DNSSEC and DANE introduction

Viktor Dukhovni

<viktor@twosigma.com>

<ietf-dane@dukhovni.org>

NYLUG | Two SIgma | October 17th 2018

DNSSEC

1. Overview
2. Best Practice
3. Metrics

NYLUG | Two SIgma | October 17th 2018

Legacy DNS issues

- DNS packets are too easy to spoof even "off path"
  - 16-bit query ID + 16 bit port number
- "On path" attackers can modify data at will
  - Caches can tamper with upstream authoritative data
- Can't trust security-relevant MX and SRV RRs

Goals

- End-to-end tamper-evident data integrity
- Intermediate caches need not be trusted
- Downgrade resistance, no false insecure delegations
- But, passive monitoring (privacy) not in scope

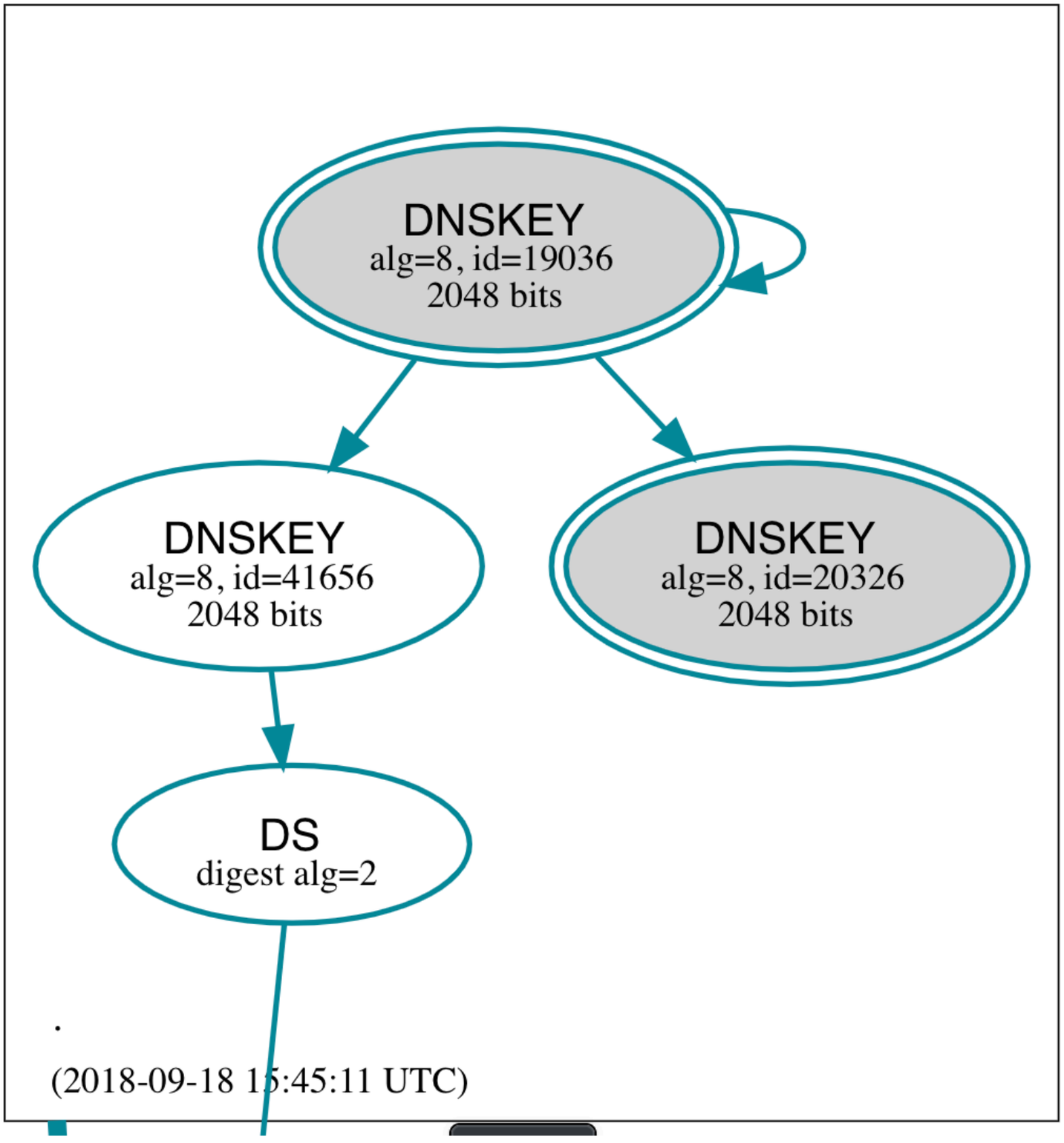
Features

- **Trust-anchor** keys validate the whole tree or a subtree
- **Signed delegation** across zone cuts
- Signatures cover non-glue RRsets
- Authenticated ***denial of existence*** (**DoE**) avoids downgrades
  - **NoData** & **NXDomain** are signed answers not errors

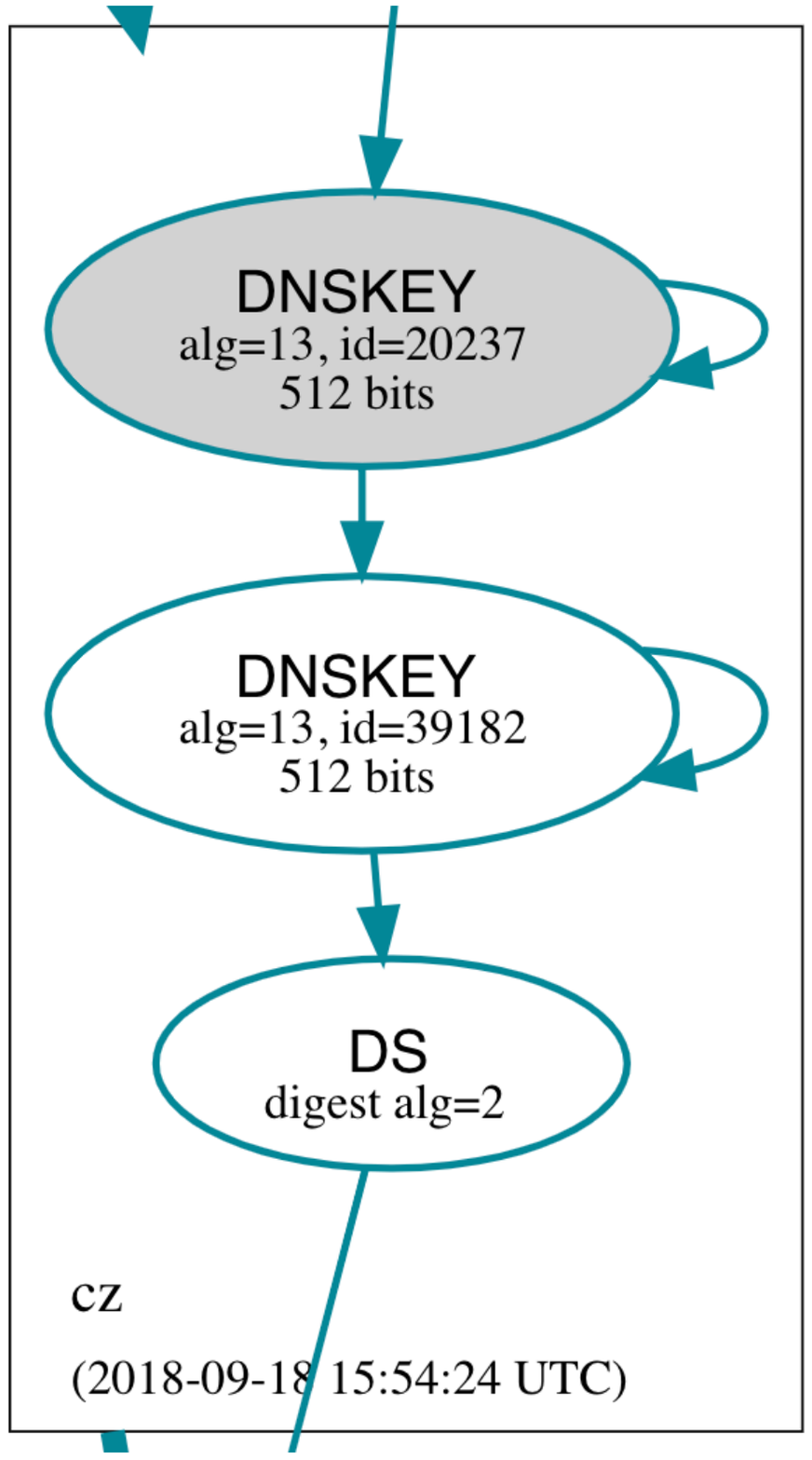
Keys

- Long term ***Key Signing Keys*** (**KSK**s) sign the *zone apex* DNSKEY RRset
- Shorter-term ***Zone Signing Keys*** (**ZSK**s) sign the zone content.

```
$ dig +noall +ans +nocl +nottl -t DNSKEY cz.
; RDATA: flags, protocol, algorithm, key
cz. DNSKEY 257 3 13 nq...DQ== ; KSK
cz. DNSKEY 256 3 13 Ww...Eg== ; ZSK
```



http://dnsviz.net/d/root/dnssec/



http://dnsviz.net/d/cz/dnssec/

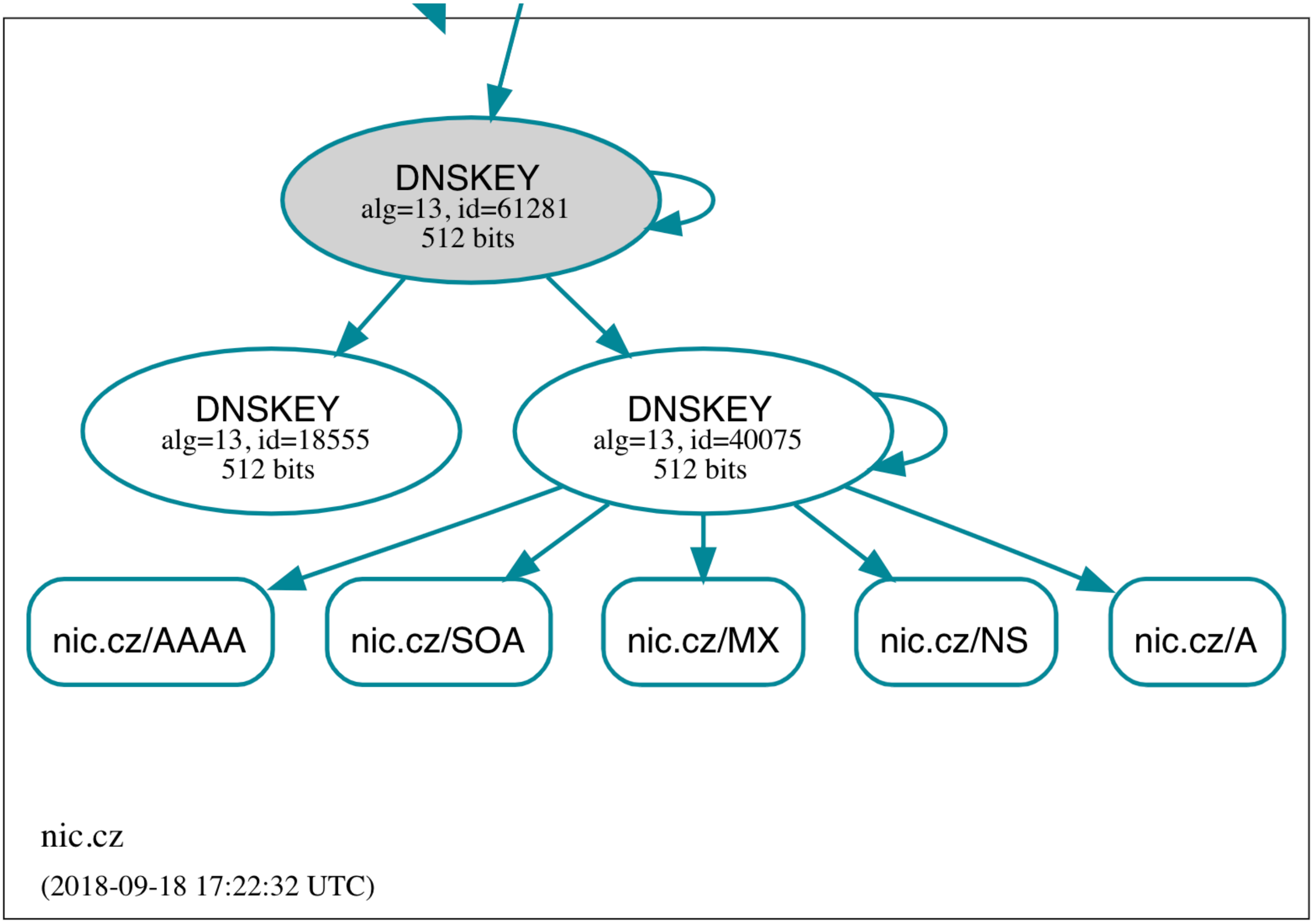
Signed delegation

- ***Delegation Signer*** (**DS**) records in parent zone validate child KSK(s)

```
$ dig +noall +ans +nocl +nottl -t DS cz.
; RDATA: keyid, keyalg, hashalg, hash
cz. DS 20237 13 2 CFF0...78E2
```

  - keyalg 13 = ECDSA P-256, hashalg 2 = SHA2-256

- Each glue NS RRset in parent needs signed DS or DoE

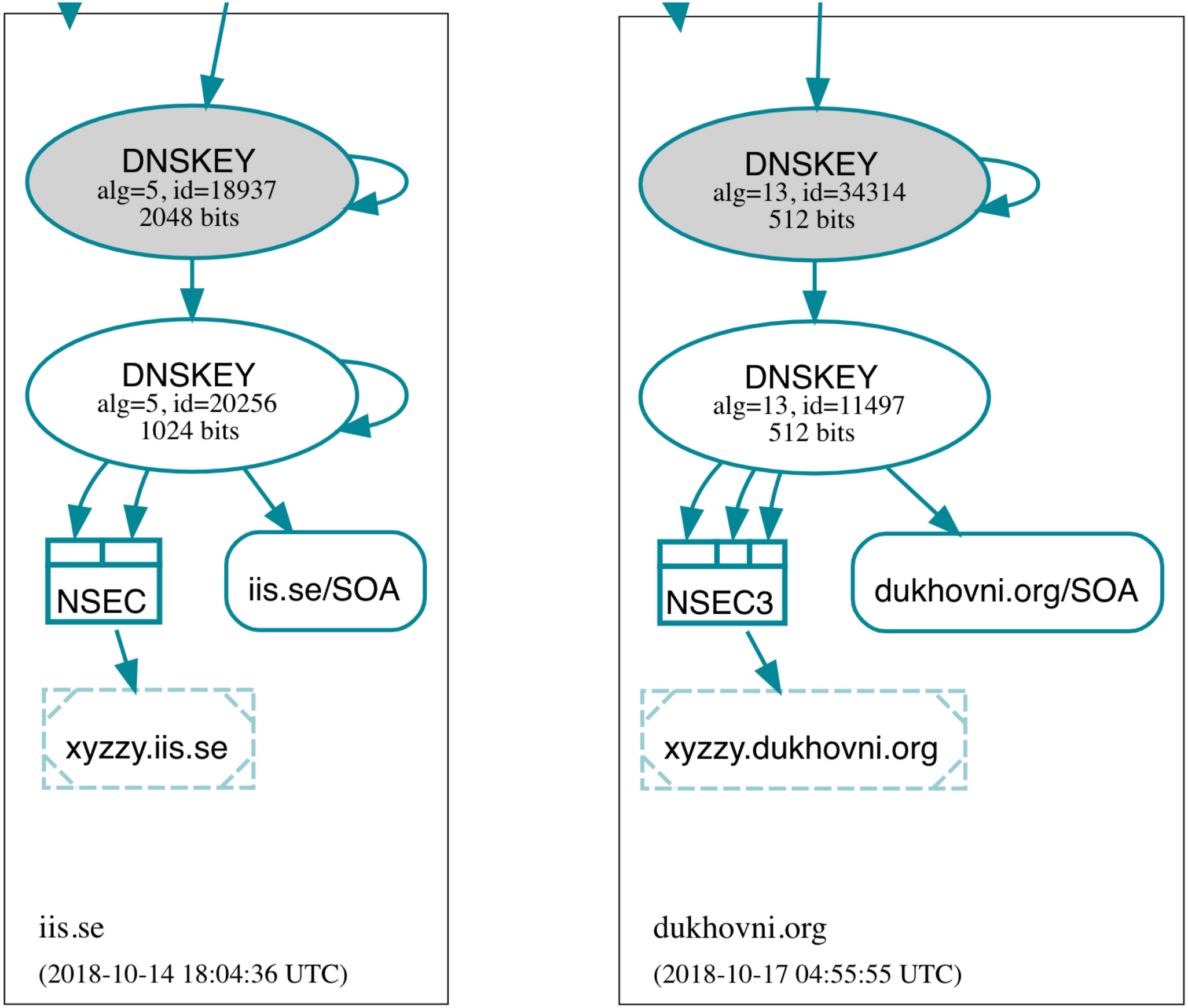

DNSKEY
alg=13, id=61281
512 bits
DNSKEY
alg=13, id=18555
512 bits
DNSKEY
alg=13, id=40075
512 bits
nic.cz/AAAA
nic.cz/SOA
nic.cz/MX
nic.cz/NS
nic.cz/A
nic.cz
(2018-09-18 17:22:32 UTC)

Denial of existence

- NSEC chain lists (node, next node, type bitmask)

```
$ dig +dnssec -t a xyzzy.iis.se
iis.se. NSEC _dmarc.iis.se. A NS SOA ...
xwin.iis.se. NSEC zkt.iis.se. NS RRSIG NSEC
```

- NSEC3 replaces ordered names with ordered hashes
  - Discourages zone walking
  - Optionally further iterated, but just once (0) is enough
  - ***Opt-out bit*** skips unsigned delegations, for lightly signed TLDs, don't use in your own zones.


DNSKEY
alg=5, id=18937
2048 bits
DNSKEY
alg=5, id=20256
1024 bits
NSEC
iis.se/SOA
xyzzy.iis.se
iis.se
(2018-10-14 18:04:36 UTC)
DNSKEY
alg=13, id=34314
512 bits
DNSKEY
alg=13, id=11497
512 bits
NSEC3
dukhovni.org/SOA
xyzzy.dukhovni.org
dukhovni.org
(2018-10-17 04:55:55 UTC)

BIND authoritative

```
options {
   ...
   key-directory "keys";
   dnssec-enable yes;
   dnssec-dnskey-kskonly yes;
   sig-validity-interval 14;
};
zone "dukhovni.org" {
   type master;
   file "master/dukhovni.org";
   auto-dnssec maintain;
   inline-signing yes;
};
```

Unbound recursive

```
server:
   module-config: "validator iterator"
   auto-trust-anchor-file: /etc/unbound/
root.key
   qname-minimisation: no
   cache-max-ttl: 7200
   cache-max-negative-ttl: 1200
```

Zone signing

```
# d=example.net
# dnssec-keygen -K "$kdir" -r /dev/urandom \
  -a ECDSAP256SHA256 -3 -f KSK $d
# dnssec-keygen -K "$kdir" -r /dev/urandom \
  -a ECDSAP256SHA256 -3 $d
# chown named "$kdir"/K$d.+013+*
```

```
    auto-dnssec maintain; // add to zone stanza to
    inline-signing yes;   // automate signing

# rndc reconfig; rndc loadkeys $d
# salt=$(openssl rand -hex 8 | tr a-f A-F)
# rndc signing -nsec3param 1 0 0 $salt $d.
# dig +noall +ans -t dnskey $d | \
  dnssec-dsfromkey -2 -f - $d
```

https://securityblog.switch.ch/2014/11/13/dnssec-signing-your-domain-with-bind-inline-signing/

ZSK rollover

- Don't pre-expire keys, set expiration "just-in-time" when adding replacement keys.

- If replacement is postponed, disaster avoided.

```
# d=example.net; old=K$d.+013+NNNNN
# rnd=/dev/urandom
# dnssec-settime -I +2890mi -D +8d $old
# dnssec-keygen -r $rnd -i 2d -S $old.key
# chown named K$d.+013*
# rndc loadkeys $d
```

Best Practice

- Set public server max EDNS0 buffer to avoid fragments (especially with IPv6, ~1216 bytes). [Coincidence? Root zone DNSKEY: 1169, NS: 1097, DoE ~1020]
- Algorithms 13 (P-256) and 8 (RSA with SHA256)
- RSA KSK ≥ 1536 bits, ZSK ≥ 1280 bits
- EdDSA not yet supported by most resolvers
- Automate zone signing, 7–30 day signature lifetime

DNSSEC History

- 2008 Dan Kaminsky BlackHat talk validates DJB
- Resolvers add port randomization counter-measure
- .ORG signed in 2009
- ICANN signs root zone in 2010 (KSK-2010, id 19036)
- .COM (2011) and other gTLDs follow (just .AERO left)
- 125 of 247 ccTLDs presently signed

Best Practice

- **Monitor** your deployment
  - Check for signatures too close to expiration
  - Check for working denial of existence
  - Slave nameserver synchronization
  - Firewalls must not drop CAA, TLSA, CDS, ... queries
- Rotate KSK keys ~annually, ZSKs ~90 days

Best Practice (Resolver)

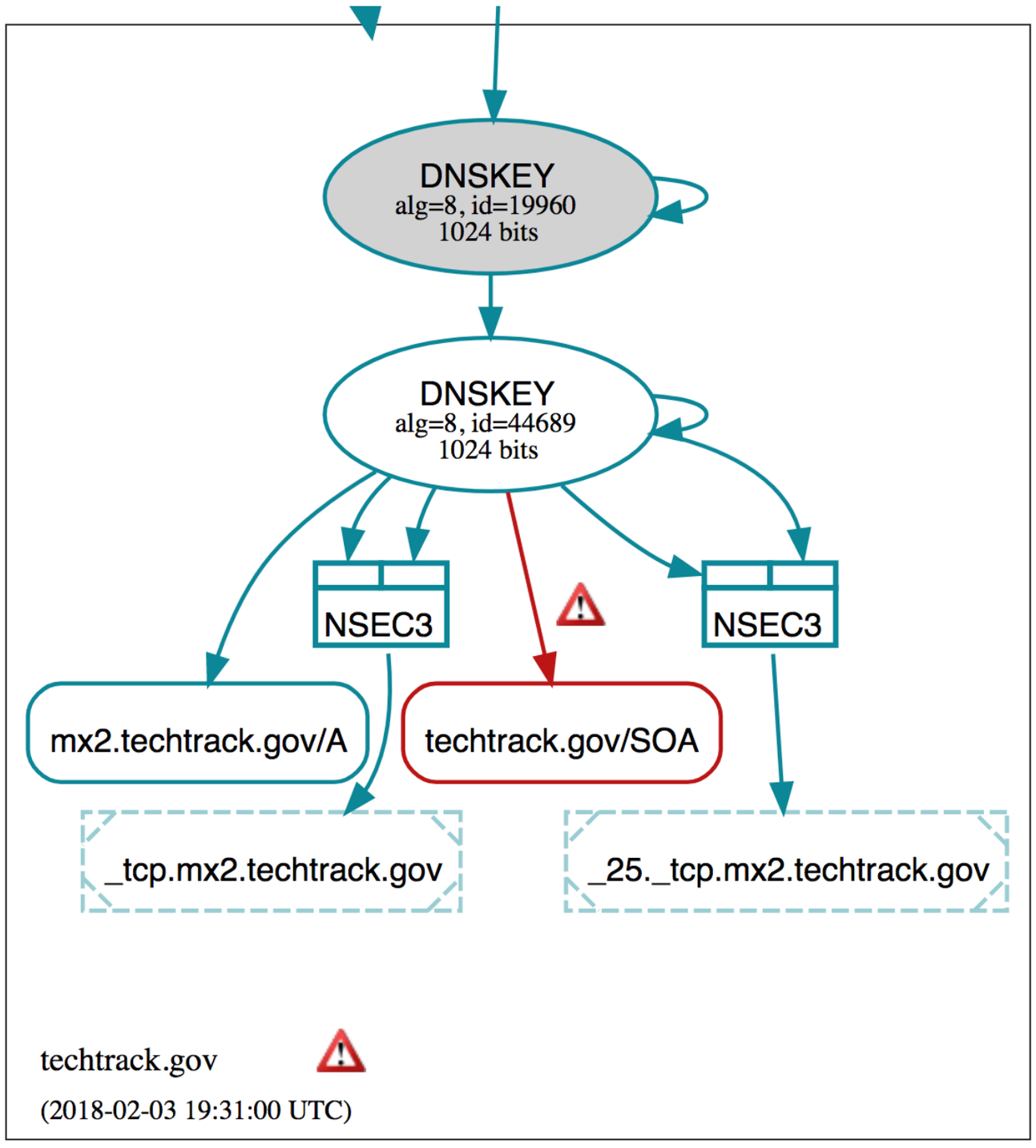
- **Monitor** resolution of "." and some key TLDs
- Make sure trust-anchor rollover can work
    - Updates by running resolver, and file permissions
    - Timing of boot-time updates (network access)
- Enable validation
- Cap cache TTLs (defaults: unbound: 1 day, BIND: 7)
- Perhaps slave the root and .arpa zones.

Checklist

- Keep name-server software up to date
- Test apex wildcard A or wildcard CNAMEs
- Test empty non-terminals (_tcp.example.com)
- Avoid SOA serial number changes after signing
- Avoid NSEC3 opt-out in most zones
- Avoid high NSEC3 (extra) iteration counts (0 is BCP!)

https://lists.dns-oarc.net/pipermail/dns-operations/2017-December/017127.html
https://lists.dns-oarc.net/pipermail/dns-operations/2018-January/017173.html

Check DNSViz



http://dnsviz.net/d/_25._tcp.mx2.techtrack.gov/WnYN-A/dnssec/

Metrics

- ~250 million domain sample
- ~9 million signed at "org-level", ~10 million estimated
- ~1.8 million ECDSA P-256, rest RSA
- KSK typically 2048-bit, ZSK typically 1024-bit

Top TLDs

DNSSEC domains x1000	TLD
3,089	NL
935	COM
820	SE
597	CZ
507	BR
503	EU
472	PL
411	FR
377	NO
145	BE
130	NET
129	NU
119	HU
97	ORG
85	DE
500	other

Reliability

- Breakage largely at parked domains
  - Many just lame delegations (ordinary DNS outage).
- Denial of existence problem only at ~500 domains
- Low breakage % TLDs: .香港 (0.00), .BR (0.04), .HK (0.06)
- High breakage TLDs: .BANK (41.9), .NRW (11.5), .RU (9.6)

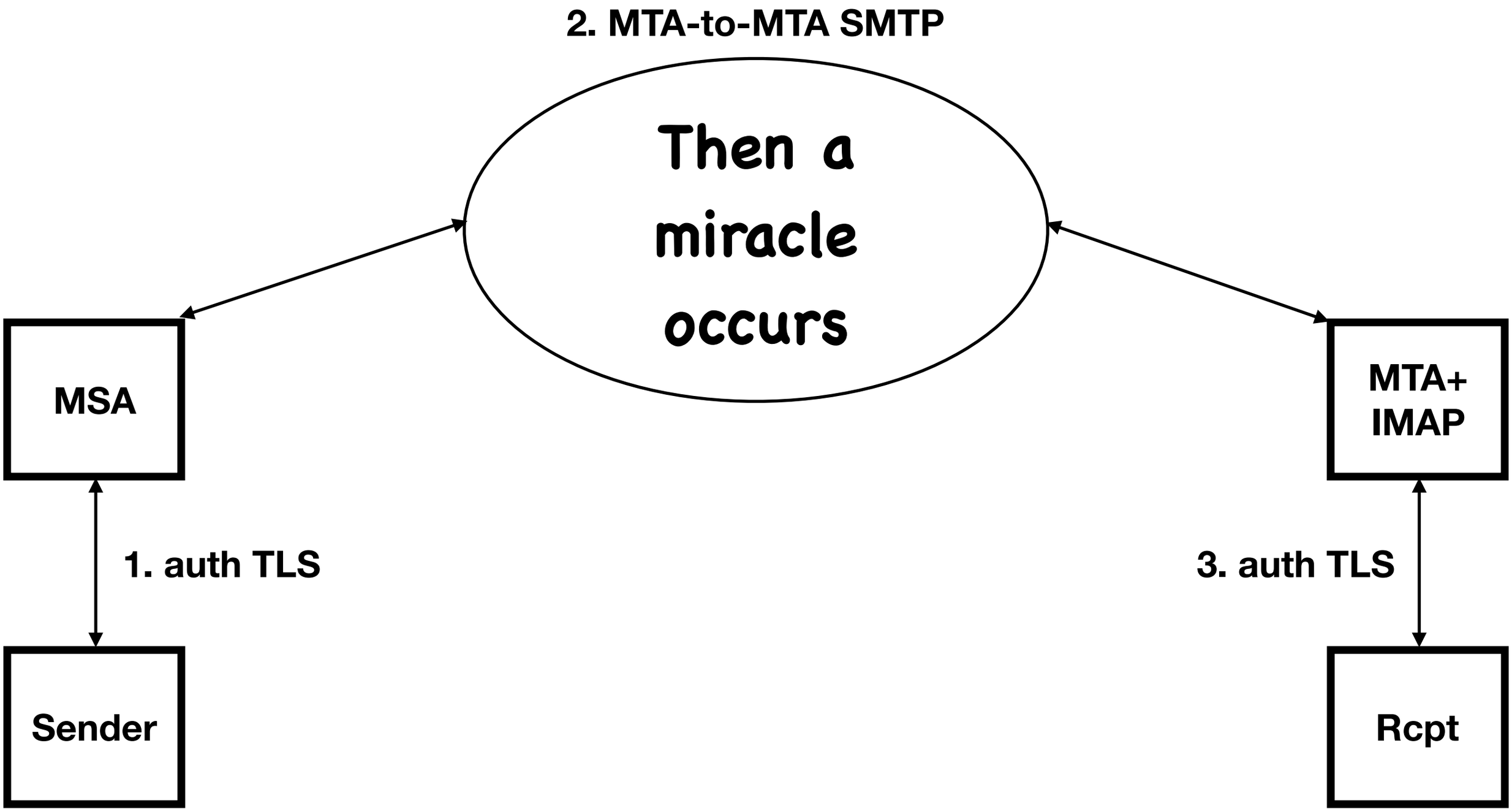
Q&A

- Before we move on to **DANE** (stands for: DNS-Based Authentication of Named Entities)

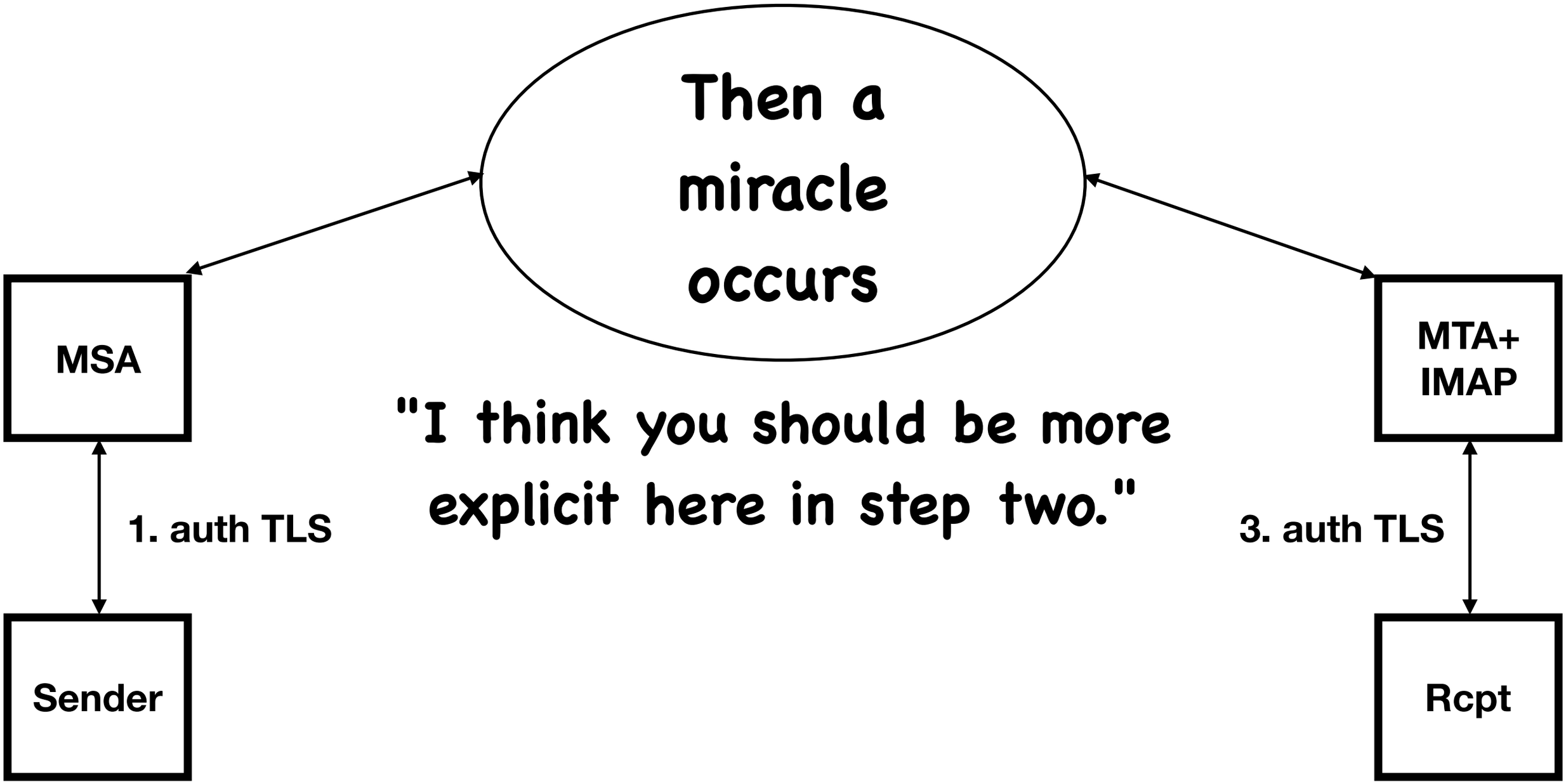
- Any DNSSEC questions?

Email Security

Email Security



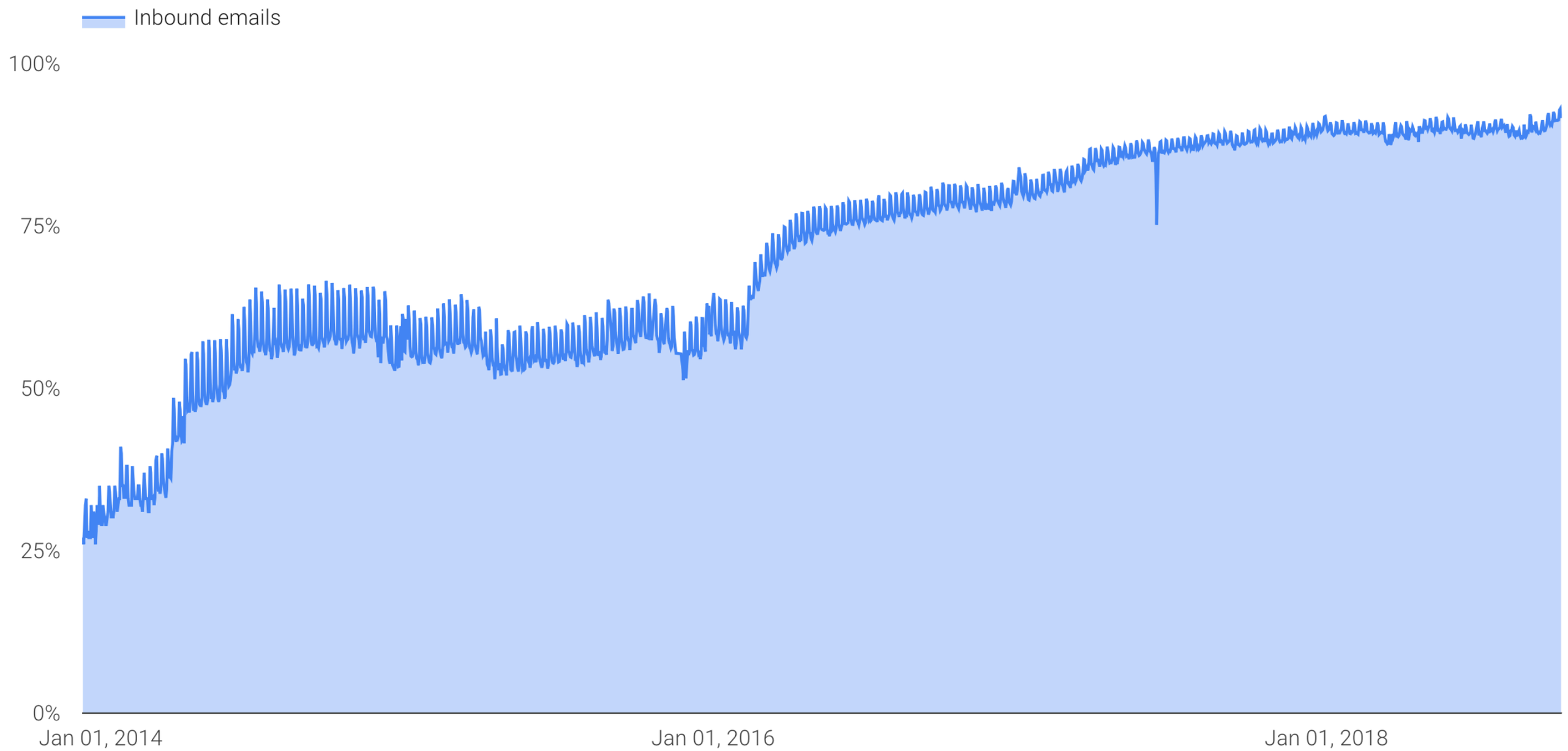
Email Security



"I think you should be more explicit here in step two."

Email Security

Gmail STARTTLS (~92% now)



https://transparencyreport.google.com/safer-email/overview

SMTP is not like HTTPS

https://tools.ietf.org/html/rfc7672#section-1.3

- Must trust DNS for authentic MX hosts
- Web CA trust would be problematic
  - Too many CAs to trust, but no user to "click OK"
  - Can't avoid trusting them all

SMTP TLS wish list

- If everyone encrypts security is there when needed
- Resist active attacks:
  - Downgrade-resistant, even on first contact
  - Work in mixed environment with legacy systems
  - Securely signals which peers to encrypt
  - Robust peer authentication

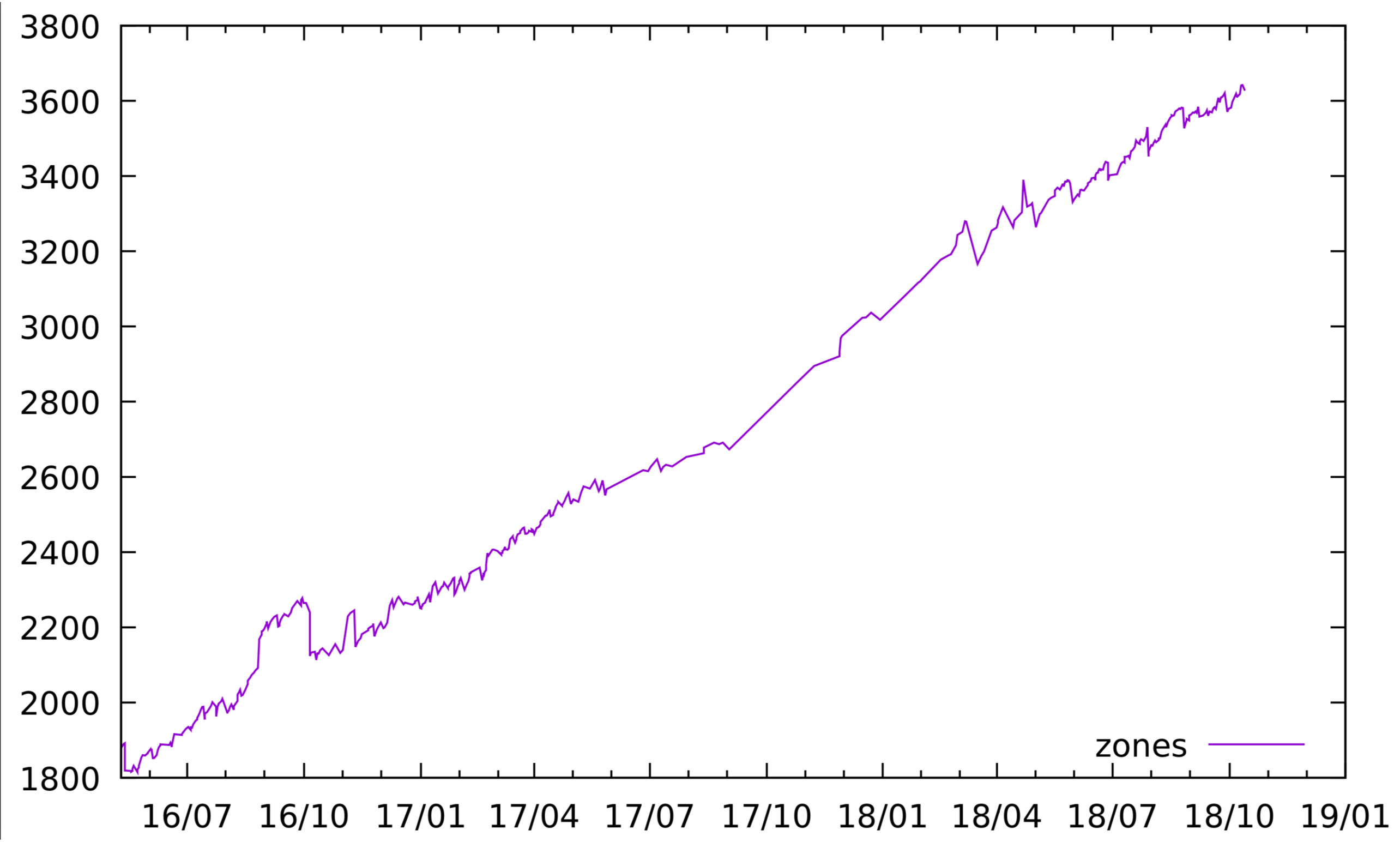
DANE (MTA-to-MTA SMTP)

- STARTTLS only protects email against passive monitoring
- DANE (RFC7671, RFC7672) adds active attack resistance
  - DNSSEC guards against MX record forgery
  - Presence of DANE TLSA records is a contract to support STARTTLS **and** present a matching cert chain
  - Authenticate domain control via DNSSEC, no extraneous trusted third parties
  - DNSSEC ensures downgrade protection

Well known DANE domains

gmx.at
transip.be
travelbirdbelgique.be
nic.br
registro.br
gmx.ch
open.ch
anubisnetworks.com
geektimes.com
gmx.com
habr.com
mail.com
societe.com
solvinity.com
t-2.com
trashmail.com
xfinity.com
xfinitymobile.com
active24.cz
clubcard.cz
cuni.cz
cvc.cz
destroystores.cz
itesco.cz
klubpevnehozdravi.cz
knizni-magazin.cz
localssrcapp.cz
nic.cz
optimail.cz
smtp.cz
bayern.de
bund.de
elster.de
fau.de
freenet.de
gmx.de
jpberlin.de
kabelmail.de
lrz.de
mail.de
posteo.de
ruhr-uni-bochum.de
tum.de
uni-erlangen.de
unitybox.de
unitymedia.de
web.de
dk-hostmaster.dk
egmontpublishing.dk
netic.dk
tilburguniversity.edu
transip.eu
insee.fr
octopuce.fr
comcast.net
dd24.net
dns-oarc.net
gmx.net
habramail.net
hr-manager.net
inexio.net
mpssec.net
mylobu.net
t-2.net
transip.net
xs4all.net
xworks.net
ardanta.nl
bhosted.nl
bit.nl
boozyshop.nl
deltion.nl
hierinloggen.nl
hr.nl
hro.nl
interconnect.nl
intermax.nl
markteffectmail.nl
ouderportaal.nl
overheid.nl
pathe.nl
politie.nl
previder.nl
rotterdam.nl
transip.nl
truetickets.nl
uvt.nl
verschoore.nl
xs4all.nl
domeneshop.no
handelsbanken.no
rushtrondheim.no
webcruitermail.no
aegee.org
debian.org
freebsd.org
gentoo.org
ietf.org
isc.org
lazarus-ide.org
netbsd.org
openssl.org
samba.org
torproject.org
asf.com.pt
handelsbanken.se
iis.se
minmyndighetspost.se
skatteverket.se
t-2.si
mail.co.uk
govtrack.us

#Zones of DANE MX hosts



Deploying DANE

- Deploying DNSSEC is the main barrier
- Coordinating TLSA records and cert chain may look hard
- We'll make it easy

Inbound DANE

- Need some STARTTLS-capable SMTP server
- DNSSEC-signed MX records
- DNSSEC-signed TLSA records for each MX host
  - Provider's responsibility if MX hosts outsourced!
    - Including management of key and certificate rotation

Outbound DANE

- Need DNSSEC validating resolver, **local** to the MTA
- DANE-enabled MTA (Postfix, Exim, Halon, PowerMTA, mailinabox.email, Cisco ESA, ...)
- Enable DANE as documented
- Perhaps a few policy exceptions:

`https://github.com/danefail/list`

Postfix + DANE

- Working server-side STARTTLS, e.g. Let's Encrypt with `fullchain.pem`

- DNSSEC + TLSA records always matching cert chain

- Local (loopback) validating resolver + `main.cf`:

```
dbtype = ${default_database_type}
cfgdir = ${config_directory}
indexed = ${dbtype}:${cfgdir}/

smtp_dns_support_level = dnssec
smtp_tls_security_level = dane
smtp_tls_policy_maps = ${indexed}tls-policy
```

TLSA records

- **3 1 1: certificate usage DANE-EE(3):**
  - Publishes end-entity (server) public key SHA256 hash
- **2 1 1: certificate usage DANE-TA(2):**
  - Publishes trust-anchor (CA) public key SHA256 hash
  - If the CA is secure enough for your needs
- Rest of record is hash value:

```
$ dig +nosplit +short -t tlsa _25._tcp.mail.ietf.org
3 1 1 0C72....D3D6
```

TLSA record rollover

- Need matching TLSA in place when chain is updated
- TLSA records can include present and future values
- Publish **keys** well in advance of obtaining certificates
- Two models (EE == end-entity, TA == trust-anchor):
  - EE Key + Next EE Key: (3 1 1 + 3 1 1)
  - EE Key + TA Key: (3 1 1 + 2 1 1)

Current + Next

- Generate next key when deploying current key and cert
- Deploy new chain, and publish new TLSA records:

```
_25._tcp.mx.example.com. IN TLSA 3 1 1 curr-pubkey-sha256
_25._tcp.mx.example.com. IN TLSA 3 1 1 next-pubkey-sha256
```

- Weeks later, obtain certificate for pre-generated *next* key†
  - But first, make sure TLSA record is already in place
- Repeat!

† With Let's Encrypt, use "--csr" option to use new key,
or else "`certbot renew --reuse-key`" to use current key.
Better support in certbot coming soon.

Current + Issuer CA

- Publish TLSA RRs for server key & issuer CA key

```
_25._tcp.mx.example.com. IN TLSA 3 1 1 ee-pubkey-sha256
_25._tcp.mx.example.com. IN TLSA 2 1 1 ta-pubkey-sha256
```

- Deploy certificates from same CA, if EE key changes:
  - Promptly update **3 1 1** hash to match new EE key
- If CA key changes, keep previous EE key
  - Obtain cert for previous key from new CA
  - Promptly update **2 1 1** hash to match new CA key

Reliability

- Automate:
  - TLSA record updates and zone re-signing
  - Key rollover
  - Cert chain acquisition and deployment
  - `certbot` "`--csr`" and "`--reuse-key`" options
- Have working contacts in WHOIS, SOA, postmaster, TLSRPT (RFC8460).

Monitor

- DNSSEC DS and DNSKEY records
- STARTTLS availability
- TLSA records matching of live cert chain
- Check TLSA RRs for all certificate types: RSA, ECDSA, ... (if more than one configured)

Operational BCP

- Publish the current and next TLSA record
- Don't offer STARTTLS selectively to just some clients
- Use a separate certificate for each MX host
- Stagger certificate rotation for separate MX hosts

DANE tools

- https://dane.sys4.de/ and dane-users@sys4.de
- https://github.com/letoams/hash-slinger
- https://github.com/PennockTech/smtpdane
- https://github.com/vdukhovni/danecheck
- openssl s_client†

† Requires OpenSSL 1.1.0 or later

```
$ danesmtp() {
   local host=$1; shift
   local opts=(-starttls smtp -connect "$host:25" \
               -verify 9 -verify_return_error -brief \
               -dane_ee_no_namechecks -dane_tlsa_domain "$host")
   set -- $(dig +short +nosplit -t tlsa "_25._tcp.$host" |
             egrep -i '^[23] [01] [012] [0-9a-f]+$')
   while [ $# -ge 4 ]
   do
      opts=("${opts[@]}" "-dane_tlsa_rrdata" "$1 $2 $3 $4")
      shift 4
   done
   (sleep 1; printf "QUIT\r\n") | openssl s_client "${opts[@]}"
}
```

```
$ danesmtp mail.ietf.org
...
Protocol version: TLSv1.2
Ciphersuite: ECDHE-RSA-AES256-GCM-SHA384
Peer certificate: OU = Domain Control Validated, CN = *.ietf.org
Hash used: SHA512
Verification: OK
DANE TLSA 3 1 1 ...e7cb23e5b514b56664c5d3d6 matched EE certificate at depth 0
...

$ echo $?
0
```

Coexisting with DANE

- DANE senders skip MX hosts that ***fail*** TLSA lookups
- When all MX hosts are skipped, delivery is deferred
- For DNSSEC-signed domains **without** TLSA records:
  - TLSA Denial of Existence (DoE) must function correctly
- DANE is first application protocol to need reliable DoE

No DNS RRtype filters

- Some firewalls offer misguided filtering features, blocking TLSA, CAA, CDS, ... lookups
- These break more than DANE
- Avoid filters that block queries for some record types
- Monitor correct responses for unexpected types:

```
$ dig -t TYPE12345 example.com.          -> NODATA
$ dig -t TYPE12345 n.x.example.com.      -> NXDomain
```

https://tools.ietf.org/html/draft-ietf-dnsop-no-response-issue

DANE SMTP Survey

- Monitors domains directly delegated from public suffixes
- Notifies operators of botched key/cert rotation
- Sourced from ICANN CZDS, Verisign, FarSight Security, SIDN, `https://scans.io/`, open access for .se, .nu, .fr, ...
- Covers ~250 million candidate domain names
- Captures DS, DNSKEY, MX, A, AAAA, TLSA records
- Captures certificate chains of MX hosts

Survey Stats

- 8.9 million domains with DNSSEC-validated MX
- 323 thousand domains with DANE SMTP
- 10s of millions of users (gmx.de, web.de, comcast.net)
- 5520 DANE MX hosts in 3627 zones
- ~500 domains with TLSA record lookup problems
- ~250 domains with wrong TLSA records or no STARTTLS

MTA-STS

- MTA-STS: compromise for the DNSSEC-challenged
  - Still can and **should** prefer DANE ***outbound***
  - Authenticates domain control via CA leap of faith!
  - Vulnerable to MiTM at cert bootstrap
  - Vulnerable to weakest root CA, and unauthorized certs
  - Open to downgrade on first (or irregular) contact
  - Complex mix of HTTPS, unsigned DNS and SMTP