

DANE  
Internet-Draft  
Intended status: Best Current Practice  
Expires: January 16, 2014

V. Dukhovni  
Unaffiliated  
W. Hardaker  
Parsons  
July 15, 2013

DANE TLSA implementation and operational guidance  
draft-dukhovni-dane-ops-01

## Abstract

This memo provides operational guidance to server operators to help ensure that clients will be able to authenticate a server's certificate chain via published TLSA records. Guidance is also provided to clients for selecting reliable TLSA record parameters to use for server authentication. Finally, guidance is given to protocol designers who wish to make use of TLSA records to secure protocols using a TLS and TLSA combination.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 16, 2014.

## Copyright Notice

Copyright (c) 2013 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 . . . . .	2
1.1. Terminology . . . . .	3
2. DANE TLSA record overview . . . . .	4
2.1. Example TLSA record . . . . .	5
3. General DANE Guidelines . . . . .	6
3.1. TLS Requirements . . . . .	6
3.2. DANE DNS Record Size Guidelines . . . . .	6
3.3. Certificate Name Check Conventions . . . . .	7
3.4. Service Provider and TLSA Publisher Synchronization . . . . .	7
3.5. TLSA Base Domain and CNAMEs . . . . .	8
3.6. TLSA Base Name Priorities . . . . .	9
3.7. Interaction with Certificate Transparency . . . . .	9
3.8. Design Considerations for Protocols Using DANE . . . . .	10
3.9. TLSA Records and Trust Anchor Digests . . . . .	12
3.10. Trust anchor public keys . . . . .	13
4. Type Specific DANE Guidelines . . . . .	14
4.1. Type 3 Guidelines . . . . .	14
4.2. Type 2 Guidelines . . . . .	14
4.3. Type 1 Guidelines . . . . .	14
4.4. Type 0 Guidelines . . . . .	14
5. Note on DNSSEC security . . . . .	15
6. Acknowledgements . . . . .	16
7. Security Considerations . . . . .	16
8. References . . . . .	17
8.1. Normative References . . . . .	17
8.2. Informative References . . . . .	18
Authors' Addresses . . . . .	18

## 1. Introduction

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

In the context of this memo, channel security is assumed to be provided by TLS or DTLS. The Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) protocols provide secured TCP and UDP communication over the Internet Protocol. By convention, "TLS" will be used through this document and, unless otherwise specified, the text applies equally as well to the DTLS protocol. Used without authentication, TLS provides protection only against eavesdropping. With authentication, TLS also provides protection

against man-in-the-middle (MITM) attacks. Since the publication of the TLS 1.0 specification in [RFC2246], two updates to the protocol have been published: TLS 1.1 [RFC4346] and TLS 1.2 [RFC5246]. The DTLS protocol was later documented in [RFC6347].

As described in the introduction of [RFC6698], TLS authentication via the existing public Certificate Authority (CA) Public Key Infrastructure (PKI) suffers from an over-abundance of trusted certificate authorities capable of issuing certificates for any domain of their choice. DNS-Based Authentication of Named Entities (DANE) leverages the DNSSEC infrastructure to publish trusted keys and certificates for use with TLS via a new TLSA record type. DNSSEC validated DANE TLSA records have created a new PKI designed to augment or replace the trust model of the existing public CA PKI.

When a TLS client goes to the trouble of authenticating a certificate presented by a TLS server, it should not continue to use the server in case of authentication failure or else authentication serves no purpose. Consequently, if a client cannot reliably authenticate correctly configured, legitimate servers via a particular combination of TLSA parameters, then the client should treat that combination of parameters as unusable. Otherwise, the client risks routinely dropping connections to legitimate servers. Servers publishing TLSA records **MUST** be configured to allow correctly configured clients to successfully authenticate the server's TLS certificate.

If a TLSA record is found as unusable because of a parameter combination, it is protocol specific as to whether the connection should be established anyway without security, with only TLS encryption and not authentication, or to refuse to connect entirely.

### 1.1. Terminology

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

This memo is being discussed on the dane@ietf.org mailing list.

The following terms are used throughout this document:

**Service Provider:** A company or organization that offers to host a service on behalf of a Client Domain. The original domain name associated with the service is typically still within the control of the client, however, and the service provider is frequently referred to by a redirection resource record. Example redirection records include MX, SRV, and CNAME. Many times the Service Provider provides services for many customers and must carefully

manage TLS credentials offered to their clients to ensure name matching is handled easily by clients.

Client Domain: Clients that make use of a Service Provider to outsource their services will be referred to as "Client Domains".

TLSA Publisher: The entity responsible for publishing a TLSA record within a DNS zone. This zone will be considered DNSSEC signed, unless otherwise specified. If the Client Domain is not outsourcing their DNS service, the TLSA Publisher will be the client themselves. Otherwise the TLSA Publisher may be the outsourced DNS service instead.

public key: The term "public key" will be an informal short-hand for the subjectPublicKeyInfo from a PKIX certificate.

SNI: The "Server Name Indication", or SNI, describes the process by which a TLS client requests to connect to a particular service name for a TLS server ([RFC3546]). Without this TLS extension, a TLS server has no choice but to offer a PKIX certificate with a default server name. Service Providers that are expected to host services for many clients need to present the correct certificate for the correct client, and the SNI extension provides a hint to the server which certificate should be transmitted to the client.

## 2. DANE TLSA record overview

[RFC6698] specifies a protocol for publishing TLS server certificate associations via DNSSEC. The DANE TLSA specification defines multiple TLSA RR types via combinations of the following 3 parameters:

- o The TLSA Certificate Usage field. Section 2.1.1 of [RFC6698] specifies 4 values ranging from 0 to 3.
- o The selector field. Section 2.1.2 of [RFC6698] specifies 2 values ranging from 0 to 1.
- o The matching type field. Section 2.1.3 of [RFC6698] specifies 3 values ranging from 0 to 2.

We may consider the TLSA Certificate Usage values 0 through 3 to be a combination of two one-bit flags. The low-bit chooses between referencing trust-anchor (TA) and end-entity (EE) certificates. The high bit chooses between public PKI issued and domain-issued certificates:

- o When the low bit is set (TLSA Certificate Usages 1 and 3) the TLSA record matches an EE (server) certificate.
- o When the low bit is not set (TLSA Certificate Usages 0 and 2) the TLSA record matches a trust-anchor (a certificate authority) that issued a certificate somewhere in the certificate chain that authenticates the final end-entity certificate.
- o When the high bit is set (TLSA Certificate Usages 2 and 3) the server certificate chain is domain-issued and may be verified without reference to the existing public certificate authority PKI. Trust is entirely placed on the content of the TLSA records obtained via DNSSEC.
- o When the high bit is not set (TLSA Certificate Usages 0 and 1) the TLSA record publishes a server policy stating that its certificate chain must pass PKIX validation [RFC5280], with the DANE TLSA record used to constrain the server certificate chain to contain the referenced CA or EE certificate.

The selector field specifies whether the TLSA RR matches the whole certificate or just its subjectPublicKeyInfo (i.e. an ASN.1 DER encoding of the certificate's algorithm id, any parameters and the public key data). A selector field of "0" specifies the whole certificate. A selector field of "1" specifies just the public key.

The matching type field specifies how the TLSA RR Certificate Association Data field is to be compared with the certificate or public key. A value of "0" means exact match, the DER encoding of the certificate or public key is given in the TLSA RR. A "1" value means a SHA-256 digest and "2" means a SHA-512 digest. Of these, only SHA-256 is mandatory to implement. Clients SHOULD implement SHA-512, but servers SHOULD NOT exclusively publish SHA-512 digests. Unless a "second preimage" attack is found against SHA-256, servers should only publish SHA-256 digests.

## 2.1. Example TLSA record

In the example TLSA record below:

```
_25._tcp.mail.example.com. IN TLSA 3 0 1 (  
    E8B54E0B4BAA815B06D3462D65FBC7C0  
    CF556ECCF9F5303EBFBB77D022F834C0 )
```

The TLSA Certificate Usage is "3", the selector is "0" and the matching type is "1". The rest of the record is the certificate association data field, which is in this case the SHA-256 digest of the server certificate.

### 3. General DANE Guidelines

These guidelines provide guidance for using or designing protocols for DANE, regardless of what type the TLSA record will actually contain.

#### 3.1. TLS Requirements

TLS clients that support DANE/TLSA MUST support at least TLS 1.0 and SHOULD support TLS 1.2. TLS clients and servers using DANE SHOULD support the "Server Name Indication" extension of TLS.

#### 3.2. DANE DNS Record Size Guidelines

Selecting a combination of TLSA parameters to use requires careful thought. One important consideration is the size of the resulting TLSA record based on the parameters chosen.

##### 3.2.1. UDP and TCP Considerations

Deployments SHOULD avoid TLSA record sizes that cause UDP fragmentation.

Although DNS over TCP would provide the ability to transfer larger DNS records between clients and servers, it is not yet widely deployed or permitted through many firewalls. TCP must be expected to be deployed on all the DNS servers and DNS clients for it to be a truly viable large-record solution.

##### 3.2.2. Packet Size Considerations for TLSA Parameters

Server operators SHOULD NOT publish "TLSA \* 0 0" records, as even a single certificate is generally too large to be reliably delivered via DNS without TCP being widely available. Furthermore, two full certificates may need to be published in the TLSA RRset for certificate rollover.

While "TLSA \* 1 0" records, which publish full public keys without the full X.509 wrapping, are generally more compact, these too should be used with caution. Servers SHOULD publish digests within TLSA records instead. The complete certificate should, instead, be transmitted to the client in band during the TLS handshake.

### 3.3. Certificate Name Check Conventions

Certificates presented by a TLS server will contain either a Common Name (CN) or subjectAltName (or both), according to [RFC5280]. The server's hostname should be published within these fields, ideally within the subjectAltName. This section discusses what must be done to match an expected name against the name found within a certificate, if required.

The TLSA Publisher for TLSA records for a given service MUST ensure that at least one of these TLSA records will match the server's certificate chain. If SNI is not employed for a TLS connection, the TLSA record must match the server's default certificate. If the SNI extension is sent by the client with a host\_name (see [RFC3546] Section 3.1) equal to the base domain of the TLSA RRset, at least one TLSA record must match the certificate presented by the server for that host\_name.

When, for example, the TLSA RRset is published at

`_25._tcp.mail.example.com`

the TLSA base domain is mail.example.com. At least one of the TLSA records in the `_25._tcp.mail.example.com` RRset MUST match the server certificate chain, provided the client TLS handshake included the SNI extension with a host\_name of "mail.example.com".

Note: Except with TLSA Certificate Usage "3", where name checks are not applicable (see Section 4.1), DANE aware clients SHOULD use the base domain of the TLSA RRset to verify that the client has reached the correct server by checking that the TLSA base domain is matched by one of the subjectAltName ([RFC5280]) in the server certificate. The commonName from the certificate subject DN MAY be used only when no subjectAltNames of type 'dns' are present. Additional acceptable names may be specified by protocol specific DANE RFCs. For example, with SMTP both the destination domain name and the MX host name are acceptable in the server certificate.

Since the server's ability to respond with the right certificate chain requires the TLS client to provide the correct SNI information, DANE PKI aware clients SHOULD send the SNI extension with a host\_name value of the base domain of the TLSA RRset (otherwise they risk failure to authenticate the server).

### 3.4. Service Provider and TLSA Publisher Synchronization

Complications arise when the TLSA Publisher is not the same entity as the Service Provider. In this situation, the TLSA Publisher and the

Service Provider must cooperate to ensure that TLSA records don't fall out of sync with the server certificate configuration.

Ideally, the TLSA Publisher and the Service Provider should be the same entity. If a TLSA record must be published in the client's base domain, CNAME records can be easily used to point at the real TLSA record in the Service Provider's zone assuming certificate usage 3. TLSA records are published by the Service Provider (see Section 3.5). Having the master TLSA record in the Service Provider's zone avoids the complexity of bilateral coordination of server certificate configuration and TLSA record management.

For example, with SMTP, the customer's MX records can be pointed at the Service Provider's MX hosts. When the customer's DNS zone is signed, the MX records can be securely used as the base names for TLSA records managed by the Service Provider.

### 3.5. TLSA Base Domain and CNAMEs

When the protocol does not support service location indirection via MX, SRV or similar DNS records, the service may be redirected via a CNAME. A CNAME is a more blunt instrument for this purpose, since unlike an MX or SRV record, it remaps the origin domain to the target domain for all protocols. Also Unlike MX or SRV records, CNAME records may chain (though clients will generally impose implementation dependent maximum nesting depths).

When CNAMEs are employed, the best place to seek DANE TLSA records is in the Service Provider's domain, as discussed in Section 3.4. Therefore, DANE PKI clients connecting to a server whose domain name is a CNAME alias SHOULD follow the CNAME hop-by-hop to its ultimate target host (noting at each step whether the CNAME is DNSSEC validated) and use the resulting target host as the base domain for TLSA lookups. Standards defining how to use DANE anchored TLS for each application protocol are expected to specify where to locate TLSA RRs when the destination is referred to by a CNAME.

If CNAMEs were not followed, Client Domains would need to publish TLSA records that match the Service Provider's certificate chain or always use an entity that was both the Service Provider and the TLSA publisher. Having the TLSA base domain be different than the Service Provider's domain imposes a difficult key management burden on the Client Domain and the Service Provider.

It is possible to publish CNAMEs in the Client Domain pointing to the Service Provider's TLSA RRset if the TLSA certificate usage field is set to 3. Otherwise, a client that used the alias name (from the hosted domain rather than the Service Provider's domain) as the base



domain to obtain the TLSA RRset would look for the hosted domain in the server certificate when performing name checks, and would generally fail to authenticate the server except in the rare cases when the server's certificate does include the Client Domain. SNI SHOULD be used to help perform the right certificate selection by the server, although this imposes a management burden on the TLS server that could be avoided by ensuring the TLSA base domain is within the Service Provider's control in the first place.

Example CNAME record for a TLSA domain:

```
; TLSA RRs aliased to Service Provider, but the base domain is
; the hosted domain. Likely to fail name check unless Service
; Provider usage is "3".
;
_25._tcp.mail.example.com. IN CNAME _25._tcp.mail.example.net.
_25._tcp.mail.example.net. IN TLSA 3 1 1 ...
```

Note: when the TLSA RRset query domain (base domain plus port and protocol prefixes) resolves to a DNSSEC validated CNAME that points to a DNSSEC signed zone with the actual TLSA records, as the above example indicates, it has no effect on the value of the base domain, which remains the original domain to which the client prefixed the port and protocol. In the example above, the base domain is "mail.example.com" and not "mail.example.net".

Though CNAMEs are illegal on the right hand side of most indirection records, such as MX and SRV records, they are supported by some implementations. In this case, if the MX or SRV host is a CNAME alias the client MAY "chase" the CNAME and SHOULD use the target hostname as the base domain for TLSA records as well as the host\_name in SNI, provided the CNAME RR is found to be "secure" at each step in the CNAME expansion.

### 3.6. TLSA Base Name Priorities

There are multiple steps within a chaining DNS lookup process that TLSA base names can be pulled from. This section will discuss what the preferred selection points are. TBD.

1. Final Domain Name
2. Redirect Name
3. Initial Name

### 3.7. Interaction with Certificate Transparency

[RFC6962] Certificate Transparency or CT for short, defines an approach to mitigate the risk of rogue or compromised public CAs issuing unauthorized certificates. This section clarifies the interaction of CT and DANE. CT is a protocol and auditing system that applies only to public CAs, and only when they are free to issue unauthorized certificates for a domain. If the CA is not a public CA, or DANE TLSA RRs constrain the end-entity certificate to a fixed public key, there is no role for CT, and clients SHOULD NOT apply CT checks.

When a server is authenticated via a DANE TLSA RR with TLSA Certificate Usage "1" or "3" (that is an end-entity certificate association), the domain owner has unambiguously specified the certificate associated with the given service. Even if a rogue CA were able to issue an unauthorized end-entity certificate that binds a public key to a name in that domain, barring "second preimage" attacks on the hashing algorithms in use, any such certificate would not match the TLSA record and would be rejected. Therefore, when a TLS client authenticates the TLS server via a TLSA certificate association with usage "1" or "3", CT checks SHOULD NOT be performed. Publication of the server certificate or public key (digest) in a DNSSEC signed zone by the domain owner assures the client that the certificate is not an unauthorized certificate issued by a rogue CA without the domain owner's consent.

When a server is authenticated via a DANE TLSA RR with TLSA usage "2" and the server certificate does not chain to a known public root CA, CT cannot apply (CT logs only accept chains that start with a known root). Since TLSA Certificate Usage "2" is generally intended to support non-PKIX trust anchors, clients SHOULD NOT perform CT checks with usage "2" using unknown root CAs. A server operator that wants CT checks SHOULD publish TLSA RRs with usage "0", or can obviate them with usage "1" or "3".

CT checks remain applicable with TLSA Certificate Usage "0" when the client supports both DANE and CT and the trusted PKIX root issuer is a known public root.

### 3.8. Design Considerations for Protocols Using DANE

#### 3.8.1. Design Considerations for non-PKIX Protocols

For some application protocols, the existing public CA PKI may not be viable. For these (non-PKIX) protocols, servers SHOULD NOT suggest publishing TLSA records with TLSA Certificate Usage "0" or "1", as clients cannot be expected to perform [RFC5280] PKIX validation or [RFC6125] identity verification.

Protocols designed for non-PKIX use SHOULD choose to treat any TLSA records with TLSA Certificate Usage "0" or "1" as unusable. After verifying that the only available TLSA Certificate Usage types are "0" or "1", protocol definitions MAY instruct clients to either refuse to initiate a connection or to connect via unauthenticated mandatory TLS if no alternative authentication mechanisms are available.

If non-PKIX protocols do allow for publication of TLSA records with TLSA Certificate Usage "0" or "1", clients SHOULD make use of the TLSA verification to the fullest extent possible.

#### 3.8.1.1. TLSA Certificate Usage 1

With non-PKIX protocols, clients using TLSA Certificate Usage "1" records MAY ignore the PKIX validation requirement, and authenticate the server per the content of the TLSA record alone. Since servers will hopefully rely on SNI to select the correct certificate for presentation, the client SHOULD use the SNI extension to signal the base domain of the TLSA RRset.

#### 3.8.1.2. TLSA Certificate Usage 0

With TLSA Certificate Usage "0" in non-PKIX protocols, the usability of the TLSA records depends on its matching type.

If the matching type is "0", the TLSA record contains the full certificate or full public key of the trusted certificate authority. In this case the client has all the information it needs to match the server trust-chain to the TLSA record. The client MAY ignore the PKIX validation requirement and authenticate the server via its DANE TLSA records alone (sending SNI with the base domain as usual). The client SHOULD use the base domain of the TLSA record(s) in certificate name checks.

If the matching type is not "0", the TLSA record contains only a digest of the trust certificate authority certificate or public key. The full certificate may not be included in the server's certificate chain and the client may not be able to match the server trust chain against the TLSA record when a non-PKIX protocol is being used, as the client won't have a default CA trust list. See Section 3.9.1 for a more complete discussion of this case. The client cannot reliably authenticate the server in this case and SHOULD treat the TLSA record as unusable.

If the client is configured with a set of trusted CAs that are believed to be sufficiently complete to authenticate all the servers it expects to communicate with, then it MAY elect to honor

certificate usage "0" TLSA records that publish digests of the trusted CA certificate or public key.

### 3.9. TLSA Records and Trust Anchor Digests

With TLSA records that match the EE certificate, the TLS client has no difficulty matching the TLS record against the server certificate, as this certificate is always present in the TLS server certificate chain. The TLS client can, if necessary, extract the public key from the server certificate, and can compute the appropriate digest.

With DANE TLSA records that match the digest of a TA certificate or public key, a complication arises when the TA certificate is omitted from the server's certificate chain. This can happen when the trust-anchor is a root certificate authority, as stated in section 7.4.2 of [RFC5246]:

The sender's certificate MUST come first in the list. Each following certificate MUST directly certify the one preceding it. Because certificate validation requires that root keys be distributed independently, the self-signed certificate that specifies the root certificate authority MAY be omitted from the chain, under the assumption that the remote end must already possess it in order to validate it in any case.

This means that TLSA records that match a TA certificate or public key digest are not entirely sufficient to validate the peer certificate chain. If no matching certificate is found in the server's certificate chain, the chain may be signed by an omitted root CA whose digest matches the TLSA record. We will consider each trust-anchor TLSA Certificate Usage in turn.

#### 3.9.1. Trust Anchor Digests With TLSA Certificate Usage 0

In this case, from the server's perspective, the omission of the root CA seems reasonable, since in addition to authentication via DANE TLSA records, the client is expected to perform [RFC5280] PKIX validation of the server's trust chain and thus to already have a copy of the omitted root certificate.

From the client's perspective the situation is more nuanced. Despite the server's indicated preference for PKIX validation, the client may not possess (or may not fully trust) a complete set of public root CAs. This is especially likely in protocols where the existing public CA PKI is not applicable, as described in Section 3.8.1. If it is likely that a client lacks a sufficiently complete list of trusted CAs, and that a non-negligible number of DNS servers publish TLSA Certificate Usage 0 TLSA records with digests of omitted root

CAs, then such a client SHOULD treat such TLSA records as "unusable". Simply ignoring PKIX validation is not an option, since the client will also be unable to match the TLSA record without position of the root certificate. The client MAY choose fall back to unauthenticated TLS, if PKIX is also not an option (see [I-D.ietf-dane-srv]) or refuse to initiate a connection.

### 3.9.2. Trust Anchor Digests With TLSA Certificate Usage 2

With TLSA Certificate Usage "2", there is no expectation that the client is pre-configured with the trust anchor certificate. With TLSA Certificate Usage "2" clients are expecting to rely on the TLSA records alone. But, with a matching type other than "0" the TLSA records contain neither the full trust anchor certificate nor the full public key. If the TLS server's certificate chain does not contain the trust-anchor certificate, clients will be unable to authenticate the server.

TLSA Publishers that publish TLSA Certificate Usage "2" with a non-zero matching type MUST ensure that the corresponding server is configured to include the associated trust anchor certificate in its TLS handshake certificate chain, even if that certificate is a self-signed root CA and would have been optional in the context of the existing public CA PKI.

Since servers are expected to always provide usage "2" trust anchor certificates (either via DNS or else via the TLS handshake), clients SHOULD fully support this TLSA Certificate Usage. Clients MAY choose to treat it as unusable if experience proves that servers don't consistently live up to their obligations.

### 3.10. Trust anchor public keys

TLSA records with TLSA Certificate Usage "0" or "2", selector "1" and a matching type of "0" publish the full public key of a trust anchor via DNS. In section 6.1.1 of [RFC5280] the definition of a trust anchor consists of the following four parts:

1. the trusted issuer name,
2. the trusted public key algorithm,
3. the trusted public key, and
4. optionally, the trusted public key parameters associated with the public key.

Items 2-4 are precisely the contents of the `subjectPublicKeyInfo` published in the TLSA record, but the issuer name is not included in the public key.

With TLSA Certificate Usage "0", when the client is able to perform PKIX validation, the client can construct a complete PKIX trust chain as it will have access to the trust anchor name. So in that case, the client can verify that the server certificate chain is issued by a trust anchor that matches the TLSA record.

With TLSA Certificate Usage "2", the client may not have the missing trust anchor certificate, and cannot generally verify whether a particular certificate chain is "issued by" the trust anchor described in the TLSA record. If the server certificate chain includes a CA certificate whose public key matches the TLSA record, the client can match that CA as the intended issuer. Otherwise, the client can only check that the topmost certificate in the server's chain is "signed by" by the trust anchor public key in the TLSA record.

Since trust chain validation via bare public keys rather than trusted CA certificates may be difficult to implement using existing TLS libraries, servers SHOULD include the trust anchor certificate in their certificate chain when the TLSA Certificate Usage is "2".

If none of the server's certificate chain elements match a public key specified in full (`selector = 0`, `match type = 0`) in a TLSA record, clients SHOULD attempt to check whether the topmost certificate in the chain is signed by the provided public key, and if so consider the server trust chain valid, with authentication complete if name checks are also successful.

#### 4. Type Specific DANE Guidelines

##### 4.1. Type 3 Guidelines

##### 4.2. Type 2 Guidelines

##### 4.3. Type 1 Guidelines

##### 4.4. Type 0 Guidelines

TLSA Certificate Usage "0" allows a domain to publish constraints on the set of certificate authorities trusted to issue certificates for its TLS servers. It is expected that clients will only accept trust chains which contain a match for one of the published TLSA records. This is simple for TLSA Certificate Usage "1" where the PKIX trust chain always contains the leaf server certificate. The situation for TLSA Certificate Usage "0" is more subtle.

TLSA Publishers may publish TLSA records for a particular public root CA, expecting that clients will then only accept chains anchored at that root. It is possible, however, that the client's set of trusted certificates includes some intermediate CAs, either with or without the corresponding root CA. When a client constructs a trust chain leading from a trusted intermediate CA to the server leaf certificate, such a chain may omit any trusted roots published in the server's TLSA records.

If the omitted root is also trusted, the client may erroneously reject the server chain if it fails to determine that the shorter chain it constructed extends to a longer trusted chain that matches the TLSA records. This means that a client SHOULD not always stop extending the chain when the first locally trusted certificate is found. If no TLSA records have matched any of the elements of the chain, it MUST attempt to build a longer chain if the trusted certificate found is not self-issued, in the hope that a certificate closer to the root may in fact match the server's TLSA records.

#### 5. Note on DNSSEC security

Clearly the security of the DANE TLSA PKI rests on the security of the underlying DNSSEC infrastructure. While this memo is not a guide to DNSSEC security, a few comments may be helpful to TLSA implementors.

With the existing public CA PKI, name constraints are rarely used and public root CAs can issue certificates for any domain of its choice. With DNSSEC, the situation is different. Only the registrar of record can update a domain's DS record in the registry parent zone (in some cases, however, the registry is the sole registrar). With gTLDs, for which multiple registrars compete to provide domains in a single registry, it is important to make sure that rogue registrars cannot easily initiate an unauthorized domain transfer, and thus take over DNSSEC for the domain. DNS Operators SHOULD use a registrar lock of their domains to offer some protection of this possibility.

When the registrar is also the DNS operator for the domain, one needs to consider whether the registrar will allow orderly migration of the domain to another registrar or DNS operator in a way that will

maintain DNSSEC integrity. TLSA Publishers SHOULD ensure their registrar publishes a suitable domain transfer policy.

DNSSEC signed RRsets cannot be securely revoked before they expire. Operators should plan accordingly and not generate signatures with excessively long duration. For domains publishing high-value keys, a signature lifetime of a few days is reasonable, and the zone should be resigned every day. For more domains with less critical data, a reasonable signature lifetime is a couple of weeks to a month, and the zone should be resigned every week. Monitoring of the signature lifetime is important. If the zone is not resigned in a timely manner, one risks a major outage with the entire domain becoming invalid.

## 6. Acknowledgements

The authors would like to thank Phil Pennock for his comments and advice on this document.

Acknowledgments from Viktor: Thanks to Tony Finch who finally prodded me into participating in DANE working group discussions. Thanks to Paul Hoffman who motivated me to produce this memo and provided feedback on early drafts.

## 7. Security Considerations

Application protocols that cannot make use of the existing public CA PKI (so called non-PKIX protocols), may choose to not implement certain PKIX-dependent TLSA record types defined in [RFC6698], or may choose to make a best-effort use of such records. In neither case is security compromised, since by assumption PKIX verification is simply not an option for these protocols. When the TLS server is authenticated based on the TLSA records alone, the client is as well authenticated as possible, treating the TLSA records as unusable would lead to weaker security.

Therefore, when TLSA records are used with protocols where PKIX does not apply, the recommended trade-off is for servers to not publish PKIX-dependent TLSA records, and for clients to use them as best they can, but otherwise treat them unusable. Of course when PKIX validation is an option clients SHOULD perform PKIX validation per [RFC6698].



## 8. References

### 8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2246] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.
- [RFC3546] Blake-Wilson, S., Nystrom, M., Hopwood, D., Mikkelsen, J., and T. Wright, "Transport Layer Security (TLS) Extensions", RFC 3546, June 2003.
- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.
- [RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.
- [RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, March 2005.
- [RFC4346] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.
- [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, May 2008.
- [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, March 2011.
- [RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, January 2012.

- [RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, August 2012.
- [RFC6962] Laurie, B., Langley, A., and E. Kasper, "Certificate Transparency", RFC 6962, June 2013.

## 8.2. Informative References

- [I-D.ietf-dane-srv] Finch, T., "Using DNS-Based Authentication of Named Entities (DANE) TLSA records with SRV and MX records.", draft-ietf-dane-srv-02 (work in progress), February 2013.

## Authors' Addresses

Viktor Dukhovni  
Unaffiliated

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

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

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

DANE  
Internet-Draft  
Intended status: Experimental  
Expires: January 16, 2014

V. Dukhovni  
Unaffiliated  
W. Hardaker  
Parsons  
July 15, 2013

SMTP security via opportunistic DANE TLS  
draft-dukhovni-smtp-opportunistic-tls-01

Abstract

This memo describes a protocol for opportunistic TLS security based on the DANE TLSA DNS record. The design goal is an incremental transition of the Internet email backbone (MTA to MTA SMTP traffic) from today's unauthenticated and unencrypted connections to TLS encrypted and authenticated delivery when the client is DANE TLSA aware and the server domain publishes DANE TLSA records for its MX hosts.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 16, 2014.

Copyright Notice

Copyright (c) 2013 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 . . . . .	2
1.1. Background . . . . .	2
1.2. SMTP Channel Security . . . . .	3
1.3. Terminology . . . . .	5
2. Hardening Opportunistic TLS . . . . .	5
2.1. TLS discovery . . . . .	5
2.1.1. MX resolution . . . . .	6
2.1.2. TLSA record lookup . . . . .	7
2.2. DANE authentication . . . . .	8
2.2.1. TLSA certificate usages . . . . .	8
2.2.2. Certificate matching . . . . .	11
3. Opportunistic TLS for Submission . . . . .	13
4. Mandatory TLS Security . . . . .	14
5. Acknowledgements . . . . .	14
6. Security Considerations . . . . .	15
7. Normative References . . . . .	15
Authors' Addresses . . . . .	17

## 1. Introduction

Lacking verified DNS and "Server Name Indication" (SNI), there has historically been no scalable way for SMTP server operators to provide certificates which match a trustable identifier. It's only with the deployment of DNSSEC and DANE that authenticated TLS for SMTP to MX becomes possible between parties that have not already established an identity convention out-of-band.

### 1.1. Background

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

As described in the introduction of [RFC6698], TLS authentication via the existing public Certificate Authority (CA) Public Key Infrastructure (PKI) suffers from an over-abundance of trusted certificate authorities capable of issuing certificates for any domain of their choice. DNS-Based Authentication of Named Entities (DANE) leverages the DNSSEC infrastructure to publish trusted keys and certificates for use with TLS via a new TLSA record type. DNSSEC validated DANE TLSA records have created a new PKI designed to augment or replace the trust model of the existing public CA PKI.

In the context of this memo, channel security is assumed to be provided by TLS. The Transport Layer Security (TLS) protocols provide secured TCP communication. Used without authentication, TLS provides protection only against eavesdropping. With authentication, TLS also provides protection against man-in-the-middle (MITM) attacks. Since the publication of the TLS 1.0 specification in [RFC2246], two updates to the protocol have been published: TLS 1.1 [RFC4346] and TLS 1.2 [RFC5246].

## 1.2. SMTP Channel Security

The Simple Mail Transport Protocol (SMTP) ([RFC5321]) is multi-hop store & forward, while TLS security is hop-by-hop. The number of hops from the sender's Mail User Agent to the recipient mailbox is rarely less than 2 and is often higher. Some hops may be TLS protected, some may not. The same SMTP TCP endpoint can serve both TLS and non-TLS clients, with TLS negotiated via the SMTP STARTTLS command ([RFC3207]). MX RRs abstract hop destinations via DNS. SMTP addresses are not transport addresses and are security agnostic. Unlike HTTP, there is no URI scheme for email addresses to designate whether the SMTP server should be contacted with or without security.

A Mail Transport Agent (MTA) may need to forward a message to a particular email recipient <user@example.com>. To deliver the message, the MTA needs to retrieve the MX hosts of example.com from DNS, and then deliver the message to one of them. Absent DNSSEC, the MX lookup is vulnerable to man-in-the-middle and cache poisoning attacks. As a result, verifying MX host certificates without using DNSSEC is futile as the attacker can simply forge DNS replies and issue bogus MX records, directing traffic to a server of his choice.

One might try to harden STARTTLS with SMTP against DNS attacks by requiring each MX host to possess an X.509 certificate for the recipient domain that is obtained from the message envelope and is not subject to DNS reply forgery. Unfortunately, this is impractical, as email for many domains is handled by third parties, which are not in a position to obtain certificates for all the domains they serve. Deployment of SNI (see [RFC6066] Section 3.1) is no panacea, since the key management is operationally challenging at large scale unless the email service provider is also the domain's registrar and its certificate issuer; this is rarely the case for email.

A man-in-the-middle can also suppress the MX host's STARTTLS EHLO response. Unless the sending MTA is statically configured to use TLS for mail sent to example.com, the message will be sent unauthenticated and in the clear. Sender-side configuration of peer-domains for which TLS must be used can protect an organization and a

few of its business partners, but is not a viable approach to securing the Internet email backbone. Internet email requires contacting many new domains for which security configurations can not be established in advance.

With the existing public Certificate Authority (CA) Public Key Infrastructure (PKI), neither the recipient domain, nor the MX hostname are suitable SMTP server authentication identities. Large scale deployment of authenticated TLS based on this PKI is not possible in the context of MTA to MTA SMTP. SMTP secure channels authenticated via the public CA PKI are rarely used and only between domains that make bilateral arrangements with their business partners. At this time, MTA to MTA traffic between Internet connected organizations typically does not use TLS at all, or uses TLS opportunistically without authentication, for protection only against passive eavesdropping.

Note, the above does not apply to mail submission [RFC6409], where a mail user agent is pre-configured to send all email to a fixed Mail Submission Agent (MSA). Submission servers usually offer TLS and the Mail User Agent (MUA) can be statically configured to require TLS with its chosen MSA. The situation changes when submission servers are configured dynamically via SRV records (see [RFC6186] Section 6, although this is not yet widely deployed). Applications to submission via SRV records will be discussed later in this memo.

With little opportunity to use the existing public CA PKI, MX hosts that support STARTTLS often use self-signed or private-CA issued X.509 certificates. They are rarely configured with a comprehensive list of trusted CAs and do not check CRLs or implement OCSP. In essence, they don't and can't use the existing public CA PKI. This is not simply a result of complacency on the part SMTP server administrators and MTA developers. Nor is it just a result of the relative maturity of the SMTP infrastructure when TLS was introduced. Rather, the abstraction of the SMTP transport endpoint via DNS MX records, often across organization boundaries, limits the use of public CA PKI with SMTP to a small set of sender-configured peer domains.

This does not mean, however, that the Internet email backbone cannot benefit from TLS. The fact that transport security is not explicitly specified in either the recipient address or the MX record means that new protocols can furnish out-of-band information to SMTP, making it possible to simultaneously discover both which peer domains support secure delivery via TLS and how to verify the authenticity of the associated MX hosts. The first such mechanism that can work an Internet scale is DANE TLSA, but use of DANE TLSA with MTA to MTA SMTP must be cognizant of the lack of any realistic role for the existing public CA PKI.

### 1.3. Terminology

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

## 2. Hardening Opportunistic TLS

This section describes opportunistic SMTP over TLS security, where traffic from DANE TLSA aware SMTP clients to domains that implement DANE TLSA records in accordance with this memo is secure. Traffic to other domains continues to be sent in the same manner as before (either manually configured for security or unencrypted and unauthenticated). It is hoped that, over time, more domains will implement DNSSEC and publish DANE TLSA records for their MX hosts. This will enable an incremental transition of the email backbone to authenticated TLS delivery.

Since email addresses and MX hostnames (or submission SRV records) neither signal nor deny support for TLS by the receiving domain, it is possible to use DANE TLSA records to securely signal TLS support and simultaneously to provide the means by which SMTP clients can successfully authenticate legitimate SMTP servers.

### 2.1. TLS discovery

As noted previously (Section 1.2), opportunistic TLS with SMTP servers that advertise TLS support via STARTTLS is subject to a man in the middle downgrade attack. Some SMTP servers erroneously advertise STARTTLS in default configurations that are not in fact TLS capable, and clients need to be prepared to retry plaintext delivery after STARTTLS fails. Downgrade resistant mechanisms for a server to advertise TLS support via DNSSEC validated DANE TLSA records are specified below. DNSSEC validated TLSA records are unlikely to be published by default for servers that do not in fact support TLS, and thus clients can safely interpret their presence as a commitment by the server operator to implement STARTTLS.

SMTP is a store & forward protocol. An MTA that is not the final destination for a message recipient forwards the message one hop closer to the recipient's mailbox. To do so, it must determine the appropriate next-hop destination.

Typically, the next-hop destination defaults to the domain part of the recipient address, which is then subject to MX resolution. The next-hop destination may also be configured by the MTA administrator to be a next-hop destination host (explicitly exempt from MX resolution), or a next-hop destination domain (subject to MX resolution), which takes the place of the domain part of the recipient address. In the language of [RFC5321] Section 5.1, we'll refer to this next-hop destination host or domain as "the initial name".

#### 2.1.1. MX resolution

If the initial name is a next-hop domain subject to MX resolution, a DNSSEC validated "MX" lookup is performed, to obtain the list of associated MX hosts. If no MX records are found, or if the initial name is a next-hop host not subject to MX resolution, it is resolved to one or more network addresses, by performing DNSSEC validated "A" and/or "AAAA" lookups.

Following [RFC5321] Section 5.1, if the "A", "AAAA" or "MX" lookup of the initial name yields a CNAME, we replace it with the resulting name as if it were the initial name and try the same lookup again with the new name. MTAs typically support limited recursion in CNAME expansion so this replacement is performed recursively. If initially, or at any stage of recursion, the response is "bogus", MX resolution fails with a temporary error. Mail delivery SHOULD either be deferred or attempted via any alternative delivery channel configured by the MTA administrator (which may also employ opportunistic DANE TLS).

If at any stage the response is "insecure", opportunistic DANE TLS is not applicable, and mail delivery SHOULD proceed with pre-DANE opportunistic TLS (subject to its various MITM attacks).

If at each and every stage the response is "secure", and the initial name is a next-hop host name not subject to MX resolution, the resulting final name becomes the next-hop destination and is the base domain for TLSA record lookup.

If at each and every stage the response is "secure", and the initial name is a next-hop domain subject to MX resolution, and no MX records are found, the resulting final name is the next-hop destination and is the base domain for TLSA record lookup.



If at each and every stage the response is "secure", and the initial name is a next-hop domain subject to MX resolution, and one or more MX records are found, the MX records MUST be sorted by preference. A better (numerically lower) MX preference for a host that does not support TLS MUST NOT be preempted by a worse (numerically higher) MX preference for a host that does support TLS. In other words, avoiding delivery loops trumps any preference for channel security.

In each delivery attempt via a candidate MX host, the MX host SHOULD be treated as though it were the initial next-hop destination host (which is, of course, not subject to further MX resolution) with the associated TLSA base domain determined as above.

CNAMEs are not legal in the exchange field of MX records, thus MTAs MAY skip over MX records in which the MX exchange is a CNAME. There is some additional risk, in this case, that the MTA may fail to notice that it is one of the MX hosts for the destination and that it must skip MX records with equal or worse (numerically higher precedence). If an MTA does allow CNAMEs to be used in MX records it SHOULD process them recursively as described above to determine whether opportunistic DANE TLS is applicable and if so the associated TLSA RRset base domain.

#### 2.1.2. TLSA record lookup

When all the DNSSEC lookups, "CNAME", "MX", "A" or "AAAA", used to obtain a given TLSA base domain (one for each candidate MX host if multiple DNSSEC validated MX hosts were found) is "secure", and the SMTP client is configured for opportunistic DANE TLS, it SHOULD locate the TLSA RRset corresponding to this base domain. If, for example, the base domain is "mail.example.com", the TLSA RRset is obtained via a DNSSEC query of the form:

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

Typically, the destination TCP port is 25, but this may be different with custom routes specified by the MTA administrator or when an MUA connects to a submission server on port 587. The SMTP client MUST use the appropriate "\_<port>" prefix in place of "\_25" when the port number is not equal to 25. The query response may be a CNAME (or a DNAME + CNAME combination), or the TLSA RRset. DNAME processing with DNSSEC can be done using standard DNAME resolution techniques and will not be discussed in detail here. The SMTP client MUST check the security status of the response.

If the response is "bogus", delivery via the host in question SHOULD NOT proceed, otherwise the SMTP client is vulnerable to man in the middle STARTTLS downgrade attacks. If the response is "insecure",

opportunistic DANE TLS is not applicable for the host in question, and the SMTP client SHOULD proceed with legacy opportunistic TLS. If the response is "secure" and the record is a CNAME or DNAME, the SMTP client restarts the TLSA query at the target domain, following CNAMEs as appropriate.

If, after possible CNAME indirection, the response is "secure" and at least one TLSA record is found (even if not usable because it is unsupported by the implementation or administratively disabled) the next-hop host has committed to TLS support. The SMTP client SHOULD NOT deliver mail via such a next-hop host unless a TLS session is negotiated via STARTTLS. This avoids the man in the middle STARTTLS downgrade attacks.

When usable TLSA records are available, a client SHOULD NOT deliver mail via a server that fails to match at least one TLSA record. This is not a "must" because clients may incrementally deploy opportunistic DANE TLS only for selected peer domains. At times, clients may need to disable opportunistic DANE TLS for peers that fail to interoperate due to misconfiguration or software defects on either end. For opportunistic DANE TLS to be robust (resistant to failures), servers MUST live up to the promises stated by the existence of the TLSA record, but it is not always possible to compel clients to use a security policy chosen by the server. Given a robust security protocol, clients will hopefully, over time, willingly choose to adopt it.

SMTP over opportunistic TLS using DANE implementations and publishers need to follow the guidance outlined in [I-D.dukhovni-dane-ops]'s "Certificate Name Check Conventions", "Service Provider and TLSA Publisher Synchronization" and "TLSA Base Domain and CNAMEs" sections.

Note: treating CNAMEs in MX hosts in the same manner as CNAMEs with non-MX destinations is consistent with [RFC5321] where non-MX destinations are considered equivalent to destinations with a single preference 0 MX record with the query domain identical to the MX host.

## 2.2. DANE authentication

### 2.2.1. TLSA certificate usages

As noted in the introduction, the existing public CA PKI is not viable for the Internet email backbone. TLSA records for MX hosts or submission servers that are to be found via SRV records SHOULD NOT include certificate usage "0" or "1", as in both cases SMTP clients cannot be expected to perform [RFC5280] PKIX validation or [RFC6125] identity verification.

If despite this recommendation SMTP servers do publish TLSA records with certificate usage "0" or "1", clients SHOULD make use of these to the fullest extent possible.

TLSA Publishers should follow the TLSA publication size guidance found in [I-D.dukhovni-dane-ops] about "DANE DNS Record Size Guidelines".

#### 2.2.1.1. Certificate usage 3

Since opportunistic DANE TLS will be used by non-interactive MTAs, with no user to "press OK" when authentication fails, reliability of peer authentication is paramount. TLSA records published for SMTP servers SHOULD be "3 1 1" records to support opportunistic SMTP over TLS with DANE. This record specifies the SHA-256 digest of the server's public key.

Authentication via certificate usage "3" TLSA records involves no certificate authority signature checks. It also involves no server name checks, and thus does not impose any new requirements on the names contained in the server certificate (SNI is not required when the TLSA record matches the public key of the server's default certificate). It uses the SHA-256 digest which all clients are obligated to support, and works across certificate renewals with the same key.

Two TLSA records will need to be published before updating a server's public key, 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 time out the previous TLSA RRset, which contains only the old key, the server may be reconfigured to use the new private key and associated public key certificate. The amount of time a server should wait before using a new key that is referenced by new TLSA records should be  $2 * \text{the TTL of the previously published TLSA records}$ . Once the server is using a new key, the obsolete TLSA RR can be removed from DNS, leaving only the RR that matches the new key.

#### 2.2.1.2. Certificate usage 2

Some domains may prefer to reduce the operational complexity of publishing unique TLSA RRs for each TLS service. If the domain employs a common issuing certificate authority to create certificates for multiple TLS services, it may be simpler to publish the issuing authority's public key as a trust-anchor for the certificate chains of all relevant services. The TLSA RRs for each service issued by the same TA may then be CNAMEs to a common TLSA RRset that matches the TA. In this case, the certificate chain presented in the TLS handshake of each service SHOULD include the TA certificate, as SMTP clients cannot generally be expected to have domain-issued trust-anchor certificates in their trusted certificate store. TLSA Publishers should publish either "2 1 1" or "2 0 1" TLSA parameters, which specifies the SHA-256 digest of the trust-anchor public key or certificate. As with regular certificate rollover discussed in Section 2.2.1.1, two such TLSA RRs need to be published to facilitate TA certificate rollover.

The usability of "2 1 1" or "2 0 1" TLSA RRs with SMTP is not assured. Unless server operators employing these RRs universally ensure that the corresponding TA certificate is included in the SMTP server's TLS handshake trust chain, then clients MAY enable support for these RRs. If sufficiently many server administrators are negligent in deploying these RRs, SMTP clients should be hesitant to support them, since mail delivery will not work to many destination domains if they do. We encourage server operators to implement these RRs, if appropriate, to their organization, provided they do so with care. It is critical to never forget to include trust-anchor certificates in server trust chains. SMTP client implementations are encouraged to support these TLSA RRs by default, unless future experience proves optimism for publishing correct certificate chains unfounded.

Clients employing opportunistic DANE TLS MAY choose to treat any TLSA records with certificate usage "0" as unusable. They may then choose to connect via unauthenticated mandatory TLS if no alternative authentication mechanisms are available.

#### 2.2.1.3. Certificate usage 1

SMTP servers SHOULD NOT publish TLSA RRs with certificate usage "1". SMTP clients that implement this specification SHOULD ignore the PKIX validation requirement when they encounter certificate usage "1", and SHOULD authenticate the server per the content of the TLSA record alone. That is, SMTP clients should treat certificate usage "1" as certificate usage "3"; clients SHOULD NOT apply name checks, expiration checks, or process the server certificate content beyond possibly extracting its public key for matching against corresponding TLSA RRs.

#### 2.2.1.4. Certificate usage 0

SMTP servers SHOULD NOT publish TLSA RRs with certificate usage "0". Since PKIX validation is not possible with opportunistic DANE TLS, SMTP clients SHOULD treat certificate usage "0" RRs as though they were certificate usage "2" RRs. But, with certificate usage "0" the usability of the TLSA record depends more strongly on its matching type. Name checks, key usage checks, etc, apply identically to the leaf certificate with TLSA usages "0" and "2".

If the matching type is "0" (the server should also avoid this matching type and should publish usage "3" or "2" public key or certificate digests), the TLSA record contains the full certificate or full public key of the trusted certificate authority. In this case the client has all the information it needs to match the server trust-chain to the TLSA record. The client SHOULD ignore the PKIX validation requirement, and verify the server's trust chain via its DANE TLSA records only (name checks still apply as with usage "2").

If the matching type is not "0", the TLSA record contains only a digest of the trust certificate authority certificate or public key. The server operator publishing usage 0 TLSA records may expect that clients already have the issuing authority certificate on hand, and may omit it from the server's certificate chain. As a result, the client may not be able to match the server trust chain against the TLSA record if it, in fact, does not have a copy of the certificate authority certificate or public key.

SMTP clients that implement this specification SHOULD treat TLSA records with certificate usage "0" and a digest matching type as unusable, but MAY be explicitly configured to support them when it is believed that clients possess a sufficiently complete set of trusted public CA certificates. This is most plausible with an MUA which only needs enough CA certificates to authenticate its preferred submission service.

#### 2.2.2. Certificate matching

When at least one usable "secure" TLSA record is found, the SMTP client SHOULD use TLSA records to authenticate the next-hop host, mail SHOULD not be delivered via this next-hop host if authentication fails, otherwise the SMTP client is vulnerable to TLS man in the middle attacks.

With TLSA certificate usages "2" or "0" the TLSA base domain SHOULD be the domain the client looks for in the server certificate to check that it has reached the correct server, if the TLSA base domain was obtained indirectly via an MX lookup, the CNAME resolved name used in

the MX lookup SHOULD also be accepted. Barring CNAME expansion, this additional domain is typically the envelope recipient domain, but may be different when the client is an MUA sending all mail to a submission server, or an MTA configured with explicit next-hop destination overrides for the message recipient.

Accepting certificates with the next-hop domain in addition to the next-hop MX host allows a domain with multiple MX hosts to field a single certificate bearing the email domain name across all the MX hosts, this is also compatible with pre-DANE SMTP clients that are configured to look for the email domain name in server certificates.

The client MUST NOT perform certificate usage name checks with certificate usage "3" (or equivalently "1"), since with these the server is authenticated directly by matching the TLSA RRset to its certificate or public key without resort to any issuing authority. The certificate content is ignored except in so far as it is used to match the certificate with the TLSA RRset.

To ensure that the server sends the right certificate chain, the SMTP client MUST send the TLS SNI extension containing the TLSA base domain. Since DANE-aware clients are obligated to send SNI information, which requires at least TLS 1.0, SMTP servers for which DANE TLSA records are published MUST support TLS 1.0 or later with any client authorized to use the service.

Each SMTP server MUST present a certificate trust chain (see [RFC2246] 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 RRset includes records with a matching type indication a digest record (i.e., a value other than "0"), the SHA-256 digest of any object SHOULD be provided along with any other digest published, since clients may support only SHA-256. Unless SHA-256 proves vulnerable to a "second preimage" attack, it should be the only digest algorithm used in TLSA records.

If the server's TLSA records match the server's default certificate chain, the server need not support SNI. The server need not include the extension in its TLS HELLO, simply returning a matching certificate chain is sufficient. Servers MUST NOT enforce the use of SNI by clients, if the client sends no SNI extension, or sends an SNI extension for an unsupported domain the server MUST simply use its default certificate chain. The client may be using unauthenticated opportunistic TLS and may not expect any particular certificate from the server.

The client may even offer to use anonymous TLS ciphersuites and servers SHOULD support these, no security is gained by forcing the use of a certificate the client will ignore. Indeed support for anonymous ciphersuites in the server makes audit trails more useful if the chosen ciphersuite is logged, as this will in many cases record which clients did not care to authenticate the server. (The Postfix SMTP server supports anonymous TLS ciphersuites by default, and the Postfix SMTP client offers these at its highest preference when server authentication is not applicable).

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, therefore authentication via this protocol is expected to be less prone to connection failure caused by incompatible configuration of the client and server.

### 3. Opportunistic TLS for Submission

Prior to [RFC6409], the SMTP submission protocol was poster child for PKIX TLS. The MUA typically connects to one or more submission servers explicitly configured by the user. There is no indirection via insecure MX records, and unlike web browsers, there is no need to authenticate a large set of TLS servers. Once TLS is enabled for the desired submission server or servers, provided the server certificate is correctly maintained, the MUA is able to reliably use TLS to authenticate the submission server.

[RFC6409] aims to simplify the configuration of the MUA submission service by dynamically deriving the submission service from the user's email address. This is done via SRV records, but at the cost of introducing the same TLS security problems faced by MTA to MTA SMTP. Prompting the user when the SRV record domain is different from the email domain is not a robust solution.

The protocol defined in this memo can also be used to opportunistically secure the submission service association. If the email domain is DNSSEC signed, the SRV records are "secure" and the SRV host publishes secure TLSA records for submission, then the MUA can safely auto-configure to authenticate the submission server via DANE. When DANE TLSA records are not available, the client SHOULD fall back to legacy behavior.

#### 4. Mandatory TLS Security

An MTA implementing this protocol for sending email to Internet destinations may need a greater security assurance when sending email to selected destinations that to which the sending organization sends sensitive email and may have regulatory obligations to protect its content. This protocol is not in conflict with this requirement, and in fact it can often simplify authenticated delivery to such destinations.

Specifically, with domains that publish DANE TLSA records for their MX hosts a sending MTA can be configured to use the receiving domains's DANE TLSA records to authenticate the corresponding MX hosts, thereby obviating the complex manual provisioning process. In anticipation of, or in response to, a failure to obtain the expected TLSA records, the sending system's administrator may choose from a selection of fallback options, if supported by the sending MTA:

- o Defer mail if no usable TLSA records are found. This is useful when the destination is known to publish TLSA records, and lack of TLSA records is most likely a transient misconfiguration.
- o Authenticate the peer via a manually configured certificate digest. This may be obtained, for example, after a problem is detected and confirmed to be valid by some out-of-band mechanism.
- o Authenticate the peer via the existing public CA PKI, if the peer server has usable CA issued certificates. In many cases the sending MTA will need custom certificate name matching rules to match the destination's gateways. And the sending server must explicitly configure policy for the destination to always require TLS to prevent MITM attacks.
- o Send via unauthenticated mandatory TLS. This is useful if the requirement is merely to always encrypt transmissions to protect against only eavesdropping, and the possibility of MITM attacks is less of a concern than timely email delivery.

It should be noted that barring administrator intervention, email SHOULD be deferred when DNSSEC lookups fail, (as distinct from "secure" non-existence of TLSA records, or secure evidence that the domain is no longer signed). In addition to configuring fallback strategies when TLSA records are unexpectedly absent, administrators may, in hopefully rare cases, need to disable DNSSEC lookups for a destination to work around a DNSSEC outage.

#### 5. Acknowledgements



The authors would like to thank Phil Pennock for his comments and advice on this document.

Acknowledgments from Viktor: Thanks to Tony Finch who finally prodded me into participating in DANE working group discussions. Thanks to Paul Hoffman who motivated me to produce this memo and provided feedback on early drafts. Thanks also to Wietse Venema who created Postfix, and patiently guided the Postfix DANE implementation to production quality.

## 6. Security Considerations

This protocol leverages DANE TLSA records to implement MITM resistant opportunistic channel security for SMTP. For destination domains that sign their MX records and publish signed TLSA records for their MX hosts, this protocol allows sending MTAs (and perhaps dynamically configured MUAs) 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 and negotiated out-of-band key and TLS configuration exchanges.

## 7. Normative References

- [I-D.dukhovni-dane-ops]  
Dukhovni, V., "DANE TLSA implementation and operational guidance", draft-dukhovni-dane-ops-00 (work in progress), May 2013.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2246] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.
- [RFC3207] Hoffman, P., "SMTP Service Extension for Secure SMTP over Transport Layer Security", RFC 3207, February 2002.

- [RFC3546] Blake-Wilson, S., Nystrom, M., Hopwood, D., Mikkelsen, J., and T. Wright, "Transport Layer Security (TLS) Extensions", RFC 3546, June 2003.
- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.
- [RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.
- [RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, March 2005.
- [RFC4346] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.
- [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, May 2008.
- [RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, October 2008.
- [RFC6066] Eastlake, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, January 2011.
- [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, March 2011.
- [RFC6186] Daboo, C., "Use of SRV Records for Locating Email Submission/Access Services", RFC 6186, March 2011.
- [RFC6409] Gellens, R. and J. Klensin, "Message Submission for Mail", STD 72, RFC 6409, November 2011.
- [RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, August 2012.

Authors' Addresses

Viktor Dukhovni  
Unaffiliated

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

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

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

DANE  
Internet-Draft  
Intended status: Informational  
Expires: August 18, 2014

O. Gudmundsson  
Shinkuro Inc.  
February 14, 2014

Harmonizing how applications specify DANE-like usage  
draft-ogud-dane-vocabulary-02

Abstract

There is no standard terminology as how to talk about use of DNS in various application contexts, this document goal is to facilitate creation of such a vocabulary/taxonomy.

This document started out as proposal for specific word usage for specifications of adding DANE like technology by different protocols/services. DANE is a method for specifying in DNS records acceptable keys/certificates for application servers.

The terms defined in this document should be applicable to all uses of service specification that uses DNS records.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 18, 2014.

Copyright Notice

Copyright (c) 2014 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 . . . . .	2
1.1. Requirements notation . . . . .	3
2. Proposed Terms . . . . .	3
2.1. DNS Navigation Records . . . . .	3
2.2. DNS Integrity . . . . .	4
2.3. Service Specification Records (SSR) . . . . .	4
2.4. Service Address Records (SAR) . . . . .	5
2.5. Application Authentication Records (AAR) . . . . .	6
2.6. Offered Name: Name used when indirection records are used . . . . .	6
3. Example specification . . . . .	7
4. IANA considerations . . . . .	7
5. Security considerations . . . . .	7
6. Internationalization Considerations . . . . .	8
7. Acknowledgements . . . . .	8
8. References . . . . .	8
8.1. Normative References . . . . .	8
8.2. Informative References . . . . .	8
Appendix A. Document history . . . . .	9
Author's Address . . . . .	9

## 1. Introduction

DNS [RFC1034] is being used by many protocols to express where services are located on the internet, today there is no good way to express exactly what people have in mind when specifying a new service/protocol exactly and in concise manner how the service is looked up in the DNS.

DANE [RFC6698] is a powerful new way to provide/amend how authentication/authorization/confidentiality of a connection to a server can be protected by leveraging DNSSEC [RFC4033] [RFC4034] [RFC4035] for the establishment of TLS connection [RFC5246] [RFC6347] which in many cases uses PKIX [RFC5280]. All of these technologies are complicated. People familiar with one or two are not necessarily familiar with all the parts that needed to apply DANE like mechanism to other protocols.

The goal of this document is three fold:

- o To provide common vocabulary for usage of DNS records in service specification.
- o To provide an overview of the non protocol specific parts needed to specify an DANE like addition.
- o To provide a common framework for such specifications making it easy to review/compare the specifications. An important goal is to allow the new specifications to avoid repeating explanations and/or definitions.

Number of RFC's in the past have tried to use consistent terminology when specifying how to access services both in the context of security TLS with X.509 [RFC6125] and without security [RFC2782]. The terminology in this document is not identical but concepts are similar. The hope is that once the standard terminology is specified, as simple documents can provide a mapping if one is needed.

This version of the document aims to hide complexity and focus on generalities. This is done to make it easier for the reader to decide if the terms here are of use and if it is worthwhile for the DANE WG to adopt this document. Descriptions of complexities can be added in later versions if the WG decides that is needed.

When notation "foo/bar" is used below that is because the editor is not sure if both apply or which one is more appropriate, please advise.

#### 1.1. Requirements notation

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

#### 2. Proposed Terms

The terms below are being proposed to avoid confusion when reading protocol specifications related to DNS and DANE, for various application protocols.

At this point all the terms below are proposals and better terms are welcome.

##### 2.1. DNS Navigation Records

DNS Navigation refers to any records used to traverse the DNS tree to find the records requested. This includes

NS records: that provide a referral to DNS servers for more specific part of the name being looked up. Example: name server for "example." will hand out a referral to server for "bar.example." when asked about "foo.bar.example."

CNAME records: records that change the location of an record, this for all practical purposes a pointer that only applies to that specific name.

DNAME records: specify a rewrite rule for a name to a new name. Example: "bar.example." DNAME "foo.example." means that "www.bar.example." is to be looked up as "www.foo.example.". DNAME applies to names that are longer than the name it, i.e. "bar.example." is not rewritten but "www.bar.example." is.

DANE specification explicitly requires all of these records to be validated by DNSSEC.

See section Section 2.2

While traversing the DNS tree other records like A and AAAA are used but these records do not change the "navigation", these records do not explicitly need to be protected as the data retrieved from the addresses is expected to be protected.

## 2.2. DNS Integrity

DNSSEC defines a records and procedures to provide integrity and authentication to data stored in DNS [RFC4034]. The records used to provide the keying information and chain of trust are DNSKEY, DS records. NSEC/NSEC3 provide information about existence/non-existence of the requested information. RRSIG provides a digital signature for a RRset.

DNSSEC provides both Integrity and Authenticity i.e. it says the records came from the right source and have not been changed.

Any DNS record that is DNS Integrity protected, will pass DNSSEC validation for all DNS Navigation records leading to the name and the record itself also passes DNSSEC validation.

In the case of CNAME and DNAME that go "sideways" i.e. to a different branch of the DNS tree, both branches MUST be validated.

## 2.3. Service Specification Records (SSR)

Protocols have different ways to express servers.

- o Web servers are frequently specified by name i.e. the "www" prefix, thus its service specification record is: "address record stored at www.<domain>".
- o Email servers have a special RR type (MX): SRR= "MX record at <domain>"),
- o Jabber uses SRV records: SSR="SRV record at \_xmpp-server.tcp.<domain>",
- o ENUM uses NAPTR records etc.
- o In addition there are also protocols that use a combination like S-NAPTR a schema where NAPTR records are used to specify where to look for SRV records. For all practical purposes NAPTR + SRV should combined be treated as the Service Specification.

For a DANE like specification it has to be clear as what the service specification records are and these records require DNS Integrity.

NOTE: when a client supplies a string to the server as a indicator of what service the the client wants, the string supplied MAY depend on redirection in DNS navigation as well as results of NAPTR records, etc. See section Section 2.6.

NOTE: when NAPTR records as are used they should be treated same way as DNS Navigation records even though strictly speaking it is the application that evaluates the NAPTR record.

NOTE: When there is a CNAME at the name service is expected to be specified at, that can be either a DNS Navigation record or a Service Specification Record. Protocol specification should provide guidance on interpretation.

#### 2.4. Service Address Records (SAR)

These are the address records for the servers that offer the service.

In some cases the Service Specification records reside at the same name or are the same as the Service Address records. Example: original TLS/DANE[RFC6698], thus both SSR and SAR records are covered by the same DNS integrity rule.



## 2.5. Application Authentication Records (AAR)

This term refers to the records that provide information about what are acceptable keys or certificates for the servers to offer.

Application Authentication Records MUST be protected by DNS Integrity and each protocol specification MUST explicitly state where/how to look up the Authentication records.

In some cases all the servers for a service will have the same authentication information, in other cases it is going to be on a server by server case. In the first case it is "natural" to store the Authentication records "at" the Service Specification records. In the second case it more natural to store them "at" the Address Records. In this context "at" means the authentication records are stored at name that is an extension of the location example: "\_443.\_tcp.www.example.com" for [RFC6698]. It is possible that neither of these locations is the right one and in that case the specification MUST explicitly express rules as how to find the Authentication Records.

Note: above that there is no a requirement that the Application Address records be covered by DNS Integrity. This is because when the Application Authentication records reside "at" the address records, DNS Integrity is inherited. On the other hand when when Application Authentication Records are stored "at" the Service Specification Record, DNS Integrity for the address records is optional, as any connection to a bogus/wrong server should fail the Authentication tests performed at connection time.

Note: When a Address record search has a CNAME at or DNAME above, the name queried, where should the Authentication Records reside ? With CNAME or with final address record ?

## 2.6. Offered Name: Name used when indirection records are used

In many protocols one of the first items presented by the application is a <name> that is "related to"/"derived from" the original query name. When DNAME is used the name queried for might be required to be rewritten into a new name.

To disambiguate these cases following prefix terms are defined. Similar rules apply NAPTR + SRV combinations. It is important for many applications to be able to express what name is presented by the application to the server at connection time.

Query: The name the application issued the query for to discover SSR /service.

Final: The name after all the indirection records have been applied.

SRV The name on the SRV record used.

NAPTR The name on the first NAPTR record used, prefix with Final if that is the one wanted.

Intermediate A particular location in the indirection chain. The specification needs to handle this case if it ever occurs.

NOTE: not sure this is needed???ogud???

### 3. Example specification

This section is an short example for a protocol that is like SSH [RFC4253] we will call this protocol HISS. This is not an actual full specification, just here to give an idea of how to go about extending DANE-like to a random protocol using the terminology from this document.

Location of HISS protocol DNS records:

Service Specification Records:

HISS uses address records as the service specification record. This record MUST have "DNS Integrity" as explained in RFC-to-be-this-document. CNAME/DNAME are treated as a DNS Navigation record.

Service Address Records:

see: Service Specification Records.

Application Authentication Records:

The protocol uses the DNS HISSFP that is stored at the same name as the service is specified. The HISSFP record, if present, takes precedence over keys stored in client cache.

Offered Name

Not used.

The HISS protocol and HISSFP DNS RR do not exist

### 4. IANA considerations

None

[RFC Editor: Please remove this section before publication ]

### 5. Security considerations

This documents goal is to improve specifications of adding security via DANE technology to protocols, thus the overwriting goal is to decrease confusion and increase clarity, with the end goal of improving security. This document does not specify a protocol. XX  
Needs more work XX

## 6. Internationalization Considerations

When selecting terms to use in standards documents it is important to select words that do not confuse international readers. This document goes out of its way in selecting English terms that are dissimilar to avoid confusions.

## 7. Acknowledgements

Number of people have commented that this is interesting work. Peter Saint-Andre tried to apply the terms to one of his documents and provided many good suggestions.

## 8. References

### 8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, August 2012.

### 8.2. Informative References

- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.
- [RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.
- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.
- [RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.

- [RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, March 2005.
- [RFC4253] Ylonen, T. and C. Lonvick, "The Secure Shell (SSH) Transport Layer Protocol", RFC 4253, January 2006.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.
- [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, May 2008.
- [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, March 2011.
- [RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, January 2012.

#### Appendix A. Document history

[RFC Editor: Please remove this section before publication ]

02 Textual improvements, applied comments from Peter Saint-Andre.

01 Added definition of offered names, expanded DNAME/CNAME text added NAPTR and SRV.

00 Initial version

#### Author's Address

Olafur Gudmundsson  
Shinkuro Inc.  
4922 Fairmont Av, Suite 250  
Bethesda, MD 20814  
USA

Email: ogud@ogud.com