



# Internet Infrastructure Security

Thomas C. Schmidt

t.schmidt@haw-hamburg.de

HAW Hamburg, Dept. Informatik



# 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

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

## We have trusted

- ⇒ Name resolution (DNS)
- ⇒ Packet delivery (Routing & Forwarding)
- ⇒ Application origination (Certification ?)



# Who is Involved?

## o DNS:

- Recursive resolvers
- Caches
- Authoritative nameservers

## o Routing:

- Control plane: many BGP speakers
- Forwarding plane: eyeball, transit & origin ISPs

## o Application:

- Application server
- Indirect (hidden) contributors
- Certification authority ?



# Who Would Do Harm?



## Securing Names

# DNSSEC



# Attacking Names

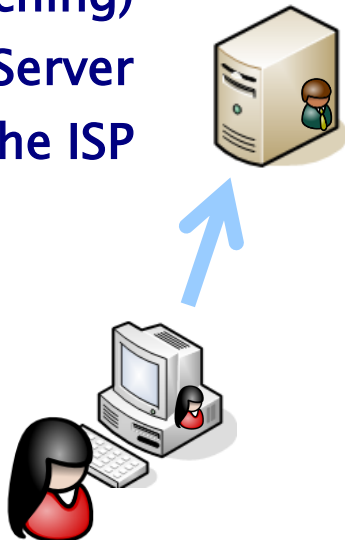


**www.  
mybank.  
com**



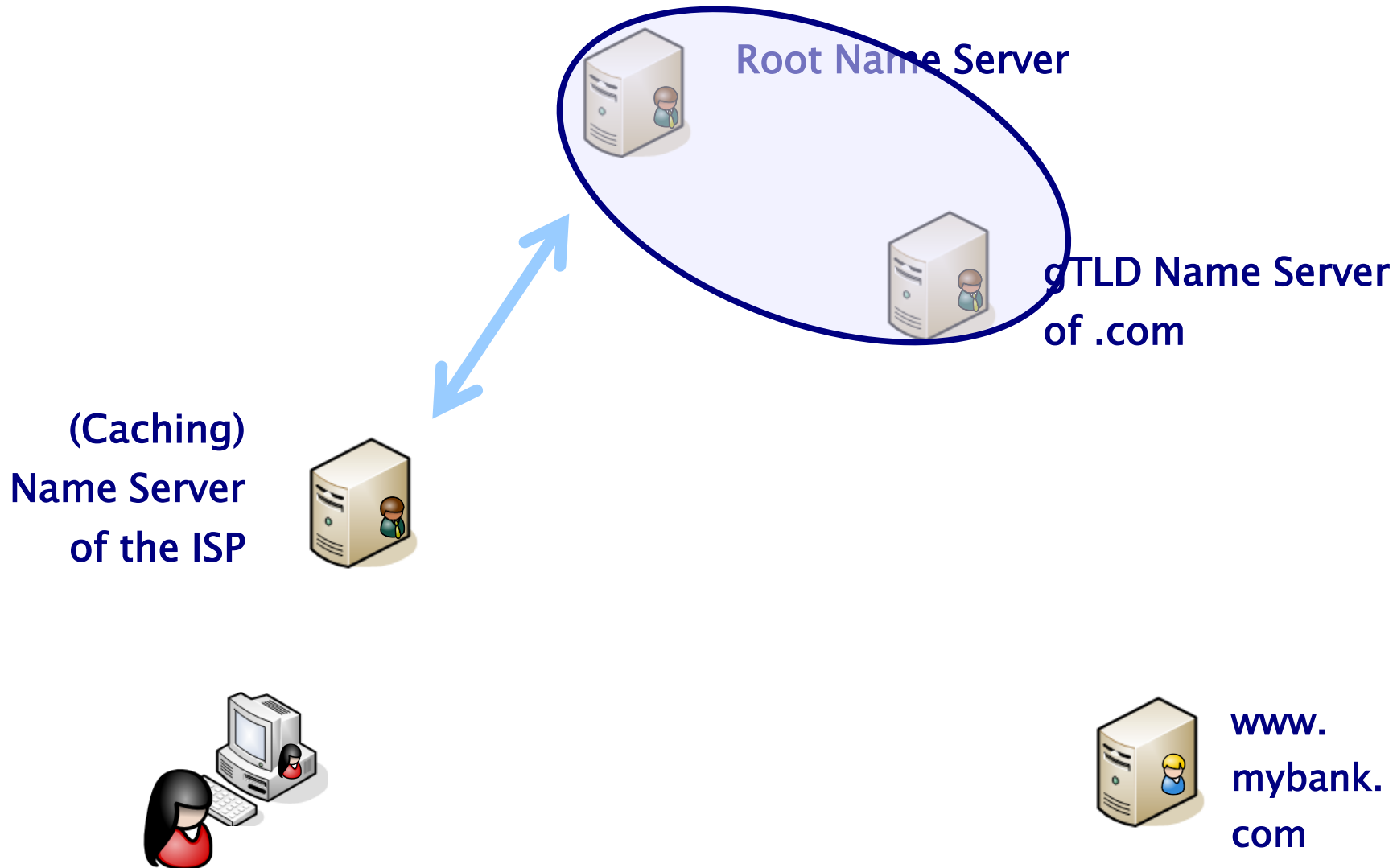
# Attack Vectors (1)

(Caching)  
Name Server  
of the ISP

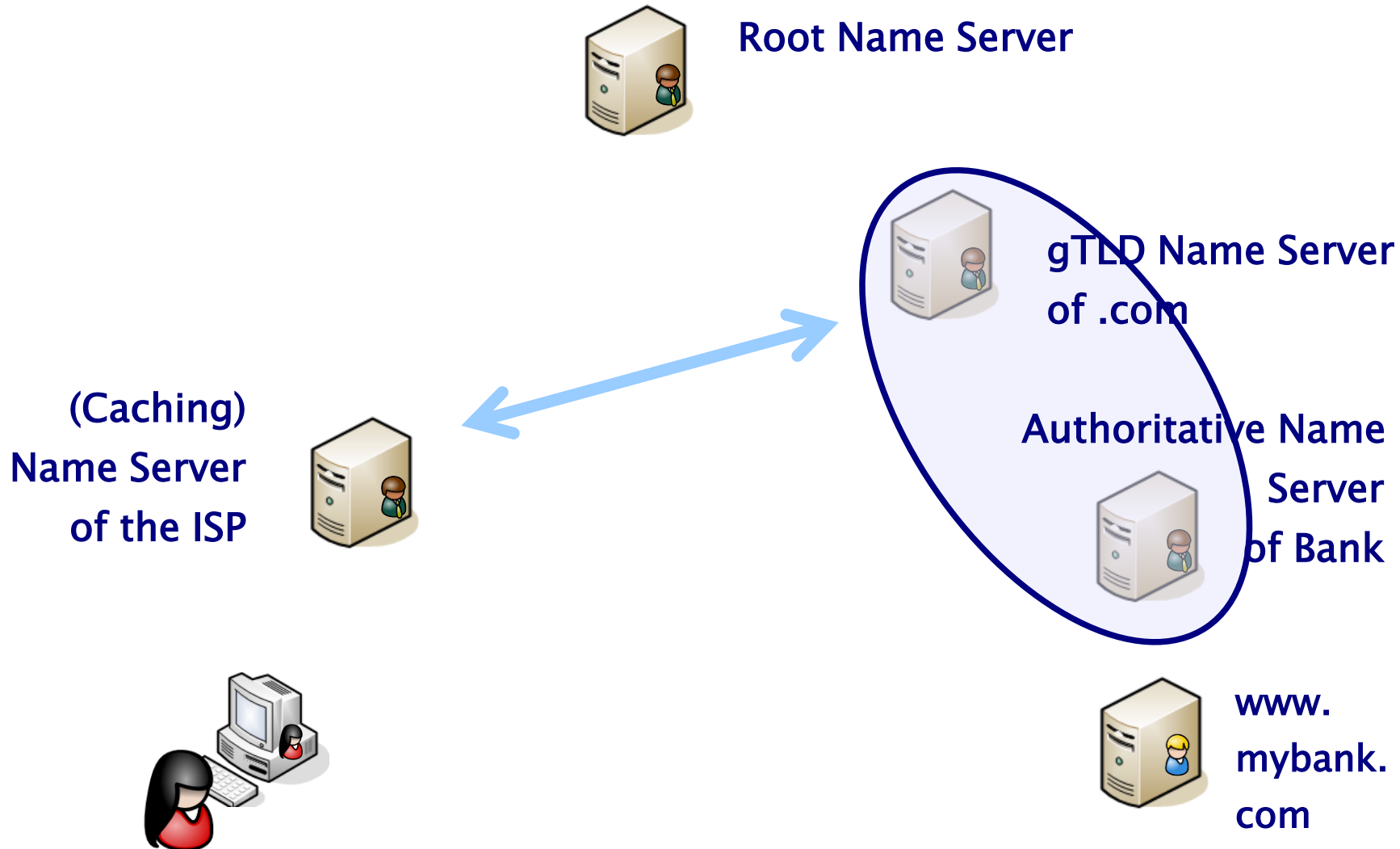


 [www.  
mybank.  
com](http://www.mybank.com)

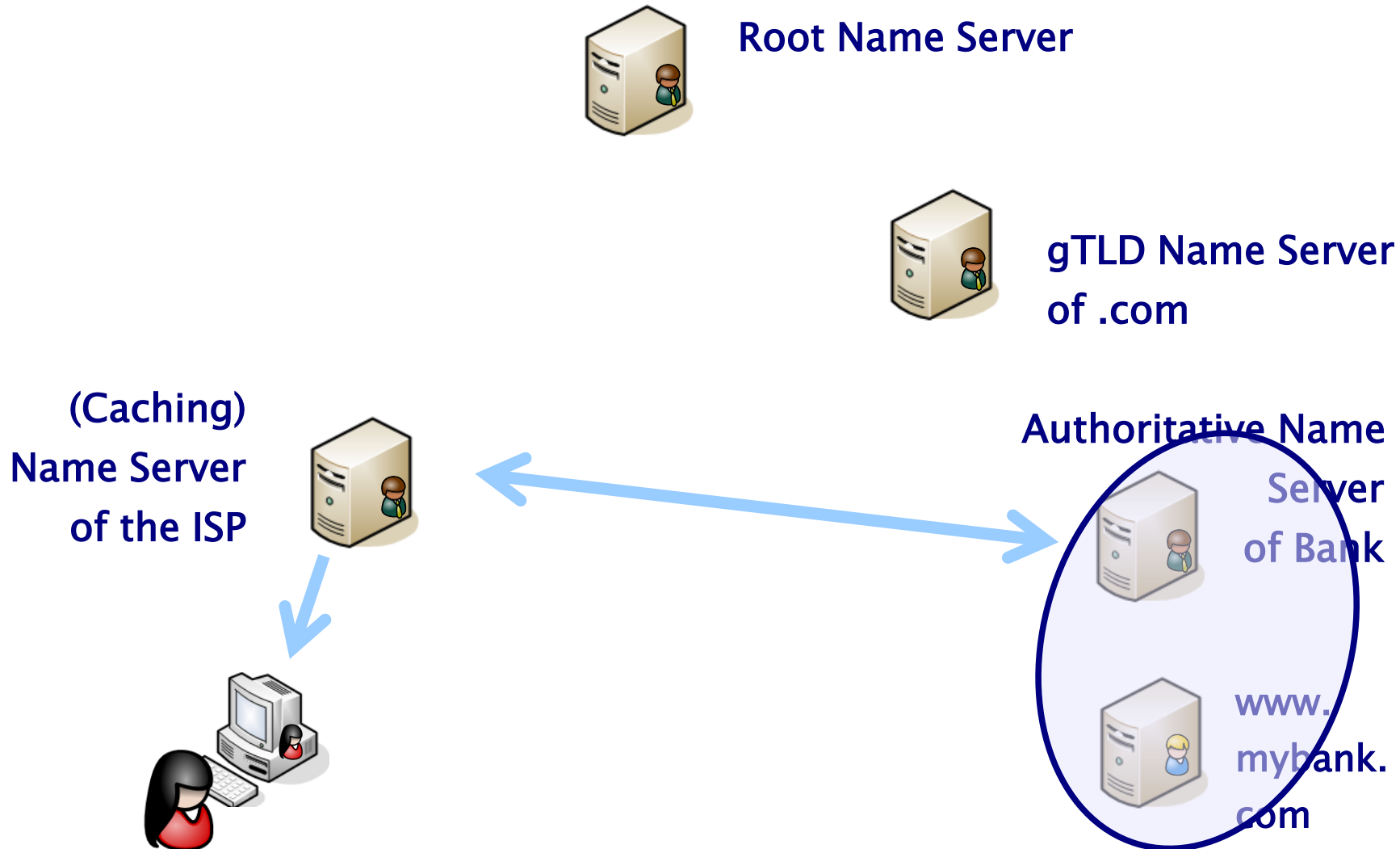
# Attack Vectors (2)



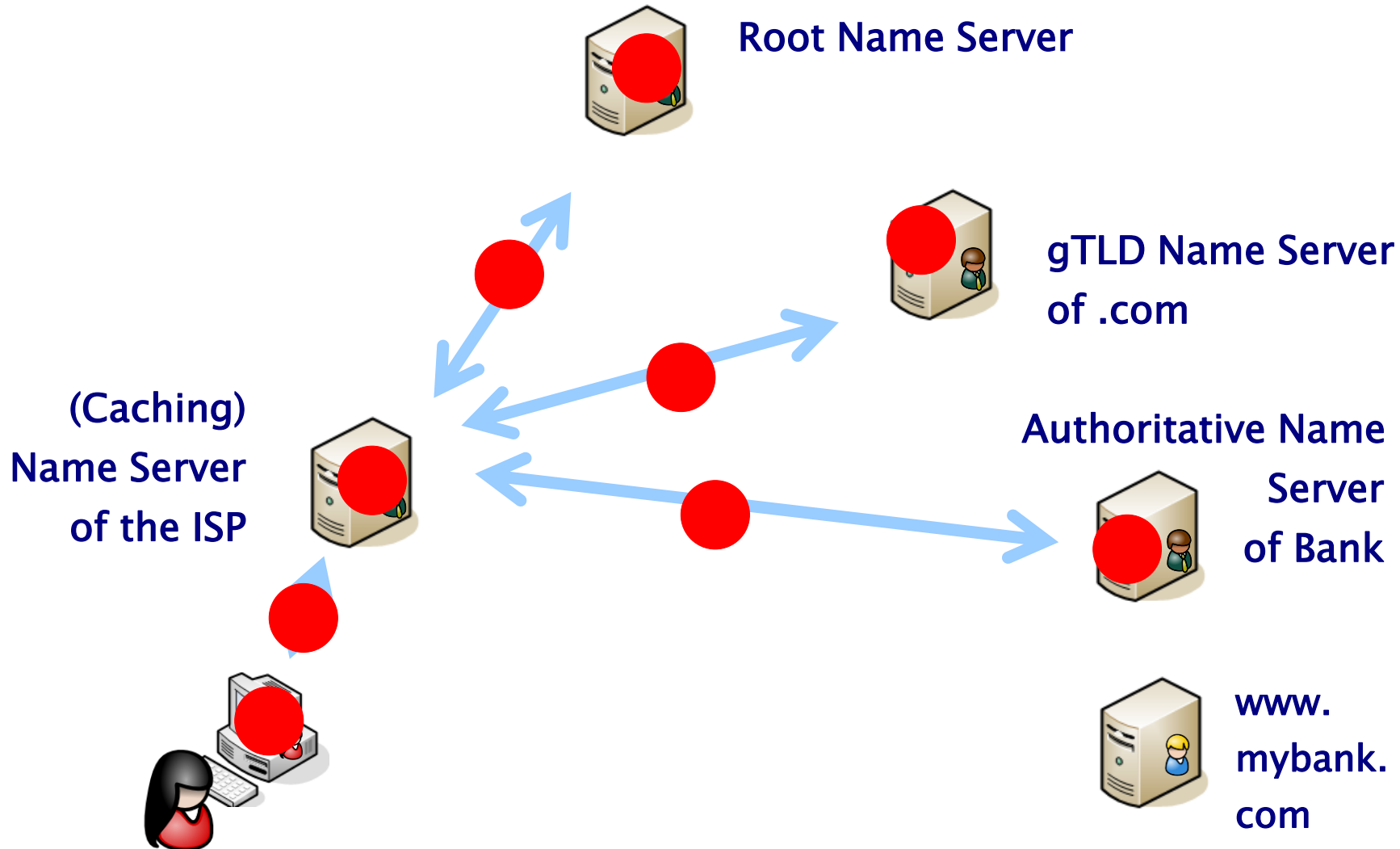
# Attack Vectors (3)



# Attack Vectors (4)



# Attack Surface of the DNS



# 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 on the Internet.




# DNSSEC Design Objectives

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

## Goals

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

## Non-Goals

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

# DNSSEC Fundamentals

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





# Signing Resource Record Sets

## o RRset:

```
- www.opendnssec.se. 7200 IN A 192.168.10.3
- A 10.0.0.3
- A 172.25.215.
```

## o DNSKEY RDATA:

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

Flags, Protocol, Algorithm

## o 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



# Delegation Signer (DS) Records

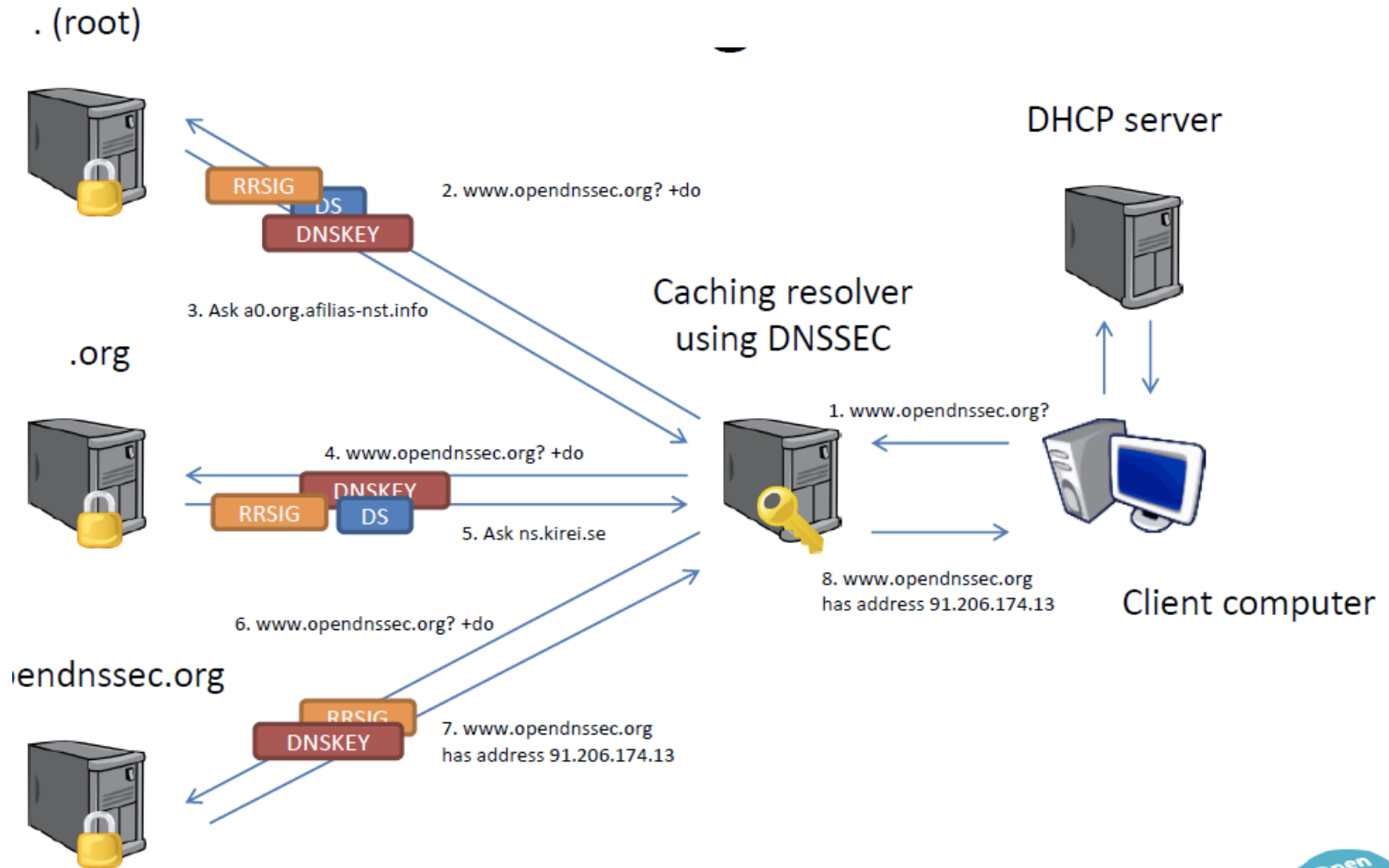
- o Handle for building the chain of trust along names
- o A DS record (the hash of the DNSKEY) is published at the parent zone to delegate trust to the child zone
- o Example (name, types, key-tag, algorithm, digest-type, digest):

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

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



# Resolving DNSSEC



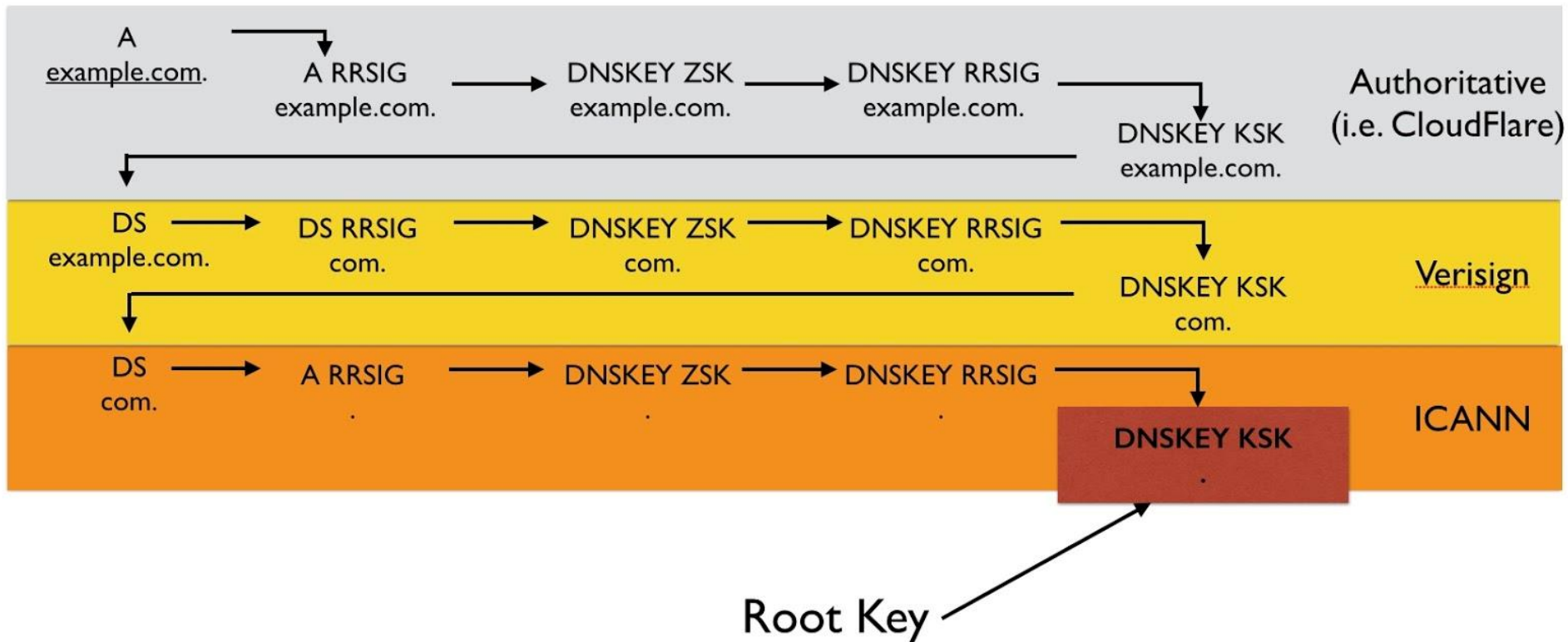
# DNSSEC Crypto

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



# DNSSEC Trust Chain

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



# Walking the Chain of Trust

Locally configured

Trusted key: . **8907**

\$ORIGIN .

1

2

3

```
. DNSKEY (...) 5TQ3s... (8907) ; KSK
DNSKEY (...) lasE5... (2983) ; ZSK
RRSIG DNSKEY (...) 8907 . 69Hw9...

net. DS 7834 3 1ab15...
RRSIG DS (...) . 2983
```

4

\$ORIGIN net.

5

```
net. DNSKEY (...) q3dEw... (7834) ; KSK
DNSKEY (...) 5TQ3s... (5612) ; ZSK
RRSIG DNSKEY (...) 7834 net. cMas...

foo.net. DS 4252 3 1ab15...
RRSIG DS (...) net. 5612
```

6

\$ORIGIN foo.net.

7

```
foo.net. DNSKEY (...) rwx002... (4252) ; KSK
DNSKEY (...) sovP42... (1111) ; ZSK
RRSIG DNSKEY (...) 4252 foo.net. 5t...

www.foo.net. A 193.0.0.202
RRSIG A (...) 1111 foo.net. a3...
```

8

9



# Chain of Trust Verification, Summary

- o Data in zone can be trusted if signed by a Zone-Signing-Key
- o Zone-Signing-Keys can be trusted if signed by a Key-Signing-Key
- o Key-Signing-Key can be trusted if pointed to by trusted DS record
- o 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)



# 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

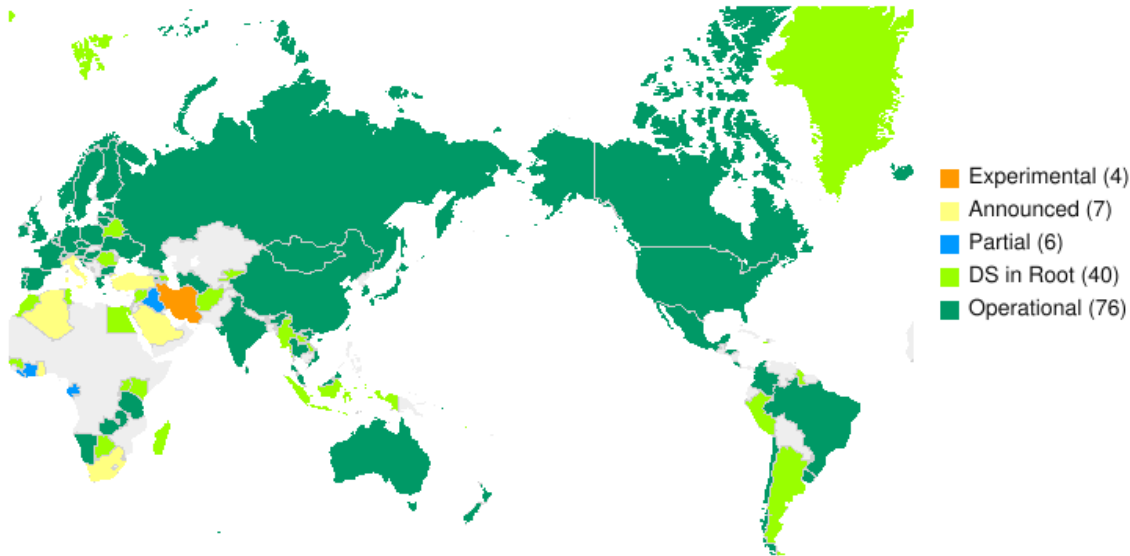
- o Provide proof of non-existence
- o NSEC points to the next label (domain name) in the zone, allows for zone walk (“get next”)
  - Zone walk often unwanted
- o NSEC3 prevents ‘walking in the clear’
  - Translates into hashes (linked list of hashed names)
  - Non-existence of hash proves non-existence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



# DNSSEC Deployment

## Country Top Level Domains

ccTLD DNSSEC Status on 2016-06-20



## 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/maps/>



# DNS over HTTPs (DoH)

- o Recent counter-approach – RFC 8484 (10/18)
  - Web-centric Over-the-Top service (OTT)
  - Full privacy on the net – queries TLS-encrypted
  - Rapid deployment by (a handful of) centralized servers: Google, Cloudflare, ...
  - Different trust model: Trust the (Central) Server provider
- o Problems from OTT
  - DNS records remain unauthenticated by RFC 8484
  - Centralized approach – no more distributed caching
  - Querier visible to DoH server *and* beyond: Modern DNS resolvers use EDNS subnet field (RFC7871)
  - Invisible to local providers – hinders debugging and performance optimization



## Securing IP Origins

# RPKI



# How can an Attacker Try to Hijack Your IP Prefix?

## You

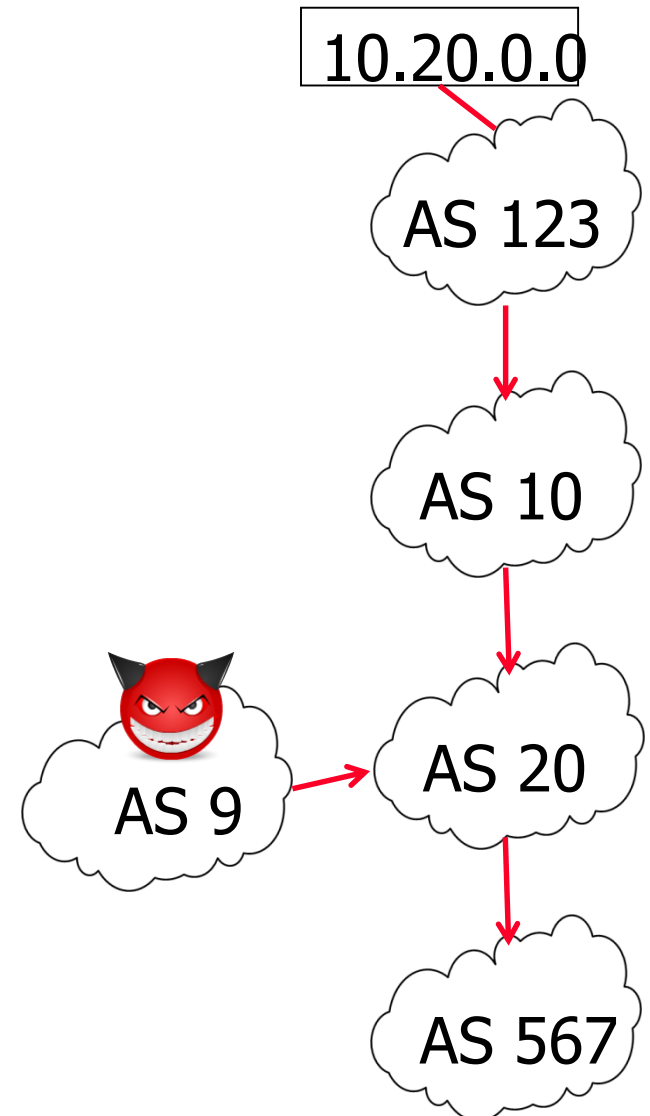
- o AS 123 announces IP prefix 10.20.0.0/16

## Me

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

## Attacker

- o Announces 10.20.0.0/16
- o Announces 10.20.30.0/24



# Hijacks in the Real World?

## Prominent examples

Firefox

Pakistan Blamed for Worldwide YouTube Break

thelede.blogs.nytimes.com/2008/02/25/pakistan-blamed-for-...  
Meistbesuchte Seiten http://www.potaroo.n... IP to ASN Lookup v1.0

**The New York Times**  
Saturday, October 29, 2011

WORLD U.S. N.Y. / REGION BUSINESS TECHNOLOGY SCIENCE HEALTH SPORTS OPINION

**Fontainebleau Fall for Bleu**

**The Lede**  
Blogging the News With Robert Mackey

February 25, 2008, 9:34 AM

**Pakistan Blamed for Worldwide YouTube Break**  
By MIKE NIZZA

If all had gone according to plan, Pakistan would have been the latest government taking part in an unsettling trend from Brazil to Thailand: YouTube blocking. Unlike its predecessors, though, Pakistan also affected thousands of people beyond its borders.

In case you were wondering on Sunday why you couldn't watch the video of the moment – President Nicolas Sarkozy telling a man to “get lost” – YouTube's answer was simple: Pakistan. Here is what the company had to say, via CNET:

“For about two hours, traffic to YouTube was routed according to erroneous Internet Protocols,” said YouTube spokesperson Ricardo Reyes in a statement. “Many users around the world could not access our site. We have

http://thelede.blogs.nytimes.com/

Firefox

Internetangriff: US-Kommission wirft China Datenentführung vor

http://www.spiegel.de/netzwelt/netzpolitik/0,1518,729690,00.html  
Meistbesuchte Seiten http://www.potaroo.n... IP to ASN Lookup v1.0 Englische Grammatik... Latest Headlines

**SPiegel ONLINE NETZWELT**

NACHRICHTEN VIDEO THEMEN FORUM ENGLISH DER SPIEGEL SPIEGEL TV ABO SHOP

Home Politik Wirtschaft Panorama Sport Kultur Netzwelt Wissenschaft einestages Karriere Uni Schule Reise Auto

Nachrichten > NetzWelt > Netzpolitik > Cyberwar Login | Registrieren

**THEMA Cyberwar**  
Alle Artikel und Hintergründe

17.11.2010 Drucken | Senden | Feedback | Meinungen

**Internetangriff**  
**US-Kommission wirft China Datenentführung vor**  
Von Konrad Lischka

VERWANDTE THEMEN  
Computersicherheit  
Volksrepublik China

ALLE THEMENSEITEN >>>

LÄNDERLEXIKON CHINA >>>

Fläche: 9.572.900 km<sup>2</sup>  
Bevölkerung: 1.341,335 Mio. Einwohner  
Hauptstadt: Peking  
Staatsoberhaupt: Hu Jintao  
Regierungschef: Wen Jiabao

Mehr auf der Themenseite | Wikipedia | Lexikon | China-Reisesite

**Nächsten Sie ein anderes Land erkunden?**  
Alle Länder im Überblick >>>

Netzwerkabel: Letztere China Daten aus den USA über eigene Server um?

Google free public DNS ser...

www.zdnet.com/google-free-public-dns-services-were-briefly-corrupted-7...  
Most Visited FPQ-Test Event Time Announce... TimeZone http://www.potaroo.n... Studie: Tausende eing...

**ZDNet**

UK Edition UK News IT Budgeting IT Jobs 3D Printing Hybrid Cloud Wearables

FIRST TAKE: Lenovo Yoga 3 Pro: Thin, light and flexible

Topic: Networking Follow via: RSS

**Google free public DNS services were briefly corrupted**

Summary: For some South American users, Google's free public Domain Name System servers were corrupted for less than a half hour.

By Steven J. Vaughan-Nichols for Networking | March 17, 2014 - 21:44 GMT  
Follow @sjvn  
Get the ZDNet Most Read News Alerts - UK newsletter now

Without the Domain Name System (DNS), we're all lost on the Internet. DNS provides the service that translates our human readable Web addresses such as google.com to their real, but mysterious Internet Protocol version 4 (IPv4) addresses, such as 8.8.8.8 or IPv6's 2001:4860:4860::8888. The problem with this master yellow pages directory to the Internet is that DNS records themselves can be corrupted or your communications with the DNS servers interrupted by a man-in-the-middle (MM) attack.

**Google DNS 8.8.8.8/32 was hijacked for ~22min yesterday, affecting networks in Brazil & Venezuela #bgp #hijack #dns pic.twitter.com/wBuui8dwO**

BGPmon.net  
BGPmon.net  
Follow

**BGPmon.net**  
Google DNS 8.8.8.8/32 was hijacked for ~22min yesterday, affecting networks in Brazil & Venezuela #bgp #hijack #dns pic.twitter.com/wBuui8dwO  
Reply Retweet Favorite + More  
BGPMON

Read psm-cdh.com

**Related Stories**

- Australians ease off broadband above 24Mbps: ABS
- Google's 'Physical Web' project looks to inject Internet of Things with more search
- Broadband Britain: Where we've come from, where we are going
- Akamai shows global and US internet speeds increasing

**The best of ZDNet, delivered**

**ZDNet Newsletters**  
Get the best of ZDNet delivered straight to your inbox

Enter your email address

ZDNet Most Read News Alerts - UK  
Major news is breaking. Are you ready? This newsletter has only the most important tech news nothing else.

Subscribe Now

f t in x+ s

**Resource Centre**

**Caveat: Reasons may also be misconfiguration ;-)**

# Outline

## Part 1 - Background:

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

⇒ All of this in a cryptographically strong way

## Part 2 – Deployment, Implementation RTRlib:

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



# Problem

- o BGP is based on trust between peers

## Implications

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

## Compromise

- o Filtering
  - o 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

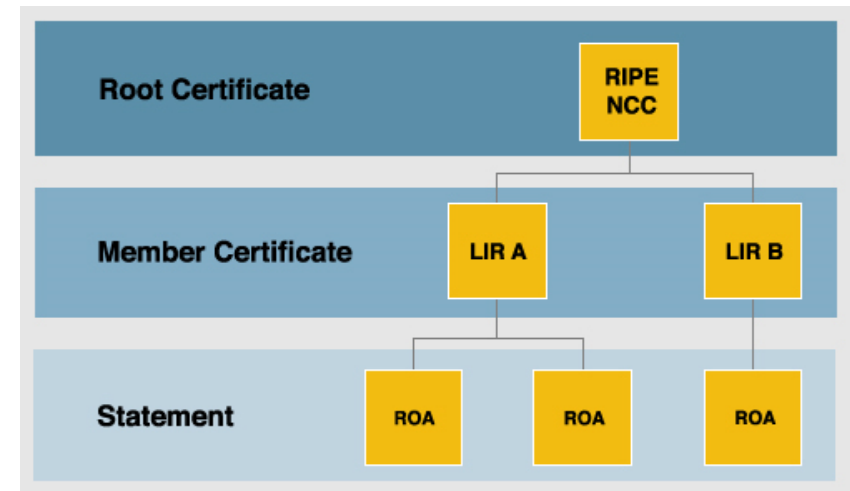
- o Cryptographic operations are complex
- o Minimal additional load at routers

In the following we concentrate on 1.



# Resource Public Key Infrastructure (RPKI)

- o System that allows to attest the usage of IP addresses and ASNs (i.e., Internet resources)
- o RPKI includes cryptographically provable certificates
- o Certificate hierarchy reflects IP-/AS-allocation in the Internet
  - o Currently, each RIR creates a self-signed root certificate



Source: RIPE

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



# Routing Origination Authorization (ROA)

- o 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
- o ROA will be signed with the certificate of the RPKI
- o Note: Multiple ROAs per IP prefix possible

Example:

ROA

```
Valid from
01/10/2012    10.20.0.0/16-24 -> AS 123
to
01/10/2013    80.90.0.0/16-16 -> AS 123
+ E2E Cert
```

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 1<sup>st</sup> Oct. 2012 until 1<sup>st</sup> Oct. 2013



# RPKI & ROA

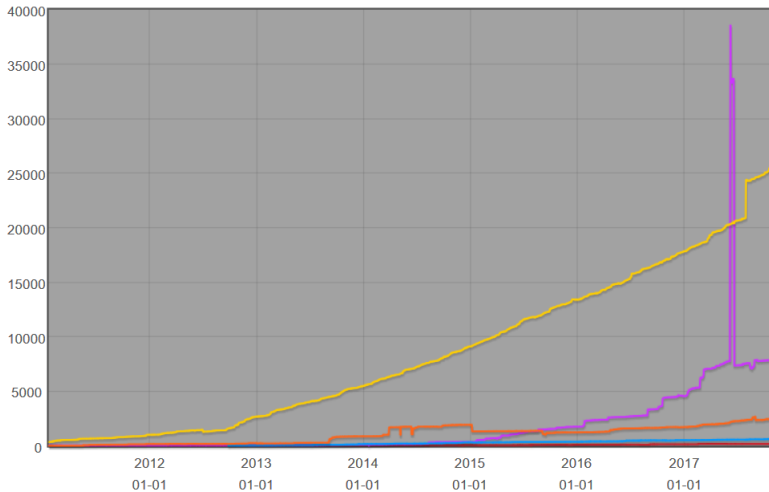
- o All certificates including ROAs will be published in RPKI repositories
  - Each RIR (including RIPE NCC ;) operates one
  - ISPs can maintain their own repository
  - Information of all repositories describe the overall picture
- o Check if AS is allowed to announce IP prefix  
= check the corresponding 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 prefixes in ROAs  AfrINIC  APNIC  ARIN  LACNIC  RIPE NCC

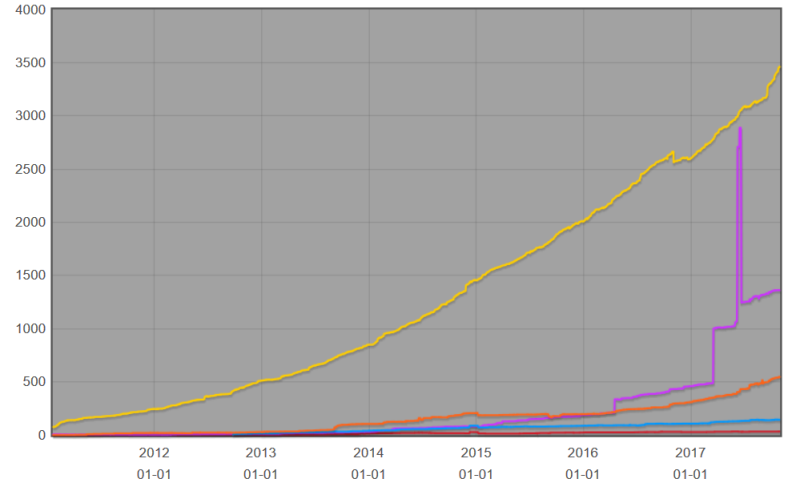
This graph shows the total amount of distinct IPv4 prefixes found in the ROAs



IPv4

IPv6 prefixes in ROAs  AfrINIC  APNIC  ARIN  LACNIC  RIPE NCC

This graph shows the total amount of distinct IPv6 prefixes found in the ROAs



IPv6

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

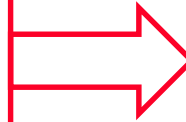


# Prefix Origin Verification & RPKI

Validation process consists of two steps

## 1. Validation of ROAs

- Performed at external cache



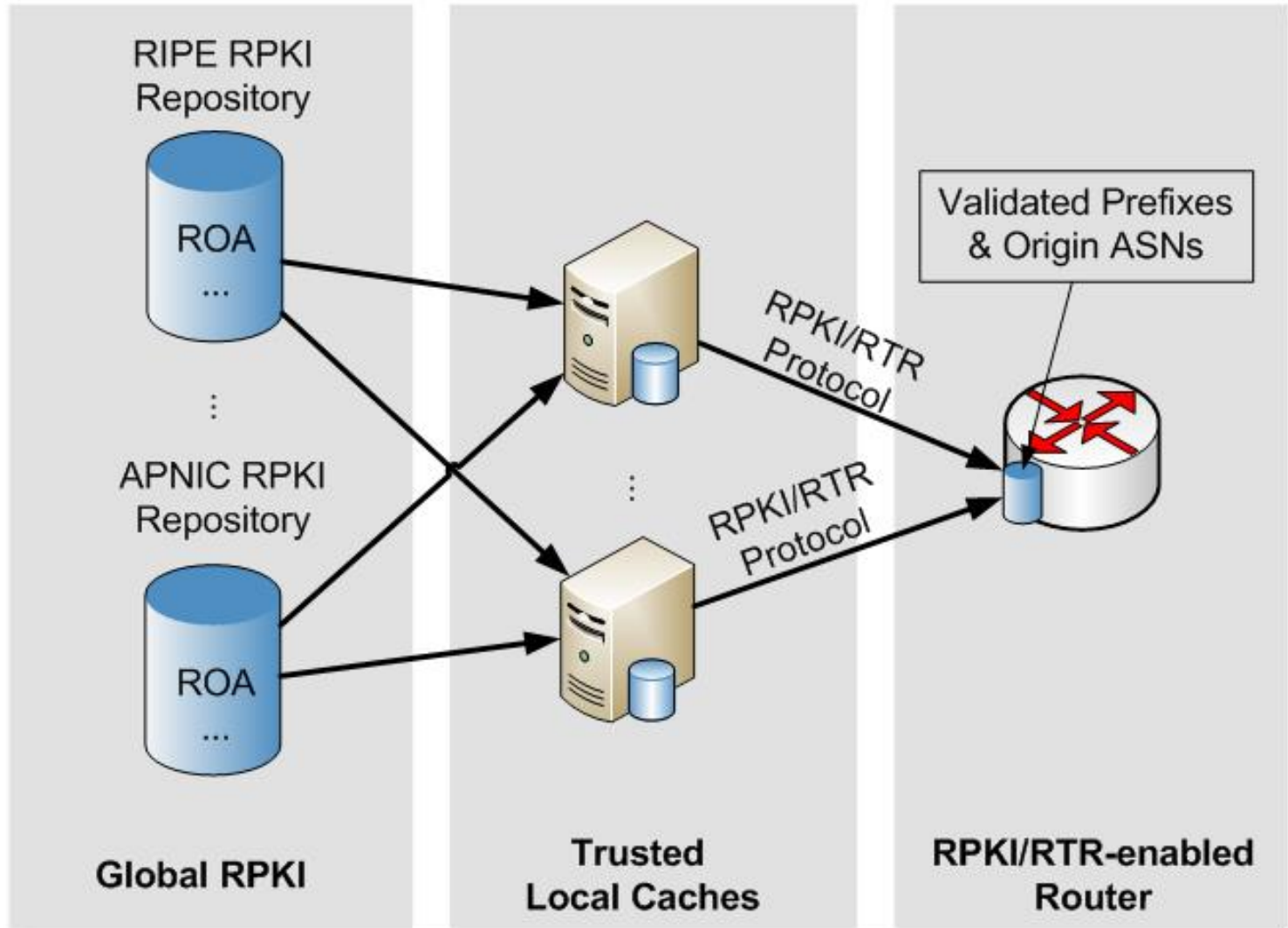
## 2. Validation of BGP updates

- Performed at BGP router
- No additional cryptographic operations necessary

- o 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
- o 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

- o 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

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

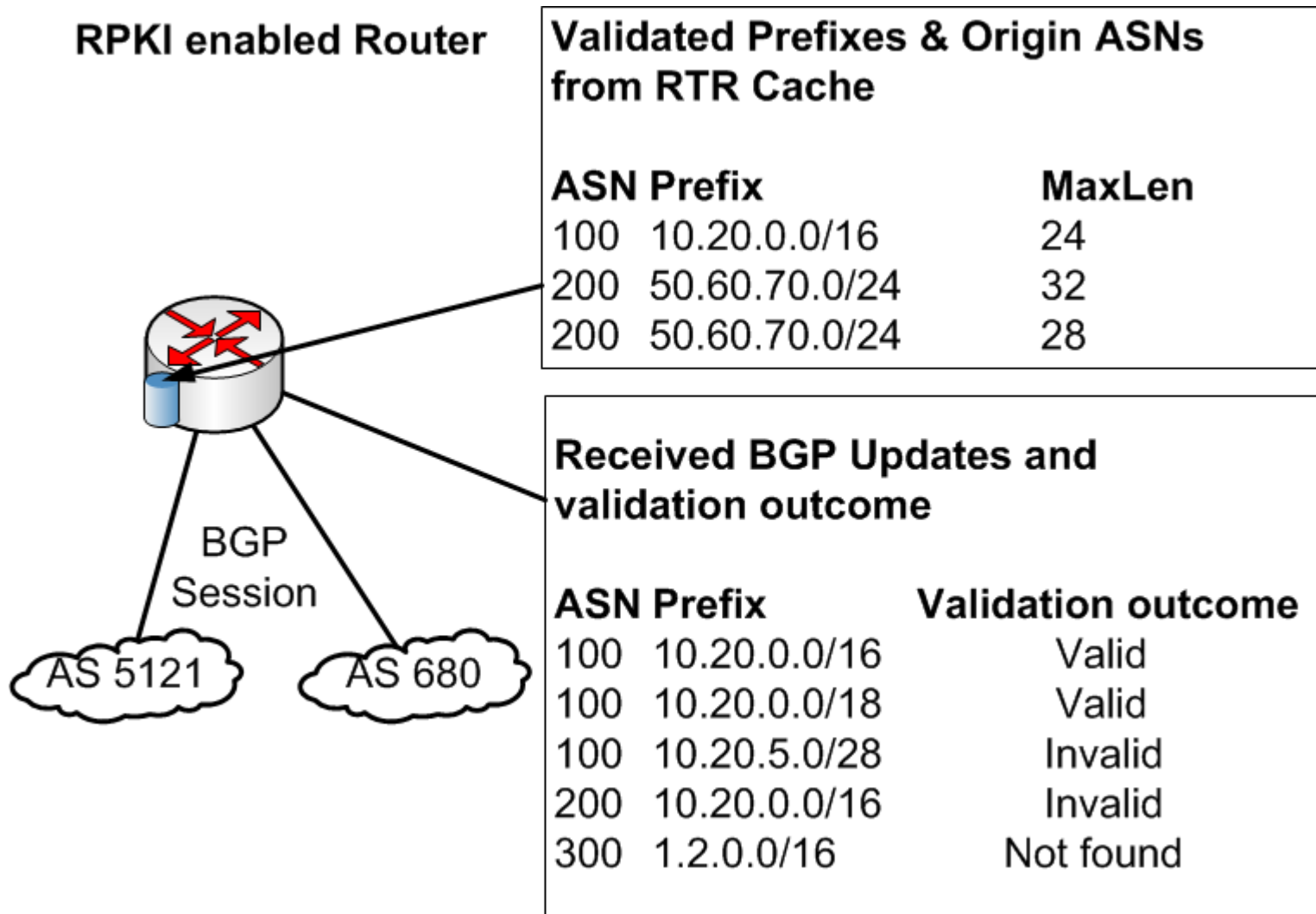
## Not Found

- o If no covering ROA exists





# Validation Outcome - Examples



# How to Participate?

## Create ROAs

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

## Setup RTR Cache Server (or use an external)

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

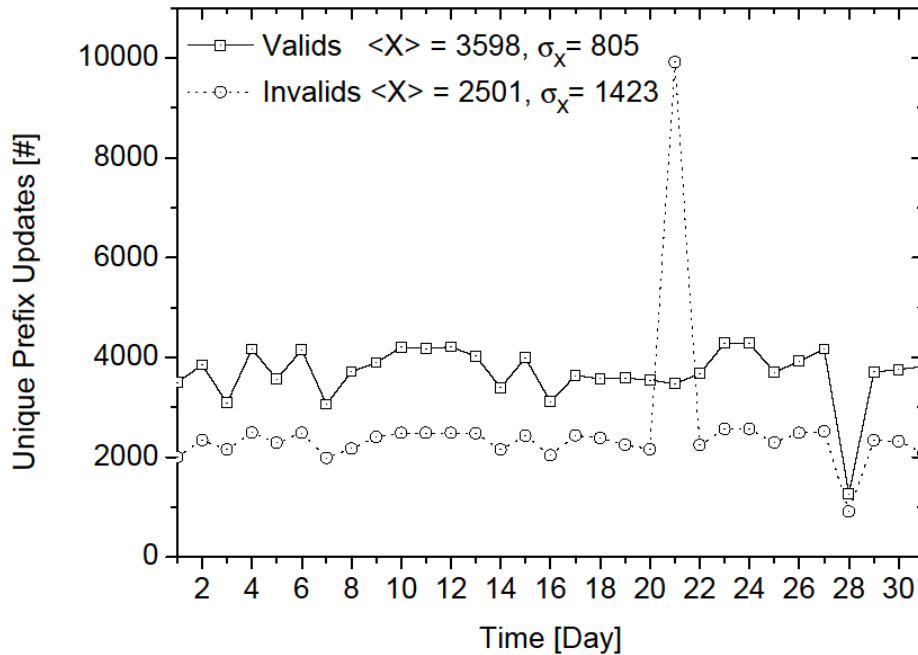
## Enable your BGP Router

- o Juniper and Cisco have official support for RPKI since years.
- o Open-Source: FRR, (Quagga), BIRD by RTRlib



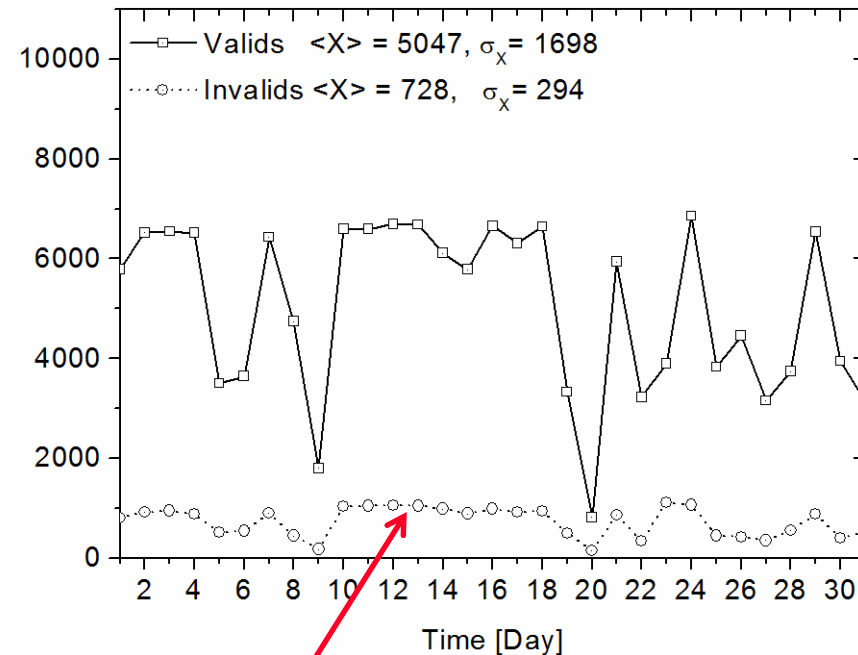
# Valide vs. Invalide BGP Updates

Januar 2012



Number of invalids  
decreases over time

Mai 2012



Are these updates  
really hijacks??



# Some Common Pitfalls - Examples

## Case 1: Missing Customer (or Sibling) Legitimation

- o ROA created: 12.0.0.0/8-9 -> AS 7018
  - o AS 27487 announces 12.0.19.0/24
  - o AS 2386 announces 12.1.216.0/24
- ⇒ Consider sub-allocations, start most specific
- Both announcements are invalid if no ROAs exists

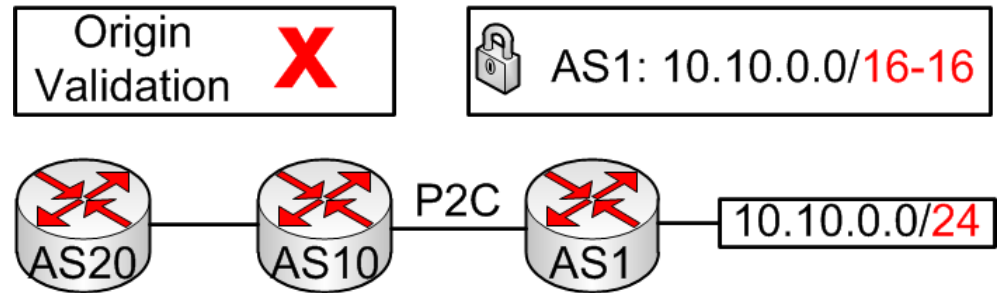
## Case 2: (De-)Aggregation

- o ROA created: 78.192.0.0/10-10 -> AS 12322
  - o Usual announcement: 78.192.0.0/10
  - o For 30 minutes: 78.192.10.0/24 ...
- ⇒ Configure the max ROA prefix length explicitly

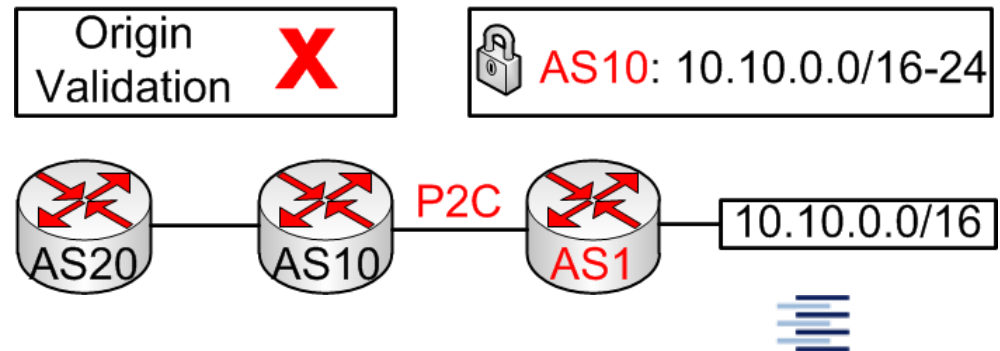


# Common Pitfalls – Overview (1)

Valid origin, announced prefix is more specific

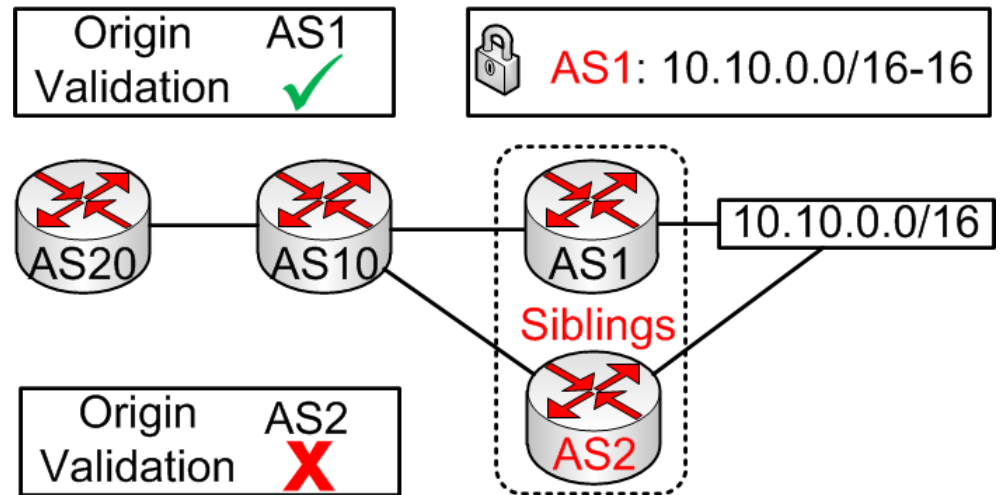


Provider does not consider customers



# Common Pitfalls – Overview (2)

Additional AS of a company is not authorized



# Implementing the RPKI Router Part

## **RTRLIB**



# What is the RTRlib?

## General objective

- o Implementation of the RPKI-RTR client protocol in C

## Details

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

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





# Applications

## o Extension of BGP daemons

- Now part of FRR, (Quagga), BIRD (code-wise), and commercial products

## o Monitoring of the RPKI deployment

- Integrate the library in your Python/Perl ... scripts
- Particularly suitable for real-time monitoring

## o Testing purposes

- Evaluate performance of your RPKI/RTR cache server
- Play around with BGP update validation

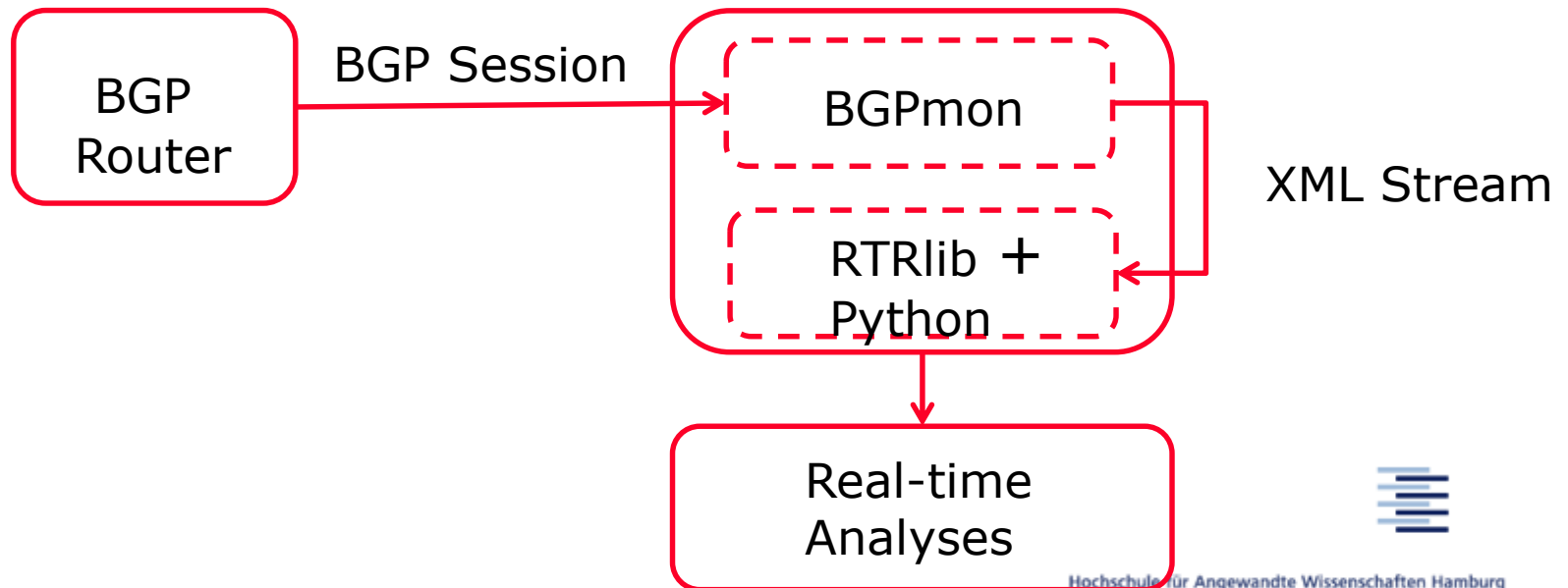


# Using RTRlib for Monitoring

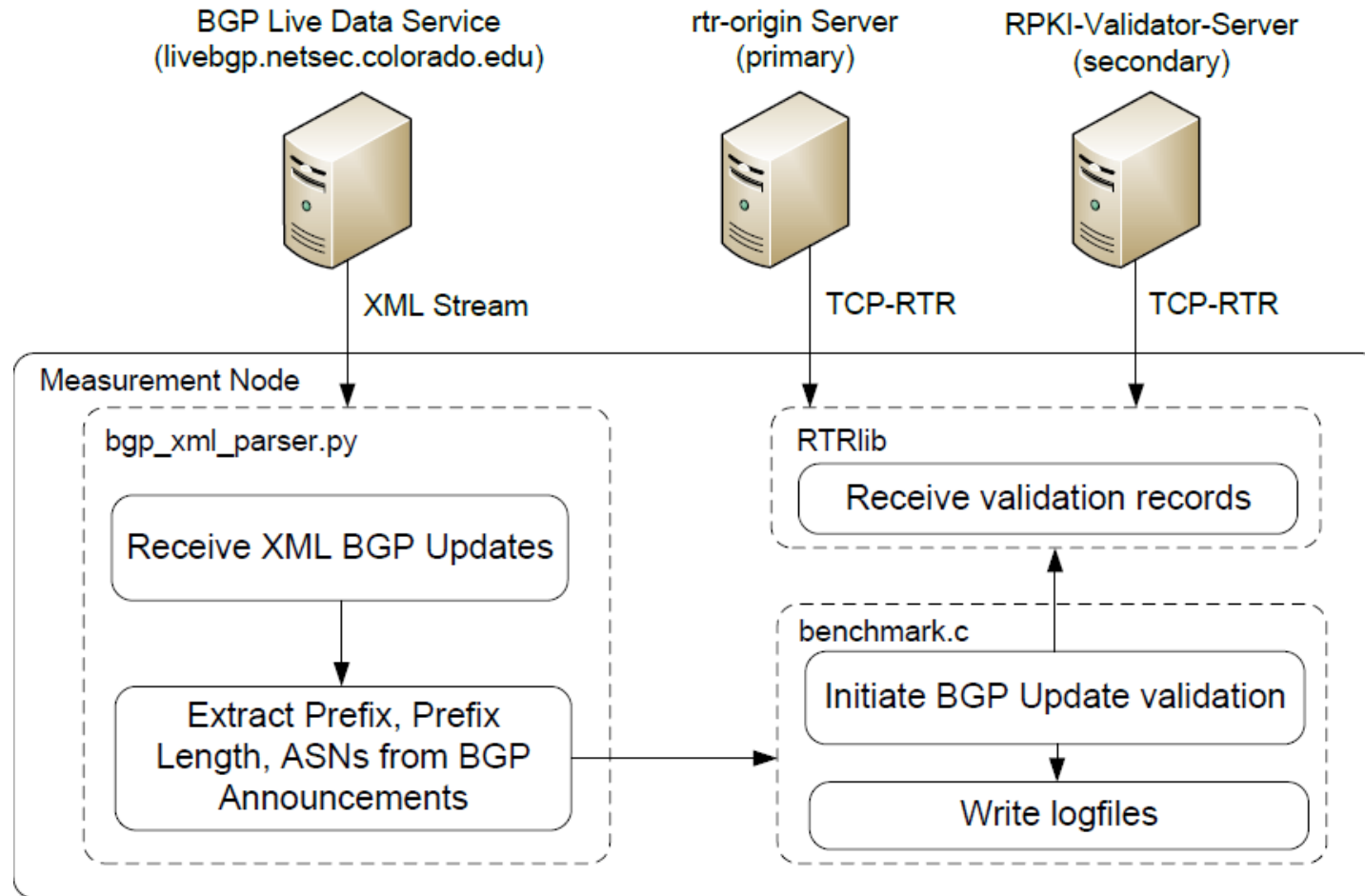
**Objective:** Emulate update validation of your BGP peer

**Setup – No Firmware Change at Your Router:**

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



# Example: Monitoring Scenario



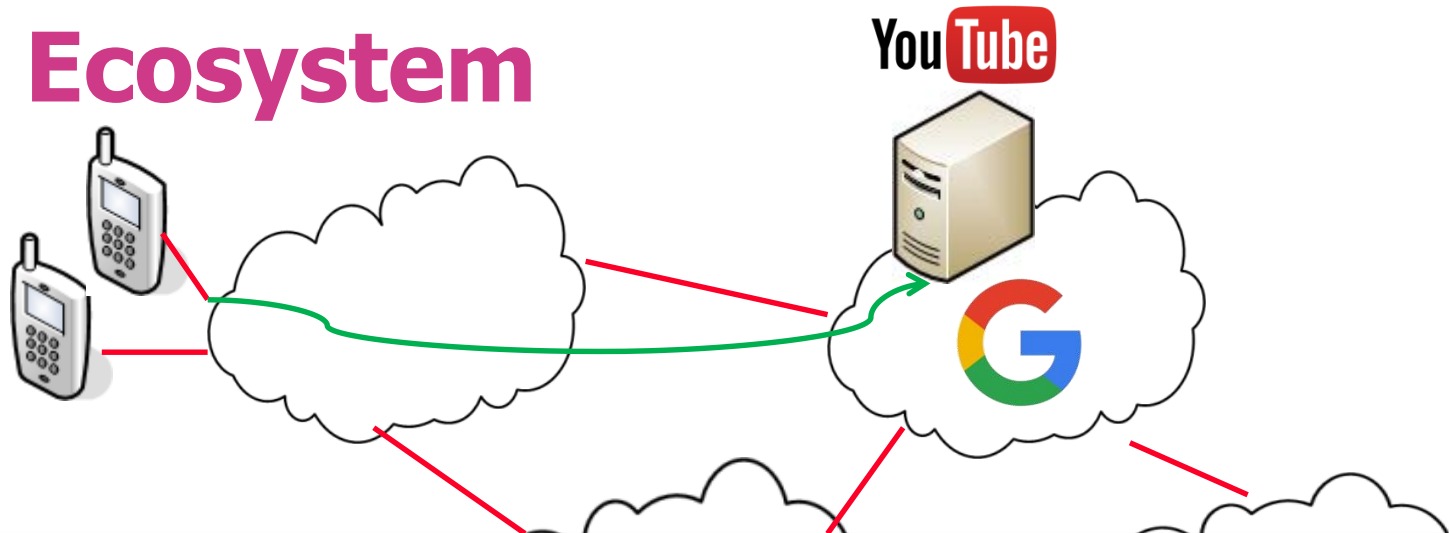
# Objectives [ACM HotNets'15]

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

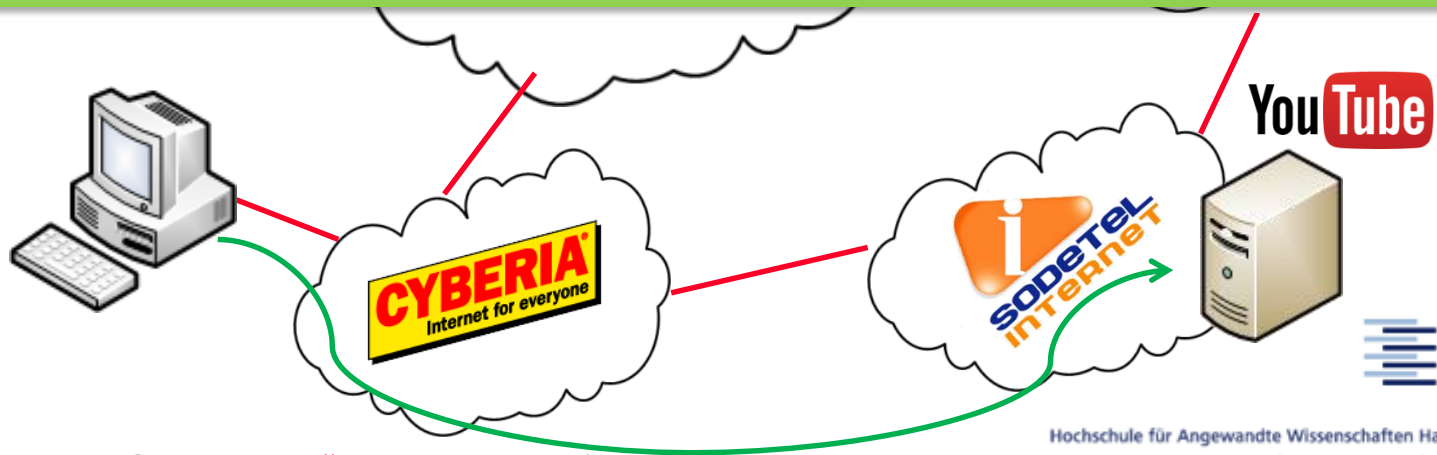
Which web servers are secured by the RPKI?



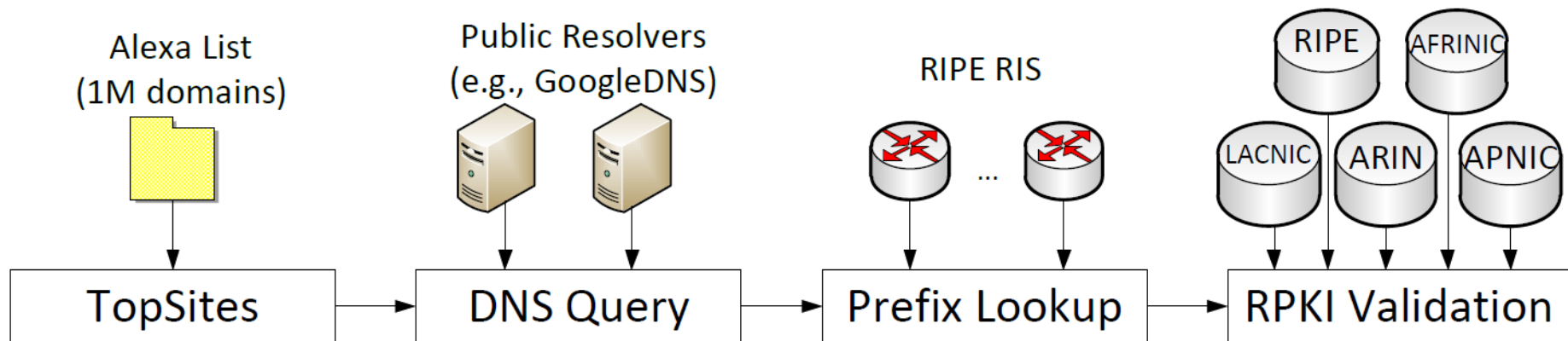
# Web Ecosystem



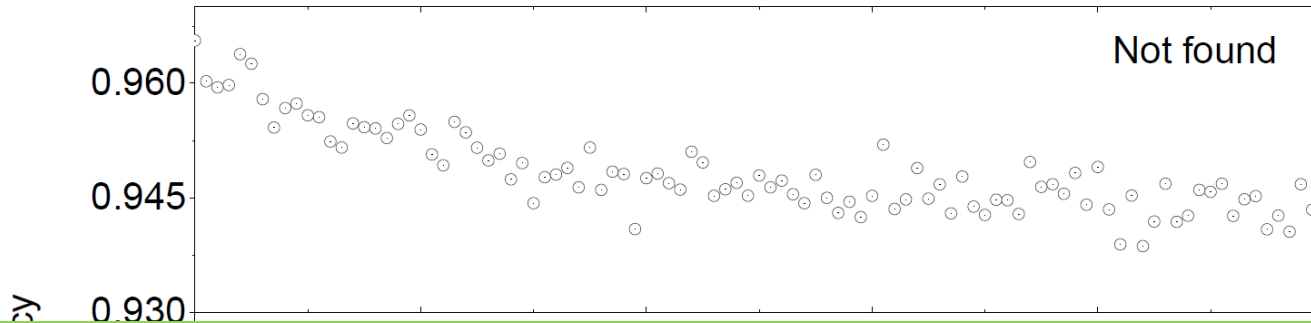
CDNs make web access faster.  
But measurements and security more challenging



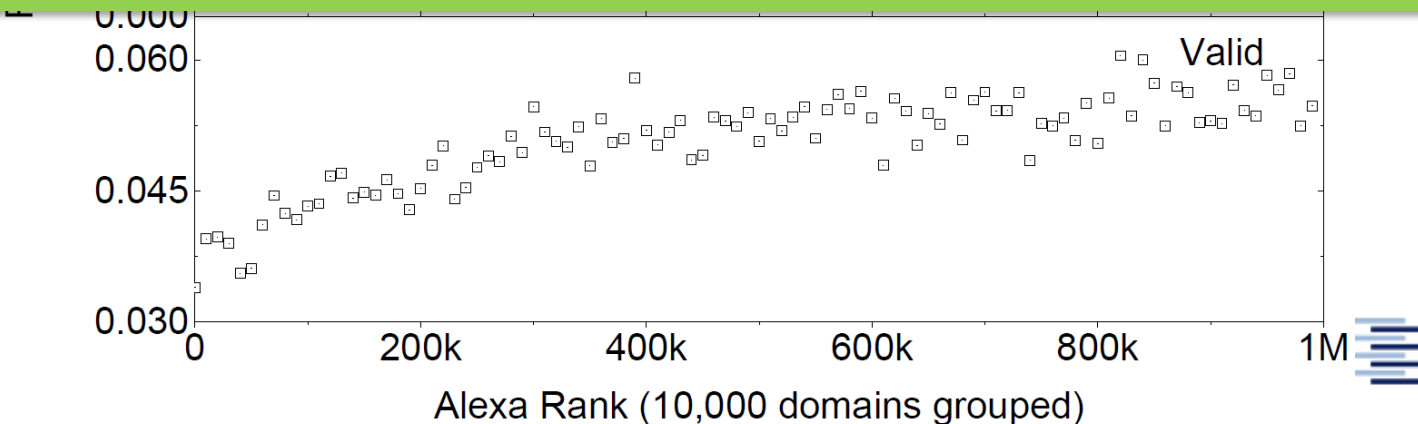
# Measurement Methodology



# RPKI Validation Outcome for 1M Web Sites



**More popular sides are less secured!**



# Validation in Web-Browser



ASN:	6724
AS Name:	STRATO STRATO AG
IP Address:	81.169.145.33
BGP Prefix:	81.169.144.0/22
Validation Result:	Valid



# Study: ROA and ROV [SIGCOMM CCR '18]

Route Origin  
Authorization (ROA)

Prefix owner authorizes AS to  
originate a set of prefixes

Route Origin  
Validation (ROV)

BGP router validates received  
routes using ROA information

# Motivation & Research Problem

Goal: Which ASes use ROV-based filtering policies?

Assess impact of defense mechanisms

Track deployment over time

Create an incentive to deploy

Challenge: Private router configurations must be inferred

# Controlled Experiments: Setup

## Hand-crafted ROAs *and* BGP Updates

Goal: Find ASes that filter invalid routes

### BGP

Announce prefixes  $P_A$  (Anchor)  
and  $P_E$  (Experiment)

- ✓ Same RIR DB route object
- ✓ Same prefix length
- ✓ Announced at the same time
- ✓ Announced to same peers
- ✓ Announced from same origin AS

### RPKI

Issue ROAs for  
both prefixes

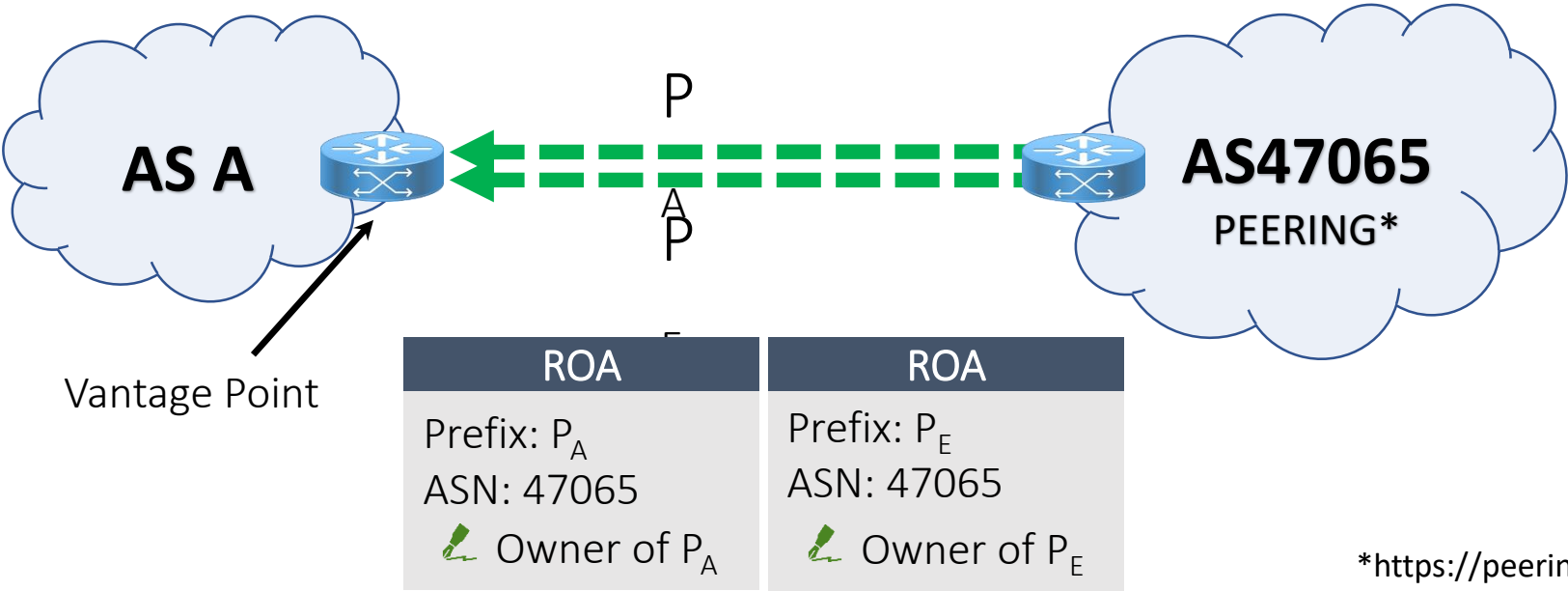
$P_A$  announcement is always *valid*.

Periodically change ROA for  $P_E$  :

- Flips announcement from *valid* to *invalid* to *valid* daily.

# Controlled Experiments

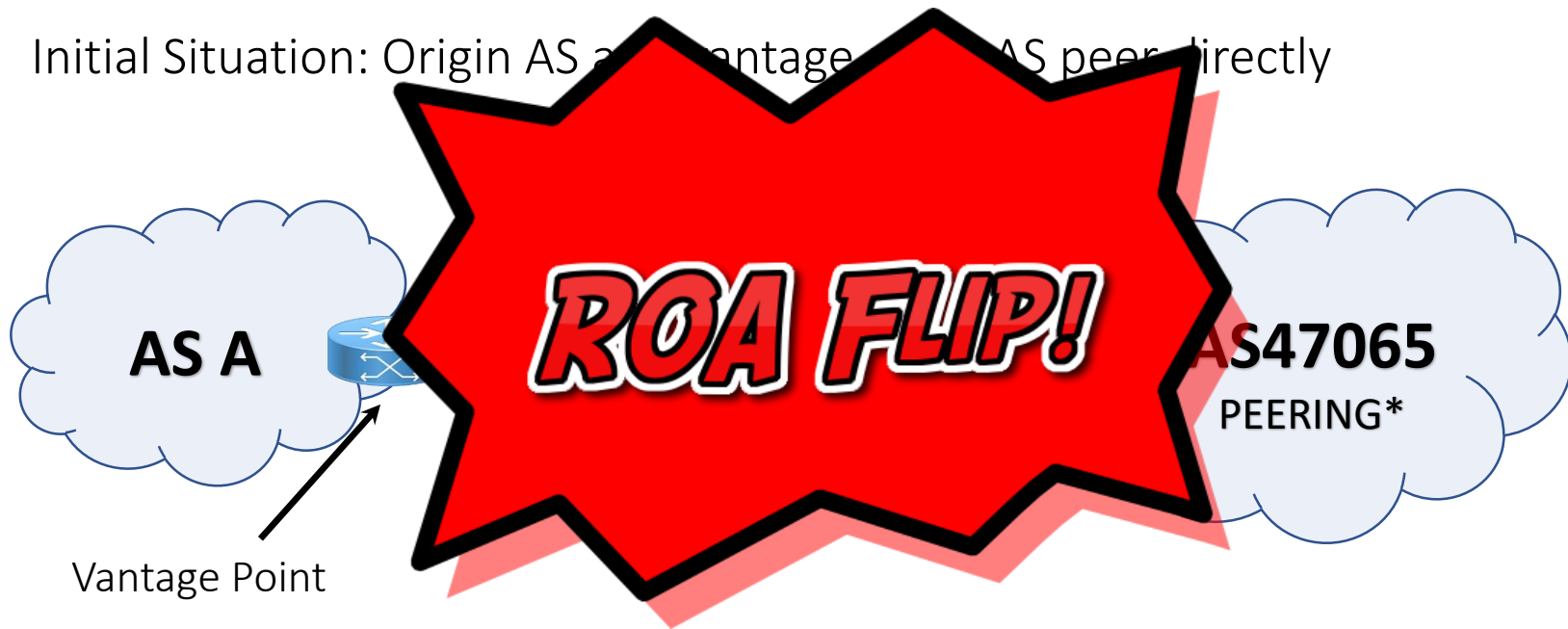
Initial Situation: Origin AS and vantage point AS peer directly



\*<https://peering.usc.edu/>

# Controlled Experiments

Initial Situation: Origin AS a vantage AS peer directly





\*<https://peering.usc.edu/>

# Controlled Experiments

Observation: Vantage point exports no route for  $P_E$



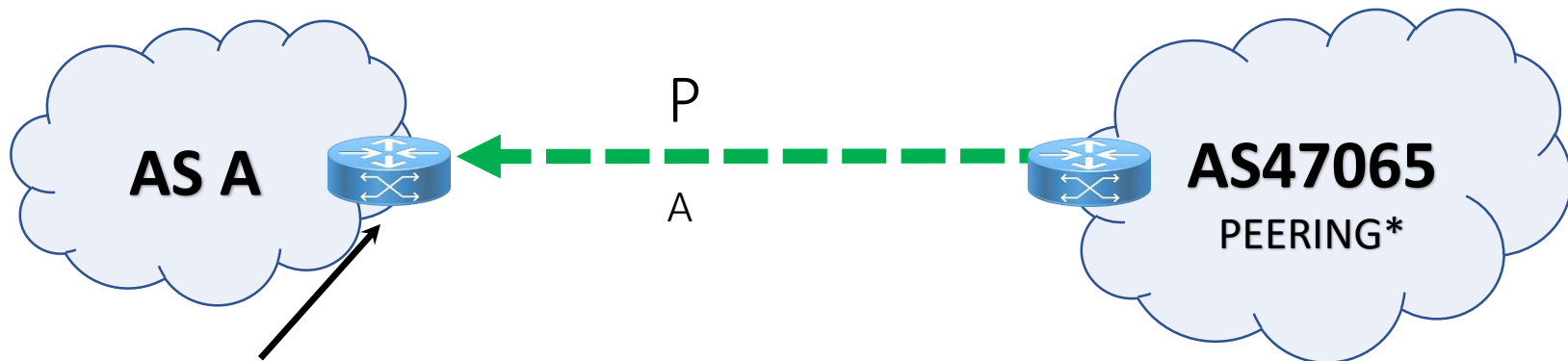
Vantage Point

ROA	ROA
Prefix: $P_A$	Prefix: $P_E$
ASN: 47065	ASN: 51224
 Owner of $P_A$	 Owner of $P_E$

\*<https://peering.usc.edu/>

# Controlled Experiments

Observation 1: Vantage point exports no route for  $P_E$



Vantage Point

Conclusion: Vantage point is using ROV-based filtering

\*<https://peering.usc.edu/>

# Controlled Experiments Results

Before October 20<sup>th</sup> 2017:

- (At least) Three ASes drop invalid routes

October 20<sup>th</sup> 2017:

- AMS-IX Route Server changes ROV based filtering to 'opt-out'
- 50+ ASes "drop" invalid routes

Full talk on [Youtube](#)



# Software Download

- o RTRlib – The RPKI RTR Client C Library  
<http://rpki.realmv6.org/>
- o RIPE NCC RPKI Validator (Cache Server)  
<http://www.ripe.net/lir-services/resource-management/certification/tools-and-resources>
- o rpki.net project  
<https://trac.rpki.net/>
- o RPSTIR – BBN RPKI Validator  
<http://sourceforge.net/projects/rpstir/>
- o Firefox Add-on RPKI Validator  
<https://addons.mozilla.org/addon/rpki-validator/>



# Further Reading

## Overview Article

- o 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](http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_12-1/121_resource.html)

## RPKI deployment

- o 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.

## RoV deployment

- o Andreas Reuter, Randy Bush, Italo Cunha, Ethan Katz-Bassett, Thomas C. Schmidt, Matthias Wählisch: *Towards a Rigorous Methodology for Measuring Adoption of RPKI Route Validation and Filtering*, *ACM Sigcomm Computer Communication Review*, Vol. **48**, No. 1, p. 19–27, January 2018.



# Further Reading (cont'd)

## IETF – Standardization

- o Secure Inter-Domain Routing Group (SIDROPS)  
<http://datatracker.ietf.org/wg/sidrops/>
- o Draft “BGP Prefix Origin Validation”  
<http://tools.ietf.org/html/rfc6810>
- o Draft “The RPKI/Router Protocol”  
<http://tools.ietf.org/html/rfc6811>

## RPKI/RTR Implementation Report

- o <http://tools.ietf.org/html/rfc7128>

## RTRlib Reference Report

- o 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

- o Application transport today provides encryption, integrity protection, privacy, +++
  - Examples are TLS, DTLS, IPSec, S/MIME, SSH, ...
- o 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

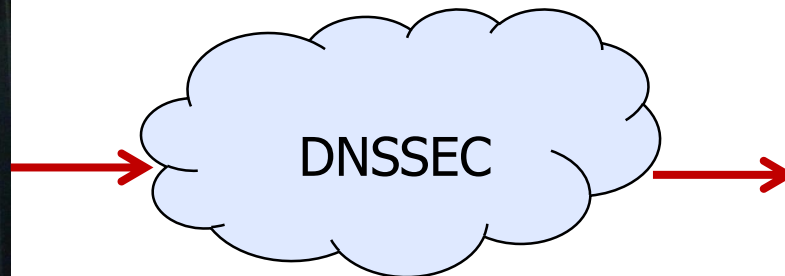


# Threats & Flaws of the CA Approach

- o 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)
- o Tolerance & delegation may lead to unexpected endpoints
  - Often self-signed or expired certificates
  - CDNs officially terminate TLS sessions
- o We learn CA keys out of band
  - Local misuse by configuration („TLS-proxies“)
- o Key revocation problem
  - Revocation lists slow, not scalable
  - After compromise, everybody wants to revoke →Heartbleed!

# DNS Based Authentication by Named Entities (DANE)

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



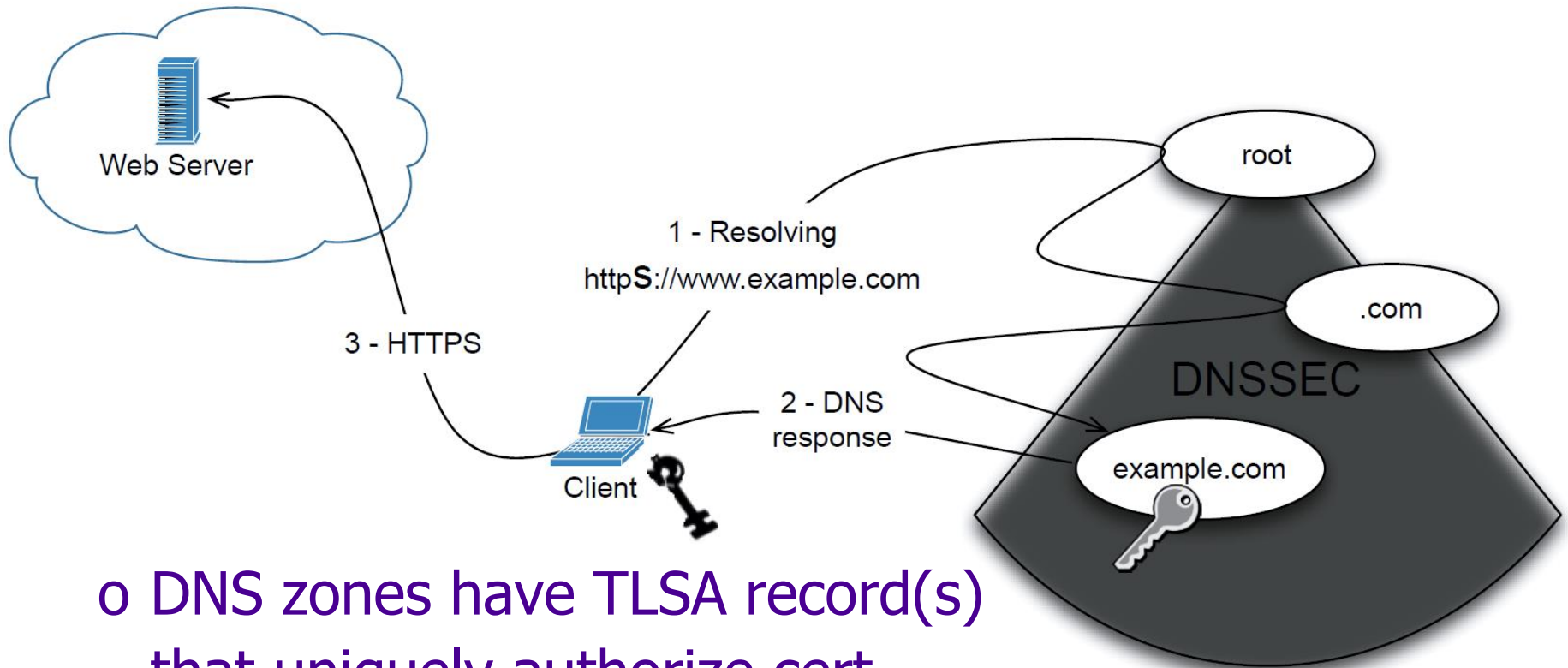
# TLSA Records in DNSSEC

- o DNS record type to authenticate remote endpoints in transport: SSL/TLS (web, mail, ...)
- o TLSA key: `_port._proto.domain.tld` –  
`_443._tcp.good.dane.verisignlabs.com`
- o TLSA value: Meta-data + Certificate Association Data (raw cert data in hex) –  
(0 0 1 d2abde240d7cd3ee6b4b28c54df034b9  
7983a1d16e8a410e4561cb106618e971)





# DANE verification process



- o 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:**

- o Server-centric security toolset – mainly inter-SMTP mail security
- o Emerging building blocks for 'Secure Email' with clients (↪ Thunderbird)
- o 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

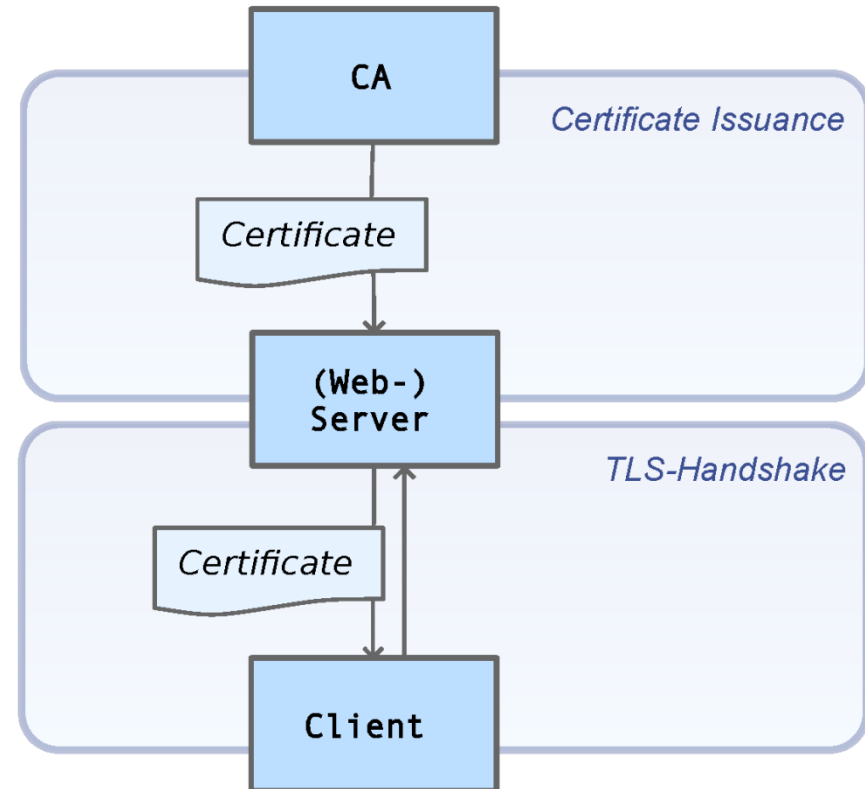
- Certify any resource
- Remain in secrecy
- Lie about time of issuing

A Client alone cannot

- Verify correctness of the CA

Public trust anchors can help

- DANE per name
- CT per certificate

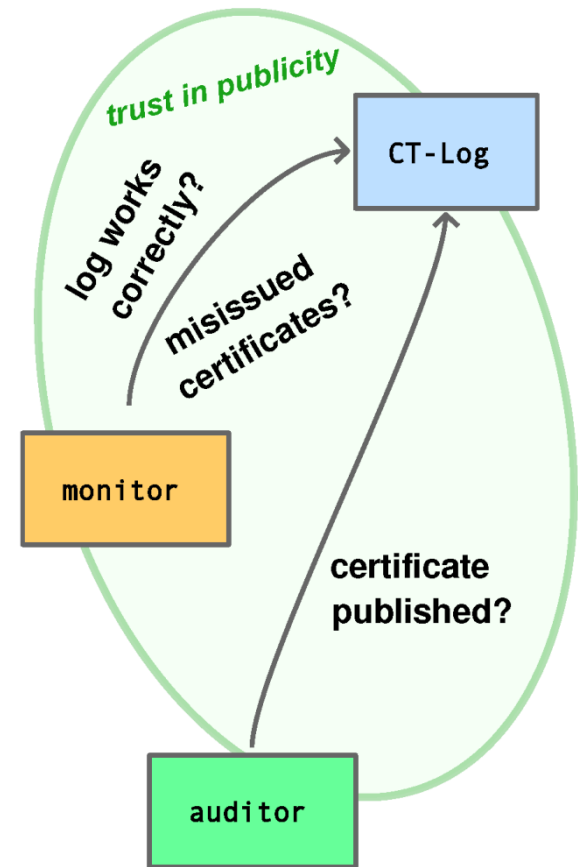


Graphics by Theodor Nolte

# CT: Replicate Certificates in Public

Approach (Google, RFC6962):

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



Graphics by Theodor Nolte

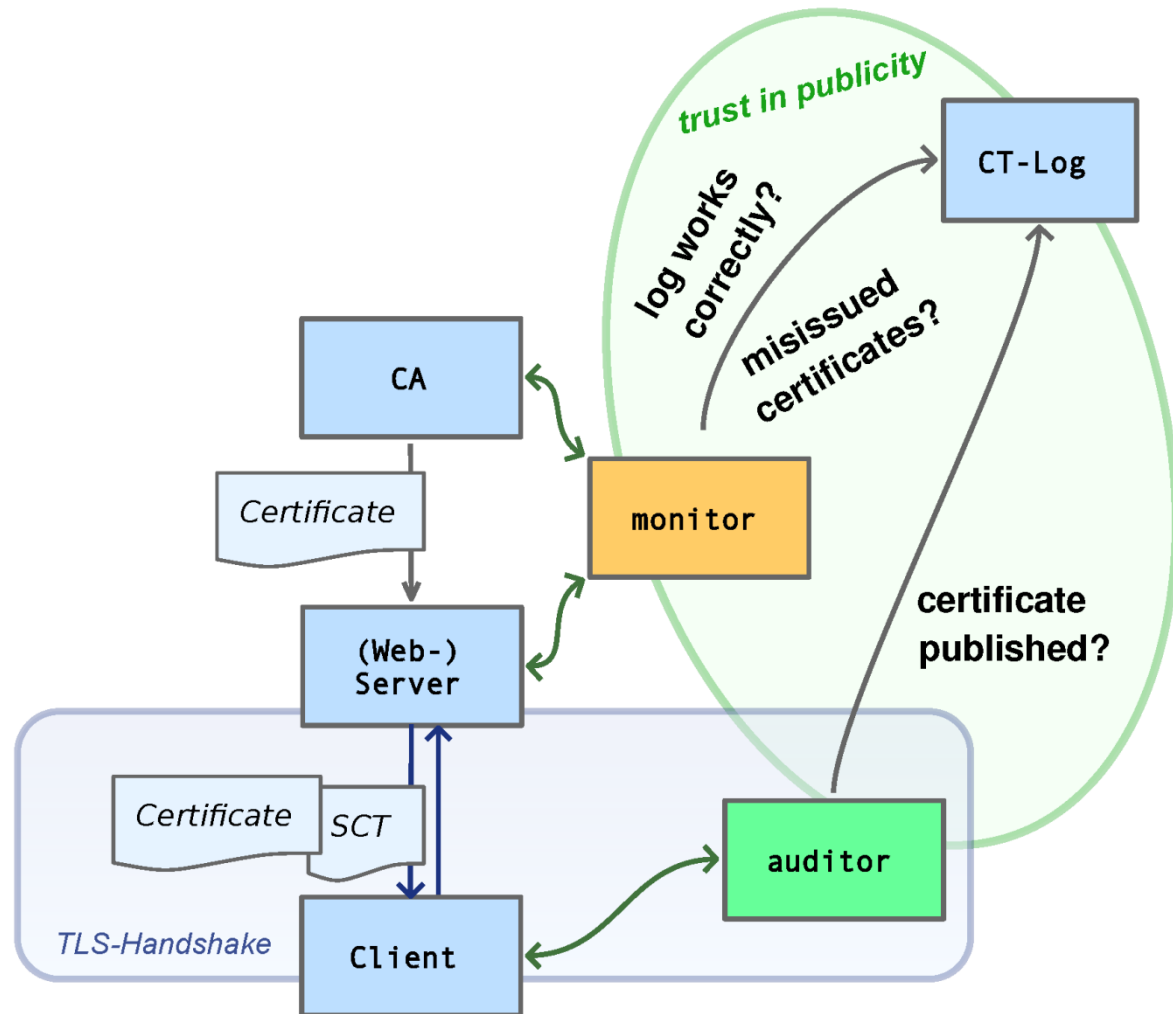
# CT Enforces Visibility

## Publication/Monitoring

- o CAs
- o Resource Owners
- o 3<sup>rd</sup> Parties

## Verification/Auditing

- o **Clients** based on SCTs

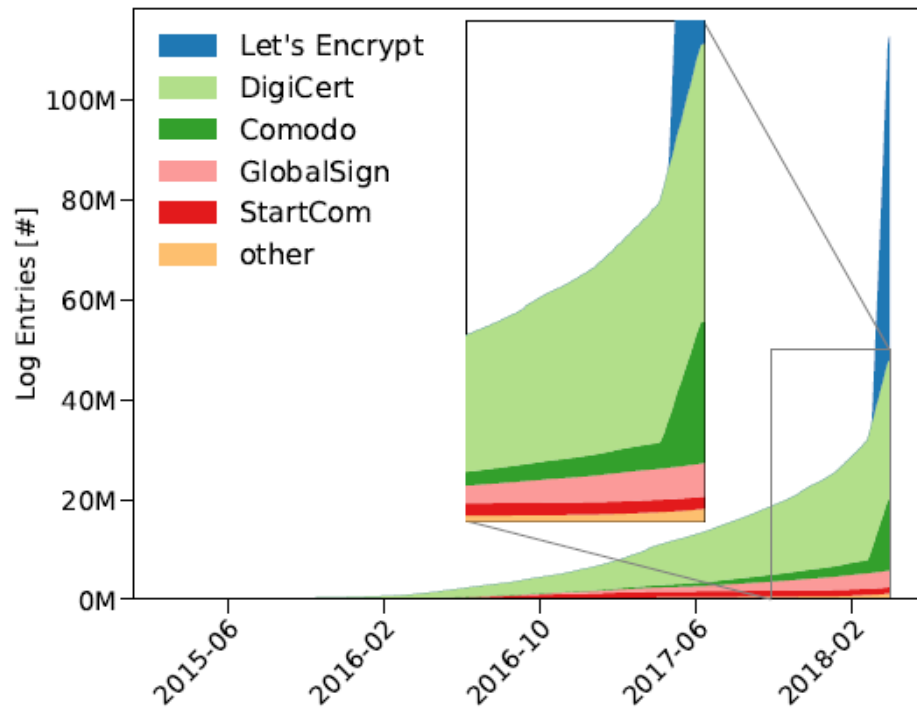


# CT Deployment [ACM IMC '18]

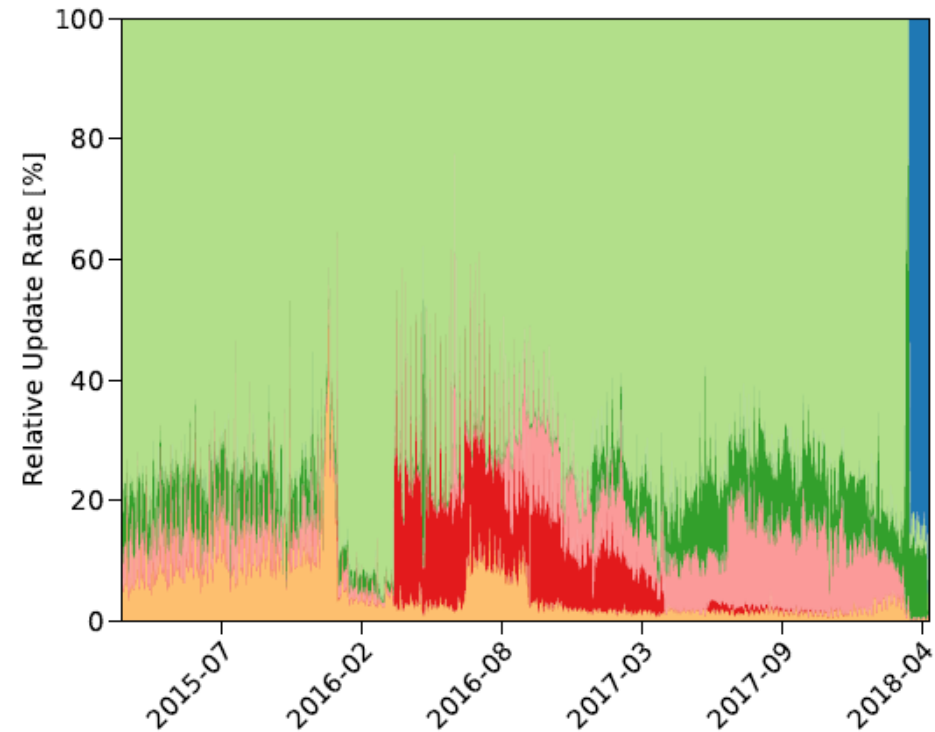
- o Initial certificate deployment in Logs remained low
- o Google announced in October 2016, they would only regard certificates trusted if published in Logs – with little impact
- o Google announced and implemented this policy in Chrome as of April 18th, 2018
- o This led to an explosion of deployment – and a sharp monopolization of CT logs



# Approaching the Chrome Deadline: Logging of Precertificates



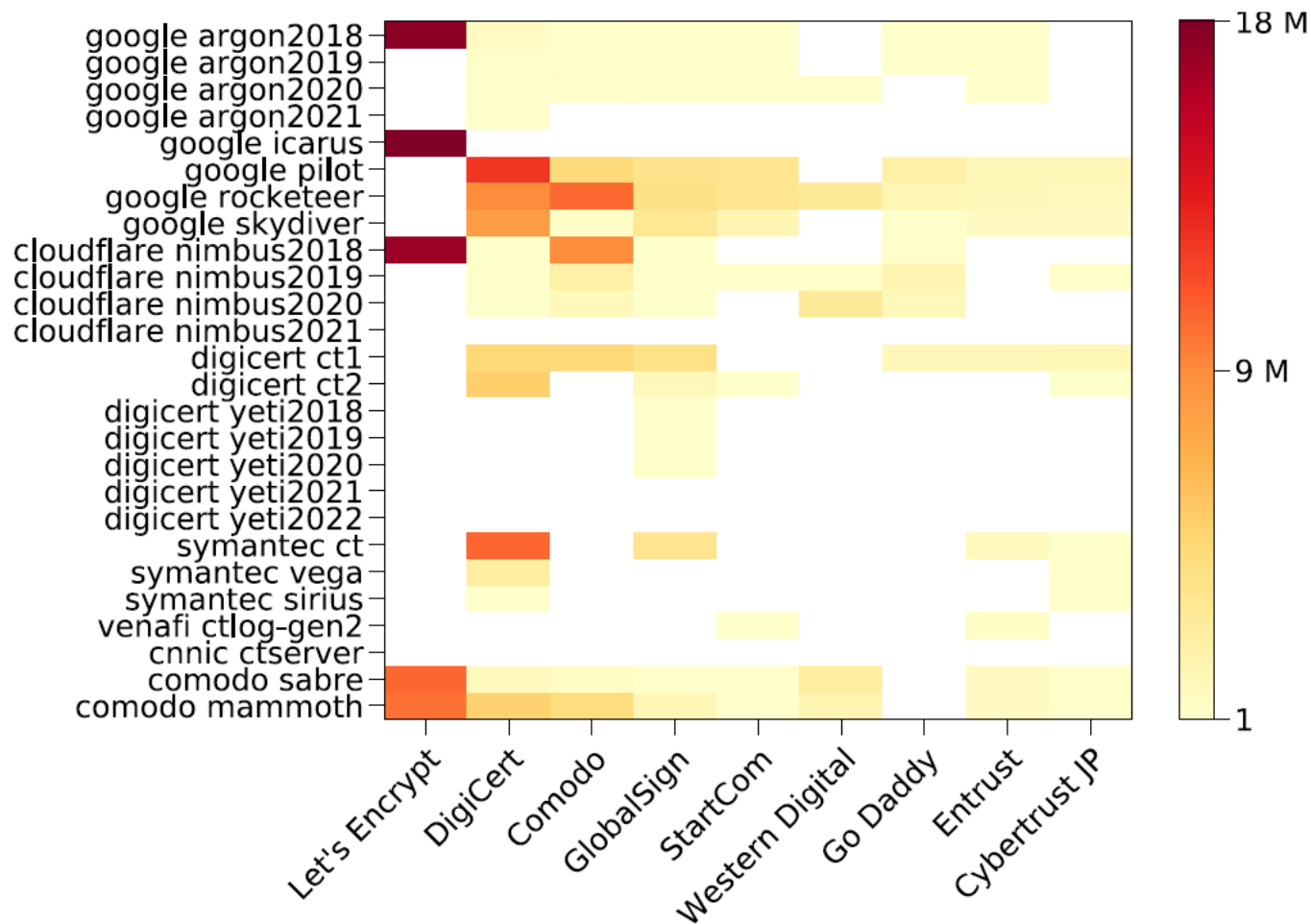
(a) Cumulative growth of logged precertificates by Certification Authority (CA).



(b) Relative update rate per CA and day. Let's Encrypt dominates after starting to log.



# Distribution of Precertificate Logging: CAs versus CT-Logs



# Leakage of DNS Subdomains

- o New attack vector brought by CT:  
Publication of (future) subdomain names
  - FQDNs of services accessible in CT-Logs
- o Subdomain enumeration – as prevented by DNSSEC – is a common attack preparation
- o Measurement: CT Honeypot
  - Inject hashed subdomain names
  - Measure DNS queries
  - Result: multiple queries within seconds



# Résumé on CT

- o 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
- o Technically issuing of illegitimate certificates remains unhindered
- o Privacy issue of CT
  - Logs see certificate queries
  - Leak subdomains
  - But reveal potential phishing domains:  
`appleid.apple.com-7etr6eti.gq`



# Bibliography

o Drafts, RFCs:

[tools.ietf.org](https://tools.ietf.org), <http://www.rfc-editor.org>

o Material for DNSSEC:

<http://www.dns-school.org/>

## CT deployment:

o Quirin Scheitle, Oliver Gasser, Theodor Nolte, Johanna Amann, Lexi Brent, Georg Carle, Ralph Holz, Thomas C. Schmidt, Matthias Wählisch, *The Rise of Certificate Transparency and Its Implications on the Internet Ecosystem*, **In:** *Proc. of ACM Internet Measurement Conference (IMC 2018)*, ACM : New York, NY, USA, October 2018.

