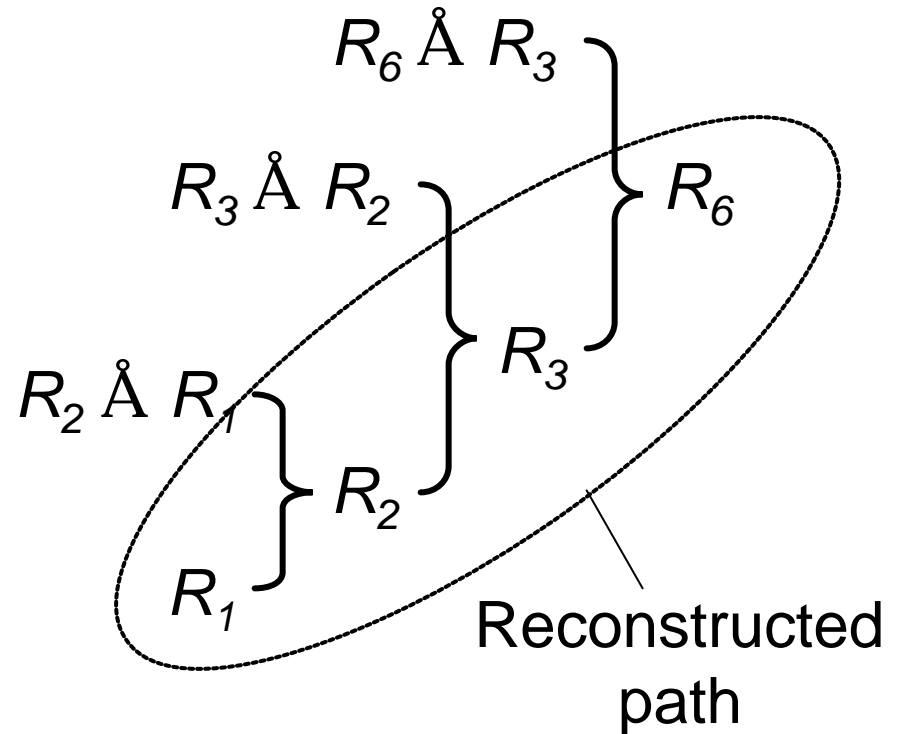
Attack path



Distinct edge-ids



Path reconstruction



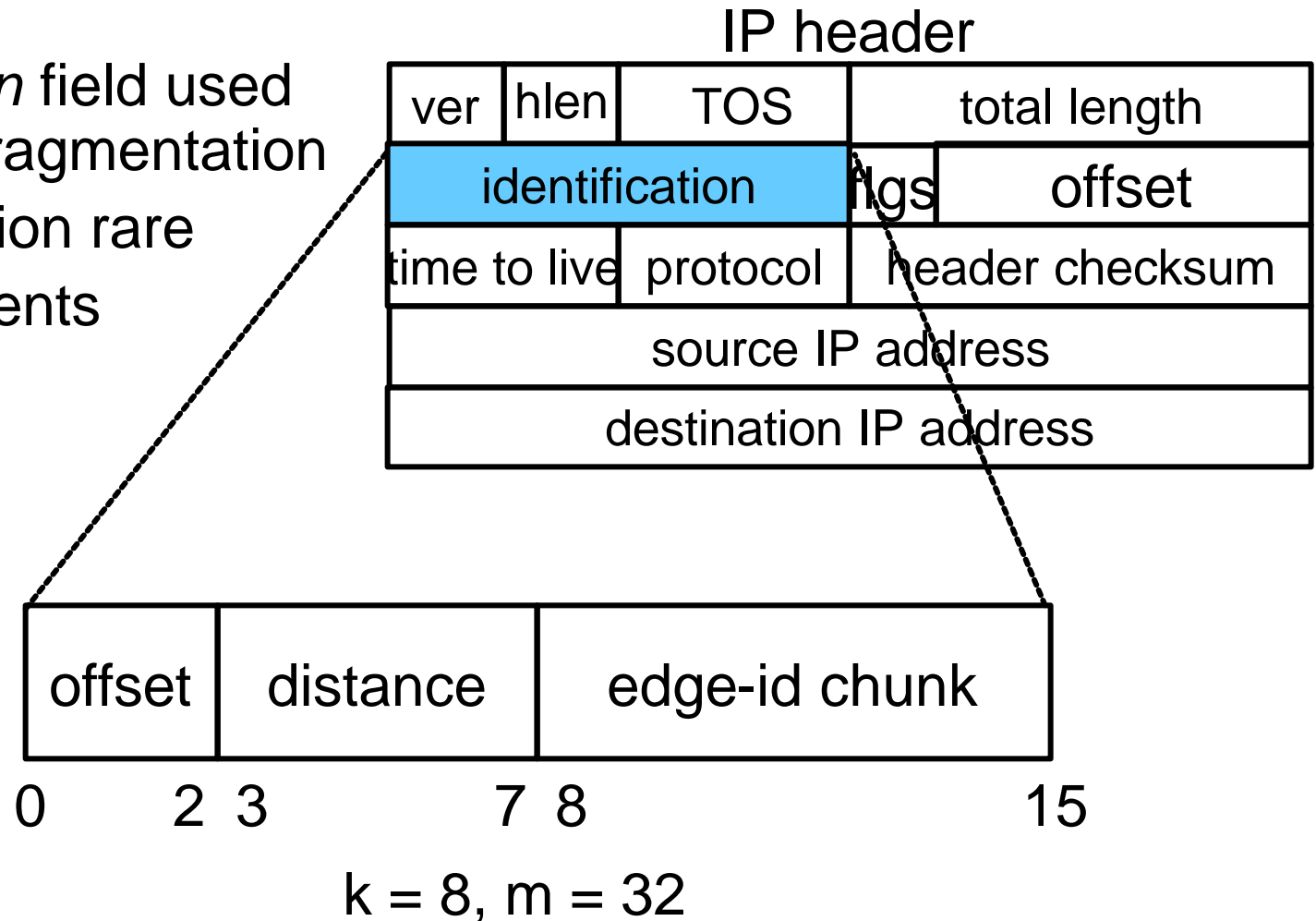
# Edge-id chunks

---

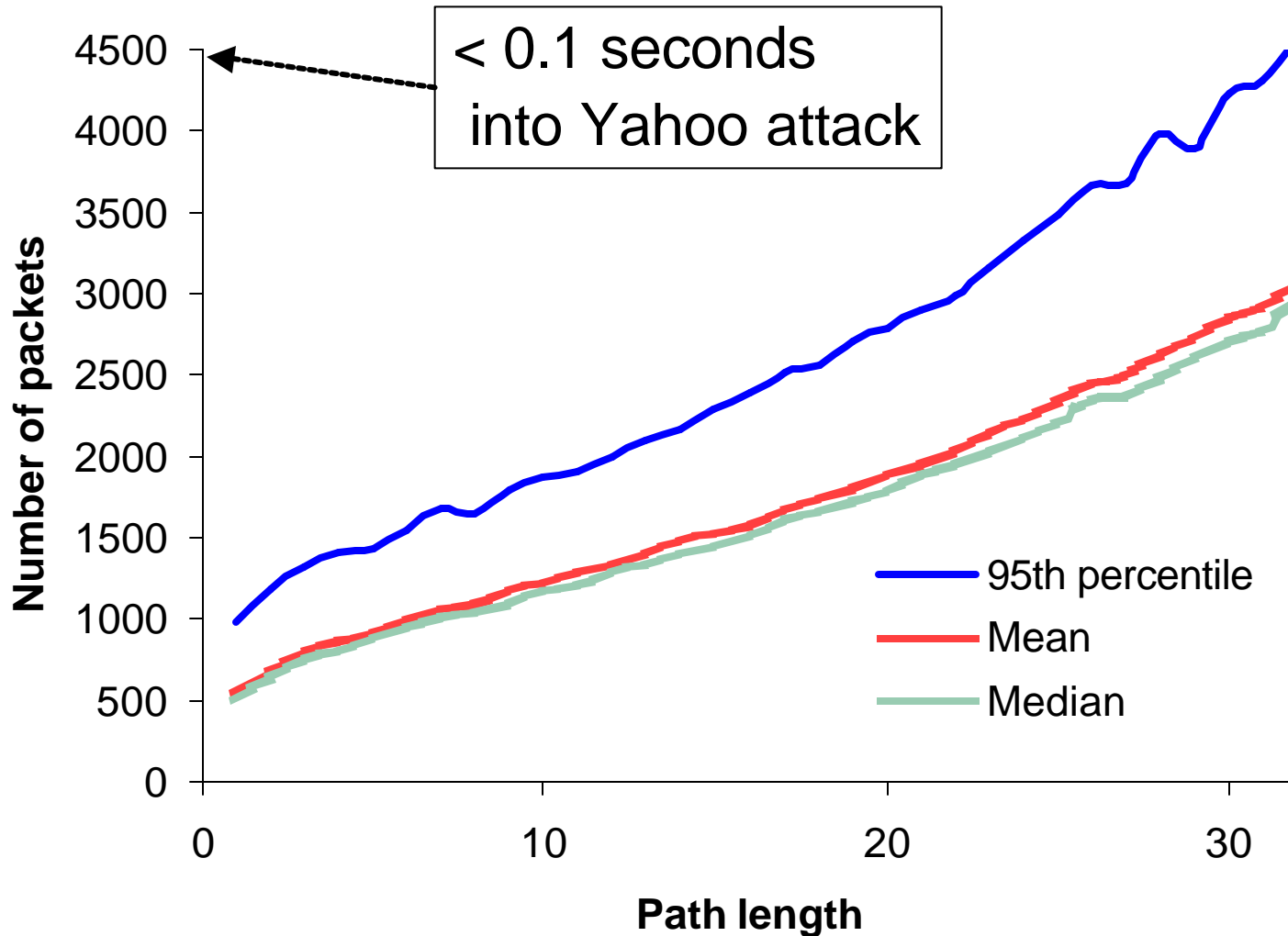
- Divide each edge-id into  $k$  chunks
- When marking packet
  - Pick edge-id chunk at random
  - Write chunk and its offset into packet
- Increases reconstruction time by  $k \ln(k)$
- Chunks may not be unique
  - Augment edge-id with hash of  $m$  bits
  - Validate chunk combinations at victim

# One opportunistic encoding

- *Identification* field used for packet fragmentation
- Fragmentation rare
- Mark fragments separately



# Experimental convergence time



# Weaknesses

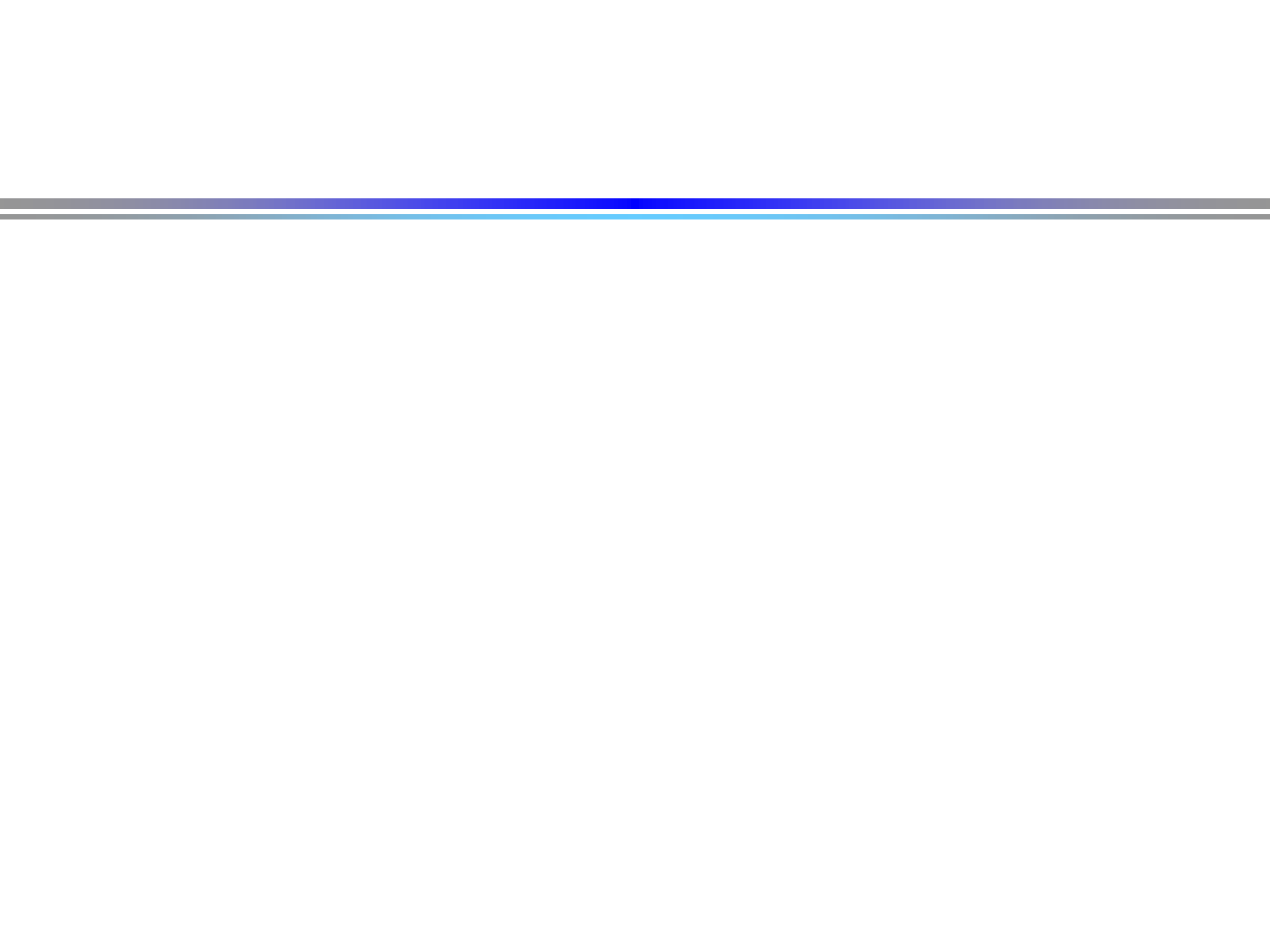
---

- Path validation/authentication
- Robustness in highly distributed attacks
  - *Both addressed nicely in [Song&Perrig00]*
- Compatibility issues (IPsec AH, IPv6)
- Origin laundering (reflectors, tunnels, etc)

# Summary

---

- An efficient algorithm for tracing anonymous attacks: *edge sampling*
- Hybrid algorithm with reduced per-packet space overhead (at a cost)
- Potential encoding into current IP packet header



# Traceback is only one part of the problem

---

- **Detection**

- How do you know you're under attack?
- What identifies an attack?

- **Countermeasures**

- What do you do after you backtrace the attack?
- How to filter attack but not customer traffic?
- How to enable inter-ISP cooperation?

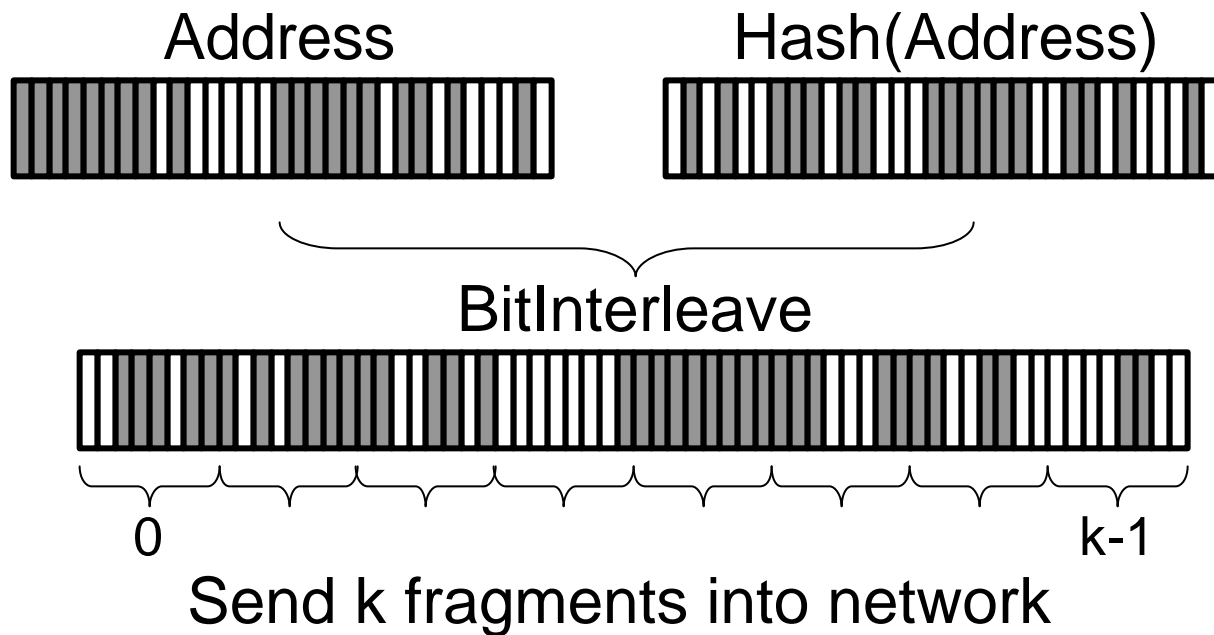
# Ingress/egress filtering

---

- Idea is that routers block packets with “invalid” source addresses
- Great idea, but...
  - Really only possible at edge of network
  - Requires universal deployment
  - Management overhead for multi-homed hosts
  - Rarely employed inside enterprises  
(still  $2^{16}$  “valid” source addresses at UW)

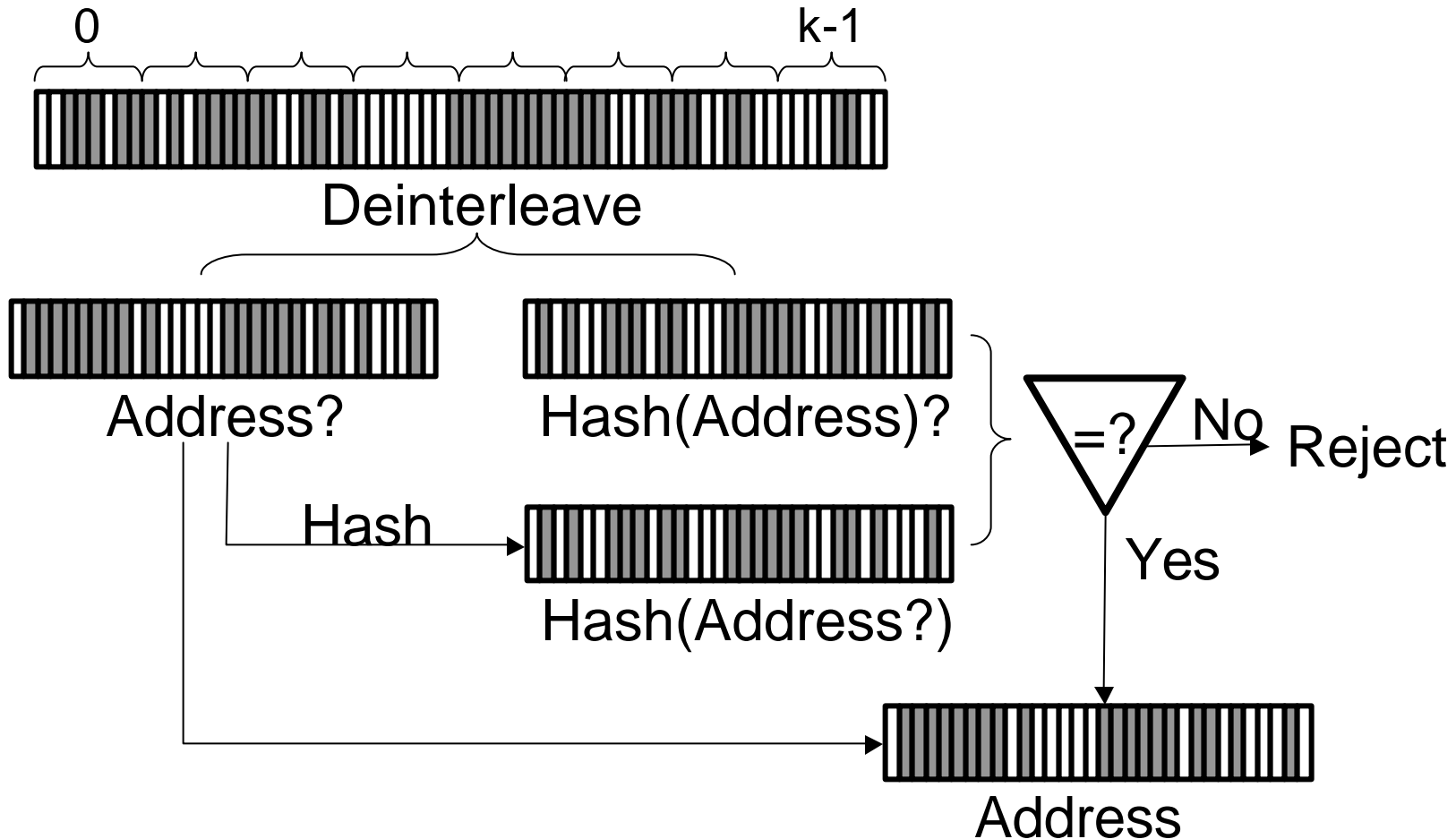
# Augment router id with hash

Interleave address data with random hash



# Hash-based consistency check

Combine k fragments from network



# Modified path reconstruction

---

- Sort edge-id fragments by distance
  - For each distance, sort by offset
  - Try all combinations until hash matches
- Probability of “false positive” at distance  $d$

$$1 - \left(1 - \frac{1}{2^h}\right)^{m^k}$$

$m$ : # disjoint edges at distance  $d$

$h$ : length of hash in bits

- Design tension in size of  $k$  and  $m$

# ICMP Traceback

## [Bellovin, Draft spec]

---

- **Sample packets in routers (~1/20,000)**
- **Generate special ICMP message**
  - Includes forward hop info, reverse hop info, authentication & origin packet contents
  - Set TTL to 255
- **Reconstruct path at victim**
  - TTL ordered by distance
  - Match forward/reverse hop info

# Comparison

---

- Very similar, but out-of-band
- Advantages
  - Has authentication built in
  - Few backwards-compatibility issues
- Disadvantages
  - ICMP increasingly filtered
  - Painful to trace through discontinuities caused by partial deployment
  - Must associate attack and signal
- Open challenges for both
  - Chaff (clogging attacks) and laundering

# Edge sampling algorithm

---

- **Probabilistically mark packets in routers**
  - *Edge*: Two adjacent router addresses
  - *Distance*: edges traversed since marked – starts at 0
  - If packet **not** marked, increment distance
- **Reconstruct path at victim**
  - Collect marked packets, construct graph
  - # packets needed to reconstruct path

$$E(X) < \frac{\ln(d)}{p(1-p)^{d-1}}$$

$p$ : marking probability

$d$ : length of path in hops