Measuring DANE TLSA Deployment

Liang Zhu, Duane Wessels, Allison Mankin, John Heidemann

Measuring DANE TLSA Deployment

Liang Zhu[1] Duane Wessels[2] Allison Mankin[2] John Heidemann[1]

[1]University of Southern California [2]Verisign Labs

Abstract. The DANE (DNS-based Authentication of Named Entities) framework uses DNSSEC to provide a source of trust, and with TLSA it can serve as a root of trust for TLS certificates. This serves to complement traditional certificate authentication methods, which is important given the risks inherent in trusting hundreds of organizations—risks already demonstrated with multiple compromises. The TLSA protocol was published in 2012, and this paper presents the first systematic study of its deployment. We studied TLSA usage, developing a tool that actively probes all signed zones in `.com` and `.net` for TLSA records. We find the TLSA use is early: in our latest measurement, of the 485k signed zones, we find only 997 TLSA names. We characterize how it is being used so far, and find that around 7–13% of TLSA records are invalid. We find 33% of TLSA responses are larger than 1500 Bytes and will very likely be fragmented.

1 Introduction

The Domain Name System (DNS) is central to Internet use. Originally used mainly to map between names and IP addresses or services, DNS has grown to support many other applications. To protect DNS information from modification or forgery, DNS Security Extensions (DNSSEC) provides integrity of DNS data via a cryptographic chain of trust following the DNS hierarchy. Traditional public-key infrastructure (PKI) places trust in multiple Certification Authorities (CAs, or PKI-CAs). Rather than the PKI's many roots and shallow tree, DNSSEC provides a single root and deeper hierarchy, decreasing the size of the root of trust and thus its vulnerability to compromise.

DANE (DNS-based Authentication of Named Entities) takes advantage of the DNSSEC-provided root of trust to authenticate TLS (Transport Layer Security) certificates. It places *TLSA records* in the DNS hierarchy and uses DNSSEC to validate their integrity. DANE TLSA therefore complements PKI Certificate Authorities, allowing TLS-users to better control certificates validation and protect against classes of CA compromise (as has occurred before [6,5]).

The DANE standard was published in 2012 [14], relatively recently. Although standardized, little is known about how actively DANE is being used. After two years, how many domains are using DANE? How widely used is TLSA? In this paper, we start to answer these questions.

This paper presents the first systematic study of DANE TLSA deployment. Its first contribution is to describe an efficient methodology to actively probe DANE TLSA deployment status for three protocols on six ports. We apply this

method to get a *complete* view of DANE usage at the two largest generic top-level domains (gTLDs): `.com` and `.net` for more than four months. (Passive monitoring would provide an incomplete view.) Our second contribution is to track DANE TLSA growth. We see that deployment remains very early, with only 997 TLSA names of the 485k DNSSEC zones, but *use is steadily increasing* (§ 4.1). Our third contribution is to evaluate DANE usage in two ways. We check for *correct* use of TLSA, and we consistently find 7%-13% of TLSA records are invalid: no IP address, no certificate, or a mismatch between the TLSA record and the certificate (§ 4.3). This high rate of misconfiguration suggests that DANE remains experimental, not mandatory. DANE has several operational modes, so we also characterize which are most popular, finding that DANE is most often used (76% of the time) to establish trust independent of the public CA hierarchy (§ 4.4). Our final contribution is to evaluate how DANE interacts with UDP. We find that find 33% of TLSA responses exceed the Ethernet MTU and will be fragmented at the IP layer, raising potential performance issues (§ 4.5).

2 Background and Related Work

We next briefly review DNS, DNSSEC, and DANE TLSA, and prior work observing DNS.

2.1 DNS

DNS is a protocol that maps *domain names* to *Resource Records* (RRs) using globally distributed servers to provide a consistent global namespace distributed across millions of servers [20,21]. There are many types of RRs, from `A` and `AAAA` for IPv4 and IPv6 addresses, or `MX` for the mail server for a given domain. RRs are stored in *zones*, with each zone managing part of the namespace (`*.example.com`). An *authoritative* name server provides the answer for a zone.

2.2 DNSSEC

DNS originally assumed a cooperative internet, so its basic design did not protect against malicious attempts to alter or pollute the namespace. DNSSEC was developed to protect the integrity of DNS responses by cryptographically signing the zone, allowing anyone to verify that RRs are what the zone owner intended [2,4,3]. For most users, DNSSEC trust is anchored in the single, signed root zone. The chain of trust extends to lower levels of the namespace via signed delegations in the form of Delegation Signer (DS) records. The operator of a signed zone is responsible for publishing its DS records in its parent zone.

2.3 DANE TLSA

DANE (DNS-based Authentication of Named Entities) is the idea that DNSSEC allows the DNS hierarchy to provide a root of trust for different types of information. DANE can compliment the traditional public-key infrastructure (PKI)

Certification Authorities (CAs, or PKI-CAs) by giving operators of TLS-based services more control over how they indicate the authenticity of their TLS certificates. The CA model has come under scrutiny recently due to CA compromises and a lack of transparency and choice in the set of trusted CAs.

The DANE framework can provide trust for many different applications. DANE TLSA [14] is the first of these, providing methods to verify X.509 server certificates used in Transport Layer Security [9]. This paper addresses the deployment of DANE TLSA; studies of TLSA design and TLS performance with DANE are not our focus.

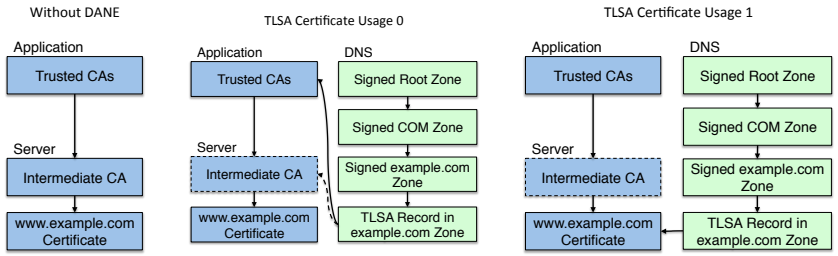
In DANE TLSA, domain name owners publish "certificate association data" in the DNS as TLSA RRs. TLSA RRs specify how applications can verify an X.509 certificate. Options range from providing the X.509 certificate itself, identifying a particular already-known intermediate CA, any already-known intermediate CA, or specify a new CA to serve as trust anchor. It can further specify whether the entire certificate, or only the public key, must be matched, and whether matching is based on hashes or an exact match of the given data. Figure 1 highlights the different ways that TLSA complements and constrains traditional CA methods.

TLSA associates records with particular network services to support different certificates for different services on the same host. TLSA names are prefixed with a port number and protocol name. For example, the TLSA record for the HTTPS service at `www.example.com` is stored under the name `_443._tcp.www.example.com`.

2.4 Related Work

We are not the first to scan the DNS. Several groups have walked the DNS reverse hierarchy, and ISC makes this data available publicly [16]. Others have used active DNS queries to study potential DNSSEC DDoS attacks [33], and uncover operational practices of EDNS-Client-Subnet (ECS) adopters [32]. SecSpider has tracked DNSSEC deployment and health since 2005 [26]. NIST monitors DNSSEC deployment of a set of government, industry and university domains [23]. Our probing is similar to these measurements, but targeted tracking growth of DANE TLSA. The Internet Society also provides pointers to DNSSEC deployment reports [8]. We provide data about DANE TLSA that could fit in such a report.

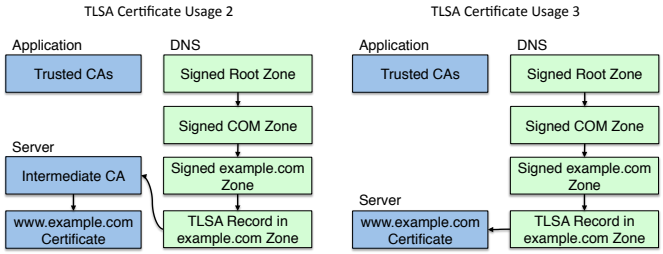
To support testing, several organizations have created correct or intentionally misconfigured DANE sites [7,37,24]. Other groups have created websites or tools that allow one to validate DANE TLSA (and DNSSEC) by request [10,22,19], both with IPv4 and IPv6 [31,1], and published DANE TLSA enabled mail servers [19]. Our measurements complement test cases and on-demand tests by evaluating deployment correctness as seen in the field, at least for two large TLDs.



(a) Certificate validation without DANE TLSA.

(b) TLSA Usage 0. The TLSA record constrains the PKIX validation to a specific certificate authority.

(c) TLSA Usage 1. The TLSA record constrains the PKIX validation to a specific server certificate.



(d) TLSA Usage 2. The TLSA record constrains the PKIX validation to a CA trust anchor that might not be known to the application.

(e) TLSA Usage 3. The TLSA record specifies a server certificate that might not validate via the application's PKIX mechanism.

Fig. 1: The different ways that DANE TLSA complements and constrains certificate validation in applications.

3 Monitoring DANE TLSA Deployment

To understand current DANE TLSA deployment we are interested in long-term observations of its use and growth. We have developed *PryDane*, a new tool that takes a set of zone names as input, then evaluates all those that use DNSSEC to see which also use TLSA. For zones with TLSA records, it also validates whether records match the the servers' certificates.

3.1 How to Track TLSA-Enabled Names

Our goal is to track increased deployment of DANE TLSA over time. Since the DNS is large, TLSA records are currently rare, and we need to probe regularly, our first challenge is to *efficiently* search DNS for TLSA use. Two possible

```
for all $DOMAIN in DS records of COM and NET zones do
  check _443._tcp.$DOMAIN
  check _443._tcp.www.$DOMAIN
  for SMTP $PORT 25, 465, 587 do
    if MX record points to $MX then
      check _$PORT._tcp.$MX
    else
      check _$PORT._tcp.$DOMAIN
  for $NAME _xmpp-client._tcp.$DOMAIN, _xmpp-server._tcp.$DOMAIN do
    if $NAME SRV record points to $PORT and $TARGET then
      check _$PORT._tcp.$TARGET
    else
      for XMPP $PORT 5222, 5269 do
        check _$PORT._tcp.jabber.$DOMAIN
        check _$PORT._tcp.xmpp.$DOMAIN
```

Fig. 2: Pseudo code of our probing system.

methods present themselves: passive collection from live traffic, and TLD zone files. Each has advantages and disadvantages.

Passive collection of DNS (for example, [11]) can provide usage and popularity with basic DNS data. It also can collect data across the entire DNS namespace. However, passive collection is likely incomplete, missing zones that never happen to be used during observation, and collection can be complex and sometimes unreliable.

Zone files are available for all gTLDs through ICANN's Centralized Zone Data Service [15]. We find that zone files are generally more reliable and easier to process, and they guarantee complete coverage within the TLD. They do not, however, indicate which names are actually being queried, and do not cover the entire DNS namespace since most ccTLDs do not make their zone files available. Nonetheless, for this study, we have chosen to use TLD zone files as our data source. So far we have only used the `.com` and `.net` zone files. See § 5 for details about the zone files we used. Including more gTLDs is future work.

To get a set of meaningful targets to probe, we select *all* DNSSEC signed names by extracting those delegations that have accompanying `DS` records. We ignore non-DNSSEC names because TLSA records are only trustworthy when their integrity is ensured, and use of TLSA without DNSSEC is an error.

We probe several services that are TLSA early-adopters: HTTPS, SMTP (mail [29]), and XMPP (Jabber [28,18]). Other services that may use TLS are VPNs and secure SIP, but we omit these because we know of no deployments that support DANE TLSA. Table 1 lists when protocol support for TLSA began, and our gradual addition of protocol coverage. We look for DANE TLSA use with service discovery methods specific to each protocol as shown in Figure 2. Generally these probe only with the target domain, but some MX records point to e-mail servers in domains outside our targets (`.com` and `.net`).

protocol	port	TLSA Support date	probing start
HTTPS [9]	443	2013-03-04 [10]	2014-07-14
SMTP [13]	587	2014-01-15 [29]	2014-07-14
	25		2014-10-02
	465		2014-10-02
XMPP [27]	5222	experimental	2014-10-14
	5269	[28,18]	2014-10-14

Table 1: Protocol and implementation support for TLSA, and when our coverage begins for each.

scanned size	130.30M	(100%)	
non-DNSSEC	129.82M	(99.63%)	
DNSSEC	485k	(0.37%)	
non-TLSA zones	∼485k		
TLSA zones	443		[100%]
com and net	365		[82.4%]
other zones	78		[17.6%]
TLSA names	997		{100%}
HTTPS (443)	393		{39.5%}
SMTP (25)	314		{31.5%}
(465)	87		{8.7%}
(587)	105		{10.5%}
XMPP (5222)	49		{4.9%}
(5269)	49		{4.9%}

Table 2: Number of TLSA names on Dec. 3, 2014. We discovered a few TLSA names outside the zone of `.com` and `.net`, since some mail servers are not in those two TLDs.

4 Observations and Findings

Our measurements provide several key findings: estimates of DANE TLSA deployment, growth, and correctness.

4.1 The Number of TLSA Enabled Names

As of Dec. 3, 2014, PryDane monitors 485k DNSSEC secured `.com` and `.net` zones. Among those, 997 TLSA names are found (Table 2).

Our measurement shows the deployment of DANE TLSA is steady increasing overall (Figure 3), although the fluctuation of the curve also exists, which we think is caused by the experimental deployment and occasional DNS failure. Adding protocols results in jumps in the number of total names that we find on 2014-10-02 and 2014-10-14. We also find that port-443 TLSA names increase faster than port-587 names which does not increase much (almost flat curve).

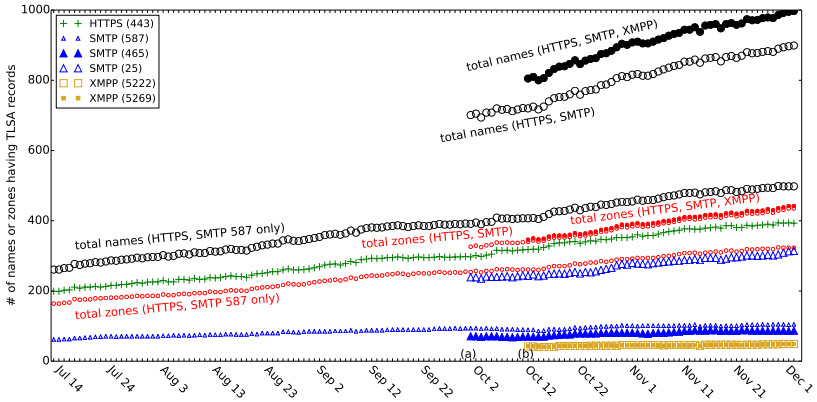


Fig. 3: Number of TLSA names and zones over 142 days. New probes are added during our measurement: (a) add probing SMTP port 25 and 465; (b) add probing XMPP port 5222 and 5269.

zone	**date**	**total**	**DNSSEC**	**TLSA**	P_{dnssec}	P_{tlsa}
com	2014-12-03	115.2M	405k	183	.0035	.0005
net	2014-12-03	15.1M	79k	182	.0053	.0023

Table 3: Sample zone numbers and penetration of DNSSEC and DANE TLSA at the end of our current observation

If we project the current linear trend, the population will double in 6 months. Growth so far is largely linear but our collection methodology will allow longer observation to determine if usage increases and follows an S (sigmoid) curve.

Currently there are only few applications, such an add-on [10] available for common browsers and Postfix mail server [29], supporting DANE TLSA. Deployment of DANE TLSA should pick up quickly as the application support is implemented.

4.2 Compare DANE TLSA and DNSSEC Deployment

To understand the deployment of DANE TLSA, we compare the growth of DNSSEC and DANE TLSA over time. We find DANE TLSA is growing well given it's relative immaturity.

To compare them we consider the penetration of each technology into its base of possible users. We define the penetration of DANE TLSA (P_{tlsa}) as the fraction of the number of TLSA zones over all DNSSEC zones (N_{tlsa}/N_{dnssec}). Since N_{tlsa} is limited by N_{dnssec} (DANE TLSA replies on DNSSEC for authentication), we normalize by the number of DNSSEC zones. We consider a zone to be *TLSA active* if that zone contains at least one TLSA record. Similarly, the

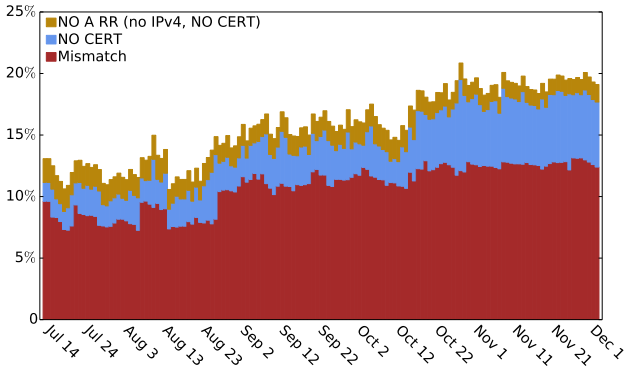


Fig. 4: TLSA validation (ports 443 and 587 only) without cert usage 0 over 142 days. There are consistently 7%-13% TLSA enabled names do not match servers' certificates (lower red bars).

penetration of DNSSEC (P_{dnssec}) is the fraction of zones using DNSSEC over all active zones (N_{dnssec}/N_{all}).

We obtain DNSSEC data from TLDs operators. Combining those with our measurement results, we track P_{tlsa} and P_{dnssec} during our time of observation. The absolute values of penetration are small for both TLDs (less than 0.6%) Table 3, indicating DNSSEC deployment is still modest, 9 years after standardization. Compared to current DNSSEC deployment, DANE TLSA seems promising given its novelty (only 2 years after standardization). We observe $P_{dnssec} > P_{tlsa}$ because of greater DNSSEC maturity. Ideally we would compare the first year of DNSSEC deployment with current DANE TLSA deployment. However, the correct data of early DNSSEC deployment is not available because many zones had DNSSEC signed names before the signing of `.com` and `.net` zones.

4.3 TLSA Record Validation

Having found TLSA records, we also check them for correct usage. We find 7–13% TLSA records are consistently showing invalid over two months (Figure 4). We identify several problems: no IP address, no certificate, or a mismatch between the TLSA record and the certificate at the IP address. We categorize our TLSA validation results in the following.

No IPv4: There are some domain names having an associate TLSA record, but without an `A` record (no IPv4 address). In this case, it's impossible to get a certificate through IPv4, thus no validation could be done. Over 142 days of our observation, 24 unique domain names in total fall into this case. Among those, 5 domain names consistently report no IPv4 addresses every day. To further study the consistent no-IPv4 names, we queried `AAAA` record (IPv6) for those names. We found 3 of them pointing to the same `CNAME` record having a IPv6 address, one of them has an IPv6 address, and the rest don't have IPv6 address.

No certificate: For some TLSA records, we were unable to retrieve the server's certificate. In these cases our call to the `OpenSSL` command timed out. We believe this problem is cased by the remote server, since the probing machine is well connected to internet and has no problems fetching certificates in all the other cases.

Mismatching: Sometimes the TLSA record and certificate exist, but they don't match based on the given options in the TLSA record. We think this is mostly caused by expiration of either certificate or TLSA record, and one of them is not updated correspondingly. This accounts for most of the invalid cases. The lacking of feedback from users also makes web operators pay little attention to those deployed TLSA records, since TLSA is not widely used at this time.

We plan to add more functionality to our TLSA validation process. We validate the certificate through IPv4 addresses after getting a TLSA record. We would like to also check certificates through IPv6, however our probing systems currently sit on a network without global IPv6 reachability. We leave validating IPv6 certificate as a future work. For simplicity, we currently assume DNS integrity without validating DNSSEC chain, and only checks whether the certificate matches the corresponding TLSA records or whether a trust anchor is found, based on the different options. To support DNSSEC validation in our measurement, we plan to use cache to avoid constantly fetching common DNSSEC keys, potentially improving performance. TLSA records in an unsigned domain is also an error because their integrity cannot be protected by DNSSEC. Measuring this kind of error would require probing *all* domains, an expensive task inconsistent with our goal of minimizing network traffic. Exploration of this class of error is possible future work.

There are several websites built to allow one to validate DANE TLSA and DNSSEC by request [10,22,31]. Our measurements complement these on-demand tests and show a broader view of DANE TLSA healthiness.

4.4 Observed TLSA Parameters

TLSA can specify several different trust relationships, such as requiring a specific CA or certificate. More explanation about TLSA option is presented in § 2.3. We next study which are currently in use.

We study the latest one-day sample (Figure 5). We observe that the major group of combination is domain-issued certificate (76%, certificate usage: 3) matching full certificate (71%, selector: 0) with SHA-256 (84%, matching type: 1), and this does not change much over the time of our observation. The dominant use of domain-issued certificate indicates that most DANE TLSA cases are actually independent from CA without serving its trust source. SHA-256 is currently strong enough and it's not necessary to use stronger algorithm bringing more bits in DNS response, causing the problem of larger DNS packets § 4.5. There is a small number (1.5%) of TLSA records using exact matching (matching type: 0) which may bring the problems of large response packets § 4.5. We recommend not to use full certificate matching unless TLSA record is used to deliver the server's certificate.

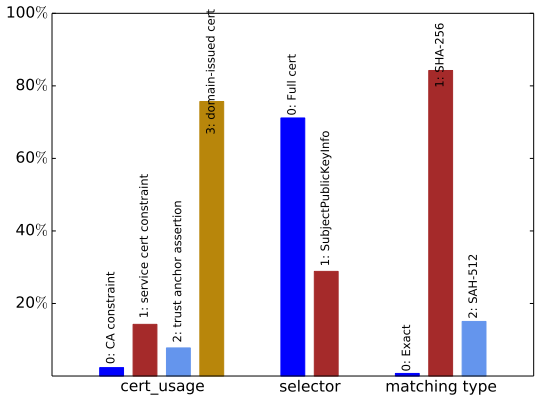


Fig. 5: Distribution of different options in TLSA record. This figure shows one sample of total 1123 TLSA records in 997 TLSA responses captured on Dec. 3, 2014.

4.5 Problematically Large TLSA Packets

When TLSA response blows up to more than 1500 bytes, it suffers IP fragmentation causing various problems: resend-all loss recovery [17], middleboxes block fragments [38], and fragmentation attacks [12]. Large TLSA packets also force DNS to fallback to retry using TCP, and fragments have to be re-assembled, adding the extra resolving latency.

There are several causes leading to large TLSA response. First, a TLSA response can contain multiple TLSA records, either for certificate rollover or for different assertions [14]. In the sample of Dec. 3, 2014, we observe 9.5% out of 997 TLSA responses contain more than one TLSA record. As number of TLSA records increases, the packet size rises correspondingly. Second, the exact matching of certificate in TLSA record without using a hash value adds much more to response packets. We examine the sizes of current SSL certificates by using data collected by Rapid7 Labs [30]. We find median size of X.509 certificate is 774 B, indicating that a TLSA response containing 2 full certificates gets IP fragmented. Third, with DNSSEC enabled and multiple RRs in authority and additional sections, a TLSA response is more likely to be problematically large, which is the common problem of DNS response, not limited to TLSA. To examine the actual TLSA response sizes, we actively query the corresponding authoritative servers for those TLSA names we found. We find that 33% TLSA responses are larger than 1500 bytes, leading to the problems of IP fragmentation (Figure 7). Those large response packets are mostly caused by the several RRs with different names in authority and additional sections.

We suggest that DNS zone operators limit the number of TLSA records for one domain name, use hash matching instead of exact matching, and limit the

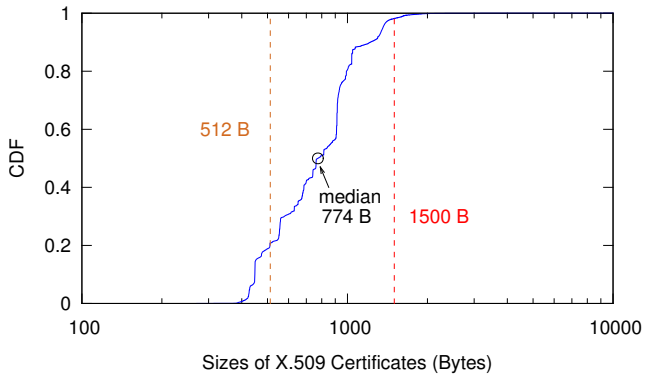


Fig. 6: Cumulative distribution of certificate sizes based on IPv4 SSL certificate data from Rapid7 Labs [30]. Date: 2014-09-29

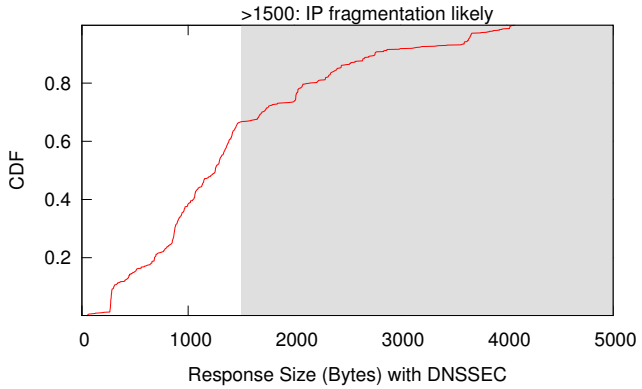


Fig. 7: Cumulative distribution of the response sizes with DNSSEC from authoritative servers of the 997 TLSA names on Dec. 3, 2014

number of RRs in authority and additional sections to avoid future possible IP fragmentation.

4.6 Different Certificates Through IPv4 and IPv6

The difference between IPv4 and IPv6 certificates is problematic for DANE TLSA with usage "domain-issued certificate", because one domain names normally (more than 90% as we observed) has one associated TLSA record, in which case, one TLSA record cannot match two different certificates.

To detect this circumstance (different IPv4 and IPv6 certificates for the same name), we conduct additional measurement from another vantage point with working IPv6 access. (Our main probing server does not have IPv6 connectivity.) For each TLSA enabled name we detect, we actively fetch certificate through

IPv4 and IPv6 if they have one, and we compare the two certificates. As of Oct 1, 2014, we find 238 out of 390 TLSA names have both IPv4 and IPv6 certificates, among which we detect 15 names (under 10 different sub-domain) have different certificates between IPv4 and IPv6. Operators might forget to update one of them when rolling over the certificates, leading to this inconsistency. We suggest domain name owners pay attention attention to this problem if they prepare to deploy DANE TLSA in their domain.

5 Representativeness of Our Results

Although we study two of the largest TLDs (`.com` and `.net`), they are only a subset of the Internet. Some other TLDs have as many or more signed delegations, however those ccTLD zone files are not generally available. We believe the data we study is large enough to provide an overview of current deployment of DANE TLSA. We do not know of any bias in the subset that we measure.

Our dataset is large: as of Dec. 3 2014, there are 115.2M and 15.1M active zones in `.com` and `.net` respectively [36,34].

Second, we probed *all* DNSSEC signed sub-zones from these two TLDs, by extracting *all* DS records in the zone files. On Dec. 3 2014, we probed 405k DNSSEC signed `.com` zones, and 79k signed `.com` zones [35]. We only probe DNSSEC signed zones because DANE relies on DNSSEC for integrity. While a TLSA record can be placed in non-DNSSEC-signed zones, such records are not effective because they lack the integrity verification provided by DNSSEC.

Third, we explore three major secure services (`HTTPS`, `SMTP` and `XMPP`) that are most likely to use TLSA records. Other services using TLS are VPN and SIP applications. However, we know of no deployments using DANE TLSA for them.

6 Conclusion and Future Work

This paper presents the first measurement of DANE TLSA deployment. The main results are summarized as follows. Current CA-based certificate authentication works well in most cases and people don't feel the need to use a completely new authentication protocol, although DANE provides several benefits, such as reducing attack surface and making Secure/Multipurpose Internet Mail Extensions (S/MIME) global deployment possible [25]. Our measurement shows DANE TLSA use is early. However, the increasing trend of DANE TLSA deployment emerges. Our TLSA validation shows current DANE deployment has security inconsistency. Among TLSA records found, there are consistently around 7%-13% TLSA records mismatching server's certificates over the time of our observation. We observed that the most common (71%-84%) usage of TLSA record is: domain-issued certificate matching full certificates with SHA-256. We find 33% TLSA responses suffering IP fragmentation, resulting in fragmentation attacks and additional latency of query processing.

Our monitoring system PryDane is continuously running to keep track of new deployment of DANE. We are working on releasing the source code. (Pseudocode is shown in Figure 2.) We are exploring different services leading to TLSA records deployed in DNS, other than `SMTP` and `HTTPS`. We are also extending PryDane to capture other possible DANE cases, such as OPENPGPKEY [39], and adding IPv6 certificate validation. Our current measurements cover `.com` and `.net` with direct access to the zones; future work may explore other DNSSEC signed zones, or passive DNS analysis of TLSA.

Acknowledgments: Liang Zhu began this work on an internship at Verisign. The work of Liang Zhu and John Heidemann in this paper is partially sponsored by the Department of Homeland Security (DHS) Science and Technology Directorate, HSARPA, Cyber Security Division, via SPAWAR Systems Center Pacific under Contract No. N66001-13-C-3001, and via BAA 11-01-RIKA and Air Force Research Laboratory, Information Directorate under agreement number FA8750-12-2-0344. The U.S. Government is authorized to make reprints for Governmental purposes notwithstanding any copyright. The views contained herein are those of the authors and do not necessarily represent those of DHS or the U.S. Government.

References

1. NLnetLabs. Ldns (ldns-dane). http://www.nlnetlabs.nl/projects/ldns/.
2. R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Dns security introduction and requirements. RFC 4033, Mar. 2005.
3. R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Protocol modifications for the dns security extensions. RFC 4035, Mar. 2005.
4. R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Resource records for the dns security extensions. RFC 4034, Mar. 2005.
5. S. Bhat. Gmail Users in Iran Hit by MITM Attacks. http://techie-buzz.com/tech-news/gmail-iran-hit-mitm.html, Aug 2011.
6. Comodo. Comodo Fraud Incident. https://www.comodo.com/Comodo-Fraud-Incident-2011-03-23.html, Mar 2011.
7. Deploy360 Porgramme. Dane test sites. http://www.internetsociety.org/deploy360/resources/dane-test-sites/.
8. Deploy360 Porgramme. Dnssec statistics. http://www.internetsociety.org/deploy360/dnssec/statistics.
9. T. Dierks and E. Rescorla. The transport layer security (tls) protocol version 1.2. RFC 5246, Aug. 2008.
10. DNSSEC/TLSA Validator. https://www.dnssec-validator.cz.
11. Edward Bjarte Fjellskal. PassiveDNS tool. https://github.com/gamelinux/passivedns.
12. A. Herzberg and H. Shulmanz. Fragmentation considered poisonous. In *Proc. of IEEE Conference on Communications and Network Security (CNS)*, Oct. 2013.
13. P. Hoffman. Smtp service extension for secure smtp over transport layer security. RFC 3207, Feb. 2002.
14. P. Hoffman and J. Schlyter. The dns-based authentication of named entities (dane) transport layer security (tls) protocol: Tlsa. RFC 6698, Aug. 2012.
15. ICANN. The Centralized Zone Data Service. https://czds.icann.org/.

16. Internet Systems Consortium. Internet domain survey. web page http://www.isc.org/solutions/survey accessed January 2008, Jan. 2008.
17. C. A. Kent and J. C. Mogul. Fragmentation considered harmful. *SIGCOMM Comput. Commun. Rev.*, 25(1):75–87, Jan. 1995.
18. I. Learmonth and S. Gunasekaran. Bootstrapping Trust with DANE. https://www.hackerleague.org/hackathons/kings-of-code-hack-battle-at-tnw-europe-conference-2014/hacks/bootstrapping-trust-with-dane, Apr 2014.
19. Mail Server Security Test. https://www.tlsa.info/.
20. P. Mockapetris. Domain names - concepts and facilities. RFC 1034, Nov. 1987.
21. P. Mockapetris. Domain names—implementation and specification. RFC 1035, Nov. 1987.
22. NIST. Danelaw. https://www.had-pilot.com/dane-tests.html.
23. NIST. Estimating ipv6 and dnssec external service deployment status. http://fedv6-deployment.antd.nist.gov.
24. NIST. Tlsa test tree. https://www.had-pilot.com/tlsa-test.html.
25. E. Osterweil, B. Kaliski, M. Larson, and D. McPherson. Reducing the x. 509 attack surface with dnssecs dane. *SATIN: Securing and Trusting Internet Names (March 2012)*, 2012.
26. E. Osterweil, M. Ryan, D. Massey, and L. Zhang. Quantifying the operational status of the dnssec deployment. In *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, IMC '08, pages 231–242, New York, NY, USA, 2008. ACM.
27. E. P. Saint-Andre. Extensible messaging and presence protocol (xmpp): Core. RFC 3920, Oct. 2004.
28. P. Pennock. XMPP & DANE with Prosody. http://bridge.grumpy-troll.org/2014/05/xmpp-dane-with-prosody, May 2014.
29. Postfix. http://www.postfix.org/TLS_README.html.
30. M. Schloesser, B. Gamble, J. Nickel, C. Guarnieri, and H. Moore. Project Sonar: IPv4 SSL Certificates. https://scans.io/study/sonar.ssl, Sept 2014.
31. SIDN labs. Tlsa validator. https://check.sidnlabs.nl/dane.
32. F. Streibelt, J. Böttger, N. Chatzis, G. Smaragdakis, and A. Feldmann. Exploring edns-client-subnet adopters in your free time. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC '13, pages 305–312, New York, NY, USA, 2013. ACM.
33. R. van Rijswijk-Deij, A. Sperotto, and A. Pras. Dnssec and its potential for ddos attacks: A comprehensive measurement study. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, IMC '14, pages 449–460, New York, NY, USA, 2014. ACM.
34. Verisign. Daily zone counts. http://www.verisigninc.com/en_US/channel-resources/domain-registry-products/zone-file-information/index.xhtml.
35. Verisign. Dnssec scoreboard. http://scoreboard.verisignlabs.com.
36. Verisign. The Domain Name Industry Brief. www.verisigninc.com/assets/domain-name-report-december2014.pdf, December 2014.
37. Verisign Labs. Dane/tlsa demonstration. http://dane.verisignlabs.com/.
38. N. Weaver, C. Kreibich, B. Nechaev, and V. P. xson. Implications of Netalyzr's DNS measurements. In *Proc. of Workshop on Securing and Trusting Internet Names (SATIN)*, Apr. 2011.
39. P. Wouters. Using dane to associate openpgp public keys with email addresses. Work in progress (draft-wouters-dane-openpgp-02), Feb. 2014.