

Internet Engineering Task Force (IETF)  
Request for Comments: 7672  
Category: Standards Track  
ISSN: 2070-1721

V. Dukhovni  
Two Sigma  
W. Hardaker  
Parsons  
October 2015

## SMTP Security via Opportunistic DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS)

### Abstract

This memo describes a downgrade-resistant protocol for SMTP transport security between Message Transfer Agents (MTAs), based on the DNS-Based Authentication of Named Entities (DANE) TLSA DNS record. Adoption of this protocol enables an incremental transition of the Internet email backbone to one using encrypted and authenticated Transport Layer Security (TLS).

### Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc7672>.

### Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction .....	3
1.1. Terminology .....	4
1.2. Background .....	6
1.3. SMTP Channel Security .....	6
1.3.1. STARTTLS Downgrade Attack .....	7
1.3.2. Insecure Server Name without DNSSEC .....	7
1.3.3. Sender Policy Does Not Scale .....	8
1.3.4. Too Many Certification Authorities .....	9
2. Identifying Applicable TLSA Records .....	9
2.1. DNS Considerations .....	9
2.1.1. DNS Errors, "Bogus" Responses, and "Indeterminate" Responses .....	9
2.1.2. DNS Error Handling .....	11
2.1.3. Stub Resolver Considerations .....	12
2.2. TLS Discovery .....	13
2.2.1. MX Resolution .....	14
2.2.2. Non-MX Destinations .....	16
2.2.3. TLSA Record Lookup .....	18
3. DANE Authentication .....	20
3.1. TLSA Certificate Usages .....	20
3.1.1. Certificate Usage DANE-EE(3) .....	21
3.1.2. Certificate Usage DANE-TA(2) .....	22
3.1.3. Certificate Usages PKIX-TA(0) and PKIX-EE(1) .....	23
3.2. Certificate Matching .....	24
3.2.1. DANE-EE(3) Name Checks .....	24
3.2.2. DANE-TA(2) Name Checks .....	24
3.2.3. Reference Identifier Matching .....	25
4. Server Key Management .....	26
5. Digest Algorithm Agility .....	27
6. Mandatory TLS Security .....	27
7. Note on DANE for Message User Agents .....	28
8. Interoperability Considerations .....	28
8.1. SNI Support .....	28
8.2. Anonymous TLS Cipher Suites .....	29
9. Operational Considerations .....	29
9.1. Client Operational Considerations .....	29
9.2. Publisher Operational Considerations .....	30
10. Security Considerations .....	30
11. References .....	31
11.1. Normative References .....	31
11.2. Informative References .....	33
Acknowledgements .....	34
Authors' Addresses .....	34

## 1. Introduction

This memo specifies a new connection security model for Message Transfer Agents (MTAs). This model is motivated by key features of inter-domain SMTP delivery, principally, the fact that the destination server is selected indirectly via DNS Mail Exchange (MX) records and that neither email addresses nor MX hostnames signal a requirement for either secure or cleartext transport. Therefore, aside from a few manually configured exceptions, SMTP transport security is, by necessity, opportunistic (for a definition of "Opportunistic Security", see [RFC7435]).

This specification uses the presence of DANE TLSA records to securely signal TLS support and to publish the means by which SMTP clients can successfully authenticate legitimate SMTP servers. This becomes "opportunistic DANE TLS" and is resistant to downgrade and man-in-the-middle (MITM) attacks. It enables an incremental transition of the email backbone to authenticated TLS delivery, with increased global protection as adoption increases.

With opportunistic DANE TLS, traffic from SMTP clients to domains that publish "usable" DANE TLSA records in accordance with this memo is authenticated and encrypted. Traffic from legacy clients or to domains that do not publish TLSA records will continue to be sent in the same manner as before, via manually configured security, (pre-DANE) opportunistic TLS, or just cleartext SMTP.

Problems with the existing use of TLS in MTA-to-MTA SMTP that motivate this specification are described in Section 1.3. The specification itself follows, in Sections 2 and 3, which describe, respectively, how to locate and use DANE TLSA records with SMTP. In Section 6, we discuss the application of DANE TLS to destinations for which channel integrity and confidentiality are mandatory. In Section 7, we briefly comment on the potential applicability of this specification to Message User Agents.

## 1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

The following terms or concepts are used throughout this document:

**Man-in-the-middle (MITM) attack:** Active modification of network traffic by an adversary able to thereby compromise the confidentiality or integrity of the data.

**Downgrade attack:** (From [RFC4949].) A type of MITM attack in which the attacker can cause two parties, at the time they negotiate a security association, to agree on a lower level of protection than the highest level that could have been supported by both of them.

**Downgrade-resistant:** A protocol is "downgrade-resistant" if it employs effective countermeasures against downgrade attacks.

**"Secure", "bogus", "insecure", "indeterminate":** DNSSEC validation results, as defined in Section 4.3 of [RFC4035].

**Validating security-aware stub resolver and non-validating security-aware stub resolver:**

Capabilities of the stub resolver in use, as defined in [RFC4033]; note that this specification requires the use of a security-aware stub resolver.

**(Pre-DANE) opportunistic TLS:** Best-effort use of TLS that is generally vulnerable to DNS forgery and STARTTLS downgrade attacks. When a TLS-encrypted communication channel is not available, message transmission takes place in the clear. MX record indirection generally precludes authentication even when TLS is available.

**Opportunistic DANE TLS:** Best-effort use of TLS that is resistant to downgrade attacks for destinations with DNSSEC-validated TLSA records. When opportunistic DANE TLS is determined to be unavailable, clients should fall back to pre-DANE opportunistic TLS. Opportunistic DANE TLS requires support for DNSSEC, DANE, and STARTTLS on the client side, and STARTTLS plus a DNSSEC published TLSA record on the server side.

**Reference identifier:** (Special case of [RFC6125] definition.) One of the domain names associated by the SMTP client with the destination SMTP server for performing name checks on the server certificate. When name checks are applicable, at least one of the reference identifiers **MUST** match an [RFC6125] DNS-ID (or, if none are present, the [RFC6125] CN-ID) of the server certificate (see Section 3.2.3).

**MX hostname:** The RRDATA of an MX record consists of a 16 bit preference followed by a Mail Exchange domain name (see [RFC1035], Section 3.3.9). We will use the term "MX hostname" to refer to the latter, that is, the DNS domain name found after the preference value in an MX record. Thus, an "MX hostname" is specifically a reference to a DNS domain name rather than any host that bears that name.

**Delayed delivery:** Email delivery is a multi-hop store-and-forward process. When an MTA is unable to forward a message that may become deliverable later, the message is queued and delivery is retried periodically. Some MTAs may be configured with a fallback next-hop destination that handles messages that the MTA would otherwise queue and retry. When a fallback next-hop destination is configured, messages that would otherwise have to be delayed may be sent to the fallback next-hop destination instead. The fallback destination may itself be subject to opportunistic or mandatory DANE TLS (Section 6) as though it were the original message destination.

**Original next-hop destination:** The logical destination for mail delivery. By default, this is the domain portion of the recipient address, but MTAs may be configured to forward mail for some or all recipients via designated relays. The original next-hop destination is, respectively, either the recipient domain or the associated configured relay.

**MTA:** Message Transfer Agent ([RFC5598], Section 4.3.2).

**MSA:** Message Submission Agent ([RFC5598], Section 4.3.1).

**MUA:** Message User Agent ([RFC5598], Section 4.2.1).

**RR:** A DNS resource record as defined in [RFC1034], Section 3.6.

**RRset:** An RRset ([RFC2181], Section 5) is a group of DNS resource records that share the same label, class, and type.

## 1.2. Background

The Domain Name System Security Extensions (DNSSEC) add data origin authentication, data integrity, and data nonexistence proofs to the Domain Name System (DNS). DNSSEC is defined in [RFC4033], [RFC4034], and [RFC4035].

As described in the introduction of [RFC6698], TLS authentication via the existing public Certification Authority (CA) PKI suffers from an overabundance of trusted parties capable of issuing certificates for any domain of their choice. DANE leverages the DNSSEC infrastructure to publish public keys and certificates for use with the Transport Layer Security (TLS) [RFC5246] protocol via the "TLSA" DNS record type. With DNSSEC, each domain can only vouch for the keys of its delegated sub-domains.

The TLS protocol enables secure TCP communication. In the context of this memo, channel security is assumed to be provided by TLS. Used without authentication, TLS provides only privacy protection against eavesdropping attacks. Otherwise, TLS also provides data origin authentication to guard against MITM attacks.

## 1.3. SMTP Channel Security

With HTTPS, TLS employs X.509 certificates [RFC5280] issued by one of the many CAs bundled with popular web browsers to allow users to authenticate their "secure" websites. Before we specify a new DANE TLS security model for SMTP, we will explain why a new security model is needed. In the process, we will explain why the familiar HTTPS security model is inadequate to protect inter-domain SMTP traffic.

The subsections below outline four key problems with applying traditional Web PKI [RFC7435] to SMTP; these problems are addressed by this specification. Since an SMTP channel security policy is not explicitly specified in either the recipient address or the MX record, a new signaling mechanism is required to indicate when channel security is possible and should be used. The publication of TLSA records allows server operators to securely signal to SMTP clients that TLS is available and should be used. DANE TLSA makes it possible to simultaneously discover which destination domains support secure delivery via TLS and how to verify the authenticity of the associated SMTP services, providing a path forward to ubiquitous SMTP channel security.

### 1.3.1. STARTTLS Downgrade Attack

SMTP [RFC5321] is a single-hop protocol in a multi-hop store-and-forward email delivery process. An SMTP envelope recipient address does not correspond to a specific transport-layer endpoint address; rather, at each relay hop, the transport-layer endpoint is the next-hop relay, while the envelope recipient address typically remains the same. Unlike HTTP and its corresponding secured version, HTTPS, where the use of TLS is signaled via the URI scheme, email recipient addresses do not directly signal transport security policy. Indeed, no such signaling could work well with SMTP, since TLS encryption of SMTP protects email traffic on a hop-by-hop basis while email addresses could only express end-to-end policy.

With no mechanism available to signal transport security policy, SMTP relays employ a best-effort "opportunistic" security model for TLS. A single SMTP server TCP listening endpoint can serve both TLS and non-TLS clients; the use of TLS is negotiated via the SMTP STARTTLS command [RFC3207]. The server signals TLS support to the client over a cleartext SMTP connection, and, if the client also supports TLS, it may negotiate a TLS-encrypted channel to use for email transmission. The server's indication of TLS support can be easily suppressed by an MITM attacker. Thus, pre-DANE SMTP TLS security can be subverted by simply downgrading a connection to cleartext. No TLS security feature can prevent this. The attacker can simply disable TLS.

### 1.3.2. Insecure Server Name without DNSSEC

With SMTP, DNS MX records abstract the next-hop transport endpoint and allow administrators to specify a set of target servers to which SMTP traffic should be directed for a given domain.

A TLS client is vulnerable to MITM attacks unless it verifies that the server's certificate binds the public key to a name that matches one of the client's reference identifiers. A natural choice of reference identifier is the server's domain name. However, with SMTP, server names are not directly encoded in the recipient address; instead, they are obtained indirectly via MX records. Without DNSSEC, the MX lookup is vulnerable to MITM and DNS cache poisoning attacks. Active attackers can forge DNS replies with fake MX records and can redirect email to servers with names of their choice. Therefore, secure verification of SMTP TLS certificates matching the server name is not possible without DNSSEC.

One might try to harden TLS for SMTP against DNS attacks by using the envelope recipient domain as a reference identifier and by requiring each SMTP server to possess a trusted certificate for the envelope recipient domain rather than the MX hostname. Unfortunately, this is impractical, as email for many domains is handled by third parties that are not in a position to obtain certificates for all the domains they serve. Deployment of the Server Name Indication (SNI) extension to TLS (see Section 3 of [RFC6066]) is no panacea, since SNI key management is operationally challenging except when the email service provider is also the domain's registrar and its certificate issuer; this is rarely the case for email.

Since the recipient domain name cannot be used as the SMTP server reference identifier, and neither can the MX hostname without DNSSEC, large-scale deployment of authenticated TLS for SMTP requires that the DNS be secure.

Since SMTP security depends critically on DNSSEC, it is important to point out that SMTP with DANE is consequently the most conservative possible trust model. It trusts only what must be trusted and no more. Adding any other trusted actors to the mix can only reduce SMTP security. A sender may choose to further harden DNSSEC for selected high-value receiving domains by configuring explicit trust anchors for those domains instead of relying on the chain of trust from the root domain. However, detailed discussion of DNSSEC security practices is out of scope for this document.

### 1.3.3. Sender Policy Does Not Scale

Sending systems are in some cases explicitly configured to use TLS for mail sent to selected peer domains, but this requires configuring sending MTAs with appropriate subject names or certificate content digests from their peer domains. Due to the resulting administrative burden, such statically configured SMTP secure channels are used rarely (generally only between domains that make bilateral arrangements with their business partners). Internet email, on the other hand, requires regularly contacting new domains for which security configurations cannot be established in advance.

The abstraction of the SMTP transport endpoint via DNS MX records, often across organizational boundaries, limits the use of public CA PKI with SMTP to a small set of sender-configured peer domains. With little opportunity to use TLS authentication, sending MTAs are rarely configured with a comprehensive list of trusted CAs. SMTP services that support STARTTLS often deploy X.509 certificates that are self-signed or issued by a private CA.



#### 1.3.4. Too Many Certification Authorities

Even if it were generally possible to determine a secure server name, the SMTP client would still need to verify that the server's certificate chain is issued by a trusted CA (a trust anchor). MTAs are not interactive applications where a human operator can make a decision (wisely or otherwise) to selectively disable TLS security policy when certificate chain verification fails. With no user to "click OK", the MTA's list of public CA trust anchors would need to be comprehensive in order to avoid bouncing mail addressed to sites that employ unknown CAs.

On the other hand, each trusted CA can issue certificates for any domain. If even one of the configured CAs is compromised or operated by an adversary, it can subvert TLS security for all destinations. Any set of CAs is simultaneously both overly inclusive and not inclusive enough.

## 2. Identifying Applicable TLSA Records

### 2.1. DNS Considerations

#### 2.1.1. DNS Errors, "Bogus" Responses, and "Indeterminate" Responses

An SMTP client that implements opportunistic DANE TLS per this specification depends critically on the integrity of DNSSEC lookups, as discussed in Section 1.3.2. This section lists the DNS resolver requirements needed to avoid downgrade attacks when using opportunistic DANE TLS.

A DNS lookup may signal an error or return a definitive answer. A security-aware resolver **MUST** be used for this specification. Security-aware resolvers will indicate the security status of a DNS RRset with one of four possible values defined in Section 4.3 of [RFC4035]: "secure", "insecure", "bogus", and "indeterminate". In [RFC4035], the meaning of the "indeterminate" security status is:

An RRset for which the resolver is not able to determine whether the RRset should be signed, as the resolver is not able to obtain the necessary DNSSEC RRs. This can occur when the security-aware resolver is not able to contact security-aware name servers for the relevant zones.

Note that the "indeterminate" security status has a conflicting definition in Section 5 of [RFC4033]:

There is no trust anchor that would indicate that a specific portion of the tree is secure.

In this document, the term "indeterminate" will be used exclusively in the [RFC4035] sense. Therefore, obtaining "indeterminate" lookup results is a (transient) failure condition, namely, the inability to locate the relevant DNS records. DNS records that would be classified "indeterminate" in the sense of [RFC4035] are simply classified as "insecure".

We do not need to distinguish between zones that lack a suitable ancestor trust anchor, and delegations (ultimately) from a trust anchor that designate a child zone as being "insecure". All "insecure" RRsets MUST be handled identically: in either case, non-validated data for the query domain is all that is and can be available, and authentication using the data is impossible. As the DNS root zone has been signed, we expect that validating resolvers used by Internet-facing MTAs will be configured with trust anchor data for the root zone and that therefore domains with no ancestor trust anchor will not be possible in most deployments.

As noted in Section 4.3 of [RFC4035], a security-aware DNS resolver MUST be able to determine whether a given non-error DNS response is "secure", "insecure", "bogus", or "indeterminate". It is expected that most security-aware stub resolvers will not signal an "indeterminate" security status (in the sense of [RFC4035]) to the application and will instead signal a "bogus" or error result. If a resolver does signal an [RFC4035] "indeterminate" security status, this MUST be treated by the SMTP client as though a "bogus" or error result had been returned.

An MTA using a non-validating security-aware stub resolver MAY use the stub resolver's ability, if available, to signal DNSSEC validation status based on information the stub resolver has learned from an upstream validating recursive resolver. Security-oblivious stub resolvers [RFC4033] MUST NOT be used. In accordance with Section 4.9.3 of [RFC4035]:

... a security-aware stub resolver MUST NOT place any reliance on signature validation allegedly performed on its behalf, except when the security-aware stub resolver obtained the data in question from a trusted security-aware recursive name server via a secure channel.

To avoid much repetition in the text below, we will pause to explain the handling of "bogus" or "indeterminate" DNSSEC query responses. These are not necessarily the result of a malicious actor; they can, for example, occur when network packets are corrupted or lost in transit. Therefore, "bogus" or "indeterminate" replies are equated in this memo with lookup failure.

There is an important non-failure condition we need to highlight in addition to the obvious case of the DNS client obtaining a non-empty "secure" or "insecure" RRset of the requested type. Namely, it is not an error when either "secure" or "insecure" nonexistence is determined for the requested data. When a DNSSEC response with a validation status that is either "secure" or "insecure" reports either no records of the requested type or nonexistence of the query domain, the response is not a DNS error condition. The DNS client has not been left without an answer; it has learned that records of the requested type do not exist.

Security-aware stub resolvers will, of course, also signal DNS lookup errors in other cases, for example, when processing a "SERVFAIL" [RFC2136] response code (RCODE) [RFC1035], which will not have an associated DNSSEC status. All lookup errors are treated the same way by this specification, regardless of whether they are from a "bogus" or "indeterminate" DNSSEC status or from a more generic DNS error: the information that was requested cannot be obtained by the security-aware resolver at this time. Thus, a lookup error is either a failure to obtain the relevant RRset if it exists or a failure to determine that no such RRset exists when it does not.

In contrast to a "bogus" response or an "indeterminate" response, an "insecure" DNSSEC response is not an error; rather, as explained above, it indicates that the target DNS zone is either delegated as an "insecure" child of a "secure" parent zone or not a descendant of any of the configured DNSSEC trust anchors in use by the SMTP client. "Insecure" results will leave the SMTP client with degraded channel security but do not stand in the way of message delivery. See Section 2.2 for further details.

### 2.1.2. DNS Error Handling

When a DNS lookup failure (an error, "bogus", or "indeterminate", as defined above) prevents an SMTP client from determining which SMTP server or servers it should connect to, message delivery **MUST** be delayed. This naturally includes, for example, the case when a "bogus" or "indeterminate" response is encountered during MX resolution. When multiple MX hostnames are obtained from a successful MX lookup but a later DNS lookup failure prevents network address resolution for a given MX hostname, delivery may proceed via any remaining MX hosts.

When a particular SMTP server is securely identified as the delivery destination, a set of DNS lookups (Section 2.2) **MUST** be performed to locate any related TLSA records. If any DNS queries used to locate TLSA records fail (due to "bogus" or "indeterminate" records, timeouts, malformed replies, SERVFAIL responses, etc.), then the SMTP

client **MUST** treat that server as unreachable and **MUST NOT** deliver the message via that server. If no servers are reachable, delivery is delayed.

In the text that follows, we will only describe what happens when all relevant DNS queries succeed. If any DNS failure occurs, the SMTP client **MUST** behave as described in this section, by "skipping" the SMTP server or destination that is problematic. Queries for candidate TLSA records are explicitly part of "all relevant DNS queries", and SMTP clients **MUST NOT** continue to connect to an SMTP server or destination whose TLSA record lookup fails.

### 2.1.3. Stub Resolver Considerations

A note about DNAME aliases: a query for a domain name whose ancestor domain is a DNAME alias returns the DNAME RR for the ancestor domain along with a CNAME that maps the query domain to the corresponding sub-domain of the target domain of the DNAME alias [RFC6672]. Therefore, whenever we speak of CNAME aliases, we implicitly allow for the possibility that the alias in question is the result of an ancestor domain DNAME record. Consequently, no explicit support for DNAME records is needed in SMTP software; it is sufficient to process the resulting CNAME aliases. DNAME records only require special processing in the validating stub resolver library that checks the integrity of the combined DNAME + CNAME reply. When DNSSEC validation is handled by a local caching resolver rather than the MTA itself, even that part of the DNAME support logic is outside the MTA.

When a stub resolver returns a response containing a CNAME alias that does not also contain the corresponding query results for the target of the alias, the SMTP client will need to repeat the query at the target of the alias and should do so recursively up to some configured or implementation-dependent recursion limit. If at any stage of CNAME expansion an error is detected, the lookup of the original requested records **MUST** be considered to have failed.

Whether a chain of CNAME records was returned in a single stub resolver response or via explicit recursion by the SMTP client, if at any stage of recursive expansion an "insecure" CNAME record is encountered, then it and all subsequent results (in particular, the final result) **MUST** be considered "insecure", regardless of whether or not any earlier CNAME records leading to the "insecure" record were "secure".

Note that a security-aware non-validating stub resolver may return to the SMTP client an "insecure" reply received from a validating recursive resolver that contains a CNAME record along with additional answers recursively obtained starting at the target of the CNAME. In

this case, the only possible conclusion is that some record in the set of records returned is "insecure", and it is, in fact, possible that the initial CNAME record and a subset of the subsequent records are "secure".

If the SMTP client needs to determine the security status of the DNS zone containing the initial CNAME record, it will need to issue a separate query of type "CNAME" that returns only the initial CNAME record. Specifically, as discussed in Section 2.2.2, when "insecure" A or AAAA records are found for an SMTP server via a CNAME alias, the SMTP client will need to perform an additional CNAME query in order to determine whether or not the DNS zone in which the alias is published is DNSSEC signed.

## 2.2. TLS Discovery

As noted previously (in Section 1.3.1), opportunistic TLS with SMTP servers that advertise TLS support via STARTTLS is subject to an MITM downgrade attack. Also, some SMTP servers that are not, in fact, TLS capable erroneously advertise STARTTLS by default, and clients need to be prepared to retry cleartext delivery after STARTTLS fails. In contrast, DNSSEC-validated TLSA records **MUST NOT** be published for servers that do not support TLS. Clients can safely interpret their presence as a commitment by the server operator to implement TLS and STARTTLS.

This memo defines four actions to be taken after the search for a TLSA record returns "secure" usable results, "secure" unusable results, "insecure" or no results, or an error signal. The term "usable" in this context is in the sense of Section 4.1 of [RFC6698]. Specifically, if the DNS lookup for a TLSA record returns:

- A "secure" TLSA RRset with at least one usable record: Any connection to the MTA **MUST** employ TLS encryption and **MUST** authenticate the SMTP server using the techniques discussed in the rest of this document. Failure to establish an authenticated TLS connection **MUST** result in falling back to the next SMTP server or delayed delivery.
- A "secure" non-empty TLSA RRset where all the records are unusable: Any connection to the MTA **MUST** be made via TLS, but authentication is not required. Failure to establish an encrypted TLS connection **MUST** result in falling back to the next SMTP server or delayed delivery.

An "insecure" TLSA RRset or DNSSEC-authenticated denial of existence of the TLSA records:

A connection to the MTA **SHOULD** be made using (pre-DANE) opportunistic TLS; this includes using cleartext delivery when the remote SMTP server does not appear to support TLS. The MTA **MAY** retry in cleartext when delivery via TLS fails during the handshake or even during data transfer.

Any lookup error: Lookup errors, including "bogus" and "indeterminate" as explained in Section 2.1.1, **MUST** result in falling back to the next SMTP server or delayed delivery.

An SMTP client **MAY** be configured to mandate DANE-verified delivery for some destinations. With mandatory DANE TLS (Section 6), delivery proceeds only when "secure" TLSA records are used to establish an encrypted and authenticated TLS channel with the SMTP server.

When the original next-hop destination is an address literal rather than a DNS domain, DANE TLS does not apply. Delivery proceeds using any relevant security policy configured by the MTA administrator. Similarly, when an MX RRset incorrectly lists a network address in lieu of an MX hostname, if an MTA chooses to connect to the network address in the nonconformant MX record, DANE TLSA does not apply for such a connection.

In the subsections that follow, we explain how to locate the SMTP servers and the associated TLSA records for a given next-hop destination domain. We also explain which name or names are to be used in identity checks of the SMTP server certificate.

### 2.2.1. MX Resolution

In this section, we consider next-hop domains that are subject to MX resolution and have MX records. The TLSA records and the associated base domain are derived separately for each MX hostname that is used to attempt message delivery. DANE TLS can authenticate message delivery to the intended next-hop domain only when the MX records are obtained securely via a DNSSEC-validated lookup.

MX records **MUST** be sorted by preference; an MX hostname with a worse (numerically higher) MX preference that has TLSA records **MUST NOT** preempt an MX hostname with a better (numerically lower) preference that has no TLSA records. In other words, prevention of delivery loops by obeying MX preferences **MUST** take precedence over channel security considerations. Even with two equal-preference MX records, an MTA is not obligated to choose the MX hostname that offers more

security. Domains that want secure inbound mail delivery need to ensure that all their SMTP servers and MX records are configured accordingly.

In the language of [RFC5321], Section 5.1, the original next-hop domain is the "initial name". If the MX lookup of the initial name results in a CNAME alias, the MTA replaces the initial name with the resulting name and performs a new lookup with the new name. MTAs typically support recursion in CNAME expansion, so this replacement is performed repeatedly (up to the MTA's recursion limit) until the ultimate non-CNAME domain is found.

If the MX RRset (or any CNAME leading to it) is "insecure" (see Section 2.1.1) and DANE TLS for the given destination is mandatory (Section 6), delivery **MUST** be delayed. If the MX RRset is "insecure" and DANE TLS is not mandatory, the SMTP client is free to use pre-DANE opportunistic TLS (possibly even cleartext).

Since the protocol in this memo is an Opportunistic Security protocol [RFC7435], the SMTP client **MAY** elect to use DANE TLS (as described in Section 2.2.2 below), even with MX hosts obtained via an "insecure" MX RRset. For example, when a hosting provider has a signed DNS zone and publishes TLSA records for its SMTP servers, hosted domains that are not signed may still benefit from the provider's TLSA records. Deliveries via the provider's SMTP servers will not be subject to active attacks when sending SMTP clients elect to use the provider's TLSA records (active attacks that tamper with the "insecure" MX RRset are of course still possible in this case).

When the MX records are not (DNSSEC) signed, an active attacker can redirect SMTP clients to MX hosts of his choice. Such redirection is tamper-evident when SMTP servers found via "insecure" MX records are recorded as the next-hop relay in the MTA delivery logs in their original (rather than CNAME-expanded) form. Sending MTAs **SHOULD** log unexpanded MX hostnames when these result from "insecure" MX lookups. Any successful authentication via an insecurely determined MX host **MUST NOT** be misrepresented in the mail logs as secure delivery to the intended next-hop domain.

In the absence of DNS lookup errors (Section 2.1.1), if the MX RRset is not "insecure", then it is "secure", and the SMTP client **MUST** treat each MX hostname as described in Section 2.2.2. When, for a given MX hostname, no TLSA records are found or only "insecure" TLSA records are found, DANE TLSA is not applicable with the SMTP server in question, and delivery proceeds to that host as with pre-DANE opportunistic TLS. To avoid downgrade attacks, any errors during TLSA lookups **MUST**, as explained in Section 2.1.2, cause the SMTP server in question to be treated as unreachable.

### 2.2.2. Non-MX Destinations

This section describes the algorithm used to locate the TLSA records and associated TLSA base domain for an input domain that is not subject to MX resolution, that represents a hostname from a "secure" MX RRset, or that lacks MX records. Such domains include:

- o Any host that is configured by the sending MTA administrator as the next-hop relay for some or all domains and that is not subject to MX resolution.
- o A domain that has MX records. When a domain has MX records, we treat each MX host listed in those MX records as though it were a non-MX destination -- that is, in the same way as we would treat an administrator-configured relay that handles mail for that domain. (Unlike administrator-specified relays, MTAs are not required to support CNAME expansion of next-hop names found via MX lookups.)
- o A next-hop destination domain subject to MX resolution that has no MX records. In this case, the domain's name is implicitly also its sole SMTP server name.

Note that DNS queries with type TLSA are mishandled by load-balancing nameservers that serve the MX hostnames of some large email providers. The DNS zones served by these nameservers are not signed and contain no TLSA records. These nameservers **SHOULD** provide "insecure" negative replies that indicate the nonexistence of the TLSA records, but instead they fail by not responding at all or by responding with a DNS RCODE [RFC1035] other than NXDOMAIN, e.g., SERVFAIL or NOTIMP [RFC2136].

To avoid problems delivering mail to domains whose SMTP servers are served by these problematic nameservers, the SMTP client **MUST** perform any A and/or AAAA queries for the destination before attempting to locate the associated TLSA records. This lookup is needed in any case to determine (1) whether or not the destination domain is reachable and (2) the DNSSEC validation status of the chain of CNAME queries required to reach the ultimate address records.

If no address records are found, the destination is unreachable. If address records are found but the DNSSEC validation status of the first query response is "insecure" (see Section 2.1.3), the SMTP client **SHOULD NOT** proceed to search for any associated TLSA records. In the case of these problematic domains, TLSA queries would lead to DNS lookup errors and would cause messages to be consistently delayed and ultimately returned to the sender. We don't expect to find any



"secure" TLSA records associated with a TLSA base domain that lies in an unsigned DNS zone. Therefore, skipping TLSA lookups in this case will also reduce latency, with no detrimental impact on security.

If the A and/or AAAA lookup of the initial name yields a CNAME, we replace it with the resulting name as if it were the initial name and perform a lookup again using the new name. This replacement is performed recursively (up to the MTA's recursion limit).

We consider the following cases for handling a DNS response for an A or AAAA DNS lookup:

**Not found:** When the DNS queries for A and/or AAAA records yield neither a list of addresses nor a CNAME (or CNAME expansion is not supported), the destination is unreachable.

**Non-CNAME:** The answer is not a CNAME alias. If the address RRset is "secure", TLSA lookups are performed as described in Section 2.2.3 with the initial name as the candidate TLSA base domain. If no "secure" TLSA records are found, DANE TLS is not applicable and mail delivery proceeds with pre-DANE opportunistic TLS (which, being best-effort, degrades to cleartext delivery when STARTTLS is not available or the TLS handshake fails).

**Insecure CNAME:** The input domain is a CNAME alias, but the ultimate network address RRset is "insecure" (see Section 2.1.1). If the initial CNAME response is also "insecure", DANE TLS does not apply. Otherwise, this case is treated just like the non-CNAME case above, where a search is performed for a TLSA record with the original input domain as the candidate TLSA base domain.

**Secure CNAME:** The input domain is a CNAME alias, and the ultimate network address RRset is "secure" (see Section 2.1.1). Two candidate TLSA base domains are tried: the fully CNAME-expanded initial name and, failing that, the initial name itself.

In summary, if it is possible to securely obtain the full, CNAME-expanded, DNSSEC-validated address records for the input domain, then that name is the preferred TLSA base domain. Otherwise, the unexpanded input domain is the candidate TLSA base domain. When no "secure" TLSA records are found at either the CNAME-expanded or unexpanded domain, then DANE TLS does not apply for mail delivery via the input domain in question. And, as always, errors, "bogus" results, or "indeterminate" results for any query in the process MUST result in delaying or abandoning delivery.

### 2.2.3. TLSA Record Lookup

When the SMTP server's hostname is not a CNAME or DNAME alias, the list of associated candidate TLSA base domains (see below) consists of just the server hostname.

When the hostname is an alias with a "secure" (at every stage) full expansion, the list of candidate TLSA base domains (see below) is a pair of domains: the fully expanded server hostname first, and the unexpanded server hostname second.

Each candidate TLSA base domain (alias-expanded or original) is in turn prefixed with service labels of the form "`<port>._tcp`". The resulting domain name is used to issue a DNSSEC query with the query type set to TLSA ([RFC6698], Section 7.1).

The first of these candidate domains to yield a "secure" TLSA RRset becomes the actual TLSA base domain.

For SMTP, the destination TCP port is typically 25, but this may be different with custom routes specified by the MTA administrator, in which case the SMTP client MUST use the appropriate number in the "`<port>`" prefix in place of "25". If, for example, the candidate base domain is "mx.example.com" and the SMTP connection is to port 25, the TLSA RRset is obtained via a DNSSEC query of the form:

`_25._tcp.mx.example.com. IN TLSA ?`

The query response may be a CNAME or the actual TLSA RRset. If the response is a CNAME, the SMTP client (through the use of its security-aware stub resolver) restarts the TLSA query at the target domain, following CNAMEs as appropriate, and keeps track of whether or not the entire chain is "secure". If any "insecure" records are encountered or the TLSA records don't exist, the next candidate TLSA base domain is tried instead.

If the ultimate response is a "secure" TLSA RRset, then the candidate TLSA base domain will be the actual TLSA base domain, and the TLSA RRset will constitute the TLSA records for the destination. If none of the candidate TLSA base domains yield "secure" TLSA records, then the SMTP client is free to use pre-DANE opportunistic TLS (possibly even cleartext).

TLSA record publishers may leverage CNAMEs to reference a single authoritative TLSA RRset specifying a common CA or a common end-entity certificate to be used with multiple TLS services. Such CNAME expansion does not change the SMTP client's notion of the TLSA base domain; thus, when `_25._tcp.mx.example.com` is a CNAME, the base

domain remains `mx.example.com`, and this is still the reference identifier used together with the next-hop domain in peer certificate name checks.

Note that shared end-entity certificate associations expose the publishing domain to substitution attacks, where an MITM attacker can reroute traffic to a different server that shares the same end-entity certificate. Such shared end-entity TLSA records **SHOULD** be avoided unless the servers in question are functionally equivalent or employ mutually incompatible protocols (an active attacker gains nothing by diverting client traffic from one such server to another).

A better example, employing a shared trust anchor rather than shared end-entity certificates, is illustrated by the DNSSEC-validated records below:

```
example.com.           IN MX 0 mx1.example.com.
example.com.           IN MX 0 mx2.example.com.
_25._tcp.mx1.example.com. IN CNAME tlsa201._dane.example.com.
_25._tcp.mx2.example.com. IN CNAME tlsa201._dane.example.com.
tlsa201._dane.example.com. IN TLSA 2 0 1 e3b0c44298fc1c149a...
```

The SMTP servers `mx1.example.com` and `mx2.example.com` will be expected to have certificates issued under a common trust anchor, but each MX hostname's TLSA base domain remains unchanged despite the above CNAME records. Correspondingly, each SMTP server will be associated with a pair of reference identifiers consisting of its hostname plus the next-hop domain "`example.com`".

If, during TLSA resolution (including possible CNAME indirection), at least one "secure" TLSA record is found (even if not usable because it is unsupported by the implementation or support is administratively disabled), then the corresponding host has signaled its commitment to implement TLS. The SMTP client **MUST NOT** deliver mail via the corresponding host unless a TLS session is negotiated via STARTTLS. This is required to avoid MITM STARTTLS downgrade attacks.

As noted previously (in Section 2.2.2), when no "secure" TLSA records are found at the fully CNAME-expanded name, the original unexpanded name **MUST** be tried instead. This supports customers of hosting providers where the provider's zone cannot be validated with DNSSEC but the customer has shared appropriate key material with the hosting provider to enable TLS via SNI. Intermediate names that arise during CNAME expansion that are neither the original name nor the final name are never candidate TLSA base domains, even if "secure".

### 3. DANE Authentication

This section describes which TLSA records are applicable to SMTP opportunistic DANE TLS and how to apply such records to authenticate the SMTP server. With opportunistic DANE TLS, both the TLS support implied by the presence of DANE TLSA records and the verification parameters necessary to authenticate the TLS peer are obtained together. In contrast to protocols where channel security policy is set exclusively by the client, authentication via this protocol is expected to be less prone to connection failure caused by incompatible configuration of the client and server.

#### 3.1. TLSA Certificate Usages

The DANE TLSA specification [RFC6698] defines multiple TLSA RR types via combinations of three numeric parameters. The numeric values of these parameters were later given symbolic names in [RFC7218]. The rest of the TLSA record is the "certificate association data field", which specifies the full or digest value of a certificate or public key.

Since opportunistic DANE TLS will be used by non-interactive MTAs, with no user to "click OK" when authentication fails, reliability of peer authentication is paramount. Server operators are advised to publish TLSA records that are least likely to fail authentication due to interoperability or operational problems. Because DANE TLS relies on coordinated changes to DNS and SMTP server settings, the best choice of records to publish will depend on site-specific practices.

The certificate usage element of a TLSA record plays a critical role in determining how the corresponding certificate association data field is used to authenticate a server's certificate chain. Sections 3.1.1 and 3.1.2 explain the process for certificate usages DANE-EE(3) and DANE-TA(2), respectively. Section 3.1.3 briefly explains why certificate usages PKIX-TA(0) and PKIX-EE(1) are not applicable with opportunistic DANE TLS.

In summary, we RECOMMEND the use of "DANE-EE(3) SPKI(1) SHA2-256(1)", with "DANE-TA(2) Cert(0) SHA2-256(1)" TLSA records as a second choice, depending on site needs. See Sections 3.1.1 and 3.1.2 for more details. Other combinations of TLSA parameters either (1) are explicitly unsupported or (2) offer little to recommend them over these two.

### 3.1.1. Certificate Usage DANE-EE(3)

Authentication via certificate usage DANE-EE(3) TLSA records involves simply checking that the server's leaf certificate matches the TLSA record. In particular, the binding of the server public key to its name is based entirely on the TLSA record association. The server **MUST** be considered authenticated even if none of the names in the certificate match the client's reference identity for the server.

The expiration date of the server certificate **MUST** be ignored: the validity period of the TLSA record key binding is determined by the validity interval of the TLSA record DNSSEC signature.

With DANE-EE(3), servers need not employ SNI (they may ignore the client's SNI message) even when the server is known under independent names that would otherwise require separate certificates. It is instead sufficient for the TLSA RRsets for all the domains in question to match the server's default certificate. Of course, with SMTP servers it is simpler still to publish the same MX hostname for all the hosted domains.

For domains where it is practical to make coordinated changes in DNS TLSA records during SMTP server key rotation, it is often best to publish end-entity DANE-EE(3) certificate associations. DANE-EE(3) certificates don't suddenly stop working when leaf or intermediate certificates expire, nor do they fail when the server operator neglects to configure all the required issuer certificates in the server certificate chain.

TLSA records published for SMTP servers **SHOULD**, in most cases, be "DANE-EE(3) SPKI(1) SHA2-256(1)" records. Since all DANE implementations are required to support SHA2-256, this record type works for all clients and need not change across certificate renewals with the same key.

### 3.1.2. Certificate Usage DANE-TA(2)

Some domains may prefer to avoid the operational complexity of publishing unique TLSA RRs for each TLS service. If the domain employs a common issuing CA to create certificates for multiple TLS services, it may be simpler to publish the issuing authority as a trust anchor (TA) for the certificate chains of all relevant services. The TLSA query domain (TLSA base domain with port and protocol prefix labels) for each service issued by the same TA may then be set to a CNAME alias that points to a common TLSA RRset that matches the TA. For example:

```
example.com.           IN MX 0 mx1.example.com.
example.com.           IN MX 0 mx2.example.com.
_25._tcp.mx1.example.com. IN CNAME tlsa201._dane.example.com.
_25._tcp.mx2.example.com. IN CNAME tlsa201._dane.example.com.
tlsa201._dane.example.com. IN TLSA 2 0 1 e3b0c44298fc1c14....
```

With usage DANE-TA(2), the server certificates will need to have names that match one of the client's reference identifiers (see [RFC6125]). The server MAY employ SNI to select the appropriate certificate to present to the client.

SMTP servers that rely on certificate usage DANE-TA(2) TLSA records for TLS authentication MUST include the TA certificate as part of the certificate chain presented in the TLS handshake server certificate message even when it is a self-signed root certificate. Many SMTP servers are not configured with a comprehensive list of trust anchors, nor are they expected to be at any point in the future. Some MTAs will ignore all locally trusted certificates when processing usage DANE-TA(2) TLSA records. Thus, even when the TA happens to be a public CA known to the SMTP client, authentication is likely to fail unless the TA certificate is included in the TLS server certificate message.

With some SMTP server software, it is not possible to configure the server to include self-signed (root) CA certificates in the server certificate chain. Such servers either MUST publish DANE-TA(2) records for an intermediate certificate or MUST instead use DANE-EE(3) TLSA records.

TLSA records with a matching type of Full(0) are discouraged. While these potentially obviate the need to transmit the TA certificate in the TLS server certificate message, client implementations may not be able to augment the server certificate chain with the data obtained from DNS, especially when the TLSA record supplies a bare key (selector SPKI(1)). Since the server will need to transmit the TA certificate in any case, server operators SHOULD publish TLSA records

with a matching type other than Full(0) and avoid potential interoperability issues with large TLSA records containing full certificates or keys.

TLSA Publishers employing DANE-TA(2) records SHOULD publish records with a selector of Cert(0). Such TLSA records are associated with the whole trust anchor certificate, not just with the trust anchor public key. In particular, the SMTP client SHOULD then apply any relevant constraints from the trust anchor certificate, such as, for example, path length constraints.

While a selector of SPKI(1) may also be employed, the resulting TLSA record will not specify the full trust anchor certificate content, and elements of the trust anchor certificate other than the public key become mutable. This may, for example, allow a subsidiary CA to issue a chain that violates the trust anchor's path length or name constraints.

### 3.1.3. Certificate Usages PKIX-TA(0) and PKIX-EE(1)

Note that this section applies to MTA-to-MTA SMTP, which is normally on port 25 -- that is, to servers that are the SMTP servers for one or more destination domains. Other uses of SMTP, such as in MUA-to-MSA submission on ports 587 or 465, are out of scope for this document. Where those other uses also employ TLS opportunistically and/or depend on DNSSEC as a result of DNS-based discovery of service location, the relevant specifications should, as appropriate, arrive at similar conclusions.

As noted in Sections 1.3.1 and 1.3.2, sending MTAs cannot, without relying on DNSSEC for "secure" MX records and DANE for STARTTLS support signaling, perform server identity verification or prevent STARTTLS downgrade attacks. The use of PKIX CAs offers no added security, since an attacker capable of compromising DNSSEC is free to replace any PKIX-TA(0) or PKIX-EE(1) TLSA records with records bearing any convenient non-PKIX certificate usage. Finally, as explained in Section 1.3.4, there is no list of trusted CAs agreed upon by all MTAs and no user to "click OK" when a server's CA is not trusted by a client.

Therefore, TLSA records for the port 25 SMTP service used by client MTAs SHOULD NOT include TLSA RRs with certificate usage PKIX-TA(0) or PKIX-EE(1). SMTP client MTAs cannot be expected to be configured with a suitably complete set of trusted public CAs. Lacking a complete set of public CAs, MTA clients would not be able to verify the certificates of SMTP servers whose issuing root CAs are not trusted by the client.

Opportunistic DANE TLS needs to interoperate without bilateral coordination of security settings between client and server systems. Therefore, parameter choices that are fragile in the absence of bilateral coordination are unsupported. Nothing is lost; since the PKIX certificate usages cannot aid SMTP TLS security, they can only impede SMTP TLS interoperability.

SMTP client treatment of TLSA RRs with certificate usages PKIX-TA(0) or PKIX-EE(1) is undefined. As with any other unsupported certificate usage, SMTP clients MAY treat such records as "unusable".

### 3.2. Certificate Matching

When at least one usable "secure" TLSA record is found, the SMTP client MUST use TLSA records to authenticate the SMTP server. Messages MUST NOT be delivered via the SMTP server if authentication fails; otherwise, the SMTP client is vulnerable to MITM attacks.

#### 3.2.1. DANE-EE(3) Name Checks

The SMTP client MUST NOT perform certificate name checks with certificate usage DANE-EE(3) (Section 3.1.1).

#### 3.2.2. DANE-TA(2) Name Checks

To match a server via a TLSA record with certificate usage DANE-TA(2), the client MUST perform name checks to ensure that it has reached the correct server. In all DANE-TA(2) cases, the SMTP client MUST employ the TLSA base domain as the primary reference identifier for matching the server certificate.

**TLSA records for MX hostnames:** If the TLSA base domain was obtained indirectly via a "secure" MX lookup (including any CNAME-expanded name of an MX hostname), then the original next-hop domain used in the MX lookup MUST be included as a second reference identifier. The CNAME-expanded original next-hop domain MUST be included as a third reference identifier if different from the original next-hop domain. When the client MTA is employing DANE TLS security despite "insecure" MX redirection, the MX hostname is the only reference identifier.

**TLSA records for non-MX hostnames:** If MX records were not used (e.g., if none exist) and the TLSA base domain is the CNAME-expanded original next-hop domain, then the original next-hop domain MUST be included as a second reference identifier.



Accepting certificates with the original next-hop domain in addition to the MX hostname allows a domain with multiple MX hostnames to field a single certificate bearing a single domain name (i.e., the email domain) across all the SMTP servers. This also aids interoperability with pre-DANE SMTP clients that are configured to look for the email domain name in server certificates -- for example, with "secure" DNS records as shown below:

```

exchange.example.org.      IN CNAME mail.example.org.
mail.example.org.          IN CNAME example.com.
example.com.               IN MX      10 mx10.example.com.
example.com.               IN MX      15 mx15.example.com.
example.com.               IN MX      20 mx20.example.com.
;
mx10.example.com.          IN A       192.0.2.10
_25._tcp.mx10.example.com. IN TLSA    2 0 1 ...
;
mx15.example.com.          IN CNAME mxbackup.example.com.
mxbackup.example.com.      IN A       192.0.2.15
;_25._tcp.mxbackup.example.com. IN TLSA ? (NXDOMAIN)
_25._tcp.mx15.example.com. IN TLSA    2 0 1 ...
;
mx20.example.com.          IN CNAME mxbackup.example.net.
mxbackup.example.net.      IN A       198.51.100.20
_25._tcp.mxbackup.example.net. IN TLSA    2 0 1 ...

```

Certificate name checks for delivery of mail to exchange.example.org via any of the associated SMTP servers MUST accept at least the names "exchange.example.org" and "example.com", which are, respectively, the original and fully expanded next-hop domain. When the SMTP server is mx10.example.com, name checks MUST accept the TLSA base domain "mx10.example.com". If, despite the fact that MX hostnames are required to not be aliases, the MTA supports delivery via "mx15.example.com" or "mx20.example.com", then name checks MUST accept the respective TLSA base domains "mx15.example.com" and "mxbackup.example.net".

### 3.2.3. Reference Identifier Matching

When name checks are applicable (certificate usage DANE-TA(2)), if the server certificate contains a Subject Alternative Name extension [RFC5280] with at least one DNS-ID [RFC6125], then only the DNS-IDs are matched against the client's reference identifiers. The CN-ID [RFC6125] is only considered when no DNS-IDs are present. The server certificate is considered matched when one of its presented identifiers [RFC5280] matches any of the client's reference identifiers.

Wildcards are valid in either DNS-IDs or the CN-ID when applicable. The wildcard character must be the entire first label of the DNS-ID or CN-ID. Thus, "\*.example.com" is valid, while "smtp\*.example.com" and "\*smtp.example.com" are not. SMTP clients **MUST** support wildcards that match the first label of the reference identifier, with the remaining labels matching verbatim. For example, the DNS-ID "\*.example.com" matches the reference identifier "mx1.example.com". SMTP clients **MAY**, subject to local policy, allow wildcards to match multiple reference identifier labels, but servers cannot expect broad support for such a policy. Therefore, any wildcards in server certificates **SHOULD** match exactly one label in either the TLSA base domain or the next-hop domain.

#### 4. Server Key Management

Two TLSA records **MUST** be published before employing a new EE or TA public key or certificate: one matching the currently deployed key and the other matching the new key scheduled to replace it. Once sufficient time has elapsed for all DNS caches to expire the previous TLSA RRset and related signature RRsets, servers may be configured to use the new EE private key and associated public key certificate or may employ certificates signed by the new trust anchor.

Once the new public key or certificate is in use, the TLSA RR that matches the retired key can be removed from DNS, leaving only RRs that match keys or certificates in active use.

As described in Section 3.1.2, when server certificates are validated via a DANE-TA(2) trust anchor and CNAME records are employed to store the TA association data at a single location, the responsibility of updating the TLSA RRset shifts to the operator of the trust anchor. Before a new trust anchor is used to sign any new server certificates, its certificate (digest) is added to the relevant TLSA RRset. After enough time elapses for the original TLSA RRset to age out of DNS caches, the new trust anchor can start issuing new server certificates. Once all certificates issued under the previous trust anchor have expired, its associated RRs can be removed from the TLSA RRset.

In the DANE-TA(2) key management model, server operators do not generally need to update DNS TLSA records after initially creating a CNAME record that references the centrally operated DANE-TA(2) RRset. If a particular server's key is compromised, its TLSA CNAME **SHOULD** be replaced with a DANE-EE(3) association until the certificate for the compromised key expires, at which point it can return to using a CNAME record. If the central trust anchor is compromised, all

servers need to be issued new keys by a new TA, and an updated DANE-TA(2) TLSA RRset needs to be published containing just the new TA.

SMTP servers cannot expect broad Certificate Revocation List (CRL) or Online Certificate Status Protocol (OCSP) support from SMTP clients. As outlined above, with DANE, compromised server or trust anchor keys can be "revoked" by removing them from the DNS without the need for client-side support for OCSP or CRLs.

## 5. Digest Algorithm Agility

While [RFC6698] specifies multiple digest algorithms, it does not specify a protocol by which the SMTP client and TLSA record publisher can agree on the strongest shared algorithm. Such a protocol would allow the client and server to avoid exposure to deprecated weaker algorithms that are published for compatibility with less capable clients. When stronger algorithms are an option, deprecated algorithms SHOULD be avoided. Such a protocol is specified in [RFC7671]. SMTP clients and servers that implement this specification MUST comply with the requirements outlined in Section 9 of [RFC7671].

## 6. Mandatory TLS Security

An MTA implementing this protocol may require a stronger security assurance when sending email to selected destinations. The sending organization may need to send sensitive email and/or may have regulatory obligations to protect its content. This protocol is not in conflict with such a requirement and, in fact, can often simplify authenticated delivery to such destinations.

Specifically, with domains that publish DANE TLSA records for their MX hostnames, a sending MTA can be configured to use the receiving domain's DANE TLSA records to authenticate the corresponding SMTP server. Authentication via DANE TLSA records is easier to manage, as changes in the receiver's expected certificate properties are made on the receiver end and don't require manually communicated configuration changes. With mandatory DANE TLS, when no usable TLSA records are found, message delivery is delayed. Thus, mail is only sent when an authenticated TLS channel is established to the remote SMTP server.

Administrators of mail servers that employ mandatory DANE TLS need to carefully monitor their mail logs and queues. If a partner domain unwittingly misconfigures its TLSA records, disables DNSSEC, or misconfigures SMTP server certificate chains, mail will be delayed and may bounce if the issue is not resolved in a timely manner.

## 7. Note on DANE for Message User Agents

We note that SMTP is also used between Message User Agents (MUAs) and Message Submission Agents (MSAs) [RFC6409]. In [RFC6186], a protocol is specified that enables an MUA to dynamically locate the MSA based on the user's email address. SMTP connection security considerations for MUAs implementing [RFC6186] are largely analogous to connection security requirements for MTAs, and this specification could be applied largely verbatim with DNS MX records replaced by corresponding DNS Service (SRV) records [RFC7673].

However, until MUAs begin to adopt the dynamic configuration mechanisms of [RFC6186], they are adequately served by more traditional static TLS security policies. Specification of DANE TLS for MUA-to-MSA SMTP is left to future documents that focus specifically on SMTP security between MUAs and MSAs.

## 8. Interoperability Considerations

### 8.1. SNI Support

To ensure that the server sends the right certificate chain, the SMTP client **MUST** send the TLS SNI extension containing the TLSA base domain. This precludes the use of the Secure Socket Layer (SSL) HELLO that is SSL 2.0 compatible by the SMTP client.

Each SMTP server **MUST** present a certificate chain (see [RFC5246], Section 7.4.2) that matches at least one of the TLSA records. The server **MAY** rely on SNI to determine which certificate chain to present to the client. Clients that don't send SNI information may not see the expected certificate chain.

If the server's TLSA records match the server's default certificate chain, the server need not support SNI. In either case, the server need not include the SNI extension in its TLS HELLO, as simply returning a matching certificate chain is sufficient. Servers **MUST NOT** enforce the use of SNI by clients, as the client may be using unauthenticated opportunistic TLS and may not expect any particular certificate from the server. If the client sends no SNI extension or sends an SNI extension for an unsupported domain, the server **MUST** simply send some fallback certificate chain of its choice. The reason for not enforcing strict matching of the requested SNI hostname is that DANE TLS clients are typically willing to accept multiple server names but can only send one name in the SNI extension. The server's fallback certificate may match a different name acceptable to the client, e.g., the original next-hop domain.

## 8.2. Anonymous TLS Cipher Suites

Since many SMTP servers either do not support or do not enable any anonymous TLS cipher suites, SMTP client TLS HELLO messages SHOULD offer to negotiate a typical set of non-anonymous cipher suites required for interoperability with such servers. An SMTP client employing pre-DANE opportunistic TLS MAY also include one or more anonymous TLS cipher suites in its TLS HELLO. SMTP servers that need to interoperate with opportunistic TLS clients SHOULD be prepared to interoperate with such clients by either always selecting a mutually supported non-anonymous cipher suite or correctly handling client connections that negotiate anonymous cipher suites.

Note that while SMTP server operators are under no obligation to enable anonymous cipher suites, no security is gained by sending certificates to clients that will ignore them. Indeed, support for anonymous cipher suites in the server makes audit trails more informative. Log entries that record connections that employed an anonymous cipher suite record the fact that the clients did not care to authenticate the server.

## 9. Operational Considerations

### 9.1. Client Operational Considerations

An operational error on the sending or receiving side that cannot be corrected in a timely manner may, at times, lead to consistent failure to deliver time-sensitive email. The sending MTA administrator may have to choose between allowing email to queue until the error is resolved and disabling opportunistic or mandatory DANE TLS (Section 6) for one or more destinations. The choice to disable DANE TLS security should not be made lightly. Every reasonable effort should be made to determine that problems with mail delivery are the result of an operational error and not an attack. A fallback strategy may be to configure explicit out-of-band TLS security settings if supported by the sending MTA.

SMTP clients may deploy opportunistic DANE TLS incrementally by enabling it only for selected sites or may occasionally need to disable opportunistic DANE TLS for peers that fail to interoperate due to misconfiguration or software defects on either end. Some implementations MAY support DANE TLS in an "audit only" mode in which failure to achieve the requisite security level is logged as a warning and delivery proceeds at a reduced security level. Unless local policy specifies "audit only" or specifies that opportunistic DANE TLS is not to be used for a particular destination, an SMTP

client **MUST NOT** deliver mail via a server whose certificate chain fails to match at least one TLSA record when usable TLSA records are found for that server.

## 9.2. Publisher Operational Considerations

Some MTAs enable STARTTLS selectively. For example, they might only support STARTTLS with clients that have previously demonstrated "proper MTA behavior", e.g., by retrying the delivery of deferred messages (greylisting). If such an MTA publishes DANE TLSA records, sending MTAs that implement this specification will not attempt the initial cleartext SMTP transaction needed to establish the "proper MTA behavior", because they cannot establish the required channel security. Server operators **MUST NOT** implement selective STARTTLS if they also want to support DANE TLSA.

TLSA Publishers **MUST** follow the guidelines in Section 8 of [RFC7671].

TLSA Publishers **SHOULD** follow the TLSA publication size guidance found in Section 10.1 of [RFC7671].

TLSA Publishers **SHOULD** follow the TLSA record TTL and signature lifetime recommendations found in Section 13 of [RFC7671].

## 10. Security Considerations

This protocol leverages DANE TLSA records to implement MITM-resistant Opportunistic Security [RFC7435] for SMTP. For destination domains that sign their MX records and publish signed TLSA records for their MX hostnames, this protocol allows sending MTAs to securely discover both the availability of TLS and how to authenticate the destination.

This protocol does not aim to secure all SMTP traffic, as that is not practical until DNSSEC and DANE adoption are universal. The incremental deployment provided by following this specification is a best possible path for securing SMTP. This protocol coexists and interoperates with the existing insecure Internet email backbone.

The protocol does not preclude existing non-opportunistic SMTP TLS security arrangements, which can continue to be used as before via manual configuration with negotiated out-of-band key and TLS configuration exchanges.

Opportunistic SMTP TLS depends critically on DNSSEC for downgrade resistance and secure resolution of the destination name. If DNSSEC is compromised, it is not possible to fall back on the public CA PKI to prevent MITM attacks. A successful breach of DNSSEC enables the attacker to publish TLSA usage 3 certificate associations and thereby

bypass any security benefit the legitimate domain owner might hope to gain by publishing usage 0 or usage 1 TLSA RRs. Given the lack of public CA PKI support in existing MTA deployments, avoiding certificate usages 0 and 1 simplifies implementation and deployment with no adverse security consequences.

Implementations must strictly follow Sections 2.1.2, 2.1.3, 2.2, 2.2.1, 2.2.2, 2.2.3, 3.2, and 9.1 of this specification; these sections indicate when it is appropriate to initiate a non-authenticated connection or cleartext connection to an SMTP server. Specifically, in order to prevent downgrade attacks on this protocol, implementations must not initiate a connection when this specification indicates that a particular SMTP server must be considered unreachable.

## 11. References

### 11.1. Normative References

- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <<http://www.rfc-editor.org/info/rfc1034>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC3207] Hoffman, P., "SMTP Service Extension for Secure SMTP over Transport Layer Security", RFC 3207, DOI 10.17487/RFC3207, February 2002, <<http://www.rfc-editor.org/info/rfc3207>>.
- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <<http://www.rfc-editor.org/info/rfc4033>>.
- [RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, DOI 10.17487/RFC4034, March 2005, <<http://www.rfc-editor.org/info/rfc4034>>.
- [RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, DOI 10.17487/RFC4035, March 2005, <<http://www.rfc-editor.org/info/rfc4035>>.

- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <<http://www.rfc-editor.org/info/rfc5246>>.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <<http://www.rfc-editor.org/info/rfc5280>>.
- [RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, DOI 10.17487/RFC5321, October 2008, <<http://www.rfc-editor.org/info/rfc5321>>.
- [RFC5598] Crocker, D., "Internet Mail Architecture", RFC 5598, DOI 10.17487/RFC5598, July 2009, <<http://www.rfc-editor.org/info/rfc5598>>.
- [RFC6066] Eastlake 3rd, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, DOI 10.17487/RFC6066, January 2011, <<http://www.rfc-editor.org/info/rfc6066>>.
- [RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, DOI 10.17487/RFC6125, March 2011, <<http://www.rfc-editor.org/info/rfc6125>>.
- [RFC6186] Daboo, C., "Use of SRV Records for Locating Email Submission/Access Services", RFC 6186, DOI 10.17487/RFC6186, March 2011, <<http://www.rfc-editor.org/info/rfc6186>>.
- [RFC6672] Rose, S. and W. Wijngaards, "DNAME Redirection in the DNS", RFC 6672, DOI 10.17487/RFC6672, June 2012, <<http://www.rfc-editor.org/info/rfc6672>>.
- [RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, DOI 10.17487/RFC6698, August 2012, <<http://www.rfc-editor.org/info/rfc6698>>.



- [RFC7218] Gudmundsson, O., "Adding Acronyms to Simplify Conversations about DNS-Based Authentication of Named Entities (DANE)", RFC 7218, DOI 10.17487/RFC7218, April 2014, <<http://www.rfc-editor.org/info/rfc7218>>.
- [RFC7671] Dukhovni, V. and W. Hardaker, "The DNS-Based Authentication of Named Entities (DANE) Protocol: Updates and Operational Guidance", RFC 7671, DOI 10.17487/RFC7671, October 2015, <<http://www.rfc-editor.org/info/rfc7671>>.

## 11.2. Informative References

- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <<http://www.rfc-editor.org/info/rfc1035>>.
- [RFC2136] Vixie, P., Ed., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, DOI 10.17487/RFC2136, April 1997, <<http://www.rfc-editor.org/info/rfc2136>>.
- [RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", RFC 2181, DOI 10.17487/RFC2181, July 1997, <<http://www.rfc-editor.org/info/rfc2181>>.
- [RFC4949] Shirey, R., "Internet Security Glossary, Version 2", FYI 36, RFC 4949, DOI 10.17487/RFC4949, August 2007, <<http://www.rfc-editor.org/info/rfc4949>>.
- [RFC6409] Gellens, R. and J. Klensin, "Message Submission for Mail", STD 72, RFC 6409, DOI 10.17487/RFC6409, November 2011, <<http://www.rfc-editor.org/info/rfc6409>>.
- [RFC7435] Dukhovni, V., "Opportunistic Security: Some Protection Most of the Time", RFC 7435, DOI 10.17487/RFC7435, December 2014, <<http://www.rfc-editor.org/info/rfc7435>>.
- [RFC7673] Finch, T., Miller, M., and P. Saint-Andre, "Using DNS-Based Authentication of Named Entities (DANE) TLSA Records with SRV Records", RFC 7673, DOI 10.17487/RFC7673, October 2015, <<http://www.rfc-editor.org/info/rfc7673>>.

## Acknowledgements

The authors would like to extend great thanks to Tony Finch, who started the original version of a DANE SMTP document. His work is greatly appreciated and has been incorporated into this document. The authors would like to additionally thank Phil Pennock for his comments and advice on this document.

Acknowledgements from Viktor: Thanks to Paul Hoffman, who motivated me to begin work on this memo and provided feedback on early draft versions of this document. Thanks to Patrick Koetter, Perry Metzger, and Nico Williams for valuable review comments. Thanks also to Wietse Venema, who created Postfix, and whose advice and feedback were essential to the development of the Postfix DANE implementation.

## Authors' Addresses

Viktor Dukhovni  
Two Sigma

Email: [ietf-dane@dukhovni.org](mailto:ietf-dane@dukhovni.org)

Wes Hardaker  
Parsons  
P.O. Box 382  
Davis, CA 95617  
United States

Email: [ietf@hardakers.net](mailto:ietf@hardakers.net)