Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Freie Universität Berlin

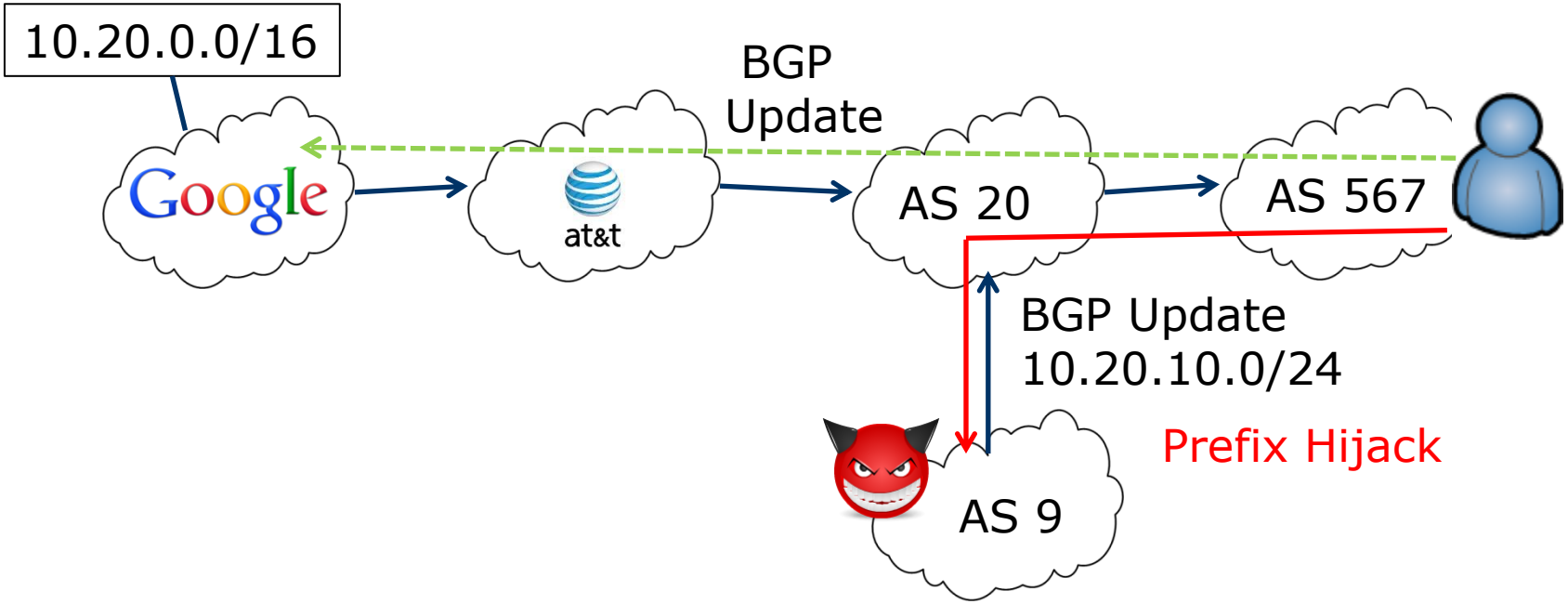
RTRlib

An Open-Source Library in C for RPKI-based Prefix Origin Validation

Matthias Wählisch, Fabian Holler,
Thomas C. Schmidt, Jochen H. Schiller

m.waehlisch@fu-berlin.de
schmidt@informatik.haw-hamburg.de

Harming the Internet Backbone

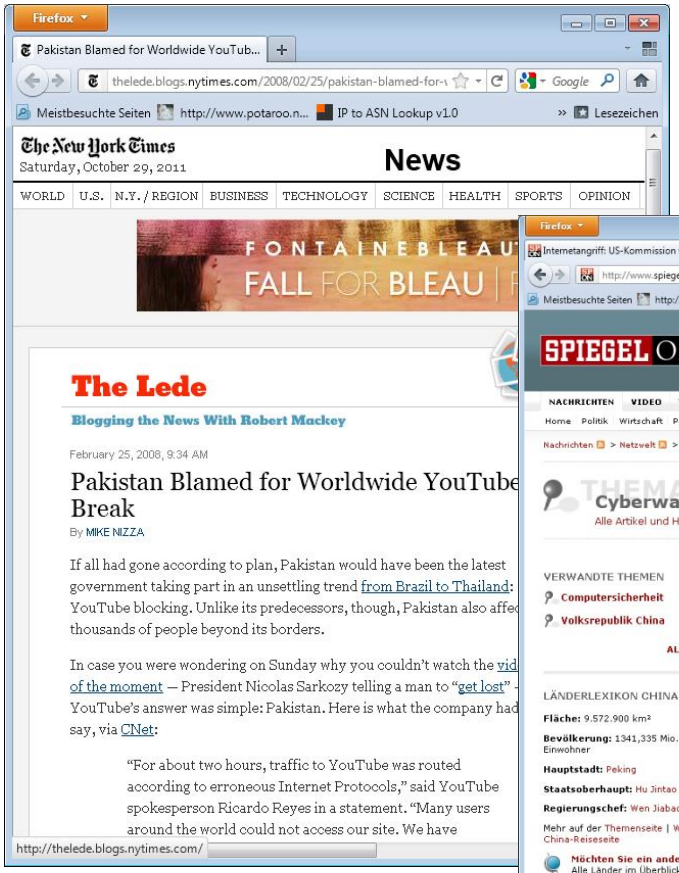


Problem

- BGP is based on trust
- Originally, no mechanism for cryptographically strong verification of AS-to-prefix mapping

Prefix Hijacking – Reality?

Prominent Examples



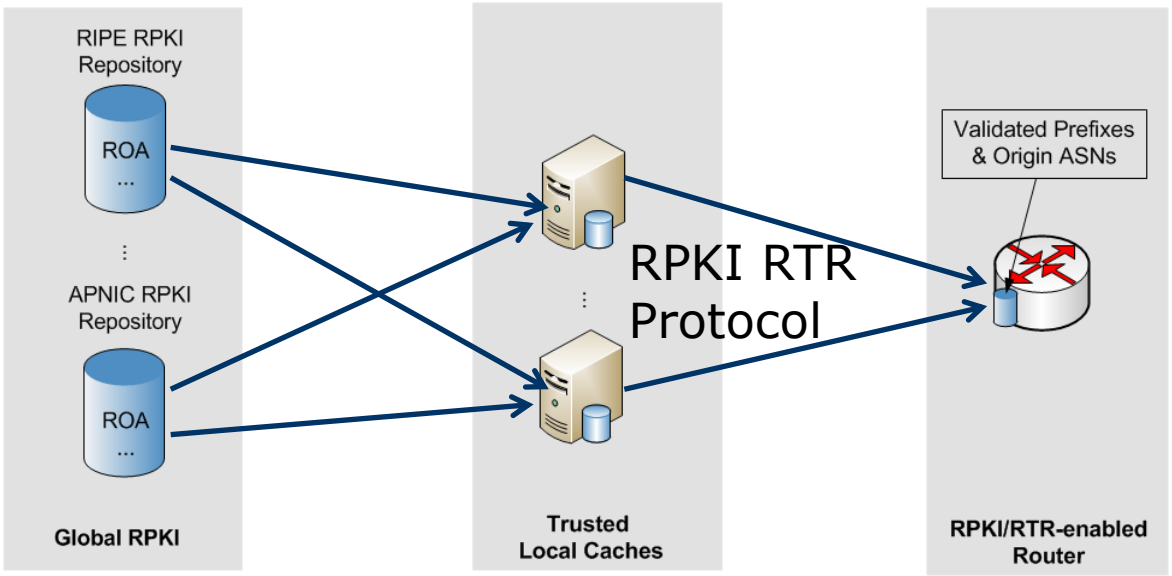
244

For about 18 minutes on April 8, 2010, China Telecom advertised erroneous network traffic routes that instructed U.S. and other foreign Internet traffic to travel through Chinese servers.* Other servers around the world quickly adopted these paths, routing all traffic to about 15 percent of the Internet's destinations through servers located in China. This incident affected traffic to and from U.S. government (".gov") and military (".mil") sites, including those for the Senate, the army, the navy, the marine corps, the air force, the office of secretary of Defense, the National Aeronautics and Space Administration, the Department of Commerce, the National Oceanic and Atmospheric Administration, and many others. Certain commercial websites were also affected, such as those for Dell, Yahoo!, Microsoft, and IBM.[116]

Although the Commission has no way to determine what, if anything, Chinese telecommunications firms did to the hijacked data, incidents of this nature could have a number of serious implications. This level of access could enable surveillance of specific users or sites.† It could disrupt a data transaction and prevent a user from establishing a connection with a site. It could even allow a diversion of data to somewhere that the user did not intend (for example, to a "spoofed" site). Arbor Networks Chief Security Officer Danny McPherson has explained that the volume of affected data here could have been intended to conceal one targeted attack.[117] Perhaps most disconcertingly, as a result of the diffusion of Internet security certification authorities,‡ control over diverted data could possibly allow a telecommunications firm to compromise the integrity of supposedly secure encrypted sessions.§

Caveat: Reasons may also be misconfiguration ;-)

Countermeasure: RPKI-based Prefix Origin Validation



Deployment started in January 2011

Research Questions

1. What is the performance overhead for prefix origin validation at routers?
2. Does prefix origin validation introduce new attacks that harm the local router system?
3. What is the current state of deployment?

⇒ We need a flexible and efficient open-source implementation of the RPKI RTR protocol → **RTRlib**

What is the RTRlib?

Freie Universität Berlin

General objective

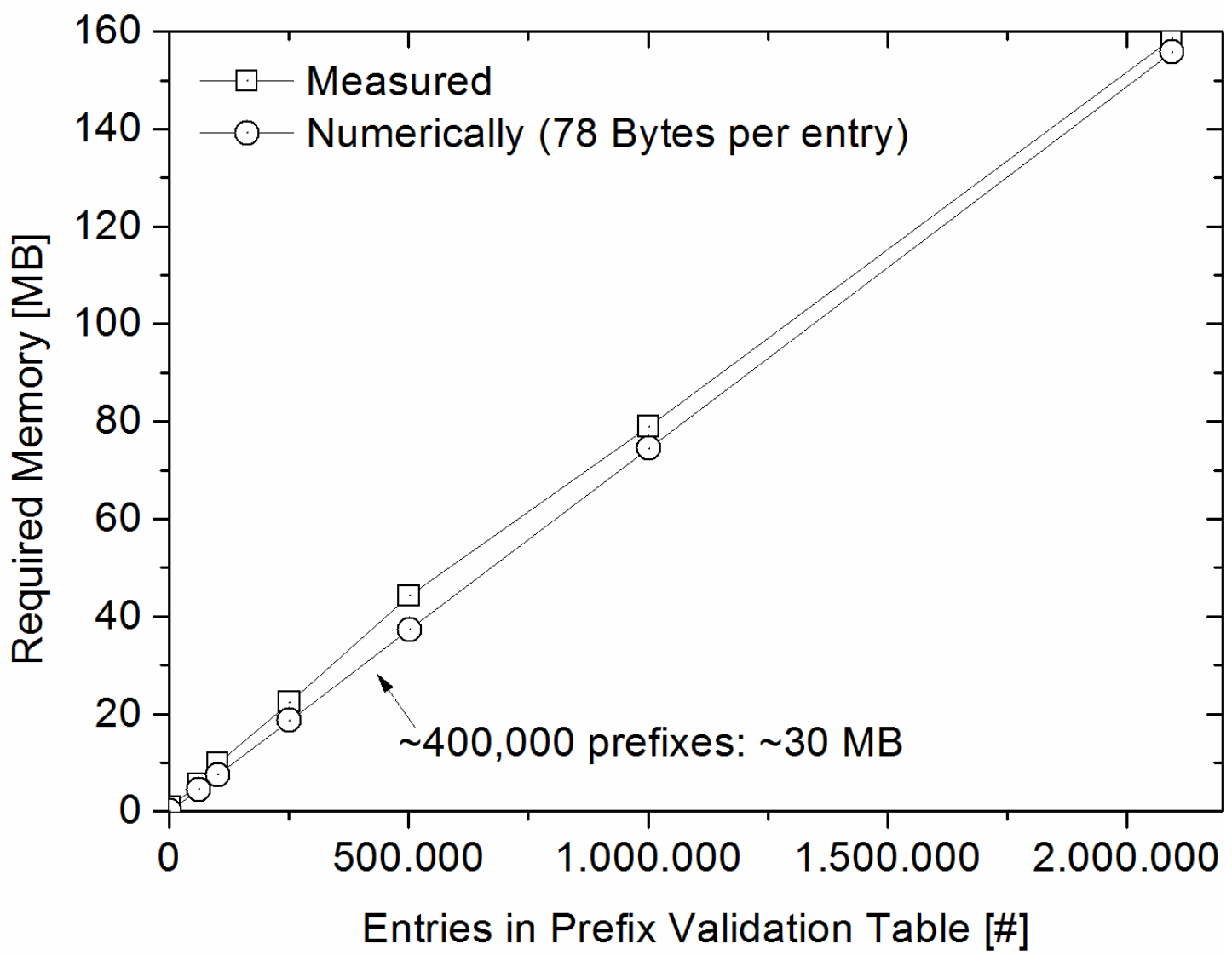
- Implementation of the RPKI-RTR client protocol in C

Details

- Fetch validated prefixes + origin ASes from RPKI cache
- Keep the routers validation database in sync
- Provide an interface between local database and routing daemon to access validated objects
- Allow also for validation of BGP updates
- Conforms to relevant IETF RFCs/drafts

It’s open-source: http://rpki.realmv6.org

Memory Consumption



Processing Time to Load Data Into Router

Freie Universität Berlin

Motivation

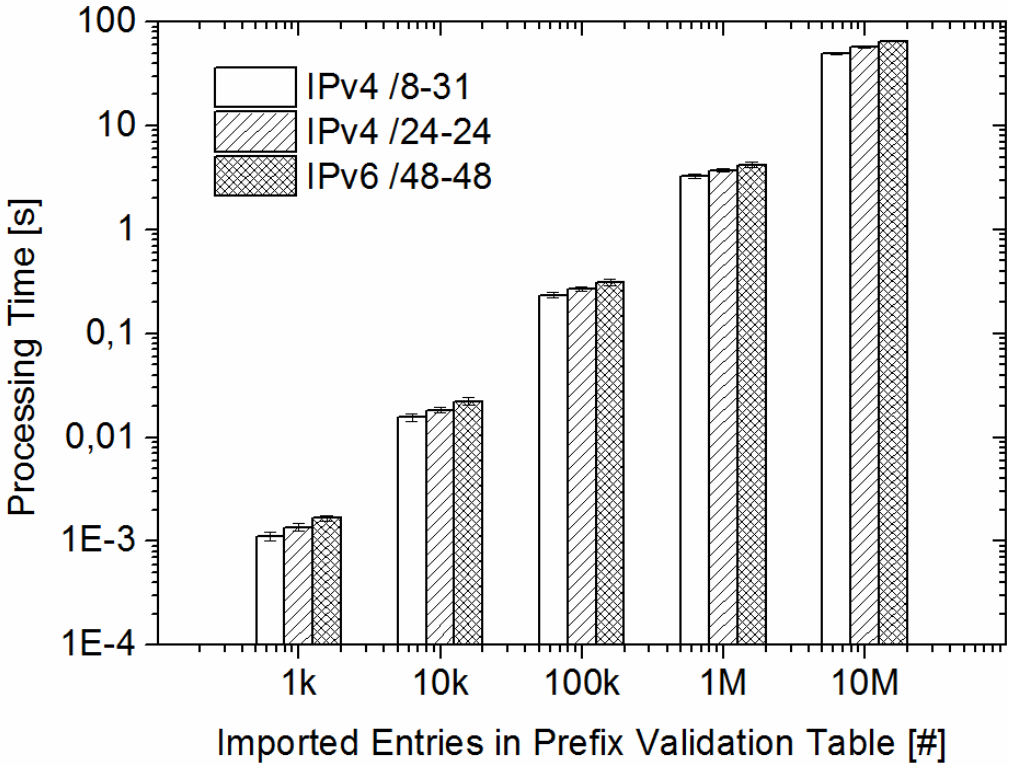
- Startup
- Cache server session reset

Evaluation Approach

- Load data directly from file

Result

- Delay will be dominated by distance to cache server



CPU Load Based on Deployment State

Freie Universität Berlin

Motivation

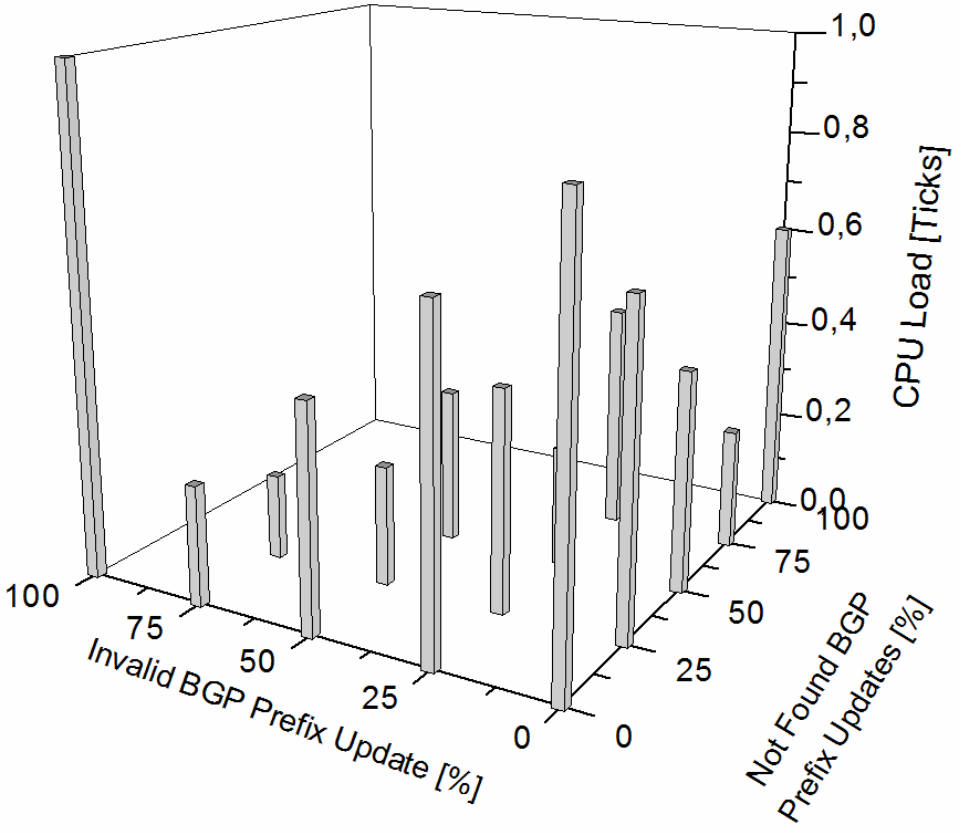
- Does deployment state influence performance?
- Complexity attacks

Evaluation Approach

- Creation of predefined mixture of validation outcome

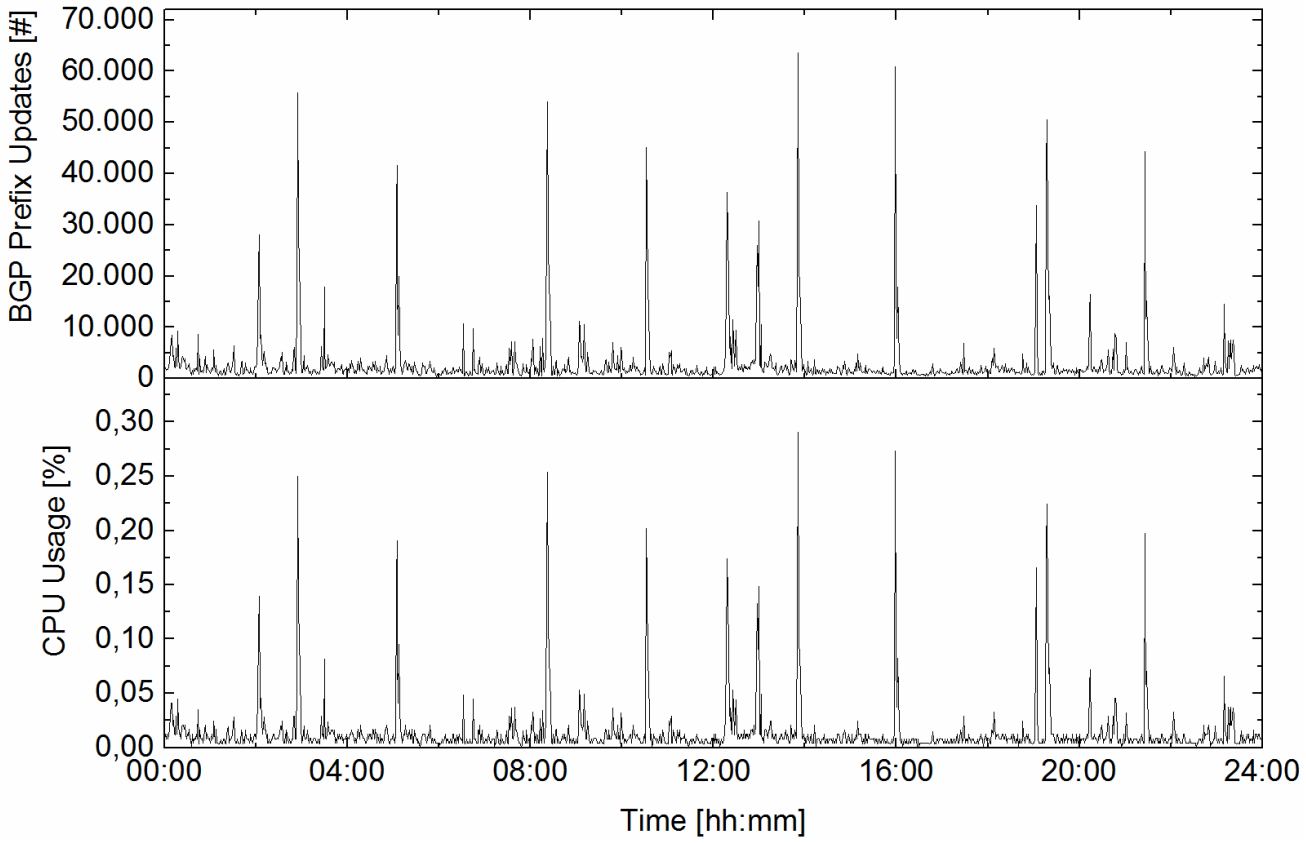
Result

- Almost no dependency



RPKI in the Wild – CPU Load

- Test on commodity hardware
  - AMD Athlon 64 X2 CPU 4200+ and 2 GB RAM
- Live BGP update stream over several months

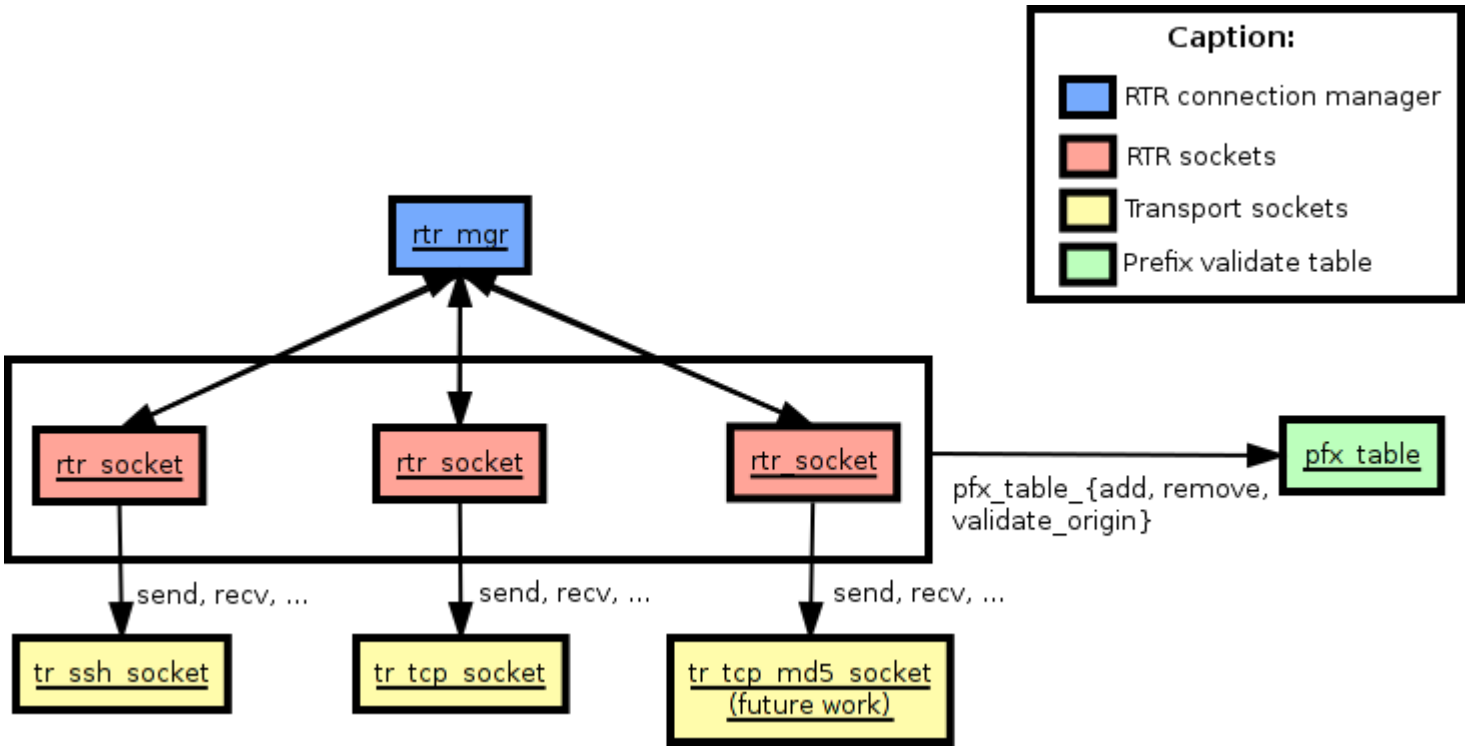


Applications

- Extension of BGP daemons
  - In discussions with BIRD developers
  - Beta version for Quagga is available
- Monitoring of the RPKI deployment
  - Integrate the library in your Python/Perl … scripts
  - Particularly suitable for real-time monitoring
- Testing purposes
  - Evaluate performance of your RPKI/RTR cache server
  - Play around with BGP update validation

RTRlib: Architectural Design

- Layered architecture to support flexibility

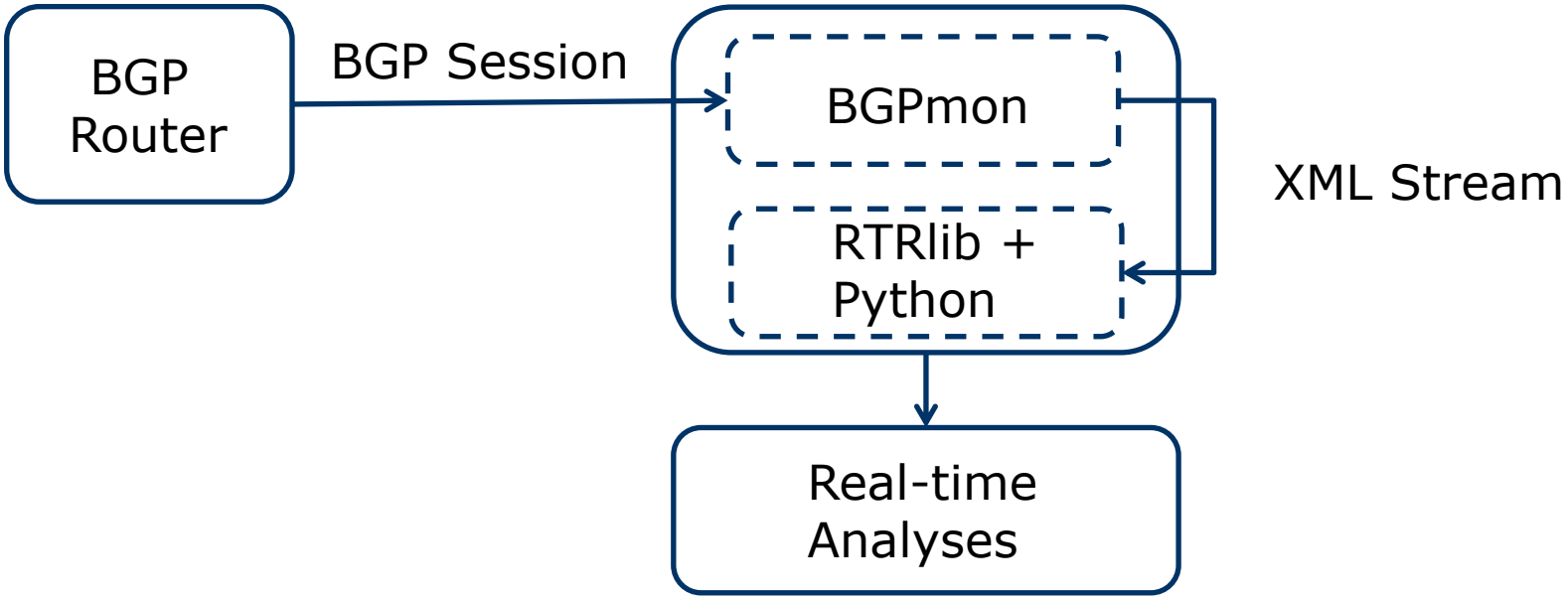


Using RTRlib for Monitoring

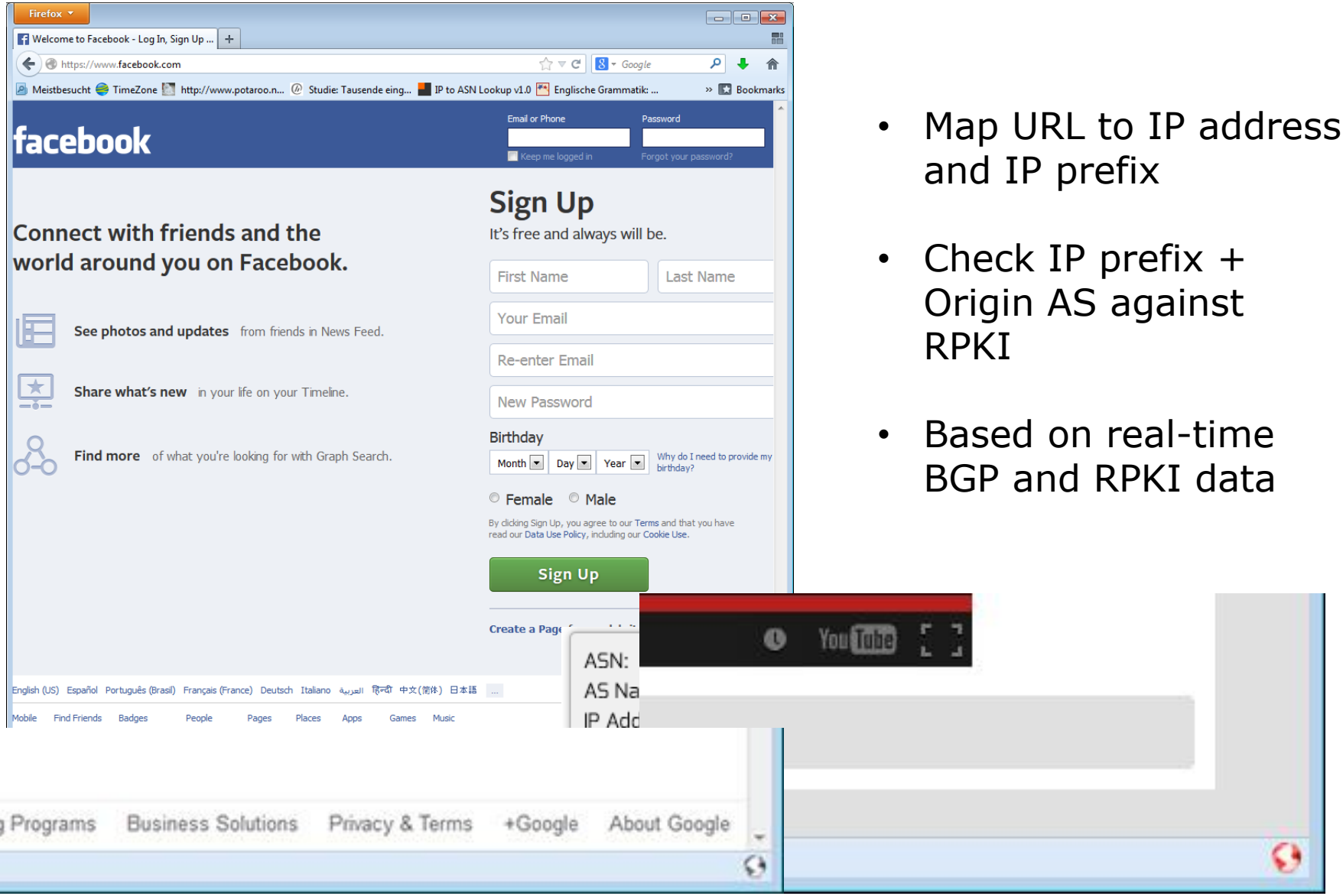
Objective: Emulate update validation of your BGP peer

Setup – No Firmware Change at Your Router:

- Tools: RTRlib + Python Script + BGPmon
- Establish peering between router and BGPmon

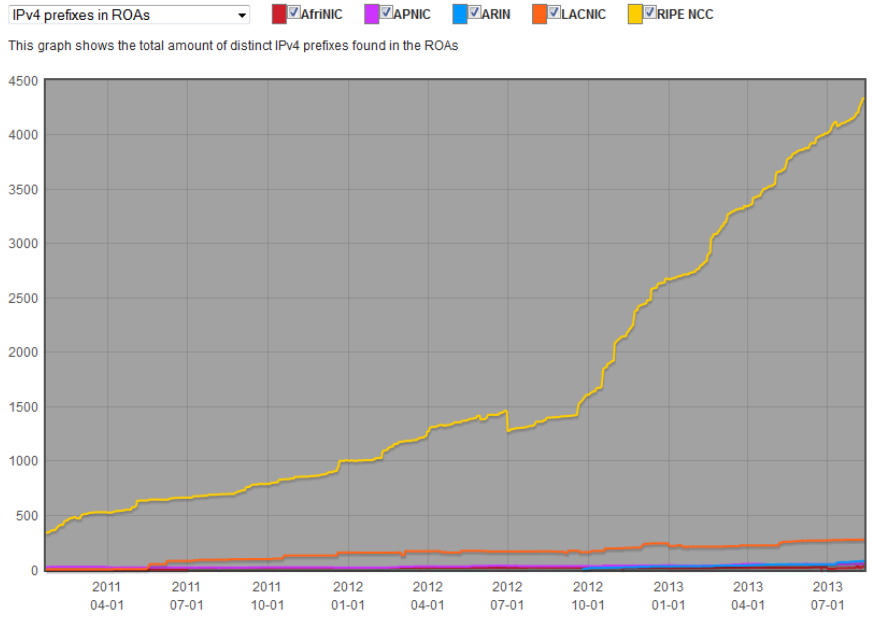


Validation of Web Server Prefix in Firefox

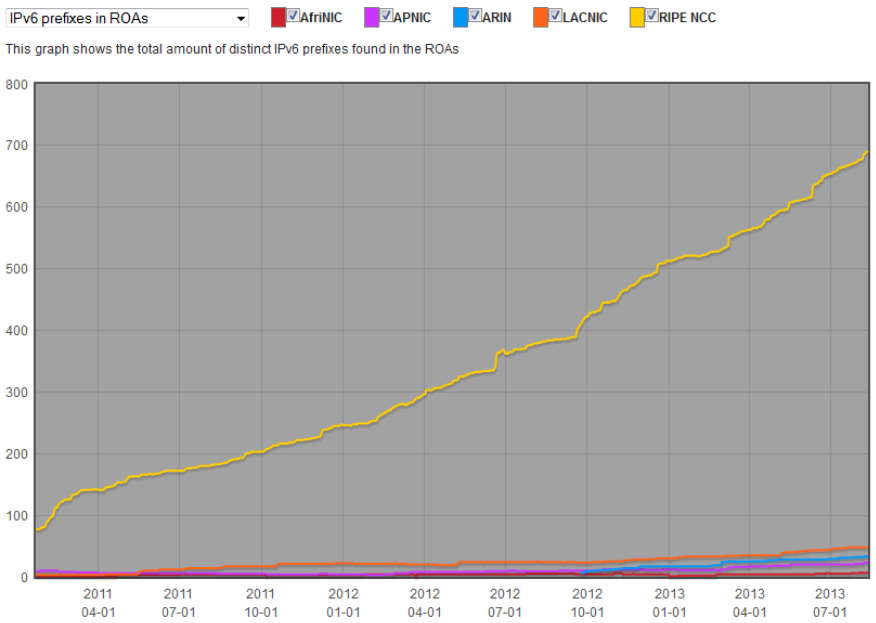


- Map URL to IP address and IP prefix
- Check IP prefix + Origin AS against RPKI
- Based on real-time BGP and RPKI data

Current # IP prefixes in ROAs



IPv4



IPv6

http://certification-stats.ripe.net/

Attacking the Local Router System – One Example?

Freie Universität Berlin

- Attacker misuses the creation of AS to prefix mapping (ROA)
- Any owner of an IP prefix can create arbitrary ROAs for this prefix
  - Example: Operators own 10.20.30.0/24
  - Create 10.20.30.1/32 AS10; 10.20.30.2/32 AS10; … lead to 255 entries
  - Easily to handle in IPv4 due to limited address space
  - Getting worse in IPv6

⇒Limit the creation of ROAs

Conclusion & Outlook

Freie Universität Berlin

Conclusion

- Prefix origin validation can start
- Load at routers is relatively negligible
- Monitoring of prefix origin validation: Use RTRlib

Outlook

- Continue the long-term study
- More extensive analysis of RPKI-based complexity attacks
- Development of approach to distinguish invalid updates from misconfigurations and real prefix hijacks

Thanks!

Freie Universität Berlin

Interested in Testing RPKI/RTR?

- Download: http://rpki.realmv6.org

Firefox Add-on RPKI Validator

- https://addons.mozilla.org/addon/rpki-validator/

GitHub

- https://github.com/rtrlib

Freie Universität Berlin

BACKUP

Example – Establish Transport

```
tr_socket* ssh_socket;                  //create a SSH connection
tr_ssh_config config = {
   "123.321.123.321",                   //IP
   22,                                  //Port
   "rpki_user",                         //SSH User
   "/etc/rpki-rtr/hostkey",             //Server hostkey
   "/etc/rpki-rtr/client.priv",         //Authentication private key
   "/etc/rpki-rtr/server.pub"           //Authentication public key
};
tr_ssh_init(&config, &ssh_socket);

tr_socket* tcp_socket;                  //create unprotected TCP conn.
tr_tcp_config tcp_config = {
   "123.321.123.321",                   //IP
   "1234"                               //Port
};
tr_tcp_init(&tcp_config, &tcp_socket);
```

Create Connection Manager and Perform Origin Validation

Freie Universität Berlin

```
//init all rtr_sockets with the same settings
//srv.pool,polling_period,cache_timeout,update_fp,conn_f
rtr_mgr_init(&p0, 60, 120, NULL, 0, NULL, 0);

//create and start the connection manager
rtr_mgr_socket mgr_sock;
rtr_mgr_start(&mgr_sock, &p0);

//validate the BGP origin ASN 12345 for 10.10.0.0/24
ip_addr prefix;
prefix.ver = IPV4;
prefix.u.addr4.addr = 0x0A0A0000;

pfxv_state result;
rtr_mgr_validate(mgr_sock, 12345, &prefix, 24, &result);
```

Prefix Origin Verification & RPKI

Validation process consists of two steps

1. Validation of ROAs

- Performed at external cache

2. Validation of BGP updates

- Performed at BGP router
- No additional cryptographic operations necessary

- IETF "RPKI/RTR protocol" manages push of *valid* ROAs from cache to BGP router
  - Implementations for Cisco and Juniper available
  - Open-source BGP daemons on the way
- Evaluation result of BGP update: VALID, INVALID, NOT_FOUND
  - Combine this with BGP policies

Protection Concepts

1. **Prefix Origin Validation**
   - Mapping of IP prefixes and origin AS necessary
     - Including cryptographic proof
     - Prefix owner should be able to authenticate *Origin AS(es)*
   - BGP router compares BGP update with mapping
2. **Path Validation**
   - BGP path information are cryptographically secured
     - Paths will be signed

Challenges

- Cryptographic operations are complex
- Minimal additional load at routers

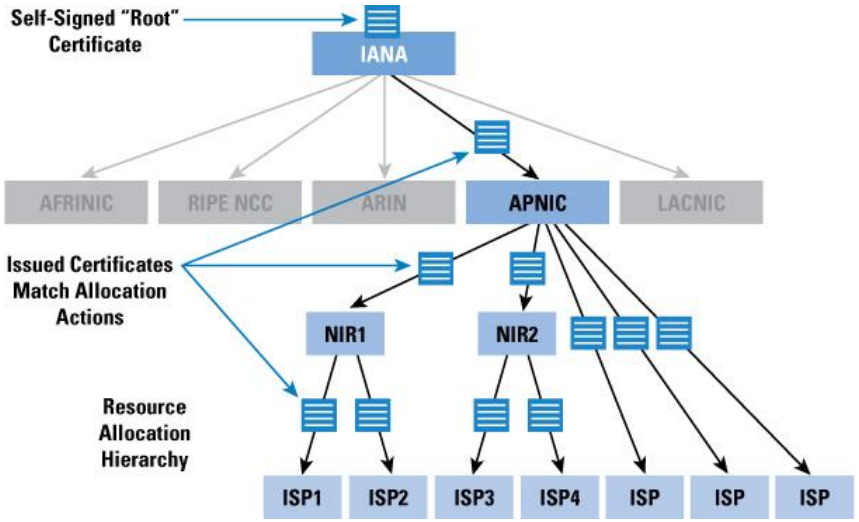
In the following we concentrate on 1.

Resource Public Key Infrastructure (RPKI)

Freie Universität Berlin

- System that allows to attest the usage of IP addresses and ASNs (i.e., Internet resources)
- RPKI includes cryptographically provable certificates

- Certificate hierarchy reflects IP-/AS-allocation in the Internet
  - Currently, each RIR creates a self-signed root certificate



Source: Cisco

- Implementation of the RPKI started January 2011
- All RIRs participate

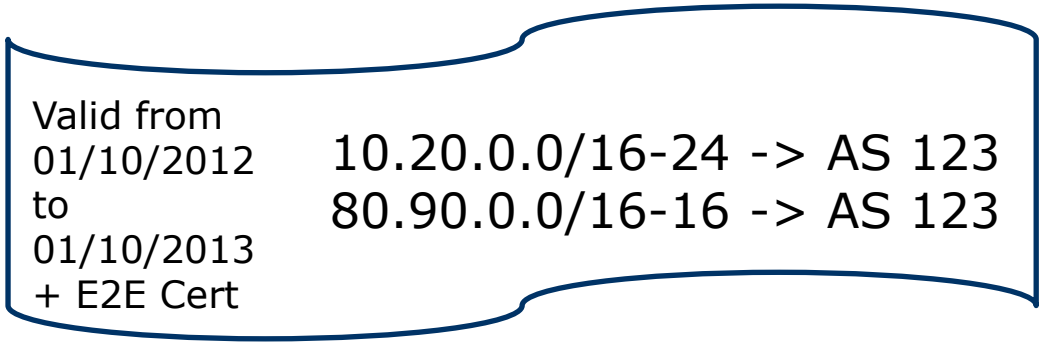
Routing Origination Authorization (ROA)

Freie Universität Berlin

- Content of an ROA
  - Set of IP prefixes with minimal and maximal (optional) length
  - An AS number allowed to announce the prefixes
  - End-Entity-Certificate
- ROA will be signed with the certificate of the RPKI
- Note: Multiple ROAs per IP prefix possible

Example:

ROA



AS 123 is allowed to announce network range 10.20.0.0/16 to 10.20.0.0/24 and 80.90.0.0/16 from 1st Oct. 2012 until 1st Oct. 2013

RPKI & ROA

- All certificates including ROAs will be published in RPKI repositories
  - Each RIR operates one
  - You could maintain your own repository
  - Information of all repositories describe the overall picture
- Check if AS is allowed to announce IP prefix
  = check the corresponding ROA
  - Corresponding ROA will be determined based on CIDR
  - ROA needs cryptographic verification
  - ROAs implements a positive attestation
    - If a ROA for a prefix exists, announcements of all origin ASes that are not included will be considered INVALID