A Longitudinal and Comprehensive Study of the DANE Ecosystem in Email

Hyeonmin Lee*
Seoul National University

Aniketh Girish*
Amrita Vishwa Vidyapeetham

Roland van Rijswijk-Deij
University of Twente & NLnet Labs

Taekyoung "Ted" Kwon
Seoul National University

Taejoong Chung
Rochester Institute of Technology

Abstract

The DNS-based Authentication of Named Entities (DANE) standard allows clients and servers to establish a TLS connection without relying on trusted third parties like CAs by publishing `TLSA` records. DANE uses the Domain Name System Security Extensions (DNSSEC) PKI to achieve integrity and authenticity. However, DANE can only work correctly if each principal in its PKI properly performs its duty: through their DNSSEC-aware DNS servers, DANE servers (e.g., SMTP servers) must publish their `TLSA` records, which are consistent with their certificates. Similarly, DANE clients (e.g., SMTP clients) must verify the DANE servers' `TLSA` records, which are also used to validate the fetched certificates.

DANE is rapidly gaining popularity in the email ecosystem, to help improve transport security between mail servers. Yet its security benefits hinge on deploying DANE correctly. In this paper we perform a large-scale, longitudinal, and comprehensive measurement study on how well the DANE standard and its relevant protocols are deployed and managed. We collect data for *all* second-level domains under the `.com`, `.net`, `.org`, `.nl`, and `.se` TLDs over a period of 24 months to analyze server-side deployment and management. To analyse the client-side deployment and management, we investigate 29 popular email service providers, and four popular MTA and ten DNS software programs.

Our study reveals pervasive mismanagement in the DANE ecosystem. For instance, we found that 36% of `TLSA` records cannot be validated due to missing or incorrect DNSSEC records, and 14.17% of them are inconsistent with their certificates. We also found that only four email service providers support DANE for both outgoing and incoming emails, but two of them have drawbacks of not checking the `Certificate Usage` in `TLSA` records. On the bright side, the administrators of email servers can leverage open source MTA and DNS programs to support DANE correctly.

*This work was done while the authors did an internship at Rochester Institute of Technology.

1 Introduction

Transport Layer Security (TLS) is responsible for securing Internet traffic in a variety of protocols such as DNS and HTTP. Coupled with a Public Key Infrastructure (PKI), TLS relies on certificates to bind entities to their public keys. Certificates are typically issued by Certificate Authorities (CAs), in a hierarchical fashion. At the top level of the hierarchy, there are root CAs, who have self-signed certificates since they cannot rely on other trusted third parties.

However, the current PKI model, discussed above, has been criticized for its potential vulnerability, since any CA can issue certificates for *any* domain name. Historically, we have observed that compromised CAs issued valid-looking but fraudulent certificates inappropriately [15, 58, 75]. Since then, a number of new protocols and extensions [40, 41, 48, 62, 68] have been proposed to mitigate these problems. However, none of these fundamentally solves the problem: *the validation process of a certificate still relies on CAs.*

To address this issue, the DNS-based Authentication of Named Entities (DANE) protocol [38] was proposed to support TLS *without* relying on trusted third-parties like CAs. At its core, a domain name owner that runs a TLS server such as HTTPS, or secure email via SMTP+STARTTLS, can publish its certificate information as a DNS record called the `TLSA` record, which can be used by TLS clients to verify the authenticity of the certificate in a non-PKI fashion. Furthermore, the integrity and authenticity of the `TLSA` records are guaranteed by the DNS Security Extensions (DNSSEC) [16–18]. Thus, a TLS server can easily publish and serve its certificate without relying on CAs, and TLS clients can also verify the certificate by (1) fetching `TLSA` records, (2) validating them using DNSSEC signatures, and (3) checking if the `TLSA` records are consistent with the certificate from the TLS server.

Due to its simple but robust security guarantees, there have been a number of attempts to deploy DANE for the Web PKI (HTTPS). However, DANE has never been adopted due to two operational challenges. First, a client (i.e., browser) may be behind a middlebox, which are notorious for discarding

TLSA or DNSSEC records. Second, the browser needs to make additional DNS queries to retrieve the TLSA and DNSSEC records, which incurs additional latency. Thus, modern web browsers do not usually support DANE [47].

Fortunately, many email service providers have begun to deploy DANE for their SMTP services as users are tolerant to millisecond-order additional delays in sending and receiving emails—moreover, DANE can solve security challenges in SMTP not solved so far, such as STARTTLS downgrade attacks [27] and receiver authentication [37].

In response to emerging threats in email security [30], the Dutch and German national governments require DANE support from vendors in public tenders [13, 19] and certain TLD registries (e.g., the .se and .nl registries) have employed financial incentives for registrars providing email hosting services to deploy DANE [59]. Finally, popular mail service providers have also begun to deploy DANE; Web.de (one of the largest free webmail providers in Germany) supports outbound DANE since 2016 [29], and Comcast (one of the largest ISPs in the US) did the same thing [26] in August 2017.

Like other PKIs, however, DANE can only function correctly when all principals fulfil their responsibilities: TLS servers presenting certificates, DNS servers publishing TLSA records, DNS clients validating DNS responses using DNSSEC, and TLS clients verifying certificates using TLSA records. Unfortunately, the complexity of DANE leads to many opportunities for mismanagement. For instance, on the server side, TLSA records may have DNSSEC errors such as expired signatures, or the certificates may be inconsistent with published TLSA records. On the client side, DNS resolvers may not validate TLSA records properly, or buggy TLS applications do not bother to check the validity of certificates.

Surprisingly little is known about the practice of the current DANE PKI ecosystem for email services. While there have been some studies of DANE [83], no prior work has studied the DANE PKI in SMTP longitudinally or comprehensively.

In this paper, we present a comprehensive study of the entire DANE ecosystem for SMTP. To study server-side behavior, our work leverages daily snapshots for 24 months and hourly snapshots for 4 months of MX records and TLSA records for *all* second level domain names that end with .com, .net, .org, .nl, or .se. For the MX records present, we retrieve the certificates of the corresponding email servers. To study client-side behavior, we investigate how DANE is supported by analyzing (1) the 29 most popular email service providers, (2) their DNS resolvers, (3) software implementations of popular or DANE-supporting mail transfer agents (MTAs), and (4) software implementations of popular DNS programs.

Our analysis reveals many instances of troubling and persistent mismanagement in the DANE PKI in SMTP:

- *First*, we find nearly 36% of TLSA records cannot be validated due to missing or incorrect DNSSEC records, e.g., some 19% are signed but lack a secure delegation (i.e., DS records).
- *Second*, even though most of the mail servers that provide TLSA records (99.5%) present their certificates through STARTTLS, we find that over 14% of them do not match the presented certificates.
- *Third*, when focusing on 29 popular email providers, we find that only four of them support DANE for their outgoing and incoming emails and one provider only supports DANE for incoming emails.
- *Finally*, we tested four popular MTA and ten popular DNS implementations to see if email providers can easily support DANE; we find that two popular MTAs correctly support DANE for both incoming and outgoing emails in conjunction with four DNS implementations that support TLSA records and DNSSEC.

Overall, our results show that DANE deployment is rare, but steadily increasing (especially in some country-code TLDs). Unfortunately, we also find widespread mismanagement of certificates and TLSA records. On the bright side, however, only a few players can easily make changes in order to bring the benefits and a greater adoption of DANE to end users, which are mainly large email providers and MTA and DNS Software providers.

To allow other researchers and administrators to reproduce and extend our work, we publicly release all of our analysis code and data to the research community at

https://dane-study.github.io

2 Background

In this section, we provide an overview of DNS, DNSSEC, DANE, and explain how they work together to secure email transport (i.e., SMTP).

DNS and DNSSEC DNS maintains the mapping between domain names and their associated values such as their IPv4 addresses (A records) and their mail servers' domain names (MX records). Unfortunately, the original DNS protocol [55] has serious security problems (e.g., no authentication of DNS records), making DNS vulnerable to numerous attacks such as DNS hijacking and cache poisoning [21, 42, 70]. To prevent such attacks, the DNS Security Extensions (commonly referred to as DNSSEC) were introduced to provide integrity and authenticity of DNS records using three new record types:

- DNSKEY records, which contain public keys used in DNSSEC.
- RRSIG records, which contain the cryptographic signatures (of DNS records) generated by the private keys; their corresponding public keys are in DNSKEY records.
- DS records, which are hashes of DNSKEYs. These records *must* be uploaded to the parent DNS zone to construct a chain of trust, which reaches up to the root DNS zone.

TLSA Records DANE introduces an additional DNS record type, called the TLSA record [38], which provides information that can verify the certificate of a corresponding domain name. There can be multiple applications that require TLS for a single domain name. Thus, a TLSA record is stored for a particular location, which is a combination of a port number, a protocol (i.e., TCP or UDP), and a base domain name. For a given base domain name, this allows specification of different certificates for different combinations (i.e., different applications). For example, to request a TLSA record for an SMTP server that has as its MX record mail.example.com and supports STARTTLS on port 25, the derived domain name must be _25._tcp.mail.example.com to fetch its TLSA record. A TLSA record consists of four fields (details in [38]):

- Certificate Usage, which specifies how the presented certificates from the TLS server can be validated with the Certificate Association Data (see below). There are four Certificate Usages: first, it can specify that the certificate for Certificate Association Data should be used as either (a) a trust anchor (i.e., a root certificate), thus permitting any leaf certificates as long as they are signed by the trust anchor (DANE-TA), or (b) a leaf certificate (DANE-EE), both of which do not require any IETF PKIX validation. In other words, if the presented certificate of which Certificate Usage in the fetched TLSA record is either DANE-TA or DANE-EE, the TLS client does not need to check if the certificate is signed by trusted CAs or is already in the root certificate stores. Similarly, the PKIX-TA usage can specify that (c) Certificate Association Data has to be used as a trust anchor, or (d) PKIX-EE for a leaf certificate. Note that the presented certificate *must* pass PKIX certification path validation using a set of root certificate stores, which are mutually agreed between the client and the server.
- Selector, which specifies the type of Certificate Association Data, indicating whether the Certificate Association Data is derived from a certificate or its subject public key.
- Matching Type, which specifies what Certificate Association Data presents, which can be the original data, its SHA-256 hash, or its SHA-512 hash.
- Certificate Association Data, which contains the full data or a digest of a certificate or its public key.

At first glance, it may seem that PKIX-TA or PKIX-EE would be more secure as they require additional PKIX validation; in fact, they only provide illusory incremental security over DANE-TA or DANE-EE. If attackers can compromise the integrity of DNSSEC, PKIX-TA or PKIX-EE can be easily replaced by forged TLSA records containing DANE-TA or DANE-EE, so that any added PKIX verification can be bypassed. Moreover, they are *even more brittle* in SMTP with STARTTLS since the TLS client and TLS server need to have a list of mutually trusted CA and TLS servers, which still relies on trusted third parties (i.e., CAs) to manage their certificates. Thus, the DANE operational practice *recommends* to avoid using PKIX-EE and PKIX-TA [28].

DANE and DNSSEC A TLS client may be vulnerable to man-in-the-middle (MITM) attacks if it cannot verify the server's certificate that binds a public key to the server's identity such as the domain name of the mail or web server. In an email protocol, however, the name of the email server is not usually encoded in the recipient address; instead, the client obtains the server name through an MX record lookup[1].

To leverage DANE, the client has to obtain TLSA records to verify the presented certificate from a TLS server. However, if there is no security guarantee that the fetched DNS records (including TLSA) are not authentic, the client can be vulnerable to active attacks such as MITM and DNS cache poisoning. Thus, a client who wishes to rely on DANE *must* use DNS resolvers that support DNSSEC, or it needs to look up and authenticate the DNS records using DNSSEC by itself.

DANE and SMTP Email service providers use SMTP (as TLS clients) to send emails to destination mail servers (i.e., TLS servers). However, SMTP has no built-in security mechanisms such as authenticating recipients or encrypting messages in transit. To overcome this limitation, an SMTP extension called STARTTLS was introduced in 2002 to encrypt the messages within a TLS session [37]. However, unlike other TLS protocols, such as HTTPS that signals TLS support explicitly through the URI scheme (e.g., https://), an email address itself cannot indicate any transport security policy. Thus, STARTTLS supports *opportunistic TLS*; a client can send a plain-text command, "STARTTLS", to express its TLS support at the initial stage of the SMTP connection. Unfortunately, STARTTLS is well-known to be vulnerable to downgrade attacks, in which a man-in-the-middle may strip out the STARTTLS command. Even worse, the STARTTLS RFC [37] does not specify what to do when the certificate presented by the TLS server is not valid, thus making many TLS clients ignore mismatches between MX records and the domain names in the certificates or continue email transmissions even with invalid certificates (e.g., self-signed certificates) [30].

With DANE, however, the destination mail server can explicitly tell the clients through TLSA records that (1) it supports TLS for secure email transmissions, (2) the presented certificate will be exactly matched with the TLSA records, and (3) the TLSA records are not forged by providing their RRSIGs, DNSKEYs, and DS records.

Figure 1 briefly illustrates how an SMTP client can use DNSSEC to verify the integrity and authenticity of the fetched TLSA records and validate the certificates.

[1] For example, a domain name (of the email server) mapped to a recipient address of user@gmail.com is gmail-smtp-in.l.google.com, which is specified in the MX record.

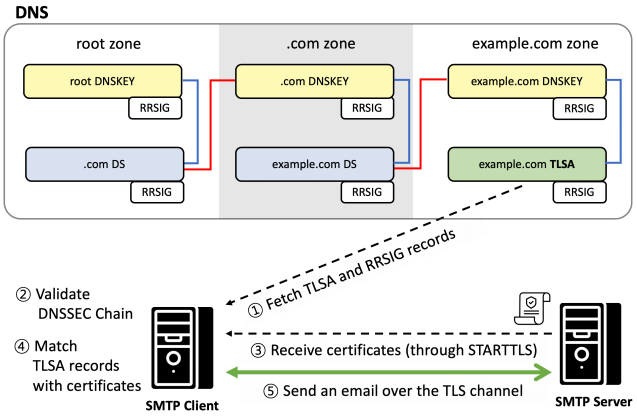

Figure 1: Overview of how DANE works along with DNSSEC and STARTTLS. The integrity and authenticity of `TLSA` records are supported through DNSSEC chain validation; Each `RRSIG` is the signature of a record set (e.g., . `TLSA` records) verified with a `DNSKEY` (blue lines) and each `DS` record is uploaded by a child zone (red lines). After DNSSEC chain verification, the SMTP client verifies the obtained certificate by matching with it to a `TLSA` record.

3 Related Work

In this section, we discuss related work concerning studies of the DANE ecosystem and security protocols for email.

DANE Deployment Liang et al. [83] studied the early stages of DANE deployment in 2014. They specifically focused on the very early stage of DANE usage for the HTTPS, SMTP, and XMPP protocols. Liang et al. found fewer than 1,000 `TLSA` records in 485K signed zones, of which 13% were invalid, which indicated that DANE usage was very rare.

There have been many attempts to deploy DANE to Web PKI in browsers [47, 72]; however, due to some problems like middleboxes blocking `TLSA` records lookup, these were abandoned. Recently, a new TLS extension [56] proposes to allow a web server to deliver its DANE records and its DNSSEC authentication chain during TLS handshakes. This extension, however, has not been standardized yet.

Dukhovni et al. publish DANE deployment statistics periodically [34, 77]; they recently found that 1.4M domains publish signed `MX` records that have `TLSA` records. Web-based debugging tools such as DANE SMTP Validator [32] and DANECheck [25] can help administrators verify correct DANE deployment.

Our study extends these prior studies in three ways.

First, we examine **all** `TLSA` records in three of the largest gTLDs and two ccTLDs with the highest DNSSEC deployment rates for 24 months to investigate the status of DANE deployment longitudinally. *Second*, we primarily focus on how recent incentives for DANE deployment [13, 19, 59] have impacted on the dynamics of DANE ecosystem; this is in contrast to the earlier work in 2014 [83] that focused on the very early stage of DANE deployment where nobody relied on DANE production systems. Since then, there have been multiple incentives introduced by national governments [13, 19] and TLD registries [59] to spur greater adoption of DANE, *which completely changed the landscape of DANE*; for example, the German and Dutch national government guidelines for secure emails state that DANE is mandatory for government bodies and on the comply-or-explain list for public tenders [13, 19] and we confirm a 1,400-fold and 3,100-fold increase of DANE usages in `.com` and `.net` domains compared to earlier work [83], which we detail in the following section. *Third*, we examine DANE deployment more comprehensively including `TLSA` validation against their corresponding certificates and (mis)configurations of the related entities (e.g., SMTP servers and clients) to study the complete DANE ecosystem in email.

Email Security SMTP has long been fraught with security issues such as sender spoofing [36, 66]. To address these problems, there have been many SMTP extensions such as DomainKeys Identified Mail (DKIM) [20], the Sender Policy Framework (SPF) [45] and Domain-based Message Authentication, Reporting, and Conformance (DMARC) [44]. Their purposes are mainly to authenticate a sender and verify the integrity of received emails, but not to encrypt email transport. Studies have focused on how many email servers support those extensions [30] or how popular email service providers actually behave [36]. To encrypt emails, STARTTLS [37] was introduced in 2002 and several studies focused on the deployment of STARTTLS [30,35,57,74]. For example, Foster et al. [35] showed that 89% of popular email service providers deployed STARTTLS. Similarly, Rijs et al. [69] also showed that 60.3% of 116 scanned domains mainly from the Netherlands support STARTTLS. However, STARTTLS was originally designed to protect messages from passive eavesdroppers, thus one of the remaining challenges was the lack of an authentication mechanism of receiver mail servers. Durumeric et al. [30] showed that 52% of SMTP servers in Alexa 1M domains presented trusted certificates, and 34.2% of their `Common Name` values are consistent with the ones in their `MX` records.

Recently, MTA-STS was proposed to authenticate email servers and resist SMTP downgrade attacks [53]. Even though MTA-STS is simple to deploy with a `TXT` record, it does not provide any security guarantee for certificates and the integrity of the record (e.g., MITM attack can take place by simply dropping the `TXT` record). Also, MTA-STS relies on trust-on-first-use (ToFU) and policy caching. Thus, the initial SMTP connection is trusted without authentication of the receiving mail server [53].

4 DANE Deployment

We study the DANE PKI in email applications with a focus on its deployment by analyzing how email servers configure their

TLD	Measurement Period	MX Records	
		Number	Percent with TLSA
.com	2017-10-22 – 2019-10-31	72,981,465	0.7%
.net	2017-10-22 – 2019-10-31	7,440,488	7.3%
.org	2017-10-22 – 2019-10-31	6,112,057	7.0%
.nl	2017-10-22 – 2019-10-31	4,369,343	9.8%
.se	2017-10-22 – 2019-10-31	860,413	38.2%

Table 1: Overview of the Daily datasets for this study. The number and percentage of the domains that have TLSA records are as-of October 31, 2019.

MX records and the corresponding TLSA records. In particular, we carry out a longitudinal study to see how the email servers have changed their MX and TLSA records over time. Let us first introduce the datasets of our study.

4.1 Datasets

Our goal in this section is to conduct a large scale and longitudinal measurement study of DANE deployment in the email ecosystem by focusing on their authoritative DNS servers.

Daily Scans: MX and TLSA records We utilise data from the OpenINTEL [60, 80] measurement platform that fetches DNS records for all registered domains in many TLDs, currently covering around 65% of the global name space. For our study, we select the data for three generic TLDs (.com, .net and .org) and two country code TLDs (.nl and .se); we find that there are 178M resolvable domains in the dataset for these TLDs. We choose the .com, .net, and .org TLDs because they are the three largest TLDs, and .nl and .se because these countries show high rates of the DNSSEC deployment [33], which is essential for DANE. For each domain, we first extract SOA and DNSKEY records with the corresponding RRSIG records, and MX records. After that, we construct a domain name to query TLSA records based on each MX record[2]. The daily snapshots were fetched for 24 months between October 22, 2017 and October 31, 2019. Table 1 summarizes this dataset.

Taken together, the daily scans represent one of the most comprehensive datasets of DANE observations.

4.2 DANE prevalence

We begin by examining how DANE has been deployed by email servers by focusing on the number of second-level domains that serve at least one TLSA record for their MX records. Figure 2 plots the fractions of .com, .net, .org, .nl, and .se second-level domains that publish at least one TLSA record for their MX records. We first notice that DANE deployment for MX records is very rare in gTLDs: only between 0.6% (.com) and 0.73% (.org) have TLSA records for their MX records. However, we also make the following observations:

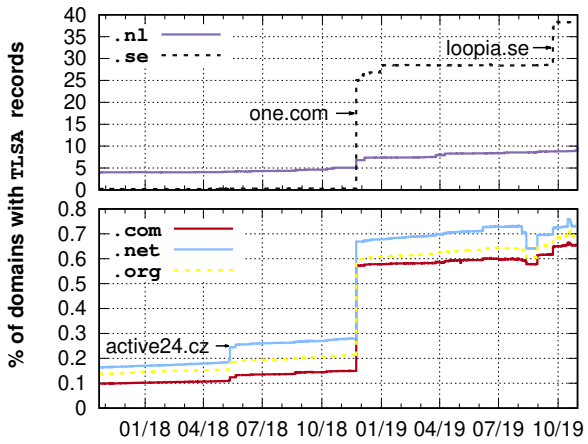

Figure 2: The percentages of domains with MX records in .com, .org, .net, .nl, and .se domains that have TLSA records from the Daily dataset are shown. 0.60% (.com) ∼ 0.73% (.net) of all domains with MX records have corresponding TLSA records in the latest snapshot.

First, we see that the fraction of MX records with TLSA records is steadily growing. For example, the fraction in .com rose from 0.10% in October 2017 to 0.65% in October 2019 showing more than 400K MX records have accompanying TLSA records.

Second, we notice that while the overall DANE deployment rate in the three gTLDs is quite low, the deployment rate is much higher in .nl and .se. Recent studies [23, 49] reported a similar trend for DNSSEC deployment in these two ccTLDs, due to the financial incentives from the registries.

Third, we observe that the growth in DANE deployment is mainly due to the fact that a small number of email service providers provide email hosting services leveraging TLSA records such as one.com and Loopia. That is, we find that the "spikes" we observe in uptake are due to some popular email service providers that provide email hosting services to many domains. For example, the spike on November 23, 2018 was due to a single hosting provider (one.com) publishing a single TLSA record, which impacted 934,066 domains that pointed their MX records to one.com to outsource their email services.[3] Similarly, Loopia (a Swedish service provider) published TLSA records for their MX records, which resulted in DANE deployment for its 76,776 domains instantly in September 2019. However, we are also able to observe drops in August 2019, which were caused by one.com that removed its TLSA records for some MX records making 12,658 .com, .net, and .org domains temporarily forgo their TLSA records.

[2]Because the SMTP protocol can use three possible port numbers (25, 465, and 587), we send three TLSA record requests for each MX record.

[3]This spike is not a coincidence; one of our co-authors presented on DANE to the operator community two days before this spike, and we know from private communication this influenced the decision of one.com to enable DANE.

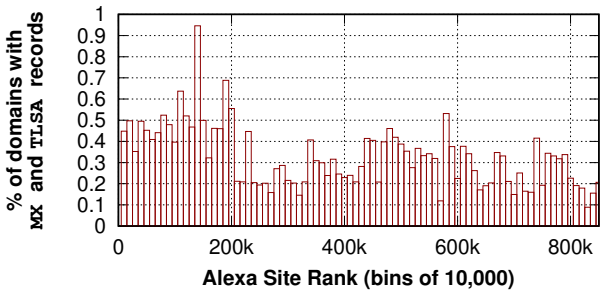

Figure 3: The percentages of domains publishing both MX and TLSA records as a function of website popularity are shown. More popular websites are more inclined to deploy DANE for their email services.

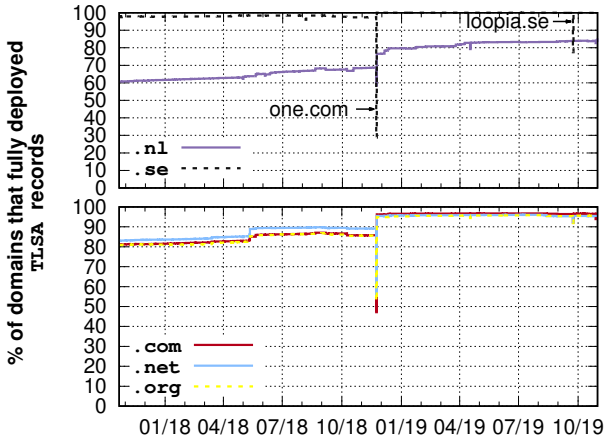

Figure 4: The percentage of domains with at least one TLSA record that also fully deployed TLSA records for all of their MX records.

This changed was reverted in September 2019. We suspect that one.com migrated these domains to other MX records. This observation suggests that email hosting services play a significant role in DANE deployment for SMTP.

Next, we examine whether popular domains are more likely to deploy DANE. Figure 3 shows the percentage of the MX records in the Alexa top 1M domains in .com, .net, .org, .nl, and .se that publish TLSA records, as of October 31, 2019. We first observe that popular websites are more likely to have TLSA records to support DANE, but the overall DANE deployment remains low even among the most popular domains; for example, the average percentage of domains with TLSA records among the top 100,000 popular domains is 0.45%, while that of the bottom 100,000 popular domains is 0.21%. However, we cannot know if all of these domains correctly deployed DANE only by analyzing TLSA records. We have to check (1) if their TLSA records are correctly signed, (2) if they support STARTTLS to present their certificates, and (3) if the certificates are consistent with the corresponding TLSA records. Thus, we perform a more detailed examination of whether they operate DANE correctly in the following sections.

4.3 Security considerations

We began by focusing on the second-level domains that serve at least one TLSA record for their MX records. However, given that domains can serve multiple MX records for better availability, it is ideal to deploy TLSA records for *all of their MX records* to stop active attackers who intentionally attempt to disrupt an SMTP connection to the mail servers with TLSA records and steer a victim SMTP client towards the mail servers that are not equipped with TLSA records.

We now try to understand if domains have *fully* deployed DANE by investigating the number of domains that have deployed TLSA records for all of their MX records. Figure 4 shows the ratio of domains that fully deployed TLSA records and we make a number of observations.

First, we found that a substantial portion of domains from .com, .net, .org, and .nl partially deployed TLSA records; on average 18% of .com, .net, .org and 39% of .nl domains did not fully deploy TLSA records in our oldest snapshot, which implies that these domains were susceptible to downgrade attacks. The fraction of these domains is, however, *steadily decreasing*; for example, in the latest snapshot, we found only less than 5% of .com, .net, and .org domains partially deployed TLSA records and 15% of .nl domains did so. *Second*, we observe that large email providers (one.com and loopia.se) partially deployed their TLSA records first and introduced DANE for all of their MX records a few days later; for example, it took two days for one.com to fully deploy TLSA records. We believe this to be an intentional action to minimize the risk of potential mistakes during the deployment and configuration of the TLSA records.

5 DANE Management

Recall that properly managing DANE for emails means that a domain owner *must* (1) enable DNSSEC correctly by publishing DNSKEY and RRSIG records, and uploading a DS record in the TLD zone, (2) publish a TLSA record, and (3) support STARTTLS and serve a certificate that can be verified using its TLSA record. Thus, we now investigate whether domains with MX and TLSA records take all the necessary steps to support DANE correctly.

5.1 Dataset

Our goal in this section is to understand how domains (i.e., email servers) with MX and TLSA records deploy and operate DANE correctly. The Daily dataset suffices for studying the deployment of TLSA records in the SMTP protocol at a coarse granularity. However, it does not tell us whether the email servers present their certificates, and whether the certificates

Vantage Point	Measurement Period	The number of TLSA	The number of Certs
Oregon	2019-07-11 through 2019-10-31	284,081	252,860
Virginia		283,820	253,070
São Paulo		283,055	251,500
Paris		284,147	253,149
Sydney		282,440	249,608

Table 2: Overview of the Hourly datasets. The number of the collected TLSA records and the certificates are as-of October 31, 2019.

are actually consistent with the TLSA records. Thus, we also collect (1) all the certificates presented by the email servers indicated in the MX records, and (2) the corresponding TLSA records, to observe dynamics at the time scale of hours.

Hourly scans: certificates and TLSA records The following steps detail our methodology to obtain certificates from the mail servers that publish TLSA records.

1. We first obtain all the MX and TLSA records from our Daily dataset, which are updated everyday.
2. We developed a measurement SMTP client that initiates an SMTP connection to an email server (that corresponds to each MX record) through each of the SMTP port numbers (i.e., 25, 465, and 587). The client then sends the STARTTLS command to upgrade the SMTP connection to TLS, and fetches the certificate every hour.
3. We also collect and validate TLSA records in terms of DNSSEC every hour.
4. We deploy the measurement SMTP client in five different vantage points around the world —Oregon (Amazon Web Services [AWS] U.S. West), Virginia (AWS U.S. East), São Paulo (AWS Brazil), Paris (AWS France), and Sydney (AWS Australia)—to comprehensively understand how email servers and their DNS servers behave. All measurement clients are perfectly synchronized to minimize discrepancies between DNS records and certificates across the vantage points.[4]

We used the above methodology to gather measurements by sending on average 11,972 TLSA record lookups as well as collecting the certificate chains every hour from July 11, 2019 to October 31, 2019. We refer to these measurements as the Hourly dataset and Table 2 summarizes this dataset.

Ethical Considerations Our primary ethical concern is to minimize the potential performance risks associated with target email servers by establishing STARTTLS connections every hour. First of all, we have not sent any emails to the email servers. We have only collected the presented certificates from the email servers after sending STARTTLS commands. We also registered a PTR record[5] for each of the five measurement clients, which indicates a domain that runs a webpage explaining the purpose of our measurements and instructions for the DNS and SMTP operators on how to opt out. During the four month measurement period, we received ten opt-out requests and excluded their domains and IP addresses from the measurement.

[4]Measurement completion times may differ depending on the vantage point. The average difference between the fastest and slowest vantage point is only 13.9 seconds. It is possible that two vantage points may fetch two different TLSA records if a rollover occurs exactly between the two scans, but we believe this to be very unlikely.

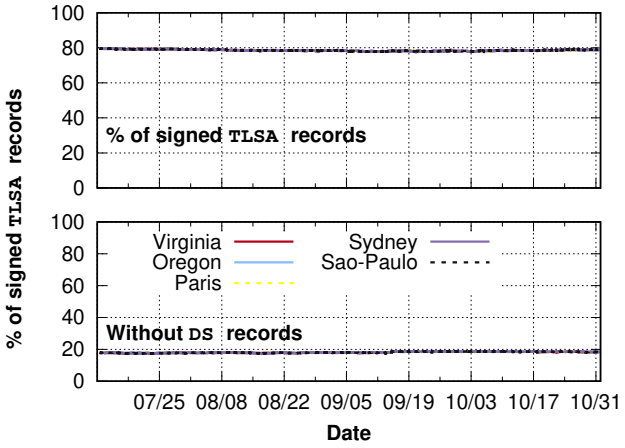

Figure 5: The percentage of signed TLSA records (top) and the percentage of them missing DS records (bottom) from the Hourly dataset are plotted. About 80% of TLSA records are signed, but 20% of them still miss DS records in the latest snapshot.

5.2 Missing Components

We now examine whether domains that publish TLSA records also (1) publish all the necessary DNSSEC records and (2) support STARTTLS.

DNSSEC Recall from section 2 that a domain that publishes TLSA records *must* properly deploy DNSSEC; TLSA records must be signed by its private key, which corresponds to the public key in the DNSKEY record, and have a DS record in the parent DNS zone to create a chain of trust. We first focus on the TLSA records published by a domain that *attempts* to deploy DNSSEC for DANE by generating RRSIGs using their DNSKEYs; consistent with prior work [50,79], we refer to these records as *signed* records.

We begin by examining the percentage of signed TLSA records using the Hourly dataset (top of Figure 5). A key observation is that DNSSEC deployment for TLSA records is pervasive, showing that 80% of TLSA records are signed.

Next, we see how many *signed* TLSA records do not have corresponding DS records; Figure 5 (bottom) shows the percentage of signed TLSA records that cannot be validated due to missing DS records. Interestingly, we observe that 18.5%

[5]This DNS record is used for the reverse DNS lookup; it maps the associated domain or host name for the IP address.

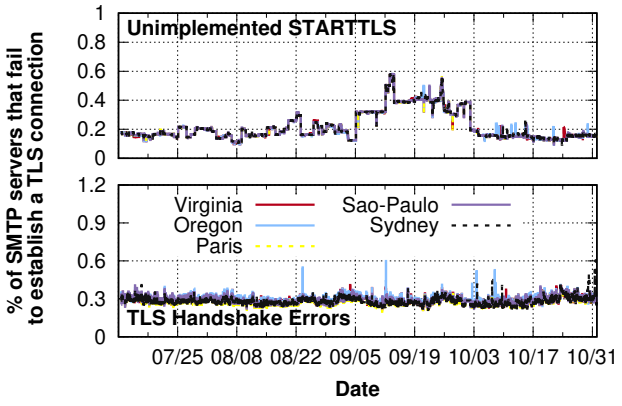

Figure 6: The percentage of the established SMTP connections that fail to initiate TLS connections is shown.

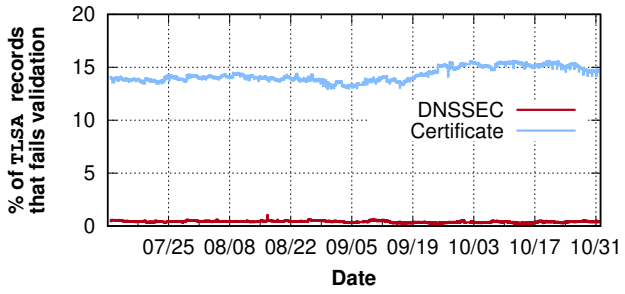

Figure 7: The percentage of TLSA records that are DNSSEC invalid due to (1) wrong DNSSEC configurations such as expired RRSIGs and (2) TLSA records inconsistent with the certificates.

of the signed TLSA records do not have DS records. This is somewhat in line with a recent study [22], which showed that about 30% of signed domains do not upload DS records because of mismanagement by large hosting service providers that provide authoritative DNS servers for their customers. This means that the email servers that use those TLSA records do not profit from any of the security benefits that DANE provides even if they present certificates through STARTTLS, which are consistent with their TLSA records. This is because DANE-supporting email servers for outgoing emails *should not* use TLSA records that cannot be validated through DNSSEC. Installing DS records in the parent zone often requires a manual process where the domain administrator typically needs to contact its registrar. Thus we believe that the CDNSKEY and CDS protocols, which allow the domain owner to directly upload the DS record to the registry could mitigate this overhead [46, 82].

STARTTLS Now, we turn our attention to the email servers running on the MX records. Our goal is to understand how the email servers support STARTTLS to present their certificates. Recall that we set up SMTP clients for testing purposes. For this goal, we first register PTR records for the IP addresses used for the SMTP clients to prevent our connections from being denied by the email servers; in this way, each SMTP client can initiate an SMTP connection for each email server. However, when we attempt to make an SMTP connection, we notice that on average some 20 email servers block our connections in each round. Even though we register PTR records in our DNS server and send not-spam requests to well-known block removal centers such as Spamhaus [73], some email servers still do not allow us to initiate SMTP connections because of their custom block lists. SMTP error codes explicitly show us that our connections are rejected due to their spam filters. We identify the STARTTLS related errors by pairing the error codes and messages such as 500 indicating that the email server does not understand the command, or 502 indicating that the (STARTTLS) command is not implemented.

We also consider errors in negotiating TLS connections such as malformed certificate structures, handshake failures, and no certificates suggested. Figure 6 plots the fraction of the established SMTP connections for which we cannot negotiate STARTTLS connections; note that on average 0.22% of email servers do not implement STARTTLS, and 0.29% of those supporting STARTTLS provide no or malformed certificates. Taken together, these results show that *the majority of the failure of correct DANE deployment is due to missing DS records rather than absence of STARTTLS support*; the average failure rate of the SMTP servers due to unimplemented STARTTLS is less than 0.2%, while the failure rate due to missing DS records is 20%.

5.3 Incorrect Components

Providing (i) a signed TLSA record and its DS record from the DNS and (ii) certificates via STARTTLS is not sufficient to properly operate DANE. The Certificate Association Data of the TLSA records must be correct and consistent with the certificate presented by the email server.

- DNSSEC validation: We examine the correctness and freshness of the RRSIGs records of TLSA records. To this end, we use Unbound [76] to fetch all the necessary DNS records (e.g., DNSKEYs and DS records and their corresponding RRSIGs), and verify the TLSA records based on the time of the scan. The reasons for the DNSSEC validation failures can be expired RRSIGs, RRSIGs inconsistent with their DNSKEYs, malformed RRSIGs, etc.
- Certificate validation: We also examine if the TLSA records are consistent with the presented certificates. To this end, we build a validation program using the OpenSSL library to verify given certificates based on the Certificate Usage in the TLSA records.[6] The reason for certificate validation failures can be mismatched Certificate Usage, Selector, Matching Type, or Certificate Association Data.

[6] We also used the attime option to have OpenSSL validate the certificates as of the time of the scan.

Figure 7 shows the distribution of the validation failure reasons during our measurement period. We make the following observations:

First, we find that most of the TLSA records configure their DNSSEC properly if they do not miss any related DNSSEC records; the average failure rate is only 0.47%. Compared with the recent study [23] reporting a 0.5% failure rate of RRSIGs of signed domains, this result indicates that TLSA records are managed similarly well. Focusing on the validation failure reason, we find that expired RRSIGs are the primary reason (70% of the failures) and the other 30% are due to non-existent DNSKEYs. *Second*, we find that on average 14.17% of the certificates cannot be validated due to a mismatch with their corresponding TLSA records; 2.7% of these errors are caused by a wrong Selector or Certificate Usage. In other words, we can make them valid simply by changing the option number of the Selector or Certificate Usage. The others (97.3%) are due to Certificate Association Data that does not match with any certificate in the chain presented by the TLS server. One possible explanation is that the administrators forgot to update either TLSA records or certificates when changing their public keys, which we consider in more detail in subsection 5.5.

5.4 Impact of TLSA Validation Failure

As explained in section 4, a popular email server (MX record) can be used by many domains, meaning that the validation failure of a single TLSA record can affect many domains that rely on its MX record. We now combine our Daily and Hourly datasets to analyze how many domains have TLSA records with missing or incorrect DANE components, allowing us to estimate the *impact* of TLSA record validity. Figure 8 shows the percentage of domains that have TLSA records that cannot be validated by sending email clients, classified by their TLDs. As the figure shows, the impact varies across TLDs; for example, only 0.006% of .se domains cannot be validated due to missing or invalid DNSSEC or STARTTLS configurations, while .org domains show a much higher error rate of 1.65%, which is 275 times higher.

Interestingly, we observe only 30 $\sim$ 150 .se domains with incorrect or missing TLSA records. We believe this success in deployment is related to the .se registry's consistent efforts to deploy TLSA records and DNSSEC by offering financial incentives to registrars [23, 51] that deploy these technologies correctly[7]. Surprisingly, for almost 8,200 .nl domains, the TLSA records were invalid for 7 hours on October 19, 2019. This was mainly due to four TLSA records sharing the same second-level domain, mailplatform.eu[8]. From manual inspection, we find that their DNSSEC signatures were not valid due to no DNSKEYs matching the DS record in the parent zone. We suspect that they made a mistake during the update of their DS records and DNSKEYs.

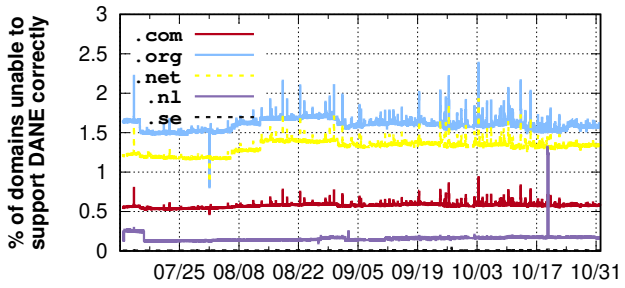

Figure 8: The percentage of domains with misconfigured TLSA records is shown.

5.5 TLSA Management

The previous sections focus on the necessary and correct components to provide valid certificates, which are consistent with the TLSA records. In this subsection, we focus on how TLSA records and the corresponding public keys are managed; more specifically, we investigate if the TLSA records are used as intended and how often public and private key pairs are changed.

Unsuitable Usages The primary purpose of DANE is to let domain owners use custom certificates for their TLS connections by using TLSA records with the DANE-EE or DANE-TA usage without relying on third party CAs. If the domain owner has a certificate issued by a CA, but serves a TLSA record with the DANE-EE or DANE-TA usage, they do not benefit fully from the security measures that DANE provides (instead, they should use the PKIX-EE or PKIX-TA Certificate Usage). Moreover, the validity periods of such certificates are usually determined by CAs, which are usually short.[9] Thus, domain owners incur additional complexity as they need to update their TLSA records whenever the certificates are re-issued. Therefore, a domain name owner *should* avoid setting their TLSA records with the DANE-EE or DANE-TA usage when they serve a certificate issued by a CA.

We first examine how the Certificate Usage field is set in TLSA records by calculating the distribution of the Certificate Usages of the TLSA records from our latest snapshot. Unsurprisingly, we observe that the vast majority of TLSA records (94.29%) use DANE-EE or DANE-TA. We then configure OpenSSL [61] to trust the set of root CA certificates in the Ubuntu 18.04 LTS root store [24]; the validation would fail if the certificates for the TLSA records are custom certificates. Surprisingly, we find that on average 90.58%

[7] Similarly, the .nl registry manages a program called Registrar Scorecard, which offers financial incentives to registrars who enable and manage Internet security protocols such as DKIM and DNSSEC [67, 78].

[8] _25._tcp.antispam.mailplatform.eu, _25._tcp.antispam-alt.mailplatform.eu, _25._tcp.mx-alt.mailplatform.eu, and _25._tcp.mx.mailplatform.eu

[9] The lifetime of the certificates issued by LetsEncrypt is 3 months [52].

and 90.37% of TLSA records with DANE-EE and DANE-TA are still valid, which means that the certificates are valid in terms of PKIX, not custom certificates. Consequently, these records could have used PKIX-EE or PKIX-TA Certificate Usages, thus having the additional benefit of certificate validation through two independent mechanisms (DANE and PKIX). We believe operators do this because they are worried that sending SMTP servers would reject their custom certificates. However, as we will see in the next section, all of the popular email service providers (i.e., sending SMTP servers) that we test do not validate the certificates of the receiving SMTP servers when they cannot find any available TLSA records.

Key Rollover Just like other PKIs, DANE also provides a method for an TLS server to change its public and private key pairs. This process is called key rollover, and the best current practice for executing such a rollover is specified in an RFC [28].

However, unlike other PKIs, DANE requires more careful consideration when performing key rollovers because of old DNS records cached on resolvers. Recall that all DNS responses (including TLSA records) each contain a TTL field indicating how long a given record may be cached. Thus, if an SMTP server simply switches to a new certificate and publishes its corresponding TLSA record immediately, the cached old TLSA records can result in a mismatch to the new certificate, causing a validation failure in some SMTP clients. Thus, before rolling over to a new certificate, the administrator needs to publish a new TLSA record in advance (at least two TTLs of the old TLSA records), while keeping the old one to let the DNS resolvers of SMTP clients fetch the new and old TLSA records together.

We examine how frequently SMTP servers roll their keys, and when they do, if they do this correctly. We only consider changes where the actual public key in the certificate and TLSA record changes. This is relevant because, as discussed earlier, TLSA records have a Matching Type option that specifies how certificates and TLSA records should be matched. If the Matching Type indicates that matches should be performed based on the public key only, the certificate can be renewed while retaining the same key (which extends the validity of the certificate without an actual key rollover).

We first filter certificates and TLSA records that we can monitor for the entire measurement period, which leaves us 10,382 certificates (and their corresponding TLSA records). Among the certificates, we find that 7,334 (70.6%) certificates have never changed their public keys.

We then see whether the other 3,048 certificates have changed their keys *correctly*. To analyze the rollover behaviors more accurately, we remove the TLSA records from our considerations when (1) their TTLs are shorter than our scan resolution (i.e., 1 hour), (2) their corresponding certificates have never been valid[10], and (3) we could not capture their corresponding certificates when the rollover happened due to server or measurement errors. After filtering, this leaves 1,460 (47.9%) TLSA records and their certificates. We make the following observations from our analysis for this dataset:

First, we observe that only 124 domains (8.5%) domains have maintained two or more types of TLSA records with mixed usages such as maintaining DANE-EE and DANE-TA together; this allows administrators to change the leaf certificate and its TLSA records with DANE-EE usage immediately as long as it is signed by the certificate that the TLSA records with DANE-TA usage specify. Due to this advantage, we find that 109 (87.9%) of them successfully roll their keys without any validation failures. *Second*, we find that 1,335 domains (91.4%) have a single TLSA record usage; in this case, the administrators need to make sure that they pre-publish the new TLSA records well in advance of a key rollover. However, we observe that *the vast majority of them (1,257 or 94.2%) experience at least one validation failure during their rollovers.* From further investigation, we observe that 939 of them (74.7%) introduced new certificates and the corresponding TLSA records at the same time *without considering the TTL of the TLSA records* or only introduced new TLSA records after changing certificates.

These results highlight the challenges for correctly updating the keys in two different places in DANE. Considering that authoritative DNS servers and SMTP servers provide two disjoint functions, administrators need to add a new TLSA record on the DNS server *in advance*, and need to install the new certificate in their SMTP server *manually after waiting at least two TTLs.*

6 Client-side DANE Support

Even if domains properly manage their TLSA records with DNSSEC and provide valid certificates that comply with the certificate-related data in TLSA records, an SMTP client cannot be protected unless it looks up and verifies TLSA records correctly. We now examine how DANE is supported in the real world by examining (1) popular email service providers and (2) popular Mail Transfer Agent (MTA) and DNS software.

6.1 Popular Email Service Providers

We first examine how popular email service providers have deployed DANE to authenticate destination mail servers and encrypt email transport. In order to obtain a list of popular email providers, we use the approach from a previous study [36]; we refer to Adobe's leaked user email database from 2013 [43] to rank the email domains based on popularity

[10] In this case, we cannot determine whether they conduct correct rollovers.

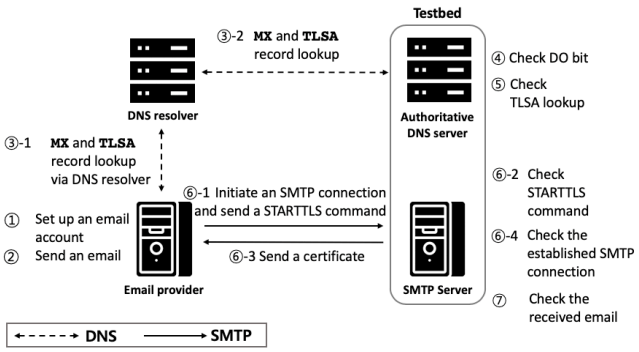

Figure 9: Timeline for measurement of an email provider's DANE support: we sign up for an account and send an email to our testbed server ①∼②; the email provider looks up our domain's `MX` record and `TLSA` record (if it supports DANE) via its DNS resolver or by itself ③; our authoritative DNS server checks if (a) the email provider has tried to look up the `TLSA` record and (b) set the `DO` bit in the header ④∼⑤; the email provider initiates an SMTP connection and sends the STARTTLS command (if it supports STARTTLS). Once the connection is made, the email is transferred ⑥; our testbed SMTP server checks if the email has been successfully delivered ⑦.

and choose the top 25 providers. We also add recent popular email service providers: `protonmail.com`, `tutanota.com`, `zoho.in`, `fastmail.com`, and `runbox.com`. In total, we have 29 popular email service providers that cover 83 million email addresses (54%) in the Adobe database. The list of the email service providers is shown in Table 3. In the following, we describe the details of our measurement methodology.

Experiment Setup

The goal of the experiments is to investigate how popular email service providers, as SMTP clients, properly support DANE. To do so, we first purchase a second-level domain name (e.g., `foo.com`) as an SMTP server in our testbed, which is configured to fully support DNSSEC by uploading `DS` records to its top-level domain, the `.com` zone. We use BIND [2] to run our authoritative DNS server, which has DNS/DNSSEC records for 15 different subdomains. Also, we use Postfix [65] as our SMTP server. We configure the SMTP server to support STARTTLS and enable the Server Name Indication (SNI) [14] extension to serve different certificates for individual subdomain names. Note that the SMTP clients (i.e., 29 email service providers) already support these functions. We test 15 subdomains mapped to different `MX` records; 14 subdomains are configured to test a different combination of DNSSEC, STARTTLS, and DANE misconfigurations, while one subdomain is correctly configured.[11]

[11] To avoid any potential caching issues at intermediate resolvers, we set the TTL values of `MX` and `TLSA` records to one second; however, if some email service providers would happen to send DNS queries to the exact same resolver (e.g., one of the multiple upstream resolvers behind Google DNS), it could ignore our TTL value, which would interfere with our experiment results. To minimize this potential issue, we tested all email service providers at least 5 times over a month to make sure they perform consistently.

We then proceed as follows as illustrated in Figure 9.

1. For each email service provider (e.g., `gmail.com`), we first set up an account as an email sender (e.g., `sender@gmail.com`).
2. For each transmission of an email, we pick one of the 15 testbed subdomains (e.g., `dnssec-invalid-rrsig.foo.com`) to which an email is sent by an email service provider (`sender@gmail.com`).
3. The email service provider first looks up an `MX` record of the testbed subdomain by sending a DNS request to its DNS resolver, which ultimately forwards to our authoritative DNS server. Thus, we can learn the IP address of the resolver on which the email service provider relies.
4. If the incoming DNS request from the resolver does not set the `DO` bit, it indicates that the resolver does not support DNSSEC.
5. As we wish to see whether DANE is enabled in the email service provider (and its DNS resolver), we check if the DNS resolver also makes a DNS request for `TLSA` records.
6. We then check if the email service provider (as an SMTP client) successfully (1) initiates an SMTP connection to our destination email server, and (2) sends the STARTTLS command. If so, our DNS server provides a valid or invalid certificate (depending on the requested subdomain name). In case of an invalid certificate, we observe if the email service provider still continues to establish the TLS connection.
7. Finally, we check if the email has been successfully delivered to our email server. If our email server fails to receive the email sent to a misconfigured test subdomain, it means that the email service provider (and its DNS resolver) has correctly validated the misconfigured subdomain, and decided not to send the email.

Experiment Configurations At first glance, measuring whether an email service provider (i.e., SMTP client) correctly supports DANE seems trivial. We can configure our DNS server to support DNSSEC and to serve `TLSA` records. Also, the destination email server (i.e., SMTP server) is configured to support STARTTLS with a certificate for each subdomain name; note that some certificates are inconsistent with the `Certificate Association Data` values in their corresponding `TLSA` records depending on the misconfiguration settings. Then, the SMTP client will send an email to the SMTP server; we will check whether the email is successfully received. This may be sufficient for studying email service providers at a coarse granularity. However we still would not understand which protocols are (not) supported, or which mechanisms are (in)correctly implemented. To understand the fine-grained behavior of every email service provider, we

have to test each protocol separately by *incorrectly configuring only one* of the DANE-related protocols while keeping the others correctly configured. To this end, we configure our test subdomains and their email servers as follows:

- DNSSEC: The DNS resolver of an email service provider *must* support DNSSEC to check the integrity and authenticity of `TLSA` records. In order to examine whether the DNS resolver validates DNS responses correctly using DNSSEC, we first introduce four different misconfigured subdomains whose `MX` records have missing, incorrect, or expired `RRSIG`s, or missing `DNSKEY`s. Then the email service provider sends an email to each of the four subdomains. We finally check whether the email has been successfully received.
  Typically, SMTP clients (i.e., email service providers) that require DNS lookups outsource DNSSEC validation to their DNS resolvers; DNSSEC-supporting resolvers fetch and validate DNS responses *on behalf of* their clients. If a DNS response is invalid, the DNS resolver returns a `SERVFAIL` response to the SMTP client. Otherwise, it forwards the DNS response to the SMTP client and sets the `Authenticated Data (AD)` bit in the response.
  In some cases, the DNS resolver that an SMTP client uses resides outside its own administrative domain (e.g., it uses a public DNS resolver like Google Public DNS [31]). We examine whether the DNS resolver is managed by a third party such as a public DNS resolver using a WHOIS lookup (e.g., its AS number). The reason we do this is that a man-in-the-middle attacker may interfere in the DNS lookup process towards a resolver outside of the SMTP client's administrative domain. For this reason, the DANE standard strongly recommends against the use of external DNS resolvers ([38], section 8.3).
- STARTTLS: The SMTP client must send the `STARTTLS` command to the destination email server (i.e., SMTP server) to fetch and validate the SMTP server's certificate. Thus, we make the SMTP client authenticate the SMTP server (before sending an email) and check if it sends the `STARTTLS` command after negotiating an SMTP connection with the SMTP server. Our SMTP server presents an invalid certificate, and we will check whether the SMTP client validates it. To this end, the DNS server does not provide the corresponding `TLSA` records. The SMTP server intentionally serves a PKIX-invalid certificate such as an expired or self-signed one, or a certificate whose `Common Name` is not consistent with the one in the `MX` record. Upon receipt of the certificate, the SMTP client either (i) detects the invalid certificate (and the SMTP connection is terminated), or (ii) accepts the invalid certificate without any authentication (thus the SMTP connection is established). Since the STARTTLS RFC [37] does not specify what a client should do for an invalid certificate, it is totally up to the implementation of the SMTP client. We then check whether the email has been successfully received, which means the SMTP client fails to validate certificates.
- DANE: Finally, we investigate whether email service providers have deployed DANE validation and whether they do so correctly. To this end, we introduce four incorrectly configured subdomains; the `TLSA` records of the four subdomains each have a wrong (1) `Certificate Usage`, (2) `Selector`, (3) `Matching Type`, or (4) `Certificate Association Data` that does not match the presented certificate.[12] Before the SMTP client sends the email, we also check (1) if its DNS resolver also has resolved a `TLSA` record from our authoritative DNS server, (2) if it initiates an SMTP connection with the `STARTTLS` command, (3) if it terminates connection after the SMTP server presents a misconfigured certificate, and (4) if it performs the validation of `TLSA` records, and (5) if it detects the `Certificate Association Data` in the `TLSA` record(s) is inconsistent with the SMTP server's certificate.

Experiment results From the experiments, we observe (1) how the email service providers deploy DNSSEC, STARTTLS, and DANE, and (2) if the corresponding protocols are correctly implemented.

First, we observe that 4 out of 29 email providers (`excite.com`, `gmail.com`, and `gmail inbox`, and `outlook.com`) use DNS resolvers that do not support DNSSEC explicitly by sending DNS requests without setting the `DO` bit. Interestingly, we found that `google.com` and `gmail inbox` have tried to fetch `MTA-STS` records [53]; note that `MTA-STS` is an alternative to DANE to authenticate destination email servers. As they cannot check the integrity and authenticity of the `MTA-STS` records, however, they are vulnerable to man-in-the-middle attacks [53], which can manipulate or drop `MTA-STS` lookups or redirect them to a wrong destination mail server. Among the 26 email service providers whose DNS resolvers enable the `DO` bit, *only seven email service providers* fetch `DNSKEY`s and `DS` records. It is a serious issue that the 19 email service providers do not fetch `DNSKEY`s and `DS` records even if the `DO` bit is set. Thus, 23 (i.e., 4 + 19) email service providers are still susceptible to DNS poisoning attacks. This result is in line with the recent study [22], which showed that 82% of the DNS resolvers managed by local ISPs actually do not perform DNSSEC validation. Even more alarmingly, of the seven email service providers that do fetch `DNSKEY`s and `DS` records, we find that three email providers (`mynet.com`, `sapo.pt`, and `sina.com`) explicitly disable DNSSEC validation by setting the `CD` bit. Thus, their resolvers incur the communications overhead for DNSSEC responses including `DNSKEY`s and `DS` records, whose sizes are much larger (by a factor of $6\times \sim 12\times$) than those of DNS (i.e., non-DNSSEC) responses [81], but do not bother to validate the results. Finally, we observe

[12] All other settings such as DNSSEC and STARTTLS are correct.

Mail Provider	DNSSEC					STARTTLS				DANE						
	DO bit	Requested		Valid-ation	Same Op.	Cmd. Sent	Correctly Rejected			TLSA		No Cert	Correctly Rejected Wrong			
		DNSKEY	DS				Expired Cert	Self Signed	CN Unmatch	Pub.	Req.		Usage	Selector	Match	Cert
mail.com	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓	✗	✓	✓	✓
comcast.net	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓
gmx.com	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓
tutanota.com	✓	✓	✓	✓	✗	✓	✗	✗	✗	✓	✓	✓	✗	✓	✓	✓
mynet.com	✓	✓	✓	✗	✗	✓	✗	✗	✗	✗	✗	-	-	-	-	-
sapo.pt	✓	✓	✓	✗	✗	✓	✗	✗	✗	✗	✗	-	-	-	-	-
sina.com	✓	✓	✓	✗	✗	✗	-	-	-	✗	✗	-	-	-	-	-
protonmail.com	✓	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	-	-	-	-	-
aol.com	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	-	-	-	-	-
fastmail.com	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	-	-	-	-	-
freemail.hu	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	-	-	-	-	-
mail.ru	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	-	-	-	-	-
naver.com	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	-	-	-	-	-
rediffmail.com	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	-	-	-	-	-
yahoo.com	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	-	-	-	-	-
zoho.in	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	-	-	-	-	-
daum.net	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	-	-	-	-	-
interia.pl	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	-	-	-	-	-
inbox.lv	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	-	-	-	-	-
icloud.com	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	-	-	-	-	-
runbox.com	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	-	-	-	-	-
seznam.cz	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	-	-	-	-	-
o2.pl	✓	✗	✗	✗	✓	✗	-	-	-	✗	✗	-	-	-	-	-
wp.pl	✓	✗	✗	✗	✓	✗	-	-	-	✗	✗	-	-	-	-	-
sohu.com	✓	✗	✗	✗	✗	✗	-	-	-	✗	✗	-	-	-	-	-
t-online.de	✓	✗	✗	✗	✓	✗	-	-	-	✗	✗	-	-	-	-	-
excite.com	✗	✗	✗	✗	✓	✗	-	-	-	✗	✗	-	-	-	-	-
gmail.com	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	-	-	-	-	-
outlook.com	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	-	-	-	-	-

Table 3: Table showing the top 29 popular email providers' support for DNSSEC, STARTTLS, and DANE; if email providers do not support STARTTLS, we do not test if they accept an expired, self-signed, `Common Name` mismatched certificate (hence the –). Similarly, if they do not fetch `TLSA` records we also do not test if they accept the wrong `TLSA` records (hence the –).

that 9 out of 29 mail service providers use DNS resolvers outside their own network, which makes them vulnerable to man-in-the-middle attacks (Same Op. column in Table 3).

Second, we also observe that 24 out of the 29 mail service providers support STARTTLS; this is in line with a prior study [30], which showed that 81.8% of Alexa 1M domains support STARTTLS. However, we find that *none* of the 24 email service providers correctly verify presented certificates; they successfully complete the TLS handshake even though destination email servers present expired or self-signed certificates, or even certificates whose `Common Name` fields are inconsistent with their corresponding `MX` records. We believe this is due to the lack of specifying what to do in case of invalid certificates in STARTTLS [37]. This result is also in line with prior work [30] that studied the STARTTLS support of popular email service providers; only 52% of popular email servers present *valid certificates*. However, our results suggest that popular email service providers never authenticate the certificates of the counterparts, which strongly motivates the need to deploy DANE for securing incoming and outgoing emails.

Third, we find that only four email service providers (`mail.com`, `comcast.net`, `gmx.com`, `tutanota.com`) actually fetch `TLSA` records. Fortunately, we find that these four email service providers correctly reject `TLSA` records if their `Selector`, `Matching Type`, or `Certificate Association Data` field is not valid. Equally, they also refuse to connect if our test server refuses to initiate a TLS connection (No Cert column). However, we observe that `mail.com`, `tutanota.com` do not check whether the `Certificate Usage` value of the `TLSA` record is consistent with the certificate. That is, we present a self-signed certificate through STARTTLS, but the `TLSA` record sets its `Certificate Usage` to `PKIX-EE`. Given that self-signed certificates can never be PKIX valid, they should have rejected the invalid certificates during the TLS handshake. There are two possible hypotheses for this; they might (1) ignore a `TLSA` record whose `Certificate Usage` is either `PKIX-TA` or `PKIX-EE` (as these usages are not recommended [28]), or (2) skip the PKIX certificate validation except for checking the `Certificate Association Data`. To test our hypothesis, we introduce another subdomain that serves a `TLSA` record with the `PKIX-EE` usage and with a wrong `Certificate Association Data`. Thus, if they ignore the entire `TLSA` record, then our certificate would be accepted and the email would be delivered successfully; if they skip the PKIX validation, the invalid certificate would be rejected, thus the email would not be transmitted. From the additional experiment, we find that the email is not transmitted to this subdomain, implying that two mail servers currently skip the PKIX validation except for checking the `Certificate Association Data`.

6.2 Popular MTAs and DNS software

To deploy DANE in the SMTP protocol at a larger scale, the software of Mail Transfer Agents (MTAs) and DNS resolvers/servers *must* be correctly implemented. If email service providers wish to support DANE, (1) the software of their DNS servers and DNS resolvers must be able to understand TLSA records and to support DNSSEC to validate DNS responses, and (2) their SMTP software must look up and validate TLSA records along with the corresponding certificates. More specifically, sending MTAs (i.e., SMTP clients) must be able to (1) look up TLSA records by themselves, or use their DNS resolvers to look up and validate TLSA records, (2) send the STARTTLS command to receiving MTAs (i.e., SMTP servers), and (3) validate the certificates of the receiving MTAs with the corresponding TLSA records. The receiving MTAs must (1) deploy DNS servers that can serve TLSA records and support DNSSEC to sign their DNS records, and (2) support STARTTLS to present their certificates consistent with the TLSA records.

However, it remains unclear whether the MTA and DNS software achieves the above objectives [39]. In this section, we examine whether the popular MTA and DNS software correctly supports DANE from two perspectives:

- DANE for outgoing emails: Unlike other SMTP extensions that impose responsibilities on receiving MTAs to authenticate sending MTAs (e.g., SPF, DKIM, and DMARC), DANE requires the sending MTAs and their DNS resolvers to execute the following tasks: (1) fetch the receiving MTA's certificate through STARTTLS, and (2) verify the certificates with their TLSA records. Thus, we first examine whether popular SMTP software supports STARTTLS for their outgoing emails, sends TLSA requests, and verifies the fetched certificates. Additionally, we also check whether the SMTP software resolves DNS records by itself (thus an SMTP client becomes a recursive resolver), or relies on DNS resolvers to look up DNS records on its behalf (thus an SMTP client becomes a stub resolver). As discussed in subsection 6.1, if the SMTP client software looks up TLSA records, it is recommended to resolve the DNS records by itself to block man-in-the-middle attacks. For the MTA software leveraging external recursive resolvers, we also check whether the popular DNS software understands TLSA records and supports DNSSEC as a recursive resolver.
- DANE for incoming emails: It is relatively easy to enable DANE for incoming emails. The MTA software needs to enable STARTTLS and its DNS server needs to serve TLSA records that are signed correctly and consistent with the certificate.

Selecting popular MTA and DNS software To obtain a list of popular open source MTA programs, we refer to a prior study that showed four popular MTAs (Exim, Postfix, Sendmail, Exchange[13]), together had a 61% market share in 2015 [30]. To obtain a list of popular open source DNS programs, we refer to prior work [54] that identified DNS software programs running on second-level domains for the .com, .net, .org TLDs. In total, we investigated ten DNS software programs.

MTA Software	SMTP as a Client				SMTP as a Server
	DNS Resolver	START-TLS	TLSA records Req.	TLSA records Valid.	START-TLS
Postfix 3.4.7 [65]	Stub	✓	✓	✓	✓
Exim 4.92.3 [4]	Stub	✓	✓	✓	✓
sendmail 8.15.2 [71]	Stub	✓	✗	-	✓
Exchange Server 2019 [3]	Stub	✓	✗	-	✓

Table 4: Experiment results on four popular SMTP software implementations of their support for STARTTLS and DANE.

DNS Software	DNS		Support	
	Auth.	Recursive	DNSSEC	TLSA
BIND9 9.14.7 [2]	✓	✓	✓	✓
PowerDNS 4.2.0 [9]	✓	✓	✓	✓
Microsoft DNS [7]	✓	✓	✓	✓
Simple DNS Plus 8.0.110 [10]	✓	✓	✓	✓
NSD 4.2.2 [8]	✓	✗	✓	✓
KnotDNS 2.9.0 [5]	✓	✗	✓	✓
YADIFA 2.3.9 [11]	✓	✗	✓	✓
djbdns 1.05 [84]	✓	✓	✗	✗
MaraDNS 3.4.01 [6]	✓	✓	✗	✗
posadis 0.60.6 [63]	✓	✓	✗	✗

Table 5: Experiment results on ten popular DNS software implementations of DNSSEC and DANE (TLSA records). Among them, seven implementations support both protocols correctly.

Results

Our results are summarized in Table 4 and Table 5; we make the following observations. *First*, we note that all of the SMTP programs rely on external recursive resolvers to resolve TLSA records[14]. Considering that a stub resolver can check the authenticity of TLSA records only by the AD bit set by its recursive resolver, a sending MTA may wish to install its own recursive resolver supporting DNSSEC and DANE (Table 5) to verify the DNS records by itself, thereby reducing the attack vectors.

Second, we notice that all of the MTA programs support STARTTLS for both incoming and outgoing emails. However, we find that only Exim and Postfix support DANE.

Third, focusing on the DNS software, we find that seven of the tested DNS programs support DNSSEC. Thus, receiving MTAs (i.e., SMTP servers) that wish to assure the authenticity of their identities and guarantee the confidentiality of email transport, can easily deploy DANE by serving signed TLSA records. However, we find three DNS programs cannot fetch TLSA records yet. Thus the receiving MTAs outsourcing their

[13] This is not open source, but commercial software running on Microsoft Windows Server.

[14] However, we learned that some commercial SMTP programs look up DNS records by themselves such as Cisco's Async OS Email Security Appliance [1]

DNS lookups to those resolvers cannot authenticate sending MTAs even if they use the DANE-supporting MTA software.

6.3 Summary

In summary, DANE support in practice is poor among 29 popular email service providers: only five of them support DANE for incoming emails and four of them support DANE for outgoing emails. Among the four email service providers supporting DANE for both incoming and outgoing emails, one relies on external DNSSEC-aware resolvers, which might be vulnerable to MITM attacks. On the bright side, DANE support in the popular MTA and DNS programs is pervasive; all MTAs support STARTTLS for incoming emails, and two of them validate the presented certificates with their `TLSA` records for outgoing emails. Also, seven DNS programs support both DNSSEC and `TLSA` records; as to the others not supporting DNSSEC and DANE, the latest versions of `djbdns` and `posadis` were released more than 15 years ago [12, 64], and MaraDNS does not support DANE yet despite being updated recently. Thus, we believe that the administrators of those email service providers that do not support DANE yet can easily support DANE by updating and configuring MTA and DNS software.

7 Conclusion

This paper presents a longitudinal and comprehensive study of the DANE ecosystem in SMTP—encompassing 178M second-level domains and 29 popular email service providers to understand the security implications of how DANE is (mis)managed. We found that (1) DANE deployment is scarce but increasing, (2) more than one third of all the `TLSA` records cannot be validated due to missing or incorrect DNSSEC records, and (3) 14% of the certificates are inconsistent with their `TLSA` records. On the SMTP client side, we measured 29 popular email service providers to understand how they support DANE; we found only four of them support DANE for both outgoing and incoming emails, and one email service provider does so only for incoming emails. We also tested four MTA and ten DNS software programs, and found that two of the MTA and seven of the DNS programs support DANE correctly, which implies that the administrators willing to deploy DANE would not find any operational challenges.

Acknowledgments

We thank the anonymous reviewers and our shepherd, Paul Pearce, for their helpful comments. This research was supported in part by NSF grants CNS-1850465 and CNS-1901090, an Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2016-0-00160, Versatile Network System Architecture for Multi-dimensional Diversity), SURFnet Research on Networks and EU H2020 CONCORDIA (#830927).

References

[1] AsyncOSEmailSecurityAppliance. https://www.cisco.com/c/ko_kr/products/security/email-security-appliance/index.html.

[2] BIND9. https://www.isc.org/bind/.

[3] ExchangeServer. https://docs.microsoft.com/ko-kr/Exchange/exchange-server?view=exchserver-2019.

[4] Exim. https://www.exim.org/.

[5] KnotDNS. https://www.knot-dns.cz/.

[6] MaraDNS. https://maradns.samiam.org/.

[7] MicrosoftDNS. https://docs.microsoft.com/ko-kr/windows-server/networking/dns/dns-top.

[8] NSD. https://www.nlnetlabs.nl/projects/nsd/about/.

[9] PowerDNS. https://www.powerdns.com/downloads.html.

[10] SimpleDNSPlus. https://simpledns.com/.

[11] YADIFA. https://www.yadifa.eu/.

[12] djbdns 1.05 Release. https://github.com/abh/djbdns/blob/master/CHANGES.

[13] STARTTLS en DANE. 2016. https://www.forumstandaardisatie.nl/standaard/starttls-en-dane.

[14] D. E. 3rd. Transport Layer Security (TLS) Extensions: Extension Definitions. RFC 6066, IETF, 2011.

[15] C. Arthur. DigiNotar SSL certificate hack amounts to cyberwar, says expert. *The Guardian*. http://www.theguardian.com/technology/2011/sep/05/diginotar-certificate-hack-cyberwar.

[16] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. DNS Security Introduction and Requirements. RFC 4033, IETF, 2005. http://www.ietf.org/rfc/rfc4033.txt.

[17] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Protocol Modifications for the DNS Security Extensions. RFC 4035, IETF, 2005. http://www.ietf.org/rfc/rfc4035.txt.

[18] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Resource Records for the DNS Security Extensions. RFC 4034, IETF, 2005. http://www.ietf.org/rfc/rfc4034.txt.

[19] BSI TR-03108-1: Secure E-Mail Transport. 2016. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03108/TR03108-1.pdf.

[20] D. Crocker, T. Hansen, and M. Kucherawy. DomainKeys Identified Mail (DKIM) Signatures. RFC 6376, IETF, 2011. http://www.ietf.org/rfc/rfc6376.txt.

[21] T. Chung, D. Choffnes, and A. Mislove. Tunneling for Transparency: A Large-Scale Analysis of End-to-End Violations in the Internet. *IMC*, 2016.

[22] T. Chung, R. van Rijswijk-Deij, B. Chandrasekaran, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson. A Longitudinal, End-to-End View of the DNSSEC Ecosystem. *USENIX Security*, 2017.

[23] T. Chung, R. van Rijswijk-Deij, D. Choffnes, A. Mislove, C. Wilson, D. Levin, and B. M. Maggs. Understanding the Role of Registrars in DNSSEC Deployment. *IMC*, 2017.

[24] Certmgr - Mono Certificate Manager. http://manpages.ubuntu.com/manpages/bionic/man1/certmgr.1.html.

[25] Check a DANE TLS Service. https://www.huque.com/bin/danecheck.

[26] Comcast supporting outbound DANE. https://www.internetsociety.org/blog/2017/08/comcast-supporting-outbound-dane/.

[27] V. Dukhovni and W. Hardaker. SMTP Security via Opportunistic DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS). RFC 7672, IETF, 2015.

[28] V. Dukhovni and W. Hardaker. The DNS-Based Authentication of Named Entities (DANE) Protocol: Updates and Operational Guidance. RFC 7671, IETF, 2015.

[29] V. Dukhovni. NEWSFLASH: DANE TLSA records published for web.de. 2016. https://mailarchive.ietf.org/arch/msg/dane/KWMzQLebCeOSgDXhtFAat5NMD60.

[30] Z. Durumeric, D. Adrian, A. Mirian, J. Kasten, E. Bursztein, N. Lidzborski, K. Thomas, V. Eranti, M. Bailey, and J. A. Halderman. Neither Snow Nor Rain Nor MITM ... An Empirical Analysis of Email Delivery Security. *IMC*, 2015.

[31] W. B. De Vries, R. van Rijswijk-Deij, P.-T. de Boer, and A. Pras. Passive Observations of a Large DNS Service: 2.5 Years in the Life of Google. *Network Traffic Measurement and Analysis Conference (TMA)*, 2018.

[32] DANE SMTP Validator. https://dane.sys4.de/.

[33] DNSSEC Deployment Report. https://rick.eng.br/dnssecstat/.

[34] DNSSECDeploymentStatistics. https://stats.dnssec-tools.org/.

[35] I. Foster, J. Larson, M. Masich, A. C. Snoeren, S. Savage, and K. Levchenko. Security by Any Other Name: On the Effectiveness of Provider Based Email Security. *CCS*, 2015.

[36] H. Hu and G. Wang. End-to-End Measurements of Email Spoofing Attacks. *USENIX Security*, 2018.

[37] P. Hoffman. SMTP Service Extension for Secure SMTP over Transport Layer Security. IETF RFC 3207, IEFT, 2002.

[38] P. Hoffman and J. Schlyter. The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA. RFC 6698, IETF, 2012.

[39] S. Huque. Whither DANE? 2019. https://indico.dns-oarc.net/event/31/contributions/707/attachments/682/1125/whither-dane.pdf.

[40] HSTS Preload List. https://opensource.google.com/projects/hstspreload.

[41] J. H. C. Jackson and A. Barth. HTTP Strict Transport Security (HSTS). RFC 6797, IETF, 2012.

[42] D. Kaminsky. It's the End of the Cache as We Know It. Black Hat, 2008. https://www.blackhat.com/presentations/bh-jp-08/bh-jp-08-Kaminsky/BlackHat-Japan-08-Kaminsky-DNS08-BlackOps.pdf.

[43] D. Kocieniewski. Adobe Announces Security Breach. *The New York Times*, 2013. https://www.nytimes.com/2013/10/04/technology/adobe-announces-security-breach.html.

[44] M. Kucherawy and E. Zwicky. Domain-based Message Authentication, Reporting, and Conformance (DMARC). RFC 7489, IETF, 2015. https://tools.ietf.org/html/rfc7489.

[45] S. Kitterman. Sender Policy Framework (SPF) for Authorizing Use of Domains in Email. RFC 7208, IETF, 2014. https://tools.ietf.org/html/rfc7208.

[46] W. Kumari, O. Gudmundsson, and G. Barwood. Automating DNSSEC Delegation Trust Maintenance. RFC 7344, IETF, 2014.

[47] A. Langley. Why not DANE in browsers. 2015. https://www.imperialviolet.org/2015/01/17/notdane.html.

[48] B. Laurie, A. Langley, and E. Kasper. Certificate Transparency. RFC 6962, IETF, 2013. http://www.ietf.org/rfc/rfc6962.txt.

[49] T. Le, R. V. Rijswijk-Deij, L. Allodi, and N. Zannone. Economic Incentives on DNSSEC Deployment: Time to Move from Quantity to Quality. *NOMS*, 2018.

[50] W. Lian, E. Rescorla, H. Shacham, and Stefan. Measuring the Practical Impact of DNSSEC Deployment. *USENIX Security*, 2013.

[51] A.-M. E. Löwinder. DNSSEC Deployment in Sweden: How Do We Do It? ICANN50, 2014. https://london50.icann.org/en/schedule/wed-dnssec/presentation-dnssec-deployment-sweden-25jun14-en.pdf.

[52] Let's Encrypt. https://letsencrypt.org.

[53] D. Margolis, M. Risher, G. Inc., B. Ramakrishnan, O. Inc., A. Brotman, C. Inc., J. Jones, and M. Inc. SMTP MTA Strict Transport Security (MTA-STS). IETF, 2018.

[54] D. Moore. DNS server survey. http://mydns.bboy.net/survey/.

[55] P. Mockapetris. Domain Names - Concepts and Facilities. RFC 1034, IETF, 1987.

[56] M.Shore, R. Barnes, S. Huque, and W. Toorop. A DANE Record and DNSSEC Authentication Chain Extension for TLS draft-ietf-tls-dnssec-chain-extension-07. IETF, 2018.

[57] Massive growth in SMTP STARTTLS deployment. https://www.facebook.com/notes/protect-the-graph/massive-growth-in-smtp-starttls-deployment/1491049534468526.

[58] Mozilla piles on China's SSL cert overlord: We don't trust you either. http://bit.ly/1GBPwfG.

[59] New incentives for security standards DNSSEC and DANE. 2019. https://www.sidn.nl/en/news-and-blogs/new-incentives-for-security-standards-dnssec-and-dane.

[60] OpenINTEL. https://www.openintel.nl/.

[61] OpenSSL. https://www.openssl.org/.

[62] I. Petrov, D. Peskov, G. Coard, T. Chung, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson. Measuring the Rapid Growth of HSTS and HPKP Deployments. University of Maryland, 2017. http://www.cs.umd.edu/content/measuring-rapid-growth-hsts-and-hpkp-deployments.

[63] Posadis. http://posadis.sourceforge.net/.

[64] Posadis 0.60.6 Release. http://posadis.sourceforge.net/release/041225.

[65] Postfix. http://www.postfix.org/.

[66] Z. Ramzan and C. Wuest. Email Spoofing Attack statistics. *CEAS*, 2007.

[67] Registrar Scorecard yields great results. 2019. https://www.sidn.nl/en/news-and-blogs/registrar-scorecard-yields-great-results.

[68] Q. Scheitle, T. Chung, J. Hiller, O. Gasser, J. Naab, R. van Rijswijk-Deij, O. Hohlfeld, R. Holz, D. Choffnes, A. Mislove, and G. Carle. A First Look at Certification Authority Authorization (CAA). *CCR*, 48(2), 2018.

[69] R. Sean and M. van der Meer. The state of StartTLS. 2014. https://caldav.os3.nl/_media/2013-2014/courses/ot/magiel_sean2.pdf.

[70] S. Son and V. Shmatikov. The hitchhiker's guide to DNS cache poisoning. *Security and Privacy in Communication Networks*, Springer, 2010.

[71] Sendmail. https://www.proofpoint.com/us/open-source-email-solution.

[72] Support for DNSSEC/DANE/TLSA validation. https://bugzilla.mozilla.org/show_bug.cgi?id=1479423.

[73] The Spamhaus Project. https://www.spamhaus.org/.

[74] The current state of SMTP STARTTLS deployment. https://www.facebook.com/notes/protect-the-graph/the-current-state-of-smtp-starttls-deployment/1453015901605223/.

[75] Trustwave to escape 'death penalty' for SSL skeleton key. 2012. http://bit.ly/1RbPlNe.

[76] Unbound. https://nlnetlabs.nl/projects/unbound/about/.

[77] Update on stats 2019-10. 2019. https://mail.sys4.de/pipermail/dane-users/2019-November/000534.html.

[78] A. Veenman. SIDN extends DNSSEC discount until July 1, 2018. 2014. https://www.ispam.nl/archives/38957/sidn-verlengt-dnssec-kortingsregeling-tot-1-juli-2018/.

[79] N. L. M. van Adrichem, N. Blenn, A. R. Lúa, X. Wang, M. Wasif, F. Fatturrahman, and F. A. Kuipers. A measurement study of DNSSEC misconfigurations. *Sec. Info.*, 4(8), 2015.

[80] R. van Rijswijk-Deij, M. Jonker, A. Sperotto, and A. Pras. A High-Performance, Scalable Infrastructure for Large-Scale Active DNS Measurements. *IEEE Journal on Selected Areas in Communications*, 34(6), 2016.

[81] R. van Rijswijk-Deij, A. Sperotto, and A. Pras. DNSSEC and Its Potential for DDoS Attacks (A Comprehensive Measurement Study). *IMC*, 2014.

[82] P. Wouters and O. Gudmundsson. Managing DS Records from the Parent via CDS/CDNSKEY. RFC 8078, IETF, 2017.

[83] L. Zhu, D. Wessels, A. Mankin, and J. Heidemann. Measuring DANE TLSA Deployment. *TMA*, 2015.

[84] djbdns. http://cr.yp.to/djbdns.html.

A Terminology

In this section, we provide a glossary of terms and their definitions.

Simple Mail Transfer Protocol (SMTP) is a protocol for internet electonic mail transmission. Mail servers (or Mail Transfer Agent) use SMTP to send and receive emails.

`MX` records is a DNS record to specify which mail servers are willing to act as a mail exchange for the associated domain.

Mail Transfer Agent (MTA) is a software that transfers email messages; it receives incoming emails from sources and delivers outgoing emails to their destinations.

Domain Name System (DNS) is a hierarchical and decentralized naming system for computers or other resources connected to the Internet. It associates various resources (e.g., IP addresses) with domain names.

Top-Level Domains (TLDs) are domains under the root zone in DNS. A second-level domain name comes after the dot such as `.com` and `.se`.

Country Code Top-Level Domain (ccTLD) is one of the categories of TLD, which is reserved for a country or territory identified with a country code such as `.se`, `.nl`.

Generic Top-Level Domain (gTLD) is one of the categories of TLD, which is not country-specific but paired with different classes or organizations such as `.com`, `.net`.