Cryptographic Approaches for Securing Routing Protocols

Adrian Perrig

perrig@cmu.edu

Carnegie Mellon CyLab
PITTSBURGH, PA
www.cylab.cmu.edu

Why Secure Routing?

- Current routing protocols assume trusted environment!
- Even misconfigurations severely disrupt Internet routing
- Secure routing goals
  - Reduce misconfiguration impact
  - Robust against external malicious nodes (no compromised nodes)
  - Robust against compromised nodes (Byzantine failures)

Routing Protocol Attacks

- Current routing protocols are vulnerable
  - Prevent route establishment
  - Attracting traffic (e.g., blackhole attack)
  - Repelling traffic
  - Gratuitous detours
  - Cause route instabilities / route flapping
  - Denial-of-Service (DoS): router overload
    - Almost all attacks appear as DoS attacks, since routing is a service, however, we only consider router resource consumption as routing DoS attacks

Approaches to Secure Routing

- Detection/recovery
  - Use intrusion-detection techniques to detect malicious behavior
- Prevention
  - Use cryptographic techniques to prevent malicious behavior
- Robustness
  - Use robustness techniques to reduce impact of malicious behavior
  - E.g., use multipath routing to improve probability of packet delivery

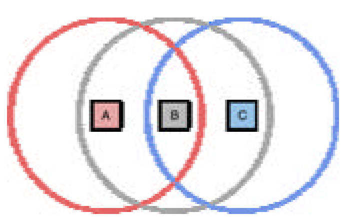
Outline

- **Secure ad hoc network routing protocols**
  - **SEAD: Secure Efficient Ad-hoc network Distance vector routing protocol**
    - **Joint work with Yih-Chun Hu and David Johnson**
    - **Defend against shortening hop count**
- **Secure Internet routing protocols**
  - **SPV: Secure Path Vector**
    - **Joint work with Yih-Chun Hu and Marvin Sirbu**
    - **Secure BGP routing protocol**

Ad Hoc Networks

- **No infrastructure, or out-of-range base station**
- **Devices self-organize to form a network**

- **Ad hoc network routing protocol extends communication range**



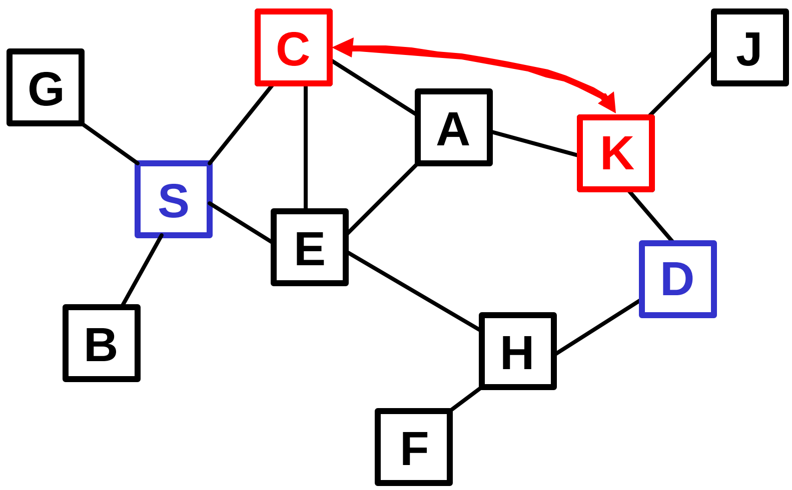
Ad Hoc Network Applications

- Ad hoc networks provide connectivity in various environments
  - Rooftop networks
  - Corporate ad hoc networks
  - Emergency response, disaster relief
  - Devices protecting critical infrastructures
  - Networks of cars relaying safety information
  - Satellite networks in space
  - Military applications

Security Threats to Ad Hoc Networks

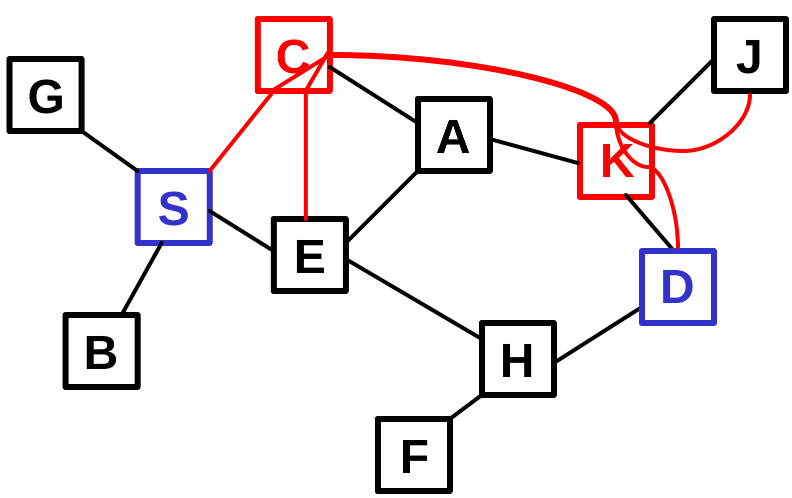
- Wireless communication allows attacker to
  - Eavesdrop on all communication
  - Inject malicious messages into the network
- Current ad hoc network routing protocols designed for trusted environments
  - Highly susceptible to attacks!
  - Skilled attacker can prevent communication
- Sample ad hoc network attacks
  - Wormhole attack
  - Rushing attack

What is a Wormhole?



Nodes C and K open a tunnel

What is a Wormhole?



C and K act as repeaters for their neighbors

Why is that an Attack?

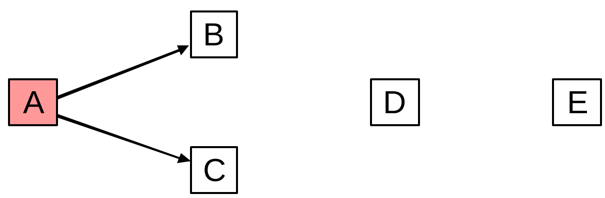
- Routing protocol sees wormhole as a link
- But attacker could selectively forward only routing packets, but not data
- Routing protocol generally chooses route through wormhole because it is the shortest route
- Attacker does not need to compromise any nodes or keys!
- Result: an attacker can cripple the network when using a routing protocol that does not protect against wormholes

Rushing Attack

- In a rushing attack, an attacker exploits duplicate suppression in broadcasts to suppress legitimate packets by quickly forwarding its own packets
- Methods for rushing
  - Forwarding REQUEST without checking signature
  - Using a longer transmission range
  - Ignoring delays specified by the MAC layer
  - "Tunneling" a REQUEST over another medium

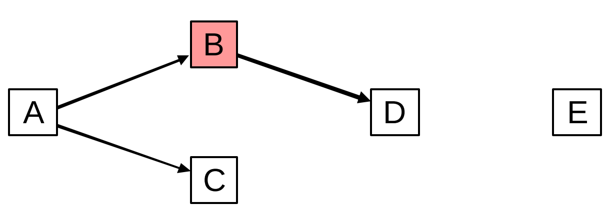
Example Rushing Attack

- A sends a ROUTE REQUEST



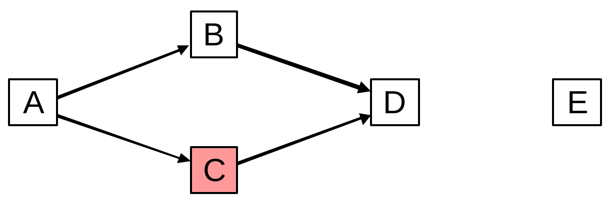
Example Rushing Attack

- A sends a ROUTE REQUEST
- B forwards the REQUEST without checking the signature, or otherwise rushes the REQUEST



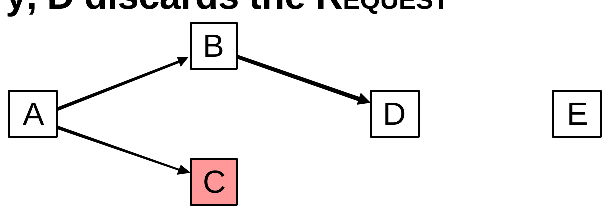
Example Rushing Attack

- A sends a ROUTE REQUEST
- B forwards the REQUEST without checking the signature, or otherwise rushes the REQUEST
- **C correctly processes the REQUEST, and forwards it later as a result**



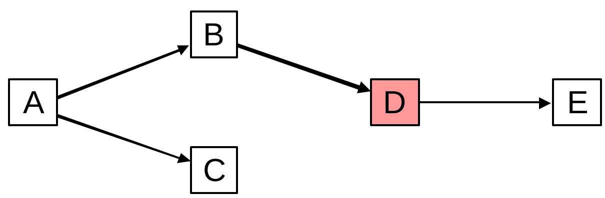
Example Rushing Attack

- A sends a ROUTE REQUEST
- B forwards the REQUEST without checking the signature, or otherwise rushes the REQUEST
- C correctly processes the REQUEST, and forwards it later as a result
- **Since D has already heard a REQUEST from this discovery, D discards the REQUEST**



Example Rushing Attack

- B rushes the REQUEST
- C forwards it later
- Since D has already heard a REQUEST from this discovery, D discards the REQUEST
- **A discovers a path through B because B rushed the REQUEST**



Basic Distance Vector Routing

- **Each node maintains a routing table**

Example table at **A**:

Destination	Metric	Next Hop
A	0	-
B	1	B
C	2	B

- **Computed using Distributed Bellman-Ford**
  - **Each node periodically broadcasts its routing table**
  - **For each routing table entry received, compare best known route with new information**

DSDV: Using Sequence Numbers to Prevent Routing Loops

Adding sequence numbers guarantees loop-freedom:

- Each node maintains a sequence number
- Node increments its own sequence number each time it sends a routing update about itself
- Each update includes sequence number and metric
- An advertised route is "better" if either:
  - It has a greater (more recent) sequence number, or
  - Sequence numbers are equal, and the metric is lower
- Only the most recent sequence number matters

Attacks to defend against: Claim lower metric or higher sequence number

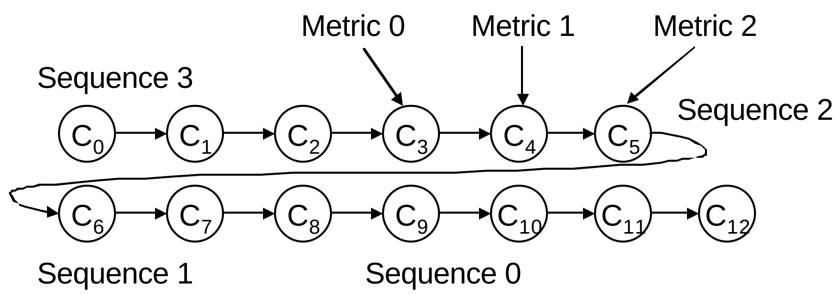
SEAD Protocol Properties

SEAD (Secure Efficient Ad hoc Distance vector):

- Uses one-way hash chains to authenticate metric and sequence number
- Assumes a limit k-1 on metric (as in other distance vector protocols such as RIP, where k=16)
  - Metric value infinity can be represented as k

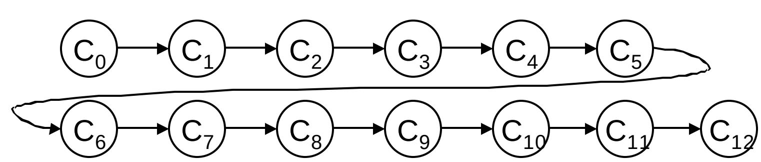
SEAD Metric Authenticators

- Each node generates a hash chain and distributes the last element (C_{12}) for verification
- Each sequence number has 3 hash chain values
- Within a sequence number
  - $C_{\{0,3,6,9\}}$ represent metric 0
  - $C_{\{1,4,7,10\}}$ represent metric 1
  - $C_{\{2,5,8,11\}}$ represent metric 2



SEAD Metric Authenticator Properties

- SEAD metric authenticator prevents blackhole attack
  - Assume all nodes know authentic C_{12}
  - Consider source announces C_9 for metric 0
  - Neighbor announces C_{10} for metric 1
  - Attacker cannot announce lower metric!
  - Due to flooding, useless to announce lower metric with lower sequence number

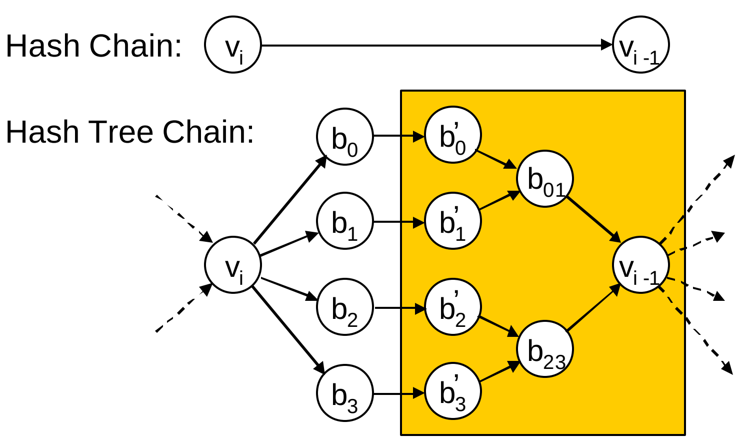


Remaining Problems

- **"Same Metric" Fraud attack**
  - **Attack: Replay metric and authenticator attacker hears**
  - **Solution: Tie forwarding node address to authenticator**
- **Denial-of-Service attack:**
  - **Attack: Claim a very high sequence number**
  - **Solution: Each sequence number gets own chain**
- **Larger metric spaces:**
  - **Verifying even one sequence number may be expensive (e.g., if metric is based on latency or policy)**
  - **Solution: Cheaper hash-chain following**

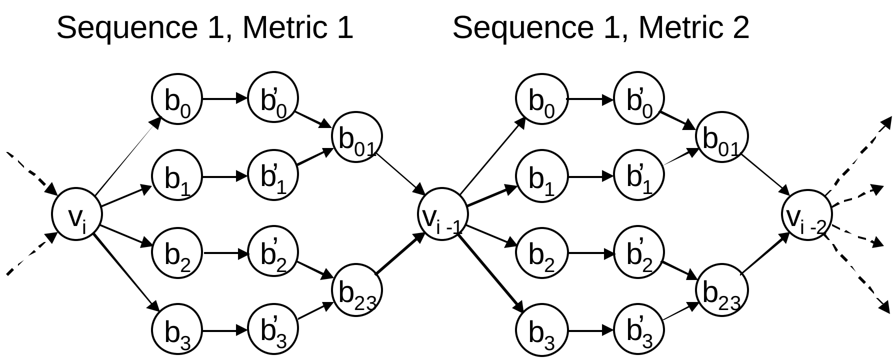
Hash Tree Chains

- **Each step in a hash tree chain is a one-time signature**



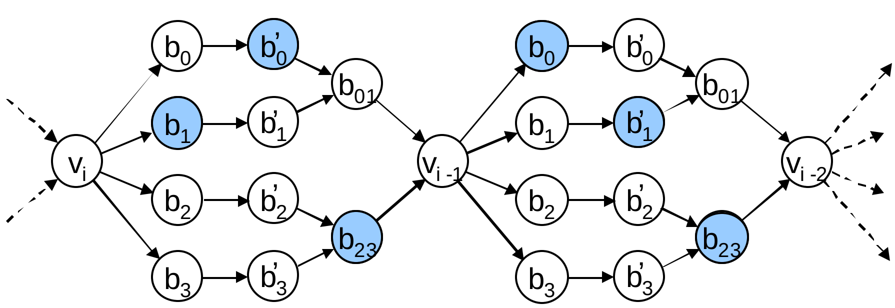
Using Hash Tree Chains

- **As before, one step in the one-way chain corresponds to a** *(sequence number, metric)* **pair**



Using Hash Tree Chains

- **As before, one step in the one-way chain corresponds to a** *(sequence number, metric)* **pair**
- **Each b_i corresponds to a forwarding node**
- **Attacker must gather correct b_i to replay metric**



SPV: Secure Path Vector Routing

- Joint work with Yih-Chun Hu and Marvin Sirbu
- Presented at ACM Sigcomm 2004
- SPV adds security to BGP routing protocol
  - Use of highly efficient one-way function to provide security
  - Key insight: authentication of autonomous systems on path not necessary

BGP Essentials

- BGP is Internet's interdomain routing protocol
  - Destinations are prefixes (CIDR blocks)
  - Route includes list of autonomous systems (AS)
- A path vector protocol
  - Each AS maintains routes to each prefix
  - It advertises a (potentially different) subset of those routes to each of its peers
  - Each advertised route includes an ASPATH attribute (a list of ASes the route traverses)

Three Important Attacks

- Unauthorized AS advertises a prefix
  - E.g., small ISP advertises Google's prefix
  - ASes closer to the small ISP than to Google will send Google's packets to the ISP
- ASPATH truncation
  - Reduces ASPATH length, causing downstream AS to prefer attacker's route
- ASPATH alteration
  - Remove undesirable ASNs from the path to cause downstream ASes to prefer attacker's route

S-BGP (Kent et al.)

S-BGP checks two things:

- Originating AS is authorized to advertise prefix
- Each AS receives delegation from previous AS

Requires identification of delegating AS

Disadvantages:

- S-BGP requires the use of computationally expensive digital signatures
  - Signing is 10,000 times slower than one-way function
  - Verification is 1,000 times slower
- Poor incremental deployment properties

Our Key Observation

- SPV protects the ASPATH by:
  - Each AS checks that the previous AS correctly inserted its own ASN in the ASPATH
  - Using cryptography to make unauthorized ASPATH modification difficult

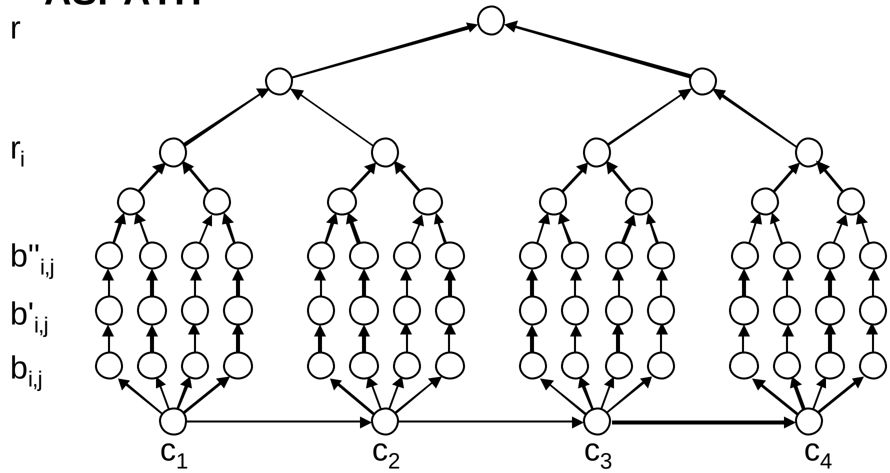
- High levels of security can be achieved without identifying the AS that inserts an ASN:
  - Append-only ASPATH

- SPV protects the ASPATH by:
  - Each AS checks that the previous AS correctly inserted its own ASN in the ASPATH
  - Using cryptography to make unauthorized ASPATH modification difficult

- Properties:
  - Without breaking the crypto, a node cannot change or remove ASNs from ASPATH
  - Desirable incremental deployment properties
  - However, collaborating attackers can insert bogus ASNs between themselves

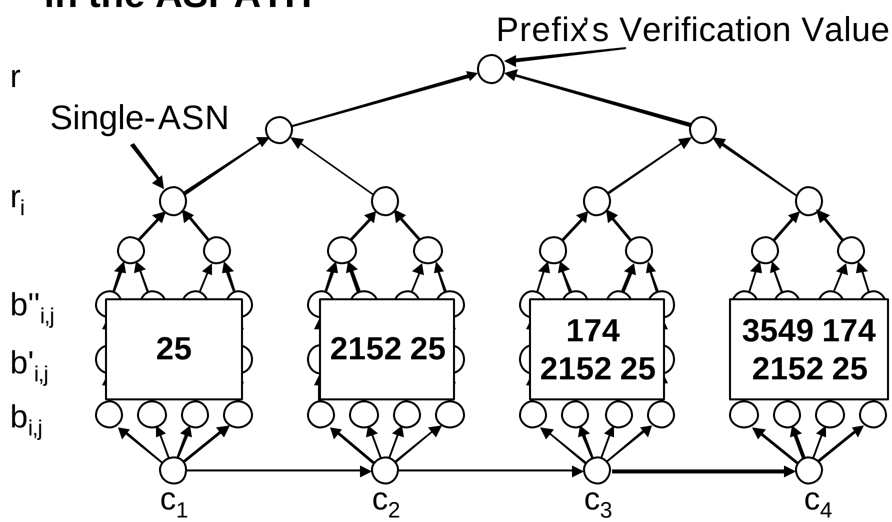
Our ASPATH Protector

- The goal of the ASPATH protector is to prevent an attacker from modifying the encoded ASPATH



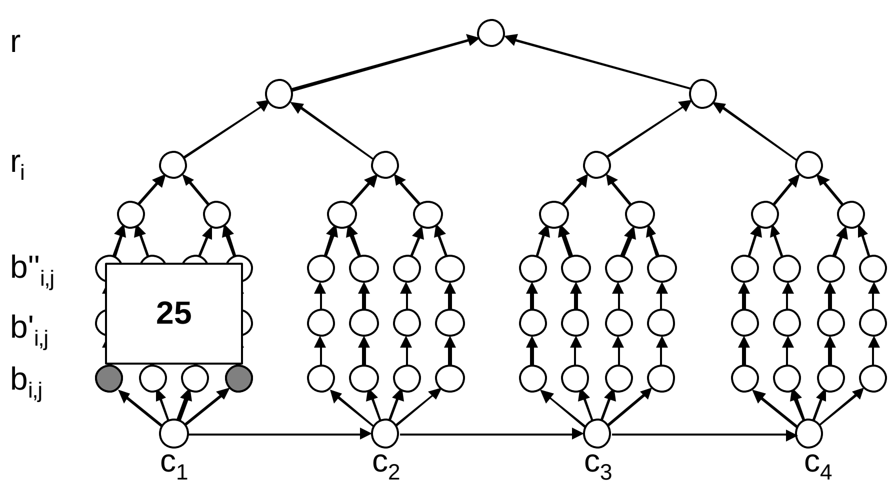
Our ASPATH Protector

- **Each one-time signature signs a single position in the ASPATH**



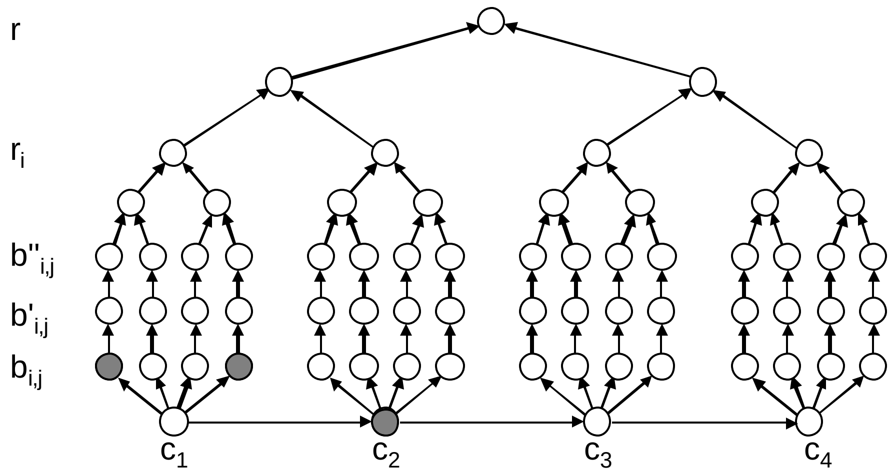
Using the ASPATH Protector

- **Originating AS encodes its ASN**
  - **Disclose lower values based on H(ASPATH)**



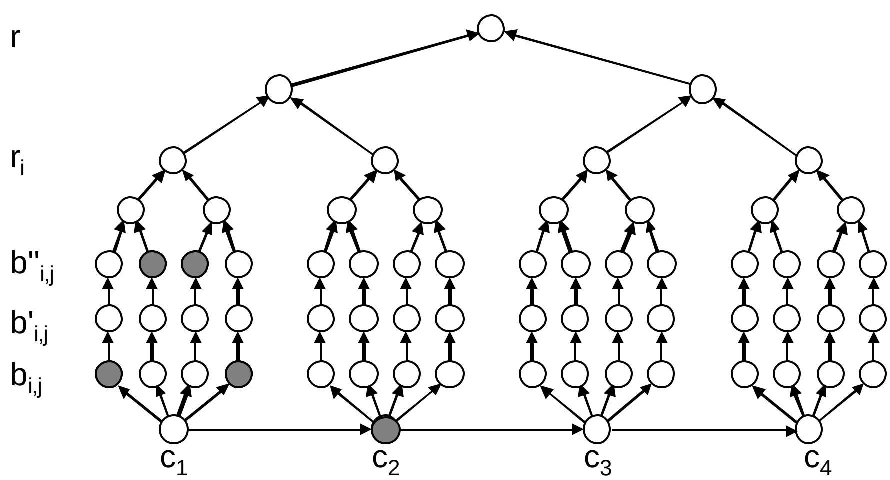
Using the ASPATH Protector

- **Originating AS encodes its ASN**
  - **Disclose next signing key**



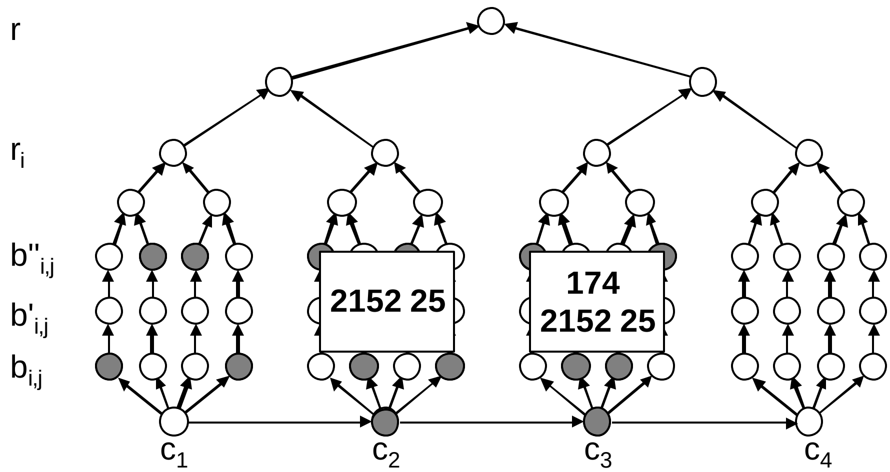
Using the ASPATH Protector

- **Originating AS encodes its ASN**
  - **Disclose upper values needed to verify**



Using the ASPATH Protector

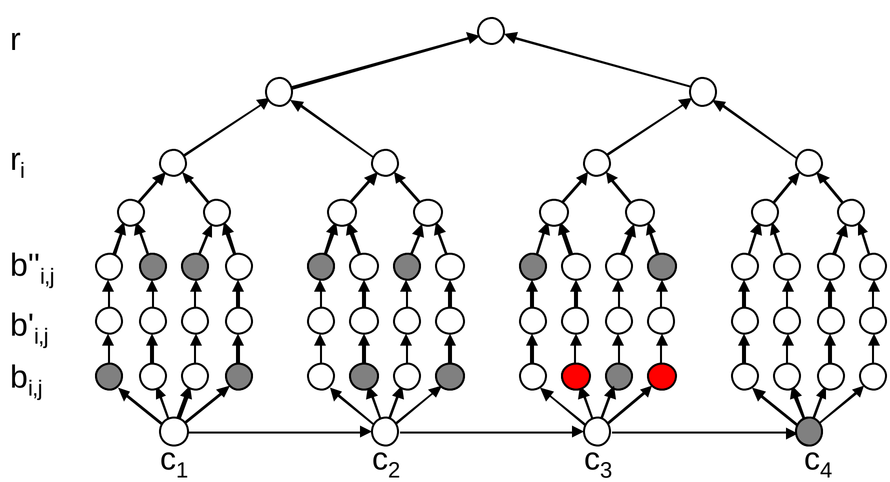
- Originating AS encodes its ASN
- Each AS in turn encodes its ASN



ASPATH Protector Security

An AS receives 128.32.0.0/16 along 174 2152 25

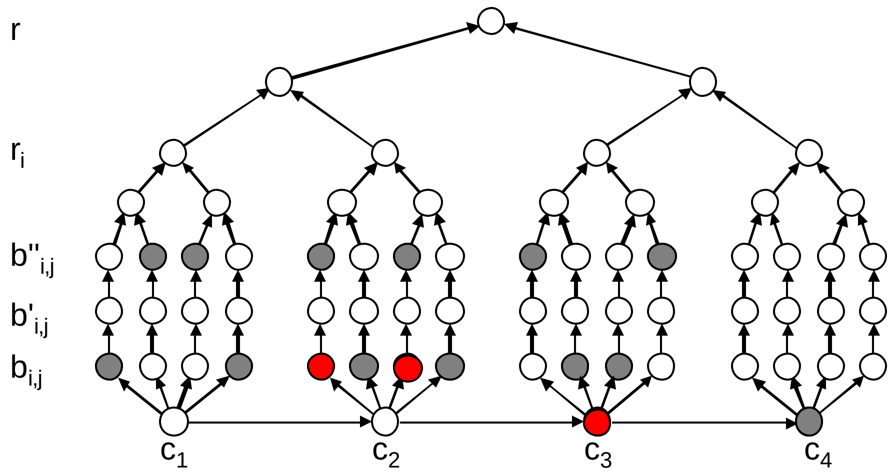
- To change the last AS from 174 to 123:



ASPATH Protector Security

An AS receives 128.32.0.0/16 along 174 2152 25

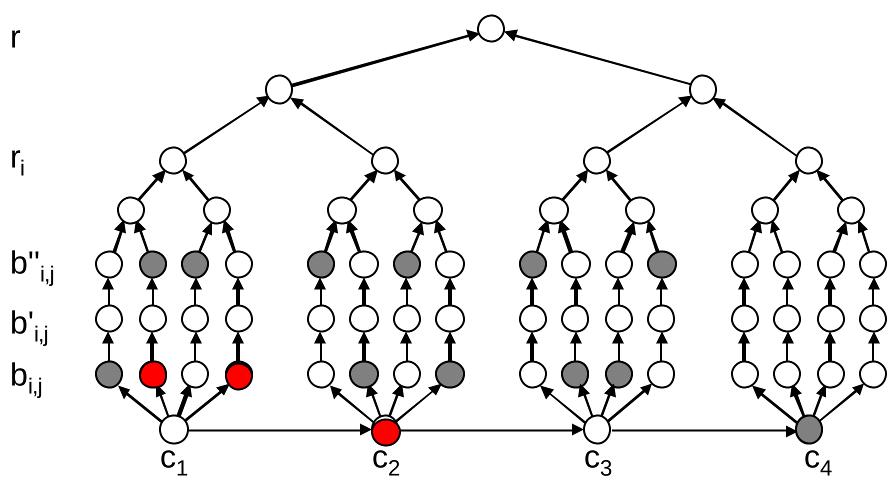
- **To truncate by removing ASes 174 and 2152:**



ASPATH Protector Security

An AS receives 128.32.0.0/16 along 174 2152 25

- **To originate a route to 128.32.0.0/16:**

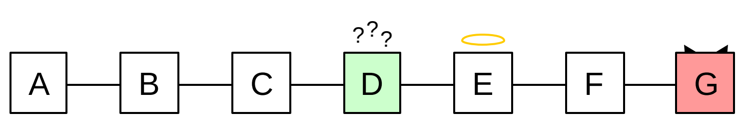


How Much Security is Needed?

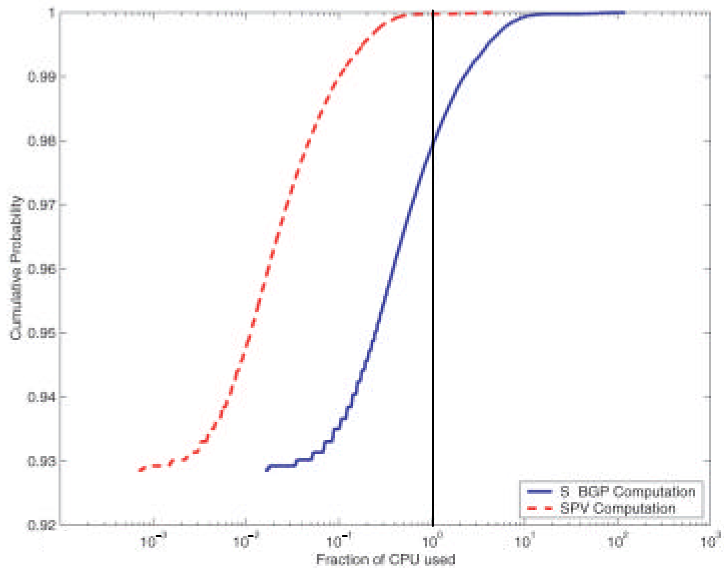
- Security can be measured in the amount of effort required to break the scheme
  - E.g., on average, given an 80-bit value *x*, you need to perform 2^{79} hash operations to find *y* such that H(*y*) = *x*, if H returns 80-bit values
- SPV uses large structures; to provide such high assurances requires too much overhead
  - Resulting UPDATEs are over the 4k limit
- However, there are only 2^{16} possible ASNs, which limits the useful work an attacker can do
- So, SPV attacks are cheap but rarely possible

Incremental Deployment

- What if an intermediate AS doesn't deploy a secure version of BGP?
- If D is non-deploying but E is legitimate:
  - In S-BGP, G can remove E and add arbitrary ASNs after D
  - In SPV, E will have included D in the ASPATH protector, so it's as if D had deployed SPV



Computational Cost



Conclusion

- **Almost all networking protocols were designed for trustworthy environments, now time has come to secure them**
- **Secure routing is an exciting area where we can apply our crypto protocols**