ATHENE
National Research Center for Applied Cybersecurity

Downgrading DNSSEC
How to Exploit Crypto Agility for Hijacking Signed Zones

Elias Heftrig, Haya Shulman, Michael Waidner

Contributions Summary

- Analysis of conditions under which DNS resolvers can be forced to skip DNSSEC validation
  - Vulnerabilities affecting major DNS providers and many dependent systems on the Internet
- Development of DNS cache poisoning attacks utilizing the attack vectors
- Evaluation of the DNSSEC ecosystem on the Internet
- Exploration of factors in the specification that promote the vulnerabilities

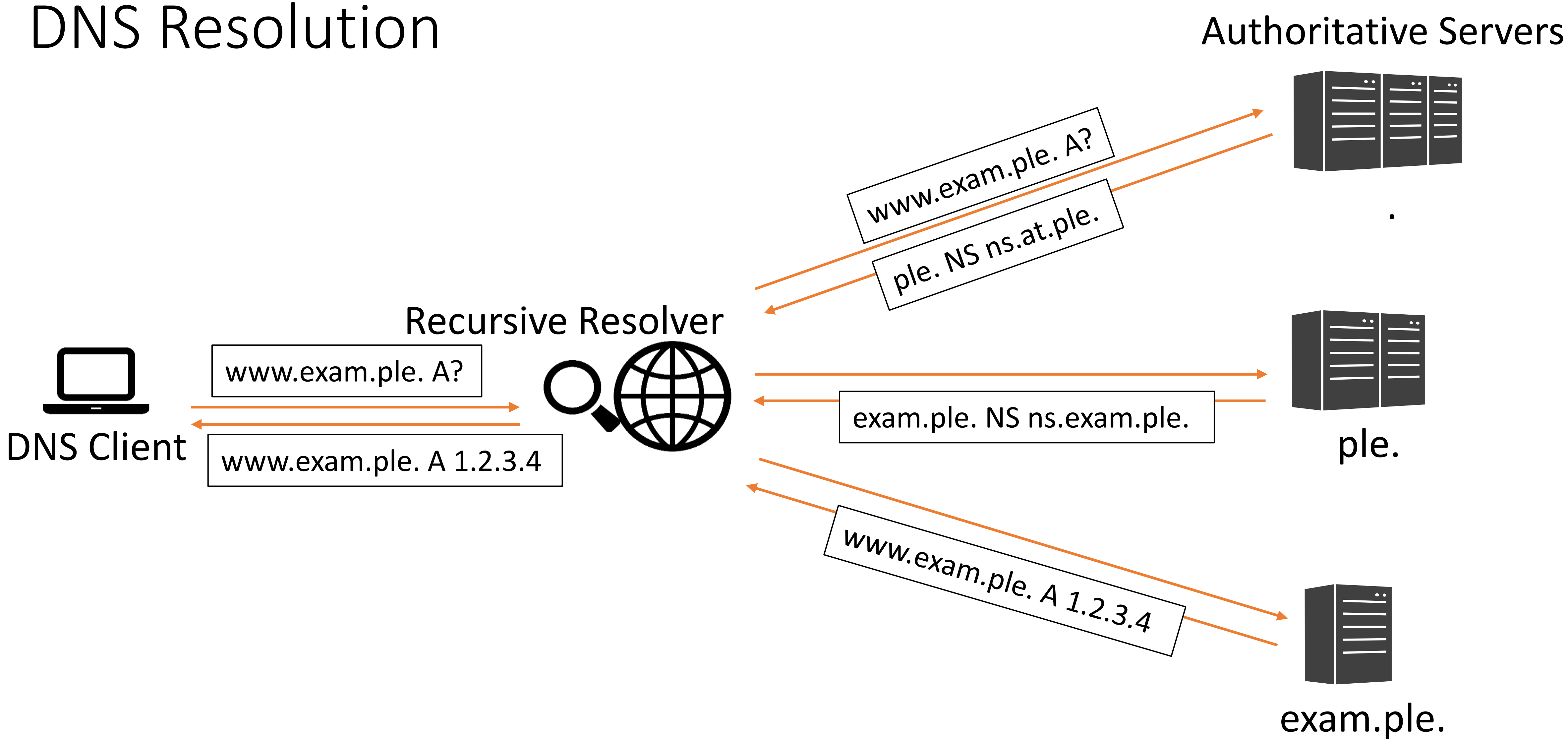
Agenda

- DNS(SEC) Overview
- Downgrading DNSSEC
- Specification Analysis
- Conclusion

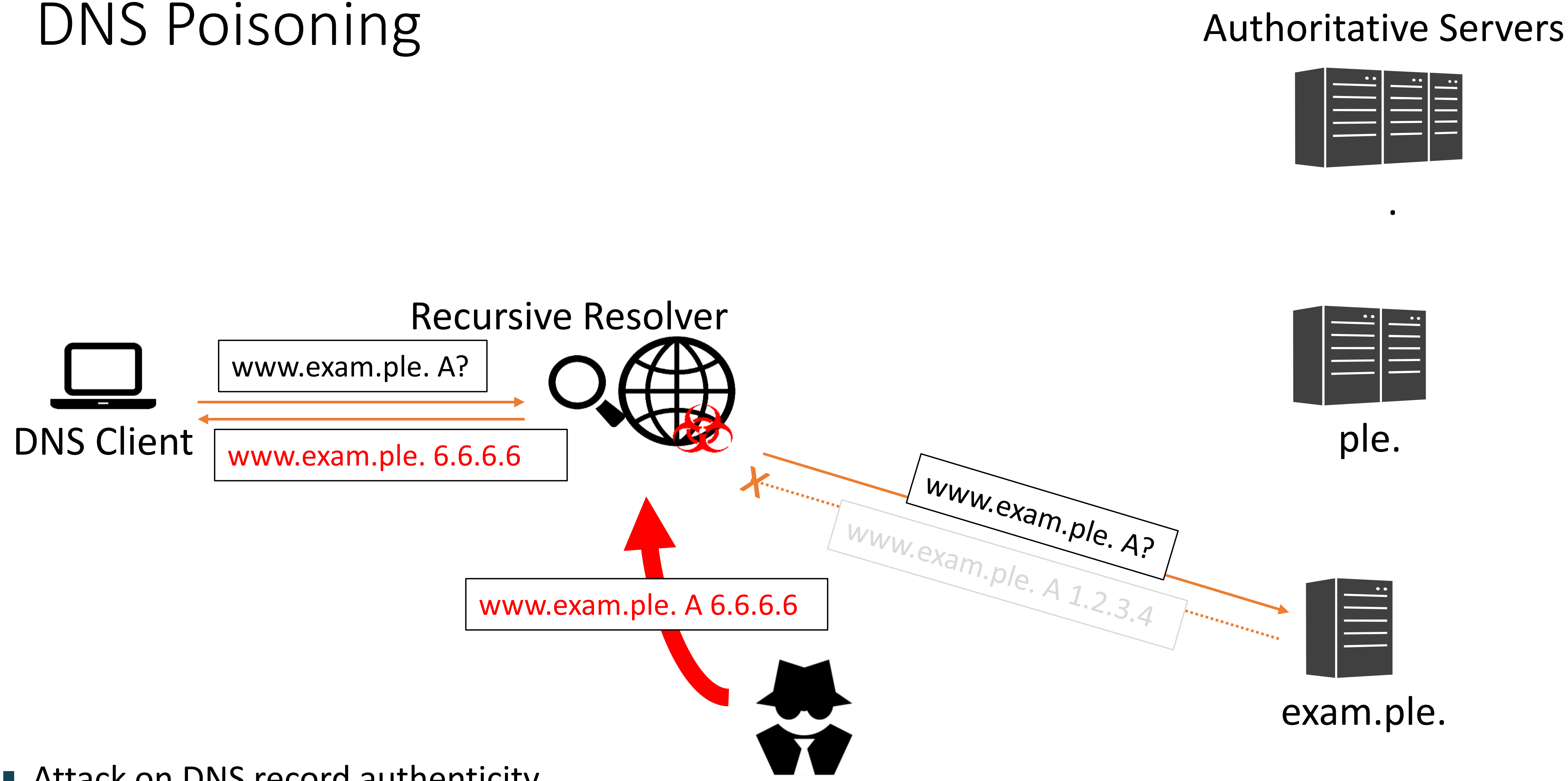
Agenda

- **DNS(SEC) Overview**
- Downgrading DNSSEC
- Specification Analysis
- Conclusion

DNS Resolution

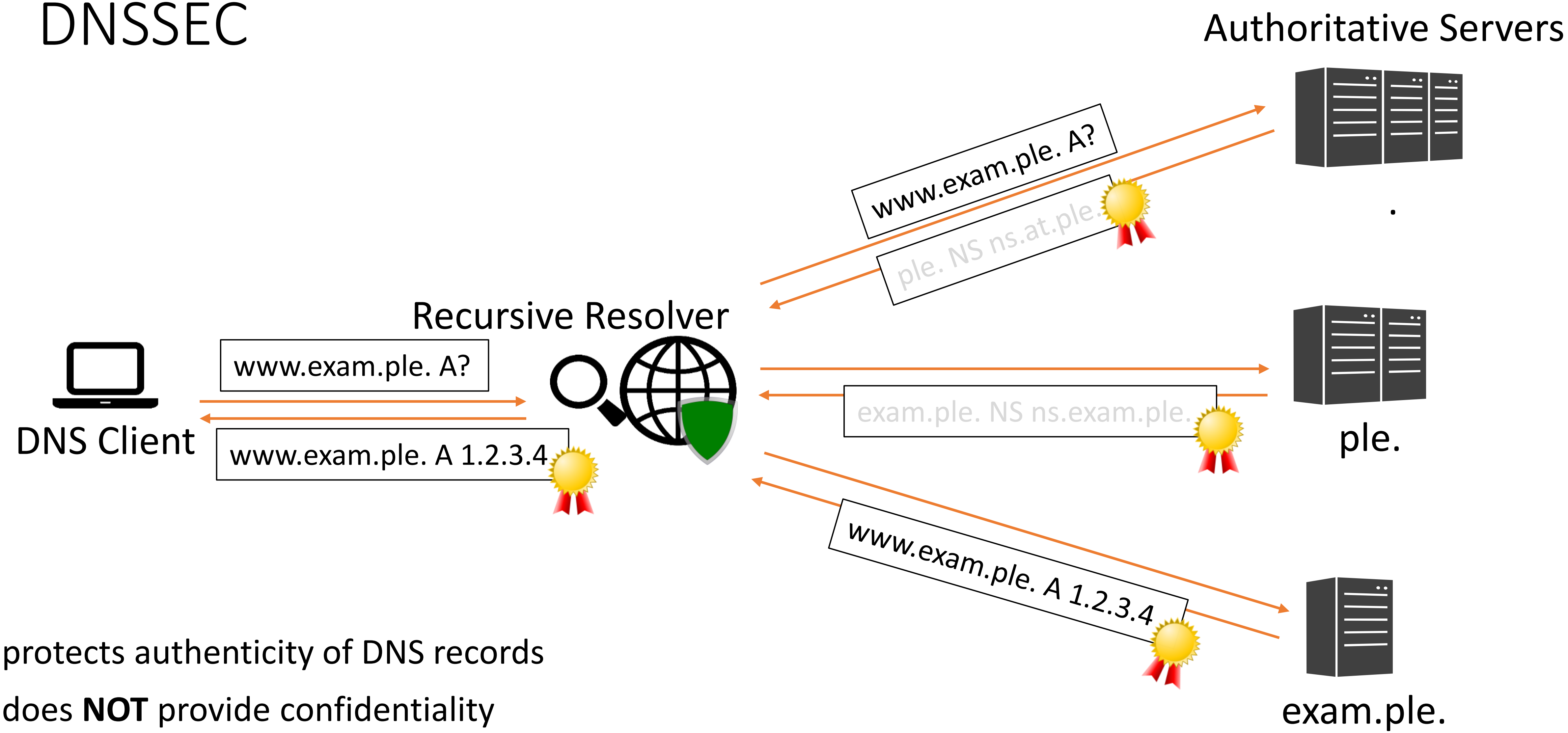


DNS Poisoning



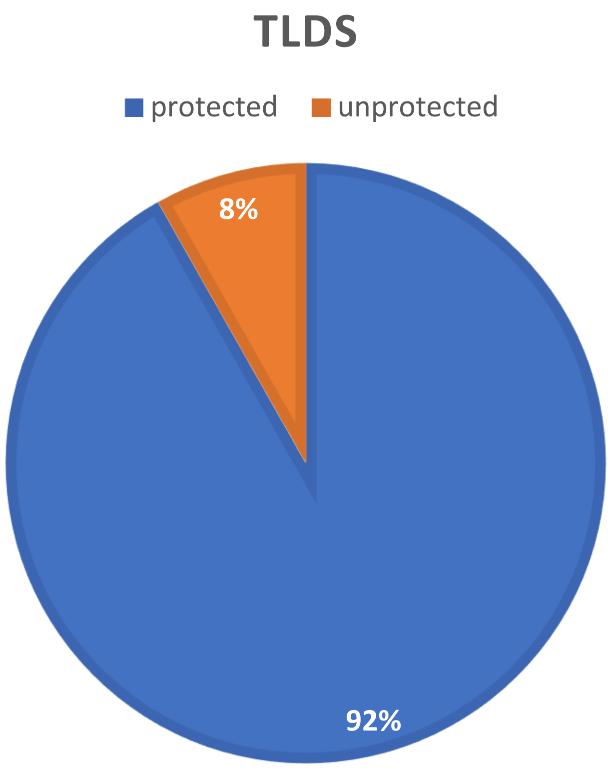
- Attack on DNS record authenticity

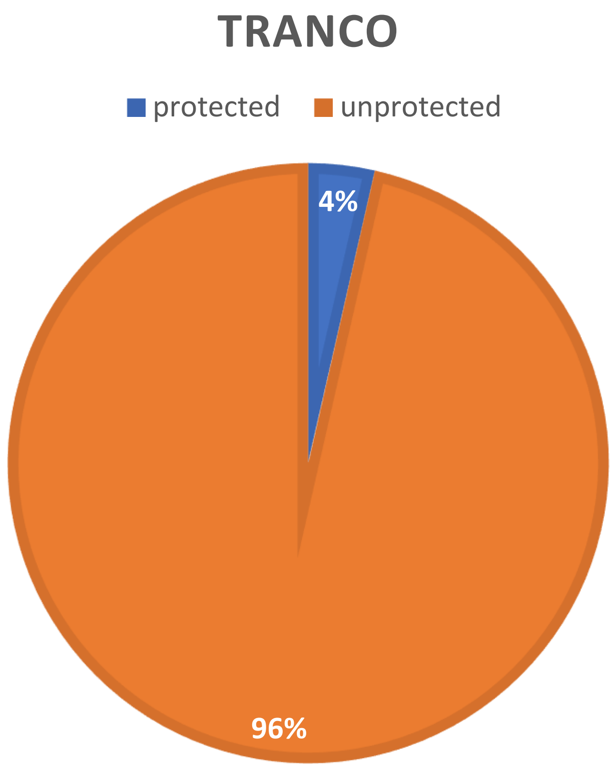
DNSSEC



- protects authenticity of DNS records
- does **NOT** provide confidentiality
- uses a PKI aligned with the DNS for signature validation

Measurements Setup

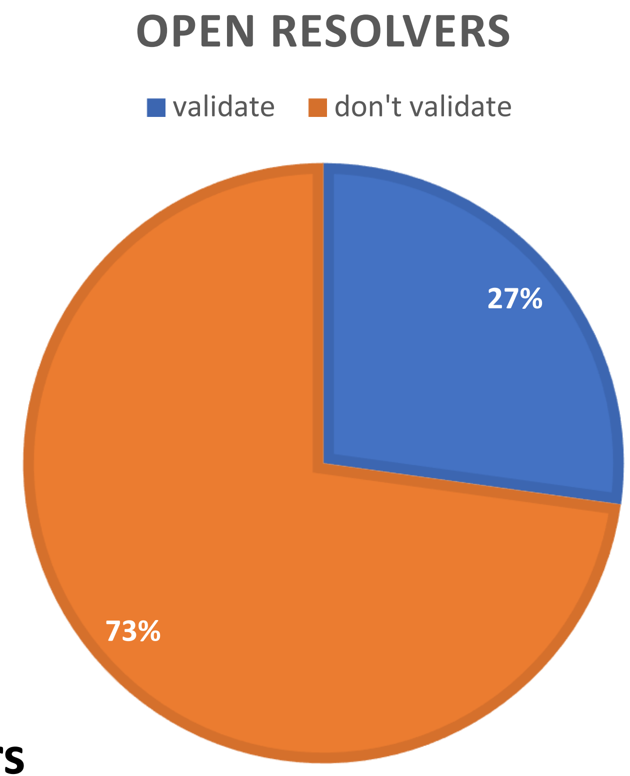


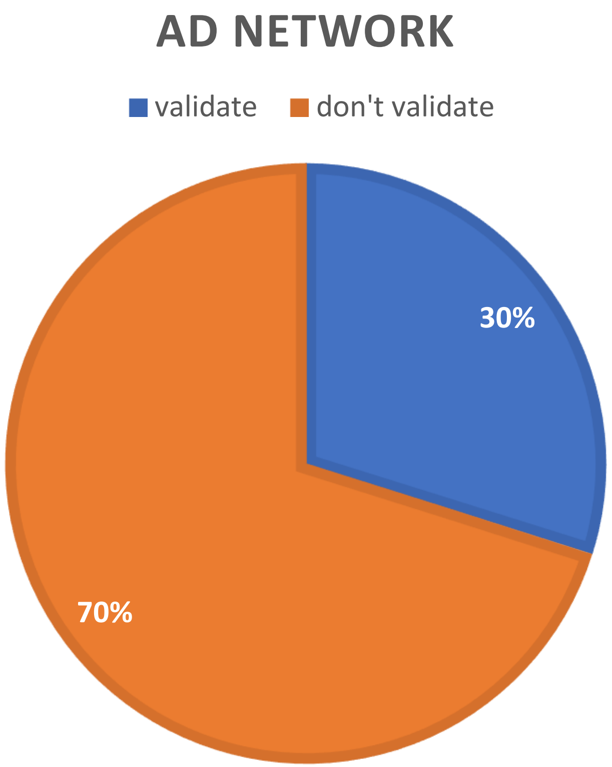


Domains

- All Top-level Domains (TLDs) and Tranco Top 1M
- “protected” := signed and linked to the public chain of trust

Measurements Setup





Resolvers

- 9 Validating Resolvers in the Lab (4 popular Linux-hosted, 5 Windows Server Flavors)
- 8 Popular public validating resolver Services (Cloudflare 1.1.1.1, Google Public DNS, …)
- 8,829 Open resolvers sampled from portscans on the IPv4 Address space
- Resolvers used by 8,977 Web clients distributed over the globe, measured using an ad network

DNSSEC Algorithm Agility

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)

~ newer

RSA

ECDSA

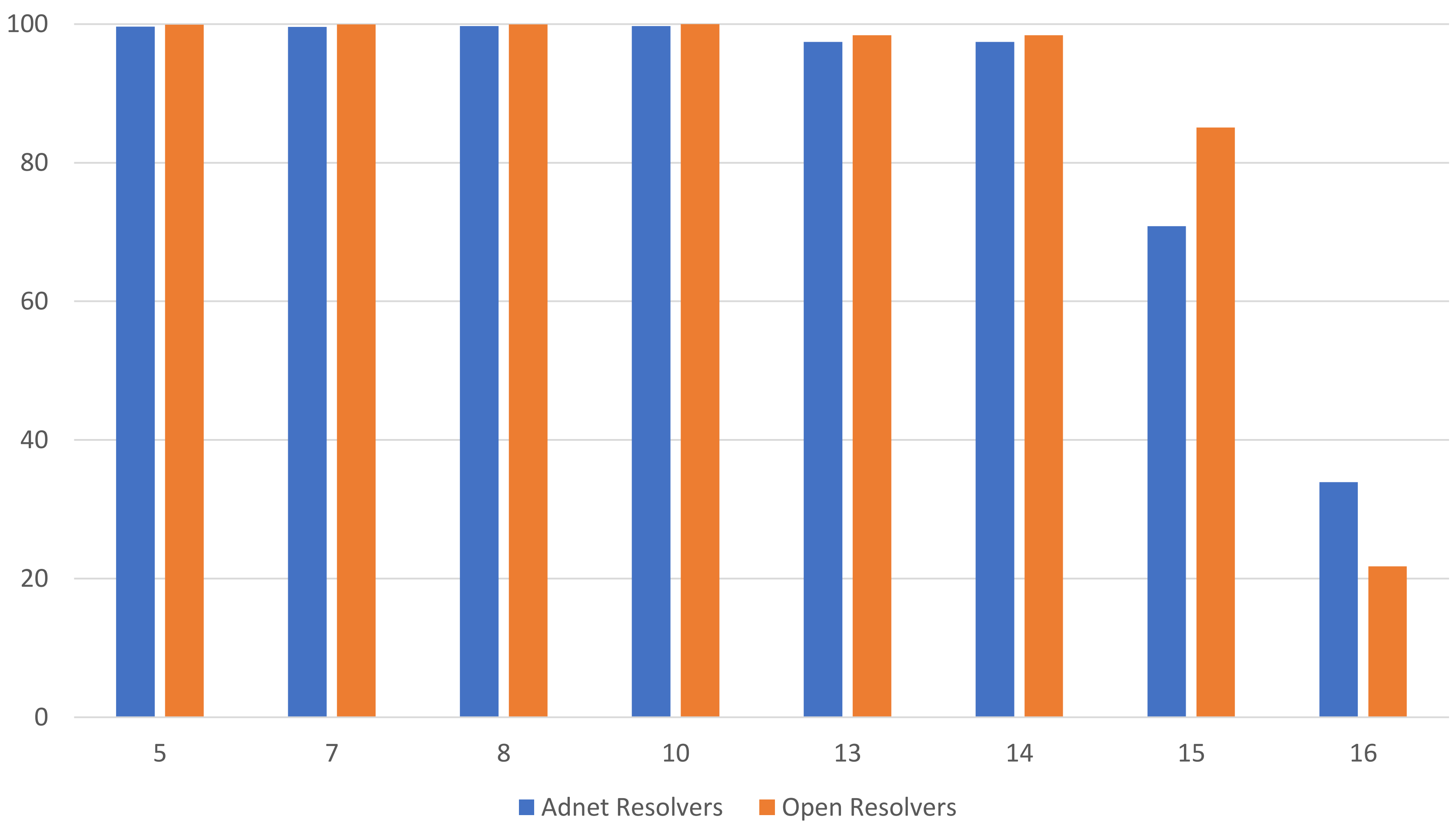
EdDSA

phasing out

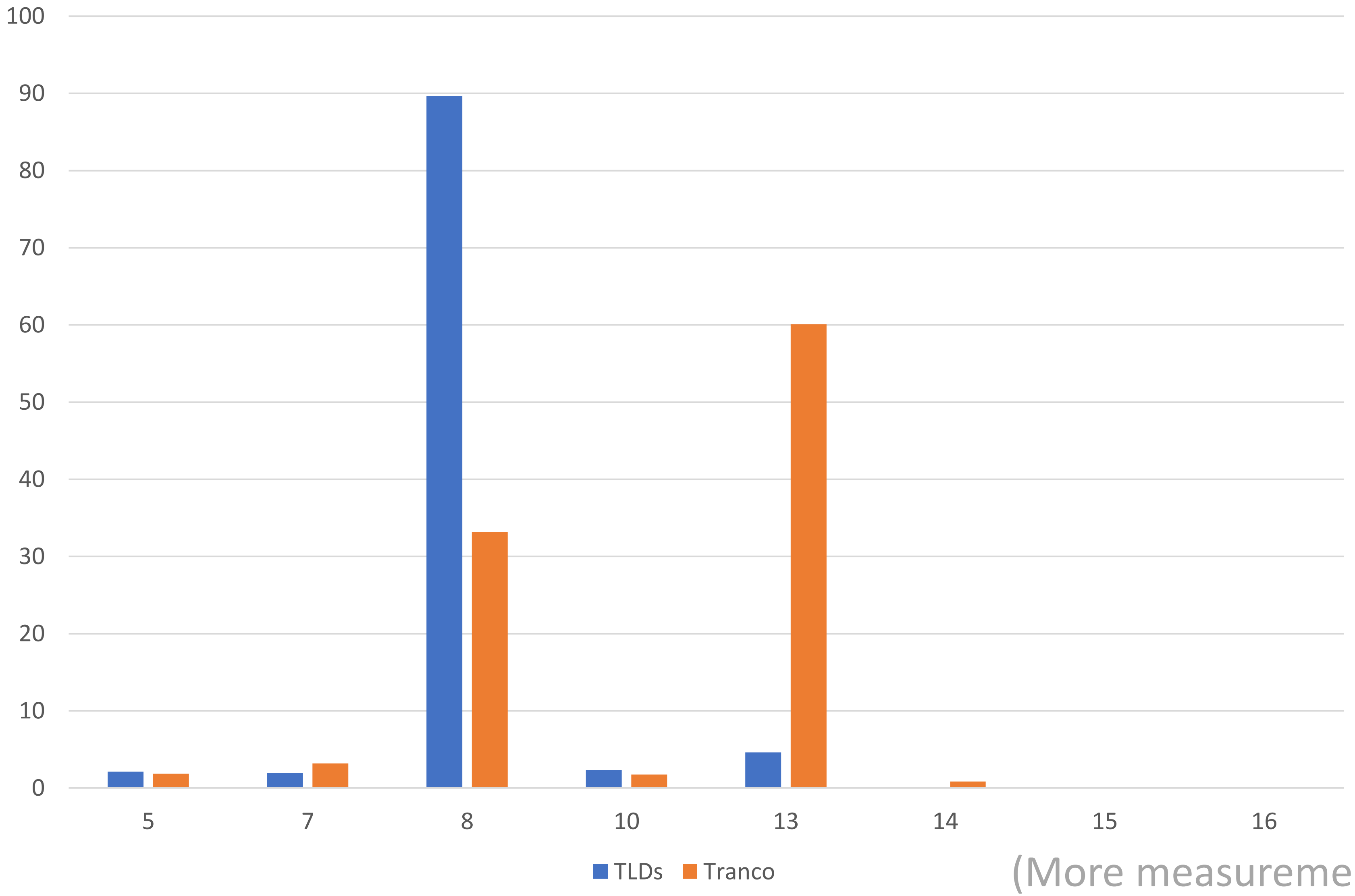
phasing in

- Rules for Algorithm Support in DNSSEC Software, acc. [RFC8624]
- No negotiation included

DNSSEC Algorithm Support in Resolvers



DNSSEC Algorithm Usage in Domains



(More measurements in the paper)

Agenda

- DNS(SEC) Overview
- **Downgrading DNSSEC**
- Specification Analysis
- Conclusion

Attack Model

Attack Setup

- Attacker Model: On-path Attacker (~ Threat Model of DNSSEC)
- Positioned between the resolver and the authoritative name server

Attack Ingredients

- Disable DNSSEC validation, by manipulating the chain of trust
- Inject Poisonous Payload

DNSSEC Manipulation Methodologies

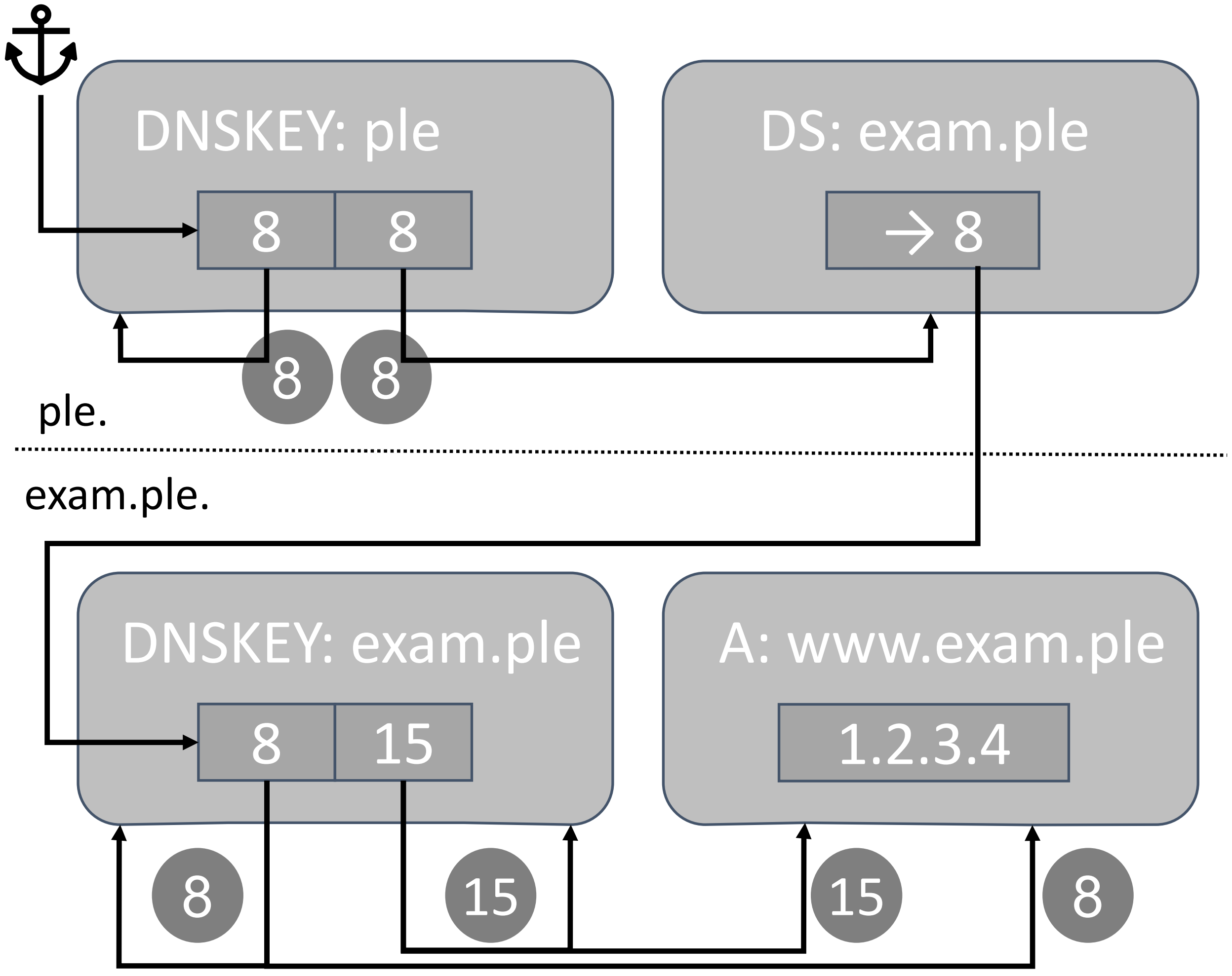
Attack vectors

(a) Strip the RRSIG over the target DNS RRset
(b) Strip the RRSIG over the DNSKEY RRset
(c) Strip the DNSKEY RRset
(d) Rewrite the AlgorithmNumber field in the RRSIG

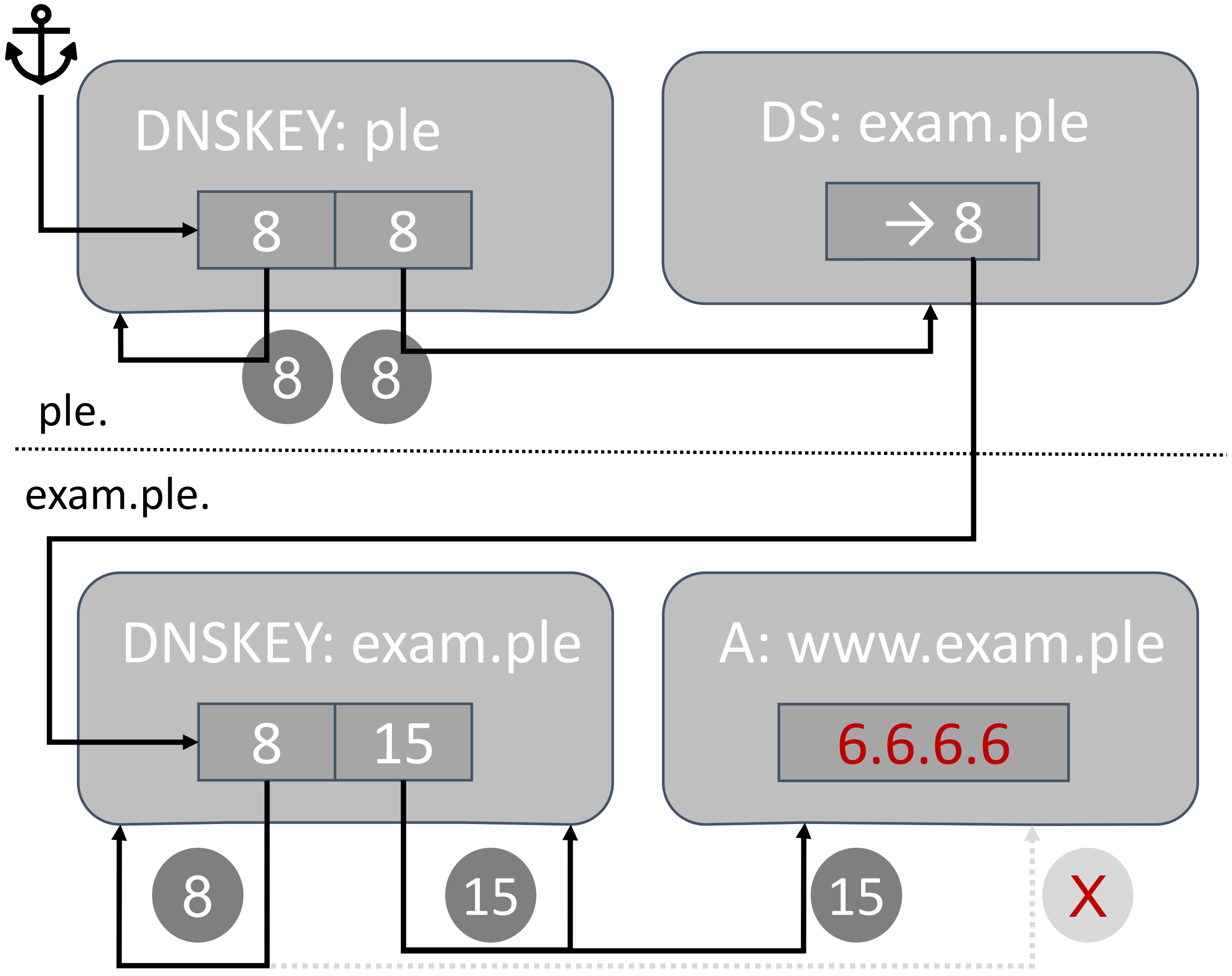
Applied to

- Single-algorithm domains (99.14% of protected Tranco Top1M)
- Dual-algorithm domains
  - one supported and one unsupported algorithm
  - Goal of (a)-(c): forcing the resolver along an unsupported validation path

(a) Stripping the RRSIG over the target RRSet in a Dual-Algorithm Zone



- Before



- After

Vulnerability Evaluations

Vulnerable Resolvers in the Lab

- Windows Server: (b) and (c)
- All tested platform versions

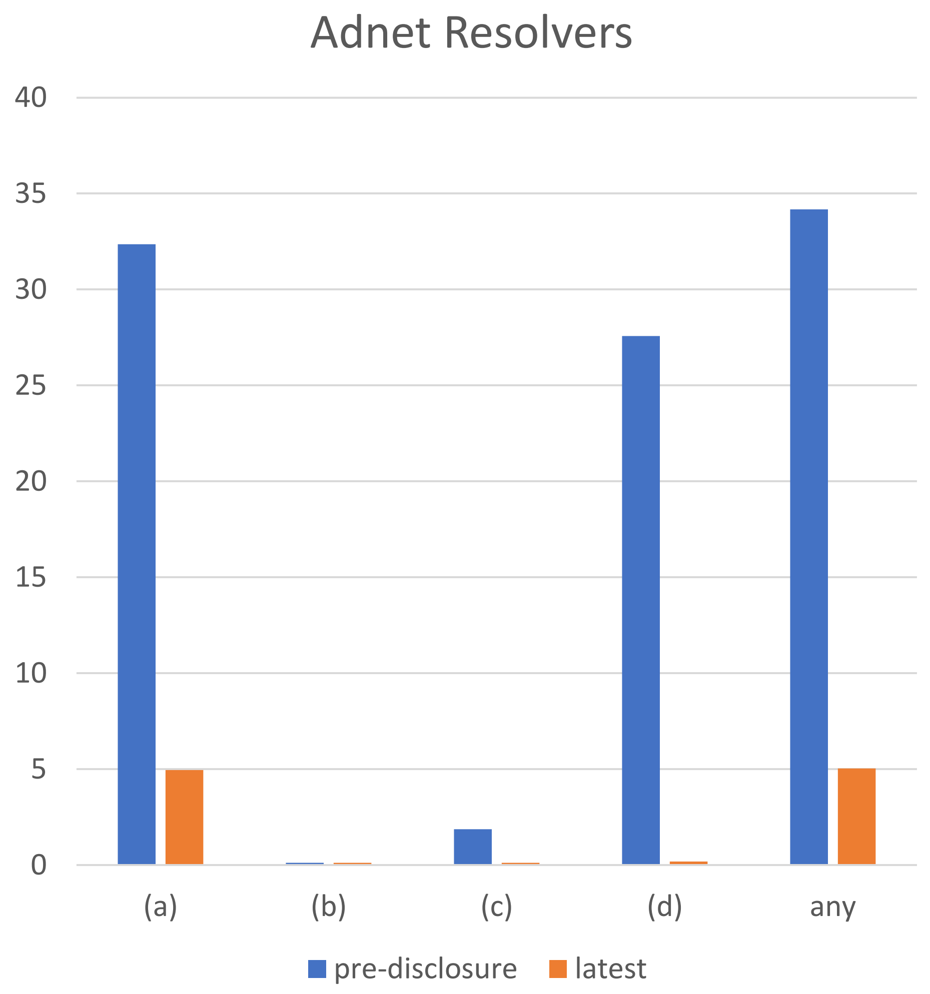
Vulnerable Popular Open Resolver Services

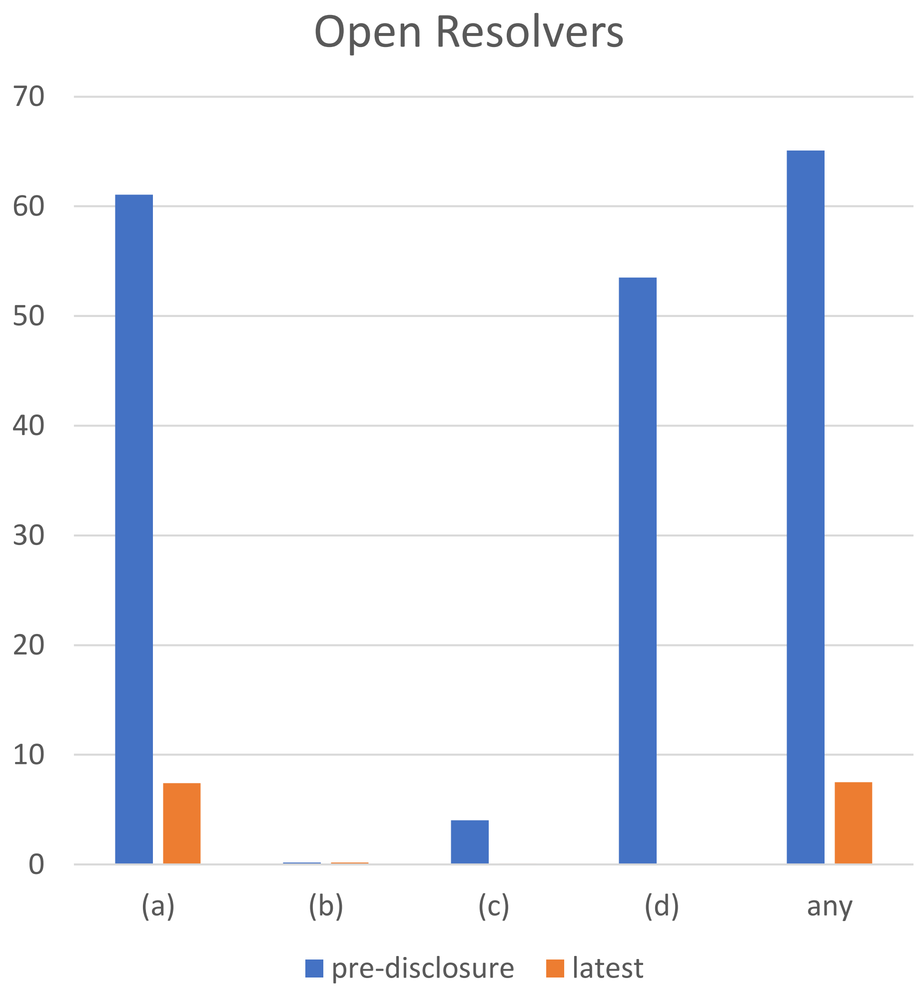
- Google: (a) and (d)
- Cloudflare: (a)
- OpenDNS: (c)

Generally

- Attack vectors (a) – (c) found effective on dual-algorithm domains only

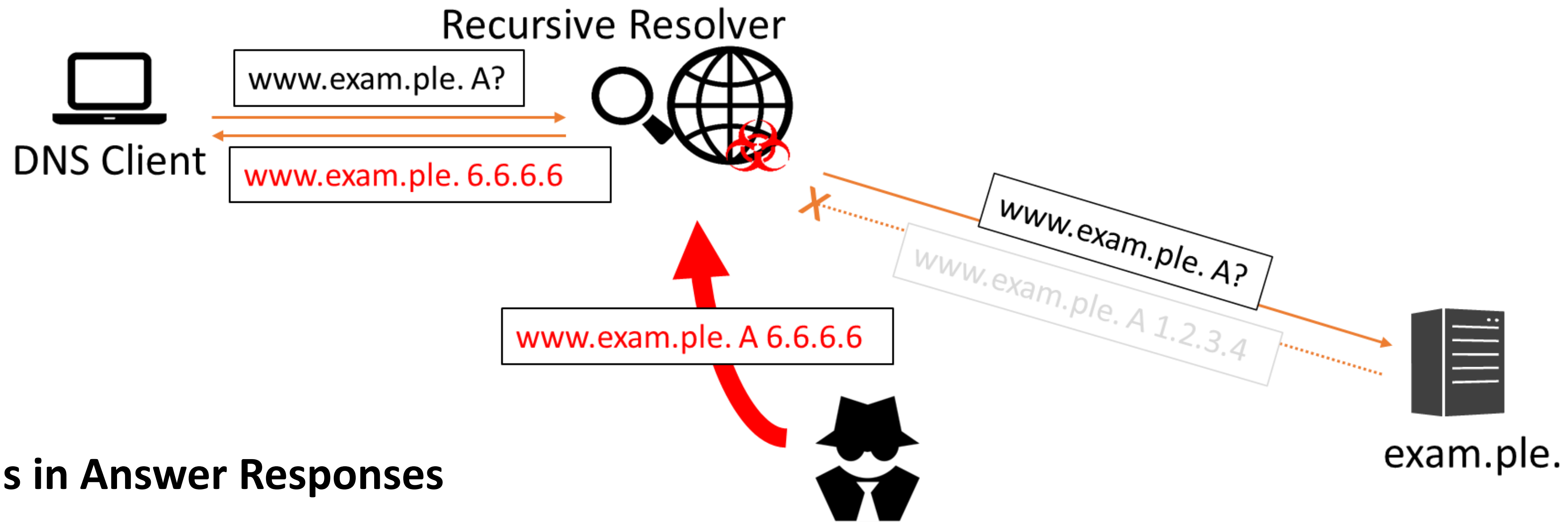
Vulnerability Evaluations





(mind the scales)

DNS Cache Poisoning Methodologies



Manipulating Records in Answer Responses

- Attacker simply injects a poisonous answer record

Hijacking a Secure Domain

- Attacker manipulates answer responses for an attacker-triggered authoritative NS-type request
- Victim resolver will send follow-up requests directly to attacker

DNS Cache Poisoning Methodologies

Hijacking Secure Delegation

- Attacker injects DS records for attacker-owned DNSKEY
- To take over the DNSSEC of the domain

Hijacking Secure Delegation

```
+ before +
IN DS 29449 13 2 f34135...eecc
IN DS 29449 13 4 8e1ec0.....180f
IN RRSIG DS 8 ...
IN RRSIG DS 16 ...

+ after +
IN DS 5342 13 2 bd638a.....4303
IN RRSIG DS 16 // invalid
```

Disabling Secure Delegation

- Attacker injects DS records not supported by the resolver
- To disable the DNSSEC of the domain

Disabling Secure Delegation

```
+ before +
IN DS 5342 8 2 f34135.....eecc
IN DS 5342 8 4 8e1ec0.....180f
IN RRSIG DS 13
IN RRSIG DS 16

+ after +
IN DS 5342 16 2 f34135.....eecc
IN DS 5342 16 4 8e1ec0.....180f
IN RRSIG DS 16 // invalid
```

Agenda

- DNS(SEC) Overview
- Downgrading DNSSEC
- **Specification Analysis**
- Conclusion

Exploited Attack Surface

AlgorithmNumber field in the RRSIG records effectively unprotected

- Used by the resolver before validating the signature
- Allows the attacker to manipulate the algorithm number

Algorithm presence out-of-scope of NSEC

- Leaves the attacker an opportunity to strip off specific DNSSEC records

Requirements on Algorithm Presence

One Core RFC mandates DNSSEC Record Presence for Signature Algorithms in Zones

DS ⟶ DNSKEY ⟶ RRSIGs on all zone data

- Was a step into the right direction
- But explicitly declared to not apply to resolvers by follow-up specification

Suggested Fix

- Require resolvers to insist on presence of a least one supported algorithm according to

  supported DS ⟶ supported DNSKEY ⟶ supported RRSIGs on all obtained zone data

- And send SERVFAIL if hurt

Overloaded Core Terminology

Validation States

- *Secure*, *Insecure*, *Bogus*, *Indeterminate* have differing definitions two of the core RFCs
- Noticed in follow-up specification but never reconciled
  - Even explicitly left open whether it should be reconciled at all ([RC8499] "DNS Terminology")
  - Or dependents just define their way out of it ([RFC7672])
- States declared important but miss clear specification of meaning and consequences
- Forces developers to settle for one or come up with their own interpretations

(further issues and explanations in the paper)

Agenda

- DNS(SEC) Overview
- Downgrading DNSSEC
- Specification Analysis
- **Conclusion**

Conclusions

- Cryptographic agility is an important feature for future-proofing DNSSEC
  - But also exposes to new attacks
- Specification needs to be balanced between implementation freedom and clear requirements
  - Because DNS developers are strongly incentivized to favor robustness over security
  - In this case, more of the latter would have prevented vulnerabilities

Thank you for your attention!