

DNSSEC

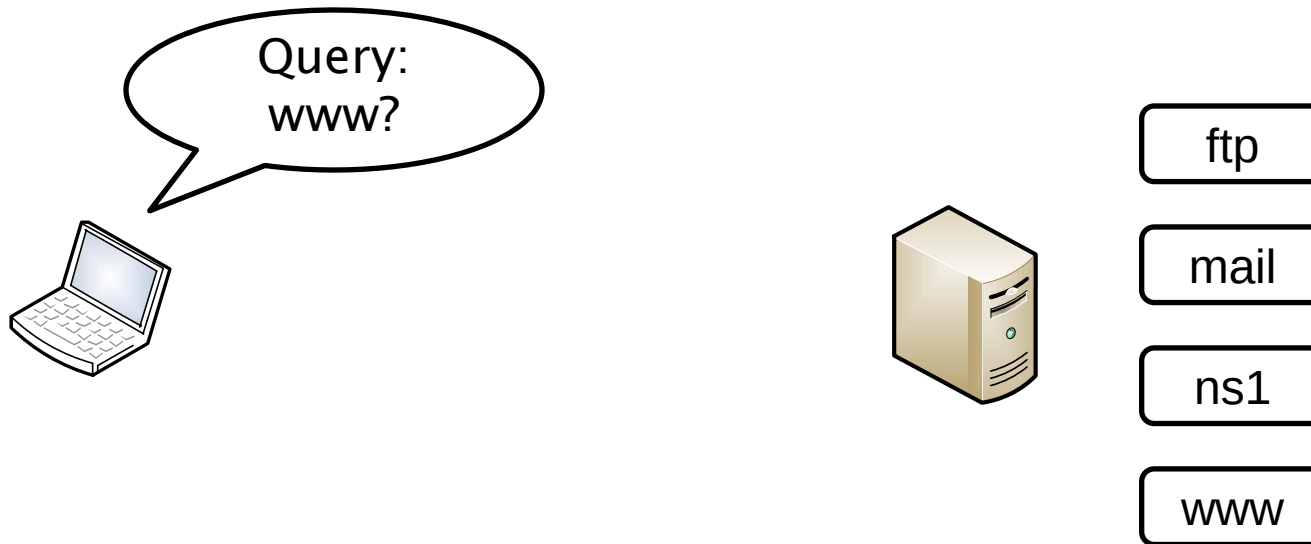
and the Hassle of Negative Responses

Matthäus Wander

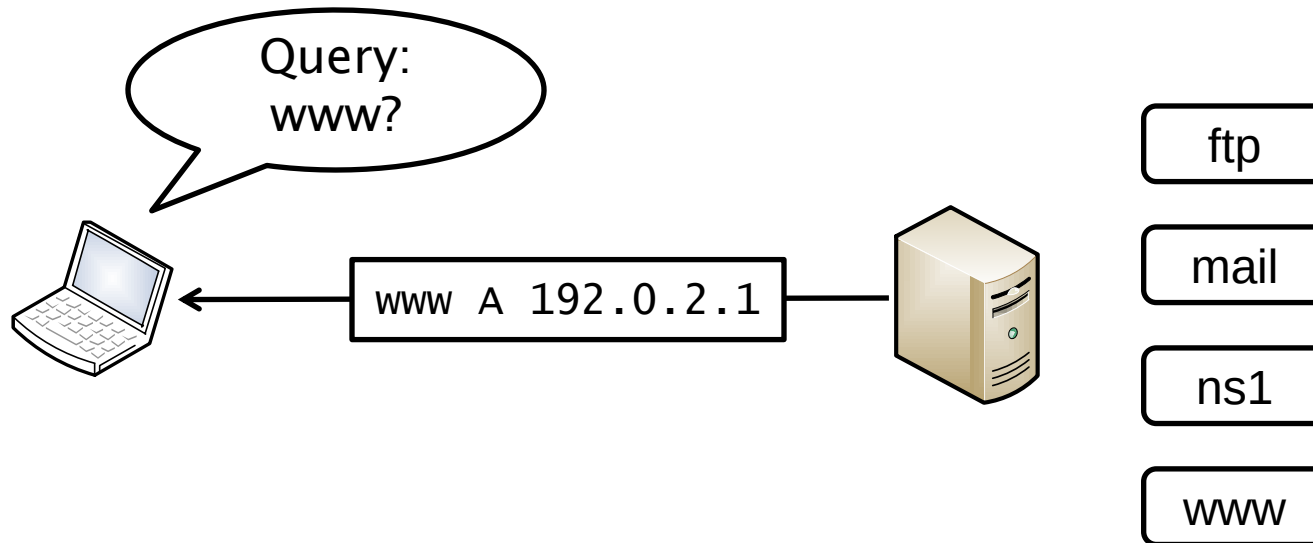
<matthaeus.wander@uni-due.de>

Erlangen, April 20, 2015

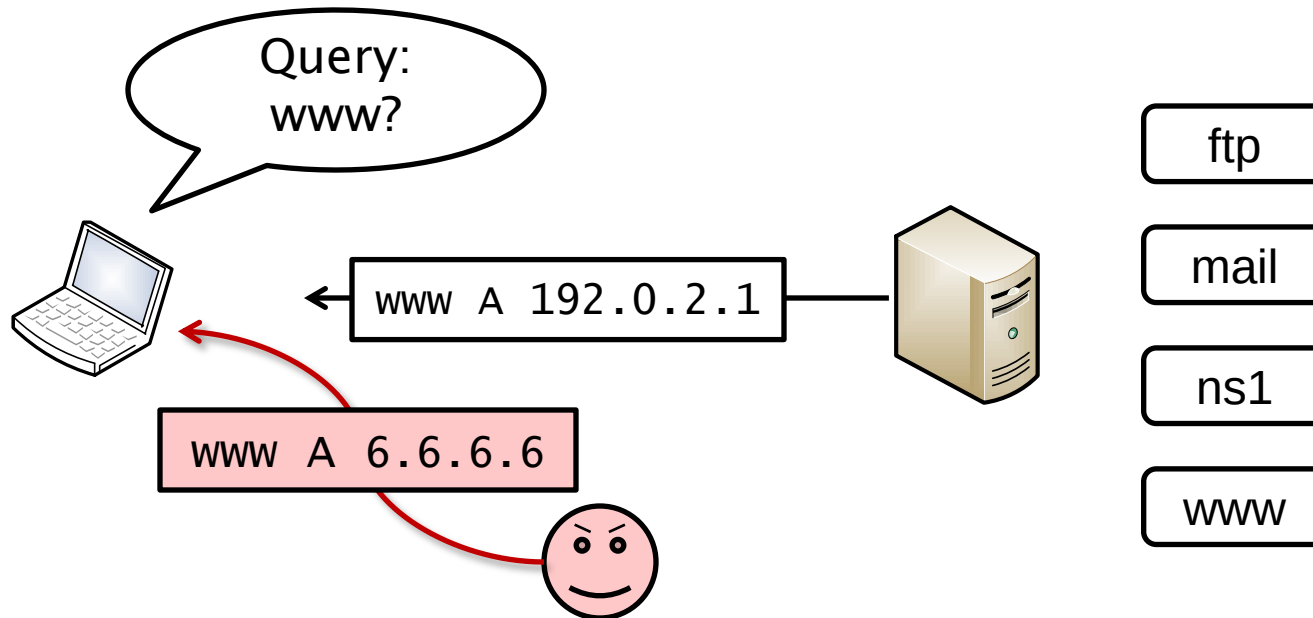
Security Goal of DNSSEC



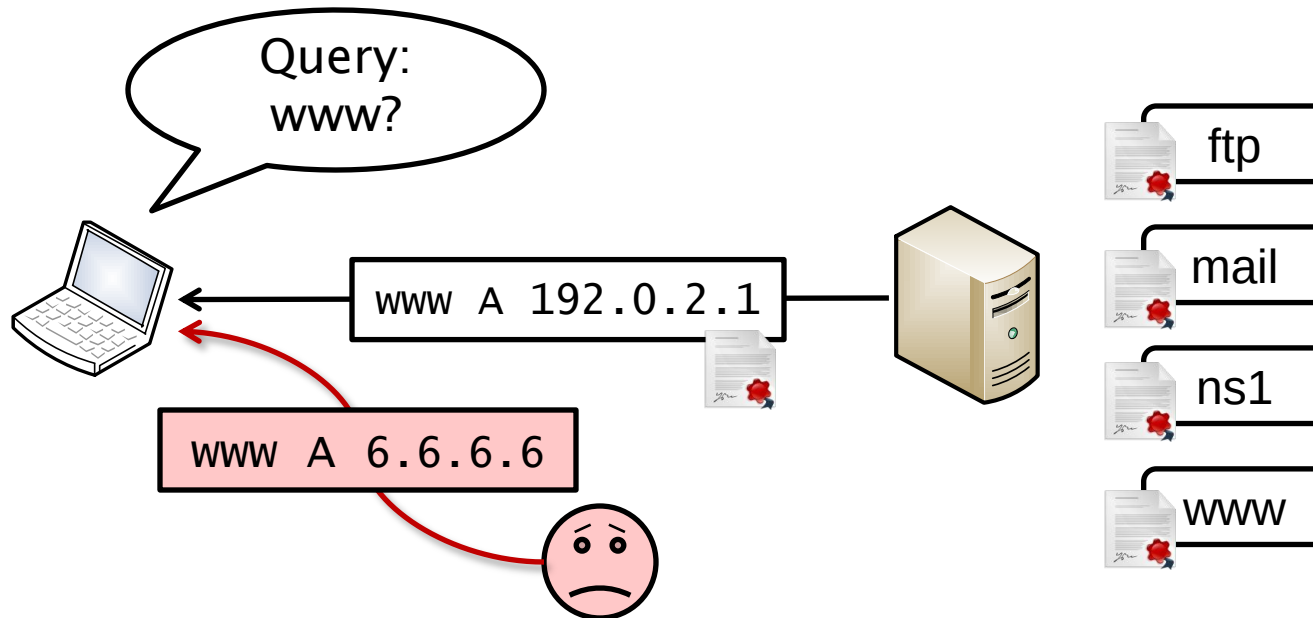
Security Goal of DNSSEC



Security Goal of DNSSEC

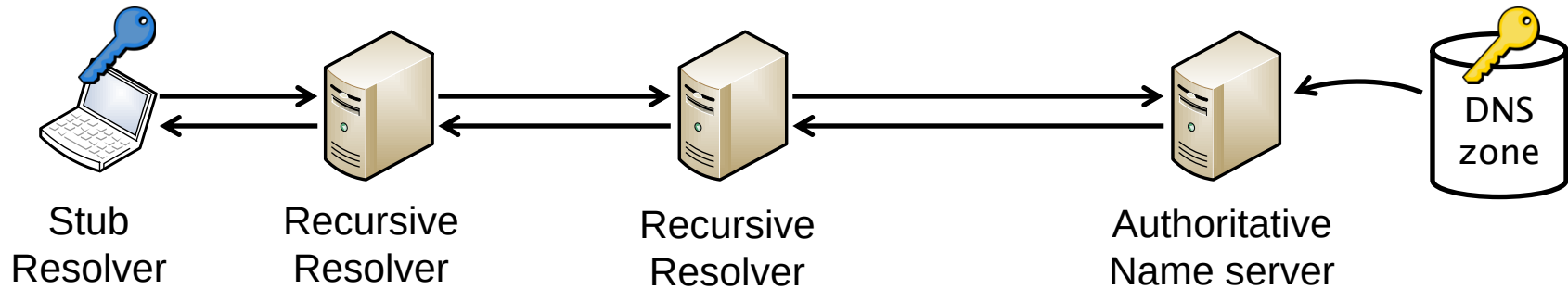


Security Goal of DNSSEC

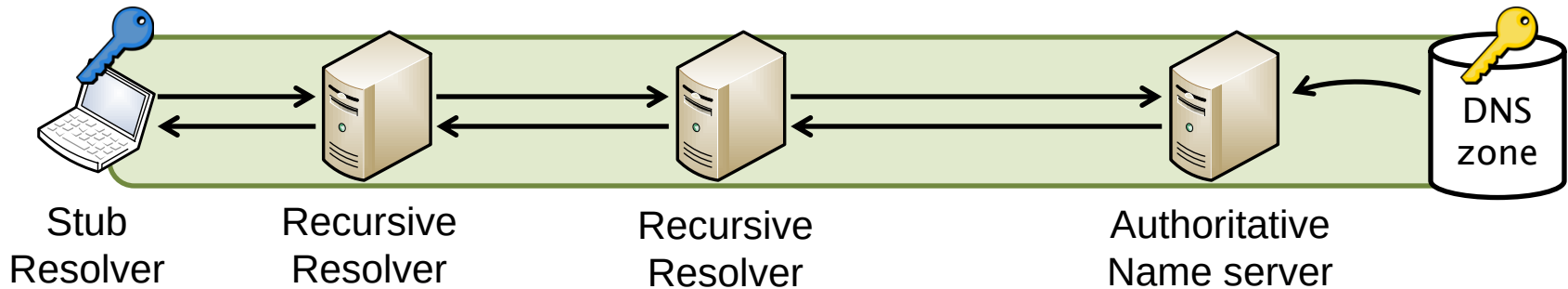


- Data integrity and authenticity
- Signatures over resource records (data sets)

End-to-End Security



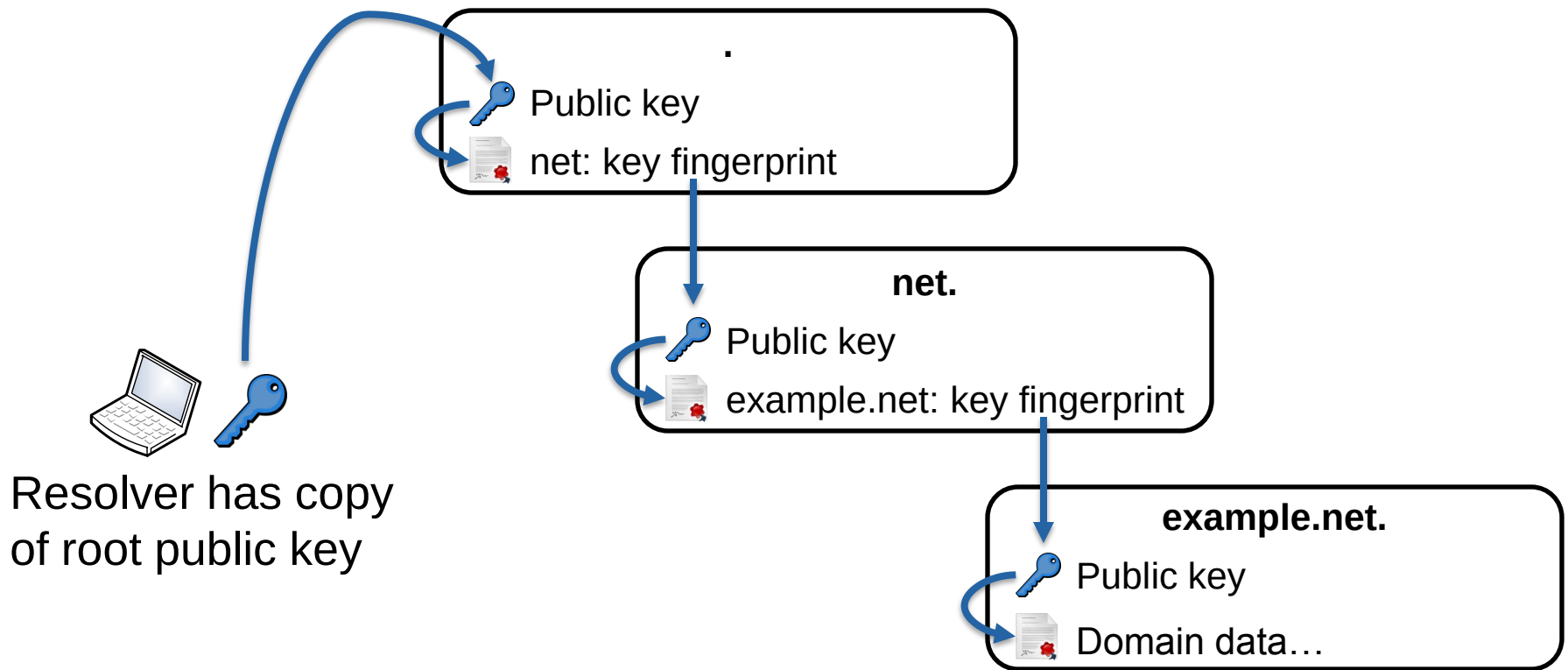
End-to-End Security



- End-to-end security principle
 - Between **validator** and **signer**
- Offline signing
 - Pre-generate all signatures

Public Key Distribution

- Public keys distributed in-band
- Authenticated by parent domain



RESEARCH WORK

NEGATIVE RESPONSES

ATTACKING NSEC3

Research Work

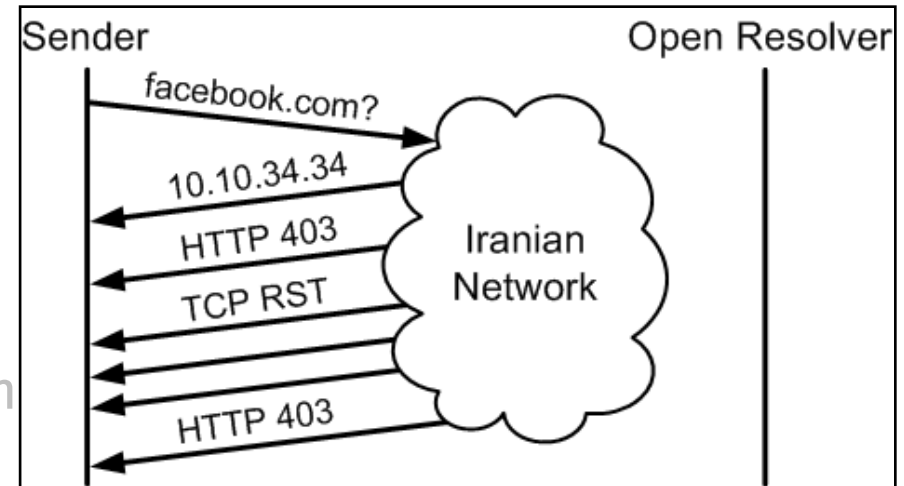
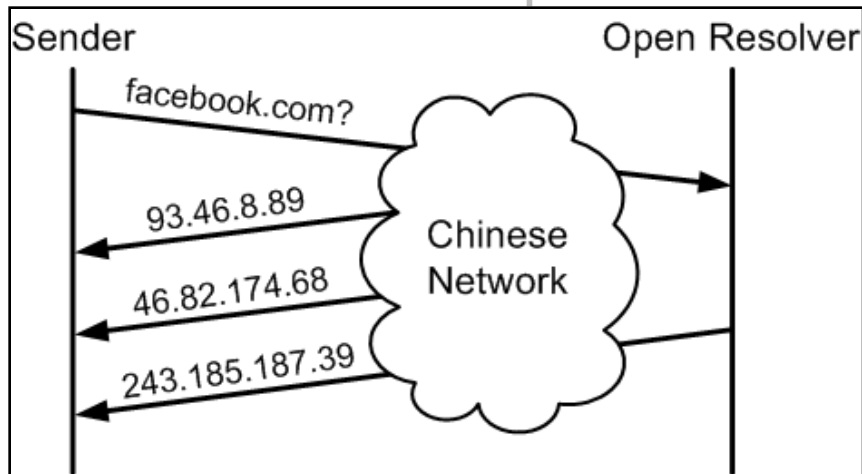
- Measurement analysis
 - DNS injection
 - DNSSEC adoption
 - Server-side: signed domains
 - Client-side: validators
- Attack methods on NSEC3 privacy goal

Research Work

- Measurement analysis

- DNS injection

- DNSSEC adoption



EC3 privacy goal

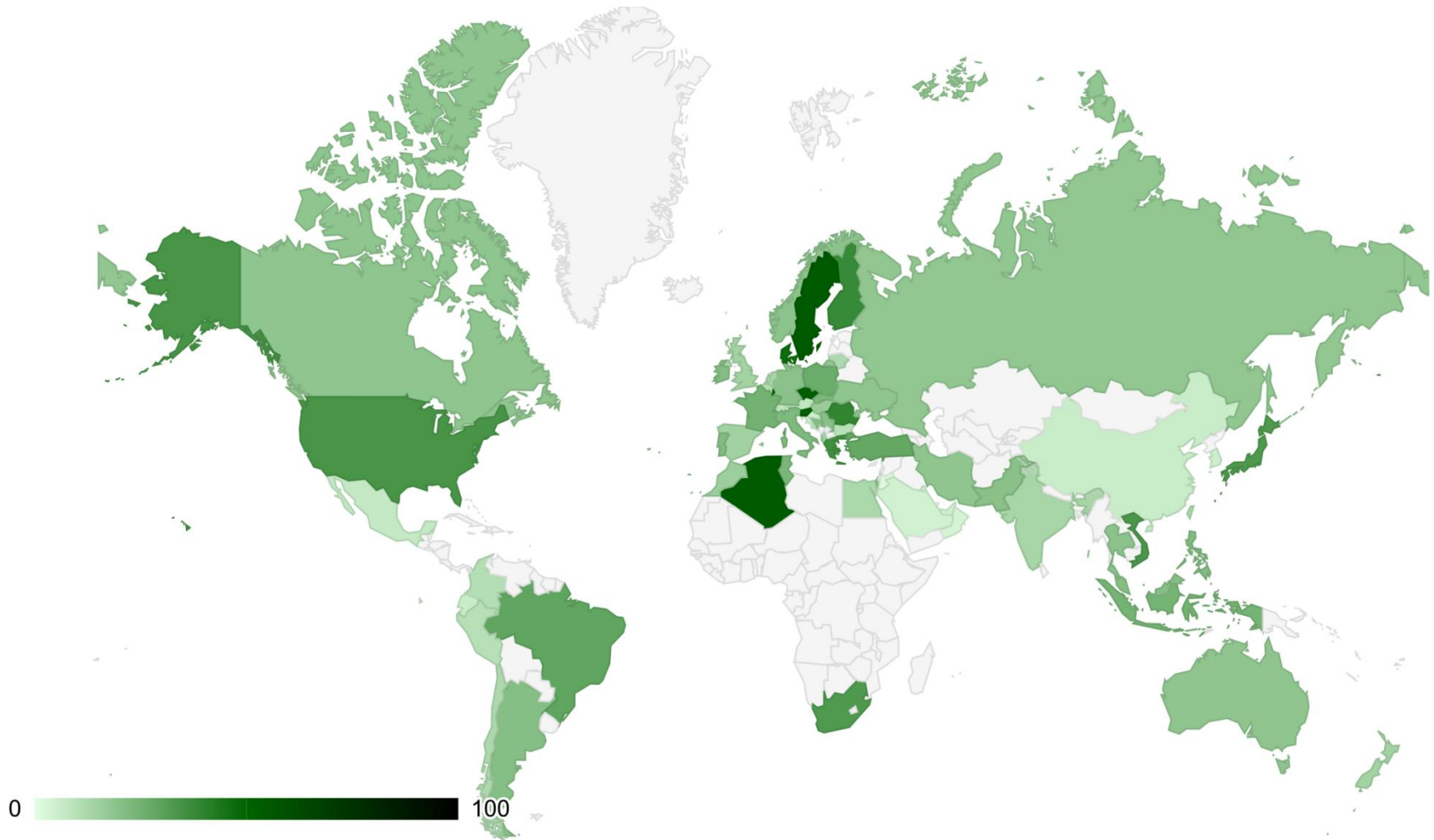
Research Work

- Measurement analysis
 - DNS injection
 - DNSSEC adoption
 - Server-side: signed do

Result	Count
No DNSKEY (dangling DS)	17,751
No trusted DNSKEY (dangling DS)	1,066
No RRSIG for trusted DNSKEY	238
Signature expired	2,138
Signature verify failure	5
Validation failure	21,198
Validation success	3,416,700

TLD	NSEC(3)	Secure Del.
nl	NSEC3, opt-out, $i = 5$	2,279,702
br	mixed ²	566,694
cz	NSEC3, $i = 10$	448,984
com	NSEC3, opt-out, $i = 0$	426,182
se	NSEC	349,514
eu	NSEC3, opt-out, $i = 1$	320,311
fr	NSEC3, opt-out, $i = 1$	205,662
no	NSEC3, opt-out, $i = 5$	119,759
be	NSEC3, opt-out, $i = 5$	92,385
net	NSEC3, opt-out, $i = 0$	81,391
org	NSEC3, opt-out, $i = 1$	46,382
ovh	NSEC3, opt-out, $i = 1$	29,372
nu	NSEC3, $i = 5$	21,126
de	NSEC3, opt-out, $i = 15$	20,004
pl	NSEC3, opt-out, $i = 12$	18,110
		<i>[632 others omitted]</i>
Total:		5,146,705

Client Validation Ratio per Country



Research Work

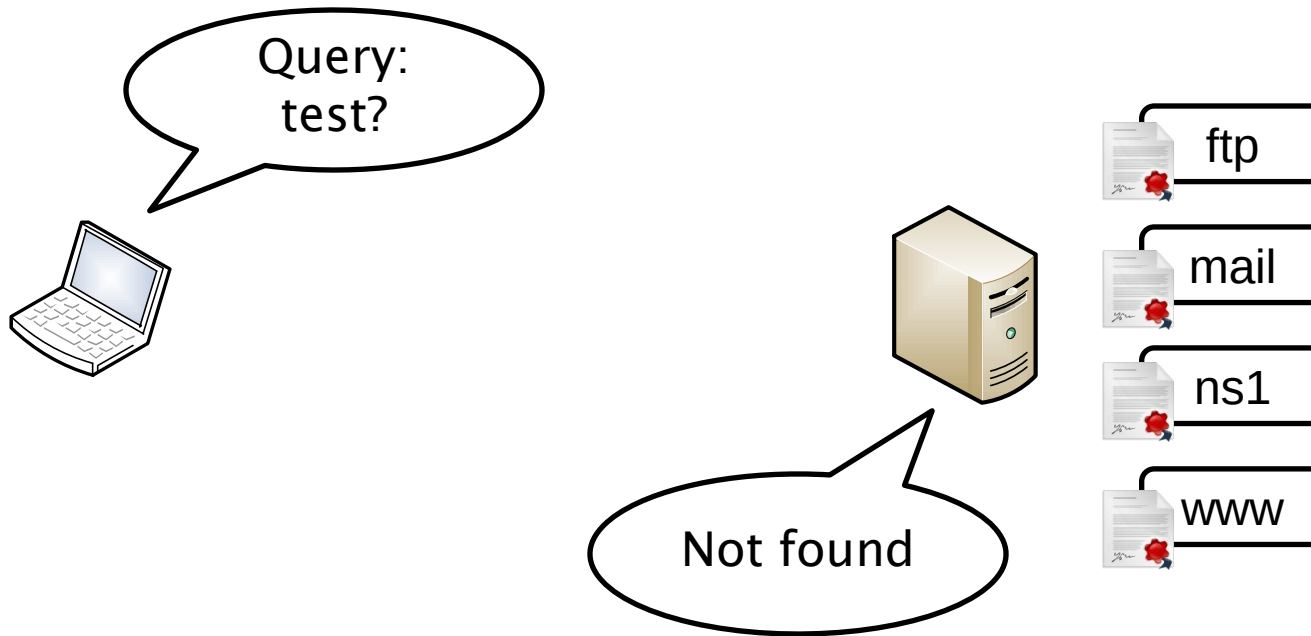
- Measurement analysis
 - DNS injection
 - DNSSEC adoption
 - Server-side: signed domains
 - Client-side: validators
- Attack methods on NSEC3 privacy goal

RESEARCH WORK

NEGATIVE RESPONSES

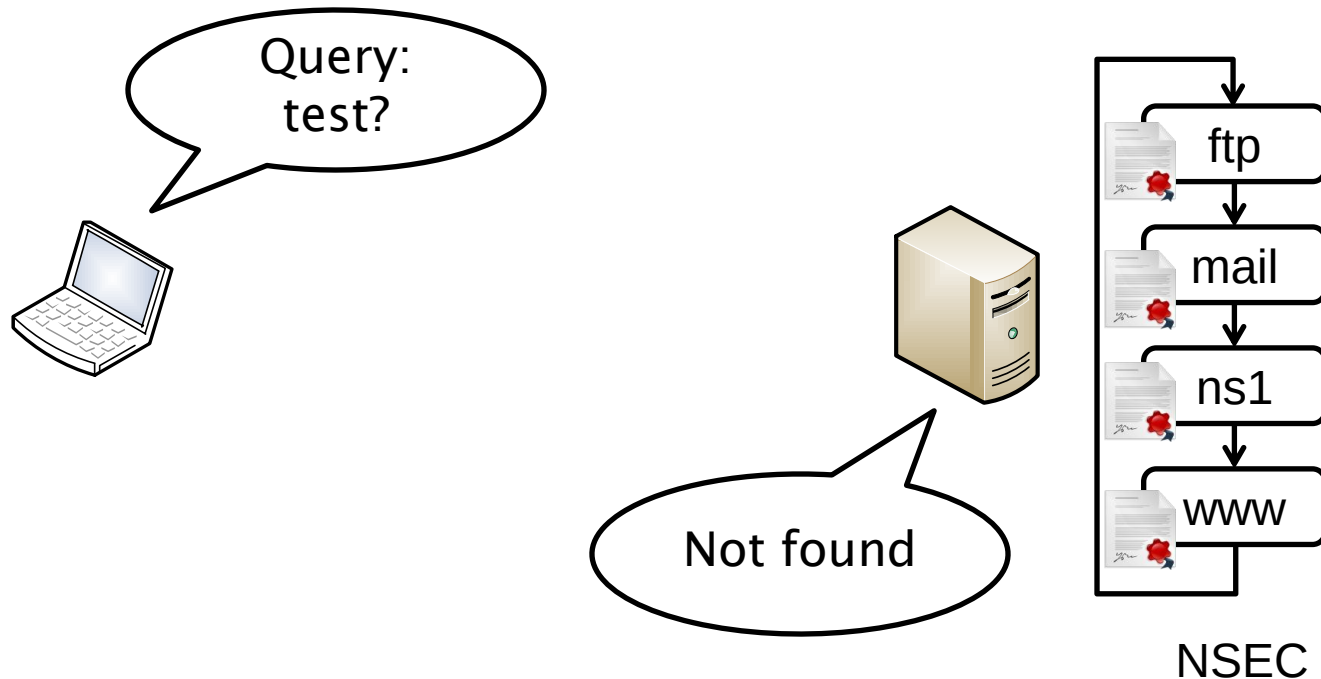
ATTACKING NSEC3

Secure Denial of Existence: NSEC



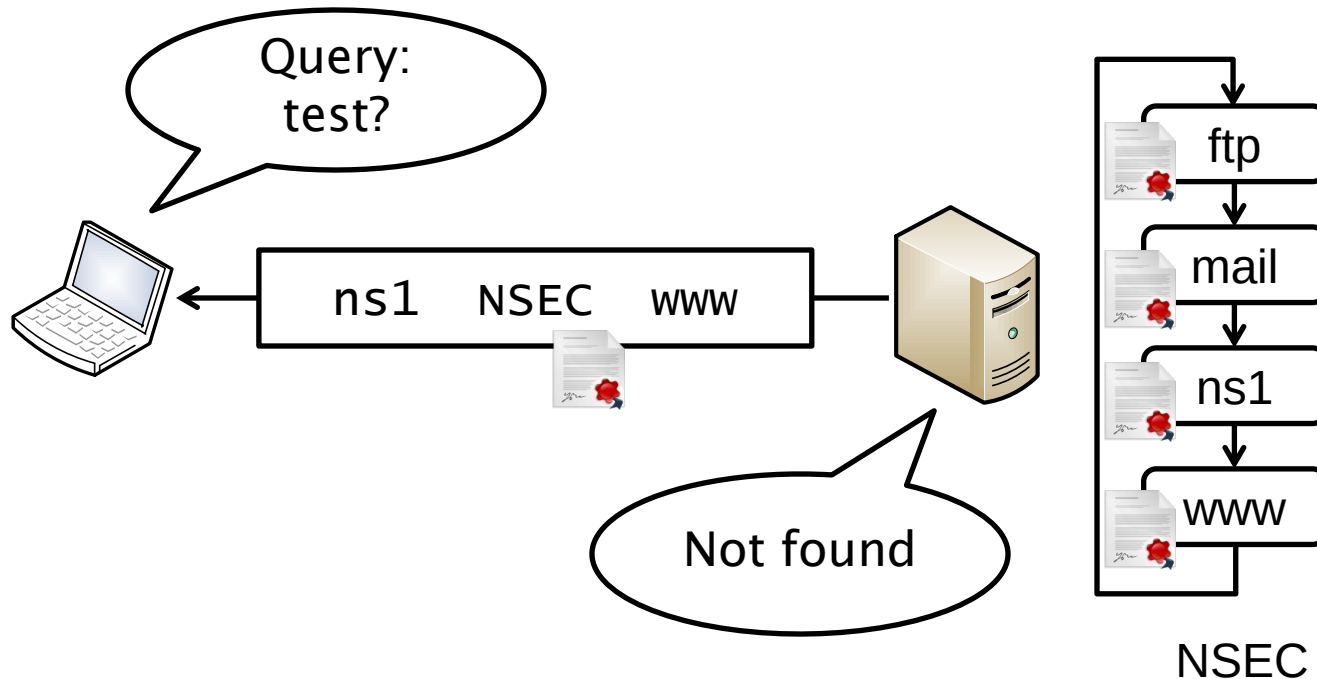
- Query name: x

Secure Denial of Existence: NSEC



- Query name: x

Secure Denial of Existence: NSEC



- Query name: x
- Response: (r_1, r_2) with $r_1 < x < r_2$

Zone Enumeration

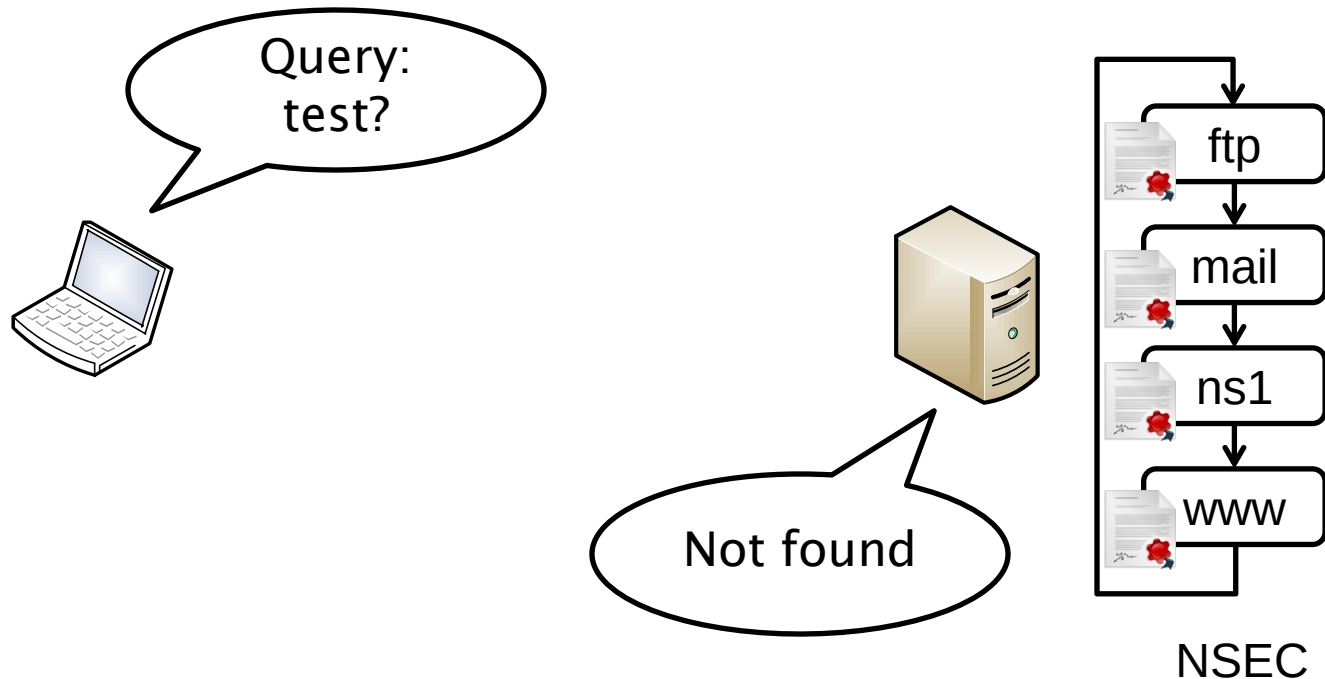
- Copy DNSSEC zone by crawling NSEC records
- Nominet (.uk) position:

„Zone file enumeration is a show-stopper for us that will prevent us from fully implementing DNSSEC.“

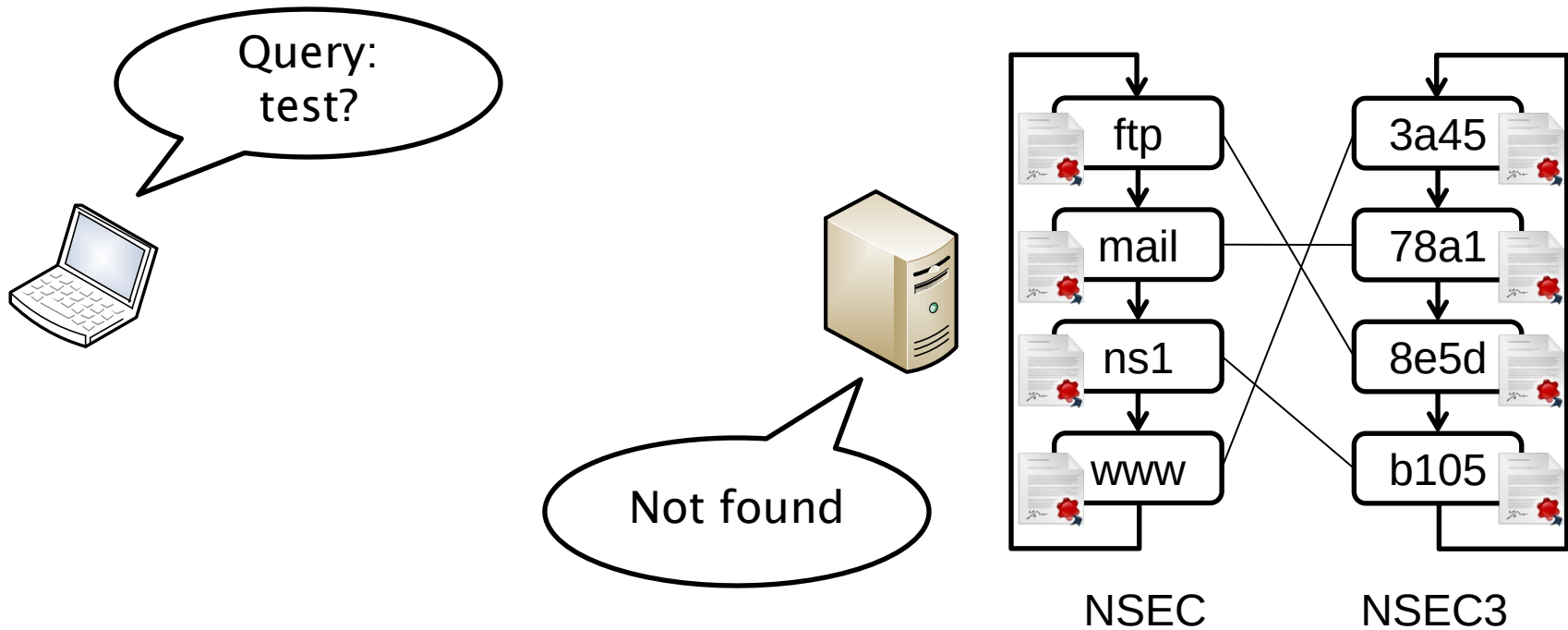
- DENIC (.de) position:

„In conflict with Germany's Federal Data Protection Act“

Secure Denial of Existence: NSEC3

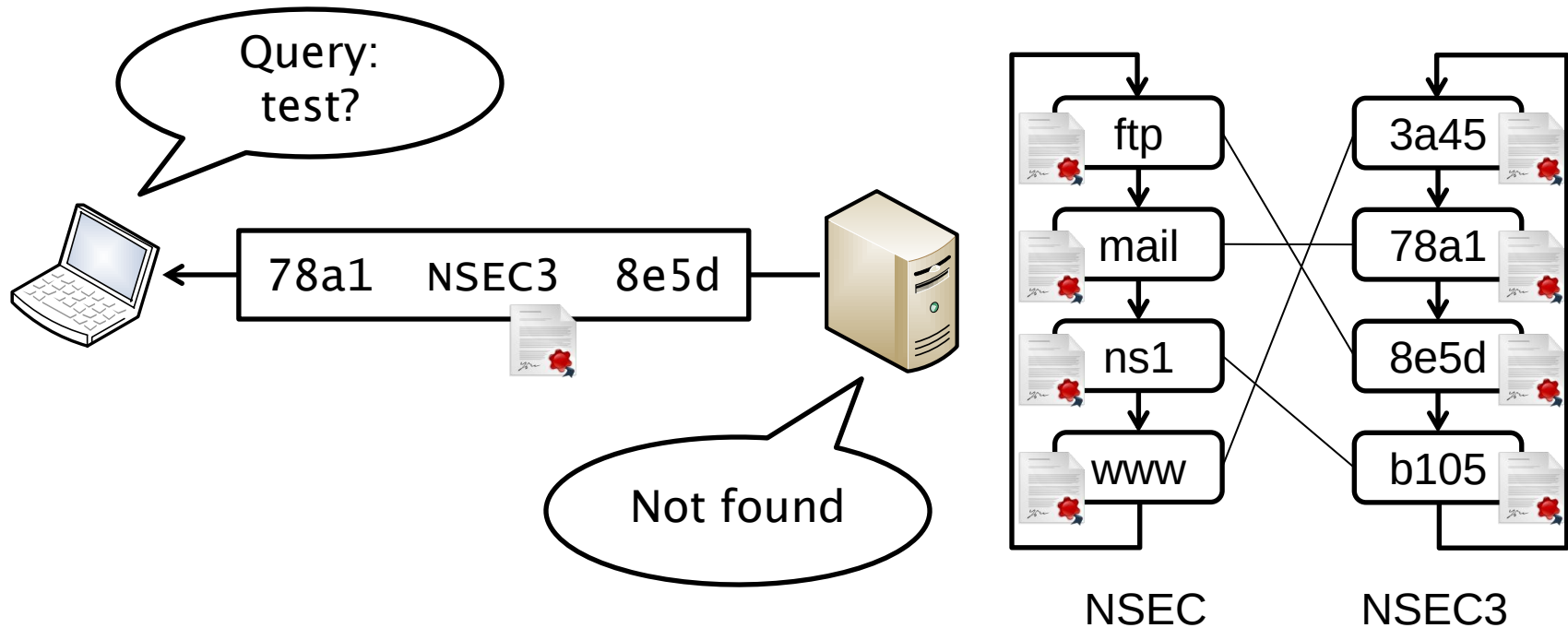


Secure Denial of Existence: NSEC3



- Hash query name: $h(x)$

Secure Denial of Existence: NSEC3



- Hash query name: $h(x)$
- Response: $(h(r_1), h(r_2))$ with $h(r_1) < h(x) < h(r_2)$

NSEC3 Definition

$$\begin{aligned} h(x, s, 0) &= \text{SHA1}(\overset{\text{domain name}}{x} || \overset{\text{salt}}{s}) \\ h(x, s, i) &= \text{SHA1}(\overset{\text{additional iterations}}{\uparrow} h(x, s, i-1) || s), \text{ for } i > 0 \end{aligned}$$

- Repeated SHA1 computation
- Salt is identical for all names in a DNSSEC zone

NSEC vs. NSEC3

- NSEC:
 - Discloses domain name database
 - Compatible with offline signing
- NSEC3:
 - Privacy: Protects from zone enumeration
 - Compatible with offline signing
 - CPU: overhead on server for hashing query names
 - Network: 40–50% larger response sizes

⇒ **How well does NSEC3 prevent zone enumeration?**

RESEARCH WORK

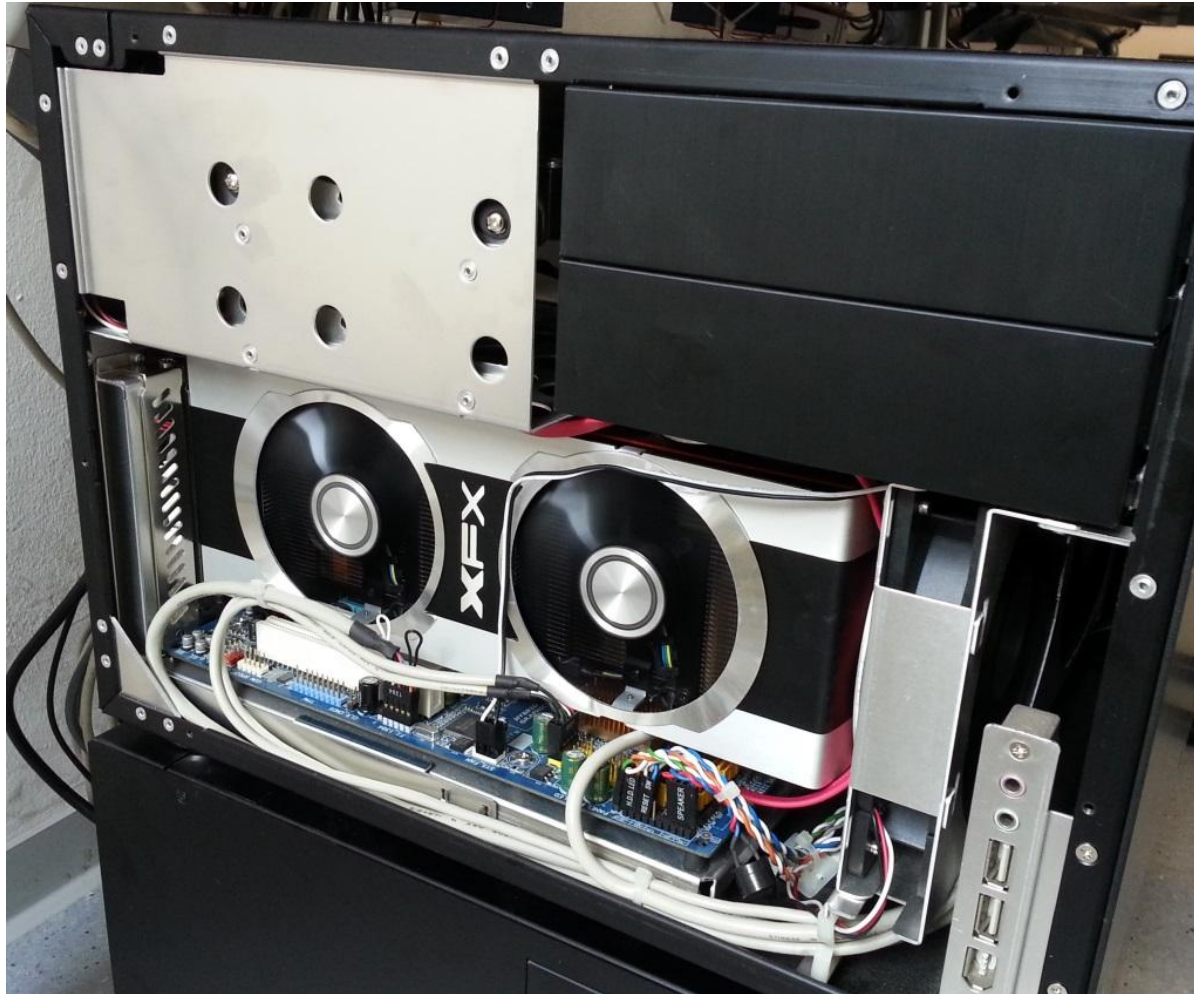
NEGATIVE RESPONSES

ATTACKING NSEC3

ATTACKING NSEC3

1. Collect hashes
2. Reverse hashes

GPU Computing

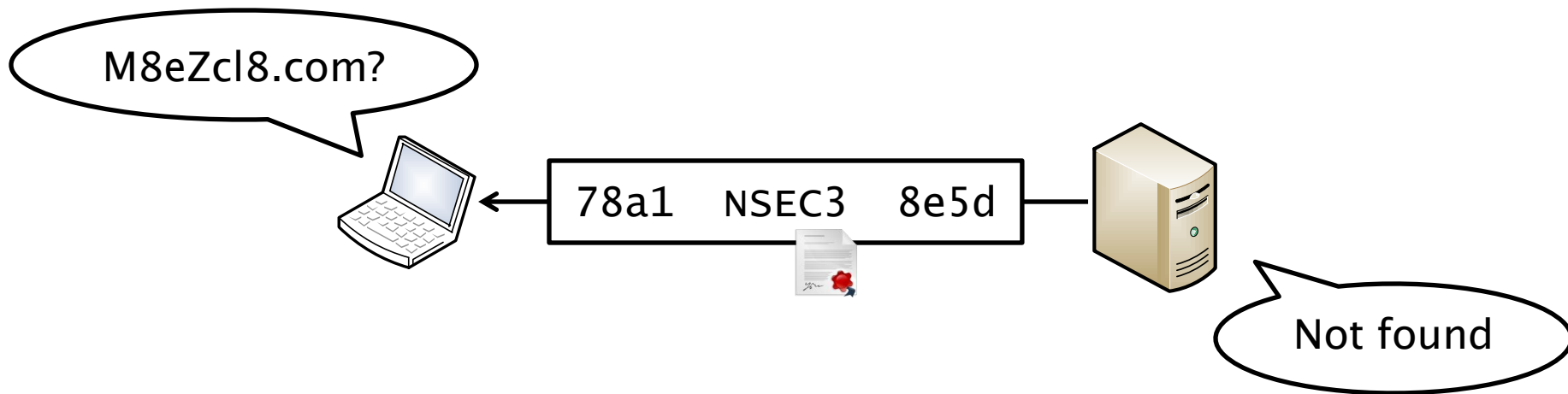


ATTACKING NSEC3

1. Collect hashes
2. Reverse hashes

Hash Crawling

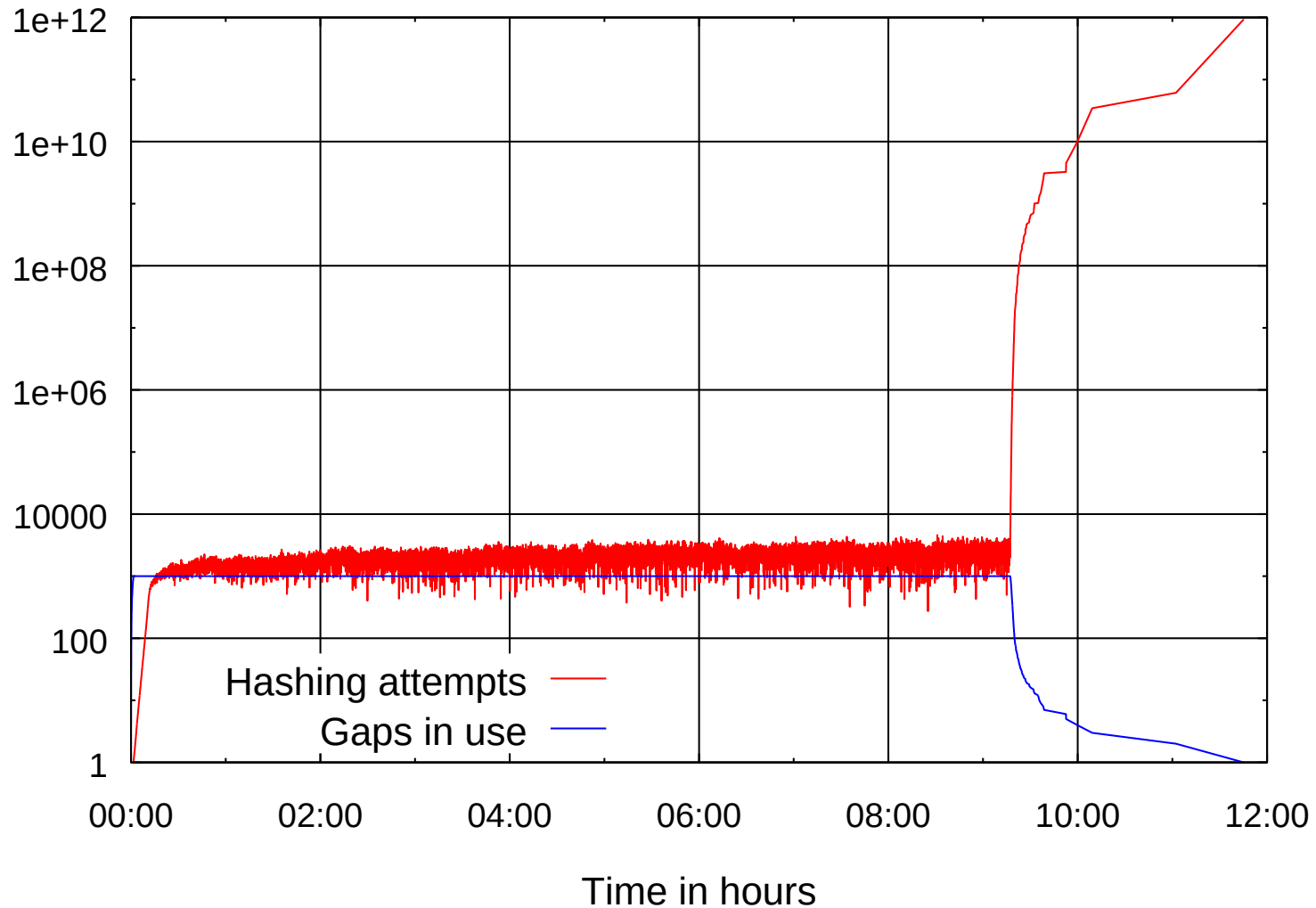
- Retrieve NSEC3 hashes from name server
 - Send queries for random non-existing names



- Check hash before sending query



Crawling 345,000 Hashes from .com



May 2014

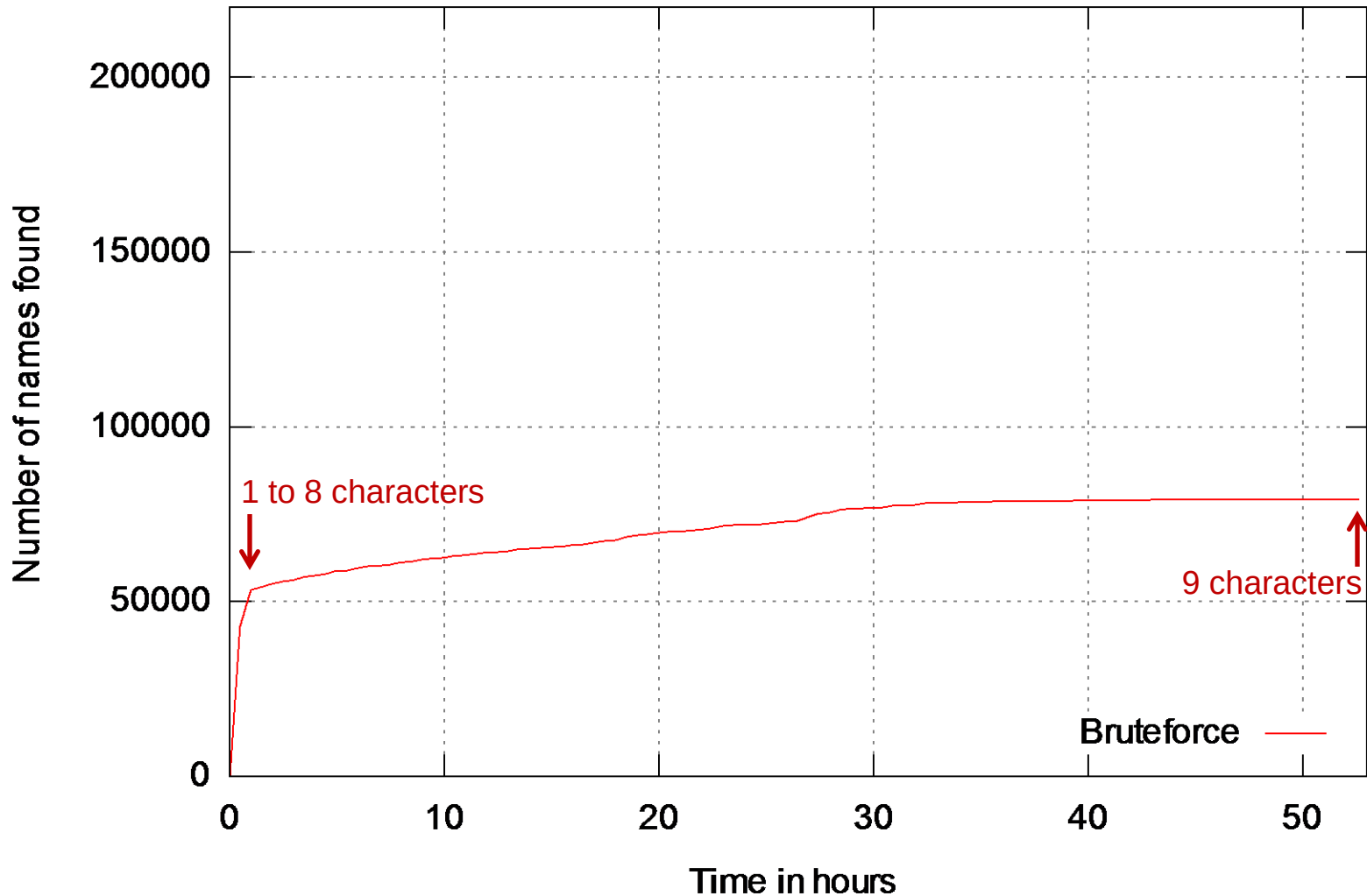
ATTACKING NSEC3

1. Collect hashes
2. Reverse hashes

Hash Breaking Methods

- Brute-force attack
 - Exhaustive search (aaa, aab, aac, ...)
- Dictionary attack
 - Read candidate names from file
- Markov attack
 - Derive candidate names from language model

Brute-Force Attack

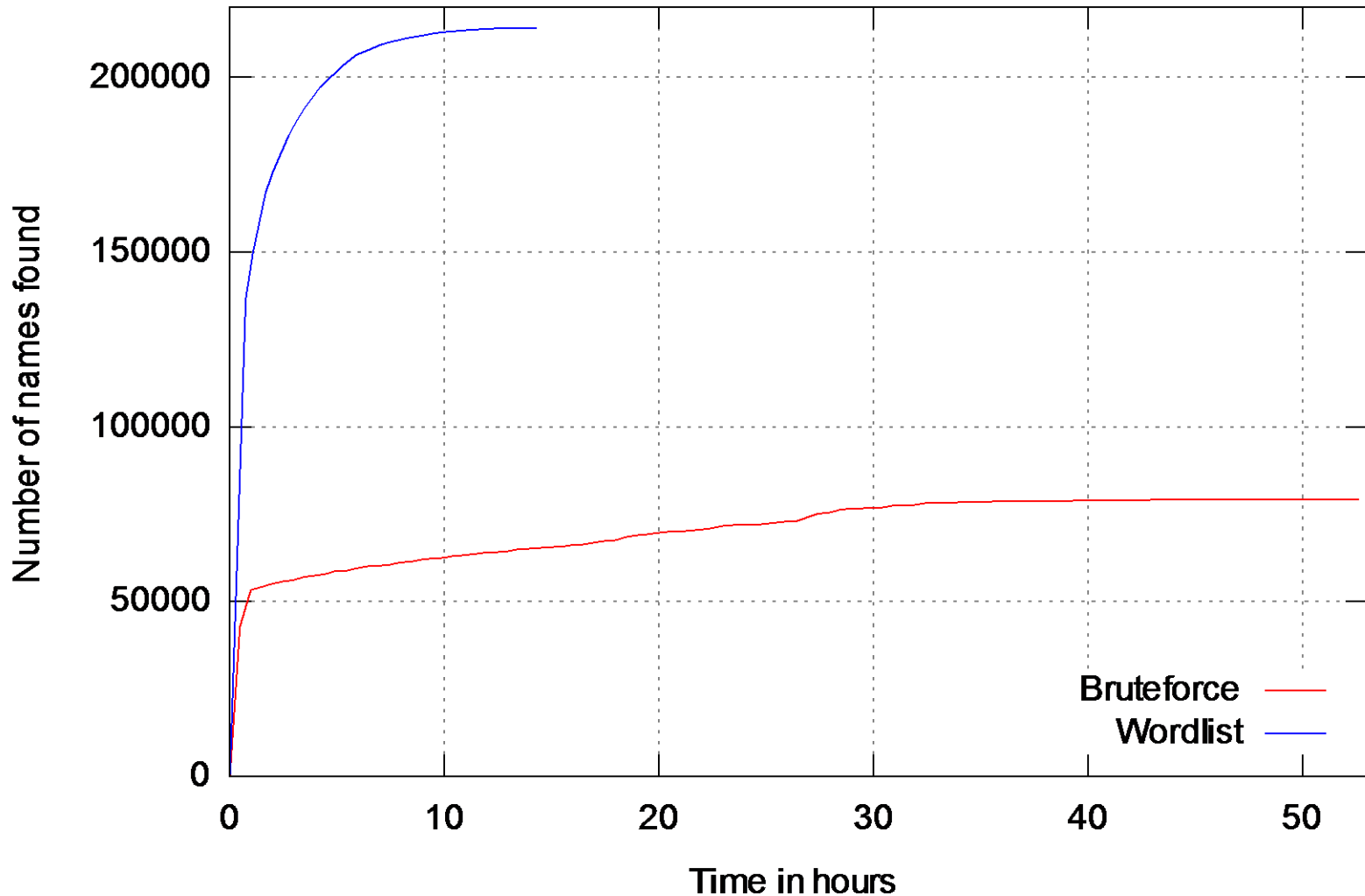


Dictionary Attack

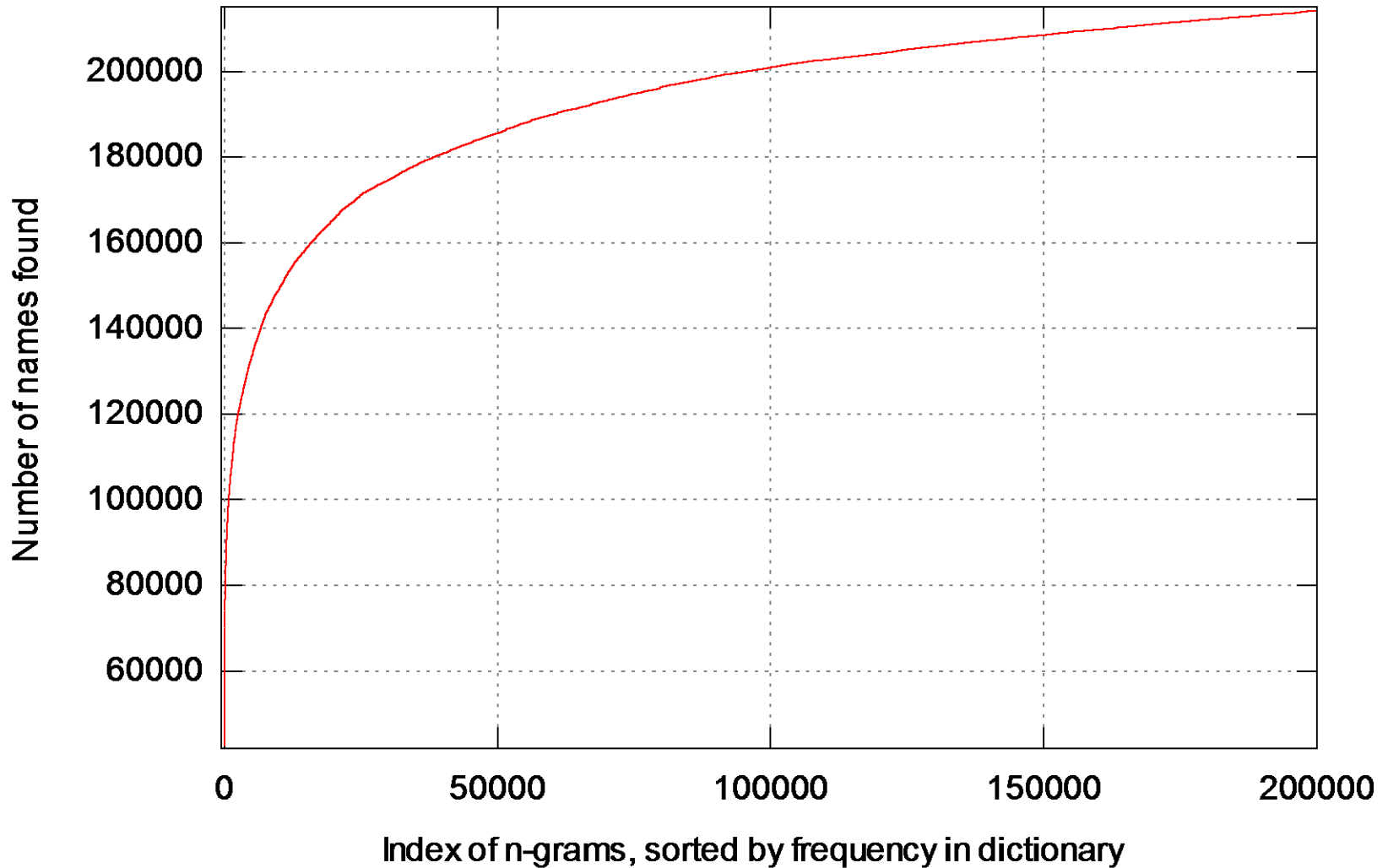
- Effectiveness depends on dictionary quality
 - Good public lists (Alexa, Quantcast)
- 7.1 mio candidate names
- Generate more candidates by inserting strings
 - Insert most common 200,000 n-grams (with $1 \leq n \leq 15$)
 - 1.7×10^{13} candidate names

foo.com. $\longrightarrow$ efoo.com.
 ↑↑↑↑ feoo.com.
 ↑ foeo.com.
 ↑ fooe.com.
 e

Dictionary Attack



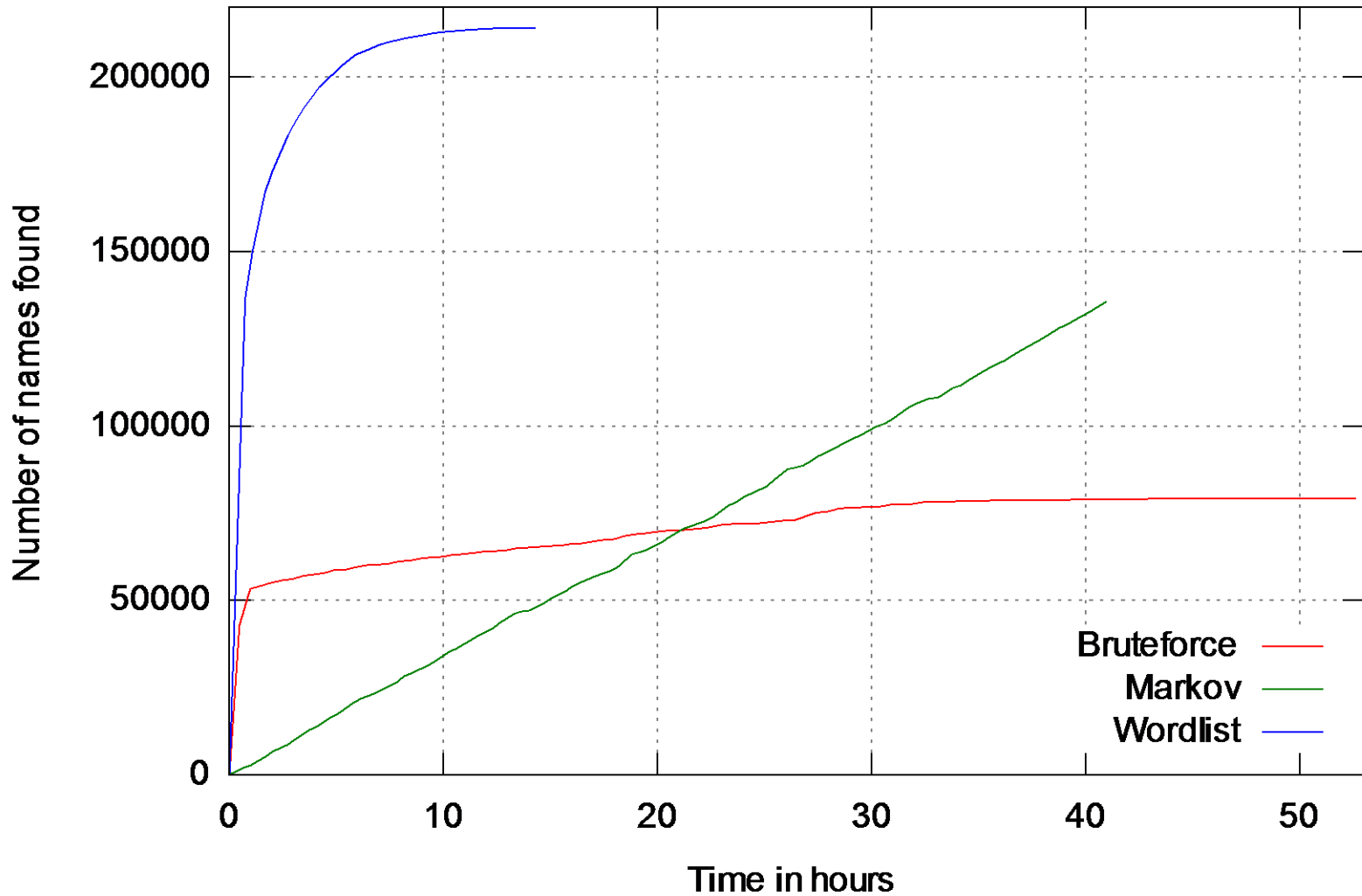
Effectiveness of Insertion Wordlist



Markov Attack

- Some strings are more probable than others
 - ,th‘ more common in English than ,tx‘
 - Model language with 1st order Markov chains
- Markov attack requires training
 - Use hits from brute-force and dictionary attacks
 - Markov model derived from names found
- Enumerate most probable names
 - Omit others (with given time frame)

Markov Attack

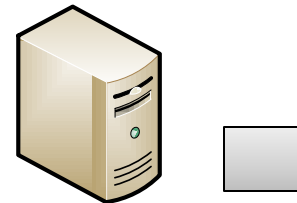


Countermeasure: Adjust Iterations

- Iterations increase workload linearly
 - For attacker and server operator
 - Unlike e.g. cipher key length



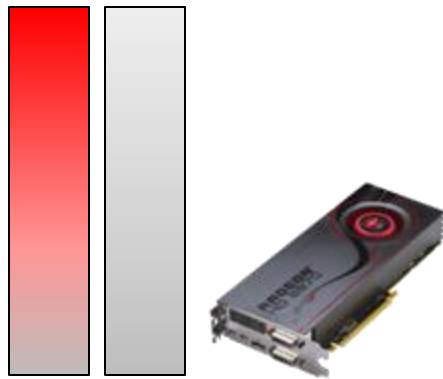
Attacker



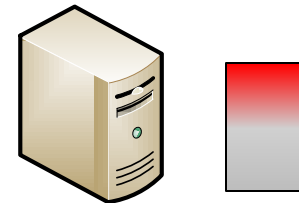
Server operator

Adjusting iterations

- Iterations increase workload linearly
 - For attacker and server operator
 - Unlike e.g. cipher key length



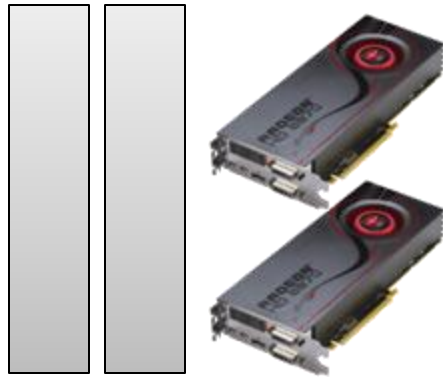
Attacker



Server operator

Adjusting iterations

- Iterations increase workload linearly
 - For attacker and server operator
 - Unlike e.g. cipher key length

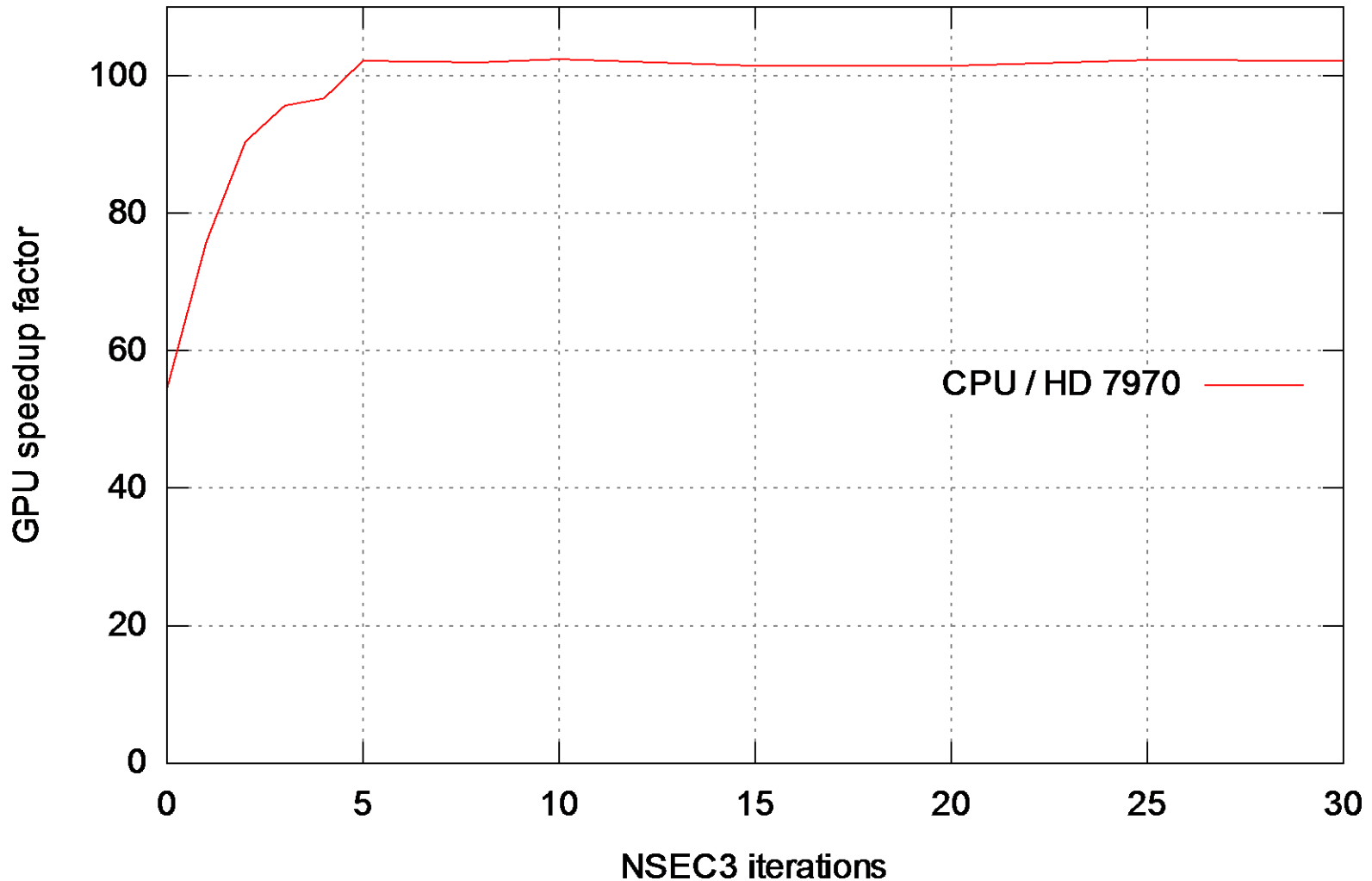


Attacker



Server operator

Efficiency of CPU vs. GPU



Conclusions

- DNSSEC authenticates domain name data
- Negative responses with NSEC/NSEC3
 - NSEC discloses domain database
 - NSEC3 incurs additional server costs
- Break NSEC3 hashes efficiently with GPUs
 - Best method: dictionary with inserted n-grams
- Reversed 65% .com NSEC3 hashes in 5 days
- Increasing iterations is not economical