APNIC 40

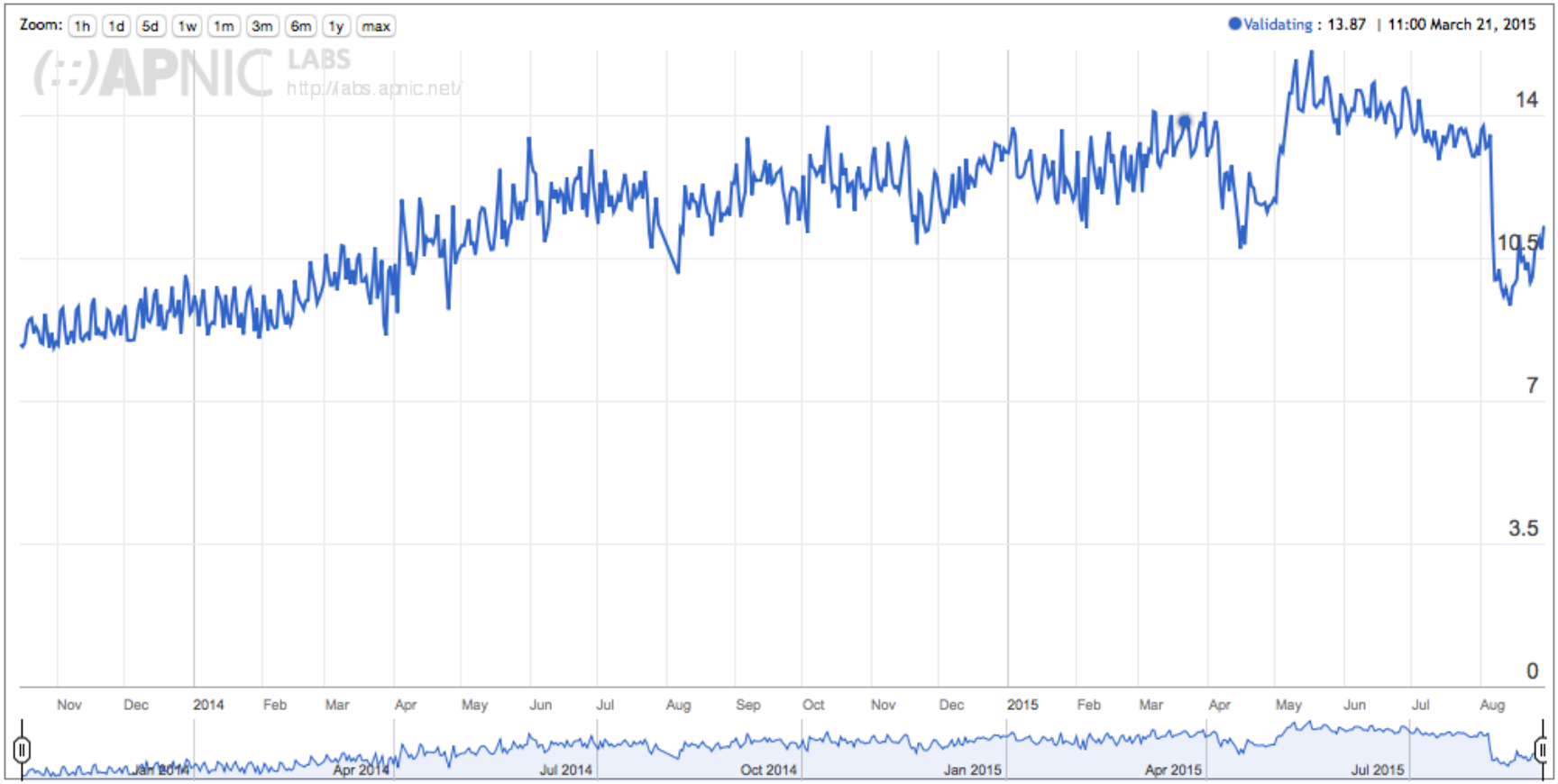
Testing Rolling Roots

Geoff Huston
APNIC Labs

JAKARTA, INDONESIA
3-10 September 2015

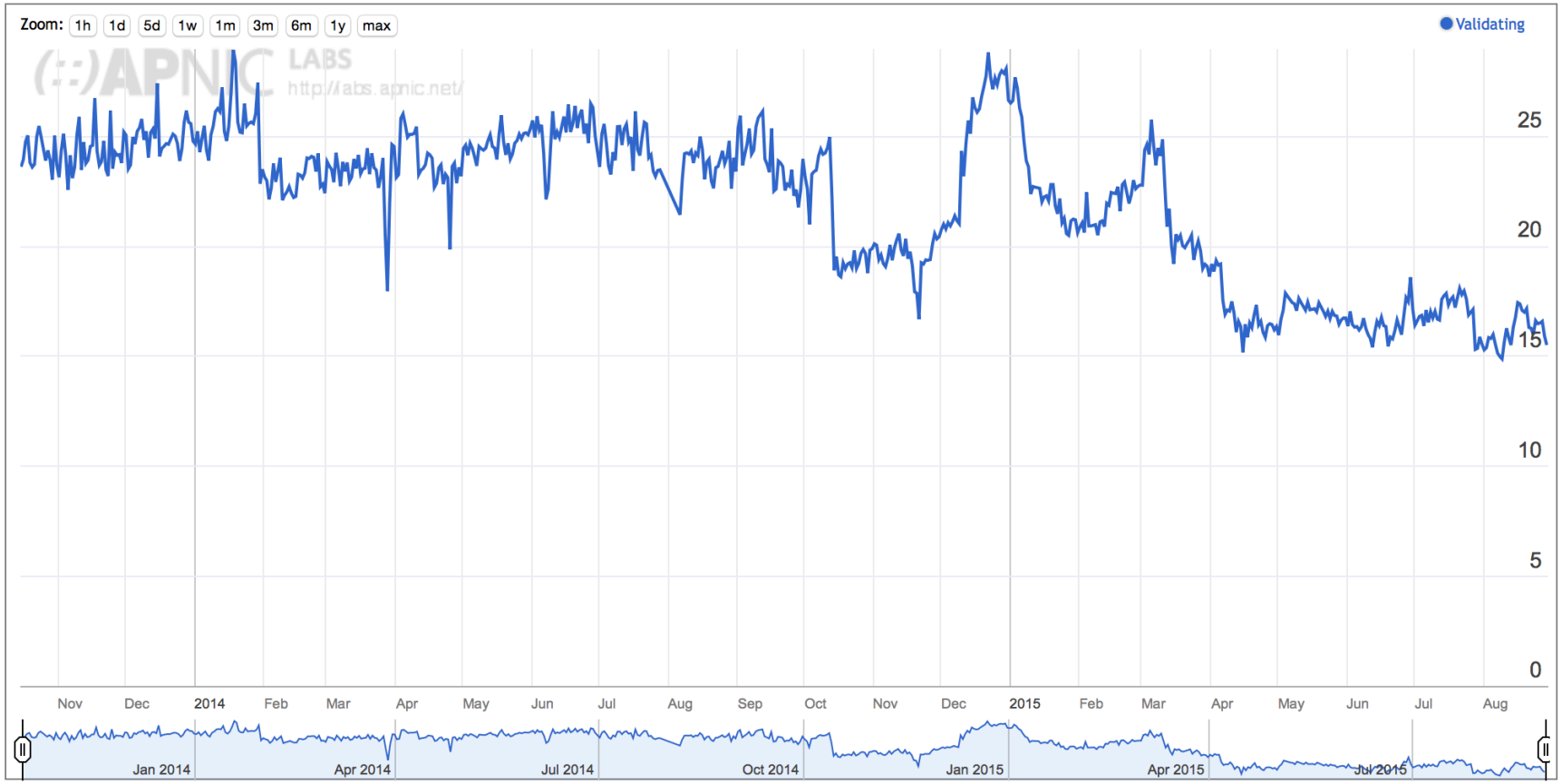
#apnic40

Use of DNSSEC in Today's Internet



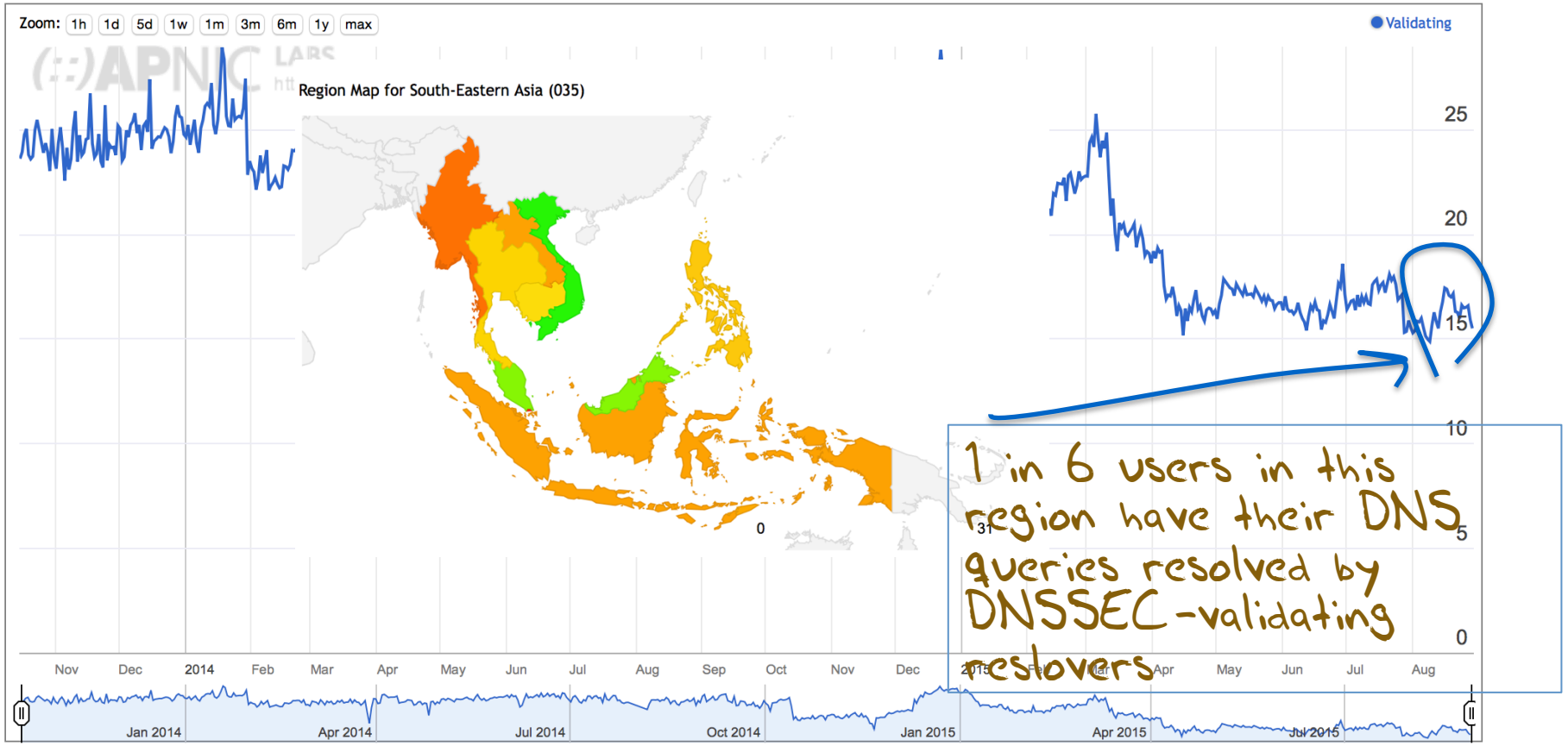
Use of DNSSEC in SE Asia

Use of DNSSEC Validation for South-Eastern Asia (XU)



Use of DNSSEC in SE Asia

Use of DNSSEC Validation for South-Eastern Asia (XU)



Why is this relevant?

Because…

the root zone managers are preparing to roll the DNS Root Zone Key Signing Key

(and this 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.

, VA - location of first DNSSEC key signing ceremony

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.

ZSK?

- Zone Signing Key
- Used to generate the digital signature RRSIG records in the root zone
- The ZSK is rolled regularly every quarter
- The DNSKEY record for the ZSK is signed by the KSK

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”
  - Most of the time the KSK is kept offline in highly secure facilities

The Eastern KSK Repository

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

The Western KSK Repository

El Segundo, California *

The Ultra Secret Third KSK Repository in Amsterdam

KSK spotting by George Michaelson

The Uruguay Mobile KSK

KSK spotting by George Michaelson

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

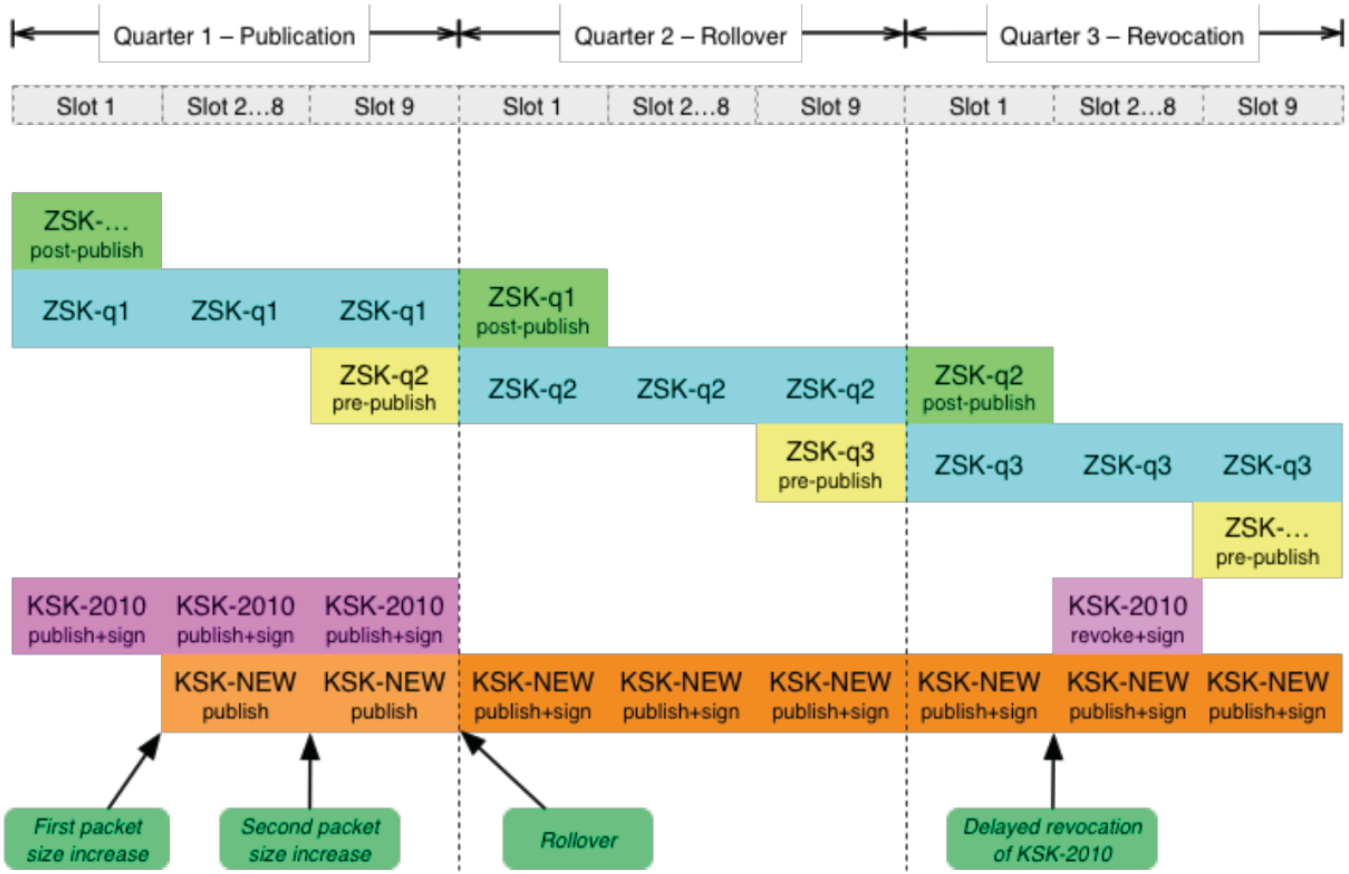
Rolling the KSK?

- 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

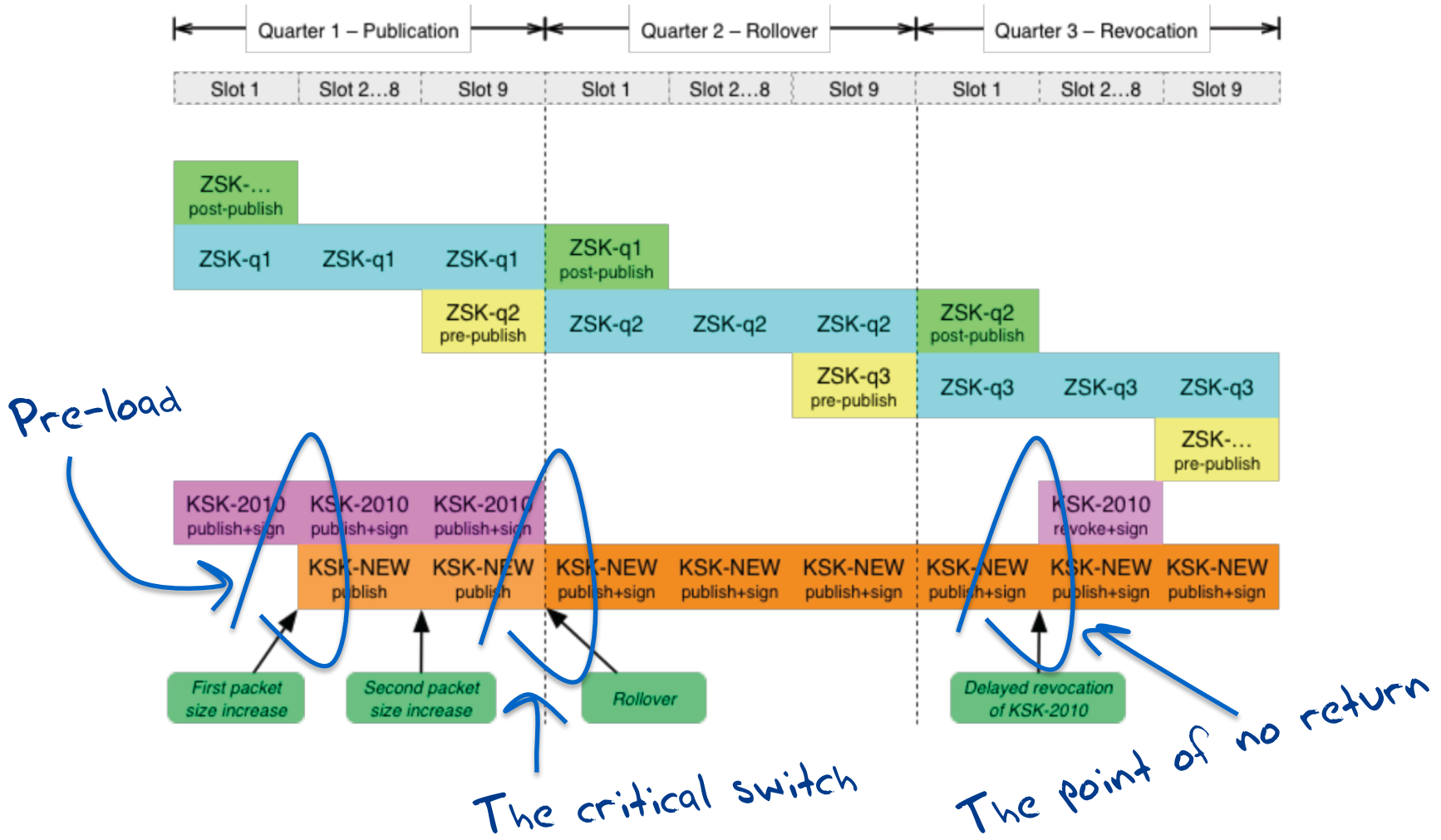
Easy, Right?

- Publish a new KSK and include it in DNSKEY responses
- 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



The RFC5011 Approach



Just Like Last Time?

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 paths when validation fails
  - (as long as resolvers keep their code up to date, which everyone does)
- And now we only roll processes
- And ***everyone*** DNS responses

So all this will go w

Nobody will even notice the KSK roll at the root

Truly Ruly!

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

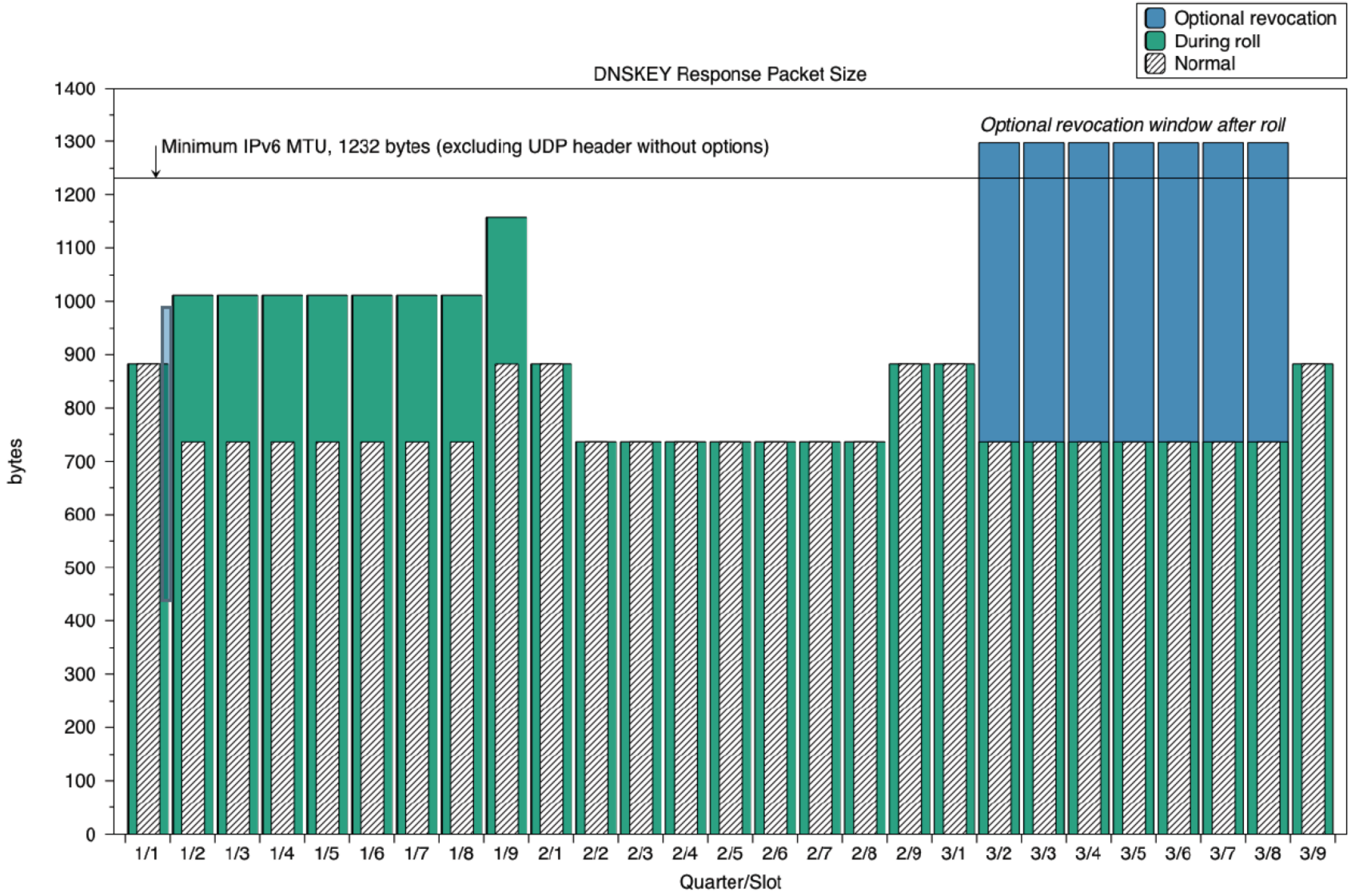
Technical Concerns

- Some DNSSEC validating resolvers do not support RFC5011
  - How many resolvers may be affect
  - How many user
  - resolvers return SERVFAIL?
    - 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

DNS Response Sizes



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

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

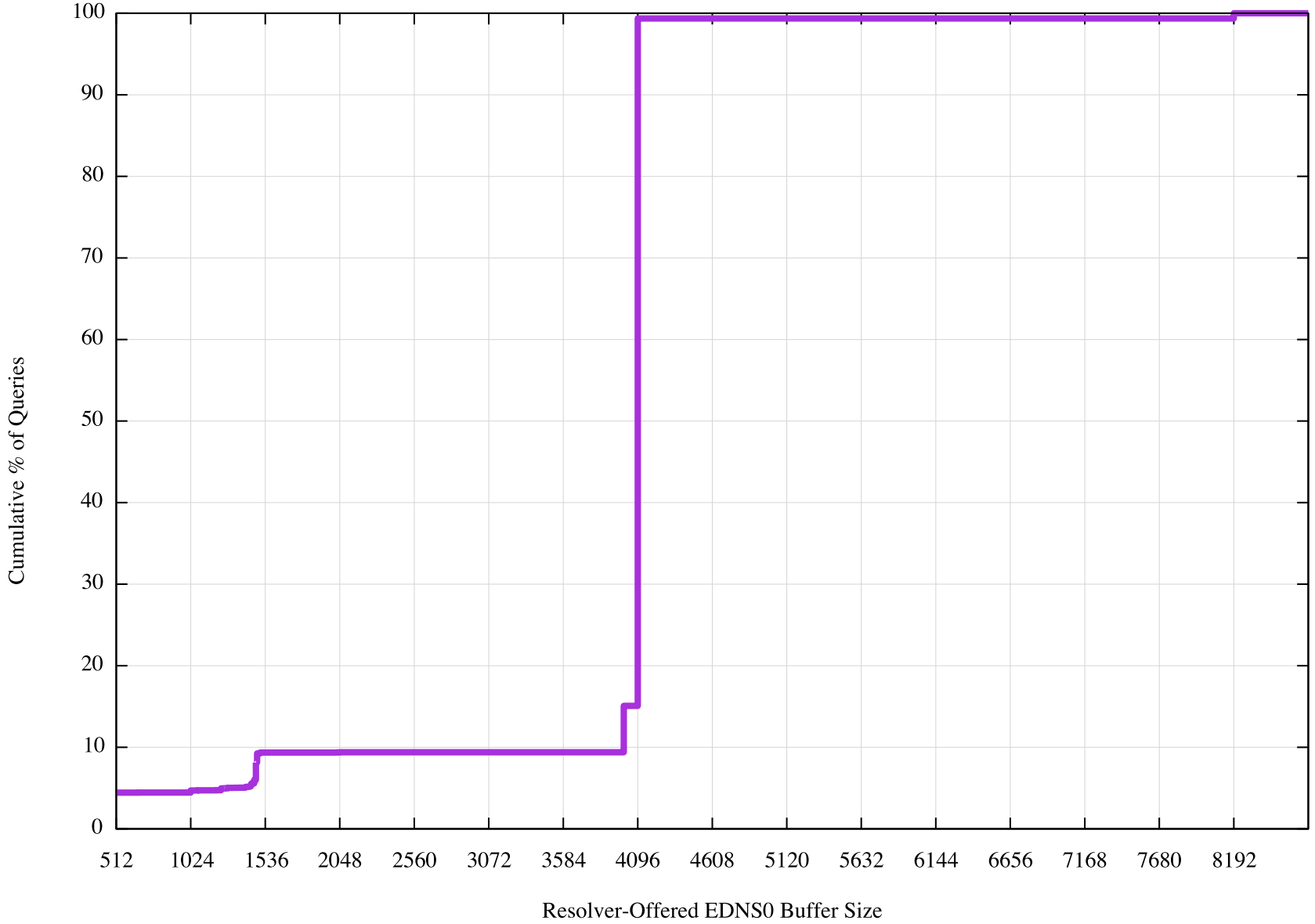
= 83% of queries

777,371 **resolvers**

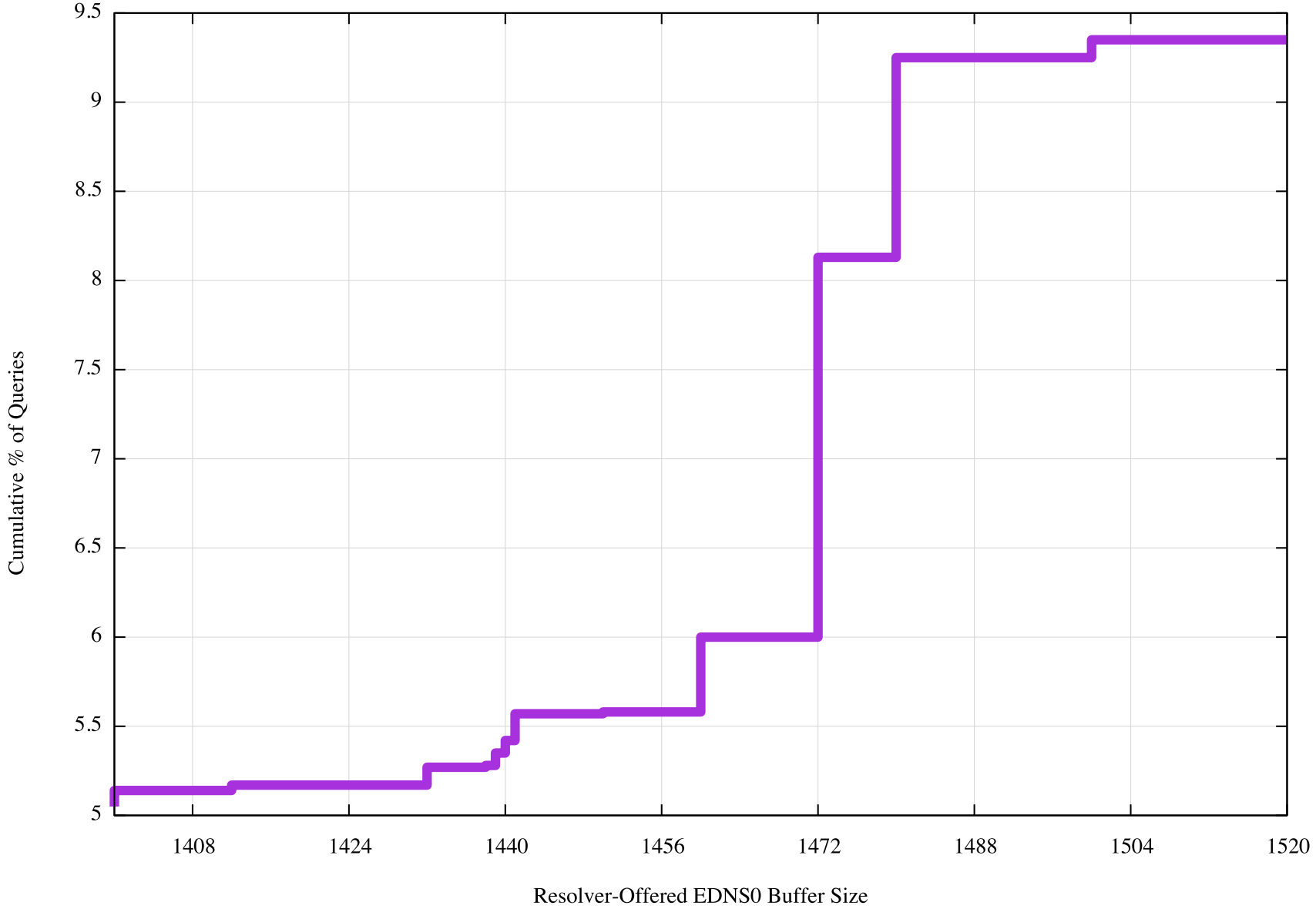
649,304 resolvers with EDNS(0) and DNSSEC OK set

= 84% of resolvers

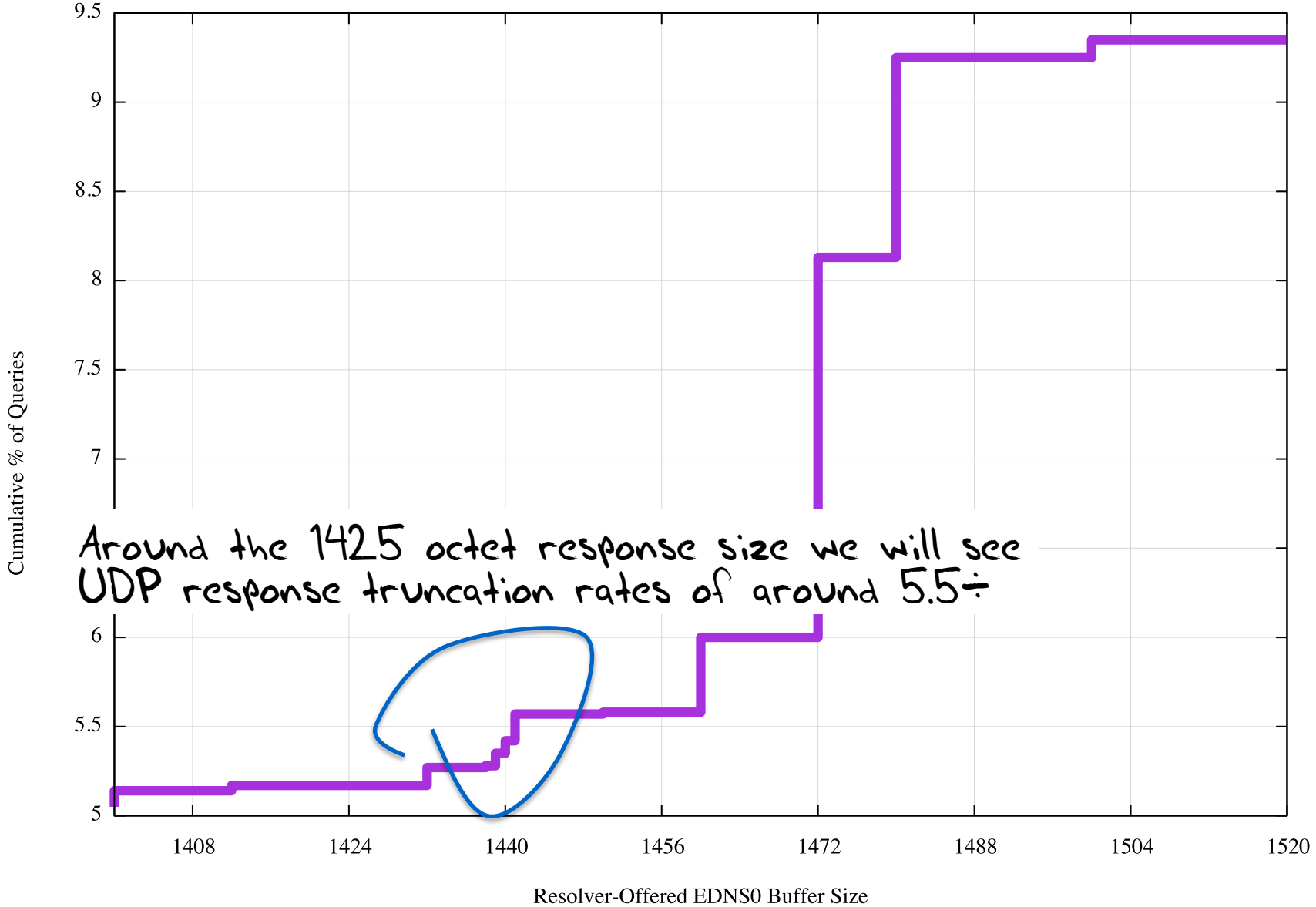
EDNS(0) UDP Buffer sizes



EDNS(0) UDP Buffer sizes



EDNS(0) UDP Buffer sizes



Small vs Large

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 test

–

– the DS record

– queried for the DS record (but no blot)

505,045 timed out (but no blot!)

5,656 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: 47,826,735

Queries over V6: 394,8[illegible]

Number [illegible]: 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 - 1

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 - 2

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 - 3

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

Some Observations - 4

The larger DNS responses will probably work

- The “fall back to TCP” will rise to 6% of queries when the response size get to around 1,350 octets
- And the DNS failure rate appears to rise by around 1 - 2 %

- BUT .org currently runs at 1,650 octets and nobody is screaming failure

- So it will probably work

Some Observations - 5

We can’t measure automated key take up

- We can’t see how many resolvers fail to use RFC5011 notices to pick up the new KSK as a Truct Anchor in advance
- We will only see it via failure on key roll

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?

Comments - 1

Why Now?

What is the imperative to roll the key now? Could we use more time to improve preparedness for this roll? For example, could we use further time to introduce some explicit EDNS(0) signalling options in resolvers to expose RFC5011 capability?

Comments - 2

Measuring and Testing?

What measurements are planned to be undertaking during the key roll process? What are the threshold metrics for proceeding to the next phase? What is the threshold metric to proceed with the revocation of the old KSK?

Comments - 3

Algorithm Change

The report's language around the potential for algorithm change is unclear. There appears to be a strong bias to retention of RSA as the KSK algorithm, despite evidence that ECDSA is both shorter and potentially faster to compute. Whilst the report argues for a reduced risk of large packets, it doesn't clearly explain why larger RSA-based DNS response payloads would be preferable to smaller ECDSA DNS response payloads.

Comments - 4

Scheduling

The report notes as a constraint that a key roll must be aligned with existing Quarter and 10-day periods used in existing processes. This has the potential consequence of scheduling the critical change in the root zone on a weekend, or on a major public holiday. Why?

Comments - 5

Testing for RFC5011

The report notes that there is no easy way to test is a resolver has picked up the new KSK via 5011 signalling (or otherwise)? Has the team working on this explored the use of sentinel zones signed by the new KSK? For example it could be envisaged that the roll process could use the incoming KSK to sign some sentinel record in the root zone allowing measurement of the extent to which resolvers are able to use the KSK as a trust anchor to validate the sentinel record. It would be helpful to understand why such potentially measurable actions are not viable options in this particular context.

Comments - 6

Serialization

The report assumes a single new KSK. What are the issues of introducing 2 or even 3 new KSKs at this point?

Comments - 7

All together all at once?

Why do all root zones flip to use the new KSK all at the same time?

Why is there not a period of dual sigs over the root ZSK?

Why not allow each root server to switch from old to old+new to new using a staggered timetable?

There may be perfectly sound reasons why all together all at once is a better option than staggered introduction, but report does not appear to provide any such reasons.