DANE und DNSSEC

X.509 Zertifikate im DNS statt PKI

Prof. Erwin Hoffmann

Provadis Hochschule, Frankfurt/Main

21.04.2015

24.3.2015: Google deckt Missbrauch im SSL-Zertifizierungssystem auf

Google ist auf mehrere gefälschte SSL-Zertifikate für Google-Domains gestoßen, die von der Zwischen-Zertifizierungsstelle MCS Holding herausgegeben wurden. Deren CA-Zertifikat wird von der Root-Certificate-Authority China Internet Network Information Center (CNNIC) beglaubigt.

CNNICs Root-Zertifikat findet sich in den meisten Betriebssystemen und Browsers, die damit den gefälschten Zertifikaten von MCS vertrauen. Einzig die Webbrowser Chrome und Firefox (ab Version 33) sowie ChromeOS weisen sie für Google-Domains zurück, da den Domains dort SSL-Zertifikate ausdrücklich zugeordnet werden (Public-Key Pinning)[1].

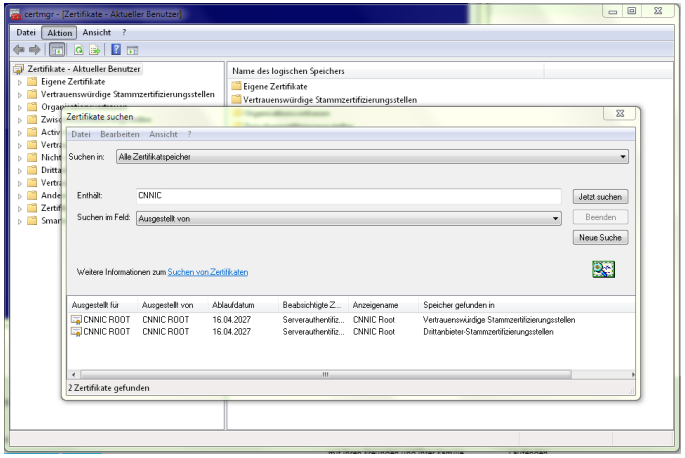


Abbildung : Viele Betriebssysteme (hier Windows 7 Professional) und Browser vertrauen von CNNIC beglaubigten Zwischenzertifizierungsstellen wie MCS, die gefälschte Domain-Zertifikate in Umlauf gebraucht haben.

[1] `http://www.heise.de/security/meldung/Google-deckt-erneut-Missbrauch-im-SSL-Zertifizierungssystem-auf-2583414.html`

Kurzer Blick in die Vergangenheit ....

Dieser Vorfall ist der letzte in einer langen Folge von Missbräuchen von X.509 (Stammzertifikaten):

- 1. September 2011: *CA hack: more bogus certificates* (DigiNotor)[2]
- 7. Februar 2012: *Trustwave issued a man-in-the-middle certificate*[3]
- 6. Juni 2012: *Super-Spion Flame trug Microsoft-Signatur*[4]

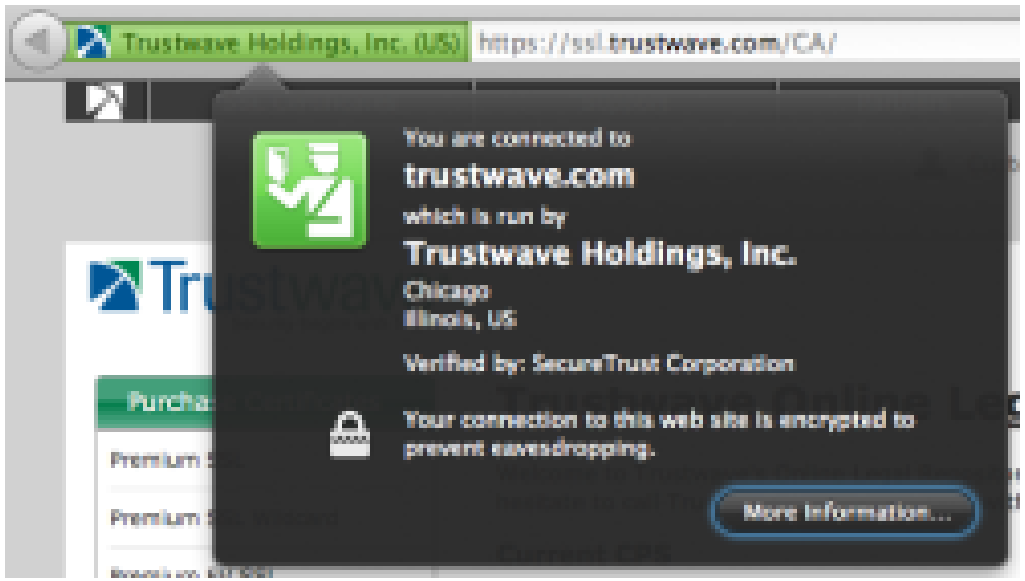


Abbildung : Alle Browser und Betriebssysteme beinhalten das X.509 root Zertifikat von *Trustwave*

[2] http://www.h-online.com/security/news/item/CA-hack-more-bogus-certificates-1334651.html

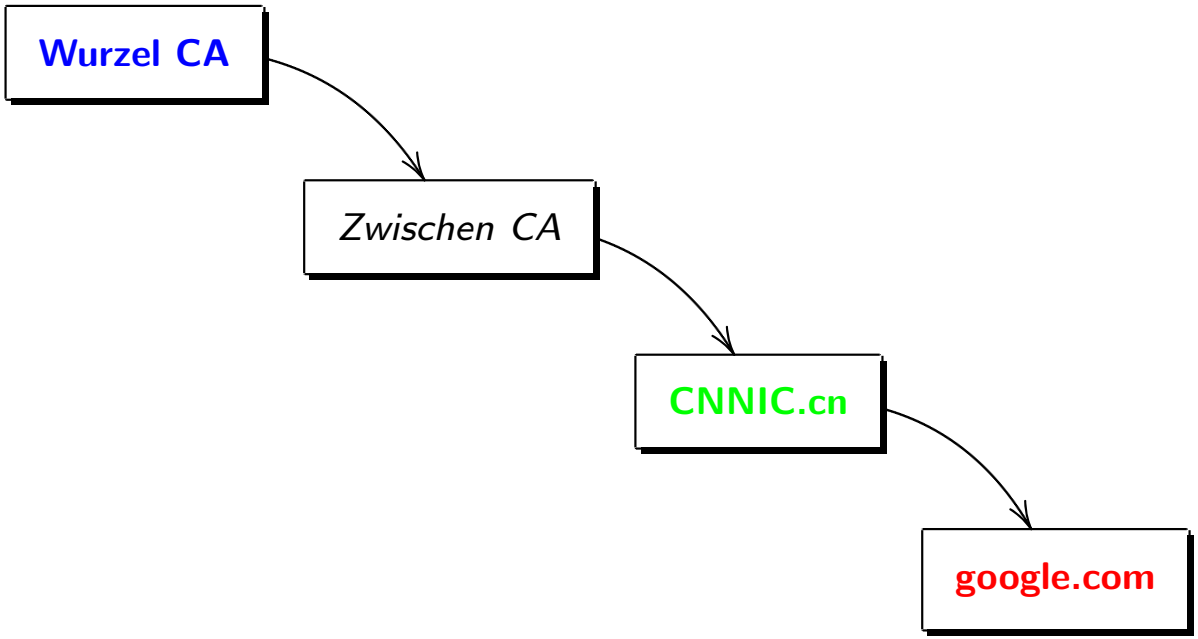
[3] http://www.h-online.com/security/news/item/Trustwave-issued-a-man-in-the-middle-certificate-1429982.html

[4] http://www.heise.de/security/meldung/Super-Spion-Flame-trug-Microsoft-Signatur-1590335.html

Wo liegt das Problem ?

Gebrochene Zertifikatskette

Jede *Certificte Authority* (**CA**) – *Trust Center* – darf X.509 Zertifikate bzw. auch Stammzertifikate für andere Organisationen ausstellen:



Gebrochene Zertifikatskette

↪ Die gesamte PKI-Vertrauenskette (*Trust Chain*) ist insbesondere durch die Unterminierung durch die NSA und andere Geheimdienste gebrochen.

Wir müssen den Stamm-Zertifikaten in den Bowsern und unseren Betriebssystemen vertrauen (oder auch nicht) ...

Stamm-Zertifikate im MacOS Schlüsselbund

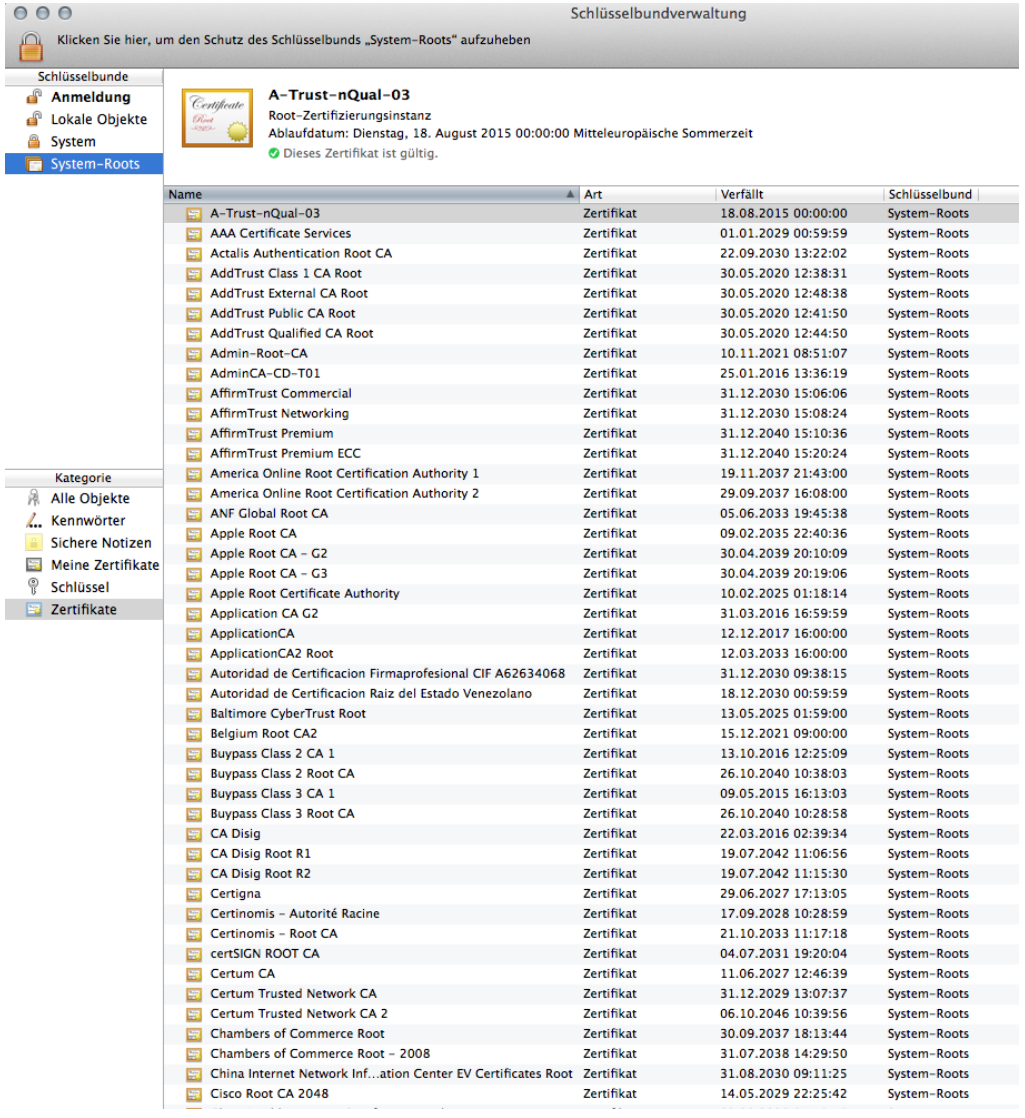


Abbildung : Auszug aus dem MacOS X Schlüsselbund

↪ Browser wie Chrome und FireFox bringen eigene Stamm-Zertifikate mit!

Welche Bedeutung besitzen X.509 Zertifikate ?

Einteilung der Zertifikate

Bei den X.509 Zertifikaten unterscheiden wir zwischen

- **Stammzertifikate** – zur *Beglaubigung* (Legitimation) anderer X.509 Zertifikate und
- **Standardzertifikate** – zur *Signatur* (None-Repudiation) von Personen, Rechnern, Domains, Programm-Code, Schlüsseln ...

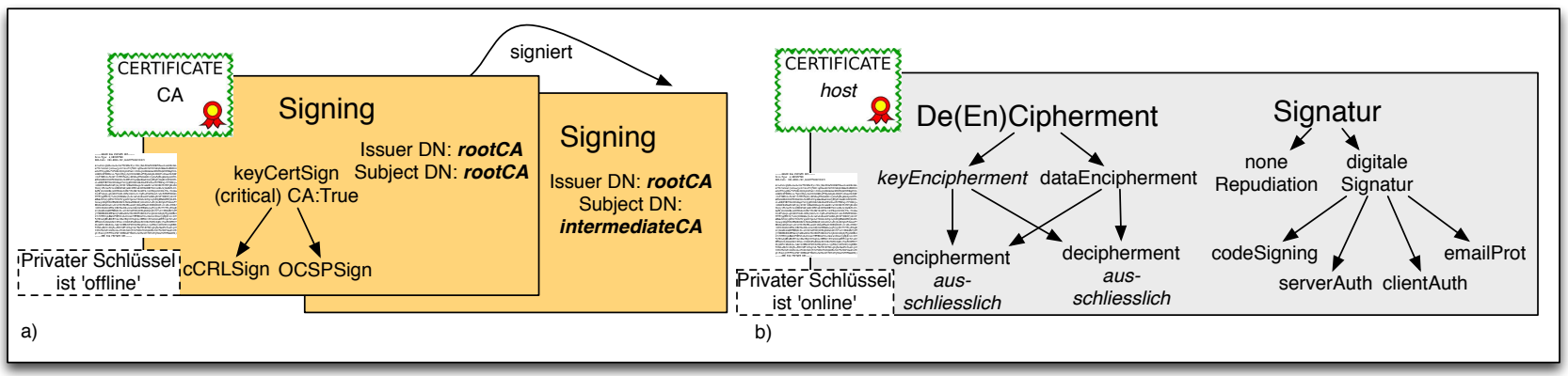


Abbildung : Taxonomie der X.509 Zertifikate

↪ Standardzertifikate werden beim RSA-Algorithmus (auch) zum (asymmetrischen) *Verschlüsseln* des **Key-Materials** eingesetzt!

Bei *Standardzertifikaten* muss der zugehörige *Private Schlüssel* als Teil des **private keys** online vorhanden sein; bei *Stammzertifikaten* wird er nur zum Zeitpunkt der Beglaubigung benötigt.

X.509

Der Standard *X.509* entstammt dem Repertoire von Standards die *CCITT*[5] (heute: *ITU-T*[6]) stammen. Ein zentraler Baustein ist der Verzeichnisdienst-Standard *X.500*, der heute in etwas abgewandelter Eigenschaft als *LDAP*[7] eingesetzt wird.

Der Standard *X.509* beschreibt den Aufbau (einschliesslich Syntax und Attribute) von *Zertifikaten*. Zertifikate sind komplexe Gebilde und beinhalten Informationen

- über den Besitzer, speziell dessen 'Name' als sog. *Distinguished Name* (**DN**),
- den Herausgeber des Zertifikates (ebenfalls mittels seines **DN**),
- den Public Key (einschliesslich Generierungs-Algorithmus) des Besitzers,
- den Verwendungszweck des Zertifikates sowie
- die Signatur des Herausgebers, der diese Elemente beglaubigt bzw. das Zertifikat damit legitimiert.

[5] Comité Consultatif International Téléphonique et Télégraphique

[6] ITU Telecommunication Standardization Sector

[7] Lightweight Directory Access Protocol

Aufbau von X.509v3 Zertifikaten

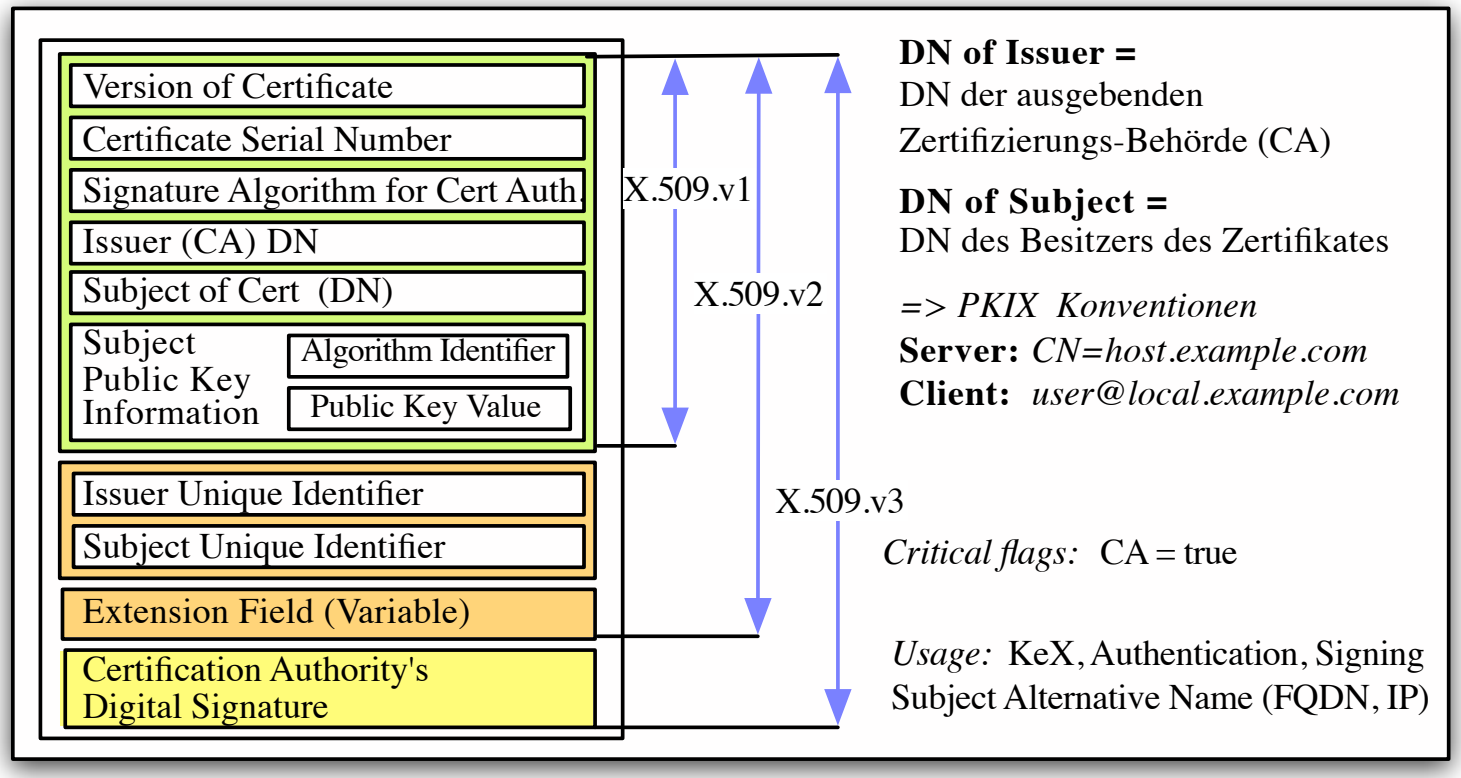


Abbildung : Allgemeiner Aufbau von X.509Zertifikaten

X.509v3

Die heute aktuelle Form des Zertifikats liegt in Form der Version *X.509v3* vor. Hierbei sind umfangreiche *Extensions* des Zertifikats möglich, die insbesondere dessen Verwendungszweck genauer bestimmen können:

- Erweiterungs-ID
- Kritikalitäts-Flag
- Inhalt der Erweiterung

Typ	**Bedeutung**
Basic Constraints	Angabe, ob der Inhaber eine *Certificate Authority* **CA** ist, also Zertifikate ausstellen und signieren darf: $CA = True$ sowie, wie viele untergeordnete CAs gestattet sind (*PathLen*).
Authority Key Identifier *Suject Key Identifier*	Informationen zum *Private Key* wie Hashwert zur Unterstützung von Zertifikatsketten.
Key Usage OID: 2.5.29.0F (hex)	Einsatz des Zertifikats (z.B. Stammzertifikat zum Signieren, Benutzer-Zertifikat zum Verschlüsseln von Dateien, Authentisierung von E-Mail etc.)
Extended Key Usage OID: 2.5.29.37 (hex)	Von den Firmen Netscape und Microsoft vorgelegte Erweiterungen, abhängig davon, ob es sich um ein CA-Zertifikat handelt oder nicht.
Issuer Alternative Name *Subject Alternative Name*	Hinterlegung von FQDN, E-Mail Adresse oder URL für den Herausgeber bzw. Besitzer des Zertifikates zur zusätzlichen Verifikation (E-Mail Adresse)
Certificate policies	Ergänzende Angaben z.B. zu Einschränkung von Zertifikatsketten.
CRLDistributionPoint	URL mit der Adresse, wie und woher eine Liste von CRLs bezogen werden kann.

Beispiel von X.509 Zertifikaten

Das Standardformat von Zertifikaten regelt der Standard X.509. Öffentlich zugängig wird der Aufbau von X.509v3 Zertifikaten aktuell in *RFC 5280* ('Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile') beschrieben.

openssl x509 -in orioncert.pem -noout -text

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 2 (0x2)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=DE, ST=RLP, L=Hoehn, O=fehnet, CN=fehnet.de/emailAddress=postmaster@fehnet.de, CN=fehnet.de
        Validity
            Not Before: Oct 11 09:56:00 2009 GMT
            Not After : Oct 11 09:54:31 2010 GMT
        Subject: C=DE, ST=RLP, L=Hoehn, O=fehnet, CN=orion.fehnet.de/emailAddress=postmaster@fehnet.de, CN=orion.fehnet.de
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:d3:f7:ca:d8:6b:56:a8:c5:36:00:a5:bc:52:44:
                    25:b3:06:6f:72:07:08:92:b8:aa:c4:7a:bd:9e:96:
                    ee:ea:3b:c4:24:04:3f:78:56:c6:87:fb:18:07:82:
                    e3:89:8f:17:20:0d:22:c4:a3:4e:18:3f:b3:d0:35:
                    5f:16:4e:f4:e2:63:cb:65:cb:10:1d:26:e8:e5:34:
                    56:1e:b8:13:2c:09:0c:4f:84:a9:a0:aa:ff:8a:98:
                    5e:a8:fb:ba:fc:87:40:52:bf:8f:b0:11:f5:6a:3d:
                    14:c4:4d:9f:79:96:f6:7f:81:7b:38:38:e4:56:f3:
                    0b:67:7f:4e:e9:e1:e8:40:b1
                Exponent: 65537 (0x10001)
    Signature Algorithm: sha256WithRSAEncryption
        92:ac:5e:7c:0b:36:f9:8b:c3:3c:94:d3:ca:76:a7:05:65:67:
        a6:83:2c:cf:fa:7d:a8:42:56:8b:c5:61:5c:f8:13:58:6d:c7:
        50:b2:96:00:44:0f:86:ea:97:8a:c7:80:cc:18:0d:9b:67:82:
        89:1d:f4:71:e5:0e:b6:1e:76:6e:c4:46:3d:ae:29:5c:65:99:
        db:54:c2:1f:f9:83:68:2b:71:5d:ac:45:70:30:2a:16:e4:4c:
        d2:14:6e:31:ab:d2:f6:68:c5:9f:0c:9e:e7:62:85:c3:e4:ab:
        e7:20:75:4f:9c:34:59:30:ba:a8:a6:7e:b9:e0:e8:75:ae:f4:
        e2:1b
```

Public Key Infrastruktur für was ?

Trust Center = Anker der Public Key Infrastructure (PKI)

Ein Trustcenter stellt als vertrauenswürdige Instanz den zentralen Teil einer PKI dar. Daher hat der Gesetzgeber strenge Auflagen an den Aufbau und Betrieb eines *Trust Centers* gemacht, das als *Certificate Authority* **CA** Zertifikate ausstellen bzw. signieren darf.

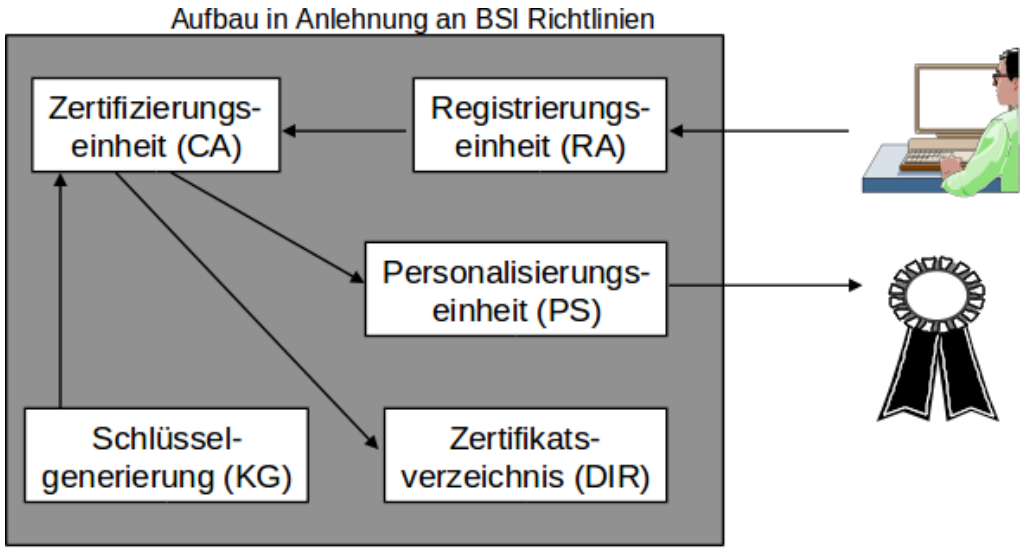


Abbildung : Aufbau eines Trust Centers

Ein Trust-Center fungiert somit als **CA** und besitzt folgende Merkmale:

- Eine *RootCA* ist immer 'offline', d.h. vom Internet getrennt.
- Ein Trust Center besitzt eine besonders geschützte Infrastruktur (*Hardware Security Module* **HSM**).
- Bereitstellung einer *Registration Authority* **RA**.
- Authentisierung von SubCAs.
- Zertifikationsausgabe an SubCAs.
- Rückruf von Zertifikaten, die hiervon ausgestellt wurden.
- Unterhaltung von *Certificate Revocation Lists* **CRL**.

Certificate Authorities

Authentisiert sich ein Kommunikationspartner durch sein (gültiges) X.509 Zertifikat (und den *Private Key*) und besitzt der jeweils andere Kommunikationspartner das Stammzertifikat, das Ausgang der Zertifikatskette war, so kann dieser das Zertifikat qualifiziert *verifizieren*.

Vom technischen Standpunkt spielt es keine Rolle, ob die Zertifikatskette bei einem 'öffentlichen' Zertifikatsanbieter (z.B. *GlobalSign*) beginnt, oder beim eigenen Betrieb. Die folgenden Schritte sind aber in jedem Fall vorzunehmen:

- Zunächst ist ein Stammzertifikat für 'Root', d.h lokale die *Certificat Authority* **CA** zu erstellen. Dies ist notwendigerweise 'selbst-signiert':
  $\hookrightarrow$ Root Certificate: `Subject = Issuer`.
- Es muss die *Basic Contraints CA = True* sowie eine qualifizierte *PathLen* aufweisen.
- Üblicherweise werden nun (langlebige) untergeordnete *CA*s zur Erstellung unterschiedlicher Ziel-Zertifikate eingerichtet; z.B. für Server- und/oder Personen-Zertifikate bzw. *CRL*s.
  Diese *CA*s sind durch 'Root' entsprechend legitimiert und weisen ebenfalls die (kritischen) *Basic Constraints CA = True* auf, sowie die *Key Usage CertSign* bzw. *CrlSign* auf.
- Ausgehend von diesen *CA*s werden die Endbenutzer-Zertifikate (mit relativ kurzer Lebensdauer) abgeleitet.
  1. Rechner-Zertifikate besitzen die Eigenschaften *Digital Signature* und *Key Agreement*.
  2. Personen-Zertifikate weisen die Merkmale *Digital Signature* sowie *Encryption* auf.

Kommerzialisierung der Zertifikate

Damit ein X.509 Zertifikat als 'gültig' eingestuft wird, muss in der Applikation, die dieses bei der Kontaktaufnahme überprüft, ein zugehöriges Stammzertifikat verfügbar sein:

$\Longrightarrow$ Betriebssysteme wie *Windows*, *Ubuntu*, aber auch Applikationen wie die Web-Browser *FireFox* und *Opera* bringen eine Satz von Stammzertifikaten mit.

Von diesen Stammzertifikaten abgeleitete X.509 Zertifikate sind besonders 'wertvoll'. Dies lassen sich entsprechende Anbieter von den Kunden bezahlen:

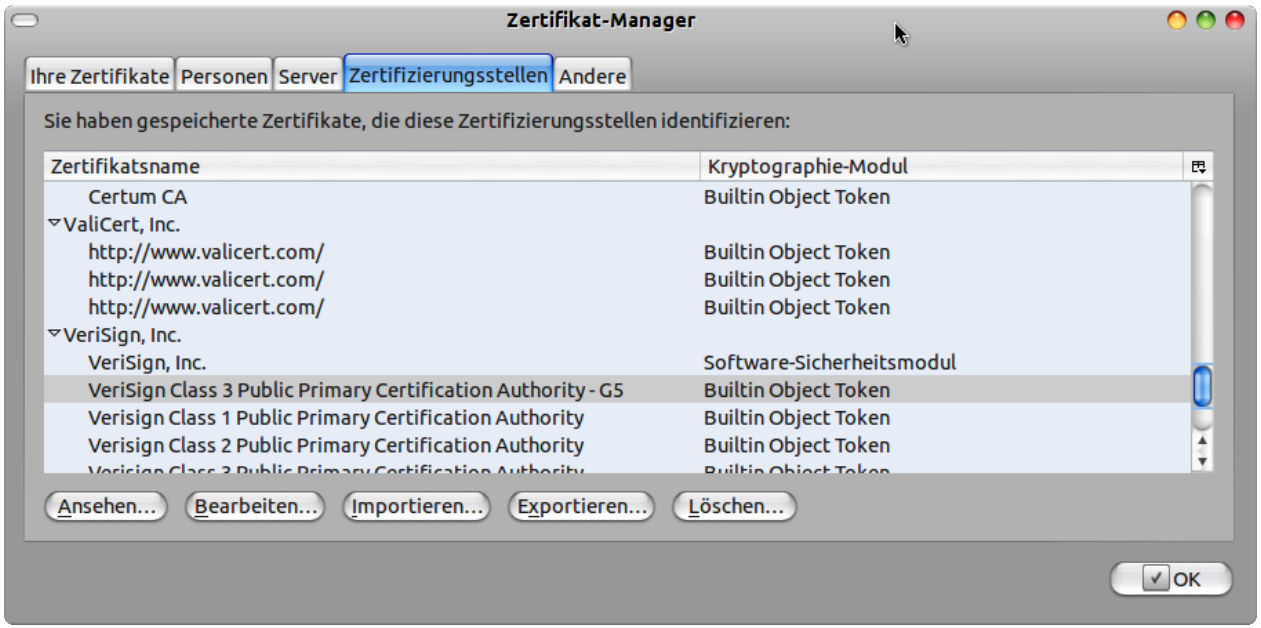


Abbildung : Mitgelieferte X.509 Stamm-Zertifikate des Firefox-Zertifikat-Managers

Klassifizierung von Zertifikaten

Arten von Zertifikaten[8]:

- *Domain-Zertifizierung*:
  Bei der Domainzertifizierung wird durch einen E-Mail-Robot eine E-Mail an eine Adresse aus dem WHOIS oder eine alternative administrative Adresse gesandt, um die Bestellung zu bestätigen. Hiermit wird sichergestellt und abschließend zertifiziert, dass die Domain sowie ein administrativer Kontakt dieser existiert. Im Zertifikat wird ausschließlich die Domain genannt, auch das SiteSeal[9] des Limitbreaker-Zertifikates, welches abweichend von der üblichen SiteSeal-Technik ebenfalls anklickbar ist, zeigt nur die Domain an, die zertifiziert wurde.
- *Identitäts-Zertifizierung*:
  Bei der Identitätszertifizierung wird ein entsprechendes Dokument zur Authentifizierung des Zertifikatsinhabers angefordert, geprüft und mit den Angaben im WHOIS abgeglichen. Beim *Platinum*-Zertifikat erfolgt außerdem ein Anruf beim Zertifikatsinhaber, um die Bestellung noch einmal abschließend zu bestätigen. Im daraufhin ausgestellten Zertifikat werden die kompletten Angaben des Zertifikatsinhabers angezeigt. Ebenso werden die vollständigen Daten in der Echtzeitüberprüfung über das jeweilige *TrustLogo* – soweit vorhanden – angezeigt.

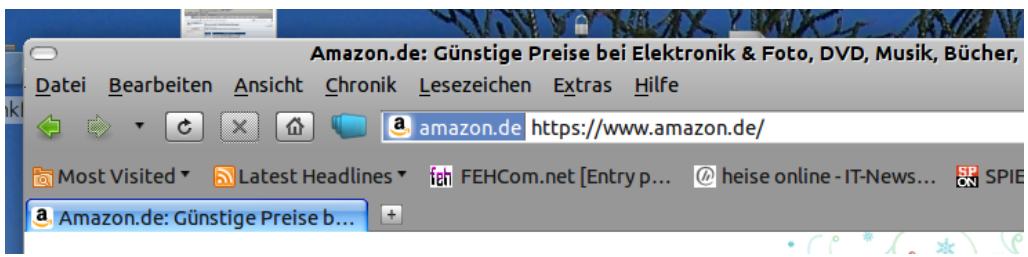


Abbildung : Einblendung des Domain Trust-Logos bei FireFox

[8] Quelle: `http://kb.psw.net/`

[9] Logo, das den Status 'Verschlüsselung' anzeigt

Zertifikats-Klassen

Zertifikate werden entsprechend ihrem Ausstellungsprozess in *Klassen* eingeteilt[10]:

Class 0 *Testzertifikate* ohne Validierung; entspricht selbst-ausgestellten Zertifikaten.

Class 1 *Domainzertifikate*, die durch Telefon- und/oder E-Mail-Robots validiert wurden.

Class 2 *Eingeschränkte Identitätszertifikate*, basierend auf einer schriftlichen Bestätigung, das die gemachten Angaben korrekt sind. Werden als SSL- bzw. Server-Zertifikate in der Regel nicht angeboten, da sie hierfür einen zu geringen Prüfungsumfang bieten, kommen aber regelmäßig zum Einsatz als S/MIME- bzw. Client-Mitarbeiter-Zertifikate. Nachdem ein Unternehmen und häufig gesondert auch deren **CA** *Class 3* validiert wurden, stellt dieser selbst Zertifikate für Mitarbeiter im Unternehmen aus, deren Identität persönlich bekannt ist bzw. schriftlich bestätigt wurde. Solche nach einer *Class 3* Validierung intern selbst ausgestellten Zertifikate gelten als *Class 2* Zertifikate

Class 3 *Identitätszertifikate*, die durch Überprüfung der Daten des Handelsregisterauszugs bei im Handelsregister eingetragenen Unternehmen, des Gewerbenachweises bei nicht im Handelsregister eingetragenen Unternehmen oder des Personalausweises bei reinen Privatpersonen validiert und deren Angaben mit den Daten in der Internet `WHOIS` Datenbank abgeglichen wurden ($\Rightarrow$ *Identitätszertifizierung*).

Class 4 *Erweiterte Identitätszertifikate*, die durch eine direkte *Face-to-Face-Identifizierung* anhand der Originaldokumente sowie der Daten im `WHOIS` analog zu *Class 3* validiert wurden (wird nirgends angeboten, weil zu kostspielig und zu aufwändig für beide Seiten)

[10] Quelle: `http://kb.psw.net/questions/31/`

Praktikabilität von PKI

Meine Ausführung über die *Public Key Infrastructure* waren bei weitem nicht erschöpfend. Zentrale Elemente der **PKI** sind

- Das *Deployment* der notwendigen Stammzertifikate.
- Die *Implementierung* des Protokolls sowohl bei den Servern als auch den Clients.
- Die Bereitstellung einer *unkompromittierten Infrastruktur* für die PKI-Information.
- Ein gemeinsames *Verständnis der Attribut-Nutzung*.
- Sowie aktuell der Möglichkeit, dass der Anwender über die Applikation über den Status des Zertifikats informiert wird.

Die Komplexität des PKI-Standards und die vorgeblichen Automatismen bringen für den Anwender ein beträchtliches Mass an In-Transparenz mit sich. Problematisch hierbei ist im Besonderen die Rolle der lokal vorhandenen Stammzertifikate. OpenSSL bietet z.B. die Möglichkeit, Zertifikate als 'vertrauenswürdig' (set trust) zu deklarieren. Ähnliche Möglichkeiten bieten auch Tools wie z.B. *XCA* und *TinyCA*; aber auch die Äquivalente unter Microsoft Windows.

Welche Anwendungen nutzen X.509 Zertifikate/PKI ?

Anwendungen, die PKI und X.509 Zertifikate nutzen

Unsere gesamte digitale Welt fundiert auf dem Einsatz von X.509 Zertifikaten:

1. X.509 Zertifikate werden zusammen mit dem Private Key benutzt eine *Authentisierung* des Kommunikationspartners zu ermöglichen und somit *Man-in-the-Middle* (**MitM**) Angriffe auszuschliessen.
2. Der Public Key in den X.509 Zertifikaten wird zur *Verschlüsselung* des kryptographischen Key Materials beim RSA Algorithmus herangezogen.
3. Das X.509 Zertifikat dient zusammen mit der Key Chain zur *Legitimation* des Kommunikationspartners.

Die folgenden Applikationen nutzen X.509 Zertifikate:

- *Transport Layer Security* (**TLS**) – bei HTTPS, SMTPS, StartTLS, STLS ....
- *Internet Key Exchange* (**IKE**) – bei IPsec (IPv4/IPv6)
- *Extensible Authentication Protocol* (**EAP**) – bei WPA2 und 'Enterprise WLAN'

↪ HTTPS in den Browsern ist sehr empfindlich gegenüber (3.); während sich SMTPS sich um (3.) überhaupt nicht kümmert. Cloud Services verlangen insbesondere (1.) sowohl beim Server als auch für den Client!

Bei der *Perfect Forward Secrecy* **PFS** werden X.509 Zertifikate nicht für den Fall (2.) eingesetzt.

Asymmetrische Verschlüsselung

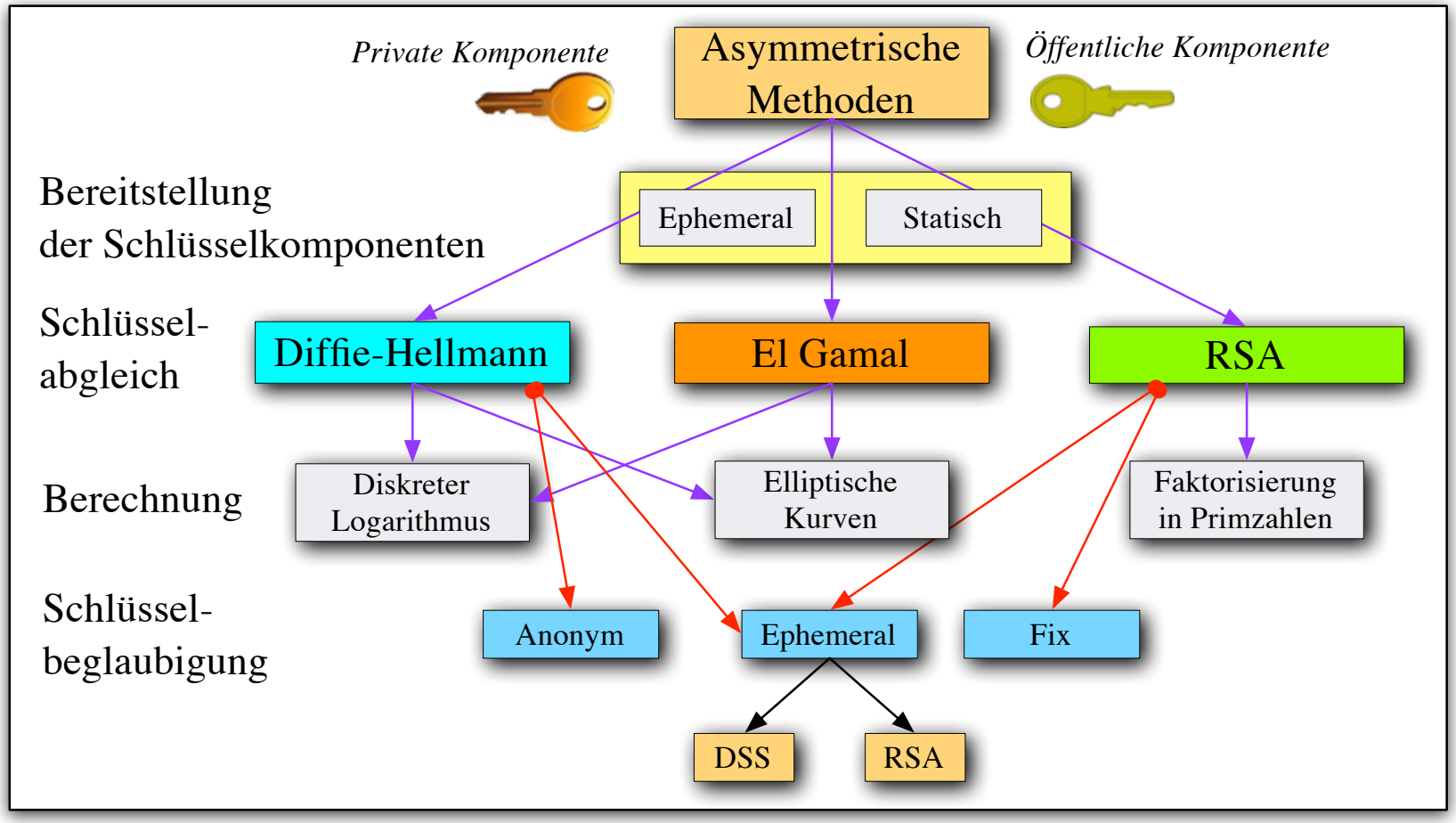


Abbildung : Taxonomie der asymmetrischen Verschlüsselung

DNS Authorized Name Entries

Alternativen zur PKI

X.509 Zertifikate sind für die bestehende Internet-Sicherheitsinfrastruktur unabdingbar. Die grösste Gefahr geht von kompromittierten Zertifikaten bzw. Stammzertifikaten aus: Kompromittierte X.509 Zertifikate sind technisch und sachlich korrekt; ein Browser z.B. kann diese Zertifikate erfolgreich *validieren* als auch *verifizieren*.

Innerhalb der PKI gibt es zwei Ansätze, die Anwender vor kompromittierten Zertifikate zu schützen:

1. *Certificate Revocation Lists* (**CRL**):
   Dem Browser (oder dem OS) wird eine Liste (Fingerprints) von fehlerhaften Zertifikaten beigefügt (z.B. über einen Update-Mechanismus).
2. *Online Certificate Status Protocol* (**OCSP**):
   Im Zertifikat findet sich eine URL, über die der Client die Gültigkeit des Protokolls (zusätzlich) verifizieren kann.

Google setzt im Chrome-Browser auf eine interne Positiv-Liste:

3. Die *Fingerprints* der Google-Zertifikate werden mitgegeben:
   Weist sich ein vermeintlicher Googles-Server mit einem Zertifikat aus, dass hierin nicht beinhaltet ist, wird eine Warnung ausgegeben.

DNS statt Public Key Infrastruktur

Die *Public Key Infrastructure* **PKI** stellt eine Meta-Infrastruktur des Internet dar, die von von zertifizierten und akkreditieren Firmen (privat) organisiert und getragen wird.

↪ Statt dieser externen Infrastruktur macht es Sinn, das bereits bestehende *Domain Name System* (**DNS**) zum Deployment der X.509 Zertifikate (und natürlich nicht der Key Files) heranzuzuiehen. Dies wurde in entsprechender Konsequenz erstmals in RFC 6698 vorgenommen. Idee:

- Der Domain-Inhaber veröffentlicht das X.509 Zertifikat für die Rechner, über die sich z.B. per HTTPS verbunden werden soll in seiner Zonendatei, für die er verantwortlich ist.
- Um festzustellen, ob ein X.509 Zertifikat vorliegt, muss der Client eine spezielle DNS-Information – einen TLSA-Record – anfordern.
- Liegt dieser vor, ist klar, dass der Server TLS-Verschlüsselung unterstützt.

Was hier so einfach klingt, ist aber mit Standard-DNS-Mitteln nicht zu erreichen und birgt zudem die Gefahr von Poisoning- und Man-in-the-Middle-Attacken per DNS.

SSH Fingerprints im DNS

Die Idee, zusätzliche Informationen für einen Rechner/Server im DNS unterzubringen, ist schon sehr alt:

- DNS `TXT` Records dienen zur Angabe beliebiger Informationen für einen Rechner.
- DNS `MX` Records erlauben die Mitteilung eines Mail eXchangers für eine Domain.
- DNS `SSHFP` Records können genutzt werden, SSH Fingerprints für eine Rechner zu hinterlegen.

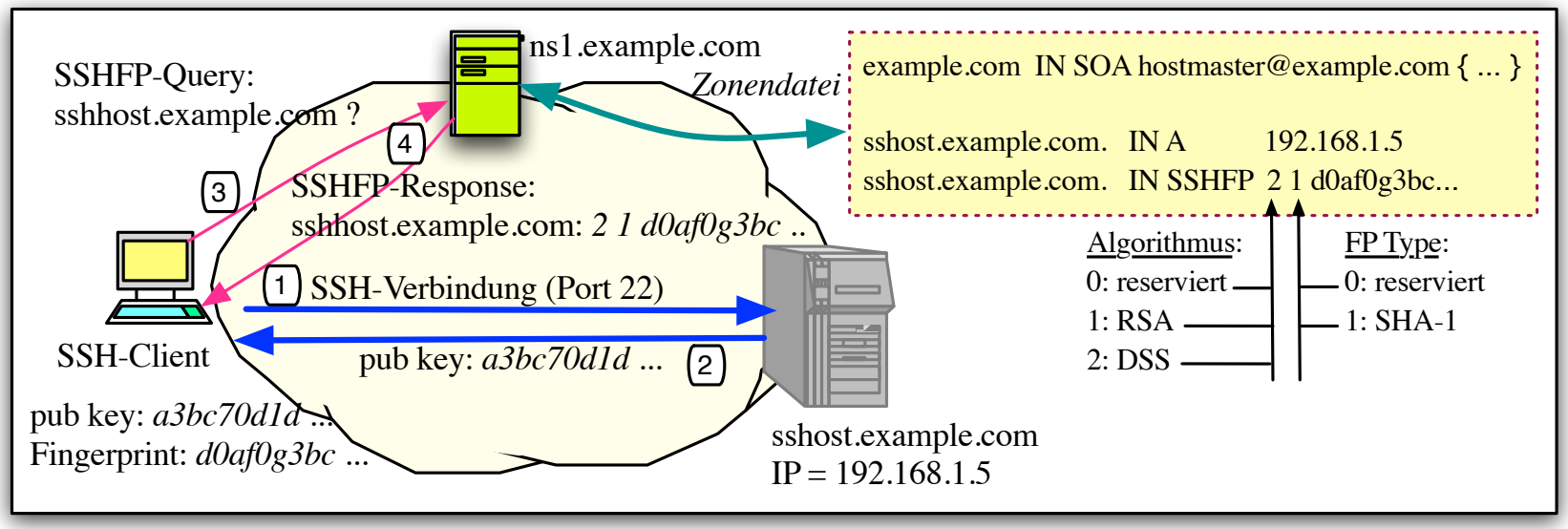


Abbildung : Ablauf der DNS-Fingerprint-Query

↪ Frage: Wie gross ist die DNS-Antwort (*Response*) vom Name-Server `ns1.example.com` für die Abfrage von `sshost.example.com` ?

DNS-Based Authentication Named Entities: DANE

Mittels der *DNS-Based Authentication Named Entities* (DANE) wird hingegen der Versuch gemacht das PKI-Vertrauensmodell komplett durch DNS zu ersetzen.

DANE nutzt hierbei 'auf der linken Seite' der DNS-Records aber nicht den Hostnamen, sondern eine an Multicast-DNS angelehnte Schreibweise des Alias (C-Name) eines Rechners, indem zusätzlich der per TLS-geschützte Dienst beschrieben wird.

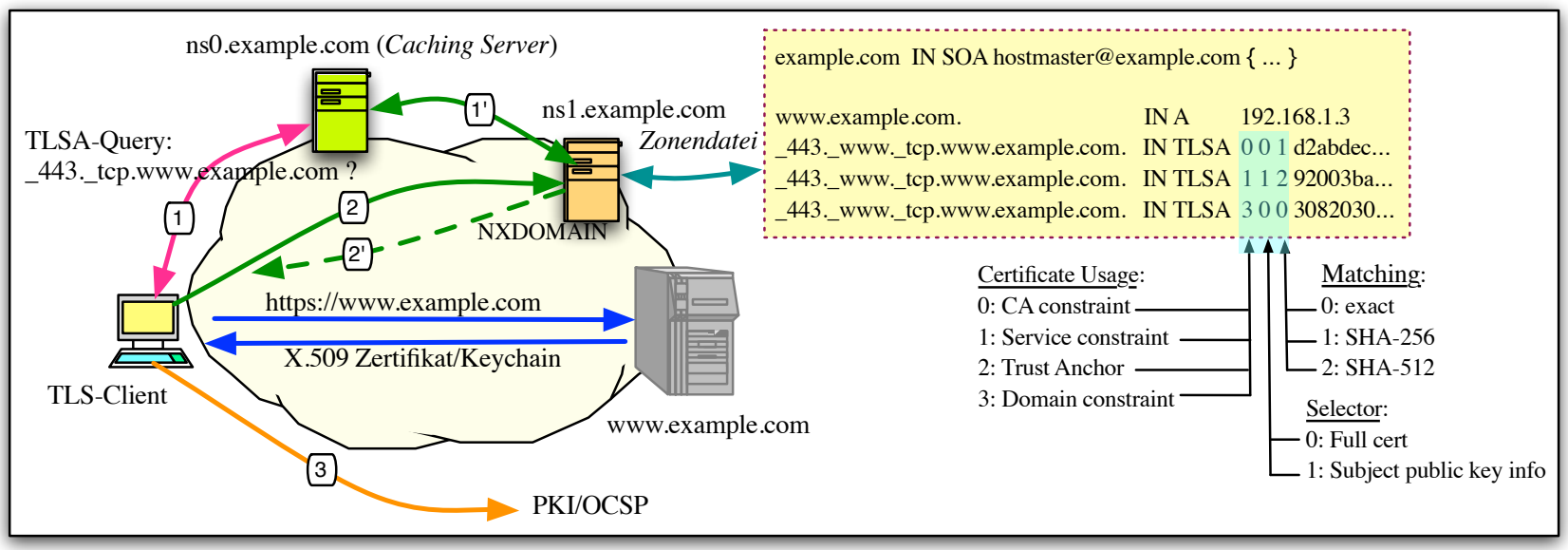


Abbildung : Abfrage eines Public Key per DANE

DNS TLSA-Records

Wie auch beim DNS SSH Fingerprinting, wird hier ein spezieller DNS Recordtyp benötigt: **TLSA**.

- Der Hostname nutzt DANE einen sog. *prefixed Name*, der nicht nur einen Rechner, sondern zusätzlich seinen Dienst, ein *Transportprotokoll* und einen *Port* ausweist: '`_443._tcp.www.example.com`'
- Der erste Teil der RDATA Section beschriebt die Nutzung des nachfolgenden Information:
  1. Certifcate Usage:
     - (1) CA constraint,
     - (2) Service constraint,
     - (3) Trust Anchor und
     - (4) Domain constraint
  2. Selector:
     - (0) Full certificate,
     - (1) Subject Public Key Info
  3. Matching:
     - (0) exact,
     - (1) SHA-256,
     - (2) SHA-512
- Es folgt das hexadezimal encodierte X.509 Zertifikat bzw. die Angabe über den Subject-DN.

↪ Frage: Wie gross ist die DNS-Antwort (*Response*) vom Name-Server `ns1.example.com` für die Abfrage von `www.example.com` ?

X.509 Zertifikate unter Einsatz von EDNS(0)

X.509 Zertifikate in DNS-Records können nicht per Standard-UDP-Nachricht übertragen werden, da diese (für IPv4) nur einen maximale Grösse von 512 Byte für den Payload ermöglicht.

↪ Die Nutzung von TLSA-Records verlangen entweder den Einsatz von TCP (für die Antwort) oder die Verwendung der DNS-Erweiterung **EDNS(0)** (RFC 6891).

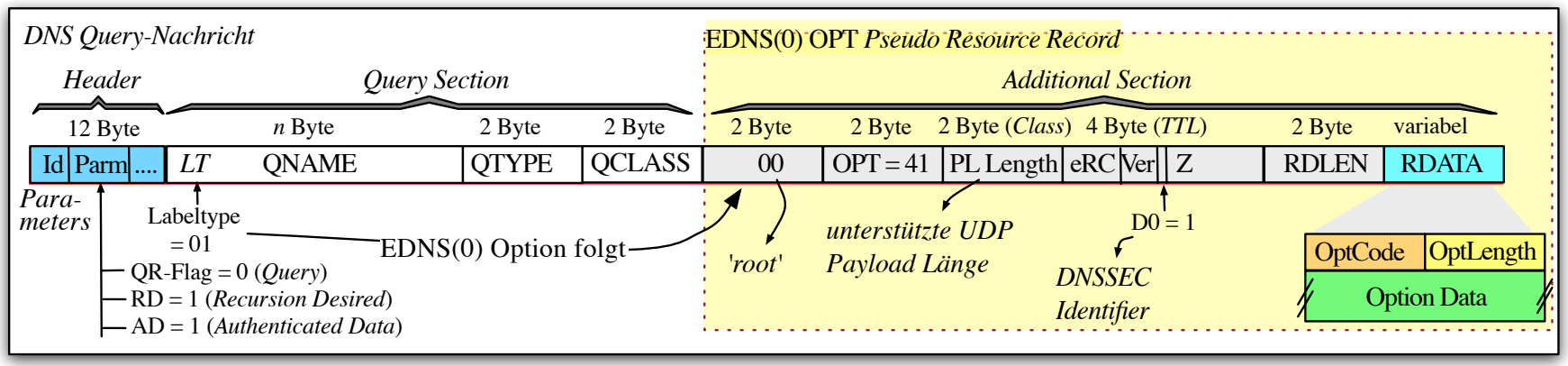


Abbildung : Aufbau einer DNS EDNS(0)-Query

↪ Die Unterstützung von EDNS(0) muss über die gesamte DNS-Kette garantiert werden. Der Client teilt die Nutzung von EDNS(0) in der ursprünglichen Query mit.

DANE und ungesichertes DNS

Für die Nutzung von DANE gelten also die folgenden Regeln:

- Der (TLS-)Client muss die Abfrage von X.509 Zertifikaten per DNS unterstützen.
- Als Fallback-Lösung – d.h. falls die Abfrage nicht möglich ist oder keine Information liefert – muss er über die Standard-PKI Mechanismen zusätzlich verfügen.
- Der Domain-Administrator muss die TLSA Records für die relevanten Server im DNS einstellen. Hierbei ist auf die TTL der DNS Records und die Gültigkeitsdauer des Zertifikats zu achten.
- Alle Komponenten müssen EDNS(0)-fähig sein (DNS-Caches, -Forwarder und Stub-Resolver).

$\hookrightarrow$ Die 'Integrität' der DNS-Information ist zu garantieren. Dies verlangt im heutigen Verständnis den Einsatz von DNSSEC; alternativ könnte aber auch DNSCurve von Dan Bernstein genutzt werden.

DNSSEC als Redefinition von DNS

Ziele von DNSSEC

DNSSEC hat das Ziel, die öffentlich im Internet deployten Informationen

- nicht im Hinblick auf die *Vertraulichkeit* der Nachricht-Übertragung zu garantieren,
- sondern 'lediglich' die *Integrität* der bezogenen DNS-Informationen sowie
- deren '*Autoritivität*' sicherzustellen.

Hierdurch ist es möglich, auch 'sensible' Informationen im DNS bereit zu stellen:

RFC 2065:

> *The extensions (DNSSEC) also provide for the storage of authenticated public keys in the DNS. This storage of keys can support general public key distribution service as well as DNS security.*

↪ Hierzu bedient man sich der Möglichkeit, die DNS-Records zu *signieren*, die für die Signatur notwendigen Public Keys zu veröffentlichen sowie die Verkettung der DNSSEC-Infrastruktur bis zur Root-Zone sicherzustellen.

Infrastruktur von DNSSEC

Um DNSSEC nutzen zu können, sind folgende Bedingungen zu schaffen:

- Ein DNSSEC-fähiger DNS Content-Server, dessen Zonendatei regelmässig und bei Änderungen des Inhalts signiert werden muss,
- EDNS(0)-fähige DNS-Cache-Server und -Forwarder
- DNSSEC-fähige (validierende) *recursive* DNS-Resolver,
- ggf. auch DNSSEC-fähige Stub-Resolver.

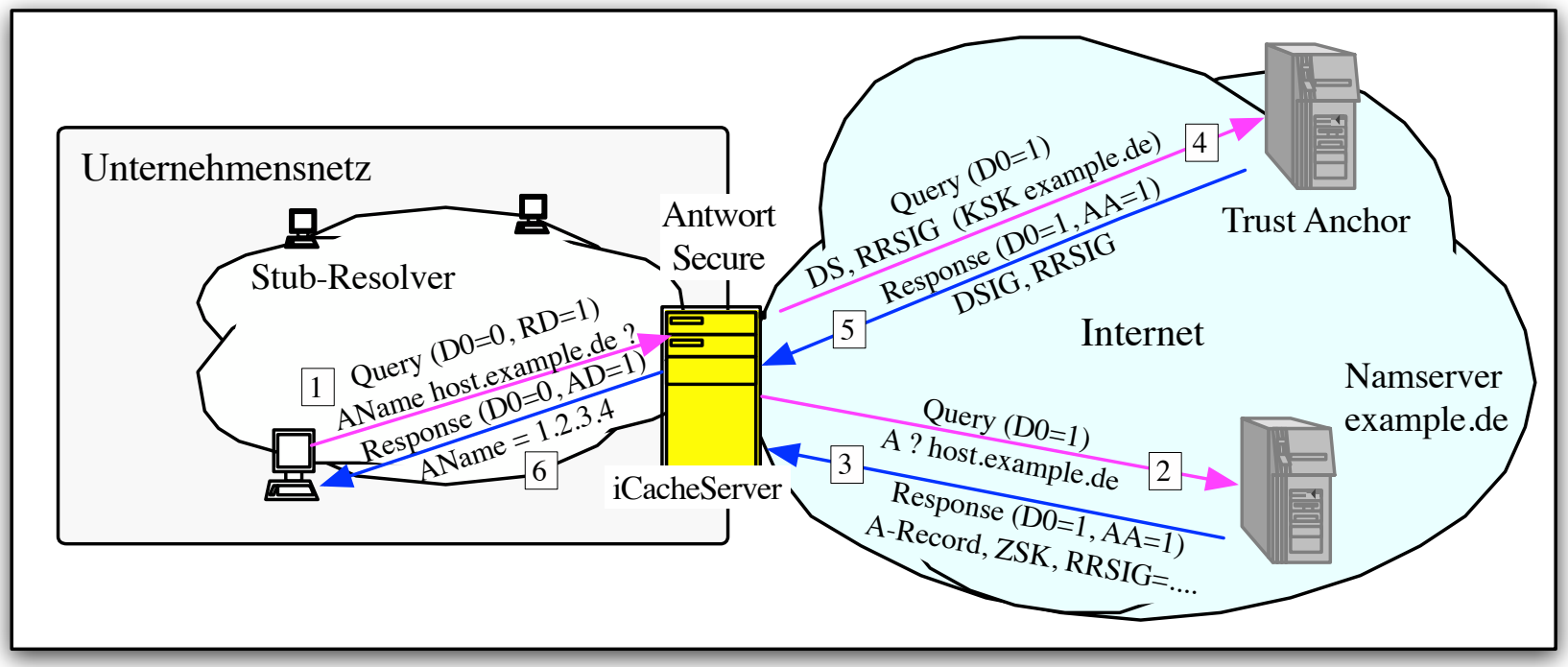


Abbildung : DNS Queries und DNSSEC Responses an nicht DNSSEC-fähige Stub-Resolver

Signieren der DNS-Zonen Daten

DNSSEC nutzt die kryptographische Signierung von DNS Records und stellt die berechneten Signaturen bereit:

- Ein DNSSEC Record `RRSIG` liefert die Signaturen für eine Anzahl kanonisierter DNS Records, wie sie mittels des *Zone Signing Keys* (**ZSK**) erzeugt wurden.
- Ein DNSSEC `DNSKEY` Record stellt für die Zone den *Public Key* zur Überprüfung der Signaturen bereit.
- Ein *Delegation Signer* (`DS`) beinhaltet die Signatur des *Key Signing Keys* (**KSK**), der vom übergeordneten Nameserver bezogen werden muss. Dieser DS ist zur Sicherstellung der *Trust Chain* auf diesem Nameserver ebenso durch einen `RRSIG` Records signiert.
- `NSEC3` Records teilen mit, welche DNS-Records überhaupt signiert werden, bwz. welche nicht.

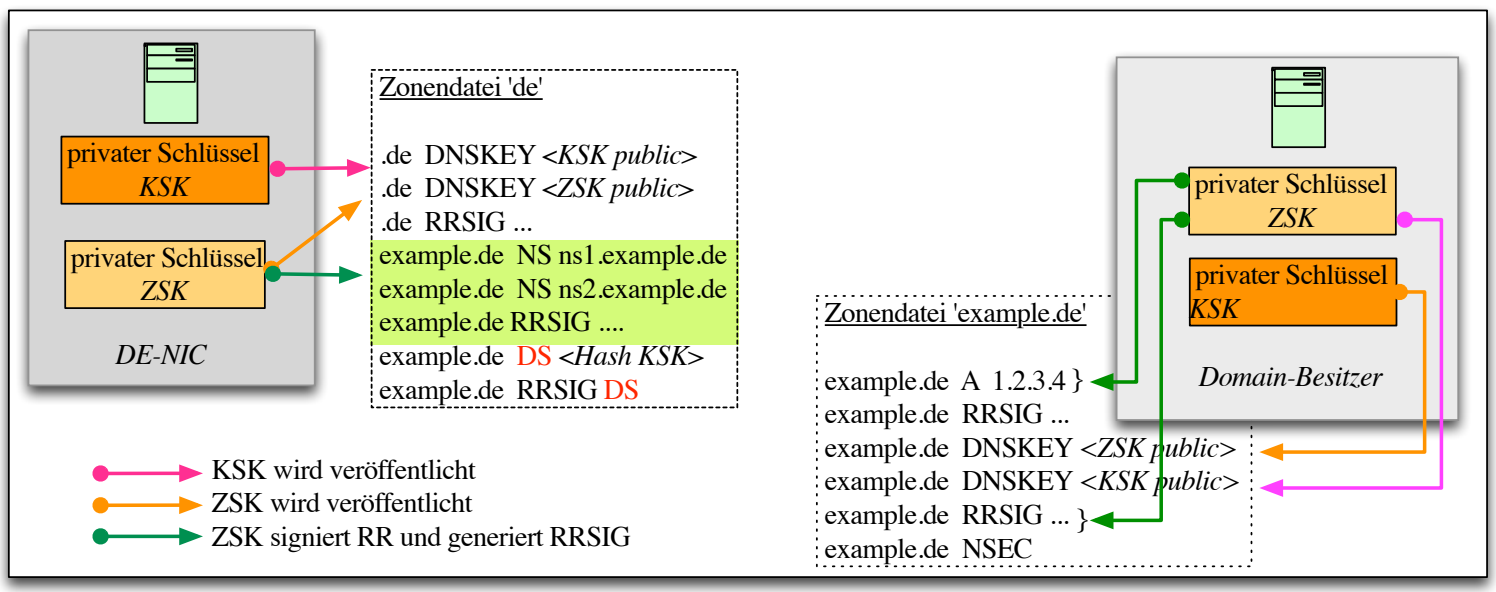


Abbildung : Trust Chain bei DNSSEC

Wiederholende Signierungen der Zonen Datei

- Aus rechentechnischen Gründen sollte der *Zone Signing Key* **ZSK** eine relative kurze Länge (512 Byte) und eine kurze Lebensdauer haben, also regelmässig getauscht werden.
- Der *Key Signing Key* (**KSK**), der vom übergeordneten Nameserver beglaubigt werden muss, ist demgegenüber länger und wird zur Erstellung immer neuer **ZSK** benutzt (und nur dafür).
- Die DNS Resource Records in der Zone müssen daher sowohl bei einer inhaltlichen Änderungen als auch periodisch
  - neu *kanonisiert* werden; also in eine *alfabetische* und *RR-Typ spezifische* Reihenfolge gebracht,
  - als auch als technischen Record-Set signiert werden.
  - Zudem existieren in der Zone zu jedem Zeitpunkt zumindest immer zwei gültige Signaturen eines Record-Sets.

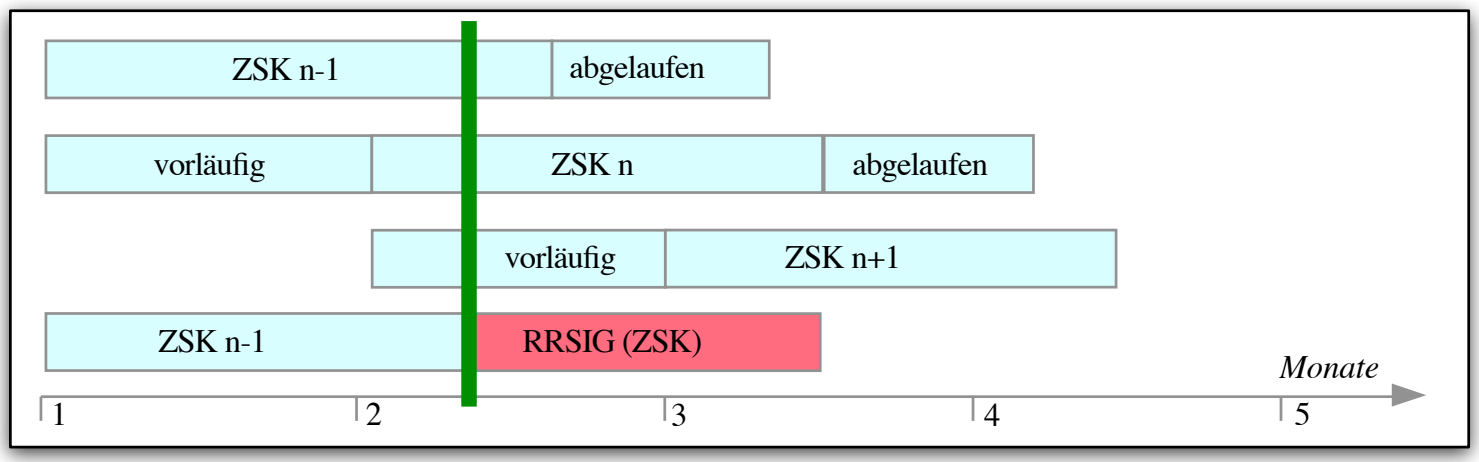


Abbildung : Key Roll-Over bei DNSSEC

Verifikation der DNS Records beim DNS Resolver

Die DNSSEC-fähigen Resolver müssen nun nicht nur

- die Signaturen der DNS Records überprüfen, sondern
- auch die Trust Chain überprüfen.

↪ Ist der Resolver ein DNS Cache-Server (iResolver) bleiben die erzielten Ergebnisse im Cache und müssen nicht bei jeder Abfrage neu bezogen werden.

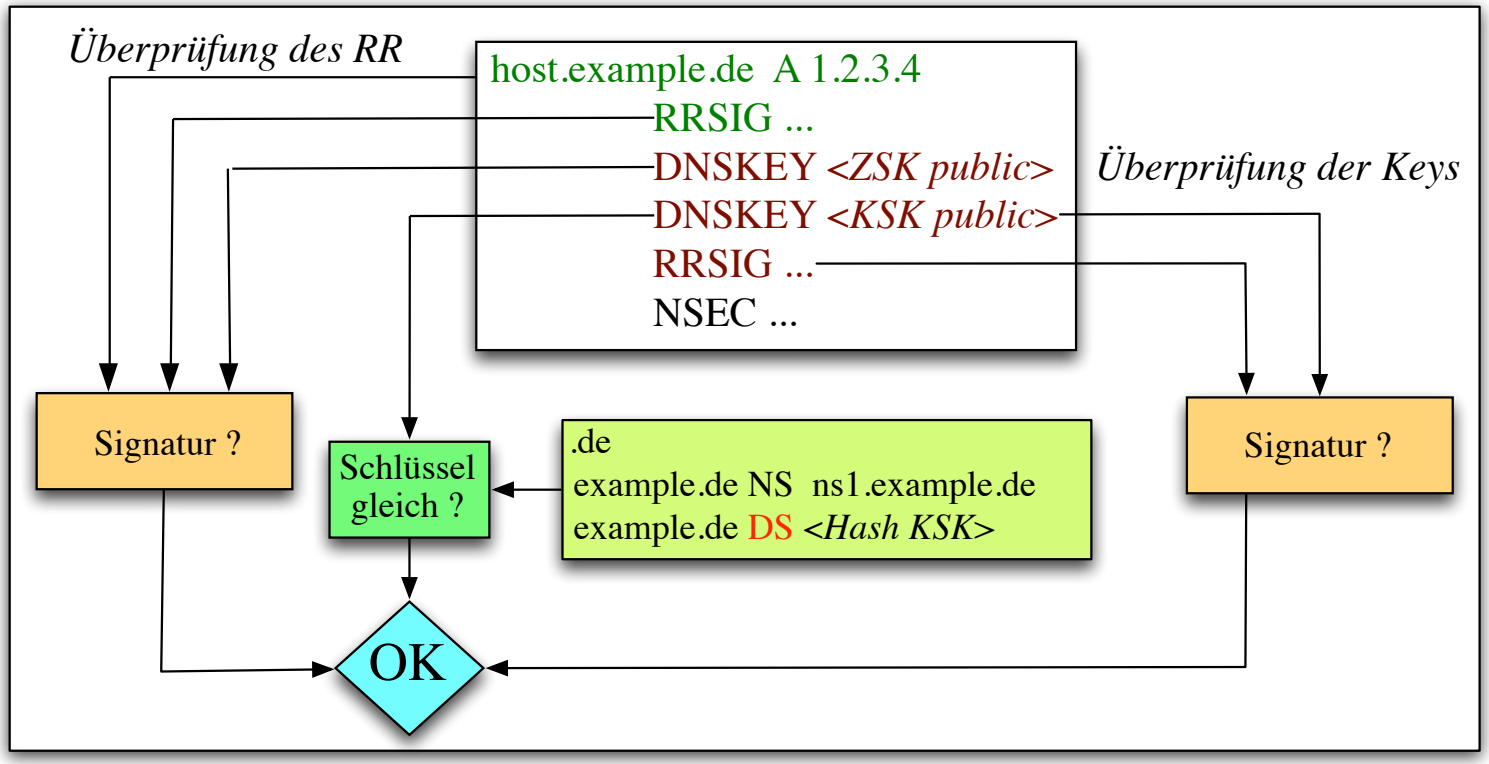


Abbildung : Überprüfung der Signaturen bei DNSSEC

Fragen / Antworten

Fragen/Antworten/Bemerkungen ?

Weiterführend Links und Quellen


A. Badach, E. Hoffmann *Technik der IP-Netze. 3. Auflage* Hanser Verlag, München, 2015


Housley, R., Ford, W., Polk, W., and D. Solo *Internet X.509 Public Key Infrastructure Certificate and CRL Profile* RFC 2459, January 1999.


Meyers, M., Ankney, R., Malpani, A., Galperin, S., and C. Adams *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP* RFC 2560, June 1999.


Bassham, L., Polk, W., and R. Housley *Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile* RFC 3279, April 2002.


D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile* RFC 5280, May 2008.


DFN *https://www.pki.dfn.de/*


CryptoShop *http://www.cryptoshop.com/index.php*


Knowledge Base *http://kb.psw.net/*


Signaturgesetz *http://www.gesetze-im-internet.de/sigg_2001/*


SMTP and Transport Layer Security
http://www.fehcom.de/qmail/smtptls.html