Under the Hood of DANE Mismanagement in SMTP

Hyeonmin Lee
Seoul National University

Md. Ishtiaq Ashiq
Virginia Tech

Moritz Müller
SIDN Labs

Roland van Rijswijk-Deij
University of Twente & NLnet Labs

Taekyoung "Ted" Kwon
Seoul National University

Taejoong Chung
Virginia Tech

Abstract

The DNS-based Authentication of Named Entities (DANE) is an Internet security protocol that enables a TLS connection *without relying on trusted third parties like CAs* by introducing a new DNS record type, `TLSA`. DANE leverages DNSSEC PKI to provide the integrity and authenticity of `TLSA` records. As DANE can solve security challenges in SMTP, such as STARTTLS downgrade attacks and receiver authentication, it has been increasingly deployed surpassing more than 1 M domains with SMTP servers that have `TLSA` records. A recent study, however, reported that there are prevalent misconfigurations on DANE SMTP servers, which hinders DANE from being proliferated.

In this paper, we investigate the reasons *why* it is hard to deploy and manage DANE correctly. Our study uses large-scale, longitudinal measurements to study DANE adoption and management, coupled with a survey of DANE operators, some of which serve more than 100 K domains. Overall, we find that keeping the `TLSA` records from a name server and certificates from an SMTP server synchronized is not straightforward even when the same entity manages the two servers. Furthermore, many of the certificates are configured to be reissued automatically, which may result in invalid `TLSA` records. From surveying 39 mail server operators, we also learn that the majority keeps using CA-issued certificates, despite this no longer being required with DANE, since they are worried about their certificates not being trusted by clients that have not deployed DANE. Having identified several operational challenges for correct DANE management, we release automated tools and shed light on unsolved challenges.

1 Introduction

With Public Key Infrastructure (PKI), Transport Layer Security (TLS) provides secure channels over the Internet. To this end, typically, Certificate Authorities (CAs) publish certificates, and the certificates are validated hierarchically, from the root to the leaf certificate.

However, the current CA-based PKI model has a fundamental vulnerability. CAs can issue certificates for any domain name, and many CAs exist; we have no choice but to trust that all of them issue certificates appropriately. History shows this trust has been broken a number of times. Several CAs were compromised and mis-issued fraudulent certificates [17, 27]. Some CAs even issued fake certificates intentionally [28, 44, 57]. These incidents shook the faith in the PKI model. Several protocols [25, 32, 35] propose mitigations for these problems. However, none of these solutions eliminate the root causes; the public CA model still allows any CA to issue a certificate for any domain name.[1]

The DNS-based Authentication of Named Entities (DANE) protocol [18, 30] was proposed in 2012 to augment or replace the use of trusted public CAs. The key idea of DANE is to leverage the Domain Name System (DNS). To use DANE, a domain owner can publish his TLS server's certificate (or public key) as a DNS record, called a `TLSA` record, to his DNS server. This `TLSA` record must be signed by the DNS Security Extensions (DNSSEC) [4–6] to guarantee its integrity. Since only a domain owner can manage DNS records of its domain, publishing `TLSA` records binds the domain and the certificate (or public key) of its TLS server. Thus, TLS clients can easily authenticate a TLS server by (i) fetching `TLSA` records from the domain's DNS server, (ii) validating their DNSSEC signatures to check the integrity and authenticity, and (iii) checking whether the `TLSA` records are consistent with the certificates from the TLS server, without relying on CAs.

Due to its simple but robust security guarantees, there have been a number of attempts to deploy DANE for numerous web applications such as HTTPS. However, it has never been adopted to validate the certificates of web servers because it introduces additional delays for browsers to fetch DNSSEC and TLSA records. It is also widely known that middleboxes may discard some DNS records such as `TXT` and `RRSIG` [34,

[1] The DNS Certification Authority Authorization (CAA) record [32] allows a domain name owner to specify the CAs authorized to issue certificates for the domain. When the record does not exist, however, it allows all CAs to issue certificates by default.

43], which hinders clients (i.e., browsers) from fetching TLSA records for validation.

Fortunately, DANE has begun to be deployed by email service providers for their SMTP services, because it can effectively solve security challenges that SMTP faces such as STARTTLS downgrade attacks [20] and SMTP service is more tolerant to millisecond-order additional delays.

A recent study [36] showed that popular email service providers such as Comcast and mail.com support DANE for their outgoing mails. Also, the .nl and .se top-level domains show relatively high DANE deployments (9.7% and 38.2% each) compared to .com, .net, and .org (less than 1%). This practice is partially due to the fact that some registries provide financial incentives to domains that deploy DANE [48, 54]. Also, the Dutch and German governments mandate DANE for certified mail service providers in their countries [2, 8].

However, another finding in that study is that server-side misconfigurations often make DANE validations fail in SMTP. First, 15% of SMTP servers that deploy TLSA records are not protected by DNSSEC that is necessary for the integrity of DNS records. Even though TLSA records are signed, 20% of the corresponding DNS servers do not upload DS records to their parent zones (so-called partial deployment). The work also showed the second reason for misconfigurations: many SMTP servers have certificates that are not consistent with the corresponding TLSA records; however, the reason for the inconsistency was not discussed.

In this paper, we present a longitudinal and comprehensive study of DANE in SMTP by observing all related entities needed to correctly operate DANE. We take hourly snapshots of DNS records from all of the second-level domains from .com, .net, .org, and .se for 20 months and collect their certificates. We also interview 39 DANE administrators to understand how they manage DANE and the challenges they face for their management. Coupled with the datasets, we draw a complete picture of the operational challenges for managing DANE.

In this paper, we make the following contributions:

- 99% of domains that outsource their SMTP servers manage DANE correctly. However, the invalid ratio jumps to more than 30% when SMTP servers are self-managed.
- In line with [36], DNSSEC is still a problem; the majority of TLSA records (99%) that experience DNSSEC validation issues are missing DS records. We also find that mismatches between TLSA records and corresponding certificates are prevalent (20%). We discover that many of these mismatched TLSA records (70%) actually match with outdated certificates of SMTP servers, which implies that the mismatches came from incorrect key rollovers.
- Most SMTP servers (87∼92%) incorrectly roll over their keys at least once regardless who manages the SMTP or name server; for example, more than (72∼84%) of SMTP servers change their public keys and corresponding TLSA records without considering the TTL of DNS caches.
- We observe that the current DANE ecosystem still relies on the CA-based PKI model. More than 94% of SMTP servers that deploy TLSA records use CA-issued certificates. We also find that this reliance causes unexpected failures in DANE management when TLSA records are not updated on time due to automatically reissued certificates from CAs.
- The survey of DANE administrators shows the reasoning behind DANE deployment and management: major reasons for DANE deployment are preventing STARTTLS stripping attacks and not trusting CAs; however, we still observe that the majority of domains that deploy DANE *use CA certificates due to the compatibility with other SMTP servers not supporting DANE.*

Our analysis reveals how DANE in the email system is managed and the reasons for mismanagement. On a more positive note, our findings demonstrate several areas of improvement where management of the DANE PKI can be automated and audited. To this end, we publicly release all of our code, datasets and survey answers to the research community at

https://dane-study.github.io

for other administrators and researchers to reproduce and benefit from our work.

2 Background

DNS and DNSSEC DNS associates various information (e.g., A records, MX records) with domains. DANE uses DNS to store the binding information between an identity of an entity and its public key. However, DNS does not provide security in its initial design; the integrity of DNS records is not guaranteed, which makes DNS vulnerable to attacks like DNS spoofing [11, 51]. Thus, the DNS Security Extensions (DNSSEC) [4–6] were proposed to provide the authentication and integrity of DNS records. For this purpose, three new DNS records were introduced:

- DNSKEY records contain public keys used to sign DNS records.
- RRSIG records contain a digital signature of DNS records generated by the private keys corresponding to public keys in DNSKEY records.
- DS records contain a digest of DNSKEY records, which are uploaded to the parent DNS zone to form a chain of trust.

Along with the new records, a domain now has three validation states [5]: (1) secure where a domain is equipped with all cryptographically correct DNSSEC-related records in the above, (2) insecure where a domain is unsigned or does not have a chain of trust (i.e., absence of DS records), and thus cannot be verified regardless of DNSKEYs (or RRSIGs)

records, (3) bogus where a domain has a chain of trust, but its DNS records are cryptographically invalid. A prior study [12] found that missing DS records are a common mistake among many DNS operators by showing that 30% of .com, .org, and .net domains with DNSKEYs do not have the corresponding DS records.

TLSA records DANE uses TLSA records to provide information that can verify the certificate of an application running on the domain. There can be multiple applications running on the same domain with different port numbers, and thus a TLSA record represents a port number, a protocol (i.e., TCP or UDP), and a base domain. For example, to request a TLSA record for an SMTP server of which the MX record is mail.foo.com, the derived domain must be _25._tcp.mail.foo.com. A TLSA record consists of four fields:

- Certificate Usage specifies how to verify certificates (or public keys) from TLS servers (e.g., SMTP server with STARTTLS). There are 4 usages depending on whose certificate is used (TA/EE) and whether PKIX validation is required (PKIX/DANE). The first two usages allow certificates from trusted CAs. Thus, a TLS server must provide a certificate chain that passes PKIX validation using root certificate stores. (i) PKIX-TA (Certificate Usage 0) allows using a root or intermediary CA's certificate. (ii) PKIX-EE (Certificate Usage 1) allows using leaf certificates issued by trusted CAs. In contrast, the next two usages do not require PKIX validation. (iii) DANE-TA (Certificate Usage 2) allows using any certificate of a root or intermediate trust anchor (TA). Thus, a server must provide a certificate chain, including a TA's (possibly self-signed) certificate, which can verify the server's leaf certificate. (iv) DANE-EE (Certificate Usage 3) allows using leaf certificates that can be self-signed; The DANE RFC [18] recommends using DANE-TA and DANE-EE since PKIX CAs offer no additional security for DANE in SMTP.
- Selector specifies whether the entire certificate or only the public key will be selected as Certificate Association Data.
- Matching Type specifies how to represent the selected certificate part. The original value, SHA-256 hash, or SHA-512 hash of the selected data can be used.
- Certificate Association Data contains the processed data depending on the above fields.

SMTP and STARTTLS The Simple Mail Transfer Protocol (SMTP) is a standard for email transmissions. However, SMTP has no security features in its initial design; for example, it sends emails in cleartext (no confidentiality). The STARTTLS extension [29] was proposed to transfer emails securely by using a TLS connection. An SMTP server can send the STARTTLS command in cleartext during the SMTP connection setup to express its TLS support to the client.

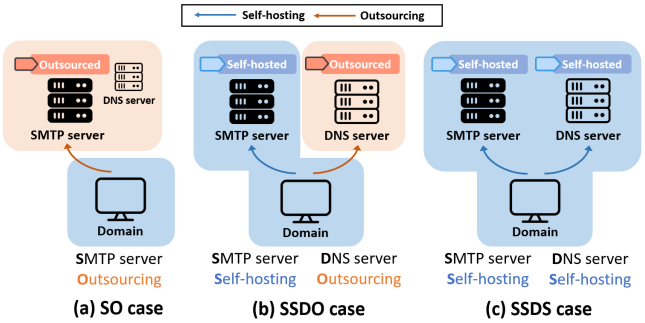

Figure 1: DANE management cases are classified depending on who manages the SMTP server and the name server. (a) SMTP server is outsourced (SO), (b) SMTP server is self-managed but name server is outsourced (SSDO), and (c) both SMTP server and name server are self-managed (SSDS). Note that the name server that serves the TLSA record is also outsourced if the SMTP server is outsourced.

However, this has two main security problems: First, as the STARTTLS command is sent as cleartext, it is vulnerable to downgrade attacks to prevent TLS negotiation by stripping the command [20]. Second, the STARTTLS standard [29] does not define what to do when the STARTTLS certificate is not valid, thus making many TLS clients not even attempt to validate the certificate [20]. With DANE, such downgrade attacks can be mitigated since the presence of TLSA records for an SMTP server are an explicit signal of STARTTLS support.[2]

How to deploy DANE SMTP The key elements for DANE are a TLSA record and its corresponding certificate. For successful DANE deployment, an SMTP server must take three steps. First of all, the base domain that serves TLSA records must have all necessary and correct DNSSEC records, making its validation status secure. Second, it must have a certificate, which is provided through an SMTP connection; the certificate may be self-signed or signed by another signing certificate. Third, it has to generate a TLSA record matched with the certificate; depending on the Certificate Usage, the administrator may want it to be matched with the signing certificate (in case of DANE-TA or PKIX-TA usage) or matched with the leaf certificate (in case of DANE-EE or PKIX-EE usage).

Where to deploy DANE SMTP At first glance, deploying DANE for an SMTP server seems straightforward because what the domain owner needs to do is to keep its certificate and TLSA record consistent.

However, it can become a bit tricky when the certificate and its corresponding TLSA record are managed by *two different entities*; domain owners may run both SMTP servers and

[2]Note that a domain serving TLSA records has to be DNSSEC-signed to support DANE. Thus, when TLSA records are not available in a given domain, the proof of non-existence such as NSEC and NSEC3 must be provided. This makes it impossible for man-in-the-middle attackers to simply drop the TLSA records for downgrade attacks.

name servers by themselves, but they can also choose to outsource their management to a popular email hosting provider (for their SMTP servers) and/or an external DNS operator (for their name servers). Thus, in practice, a domain owner has three options to deploy and manage DANE as illustrated in Figure 1.

First, a domain owner (e.g., example.com) may choose a popular email hosting provider (e.g., one.com) to outsource the SMTP server (labeled as SO). Typically, this is done by serving an MX record that delegates to an email hosting provider (e.g., mx1.one.com) such as

```
mx.example.com 600 IN MX 10 mx1.one.com.
```

Even though the MX record is served from the name server that the base domain name uses, the TLSA record will be fetched from the name server managed by the hosting provider because a TLSA record is bound to an MX record. Thus, when a domain outsources its email services to a DANE-enabled SMTP hosting provider, the provider manages both the certificate and its TLSA record. Thus, choosing a popular email hosting provider that can deploy TLSA records is an easy and effective way to support DANE. However, domain owners lose control over managing DANE because they neither manage a certificate nor TLSA records.

Second, domain owners may run and manage their SMTP servers by themselves. For their name servers, they can either (1) outsource to a popular DNS operator such as Cloudflare or their registrar's default name server (labeled as SSDO) or (2) manage it by themselves (labeled as SSDS). In the first case, *a domain owner has the responsibility to give the outsourcing DNS operator the correct TLSA record, which matches with the STARTTLS certificate that is used to encrypt the SMTP connection*. Thus, it might be problematic if the domain owners are not familiar with how to generate TLSA records (when they update their certificates) or how to give the generated TLSA records to their outsourcing DNS operators since mismatched TLSA records result in DANE validation failures. Furthermore, a recent study [13] showed that not all registrars support DNSSEC when the domain owners themselves are the DNS operator, making it impossible to deploy DANE due to missing DS records.

Why DANE validation fails In general, validating a TLSA record fails due to two reasons:

- insecure or bogus DNSSEC: DANE validation mandates correct DNSSEC deployment. When a DANE-validating client finds a TLSA record to be insecure, it ignores the TLSA record and concludes that the SMTP server does not support DANE, which brings all of the STARTTLS vulnerabilities back.[3] When the TLSA record is determined to be bogus, the client is expected to abort the SMTP connection immediately.
- Mismatched TLSA records: this happens when the certificate and its corresponding TLSA record do not match. A previous study [36] found that about 4% of TLSA records could not be validated due to such a mismatch, but the root causes were not investigated, which motivates this paper.

3 Related Work

In this section, we discuss related studies about security protocols for SMTP encryption and the DANE ecosystem.

SMTP encryption SMTP does not encrypt its messages itself. Thus, the STARTTLS extension [29] was first introduced for email encryption. Several studies [20, 26, 31, 56] reported that STARTTLS is widely deployed. However, they also found widespread mismanagement of STARTTLS; 70% of the collected STARTTLS certificates cannot be authenticated due to misconfigurations. This is somewhat expected because STARTTLS does not specify what to do for invalid STARTTLS certificates. Recently, Poddebniak et al. [50] also revealed security vulnerabilities of STARTTLS such as command injection and credential stealing in SMTP, POP3 [42] and IMAP [10] protocols, which are largely due to additional but vague negotiation processes. To overcome these limitations, MTA-STS was also proposed to authenticate email servers and encrypt email messages [41]. Compared to DANE, it is simpler to deploy MTA-STS by leveraging TXT records. However, it is still vulnerable to MITM attacks as it does not mandate DNSSEC.

Ecosystem of DANE Zhu et al. [59] measured the DANE deployment in 2015 by focusing on SLDs of .com and .net; they found that only 997 domains out of 485k signed domains have TLSA records; 13% of them were invalid. Recently, Lee et al. [36] focused on the deployment of DANE for SLDs with MX records in five TLDs and popular mail service providers in 2020. The paper showed a slow but gradually increasing DANE deployment rate; less than 1% of second-level domains of .com, .net, and .org deployed TLSA records. However, .nl and .se had deployed DANE relatively aggressively due to financial incentives from the registries. Also, they reported that 3.6% of TLSA records were not matched with the corresponding certificates, thus making them invalid. However, they could not find the root causes.

Considering DANE is still in the early stage, some efforts have been made to keep track of its deployment or to provide debugging tools for DANE administrators. For example, the NL registry (SIDN) and Dukhovni et al. publish DANE deployment statistics on websites based on their active scans and present SMTP DANE validation results on a

[3] There is a debate whether we have to regard an insecure TLSA record as an invalid TLSA record or not [19]; as DANE mandates *full* DNSSEC deployment and a validator considers insecure TLSA records *unusable*, we regard it as an invalid TLSA record. For clarity, however, we also provide the details of the invalid reasons of a TLSA record for the rest of the paper whenever we analyze them.

TLD	Measurement Period	Domains		MX records	
		All	Incorrect	All	Incorrect
.com	2019/07/13 ~ 2021/02/12	707,365	0.51%	12,323	22.20%
.net		75,921	1.06%	2,604	20.97%
.org		61,844	1.25%	1,988	18.76%
.se		363,192	0.01%	354	7.91%

Table 1: The number of SMTP servers and domains that have TLSA records, and the percentage of DANE failures of SMTP servers and domains are shown as of February 12th, 2021.

daily basis [52, 53]. Also, there are web-based DANE validation tools [16, 21, 22] that can help administrators debug and configure TLSA records.

Our study extends these prior works in two ways. First, we focus on why there are prevalent cases of mismanagement in SMTP DANE by leveraging the longitudinal datasets collected by our active measurement and the comprehensive survey from email service providers. Second, we find that the mismanagement is mainly due to the lack of automated tools for key management such as key rollovers. Hence, we design and implement a prototype of automated tools for DANE key management and discuss the potential operational challenges in practice.

4 Datasets

In this section, we present the data we collected, and analyze how DANE is deployed and operated.

4.1 DNS records and certificates

Our goal is to understand how DANE has been deployed and how well it is managed.

Daily Scans: DNS records We rely on DNS scans from four TLDs provided by OpenINTEL [49]: the .com, .net, and .org gTLDs and .se ccTLD. We choose three gTLDs (.com, .net, and .org) because they are the largest TLDs, and one ccTLD (.se) as Sweden shows the highest rate of DANE deployment [36]. For each of the four TLDs, OpenINTEL first obtains daily zone files from their registries (.com and .net from Verisign, .org from Public Internet Registry, .se from Internetstiftelsen) to obtain Name Server (NS) and Delegation Signer (DS) records for all second-level domains (SLDs). For each of these SLDs, OpenINTEL also collects DNS records from the authoritative name servers, which include A, MX, TLSA, DNSKEYs and RRSIG records.

Hourly Scans: DNS records and STARTTLS certificates The daily snapshots may be sufficient for understanding DANE behaviors in the SMTP protocol at a coarse granularity, but they have two limitations. *First*, SMTP servers with TLSA records do not necessarily mean that they support DANE correctly; for example, they may not support STARTTLS, may not present certificates during the STARTTLS handshake, or may present certificates that do not match with TLSA records, all of which make DANE validations fail. *Second*, the daily scan cannot capture the dynamics of DNS records at a timescale shorter than one day. We calculate the distribution of the TTL values of TLSA records across the entire daily dataset, and find that 93% of the TTLs of TLSA records are less than 1 day, which indicates that we would not capture their dynamics such as the changes of their certificate and TLSA records if we rely on the daily scan. To overcome these limitations, we collect the second dataset by (1) initiating an SMTP connection using collected MX records through SMTP port number 25, (2) sending the STARTTLS command to upgrade an SMTP connection with TLS, and (3) fetching the certificates every hour. We also collect their TLSA records and DNSSEC-related records every hour, and conduct DANE validation as well. In total, our snapshots span 20 months from July 13th, 2019 to February 12th, 2021, which is summarized in Table 1.

4.2 Overall DANE support

A recent study [36] reported that DANE had been increasingly deployed around 2019 as a few large email hosting providers enabled DANE support. For example, it showed that the percentage of domains with MX records that have TLSA records increased from 0.1% to 0.6% for .com domains and from 0% to 38.2% for .se domains from October 2017 to October 2019.

As our dataset partially overlaps with the one in [36], we can quickly revisit the trend; Table 1 shows the number and the percentage of domains and SMTP servers that deployed DANE. We can confirm the accelerated deployment; for example, the deployment rate increased from 0.6% to 0.97% for .com domains and from 28.41% to 41.57% for .se domains from July 2019 [36] to February 2021. When validating their TLSA records, however, we find that DANE validation failures are widespread across the SMTP servers; for example, the percentage of SMTP servers with invalid TLSA records is over 22% when they serve domains in .com. Fortunately, the percentage of impacted domains is not as high as that of SMTP servers (e.g., 0.51% in .com domains) since the outsourced SMTP operators support DANE without failures.

To understand the potential reasons behind this widespread unsuccessful DANE deployment, we first examine the correlation between the popularity of DANE-enabled SMTP servers in terms of the number of serving domains and their validation status. Figure 2 shows the CDF of the number of domains served by DANE SMTP servers that serve x domains in our latest snapshot. We make two observations.

First, we notice a disparity between DANE-valid and DANE-invalid SMTP servers in terms of the number of domains that each SMTP server serves; the incorrectly config-

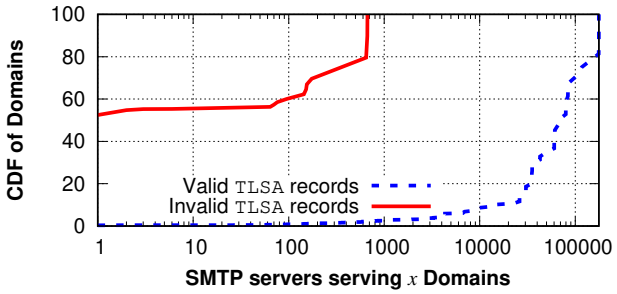

Figure 2: CDF of the number of domains served by DANE SMTP servers for valid and invalid TLSA records as of February 12, 2021.

ured DANE SMTP servers that serve only a single domain name take up 45% of the total domains and the largest incorrectly configured SMTP server serves only 667 domains. On the other hand, when focusing on the DANE-valid SMTP servers, we see that 90% of DANE-valid domains are served by popular SMTP servers that have more than 10,000 domains. It is highly likely that these domain names outsource their email services to popular email hosting providers. For instance, one.com serves more than 600,000 domains.

Recall that a name server responsible for TLSA records and an SMTP server providing the actual certificate for STARTTLS have to be managed consistently to support DANE correctly, this result may suggest that the quality of DANE management could be different depending on the two entities that manage the name server and SMTP server, respectively. Hence, to better understand why and how DANE validation fails, we now turn our attention to examine who manages DNS and SMTP servers for domains that support DANE SMTP.

5 DANE Quality vs. Managing Entity

5.1 Determining Managing Entities

Identifying whether a domain name outsources a name server (for TLSA records) or an SMTP server (for email service over STARTTLS) is not straightforward because the only publicly available information is its DNS records such as MX, NS, TLSA and their A records. One possible approach is to leverage WHOIS, but there are several challenges. First, a domain name, its MX records and NS records can be different from one another when the domain name outsources either the SMTP server or name server or both. Thus, we have to collect all registrar information of each domain name, MX and NS records, but the WHOIS infrastructure is heavily rate-limited and notoriously inconsistent [37]. Moreover, many domains are registered through privacy-preserving services that hide domain registrant information such as email address and name, which makes it challenging to identify whether the domain names (i.e., the RDATA fields in MX and NS records) are owned by the same entity [9].

To overcome these challenges, we apply two techniques. We first focus on the popularity of the MX or NS records of domains. Our high-level intuition is that the MX records or NS records that map many domains such as more than 50 domains are highly likely to be email hosting providers (for MX records) or external DNS operators (for NS records). In general, popular email hosting providers and external DNS operators manage multiple NS records and MX records with the same SLDs, respectively. For instance, the RDATA fields in NS records are mx[1-4].one.com. Thus we first group the SLDs of MX records and NS records, respectively, and check whether each domain name uses popular SMTP or name servers. This reveals that 1,193,961 (96.6%) domains rely on email hosting providers and 1,210,413 (97.9%) domains outsource DNS servers; each of these email hosting providers and outsourced DNS servers serves at least 50 domains. This confirms our findings in Figure 2 that the majority of domains with TLSA records are served by popular SMTP servers.

However, we notice that this finding is not enough to identify which domains outsource their SMTP servers. We find some corner cases where email hosting providers assign a unique MX record to their customers, but each of the MX records is mapped to the same IP address of the SMTP servers managed by the email hosting providers. A prominent example is Antagonist, which assigns a unique MX record to their customers such as mail.foo.com and mail.bar.com for their two customers, foo.com and bar.com. However, we find that *all of their MX records are mapped to the same set of IP addresses from the same set of name servers managed by Antagonist*[4]. This indicates that they outsource their SMTP servers to *Antagonist*. Thus, to prevent them from being misclassified as self-managed domains, we also group the MX records by their resolved IP address, which are classified as outsourced if the number of domains relying on the same IP address is over 50. This gives us an extra 20,707 (1.7%) domains that are found to outsource their SMTP servers.

Identifying self-managed domains is not straightforward because unpopular MX or NS records do not necessarily mean that they manage their own SMTP servers or name servers. To identify them accurately, we focus on the domains that share the same SLD with their MX records (for SMTP self-management) or NS records (for nameserver self-management). In such cases, these domains are highly likely to manage their SMTP and name servers by themselves since their owners are identical. Using this methodology, we find that 6,408 (0.5%) domains self-manage their SMTP servers and 3,365 (0.3%) domains self-manage their DNS servers.

We exclude the rest of the domains—15,052 (1.2%) domains for their MX records and 22,350 (1.8%) domains for their NS records—from further analyses. After that, we classify the domains into three cases (i.e., SO, SSDO, and SSDS)

[4] ns[1-3].webhostingserver.nl, which are Antagonist's DNS authoritative servers.

Category	SMTP servers		Domains	
	Number	Invalid TLSA (%)	Number	Invalid TLSA (%)
SO	9,766	11.51%	1,202,579	0.23%
SSDO	1,786	39.42%	1,792	40.18%
SSDS	2,840	32.29%	2,806	33.14%

Table 2: The numbers of DANE SMTP servers and their domains and the percentages of invalid TLSA records of DANE SMTP servers and their domains are shown in each category as of February 12, 2021. Note that the abbreviated categories are described in Figure 1.

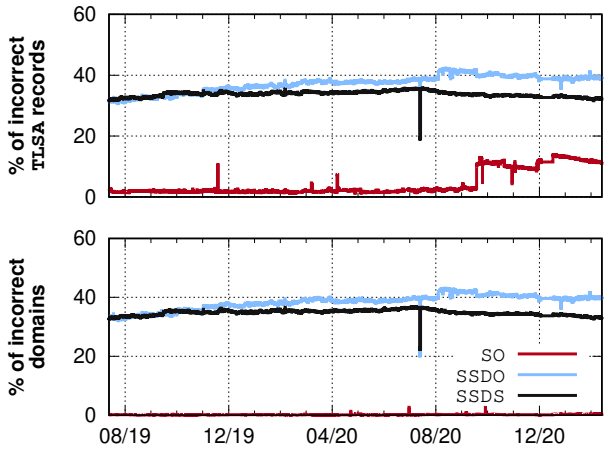

Figure 3: The percentage of incorrect TLSA records (top) and their served domains (bottom) for each category is shown.

based on the criteria of DANE managing entity we discussed in section 2, which is summarized in Table 2.

5.2 Managing entities and DANE quality

5.2.1 Overall DANE Management

We now examine how DANE management has changed over time depending on who manages the SMTP and name servers. Figure 3 shows the percentage of DANE SMTP servers that fail to deploy DANE successfully (top) and the percentage of domains associated with them for each case. Note that the incorrect DANE deployment rate of the self-managed SMTP servers (SSDS and SSDO) is much higher than outsourced SMTP servers (SO). As to the self-managed cases (SSDS and SSDO), we find its percentage of the domains with invalid TLSA records is comparable to that of the (self-managed) SMTP servers with invalid TLSA records since the latter usually serve a very small number of domains. We also see the slightly higher incorrect deployment rate of SSDO *when the name server is also outsourced* compared to SSDS, which will be detailed in the next section. On the other hand, we see only 16 (2.66%) of TLSA records in the SO case are invalid until September, 17th, 2020. However, we see a spike from

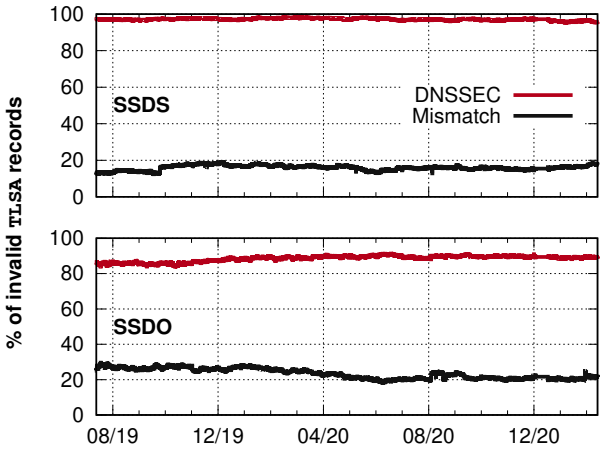

Figure 4: The percentages of TLSA validation failures due to wrong DNSSEC configuration and wrong TLSA records are shown for the self-managed SMTP categories.

September 18th, 2020 ∼ February 12th, 2021 in SO due to the following reasons. Two email hosting providers, Syix and Antagonist assign a unique MX record to each of their customers, which generates an equal number of TLSA records to the number of MX records. In fact, the 63 TLSA records managed by Syix mistakenly share the same Certificate Association Data values, and 1,655 TLSA records managed by Antagonist do the same. Also, we find that only 12 (2%) TLSA records in the SO case are misconfigured on average; we find that only 1,765 domains rely on such invalid TLSA records on average, implying that large email hosting providers are normally well-managed.

These results highlight that *self-managing SMTP servers is more error-prone*. Now, we switch our focus on *where* such mismanagement happens. To this end, we examine the TLSA records, certificates, and DNSSEC records for the misconfigured TLSA records.

5.2.2 Why TLSA Validation Fails

Next, we examine why TLSA validation fails in each case. As discussed in section 2, TLSA record validation fails due to mainly two reasons: (1) unsuccessful DNSSEC deployment and management and (2) mismatches between TLSA records and their certificates. Figure 4 plots the distributions for the SSDO and SSDS cases. Since there are only 12 invalid and unique TLSA records for the SO case, we omit the plot.

DNSSEC We notice that DNSSEC is the dominant reason of DANE management failures across all managing entities; we find that 89% and 95.7% of TLSA records in SSDO and SSDS respectively are invalid due to DNSSEC issues in our latest snapshot. In case of SO, we find that 12 unique TLSA records have DNSSEC problems, which shows that even popular email hosting providers have difficulties in deploying

DNSSEC correctly. This leads us to dig deeper into why their DNSSEC configurations are unsuccessful; we first check if they are equipped with all three DNS record types to support DNSSEC correctly (i.e., DNSKEY, RRSIG, and DS records). We observe that the majority of TLSA records that have DNSSEC problems do not have DS records, making them insecure. In case of SO, for example, 10 out of 12 TLSA records with DNSSEC problems are missing DS records. We find similar patterns in the other two categories as well; 624 (99.5%) and 866 (98.6%) invalid TLSA records are due to missing DS records in SSDO and SSDS. In contrast, we find that TLSA records with bogus state are very few; only 2, 3, and 12 TLSA records cannot be validated due to either cryptographic errors (e.g., signature from unknown keys) or expired RRSIGs in SO, SSDO, and SSDS respectively. Considering that the majority of TLSA records with DNSSEC problems are insecure, we believe that this problem can be mitigated if registrars move towards a standard of DNSSEC-by-default on their name servers by creating a chain of trust automatically. However, Chung et al. [13] pointed out that this is very rare: only one registrar (NameCheap [47]) among the top popular 20 had this policy in 2017.

Mismatches between TLSA records and certificates Interestingly, we find that, on average, 16%, 23% of TLSA records fail in DANE validation due to mismatches when the DANE managing entity is SSDS, and SSDO respectively; we also find only 6 in the SO case. This raises a serious concern since these errors may cause DANE-validating clients to abort SMTP connections regardless of their DNSSEC validation status. Also, such errors do not tend to be fixed over time; they seem to be persistent and go unnoticed by the administrators.

One possible explanation is that the parameters of a TLSA record are incorrectly set by mistakenly specifying Selector, Matching Type, or Certificate Usage even if its Certificate Association Data is generated from the correct certificate. We test this hypothesis by changing each of the parameters in three fields to see if any combination makes the DANE validation successful. Unfortunately, however, we only found that 1 (SO), 2 (SSDO), 3 (SSDS) TLSA records meet the hypothesis across the cases leaving the question still unanswered.[5]

Interestingly, we also notice that TLSA records with Selector 0 show a higher mismatch ratio than the ones with Selector 1. In case of the SSDS category, for example, we find that 11% of TLSA records use Selector 0, but 34% of them are mismatched in our latest snapshot. In contrast, we learn that only 2% of TLSA records with Selector 1 are mismatched. This may suggest that the mismatch could be due to key rollovers; TLSA records with Selector 0 need to be updated even when the reissued certificate still uses the same public key. For Certificate Usage, we find that more than 99% of TLSA records use DANE usages (i.e., DANE-TA and DANE-EE) across all managing categories as expected. In case of Matching Type, we find more than 96% of TLSA records use Matching Type 1 and less than 2% of TLSA records use Matching Type 2 across all managing categories; we find that TLSA records with Matching Type 2 show generally higher mismatch ratios than Matching Type 1; for example, in the SSDS category, we observe that the mismatch ratio of TLSA records with Matching Type 2 is 12.5% while showing 5.5% in Matching Type 2. However, we cannot identify the rationale behind this.

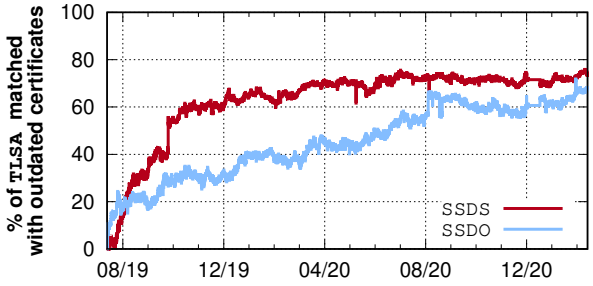

Figure 5: We plot the percentage of mismatched TLSA records at the time of the scan, that match with outdated certificates.

5.2.3 Why Mismatches Happen

The above analysis showed that the mismatches might have nothing to do with a TLSA record and its certificate captured *in the same snapshot*; we now ask whether currently mismatched TLSA records *can be correctly matched* with any of the old (and outdated) certificates that the SMTP server has ever used. This may happen when the administrators simply forget to update TLSA records after their certificates are changed.

To test our hypothesis, we first consider the TLSA records of which mismatch reasons are unknown. Then, for each snapshot, we check if we can find any outdated certificate that has expired at the snapshot but matches with the TLSA record, which is shown in Figure 5. Surprisingly, we observe an increasing trend.

This increasing trend is somewhat expected as we can compare more outdated certificates with the currently mismatched TLSA records over time. However, we find that the percentages reach up to 70% and 73% in SSDS and SSDO respectively, which indicates that *the majority of mismatches between TLSA records and certificates are due to TLSA records that have not been updated timely*.

These results indicate that many SMTP servers have

[5] Astute readers may attribute them to the administrator who calculates a hash value of a given certificate but makes a mistake while inserting it into Certificate Association Data by missing a few characters (i.e., copy & paste error). To verify this hypothesis, we also compared the edit distance of two strings (i.e., the actual Certificate Association Data on the certificate and the correct hash value calculated from the matched certificate) using the Levenshtein algorithm [38], but we could not find any single case.

updated their certificates as well as the public keys (i.e., rollovers), but failed to update the corresponding TLSA records. This is interesting because (1) they could have decided not to roll over since most of TLSA usages we measured are DANE-EE (90.6%), which allows us to use the same certificate because DANE-EE usage ignores the expiration date in a certificate in the validation process [18] and (2) it implies that conducting a rollover correctly is challenging. In the following section, we aim to answer both questions.

6 DANE Key Rollover

Like the PKI, DANE provides a method for entities to update their public and private key pairs. This process is called a key *rollover*, which is standardized in DANE RFCs [18]. However, the above analysis implies that performing key rollovers correctly is not easy. We now ask whether DANE-enabled SMTP servers perform key rollovers correctly to understand the possible challenges.

6.1 Determining SMTP servers that roll over

First, we examine how many SMTP servers have conducted rollovers during our measurement period. We find that, among the 13,902 SMTP servers we observe, 10,334 (74.3%) have changed their certificates; however, changing a certificate does not necessarily mean that they have updated their public and private keys. Thus, we check if they moved on to a new public key in the certificate, which leaves us 8,837 (63.6%) SMTP servers.

Changing a public key also does not necessarily indicate a rollover if the certificate associated with the public key is not what Certificate Usage in the corresponding TLSA record refers to; this usually happens when the Certificate Usage of the TLSA record is DANE-TA, but the leaf certificate is reissued from the same trust anchor. We find 2,560 (29%) of 8,837 SMTP servers are such cases. Finally, to analyze the rollover behaviors more precisely, we only consider the SMTP servers, of which (1) TTLs of their TLSA records are shorter than our scan resolution (i.e., one hour) and (2) certificates have ever been considered valid[6], which leaves us 2,569 (29%) SMTP servers to analyze.

6.2 How to Roll Over Correctly

DANE rollovers require synchronous management between SMTP and name servers. This is because when changing the public key (and its certificate), the old TLSA record may still be cached on and served from local resolvers . Recall that all DNS responses (including TLSA records) have a TTL field, which indicates that how long the DNS record can be cached;

[6]We do so because we cannot determine whether the rollover is done correctly if they have never been valid.

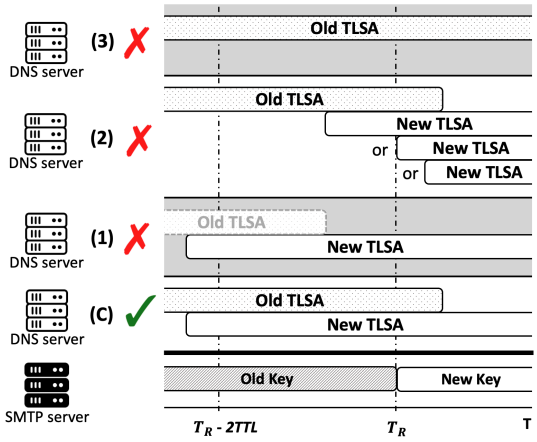

Figure 6: Rollover procedure in DANE and how validation fails is shown. For a correct rollover (C), first, the old TLSA record in the DNS server should be removed after the new key is introduced in the SMTP server at T_R, and second, the new TLSA record has to be introduced at least two TTLs before the key changes at T_R. On the other hand, the incorrect rollovers may happen when (1) the old TLSA record is removed too early, (2) the new TLSA record is introduced too late, or (3) they are never introduced.

if an SMTP server publishes a new TLSA record immediately after updating the certificate, it is possible that some SMTP clients may fetch the cached TLSA records (i.e., old TLSA records) in their DNS resolvers and cannot finish DANE validation successfully. Thus, SMTP servers must publish the new TLSA records *in advance*, at least two TTLs before moving on to the new certificate [18]. Figure 6 illustrates the correct rollover procedure as well as incorrect rollover cases.

6.3 Incorrect Rollovers

6.3.1 Early retired, late introduced, and absent TLSA records

We now examine how SMTP servers have performed rollovers; if we find any incorrect rollover, we classify it into one of the cases in Figure 6. Table 3 shows the results depending on the management case. First, we notice that *more than 87% of SMTP servers in each case perform rollovers incorrectly at least once during our measurement period*. This is discouraging because even large email hosting providers are not an exception. For example, argewebhosting.nl hosting 30,681 domains update its TLSA records one hour late after updating the certificate whenever they rollover, which results in DANE validation failures until the new TLSA record is introduced.

Second, across the three management cases, we observe that the major reason of incorrect rollovers is the late introduction of new TLSA records, while the early retirement of the

Category	Domains		SMTP servers		Incorrect Rollover Case		
	Total	Wrong Rollover	Total	Wrong Rollover	Early Retirement old TLSA	Late Introduction new TLSA	No Introduction new TLSA
SO	54,052	34,056 (63.0%)	277	255 (92.1%)	1 (0.4%)	216 (84.7%)	58 (22.8%)
SSDO	278	242 (87.1%)	275	240 (87.3%)	9 (3.9%)	173 (72.1%)	87 (36.1%)
SSDS	585	546 (93.3%)	594	544 (91.6%)	55 (10.1%)	450 (82.7%)	179 (32.9%)

Table 3: The percentages of SMTP servers that have ever roll-overed incorrectly with the reasons for each category and the impacted numbers of domains are shown. For the incorrect rollovers, the percentages of individual cases are also shown. Note that an SMTP server may have incorrectly roll-overed multiple times with different reasons during our measurement period, which makes the sum of the percentages of the reasons over 100%.

old TLSA records rarely happens. This possibly indicates that withdrawing the old TLSA record and the old certificate and introducing the new TLSA record and the certificate may happen simultaneously, but we cannot know if this is true since our scanning resolution (1 hour) cannot detect such changes.

Third, we found the prevalent cases of missing new TLSA records. The other two errors (i.e., early retirement of the old TLSA records or late introduction of the new TLSA records) cause DANE invalidations over a short period since when the new certificate or TLSA record is introduced later, the new TLSA record will be validated from that moment. If the new TLSA record is never introduced even after the rollover, it may cause permanent DANE validation failures. Moreover, we also find that some SMTP servers that introduce new TLSA records late during a rollover, never introduce a new TLSA record during another rollover, which makes the sum of the percentages of incorrect rollover cases over 100%.

One may suspect that the high failure rate could be that the new TLSA records might be introduced after our measurement period ends. However, we find that a substantial portion of SMTP servers *never change their TLSA records during their multiple rollovers*; 38 (65.6%), 68 (78.2%), and 106 (59.2%) SMTP servers do so in the SO, SSDO, and SSDS cases, respectively.

6.3.2 DANE-EE with PKIX certificates

One of the advantages of DANE is that a STARTTLS certificate need not be issued by certificate authorities (CAs) by using DANE-TA or DANE-EE usages, which allows using self-signed leaf or TA certificates. In accordance with the motivation of DANE and its best practice [18], we find that 8,201 (90.6%), 419 (4.6%), and 397 (4.4%) of DANE SMTP servers serve TLSA records with a DANE-EE, DANE-TA, and both of two usages, respectively.

However, this raises a question of why many SMTP servers do rollovers multiple times even though the majority usage is DANE-EE, which theoretically does not need to update the certificate. To answer this question, we first check how many of the certificates whose TLSA record has DANE-EE usage are *issued by popular CAs* by validating them with the well-known CA certificates. Interestingly, we find that 7,976 (94.4%) of these certificates are issued by popular CAs; more specifically,

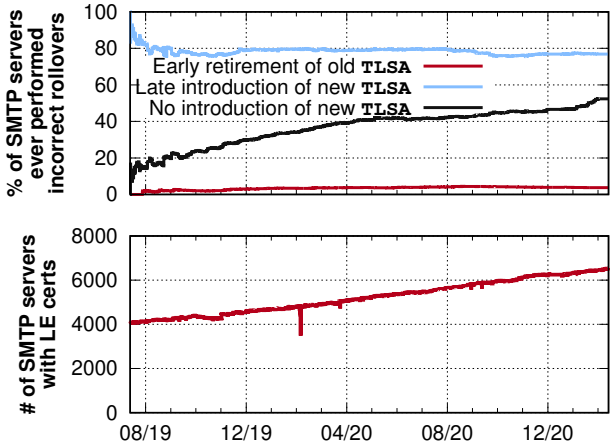

Figure 7: The percentage of the SMTP servers with Let's Encrypt (LE) certificates that perform incorrect rollovers due to each of the three errors (top) and the number of the SMTP servers that use Let's Encrypt (LE) certificates (bottom) until date *x* are shown.

two CAs, Let's Encrypt and Sectigo, issue 86.7% of the total leaf certificates in our latest snapshot.

At first glance, using a certificate issued by popular CAs does not cause any problems from an operational perspective since this certificate can be verified as long as its corresponding TLSA records are generated from the certificate. In practice, however, it could bring unexpected outcomes especially when the certificate is issued by automated CAs such as Let's Encrypt and Sectigo; these CAs often have much shorter certificate lifetimes, and certificates are automatically reissued on a regular basis (e.g., every three months). Furthermore, many of these automated tools such as certbot reissue a certificate with a new private and public key by default [15], *which means that the DANE administrator has to update the TLSA record accordingly*. Thus, there must be automated tools for updating TLSA records as well to set up a schedule to withdraw the old TLSA record and introduce the new TLSA record at the right timing; however, to the best of our knowledge, we cannot find such tools at the time of writing.

To measure these unexpected consequences, we focus on the certificates issued by Let's Encrypt because (1) it covers 77.6% of the collected leaf certificates whose TLSA record usage is DANE-EE and (2) Let's Encrypt exclusively issues certificates with automation tools at least every three months. After that, we see what type of incorrect rollovers an SMTP server has made until a given end date from the start date of our measurements. Figure 7 shows the distribution as we move the end date forward. *First*, we find that the percentage of SMTP servers that retires the old TLSA record early and introduces the new TLSA record late is almost steady.

Second, we observe a rapid growth of rollover failures due to missing new TLSA records; this is concerning because more SMTP servers seem to keep the initial TLSA record unchanged and do not update it even after certificate reissuance. Also, we find that 31.0% of SMTP servers that have introduced the new TLSA record late also sometimes do not update the TLSA record at all during another rollover[7], which suggests the behavior of updating the TLSA record when the certificate is reissued seems to be unpredictable; we believe this is due to the lack of automation support for synchronization between TLSA records and certificate reissuance.

Moreover, obtaining the public key and private key of the next certificate before getting it reissued from these automated CAs is not possible; instead, an administrator should get the next certificate in advance before the rollover to make and publish the new TLSA record[8].

Also, to the best of our knowledge, there is no DNS authoritative software support that coordinates with certificate issuance to automatically withdraw the old TLSA records and introduce new TLSA records, which makes the rollover error-prone.

6.3.3 CA Rollover

One possible option to prevent the late introduction of the new TLSA record is to use the DANE-TA usage, which allows *any leaf certificate* as long as it is issued by the certificate matching the Certificate Association Data; thus, the SMTP server can introduce the leaf certificate and the new TLSA record at the same time. We find that 571 (8.7%) of SMTP servers use TLSA records with DANE-TA usage.

Next, we see if the DANE-TA usage actually helps mitigate incorrect rollovers; Figure 8 (top) shows the percentage of SMTP servers with Let's Encrypt (LE) certificates that perform rollovers incorrectly. We make a number of observations. First, we can confirm that the DANE-TA usage effectively decreases the number of incorrect rollovers; for example, comparing the SMTP servers with the DANE-EE usage that perform rollovers (Figure 7), we can confirm that the ratio of invalid TLSA records caused by late introduction of new TLSA records dropped to 7.2% from 76% based on the snapshots on October 1st, 2020.

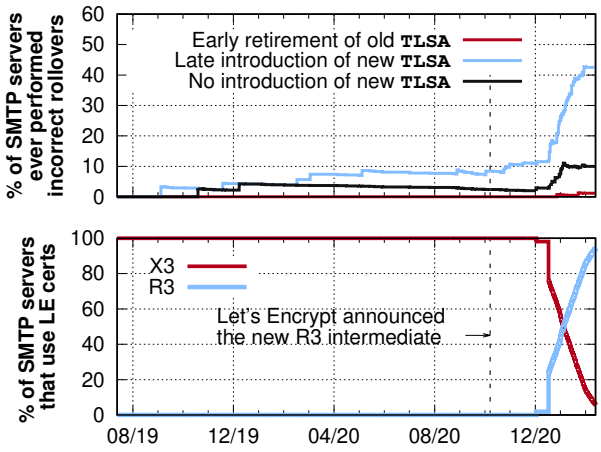

Figure 8: The percentage of SMTP servers that use Let's Encrypt (LE) certificates and have rolled over incorrectly increases (top) as Let's Encrypt starts to sign their certificates with the new R3 intermediate in October 2020.

However, we notice that this percentage suddenly increases from early October 2020. This is because Let's Encrypt (LE) announced a new intermediate certificate (called R3) on October 7th, 2020 [39] and decided to withdraw the former signing certificate, X3 [40]. We can confirm this transition by monitoring that the percentage of DANE SMTP servers with certificates signed by X3 drops but that of the ones with certificates signed by R3 grows around the late October, 2020 in Figure 8 (bottom).

Note that *the SMTP servers relying on the DANE-TA usage must update their TLSA records and follow the same best practice as described in Figure 6*. However, similar to what we observed in Figure 7, we find that most of them do not update their TLSA records with the DANE-TA usage properly, thus making both the percentage of late introduction of new TLSA records *with DANE-TA* and that of missing new TLSA records *with DANE-TA* rapidly increase right after the introduction of R3.

This suggests that the DANE-TA usage is not always resilient against certificate reissuance if the DANE administrator does not have control over the signing certificate. This raises another important question: "why do DANE administrators use PKIX certificates even when their TLSA record is configured with the DANE-EE or DANE-TA usage, which seems to be contradictory to the motivation of DANE?" The answer to this question will become clear later when we discuss the survey of DANE operators.

[7] This makes the sum of the percentages of late and missing introduction of new TLSA records over 100%.

[8] Alternatively, an administrator can generate a public and private key pair herself to make a certificate signing request (CSR) and modify a certbot command to ask Let's Encrypt to issue a certificate with the generated public key [14]

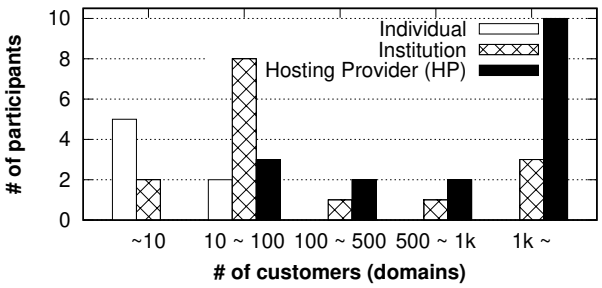

Figure 9: The distribution of how many domains each participant serves across each category is shown.

7 DANE Management in Practice

Our datasets give us an opportunity to understand how DANE is managed quantitatively. However, we use only publicly accessible information mainly from DNS or SMTP scans, making it hard for us to understand *why* operators use it, what they do to manage it, and what challenges they have. To bridge the gap between the view of how we see the DANE SMTP servers and how organizations serving mail services view it, we conducted a survey in early 2021.

7.1 Survey Methodology

We have collaborated with (1) `.nl` and `.se` registries where DANE is widely deployed in their second-level domains with `MX` records (e.g., their percentages of domains with MX records that have `TLSA` records were 9.8% and 38.2% in 2019 [36]) to share our survey with their registrars and (2) three network operator groups: NANOG [45], DENOG [23], and NLNOG [46].

In total, we received answers from 39 email operators and classified them into three categories depending on the purposes of SMTP servers: individual, institution, and hosting provider (HP). Figure 9 shows a summary; we believe that the composition of participants is broad enough to cover a wide spectrum of operators in the DANE community. Also, we have 10 participants from hosting providers who manage more than 1,000 domains; as DANE is usually managed well by the hosting providers that serve a large number of domains, we expect to learn lessons for better DANE management.

Ethical considerations Our survey focuses on organizations (and their policies), not individual people. Furthermore, we do not collect any personal information and our analyses are also not based on human subjects. Thus, our survey did not require IRB approval, which was confirmed through offline conversations with our institution's Institutional Review Board (IRB).[9]

[9]We made our questionnaire publicly available at `https://dane-study.github.io`.

7.2 Deployment and Management

Reasons of (not) supporting DANE: We first try to understand the motivation behind DANE support from the 24 (61.5%) participants who deployed DANE for SMTP. We note that they also deployed other SMTP security extensions; SPF (24), DKIM (23), DMARC (21), and MTA-STS (9), which indicates that they are aware of security challenges in SMTP. Interestingly, we find that 9 administrators (4 Individuals, 1 Institution, and 4 HPs) also use MTA-STS [41]; this is particularly interesting because MTA-STS aims to authenticate receivers without mandating DNSSEC. We also confirm this by asking them why they deployed DANE; (1) 14 out of 24 participants indicated their main purpose was to protect their customers (domains) from STARTTLS stripping attacks, and (2) 7 (out of 24) participants indicated that they do not trust CAs, which implies that they want to control their own certificates. These results are in line with our findings that most of the `TLSA` usages (90.7%) of certificates are `DANE-EE`. As one participant replies that he does not know whether DANE is supported or not, there are 14 (39.5%) participants who do not support DANE for their domains. Among the 14 participants supporting no DANE, 11 participants provided the reason; 5 (out of 11) answered they do not support DANE due to its *operational complexity*. For further analyses, we focus on the 24 participants who support DANE for their SMTP servers.

DNSSEC: We find that all of the 24 participants indicated that they also support DNSSEC. When we ask whether they have faced any problems with managing DNSSEC, 12 of them indicated that they have never experienced any issues, among which 7 participants provided their `MX` records. However, we investigate their data in our datasets and find that one of them had wrong `RRSIG`s for 14 hours making the validation status of his `TLSA` record `bogus`. The other 12 participants suffered from DNSSEC issues related with `RRSIG` records such as expired signatures. This is in line with our findings that DNSSEC is the major hurdle for correct DANE deployment.

STARTTLS certificates: We have observed that the SMTP servers of 16 participants (out of 24 DANE-supporting ones) have `TLSA` records with `DANE-EE` usage, but all of them use CA-issued certificates. We can confirm the reason for this behavior; we find that 22 (out of 24) participants use CA certificates, among which 12 indicated that they do so mainly for *compatibility with other SMTP servers that do not support DANE yet*. Since the DANE deployment rate is still low, it looks like a safer choice for SMTP servers to use *generally-trusted* certificates rather than serving self-signed certificates.[10] However, we believe this leads to a chicken-and-egg problem; (1) even popular email servers do serve CA-issued certificates with `DANE-EE TLSA` records since SMTP

[10]One administrator said *"Not every mail server supports DANE validation, hence a non-self-signed certificate is more trustworthy for those. On the other hand if I'd had only remote servers supporting DANE, I would not care."*

clients rarely support DANE and (2) since the certificates still look valid without DANE validation, the SMTP clients do not bother to support DANE or check the certificates with TLSA records, which does not seem to improve the current situation.

DANE management: We find that 19 (out of DANE-supporting 24) participants indicated that they have never experienced any DANE misconfigurations. However, we find some inconsistency between their responses and what we observe from our datasets; (1) we find the MX records of 10 (out of 19) participants in our dataset and we analyze that 3 had experienced misconfigurations: insecure and bogus DNSSEC records, and a TLSA record mismatch. (2) Also, we find that 11 (out of 24) participants indicated that they have performed rollovers. However, we find that 4 of them indicated that they update TLSA records and their certificates *simultaneously*, which causes transient DANE validation failures preventing them from receiving emails from DANE-validating clients. We believe the discrepancy between how they perceive their management and actual errors discovered from our dataset can be attributed to the challenges for detecting DANE validation failures. First, the SMTP server has to keep monitoring their DNSSEC records, certificates, and TLSA records consistently. Also, when the SMTP servers roll over their TLSA records without considering the TTLs, the clients may use the stale TLSA records cached in their local resolvers. Note that this issue resolves itself as the TTL expires, making it hard for administrators of the SMTP servers to detect such intermittent errors. For the 6 (out of 11) participants who indicated that they upload TLSA records before publishing the certificates, we find that four of them indicated using *self-developed automated scripts* to introduce the new certificates and their TLSA records at the right time. This implies that (1) automation is indeed needed for successful rollovers in DANE but (2) there is a lack of software support for the automatic rollover as all of them made scripts by themselves.

8 Discussion

We have observed pervasive DANE mismanagement mainly due to the complex procedure for key management such as when updating the key. We now ask if SMTP servers who deploy DANE for the first time, who do not have to consider issues we discovered such as old TLSA records or TTL, also experience DANE misconfiguration. Thus, in this section, we examine how well such SMTP servers can deploy DANE initially without any problems and also discuss and develop automated tools to help DANE operators manage DANE correctly.

8.1 Initial DANE Deployment

To find those who deploy DANE for the first time, we set the reference period to be the first three-month snapshot from July 13th, 2019 to October 12th, 2019. If an SMTP server has an MX record without a TLSA record for these three months, and then it has a new MX record with a TLSA record in the following period, it is assumed to deploy DANE for the first time. From this process, we identify 6,957 (50%) SMTP servers who deploy DANE for the first time out of 13,902 SMTP servers we monitored during our measurement period.[11]

SMTP Servers	DNSSEC		TLSA Mismatch	
	Insecure	Bogus	Wrong Fields	Unknown
3,051	2,972 (97.4%)	15 (0.5%)	12 (0.4%)	314 (10.3%)

Table 4: The percentage of each case of incorrect initial deployments is shown.

For this analysis, we consider only their first snapshots to see if they correctly deploy DANE and, if not, we see what the first problem is that they face. Table 4 shows a summary. First of all, we find that 3,051 (49.2%) SMTP servers fail in deploying DANE successfully in their first deployments. Next, we see each of the reasons why they fail; we notice that the vast majority of them (97.4%) are not DNSSEC-signed, which is in line with our previous findings. We also find 12 SMTP servers configured their TLSA records with wrong parameters. Interestingly, we still find that a total of 314 (10.3%) TLSA records are invalid due to mismatches, among which 7 records have their Certificate Association Data values of the SHA256 hash of an empty string, and 8 records use DANE-TA usage with a hash of the TA's public key but SMTP servers serve a different public key. These results further underscore the challenges of deploying DANE correctly, which can weaken security in the DANE PKI.

8.2 Automated Tools

We have observed that most DANE mismanagement comes from a lack of support for an automation process. We believe this problem to be analogous to what the Web's PKI ecosystem faced about a decade ago due to lots of manual processes involved to deploy TLS in web servers. Fortunately, the situation has improved greatly as lots of tools for automated certificate issuance such as certbot have been introduced and widely adopted [3, 7, 55]. For individuals or small institutions that do not have enough resources to manage TLSA records and certificates by themselves, we aim to provide an automation tool on top of popular open source-based MTA software, Mail-In-a-Box [1][12], which supports DANE by installing a name server together to post TLSA records from

[11] We cannot capture the domains that had retracted DANE support before the start date of our measurement, but deployed DANE again after then. However, these SMTP servers did not use DANE at least for 3 months, we believe it is still worthwhile investigating whether they deploy DANE correctly or not.

[12] The software uses a specific convention for MX records by creating a subdomain, box (e.g., box.example.com); we find that 35% of MX records with this subdomain in our latest snapshot.

a certificate issued from Let's Encrypt. However, it never updates the TLSA record nor the certificate, which we believe is due to the complexity of rollovers. In this section, we implement an automatic rollover and discuss further challenges.

Implementing a rollover: We implement one of the best practices for rollovers, the *Double TLSA* scheme [30], which always manages two TLSA records, one for the current public key (i.e., active key), and the other for the next public key (i.e., standby key).[13] Here is an overview of how it works: (1) we first generate two public/private key pairs (for the active key and standby key) using openssl and generate two TLSA records with these two keys, (2) we send a certificate signing request (CSR) with *the active key* to get a cert from Let's Encrypt, (3) the SMTP server now serves the certificate and the name server serves two TLSA records, (4) every two months, we generate (i) a CSR with *the standby key* to get a certificate and (ii) another key pair for the new standby TLSA record, (5) we replace (i) the old certificate with the new one and (ii) the old TLSA record of the old certificate with the TLSA record of the new standby key.

Challenges in automation: At first glance, it looks straightforward for domain owners to manage DANE rollovers using automation. However, we still find challenges when either the name server is self-managed or outsourced. When a domain owner manages its nameserver, she has to enable DNSSEC by herself; fortunately, generating DNSSEC records such as DNSKEYs and RRSIGs for a domain is a simple process because lots of popular DNS software supports it by simply enabling an option [24]. However, *constructing a chain of trust still requires a manual process*; the domain owner must ask her registrar to upload the DS record to their registry [13].[14]

When a domain owner outsources a nameserver to an external DNS operator such as Cloudflare, it has to provide two functionalities: (1) it must support APIs for the domain owner to update her DNS entries via automated scripts, and (2) it must support TLSA records in their name server. We examine 25 popular DNS operators in terms of the number of domains that they serve [13]; however, we find that only 5 DNS operators (Cloudflare, GoDaddy, 1&1, Network Solutions, OVH) do support both API and TLSA records; 3 DNS operators (Google Domains, eNOM, HostGator) supports TLSA records but do not support it via API; the others do not support neither of them.

9 Conclusion

In this paper, our goal was to investigate *why* DANE mismanagement is so prevalent. We first used a longitudinal dataset spanning 20 months to validate TLSA records of SMTP servers and found that more than 18% of SMTP servers used by domains of .com, .org, and .net are mismanaged. Most of such mismanagements happen when domain owners self-manage their SMTP servers mainly due to (1) unsuccessful deployment of DNSSEC (92%) and (2) key changes due to automatic certificate reissuances (70%). We also discovered insecure practices while performing key rollovers; 90% of SMTP servers performed rollovers incorrectly by mishandling TTLs of TLSA records, or did not update TLSA records timely due to changes of leaf and CA certificates. We confirmed our findings through DANE surveys . Finally, we modified the popular MTA software, Mail-In-a-Box, to support automatic key rollovers by implementing the *Double TLSA* scheme, but also found several systematic barriers against automation. Taken together, our results shed light on the difficulties that domain owners face when trying to deploy and manage DANE.

[13] Our pull request to the main repository has been made and waiting for the merge into master.

[14] CDS and CDNSKEY were introduced for uploading a DS record automatically [33, 58], but they have been hardly adopted by registries.

Acknowledgments

We thank the anonymous reviewers and our shepherd, Ben Stock, for their helpful comments. This research was supported in part by NSF grants CNS-2053363 and CNS-2051166, the EU H2020 CONCORDIA project (830927), the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2021-2020-0-01602) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation), and the IITP grant funded by the Korea government (MSIT) (No.2020-0-00325, Traceability Assurance Technology Development for Full Lifecycle Data Safety of Cloud Edge). Also, this work was made possible by OpenINTEL (https://www.openintel.nl/), a joint project of the University of Twente, SIDN, SURF and NLnet Labs.

References

[1] Mail-in-a-Box. https://mailinabox.email/.

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

[3] J. Aas, R. Barnes, B. Case, Z. Durumeric, P. Eckersley, A. Flores-López, J. A. Halderman, J. Hoffman-Andrews, J. Kasten, E. Rescorla, S. Schoen, and a. B. Warren. Let's Encrypt: An Automated Certificate: Authority to Encrypt the Entire Web. *CCS*, 2019.

[4] 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.

[5] 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.

[6] 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.

[7] M. Bernhard, J. Sharman, C. Z. Acemyan, P. Kortum, D. S. Wallach, and J. A. Halderman. On the Usability of HTTPS Deployment. *CHI*, 2019.

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

[9] F. Cangialosi, T. Chung, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson. Measurement and Analysis of Private Key Sharing in the HTTPS Ecosystem. *CCS*, 2016.

[10] M. Crispin. INTERNET MESSAGE ACCESS PROTOCOL. IETF RFC 3501, IEFT, 2003.

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

[12] 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.

[13] 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.

[14] Can I use an existing private key or Certificate Signing Request (CSR) with Certbot? https://certbot.eff.org/faq#can-i-use-an-existing-private-key-or-certificate-signing-request-csr-with-certbot.

[15] Certbot User Guide. https://certbot.eff.org/docs/using.html?highlight=renew#user-guide.

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

[17] Comodo-Fraud-Incident-2011-03-23. https://www.comodo.com/Comodo-Fraud-Incident-2011-03-23.html.

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

[19] V. Dukhovni. DANE in SMTP—the sky is not falling. 2020. http://dnssec-stats.ant.isi.edu/~viktor/test.html.

[20] 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.

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

[22] DANE SMTP Validator by SIDN Labs. https://check.sidnlabs.nl/dane/.

[23] German Network Operators' Group. https://www.denog.de/de/.

[24] DNSSEC signzone manual pages. https://ftp.isc.org/isc/bind9/cur/9.9/doc/arm/man.dnssec-signzone.html.

[25] C. Evans, C. Palmer, and R. Sleevi. Public Key Pinning Extension for HTTP. RFC 7469, IETF, 2015. http://www.ietf.org/rfc/rfc7469.txt.

[26] 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.

[27] Fake DigiNotar web certificate risk to Iranians. *BBC News*. https://www.bbc.com/news/technology-14789763.

[28] French gov used fake Google certificate to read its workers' traffic. https://www.theregister.co.uk/2013/12/10/french_gov_dodgy_ssl_cert_reprimand/.

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

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

[31] R. Holz, J. Amann, O. Mehani, M. Wachs, and M. A. Kaafar. TLS in the wild: an Internet-wide analysis of TLS-based protocols for electronic communication. *NDSS*, 2015.

[32] P. Hallam-Baker and R. Stradling. DNS Certification Authority Authorization (CAA) Resource Record. IETF, 2013.

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

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

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

[36] H. Lee, A. Girish, R. van Rijswijk-Deij, T. T. Kwon, and T. Chung. A Longitudinal and Comprehensive Study of the DANE Ecosystem in Email. *USENIX Security*, 2020.

[37] S. Liu, I. Foster, S. Savage, G. M. Voelker, and L. K. Saul. Who is .com? Learning to Parse WHOIS Records. *IMC*, 2015.

[38] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics doklady*, 10(8), 1966.

[39] Let's Encrypt R3. https://crt.sh/?id=3479778542.

[40] Let's Encrypt's New Root and Intermediate Certificates. https://letsencrypt.org/2020/09/17/new-root-and-intermediates.html.

[41] 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.

[42] J. Myers and M. Rose. Post Office Protocol - Version 3. RFC 1939, IETF, 1996.

[43] Measuring Middlebox Interference with DNS Records. https://blog.mozilla.org/security/2020/11/17/measuring-middlebox-interference-with-dns-records/.

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

[45] North American Network Operators' Group. https://www.nanog.org/.

[46] Netherlands Network Operators' Group. https://nlnog.net/.

[47] Nameservers and TLDs supported/unsupported by DNSSEC. https://www.namecheap.com/support/knowledgebase/article.aspx/9718/2232/nameservers-and-tlds-supportedunsupported-by-dnssec.

[48] 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.

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

[50] D. Poddebniak, F. Ising, H. Böck, and S. Schinzel. Why TLS is better without STARTTLS: A Security Analysis of STARTTLS in the Email Context. *USENIX Security*, 2021.

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

[52] SIDN Labs DANE Deployment Statistics. https://stats.sidnlabs.nl/en/mail.html#dane.

[53] SMTP DANE TLS Adoption Survey. https://stats.dnssec-tools.org/explore/.

[54] STARTTLS and DANE for outgoing mail mandatory for government organisations. 2019. https://www.sidn.nl/en/news-and-blogs/starttls-and-dane-for-outgoing-mail-mandatory-for-government-organisations.

[55] C. Tiefenau, E. von Zezschwitz, M. Häring, K. Krombholz, and M. Smith. A Usability Evaluation of Let's Encrypt and Certbot: Usable Security Done Right. *CCS*, 2019.

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

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

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

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