Internet Infrastructure Security

Thomas C. Schmidt

t.schmidt@haw-hamburg.de

HAW Hamburg, Dept. Informatik

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Agenda

- Motivation
  - Implicit Trust
- DNSSEC
  - DNS Vulnerabilities
  - DNSSEC Operations
  - Chain of Trust
  - Deployment
- RPKI
  - BGP Hijacking
  - Origin Attestation
  - Deployment, Implementation: RTRlib
- DANE
  - Secure E2E Bootstrapping
  - DANE & DNSSEC
- Certificate Transparency

How Trustworthy is the Internet Infrastructure

MOTIVATION

Whom Do We Trust on the Internet?

When invoking a service

- We use names that the infrastructure resolves
- We send packets that the infrastructure guides
- We use application interfaces that appear authentic

We have trusted

⇒ Name resolution (DNS)

⇒ Packet delivery (Routing & Forwarding)

⇒ Application origination (Certification ?)

Who is Involved?

- DNS:
  - Recursive resolver
  - Caches
  - Authoritative nameservers
- Routing:
  - Control plane: many BGP speakers
  - Forwarding plane: eyeball, transit & origin ISPs
- Application:
  - Application server
  - Indirect (hidden) contributors
  - Certification authority ?

Who Would Do Harm?

Securing Names

DNSSEC

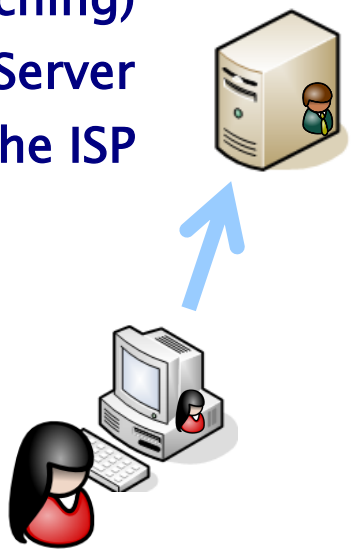
Attacking Names

www.
mybank.
com

Source: Klaus-Peter Kossakowski

Attack Vectors (1)

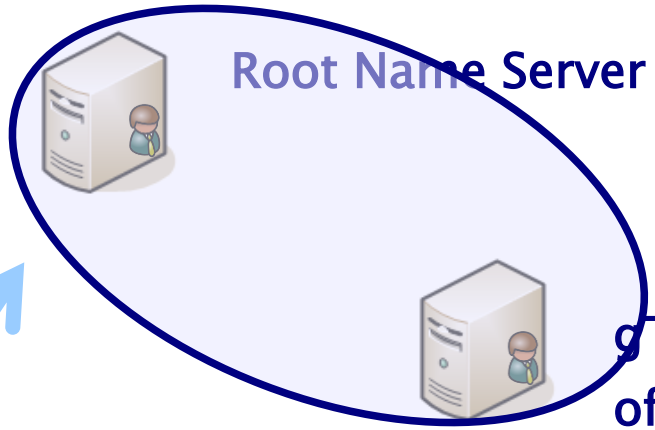
(Caching)
Name Server
of the ISP



www.
mybank.
com

Source: Klaus-Peter Kossakowski

Attack Vectors (2)



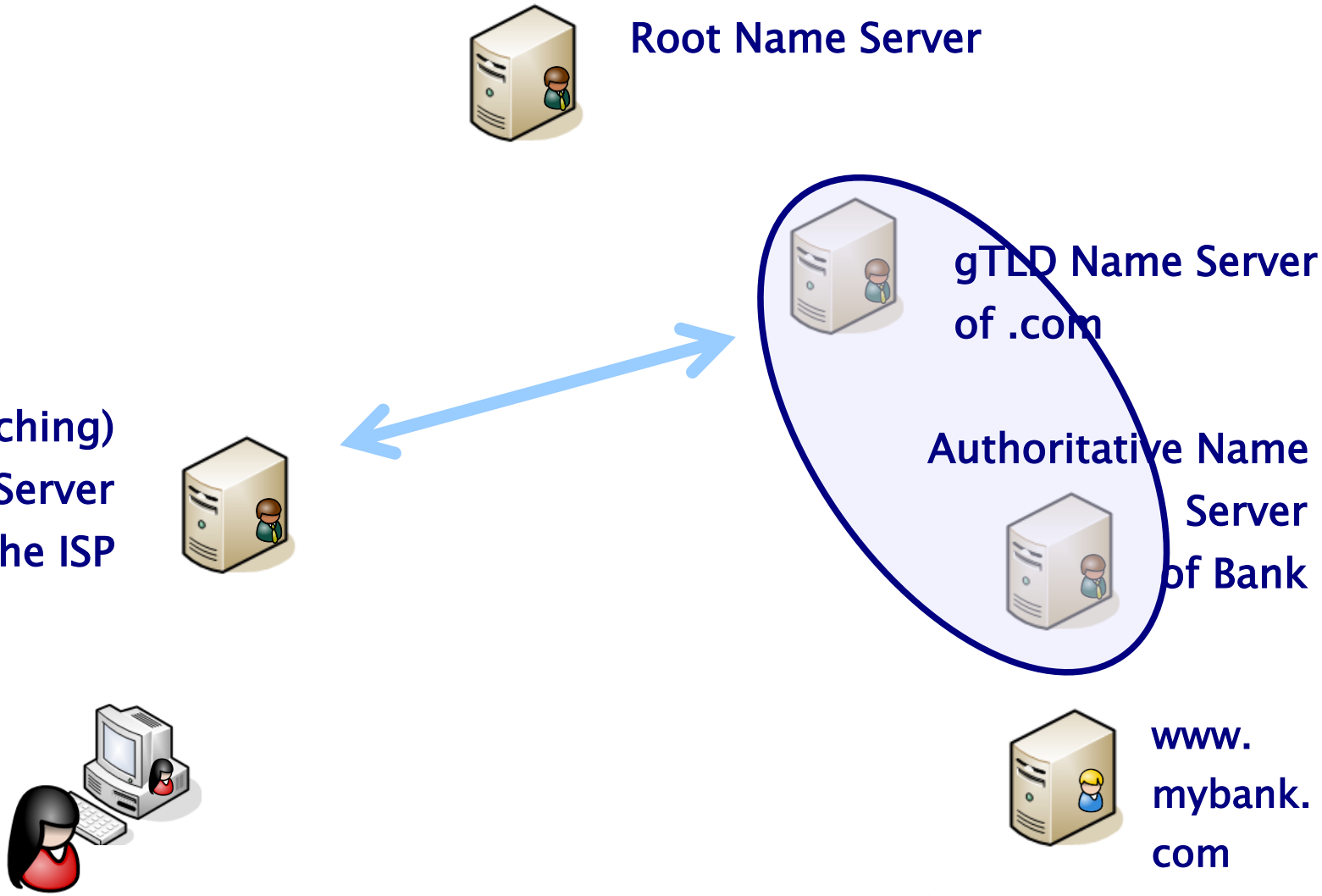
gTLD Name Server of .com

(Caching) Name Server of the ISP

www. mybank. com

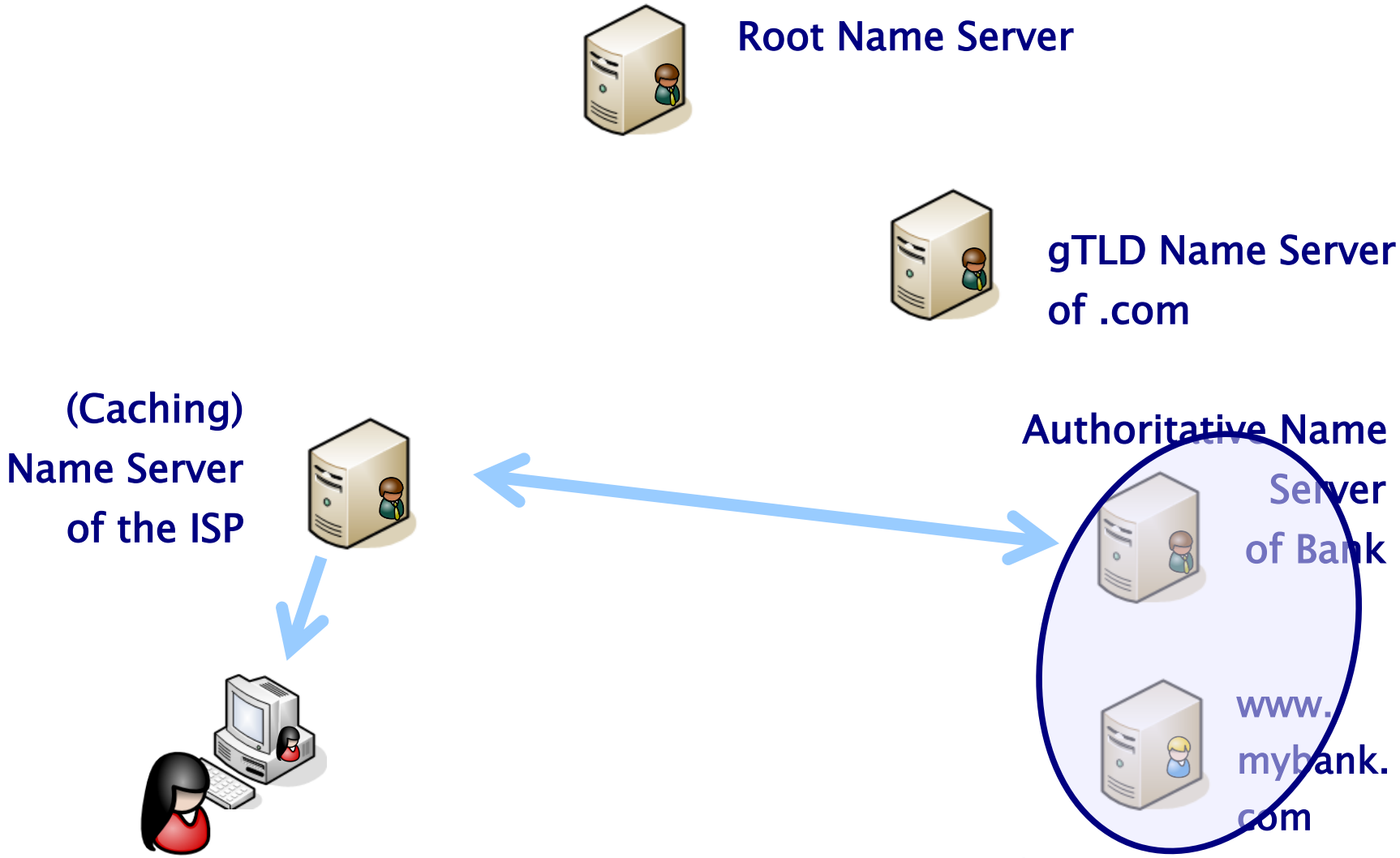
Source: Klaus-Peter Kossakowski

Attack Vectors (3)



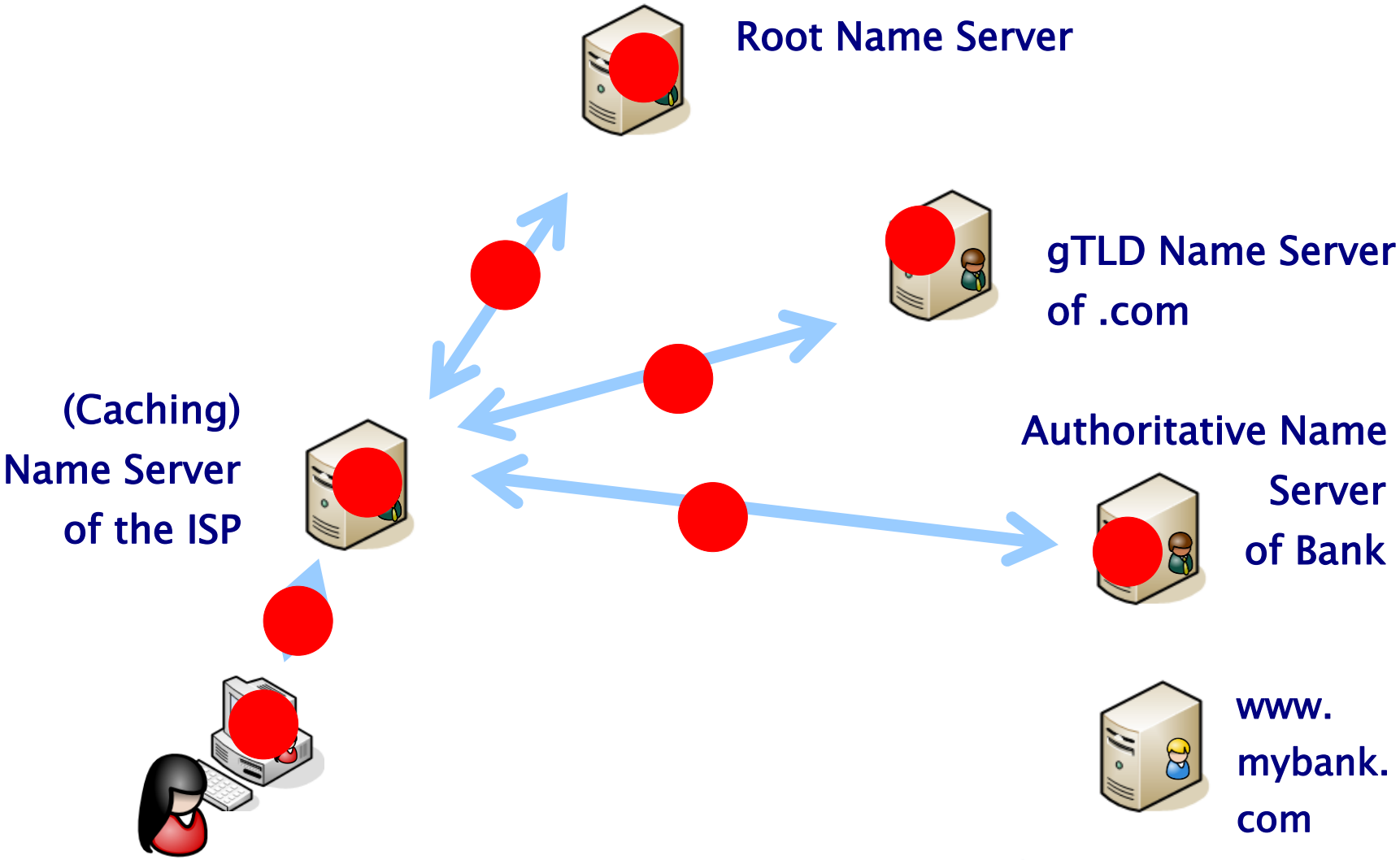
Source: Klaus-Peter Kossakowski

Attack Vectors (4)



Source: Klaus-Peter Kossakowski

Attack Surface of the DNS



Source: Klaus-Peter Kossakowski

RFC 973 on Trust in DNS, Jan. 1986

RFC 973 January 1986
Domain System Changes and Observations

UDP checksums

Many versions of UNIX generate incorrect UDP checksums, and most ignore the checksum of incoming UDP datagrams. The typical symptom is that your UNIX domain code works fine with other UNIXes, but won't communicate with TOPS-20 or other systems. (JEEVES, the TOPS-20 server used for 3 of the 4 root servers, ignores datagrams with bad UDP checksums.)

Making up data

There are lots of name servers which return RRs for the root servers with 99999999 or similar large values in the TTL. For example, some return RRs that suggest that ISIF is a root server. (It was months ago, but is no longer.)

One of the main ideas of the domain system is that everybody can get a chunk of the name space to manage as they choose. However, you aren't supposed to lie about other parts of the name space. Its OK to remember about other parts of the name space for caching or other purposes, but you are supposed to follow the TTL rules.

Now it may be that you put such records in your server or whatever to ensure a server of last resort. That's fine. But if you export these in answers to queries, you should be shot. These entries get put in caches and never die.

Suggested domain meta-rule:

If you must lie, lie only to yourself.

Paul Mockapetris

(Inventor of the DNS)

Problem: This guideline does not comply anymore with malicious activities in the Internet.

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

DNSSEC Design Objectives

- Current specifications: RFCs 4033, 4034, 4035 (orig. 2535)

Goals

- Provide integrity (prevent spoofing) by
  - Authenticating messages of name servers, and
  - Authenticating resource records
- Proof of non-existence (prevent DoS against names)

Non-Goals

- Confidentiality by hiding DNS data or requests
- Authorization of requests or requestors
- Protection against DDoS attacks (via traffic amplification)

DNSSEC Fundamentals

- DNSSEC uses Public Key Crypto to
  - Authenticate/verify Resource Record Sets (RRSets)
  - Authenticate/verify zone delegations
- Each Zone has key(s) for signing its RRSets
  - Trust chain follows zone delegation
- Public keys are stored in the DNS: DNSKEY Resource Record
- Signatures are stored in the DNS: RRSIG Resource Record

Signing Resource Record Sets

- RRset:
  - www.opendnssec.se. 7200 IN A 192.168.10.3
    - A 10.0.0.3
    - A 172.25.215.

- DNSKEY RDATA:
  - opendnssec.se. 3600 IN DNSKEY 256 3 5
    AQOvhvXXU61Pr8sCwELcqqq1g4JJCALG4C9EtraBKVd+vG
    IF/unwigfLOAO3nHp/cgGrG6gJYe8OWKYNgq3kDChN

Flags, Protocol, Algorithm

- RRSIG RDATA:
  - opendnssec.se. 3600 IN RRSIG A 5 2 3600
    20050611144523 20050511144523 3112 opendnssec.se.
    VJ+8ijXvbrTLeoAiEk/qMrdudRnYZM1VIqhNvhYuAcYKe2X/jqYfMfj
    fSUrmhPo+0/GOZj66DJubZPmNSYXw==

Type covered, Algorithm, # of labels covered, orig. TTL

Signature time range, key tag, signer's name

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

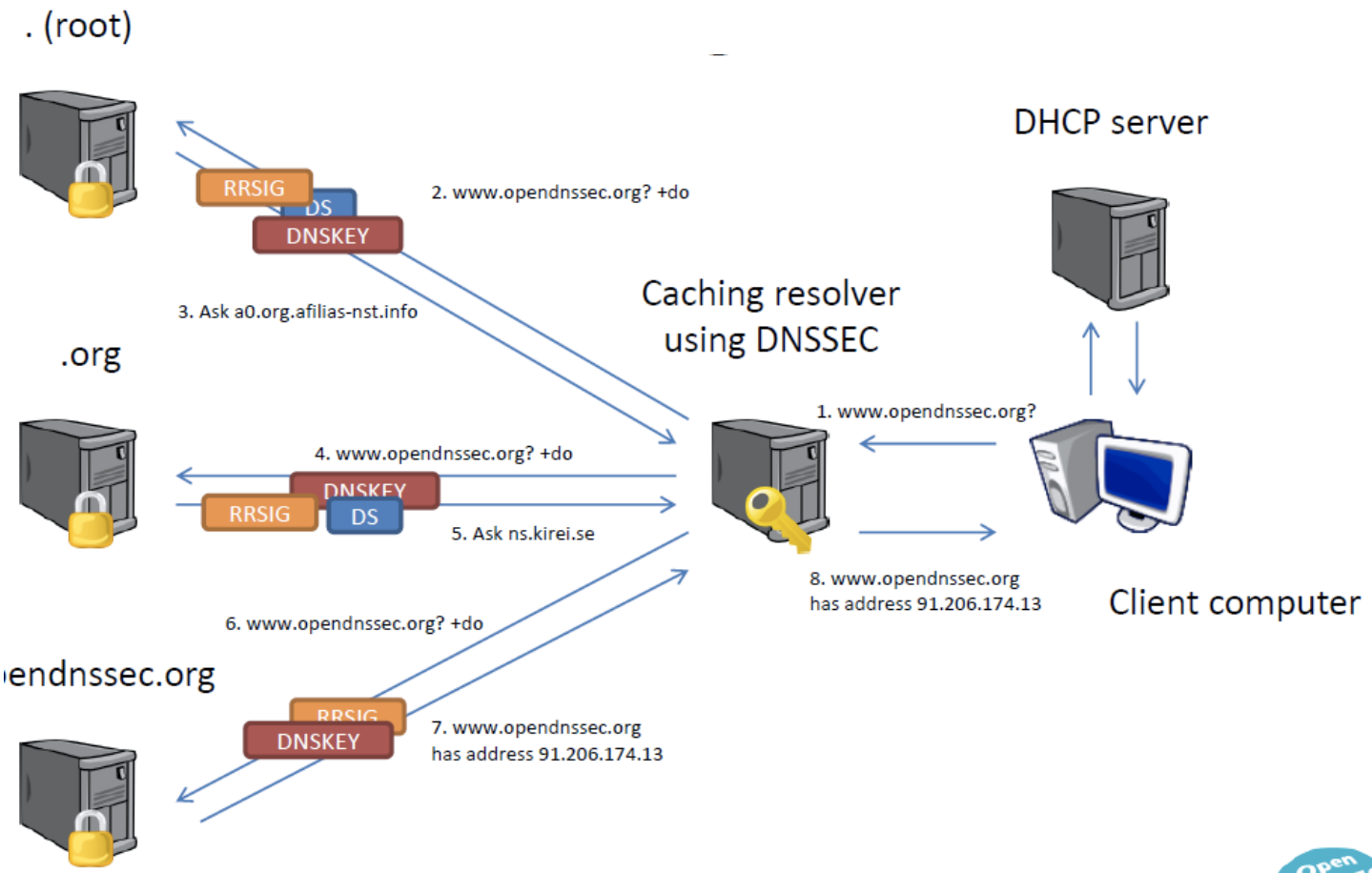
Delegation Signer (DS) Records

- Handle for building the chain of trust along names
- A DS record (the hash of the DNSKEY) is published at the parent zone to delegate trust to the child zone
- Example (name, types, key-tag, algorithm, digest-type, digest):

```
opendnssec.se. IN DS 27295 5 1
5AEF372D65BC594A7AF5E0E77CDDA55E0C43A56A
```

- The DS records are signed by the parent
- Important: DS should not be in child's zone!

Resolving DNSSEC



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Source: OpenDNSSEC Consort.

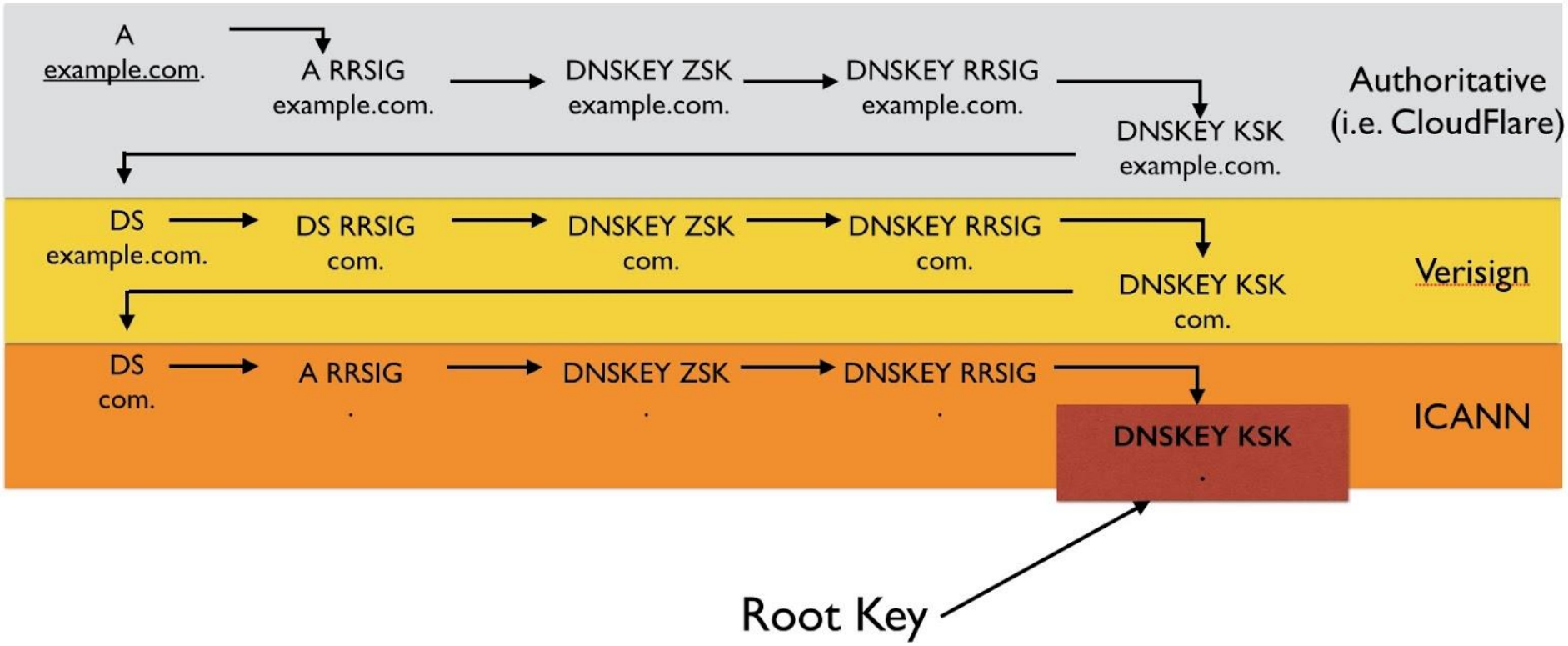
DNSSEC Crypto

- Problem: Keys in trust chain locked with parents – changes are difficult …
- Solution: Two keys
  - The Key Signing Key (KSK) for trust establishment
  - The Zone Signing Key (ZSK) for signing RRs
- **KSK** signs the ZSK, it maybe offline for protection
  - Changes involve third parties
- **ZSK** signs daily DNS changes, needed 'on disk'
  - Changes without third parties
  - Caveat: keys may be cached

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

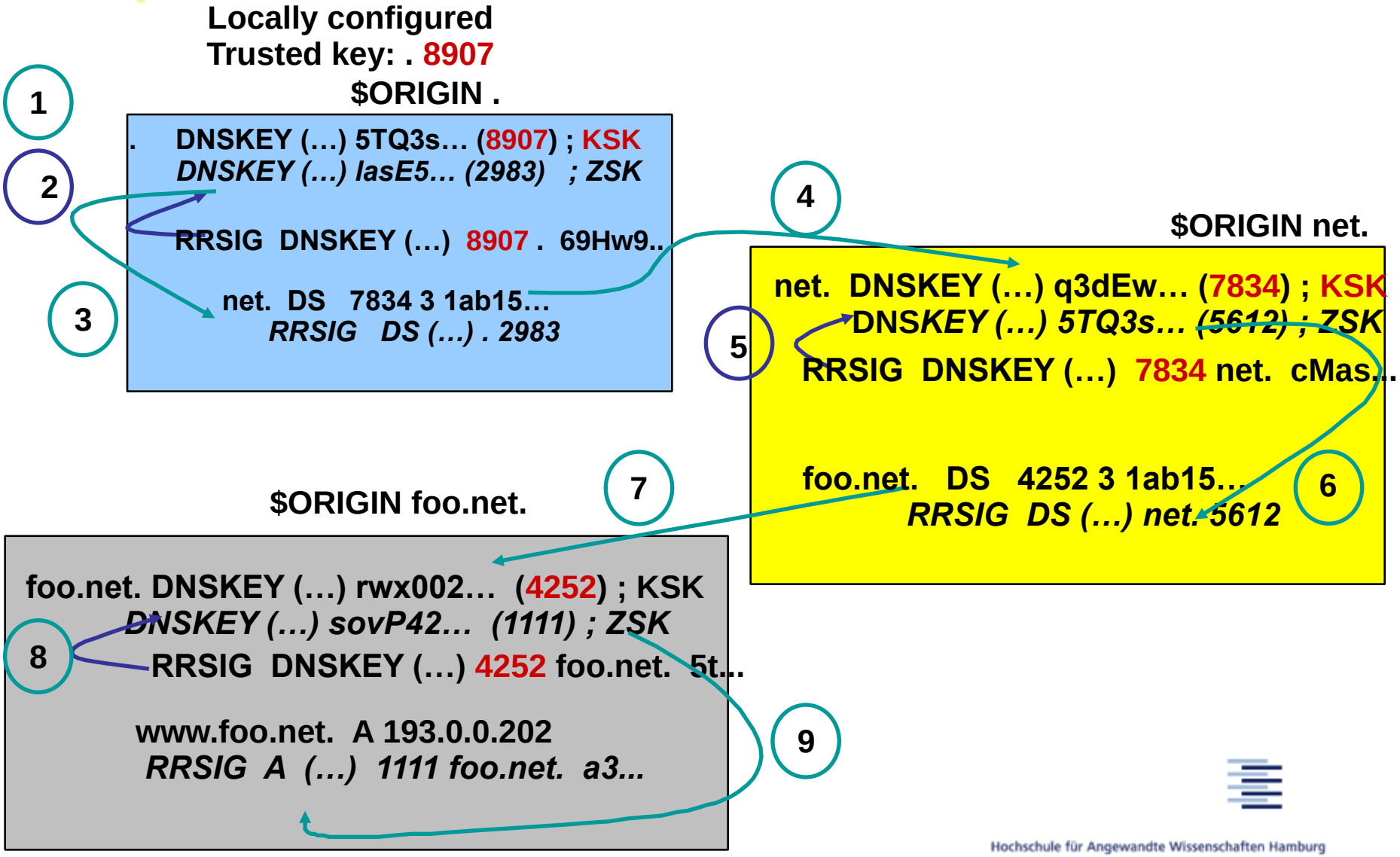
DNSSEC Trust Chain

The root of trust is the KSK DNSKEY for the DNS root. This key is universally known and published.



Source: Cloudflare

Walking the Chain of Trust



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Source: NLnet Labs

Chain of Trust Verification, Summary

- Data in zone can be trusted if signed by a Zone-Signing-Key
- Zone-Signing-Keys can be trusted if signed by a Key-Signing-Key
- Key-Signing-Key can be trusted if pointed to by trusted DS record
- DS record can be trusted
  - if signed by the parents Zone-Signing-Key
  - or
  - DS or DNSKEY records can be trusted if exchanged out-of-band and locally stored (Secure entry point)

Source: NLnet Labs

Deployment Options@Clients

Lookup procedure

1. Full DNS(SEC) resolver
   - Fully DNSSEC compliant
   - Performs DNSSEC validation on its own
2. Stub resolvers: Minimal DNS resolvers, usually deployed on end hosts to perform recursive DNS queries
   a) Client completely trusts local DNS server (e.g., from the ISP)
   b) Client decides autonomously how to handle unauthenticated data
      - DNS query includes DO bit (DNSSEC OK Bit): Enforce the server to perform validation
      - DNS server performs DNSSEC validation and answers with AD flag (Authenticated Data) or error
      - Typical deployment model

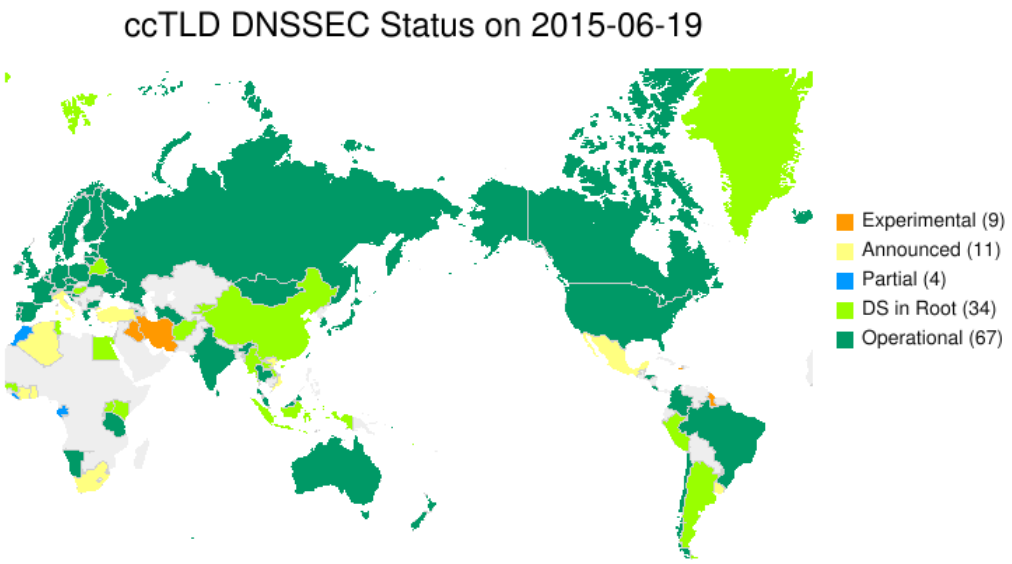
NSEC/NSEC3

- Provide proof of non-existence
- NSEC points to the next label (domain name) in the zone, allows for zone walk ("get next")
  - Zone walk often unwanted
- NSEC3 prevents 'walking in the clear'
  - Translates into hashes (linked list of hashed names)
  - Non-existence of hash proves non-existence of name
- Creates new RRs: NSEC, NSEC3 and NSEC3PARAM

Source: NLnet Labs

DNSSEC Deployment

Country Top Level Domains



Generic TLDs Secured with DNSSEC

.com	507,430
.net	90,910
.org	54,129
.info	15,181
.edu	67
…	

http://www.internetsociety.org/deploy360/dnssec/statistics/

Source: Internet Society

Securing IP Origins

RPKI

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

How can an Attacker Try to Hijack Your IP Prefix?

You

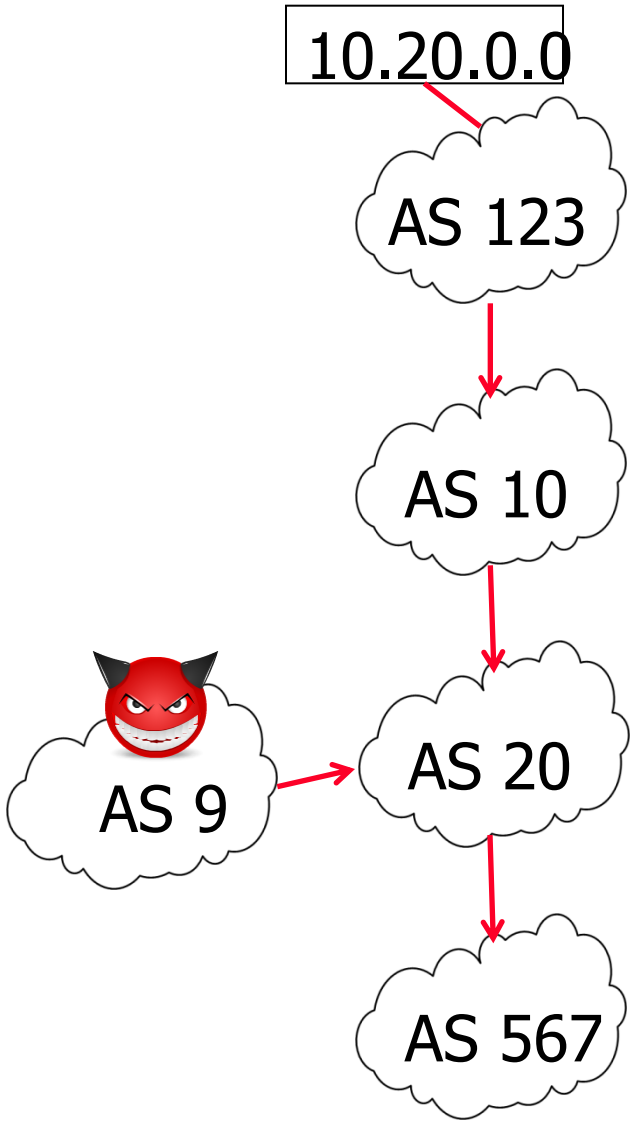
- AS 123 announces IP prefix 10.20.0.0/16

Me

- Receive a BGP update with path 123, 10, 20, 567
- Receive a BGP update with path 9, 20
- Receive a more specific prefix

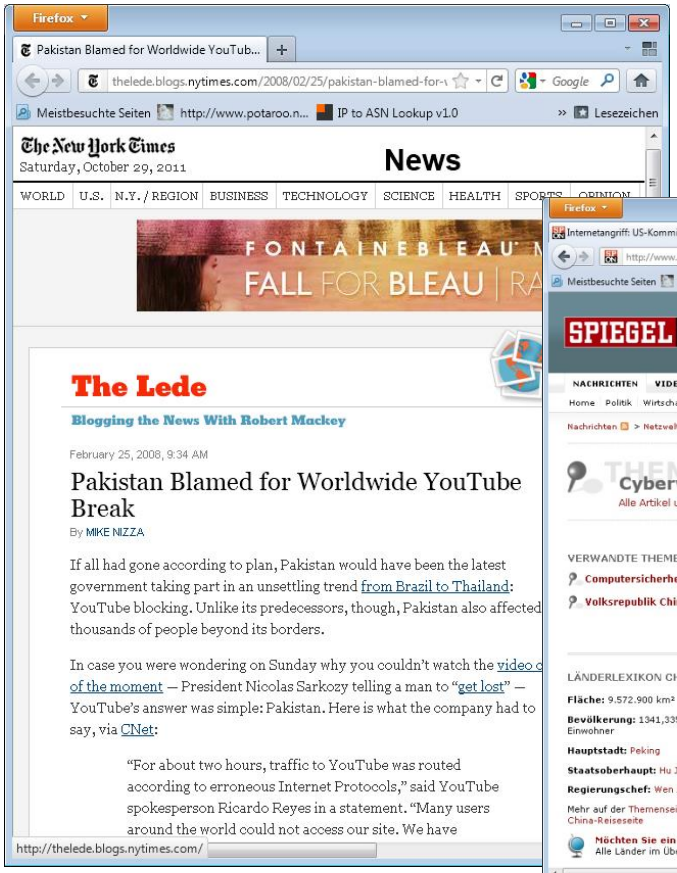
Attacker

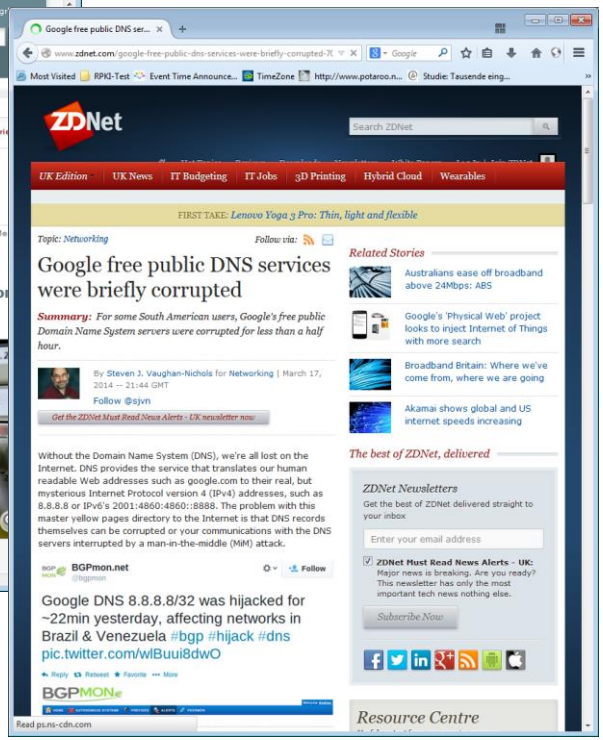
- Announces 10.20.0.0/16
- Announces 10.20.30.0/24



Hijacks in the Real World?

Prominent examples





Caveat: Reasons may also be misconfiguration ;-)

Outline

Part 1 - Background:

- How can you prevent your network from prefix hijacking?
- How can you perform prefix origin validation?

⇒ All of this in a cryptographically strong way

Part 2 - RTRlib:

- Open-source implementation of validation functions
- How to monitor your network with RTRlib

Problem

- BGP is based on trust between peers

Implications

- Any BGP speaker can claim to own an IP prefix
- Any BGP speaker can modify the AS path
- Receiver of a BGP update cannot verify the correctness of the data

Compromise

- Filtering
- Considering data of the Internet Routing Registry

⇒ This is not enough anymore!

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 hop-wise
   - BGP routers validate hops

Challenges

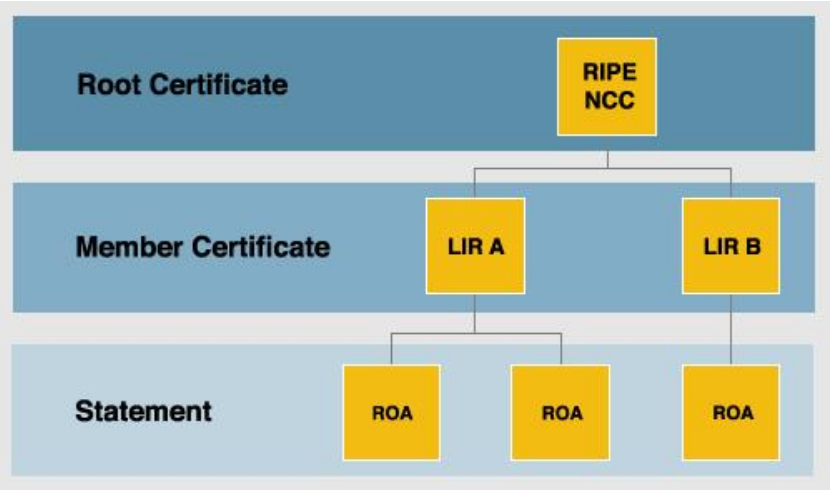
- Cryptographic operations are complex
- Minimal additional load at routers

In the following we concentrate on 1.

Resource Public Key Infrastructure (RPKI)

- 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: RIPE

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

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Routing Origination Authorization (ROA)

- 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

Valid from
01/10/2012
to
01/10/2013
+ E2E Cert

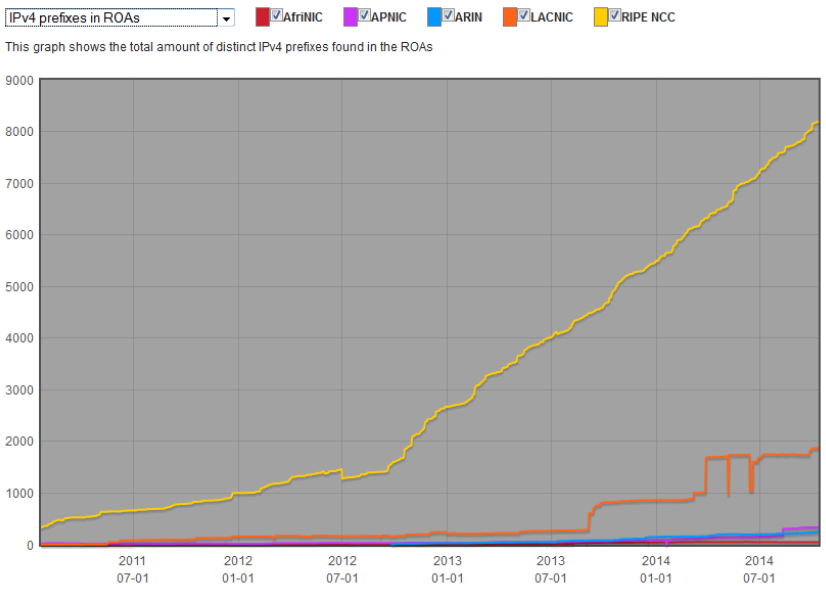
10.20.0.0/16-24 -> AS 123
80.90.0.0/16-16 -> AS 123

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

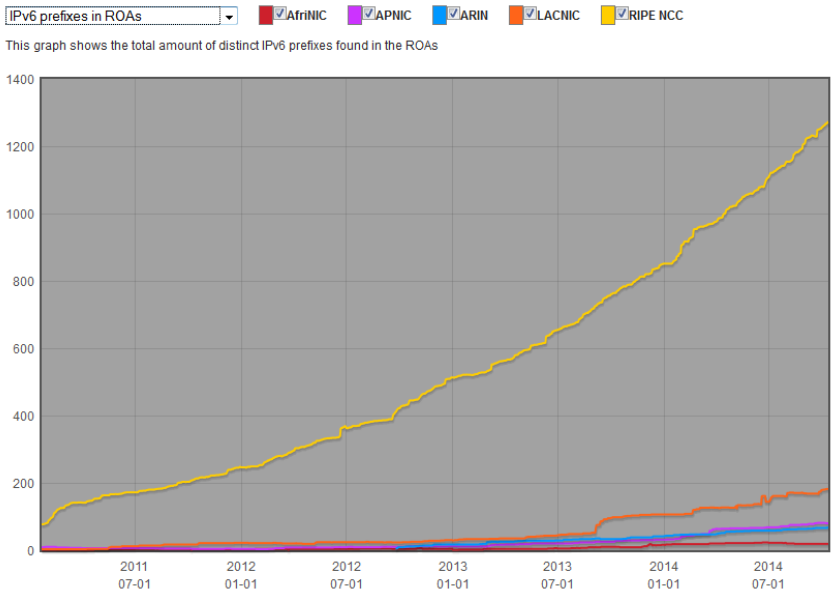
RPKI & ROA

- All certificates including ROAs will be published in RPKI repositories
  - Each RIR (including RIPE NCC ;) operates one
  - ISPs can maintain their 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

Current Deployment: # IP prefixes in ROAs



IPv4



IPv6

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

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Prefix Origin Verification & RPKI

Validation process consists of two steps

1. Validation of ROAs

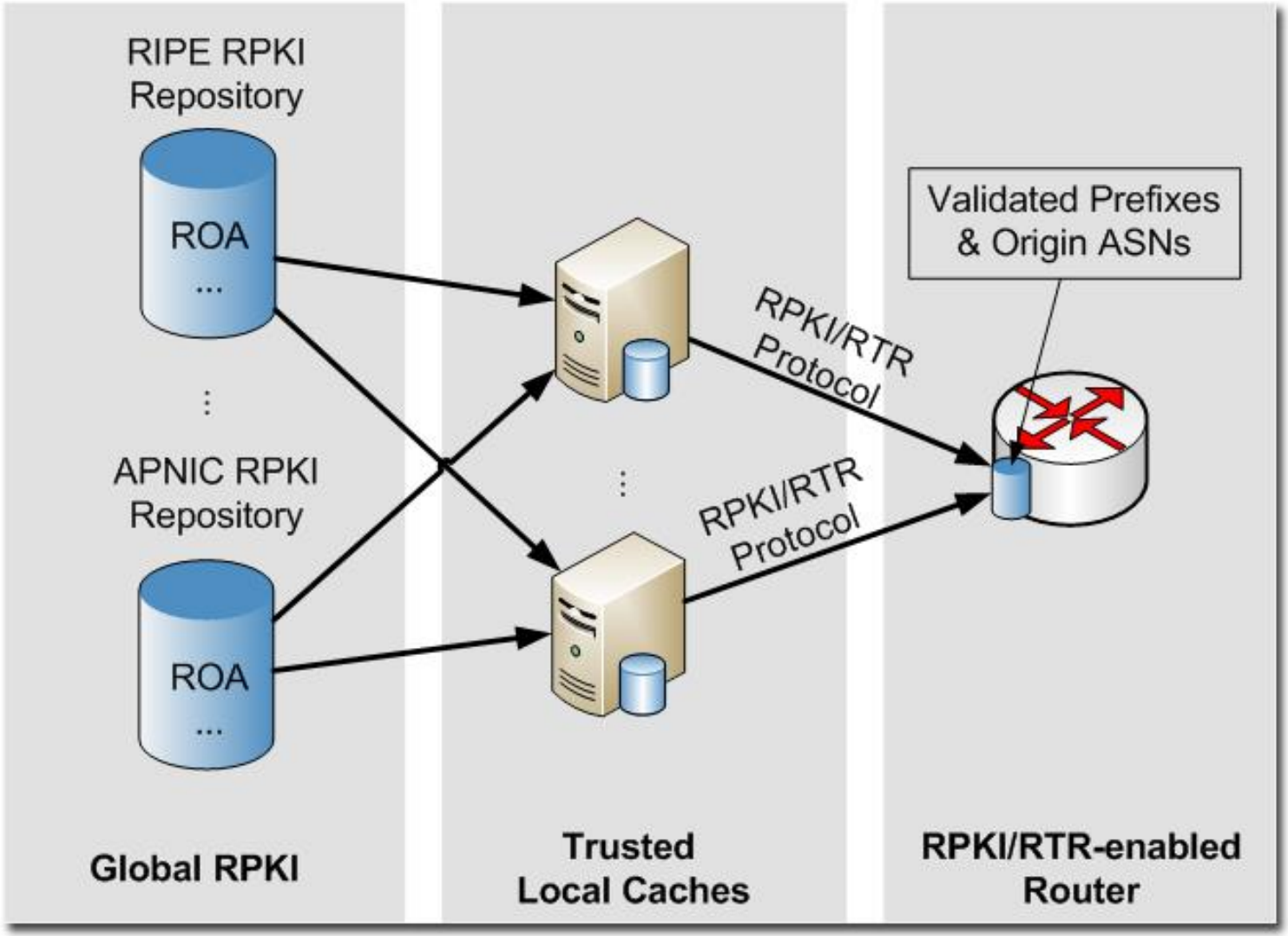
- Performed at external cache

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 the outcome with BGP policies

Architecture Overview



Validation Outcome

Validation of an ASN/Prefix pair against RPKI results in either

Valid

- If at least one valid ROA exists that covers the announced prefix and matches the BGP origin AS, with max length less or larger than the BGP prefix length

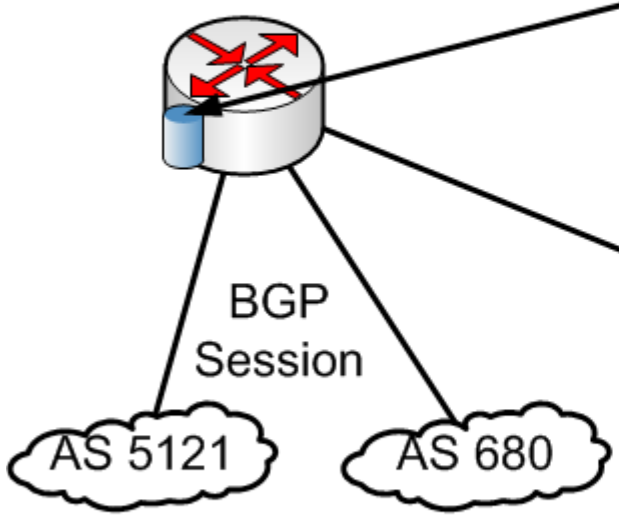
Invalid

- If no covering ROA matches the BGP origin AS or the announced prefix is more specific

Not Found

- If no covering ROA exists

Validation Outcome - Examples



Validated Prefixes & Origin ASNs from RTR Cache

ASN	Prefix	MaxLen
100	10.20.0.0/16	24
200	50.60.70.0/24	32
200	50.60.70.0/24	28

Received BGP Updates and validation outcome

ASN	Prefix	Validation outcome
100	10.20.0.0/16	Valid
100	10.20.0.0/18	Valid
100	10.20.5.0/28	Invalid
200	10.20.0.0/16	Invalid
300	1.2.0.0/16	Not found

How to Participate?

Create ROAs

- Login into the RIPE LIR Portal https://certification.ripe.net/
- Enable group "certification"
- Add your ROAs to the RPKI

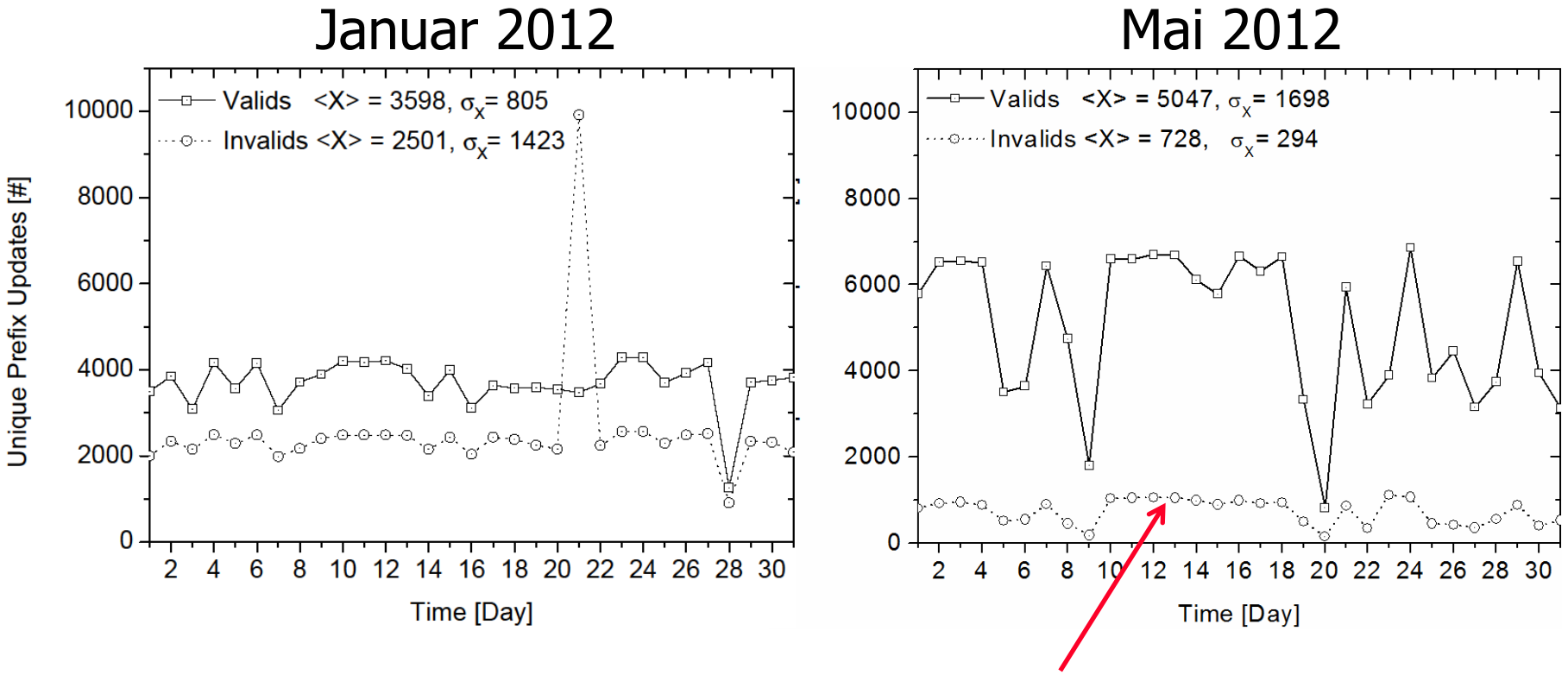
Setup RTR Cache Server (or use an external)

- Download the RIPE validator, for example
  - http://www.ripe.net/lir-services/resource-management/certification/tools-and-resources

Enable your BGP Router

- Juniper: official support for RPKI since release 12.2.
- Cisco: official support for ASR1000, 7600, ASR903 and ASR901 in releases 15.2(1)S or XE 3.5

Valide vs. Invalide BGP Updates



Number of invalids decreases over time

Are these updates really hijacks??

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Some Common Pitfalls - Examples

Case 1: Missing Customer (or Sibling) Legitimation

- ROA created: 12.0.0.0/8-9 -> AS 7018
- AS 27487 announces 12.0.19.0/24
- AS 2386 announces 12.1.216.0/24

Both announcements are invalid if no ROAs exists

⇒ Consider sub-allocations, start most specific

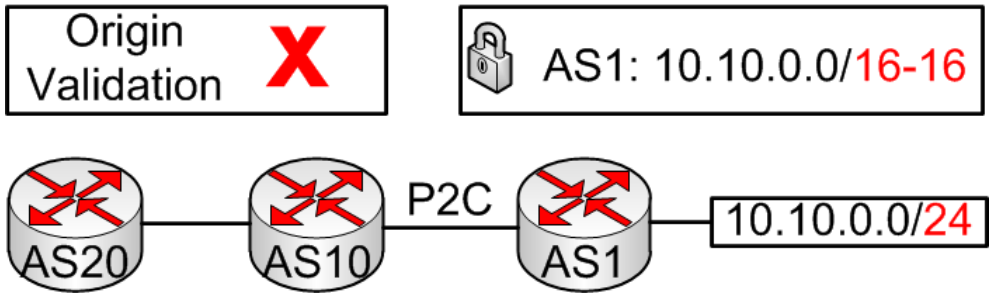
Case 2: (De-)Aggregation

- ROA created: 78.192.0.0/10-10 -> AS 12322
- Usual announcement: 78.192.0.0/10
- For 30 minutes: 78.192.10.0/24 …

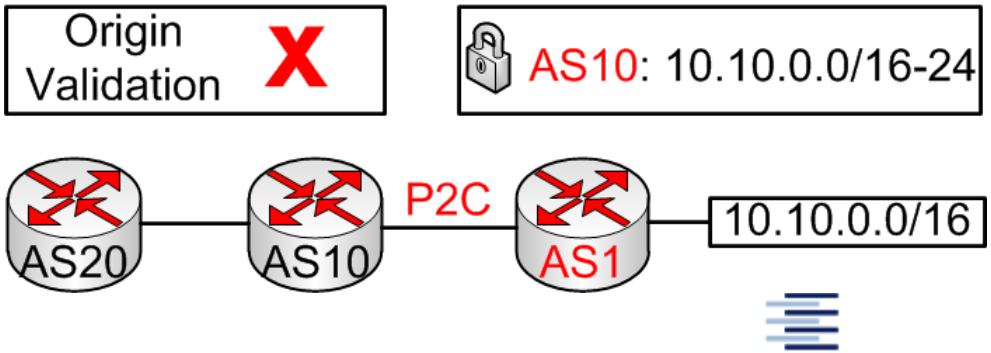
⇒ Configure the max ROA prefix length explicitly

Common Pitfalls – Overview (1)

Valid origin, announced prefix is more specific



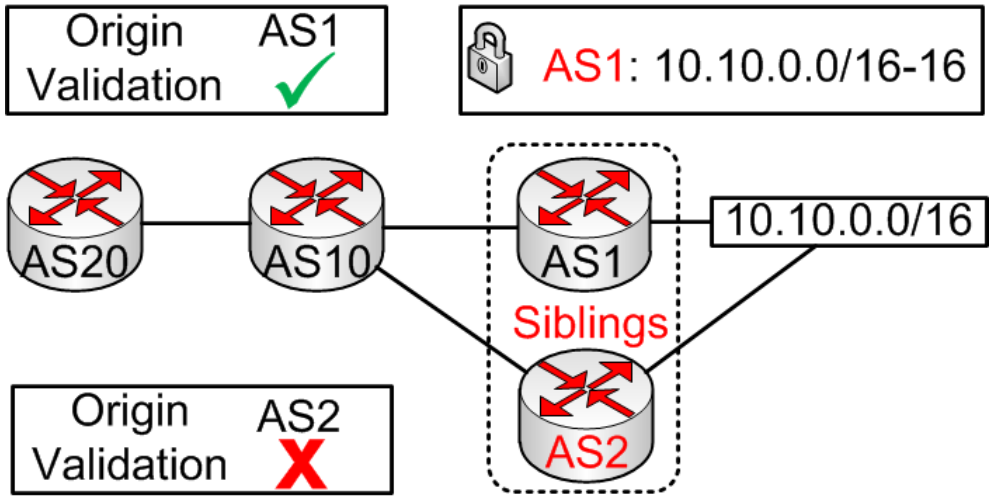
Provider does not consider customers



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Common Pitfalls – Overview (2)

Additional AS of a company is not authorized



Implementing the RPKI Router Part

RTRLIB

What is the RTRlib?

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

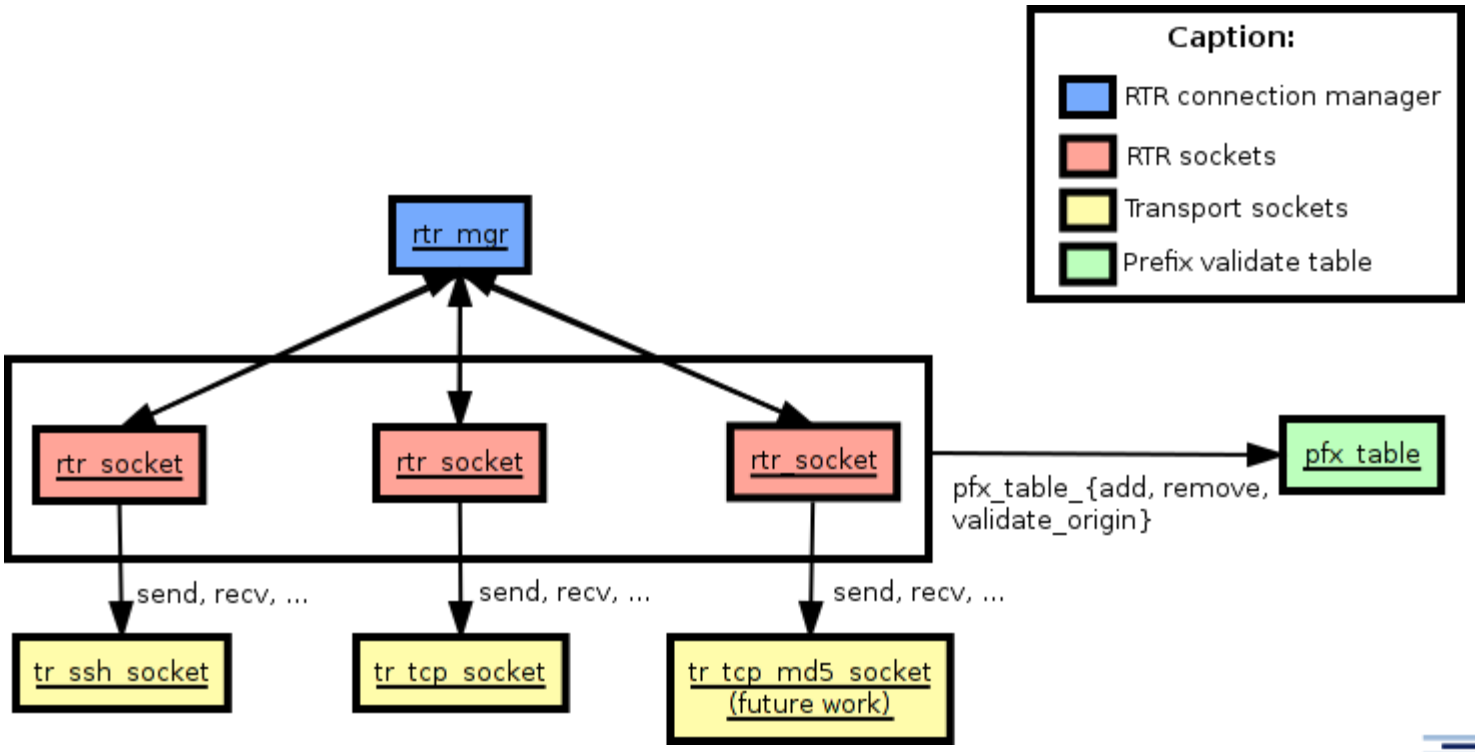
It's open-source: http://rpki.realmv6.org

Applications

- Extension of BGP daemons
  - In discussions with BIRD developers
- 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

Architectural Design

- Layered architecture to support flexibility



Example – Establish Transport

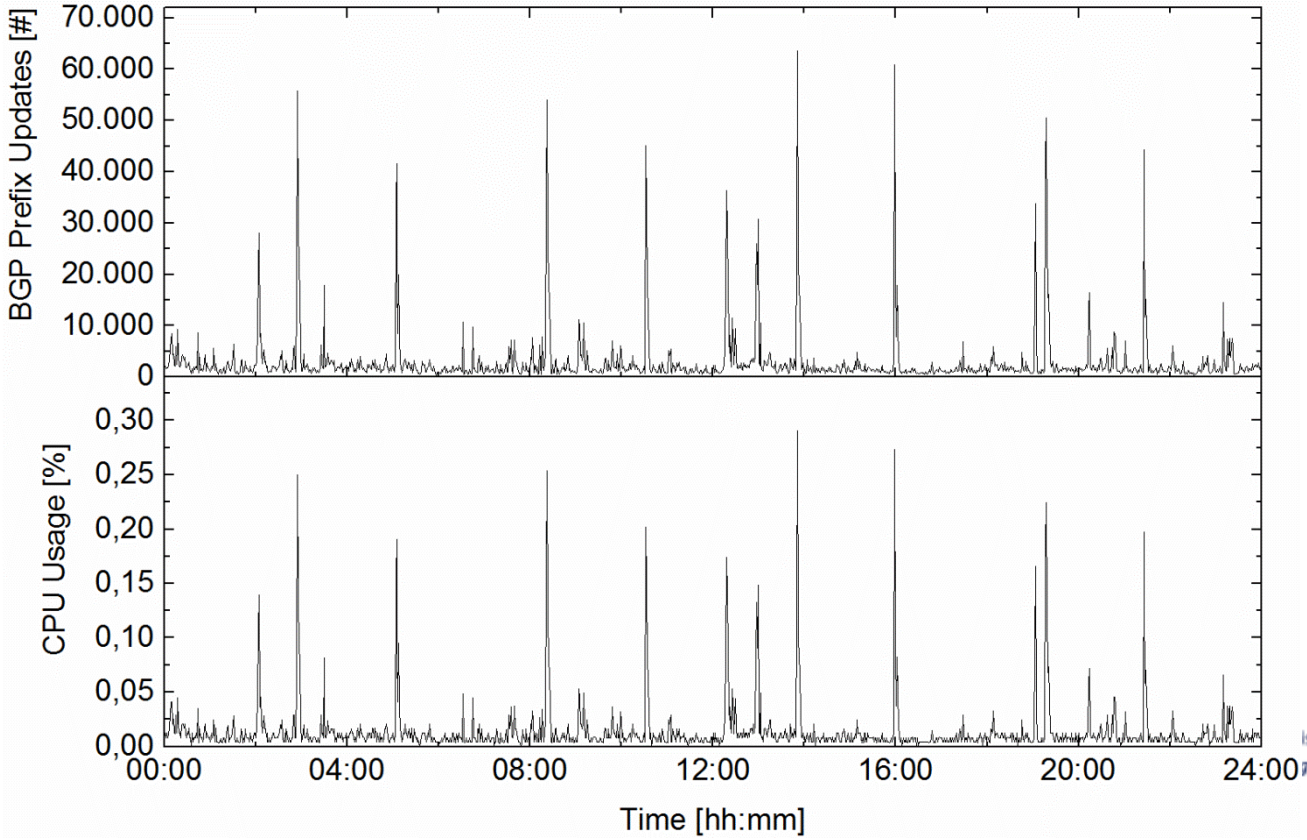
```
tr_socket* ssh_socket;                  //create a SSH connection
tr_ssh_config config = {
  "123.321.123.321",                    //IP
  22,                                   //Port
  "rpki_user",                          //SSH User
  "/etc/rpki-rtr/hostkey",              //Server hostkey
  "/etc/rpki-rtr/client.priv",          //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
```

Create Connection Manager and Perform Origin Validation

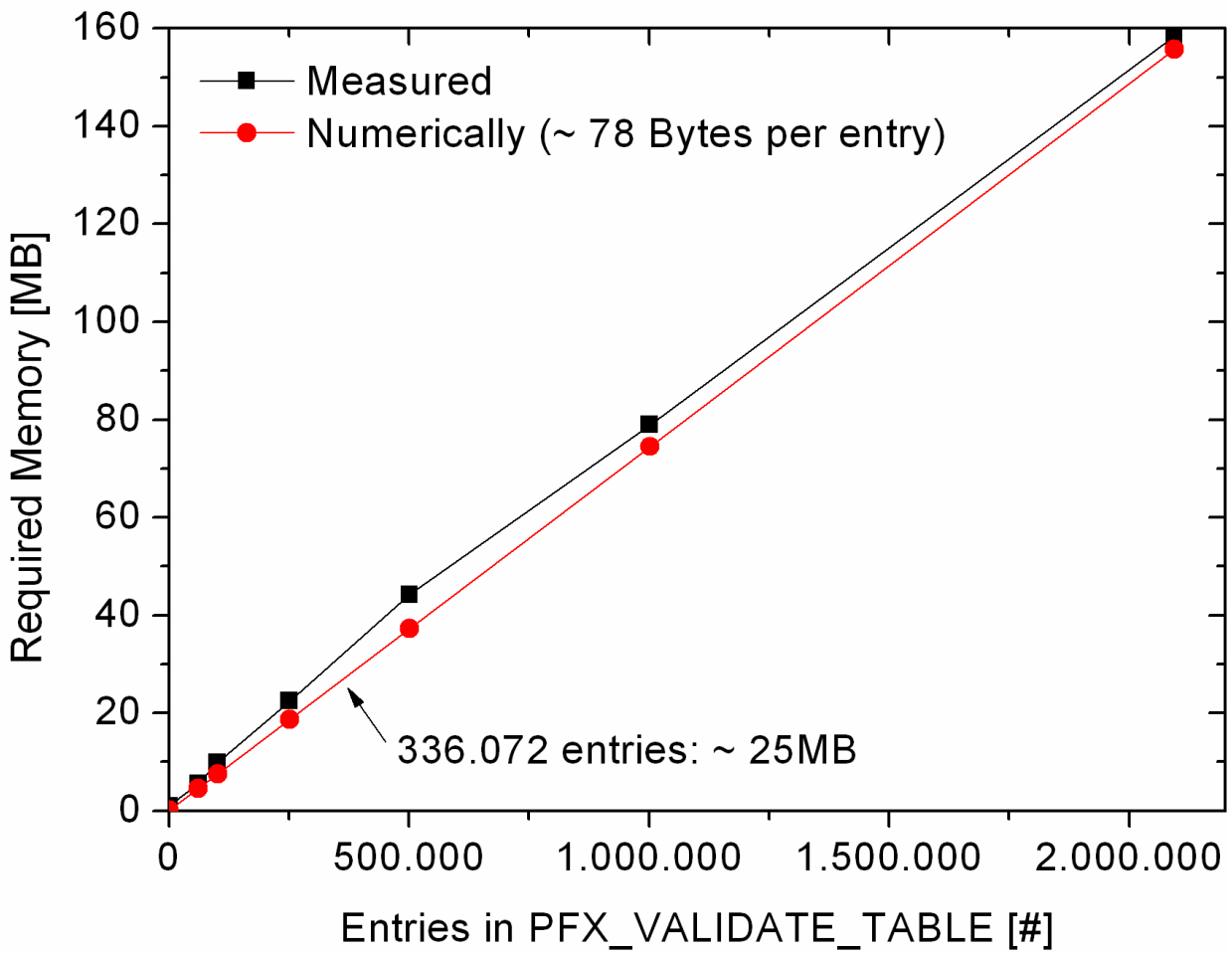
```
//init all rtr_sockets with the same settings
//srv.pool,polling_period,cache_timeout,update_fp,conn_f
rtr_mgr_init(&p0, 60, 120, NULL, 0, NULL, 0);

//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);
```

RPKI in the Wild – CPU-Load

- Test on commodity hardware
  - AMD Athlon 64 X2 CPU 4200+ and 2 GB RAM
- Live BGP update stream of a day



Memory Consumption

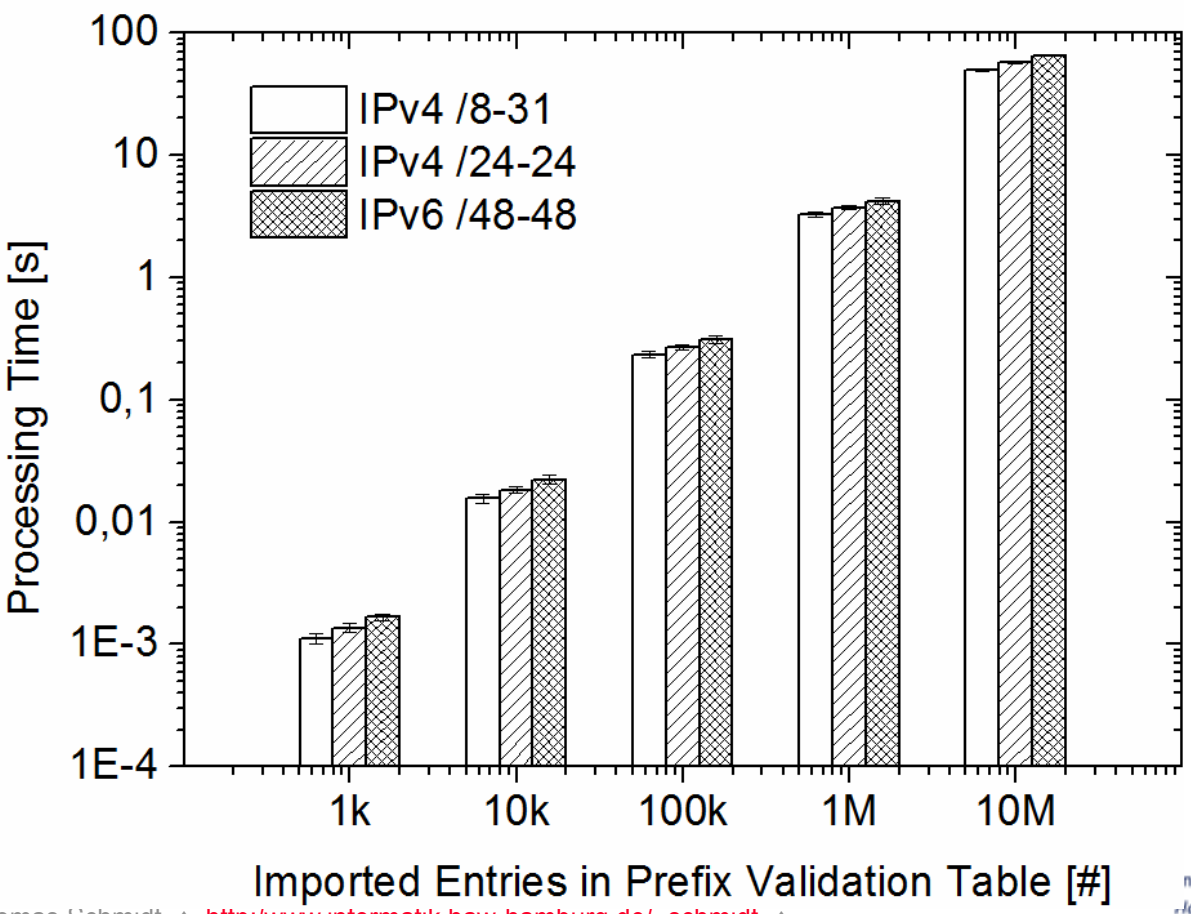


Side note:
Including
1 million entries
from a file takes
~4 seconds

10 million entries
~10 seconds

Delay from Loading RPKI Data

Motivation: Boot of a router, reset of cache server

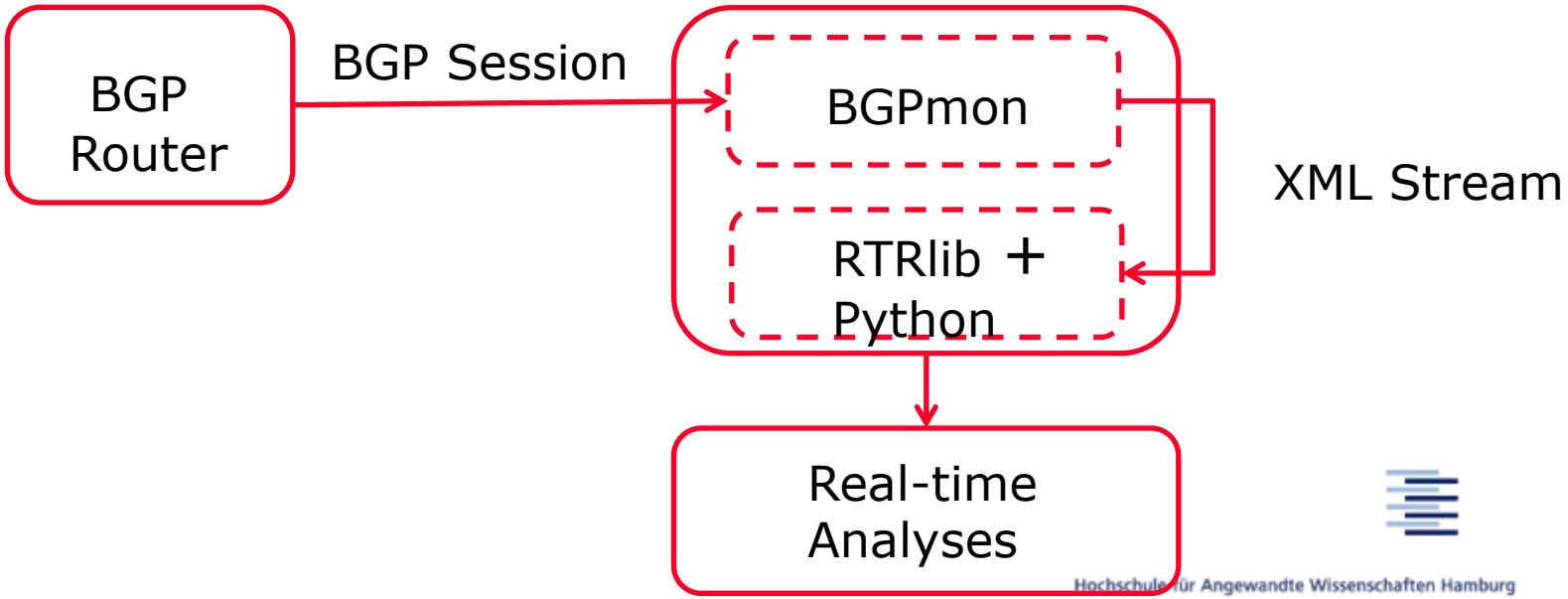


Using RTRlib for Monitoring

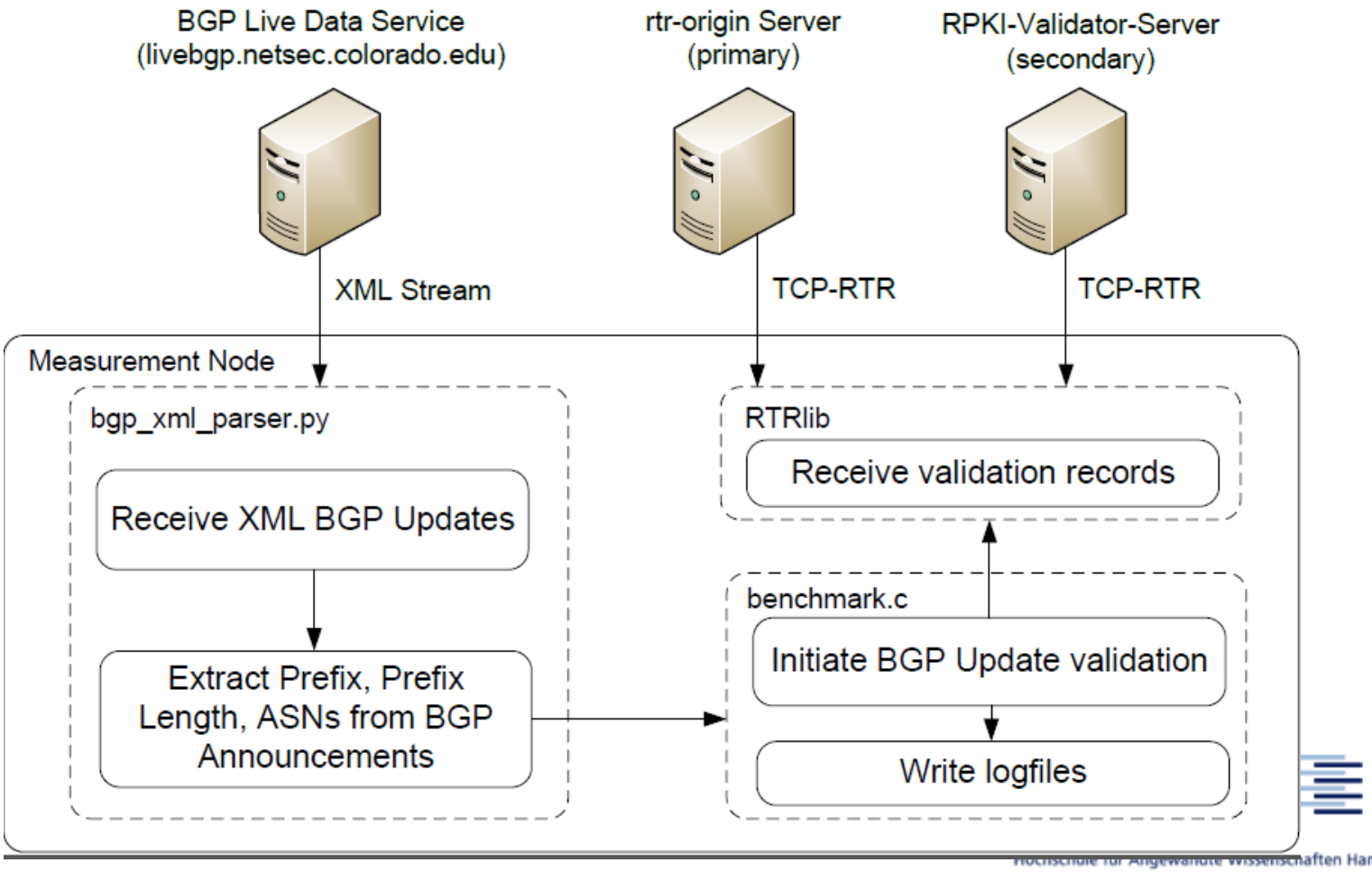
Objective: Emulate update validation of your BGP peer

Setup – No Firmware Change at Your Router:

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



Example: Monitoring Scenario

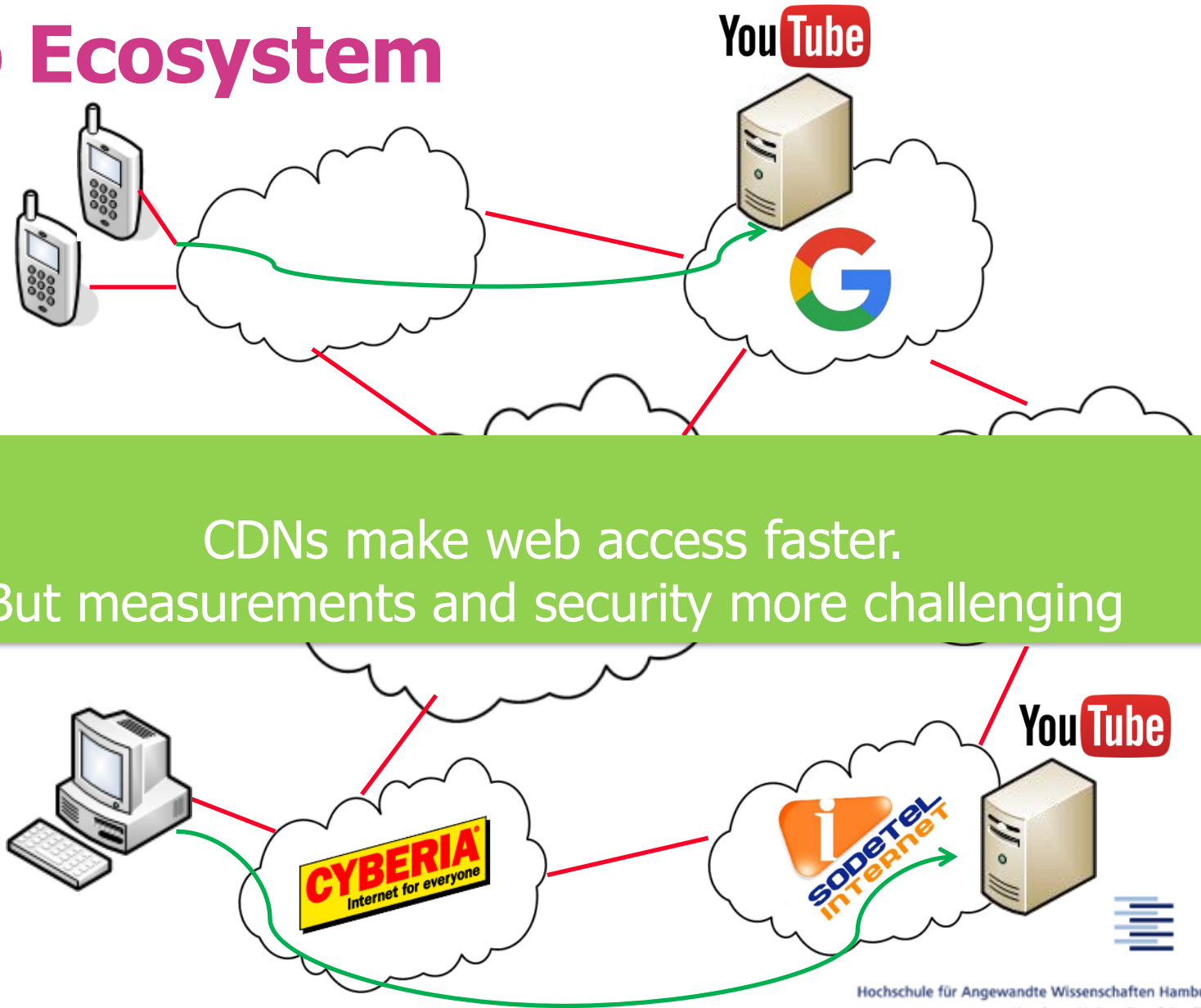


Objectives [ACM HotNets'15]

Empirically explore the relationship between web hosting infrastructure and RPKI deployment.

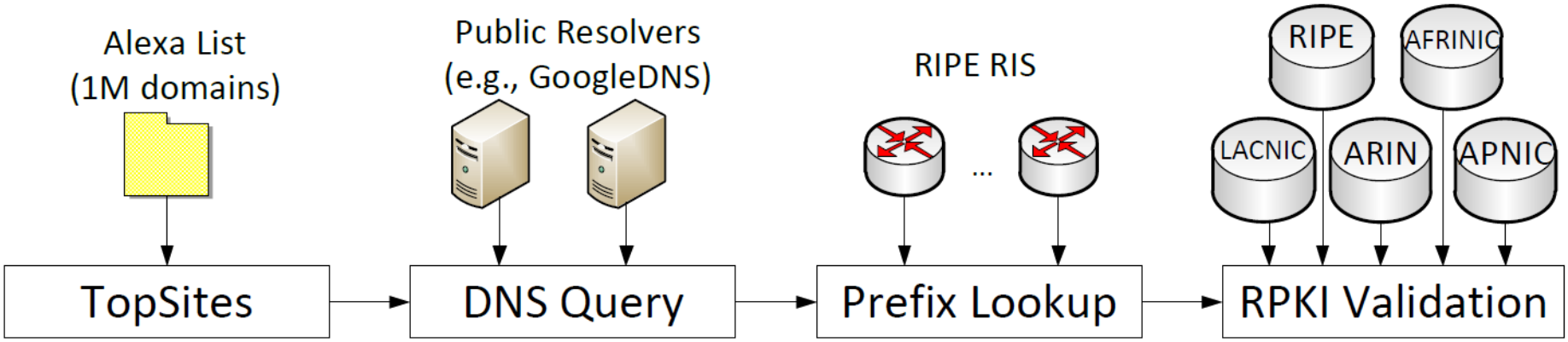
Which web servers are secured by the RPKI?

Web Ecosystem



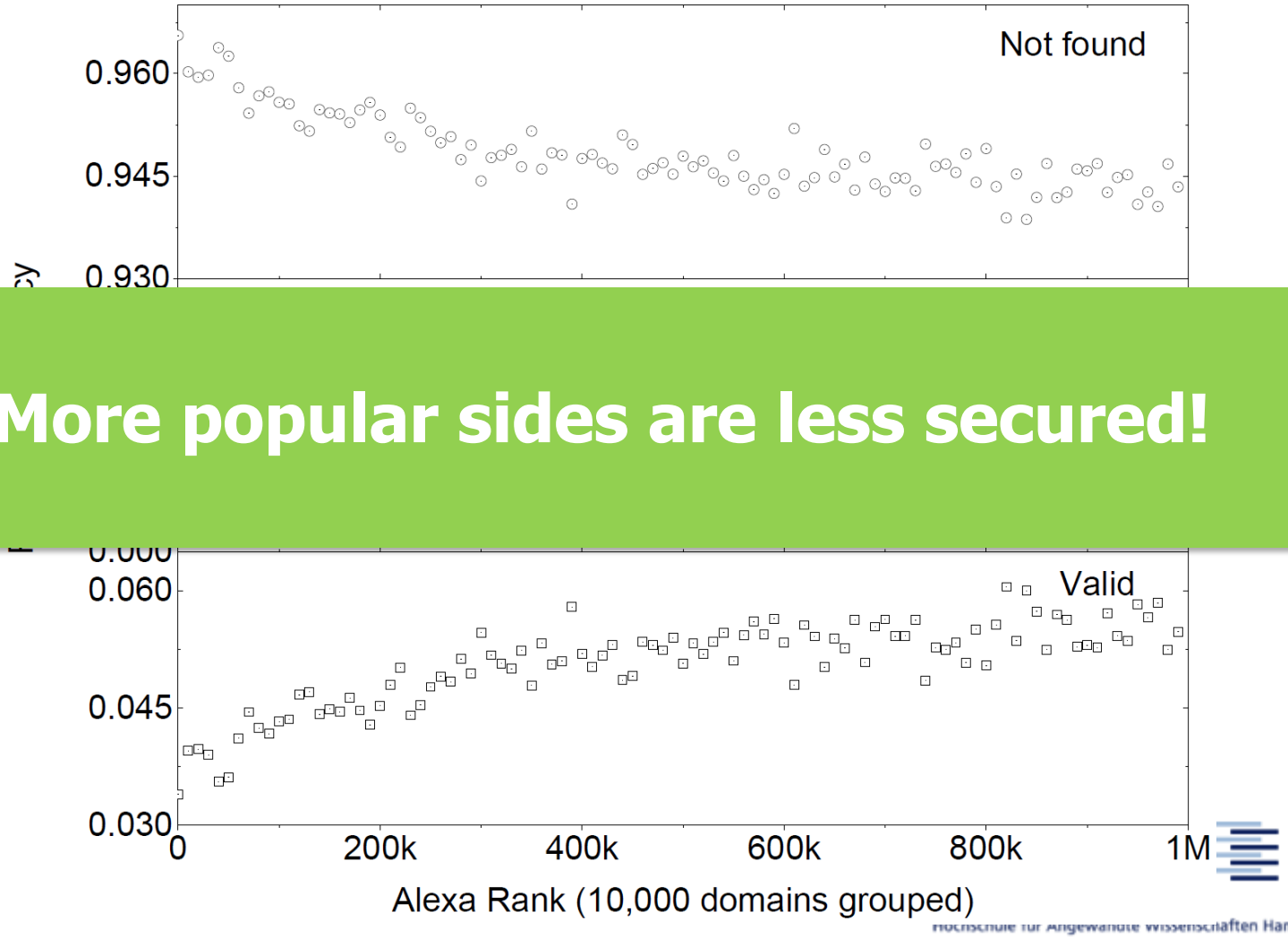
CDNs make web access faster.
But measurements and security more challenging

Measurement Methodology



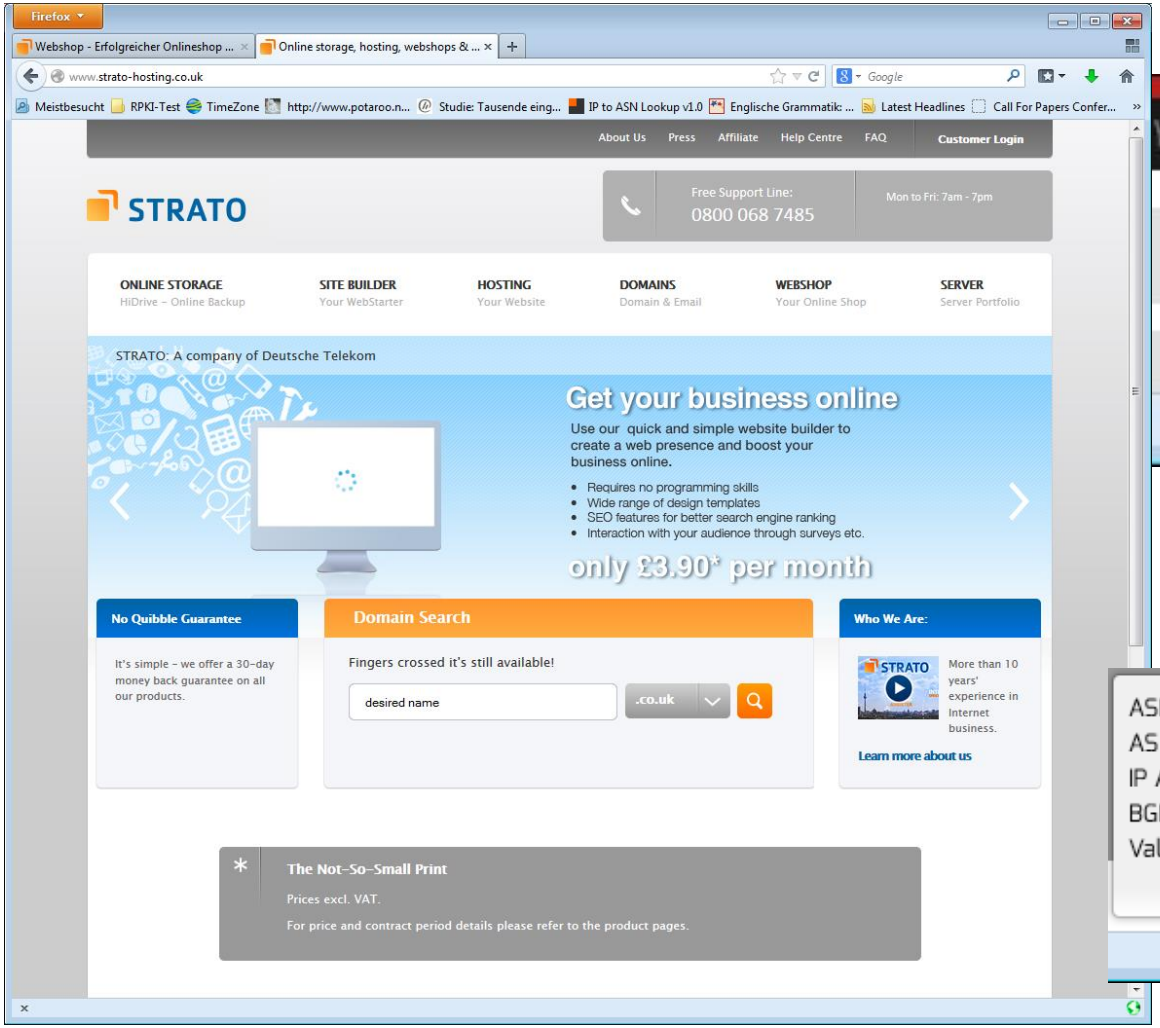
Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

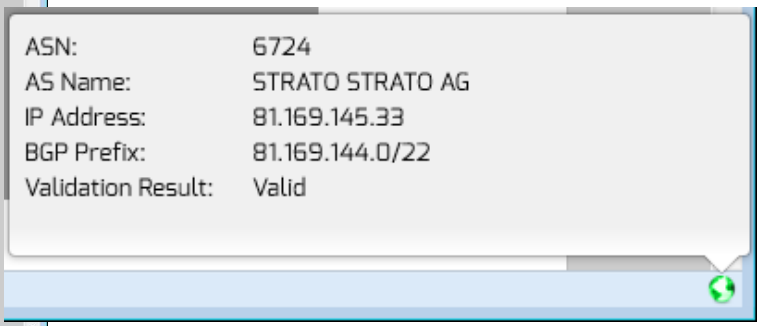
RPKI Validation Outcome for 1M Web Sites



More popular sides are less secured!

Validation in Web-Browser





Software Download

- RTRlib – The RPKI RTR Client C Library
  http://rpki.realmv6.org/

- RIPE NCC RPKI Validator (Cache Server)
  http://www.ripe.net/lir-services/resource-management/certification/tools-and-resources

- rpki.net project
  https://trac.rpki.net/

- RPSTIR – BBN RPKI Validator
  http://sourceforge.net/projects/rpstir/

- Firefox Add-on RPKI Validator
  https://addons.mozilla.org/addon/rpki-validator/

Further (General) Information

RIPE NCC Information About RPKI

- Entry point
  http://www.ripe.net/certification
- Cisco and Juniper configuration examples
  http://www.ripe.net/lir-services/resource-management/certification/router-configuration

Overview Article

- Geoff Huston: "Resource Certification", In: The Internet Protocol Journal, Volume 12, No.1, Cisco, 2009,
  http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_12-1/121_resource.html

Further (Detailed) Information

IETF – Standardization

- Secure Inter-Domain Routing Group (SIDR)
  http://datatracker.ietf.org/wg/sidr/
- Draft "BGP Prefix Origin Validation"
  http://tools.ietf.org/html/rfc6810
- Draft "The RPKI/Router Protocol"
  http://tools.ietf.org/html/rfc6811

RPKI/RTR Implementation Report

- http://tools.ietf.org/html/rfc7128

RTRlib Reference Report

- M. Wählisch, F. Holler, TC. Schmidt, J. H. Schiller, *RTRlib: An Open-Source Library in C for RPKI-based Prefix Origin Validation*, In: Proc. of USENIX Security Workshop CSET'13, Berkeley, CA, USA:USENIX Assoc., 2013.

Securing Application Endpoints

DANE

E2E Application Security

- Application transport today provides encryption, integrity protection, privacy, +++
  - Examples are TLS, DTLS, IPSec, S/MIME, SSH, …
- Secure channels require bootstrapping
  - Built from CA hierarchies
  - Relies on (a) trust of root CAs, and (b) integrity of trust delegation
  - One compromise invalidates the complete chain of trust

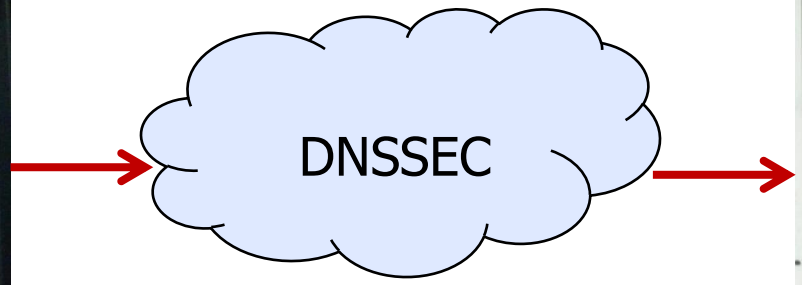
Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Threats & Flaws of the CA Approach

- CAs are vulnerable!
  - No namespace constraints - any CA can issue certificates for any entity on the Internet
  - July 10,2011 an attacker created a wildcard certificate for Google (DigiNotar)
- Tolerance & delegation may lead to unexpected endpoints
  - Often self-signed or expired certificates
  - CDNs officially terminate TLS sessions
- We learn CA keys out of band
  - Local misuse by configuration („TLS-proxies")
- Key revocation problem
  - Revocation lists slow, not scalable
  - After compromise, everybody wants to revoke ➦Heartbleed!

DNS Based Authentication by Named Entities (DANE)

- Replace trust from CAs to DNSSEC Infrastructure
- Built on top of DNSSEC: Defines new TLSA DNS record (RFC 6698)
  - May constrain the CA, or
  - Provide certificate directly from DNS



TLSA Records in DNSSEC

- DNS record type to authenticate remote endpoints in transport: SSL/TLS (web, mail, …)
- TLSA key: _port._proto.domain.tld –

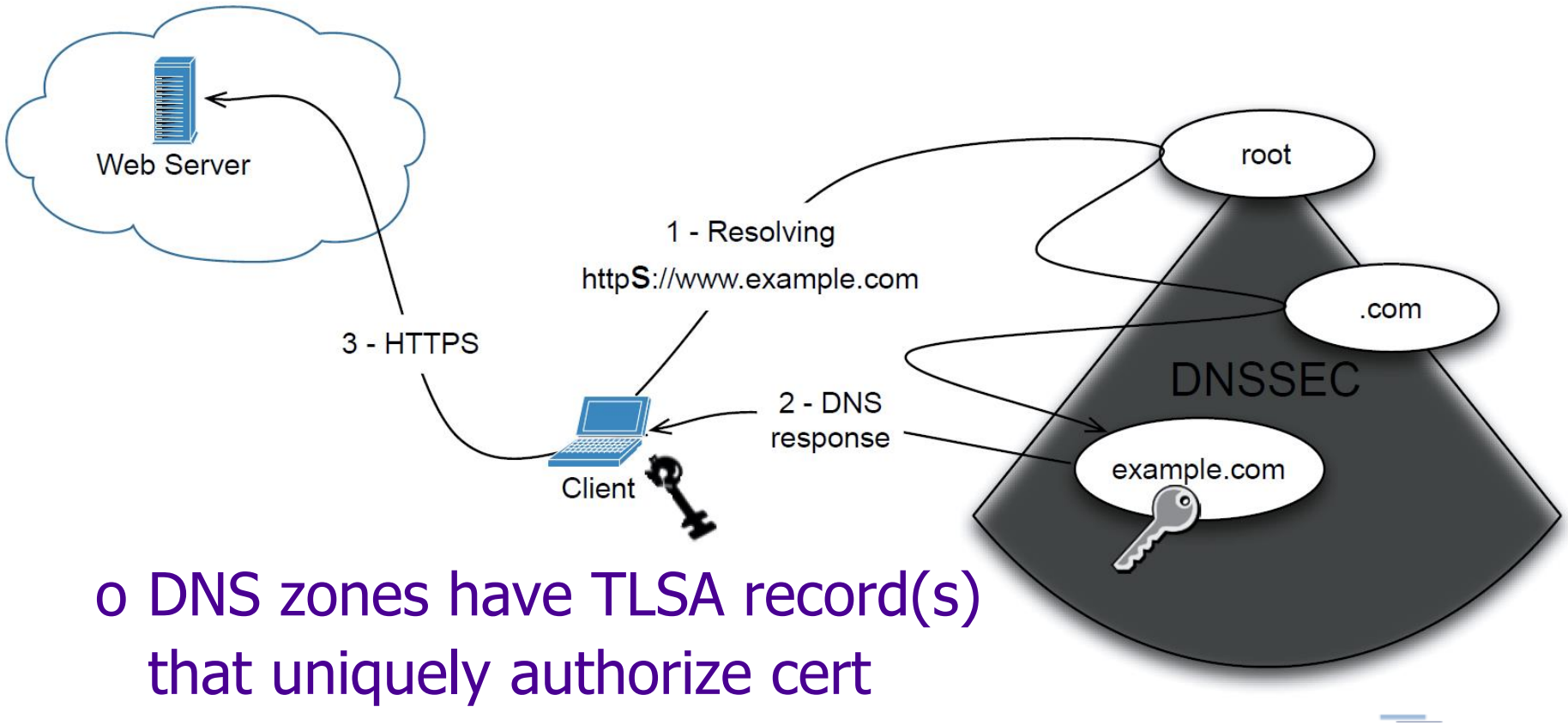
  `_443._tcp.good.dane.verisignlabs.com`

- TLSA value: Meta-data + Certificate Association Data (raw cert data in hex) –

```
(0 0 1 d2abde240d7cd3ee6b4b28c54df034b9
7983a1d16e8a410e4561cb106618e971)
```

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

DANE verification process



- DNS zones have TLSA record(s) that uniquely authorize cert used by servers

Source: Eric Osterweil from Verisign Labs

DANE Résumé

Promise: Providing security between authorized transport endpoints (Web, Mail, …)

Reality:

- Server-centric security toolset – mainly inter-SMTP mail security
- Emerging building blocks for 'Secure Email' with clients (➦ Thunderbird)
- Internet Society (ISOC) has a deployment program called Deploy 360:
  `http://www.internetsociety.org/deploy360/resources/dane/`

Enhancing Certificate Reliability

CT - CERTIFICATE TRANSPARENCY

Where the CA Approach Falls Short

A CA in the trust chain can

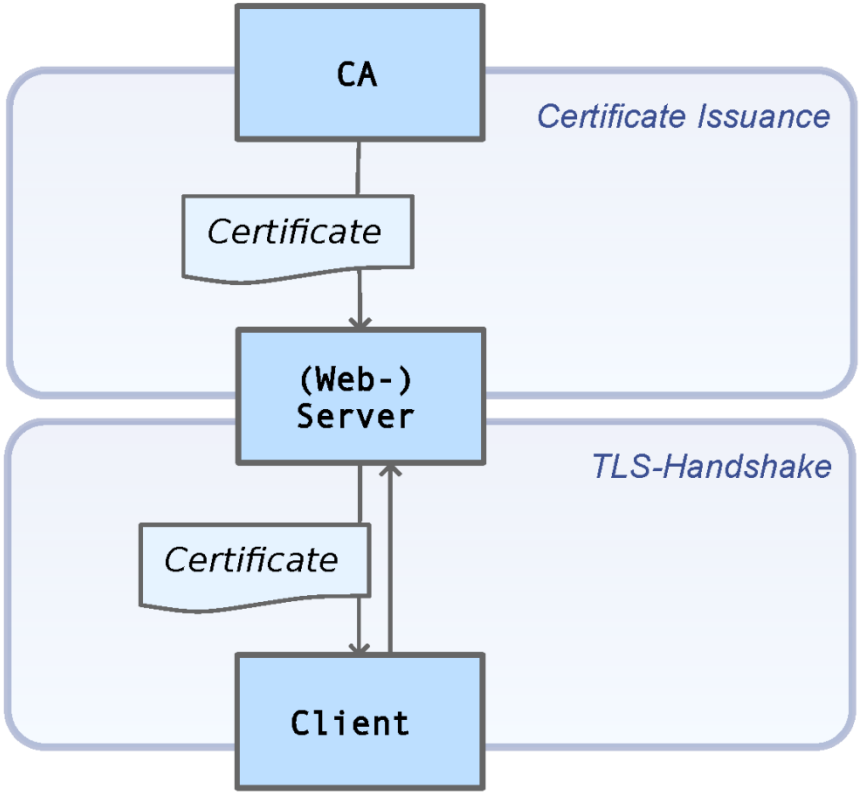
- Certify any resource
- Remain in secrecy
- Lie about time of issuing

A Client alone cannot

- Verify correctness of the CA

Public trust anchors can help

- DANE per name
- CT per certificate

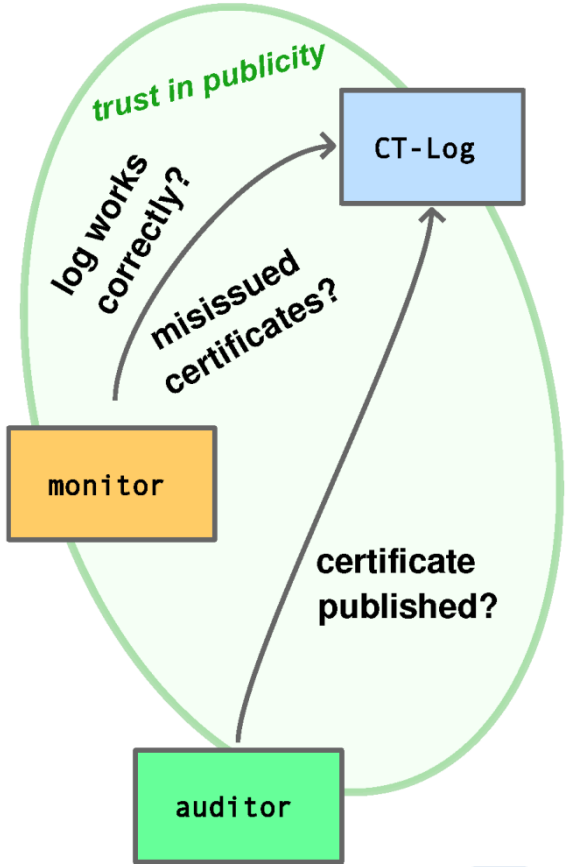


Graphics by Theodor Nolte

CT: Replicate Certificates in Public

Approach (Google, RFC6962):

- Publish certificates to independent CT-Logs
  - Purpose of "monitoring"
  - Requires valid trust chain
- Logs promise to
  - Provision certificate history online
  - Maintain immutable entries
  - Hold correct time-stamps: Returns Signed Certificate Timestamps (SCT)
- Clients check logs
  - Purpose of auditing
  - SCT serves as log promise
  - Refuse unpublished or incorrect certificates



Graphics by Theodor Nolte

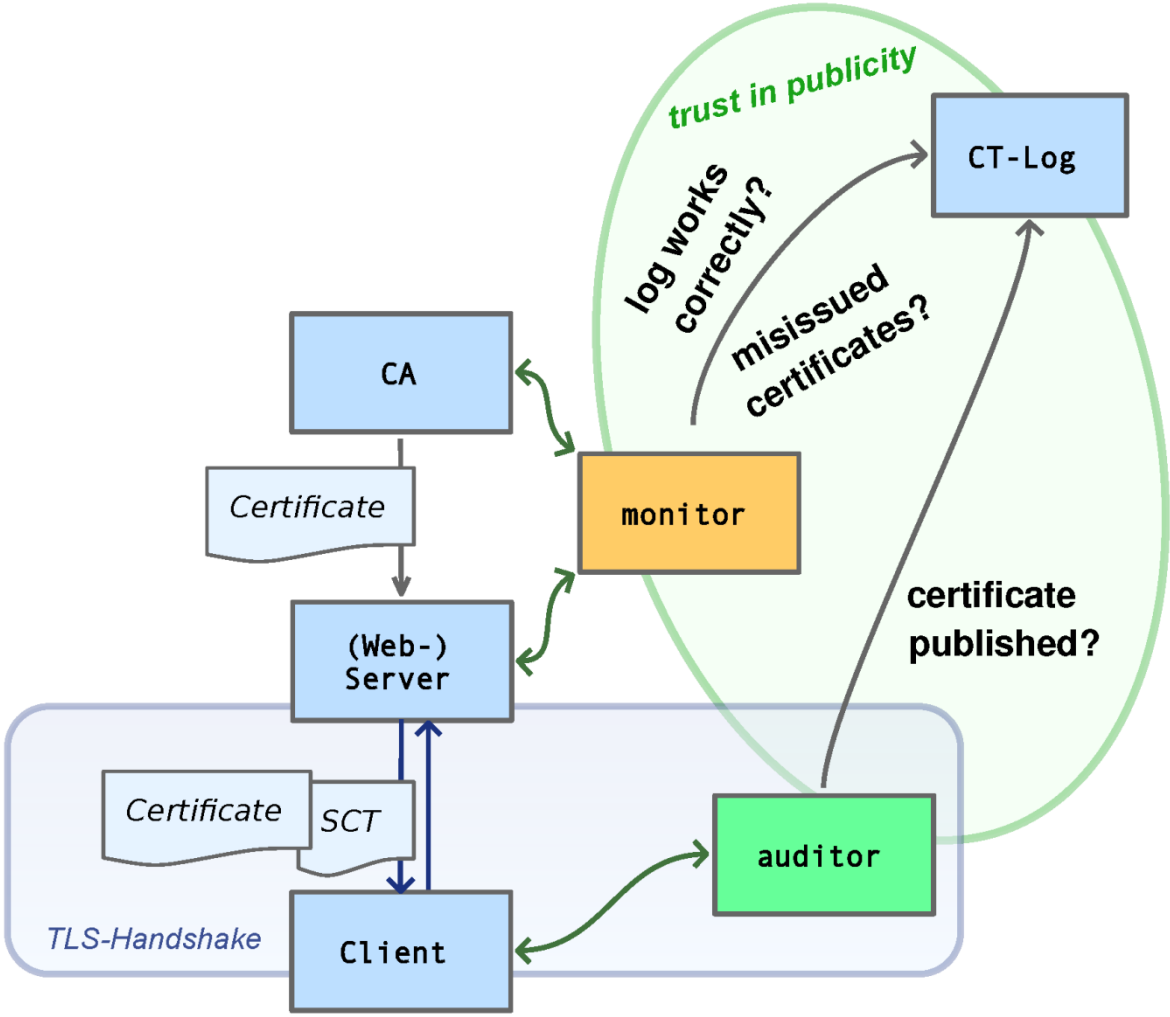
CT Enforces Visibility

Publication/Monitoring

- o CAs
- o Resource Owners
- o 3rd Parties

Verification/Auditing

- o Clients based on SCTs



Graphics by Theodor Nolte

Résumé on CT

- CT makes the use of TLS certificates transparent
  - CAs and resource owners can publish
  - Clients should check/enforce publication
  - Integrity should be monitored, forgery becomes visible
- Technically issuing of illegitimate certificates remains unhindered
- Privacy issue of CT
  - Logs see certificate queries
  - May know about each TLS request

Bibliography

- Drafts, RFCs:
  tools.ietf.org, http://www.rfc-editor.org
- Material for DNSSEC:
  http://www.dns-school.org/
- RPKI deployment:
  Matthias Wählisch, Robert Schmidt, Thomas C. Schmidt, Olaf Maennel, Steve Uhlig, Gareth Tyson: *RiPKI: The Tragic Story of RPKI Deployment in the Web Ecosystem,* In: Proc. of Fourteenth ACM Workshop on Hot Topics in Networks (HotNets), ACM : New York, Nov. 2015.