DNSSEC als Alternative zur klassischen CA

Daniel Feuchtinger, Helmut Reiser, Bernhard Schmidt[1]

Abstract: Der Betrieb klassischer Certificate-Authorities (CAs) wird organisatorisch immer aufwändiger. Die Komplexität, die Abhängigkeit von Browser-Herstellern durch eine Verankerung der Root-Zertifikate sowie diverse Sicherheitsvorfälle mit CAs und eine Vielzahl von „unbekannten" Root-Zertifikaten im Certificate-Store der Browser führen zu zunehmender Kritik an den etablierten Verfahren.

In dieser Arbeit wird untersucht, inwieweit DNSSEC mit seinen Mechanismen und Erweiterungen die Aufgaben einer klassischen CA übernehmen kann und welche Auswirkungen sich beim Einsatz solcher Zertifikate ergeben. Im Bereich der E-Mail Verschlüsselung lassen sich durch DNSSEC z.B. Zertifikate und Schlüssel sehr einfach verteilen, sind auch für Empfänger fremder Domains einfach zu nutzen und neben einer automatischen Verschlüsselung zwischen Mailservern ist auch eine vollautomatische Ende-zu-Ende Verschlüsselung möglich.

Keywords: DNSSEC; DANE; CA; PKI; Zertifikate

1 Einleitung

Die durch klassische Certificate-Authorities (CAs) bereitgestellte Public-Key-Infrastructure (PKI) hat viele technische und organisatorische Nachteile. Die immer länger werdenden und trotzdem unvollständigen Certificate-Revocation-Lists (CRLs), das Online-Certificate-Status-Protocol (OCSP) und die Pflege der Root-Zertifikate etwa in Browsern oder Betriebssystemen sollen hier als Beispiele für technische Probleme dienen. Die indifferente Verteilung von Vertrauen an eine Vielzahl von weitgehend unbekannten CAs und der damit fast zwangsläufig einhergehende Vertrauensbruch von Seiten einiger, nicht nur kleiner CAs [Fo11], sowie die Macht, die vor allem die Browser-Hersteller darüber haben, welchen CAs vertraut wird, sollen hier stellvertretend für die organisatorischen Probleme genannt werden. Das CA/Browserforum[2] definiert Richtlinien die von CAs zu erfüllen sind, damit deren Root-Zertifikate mit den Browsern ausgeliefert (verankert) werden. Die damit verbundenen organisatorischen und technischen Anforderungen haben sich in den letzten Jahren massiv verschärft. So wurden bspw. die Revalidierungszeiträume von 39 Monaten auf 825 Tage verkürzt. Die CA muss ihre Sicherheitsprozesse einmal jährlich nach den in [CA17] Abschnitt 17.1 angegebenen Standards (z.B. [ET13]) unabhängig auditieren lassen. Diese

[1] Leibniz-Rechenzentrum, Boltzmannstr. 1, 85748 Garching, daniel.feuchtinger@lrz.de, helmut.reiser@lrz.de, bernhard.schmidt@lrz.de

[2] https://cabforum.org

Regelungen können mit jeder Änderung der Guidelines verschärft werden und es droht der Verlust der Browserverankerung der Root-CAs bei Fehlern. Gleichzeitig versuchen einzelne Browser-Hersteller durch ihre Marktmacht, weitere Standards zu setzen oder Regelungen des CA/Browserforums in Frage zu stellen. Google implementiert in Chrome Certificate Transparancy [Ce17] und Mozilla stellt eine Auditierung nach dem ETSI Standard in Frage [Gr17]. Für Betreiber von CAs ergibt sich dadurch ein hohes Maß an Unsicherheit und das latente Risiko, dass Root-Zertifikate aus Browsern genommen werden und damit die Validierung für alle von dieser CA ausgestellten Zertifikate erheblich erschwert wird.

Das Domain-Name-Systen (DNS) ist ohnehin essentieller Bestandteil der klassischen PKI (OCSP baut auf DNS, Lets Encrypt validiert optional via DNS etc.). Durch DNSSEC werden die DNS-Informationen kryptographisch abgesichert und darauf aufbauend können mit DANE Zertifikate direkt kryptographisch sicher validiert werden. Auf den ersten Blick kann DNSSEC und DANE also eine klassische CA ersetzen. Welche Vorteile eine DNSSEC-DANE-PKI gegenüber den klassischen CAs hat und welche Voraussetzungen erfüllt sein müssen, um die klassische CA zu ersetzen, soll hier dargestellt werden.

2 Einführung DNSSEC und DANE

Im folgenden Abschnitt werden die Grundlagen für einen CA Einsatz von DNSSEC [GH10] und DANE [HS12] vorgestellt. Ist eine Domain DNSSEC-validiert, so ist jeder Teil des DNS-Baumes von der Wurzel bis zu dieser Domain kryptographisch mit Signaturen abgesichert. Die Baumstruktur des DNS legt die Zuständigkeiten für Signaturen und den Validierungspfad eindeutig fest. Es gibt nur eine „Root-CA“, die von der Internet Corporation for Assigned Names and Numbers (ICANN) verwaltete Root-Zone, alle weiteren CAs sind Sub-CAs mit eindeutig festgelegten Zuständigkeiten. Der öffentliche Schlüssel der Root-Zone ist der einzige Schlüssel, der initial verteilt werden muss. Das Beispiel lrz.de soll die Struktur einer DNSSEC-PKI verdeutlichen (Abbildung1). Für die Vertrauenskette ist neben der ICANN noch das Deutsche Network Information Center (DENIC), das die de-Zone verwaltet, und das LRZ, das die lrz.de-Zone und alle darunter liegenden Zonen verwaltet, zuständig. ICANN und DENIC haben nur die Aufgabe, die Vertrauenskette zum LRZ herzustellen, die Verwaltung der Zertifikate liegt organisatorisch ausschließlich beim LRZ. Technisch haben die Institutionen, die Teil der Kette sind (und nur die!), die Möglichkeit, diese Kette zu ändern und damit Zertifikate zu fälschen, das muss durch Verträge ausgeschlossen werden. Bei klassischen CAs haben alle Root-CAs (und ggf. auch Sub-CAs) die technische Möglichkeit, Zertifikate zu fälschen.

DNS-based Authentication of Named Entities (DANE) [DH15b; HS12] setzt auf DNSSEC [GH10] auf und verwendet DNS-Resource-Records, um Schlüsselinformationen zu speichern, z.B. vom Typ TLSA zur Veröffentlichung von Server-Zertifikaten oder vom Typ SMIMEA zur Veröffentlichung von S/MIME-Zertifikaten für die E-Mail-Verschlüsselung und Signatur (SMIMEA ist nicht Teil von DANE, sondern eine Ergänzung [HS17]).

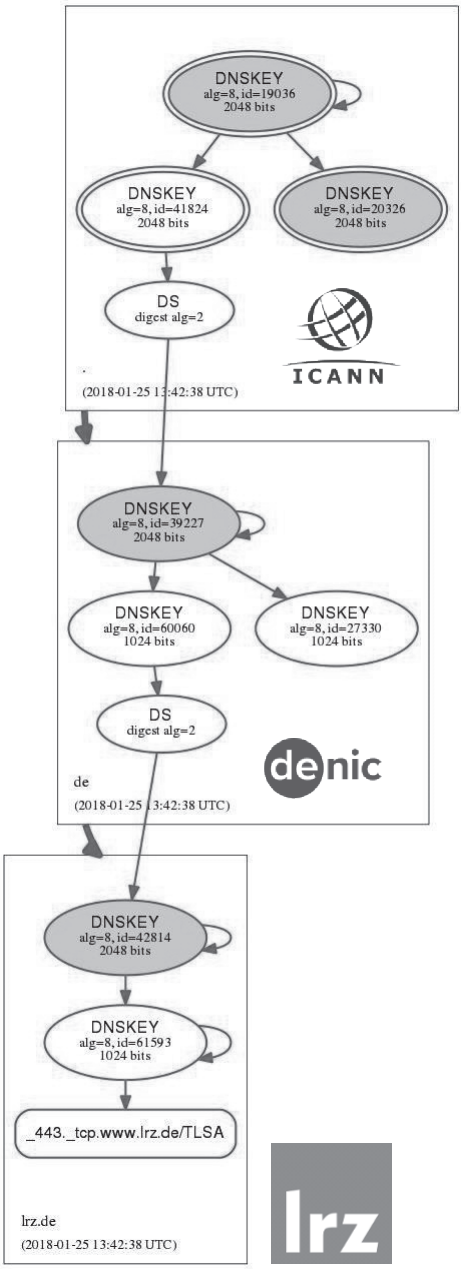

Abb. 1: DNSSEC/DANE Vertrauens-Struktur für den TLSA-Record zu www.lrz.de

Die dünnen Pfeile zeigen den technischen Validierungspfad mittels Signaturen und signierter Hashes, die dicken Pfeile zeigen die daraus folgende Delegation von Vertrauen auf Institutionsebene.

Ein Zertifikat kann durch Hinterlegen

0. des öffentlichen Schlüssels des Zertifikats im DER-Format oder
1. des Zertifikats im X.509 bzw. Public Key Infrastructure Exchange (PKIX) Format

(optional auch als sha256- oder sha512-Hash) im DNS in Form eines TLSA-Records referenziert werden. Die eigentliche Validierung erfolgt über eine der folgenden Varianten, die als Parameter (Usage Field) im TLSA-Record festgelegt wird (Nomenklatur siehe [Gu14]).

0. **CA Constraint (PKIX-TA)**: Der TLSA-Record gibt die vertrauenswürdige CA an, d.h. die CA, auf die das Zertifikat über die PKIX-Certification-Path-Validation zurückgeführt werden muss. Hier findet ein CA-Pinning durch DNSSEC/DANE statt.
1. **Service Certificate Constraint (PKIX-EE)**: Der TLSA-Record gibt das zu verwendende Zertifikat an, wobei zusätzlich die PKIX-Certification-Path-Validation durchgeführt werden muss. Hier findet ein Zertifikats-Pinning durch DNSSEC/DANE statt.
2. **Trust Anchor Assertion (DANE-TA)**: Das im TLSA-Record referenzierte Zertifikat dient als Trust-Anchor, d.h. ein Zertifikat ist vertrauenswürig, wenn es über eine Zertifikatskette auf den Trust-Anchor zurückführbar ist.
3. **Domain-Issued Certificate (DANE-EE)**: Dem über den TLSA-Record referenzierten Zertifikat wird vertraut, ohne weitere PKIX-Validierung. Damit kann auch die Prüfung der Gültigkeitsdauer und der Zertifikatssignatur entfallen, die Gültigkeitsdauer des Zertifikats wird durch die Gültigkeitsdauer des DNS-Records bzw. die der Signaturen festgelegt.

Es können mehrere TLSA-Records pro Domain angegeben werden. Um die Verwendung genauer zu spezifizieren, ist es vorgesehen, Protokoll und Port per DNS anzugeben: _port._protocol.domain. Um z.B. ein Zertifikat für den TCP-Port 443 des Hosts www.lrz.de anzugeben, wird folgender DNS-Knoten verwendet: _443._tcp.www.lrz.de.

Personenzertifikate sind nicht im DANE-RFC spezifiziert und mit den angesprochenen Mitteln ist es auch nicht möglich, eine Person im DNS abzubilden. Mit SMIMEA [HS17] ist ein Standard verfügbar, der einen DNS-Record spezifiziert, mit dessen Hilfe E-Mail-Adressen zu einer Domain referenziert werden können. Die Möglichkeiten, Zertifikate im DNS anzugeben werden vollständig von DANE übernommen, an die Stelle von _port._protocol in der Domain für den TLSA-Record kommt für den SMIMEA-Record der lokale Teil der E-Mail-Adresse als Hash (SHA256), ergänzt um `._smimecert`. Das Hashing löst einerseits Probleme der Zeichenkodierung (DNS ist hier wesentlich restriktiver als E-Mail), und kann andererseits

das Auslesen von Mail-Adressen via DNS verhindern. Ein Beispiel für einen SMIMEA-Record, der der E-Mail-Adresse hugh@test das Zertifikat des LRZ-Webservers zuordnet: `c93f1e400f26708f98cb19d936620da35eec8f72e57f9eec01c1afd6._smimecert.test. 86400 IN SMIMEA 3 1 1 71175DD9FE14CD41F1366BF5E82E25BCFFB46457BD7BF9AAEA37EFE443B0A449` Der SMIMEA-Record besagt, dass Mails von hugh@test mit dem im Record angegebenen Zertifikat signiert, und Mails an hugh@test mit dem öffentlichen Schlüssel aus dem Zertifikat verschlüsselt werden dürfen. Es können mehrere Zertifikate für eine E-Mail-Adresse via SMIMEA-Record angegeben werden. Ist die E-Mail-Adresse eindeutig einer Person zugeordnet (etwa durch eine Policy), so kann das im SMIMEA-Record referenzierte Zertifikat als Personenzertifikat betrachtet werden.

3 DNSSEC und DANE als PKI

DNSSEC und DANE stellen das Rüstzeug, um Zertifikate kryptographisch abgesichert einer Domain zuzuordnen. Das Ausstellen von Zertifikaten für eine Domain erfolgt durch das Anlegen und signieren von TLSA- oder SMIMEA-Records auf den für diese Domain autoritativen DNS-Servern. Beliebige Zertifikate können auf diesem Weg gleichzeitig signiert und veröffentlicht werden. Dabei ist zu beachten, dass die Bedeutung einiger Felder im Zertifikat ersetzt wird, wenn nicht TLSA-Records der Nutzungsvarianten PKIX-TA oder PKIX-EE (s. Abschnitt 2) verwendet werden. Diese erhalten die Bedeutung aller Zertifikatsfelder, da TLSA hier nur zusätzlich zur PKIX–Validierung gedacht ist. Bei den anderen Nutzungsvarianten ersetzen Informationen aus dem DNS einige Felder, die wichtigsten Beispiele sind:

- Die Gültigkeitsdauer des Zertifikats bestimmt sich durch die Gültigkeit der Signatur des TLSA-Records (RRSIG-Record).
- Der Common Name (CN) des Zertifikats wird durch die Domain des TLSA-Records ersetzt.
- Alternative Namen werden ignoriert, das Zertifikat kann aber auch über weitere TLSA-Records für weitere Domains verwendet werden.
- Die Signatur im Zertifikat wird gar nicht, teilweise oder vollständig geprüft, je nachdem, welche Nutzungsart der TLSA-Record definiert.
- Die Nutzungseinschränkung im Zertifikat wird ignoriert, eine Einschränkung ist durch _port._protocol gegeben.

Die DNSSEC/DANE-Validierung übernimmt ein DNSSEC-validierender DNS-Resolver und signalisiert dies über das AD-Flag in der DNS-Antwort. Die Kommunikation zwischen dem DNS-Resolver und dem Client ist nicht gegen Angriffe abgesichert, daher muss der Weg zwischen dem Resolver und dem Client sicher sein, zum Beispiel durch VPN, TSIG,

DNS-over-TLS oder am einfachsten durch die Nutzung eines lokalen validierenden Resolvers auf dem Client.

Ab Version 9.11 unterstützt der ISC BIND SMIMEA-Records, damit ist der produktive Einsatz auf einem authoritativen DNS-Server möglich. Resolver sind für unbekannte Resource Records im Allgemeinen transparent. Die Client-Unterstützung ist noch nicht so weit entwickelt, für Thunderbird gibt es das Great-DANE-Plugin[3] und für Mail-Server gibt es Filter, die aus- und eingehende Mails automatisch via SMIMEA-Record verschlüsseln, letzteres wird auch schon von Dienstleistern angeboten[4].

Ein Zertifikat ist aus DANE-Perspektive solange gültig, wie es per TLSA-Record validiert werden kann, ein Rückruf erfolgt durch das Löschen des TLSA-Records inklusive der dazugehörenden RRSIG-Records. Entscheidend für den Zeitpunkt der Wirksamkeit des Rückrufs ist die Gültigkeitsdauer der Signaturen des TLSA-Records, und der DNS-Records, die bei der Validierung verwendet werden (das sind mindestens noch die Delegations-Records). Nach Ablauf der Gültigkeit aller TLSA-Signaturen (durch wiederkehrendes Signieren der DNS-Zonen können mehrere Signaturen im Umlauf sein) kann dieses Zertifikat nicht mehr verwendet werden.

Ist ein DNSKEY kompromittiert und kann ein Angreifer beliebige Zertifikate für die betroffenen Domain inklusive Subdomains „ausgeben“, so ist das mit der Kompromittierung eines privaten CA-Schlüssels bei PKIX vergleichbar. Alle RRSIG-Records (d.h. die DNSSEC-Signaturen, auf denen das Vertrauen in die Zertifikate beruht) sind nicht mehr vertrauenswürdig und ein DNSSEC-Key-Rollover muss durchgeführt werden. Die RRSIG-Records zu den TLSA-Records, d.h. auch möglicherweise gefälschte Zertifikate, sind spätestens nach einem abgeschlossenen Key-Rollover wirksam zurückgerufen. Daraus ergibt sich ein Vorteil gegenüber den klassichen CAs, vorausgesetzt, man nutzt TLSA-Records nicht als Trust-Anchor (PKIX-TA oder DANE-TA): Bei der klassischen CA müssen alle ausgegebenen Zertifikate ausgetauscht werden, wenn der Signaturschlüssel kompromittiert wurde, bei DNSSEC/DANE genügt ein Key-Rollover. Die vor der Kompromittierung gültigen Zertifikate bzw. TLSA-Records können alle mit neuen DNSSEC-Signaturen weiter verwendet werden.

DNSSEC sieht nicht vor, widerrufene Zertifkate zu speichern, es gibt also keine wachsende Liste an zurückgerufenen und abgelaufenen Zertifikaten die gepflegt werden muss. Will man ein neues Zertifikat bzw. Schlüsselpaar verwenden, das alte aber noch für die Validierung von Signaturen behalten, so kann man das alte in Form eines Hashes verfügbar lassen und nur das neue in Form eines vollständigen Zertifikats oder öffentlichen Schlüssels zur Verfügung stellen. Kompromittierte Schlüssel und die dazugehörenden Zertifikate werden vollständig aus dem DNS gelöscht und sind damit im Nachhinein nicht mehr zuzuordnen.

Klassische CAs bieten die Möglichkeit, Signaturen mit zurückgerufenen Zertifikaten zu

[3] https://addons.mozilla.org/de/thunderbird/addon/great-dane-smim/e

[4] https://www.tb-itf.de/faq-eintrag/automatische-verschluesselung-mit-smimea.html

validieren, allerdings stellt sich hier die Frage nach dem Sinn, da es keine Möglichkeit gibt, das Mindestalter einer Signatur nachzuweisen. Es ist also nicht möglich festzustellen, dass eine Signatur vor der Kompromittierung des Schlüssels erstellt wurde und damit ist die Signatur wertlos geworden.

Wird trotzdem eine Zertifikats-History in Form der zurückgerufenen Zertifikate benötigt, so müssen eigene Lösungen gefunden werden.

4 Gegenüberstellung von DNSSEC/DANE und klassischen CAs

Für den Vergleich von DNSSEC/DANE und klassischen CAs werden hier einige der jeweils interessanten Voraussetzungen umrissen und anschließend Vor- und Nachteile diskutiert. Die kryptographischen Details werden nicht angesprochen, da sich DNSSEC und die klassischen CAs weitgehend auf die selben Algorithmen stützen.

Um ein Zertifikat gegen die klassischen CAs validieren zu können, müssen die Root-Zertifikate der CAs lokal verfügbar und aktuell sein. Zurückgerufene Zertifikate müssen gepflegt werden, beispielsweise in Form von Certificate-Revocation-Lists (CRLs), die mit der Zeit immer weiter anwachsen können und jederzeit verfügbar sein müssen. Die daraus resultierenden Probleme sollte das Online-Certificate-Status-Protocol (OCSP) lösen, das den Zertifikatsstatus dynamisch abfragbar macht. Dazu muss ein Server, ein sog. OCSP-Responder, der über widerrufene Zertifikate Auskunft gibt, dauerhaft und überall verfügbar sein. Letzteres ist schon ohne potentielle Angreifer ein schwer zu lösendes Problem. CRL-Verteilungspunkte und OCSP-Responder werden in den Zertifikaten als URL angegeben und hängen damit beide vom DNS ab. OCSP ist also über Routing und DNS angreifbar, die Anfragen des Clients können umgelenkt werden. Da OCSP ein „try-again-later“ vorsieht, welches nicht signiert werden muss [Ma09; Sa13] und welches client-seitig oft nicht zu einem Fehler führt, kann OCSP über einen Man-in-the-Middle-Angriff vollständig ausgehebelt werden. OCSP kann auch für Tracking missbraucht werden, da für jede TLS-Verbindung eine Verbindung zum OCSP-Responder aufgebaut werden muss. Aufgrund dieser Schwächen führt Chrome standardmäßig keine OCSP-Abfragen durch sondern verwendet vom Chrome-Projekt gepflegte(!) CRL-Sets, die sporadisch über Software-Updates installiert werden müssen.

Die Verwendung von DNSSEC/DANE als CA erfordert die Validierung und damit auf Client-Seite einen DNSSEC-validierenden DNS-Resolver. Insbesondere für mobile Geräte ist ein validierender DNS-Resolver in „sicherem Abstand“ nicht immer verfügbar, daher ist die Integration eines validierenden Resolvers im Betriebssystem eine Voraussetzung für eine breite Anwendung von DNSSEC-DANE als CA. Für Server-Anwendungen ist ein lokaler validierender DNS-Resolver kein Problem, daher ist der Einsatz einer DNSSEC/DANE-CA hier einfacher. Client-Anwendungen müssen die Verwendung von DANE unterstützen. Während die Unterstützung PKIX weit verbreitet ist, wird DANE in größerem Umfang derzeit nur für die Kommunikation von SMTP-Servern untereinander

verwendet. Für Firefox und Chrome gibt es ein DANE-Plugin, das aber die Validierung über die klassischen CAs nicht ersetzt. Scheitert die PKIX-Certification-Path-Validation, so wird vom Browser die übliche Warnung angezeigt, auch wenn die Validierung über das DANE-Plugin erfolgreich war. Eine DNSSEC/DANE-CA kann jeder selbst betreiben und die Sicherheitsanforderungen für die eigene Zone definieren und überwachen. Die Betreiber der Root- und der Second-Level-Domain-Zonen sind allerdings festgelegt.

Klassische CAs	
Vorteile	**Nachteile**
• Breite Unterstützung durch Betriebssysteme und Browser • Unabhängigkeit von sicherem Netz	• Root-CA-Zertifikate müssen auf allen Endgeräten gepflegt werden • Zertifikatsrückruf ist angreifbar • Vertrauen breit gestreut (Beispiel Firefox: über 150 Root-CA-Zertifikate, über 60 CAs[5] • Hohe technische Komplexität • Mißbrauchsfälle • Hoher, und weiter steigender Akkreditierungsaufwand • Abhängigkeit von Browser- und Betriebssystemherstellern • Kosten • Zuständigkeit unklar (auch mit CAA-Records!) • Kompromittierung des Signierschlüssels erfordert Austausch aller Zertifikate

DNSSEC/DANE	
Vorteile	**Nachteile**
• Baut auf vorhandene Infrastruktur auf • Zertifikatsrückruf funktioniert • Geringe Kosten, wenn DNSSEC schon vorhanden ist • Eigene Domains können flexibel und unabhängig selbst, oder von Dienstleistern (ohne Akkreditierung) verwaltet werden • Unabhängigkeit von sicherem Netz (mit lokalem Resolver) • Zuständigkeit klar • Kompromittierung des Signierschlüssels erfordert keinen Austausch von Zertifikaten	• Unterstützung durch Anwendungen fehlt • Zeitverzögerter Widerruf • Zurückgerufene und abgelaufene Zertifikate sind nicht über DNS verfügbar • Abhängigkeit von den Verwaltern der Root-Zone und der Top-Level-Domain-Zonen • Vertrauenswürdige validierende DNS-Resolver sind nicht überall verfügbar

[5] https://wiki.mozilla.org/CA

Eine private CA ist auch mit DNSSEC/DANE möglich, hat aber ähnliche Nachteile, wie eine private klassische CA, bei der das Root-Zertifikat auf den relevanten Clients installiert werden muss. Für eine private DNSSEC/DANE-CA muss ein validierender DNS-Resolver mit DNSSEC-Lookaside-Validation verwendet werden, der für eine bestimmte Zone einen eigenen DNSSEC-Trust-Anchor definiert.

5 Anwendungsfall E-Mail

Die Möglichkeiten der DNSSEC/DANE-CA werden hier für die Signatur und Verschlüsselung von E-Mails erörtert. Die Verwendung von DANE für SMTP-Server-Kommunikation ist bereits etabliert und in einem eigenen RFC [DH15a] beschrieben.

DNS mit DNSSEC gesicherten SMIMEA-Records kann die Funktion einer Zertifikatsdatenbank übernehmen und erlaubt das (auch automatisierte) Nachschlagen von Zertifikaten. Die Validierung eines Zertifikats kann eine klassische CA übernehmen, sie kann aber auch ausschließlich durch DNSSEC/DANE und unabhängig von einer klassischen CA erfolgen (SMIMEA-Nutzung DANE-TA und DANE-EE).

Auswertung des SMIMEA-Records durch Mail-Server: Eine vergleichsweise einfach umzusetzende Variante ist der Einsatz auf Mail-Servern, da ein lokaler validierender DNS-Resolver einfach eingerichtet, und nur auf den Mail-Servern notwendig ist. Der SMTP-Server kann so konfiguriert werden, dass jede ausgehende Mail verschlüsselt wird, wenn ein DNSSEC-gesicherter SMIMEA-Record für den Empfänger verfügbar ist. Für jede eingehende Mail kann genauso verfahren werden. Die Mails liegen dann automatisch verschlüsselt auf dem Server und können erst durch den Empfänger entschlüsselt werden. Passiert das auf dem Mail-Submission-Server, so geht die Mail nicht im Klartext über das Netz (vorausgesetzt, der Mail-Client verwendet TLS für die Verbindung zum Mail-Submission-Server) und wird online nur verschlüsselt gespeichert. Der SMIMEA-Record bietet keine Möglichkeit, die Verwendung zu differenzieren, also z.B. ein Flag „immer verschlüsseln“, oder „nur für Signaturverifikation“ zu setzen. Über die Art des Records lässt sich das aber zumindest teilweise erreichen, etwa indem nur ein Fingerprint hinterlegt wird, mit welchem zwar nicht automatisch verschlüsselt, aber eine Signatur überprüft werden kann. Die Wahlmöglichkeiten könnte man auch für den Nutzer konfigurierbar machen, sowohl beim Versenden, d.h. auf dem Mail-Submission-Server, als auch für Postfächer, d.h. für alle eingehenden Mails. Mailfilter für diese Szenarien sind bereits im Einsatz[6][7].

Auswertung des SMIMEA-Records durch Mail-Clients: Die Unterstützung von SMIMEA durch Mail-Clients ist noch weniger ausgereift, als die für Mail-Server. Auch hier

[6] https://www.tb-itf.de/faq-eintrag/automatische-verschluesselung-mit-smimea.html

[7] https://www.heise.de/newsticker/meldung/DANE-Automatische-Mail-Verschluesselung-mit-S-MIME-3041530.html

ist ein Problem die Verfügbarkeit eines vertrauenswürdigen validierenden DNS-Resolvers. Für Thunderbird gibt es ein vielversprechendes Plugin Great DANE[8], das die Möglichkeit bietet, Zertifikate für Empfänger zur Verschlüsselung, und für Absender zur Verifikation der Signatur automatisch via SMIMEA-Records zu finden. Die Great-DANE-Engine[9] ist auch unabhängig von Thunderbird nutzbar und kann z.B. in Webmail-Software integriert werden.

DNSSEC/SMIMEA-CA: Aus Sicht eines CA-Betreibers können SMIMEA-Records auf drei Arten eingesetzt werden:

1. Aufbauend auf eine klassische CA, jedes ausgegebene Zertifikat kann (ggf. auf Wunsch) als SMIMEA-Record veröffentlicht werden (PKIX-EE, ggf. zusätzlich PKIX-TA).
2. Aufbauend auf eine Self-Signed-CA, die SMIMEA-Records dienen hier mit DNSSEC als kryptographische Absicherung. (DANE-EE, ggf. zusätzlich DANE-TA)
3. Der Nutzer kann ein beliebiges Zertifikat vorlegen und über einen SMIMEA-Record veröffentlichen lassen (DANE-EE).

In den ersten beiden Varianten muss der Nutzer ein Certificate-Signing-Request (CSR) einreichen und z.B. persönlich seine Identität bestätigen, bevor ein Zertifikat ausgegeben und im DNS veröffentlicht wird. Mit der zweiten Variante hat man die Möglichkeit, die Art der per DNS ausgegebenen Zertifikate zu kontrollieren und den Einsatz zu protokollieren, analog zur klassischen CA (Zeitpunkt der Ausgabe, Zeitpunkt der Löschung aus dem DNS), auch wenn für den Einsatz (zumindest bei DANE-EE) eine Signatur durch die Self-Signed-CA überflüssig ist, da die Authentizität des Zertifikats ausschließlich durch DNSSEC verifiziert wird. Bei der dritten Variante ist es möglicherweise sinnvoll, über eine Policy einige Anforderungen an ein Zertifikat zu formulieren, z.B. um den Inhalt der (nicht verwendeten) Zertifikatsfelder unauffällig zu halten. Wenn diese Anforderungen automatisiert geprüft werden können, ist es sogar möglich, dass jeder Nutzer sein Zertifikat selbst hochlädt, nachdem er sich authentisiert hat.

6 Anwendungsfall Webserver-Zertifikat

Die Möglichkeiten von DNSSEC/DANE für Webserver-Zertifikate sind analog zur DNSSEC/SMIMEA-CA (vgl. Punkt 1 bis 3 des vorherigen Abschnitts). Da die verbreiteten Webbrowser DNSSEC/DANE nicht unterstützen und das Plugin für Firefox und Chrome die klassische PKIX-Verifikation nicht ersetzt, sondern nur ergänzt, ist der Nutzen einer DNSSEC/DANE-CA geringer, als bei DNSSEC/SMIMEA. DNSSEC/DANE bietet hier

[8] https://addons.mozilla.org/de/thunderbird/addon/great-dane-smime/
[9] https://github.com/grierforensics/Great-DANE-Engine

bisher „nur“ eine zusätzliche Sicherheit, allerdings auch keine Nachteile (bis auf die Pflege der TLSA-Records). Somit ist nur die erste Variante für Webserver geeignet, die für eine breitere Öffentlichkeit gedacht sind, Varianten 2 und 3 können nur als Spezialfälle dienen, wo bei den Nutzern der Seite das DANE-Plugin, ein validierender DNS-Resolver, und entsprechendes Wissen vorausgesetzt werden können (der Browser wird eine Seite ohne PKIX-Certification-Path-Validation als unsicher markieren und eine Warnung ausgeben, auch mit DANE-Plugin), was eine breite Anwendung in dieser Form verhindert. Abhilfe könnten natürlich die Browserhersteller durch Integration von DNSSEC/DANE in die Browser schaffen, oder durch ein Plugin, das die klassische PKIX-Validierung ersetzen kann.

7 Zusammenfassung

DNS-Daten sind essentiell für alle Internetanwendungen, der Zugriff auf das DNS muss daher hoch verfügbar sein. In vielen Fällen, nicht zuletzt für klassische CAs, sind DNS-Daten von sicherheitsrelevanter Bedeutung und schützenswert. Die Verbreitung von DNSSEC wird weiter zunehmen und die Datenbankfunktion des DNS für sicherheitsrelevante Anwendungen wie die Verteilung öffentlicher Schlüssel und Zertifikate verfügbar machen, während die Validierung der Zertifikate noch eine klassische CA übernimmt.

Mit der bereits vorhandenen Zuordnung von Domains zu Inhabern ist eine Aufgabe einer CA erfüllt: Der Inhaber der Domain kann die Verantwortung für Zertifikate und Schlüssel zu dieser Domain übernehmen bzw. an Dritte (z.B. den Betreiber einer klassischen CA) delegieren. In einem weiteren Schritt kann DNSSEC/DANE dann die Validierung von Zertifikaten übernehmen und bildet damit eine hochverfügbare, einfache und elegante CA, die mit nur einem Root-Zertifikat auskommt und mit einer weltweit verfügbaren Datenbank für Schlüssel und Zertifikate besticht. Dafür ist allerdings noch eine weitere Verbreitung von DNSSEC und validierenden Resolvern nötig. Ein entscheidender Schub könnte etwa durch die Unterstützung der Browser, oder der Smartphonebetriebssysteme kommen. Letzere bringen jetzt schon oft einen eigenen DNS-Resolver mit. Könnte dieser DNSSEC validieren, möglicherweise sogar selektiv (je nach Anforderung einer App), so würde das die DNSSEC/DANE-CA einem großen Anwendungskreis verfügbar, und die Ersetzung der klassischen CA realistisch machen. In kontrollierten Umgebungen wird DNSSEC/DANE bereits produktiv eingesetzt, wie z.B. in einem bayernweiten Projekt [Du17] zur sicheren Mail-Server-Kommunikation nach RFC7672 [DH15a]. Ein weiteres Anwendungsfeld könnte die automatische Verschlüsselung von Nutzer-Emails durch Mail-Submission-Server sein.

Literatur

[CA17] CA/Browser Forum: Guidelines For The Issuance And Management Of Extended Validation Certificates, Techn. Ber. Version 1.6.6, CA/Browser Forum, 2017, URL: https://cabforum.org/wp-content/uploads/EV-V1_6_6.pdf.

[Ce17] Certificate Transparency: What is Certificate Transparency?, Techn. Ber., Google, 2017, URL: https://www.certificate-transparency.org/what-is-ct.

[DH15a] Dukhovni, V.; Hardaker, W.: SMTP Security via Opportunistic DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS), RFC 7672, RFC Editor, Okt. 2015.

[DH15b] Dukhovni, V.; Hardaker, W.: The DNS-Based Authentication of Named Entities (DANE) Protocol: Updates and Operational Guidance, RFC 7671, RFC Editor, Okt. 2015.

[Du17] Duscha, S.; Schmidt, B.; Feuchtinger, D.; Reiser, H.: Einführung von DNSSEC und DANE im Bayerischen Hochschulnetz. In (Eibl, M.; Gaedke, M., Hrsg.): INFORMATIK 2017. Gesellschaft für Informatik, Bonn, S. 763–772, 2017.

[ET13] ETSI: Electronic Signatures and Infrastructures (ESI); Policy requirements for certification authorities issuing public key certificates, Techn. Ber. ETSI 102 042 V 2.4.1, European Telecommunications Standards Institute (ETSI), 2013.

[Fo11] Fox-IT), J. P. (: Interim Report, DigiNotar Certificate Authority breach, "Operation Black Tulip", Techn. Ber., FOX-IT, 2011, URL: https://cryptome.org/0005/diginotar-insec.pdf.

[GH10] Gould, J.; Hollenbeck, S.: Domain Name System (DNS) Security Extensions Mapping for the Extensible Provisioning Protocol (EPP), RFC 5910, RFC Editor, Mai 2010.

[Gr17] mozilla.dev.security.policy Group: ETSI audits not listing audit periods, Techn. Ber., Google Groups, 2017, URL: https://groups.google.com/forum/#!msg/mozilla.dev.security.policy/zAEoGqdtl6E/Q_Fr41V3BAAJ.

[Gu14] Gudmundsson, O.: Adding Acronyms to Simplify Conversations about DNS-Based Authentication of Named Entities (DANE), RFC 7218, RFC Editor, Apr. 2014.

[HS12] Hoffman, P.; Schlyter, J.: The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA, RFC 6698, http://www.rfc-editor.org/rfc/rfc6698.txt, RFC Editor, Aug. 2012, URL: http://www.rfc-editor.org/rfc/rfc6698.txt.

[HS17] Hoffman, P.; Schlyter, J.: Using Secure DNS to Associate Certificates with Domain Names for S/MIME, RFC 8162, RFC Editor, Mai 2017.

[Ma09] Marlinspike, M.: Defeating OCSP With The Chara ter '3', Techn. Ber., 2009, URL: http://www.thoughtcrime.org/papers/ocsp-attack.pdf.

[Sa13] Santesson, S.; Myers, M.; Ankney, R.; Malpani, A.; Galperin, S.; Adams, C.: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP, RFC 6960, http://www.rfc-editor.org/rfc/rfc6960.txt, RFC Editor, Juni 2013, URL: http://www.rfc-editor.org/rfc/rfc6960.txt.