Internet Survey of DANE/TLSA DNS Records Application and Use for Mail Exchanger

Erwin Hoffmann, Egbert Falkenberg
{ehoffmann,falken}@fb2.fra-uas.de

Zusammenfassung

We have performed an Internet survey for about 66 mio domains in Q4/2021 to extract the DANE/TLSA records in the DNS which are foreseen to indicate TLS capabilities and to enable X.509 certificate fingerprinting for MX services. Their particular use and the application scenarios for TLSA records are analyzed on a wide scale base indicating the acceptance of policy information in the DNS for Mail Exchangers (MX).

1 Scope and Use Case for DANE/TLSA Records in the DNS

Communication based on (E)SMTP message exchange shall be confidential. (E)SMTP [29] lacks in general privacy since it is solely a transport mechanism. Confidentiality and privacy can be enforced on the message content using either S/MIME [35] or PGP [7] – which is outside the scope of (E)SMTP. However, (E)SMTP encourages to use peer-to-peer encryption since (E)SMTP is (as host-to-host protocol) enabled to means of *Transport Layer Security* (TLS) [32]. The German 'Federal Office for Information Security' has poured this view into a technical recommendation for email servers [26].

Yet, there is no need to employ TLS in the context of the *Public Key Infrastructure* (PKI) [11] while using 'official' X.509 certificates indicating the authentication of the receiving MTA (Mail Transfer Agent) and to digitally sign the handshake. Current SMTP MTAs like *sendmail* [2], *exim* [20], *postfix* [37], or the author's fork of *qmail – s/qmail*

[22] – don't require a particular certificate chain to be verified by the (E)SMTP client; thus accepting 'self signed' X.509 certificates as well – or even allowing an 'anonymous' handshake for the TLS session. Here, a particular *Trust Store* is not required on the client side. Unlike the current situation for today's Web browsers: Here the 'verification' of a received certificate is usually mandatory and requires additional affords by the user if this fails.

Instead of the PKI, the *Domain Name System* DNS can be used as a trust anchor, due to the fact that (E)SMTP is closely related to DNS and wouldn't work without the knowledge of Mail Exchangers being present here as `MX` resource records. Adopting the idea providing *SSH fingerprints* [36] in the DNS *Domain Authenticated Name Entities* (DANE) [27, 28] can be used to host TLSA records maintaining a fingerprint of X.509 certificate for a particular service; here (E)SMTP being available over TCP port 25. However, DANE is not restricted to (E)SMTP and can be used for practically any protocol typically based on TCP or UDP.

An email client may now be enabled to look up the X.509 fingerprint in the DNS prior to receiving and accepting the certificate from the remote MTA. In case of a match, this provides strong evidence that the chosen MTA is entitled for TLS encrypted connections and is in addition authorized by the domain owner to do so. Of course, UDP based DNS messages maybe a subject of forging or suppression. Thus, additional efforts like DNSSec [3] (or DNSCurve [13]) are usually required to compensate for this weakness.

DANE according to [27, 28] and subsequently explained in [17, 16] (giving operational guidelines) seems to be closely bound to DNSSec [3]. However, DNSSec does not provide 'secure' answers as anticipated in [16], but those are given merely authenticated and indisputable by its source. However, DNS enhancements like DNSCurve, DNS/oTLS, DNS/oHTTPS, or most recently DoQUIC [6] could be used to retrieve TLSA records in a 'secure' manner.

1.1 Reseach Questions

We are interested in the deployment and technical usage of DNS TLSA resource records for Mail Exchanger (MX). Setting up TLSA records in the DNS requires coordination between the provided information in the DNS and its actual use for TLS-encrypted SMTP connections.

Here, we use an unbiased approach not requiring DNSSec authenticated DNS replies, while following a typical DNS lookup a SMTP client would perform for retrieving the required information. We probed about 66 mio domain names covering a significant part of the Internet. Internet studies regularly performed by *Viktor Dukhovni* [14] have a different scope while evaluating the potential existence of DANE records, aka DNSSec secured TLSA records.

1.2 Outline of the paper

Our paper is organized in the following sections:

- We discuss initially the Use Cases for applying DANE/TLSA records in the DNS an their implications configuring a SMTP Mail Exchanger to make use of it.
- We discuss our measurement method and our analysis chain using a DNS library based on Dan Bernstein's djbdns [5]. DNS TLSA queries are performed for 16 top-level domains, including DE and NET.
- The results allow to detail the specific anticipation of TLSA records for those top-level domains but also show the current topology of SMTP mail usage here.
- Finally, based on those observations some recommendations are given concerning the setup of Mail Exchangers in conjunction with DANE/TLSA records.

2 Use Cases of DANE/TLSA

The scope of using DNS records to store and retrieve additional TLS authentication information is laid out in [4] and was triggered by the loss of trust regarding the PKI X.509 environment, as this seemed to become compromised. Reversely, DNSSec became operational with its own trust chain. Following the idea of [36], an attempt was made to provide TLS authentication (TLSA) information in the DNS: *DNS-Based Authentication of Named Entities* (DANE) [4, 21, 17].

The use case of DANE involves two parties:

1. The *DNS domain owner* needs to evaluate its *Mail Exchanger* (MX) X.509 certificates and stores typically their fingerprint in its own zone file. For this purpose, a new DNS resource records (TLSA) is defined in [21]. Here, the DNS authoritative content server should be capable to support the particular RR format for easy use. For the recipient MTA, no changes or enhancements are required.

2. The *mail client* (typically as part of the *Mail User Agent* MUA or as client module of Internet MTAs) is enabled to perform a DNS TLSA lookup prior to starting the TLS handshake with the recipient MX. It can now evaluate the DNS response while comparing the TLSA information with the received X.509 certificate (chain) provided by the TLS-capable MTA. In most cases, The TLS connection is based on (opportunistic) *StartTLS*.

It should be noted – and comparable with DKIM – that the TLSA information is volatile. However, different from DKIM, the TLSA evaluation is only due for the (E)SMTP (RFC821) session, while *DKIM* is used to sign the (persistent) SMTP message body (RFC822). In turn, changing X.509 certificates for the receiving MTA is less critical here, though in general the problem of 'eventual consistency' in DNS is present in general.

2.1 Provisioning of TLSA information in the DNS

Upon deploying TLSA information in the DNS several constraints should be considered following RFC 6698 [21] which defines the DNS record type *52* (TLSA) with the following attributes:

- *FQDN*: What is the domain name (FQDN) to be associated with the TLSA data?

- *Usage*: What kind of X.509 certificate verification method shall the client use?

- *Selector*: What part of the X.509 certificate is referenced in the fingerprint?

- *Matching Type*: How shall the evaluation be done while matching the information present in the TLSA record?

We will discuss those items in the following sections. Outside the scope of RFC 6698 [21] is however the question of the associated *TTL* of the TLSA record and thus, how long this information should be kept as valid by a DNS cache server (*validity period*).

While RFC 6698 [21] introduces DANE/TLSA and explains the DNS wire format of the RDATA section, section 3 gives a breakdown of the TLSA prefix to be synthesized by means of prepended labels indicating with a leading 'underscore' (`0x5f` ASCII):

- `_[port].`
- `_[service].`

Here, `[port]` is a standard TCP/UDP/SCTP port in decimal format, while `[service]` is out of the set `{tcp, udp, stcp}`. Whether a hostname label or the domain label as *base* follows next is not clearly expressed, though the examples in [21, 17] suggest a hostname label. Interestingly, [38] used both assumptions in their survey.

In our case for (E)SMTP a typical TLSA FQDN would be:

- `_25._tcp.mx.example.com`

However the format of the TLSA FQDN as introduced in [21] may not only point natively to a TLSA records, but also to a *CNAME* record, requiring a redirection. [16] explicitly mentions such setups. In this case, a TLSA lookup for this domain name would provide a '`NODATA`' answer from the DNS server to be additionally processed by the recursive client.

[Fig. 1] shows a sketch including some valid samples of TLSA records in the zone of '*example.com*'.

2.1.1 Usage

The *Usage* is the most critical part of the information given in the RDATA section of the TLSA record since it defines the *scope* and *policy* of the received information during the TLS handshake while exchanging the X.509 certificates. Further, the *Usage* provides operational constraints on both sides:

- *Server-side*: Which X.509 certificate to present during the TLS handshake.
- *Client-side*: How to evaluate the received X.509 certs except for the following information given in the TLSA RDATA section.

Given the *Usage*, the mail client receives a hint how to process the certificate provided by the mail exchanger. Reversely, the domain owner has the freedom to deploy PKIX derived X.509 certificates or locally administered certificates. Unlike the case for Web services, (E)SMTP clients are not demanding certificates issued by a *Certificate Authority* (CA) but are usually satisfied with self-signed X.509 certificates as well.

Initially, [21] gives some guidance for the *Usage*:

(0) *PKIX-TA*: The X.509 certificate is considered to be a *PKIX Trust Anchor* and thus the MTA needs to provide the entire *certification chain* including it's own X.509 cert for validation and verification purposes.

(1) *PKIX-EE*: Given (intermediate) cross-signed X.509 certs in the wild, now with the *PKIX End Entity*, it is possible simply to provide the server's cert and let the other side follow the PKIX verification using its own *Trust Store* settings.

(2) *DANE-TA*: Dependency on the PKIX validation chain is not required here using a *DANE Trust Anchor*. In particular, independence from the PKIX is feasible by using self-signed certificates from the private root certificate to the one issued to the server itself.

(3) *DANE-EE*: With *DANE End Entity* the requirement for the certificate chain can be dropped and only the server cert needs to be provided during the TLS handshake.

Disregarding the fingerprint evaluation, three different methods of X.509 MTA verification are required by the mail client:

1. The X.509 certificate chain supplied by the server (the remote MTA) needs to be followed strictly: *PKIX-TA*, *DANE-TA*.
2. The certificate chain is to be evaluated using local information according to the client's Trust Store: *PKIX-EE*.

3. Only the server's X.509 certificate is relevant for verification: *DANE-EE*.

Given the last case (*DANE-EE*), this could be considered as 'dynamic' certificate pinning involving a DNS lookup and it releases the client from following the typical PKIX verification scheme.

In any case, the adjacent information in the TLSA RDATA section, *Selector*, *Matching Type*, and finally the *fingerprint* of the entire X.509 cert always refers to the server certificate only. This *Certificate Association Data Field* (RFC 6698 [21], section 2.2) has to be provisioned hexadecimal, with two values packed into one byte for the 'wire' transmission.

According to RFC 7672 [16] section 3.1.1 explicit requirements for the mail client to match the hostname ('reference identity') against the *Canonical Name* (`CN`) as part of the *Distinguished Name* (`DN`) or the elements present in the *Subject Alternative Name* (`SAN`) are ruled out and thus should not be subject of additional DNS validation. This allows a single X.509 certificate to service not only multiple recipient domains (on one MX) but in addition – as we will see later – to be used by several distinct MX servers.

2.1.2 Selector

Within the TLSA/DANE framework, the X.509 certificate is not relevant by itself; this is a matter of the certificate verification chain, which actually can be omitted given *DANE-EE* usage. Rather, the fingerprint of the certificate is matter of a comparison. Here, two choices exist:

(`0`) The X.509 certificate's fingerprint is calculated considering the entire certificate. Now, the *Owner* and *Issuer*'s DNs, the public key of the Owner, the Issuer's signature, and all technical components (like EKU) of the cert are included. The *SLL routines [10] provide some means to calculate and also verify this fingerprint.

(`1`) The fingerprint may cover the public key of the *Owner* (the *Subject*) solely, which is called the *Subject Public Key Identifier* SPKI [11].

Unfortunately, the *SSL routines do not provide a coherent computation of those values and the SPKI needs additional efforts for both for generating as well as evaluating its value.

From a practical point of view, both methods possess significant differences for the application of the certificate:

- If case (`0`) is chosen as *Selector*, each time the X.509 certificate is renewed the value of the TLSA RDATA field (the fingerprint) needs to be updated, since a fresh certificate needs to be covered.

- Choosing the SPKI (`1`) as identifier, the public key of the server may stay constant, even across certificate renewals. Thus, the strength of the underlying algorithms and key length should be carefully considered.

2.1.3 Matching Type

The *Matching Type* defines the information given in the TLS record's RDATA section after the first three bytes (octets) including this last one:

(`0`) The X.509 certificate is present. Depending on the algorithms and key length this information may stretch up to 1 KByte and thus is not suitable to be transmitted using standard DNS UDP messages.

(`1`) Instead of the certificate itself, its SHA-256 hash (*digest*) can be used to be calculated (using *Selector* `0`) over the entire cert, or (with *Selector* 1) the SPKI only. We realize, that in this case the content of the RDATA field is always restricted to 35 bytes and well suited for UDP transmission. However, we need to consider DNSSec yielding a much larger DNS response message size.

(`2`) Given the *Matching Type* (`2`) a SHA-512 hashsum is used. Thus, the length of the RDATA field is just 67 bytes. A potential security improvement w.r.t. (`1`) is questionable; in particular if *SPKI* is used as base information.

2.1.4 TTL and Key Rollover

Though no explicit suggestions for the TTL of TLSA records are given, it is clear that expired TLSA information is not helpful. [17] discusses the potential TLSA TTL in the context of a key rollover.

- Several TLSA records (RR) can be present for the same MX server (and of course any other).
- TLSA records may point (and again considering 'eventual consistency') in the DNS to retired X.509 certificates. Thus, TLSA verification may fail in the case of *Selector* (`0`) and if the TTL is chosen too long.
- In consequence, X.509 certificate rollover and TTL needs to be well coordinated.
- Potentially, a TLSA-enabled client will check all provided TLSA records for query and compares all of those matching them with the X.509 cert during the TLS handshake.
- Thus, prior of changing the server's X.509 certificate, it might be advisable to deploy the new TLSA record (additionally) in the DNS covering the forthcoming cert information.

In contrast, if *Selector* (`1`) is chosen to be provisioned alongside with the SPKI of the X.509 certs, the respective TTL could be longer.

2.2 Mail Client requirements evaluating TLSA records

A TLSA-enabled mail client needs to perform the following steps before setting up the actual SMTP connection:

1. The DNS `MX` record is fetched and thus the FQDN of the MTA is received.
2. Using this information, the IP (v4/v6) address is looked up using a `A` and/or `AAAA` query.
3. In addition, now the TLSA record is queried. Here, a synthesized address is used based on the port, the layer 4 protocol (here: TCP)

and the given FQDN of the MTA.
Let's assume the MTA name is '`mx.example.com`' the TLSA query uses for a connection on port 25 the FQDN: `_25._tcp.mx.example.com`. In case of the ESMTP *Submission* protocol, a port number '587' would be used, and in case of *SMTPS* '465' while applying the same grammar.

4. This behavior can be enhanced for an *SPF* or *DKIM* query at the client side, but this is out-of-scope here.

Now, the ESMTP client is fed with the respective DNS information and capable – while possessing the TLSA record – to verify the received X.509 certificate's fingerprint within the TLS handshake against the one retrieved from the DNS.

Typically, the following situations will arise:

- No TLSA information is deployed in the DNS: The mail client will proceed as usual.
- One or more TLSA records are received: The client needs to match one of those given the 'matching type' and 'selector' with the provided X.509 certificate by the remote MTA.
- In case a match is given, the client can proceed as usual and perhaps employs additional checks on the certificate
  - by means its validity, e.g. validation period, FQDN and/or IP address present (matched against the connection information),
  - the certificate chain according to its trust store, or
  - potentially an OCSP lookup for the certificate [34].

[Fig. 1] is a sketch of a typical set up for a mail client and the information it gathers from the DNS prior to the connection with the MTA at `mx.example.com` via (E)SMTP.

3 Measurement Set Up and Analysis

Our Internet survey and DNS query is based on two components:

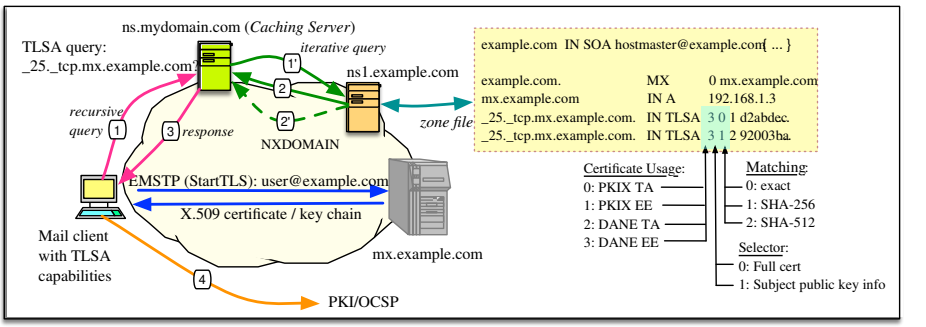

Abbildung 1: Email client with TLSA and PKIX capabilities initiating a TLS session with the MTA `mx.example.com`. ① The mail client asks its DNS recursor for the MX, IP, and TLSA record. ①' The recursor does an iterative query to the authoritative name server of the respective domain ('`example.com`'). The response is retrieved; either carrying ② the TLSA RR or a ②' `NXDOMAIN` and finally ③ forwarded to the mail client evaluating ④ the answer together with the received X.509 certificates given in TLS handshake of the MTA connected to ('`mx.example.com`').

1. *djbdnscurve6* (-40) [23] is used to provide by means of **dnscache** a local cache and iterative resolver to enhance queries.

2. *s/qmail* (4.1.12) [25] includes the command line tools **dnsmxip** and **dnstlsa** doing an recursive query (against **dnscache**) while evaluating the `MX/AAAA/A` information. From the given response **dnstlsa** automatically builds the synthesized TLSA domain name, querying the TLSA record, and returns a clear-text message from the response.

Those Unix programs are forks and new implementations of *Daniel Bernstein*'s *djbdns* [5] providing now DNS library functions for plain DNS and DNSCurve queries/responses while including full IPv6 capabilities. The clients are part of the *s/qmail* package and designed mainly to diagnose SMTP connection problems. In particular, **dnsmxip** is starting from the `MX` lookup at first with an adjacent `AAAA` and a final `A` lookup. We follow the idea of RFC 8200 [12] that IPv6 has now precedence for connecting to hosts on the Internet.

Further, **dnscache** is not DNSSec enabled. However any DNSSec records will be gratefully cached and provided to the querying recursive DNS client. By means of the DNSCurve protocol encrypted DNS messages where used were applicable.

All modules require base routines from the *fehQlibs* [24] – which come with a simple DNS stub resolver being used by **dnsmxip** and **dnstlsa**, while **dnscache** is our DNSCurve enabled resolver (aka *cache server* and *recursor*). Though EDNS0 is supported by **dnscache** the accepted DNS package size is chosen to be the (IPv6) `MTUSIZE-52` and thus 1228 bytes. For further discussion on this topic see [33].

Out-of-Scope
In our study, we left the following research items out-of-scope to be potentially covered by other evaluations:

- Since DNSSec was not used in addition, support for *EDNS0* was neither requested nor required.
- No attempt was made to set up a (E)SMTP connection with the MX and thus verifying the (potential) X.509 certificate and comparing those with the provided DNS TLSA information.
- Inconsistencies within the DNS TLSA records received for a particular MX were not considered.

3.1 Data Sets for Analysis

We used the following gTLD and ccTLD DNS domains in our analysis:

- gTLD: `INFO`, `ORG`, `NET`
- Europe: `AT`, `BE`, `CH`, `CZ`, `DE`, `ES`, `EU`, `FR`, `IT`, `PL`, `UK`, `RU`, `SE`
- America: `CA`, `BR`
- Asia & Pacific: `AU`, `CN`, `JP`, `NZ`

The zone files were taken from [1] on September 3rd, 2021. However, the zone data do not cover the TLDs completely. Evaluation took place in November/December 2021, thus with a certain delay. One needs

to consider that any domain information follows the rules of 'eventual consistency'; thus there is no way to assume an 'ad-hoc' correctness of the received domain data. However, it would be very interesting if the domain providers would give a hint on their 'domain volatility' on a regular base.

The existing domain information was used to query the MX records first and from here subsequently the TLSA information was evaluated for the given MX. Thus, if a certain domain uses a 'foreign' MX server, it is sub summarized in its origin domain.

[Fig. 2] provides a breakdown of the analyzed domains and the presence of MX records per domain according to the data for totals and MX as given in [Tab. 1].

Compared to a 'Study on Domain Name System (DNS) Abuse' by the EU [31] released in February 2022, our data sets is limited by the following known conditions:

- Totals: 66.7 mio out of 363 mio (18.5%)
- DE zone: 6 mio out of 12 mio.
- EU zone: 1.1 mio out of 3.6 mio.

Since their study was realized between March 2021 and June 2021 and ours from October to December 2021, in particular the EU zone did undergo changes due to the depletion of none-EU citizens. Apart from that, no particular biases are expected for your analysis.

3.2 MX Delegation

A significant part of organizations, institutes, and companies prefer DNS MX delegation. This means that the receiving email host (given by the MX record) is not administered within the recipient domain, but rather a delegation was set up to an external provider.

While this might be beneficial for the receiving domain, it however discloses any incoming mail to a third party mostly in clear text, since the mail is stored on the (third party) MTA queue (store & forward). The European GDPR [8] covers this case known as 'order data processing'. In the context of IT security, this is however questionable, because

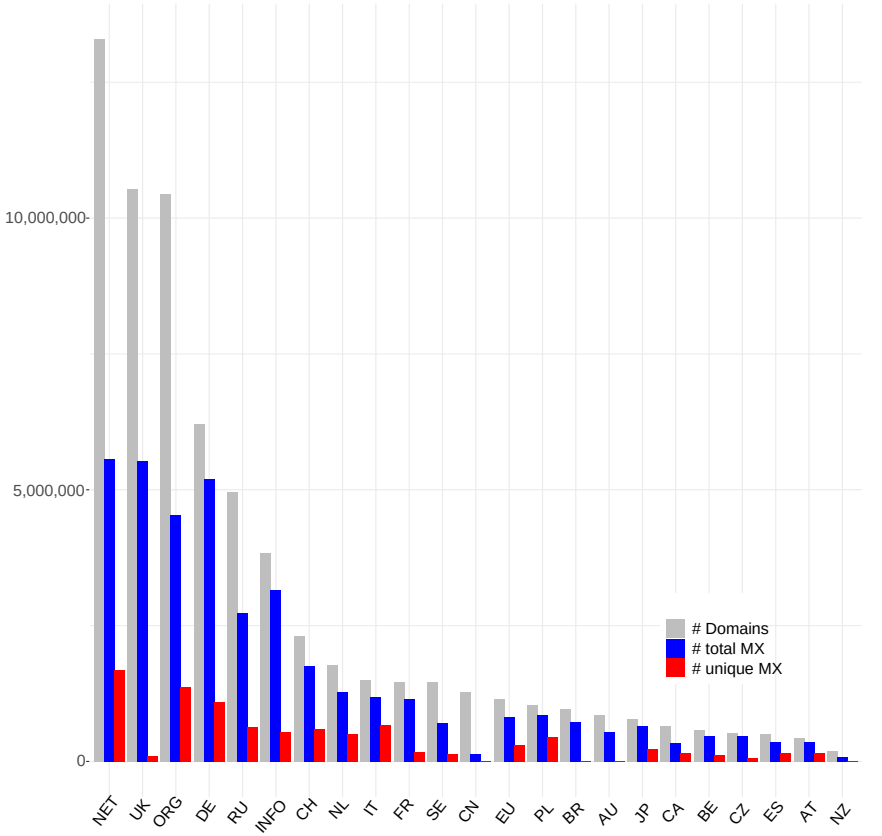

Abbildung 2: Number of analyzed domain names per gTLD and ccTLD (sorted by number of FQDN entries per TLD) together with the coverage of (total) `MX` records resolving to (unique) MX names which are used for the later TLSA lookup.

the sender (originator) of the particular mail never gave consensus that his/her message is stored on a foreign system in the first place.

For a private person, it may be perfectly legitimate to use '*gmail.com*' or '*outlook.com*' as mail provider (and visible by that address) but to forwarded mails (from users) through third-party systems might be at least questionable.

3.3 DNS Query Chain

The domain-specific DNS data were treated by the following analysis chain:

1. The domain data was subdivided into chunks based on the first letter of the domain name. Thus we had sets of data stretching `0-9`, and `a-z`. We could have sliced the domain data into sets of equal size, since in particular `0-9` are sparsely populated. But in order to keep the following analysis logically simple, a dictionary breakdown seemed to be appropriate and had little impact on performance.

2. The received `MX` information was filtered according to the following rules:

   - Only domains with an existing `MX` record were considered.
   - Domains *resolving* to '`localhost`' were omitted.
   - Domains *delegated* to `hotmail.com` or `outlook.com` were omitted as well.

Apart from the above exceptions, our query was aligned to a typical procedure a mail client would follow (here we use the mimic of *qmail-remote* as part of *s/qmail*) using a pedantic DNS lookup.

3.4 Operational Conditions and Observations

dnscache – the 'work horse' – was set up with the following conditions[1]

- Cache size [Byte]: 9000000 (8.58 MByte) – to keep DNS information
- Data size [Byte]: 30000000 (28.61 MByte) – heap memory used for TCP queries

[1] Daniel Bernstein recommended to set up a 100 MByte cache using `CACHESIZE=100000000` and `DATALIMIT=104857600` (see: `https://cr.yp.to/djbdns/cachesize.html`) together with an open file delimiter of `256`.

Depending on the input data, between 80 and 550 concurrent DNS queries were performed. The default maximum UDP connection limit for **dnscache** (initially defined to be 200) was raised up to 1000 and the corresponding TCP maximum connection limit was increased to 100 since during our initial studies we observed the dropping of UDP sessions given the lower connection limit. Forthcoming versions of *djbdnscurve6* will default to 400/40.

The 'throughput' of your analysis was impacted by two closely related factors:

1. The *validity* of the domain data thus pointing to existing domains (and not just 'parked' domain names) used and in turn

2. *incorrect domain data* (FQDN) yielding a '`servfail`' triggering multiple DNS look-ups potentially timing out.

dnscache was initially advised to perform a 'quadratic' DNS NS lookup in case an answer is not received immediately (`servfail`) making use of NS IP address randomization. UDP timeouts were given here as `{1,2,4,8,16}`. For the last part of the data gathering (after about 300 mio queries) it was reduced to `{1,2,3,4,5}` seconds while only allowing two lookups. In order to receive a final '`servfail`' 15 trials (to randomized NS) are required: The initial `MX` lookup, followed by the `AAAA` and the `A` query. Thus during those (accumulated) seconds, the respective query slot is blocked until the queried domain name is eventually regarded as 'unreachable'.

This lookup strategy is identical for mail clients and not artificially tuned to obtain a better throughput. In this sense, the result reflect truly the situation of a standard mail client trying to obtain TLSA records from the DNS.

In case a '`servfail`' is recognized occasionally, no particular actions are required. However, we've observed that particular parts of the domain name space are filled with 'garbage'. Here are three observations:

- Significant parts of domain names starting with '`my`' (among others) for the UK and NET TLD seem to be used to simply allocate and 'park' the particular domain name resulting in a '`servfail`'.

- The NS 'sc-[a-d].sinkhole.shadowserver.org' return lots of delegations to non-existing child NS (including their glue)[2].
- Additionally, in the UK domain we have often *double counting* of domain names, typically occupying the .co.uk and .uk name space.

4 Results

In our analysis and since we restrict our investigation to (E)SMTP services, here we used the domain name given in the MX record as a base prepended with TCP as service and port 25. No attempt was made to include *Submission* (on port 578) [18] or *SMTPS* (on port 465).

4.1 Bulk Results on Mail Servers, MX and TLSA Records

As depicted from [Tab. 1], we can estimate that less than 60% all domains have a valid MX server set up. Mail services are often outsourced; as can be seen by comparing the columns 'available' and 'unique' MX. Depending on the *TLD*, this relative percentage is quite different, but as a rule of thumb, one can conclude that again roughly about 75% MX services are realized by an external provider. Thus, the very same MTA is used at least to receive and queue (E)SMTP mails for external domains.

From the number of 'unique' MX the TLSA coverage is queried and valid responses are given in the second right column. However, the percentage is calculated using the ratio of 'unique' MX and MX with TLSA records given.

Here, we see the impact of MX concentration (or outsourcing): For domains with a high outsource rate (AU and BR) the TLSA use rises significantly if the MTAs in scope are provisioned with TLSA records.

As a side note: None of the MX in the China (CN) domain, which are equipped with a TLSA record are hosted in the CN domain them self. Rather, the majority of domain names here point to servers located in the NL, CH domain together with COM and NET.

[2] here, we receive 87.106.250.34 87.106.34.1 87.106.86.28 217.160.6.63 as list of IPs which have been tentatively excluded as NS for later lookups to increase speed; but didn't really solve the problem since the delegations happen at the 'sinkhole.shadowserver.org' NS.

Tabelle 1: Results on the MX (Mail Exchanger) and TLSA query/responses.

	Domains	**Number of MX**		**TLSA coverage**	
Domain	evaluated	available	unique	any	[%]
AT	4,36,132	351,897	159,878	2,366	1.48
AU	849,744	539,291	2,808	294	10.47
BE	58,3142	47,1817	109,734	6,146	5.60
BR	955,264	73,3387	2,458	464	18.88
CA	643,077	330,883	156,184	653	0.42
CH	2,304,768	1,753,310	589,175	4,270	0.72
CN	128,0957	134,696	9,342	87	0.93
CZ	526,665	469,462	69,356	2,631	3.79
DE	6,213,317	5,195,049	1,086,378	13,114	1.21
ES	498,069	363,879	159,191	1,606	1.01
EU	1,149,377	814,803	299,683	8,693	2.90
FR	1,466,033	1,145,294	180,188	3,946	2.19
INFO	3,836,691	3,151,408	540,424	6,761	1.25
IT	1,493,905	1,191,211	667,064	2,258	0.34
JP	783,319	646,255	235,005	96	0.04
NET	13,292,521	1,108,377	1,675,649	13,195	0,79
NL	1,782,553	1,282,416	511,681	63,715	12.45
NZ	192,513	8,8791	3,873	126	3.25
ORG	10,450,626	4,542,621	1,366,267	11,539	0,85
PL	1,032,558	850,241	452,154	1,493	0.33
RU	4,962,992	2,726,402	625,221	899	0.14
SE	1,465,433	702,868	13,8956	3,415	2.46
UK	10,536,401	5,527,898	104,687	4,853	4.64
all	66,736,057	38,576,497	9,145,356	152,620	1.67

4.2 Application and Use of DANE/TLSA Records for MTAs

In case TLSA records are deployed for a MTA, often not just one but several TLSA records are usually set up on the respective DNS content server. Thus, the number of 'TLSA coverage' in [Tab. 1] does not tell the returned DNS responses, but simply if any.

In a further step, we now evaluate all TLSA responses (up to a maximum of again six) and analyzed those in terms of

- *Usage*,
- *Selector*, and
- *Matching Type*.

- Fingerprint *uniqueness* and further,
- the number of responded TLSA RRs per query.

Usage

Let's now have a look at the *Usage* of the TLSA record, only considering the recommended *Usage* values {0,1,2,3}.

Tabelle 2: Breakdown of 'Usage' for received TLSA records (RR).

Domain	Number of TLSA RR	Usage 0	1	2	3
AT	4,058	1	1	475	3,581
AU	517	0	0	109	408
BE	8,764	0	0	1,347	7,417
BR	783	0	0	35	748
CA	1,215	0	0	168	1,047
CH	6,675	8	7	962	5,698
CN	87	0	0	9	114
CZ	3,806	0	0	407	3,399
DE	20,123	7	21	3,449	16,646
ES	2,860	0	0	265	2,595
EU	12,763	2	5	2,059	10,697
FR	5,826	0	0	780	5,046
INFO	10,323	15	7	1,877	8,424
IT	3,785	0	0	397	3,388
JP	209	0	0	61	148
NET	19,247	19	21	3431	15,776
NL	83,918	3	0	4,496	79,419
NZ	235	0	0	29	206
ORG	17,388	29	16	3,099	14,244
PL	2,555	0	1	178	2,376
RU	1,455	0	0	335	1,120
SE	5,758	0	0	753	5,005
UK	7,820	19	16	1,302	6,483
all	220,216	103	95	26,033	193,985

The small internal inconsistency in this table (summing up all *Usages* and compared against the number of TLSA records) is due to the fact that [21] allows more usage types; typically experimental ones, which are not covered by this analysis.

Following [Tab. 2], it seems to be clear that the TLSA deployment in the DNS is triggered by the fact that X.509 certificates do not require

PKIX evaluation and in consequence are 'authenticated' by the DNS only. The domain owners follow the current recommendations of [17] for publishing TLSA records since the *Usage* (`0`) and (`1`) are practically not present in our analysis. Further, there seems to be little intention to provide the (full) certificate chain upon the TLS handshake here, as required by *Usage* (`2`).

In terms of security, *Usage* (`2`) makes little sense only; whereas 'dynamic' certificate authorization by means of TLSA records seems to have precedence within our evaluation.

Selector

The *Selector* tells about the way the X.509 fingerprint is created and to be interpreted by the mail client.

Tabelle 3: Breakdown of 'Selector' for received TLSA records (RR).

Domain	Number of TLSA RR	Selector `0`	Selector `1`
AT	4,058	375	3,683
AU	517	98	419
BE	8,764	812	7,952
BR	783	45	738
CA	1,215	80	1,135
CH	6,675	1,217	5,458
CN	133	6	127
CZ	3,806	300	3,506
DE	20,123	5,249	14,874
ES	2,860	203	2,657
EU	12,762	1,893	10,869
FR	5,826	568	5,258
INFO	10,325	1,455	8,870
IT	3,785	356	3,429
JP	209	11	198
NET	19,246	2,869	16,377
NL	83,918	3,091	80,827
NZ	235	28	207
ORG	17,388	2,996	14,402
PL	2,555	239	2,316
RU	1,455	175	1,280
SE	5,758	444	5,314
UK	7,820	1,005	6,815
all	212,406	22,510	189,896

We have already discussed the difference between the X.509 certificate's fingerprint and the fingerprint of its public key (SPKI). Interestingly here, a 1:9 relationship is visible in favor of the SPKI [Tab. 3]. Clearly it has an operational impact: While the SPKI may stay constant over a longer period, the fingerprint of the X.509 certificate is subject of change after each renewal.

In order to cope with the renewal of certificates, their respective fingerprints need to be deployed in a key-rollover manner. We will discuss this issue again while looking at the concurrent number of TLSA fingerprints later.

Matching Type

Considering [Tab. 4] there is a clear preference to use SHA-256 instead of SHA-512 as hash function (by 45:1) for the fingerprint.

Practically, the use of SHA-256 or SHA-512 makes little difference:

- Computation speed to generate and evaluate either of both is mostly irrelevant; though favors SHA-512 over SHA-256.
- Impact on network traffic is also negligible: Whether 64 byte or 32 bytes are to be transferred is certainly not a relevant difference.
- Given security, we need to consider that a X.509 certificate spans perhaps 1000 byte (to be hashed), where as the SPKI is just in the order of few hundred byte. The main security impact originates from the avalanche quality of the hash function considering the relative small input and the repeating information in here (eg. algorithms) which reduces entropy.

Unique X.509 fingerprints

Surprisingly, deploying cryptographic information (like hash sums) in the public can be used to probe 'none-deniability' (or 'indisputablity'): If the hash sums are identical, the originating source is as well. In terms of X.509 fingerprints, we observe in [Tab. 5] significant support for this hypothesis.

We need to note that a TLSA fingerprint indisputable identifies a MTA, an email server. In [Tab. 5] the second column shows the number (of unique!) MTAs given their `MX` record. Columns 4 and 5 provide their

Tabelle 4: Breakdown of 'Matching Type' for received TLSA records (RR).

Domain	Number of TLSA RR	Matching Type 0	1	2
AT	4,058	0	3,974	84
AU	517	0	515	2
BE	8,764	0	8,538	226
BR	783	0	768	15
CA	1,215	0	1,214	1
CH	6,675	0	6,456	219
CN	113	0	132	1
CZ	3,806	0	3,705	101
DE	20,123	0	19,264	859
ES	2,860	0	2,830	30
EU	12,763	1	12,302	460
FR	5,826	0	5,687	139
INFO	10,331	8	9,946	377
IT	3,788	3	3,722	63
JP	209	0	209	0
NET	19,250	5	18,536	709
NL	83,918	0	83,195	723
NZ	235	0	234	1
ORG	17,404	16	16,717	671
PL	2,555	0	2,495	60
RU	1,455	0	1,423	32
SE	5,758	0	5,657	101
UK	7,826	12	7,544	170
all	220,206	45	215,163	4,874

fingerprints. We need to consider that occasionally the very same X.509 cert may be included with its SHA-256 and its SHA-512 hash. The evaluation of the X.509 certificates having equal hashsums is not part of this analysis.

TLSA Resource Record Deployment

Our data also allows finally to asset the number TLSA records given for a certain MX [Tab. 6].

In only one case, more than 6 TLSA records are given. Considering the frequency of deployed certificate fingerprints, most DNS (and MX) operators feel comfortable with one or two TLSA RRs per `MX` (in average 1.44). Whether the fingerprints point to different certs or the cert is covered by different *Matching Types* can not be answered by this analysis.

Tabelle 5: Unique certificate fingerprints for TLSA records (RR) per domain.

Domain	Number of TLSA MX	Number of TLSA RR	Unique fingerprint SHA-256	Unique fingerprint SHA-512
AT	2,366	4,058	352	17
AU	294	517	105	2
BE	6,146	8,764	315	15
BR	465	783	125	7
CA	633	1,215	123	2
CH	4,270	6,675	609	43
CN	87	133	50	2
CZ	69,356	3,806	304	22
DE	13,114	20,123	1,911	127
ES	1,606	2,860	197	7
EU	8,693	12,763	917	62
FR	3,946	5,826	441	52
INFO	6,761	10,331	935	61
IT	2,258	3,785	331	11
JP	96	209	48	1
NET	1,675,649	19,253	3,027	183
NL	63,715	83,918	1,054	63
NZ	126	235	55	2
ORG	1,366,267	11,539	2,177	183
PL	1,493	2,555	242	9
RU	625,221	889	256	20
SE	3,415	5,758	343	27
UK	7,840	4,853	782	39
all	3,863,817	210,848	14,699	957

5 Conclusions

Coverage and Deployment
The overall coverage of TLSA records in the DNS is still marginal (< 2%) [Fig. 3] in most areas of the Internet, with an exception for the Netherlands and Brasilia (domains `NL` and `BR`); though in the last case, we have an extreme narrow MX provider share. This coincides with the facts that

- we have a huge concentration of MX providers deploying,
- mostly 'official' X.509 certificates for their mail service are used without the need to deploy TLSA DNS records in addition.

Tabelle 6: Number of TLSA records (RR) per MX provisioned in the DNS.

	Number of		Number of TLSA records/MX							average
Domain	TLSA MX	TLSA RR	1	2	3	4	5	6	>6	**number**
AT	2,366	4,058	1,023	1,137	146	5	27	28		1.72
AU	294	517	105	159	104	12	0	19		1.76
BE	6,146	8,764	4,280	157	61	52	145	38		1.43
BR	465	783	160	296	3	4	0	1		1.68
CA	633	1,215	209	380	46	0	0	18		1.92
CH	4,270	6,675	2,418	1,588	114	49	63	38		1.56
CN	87	133	47	36	2	2	0	0		1.53
CZ	69,356	3,806	1,616	936	23	43	1	12		1.45
DE	13,114	20,123	8,400	3,780	306	153	211	263	1	1.53
ES	1,606	2,860	549	967	40	4	35	11		1.78
EU	8,693	12,763	5,768	2,427	159	104	162	73		1.47
FR	3,946	5,826	2,566	1,162	73	24	105	16		1.48
INFO	6,761	10,331	4,248	2,024	207	79	128	75		1.53
IT	2,258	3,785	1,025	1,086	75	12	45	15		1.68
JP	96	209	34	44	7	0	0	11		2.18
NET	13,195	19,253	9,200	3,101	325	190	151	217		1.56
NL	63,715	83,918	45,455	17,363	271	325	182	119		1.32
NZ	126	235	48	62	11	0	0	5		1.87
ORG	11,539	17,420	7,570	3,141	306	173	168	181		1.51
PL	1,493	2,555	598	809	35	26	20	5		1.71
RU	899	1,455	555	249	41	10	25	19		1.62
SE	3,415	5,758	1,547	1,662	86	17	57	46		1.69
UK	4,853	7,840	2,675	1,862	110	20	105	81		1.62
all	1,152,601	220,282	100,150	44,373	2,459	1,292	1,630	1,291	1	1.44

What also can be anticipated is the fact, that if an MX provider has a large market share availability, the lack of TLSA records has a significant impact on its coverage even for several distinct domains.

PKIX vs. DANE only: Usage

From the analyzed data [Tab. 2] we can conclude:

- If a TLSA record is present, there seems to be no requirement to enforce an additional PKIX verification of the mail server's X.509 certificate.
- No particular interests show up to deliver the entire X.509 certificate chain in the TLS handshake; thus only the MX server's cert is relevant. However, there is no common sense in this respect [Fig. 4].

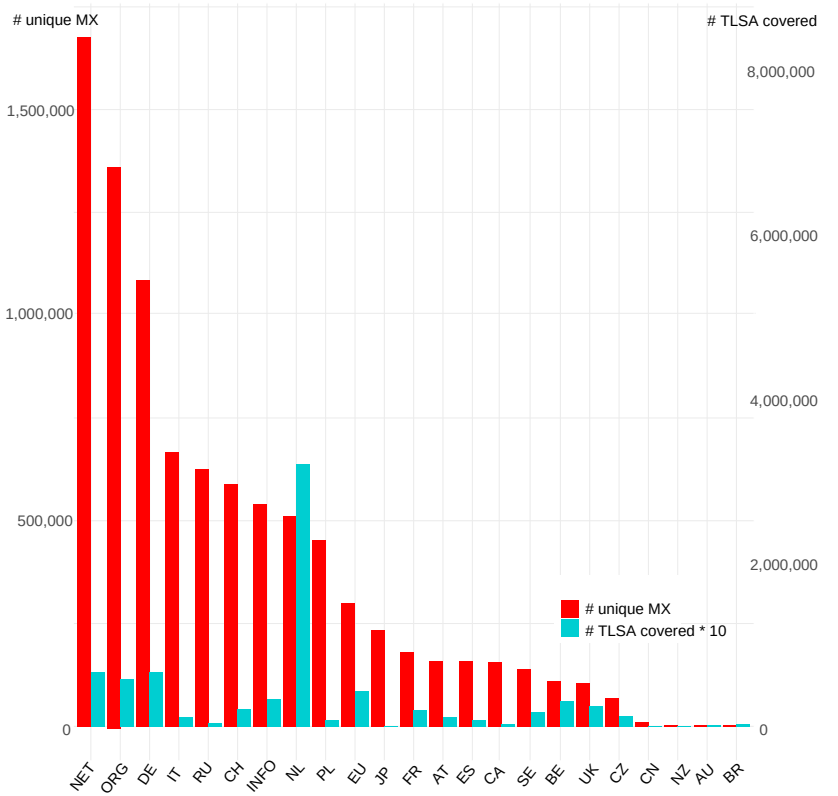

Abbildung 3: Coverage of TLSA records (small columns in light-blue and scaled by factor 10) in relation to the available (unique) `MX` records given as red columns; sorted by numbers of `MX` records per TLD.

- TLSA records are mainly of type '*DANE-EE*' using the (authenticated) DNS replies as trust anchor.

Operational Issues: Selector, Matching Type, and Redundancy

The quality and the robustness of the provided TLSA RRData information is covered (1) by the way the fingerprint is calculated [Tab. 3], (2) which hash function [Tab. 4] is used, and (3) potentially how many different X.509 certificates are deployed per MX [Tab. 6].

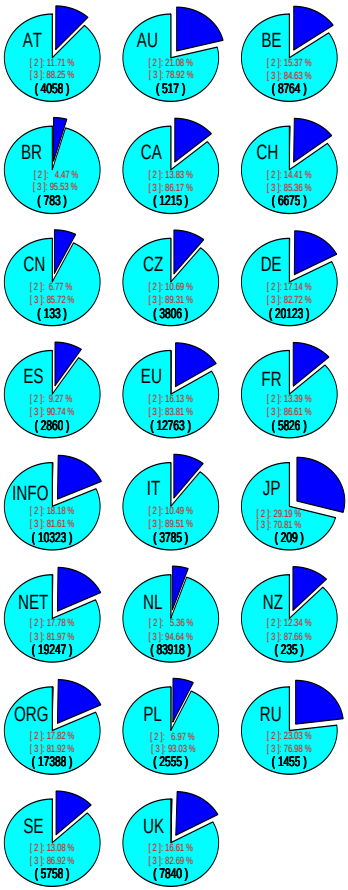

Abbildung 4: The particular TLSA ’Usage’ per domain; PKIX conforming usages are practically negligible; thus only Usage 2 and 3 are given here together with the number of received totals in parenthesis.

- MX administrators seem to follow a pragmatic approach, in particular to depend on the SPKI as fingerprint only.
- For most cases, just one or two TLSA records are published in the DNS.
- Also, for the majority of X.509 certificates accompanied by a TLSA record, the administrators seem to be comfortable with a SHA-256

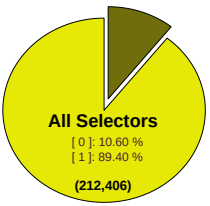

Abbildung 5: The 'Selector' for all domains investigated; total numbers of TLSA records for those selector given in parenthesis.

hash sum [Fig. 6], [Fig. 5].

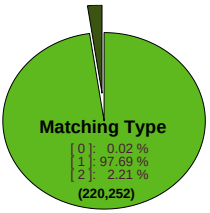

Abbildung 6: The 'Matching Type' for all domains investigated; total number of TLSA records in parenthesis.

Which MX to trust: Uniqueness

The reason, why only so few different certificate hashes are in the wild according to [Tab. 5] is yet unclear. The following assumptions though may be considered:

1. `MX` delegation as discussed before and shown in [Tab. 1] is even more prominent as anticipated initially.
2. The X.509 certificates are *wild-card* certs covering the entire MX setup.

In any case, identical X.509 certificates are inferior to potential matching the X.509 DN/SAN and the MX hostname. Unfortunately, [16] closely relates TLSA evaluation with DNSSec and concludes "secure verification of SMTP TLS certificates matching the server name is not

possible without DNSSEC" (section 1.3.2) and additionally permits wildcards within the *Subject Alternative Name* (section 3.2.3) of the X.509 certificate.

5.1 Observations and Recommendations for DNS and Mail Providers

MX TLSA Naming

- In order to unambiguously deploy a TLSA record for MX services in the DNS, the form

  – `_25._tcp.mx.example.com`

  and not

  – `_25._tcp.example.com`

  shall be used. In this way, one can 'chain' IP addresses with DNS `MX` and TLSA information together with the X.509 certificate's ownership given in the *Canonical Name* (`CN`) and/or present in the X.509's *Subject Alternative Name* (`SAN`) field [11]. It should be remembered that RFC 2821 [29] forbids the use of a `CNAME` within `MX` records.

- For the very same reason, the synthesized `FQDN` shall immediately point to the TLSA record and not to a `CNAME`.

Hash functions/values

- For various reasons [19], the use of SHA-512 is in general preferable.

- In order to save space in the DNS message, the IETF should consider to strip down the length of the result to fewer bytes in forthcoming RFCs.

- Given the better SHA-512 hash algorithm, even the upper or lower half of the SHA-512 hash sum provides significant entropy to make any potential X.509 certificate distinguishable even with reduced collision chance.

Immutable fingerprints

- It should be considered – though outside this analysis – to define a third significant hash value for a X.509 certificate: Its *immutable fingerprint*.
- This should cover the combination of the public name (the owner's *Distinguished Name* `DN` and/or the *Subject Alternative Name* `SAN`) together with the public key.

Deployment of Fingerprints for MTAs

- According to our analysis – in case of provisioned TLSA records – the overwhelming majority of MX don't expect the X.509 certificate to be suject of additional PKIX verification (resulting in a DANE-EE setup). This usage complies with RFC 7672 [16] (section 3.1.1.).
- Thus, the *Usages* PKIX-TA and PKIX-EE are practically not present. In constrast, most MTAs which support TLS (over port 25 employing *StartTLS*) – and in particular the main MX providers, like Google, Microsoft, and many others – only take benefit from PKIX compliant certificates.
- MX service providers making use of TLSA/DANE records often deploy the same X.509 certificate (and also private key) for their set of MTAs, as can be anticipated by the fingerprint.

Irrespectively of the TLSA record, the X.509 certificate includes information about the 'user', the *subject* of the certificate given as *Distinguished Name* (DN) and/or in the SAN. In the case of *Mailbox.org* and *Protonmail* not only the certificate's *fingerprints* are the same, but in addition their *private key*. In fact, since the *Selector* (`1`) indicates the use

of SPKI, we conclude that all (physically) different MTAs are considered as one logical MX on the application layer.

For instance, *s/qmail*'s sending process **qmail-remote** is able to match the domain name of the MTAs with the information present in X.509 certificate. Thus, even if the public and private keys in X.509 certificates (on different MX instances) are identical, the other fields in the cert should be accustomed accordingly, providing the possibility to uniquely identify the connected MX and should of course coincide with its network configuration.

Closing comments

The RFC 7671 and 7672 [17, 16] express a tight binding between DNSSec and TLSA and partial overruling standard procedures and best practices used for (E)SMTP mail delivery and certificate validation.

Considering a DNS setup which is not depending on DNSSec and potentially providing 'true' security, different from the approach in [16], we recommend for (E)SMTP servers accomplished by TLSA records:

- In case of *DANE-EE* TLSA record each MX server should be provisioned with a unique X.509 certificate allowing to match the CN or the SAN attributes with the *FQDN* of the MX server.
- Don't use 'wildcard' names for your MX services.
- Generate a unique public key for each MX service within the X.509 certificate.
- Follow a canonical procedure for certificate roll-over while obeying the validity period of the X.509 certificate.

Currently within the EU, a proposal is issued looking for a 'secure' DNS public recursor [9] competing with other public resolvers, for instance Quad9 [https://www.quad9.net] outside the EU. While this sounds like a honorable approach to none-disclose DNS meta-data, from the point of (E)SMTP mail services it is more important that mails today are still mostly send in clear text; even though transport encryption based on TLS is in place and 'secured' by DNSSec and TLSA.

Since (E)SMTP is a store-and-forward protocol and considering the current mail situation in the Internet ('MX delegation') as provided in

this analysis, European citizens disclose significant fractions of their entire private and confidential email communication to companies outside the EU without even realizing this. This should be of great concern.

6 Appendix

6.1 Combined results

Tabelle 7: Total evaluated domains and received MX and RR.

Domain	Number of ...			MX with TLSA		Usage				Selector		Matching Type		
	Domains	covered MX	unique MX	covered	RRs	0	1	2	3	0	1	0	1	2
INFO	3,836,691	3,151,408	540,424	6,761	10,323	15	7	1,877	8,424	1,455	8,870	8	9,946	377
NET	13,292,521	5,562,618	1,675,649	13,195	19,253	19	21	3,421	15,776	2,869	16,377	5	18,536	709
ORG	10,450,626	4,542,621	1,366,267	11,539	17,420	29	16	3,099	14,244	2,996	14,402	16	16,717	671
BR	955,264	733,387	2,458	464	783	0	0	16	748	45	738	0	768	15
CA	643,077	330,883	15,6184	653	1,215	0	0	168	1,047	80	1,135	0	1,214	1
AT	436,132	351,897	159,878	2,366	4,058	1	1	475	3,581	375	3,683	0	3,974	84
BE	583,142	471,817	109,734	6,146	8,764	0	0	1,347	7,471	812	7,952	0	8,538	226
CH	2,304,768	1,753,310	589,175	4,270	6,675	8	7	962	5,698	1,217	5,458	0	6,456	219
CZ	526,665	469,462	69,356	2,631	3,806	0	0	407	3,399	300	3,506	0	3,705	101
DE	6,213,317	5,195,049	1,086,378	13,114	20,123	7	21	3,349	16,646	5,249	14,874	0	19,264	856
ES	498,069	363,879	159,191	1,606	2,860	0	0	265	2,595	203	2,657	0	2,830	30
EU	1,149,377	814,803	299,683	8,693	12,763	2	5	2,059	10,697	1,893	10,302	1	12,302	460
FR	1,466,033	1,145,294	180,188	3,946	5,826	0	0	780	5,046	568	5,258	0	5,687	139
IT	1,493,905	1,191,211	667,064	2,258	3,785	0	0	387	3,388	356	3,429	3	3,722	63
NL	1,782,553	1,282,416	511,681	63,715	83,918	3	0	4,496	79,419	3,091	80,827	0	83,195	723
PL	1,032,558	85,0241	452,154	1,493	2,555	0	1	178	2,376	239	2,316	0	2,495	60
RU	4,962,992	2,726,402	625,221	899	1,455	0	0	335	1,120	175	1,280	0	1,423	32
SE	1,465,433	702,868	138,956	3,415	5,758	0	0	753	5,005	444	5,314	0	5,657	101
UK	10,536,401	5,527,898	104,687	4,853	7,840	19	16	1,302	6,483	1,005	6,815	12	7,644	170
AU	849,744	53,921	2,808	294	517	0	0	109	408	98	419	0	515	2
CN	1,280,957	134,696	9,342	87	133	0	0	19	114	6	127	0	132	1
JP	78,3319	646,255	235,005	96	209	0	0	61	148	11	198	0	209	0
NZ	19,2513	8,8791	3,873	126	235	0	0	29	206	28	207	0	234	1

6.2 DNS lookup details

dnsmxip
Within the first step invoking **dnsmxip**, typically the following information is revealed (displayed with the retrieving domain name):

```
nzg-journalisten.nl= mail.nzg-journalisten.nl: 10 [2a00:f10:305:0:1c00:8dff:fe00:4f3
 185.87.187.187]

sabai.ch= sabai.ch: - [104.223.45.185 104.223.37.154]

gapsfamily.org= mx20.mailspamprotection.com: 20
[146.66.121.62 35.206.105.37 146.66.121.164 146.66.121.213 146.66.121.88
 146.66.121.63 146.66.121.17
146.66.121.87 146.66.121.160 146.66.121.6 35.223.167.9 35.192.5.156 35.209.67.207
 146.66.121.83 34.70.37.227]
mx10.mailspamprotection.com: 10 [35.208.121.216 146.66.121.211 146.66.121.215
 35.224.11.180 146.66.121.217
146.66.121.105 146.66.121.82 146.66.121.69 146.66.121.216 35.208.244.18
 35.225.161.143 104.197.42.21
146.66.121.218 35.192.135.139] mx30.mailspamprotection.com: 30 [146.66.121.162
 35.238.96.225
146.66.121.80 146.66.121.165 146.66.121.212 146.66.121.219 146.66.121.9 34.69.117.62
146.66.121.166 35.208.10.124 146.66.121.214 146.66.121.100 146.66.121.161
 35.206.120.11 146.66.121.61]}
```

Listing 1: Raw results of the `MX` query showing the queried domain name, the MX record and the associated IP information as (`A/AAAA`) response.

Here, we can see the following cases (distinguished by the empty line):

a) [`nzg-journalisten.nl`]: A successful `MX` lookup including the FQDN of the MTA, its respective weight and followed in parenthesis by the (sequence of) IP address(es) assigned to.

b) [`sabai.ch`]: An unsuccessful `MX` lookup indicated by the '-' sign as weight.

c) [`gapsfamily.org`]: A delegation for the (E)SMTP email service to a third party with different MTAs and IP addresses.

In any case, multiple MX are possible; but for the following TLSA lookup only the first six `MX` entries were considered.

One consequence of the chosen breakdown approach is that domain names expressed in *Punycode* [27, 28] are entirely included in the 'x'-slice.

dnstlsa
The following TLSA query using **dnstlsa** benefits from the advantage that the received `MX` information is valid and thus an adjacent authoritative Name Server is responding. The following information is typically received:

```
maastrichtcentraal.nl= Usage: [3], Selector: [1], Type: [1]
 0966f3ce8b64dc8fe4faf2e68c4c91b794062029198a7bd5804b3aa1118fce4a  Usage: [3],
 Selector: [1], Type: [1]
 f244913ef6ca3b50e638587587629ffc84c97644736e33a97cd78fe8df898fa0

maastrichtsegevelstenen.nl= Usage: [3], Selector: [1], Type: [1]
 f244913ef6ca3b50e638587587629ffc84c97644736e33a97cd78fe8df898fa0 Usage: [3],
 Selector: [1], Type: [1]
 0966f3ce8b64dc8fe4faf2e68c4c91b794062029198a7bd5804b3aa1118fce4a
```

Listing 2: Raw results of the TLSA query resulting in a subsequent TLSA response of two RRs for the same MX

We can see that `MX` delegation is used here (and in fact, for the NL zone a lot of MX with '`mail.*.nl`' point to the same MTA) having identical X.509 certificate fingerprints.

6.3 Protonmail's MX TLSA fingerprints

The Swiss Mail provider *Protonmail* uses TLSA/DANE records in a way to setup a virtual mail service deploying the same X.509 certificate to several hosts:

```
mail.protonmail.ch= Usage: [3], Selector: [1], Type: [1]
 76bb66711da416433ca890a5b2e5a0533c6006478f7d10a4469a947acc8399
Usage: [3], Selector: [1], Type: [1]
 6111a5698d23c89e09c36ff833c1487edc1b0c841f87c49dae8f7a09e11e97
mailsec.protonmail.ch= Usage: [3], Selector: [1], Type: [1]
 6111a5698d23c89e09c36ff833c1487edc1b0c841f87c49dae8f7a09e11e97
Usage: [3], Selector: [1], Type: [1]
 76bb66711da416433ca890a5b2e5a0533c6006478f7d10a4469a947acc8399

mxext1.mailbox.org= Usage: [3], Selector: [1], Type: [1]
 4758af6f02dfb5dc8795fa402e77a8a0486af5e85d2ca60c294476aadc40b2
```

```
Usage: [3], Selector: [1], Type: [1]
 996ad31d65e03f038b8ec950f6f26611529da03e3a283e4400cba2edd04b8a
Usage: [3], Selector: [1], Type: [1]
 e41cc7633029afdba53744d7e5fc31ef507e592de9dfb33557bf3b9a792394
mxext2.mailbox.org= Usage: [3], Selector: [1], Type: [1]
 4758af6f02dfb5dc8795fa402e77a8a0486af5e85d2ca60c294476aadc40b2
Usage: [3], Selector: [1], Type: [1]
 996ad31d65e03f038b8ec950f6f26611529da03e3a283e4400cba2edd04b8a
Usage: [3], Selector: [1], Type: [1]
 e41cc7633029afdba53744d7e5fc31ef507e592de9dfb33557bf3b9a792394
mxext3.mailbox.org= Usage: [3], Selector: [1], Type: [1]
 996ad31d65e03f038b8ec950f6f26611529da03e3a283e4400cba2edd04b8a
Usage: [3], Selector: [1], Type: [1]
 4758af6f02dfb5dc8795fa402e77a8a0486af5e85d2ca60c294476aadc40b2
Usage: [3], Selector: [1], Type: [1]
 e41cc7633029afdba53744d7e5fc31ef507e592de9dfb33557bf3b9a792394
```

Listing 3: Protonmail's and Mailbox.org TLSA records

6.4 Further information

Availability of Results
The evaluated raw data sets are public available at `https://github.com/ErwinHo/DNS_TLSA_Survey` for further analysis. Our analysis extends previous ones given in [38, 30].

Comparable Surveys
Though our results are different from previous ones and are based on a distinct methodology, it might be interesting to follow the DANE mailing list reachable via `https://www.mail-archive.com/dane-users@sys4.de/info.html` for comparable investigations. Other DANE lookups covering about the same period as our analysis can be found in [15].

6.5 DNS query Recommendations

During our TLSA survey we've produced more than 300 mio DNS queries over a large part of the Internet's domain names; covering smaller and very large domains. We have used an alphabetic search order which in general works well. However, given parts of the domain name base to be contiguously tainted and full with garbage data turned out to be sub-optimal and slowing down the lookup process significantly.

We suggest to modify any future search strategy in the following respect:

- Generate a hash of the domain data (per TLD).
- Use the first two hexadecimal character of the hash sum as index; yielding 256 different sets.
- Perform the query on parallel cores or several machines; a 32 core CPU can easily handle 8 sets in parallel.
- It is not necessary to use a quadratic or exponential retry schema in case a 'servfail' is encountered.

References

[1] URL: https://zonefiles.io/.

[2] E. Allman. *cf/README for sendmail 8.12.3*. URL: https://www.sendmail.org/~ca/email/doc8.12/cf/m4/starttls.html.

[3] R. Arends u. a. *DNS Security Introduction and Requirements*. März 2005. URL: https://datatracker.ietf.org/doc/html/rfc4033.

[4] R. Barnes. *Use Cases and Requirements for DNS-Based Authentication of Named Entities (DANE)*. Okt. 2011. URL: https://datatracker.ietf.org/doc/html/rfc6394.

[5] D. Bernstein. *djbdns*. URL: https://cr.yp.to/djbdns.html.

[6] Huitema C., S. Dickinson und A. Mankin. *DNS over Dedicated QUIC Connections (draft)*. Mai 2022. URL: https://datatracker.ietf.org/doc/html/rfc9250.

[7] J. Callas u. a. *OpenPGP Message Format*. Nov. 2007. URL: https://datatracker.ietf.org/doc/html/rfc4880.

[8] European Comission. *Protection of personal data*. Sep. 2014. URL: http://eur-lex.europa.eu/legal-content/EN/LSU/?uri=celex:31995L0046.

[9] European Commission. *Equipping backbone networks with high-performance and secure DNS resolution infrastructures - Works*. Jan. 2022. URL: https://hadea.ec.europa.eu/calls-proposals/equipping-backbone-networks-high-performance-and-secure-dns-resolution-infrastructures-works_en.

[10] OpenSSL Technical Committee. URL: https://www.openssl.org/.

[11] D. Cooper u. a. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. Mai 2008. URL: https://datatracker.ietf.org/doc/html/rfc5280.

[12] S. Deering und R. Hinden. *Internet Protocol, Version 6 (IPv6) Specification*. Juli 2017. URL: https://datatracker.ietf.org/doc/html/rfc8200.

[13] M. Dempsky. *DNSCurve: Link-Level Security for the Domain Name System*. Feb. 2012. URL: https://datatracker.ietf.org/doc/draft-dempsky-dnscurve/.

[14] V. Dukhovni. URL: https://stats.dnssec-tools.org/#/.

[15] V. Dukhovni. *Update on stats 2021-11*. Nov. 2021. URL: https://www.mail-archive.com/dane-users@sys4.de/msg00473.html.

[16] V. Dukhovni und W. Hardaker. *SMTP Security via Opportunistic DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS)*. Okt. 2015. URL: https://datatracker.ietf.org/doc/html/rfc7672.

[17] V. Dukhovni und W. Hardaker. *The DNS-Based Authentication of Named Entities (DANE) Protocol: Updates and Operations Guidance*. Okt. 2015. URL: https://datatracker.ietf.org/doc/html/rfc7671.

[18] R. Gellens und J. Klensin. *Message Submission for Mail*. Nov. 2011. URL: https://datatracker.ietf.org/doc/html/rfc6409.

[19] S. Gueron, S. Johnson und J. Walker. *SHA-512/256*. 2010. URL: https://eprint.iacr.org/2010/548.pdf.

[20] P. Hazel. *Chapter 43 - Encrypted SMTP connections using TLS/SSL*. URL: https://www.exim.org/exim-html-current/doc/html/spec_html/ch-encrypted_smtp_connections_using_tlsssl.html.

[21] P. Hoffman und J. Schlyter. *The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol TLSA*. Okt. 2011. URL: https://datatracker.ietf.org/doc/html/rfc6698.

[22] E. Hoffmann. *6. Transport Layer Security with s/qmail*. URL: https://www.fehcom.de/sqmail/sqmaildoc_06.html.

[23] E. Hoffmann. *djbdnscurve6*. URL: https://www.fehcom.de/ipnet/djbdnscurve6.html.

[24] E. Hoffmann. *fehQlibs*. URL: https://www.fehcom.de/ipnet/qlibs.html.

[25] E. Hoffmann. *s/qmail*. URL: https://www.fehcom.de/sqmail/sqmail.html.

[26] Federal Office for Information Security (Germany). *BSI TR-03108-1: Secure E-Mail Transport (Version 1.0.1)*. Sep. 2016. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03108/TR03108-1.pdf.

[27] J. Klensin. *Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework*. Juni 2010. URL: https://datatracker.ietf.org/doc/html/rfc5890.

[28] J. Klensin. *Internationalized Domain Names in Applications (IDNA): Protocol*. Aug. 2010. URL: https://datatracker.ietf.org/doc/html/rfc5891.

[29] J. Klensin. *Simple Mail Transfer Protocol*. Apr. 2001. URL: https://datatracker.ietf.org/doc/html/rfc2821.

[30] H. Lee u. a. *A Longitudinal and Comprehensive Study of the (DANE) Ecosystem in Email*. Aug. 2020. URL: https://www.usenix.org/conference/usenixsecurity20/presentation/lee-hyeonmin.

[31] I. Paulovics, A. Duda und Korczynski M. (European Commission). *Study on Domain Name System (DNS) Abuse*. Jan. 2022. URL: https://op.europa.eu/en/publication-detail/-/publication/7d16c267-7f1f-11ec-8c40-01aa75ed71a1/language-en/.

[32] E. Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.3)*. Aug. 2018. URL: https://datatracker.ietf.org/doc/html/rfc8446.

[33] K. Rikitake, K. Nakao und H. Nogawa. *UDP Large-Payload Capability Detection for DNSSEC*. Mai 2008. URL: https://www.jstage.jst.go.jp/article/transinf/E91.D/5/E91.D_5_1261/_article.

[34] S. Santesson u. a. *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*. Juni 2013. URL: https://datatracker.ietf.org/doc/html/rfc6960.

[35] J. Schaad, B. Ramsdell und S. Turner. *Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification*. Apr. 2019. URL: https://datatracker.ietf.org/doc/html/rfc8551.

[36] J. Schlyter und W. Griffin. *Using DNS to Securely Publish Secure Shell (SSH) Key Fingerprints*. Jan. 2006. URL: https://datatracker.ietf.org/doc/html/rfc4255.

[37] W. Venema. *Postfix TLS Support*. URL: http://www.postfix.org/TLS_README.html.

[38] l. Zhu u. a. *Measuring DANE TLSA Deployment*. Apr. 2015. URL: https://link.springer.com/chapter/10.1007%2F978-3-319-17172-2_15.