

DNSSEC @ IANA

December 10 2007 Presentation
to USG at NIST

Purpose

- Draw on the extensive technical expertise from those working in DNSSEC and related fields;
- Update those in the field on IANA's DNSSEC deployment progress; and
- Discuss and exchange ideas regarding the technical design being considered for DNSSEC deployment at IANA. R1

Slide 2

R1 which served as our guide to parameter selection
RL, 9/24/2007

Design Goals

- Maintainability – if its not easy, it will fail
- Reliability – if there is a problem, no one will use it
- Security – it must look and be secure for people to trust it
- Target – arpa, in-addr.arpa, uri.arpa, urn.arpa, iris.arpa, ip6.arpa (IAB), root (waiting on policy)

Overall Design

- IANA's design is built on the trailblazing work by .SE. Without the generous help from Jakob Schlyter and others at .SE, I would still be lost.
- Follows operational practices document RFC4641.

“Figure 1”

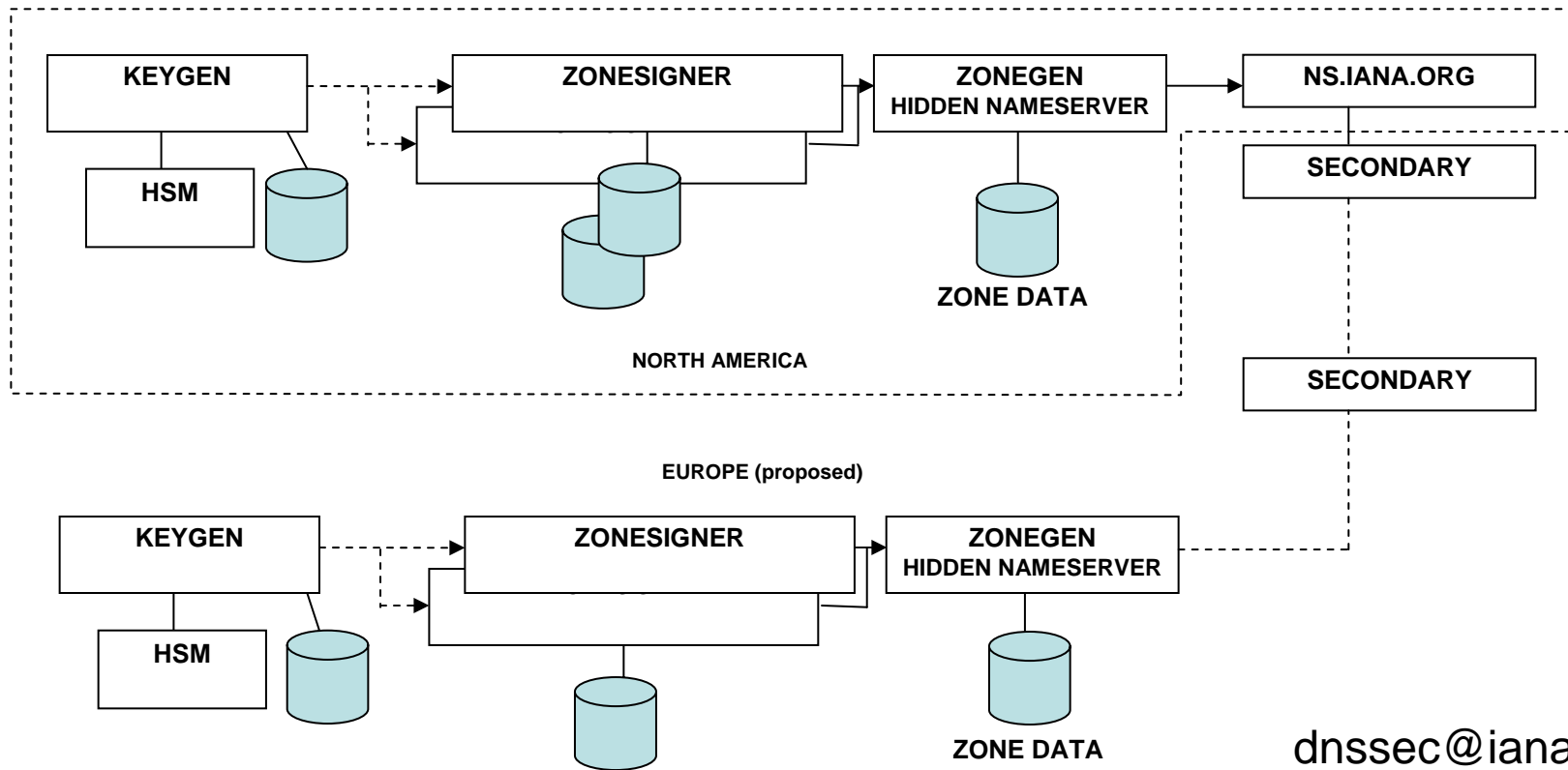
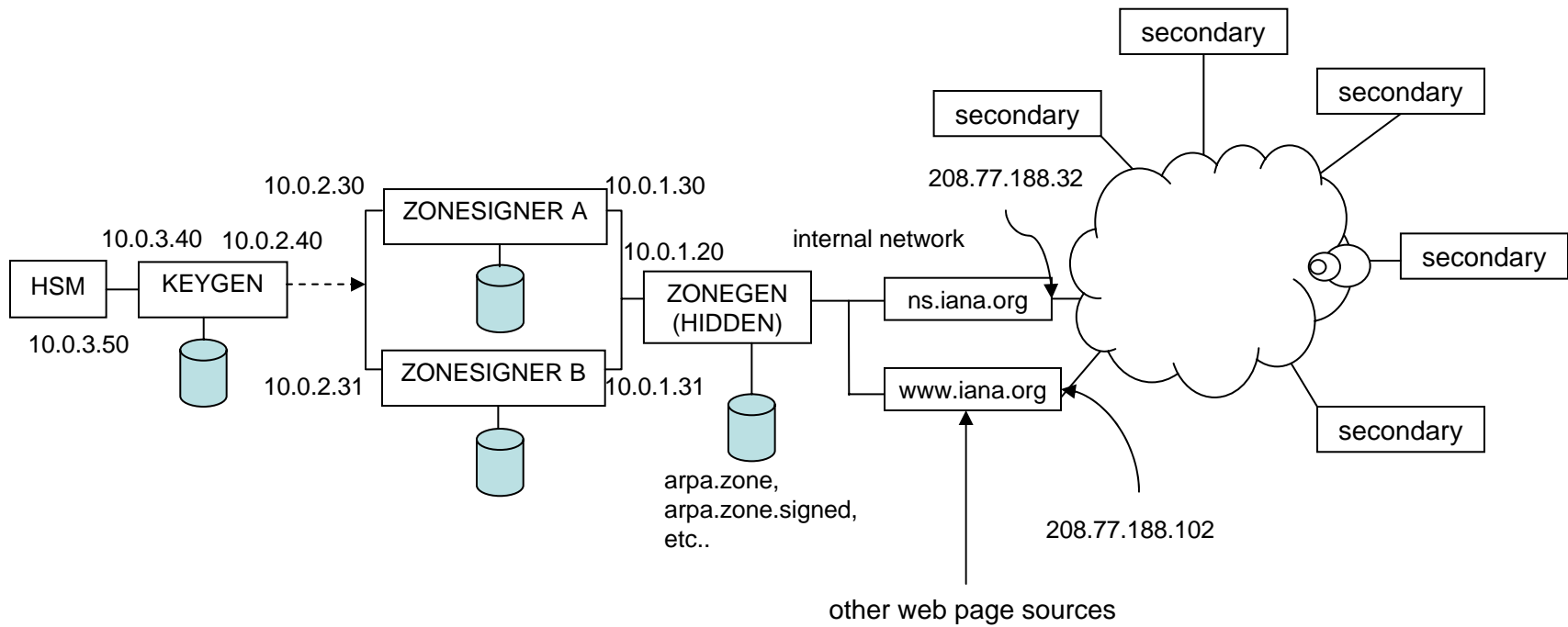


Figure 1 Details



Hardware (per site)

- 4x Dell 1RU 1950 commodity servers running Linux and FreeBSD
- 1x AEP Keyper Pro (FIPS 140-2 Level 4) external Hardware Security Module (HSM)
- 1x KVM console
- Smart cards, Flash drive
- Locked rack within ICANN cage at secure colo facility

Maintainability - Only Two Scripts

- On ZONESIGNER – **signall**: automatically run daily on multiple machines to pickup zone changes (based on SOA serial, new DS records, or expiring signatures); reload hidden master; check key status; update status web page; and email notifications.
- On KEYGEN – **keyall**: manually run monthly (when notified by email). Generates new keys and signed key bundles for ZONESIGNERs as needed. Also backs up any new keys.

Maintainability – Overlapping Keys, Rollover Script

A5

- Multiple overlapping keys (effectivity periods) to simplify rollovers.
- ZSK - three (3), old-active-new, overlapping ZSKs /w staggered effectivity periods. Use currently “active” key to sign records
- KSK - two (2) overlapping KSKs /w staggered effectivity periods. Use both to sign “key bundle” of five (5) keys
- Key generation and rollover automated in **keyall**

```
6400K+++++|+++++
2400K-----+|+++++
24001-----pppppppp+++++|+rrrr-----
08000Z-----pppppppp+|+++++rrrrr-----
92000-----p|pppppp+++++rrrr-----

keyindex file:
dn type alg tag publish date start date end date remove date
root KSK 005 64000 19750101000000 19750101000000 19761231235959 19761231235959
root KSK 005 24000 19760101000000 19760101000000 19771231235959 19771231235959
root ZSK 005 24001 19751201000000 19760101000000 19760215000000 19760229235959
root ZSK 005 08000 19760101000000 19760201000000 19760315000000 19760331235959
root ZSK 005 92000 19760201000000 19760301000000 19760415000000 19760430235959
```

A5 From rfc4641:

"Key effectivity period" The period during which a key pair is expected to be effective. This period is defined as the time between the first inception time stamp and the last expiration date of any signature made with this key, regardless of any discontinuity in the use of the key. The key effectivity period can span multiple signature validity periods.

"Signature validity period" The period that a signature is valid. It starts at the time specified in the signature inception field of the RRSIG RR and ends at the time specified in the expiration field of the RRSIG RR.

"Signature publication period" Time after which a signature (made with a specific key) is replaced with a new signature (made with the same key). This replacement takes place by publishing the relevant RRSIG in the master zone file. After one stops publishing an RRSIG in a zone, it may take a while before the RRSIG has expired from caches and has actually been removed from the DNS.

Although the "validity period" (RRSIG expiration-inception value) for the ZSK is 6 days with a "publication period" of 1 day (how often I re-compute the RRSIG and shift forward the validity period), the same keys are used for much longer "effectivity period"s. One and a half months for ZSKs with new ZSKs introduced approximately every month (within a 1/2 month window). And two years for KSKs with new KSKs introduced every year (within a 1 year window).

Administrator, 7/30/2007

Maintainability – Compromised Key, Replacement Script

- For bad ZSK (old, active, new keys)
 - old – replace key with newly generated “old” key.
 - active – use old key to sign and generate a replacement. Phase out bad key. A15
 - new – replace key with newly generated “new” key.
 - Normally done in one-step. Two-steps if “close” to a transition to account for DNS propagation delays.
- For bad KSK (2 keys)
 - One - replace key with newly generated KSK with the same effectivity period and immediately publish.
 - Both – generate two keys and phase out bad keys? A6
- Process semi-automated with **badkey** script

Slide 10

- A6** Will RFC5011 Revoke bit help here?
Administrator, 9/24/2007
- A15** This is the reason for replacing the old key.
Administrator, 10/11/2007

Reliability – Dual Signers

- Signatures on zone records are only valid for six (6) days to limit replay attacks. So an inability to sign for more than 6 days will result in DNSSEC validation to fail.
- Design: Two (2) commodity hardware based ZONESIGNERs periodically executing **signall** to make sure the zone gets signed by one of them.

Reliability - Key Backup

- Must backup even private keys to recover from catastrophe
- Encrypt and propagate new private key material as key operations generate them
- Built into regular key operations script **keyall**

Security – KSK/ZSK Split

A13

- Following .SE's lead, sensitive KSK operations are kept separate from routine ZSK signing operations by only exporting pre-signed public key bundles and a single private ZSK from KEYGEN to ZONESIGNERS.
- KEYGEN machine is connected to ZONESIGNERS only during key generation and transfer operations

A7

Slide 13

A7

Even a demo signed root might be helpful here as we could immediately publish the new DS record for arpa

Administrator, 9/24/2007

A13

man in the middle

replay

cache poisoning

Administrator, 9/24/2007

Security – HSM

- To protect against internal as well as external attacks, KSK operations (generation, signing, backup) for critical zones are performed inside the HSM.
- Do this using modified BIND tools with native PKCS11 support
- To minimize HSM operational overhead, child zones falling under .arpa will not use the HSM for KSK operations. Recovery from child zone KSK compromise can be effected quickly

Security – Key Lifetimes

- New ZSK 1024 bit every month to frustrate key guessing
- New KSK 2048 bit every year to frustrate key guessing
- Two KSKs always valid to support orderly replacement of old or compromised KSK
- Three published ZSKs to support orderly replacement and promotion of old or compromised ZSK
- 6 day (short) ZSK signature validity period to limit replay attacks while providing some time to recover from severe signing equipment failure
- 1.5 month key bundle KSK signature validity period to constrain compromised ZSK effects while not requiring daily manual resigning with KSK

Security – Key Backup

- Keys generated inside HSM (KSKs) are encrypted inside HSM before export
- Unencrypted key material (e.g. ZSK), key index, encrypted HSM keys (above), HSM configuration, and any other updated material on KEYGEN's hard drive is further encrypted using internal HSM key before transmission/backup
- Only another HSM with the same internal HSM key can decrypt this material
- Internal HSM master key backed up on N of M smartcards

Security - Meatspace

- Key generation operation requires:
 - Access to DNSSEC equipment at a secured colo facility
 - One Security Officer smartcard and PIN to enable the HSM
 - HSM User PIN to generate keys and sign the key bundle
- A minimum of two (2) authorized personnel, controlling different components above, must be present for the entire key generation operation.
- Every access to DNSSEC equipment is logged in a DNSSEC log book
- **keyall** propagates its activities to the DNSSEC Administrator via email
- Material used to (re)build KEYGEN and HSM contents will be stored in safety deposit boxes. Each box will contain one of the required 2 out of 4 HSM master key smartcards along with an encrypted backup of current KSKs and miscellaneous configuration files needed for rebuilding

A14

Slide 17

A14

I see at least five (5) people should be authorized at any time with no one person having control of both security officer card card and HSM PIN.

Administrator, 10/11/2007

Software

All software and modifications will be available as open source

KEYGEN

- keyall, kgen, badkey, and support programs
- pkcs11-backup, pkcs11-changePIN, pkcs11-encrypt, pkcs11-random
- pkcs11 modified BIND tools: dnssec-signzone and dnssec-keygen

ZONESIGNER

- signall, zsign, and support programs

ZONEGEN

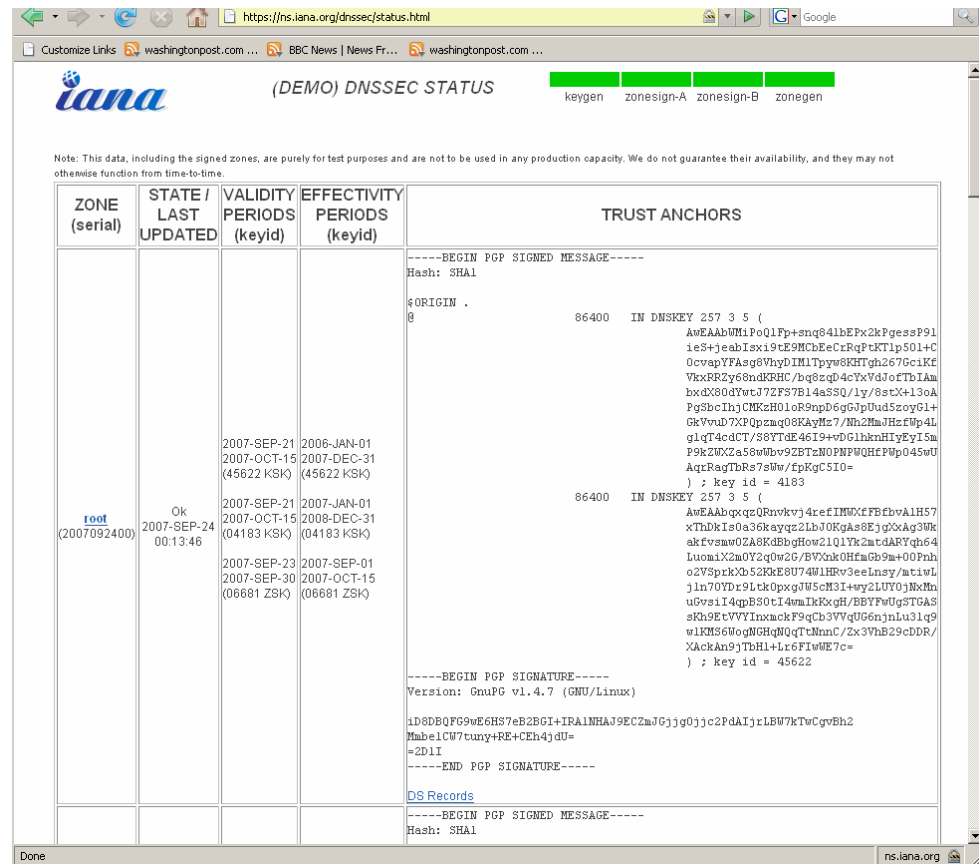
- upsite – DNSSEC status web page generator

DNSEC Status Page

<https://ns.iana.org/dnssec/status.html>

System status and publication of PGP signed trust anchors only on SSL secured site.

Domains: root, arpa, in-addr.arpa, uri.arpa, urn.arpa, iris.arpa, ip6.arpa, int



The screenshot shows the DNSSEC Status Page for the root zone. The page title is "(DEMO) DNSSEC STATUS" and it includes a note that the data is for test purposes only. The main content is a table with columns for Zone (serial), State / Last Updated, Validity Periods (keyid), Effectivity Periods (keyid), and Trust Anchors. The root zone (2007092400) is shown with a state of "Ok" and last updated on 2007-SEP-24 at 00:13:46. The trust anchors section contains a PGP signed message for the root zone, including the key ID 86400 and the key ID 4183.

ZONE (serial)	STATE / LAST UPDATED	VALIDITY PERIODS (keyid)	EFFECTIVITY PERIODS (keyid)	TRUST ANCHORS
root (2007092400)	Ok 2007-SEP-24 00:13:46	2007-SEP-21 (45622 KSK) 2007-OCT-15 (45622 KSK) 2007-SEP-21 (04183 KSK) 2007-SEP-23 (06681 ZSK) 2007-SEP-30 (06681 ZSK)	2006-JAN-01 (45622 KSK) 2007-DEC-31 (45622 KSK) 2007-JAN-01 (04183 KSK) 2007-DEC-31 (04183 KSK) 2007-SEP-01 (06681 ZSK) 2007-OCT-15 (06681 ZSK)	-----BEGIN PGP SIGNED MESSAGE----- Hash: SHA1 \$ORIGIN . \$ORIGIN 86400 IN DNSKEY 257 3 5 (AwEAAAbWU1PoQ1Fp+snq841bEfx2kPgessP91 ieS+j eabLsx19tE9McbEeCrRqPtKtLp501+C 0cvapYFAsg8VhyDIM1Tpyw6KHTgh267GcIKF VxkRRZy6ndKRHC /bq8zqD4cYxVdJofTbIAm bxkX80dYwtcJ7ZFS7B14aSSQ /ly/8stX+13oA Pg3bcIhjCMKzH0loR9npD6g9pUudSzoYGL+ GkVvD7XhQp2mq08KAYHz7 /Hk2MaJHzfWp4L q1qT4cACT /58YTB4619+00G1bkhH1yEY75m P9kZUX2a58wUw92BTzN0PFPWQHFfPp04svU AqrRagTbR7sWw /epKqCSI0=); key id = 4183 86400 IN DNSKEY 257 3 5 (AwEAAAbqxqzQrNvkj4refIMDXfFBfbvAlH57 xTbDkIe0a36kayqz2LbJ0KqAs8EjgXXAg3Wk akfvsmw02A8kDBqHow21Q1Yk2mtdARYq64 Luom1X2m0Y2q0w2G /BVYnk0Hfmg9m+00Pnh c2VSpkXb52KkE8U74u1HRv3eeLnsy /atvL jln70YDr5Ltk0pxqJ5cM3I+ry2LUY0jNkMn uGvs1I4qpB50tI4mTkKxgH /BBYFwUgSTGAS sRt9EtVYVInxckF9qCb3VYqU66nJnLu31q9 wLRMS6WogN6HqN0qTcNnnc /Zx3VhB29cDDR/ XackAn9jTbH1+Lr6F1w7eC=); key id = 45622 -----BEGIN PGP SIGNATURE----- Version: GnuPG v1.4.7 (GNU/Linux) iD8DB0FG9wE6HS7eB2EGi+IRALNHJA9EC2mJGjjg0j;c2PdAIj;Lw7kTwGvBh2 MmhelCU7tuny+RE+CEH4jdU= =2DI -----END PGP SIGNATURE----- DS Records -----BEGIN PGP SIGNED MESSAGE----- Hash: SHA1

DS Record Handling

- Will be integrated into IANA root zone management like NS records
- <https://ns.iana.org/dnssec/ds/queryds.cgi> sample
- Client side https for infrastructure (.arpa) ?

(DEMO) DNSSEC STATUS

Note: This data, including the signed zones, are purely for test purposes and are not to be used in any production capacity. We do not guarantee their availability, and they may not otherwise function from time-to-time.

DS Request Processing

Enter the TLD whose DS Record you would like included:

Please type this: **WY8JP** here: then

[PGP key](#) Comments concerning the layout, construction and functionality of this site should be sent to webmaster@iana.org.

20-JUL-2007 23:42 UTC
© 2007 The Internet Corporation for Assigned Names and Numbers.
All rights reserved.

(DEMO) DNSSEC STATUS

Note: This data, including the signed zones, are purely for test purposes and are not to be used in any production capacity. We do not guarantee their availability, and they may not otherwise function from time-to-time.

DS Request Processing

Is this information correct? Click OK if so or correct your TLD server configuration and resubmit. Once the authenticity of your request is verified and attested to, your DS records will be included in the root zone.

Your DS Record Detail:

```
IN DS 17686 5 1 9E5E81A0B71A9B6B251077F700AA730E18D712EF
se.
IN DS 17686 5 2 B78C0E213B17285C7BCC78884D81A5F09145F800C564954F856140D1 689153B9
se.
IN DS 6166 5 1 CE2B007F6D000B064B4A82E8840C19D3D09B8F8E
se.
IN DS 6166 5 2 CD9D147E24D866412216ADA5DBC257DAE6CF0FFFEF234415D6BD1114 D833F213
```

[PGP key](#) Comments concerning the layout, construction and functionality of this site should be sent to webmaster@iana.org.

20-JUL-2007 23:42 UTC
© 2007 The Internet Corporation for Assigned Names and Numbers.
All rights reserved.

Questions I have

- How's it look?
- Key handling / management? A3
- Compromised key detections and recovery in the face of disinterested users. Update vectors: A9
 - Windows update, anti-virus software updates, RFC5011, Takrem, others?
- Other suggestions?

Slide 21

A3 Fix this and oddly enough, for a man-in-the middle attack, KSK recovery might happen faster than ZSK recovery.

Difference is we can push one or all new ZSKs down a no-m-i-t-m channel but w/o rfc5011, we cant push even one new KSK down. Even with 5011, we cannot push 2 new KSKs down.

Administrator 7/12/2007
Analysis

During the validation process a resolver recursively asks for DNSKEY material then verifies it with DS records signed by the parent (I tcpdump/sniffed this). (ugly notation: A b.c = request for A record for domain b.c. K2b.c. = KSK 2 for domain b.c. (K1,K2,Z)Zb.c. = RRSIG of b.c. key bundle with zone key Zb.c.)

A b.c. ->

<- IP, (IP)Zb.c.

DNSKEY b.c. ->

<- K1b.c., Zb.c., K2b.c., (K1,K2,Z)Kb.c., (K1,K2,Z)Zb.c.

DS b.c. ->

<- DS, (DS)Zc.

DNSKEY c. ->

< K1c., Zc., K2c., (K1,K2,Z)Kc., (K1,K2,Z)Zc.

DS c. ->

<- DS, (DS)Z.

DNSKEY . ->

<- K1., Z., K2., (K1,K2,Z)K., (K1,K2,Z)Z.

Validate Z. with K1. and K2.

K1. and K2. match the trust anchors in the end user resolver

If someone compromises K2. (and assuming we notice this), we generate a new K2. which will no longer match the trust anchor in the lazy user's resolver (after caches have expired). So in this case TLDs signed with the old K2. fail. If on the other hand the ISP or another upstream provider intercepts root DNS requests and has the private K2. key, a false sense of security prevails until the trust anchors in the resolver are updated manually on the user machine, at the upstream ISP's DNS server used by its customers, or by Windows Update or other such maintenance mechanism. Although shortening the validity (RRSIG) periods over the KSKs may guard against replay attacks and a ZSK compromise (KSK RRSIG of ZSK good for about a month so for intercepted DNS root server requests, recovery from a ZSK compromise could take this long), it cannot protect against laziness and interception.

Administrator, 9/24/2007

A9 reason to keep distance between KEYGEN and ZONESIGNER: someone who has gained control of ZONESIGNER could instruct it to generate a key bundle with infinite validity period and copy private ZSK. So even w/o having the private KSK, an attacker could cache poison/impersonate for the lifetime of the KSK trust anchors in resolvers: a week to forever but typically a year.

Administrator, 9/24/2007

Try it!

Try it! “dig +dnssec -t soa . @ns.iana.org”

Questions?