



# UNIVERSITY OF TWENTE.

## **THE CHALLENGES IN USING PQC FOR DNSSEC**

Prof. dr. ir. Roland van Rijswijk-Deij

Design and Analysis of Communication Systems group

University of Twente

# WHO AM I?





# WHO AM I?

- Professor of Internet Security





# WHO AM I?

- Professor of Internet Security
- Prior roles in industry (among which a decade working for the National Research and Education Network in NL)





# WHO AM I?

- Professor of Internet Security
- Prior roles in industry (among which a decade working for the National Research and Education Network in NL)
- Two+ decades of experience with DNS





# WHO AM I?

- Professor of Internet Security
- Prior roles in industry (among which a decade working for the National Research and Education Network in NL)
- Two+ decades of experience with DNS
- Interested in DNSSEC operations research





# THIS TALK



# THIS TALK

- Quick primer: DNS + DNSSEC



# THIS TALK

- Quick primer: DNS + DNSSEC
- The problematic history of DNSSEC transport



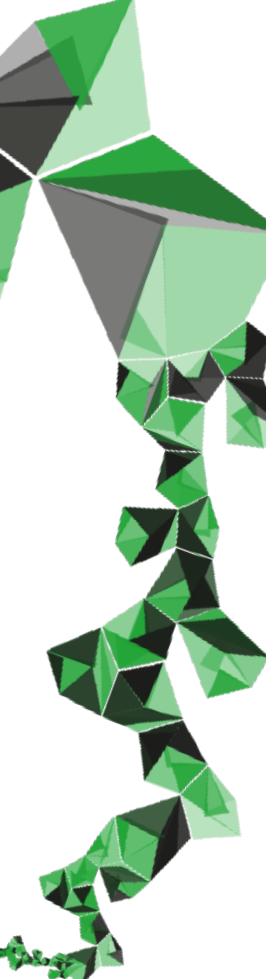
# THIS TALK

- Quick primer: DNS + DNSSEC
- The problematic history of DNSSEC transport
- Solving DNSSEC transport problems



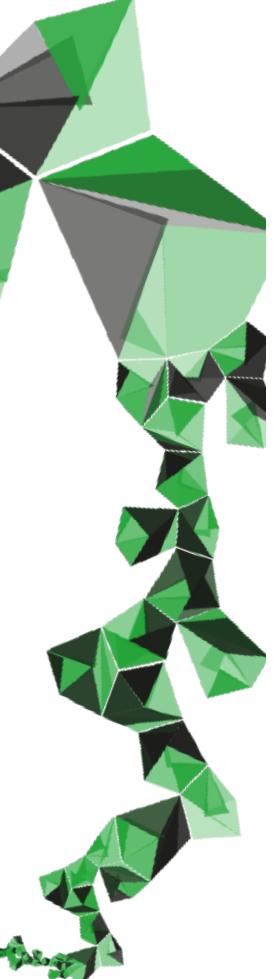
# THIS TALK

- Quick primer: DNS + DNSSEC
- The problematic history of DNSSEC transport
- Solving DNSSEC transport problems
- DNS Flag Day



# THIS TALK

- Quick primer: DNS + DNSSEC
- The problematic history of DNSSEC transport
- Solving DNSSEC transport problems
- DNS Flag Day
- PQC and DNSSEC



# THIS TALK

- Quick primer: DNS + DNSSEC
- The problematic history of DNSSEC transport
- Solving DNSSEC transport problems
- DNS Flag Day
- PQC and DNSSEC
- Q&A



# A QUICK PRIMER: THE DNS



# A QUICK PRIMER: THE DNS

- You use the Domain Name System every day



# A QUICK PRIMER: THE DNS

- You use the Domain Name System every day
- Human-readable names to machine readable information



# A QUICK PRIMER: THE DNS

- You use the Domain Name System every day
- Human-readable names to machine readable information
- Most common use: mapping names to IP addresses:

```
$ dig example.org

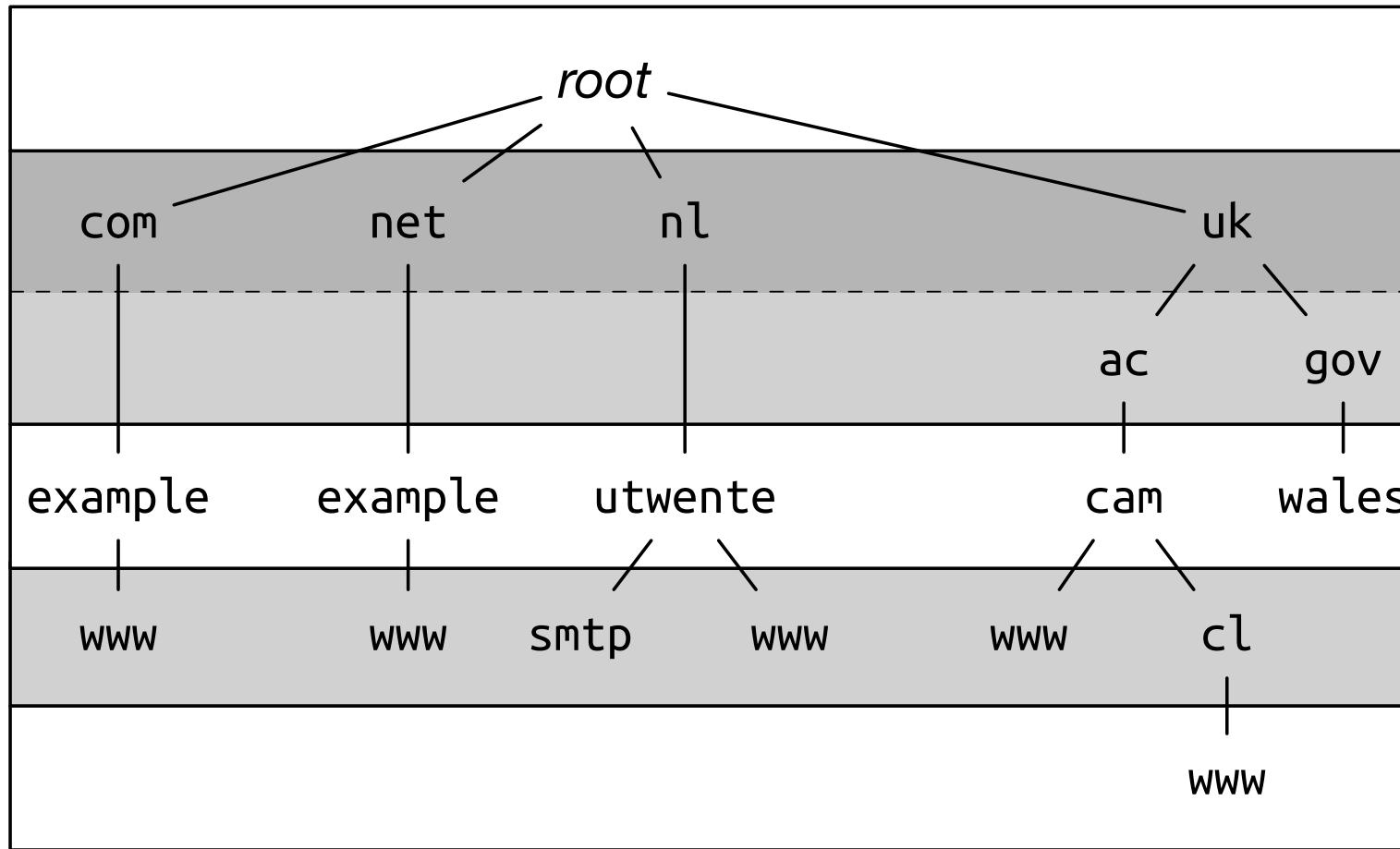
; <>> DiG 9.10.6 <>> example.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51982
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;example.org.           IN      A

;; ANSWER SECTION:
example.org.        72072   IN      A      93.184.216.34

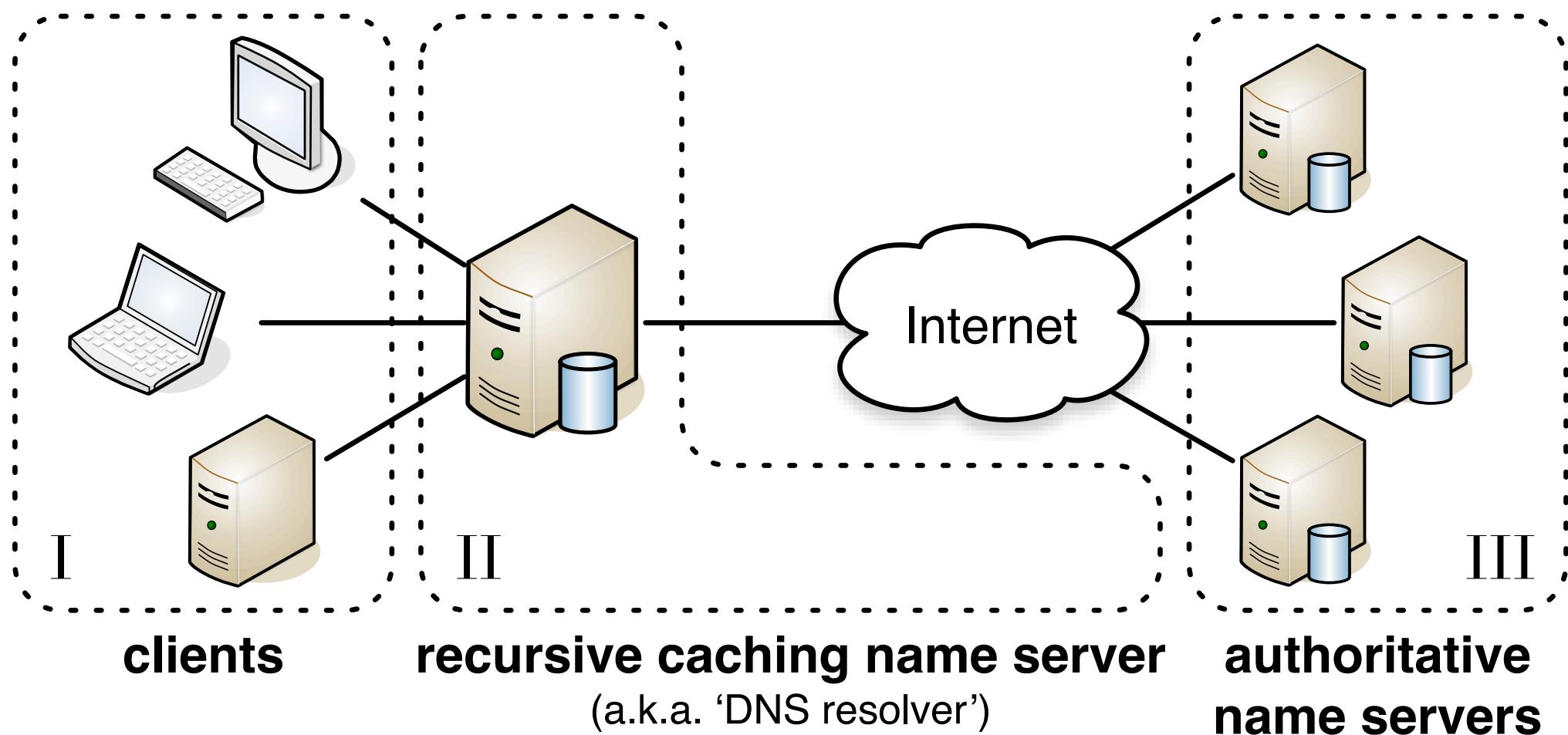
;; Query time: 25 msec
;; SERVER: 2001:610:3:200a:192:87:36:36#53(2001:610:3:200a:192:87:36:36)
;; WHEN: Thu Sep 16 14:11:09 CEST 2021
;; MSG SIZE  rcvd: 56
```

# HIERARCHICAL NAME SPACE

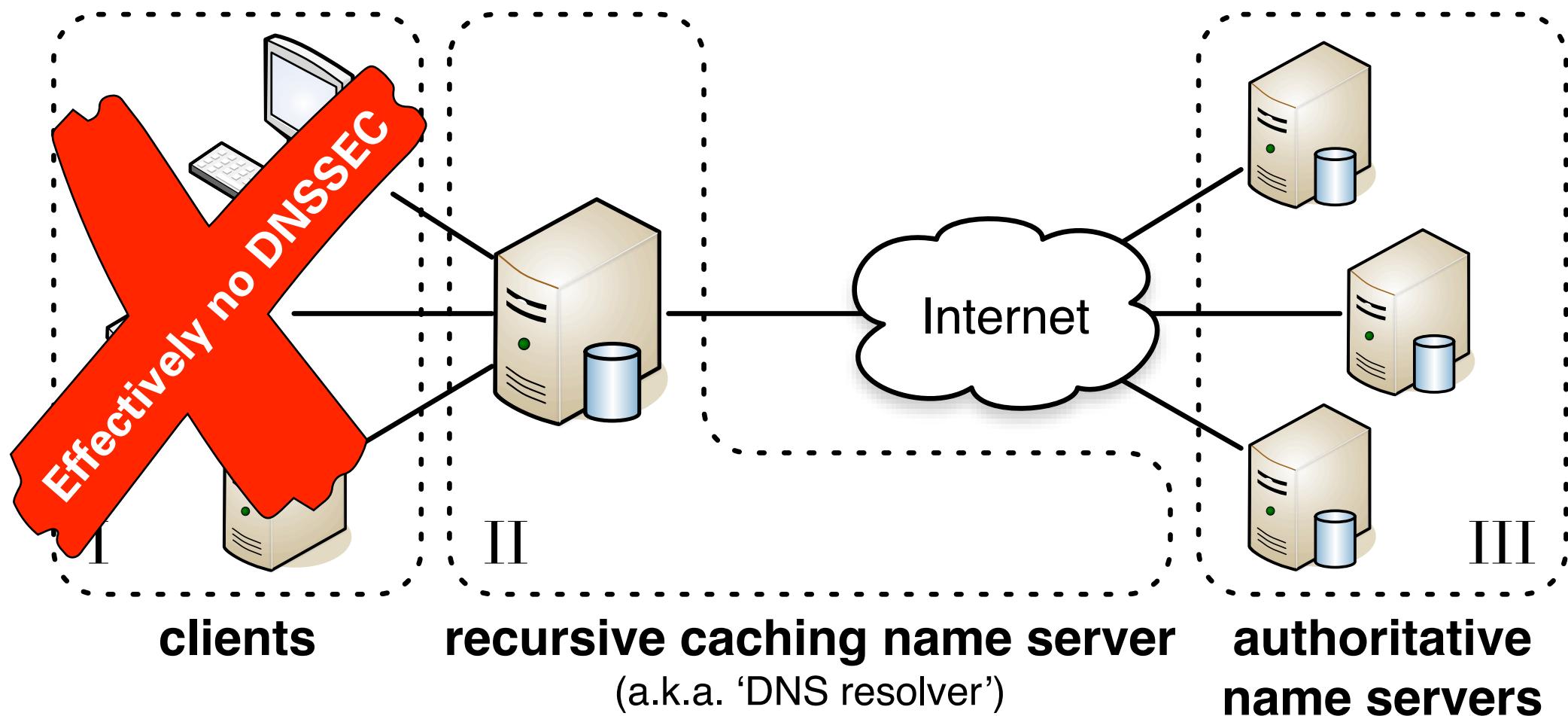


root level  
top level domains  
public suffixes  
second level domains  
third level domains  
further levels

# DNS ROLES



# DNS ROLES



# PROBLEMS WITH THE DNS





# PROBLEMS WITH THE DNS

- Original protocol dates back to the early 1980s



# PROBLEMS WITH THE DNS

- Original protocol dates back to the early 1980s
- No built-in security; everybody knew everybody on the Internet



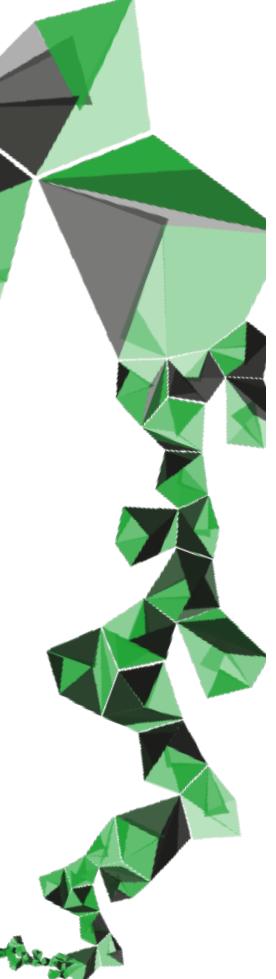
# PROBLEMS WITH THE DNS

- Original protocol dates back to the early 1980s
- No built-in security; everybody knew everybody on the Internet
- 1997: First “cache poisoning” attack on DNS



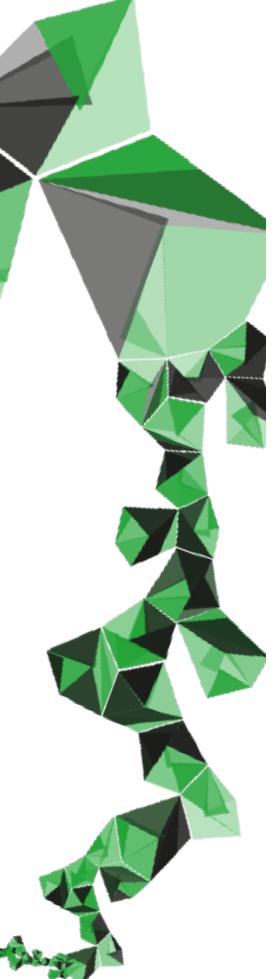
# PROBLEMS WITH THE DNS

- Original protocol dates back to the early 1980s
- No built-in security; everybody knew everybody on the Internet
- 1997: First “cache poisoning” attack on DNS
- 2008: Kaminsky variant makes cache poisoning almost trivial



# PROBLEMS WITH THE DNS

- Original protocol dates back to the early 1980s
- No built-in security; everybody knew everybody on the Internet
- 1997: First “cache poisoning” attack on DNS
- 2008: Kaminsky variant makes cache poisoning almost trivial
- 2013: Herzberg publishes variant leveraging fragmentation (we’ll come back to this)



# PROBLEMS WITH THE DNS

- Original protocol dates back to the early 1980s
- No built-in security; everybody knew everybody on the Internet
- 1997: First “cache poisoning” attack on DNS
- 2008: Kaminsky variant makes cache poisoning almost trivial
- 2013: Herzberg publishes variant leveraging fragmentation (we’ll come back to this)
- Root cause: no mechanisms for authenticity and integrity



# ENTER DNSSEC



# ENTER DNSSEC

- Goal: add authenticity and integrity to the DNS
- Note: DNSSEC does not add confidentiality*



# ENTER DNSSEC

- Goal: add authenticity and integrity to the DNS  
*Note: DNSSEC does not add confidentiality*
- Adds additional resource records to the DNS for signatures (RRSIG) and public keys (DNSKEY)



# ENTER DNSSEC

- Goal: add authenticity and integrity to the DNS  
*Note: DNSSEC does not add confidentiality*
- Adds additional resource records to the DNS for signatures (RRSIG) and public keys (DNSKEY)
- Signing is typically offline (entire DNS zone) but can also be online (e.g. for content delivery networks that use DNS for load balancing and local content cache selection)



# ENTER DNSSEC

- Goal: add authenticity and integrity to the DNS  
*Note: DNSSEC does not add confidentiality*
- Adds additional resource records to the DNS for signatures (RRSIG) and public keys (DNSKEY)
- Signing is typically offline (entire DNS zone) but can also be online (e.g. for content delivery networks that use DNS for load balancing and local content cache selection)
- Single root of trust (DNS root zone), validation along chain of trust (explained later)



# DNS ZONE

1	\$ORIGIN example.com.					
2	...					
3	<i>domain name</i>	<i>TTL</i>	<i>class</i>	<i>type</i>	<i>value</i>	
2	@	86400	IN	A	93.184.216.34	← RRset #1
3	@	86400	IN	AAAA	2606:2800:...:1946	← RRset #2
4	@	86400	IN	NS	a.iana-servers.net.	← RRset #3
5	@	86400	IN	NS	b.iana-servers.net.	
6	www	86400	IN	CNAME	example.com.	← Alias
7	sub	3600	IN	NS	ns1.example.org.	← Delegation
8	sub	3600	IN	NS	ns2.example.org.	
9	lorem.ipsum	86400	IN	A	127.0.0.1	← Results in empty non-terminal
10	*.dolor	300	IN	TXT	"Sed ut perspiciatis unde omnis..."	← Wildcard



# DNSSEC-SIGNED ZONE

	<i>domain name</i>	<i>TTL</i>	<i>class</i>	<i>type</i>	<i>value</i>	
1	example.com.	86400	IN	A	93.184.216.34	← RRset #1
2	example.com.	86400	IN	RRSIG	A 8 2 86400 ...	← signature for RRset #1
3	example.com.	86400	IN	AAAA	2606:2800::...:1946	← RRset #2
4	example.com.	86400	IN	RRSIG	AAAA 8 2 86400 ...	← signature for RRset #2
5	example.com.	86400	IN	NS	a.iana-servers.net.	} ← RRset #3
6	example.com.	86400	IN	NS	b.iana-servers.net.	
7	example.com.	86400	IN	RRSIG	NS 8 2 86400 ...	← signature for RRset #3
8	example.com.	3600	IN	DNSKEY	257 3 8 AwEAAAdCU...	} ← Set of public keys
9	example.com.	3600	IN	DNSKEY	256 3 8 AwEAAZ0a...	
10	example.com.	3600	IN	RRSIG	DNSKEY 8 2 3600 ...	← signature over DNSKEY set



# DNSSEC-SIGNED ZONE

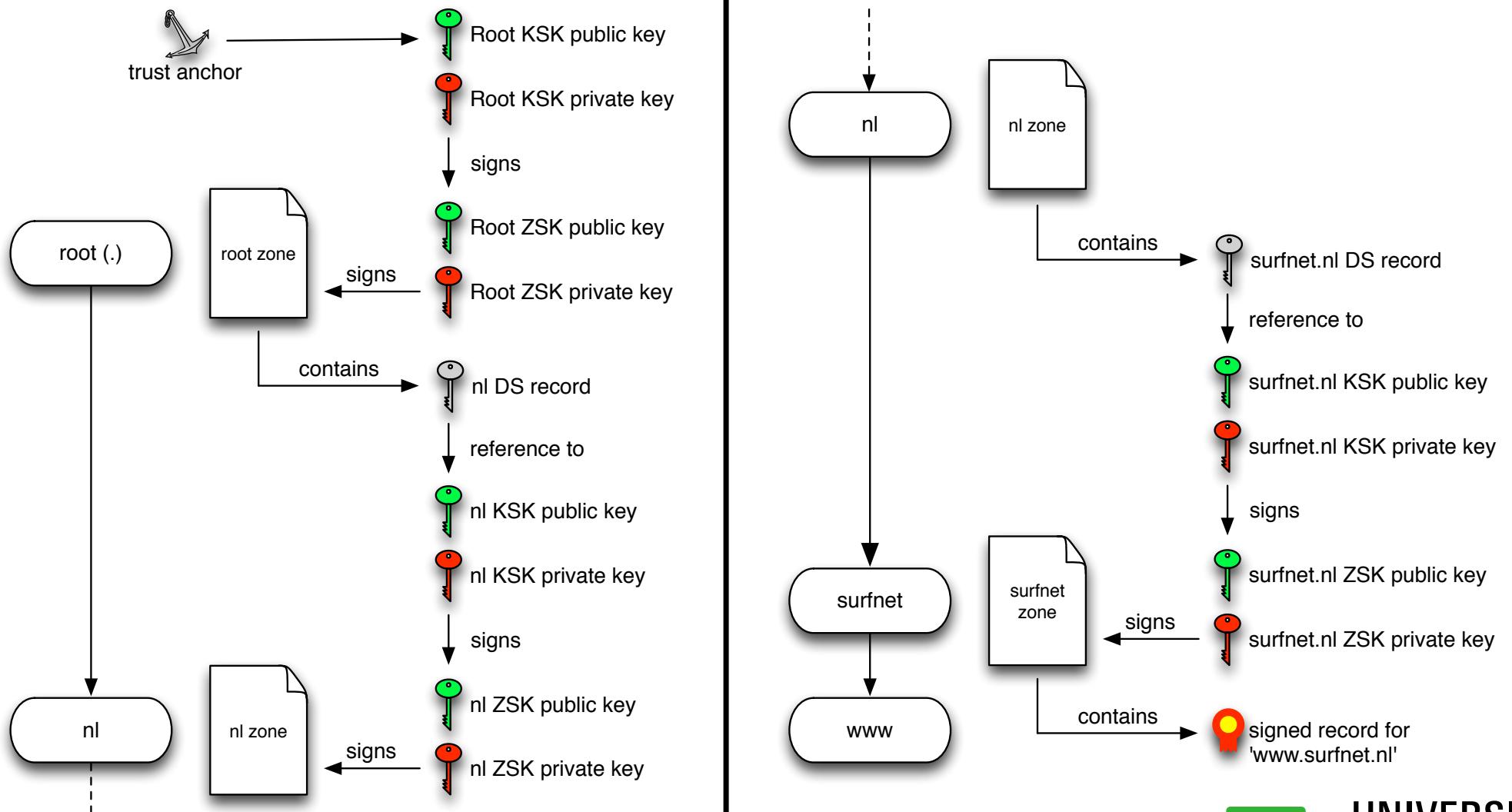
	<i>domain name</i>	<i>TTL</i>	<i>class</i>	<i>type</i>	<i>value</i>	
1	example.com.	86400	IN	A	93.184.216.34	← RRset #1
2	example.com.	86400	IN	RRSIG	A 8 2 86400 ...	← signature for RRset #1
3	example.com.	86400	IN	AAAA	2606:2800::...:1946	← RRset #2
4	example.com.	86400	IN	RRSIG	AAAA 8 2 86400 ...	← signature for RRset #2
5	example.com.	86400	IN	NS	a.iana-servers.net.	} ← RRset #3
6	example.com.	86400	IN	NS	b.iana-servers.net.	
7	example.com.	86400	IN	RRSIG	NS 8 2 86400 ...	← signature for RRset #3
8	example.com.	3600	IN	DNSKEY	257 3 8 AwEAAAdCU...	} ← Set of public keys
9	example.com.	3600	IN	DNSKEY	256 3 8 AwEAAZ0a...	
10	example.com.	3600	IN	RRSIG	DNSKEY 8 2 3600 ...	← signature over DNSKEY set



# DNSSEC-SIGNED ZONE

	<i>domain name</i>	<i>TTL</i>	<i>class</i>	<i>type</i>	<i>value</i>	
1	example.com.	86400	IN	A	93.184.216.34	← RRset #1
2	example.com.	86400	IN	RRSIG	A 8 2 86400 ...	← signature for RRset #1
3	example.com.	86400	IN	AAAA	2606:2800::...:1946	← RRset #2
4	example.com.	86400	IN	RRSIG	AAAA 8 2 86400 ...	← signature for RRset #2
5	example.com.	86400	IN	NS	a.iana-servers.net.	} ← RRset #3
6	example.com.	86400	IN	NS	b.iana-servers.net.	
7	example.com.	86400	IN	RRSIG	NS 8 2 86400 ...	← signature for RRset #3
8	example.com.	3600	IN	DNSKEY	257 3 8 AwEAAAdCU...	} ← Set of public keys
9	example.com.	3600	IN	DNSKEY	256 3 8 AwEAAZ0a...	
10	example.com.	3600	IN	RRSIG	DNSKEY 8 2 3600 ...	← signature over DNSKEY set

# DNSSEC VALIDATION





# DNSSEC TRANSPORT

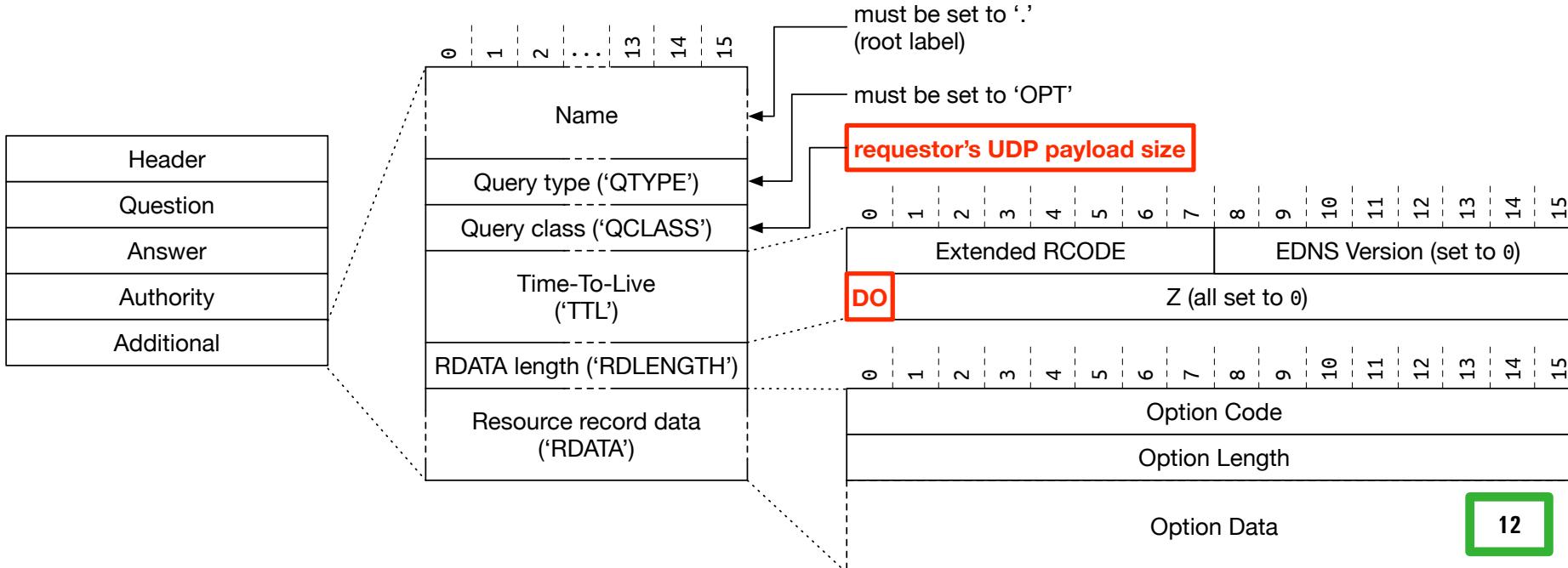


# DNSSEC TRANSPORT

- The original DNS specification allows a maximum message size of 512 bytes — too small for DNSSEC (keys + signatures!)

# DNSSEC TRANSPORT

- The original DNS specification allows a maximum message size of 512 bytes — too small for DNSSEC (keys + signatures!)
- Protocol extension EDNS0 solves this:





# EDNSO AND FRAGMENTATION



# EDNS0 AND FRAGMENTATION

- EDNS0 allows for UDP payloads of almost 64KB



# EDNS0 AND FRAGMENTATION

- EDNS0 allows for UDP payloads of almost 64KB
- Problem: this can lead to IP fragmentation if the message exceeds the Maximum Transmission Unit (MTU)



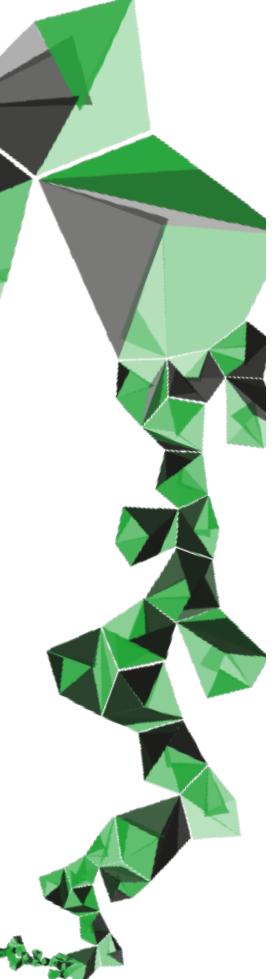
# EDNS0 AND FRAGMENTATION

- EDNS0 allows for UDP payloads of almost 64KB
- Problem: this can lead to IP fragmentation if the message exceeds the Maximum Transmission Unit (MTU)
- This was a huge problem in the “early days” of DNSSEC (roughly pre-2014)



# EDNS0 AND FRAGMENTATION

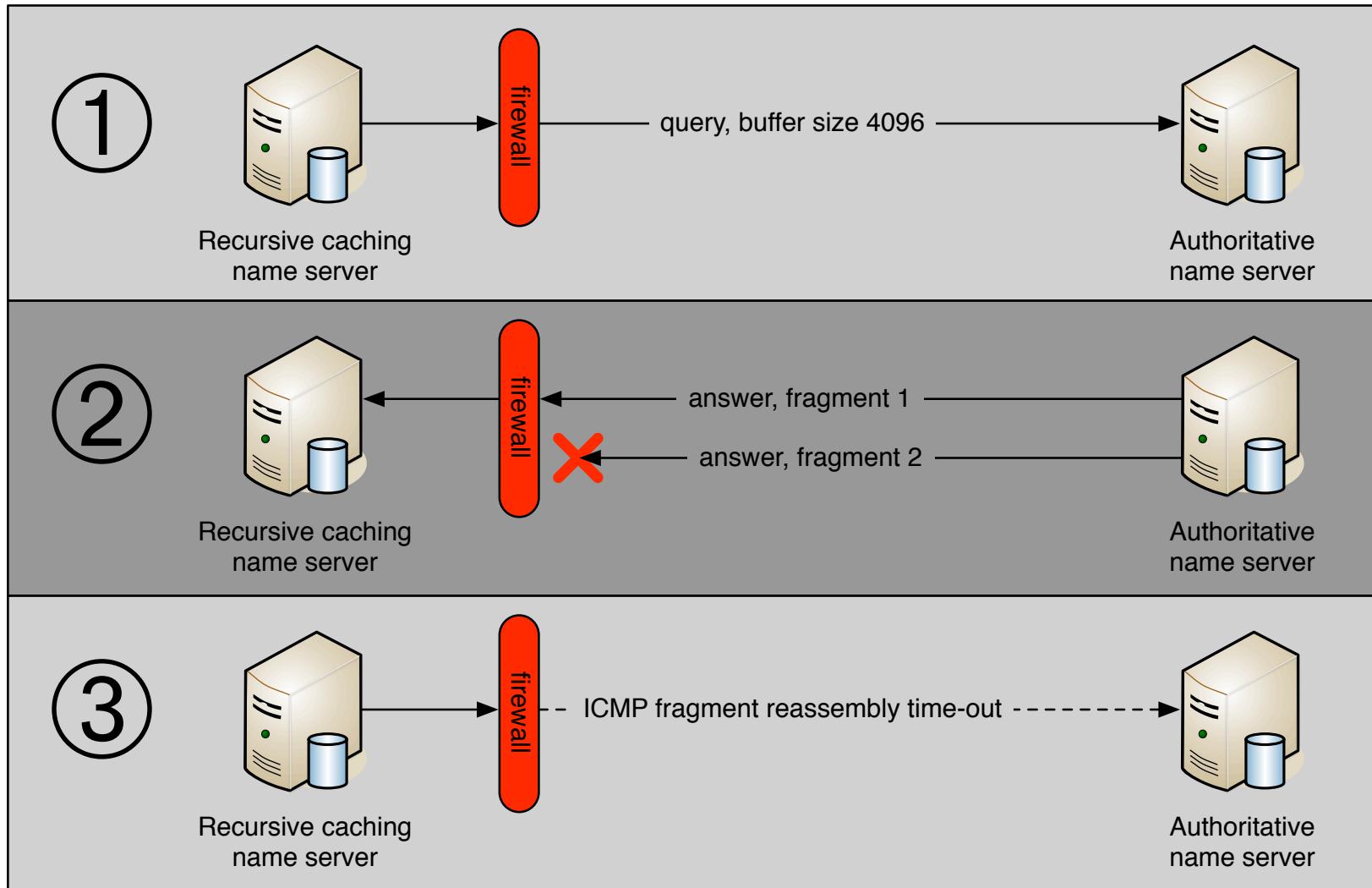
- EDNS0 allows for UDP payloads of almost 64KB
- Problem: this can lead to IP fragmentation if the message exceeds the Maximum Transmission Unit (MTU)
- This was a huge problem in the “early days” of DNSSEC (roughly pre-2014)
- Use of large RSA keys + multiple keys involved in signing a zone



# EDNS0 AND FRAGMENTATION

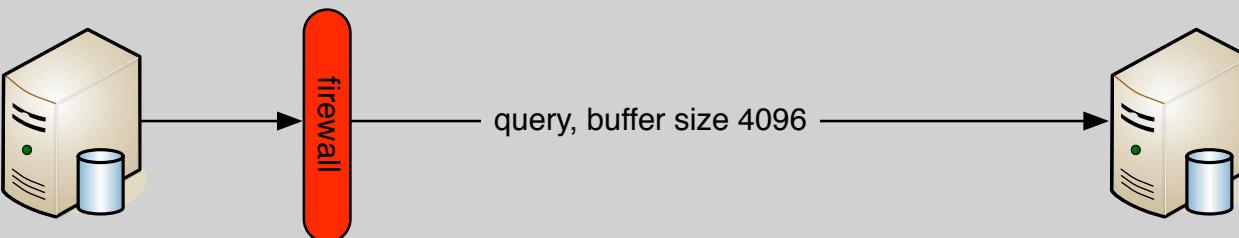
- EDNS0 allows for UDP payloads of almost 64KB
- Problem: this can lead to IP fragmentation if the message exceeds the Maximum Transmission Unit (MTU)
- This was a huge problem in the “early days” of DNSSEC (roughly pre-2014)
- Use of large RSA keys + multiple keys involved in signing a zone
- Fragmentation causes two problems (next two slides)

# UNREACHABILITY



# UNREACHABILITY

1



Research in 2012 by Van den Broek et al. and by Weaver et al. shows that up to 10% of resolvers on the Internet may struggle with fragmented UDP packets.

All mainstream DNS resolver implementations have workarounds for this problem as a consequence.

The EDNS0 RFC was also changed as a consequence of this issue

3



# HERZBERG FRAGMENT CACHE POISONING





# HERZBERG FRAGMENT CACHE POISONING

- Attack introduced in 2013



# HERZBERG FRAGMENT CACHE POISONING

- Attack introduced in 2013
- Leverages fragmentation for cache poisoning



# HERZBERG FRAGMENT CACHE POISONING

- Attack introduced in 2013
- Leverages fragmentation for cache poisoning
- Follow-up fragments do not contain any identification (unlike the first fragment which contains the query ID and source port)



# HERZBERG FRAGMENT CACHE POISONING

- Attack introduced in 2013
- Leverages fragmentation for cache poisoning
- Follow-up fragments do not contain any identification (unlike the first fragment which contains the query ID and source port)
- Put records to poison in second (or later) fragment



# HERZBERG FRAGMENT CACHE POISONING

- Attack introduced in 2013
- Leverages fragmentation for cache poisoning
- Follow-up fragments do not contain any identification (unlike the first fragment which contains the query ID and source port)
- Put records to poison in second (or later) fragment
- Irony: DNSSEC's large responses help facilitate this attack on non-validating DNS resolvers



# CONSEQUENCE: AVOID FRAGMENTATION!



# CONSEQUENCE: AVOID FRAGMENTATION!

- The DNS community wants to avoid fragmentation



# CONSEQUENCE: AVOID FRAGMENTATION!

- The DNS community wants to avoid fragmentation
- Three strategies:



# CONSEQUENCE: AVOID FRAGMENTATION!

- The DNS community wants to avoid fragmentation
- Three strategies:
  - Strip optional records from responses  
*Has some consequences for performance*



# CONSEQUENCE: AVOID FRAGMENTATION!

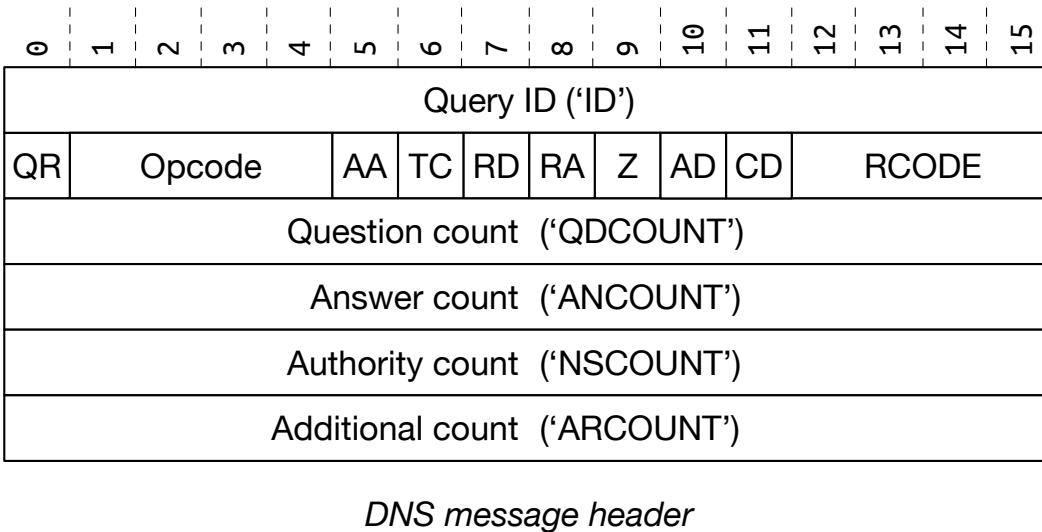
- The DNS community wants to avoid fragmentation
- Three strategies:
  - Strip optional records from responses  
*Has some consequences for performance*
  - Different cryptographic algorithms (ECC)  
*Smaller keys and signatures*



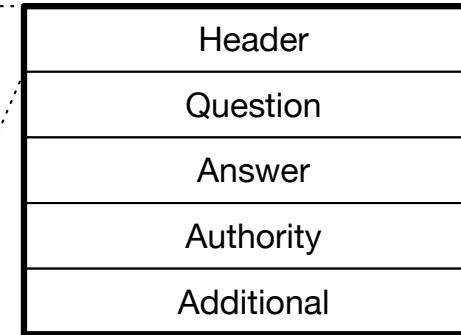
# CONSEQUENCE: AVOID FRAGMENTATION!

- The DNS community wants to avoid fragmentation
- Three strategies:
  - Strip optional records from responses  
*Has some consequences for performance*
  - Different cryptographic algorithms (ECC)  
*Smaller keys and signatures*
  - Collaborate on EDNS0 parameters that avoid fragmentation  
*DNS Flag Day (discussed later)*

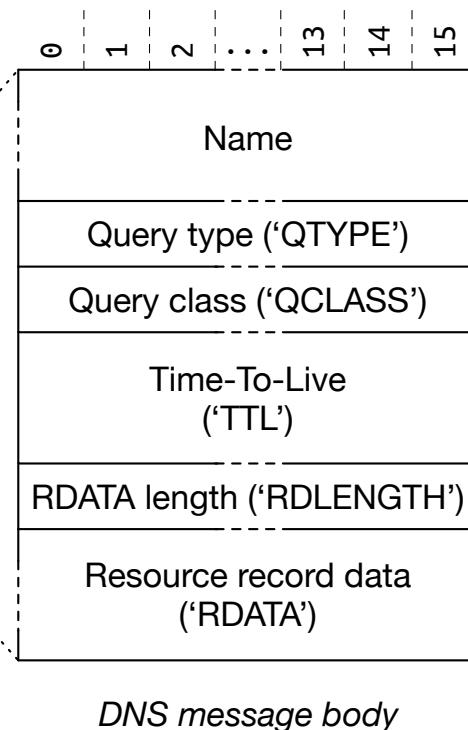
# STRIPPING OPTIONAL RECORDS



DNS message header

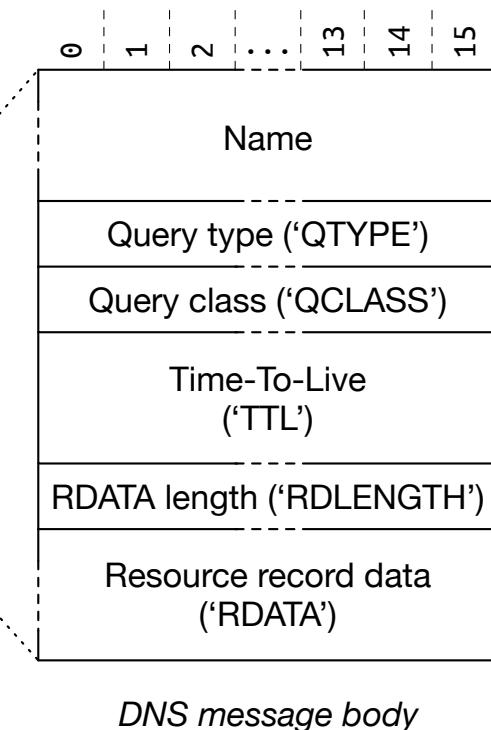
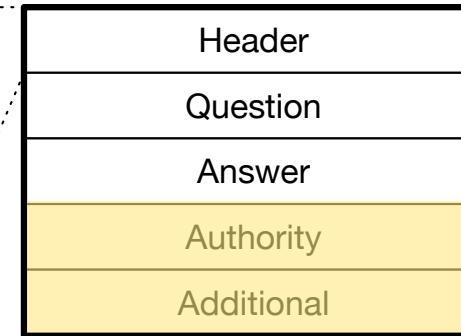
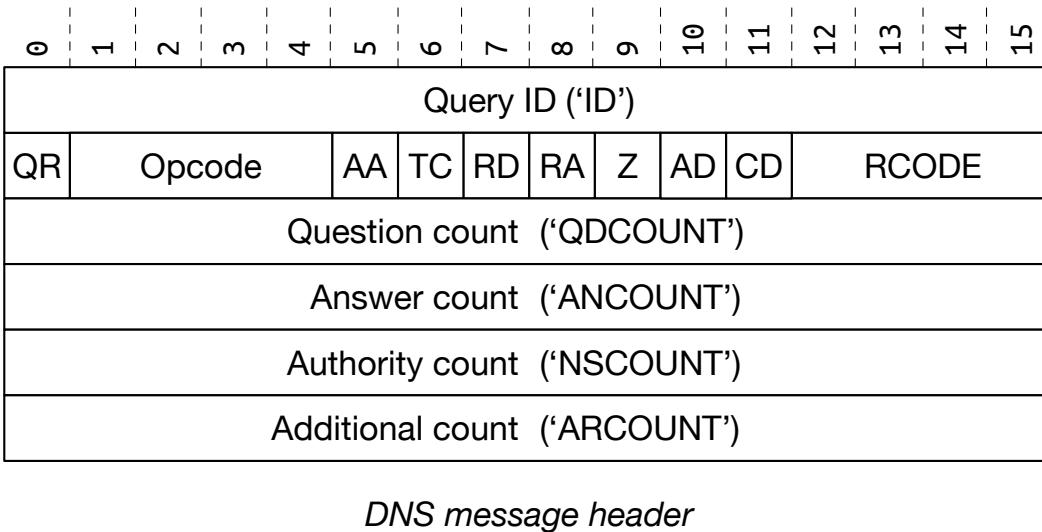


DNS message

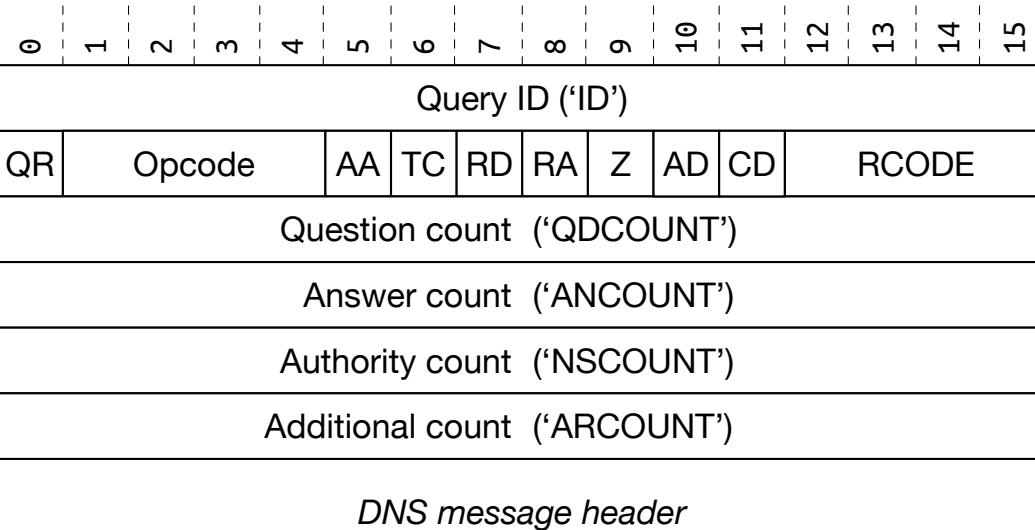


DNS message body

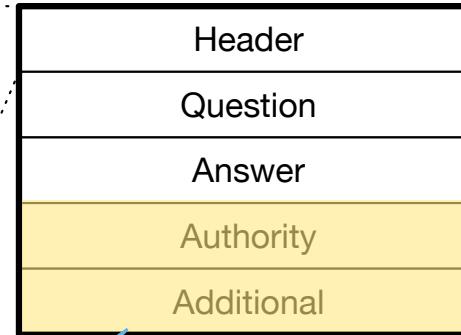
# STRIPPING OPTIONAL RECORDS



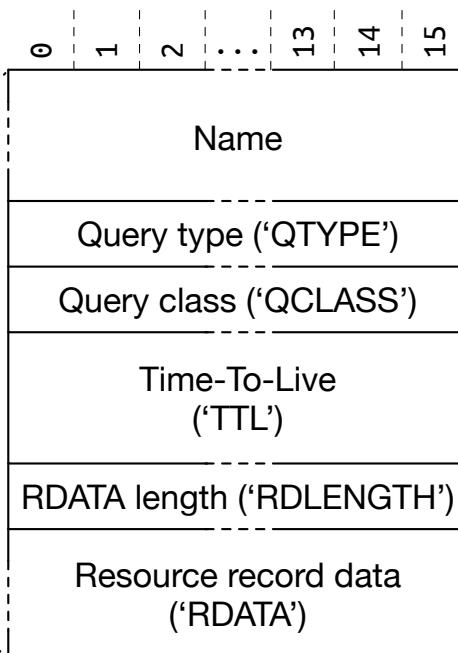
# STRIPPING OPTIONAL RECORDS



DNS message header



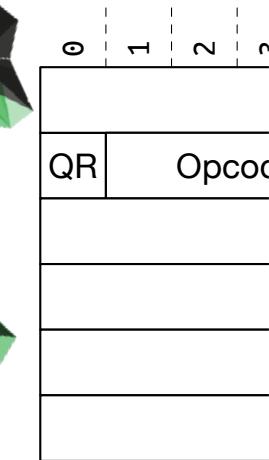
DNS message



DNS message body

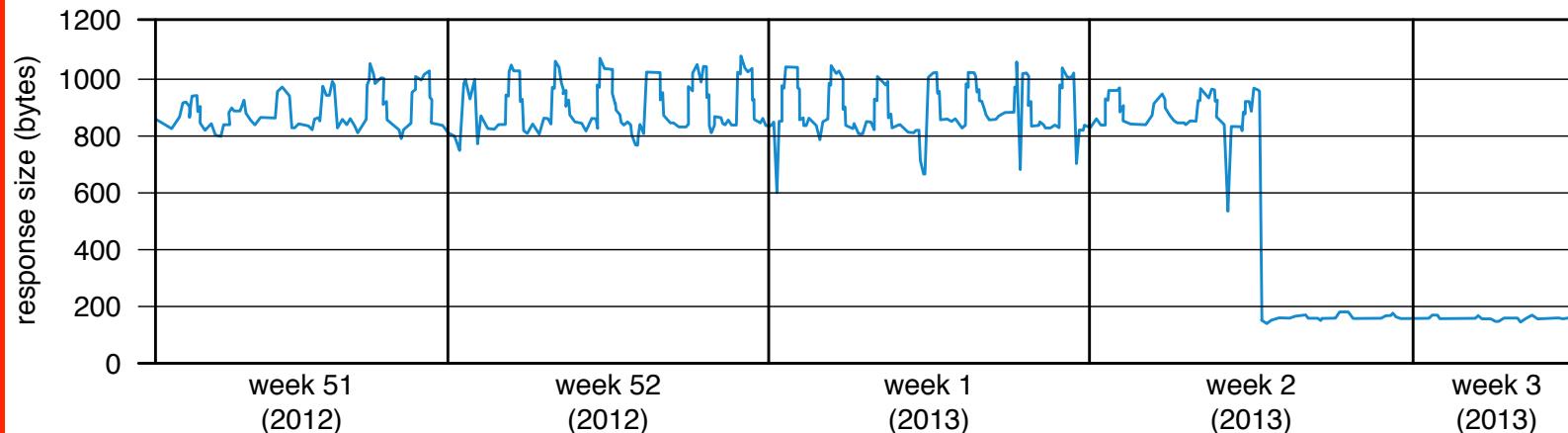
These sections are optional in many responses, stripping them reduces fragmentation significantly, as it removes records and signatures

# STRIPPING OPTIONAL RECORDS



These sections  
stripping them  
significantly,

Stripping these records vastly reduces response sizes



But there is also a performance impact  
(that is not well understood)

2 ... 13 14 15

Name

Query type ('QTYPE')

Query class ('QCLASS')

Time-To-Live ('TTL')

Record length ('RDLENGTH')

Source record data ('RDATA')

Message body

UNIVERSITY  
OF TWENTE.

# SWITCHING TO ELLIPTIC CURVE CRYPTO





# SWITCHING TO ELLIPTIC CURVE CRYPTO

- Original DNSSEC specifications assumed the use of RSA

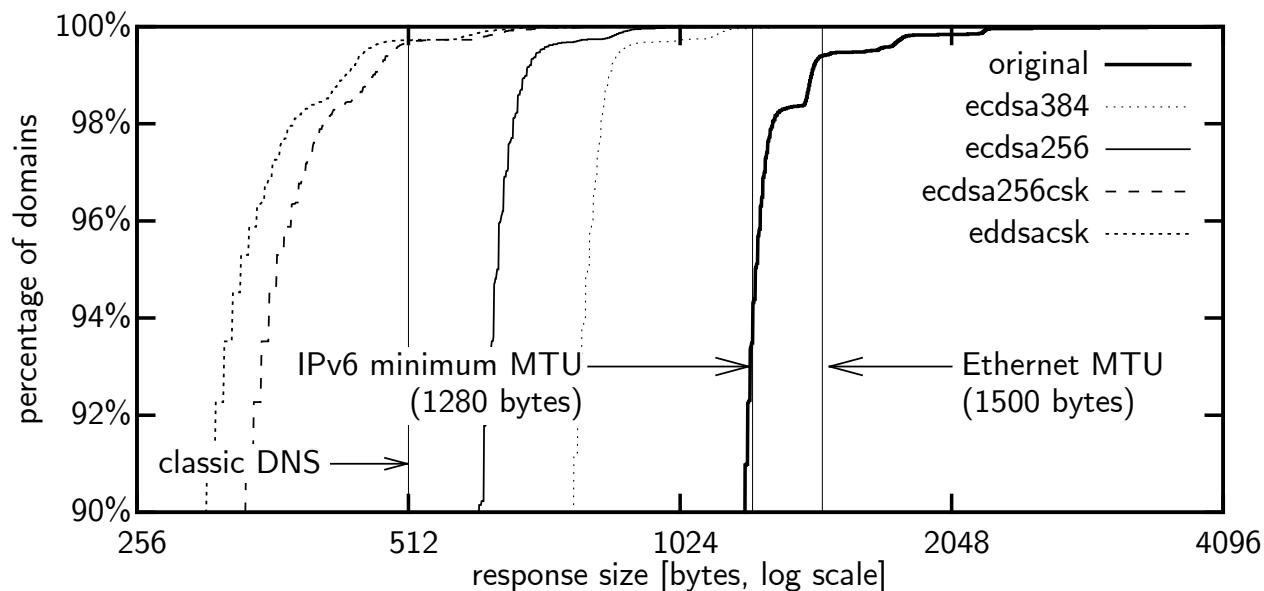


# SWITCHING TO ELLIPTIC CURVE CRYPTO

- Original DNSSEC specifications assumed the use of RSA
- ECC algorithms standardised for DNSSEC in 2012

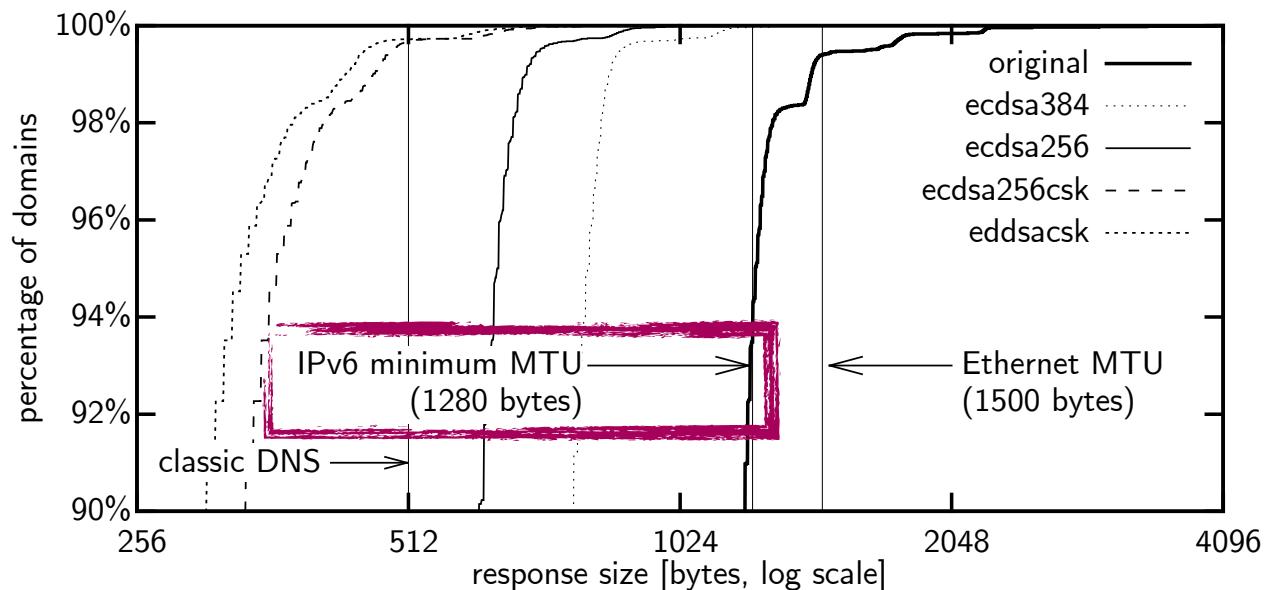
# SWITCHING TO ELLIPTIC CURVE CRYPTO

- Original DNSSEC specifications assumed the use of RSA
- ECC algorithms standardised for DNSSEC in 2012



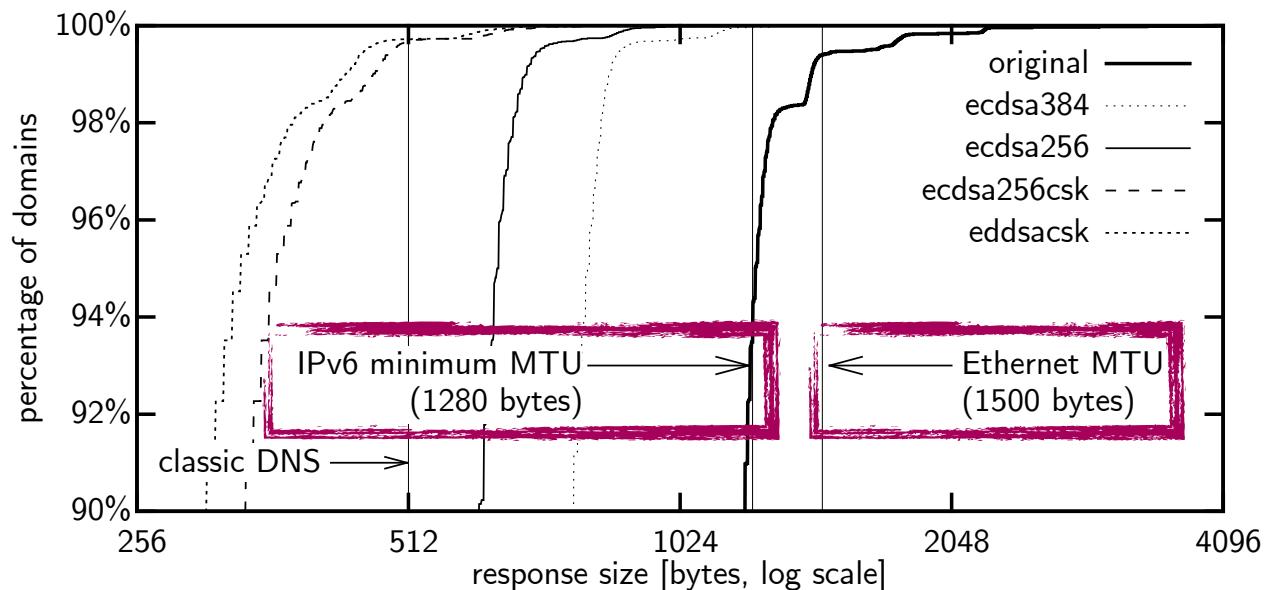
# SWITCHING TO ELLIPTIC CURVE CRYPTO

- Original DNSSEC specifications assumed the use of RSA
- ECC algorithms standardised for DNSSEC in 2012



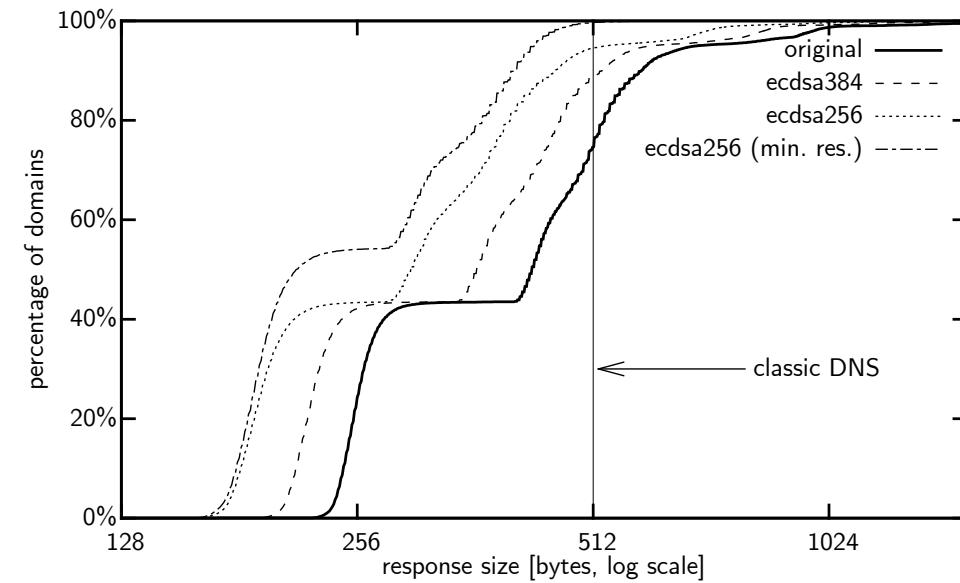
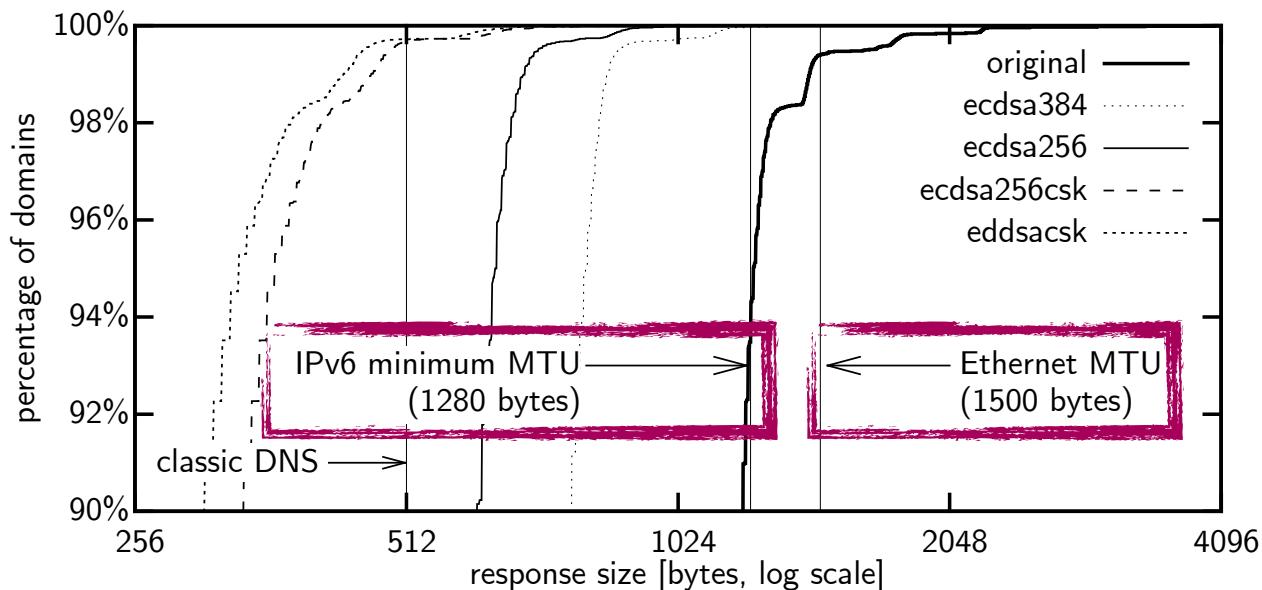
# SWITCHING TO ELLIPTIC CURVE CRYPTO

- Original DNSSEC specifications assumed the use of RSA
- ECC algorithms standardised for DNSSEC in 2012



# SWITCHING TO ELLIPTIC CURVE CRYPTO

- Original DNSSEC specifications assumed the use of RSA
- ECC algorithms standardised for DNSSEC in 2012



# ECC CHALLENGE: VALIDATION SPEED





# ECC CHALLENGE: VALIDATION SPEED

- ECC signature verification is much slower than RSA



# ECC CHALLENGE: VALIDATION SPEED

- ECC signature verification is much slower than RSA
- Verification occurs way more frequently than signing in DNSSEC → *what is the impact on operations?*



# ECC CHALLENGE: VALIDATION SPEED

- ECC signature verification is much slower than RSA
- Verification occurs way more frequently than signing in DNSSEC → *what is the impact on operations?*

ECC algorithm	OpenSSL version <sup>†</sup>	Compared to*			
		RSA		ECDSA	
		1024	2048	P-256	P-384
ECDSA P-256	0.9.8zh	27.5	8.4	-	-
	1.0.1f	26.0	7.9	-	-
	1.0.2e	11.5	3.6	-	-
ECDSA P-384	0.9.8zh	57.7	17.6	-	-
	1.0.1f	77.6	23.4	-	-
	1.0.2e	87.3	27.2	-	-
Ed25519	(1.0.2e) <sup>‡</sup>	7.9	2.5	0.7	0.1
Ed448	(1.0.2e) <sup>‡</sup>	23.4	7.3	2.0	0.3

\*the number means that the ECC algorithm is  $x$  times *slower*

<sup>†</sup>comparison of the ECC and RSA primitives for this OpenSSL version

<sup>‡</sup>independent implementations compared to this OpenSSL version

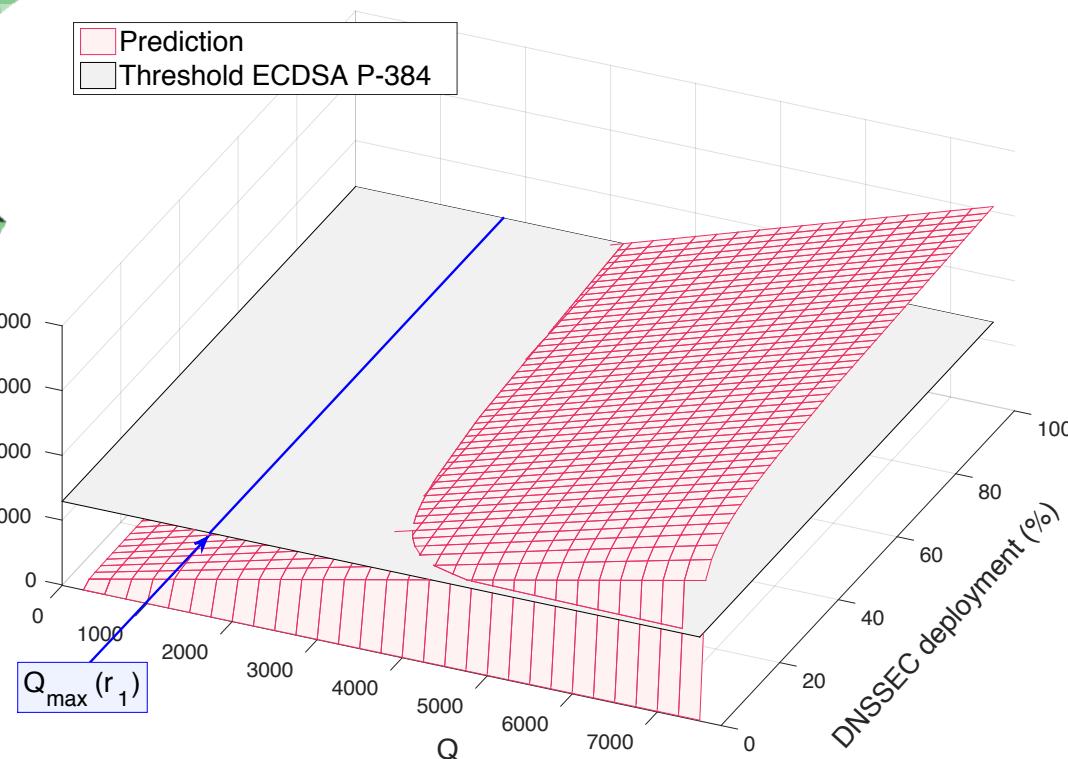


# OPERATIONAL IMPACT

- We modelled the impact of signature verification on DNSSEC operations. Conclusion: ECC use is feasible.

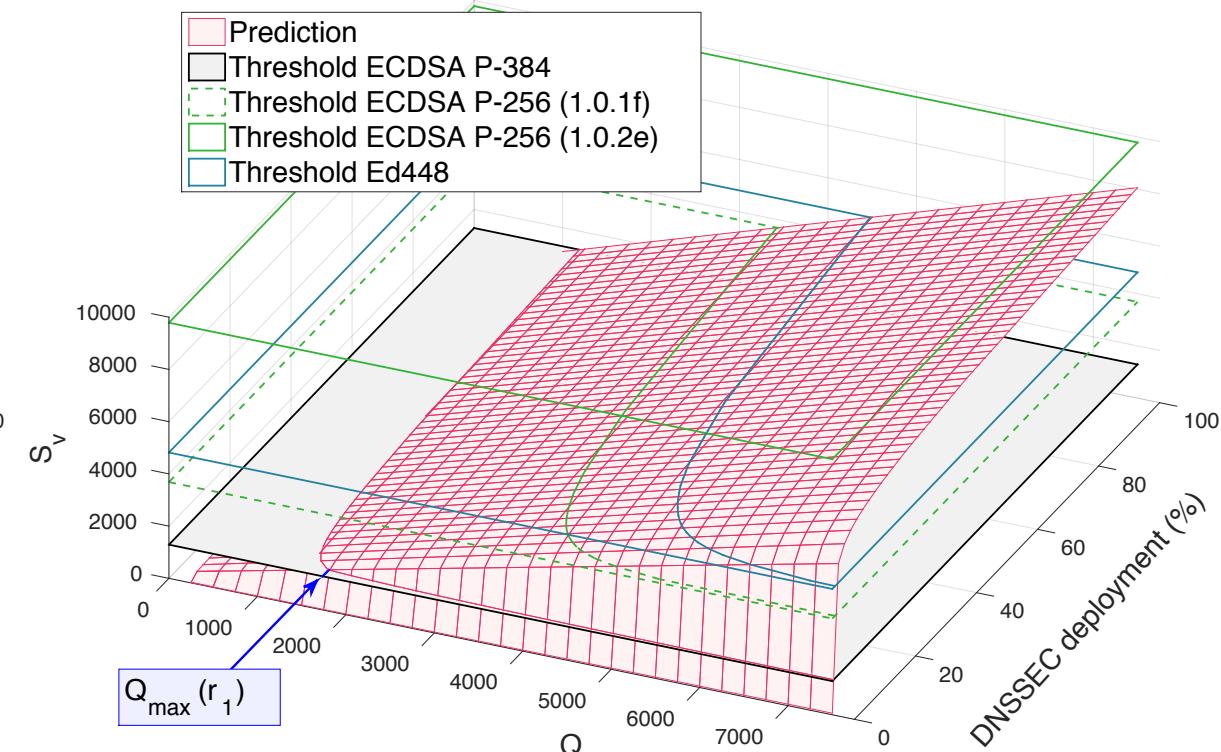
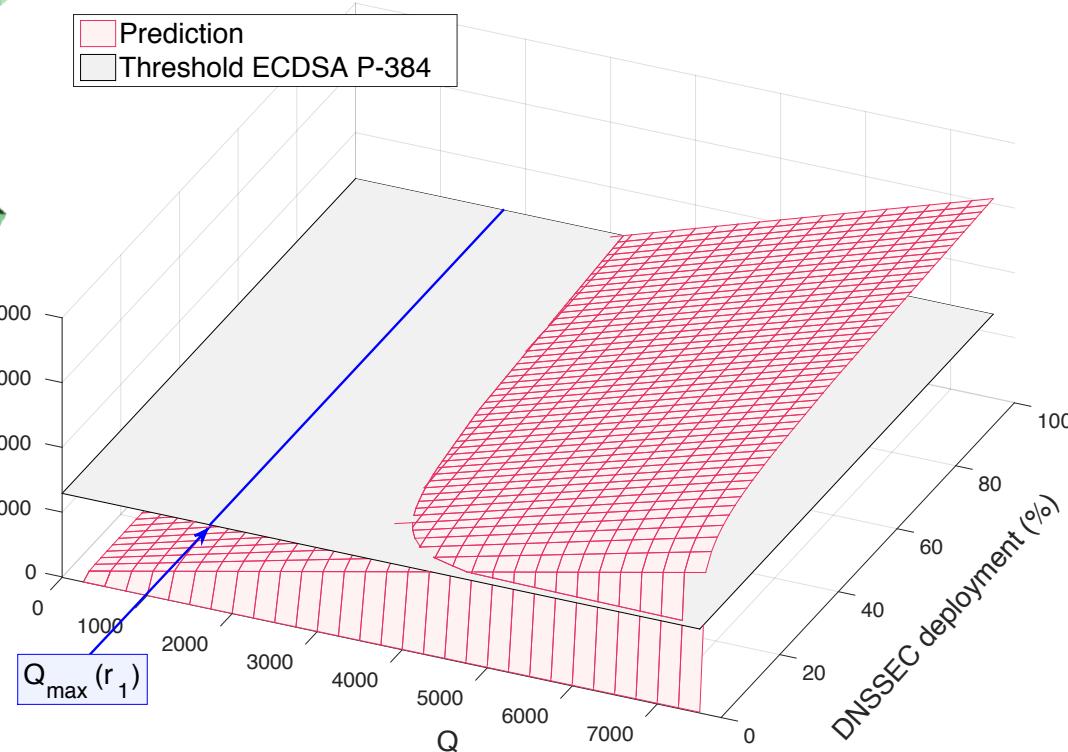
# OPERATIONAL IMPACT

- We modelled the impact of signature verification on DNSSEC operations. Conclusion: ECC use is feasible.



# OPERATIONAL IMPACT

- We modelled the impact of signature verification on DNSSEC operations. Conclusion: ECC use is feasible.



# DNS FLAG DAY 2020





# DNS FLAG DAY 2020

- Due to the operational problems with fragmentation, DNS vendors decided to work together

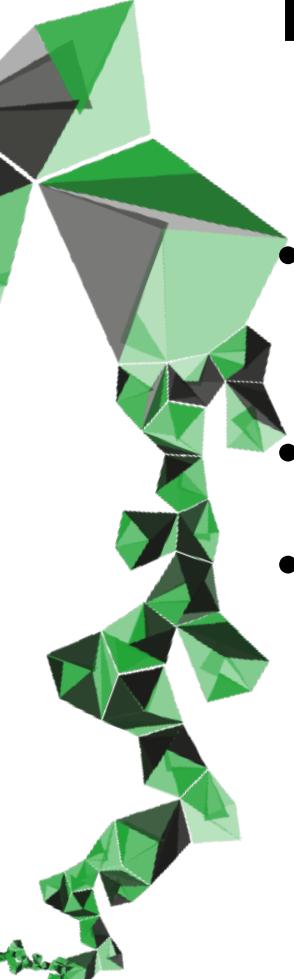




# DNS FLAG DAY 2020

- Due to the operational problems with fragmentation, DNS vendors decided to work together
- This resulted in DNS Flag Day 2020





# DNS FLAG DAY 2020

- Due to the operational problems with fragmentation, DNS vendors decided to work together
- This resulted in DNS Flag Day 2020
- All major open source vendors and several major DNS operators changed EDNS0 settings in software and operational resolvers





# DNS FLAG DAY 2020

- Due to the operational problems with fragmentation, DNS vendors decided to work together
- This resulted in DNS Flag Day 2020
- All major open source vendors and several major DNS operators changed EDNS0 settings in software and operational resolvers
- Result: maximum limit for UDP transport is now 1232 bytes





# DNS FLAG DAY 2020

- Due to the operational problems with fragmentation, DNS vendors decided to work together
- This resulted in DNS Flag Day 2020
- All major open source vendors and several major DNS operators changed EDNS0 settings in software and operational resolvers
- Result: maximum limit for UDP transport is now 1232 bytes
- Fallback to TCP otherwise





# TAKEAWAYS UNTIL NOW



# TAKEAWAYS UNTIL NOW

- DNSSEC increases the size of DNS messages



# TAKEAWAYS UNTIL NOW

- DNSSEC increases the size of DNS messages
- This causes both availability and security problems



# TAKEAWAYS UNTIL NOW

- DNSSEC increases the size of DNS messages
- This causes both availability and security problems
- The community has worked very hard to mitigate these issues



# TAKEAWAYS UNTIL NOW

- DNSSEC increases the size of DNS messages
- This causes both availability and security problems
- The community has worked very hard to mitigate these issues
- This has shaped the mindset of the community



# TAKEAWAYS UNTIL NOW

- DNSSEC increases the size of DNS messages
- This causes both availability and security problems
- The community has worked very hard to mitigate these issues
- This has shaped the mindset of the community
- **Consequence: a switch to quantum-safe algorithms has to take this mindset into account**



# WHAT ABOUT PQC IN DNSSEC?



# WHAT ABOUT PQC IN DNSSEC?

- I hope I convinced you signature size, public key size and verification speed are an issue



# WHAT ABOUT PQC IN DNSSEC?

- I hope I convinced you signature size, public key size and verification speed are an issue
- Dealing with larger signatures or keys may be unavoidable



# WHAT ABOUT PQC IN DNSSEC?

- I hope I convinced you signature size, public key size and verification speed are an issue
- Dealing with larger signatures or keys may be unavoidable
- Higher computational load for verification should be avoided



# WHAT ABOUT PQC IN DNSSEC?

- I hope I convinced you signature size, public key size and verification speed are an issue
- Dealing with larger signatures or keys may be unavoidable
- Higher computational load for verification should be avoided
- We need to balance out any changes in favour of a lower workload for DNS resolvers

# GREEN, ORANGE AND RED LINES





# GREEN, ORANGE AND RED LINES

Signature size

# GREEN, ORANGE AND RED LINES



Signature size

$\leq$  RSA 2048



# GREEN, ORANGE AND RED LINES

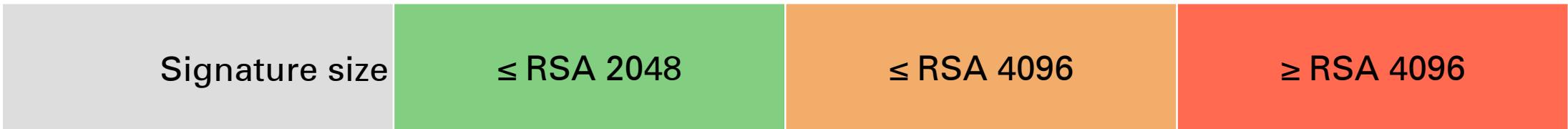
Signature size

$\leq$  RSA 2048

$\leq$  RSA 4096



# GREEN, ORANGE AND RED LINES



# GREEN, ORANGE AND RED LINES

Signature size	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ RSA 4096
Public key size			

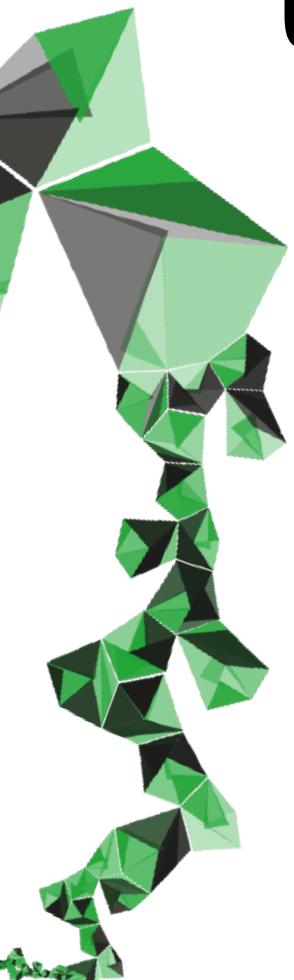


# GREEN, ORANGE AND RED LINES

Signature size	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ RSA 4096
Public key size	$\leq$ RSA 2048		

# GREEN, ORANGE AND RED LINES

Signature size	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ RSA 4096
Public key size	$\leq$ RSA 2048	$\leq$ RSA 4096	



# GREEN, ORANGE AND RED LINES

Signature size	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ RSA 4096
Public key size	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ 1 MByte*



# GREEN, ORANGE AND RED LINES

Signature size	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ RSA 4096
Public key size	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ 1 MByte*
Signing speed <i>(single CPU core)</i>			



# GREEN, ORANGE AND RED LINES

Signature size	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ RSA 4096
Public key size	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ 1 MByte*
Signing speed <i>(single CPU core)</i>	$\leq$ RSA 2048		



# GREEN, ORANGE AND RED LINES

Signature size	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ RSA 4096
Public key size	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ 1 MByte*
Signing speed <i>(single CPU core)</i>	$\leq$ RSA 2048	$\leq$ RSA 4096	



# GREEN, ORANGE AND RED LINES

Signature size	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ RSA 4096
Public key size	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ 1 MByte*
Signing speed <i>(single CPU core)</i>	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ 50ms*



# GREEN, ORANGE AND RED LINES

Signature size	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ RSA 4096
Public key size	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ 1 MByte*
Signing speed (single CPU core)	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ 50ms*
Validation speed (single CPU core)			



# GREEN, ORANGE AND RED LINES

Signature size	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ RSA 4096
Public key size	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ 1 MByte*
Signing speed (single CPU core)	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ 50ms*
Validation speed (single CPU core)	$\leq$ ECDSA P-256		



# GREEN, ORANGE AND RED LINES

Signature size	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ RSA 4096
Public key size	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ 1 MByte*
Signing speed (single CPU core)	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ 50ms*
Validation speed (single CPU core)	$\leq$ ECDSA P-256	$\leq$ ECDSA P-384	



# GREEN, ORANGE AND RED LINES

Signature size	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ RSA 4096
Public key size	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ 1 MByte*
Signing speed (single CPU core)	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ 50ms*
Validation speed (single CPU core)	$\leq$ ECDSA P-256	$\leq$ ECDSA P-384	$\geq$ ECDSA P-384



# GREEN, ORANGE AND RED LINES

Signature size	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ RSA 4096
Public key size	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ 1 MByte*
Signing speed (single CPU core)	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ 50ms*
Validation speed (single CPU core)	$\leq$ ECDSA P-256	$\leq$ ECDSA P-384	$\geq$ ECDSA P-384
Other restrictions			



# GREEN, ORANGE AND RED LINES

Signature size	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ RSA 4096
Public key size	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ 1 MByte*
Signing speed (single CPU core)	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ 50ms*
Validation speed (single CPU core)	$\leq$ ECDSA P-256	$\leq$ ECDSA P-384	$\geq$ ECDSA P-384
Other restrictions		Stateful signing	



# GREEN, ORANGE AND RED LINES

Signature size	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ RSA 4096
Public key size	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ 1 MByte*
Signing speed (single CPU core)	$\leq$ RSA 2048	$\leq$ RSA 4096	$\geq$ 50ms*
Validation speed (single CPU core)	$\leq$ ECDSA P-256	$\leq$ ECDSA P-384	$\geq$ ECDSA P-384
Other restrictions		Stateful signing	Specialised hardware



# STUDY OF NIST ROUND 3 ENTRIES

Prio	Requirement	Good	Accepted Conditionally
#1	Signature Size	$\leq 1,232$ bytes	—
#2	Validation Speed	$\geq 1,000$ sig/s	—
#3	Key Size	$\leq 64$ kilobytes	$> 64$ kilobytes
#4	Signing Speed	$\geq 100$ sig/s	—

# SUITABLE ROUND 3 ALGORITHMS

Algorithm	NIST Verdict	Approach	Private key	Public key	Signature	Sign/s	Verify/s
Crystals-Dilithium-II [29]	Finalist	Lattice	2.8kB	1.2kB	2.0kB		
Falcon-512 [31]	Finalist	Lattice	57kB	0.9kB	0.7kB	3,307	20,228
Rainbow- $I_a$ [56]	Finalist	Multivariate	101kB	158kB	66B	8,332	11,065
RedGeMSS128 [16]	Candidate	Multivariate	16B	375kB	35B	545	10,365
Sphincs <sup>+</sup> -Haraka-128s [11]	Candidate	Hash	64B	32B	8kB		
Picnic-L1-FS [17]	Candidate	Hash	16B	32B	34kB		
Picnic2-L1-FS [17]	Candidate	Hash	16B	32B	14kB		
EdDSA-Ed25519 [12]		Elliptic curve	64B	32B	64B	25,935	7,954
ECDSA-P256 [12]		Elliptic curve	96B	64B	64B	40,509	13,078
RSA-2048 [12]		Prime	2kB	0.3kB	0.3kB	1,485	49,367

# TAKEAWAYS NIST ROUND 3 ENTRIES





# TAKEAWAYS NIST ROUND 3 ENTRIES

- Thinking back to our green, orange and red lines:  
No single NIST round 3 entrant is ideal



# TAKEAWAYS NIST ROUND 3 ENTRIES

- Thinking back to our green, orange and red lines:  
No single NIST round 3 entrant is ideal
- If we consider signature size as highest priority, two most suitable candidates are Rainbow and RedGeMSS



# TAKEAWAYS NIST ROUND 3 ENTRIES

- Thinking back to our green, orange and red lines:  
No single NIST round 3 entrant is ideal
- If we consider signature size as highest priority, two most suitable candidates are Rainbow and RedGeMSS
- Challenge for both: public key size (does not fit into a DNS datagram, even over TCP the limit is 64KiB)



# TAKEAWAYS NIST ROUND 3 ENTRIES

- Thinking back to our green, orange and red lines:  
No single NIST round 3 entrant is ideal
- If we consider signature size as highest priority, two most suitable candidates are Rainbow and RedGeMSS
- Challenge for both: public key size (does not fit into a DNS datagram, even over TCP the limit is 64KiB)
- *And we only considered level 1 security (128-bit)*



# WHAT ABOUT ISOGENIES?



# WHAT ABOUT ISOGENIES?

- *Caveat*: I am not a cryptographer, nor a mathematician



# WHAT ABOUT ISOGENIES?

- *Caveat*: I am not a cryptographer, nor a mathematician
- Some observations on navigating the literature:



# WHAT ABOUT ISOGENIES?

- *Caveat*: I am not a cryptographer, nor a mathematician
- Some observations on navigating the literature:
  - No uniform metrics for public key and signature sizes



# WHAT ABOUT ISOGENIES?

- *Caveat*: I am not a cryptographer, nor a mathematician
- Some observations on navigating the literature:
  - No uniform metrics for public key and signature sizes
  - No uniform single-core performance metrics (sometimes wall clock, sometimes cycles)



# WHAT ABOUT ISOGENIES?

- *Caveat*: I am not a cryptographer, nor a mathematician
- Some observations on navigating the literature:
  - No uniform metrics for public key and signature sizes
  - No uniform single-core performance metrics (sometimes wall clock, sometimes cycles)
  - It all feels “very new”
    - sufficiently cryptoanalysed for “the big screen”?



# WHAT ABOUT ISOGENIES?

- *Caveat*: I am not a cryptographer, nor a mathematician
- Some observations on navigating the literature:
  - No uniform metrics for public key and signature sizes
  - No uniform single-core performance metrics (sometimes wall clock, sometimes cycles)
  - It all feels “very new”
    - sufficiently cryptoanalysed for “the big screen”?
  - CSIDH, CSI-FiSh, SeaSign, SQISign, where are we headed?

# STUMBLING AROUND IN THE DARK





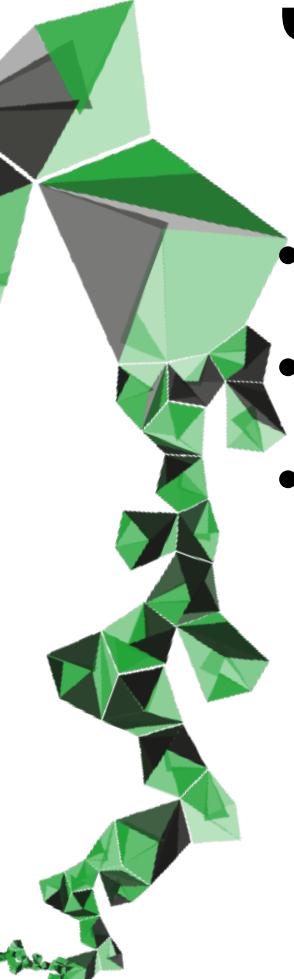
# STUMBLING AROUND IN THE DARK

- I looked at one scheme with “DNSSEC glasses on”



# STUMBLING AROUND IN THE DARK

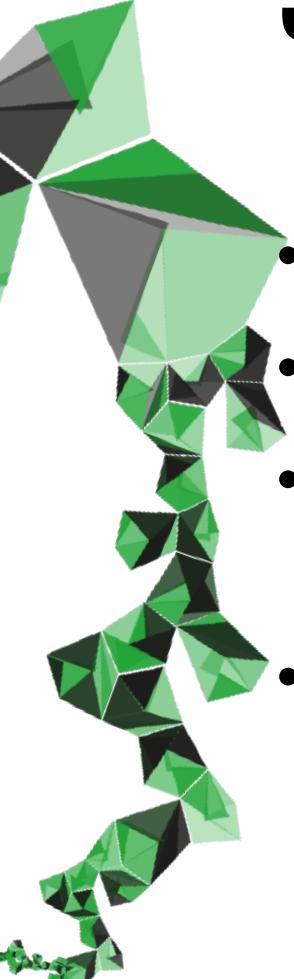
- I looked at one scheme with “DNSSEC glasses on”
- SQISign, as it looked attractive



# STUMBLING AROUND IN THE DARK

- I looked at one scheme with “DNSSEC glasses on”
- SQISign, as it looked attractive
- Key and signature size at NIST level 1 look good!

Signature size	204 bytes
Public key size	64 bytes



# STUMBLING AROUND IN THE DARK

- I looked at one scheme with “DNSSEC glasses on”
- SQISign, as it looked attractive
- Key and signature size at NIST level 1 look good!
- Signing speed is a disaster\*

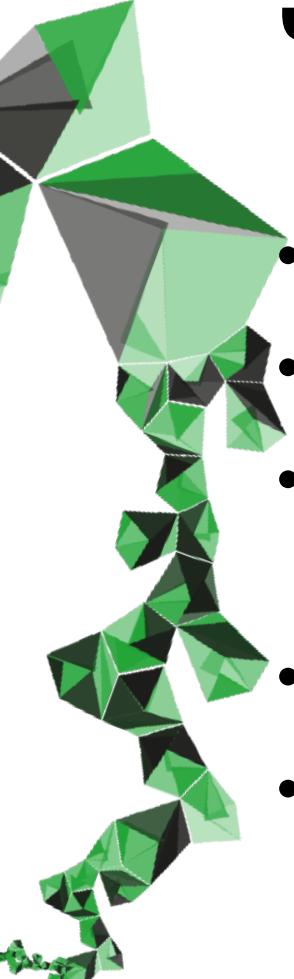
Signature size	204 bytes
Public key size	64 bytes
Signing speed (single CPU core)	2.3s



# STUMBLING AROUND IN THE DARK

- I looked at one scheme with “DNSSEC glasses on”
- SQISign, as it looked attractive
- Key and signature size at NIST level 1 look good!
- Signing speed is a disaster\*
- Validation is ~40x slower than ECDSA P-384 (so too slow)\*

Signature size	204 bytes
Public key size	64 bytes
Signing speed (single CPU core)	2.3s
Validation speed (single CPU core)	42ms

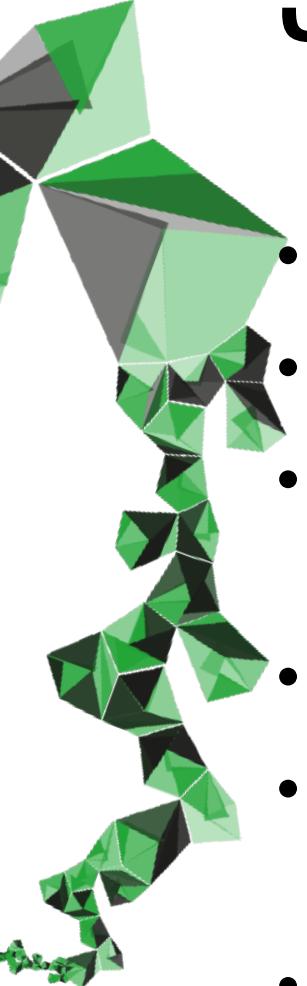


# STUMBLING AROUND IN THE DARK

- I looked at one scheme with “DNSSEC glasses on”
- SQISign, as it looked attractive
- Key and signature size at NIST level 1 look good!
- Signing speed is a disaster\*
- Validation is ~40x slower than ECDSA P-384 (so too slow)\*

Signature size	204 bytes
Public key size	64 bytes
Signing speed (single CPU core)	2.3s
Validation speed (single CPU core)	42ms

\*may work for DNS root



# STUMBLING AROUND IN THE DARK

- I looked at one scheme with “DNSSEC glasses on”
- SQISign, as it looked attractive
- Key and signature size at NIST level 1 look good!
- Signing speed is a disaster\*
- Validation is ~40x slower than ECDSA P-384 (so too slow)\*
- Will this improve?

Signature size	204 bytes
Public key size	64 bytes
Signing speed (single CPU core)	2.3s
Validation speed (single CPU core)	42ms

\*may work for DNS root



# ONGOING WORK



# ONGOING WORK

- **Prototyping NIST candidates in DNSSEC** and assessing the operational impact (ongoing Ph.D. project at UWaterloo and M.Sc. project at UTwente)



# ONGOING WORK

- **Prototyping NIST candidates in DNSSEC** and assessing the operational impact (ongoing Ph.D. project at UWaterloo and M.Sc. project at UTwente)
- **Re-thinking DNSSEC signing** based on hash-based signatures and Merkle trees (ongoing M.Sc. project at UTwente)



# ONGOING WORK

- **Prototyping NIST candidates in DNSSEC** and assessing the operational impact (ongoing Ph.D. project at UWaterloo and M.Sc. project at UTwente)
- **Re-thinking DNSSEC signing** based on hash-based signatures and Merkle trees (ongoing M.Sc. project at UTwente)
- **Drafting an RFC for hash-based signatures** in DNSSEC, as a “safe fallback” under all conditions (collaborative effort)



# ONGOING WORK

- **Prototyping NIST candidates in DNSSEC** and assessing the operational impact (ongoing Ph.D. project at UWaterloo and M.Sc. project at UTwente)
- **Re-thinking DNSSEC signing** based on hash-based signatures and Merkle trees (ongoing M.Sc. project at UTwente)
- **Drafting an RFC for hash-based signatures** in DNSSEC, as a “safe fallback” under all conditions (collaborative effort)
- **Drafting an RFC on alternative transports** for DNSSEC in a post-quantum world (collaborative effort)

# QUESTIONS/DISCUSSION



# REFERENCES

- M. Müller, J. de Jong, M. van Heesch, B. Overeinder, R. van Rijswijk-Deij. Retrofitting Post-Quantum Cryptography in Internet Protocols: A Case Study of DNSSEC. In: ACM SIGCOMM Computer Communication Review (CCR) 50 (4), 2020
- R. van Rijswijk-Deij, K. Hageman, A. Sperotto and A. Pras. The Performance Impact of Elliptic Curve Cryptography on DNSSEC Validation. In IEEE/ACM Transactions on Networking, Volume 25, Issue 2 (April 2017)
- R. van Rijswijk-Deij, A. Sperotto and A. Pras. Making the Case for Elliptic Curves in DNSSEC. ACM SIGCOMM Computer Communication Review, Volume 45, Issue 5 (October 2015)
- R. van Rijswijk-Deij, A. Sperotto and A. Pras. DNSSEC and Its Potential for DDoS Attacks - a Comprehensive Measurement Study. Proceedings of ACM IMC 2014. Vancouver, BC, Canada: ACM Press
- G. van den Broek, R. van Rijswijk-Deij, A. Sperotto and A. Pras. DNSSEC Meets Real World: Dealing with Unreachability Caused by Fragmentation. IEEE Communications Magazine, Vol. 52 (April 2014), pp. 154-160
- A. Herzberg and H. Shulman, "Fragmentation Considered Poisonous, or: One-domain-to-rule-them-all.org," in 2013 IEEE Conference on Communications and Network Security, CNS 2013, 2013, pp. 224–232
- N. Weaver, C. Kreibich, B. Nechaev, and V. Paxson, "Implications of Netalyzr's DNS Measurements," in Proceedings of the SATIN 2011 Workshop. London, UK: NPL, 2011.