



Security Aspects of IPv6 Multi-Interface

Eric Vyncke evyncke@cisco.com [@evyncke](https://twitter.com/evyncke)

Distinguished Engineer

March 2016

Agenda

- **Multi-interface (MIF)**
 - Description and IETF status
 - Security issues and some solutions
- **Other topics**
 - Proof of service chaining
 - Extension headers on the Internet

Multi-Interfaces

Back to Basics

The Internet Protocol

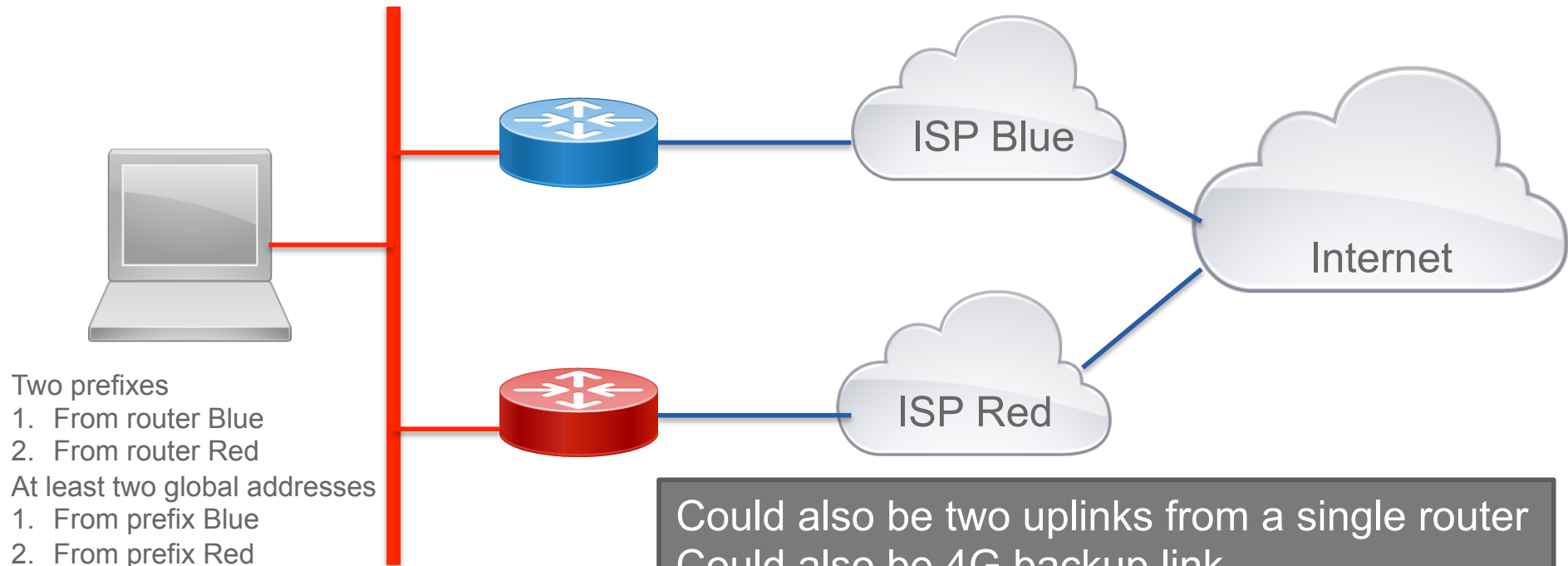
IPv4: 192.168.1.1

IPv6: FE80::0202:B3FF:FE1E:8329
2001:db8:abba:babe::1234
fd00::1:3060:2a08:1505:f6ca

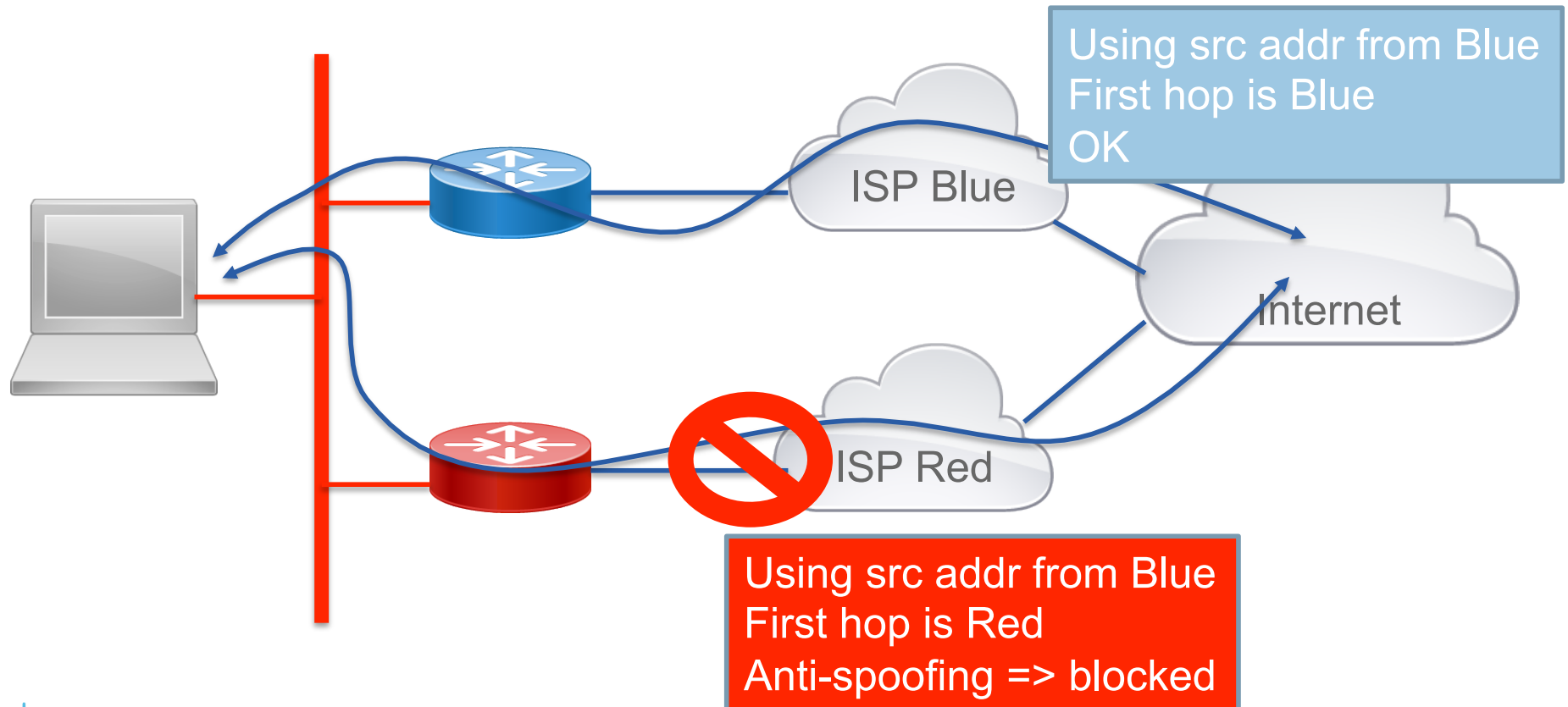
IPv6 Nodes have Multiple Addresses

- Each IPv6 nodes can have multiple addresses
 - The mandatory Link-Local Address
 - Several optional Global Addresses
 - Through DHCPv6 which can give multiple addresses
 - Through Stateless Address Auto Configuration (SLAAC)
 1. Based on several distinct Router Advertisements from each adjacent IPv6 routers
 2. Each Router Advertisements can include multiple /64 prefixes
 3. Nodes then generate 1, 2, ... Addresses per prefixes (privacy extension & EUI-64)

Simple use case: Multi-Homing (Resiliency)



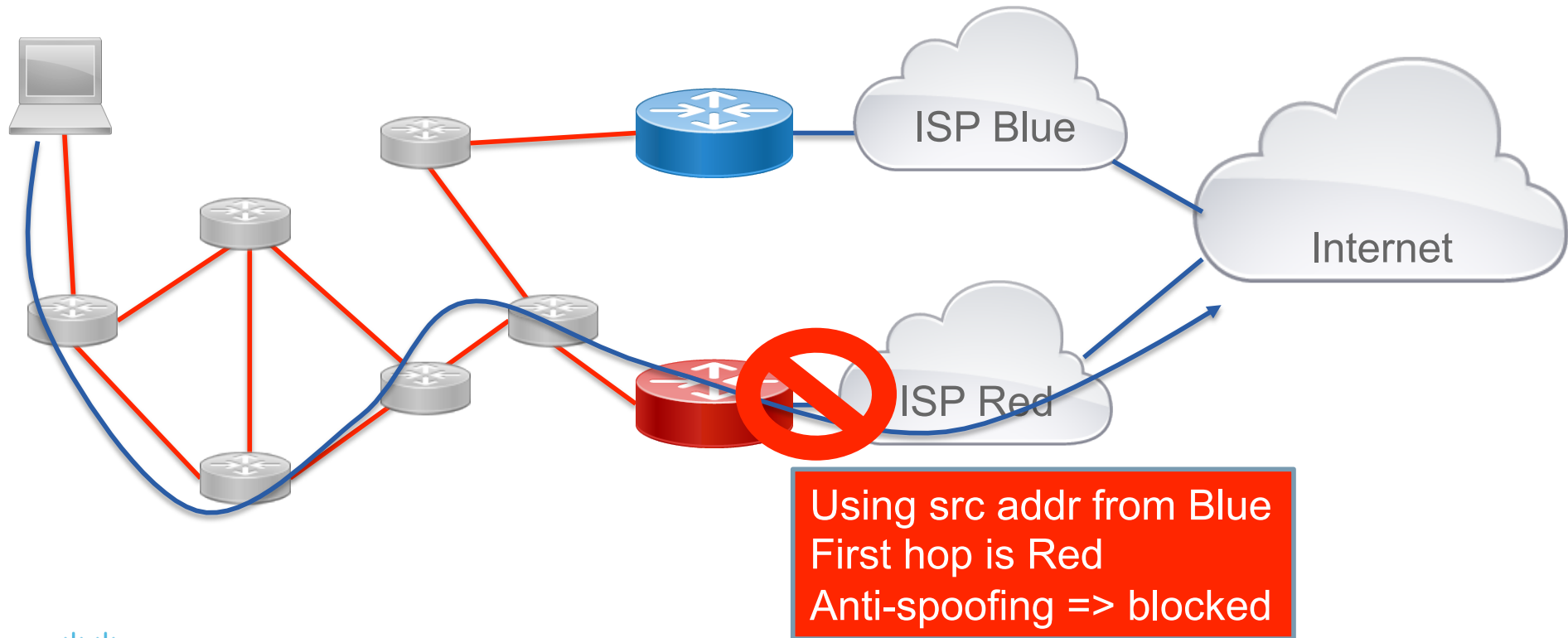
Issue with Multi-Homing (Resiliency)



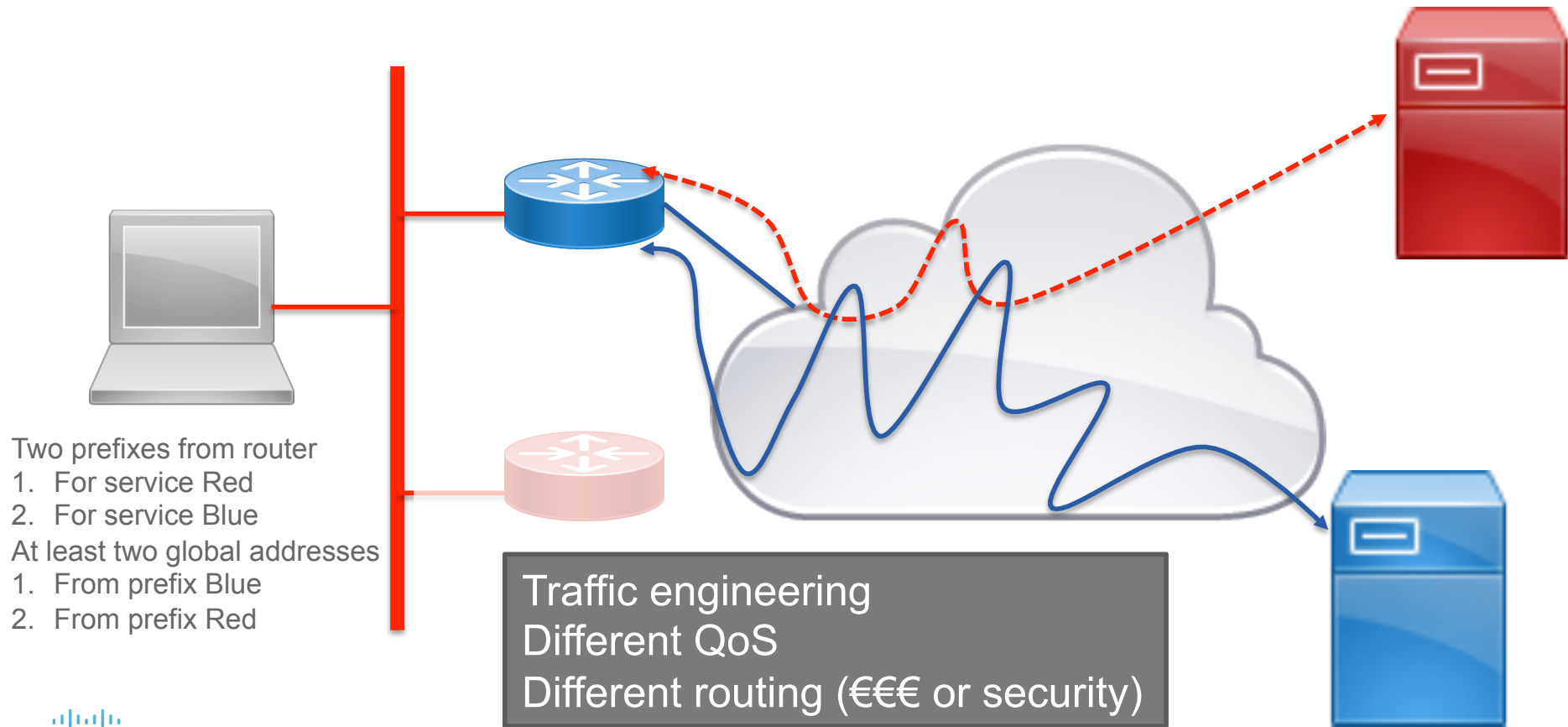
Solving the First Hop Issue

- Need to associate a prefix with first hop
- Mainly a host issue (IETF work in progress)
- Could have multiple layers of routers
 - **Source / destination routing** (IETF work in progress)

Need for **Source and** Destination Routing

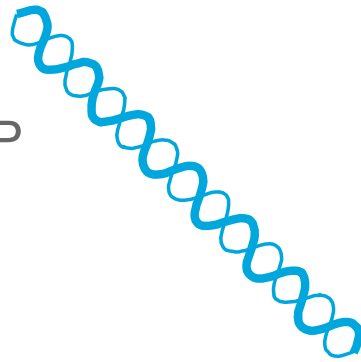


Another use case: Service Selection



Multi-Interfaces for mobile

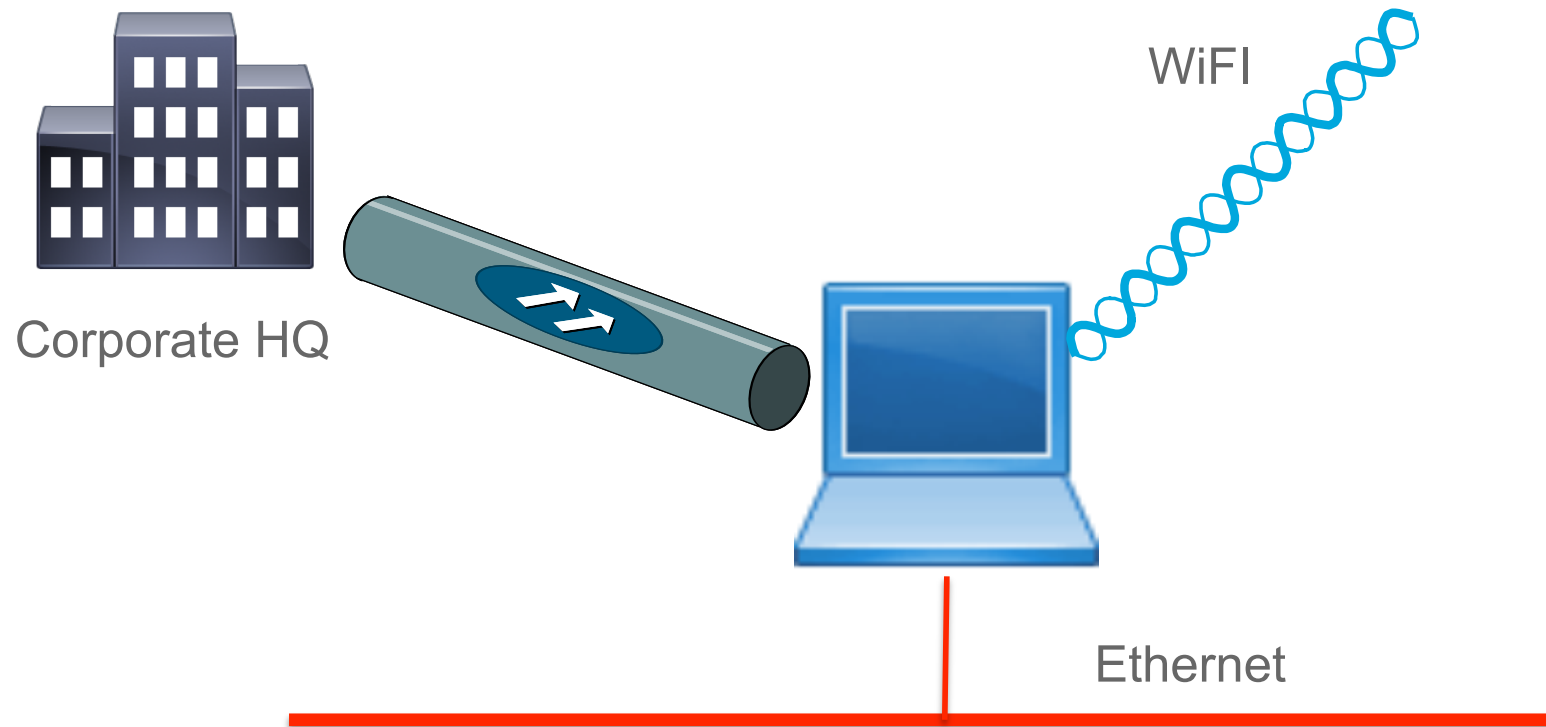
4G to mobile ISP



WiFi to Company WiFi,
Public hotspot



Multi-Interface for desktop...



Provisioning Domain: PvD (RFC 7556)

- Each connection has:
 - A specific source prefix
 - A specific next hop to default route
 - A specific DNS server and default search domain
 - HTTP proxy if any
- This is the ProVisioning Domain (PvD)
 - Dual-stack
 - ??? Huh ??? IPv6 prefix are globally unique... Not IPv4!
 - 192.168.0.100/24 over WiFi is not the same as 192.168.0.100/24 over 3G or LAN
 - Scoped by interface
 - Lifetime as long as the interface is up + any lifetime linked to PvD discovery protocol
 - PvD ID is 'assumed' to be globally unique
 - ??? Huh "assumed", are you serious????

Default route or route to a specific prefix ?

DNS server for all FQDN or only for part of them?

How to configure PvD? IETF Work in Progress

- IKEv2 can securely provision a lot ;-)
 - But can also redirect www.piratebay.com to your HR :-O
- DHCP (draft-ietf-mif-mpvd-dhcp-support)
 - *"The PVD authentication and authorization option contains information that **could be** used by the DHCPv6 client"*
 - Signature on the payload, passed as opaque by the DHCPv6 server/relay
- NDP (over RA) (draft-ietf-mif-mpvd-ndp-support)
 - The signature is linked to a provisioning domain identity
 - Content secured with the help of SeND

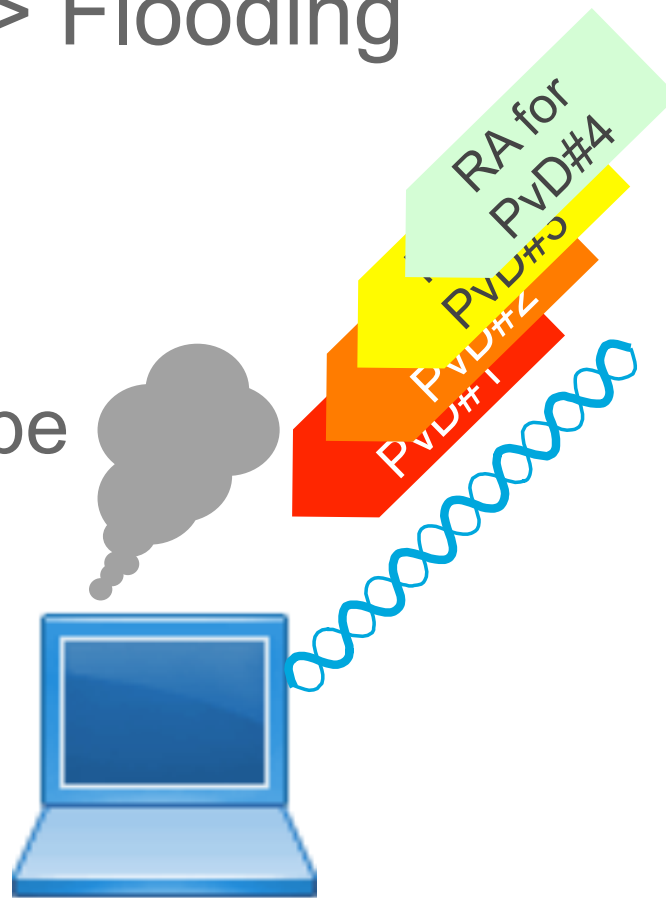
Other References

- <http://homenetting.blogspot.no/2013/09/ipv6-multi-homing.html>
- <http://homenetting.blogspot.be/2013/10/ipv6-multi-prefix-multi-homing-take-2.html>
- draft-ietf-mif-mpvd-dhcp-support
- draft-ietf-mif-mpvd-ndp-support
- draft-lamparter-rtgwg-dst-src-routing
- draft-baker-ipv6-ospf-dst-src-routing
-

Security Issues of MIF

Using RA for PvD => Flooding

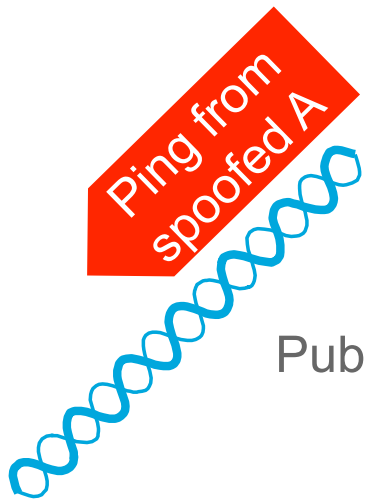
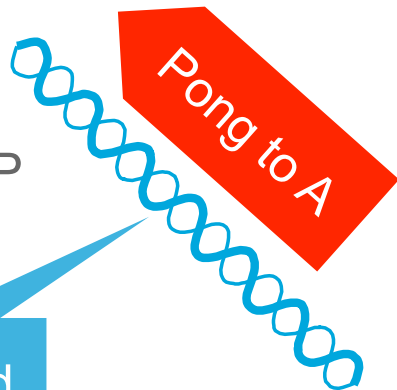
- Similar to the old Windows RA flood...
- RA provisioning can be used to flood



MIF: Reflection Attack

4G to mobile ISP
€€€€€€

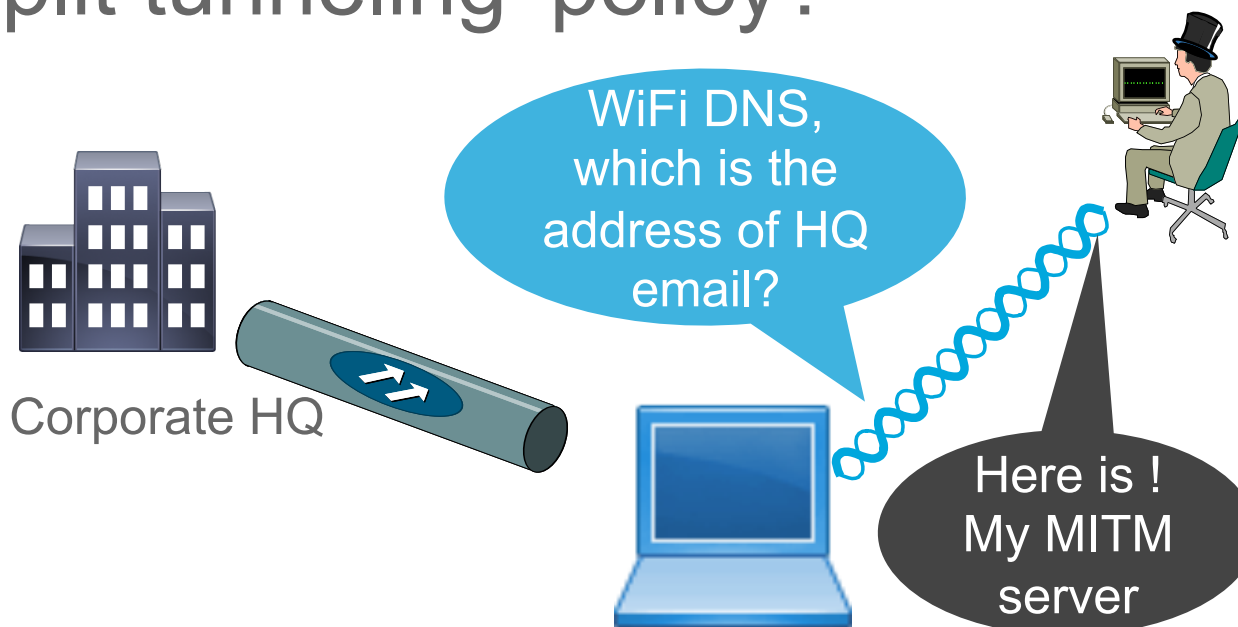
Preferred interface to prefix A



Public hotspot

Remember 'no split tunneling' policy?

- Could be played with DNS or routing...
- Even without MMI, information is leaked about private DNS content



PvD provisioning with NDP & DHCP

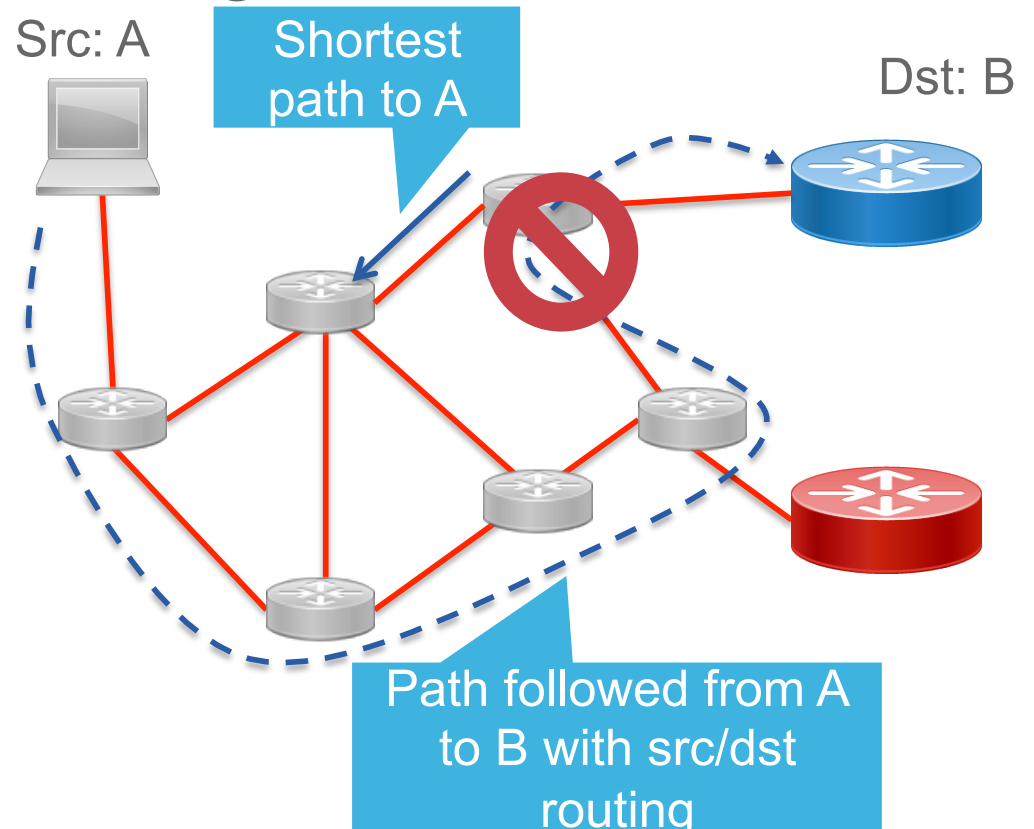
- Authentication is written as 'could' 😞
 - Trivial to inject a wrong PvD
- Moreover basically no anti-replay

PvD via DNS

- IPv6 addresses are globally unique
 - How can we leverage this?
- draft-stenberg-mif-mpvd-dns
 - Use your reverse DNS TXT/PTR request to get information
 - Transparent to ISP, CPE, ...
 - Pull model (no DoS via flooding)
 - DNSsec is your obvious friend
 - NPTv6 is your enemy of course

What about Source Routing ?

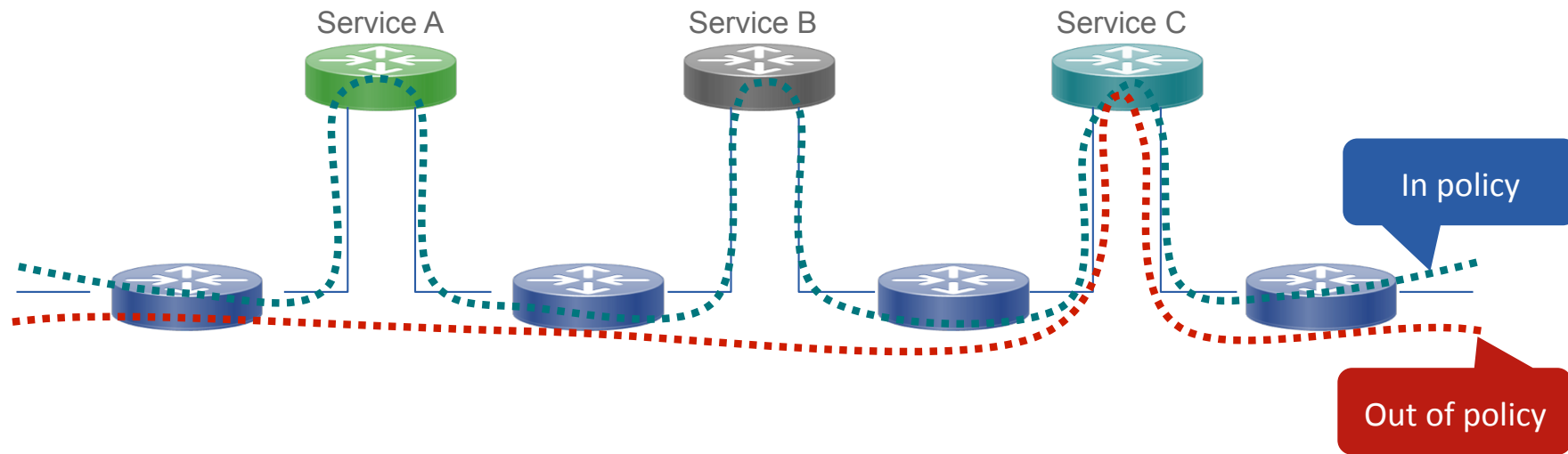
- Source Routing is really useful... But, shortest path not always taken
 - => strict uRPF is no more correct ☹
 - => cannot use uRPF-check for anti-spoofing
- Currently 'tbd'
 - uRPF could leverage scr/dst routing



Extension Headers for iOAM

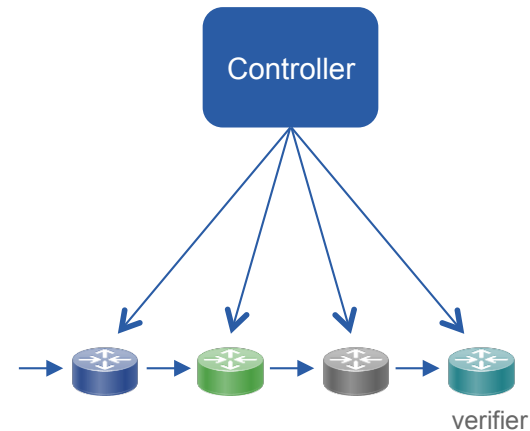
Ensuring Service Chain and Path Integrity

Service Chain: A → B → C

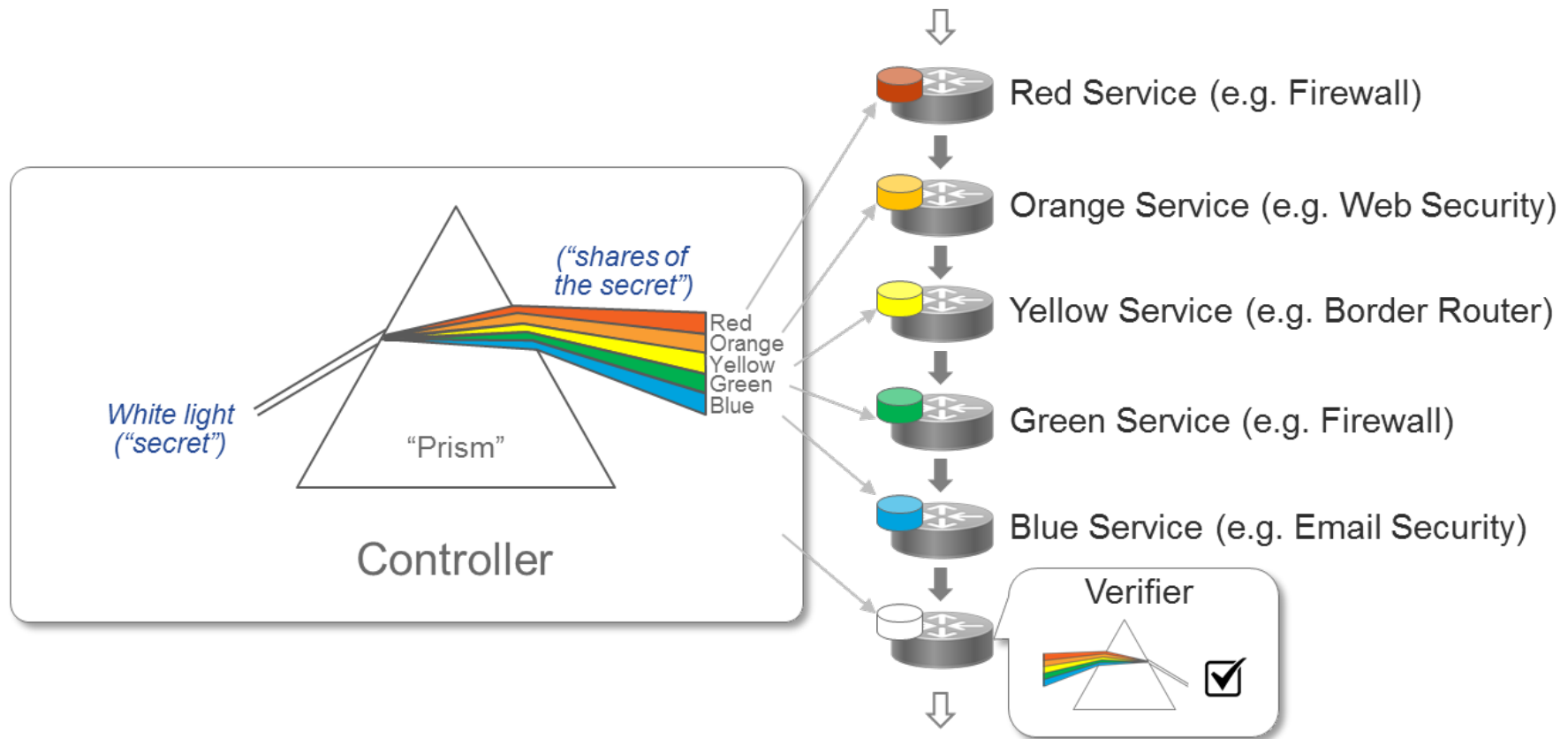


Service Chain Integrity Validation: Approach

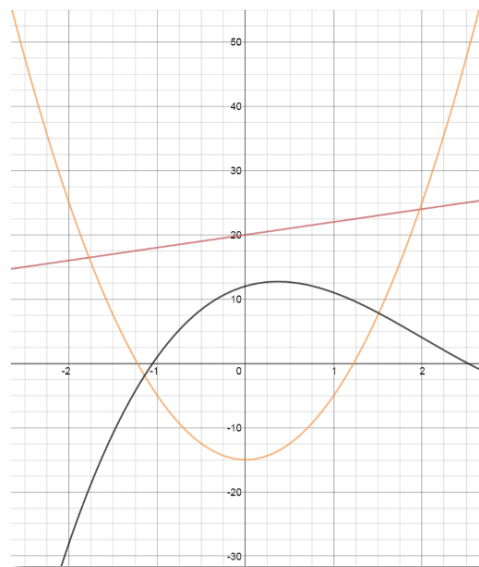
- Add meta-data to all packets that traverse a path or service chain
- The added meta-data allows a verifying node (egress node) to check whether a packet traversed the service chain correctly or not
- Security mechanisms are used on the meta-data to protect against incorrect or misuse (i.e. configuration mistakes, people playing tricks with routing, capturing, spoofing and replaying packets).



Service Chain Integrity Validation Concept



Solution Approach: Leveraging Shamir's Secret Sharing Polynomials 101



$$f2(x) = 10x^2 - 15$$

- Parabola: Min 3 points

$$f1(x) = 2x + 20$$

- Line: Min 2 points

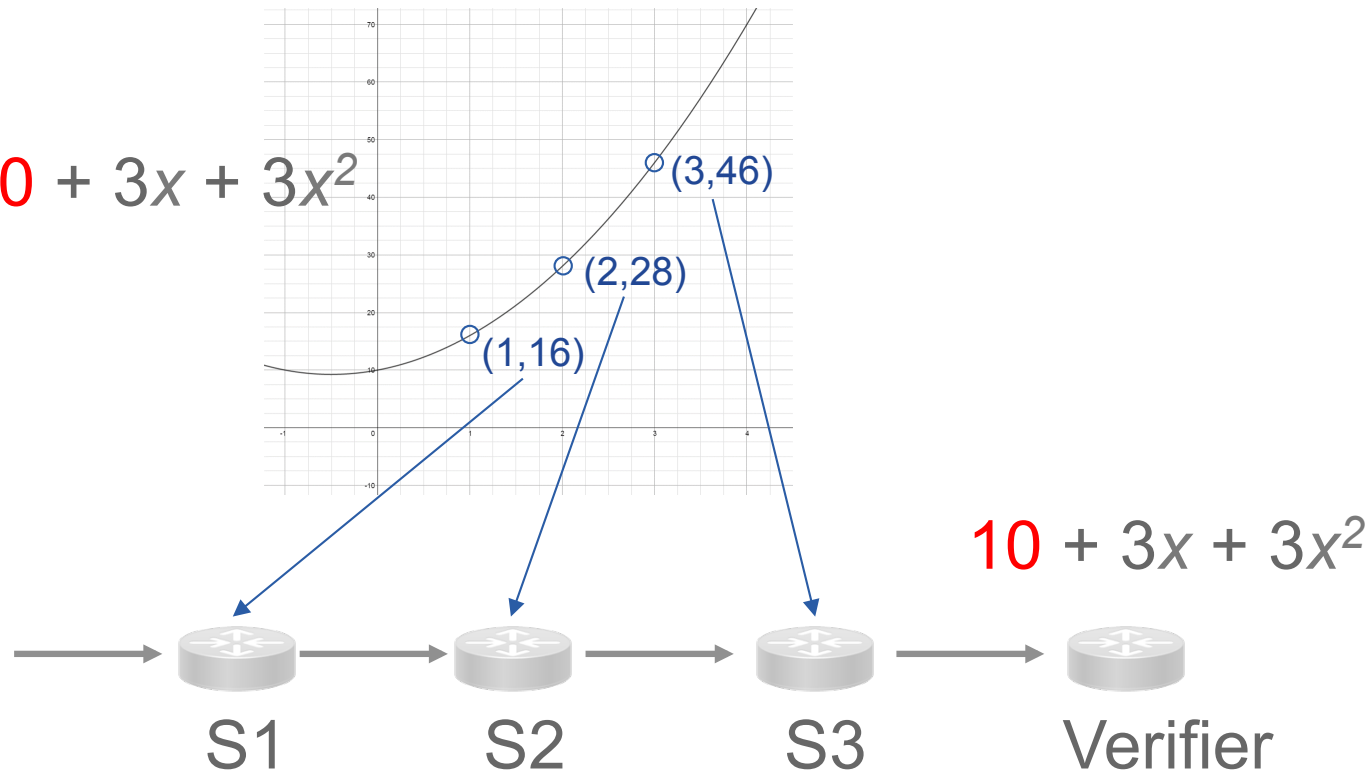
$$f3(x) = x^3 - 6x^2 + 4x - 12$$

- Cubic function: Min 4 points

General: It takes $k+1$ points to defines a polynomial of degree k .

Solution Approach: Leveraging Shamir's Secret Sharing Idea Concept


“Secret”: $10 + 3x + 3x^2$



Solution Approach: Leveraging Shamir's Secret Sharing

- Outline :
 - Each service is given a point on the curve
 - When the packet travels through each service it collects these points
 - A verifier can reconstruct the curve using the collected points
 - If there are $k+1$ services and $k+1$ points chosen, then the verifier can construct k degree polynomial and verify.
 - The polynomial cannot be constructed if a few points are missed. Any lesser points means few services are missed!
- Concern: Operationally complex to configure and recycle so many curves and their respective points for each service function

Simpler & Faster with 2 Polynomials

- POLY-1 secret, constant per chain:
 - $a_1 + b_1x + c_1x^2 + \dots$ (only known by verifier)
 - Each service gets a point on POLY-1 (for $x = 1, 2, \dots$)
- POLY-2 public, with **RND-2** random and per packet
 - $\text{RND-2} + b_2x + c_2x^2 + \dots$ (known by all services + verifier)
 - Each service generates a point on POLY-2 each time a packet crosses it (same x as in POLY-1)
- Each service adds the two points to get a point on POLY-3 and passes it to verifier by adding it to each packet.
- The verifier constructs POLY-3 from the points given by all the services and cross checks whether $\text{POLY-3} = \text{POLY-1} + \text{POLY-2}$
- *Computationally efficient: Only 3 additions and 1 multiplication per hop*
-  *All operations are done in a finite field (modulo prime)*

POLY-1
Secret – Constant

+

POLY-2
Public – Per Packet

=

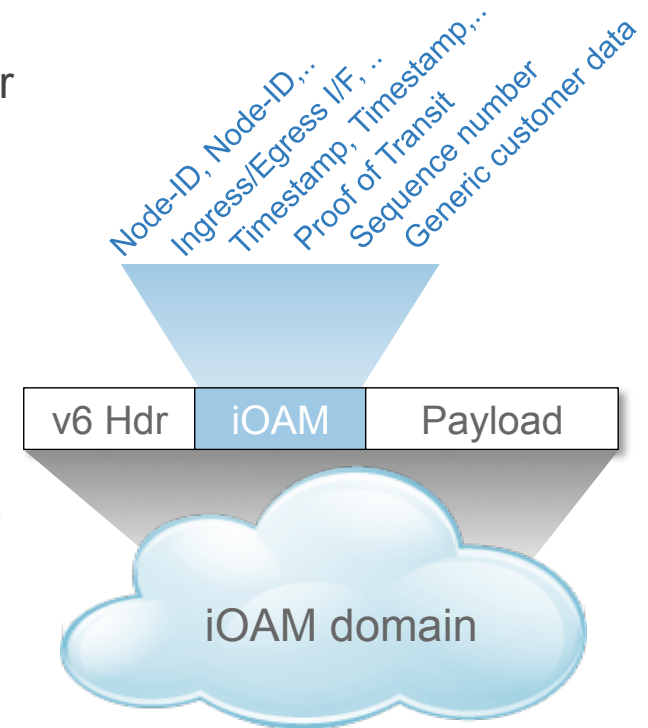
POLY-3
Secret – Per Packet

Security Considerations

- An attacker by passing few services, will miss adding a respective point on POLY-1 to corresponding point on POLY-2 , thus the verifier cannot construct POLY-3 for cross verification
- An attacker watching values, doing differential analysis across service functions (i.e. as the packets entering and leaving), cannot construct a point on POLY-1 as the operations are done over a finite field (i.e. modulo prime).
- Replay attacks could be avoided by carefully choosing POLY-2. It could be a timestamp concatenated with a random string.
- The proofs of correctness and security are based on [Shamir's Secret Sharing Scheme](#) .

In-Band OAM for IPv6

- Gather information along the path in IPv6 extension header
- In-band OAM for IPv6 (iOAM6) information carried in IPv6 extension header
 - Native v6 extension header or double-encap
- Restrict use to a specific domain
 - Domain-ingress, domain-egress, and select devices within a domain insert/remove/update the extension header
 - Information export via IPFIX/Flexible-Netflow / publish into Kafka
 - Packets with iOAM6 option handled in the fast-path of a router
- Flexible set of data carried as option headers
 - Tracing data, proof of transit data, edge-to-edge data



Extension Headers Policy? Forward? Drop ?

Extension Header Security Policy for Enterprise

- White list approach for your traffic
 - Only allow the REQUIRED extension headers (and types), for example:
 - Fragmentation header
 - Routing header type 2 & destination option (when using mobile IPv6)
 - IPsec 😊 AH and ESP
 - And layer 4: ICMPv6, UDP, TCP, GRE, ...
 - If your firewall is capable:
 - Drop 1st fragment without layer-4 header
 - Drop routing header type 0
 - Drop/ignore hop-by-hop



Source: Tony Webster, Flickr

Extension Header Loss over the Internet

- End users SHOULD filter packets with extension headers
- But, what are your ISP and its transit providers doing to your packets?



Source: Paul Townsend, Flickr

Previous Extension Headers Research by Others

- IETF-88, Nov-2013, fgont-iepg-ietf88-ipv6-frag-and-eh.pdf
 - *“Fragmentation and Extension Header Support in the IPv6 Internet”*
 - Single origin, destination = Alexa top web sites (883 unique addr)
 - Ext header size: 8 bytes and 1024 bytes; Failure rate: 45%
- IETF-89, with Tim Chown: 60% packet drops
- IETF-90, Jul-2014, iepg-ietf90-ipv6-ehs-in-the-real-world-v2.0.pdf
 - *“IPv6 Extension Headers in the Real World v2.0”*
 - Origin: RIPE Atlas probes, destination = Alexa again
 - Ext header size: 8, 256, 512 and 1024 bytes
 - Failure rate: between 60% and 90%
- December 2015, draft-ietf-v6ops-ipv6-ehs-in-real-world-02

 Campaign in June 2015

Issues with Previous Experiments

- Destination: big web sites (Alexa)
- It is expected that destination drops what is unexpected
- Not testing about Routing Header (for segment routing)

Methodology of our study

1. Determine a set of IPv6 addresses to test :
 - From Alexa's Top 1 Million list
 - From IPv6 BGP-advertised prefixes
2. TCP Traceroute without EHs :
 - Send v6 packets with TCP payload to port 80 of the destination with varying TTL => Routers in the path answer with ICMPv6 Time Exceeded
3. TCP Traceroute with EHs:
 - Same thing but adding an Extension Header before the TCP payload
4. Analysing the traceroutes

Step 1) Determining a set of IPv6 addresses to test

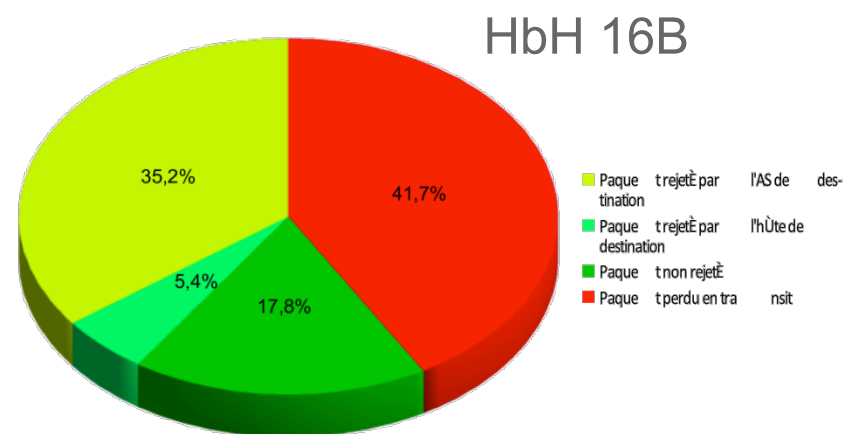
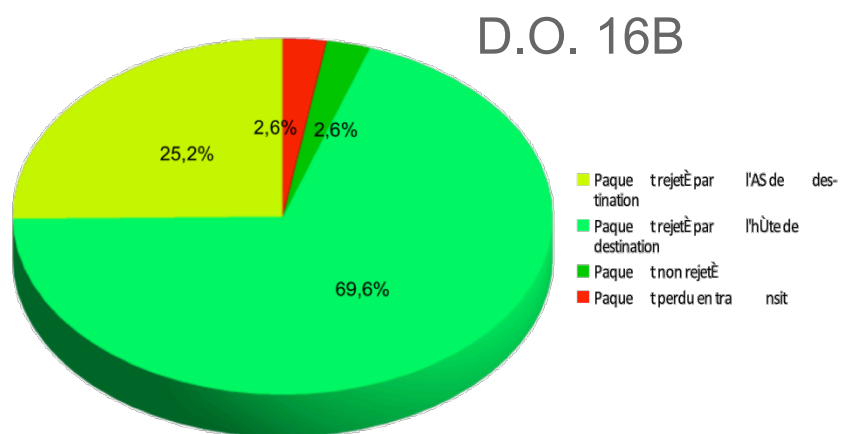
- From Alexa's Top 1 Million list :
 - Take those that have a AAAA record
 - ... with a reachable IPv6 address in the AAAA record
- From BGP-advertised IPv6 prefixes
 - Address = [prefix>::1
 - Doesn't exist ? No problem, we are supposed to reach the AS -> Enough

Methodology of our study : Analysing the traceroutes

- Is it a problem ? Depends where it was dropped !
 - If dropped by the destination organization (host or same AS): Not a problem !
 - If dropped in transit: not cool...
- Where is the dropping node ?
 - If IP corresponds to some major IXPs, we look up the corresponding ASN by knowing the addressing logic, or in a database
 - Otherwise, normal Maxmind GeoIP ASN lookup

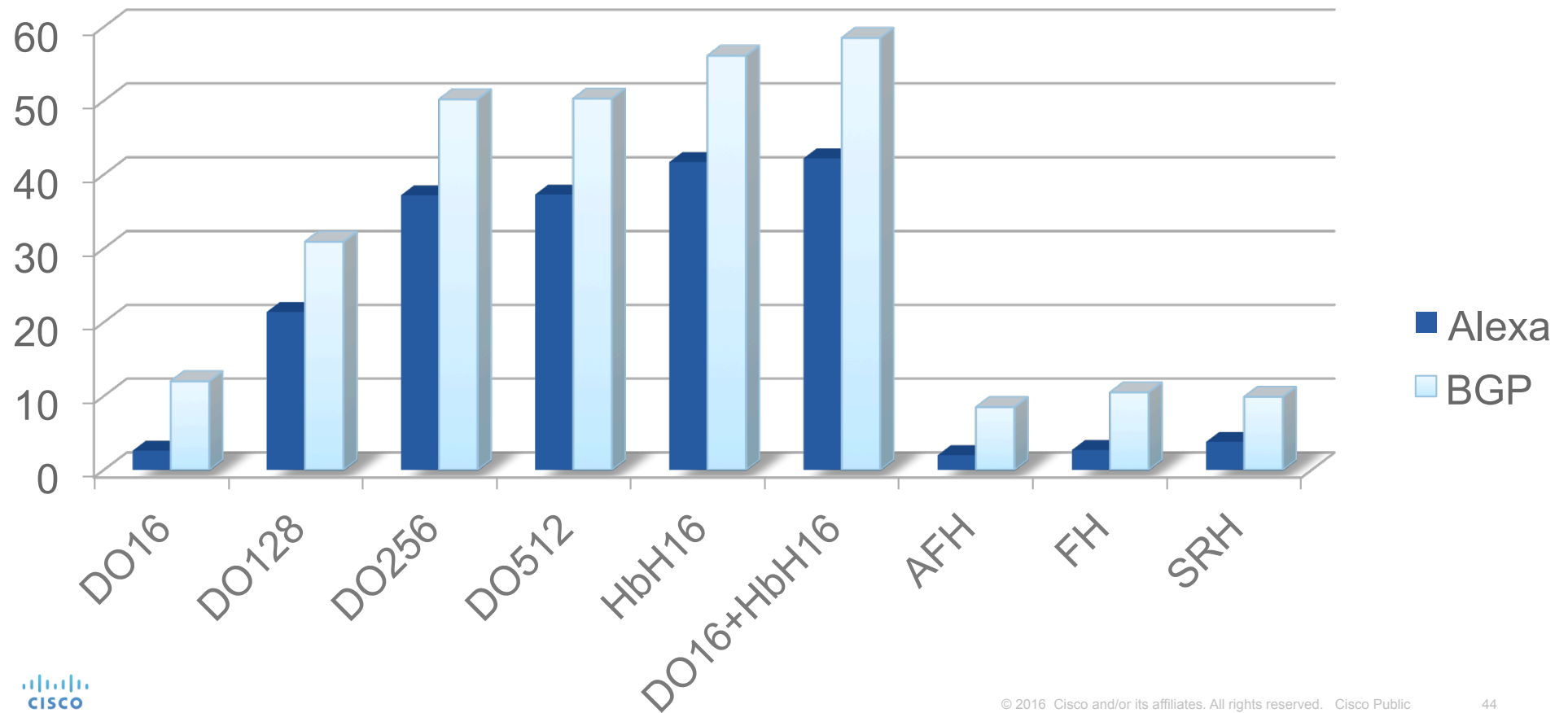
Results and analysis

- Drop rates depend on the Extension Header

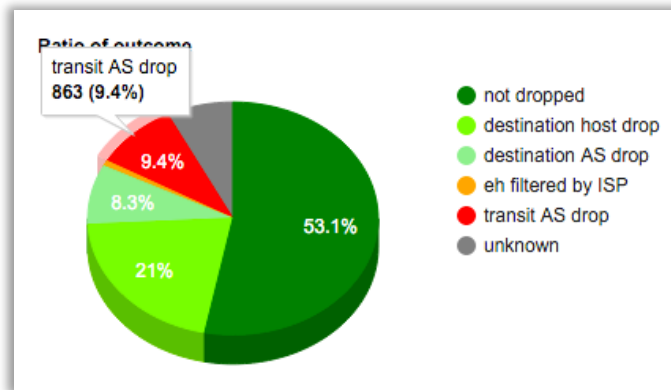


For Alexa

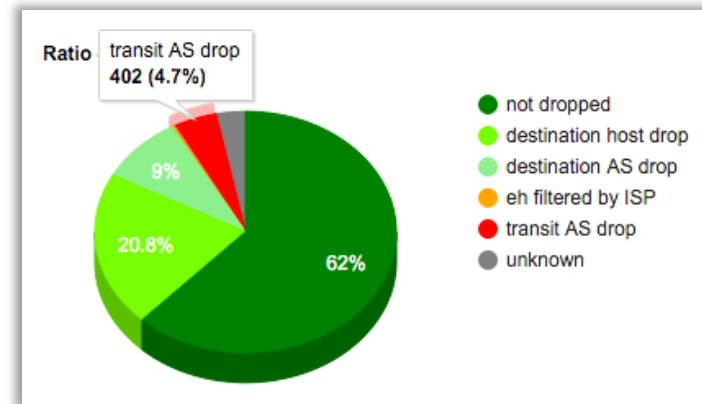
Transit Drop rates in Spring 2015



Things Keeps Improving Though



BGP in Spring 2015



BGP in Spring 2016

- Current research by Polytechnique Paris (Mehdi Kouhen) and Cisco (Eric Vyncke)
 - And VM provided by Sander Steffann
- <http://btv6.vyncke.org/exthdr/index.php?ds=bgp2016&t=fh> (work in progress!)

A last request A last wink

Please comment on this morning speakers' work 😊

IP Multicast
Internet-Draft
Intended status: Informational
Expires: June 26, 2016

E. Vyncke
Cisco
E. Rey
ERNW
A. Atlasis
NCI Agency
December 24, 2015

MLD Security
draft-vyncke-pim-mld-security-01

