DANE Overview

DNS OARC 38
Sunday, July 31st 2022
Philadelphia, PA, USA
Shumon Huque, Viktor Dukhovni

With the increasing deployment of DNSSEC, new uses are emerging that leverage the DNS to store and verify cryptographic keying material (like public keys, certificates, fingerprints, etc). The DANE (DNS-based Authentication of Named Entities) protocol and new DNS records like TLSA are among the principal enablers of these uses. This session will provide an overview of DANE and what applications can use DANE today and in the near future, and describe a project that surveys DANE deployment.

DANE

DNS-based **A**uthentication of **N**amed **E**ntities

Employing DNSSEC to securely associate cryptographic keys and/or certificates with domain names for application services, using signed DNS records.

Applications can then securely obtain, verify, and use those keys in application security protocols.

What problems does it hope to solve?

Secure association of domain names with cryptographic keys, using a system that naturally supports namespace constraints, so that only the domain owners themselves can issue and manage these associations.

Provide a complete alternative or replacement for the Public CA system.

or apply constraints on the use of Public CA issued certificates.

Enable applications to use certificate features that aren’t supported by the Public CA system.

Enable applications to use authenticated raw public keys associated with domain names.

TLS and the Internet PKI

A very large number of Internet application security protocols authenticate server names with X.509 certificates (RFC 5280); many of them using the underlying TLS layer

HTTP, IMAP, SMTP, SIP, XMPP, …

These certificates are issued and signed by the Internet PKI, composed of a set of globally trusted public “Certification Authorities” (CA).

Public CA issues - unconstrained scope

Applications need to trust a large number of global root CAs.

No namespace constraints! Any CA can issue certificates for any entity.

Our collective security is equal to the weakest one.

Furthermore many root CAs issue subordinate CA certificates to their customers, again with no namespace constraints.

Excellent paper from 2013: Analysis of the HTTPS Certificate Ecosystem: https://conferences.sigcomm.org/imc/2013/papers/imc257-durumericAemb.pdf

Public CA issues - revocation

Lack of effective revocation

Long validity period - even LetsEncrypt is 3 months.

CRL (Certificate Revocation Lists) - ungainly and not real time

OCSP (Online Certificate Status Protocol) - real time, but privacy leaking, and not even universally used

Stapled OCSP (RFC 6961) - addresses privacy threat, but not widely deployed. Needs ‘must staple’ extension too (7633) to be secure, which is difficult to deploy without wide adoption, and doesn’t solve non TLS use cases.

Public CA - functional deficiencies

Most CAs aren’t capable of issuing anything other than the most basic capabilities (e.g. alternate name forms or other extensions)

How can we support more advanced features, such as other subject alternative name forms like URI, SRVName, to better compartmentalize the security of application services running at the same domain name? We can’t today.

Public CAs basically support only DNS names and sometimes IP addresses and email addresses as identities.

(Examples: xmpp:node@example.com; _smtp.blah.example.com; …)

Fundamental reliance on the DNS

The Web/Internet PKI ultimately relies on domain names. Application services are all identified by domain names. These names need to be trusted anyway.

Domain Validated certificates are very common place.

Even Org validated or DV certificates ultimately need a way associate an organizational identity with a domain name.

DNSSEC provides a solution to trusting domain names. And DANE enables the secure mapping of domain names to cryptographic credentials for apps.

Public CA security incidents

Too many to list comprehensively, but here's a sample

- Comodo
  http://arstechnica.com/security/2011/03/how-the-comodo-certificate-fraud-calls-ca-trust-into-question/
- DigiNotar
  http://www.dutchnews.nl/news/archives/2012/11/diginotar_hack_made_possible_a.php
- http://googleonlinesecurity.blogspot.com/2011/08/update-on-attempted-man-in-middle.html

Public CA security incidents

continued …

- Trustwave http://www.computerworld.com/s/article/9224082/Trustwave_admits_issuing_man_in_the_middle_digital_certificate_Mozilla_debates_punishment
- TurkTrust: http://googleonlinesecurity.blogspot.com/2013/01/enhancing-digital-certificate-security.htm
- TeliaSonera: http://www.theregister.co.uk/2013/04/16/mozilla_threatens_teliasonera/

Public CA security incidents

continued …

- ANSSI: http://googleonlinesecurity.blogspot.com/2013/12/further-improving-digital-certificate.html
- Comodo: http://arstechnica.com/security/2015/03/bogus-ssl-certificate-for-windows-live-could-allow-man-in-the-middle-hacks/
- CNNIC: http://googleonlinesecurity.blogspot.com/2015/03/maintaining-digital-certificate-security.html

Public CA security incidents

continued …

- Symantec: https://groups.google.com/forum/m/#!topic/mozilla.dev.security.policy/Hkyq_09EDYE
- WoSign: https://groups.google.com/forum/m/#!topic/mozilla.dev.security.policy/k9PBmyLCi8I
- Symantec https://groups.google.com/forum/m/#!msg/mozilla.dev.security.policy/fyJ3EK2YOP8/chC7tXDgCQAJ

Public CA security incidents

continued …

- Symantec: https://groups.google.com/forum/m/#!topic/mozilla.dev.security.policy/Hkyg_09EDYE
- WoSign: https://groups.google.com/forum/m/#!topic/mozilla.dev.security.policy/k9PBmyLCi8I
- Symantec https://groups.google.com/forum/m/#!msg/mozilla.dev.security.policy/fyJ3EK2YOP8/chC7tXDgCQAJ

Public CA security incidents

continued …

- Digicert: https://bugzilla.mozilla.org/show_bug.cgi?id=1650910

Namespace constraints in PKI?

Technically supported in the PKIX (Internet PKI) protocol spec (see “Name Constraints” extension in RFC 5280, Section 4.2.1.10).

But these are very seldom used- sometimes for subordinate enterprise CAs.

Type specificity. Lack of criticality marking.

Not generally amenable to the Internet/Web PKI business model were each CA wants to be issue certificates for a global population of customers.

We’d need a hierarchical Internet PKI to usefully use this capability (in which case, you might as well use DNSSEC)

Certificate Transparency (CT)?

CT specifies cryptographically verifiable and unalterable logs of issued certificates by public CAs.

This can be used to retroactively detect fraudulently or mistakenly issued certificates and take action.

Band-aid. Ideally, we need to have a system that prevents these kinds of mis-issuance in the first place.

Also who operates these logs? We have yet another set of 3rd parties to trust.

What about CAA records?

CAA (RFC 8659: Certificate Authority Authorization resource record)

Zone owner publishes a CAA record at their domain authorizing only specific CAs

May help prevent “accidental” mis-issuance of certificates by other well behaved CAs.

Cannot solve the malicious CA problem.

CA issuer side check only.

How can DANE help?

Certificates and public keys (or more typically their hashes) are placed in the DNS where they can be authenticated with DNSSEC.

DNS has hierarchical and decentralized administration with a single root trust anchor (rather than a large number of unconstrained roots).

Namespace constraints are inherent.

Much more timely revocation mechanisms (shorter TTLs and record removal)

Better suited to applications that use DNS for indirection (MX, SRV, SVCB, ..).

Is it practical though?

Deployed infrastructure is becoming real. The DNS root and most TLDs are already signed. So organizations can sign their own zones and establish a complete chain of trust from the root zone trust anchor.

Validation is also widespread and growing.

However, ..

- DNSSEC deployment under the TLDs is quite sparse (~ 5% eTLD+1)
- Application protocols need to be updated to work with DANE (some can already, others need protocol revisions, implementations, and adoption) - mixed story there, but it’s early days.

DANE Protocol Specifications (for reference)

RFC 6698: DANE and TLSA record specification (August 2012)

RFC 7671: DANE Protocol: Updates & Operational Guidance

RFC 7672: SMTP Security via opportunistic DANE TLS

RFC 7673: Using DANE TLSA records with SRV records

RFC 7929: DANE Bindings for OpenPGP

RFC 8162: DANE for S/MIME Certificates

DANE TLSA Record

RFC 6698: DNS-based Authentication of Named Entities (DANE) Protocol for Transport Layer Security

Defines a new DNS record type “TLSA”, that can be used for better & more secure ways to authenticate SSL/TLS certificates

- By specifying constraints on which CA can vouch for a certificate, or which specific PKIX end-entity certificate is valid
- By specifying that a service certificate or a CA can be directly authenticated in the DNS itself.

See RFC 7671 for updates and additional operational guidance.

TLSA record parameters

Usage field:

```
    0   PKIX-TA: CA Constraint
    1   PKIX-EE: Service Certificate Constraint
    2   DANE-TA: Trust Anchor Assertion
    3   DANE-EE: Domain Issued Certificate
```

Selector field:

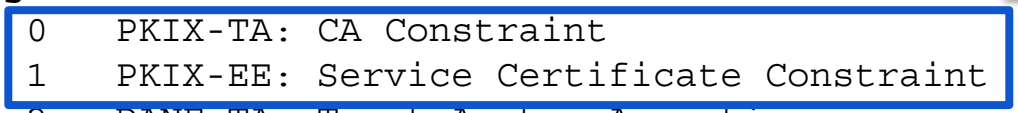
```
    0   Match full certificate
    1   Match only SubjectPublicKeyInfo
```

Matching type field:

```
    0   Exact match on selected content
    1   SHA-256 hash of selected content
    2   SHA-512 hash of selected content
```

Certificate Association Data: raw cert or key data in hex

TLSA record parameters



Co-exists with and Strengthens Public CA system

Usage field:

```
0   PKIX-TA: CA Constraint
1   PKIX-EE: Service Certificate Constraint
2   DANE-TA: Trust Anchor Assertion
3   DANE-EE: Domain Issued Certificate
```

Selector field:

```
0   Match full certificate
1   Match only SubjectPublicKeyInfo
```

Matching type field:

```
0   Exact match on selected content
1   SHA-256 hash of selected content
2   SHA-512 hash of selected content
```

Certificate Association Data: raw cert or key data in hex

TLSA record parameters

Usage field:

```
    0   PKIX-TA: CA Constraint
    1   PKIX-EE: Service Certificate Constraint
    2   DANE-TA: Trust Anchor Assertion
    3   DANE-EE: Domain Issued Certificate
```

Selector field:

```
    0   Match full certificate
    1   Match only SubjectPublicKeyInfo
```

Matching type field:

```
    0   Exact match on selected content
    1   SHA-256 hash of selected content
    2   SHA-512 hash of selected content
```

```
Certificate Association Data: raw cert or key data in hex
```

Operation without Public CAs

Usage types elaboration

0 PKIX-TA: CA Constraint
Specify which CA should be trusted to authenticate the certificate for the service. Full PKIX certificate chain validation needs to be performed.

1 PKIX-EE: Service Certificate Constraint
Define which specific service certificate (“EE cert”) should be trusted for the service. Full PKIX cert validation needs to be performed.

2 DANE-TA: Trust Anchor Assertion
Specify a domain operated CA which should be trusted independently to vouch for the service certificate.

3 DANE-EE: Domain Issued Certificate
Define a specific service certificate for the service at this domain name.

```
_25._tcp.mail.example.com. IN TLSA (
          3 1 1  d2abde240d7cd3ee6b4b28c54df034b9
                 7983a1d16e8a410e4561cb106618e971 )
```

port, protocol, domain name

```
_25._tcp.mail.example.com. IN TLSA (
                3 1 1  d2abde240d7cd3ee6b4b28c54df034b9
                       7983a1d16e8a410e4561cb106618e971 )
```

```
_25._tcp.mail.example.com. IN TLSA (
                3 1 1  d2abde240d7cd3ee6b4b28c54df034b9
                       7983a1d16e8a410e4561cb106618e971 )
```

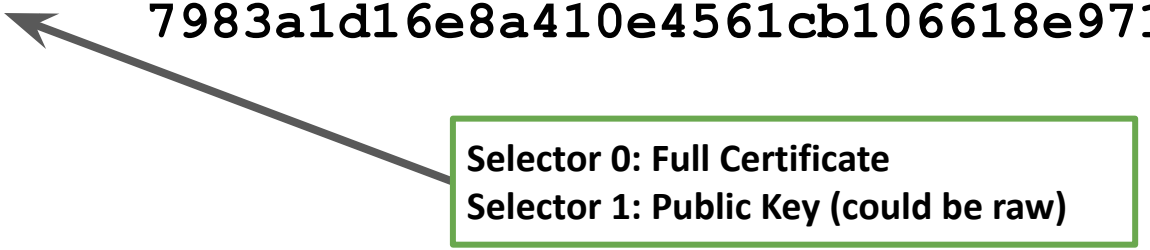
Parameters: Usage, Selector, Matching-Type

```
_25._tcp.mail.example.com. IN TLSA (
                3 1 1  d2abde240d7cd3ee6b4b28c54df034b9
                       7983a1d16e8a410e4561cb106618e971 )
```

Parameter: Usage

Usage 0: PKIX-CA: CA Constraint
Usage 1: PKIX-EE: Service Cert Constraint
Usage 2: DANE-TA: Trust Anchor Assertion
Usage 3: DANE-EE: Domain Issued Certificate

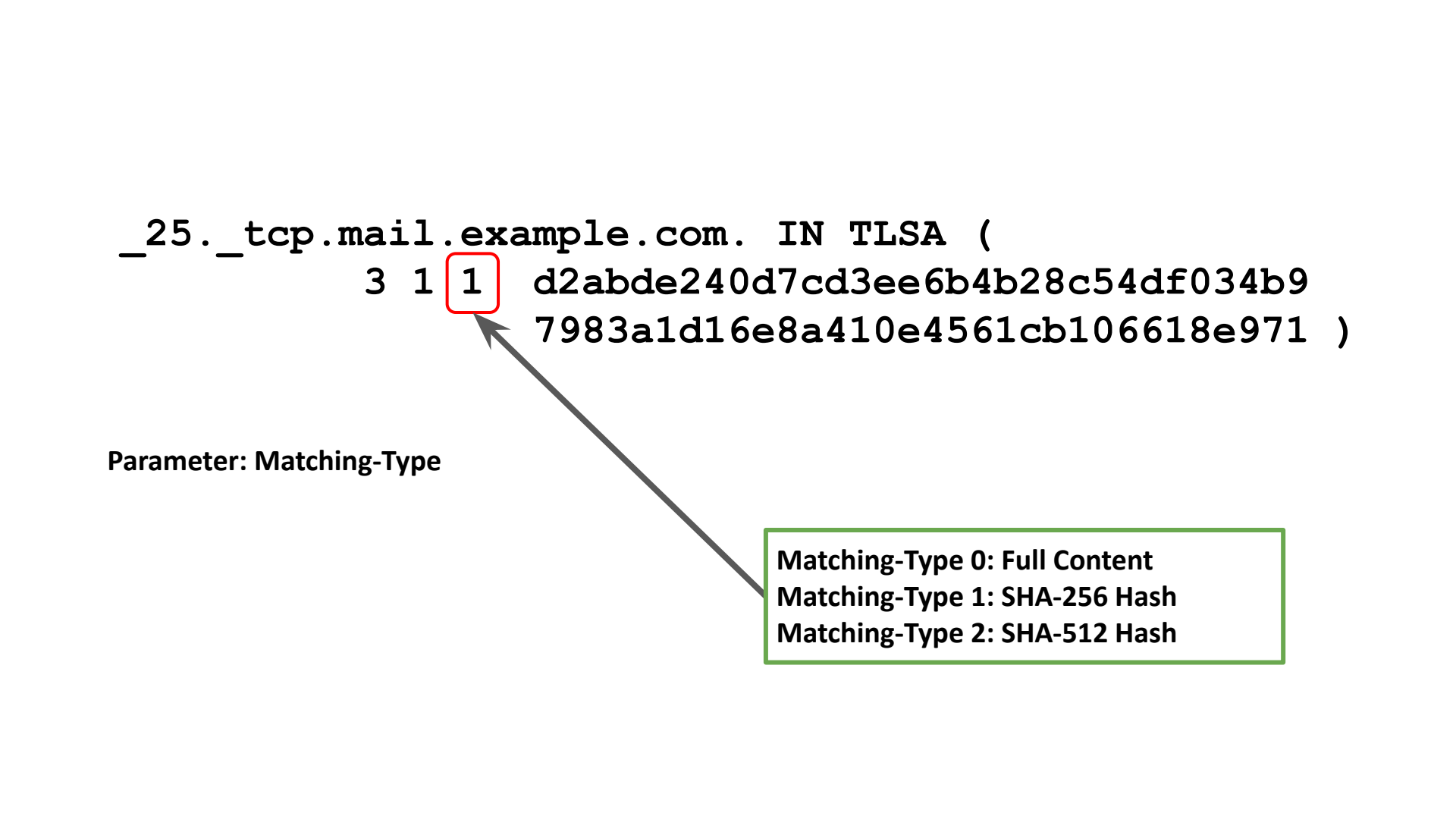
```
_25._tcp.mail.example.com. IN TLSA (
                3 1 1  d2abde240d7cd3ee6b4b28c54df034b9
                       7983a1d16e8a410e4561cb106618e971 )
```

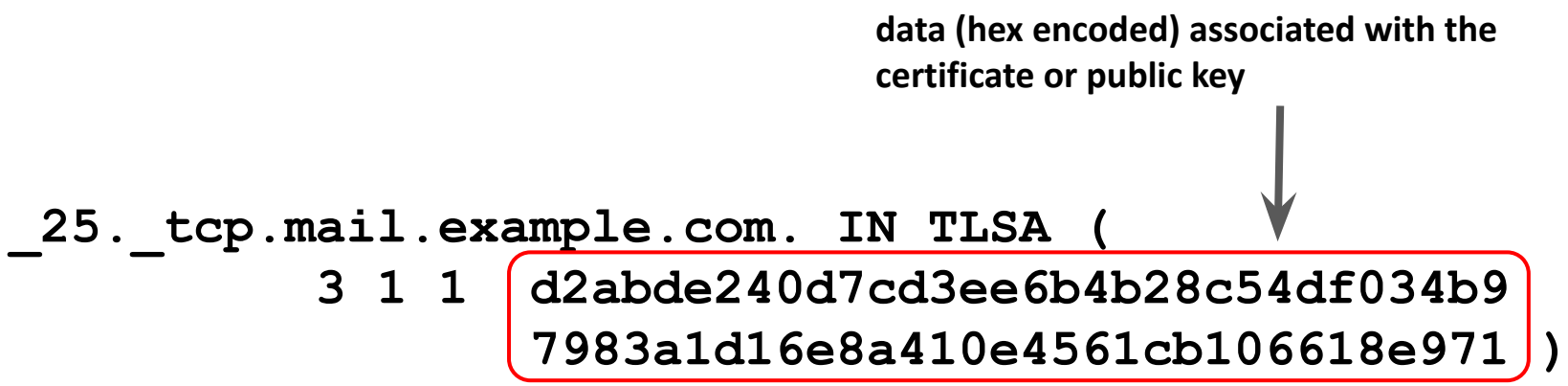


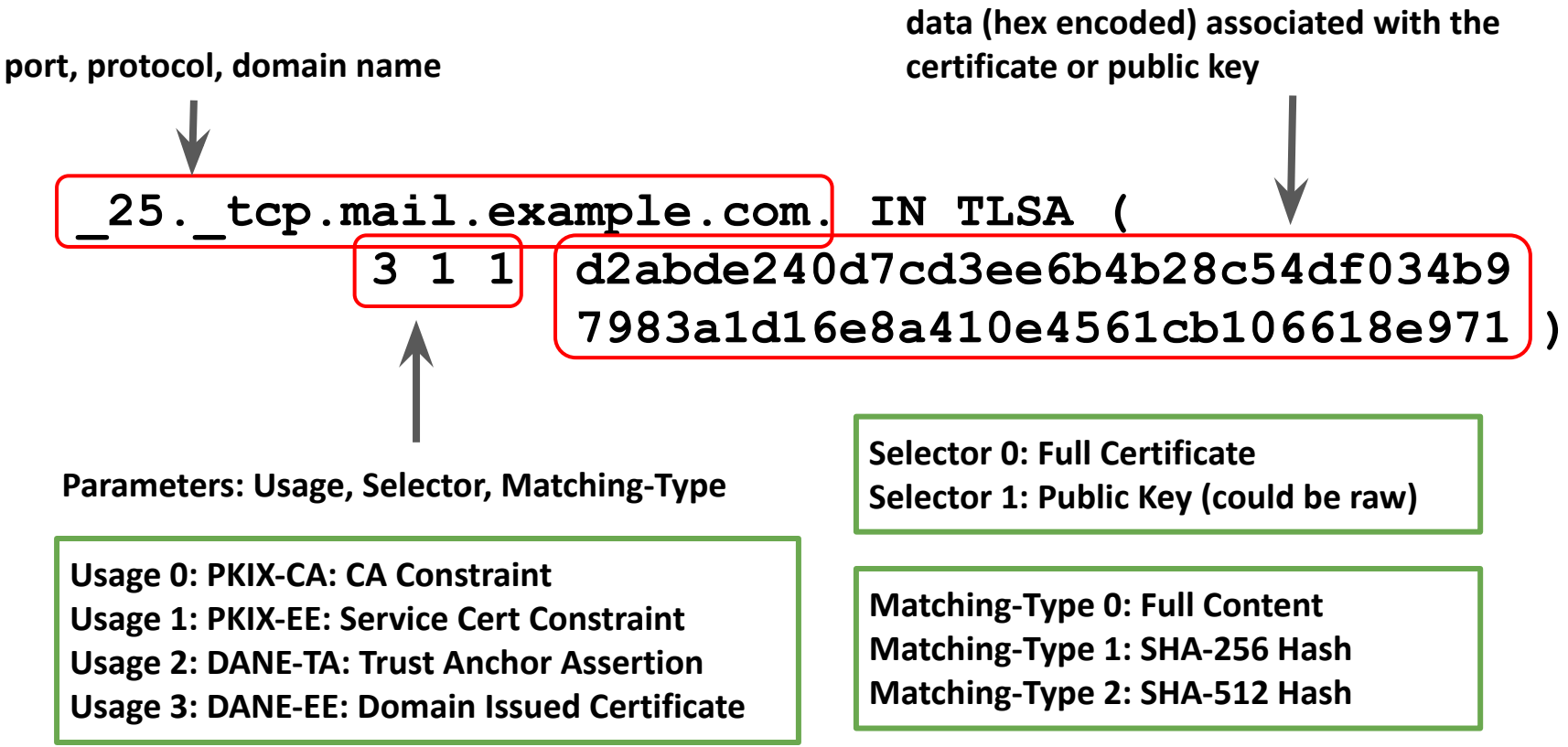
Parameter: Selector

```
_25._tcp.mail.example.com. IN TLSA (
            3 1 1  d2abde240d7cd3ee6b4b28c54df034b9
                   7983a1d16e8a410e4561cb106618e971 )
```

Parameter: Matching-Type




data (hex encoded) associated with the certificate or public key
_25._tcp.mail.example.com. IN TLSA (
3 1 1 d2abde240d7cd3ee6b4b28c54df034b9
7983a1d16e8a410e4561cb106618e971)

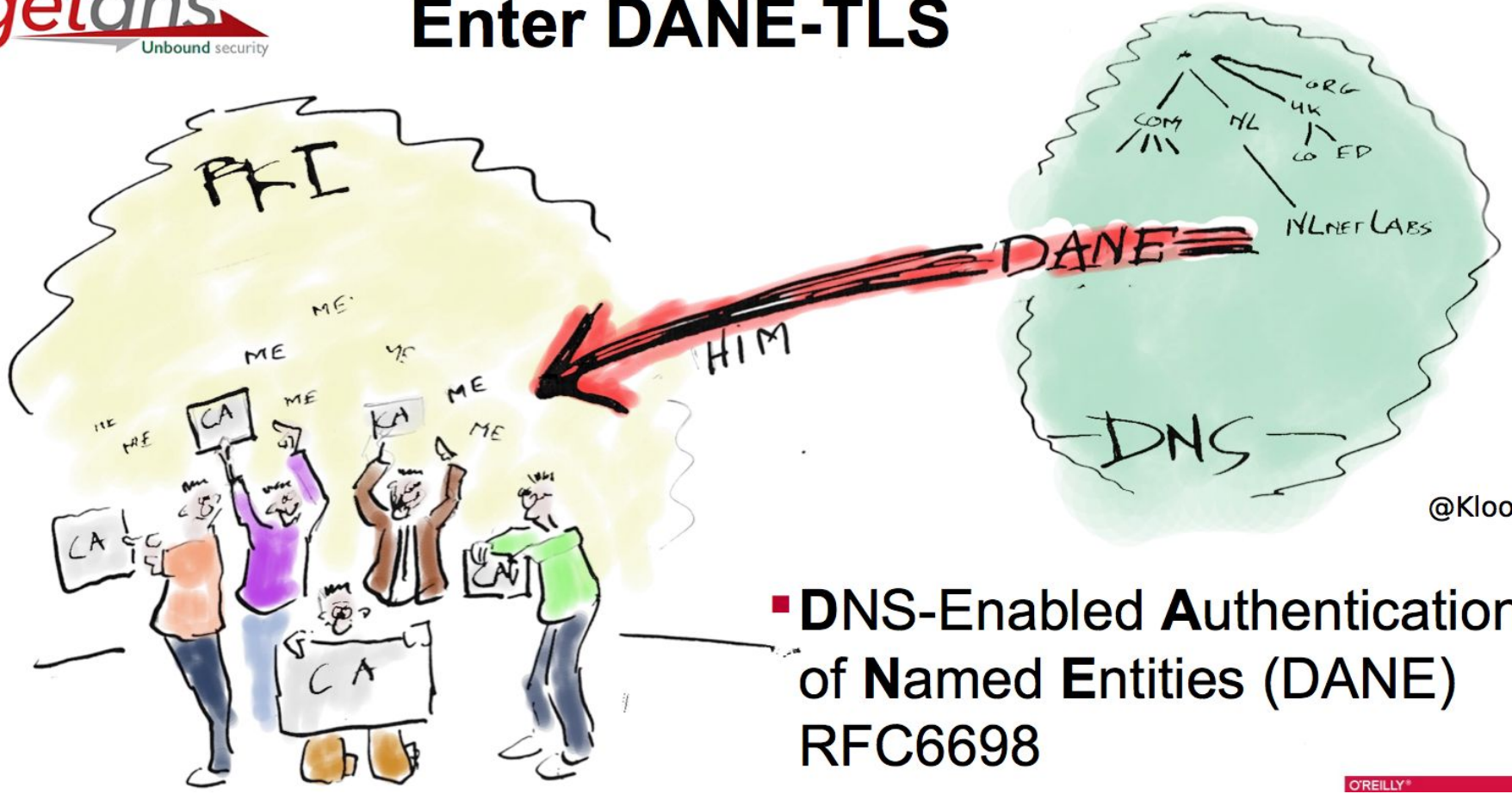


DANE record in this example specifies the SHA256 hash of the public key of the certificate that should match the End-Entity certificate. Authenticated entirely in the DNS.

Note: cartoon applies to PKIX modes only!

getdns Unbound security

Enter DANE-TLS



- **D**NS-Enabled **A**uthentication of **N**amed **E**ntities (DANE) RFC6698

DANE TLSA - what applications?

Potentially many could use them.

In practice today it’s mainly limited to SMTP (and to a smaller extent XMPP)

Some browser blockchain folks (Viktor?)

DANE for SMTP Transport Security

RFC 7672: SMTP Security via opportunistic DANE TLS

DANE to authenticate (server side of) connections between SMTP servers (specifically MTAs or Message Transfer Agents).

Without DANE, most connections between SMTP servers use encryption opportunistically. Even when encryption is used, it is vulnerable to attack:

- Attackers can strip away TLS capability advertisement and downgrade connection to plain text
- TLS connections are often unauthenticated

DANE for SMTP Transport Security

DANE addresses this security gap:

- Authenticate SMTP server's certificate using a DNSSEC signed TLSA record
- Use the presence of the TLSA record as an indicator that TLS must be used, preventing downgrade attacks.

Software support:

Postfix, Exim, Halon MTA, Power MTA, Cisco ESA, …

DANE for SMTP Transport Security

```
example.com. 86400   IN  MX   10 mail.example.com.

_25._tcp.mail.example.com. 7200 IN  TLSA   3 1 1 (
                          d2abde240d7cd3ee6b4b28c54df034b9
                          7983a1d16e8a410e4561cb106618e971 )
```

Both must be signed (plus the address records of the MX targets)

Note: only the DANE-* usage modes recommended for SMTP.

Email Provider support: Microsoft

Microsoft announcement (April 6th 2022): Support of DANE and DNSSEC in Microsoft Office 365 Exchange Online (live as of June)

https://techcommunity.microsoft.com/t5/exchange-team-blog/support-of-dane-and-dnssec-in-office-365-exchange-online/ba-p/1275494

DANE TLS for services that use SRV records

Signed SRV record to securely discover the connection endpoints of the service.

Signed TLSA records are obtained at the endpoint names.

When they are found, TLS is always used (downgrade protection)

```
_xmpp-server._tcp.example.com. 3600 IN  SRV  10 20 5269
jabber.example.com.

_5269._tcp.jabber.example.com. 600  IN  TLSA   3 1 1 (
                               A0315F0CF61CAC787140833C2C608550476
                               246DDA54122D66BB339D5 0FBB10E3 )
```

DANE TLS for the Web?

Killer app? (And the CA security incidents mentioned earlier are almost all related to the Web PKI).

Challenging proposition: introducing a competitor to the established Web PKI.

Early attempts by Google (pre-DANE) to authenticate X.509 certificates with DNSSEC (cert data in the DNS via TXT, DNSSEC auth chains in the certs).

2nd attempt: **TLS DNSSEC Chain extension (RFC 9102)**, newer, more dynamic effort to deliver DNSSEC authentication chain for the server's TLSA record in the TLS handshake.

Why send dnssec authentication chain in TLS?

Web browser folks have very specific needs where the normal way of just querying TLSA records in the DNS won't work.

- Browsers need to deal with middleboxes that impede their ability to lookup DANE and DNSSEC records.
- Latency reduction: DNSSEC involves more queries; having TLS server deliver the complete DNSSEC chain (pre-built & cached) in one shot addresses this.
- dnssec chain in TLS obviates the need for the endsystem to run a validating stub resolver (not common) or have a channel protected secure connection to an external validating resolver (also not common)

DNSSEC chain in TLS status

Ultimately this effort failed due to technical disagreements in the IETF.

The spec has been published (RFC 9102) as an experimental RFC on the independent stream (not standards track).

For the time being, in the immediate future, the Web will not be using DANE.

But other applications are looking at potentially using it.

DANE TLS for Encrypted DNS

DOT, DOQ, DOH?

For Client to Resolver, DNS over HTTPS (DOH) with Internet-PKI issued server certificates will likely become dominant.

For Recursive to Authoritative DNS, DANE TLS could have a role.

OPENPGPKEY

RFC 7929: DANE Bindings for OpenPGP Keys

Used to securely publish OpenPGP Public Keys in the DNS

DNS record name (owner name) is an encoding of the email address.

- 1st label:
  - Take local-part (LHS) of email (UTF-8 or ascii; non-ascii characters need to be normalized according to unicode rules)
  - Generate SHA256 hash of this, truncated to 28 octets, represented in hex (= 56 chars)
- 2nd label: fixed string “_openpgpkey”
- Remaining labels: domain name portion of email

Example OPENPGPKEY record

For **shuque@huque.com**

1st label: 28-octet truncated sha256 hash of "shuque" = adcd5698c7fc6c44e65e893ab7e84a638db4910d04e8e53314e8a101

2nd label: "_openpgpkey"

Remaining: domain name portion: huque.com

adcd5698c7fc6c44e65e893ab7e84a638db4910d04e8e53314e8a101.
_openpgpkey.huque.com. IN OPENPGPKEY <base64 encoding of the openpgp key>

These records can be quite large. TCP should be used to fetch them.

```
$ dig adcd5698c7fc6c44e65e893ab7e84a638db4910d04e8e53314e8a101._openpgpkey.huque.com.
OPENPGPKEY
;; Truncated, retrying in TCP mode.

adcd5698c7fc6c44e65e893ab7e84a638db4910d04e8e53314e8a101._openpgpkey.huque.com.
OPENPGPKEY

;; ANSWER SECTION:
adcd5698c7fc6c44e65e893ab7e84a638db4910d04e8e53314e8a101._openpgpkey.huque.com.3600
IN  OPENPGPKEY mQENBFNPDOkBCADAZikSR4TvRxMtU0WhbWFkZvXWOYdhWSPigqbsy7T5
PTNaALwPJaMGX5JLg/+T7kJK6WFjFfvuIc60PD5Rn71df/SqvyRdx2fW jWyjzvNfpY9IdeouIUKhWTyL+
[ … rest or rdata omitted … ]

;; WHEN: Sat Jul 30 06:13:15 EDT 2022
;; MSG SIZE  rcvd: 2337
```

SMIMEA

RFC 8162: Using DNSSEC to associate certificates with domain names for S/MIME.

S/MIME is a method of encrypting and signing MIME data used in e-mail messages.

The SMIMEA DNS record is used to associate S/MIME certificates with DNS domain names.

SMIMEA example

```
adcd5698c7fc6c44e65e893ab7e84a638db4910d04e8e53314e8
a101._smimecert.example.com. IN SMIMEA (
            3 1 1  b3dade240d7cd3ee6b4b28c54df034b9
                   7983a1d16e8a410e4561cb106618e493 )
```

Owner name format similar to OPENPGPKEY, except "_smimecert" is used.

Rdata format is similar to TLSA

SMIMEA project

Eric Osterweil (George Mason University)’s DANE SMIMEA project:

DANEportal.net - “an open portal to realize S/MIME+DANE. Our S/MIME and DANE tools are called DANEportal.net and Kurer - Democratized end-to-end message security and privacy. We've written an Outlook add-on and a Thunderbird plugin, and our portal is fully open and operational.”

Recent presentation at IEPG meeting just before IETF114.

DNS server support for DANE

All the major open source DNS server implementations support the DANE records natively (TLSA, OPENPGPKEY, SMIMEA, etc)

Older versions can typically support them using generic type and rdata encoding (RFC 3597: Handling of Unknown DNS Resource Record types)

Managed DNS provider support is more spotty.

(Some) DANE Tools & Resources

General TLSA:

- DANE Check: https://www.huque.com/bin/danecheck

SMTP TLSA specific:

- DANE SMTP Validator: https://dane.sys4.de/
- DANE SMTP Check: https://www.huque.com/bin/danecheck-smtp

TLSA Record Generator: https://www.huque.com/bin/gen_tlsa

OPENPGPKEY Generator: https://www.huque.com/bin/openpgpkey

https://github.com/baknu/DANE-for-SMTP/wiki/2.-Implementation-resources

DANE TLS Libraries

OpenSSL

GnuTLS (has a DANE specific library)

ldns (from NLNetLabs)

getdns

(Some other smaller libraries in Go and other languages)

Securing last hop?

Securing the path between end system and its recursive DNS server.

Options:

- Don’t bother. Run a full validating resolver on the client instead.
- Channel protection to recursive DNS server
  - e.g. TSIG, SIG0, GSS-TSIG, IPSEC ..
  - Very seldom done
- Run validating stub resolver on client
  - Main challenge: Middleboxes impede a non-trivial fraction of DNS queries for signed responses & newer types.

Delivering DNSSEC to the stub

Experimental Results on DNSSEC Record Delivery (IETF 114; dnsop; July 2022)

https://datatracker.ietf.org/meeting/114/materials/slides-114-dnsop-measuring-dnssec-success-01

Query	Failure Rate
A	0.022 (0.021–0.023)
A (CD=1)	0.024 (0.023–0.024)
A (DO=1)	0.387 (0.385–0.389)
A (DO=1, CD=1)	0.388 (0.386–0.390)
DNSKEY	0.023 (0.022–0.023)
SMIMEA	0.140 (0.138–0.141)
HTTPS	0.065 (0.064–0.066)
NEWONE	0.203 (0.201–0.204)
NEWTWO	0.214 (0.212–0.216)
NEWTHREE	0.281 (0.279–0.283)
NEWFOUR	0.289 (0.287–0.291)
A (WebExt API)	0.004 (0.004–0.005)

Overcoming the middlebox

TLS DNSSEC Chain (if used) can help this issue.

Queries over encrypted transports like DoH, which appear to traverse most middleboxes, also might.

DANE-like records that pre-dated DANE

RFC 4025: IPSECKEY: store IPsec keying material in the DNS

RFC 4255: SSHFP: DNS to secure publish SSH Fingerprints

Core Operational Guidance

Well managed DNSSEC infrastructure.

Ongoing monitoring of TLSA records vs certificate consistency.

Automated cert/key rollover and corresponding DANE record updates.

Newer work: DANE for TLS Client Authentication

Goal: Authentication client side of TLS connection with DANE

Target use cases (so far):

- SMTP (client) Transport Security
- IOT Device Authentication

Protocol Summary

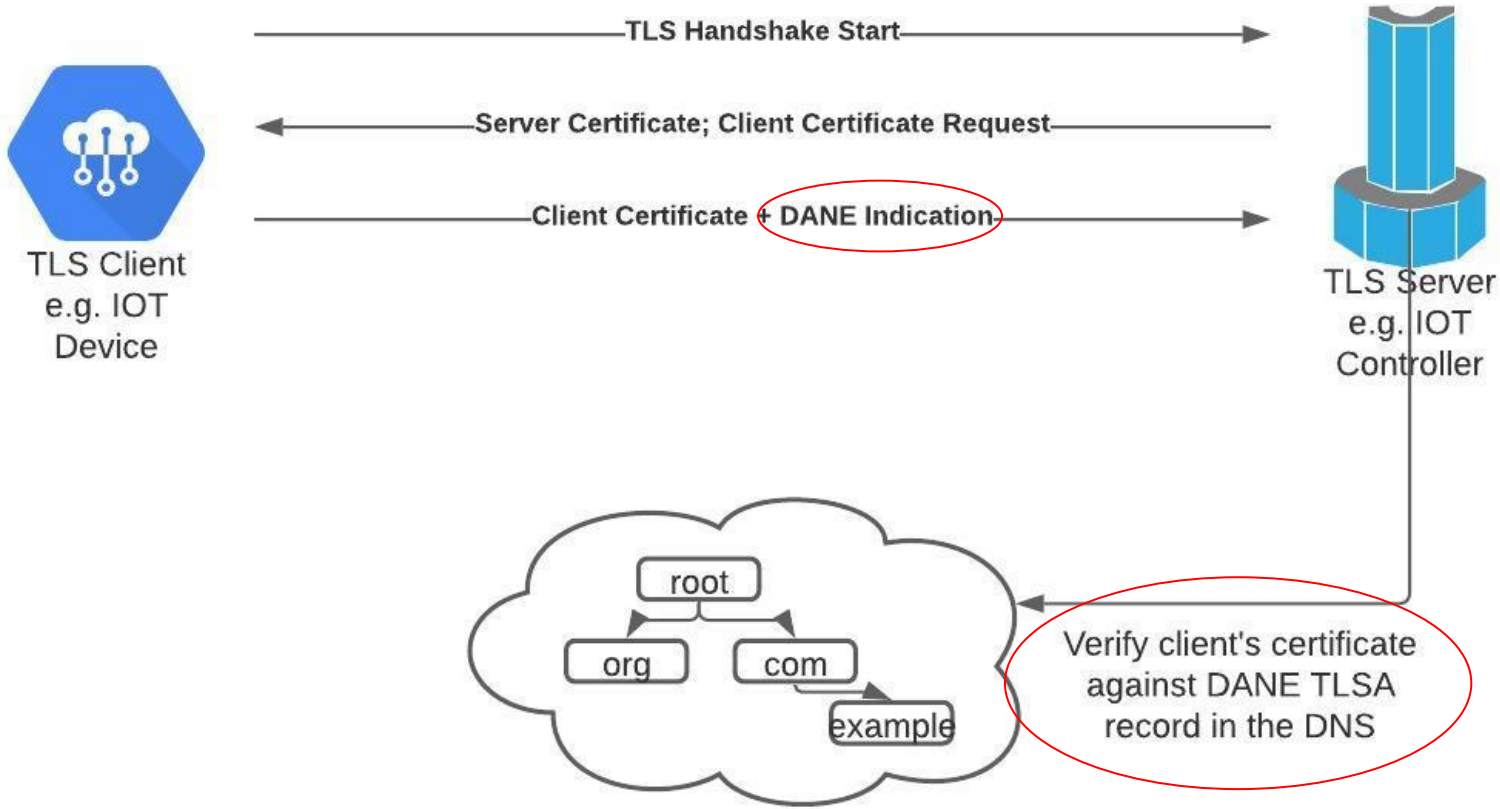
- TLS Client has a DNS domain name identity
  - A public/private key pair & a certificate binding the public key to the domain name
  - Corresponding DANE TLSA record published in DNS
- TLS server
  - Sends Certificate Request message in handshake; extracts client identity from presented certificate or DANE client ID extension, constructs TLSA query, validates DANE TLSA response with DNSSEC

Details - see work in IETF “DANCE” working group

https://www.ietf.org/archive/id/draft-ietf-dance-client-auth-00.html

https://www.ietf.org/archive/id/draft-ietf-dance-tls-clientid-00.html

```
sensor7._device.example.com. IN TLSA (
          3 1 2
          0f8b48ff5fd94117f21b6550aaee89c8
          d8adbc3f433c8e587a85a14e54667b25
          f4dcd8c4ae6162121ea9166984831b57
          b408534451fd1b9702f8de0532ecd03c )
```


TLS Handshake Start
Server Certificate; Client Certificate Request
Client Certificate + DANE Indication
TLS Client
e.g. IOT
Device
TLS Server
e.g. IOT
Controller
root
org
com
example
Verify client's certificate
against DANE TLSA
record in the DNS

DANE Survey

Viktor Dukhovni & Wes Hardaker's DANE survey

https://stats.dnssec-tools.org/about.html

Goals: promote DANE adoption

- Publish statistics that document the growth of DANE and DNSSEC adoption, promoting further adoption and use of best-practice parameters.
- Identify systems where DANE is inadvertently misconfigured, and notify their postmasters.
- DANE adoption can only grow if any published TLSA records are correct for the vast majority of domains which published them.

Data collected by the survey

- The DS RRset from the parent domain
- The DNSKEY RRset of the given domain
- The MX RRset of the given domain
- The A and AAAA records of each MX host
- Any SMTP TLSA records (_25._tcp.) of MX hosts whose A records are DNSSEC signed

Summary Statistics

Last Updated:	2022-07-30 04:50 -0700
Total number of DS Resource Record Sets:	19,106,910
Total number of working DNSKEYs:	18,906,807
Total DANE protected SMTP domains:	3,587,868

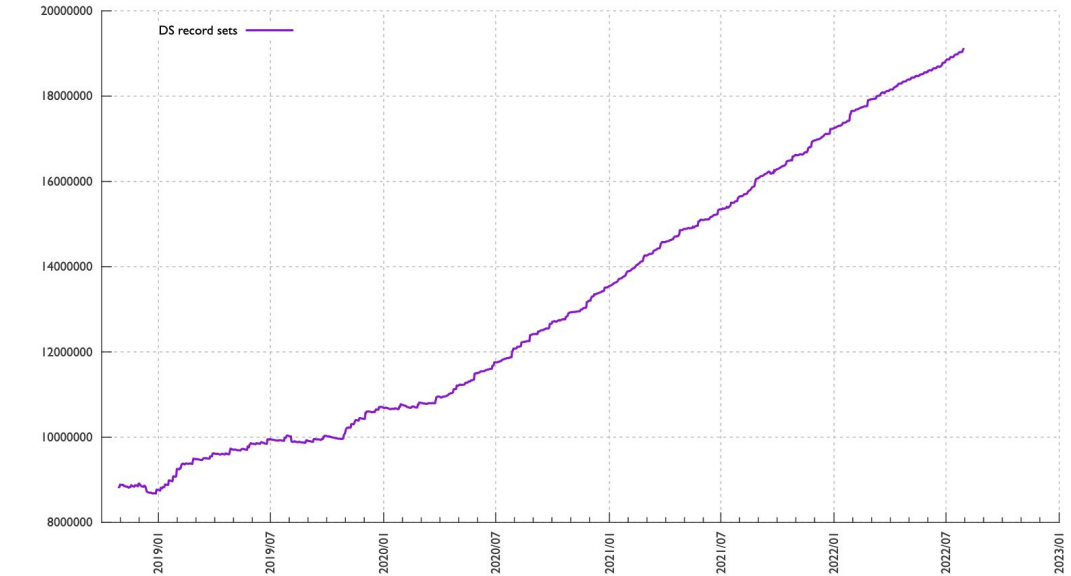
Explore the per-

For administrators th
please see our explo
in the last 24 hours.

DANE Trend graphs

DNSSEC DEPLOYMENT GROWTH | SIGNED MX AND DANE RECORDS | ZONES HOSTING DANE MAIL SERVERS

The following graph shows the growth of observed DS record sets over time (i.e. the number of signed zones):



https://stats.dnssec-tools.org/

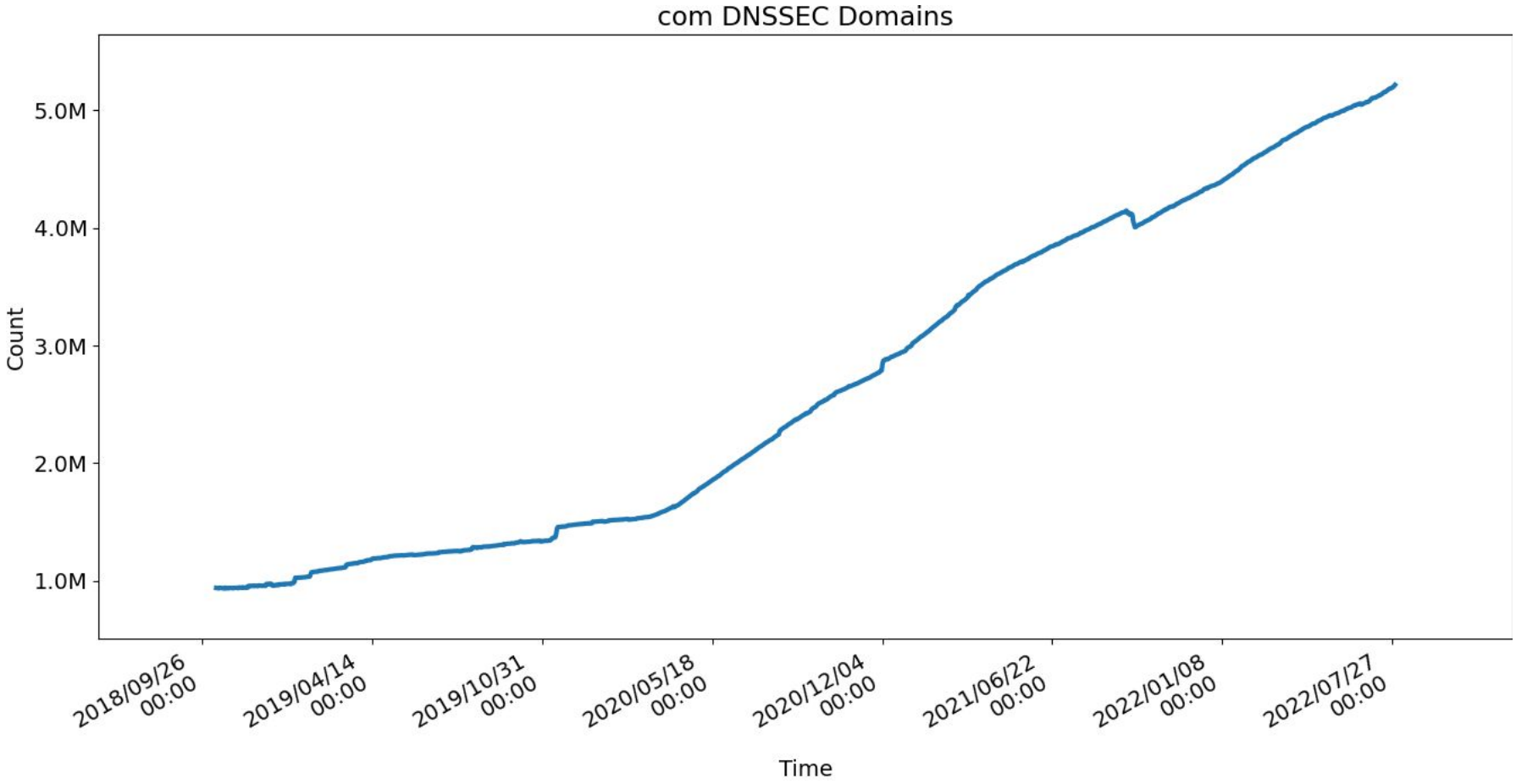
KSK ALGORITHM | ZSK ALGORITHM | RSA KSK SIZES | RSA ZSK SIZES | RSA EXPONENTS

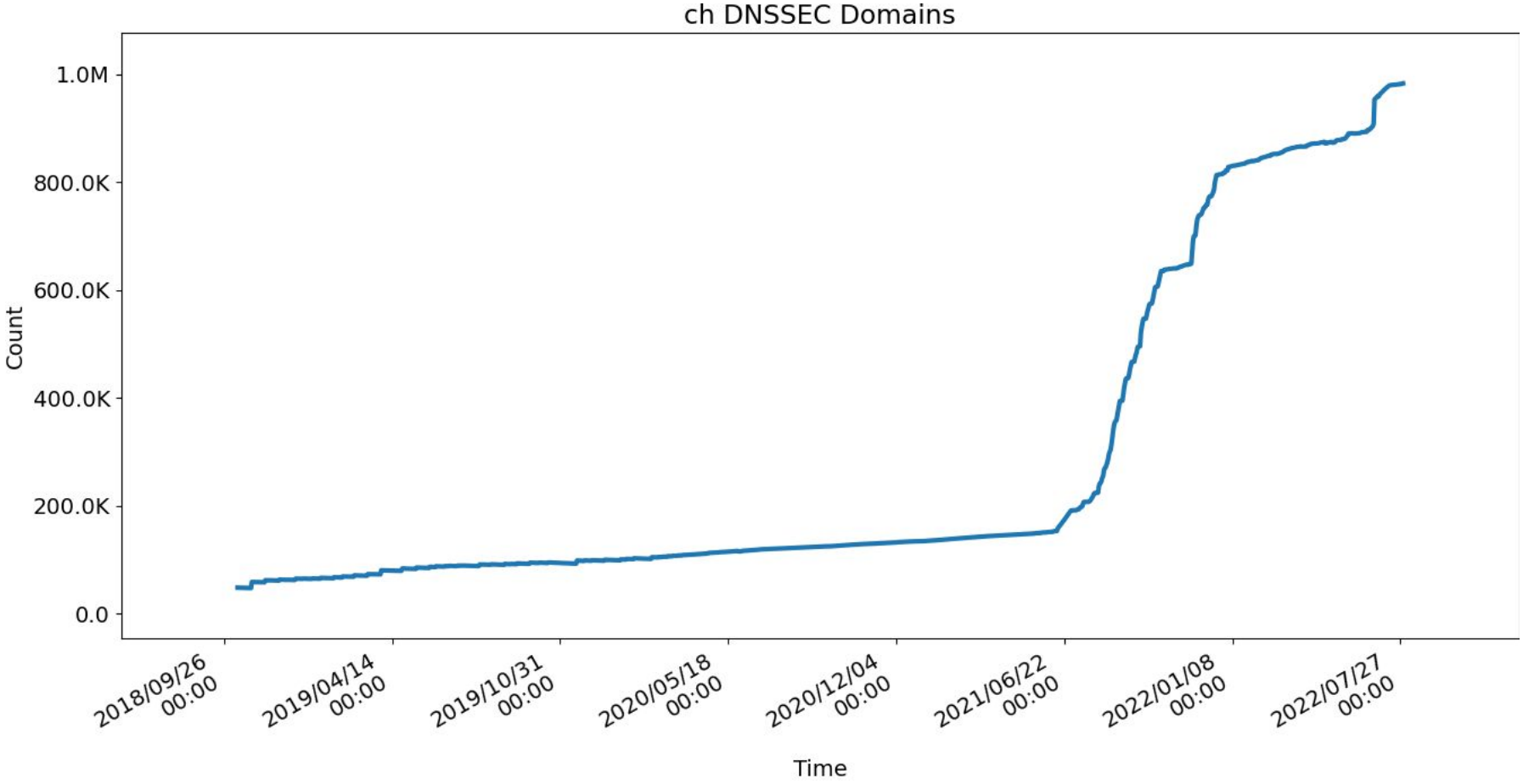
DNSKEY parameter frequency (1000 or more instances), by zone count:

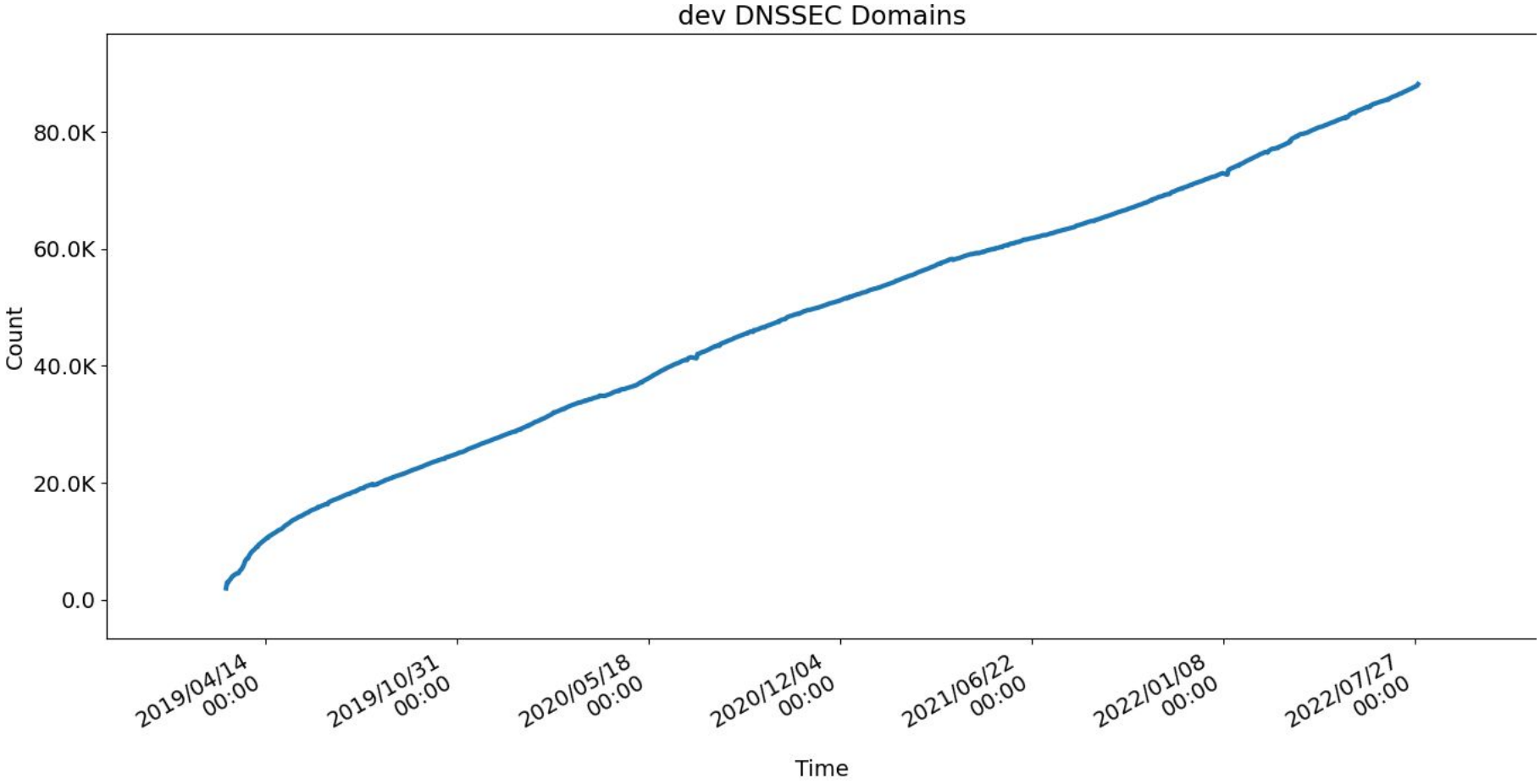
KSK Algorithm	Flags	Protocol	Domain Count ↓
8	257	3	9564070
13	257	3	8772725
10	257	3	251274
14	257	3	157244
7	257	3	136535
5	257	3	21829

DNSSEC by Top Level Domain (TLD)

TLD	Number working	DS records ↓	Percent working
com	5153798	5215788	98.81
nl	3592478	3618233	99.29
ch	980431	983186	99.72
br	836010	836463	99.95
se	786536	802970	97.95
cz	768320	775970	99.01
fr	699451	701926	99.65
eu	556886	563913	98.75
net	487300	493250	98.79


com DNSSEC Domains
5.0M
4.0M
3.0M
2.0M
1.0M
Count
2018/09/26 00:00
2019/04/14 00:00
2019/10/31 00:00
2020/05/18 00:00
2020/12/04 00:00
2021/06/22 00:00
2022/01/08 00:00
2022/07/27 00:00
Time


ch DNSSEC Domains
1.0M
800.0K
600.0K
400.0K
200.0K
0.0
Count
2018/09/26 00:00
2019/04/14 00:00
2019/10/31 00:00
2020/05/18 00:00
2020/12/04 00:00
2021/06/22 00:00
2022/01/08 00:00
2022/07/27 00:00
Time


dev DNSSEC Domains
80.0K
60.0K
40.0K
20.0K
0.0
Count
2019/04/14 00:00
2019/10/31 00:00
2020/05/18 00:00
2020/12/04 00:00
2021/06/22 00:00
2022/01/08 00:00
2022/07/27 00:00
Time

Issues found with the *ietf.org* domain:

! **Deprecated DNSSEC algorithms or weak RSA keys found. See [RFC8624].**

GO TO DNSKEY DATA

Where	Key Tag	Reason	Detail
DS	45586	Deprecated DS digest type	SHA-1 (1)
DS	45586	Deprecated DNSKEY algorithm	RSASHA1 (5)
DNSKEY	40452	Deprecated DNSKEY algorithm	RSASHA1 (5)
DNSKEY	45586	Deprecated DNSKEY algorithm	RSASHA1 (5)

DNS records for *ietf.org*

MX RECORDS | TLSA RECORDS | SMTP/TLS CERTIFICATES | DS RECORDS | DNSKEY RECORDS

- **mail.ietf.org**

Usage	Selector	Matching Type	Data
DANE-EE (3)	SPKI (1)	SHA2-256 (1)	0c72ac70b745ac19998811b131d662c9ac69dbdbe7cb23e5b514b56664c5d3d6

You can also check *ietf.org* on DNSVIZ or dane.sys4.de

– for unsigned provider zones.
+ for signed provider zones, but no MX hosts have TLSA records.
d for signed provider zones, domains served by MX hosts without DANE TLSA records
D for signed provider zones, domains served by MX hosts with DANE TLSA records.

Where the zone is signed, but DANE is not yet broadly deployed, publishing TLSA records would l

Note that in a few cases the provider is only a backup MX for the hosted domains, and the primary

signed domain count	MX host zone	DNSSEC/DANE status
2,389,935	google.com	–
1,465,592	ovh.net	–
1,250,253	one.com	D
592,311	outlook.com	–
281,131	hostpoint.ch	D
194,192	googlemail.com	–
190,548	infomaniak.ch	D
185,642	mijndomein.nl	D
168,029	argewebhosting.nl	D
157,437	transip.email	D
140,658	aftermarket.pl	+
115,291	hostnet.nl	D
111,955	mailprotect.be	–
107,856	domeneshop.no	D
98,249	loopia.se	D
90,378	wedos.net	–

FIN