black hat USA 2022

DNSSEC Downgrade Attacks

Haya Shulman, **Elias Heftrig**, Michael Waidner

Why DNSSEC Downgrade Attacks?

DNS is involved in virtually all transactions on the Internet and many mechanisms rely on its security

- when determining which IP host to send packets to
- password recovery
- ACME/Domain Validation for obtaining X.509/HTTPS Certificates
- authorization of X.509 CAs and authentication of certificates
- also: SSH host key fingerprints, IPSec Keys, …

DNSSEC is the go-for solution to achieve DNS record security

- while everybody here has probably heard of downgrade attacks on TLS
- downgrade attacks on DNSSEC have not seen much attention up until now

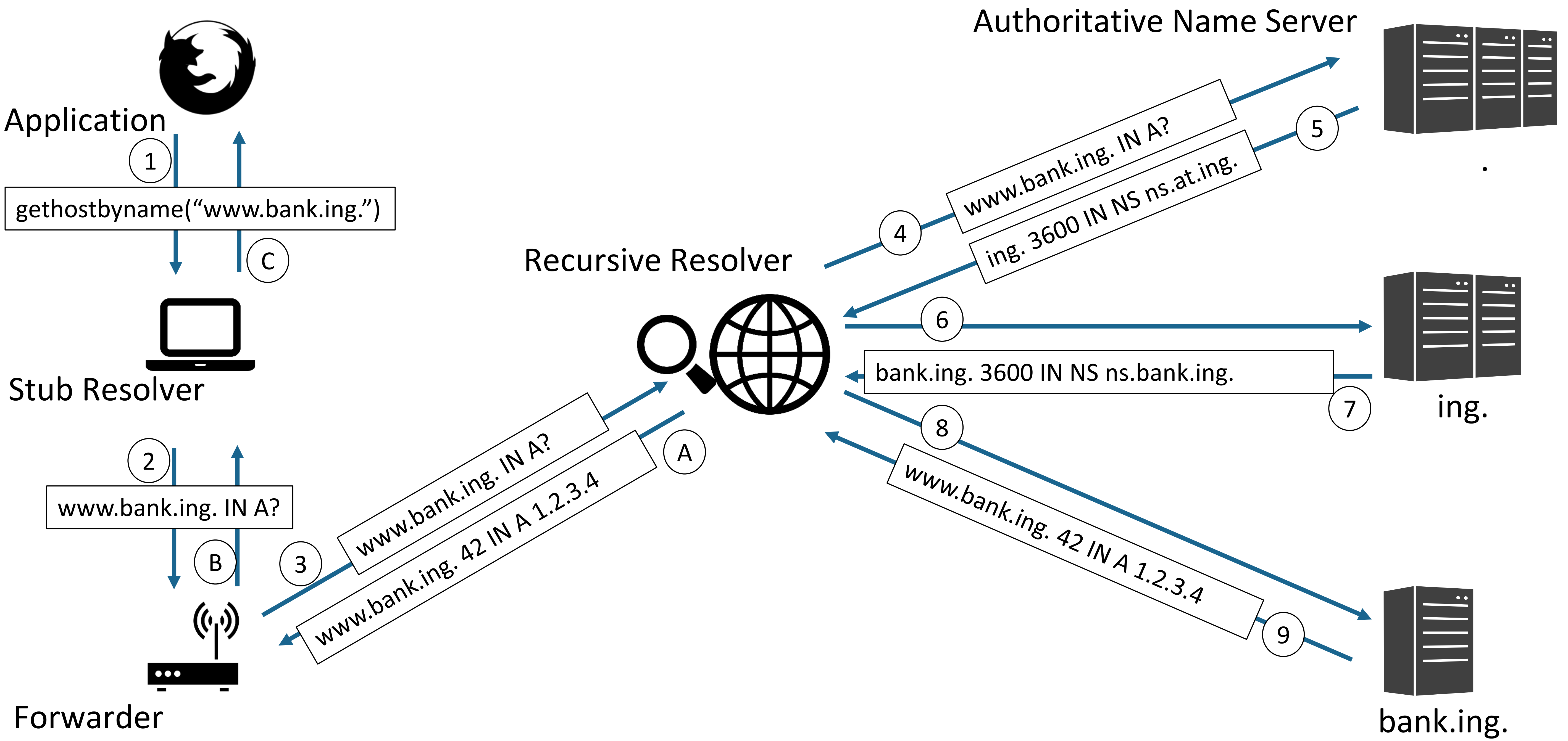
Agenda

- DNS(SEC) Refresher
- DNSSEC Downgrade Attacks
  - Attacks to Weaken Security
  - Attacks to Break Security
- Recommendations

Agenda

- ➢ **DNS(SEC) Refresher**
- ➢ DNSSEC Downgrade Attacks
    - ➢ Attacks to Weaken Security
    - ➢ Attacks to Break Security
- ➢ Recommendations

DNS Resolution – Common Setup



Application

1

gethostbyname(“www.bank.ing.”)

C

Stub Resolver

2

www.bank.ing. IN A?

B

Forwarder

3

www.bank.ing. IN A?

www.bank.ing. 42 IN A 1.2.3.4

A

Recursive Resolver

Authoritative Name Server

4

www.bank.ing. IN A?

ing. 3600 IN NS ns.at.ing.

5

.

6

bank.ing. 3600 IN NS ns.bank.ing.

7

ing.

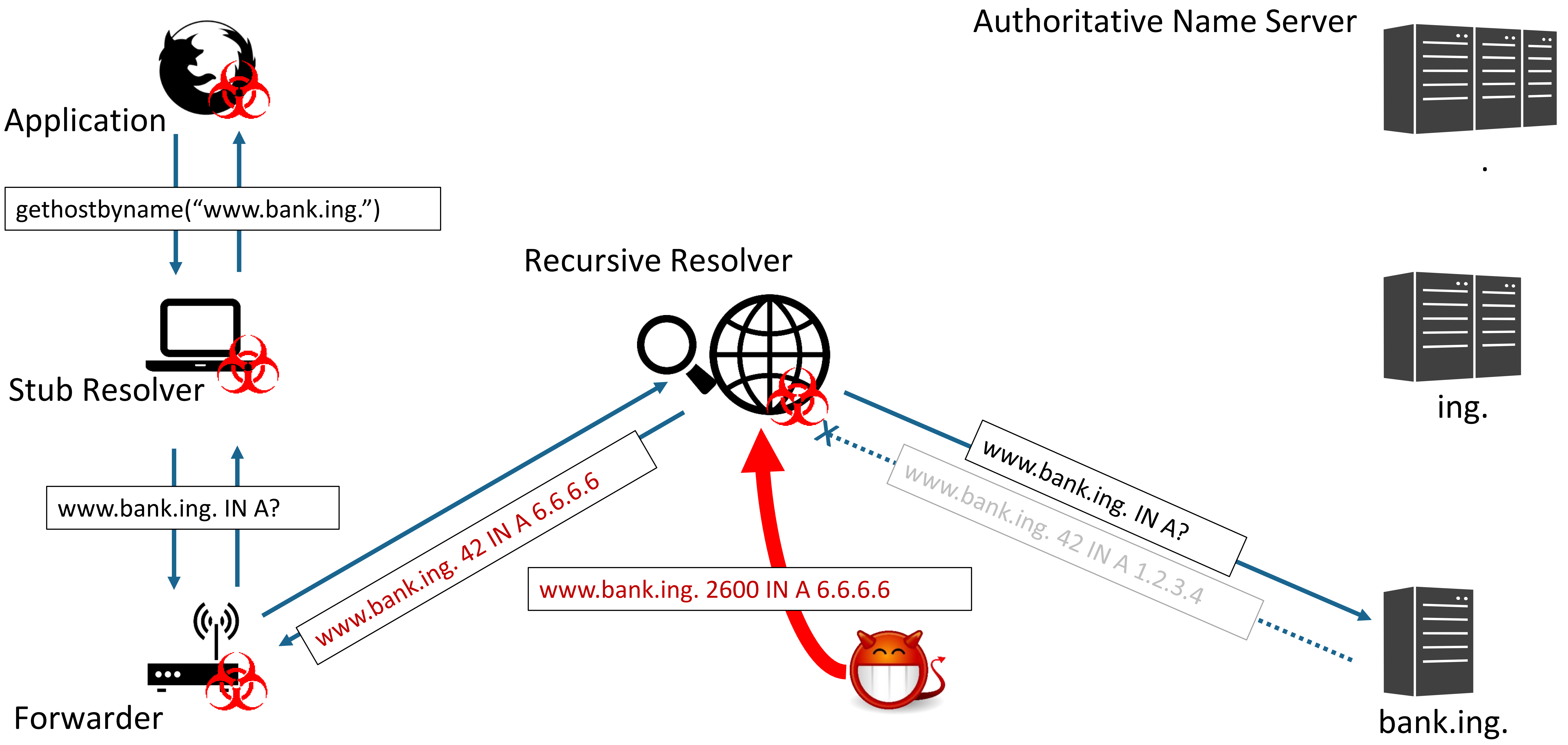
8

www.bank.ing. 42 IN A 1.2.3.4

9

bank.ing.

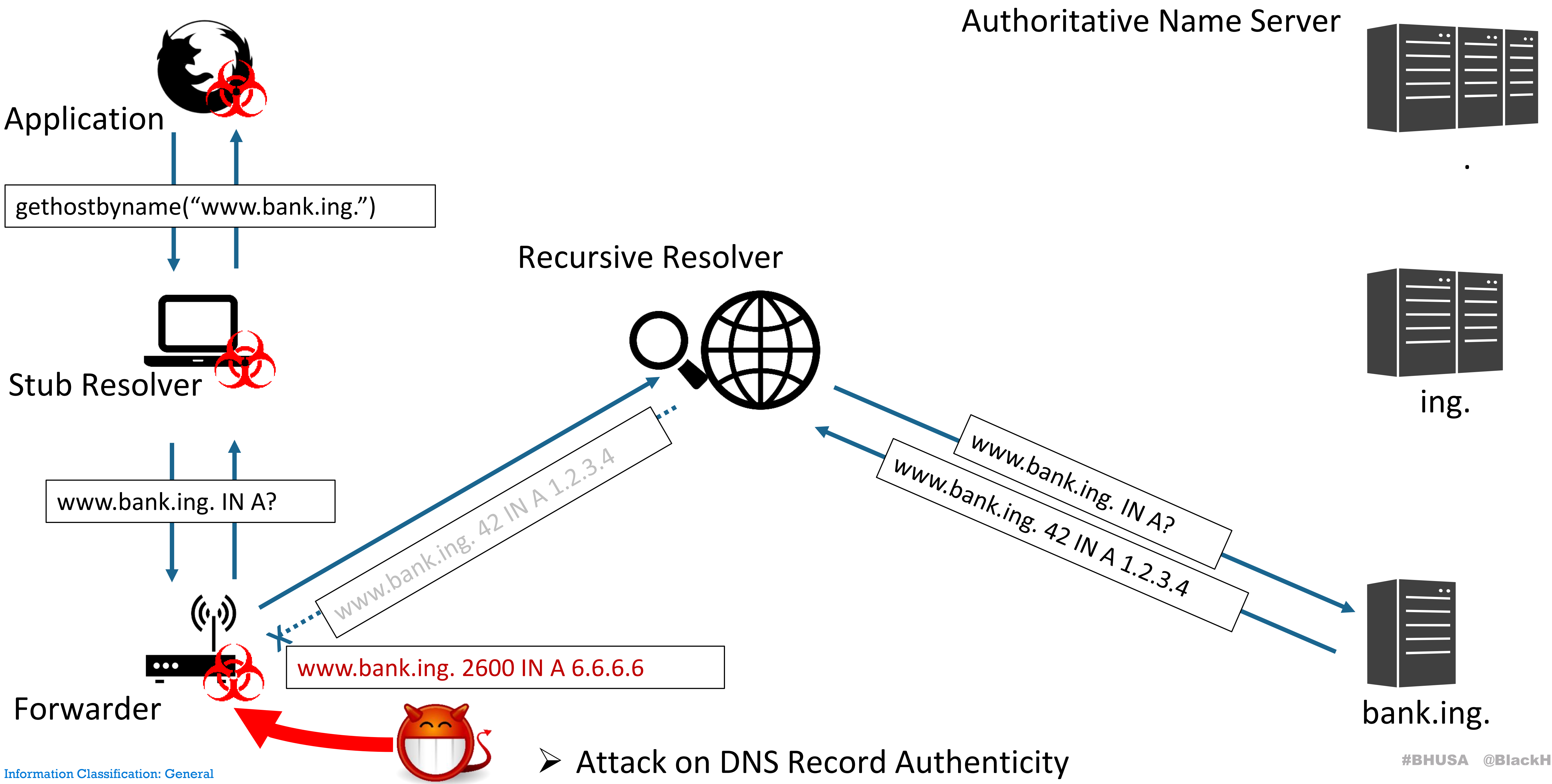
DNS Poisoning


Authoritative Name Server
.
ing.
bank.ing.
Application
gethostbyname("www.bank.ing.")
Stub Resolver
www.bank.ing. IN A?
Forwarder
Recursive Resolver
www.bank.ing. 42 IN A 6.6.6.6
www.bank.ing. 2600 IN A 6.6.6.6
www.bank.ing. IN A?
www.bank.ing. 42 IN A 1.2.3.4

- Attack on DNS Record Authenticity

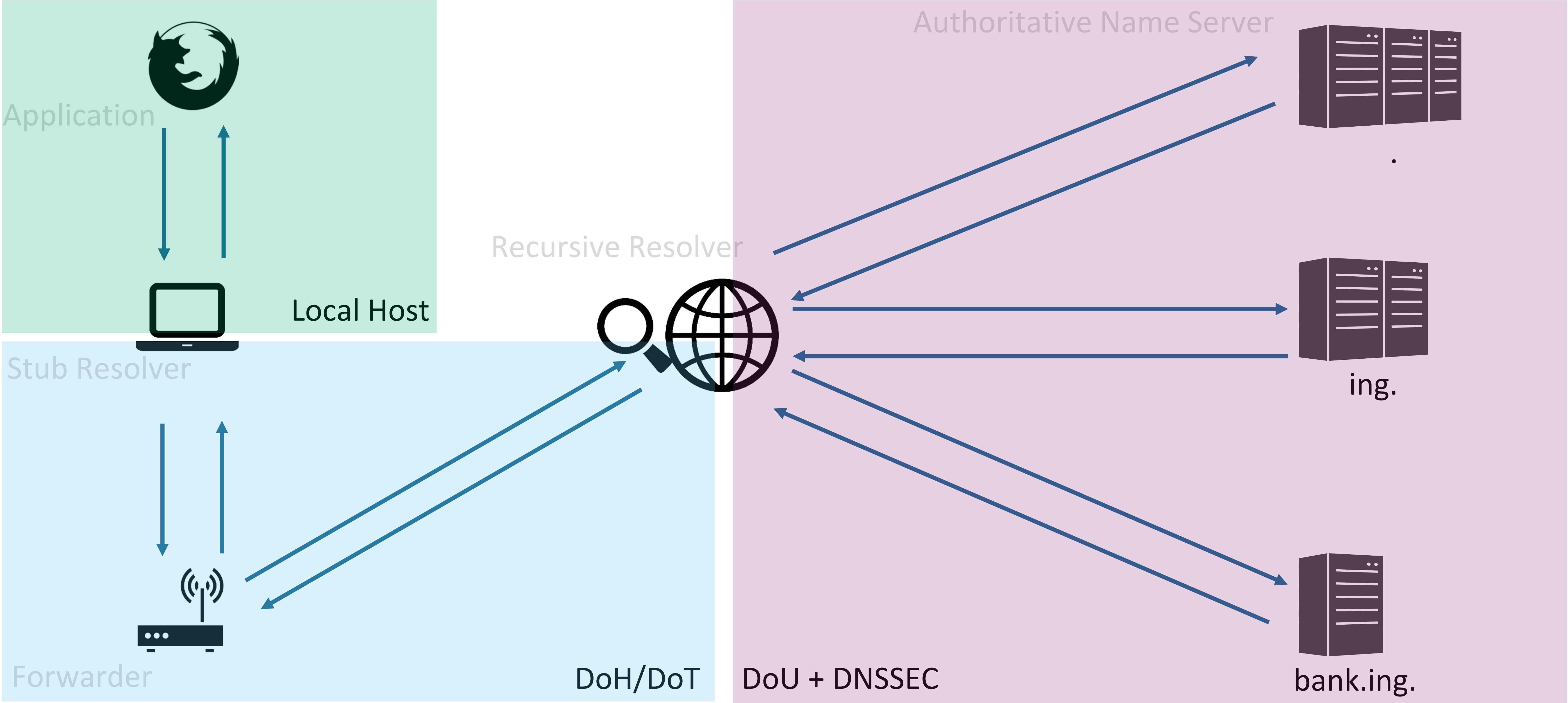
Information Classification: General

DNS Poisoning


Authoritative Name Server
.
ing.
bank.ing.
Application
gethostbyname("www.bank.ing.")
Recursive Resolver
Stub Resolver
www.bank.ing. IN A?
www.bank.ing. 42 IN A 1.2.3.4
www.bank.ing. IN A?
www.bank.ing. 42 IN A 1.2.3.4
Forwarder
www.bank.ing. 2600 IN A 6.6.6.6

- ➢ Attack on DNS Record Authenticity

Secure DNS in Practice



- Session maintenance too expensive for recursive-to-authoritative communication

DNSSEC

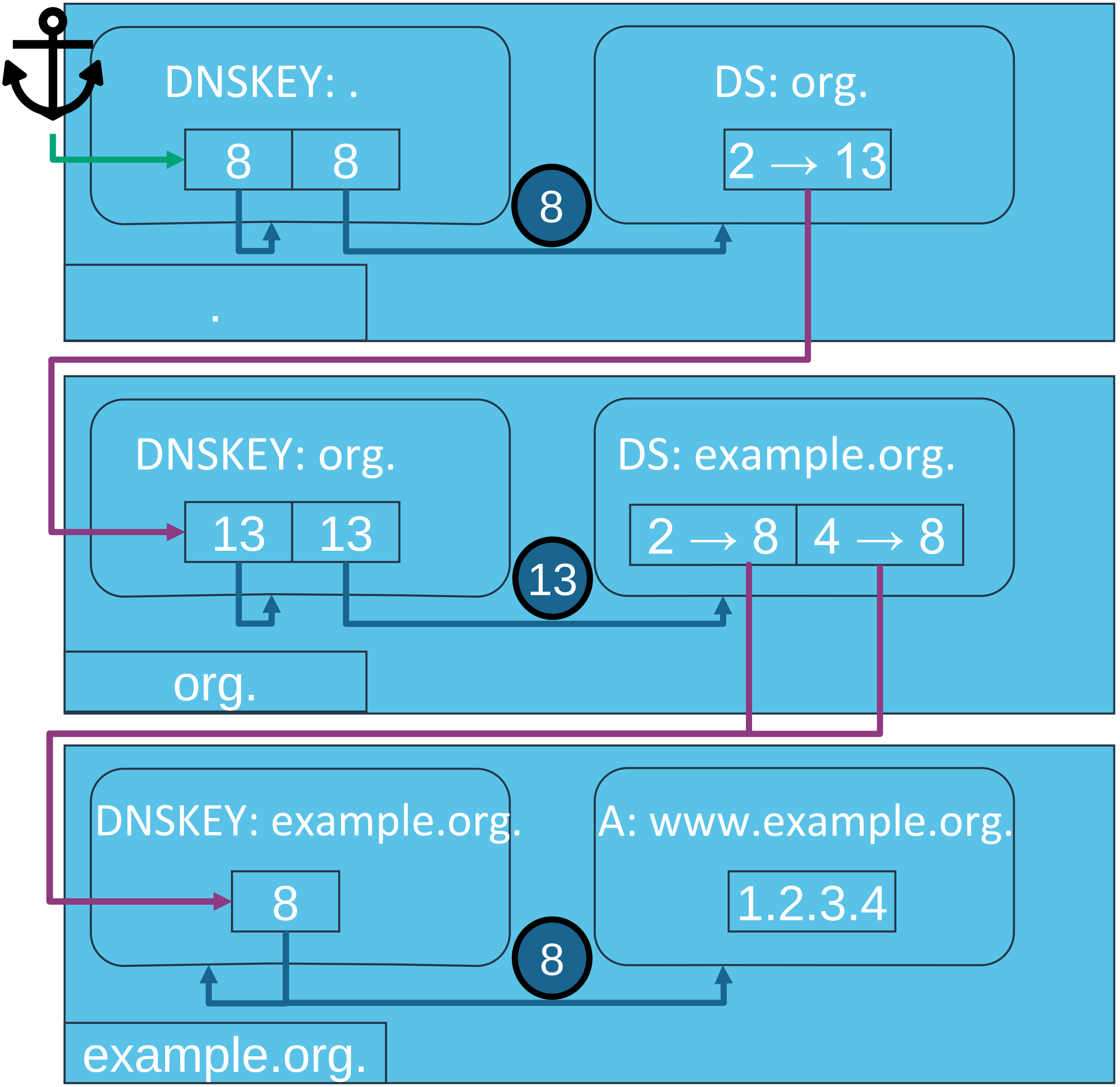
Protection Goals Provided For

- data origin authenticity
- integrity of data
- **NOT** confidentiality

Basic Principle

- protection of DNS data using cryptographic signatures
- trust in public keys delegated via a PKI
  - built into and aligned with the DNS hierarchy

DNSSEC Chain of Trust



“RRSIG” Signature Records

- cover record sets (“RRset”; same name, type and class)

DNSKEY Records

- carry public key material for verification

DS “Delegation Signer” Records

- carry digest of individual child zone DNSKEY
- conform to “certificates” in other PKIs

All DNSSEC records specify signature algorithm numbers.

DS records specify digest type numbers.

Protection of (Non-)Existence

Authenticated Denial of Existence

- uses (signed) NSEC-type records to mark empty intervals in the name space
  - specifies record types present at interval boundaries
- does not protect record presence at the level of signature algorithms

DNSSEC Record Presence Requirement for Signature Algorithms

DS ⟶ DNSKEY ⟶ RRSIGs on all zone data

- $\exists\ DS\ with\ algorithm\ a\ \Rightarrow\ \exists\ DNSKEY\ with\ algorithm\ a$
- $\exists\ DNSKEY\ with\ algorithm\ a\ \Rightarrow\ \forall\ RRsets\ in\ zone{:}\ \exists\ RRSIG\ with\ algorithm\ a$

DNSSEC Signature Algorithms

Number	Mnemonics	DNSSEC Signing	DNSSEC Validation
1	RSAMD5	MUST NOT	MUST NOT
3	DSA	MUST NOT	MUST NOT
5	RSASHA1	NOT RECOMMENDED	MUST
6	DSA-NSEC3-SHA1	MUST NOT	MUST NOT
7	RSASHA1-NSEC3-SHA1	NOT RECOMMENDED	MUST
8	RSASHA256	MUST	MUST
10	RSASHA512	NOT RECOMMENDED	MUST
12	ECC-GOST	MUST NOT	MAY
13	ECDSAP256SHA256	MUST	MUST
14	ECDSAP384SHA384	MAY	RECOMMENDED
15	ED25519	RECOMMENDED	RECOMMENDED
16	ED448	MAY	RECOMMENDED
253	PRIVATE	(MAY)	(MAY)
254	PRIVATE (OID)	(MAY)	(MAY)

SHA1

RSA

~ more secure

phasing out

ECDSA

EdDSA

private

phasing in

➢ Rules for Algorithm Support in DNSSEC Software, acc. [RFC8624]

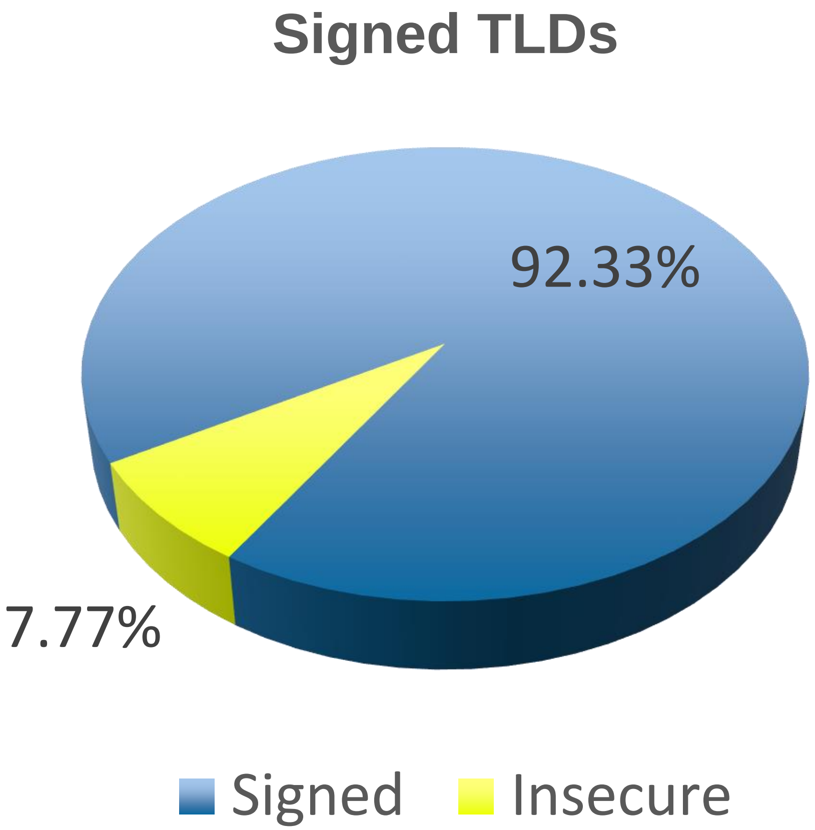
DNSSEC DS Digest Types

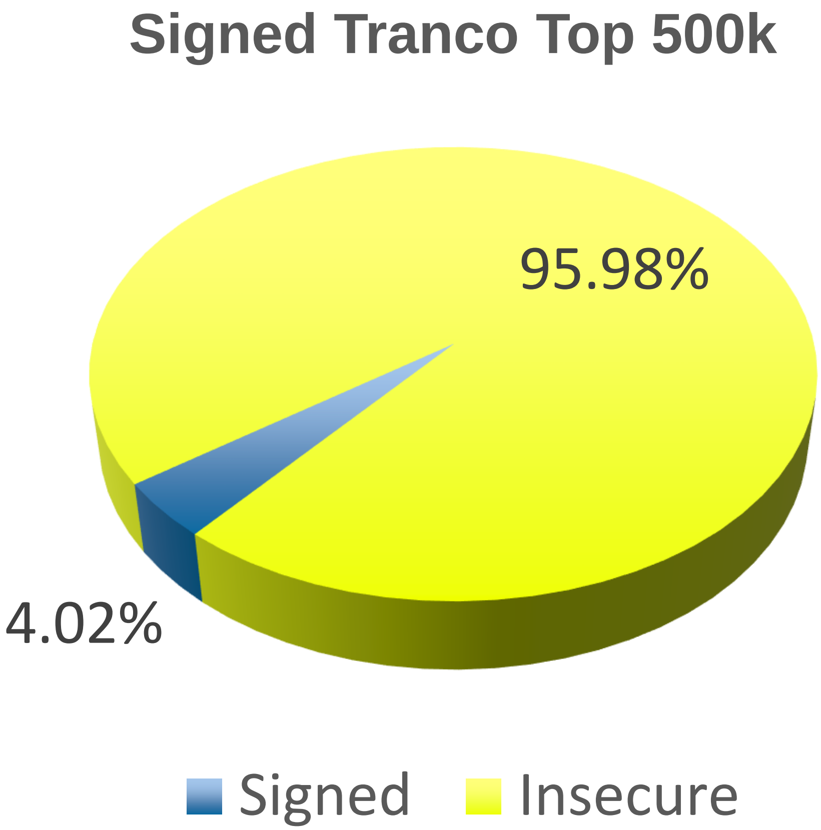
Number	Mnemonics	DNSSEC Delegation	DNSSEC Validation
1	SHA-1	MUST NOT	MUST
2	SHA-256	MUST	MUST
3	GOST R 34.11-94	MUST NOT	MAY
4	SHA-384	MAY	RECOMMENDED

in active use

- Rules for DS Digest Type Support in DNSSEC Software, acc. [RFC8624]

Investigated Domains

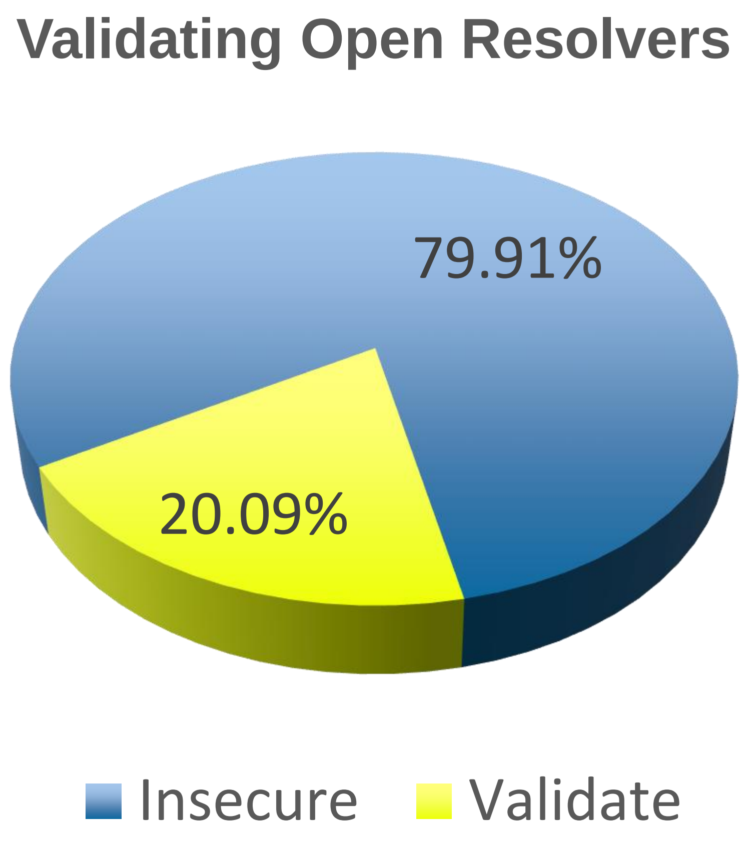




Signed Domains

- ➢ 1373 Top-Level Domains (of 1487)
- ➢ 20083 Tranco Domains (of Top 500k)
  - ➢ disregarding app. 9k domains without a validation path from the DNS root

Investigated Resolvers



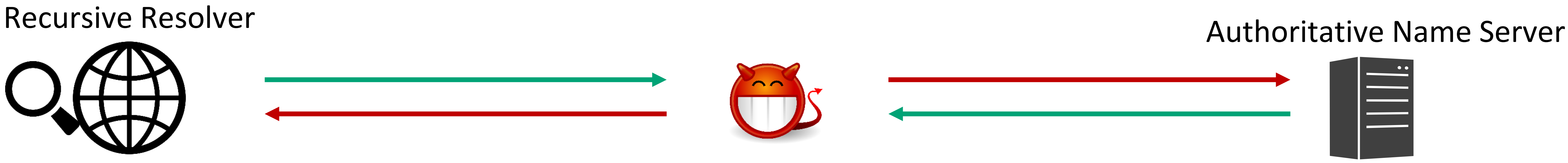
Resolvers

- 9 resolvers in the lab (Bind, Unbound, Knot, PowerDNS; 5 Windows Server Versions)
- 8 popular open resolver services (Google, Cloudflare, …)
- 15k openly accessible resolvers from a port scan on the IPv4 address space (app. 3k validating resolvers)

Agenda

- DNS(SEC) Refresher
- **DNSSEC Downgrade Attacks**
  - Attacks to Weaken Security
  - Attacks to Break Security
- Recommendations

Downgrade Attacks on DNSSEC



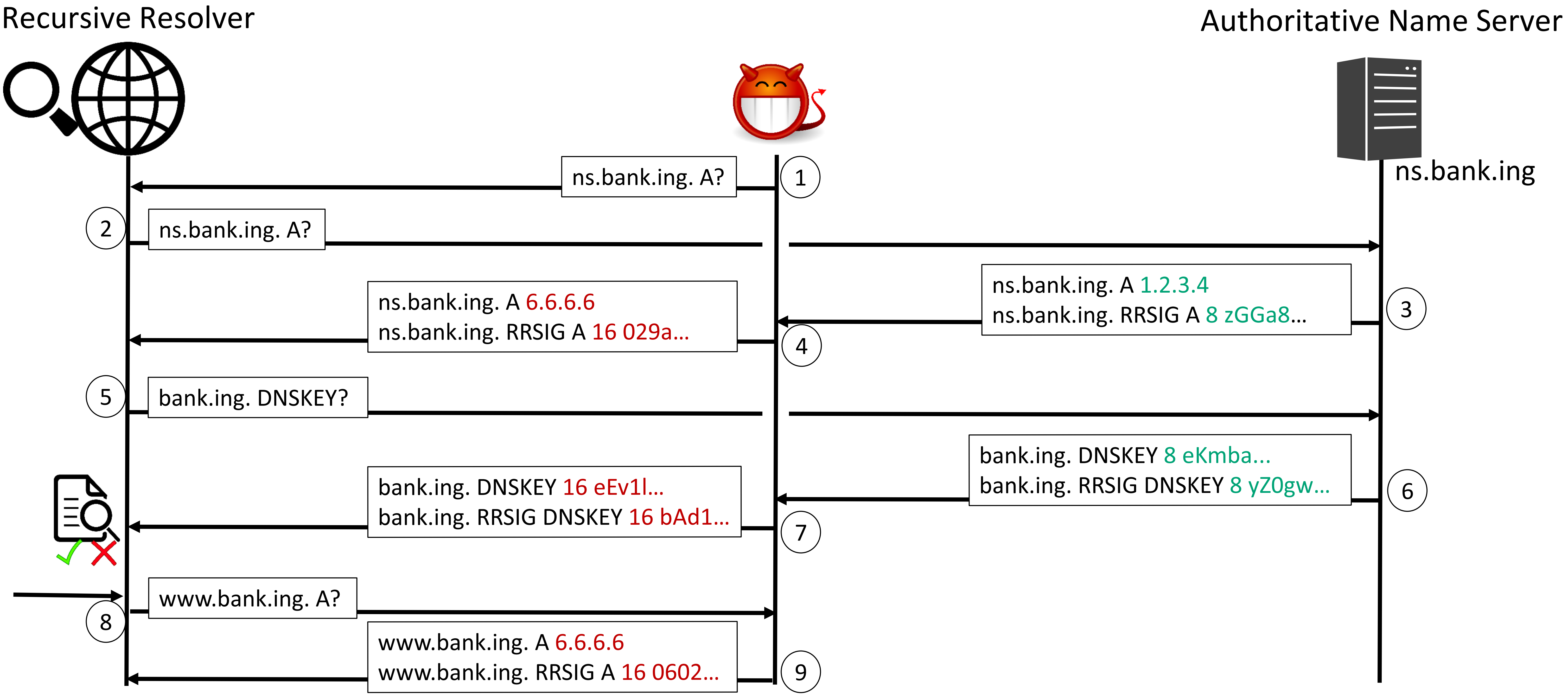
Attacker Model: On-path Attacker (~ Threat Model of DNSSEC)

- positioned between the resolver and the authoritative server
- can send, read, modify, duplicate, delay, suppress, … messages
- does not know cryptographic secrets

Further Assumptions (to keep explanations simple)

- attacker can cause trigger resolution by the resolver
- empty caches

Downgrade Attacks on DNSSEC



1 ns.bank.ing. A?

2 ns.bank.ing. A?

3 ns.bank.ing. A 1.2.3.4
ns.bank.ing. RRSIG A 8 zGGa8…

4 ns.bank.ing. A 6.6.6.6
ns.bank.ing. RRSIG A 16 029a…

5 bank.ing. DNSKEY?

6 bank.ing. DNSKEY 8 eKmba…
bank.ing. RRSIG DNSKEY 8 yZ0gw…

7 bank.ing. DNSKEY 16 eEv1l…
bank.ing. RRSIG DNSKEY 16 bAd1…

8 www.bank.ing. A?

9 www.bank.ing. A 6.6.6.6
www.bank.ing. RRSIG A 16 0602…

Information Classification: General

Agenda

- DNS(SEC) Refresher
- DNSSEC Downgrade Attacks
  - **Attacks to Weaken Security**
  - Attacks to Break Security
- Recommendations

Attacks to Weaken Security

Goal

- ➢ make the resolver use the weakest possible validation path
- ➢ and attack that weakest link in the chain of trust
- ➢ (very) roughly conforms to downgrade to “Export” in SSL

Presented here

- ➢ Downgrading to a weaker DS digest
- ➢ Downgrading to a weaker signature

The Case of SHA-1 in DNSSEC

A Note on SHA-1

- “broken” in terms of cryptanalysis
- practical attacks on DNSSEC are expected in the near future
  - attacks for non-DNSSEC cases have been demonstrated in 2019

SHA-1 in DNSSEC

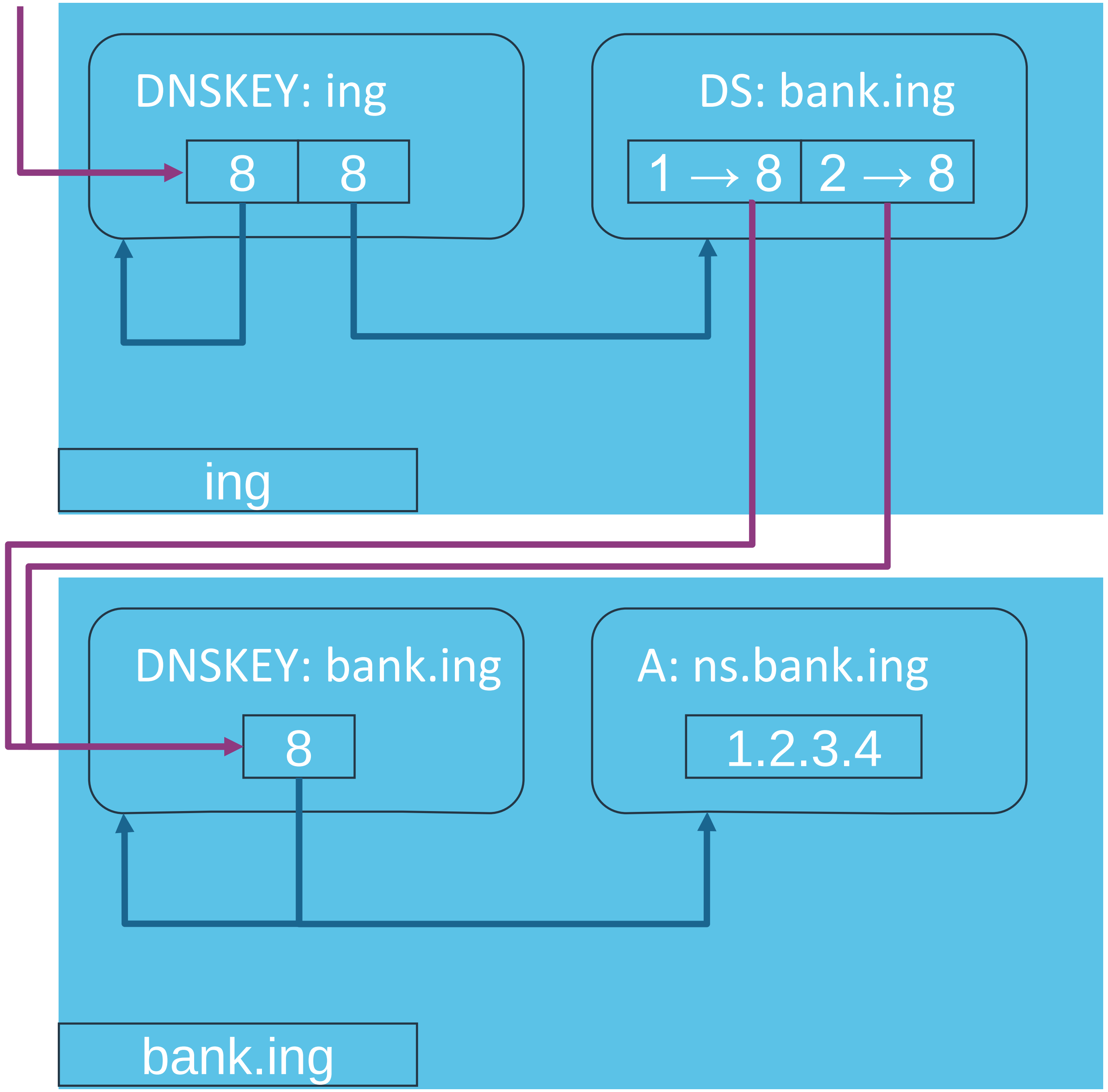
- being phased out since about 2019, but still widely used
  - algorithms 5 and 7 (“NOT RECOMMENDED”)
  - digest type 1 (“MUST NOT”)
- resolvers must still support it
  - virtually all do

		DS	DNSKEY
TLDs	any	8.64%	4.10%
	exclusively	0.22%	3.30%
Tranco	any	11.33%	6.22%
	exclusively	3.38%	5.81%

- Shares of Secure Zones using SHA-1

Downgrade to Weaker DS Digest

Downgrade to Weaker DS Digest



Preconditions

- two DS records in parent zone
  - one stronger digest, one weaker
  - both supported by the resolver
- one DNSKEY in victim zone matching both DS digests

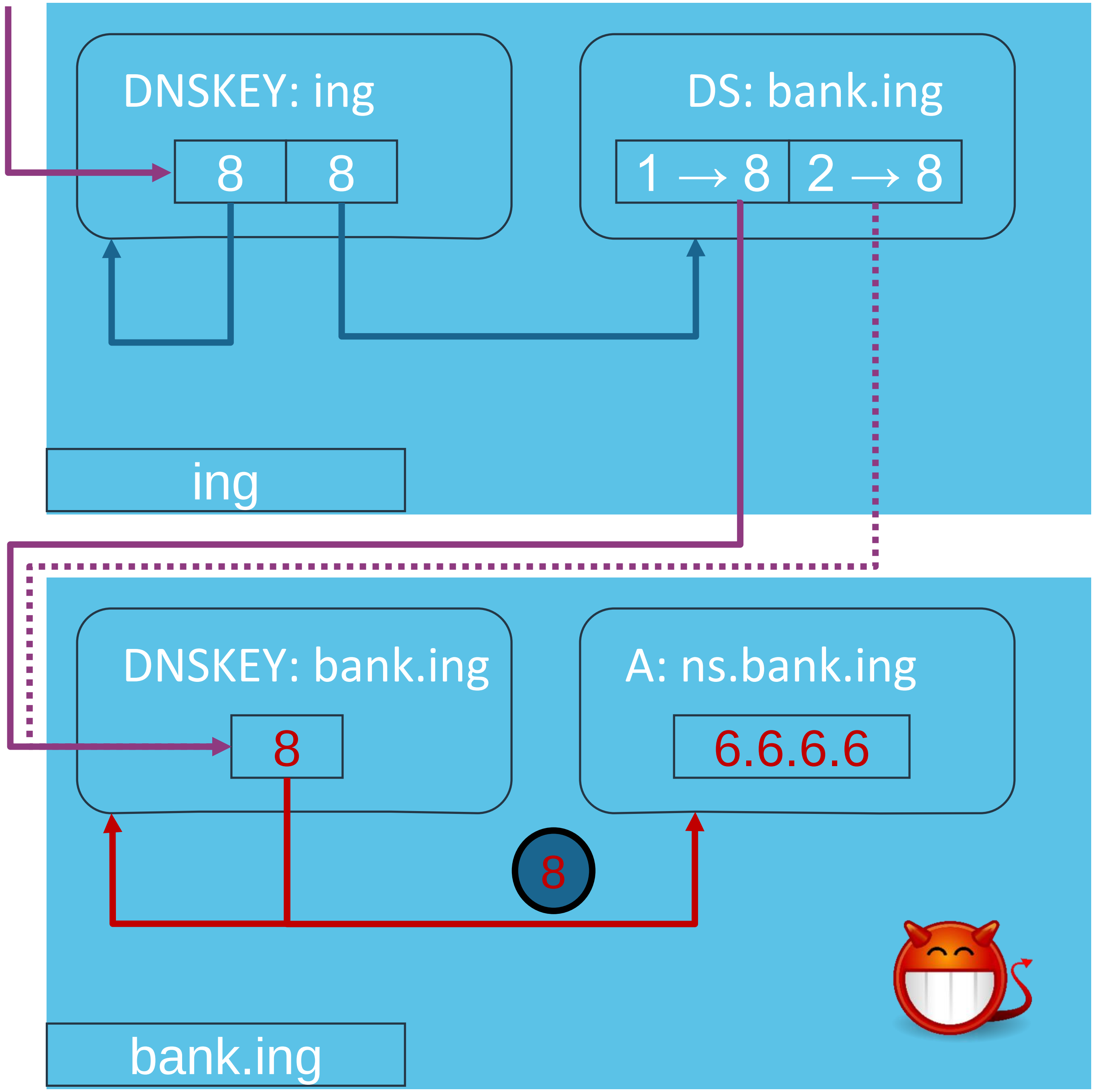
Assumption

- attacker can break the weaker digest

Note

- as outlined in RFC 4509 for SHA-1/SHA-256 (1 and 2)

Downgrade to Weaker DS Digest



Procedure

- attacker forges DNSKEY for the weaker algorithm
- replaces authentic DNSKEY and all its signatures
- spoofs target data

Observations

- stronger digest does not match the DNSKEY
- path via DS with stronger digest becomes invalid

Will the resolver fall back to the validation path via the weaker DS record?

Downgrade to Weaker DS Digest

Many Vulnerable Resolvers

Fallback to	Open Resolvers	Lab	Popular
Any weaker DS	93%	8/9	8/8
SHA-1 DS	24%	6/9	6/8

Lab

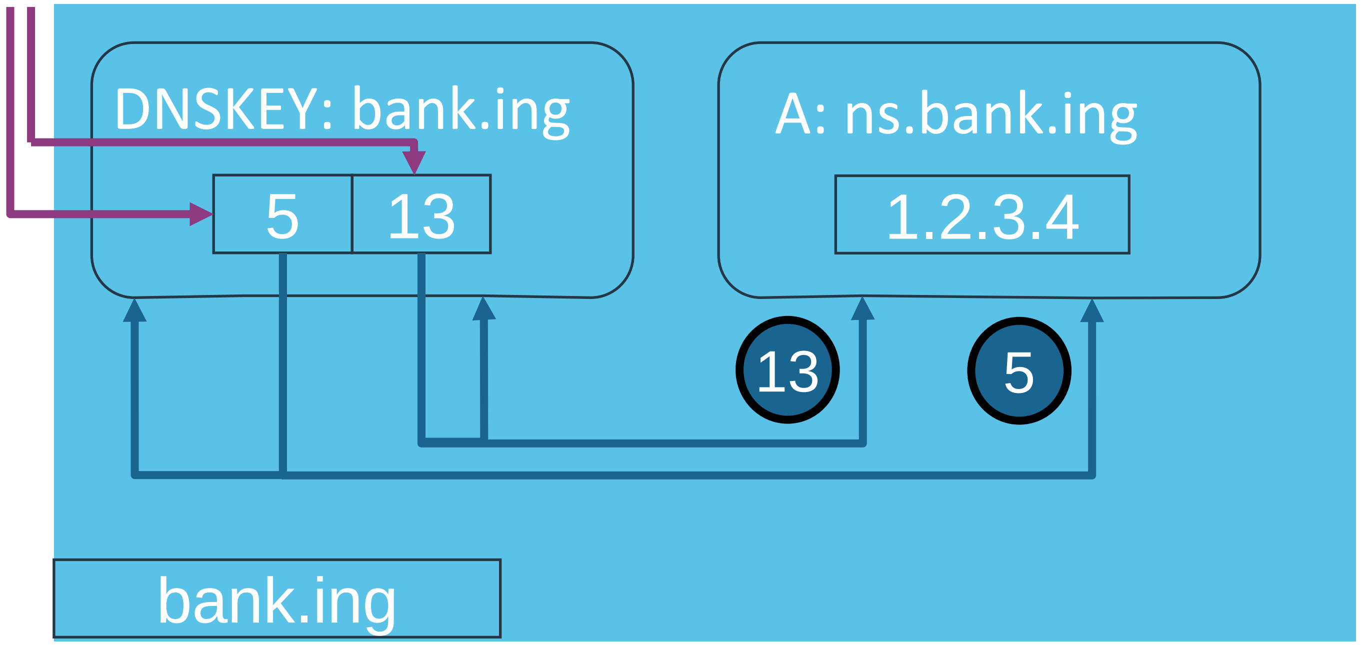
- only PowerDNS enforces strongest possible DS
- BIND9 and Knot Resolver enforce stronger-than-SHA1 DS

Popular Open Resolvers

- only Google and CZ.NIC enforce stronger-than-SHA1 DS

Downgrade to Weaker Signature

Downgrade to Weaker Signature



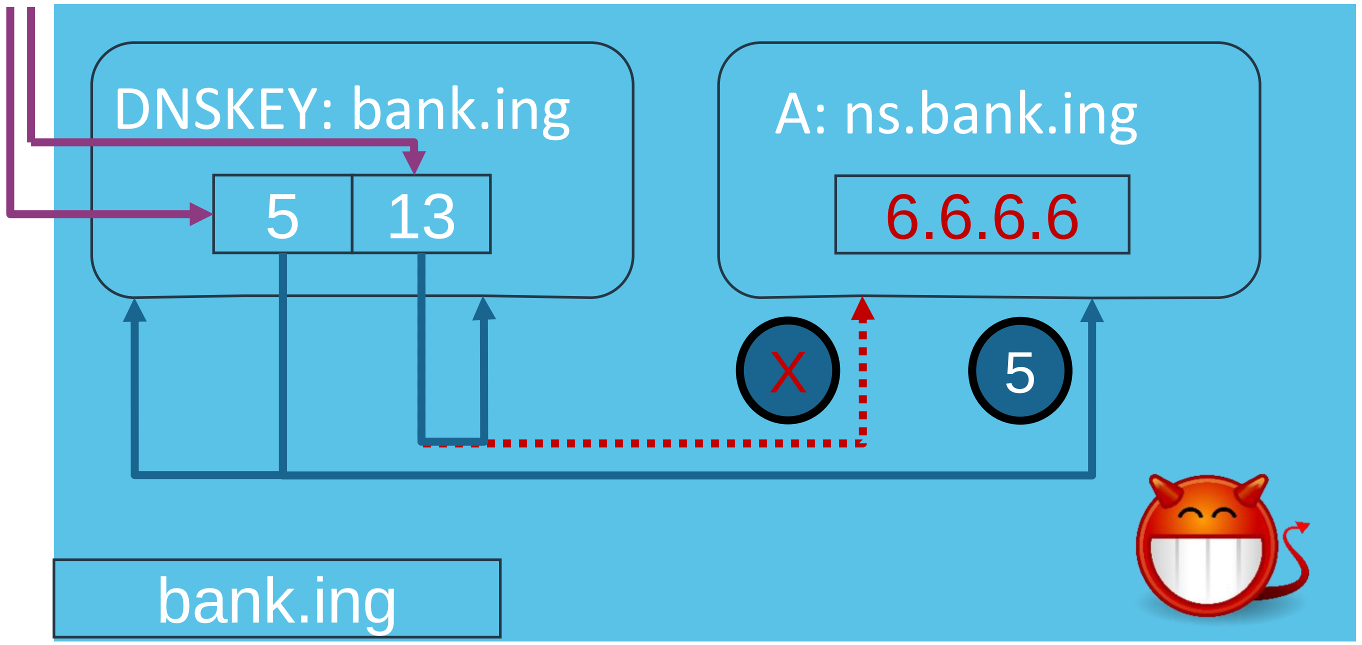
Preconditions

- zone signed with two algorithms
  - one weaker, one stronger
  - both supported by the resolver
- e.g. typical zone migrating to a new algorithm

Assumption

- attacker can forge zone data for the weaker one

Downgrade to Weaker Signature



Procedure

- attacker just places spoofed zone data in the DNS response

Observations

- Signatures of the stronger key become invalid.
- optional attacker measure: strip them off

Will the resolver accept the weaker signatures, even if stronger ones should be present and valid?

Downgrade to Weaker Signature

RFC 5702 on Algorithm Presence (DS → DNSKEY → RRSIGS on all zone data)

> “**Since each RRSet MUST be signed with each algorithm present** in the DNSKEY RRSet at the zone apex (see Section 2.2 of [RFC4035]), **a malicious party cannot filter out the RSA/SHA-2 RRSIG and force the validator to use the RSA/SHA-1 signature if both are present** in the zone. This should provide resilience against algorithm downgrade attacks, if the validator supports RSA/SHA-2.”

So… We are secure?

Downgrade to Weaker Signature

Affected Resolvers

- Turns out… all investigated resolvers fall back to weaker RRSIGS.
- even to SHA-1-based ones

RFC 6840 on Algorithm Presence (DS → DNSKEY → RRSIGS on all zone data)

> “This requirement applies to servers, not validators. **Validators SHOULD accept any single valid path.** They SHOULD NOT insist that all algorithms signaled in the DS RRset work, and they MUST NOT insist that all algorithms signaled in the DNSKEY RRset work.”

- facilitates algorithm updates of very large zones
- but bites us back while we are getting rid of SHA-1. Bad Luck ☹

Review: Attacks to Weaken DNSSEC

Countermeasures against Downgrading to Weaker DS

- require the strongest present DS digest to be used for construction of the validation path
  - especially if the weaker one is SHA-1

Countermeasures against Downgrading to Weaker Signature

- we can essentially just hope zones migrate away fast enough
  - insisting on RRSIGs of the strongest algorithm from DNSKEY risks disconnecting secure domains

- against attacker who cannot strip off records
  - insist that the strongest *present* algorithm signatures work

Agenda

- DNS(SEC) Refresher
- DNSSEC Downgrade Attacks
  - Attacks to Weaken Security
  - **Attacks to Break Security**
- Recommendations

Attacks to Break Security

Motivation

- breaking a “weaker” algorithm is still quite a bar to jump
- even SHA-1 is not quite there, yet

DNSSEC Downgrade Attacks to Break Security

- we found ways around breaking crypto
- in effect, roughly comparable to Downgrade to NULL / SSL Stripping
- exploit the validation logic that assigns security states to DNS data

DNS Record Security States

Secure

✓

- The full chain of trust is proven to be authentic.
- response to client carries records in question and the RRSIG(s) covering them
  - AD message flag set, but effectively ignored by most clients

Bogus

✗

- no valid chain of trust could be constructed, e.g. because
  - signatures failed to validate
  - DNSSEC records missing
- SERVFAIL error response to client

DNS Record Security States

Indeterminate

- not too relevant here
- assigned to infrastructure data during referrals (NS and A of NS)
- or in case of missing trust anchors (weird PKI entry)

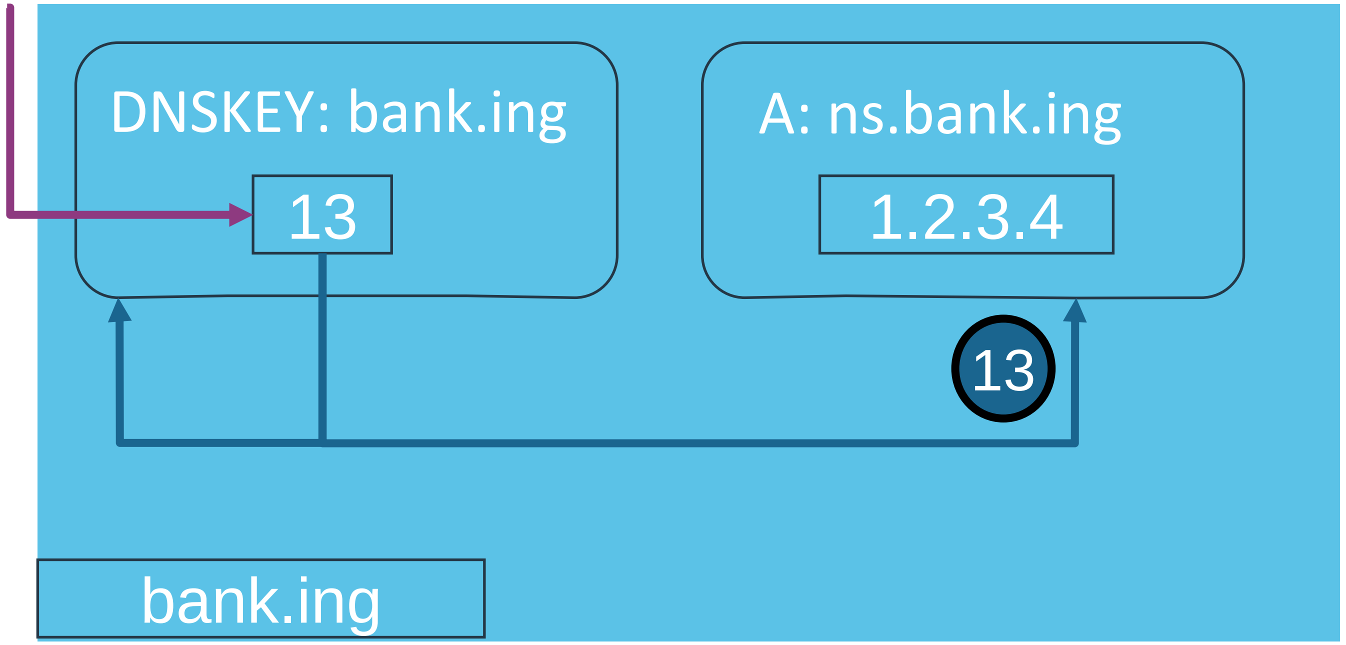
Insecure

- provably not secured in a way the resolver can validate
- e.g. by authenticated proof that **no** DS record exists at some point in the DNS hierarchy
  - authenticated DS records with unsupported digest types or signature algorithms “do not exist”
- response to client carries records in question, without AD flag

The next attacks trick the resolver into marking records *Insecure.*

Rewriting RRSIG Algorithm Numbers

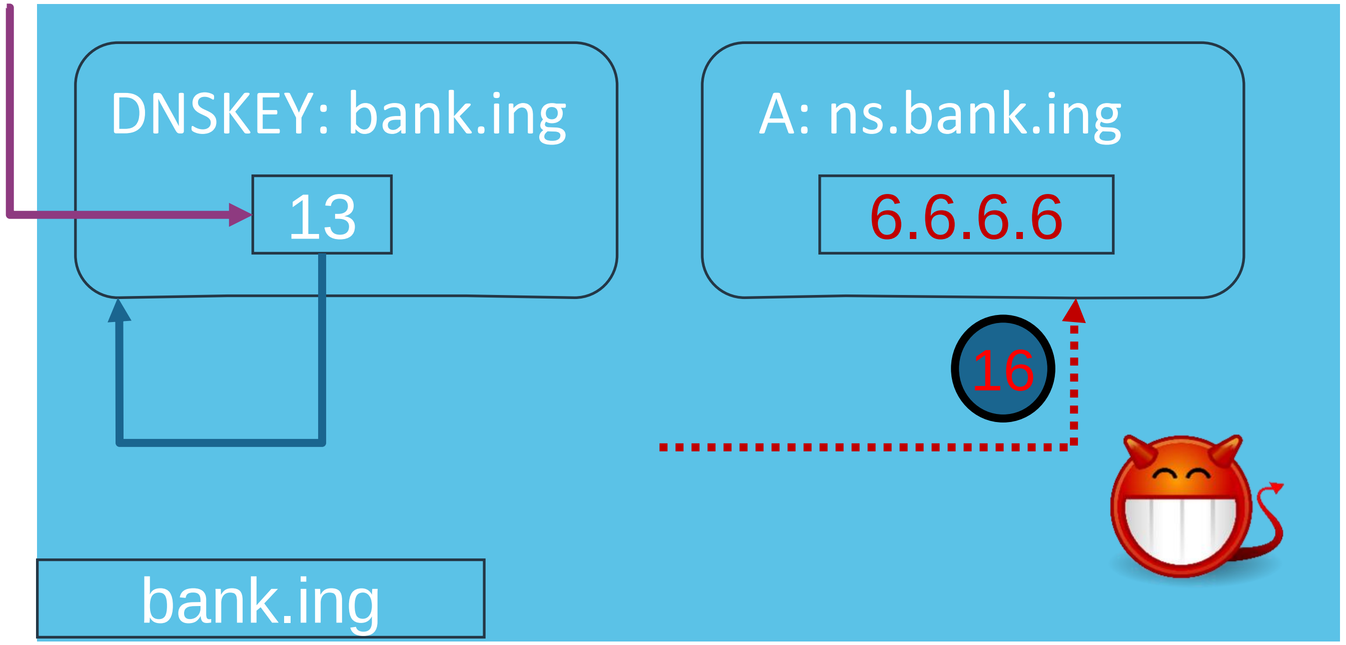
Rewriting RRSIG Algorithm Numbers



Preconditions

- ➢ just any properly protected DNSSEC zone
  - ➢ we tested for single-algorithm zones

Rewriting RRSIG Algorithm Numbers



Procedure

- attacker rewrites signature algorithm number
  - to one the resolver does not support

Note

- chain of trust broken at the last link

Vulnerable Resolvers

- Google Public DNS
- reported and fixed

Downgrade via Complex Chain of Trust

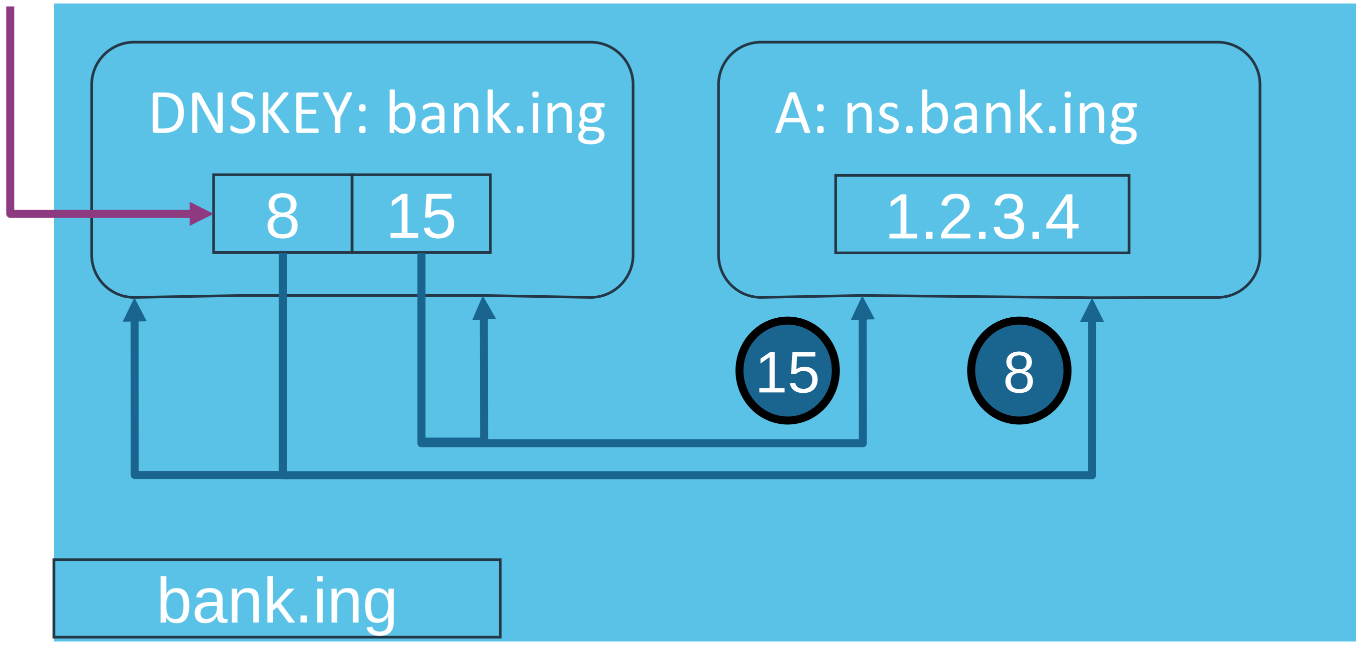
- Let’s see what can go wrong when things get experimental.

Situation

- a zone operator adds a freshly standardized algorithm
  - which is not supported by many resolvers yet
- or uses a private algorithm in addition to a non-private one

Stripping off Supported RRSIGs

Stripping Off Supported RRSIGs



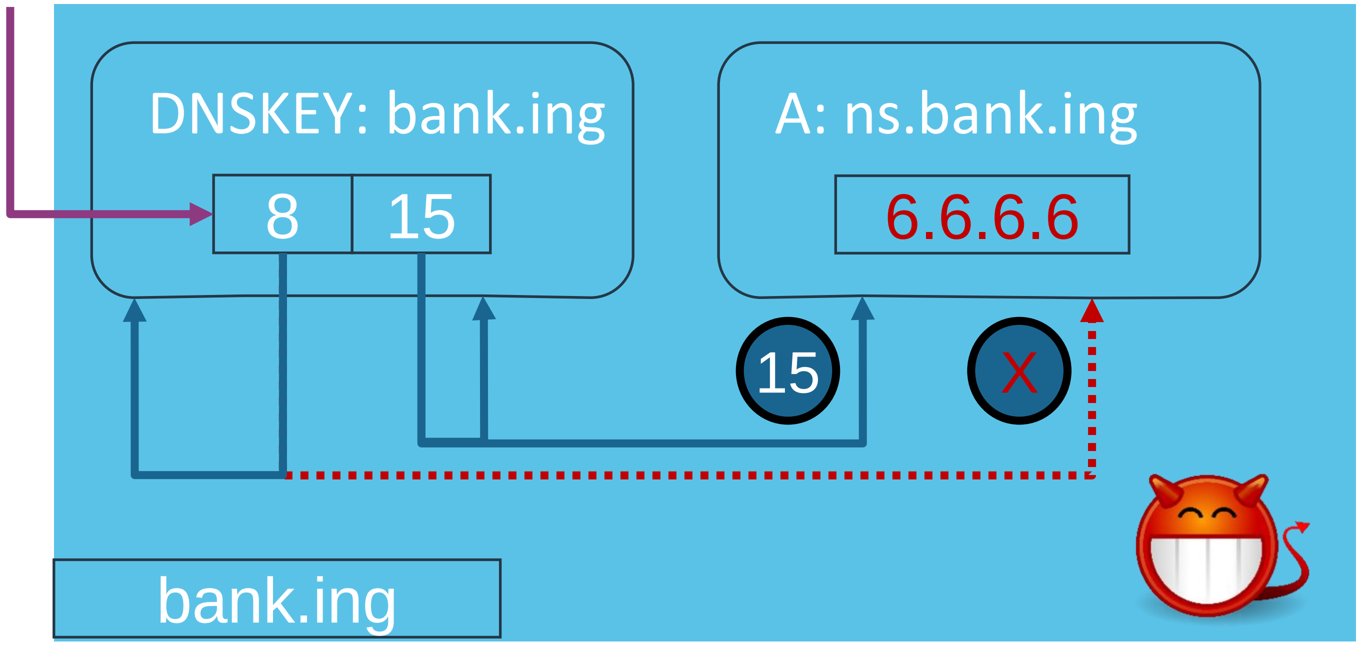
Preconditions

- the zone is signed with two different algorithms
  - one supported by the resolver
  - one unsupported (here: 15)
  - DS records at the parent at least for the supported one

Note

- DNSKEYs of both algorithms and their RRSIGs are present

Stripping Off Supported RRSIGs



Procedure

- ➢ the attacker drops the supported RRSIG records
  - ➢ from DNS messages to the resolver
  - ➢ leaving only unsupported algorithms

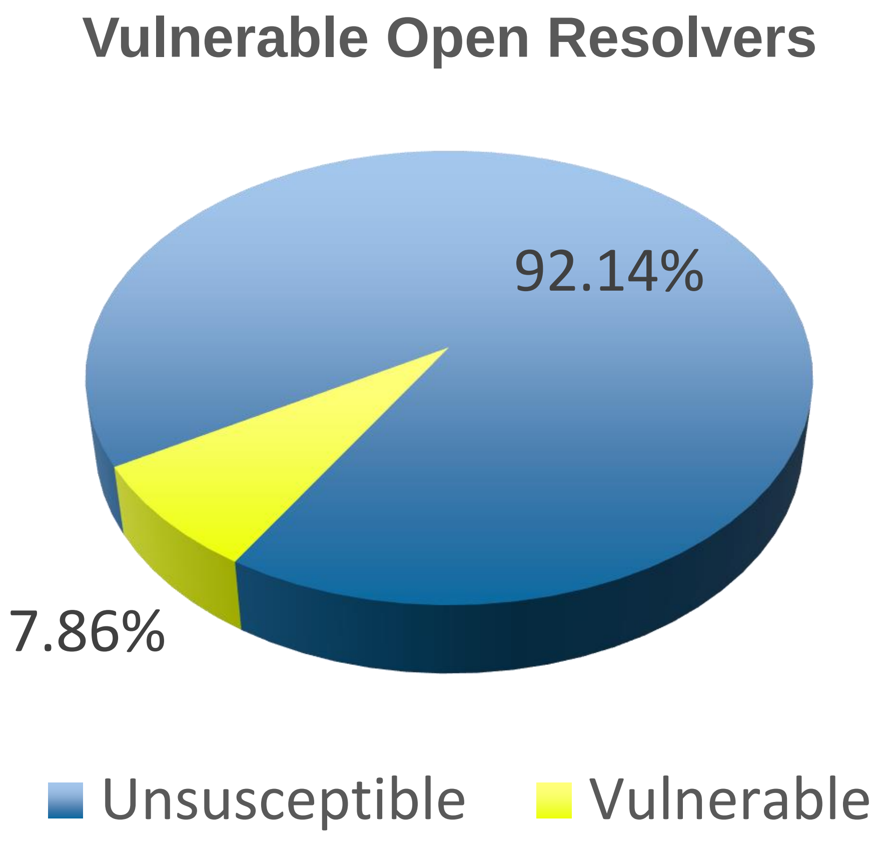
Note

- ➢ The RRSIG of Algorithm 8 should be present.

Stripping Off Supported RRSIGs

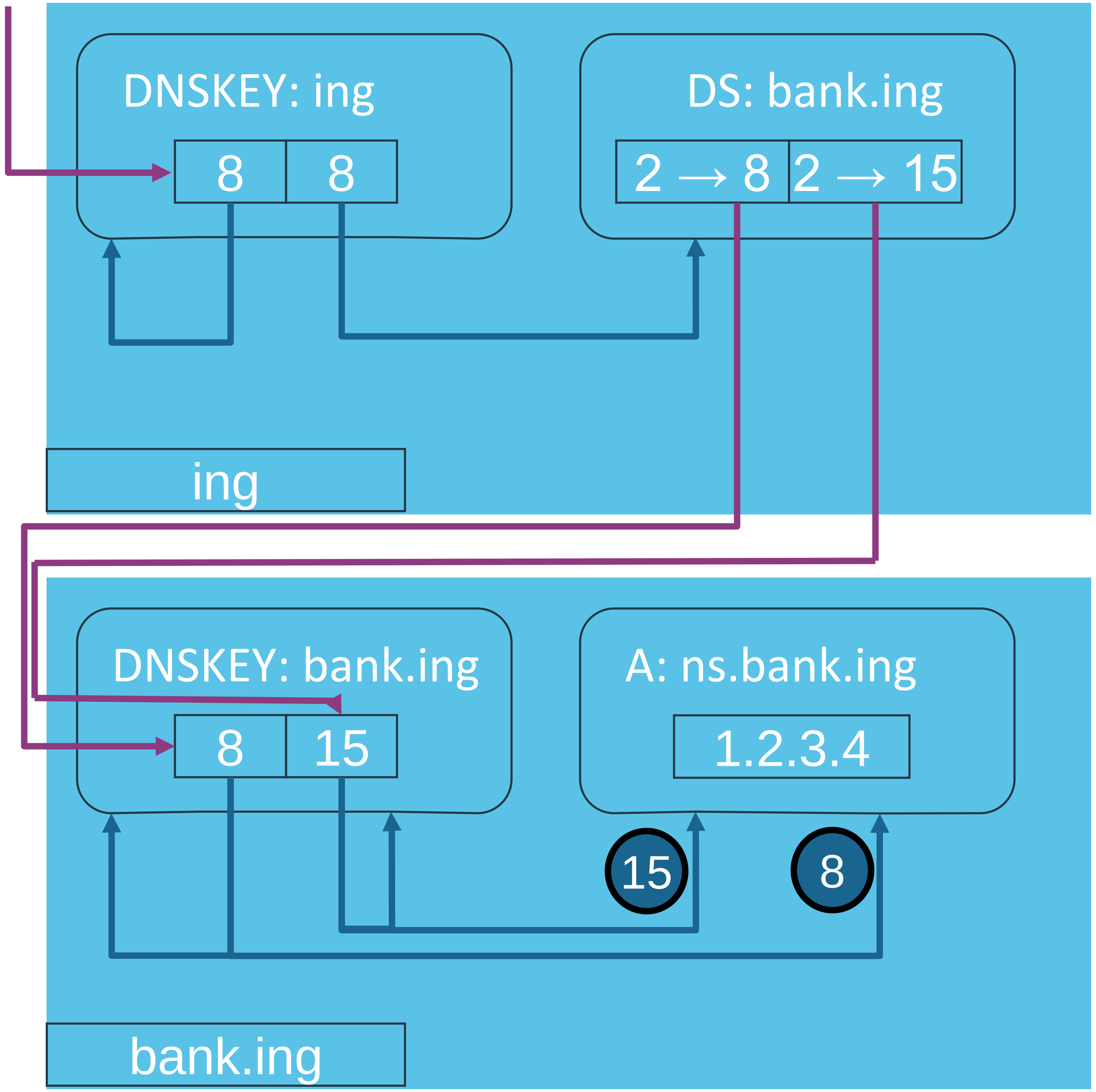
Vulnerable Resolvers

- none of the resolvers in our lab
- 2 Popular Resolver Services: Cloudflare and Google


Vulnerable Open Resolvers

Stripping off Supported DNSKEYs

Stripping off Supported DNSKEYs



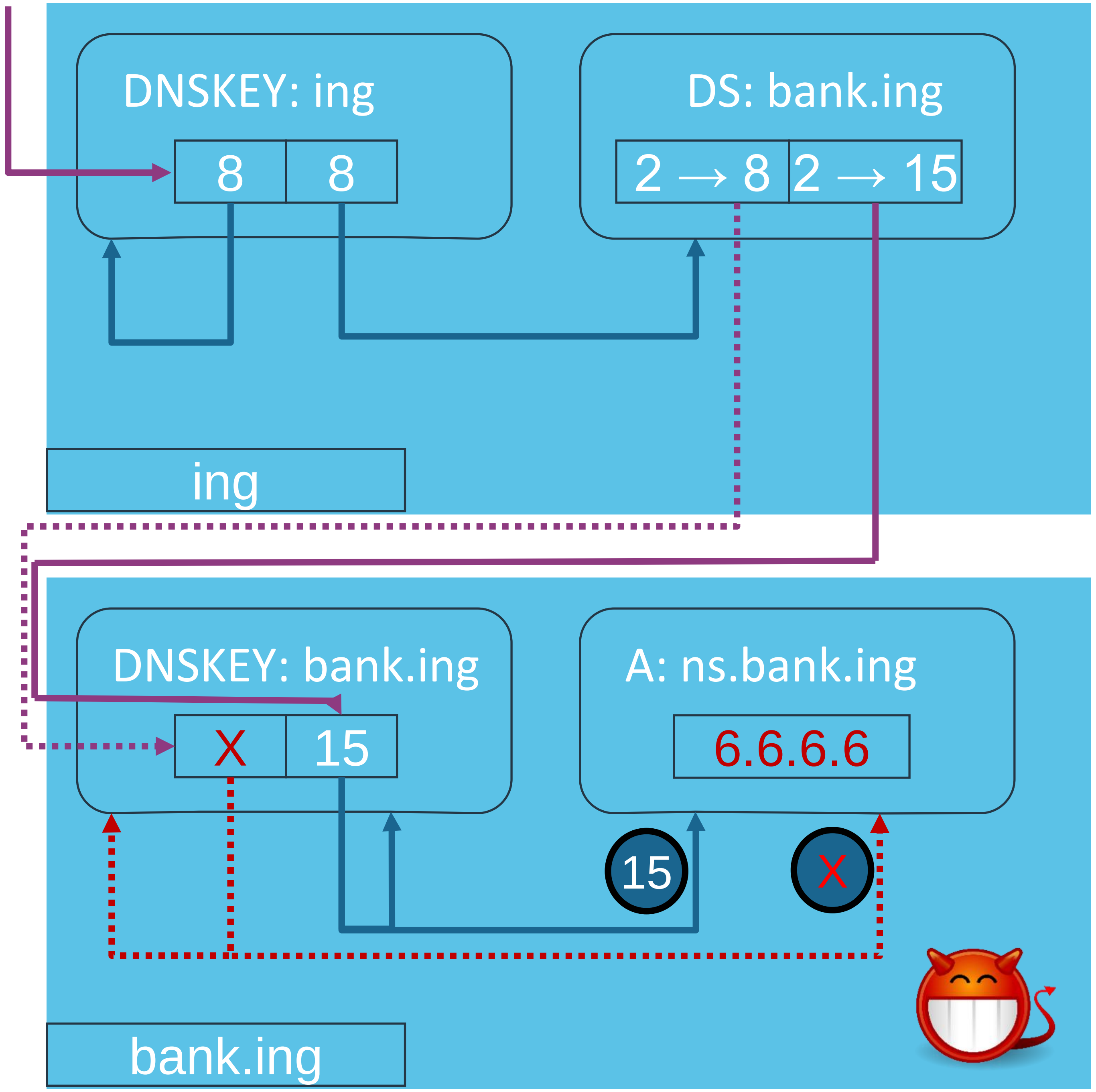
Preconditions

- zone is signed with two different algorithms
  - one supported by the resolver
  - one unsupported (here: 15)
- (at least) one DNSKEY for each
- DS records for both at the parent

Note

- DNSKEYs of both algorithms and their RRSIGs are prsent

Stripping off Supported DNSKEYs



Procedure

- ➢ the attacker drops the supported DNSKEY
  - ➢ and all its signatures
  - ➢ from any DNS messages to the resolver
  - ➢ leaving only unsupported algorithms

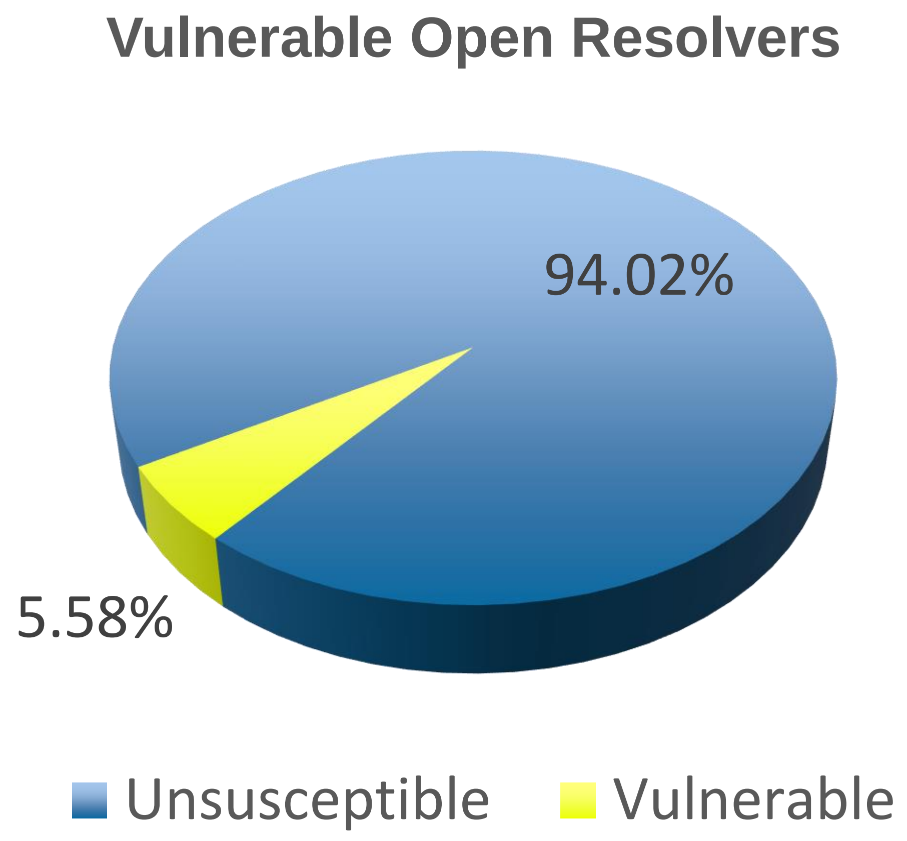
Note

- ➢ DNSKEY for algorithm 8 should be present
- ➢ RRSIGs for algorithm 8 should be present
  - ➢ stripping off the signatures not strictly necessary

Stripping off Supported DNSKEYs

Vulnerable Resolvers

- 1 Popular Open Resolver (OpenDNS)
- Windows Server Recursive DNS (all tested versions)



Review: Attacks to Break Security

Countermeasures

- when considering algorithms, resolvers should decide “insecure” solely based on the DS records
  - insist on presence of a least one supported algorithm according to specification

supported DS → supported DNSKEY → supported RRSIGs on all zone data

Agenda

- ➢ DNS(SEC) Refresher
- ➢ DNSSEC Downgrade Attacks
  - ➢ Attacks to Weaken Security
  - ➢ Attacks to Break Security
- ➢ **Recommendations**

Recommendations

Resolver Operators and Developers

- require strongest present DS digest to work for validation
- only consider DS records for deciding to mark data *insecure* because of unsupported algorithms

Zone Operators

- move away from SHA-1 ASAP
- adding additional signatures of stronger algorithms does not increase security
  - can even level security, if those are not supported by vulnerable resolvers

Thank you for your attention!

Contact: elias.heftrig@sit.fraunhofer.de