PACKET AUTHENTICATION FOR ACCOUNTABILITY

Ronald D. Williams
rdw@virginia.edu

Malathi Veeraraghavan
mv5g@virginia.edu

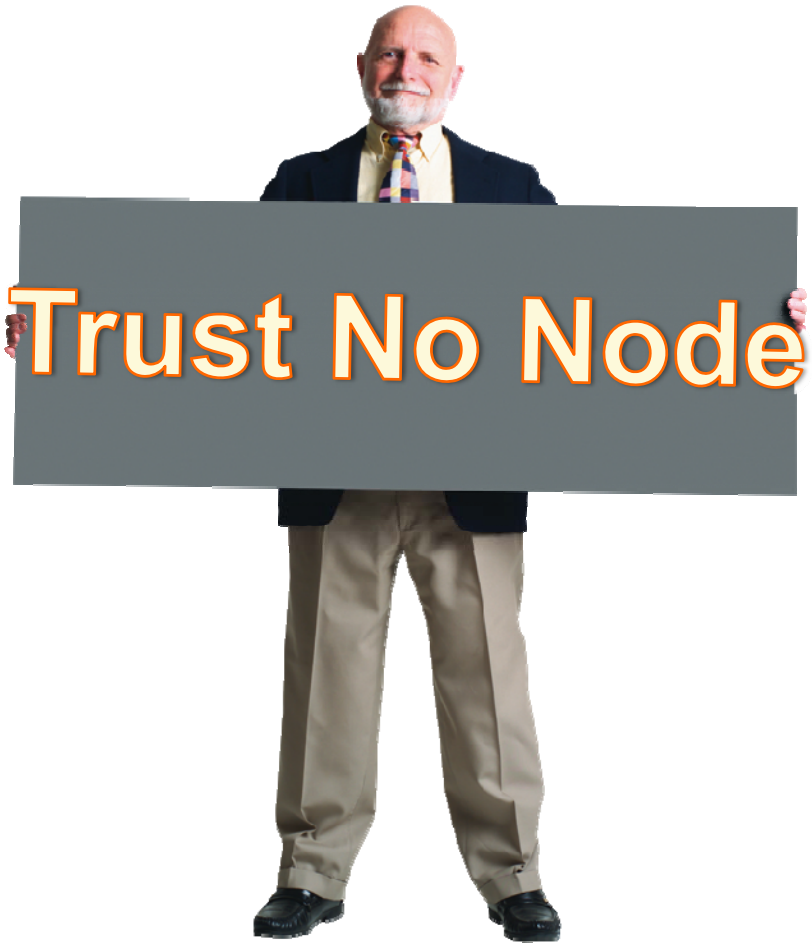
UNIVERSITY of VIRGINIA

Overview

- Attacker Model and Assumptions
- Defense Objectives
- Overview of Operation
- Operation Details
- Implementation Challenges
- Addressing the Challenges

Attacker Model

- Any node between source and destination may:
  - Modify any packet
  - Inject packets
- Nodes may be:
  - Spoofed
  - Compromised



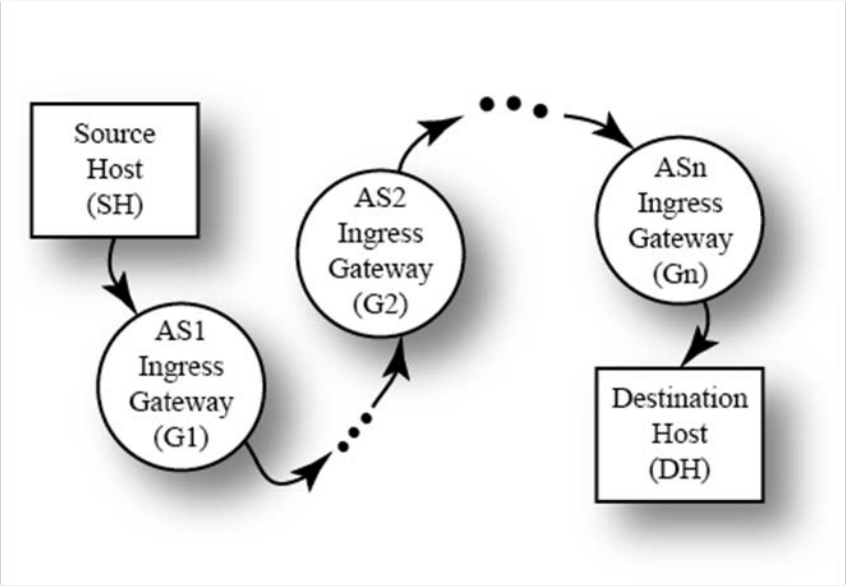
Defense Objectives

- Early detection and deletion of packets that cannot be authenticated
- Hold accountable the source of malicious packets that pass authentication
- Develop approach that involves routers rather than just the source and destination

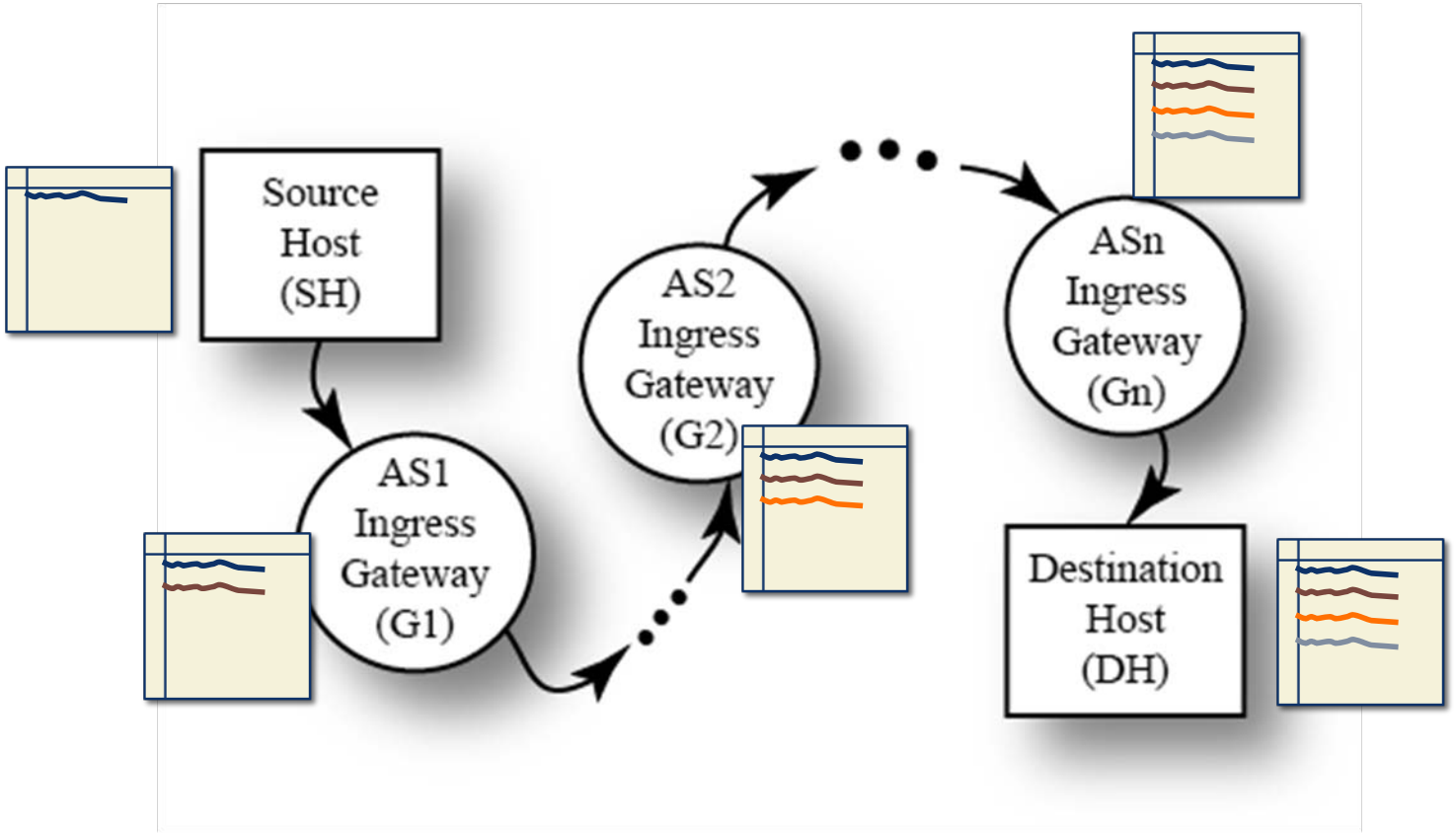
Glenn Ernest Grohe 1942

Packet Path from Source to Destination

- Packet forwarding
  - A source host sends a packet to its provider
  - The packet is forwarded within and between autonomous systems as appropriate
  - The destination host receives the packet from its provider

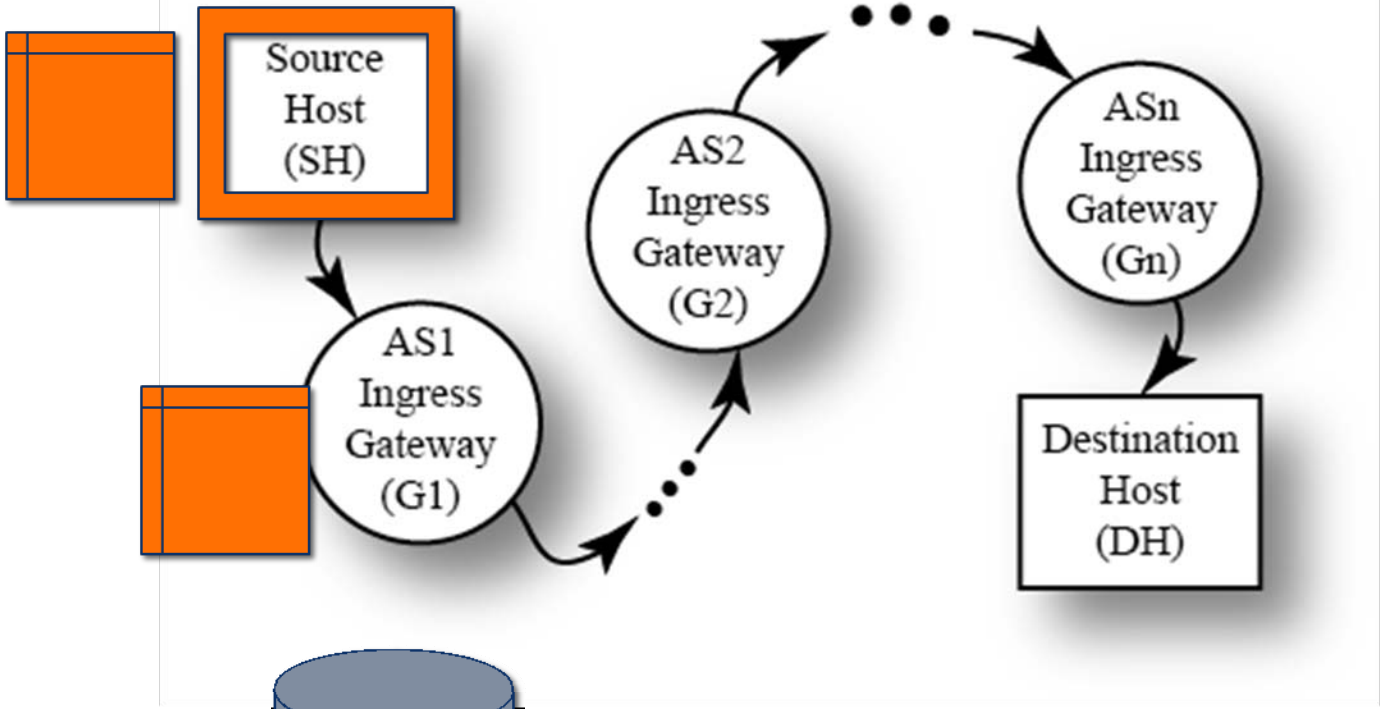


Signatures Protect the Packet



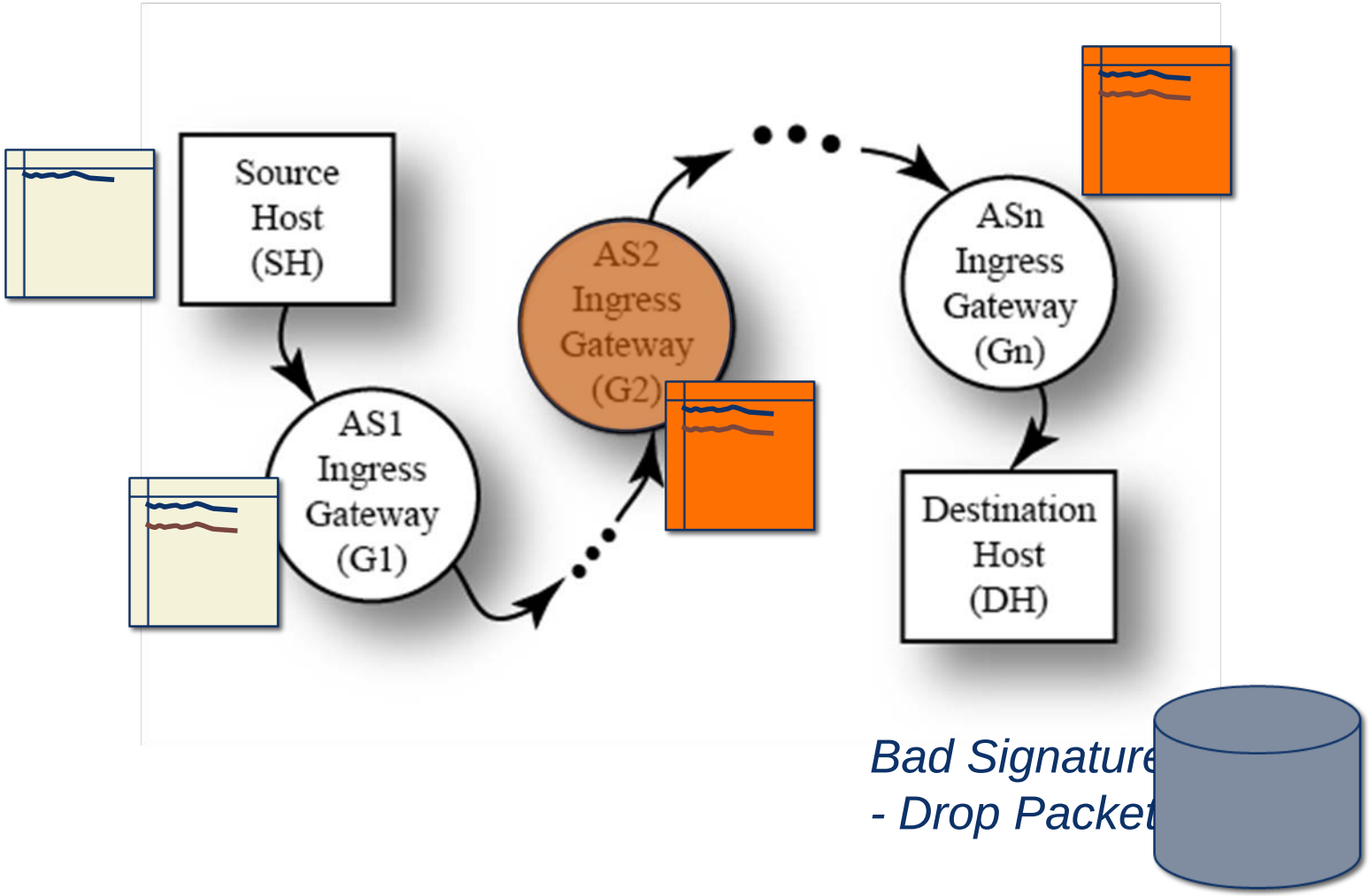
Spoofed Source Host

Cannot produce legitimate signature

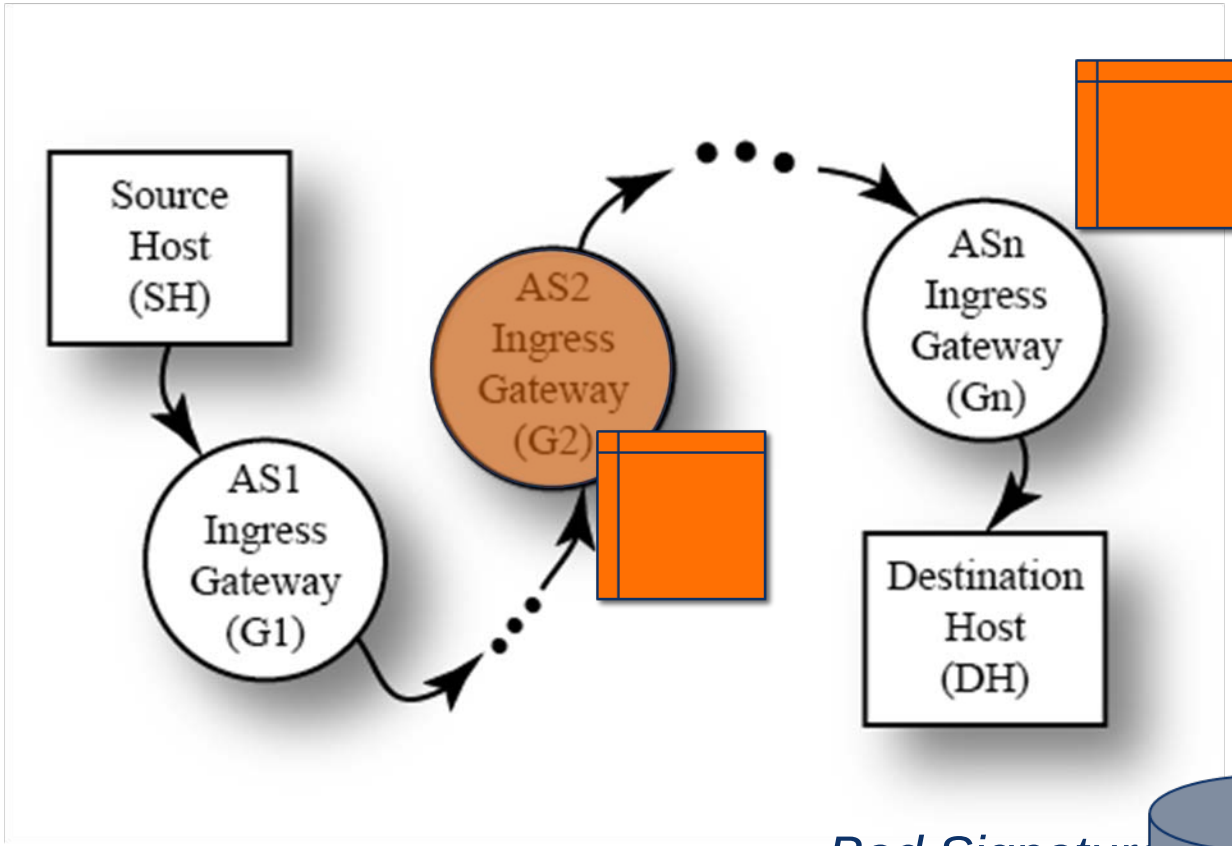


Bad Signatur
- Drop Packe

Spoofed Gateway Packet Corruption



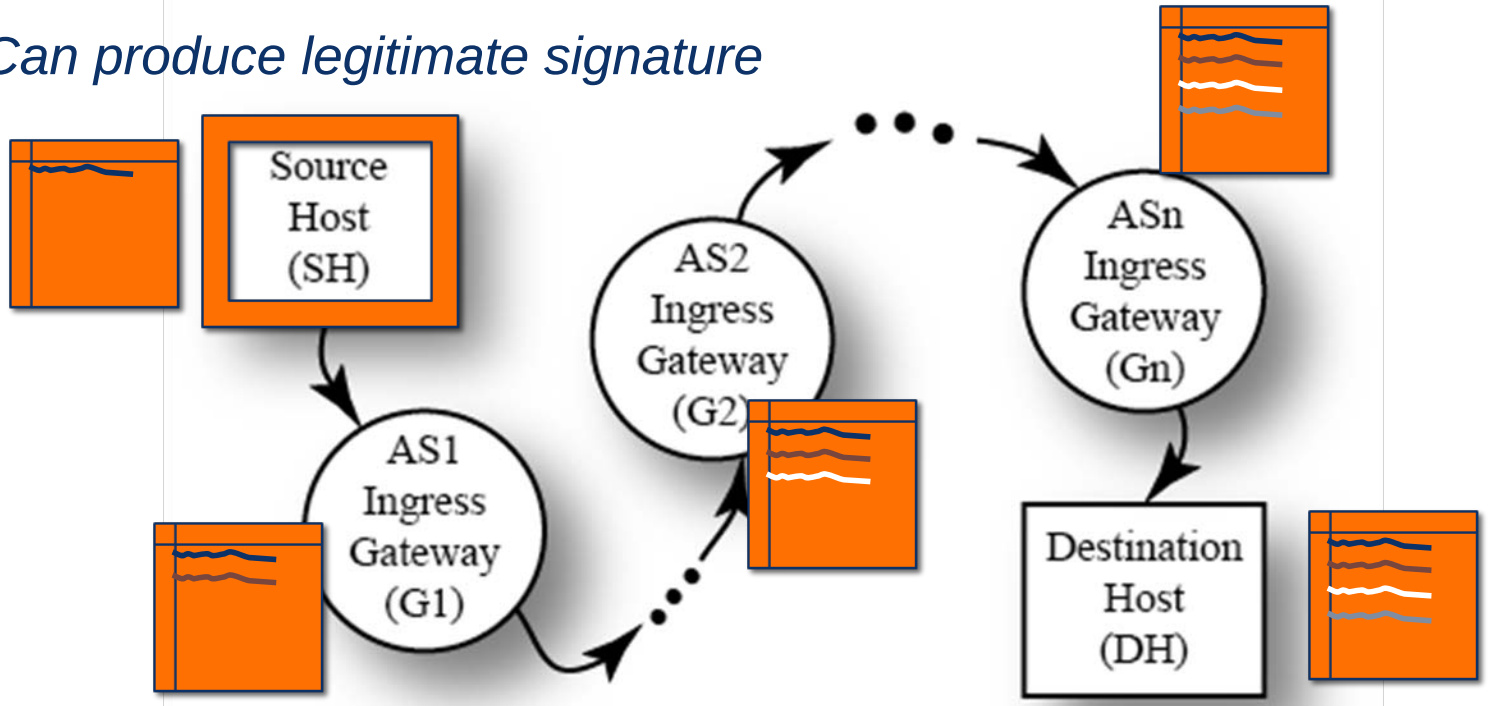
Spoofed Gateway Packet Injection



Bad Signature
- Drop Packet

Compromised Source Host

Can produce legitimate signature

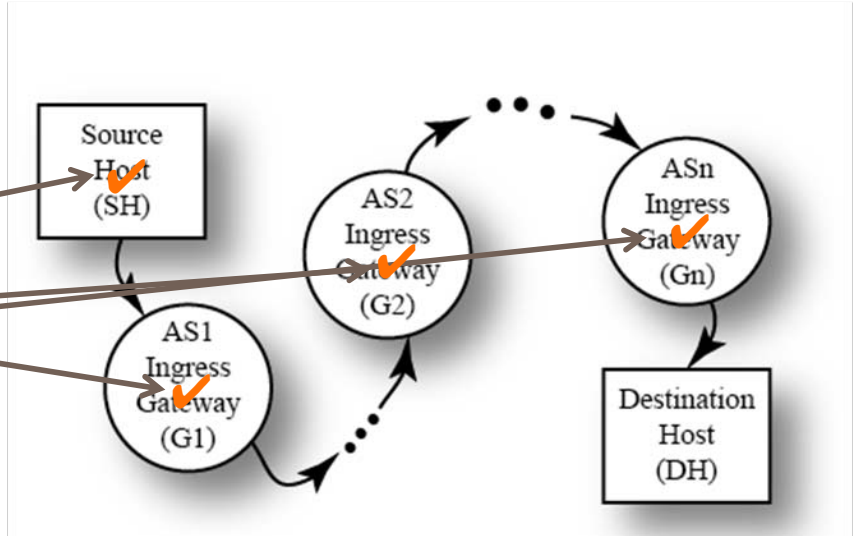


Bad packet reaches destinati
- Forensic review possible

Compromised Host Forensic Review

Legitimate packet with malicious payload

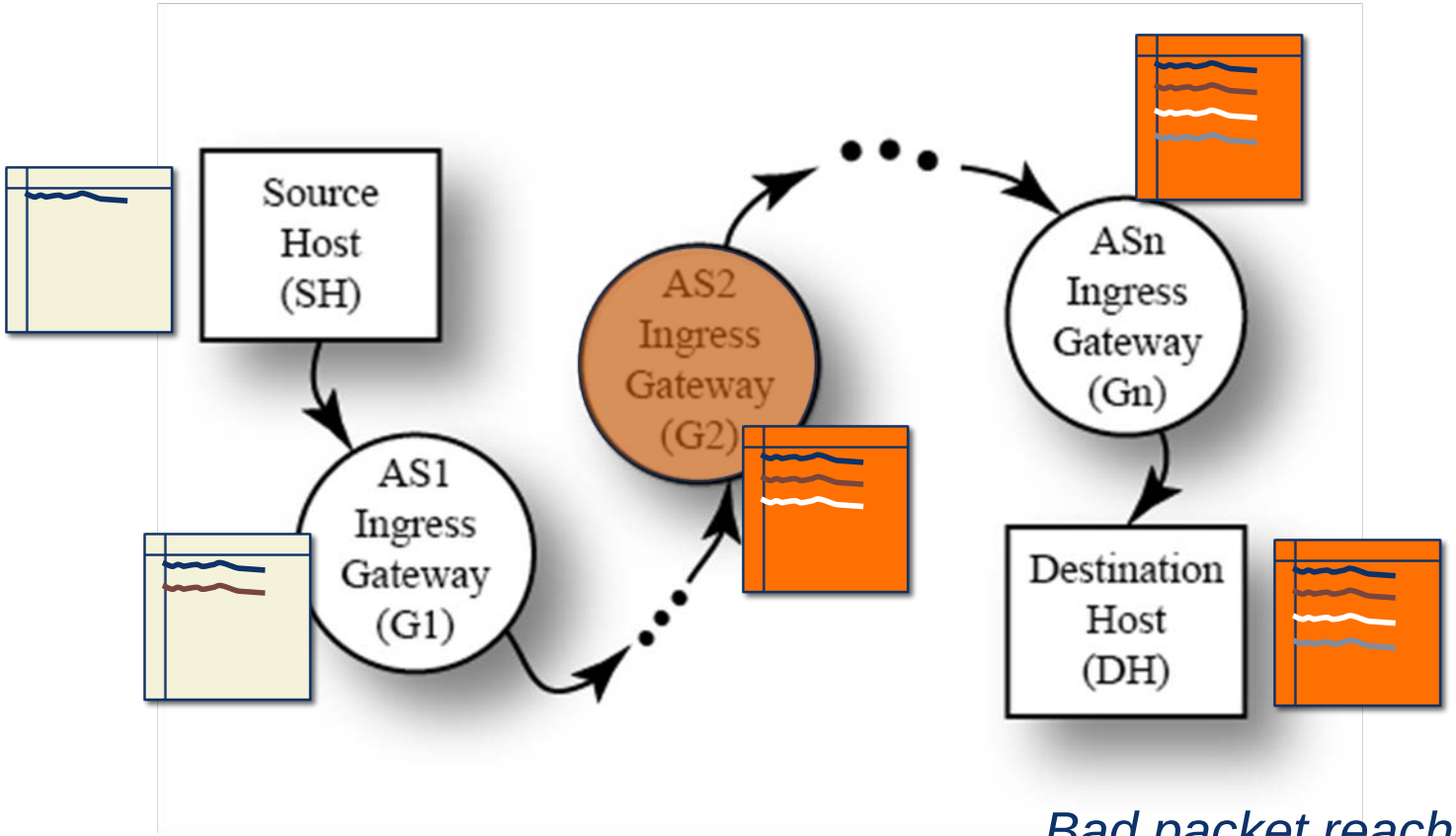
Unroll secure route record



The secure AS path record entries have been validated all the way back to the source

The source host is accountable for the malicious payload

Compromised Gateway Packet Corruption

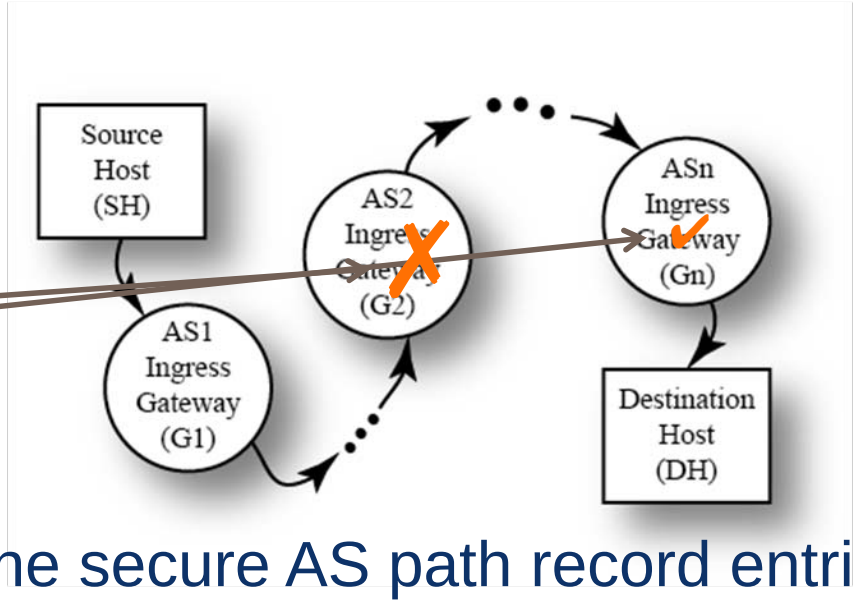


Bad packet reaches destinati
- Forensic review possible

Compromised Gateway Forensic Review

Legitimate packet with malicious payload

Unroll secure route record



The secure AS path record entries could not be validated all the way back to the source

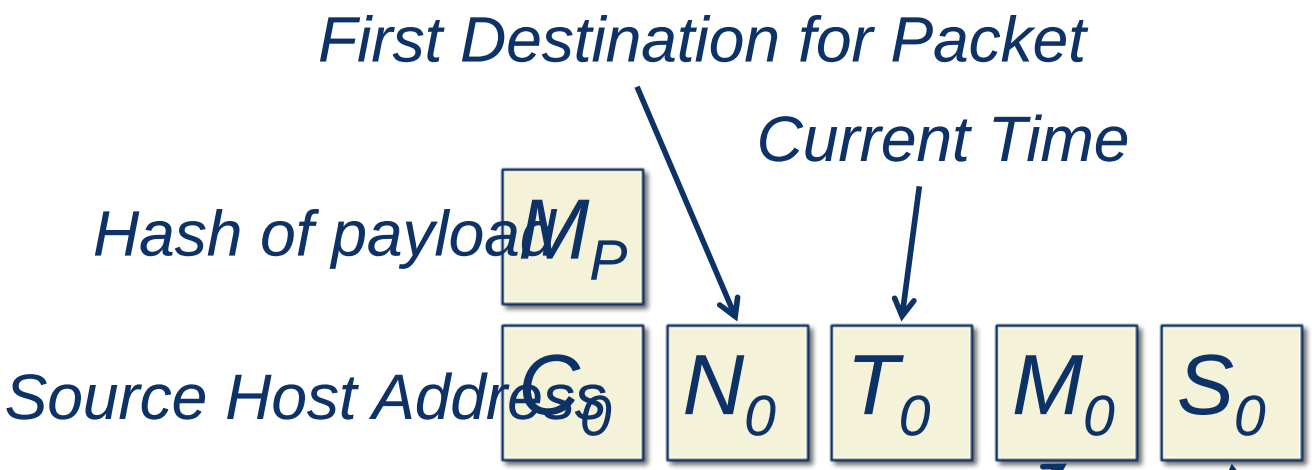
The gateway is accountable for the malicious payload

Operation Details

- The source host calculates a fixed-length message digest value, M_P, for the payload
  - This hash is used to bind the payload to the secure AS path record (SASPR)
  - This hash is a one-way function, but it is not secured by a key

$$M_P = Hash(payload)$$

The source host builds the first entry for the secure AS path record

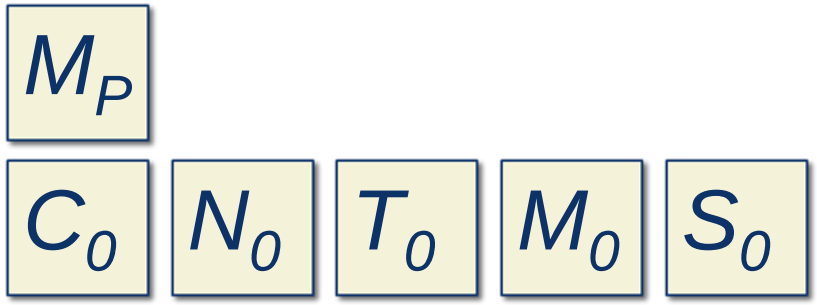


$$M_0 = Hash\left(C_0 \mid N_0 \mid T_0 \mid M_P\right)$$

$$S_0 = Encrypt_0\left(M_0\right)$$

This entry is placed in the packet header to start the secure AS path record

First Ingress Gateway

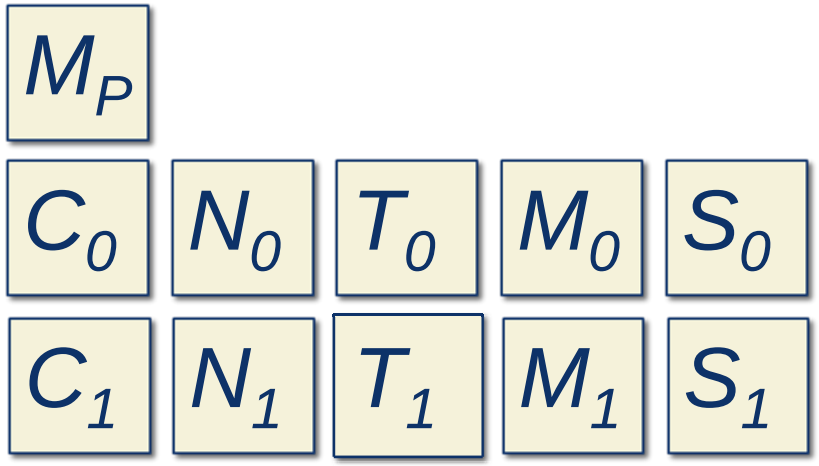


Calculate locally and compare

$$M_0 = Hash(C_0 \mid N_0 \mid T_0 \mid M_P) \qquad M_0 = Decrypt_0(S_0)$$

A match verifies the signature
Drop the packet if no match

Next Entry in the Secure AS Path Record



$$M_1 = Hash\left(C_1 \mid N_1 \mid T_1 \mid M_0 \mid M_P\right)$$

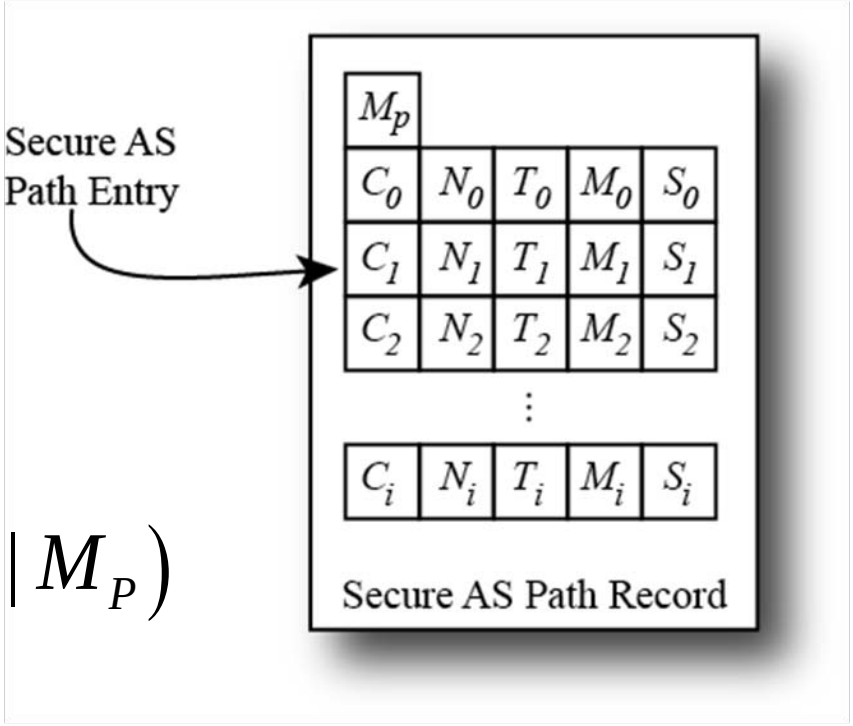
$$S_1 = Encrypt_1\left(M_1\right)$$

Secure AS Path Record

This record resides in the packet header and grows by one entry each time that it is processed by a participating node

$$M_i = Hash\left(C_i \mid N_i \mid T_i \mid M_{i-1} \mid M_P\right)$$

$$S_i = Encrypt_i\left(M_i\right)$$



Implementation Challenges

- This approach requires public key cryptography for signatures
  - A key management infrastructure is required
- Signature and hash computations are required for each header at each participating node

Public Key Infrastructure Challenge

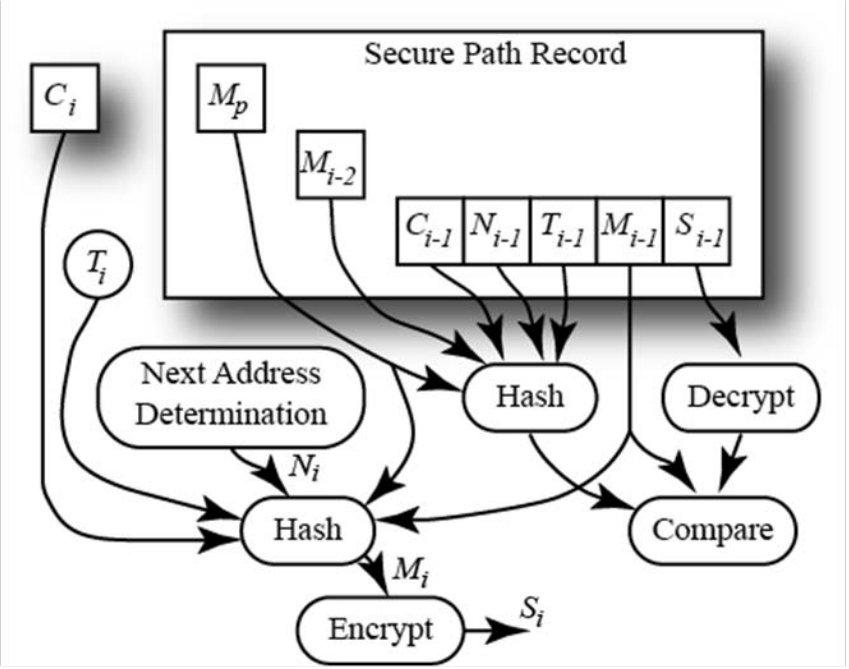
- Partially mitigated by need for participating nodes only to maintain public keys for neighboring nodes
  - Nodes are not required to maintain complete public key list
- Forensic review may require a more extensive public key list
  - This is accomplished after delivery and outside of the regular communication process
- No consideration has yet been given to improvements on PKI for this application

Computation Challenge

- The process is designed to exploit parallel processing to mask some of the processing burden
- Many researchers are working on accelerating the required processing

Parallel Implementation

- Signature validation occurs with hash and decryption operations proceeding in parallel
- Next hash may also proceed in parallel after determination of next hop address
- Next signature encryption operation follows next hash calculation



Computational Parallelism

- Public Key signature algorithms are being developed (by others) that lend themselves to highly parallel implementations
  - Lattice-base NTRU signature algorithm is reported to achieve up to three orders of magnitude performance improvement over ECC and up to five orders of magnitude performance improvement over RSA
  - Efficient hardware implementations should achieve the best possible performance
- Similar efforts are improving hash algorithms
  - Parallel hardware implementations should similarly achieve high performance

Conclusions

- This approach uses standard signature techniques to support early (in the network) detection of packets that cannot be authenticated so that these packets may be dropped early
- Packets that pass the authentication tests can be forensically processed to hold the source of the packet accountable for sending the packet
- The approach involves routers rather than just the source and destination
- The approach faces challenges of PKI and computation, but
  - PKI must only maintain at each node keys for neighboring nodes
  - The approach is structured to maximize potential parallelism of computation