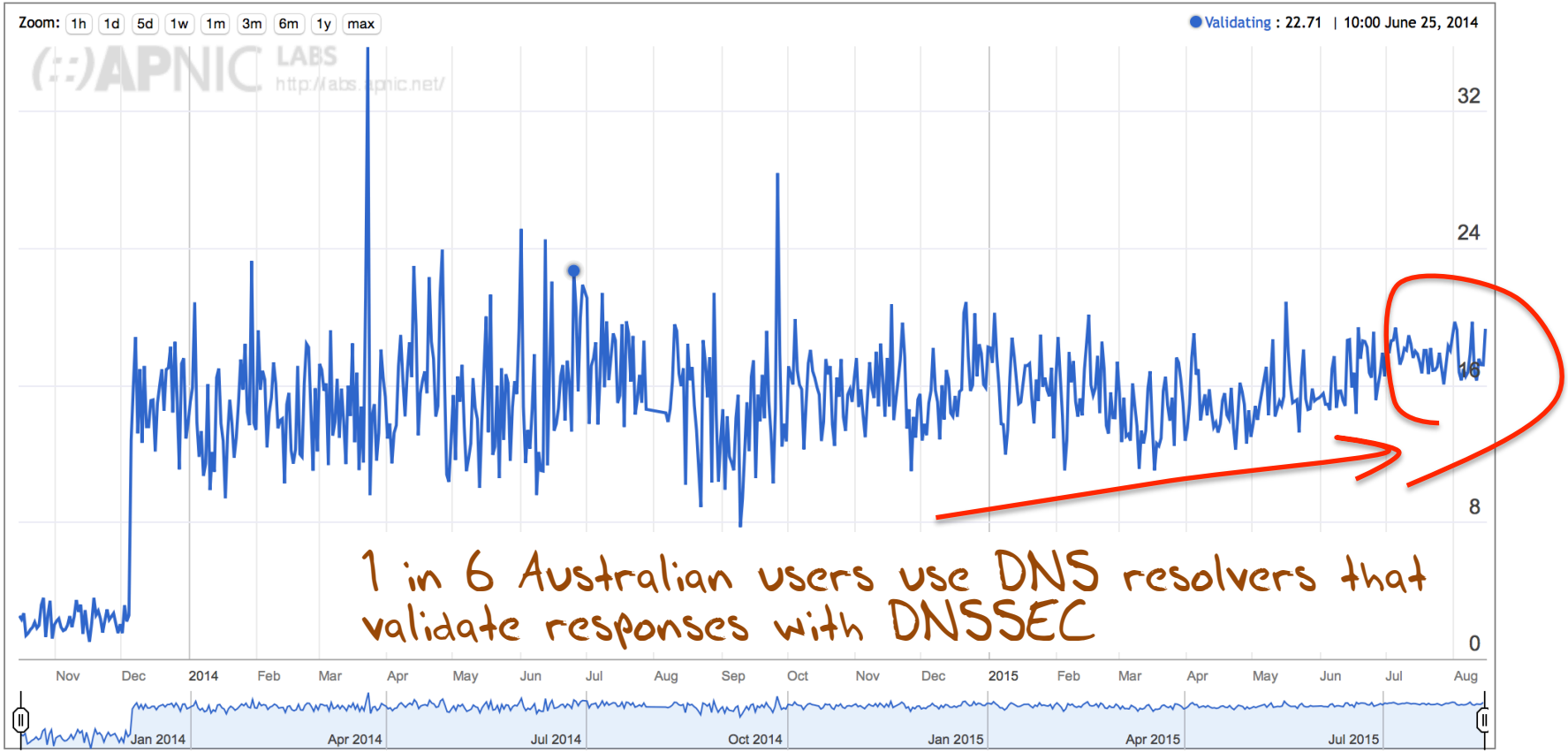
# Rolling the Keys of the DNS Root Zone

Geoff Huston
APNIC Labs

# DNSSEC in AU

## Use of DNSSEC Validation for Australia (AU)




stats.labs.apnic.net/dnssec/AU

# DNSSEC in AU

| ASN | AS Name | DNSSEC Validates | Samples |
|---|---|---|---|
| AS38719 | AUSTDOM-AS-AP Aust Domains International Pty Ltd. | 100.00% | 809 |
| AS37978 | NETTAS-AS-AP Networking Tasmania | 98.41% | 189 |
| AS38484 | VIRGIN-BROADBAND-AS-AP Virgin Broadband VISP | 97.41% | 850 |
| AS7477 | TEREDONN-AS-AP SkyMesh Pty Ltd | 96.49% | 542 |
| AS18390 | SPIN-INTERNET-AP Spin Internet Service | 93.58% | 374 |
| AS45920 | SKYMESH-AS-AP SkyMesh Pty Ltd | 91.96% | 112 |
| AS17899 | ASN-ACN ASN-ACN | 91.35% | 266 |
| AS9288 | COMCEN-AS-AP Com-Cen Pty Ltd | 89.03% | 1176 |
| AS10113 | EFTEL-AS-AP Eftel Limited. | 88.38% | 1102 |
| AS17551 | DCSINTERNET-AS-AP DCS Internet | 88.00% | 375 |
| AS9297 | COLOCITY-AS-AP Colocity Pty Ltd | 87.06% | 286 |
| AS4804 | MPX-AS Microplex PTY LTD | 85.92% | 98658 |
| AS17907 | NUSKOPE NuSkope Pty. Ltd. | 84.92% | 358 |
| AS45510 | TELCOINABOX-AU Level 10, 9 Hunter Street | 83.74% | 3186 |
| AS23963 | BORDERNET-AU-AP Bordernet Internet Pty Ltd | 81.08% | 74 |
| AS10143 | EXETEL-AS-AP Exetel Pty Ltd | 70.78% | 6632 |
| AS58950 | NOVATEL-AS-AP Novatel Telephony Pty Ltd | 67.14% | 140 |
| AS9723 | ISEEK-AS-AP ISEEK Ltd | 63.67% | 289 |
| AS56086 | ITALK-AU iTalkBB Australia | 60.00% | 125 |
| AS17659 | MAINT-AU-SPEEDCASTAUSTRALIA NewSat Ltd | 52.26% | 155 |
| AS38790 | SPIRIT-TELECOM Spirit Telecom (Australia) Pty Ltd | 49.59% | 363 |
| AS17734 | IVOISYS-AS-AU iVoisys Pty Ltd, Sydney, Australia | 37.31% | 67 |
| AS17829 | IDL-AS-AP IDL Autonomous system | 33.67% | 98 |
| AS9503 | FX-PRIMARY-AS FX Networks Limited | 32.52% | 1593 |
| AS55923 | HARBOURSAT-AS-AP Harbour IT Pty Ltd | 30.27% | 261 |
| AS4826 | VOCUS-BACKBONE-AS Vocus Connect International Backbone | 29.90% | 816 |
| AS45654 | BI-AS-AP Internet Services | 23.53% | 204 |
| AS17978 | SERVCORP SERVCORP AUSTRALIAN HOLDINGS LTD | 23.33% | 60 |

# Why is this relevant?

# Why is this relevant?

Because the root zone managers are preparing to roll the DNS Root Zone Trust Anchor Key

(and in the worst case that may break your DNS service!)

# Five Years Ago

ICANN's First DNSSEC Key Ceremony for the Root Zone

The global deployment of Domain Name System Security Extensions (DNSSEC) will achieve an important milestone on June 16, 2010 as ICANN hosts the first production DNSSEC key ceremony in a high security data centre in Culpeper, VA, outside of Washington, DC.

ars technica



MAIN MENU · MY STORIES: 25 · FORUMS · SUBSCRIBE · JOBS · ARS CONSORTIUM

RISK ASSESSMENT / SECURITY & HACKTIVISM

## DNS root zone finally signed, but security battle not over

The root of the DNS hierarchy is now protected with a cryptographic signature …

by **Iljitsch van Beijnum** - Jul 16, 2010 11:28pm CEST



Yesterday, the DNS root zone was signed. This is an important step in the deployment of DNSSEC, the mechanism that will finally secure the DNS against manipulation by malicious third parties.

, VA - location of first DNSSEC key signing ceremony

## Schneier on Security

Blog · Newsletter · Books · Essays · News · Schedule · Crypto · About Me

← Pork-Filled Counter-Islamic Bomb Device

Security Vulnerabilities of Smart Electricity Meters →

### DNSSEC Root Key Split Among Seven People

The DNSSEC root key has been divided among seven people:

> Part of ICANN's security scheme is the Domain Name System Security, a security protocol that ensures Web sites are registered and "signed" (this is the security measure built into the Web that ensures when you go to a URL you arrive at a real site and not an identical pirate site). Most major servers are a part of DNSSEC, as it's known, and during a major international attack, the system might sever connections between important servers to contain the damage.

# The Eastern KSK Repository

Secure data center in Culpeper, VA - location of first DNSSEC key signing ceremony

# The Western KSK Repository


El Segundo, California *

* No – that's not really a picture of the the El Segundo KSK repository facilitie

# The Ultra Secret Third KSK Repository in Amsterdam

# KSK?

- The Root Zone Key Signing Key signs the DNSKEY RR set of the root zone
  - The Zone Signing Key (ZSK) signs the individual root zone entries
- The KSK Public Key is used as the DNSSEC Validation trust anchor
  - It is copied everywhere as “configuration data”

# Five Years Ago…

Root DNSSEC Design Team

F. Ljunggren
Kirei
T. Okubo
VeriSign
R. Lamb
ICANN
J. Schlyter
Kirei
May 21, 2010

DNSSEC Practice Statement for the Root Zone KSK Operator

Abstract

This document is the DNSSEC Practice Statement (DPS) for the Root Zone Key Signing Key (KSK) Operator. It states the practices and provisions that are used to provide Root Zone Key Signing and Key Distribution services. These include, but are not limited to: issuing, managing, changing and distributing DNS keys in accordance with the specific requirements of the U.S. Department of Commerce.



6.3. Signature format

The cryptographic hash function used in conjunction with the signing algorithm is required to be sufficiently resistant to preimage attacks during the time in which the signature is valid.

The RZ KSK signatures will be generated by encrypting SHA-256 hashes using RSA [RFC5702].

6.4. Zone signing key roll-over

ZSK rollover is carried out quarterly automatically by the Root Zone ZSK Operator's system as described in the Root Zone ZSK Operator's DPS.

6.5. Key signing key roll-over

Each RZ KSK will be scheduled to be rolled over through a key ceremony as required, or after 5 years of operation.

RZ KSK roll-over is scheduled to facilitate automatic updates of resolvers' Trust Anchors as described in RFC 5011 [RFC5011].

After a RZ KSK has been removed from the key set, it will be retained after its operational period until the next scheduled key ceremony, when the private component will be destroyed in accordance with section 5.2.10.

# Five Years Ago…

Root DNSSEC Design Team

F. Ljunggren
Kirei
T. Okubo
VeriSign
R. Lamb
ICANN
J. Schlyter
Kirei
May 21, 2010

DNSSEC Practice Statement for the Root Zone KSK Operator

Abstract

This document is the DNSSEC Practice Statement (DPS) for the Root Zone Key S and provisions Key Distributi issuing, m dance with the s rce.


May 2010

6.3. Signatu

The crypto signing algorithm ge attacks du

The RZ KSK 6 hashes using RSA

6.4. Zone si

ZSK rollov oot Zone ZSK Operat ator's DPS.

6.5. Key sig

Each RZ KSK will be scheduled to be rolled over through a key ceremony as required, or after 5 years of operation.

RZ KSK roll-over is scheduled to facilitate automatic updates of resolvers' Trust Anchors as described in RFC 5011 [RFC5011].

After a RZ KSK has been removed from the key set, it will be retained after its operational period until the next scheduled key ceremony, when the private component will be destroyed in accordance with section 5.2.10.

# The Cast of Actors

- Root Zone Management Partners:
  - Internet Corporation for Assigned Names and Numbers (ICANN)
  - National Telecommunications and Information Administration, US Department of Commerce (NTIA)
  - Verisign
- External Design Team for KSK Roll

# Approach

- ICANN Public Consultation – 2012
- Detailed Engineering Study - 2013
- SSAC Study (SAC-063) - 2013
- KSK Roll Design Team - 2015

# 2015 Design Team Milestones

- January – June:
  Study, discuss, measure, ponder, discuss some more
- August
  – Present a draft report for ICANN Public Comment
  https://www.icann.org/public-comments/root-ksk-2015-08-06-en
  (comment close 15th September 2015)
- September
  – Prepare final report
- Pass to the Root Zone Management Partners who then will develop an operational plan and execute

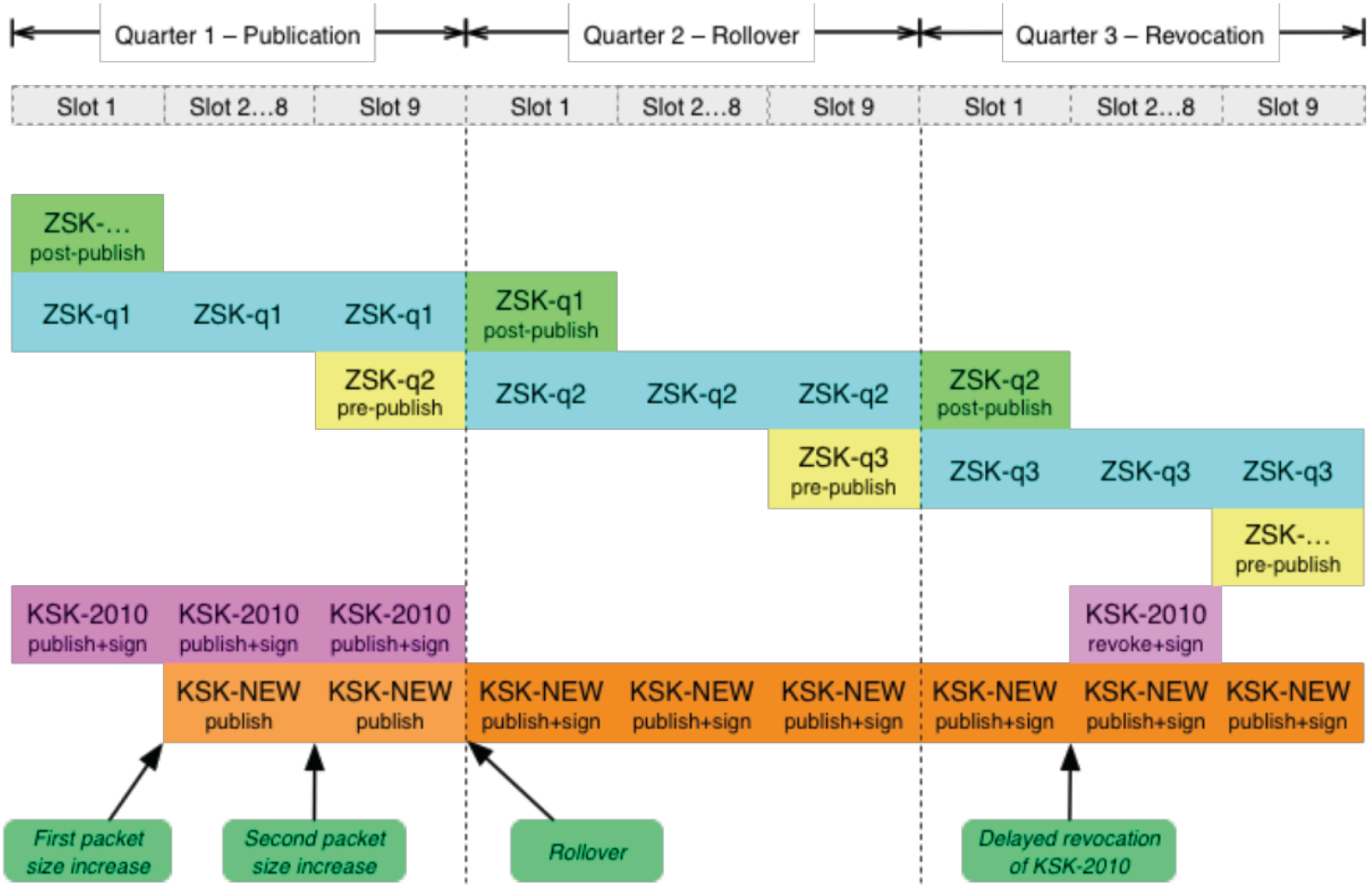
# Rolling the KSK?

- All DNS resolvers that perform validation of DNS responses use a local copy of the KSK
- They will need to load a new KSK public key and replace the existing trust anchor with this new value at the appropriate time
- This key roll could have a public impact, particularly if DNSSEC-validating resolvers do not load the new KSK

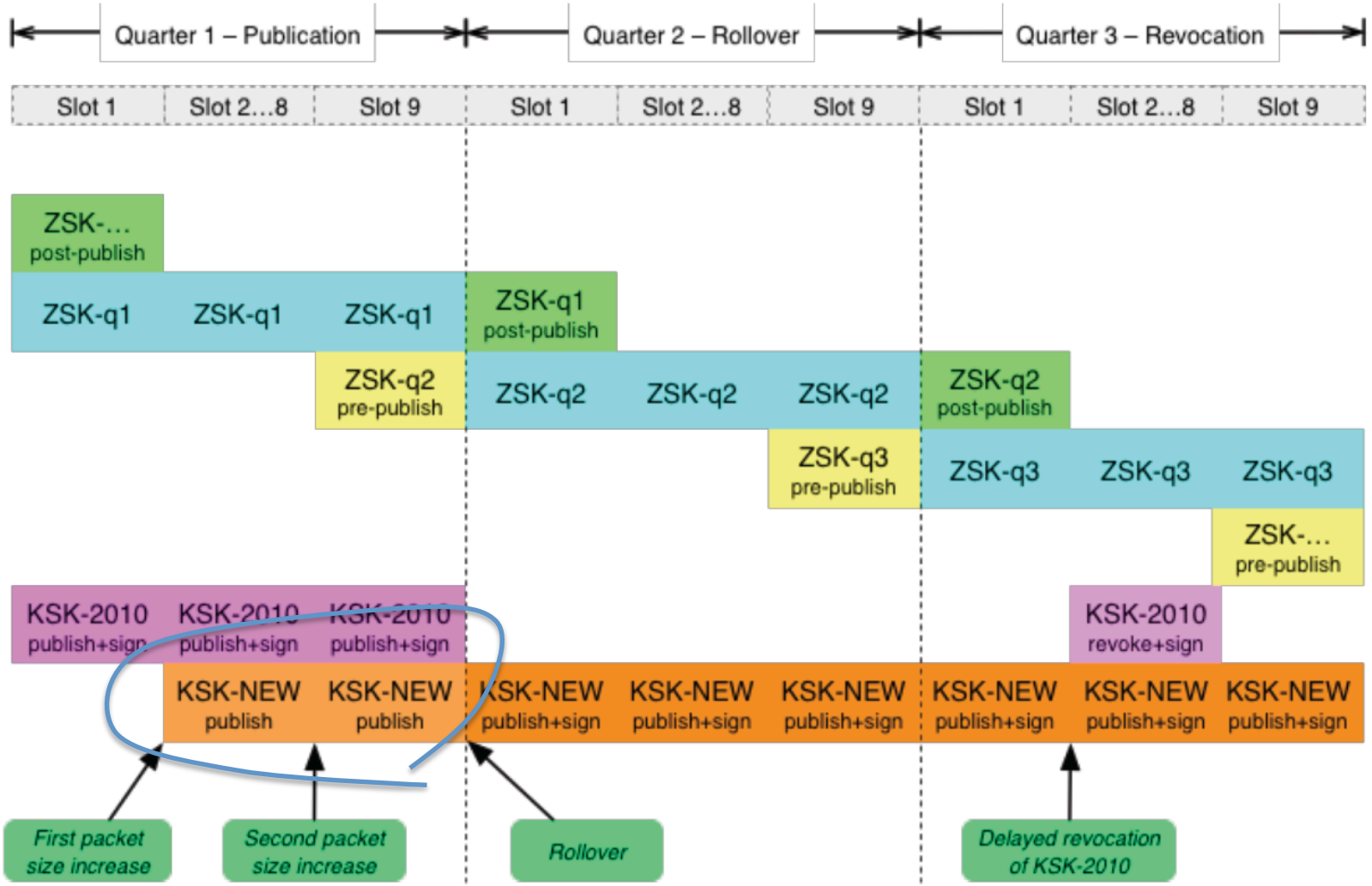
# Easy, Right?

- Publish a new KSK and include it in DNSKEY responses, signed by the old KSK
- Use the new KSK to sign the ZSK, as well as the old KSK signature
  - Resolvers use old-signs-over-new to pick up the new KSK, validate it using the old KSK, and replace the local trust anchor material with the new KSK
- Withdraw the old signature signed via the old KSK
- Revoke the old KSK

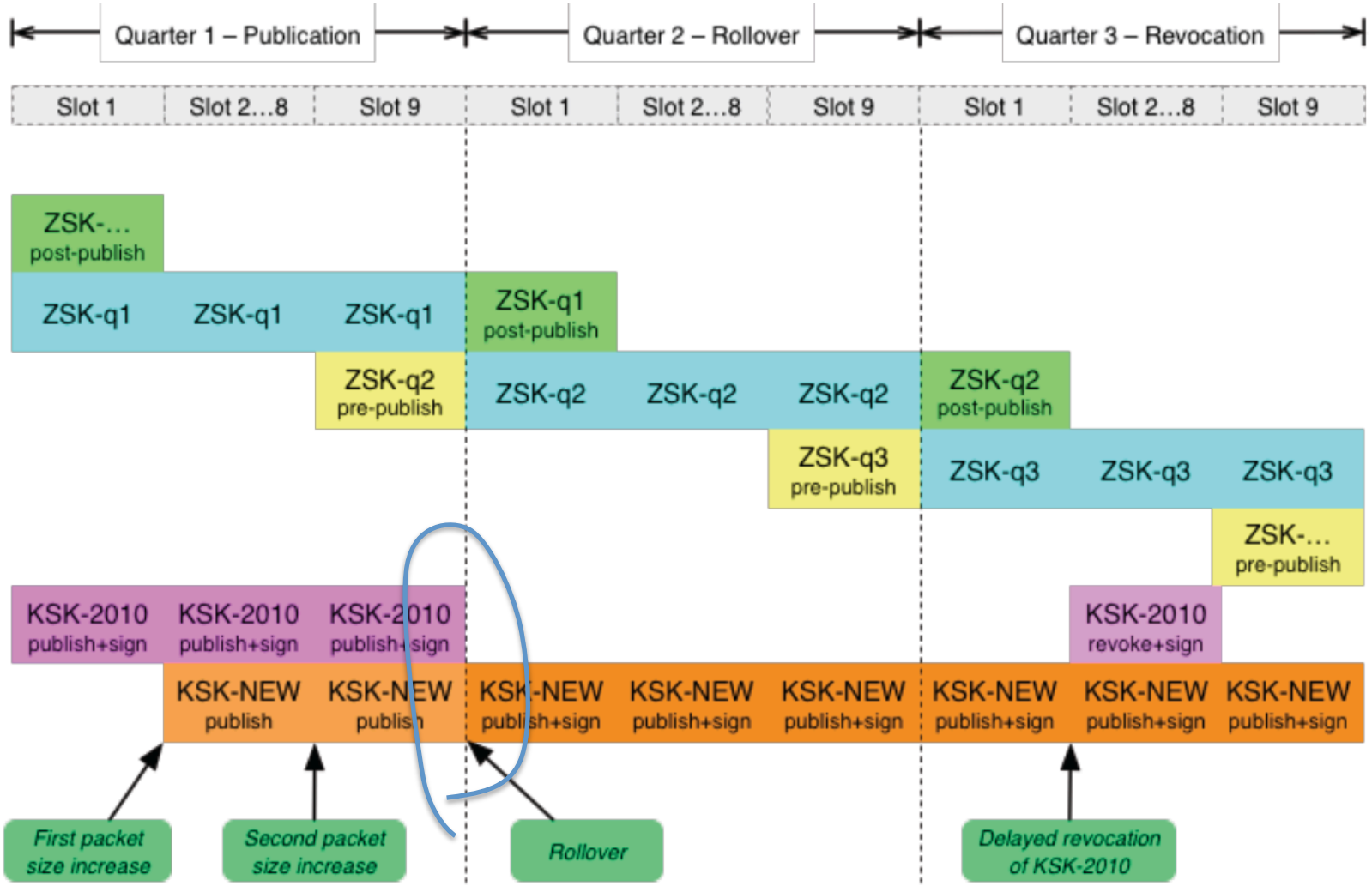
# The RFC5011 Approach



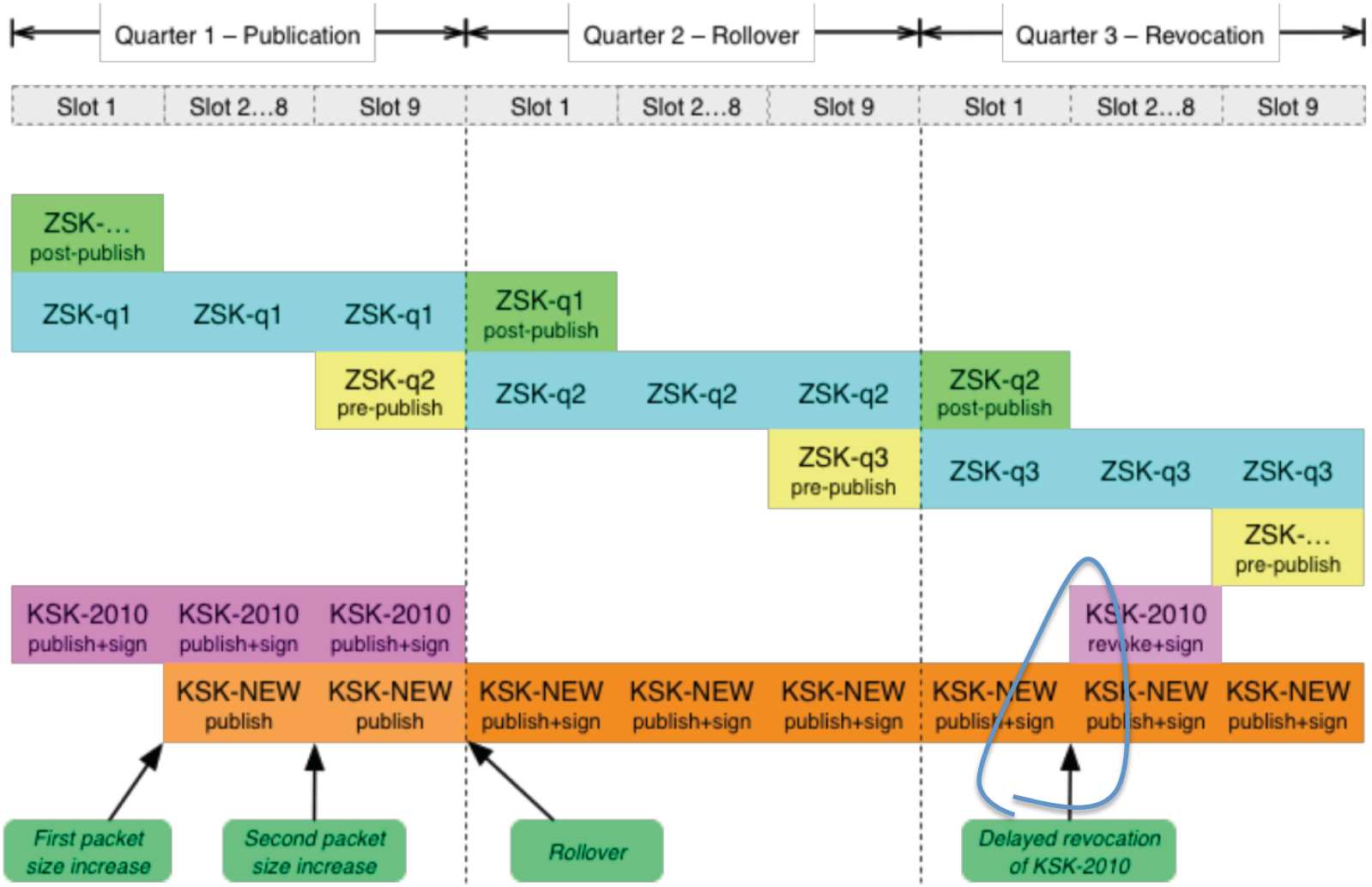
# 1. Introduce New KSK



# 2. Cutover to New KSK



# 3. Destroy Old KSK



# Easy, Right?

## Roll Over and Die?

February 2010

**George Michaelson**
**Patrik Wallström**
**Roy Arends**
**Geoff Huston**

In this month's column I have the pleasure of being joined by George Michaelson, Patrik Wallström and Roy Arends to present some critical results following recent investigations on the behaviour of DNS resolvers with DNSSEC. It's a little longer than usual, but I trust that its well worth the read.
-- *Geoff*

It is considered good security practice to treat cryptographic keys with a healthy level of respect. The conventional wisdom appears to be that the more material you sign with a given private key the more clues you are leaving behind that could enable some form of effective key guessing. As RFC4641 states: "the longer a key is in use, the greater the probability that it will have been compromised through carelessness, accident, espionage, or cryptanalysis." Even though the risk is considered slight if you have chosen to use a decent key length, RFC 4641 recommends, as good operational practice, that you should "roll" your key at regular intervals. Evidently it's a popular view that fresh keys are better keys!

The standard practice for a "staged" key rollover is to generate a new key pair, and then have the two public keys co-exist at the publication point for a period of time, allowing relying parties, or clients, some period of time to pick up the new public key part. Where possible during this period, signing is performed twice, once with each key, so that the validation test can be performed using either key. After an appropriate interval of parallel operation the old key pair can be deprecated and the new key can be used for signing.

This practice of staged rollover as part of key management is used in X.509 certificates, and is also used in signing the DNS, using DNSSEC. A zone operator who wants to roll the DNSSEC key value would provide notice of a pending key change, publish the public key part of a new key pair, and then use the new and old private keys in parallel for a period. On the face of it, this process sounds quite straightforward.

What could possibly go wrong?

# But that was then…

And this is now:

- Resolvers are now not so aggressive in searching for alternate validation paths when validation fails

  (as long as resolvers keep their code up to date, which everyone does – right?)

- And now we ***all*** support RFC5011 key roll processes
- And ***everyone*** can cope with large DNS responses

So all this will go without a hitch

Nobody will even notice the KSK roll at the root

Truly ruly!

# But that was then…

And this is now:

- Resolvers are now not so aggressive in searching for alternate validation path… …hen validation fails
  (as long a… …up to date, which everyone does – rig…
- And now w… …011 key roll processes
- And ***everyo*** …cope with large DNS responses

So all this will go without a hitch

Nobody will even notice the KSK roll at the root

Truly Ruly!

Not!

# What we all should be concerned about…

That resolvers who validate DNS responses will fail to pick up the new DNS root key automatically

- i.e. they do not have code that follows RFC5011 procedures for the introduction of a new KSK

The resolvers will be unable to receive the larger DNS responses that will occur during the dual signature phase of the rollover

# Technical Concerns

- Some DNSSEC validating resolvers do not support RFC5011
  - How many resolvers may be affected in this way?
  - How many users may be affected?
  - What will the resolvers do when validation fails?
    - Will they perform lookup ‘thrashing’
  - What will users do when resolvers return SERVFAIL?
    - How many users will redirect their query to a non-validating resolver

# Technical Concerns

- Some DNSSEC validating resolvers do not support RFC5011
  - How ma… …ay?
  - …esolvers do when validation fails?
    - Will they perform lookup 'thrashing'
  - What will users do when resolvers return SERVFAIL?
    - How many users will redirect their query to a non-validating resolver

Really hard to test this in the wild with heading down the path of fake root zones

And because of the RFC5011 30 day holddown then its not a simple "point your resolver here" kind of test

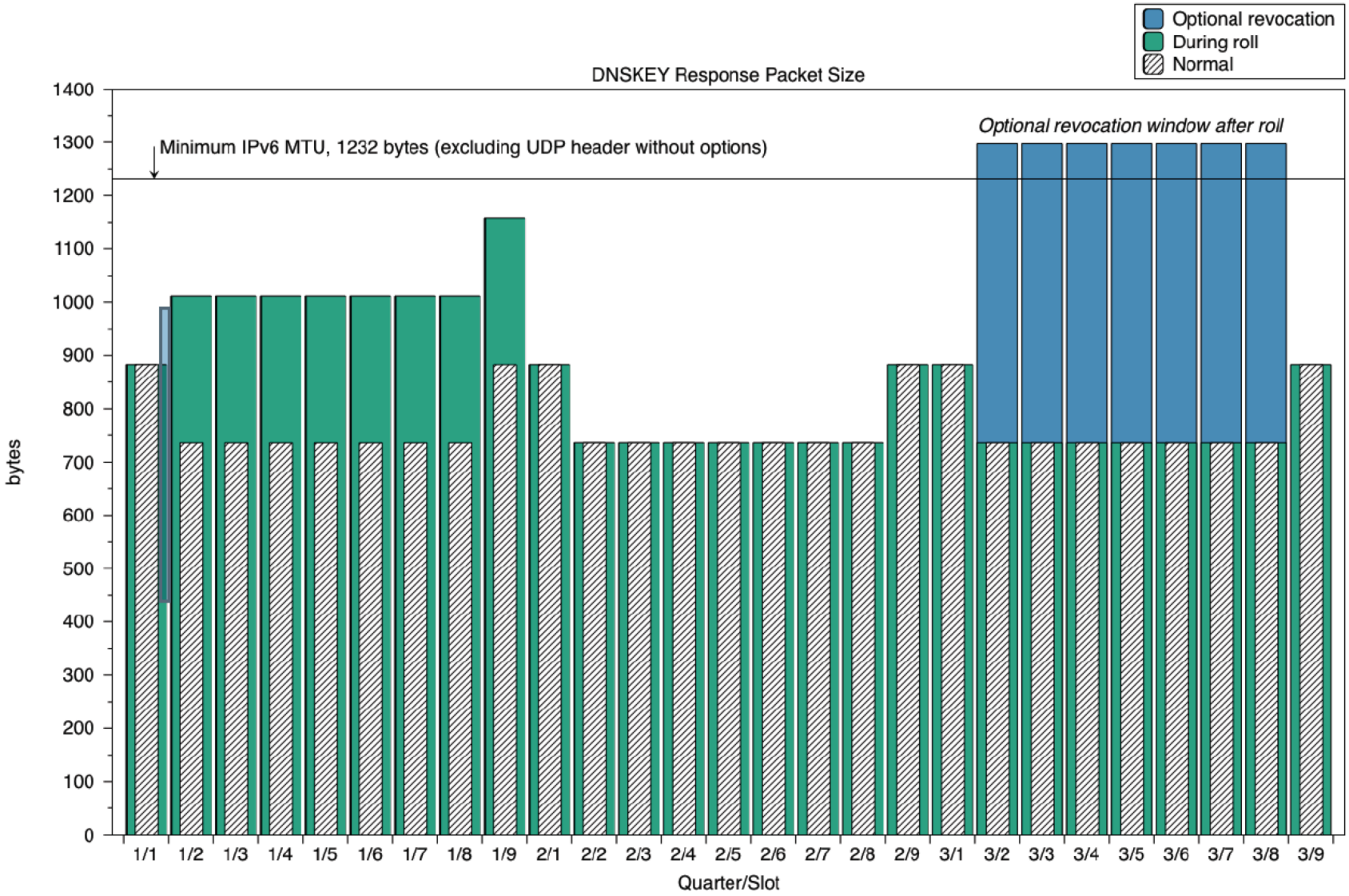
# What can be tested …

That resolvers who validate DNS responses will fail to pick up the new DNS root key automatically

- i.e. they do not have code that follows RFC5011 procedures for the introduction of a new KSK

Will resolvers be able to receive the larger DNS responses that will occur during the dual signature phase of the rollover

# DNS Response Sizes



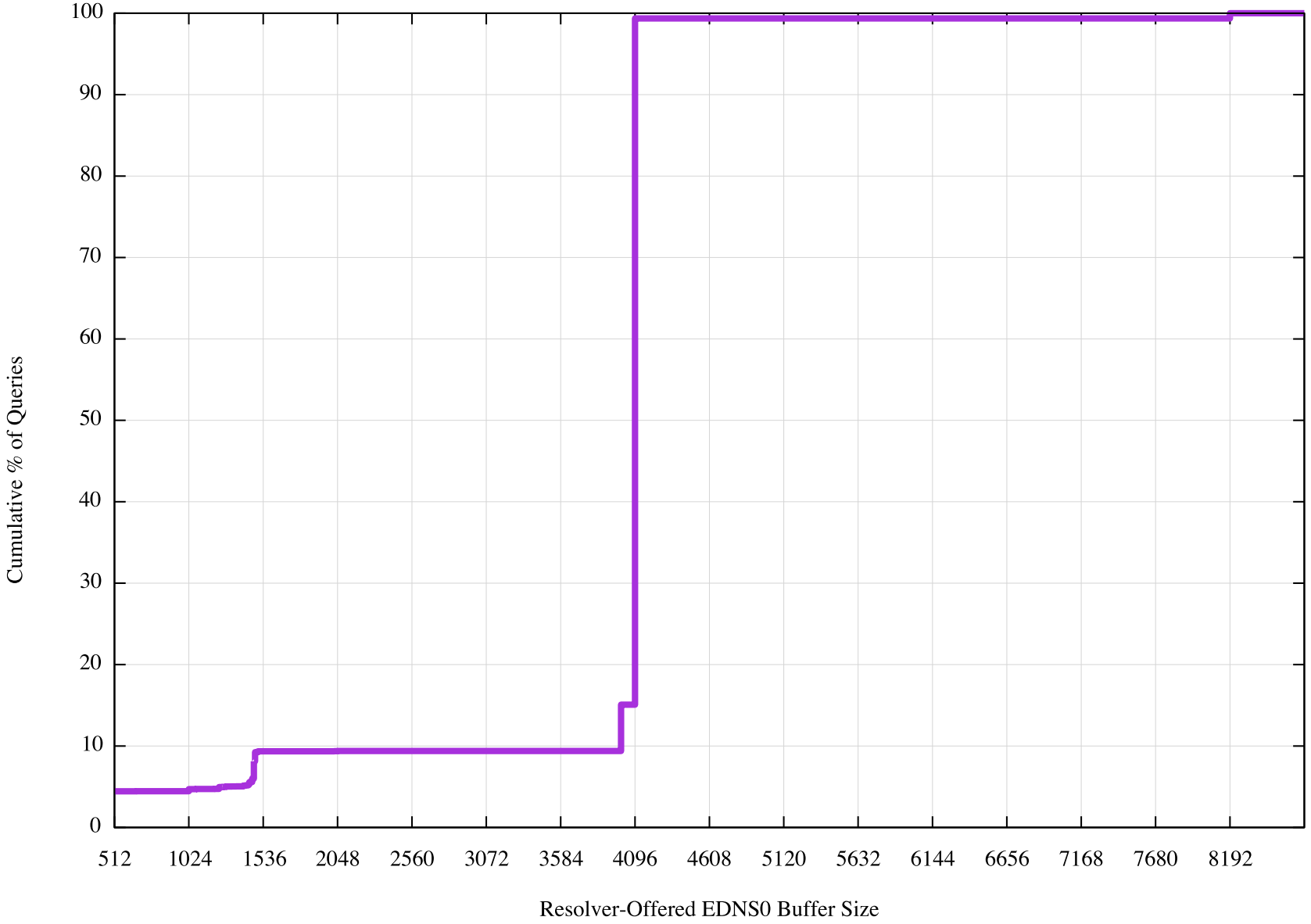
# So we've been testing large responses in the DNS

- We are interested in sending DNSSEC-aware DNS resolvers a response that is much the same size as that being contemplated in a KSK key roll
- And seeing whether they got the response

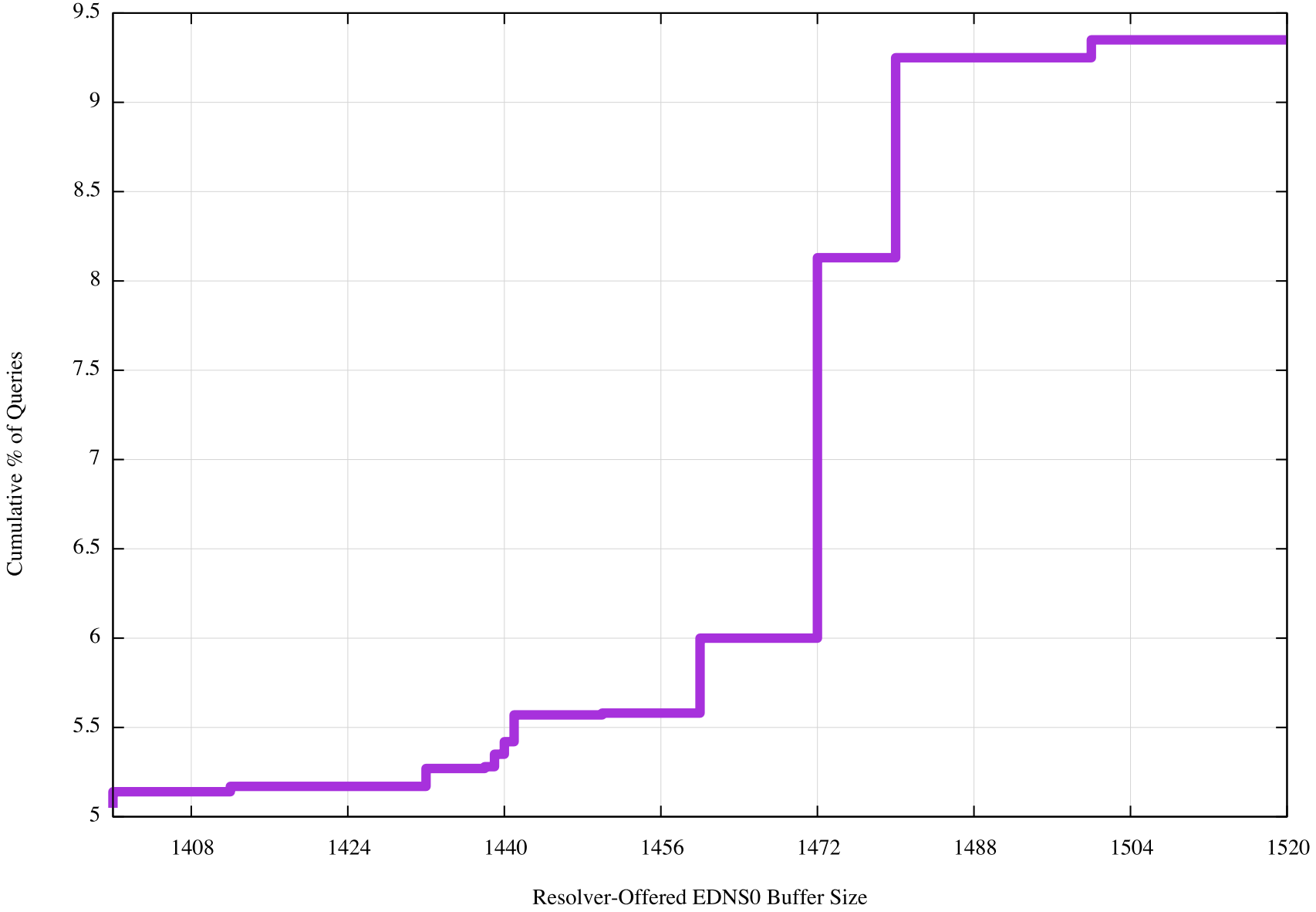
# Some Interesting Sizes

| | |
|---|---|
| 8 octets | UDP pseudo header size |
| 20 octets | IPv4 packet header |
| 40 octets | maximum size of IPv4 options in an IPv4 IP packet header |
| 40 octets | IPv6 packet header |
| 512 octets | the maximum DNS payload size that must be supported by DNS |
| 560 octets | the maximum IPv4 packet size that must be supported by IPv4 DNS UDP systems |
| 576 octets | the largest IP packet size (including headers) that must be supported by IPv4 systems |
| 913 octets | the size of the current root priming response with DNSSEC signature |
| 1,232 octets | the largest DNS payload size of an unfragmentable IPv6 DNS UDP packet |
| 1,280 octets | the smallest unfragmented IPv6 packet that must be supported by all IPv6 systems |
| **1,425 octets** | **the largest size of a ./IN/DNSKEY response with a 2048 bit ZSK** |
| 1,452 octets | the largest DNS payload size of an unfragmented Ethernet IPv6 DNS UDP packet |
| 1,472 octets | the largest DNS payload size of an unfragmented Ethernet IPv4 DNS UDP packet |
| 1,500 octets | the largest IP packet supported on IEEE 802.3 Ethernet networks |

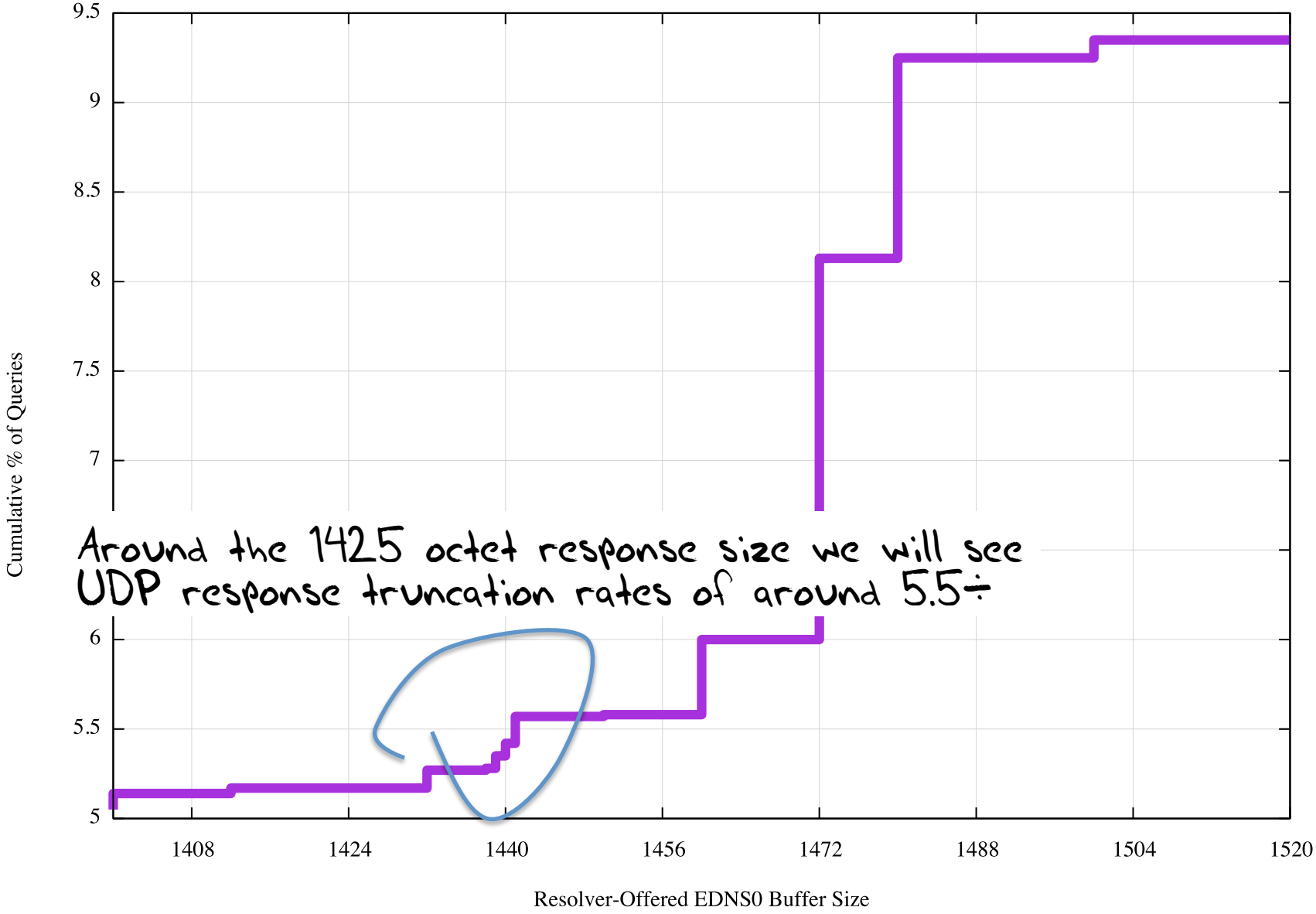
# EDNS(0) UDP Buffer sizes



# EDNS(0) UDP Buffer sizes



# EDNS(0) UDP Buffer sizes



# The Test

- We are interested in resolvers who are DNSSEC aware (queries that contain the EDNS0 option with DNSSEC OK flag set on)
- We would like to test larger responses:
  - 1,440 octets of DNS payload
- We would like to test a couple of crypto protocols
  - RSA
  - ECDSA

# EDNS(0) DNSSEC OK Set

76,456,053 **queries**

63,352,607 queries with EDNS(0) and DNSSEC OK set

= 83% of queries

777,371 **resolvers**

649,304 resolvers with EDNS(0) and DNSSEC OK set

= 84% of resolvers

# Large Responses

How well are 1,440 octet DNS responses handled when compared to much smaller responses?

# 1,440 octet RSA-signed Responses

9,113,215 tests

7,769,221 retrieved the 1x1 blot (85%)

2,644,351 queried for the DS record

849,340 queried for the DS record (but no blot fetch)

494,581 timed out (but no blot fetch)

72 appeared to fail the DNS

# 1,440 octet RSA-signed Responses

9,113,215 tests

7,769,221 retrieved the 1x1 blot

2,644,351 queried f[illegible]

8[illegible] the DS record (but no blot)

494,581 timed out (but no blot)

72 appeared to fail the DNS

Some 5÷ of experiments did not run through to completion.
For an online ad, this is not an unexpected outcome

# 1,440 octet RSA-signed Responses

9,113,215 tests

7,769,221 retrieved the 1x1 b[illegible]

2,644,351 queri[illegible]

84[illegible]cord (but no blot)

4[illegible] (but no blot)

[illegible]2 appeared to fail the DNS

This is measuring USERS not RESOLVERS. Users often use local configurations of 2 or more resolvers, and problems in one resolver appear to be fixed by the other resolver(s).

# Small vs Large

What happens when the response size grows above 1,472 octets?

| 1,440 Octets Payload | | 1,770 Octets Payload | |
|---|---|---|---|
| Experiments: | 6,542,993 | Experiments: | 6,566,645 |
| Web Fetch: | 5,880,921 | Web Fetch: | 5,992,617 |
| DS Fetch: | 181,610 | DS Fetch: | 167,119 |
| Timeout: | 480,415 | Timeout: | 401,831 |
| DNS Fail: | 47 | DNS Fail: | 5,078 |

# ECDSA vs RSA

The spec says that when a resolver encounters a zone signed only with algorithms that are not supported by the resolver then it will treat the zone as unsigned and not proceed with validation

Most resolvers determine the zone's signing algorithms from the DS record

What happens when we compare a 1,440 octet response signed by RSA and a 1,440 octet response signed by ECDSA?

# 1,440 octet ECDSA-signed Responses

9,137,436 tests

7,766,572 retrieved the 1x1 blot

2,644,564 queried for the DS record

860,163 queried for the DS record (but no blot)

505,045 timed out (but no blot!)

5,656 appeared to fail the DNS

# 1,440 octet ECDSA-signed Responses

9,137,436 tests

7,766,572 retrieved the 1x1 [illegible]

2,644,564 queri[illegible]

86[illegible]cord (but no blot)

5([illegible]but no blot!)

[illegible]50 appeared to fail the DNS

This is larger than RSA failure rates, but is still a small proportion of users affected. Some resolvers appear to have problems when presented with an unknown crypto protocol.

# IPv4 vs IPv6

Do resolvers prefer IPv4 over IPv6?

Total Queries: 47,826,735

Queries over V6: 394,816

Number of Resolvers: 109,725

Number of Resolvers

using IPv6 for queries: 2,849

# IPv4 vs IPv6

Do resolvers prefer IPv4 over IPv6?

Total Queries: 47,[illegible]

Queries ov[illegible]

[illegible]olvers: 109,725

N[illegible]mber of Resolvers

using IPv6 for queries: 2,849

In a Dual Stack environment 1% of queries and 3% of resolvers use IPv6.

What if the server was IPv6 only?

# Some Observations

There is a LOT of DNSSEC validation out there

- 87% of all queries have DNSSEC-OK set
- 30% of all DNSSEC-OK queries attempt to validate the response
- 25% of end users are using DNS resolvers that will validate what they are told
- 12% of end users don't believe bad validation news and turn to other non-validating resolvers when validation fails.

# Some Observations

There is very little V6 being used out there

- 1% of queries use IPv6 as the transport protocol when given a dual stack name server

It seems that when given a choice:

Browsers prefer IPv6

Resolvers prefer IPv4

# Some Observations

ECDSA is viable – sort of

- 1 in 5 clients who use resolvers that validate RSA-signed responses are unable to validate the same response when signed using ECDSA
- But they fail to “unsigned” rather than “invalid” so it’s a (sort of) safe fail

# What's "Too Big?"

- Out of 82,954 resolvers seen in a glueless delegation measurement experiment, 4,251 resolvers appeared to be incapable of receiving a 1,444 octet DNS response
  - 21% of these failing resolvers used IPv6
- 6% of queries shifted to TCP for the larger response
- So large DNS response packets might be a problem area

# Can it work?

If we stick to RSA and keep response sizes at or below 1,440 octets then there appears to be no obvious user impact in terms of packet size

- Some resolvers may get stuck, but users appear to use multiple resolvers

# But

- The signed **.org** DNSKEY response is 1,625 octets, and there are no obvious signals of service failures for .org names
- The potential issues surrounding large responses and the DNS may be a little more subtle than these experimental results suggest

# Where are we?

- A key roll of the Root Zone KSK will cause some resolvers to fail:
  - Resolvers who do not pick up the new key in the manner described by RFC5011
  - Resolvers who cannot receive a DNS response of ~1,300 octets
- Many users who use these failing resolvers will just switch over to use a non-validating resolver
- A small pool of users will be affected with no DNS

# Now?

Public comment:

draft report for ICANN Public Comment

https://www.icann.org/public-comments/root-ksk-2015-08-06-en

Comments close 15th September 2015

Please read & comment

Questions?