X.509/SSL/TLS

Verschluesselung

X.509

- Public-Key-Infrastruktur für digitale Zertifikate
- identifiziert Kommunikationspartner

SSL

- Secure Sockets Layer
- Verschlüsselung auf Verbindungs-Schicht

TLS

- Transport Layer Security
- TLS v1.0 = SSL v3.1

X.509

X.509 ist ein Standard der ITU, welcher ursprünglich für X.500 Datenbank/ Index-Dienste entwickelt und in seiner ersten Form 1988 veröffentlicht wurde.

Die X.509 Standards wurden von der IETF “adoptiert” um als PKI für verschiedene Protokolle zu dienen.

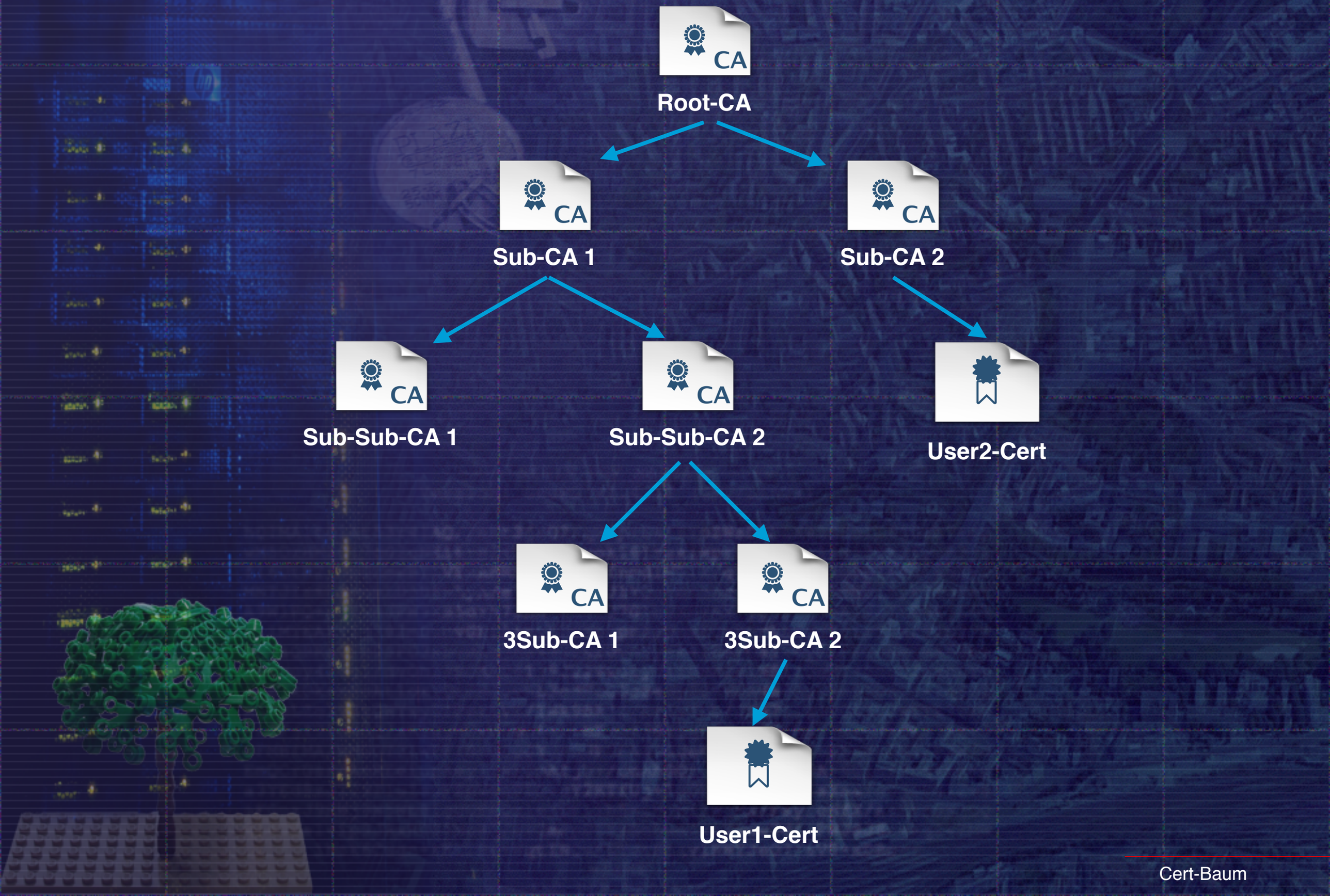
Ein X.509 Zertifikat bindet eine Identität (Namen) an einen Public-Key, wie zum Beispiel:

- E-Mail Adresse
- Hostnamen im DNS

Da X.509 ursprünglich für X.500 entwickelt wurde, sind heutzutage viele Features des Standards ungenutzt.

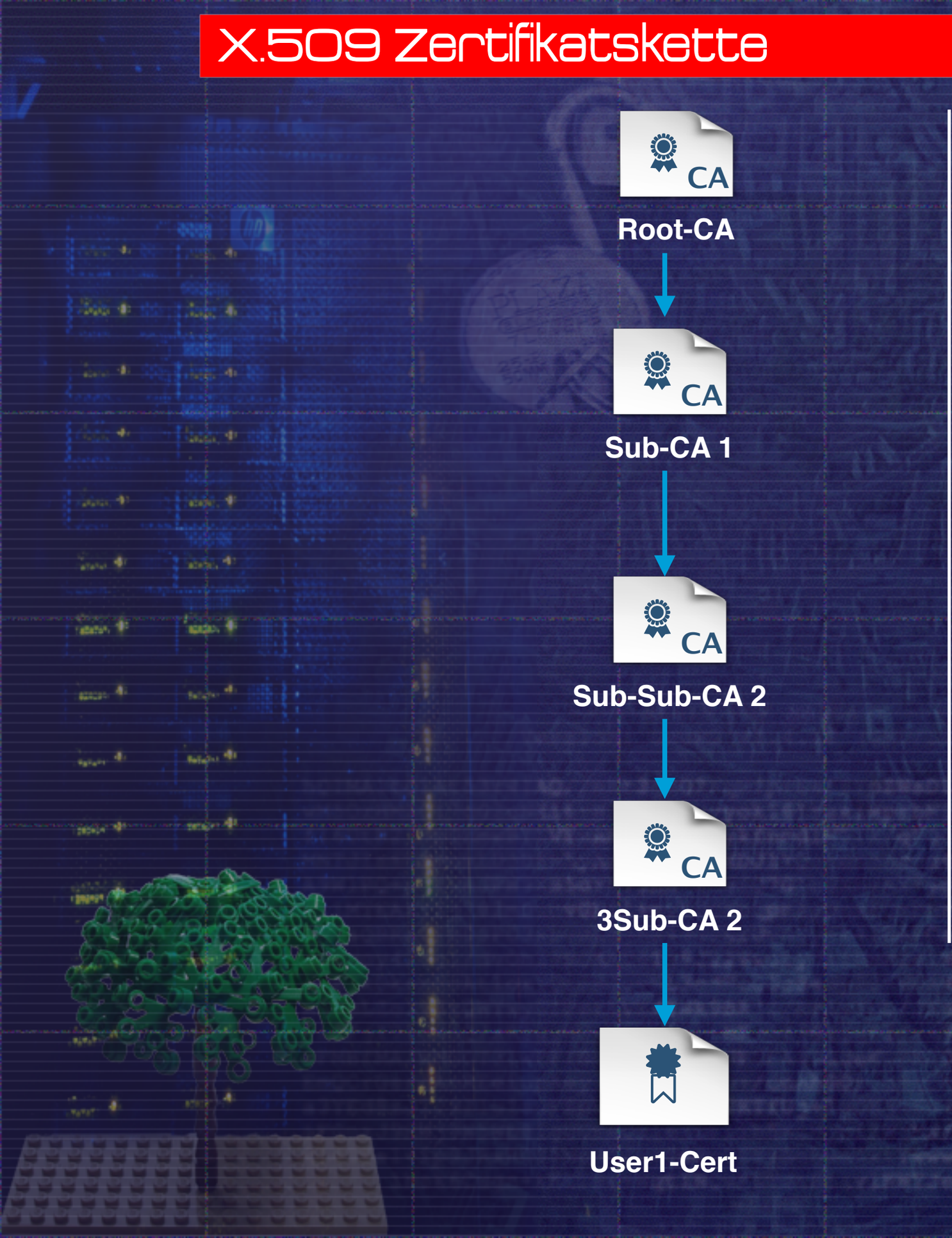
Anders als bei z.B. PGP erlaubt die normale X.509-PKI nur eine Signatur pro Zertifikat, ist also streng hierarchisch aufgebaut.

X.509 Zertifikatskette



Cert-Baum

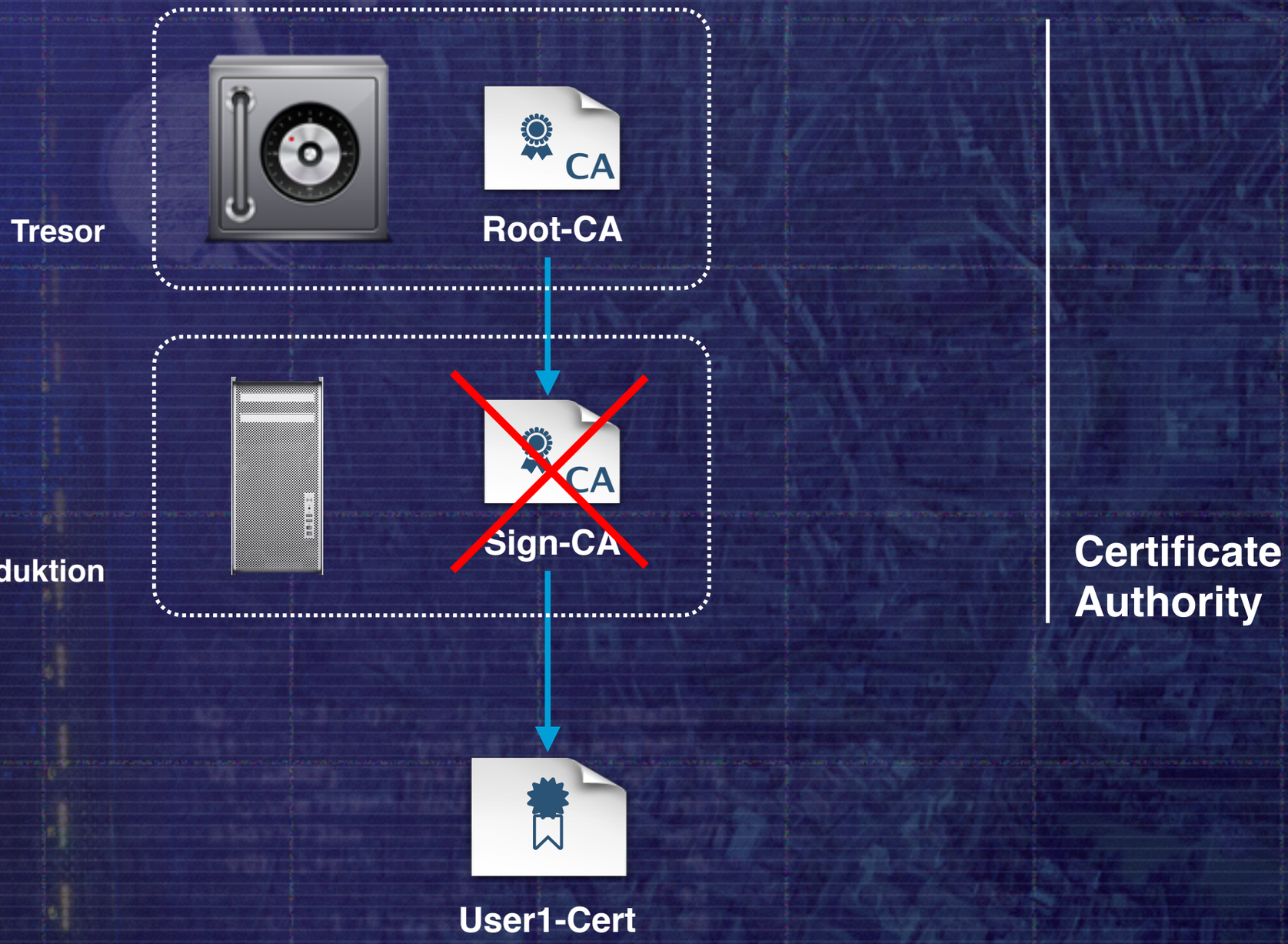
X.509 Zertifikatskette



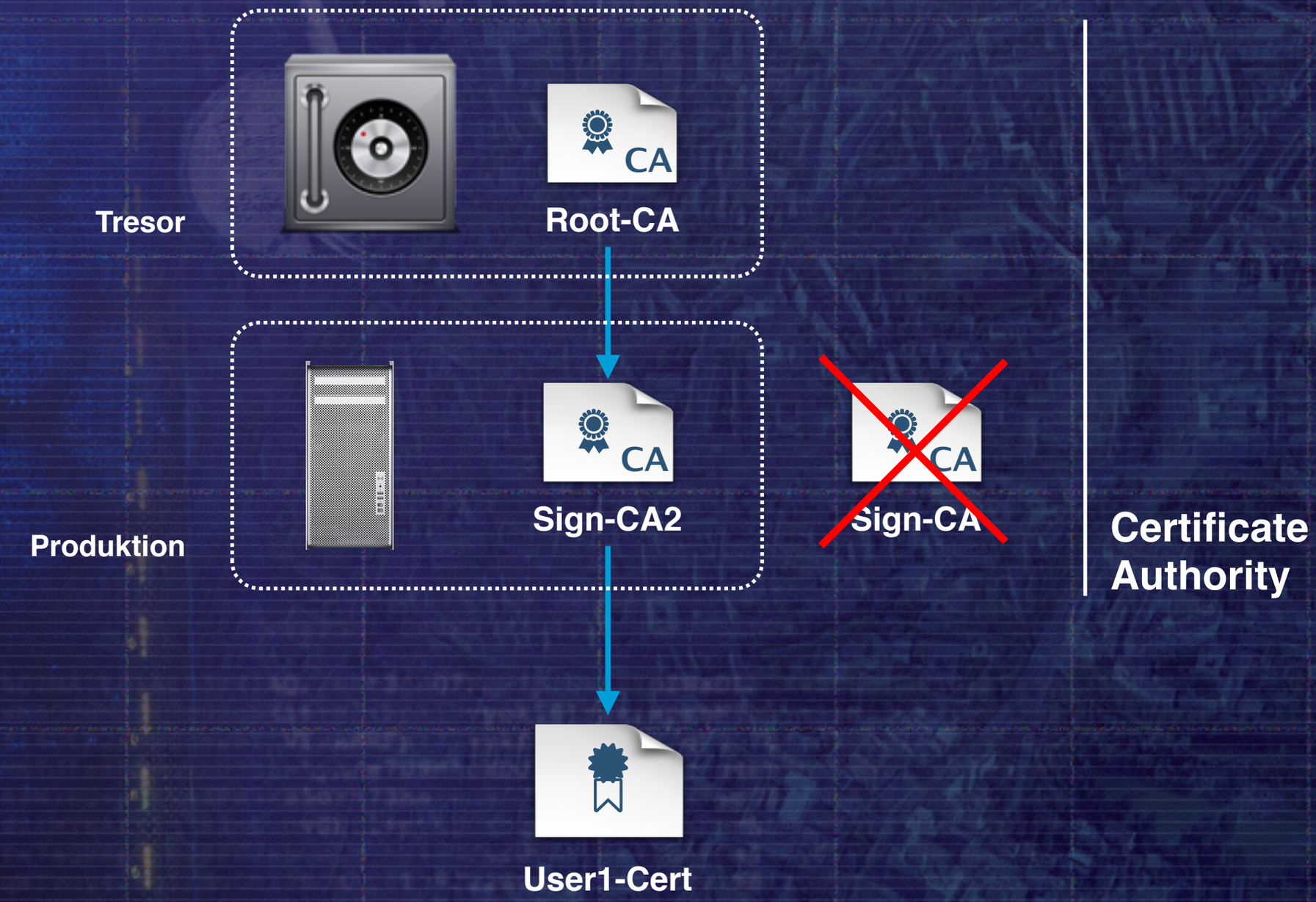
Certificate Chain (Zertifikatskette)

Cert-Baum

X.509 Zertifizierungspraxis



X.509 Zertifizierungspraxis



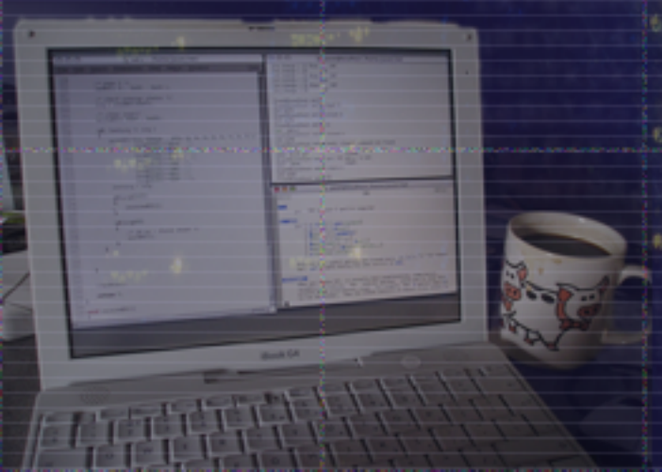
X.509 Nutzung

X.509 wird unter anderem von folgenden Protokollen genutzt:

- HTTPS
- SMTP-S (SMTP+TLS)
- IMAP-S (IMAP+TLS)
- POP3-S
- EAP

- S/MIME
- IPSec
- OpenVPN

- LDAP
- PDF (Adobe)



X.509 Schluessel

X.509 verwendet eine Kombination aus Privatem Schlüssel (“private key”) und öffentlichem Schlüssel (“public key”).

Private Schlüssel können mit einer “Passphrase” geschützt werden, meist ist dies aber für Server-Anwendungen schwer möglich, so dass Schlüssel meist nur durch das Betriebssystem geschützt werden.

X.509 kann verschiedene kryptographische Algorithmen verwenden:

- Rivest-Shamir-Adelman (RSA)
- Digital Signature Algorithm (DSA)
- Diffie-Hellman (DH)
- Key Encryption Algorithm (KEA)
- Elliptic Curve Digital Signature Algorithm (ECDSA)
- Elliptic Curve Diffie-Hellman (ECDH)

X.509 Zertifikat

Zertifikat wird an einen Distinguished Name (DN) gebunden:

- C=DE, L=Nuremberg, O=Kommunikationsnetz Franken e.V., OU=Server, CN=www.franken.de

Aussteller des Zertifikats wird ebenfalls über einen DN identifiziert:

- C=DE, L=Nuremberg, O=Kommunikationsnetz Franken e.V., OU=CA, CN=KNF Certificate Authority

Zertifikat enthält Gültigkeitszeitraum:

- Not before <Datum>, Not after <Datum>

Zertifikat enthält Public-Keys der zertifizierten Instanz

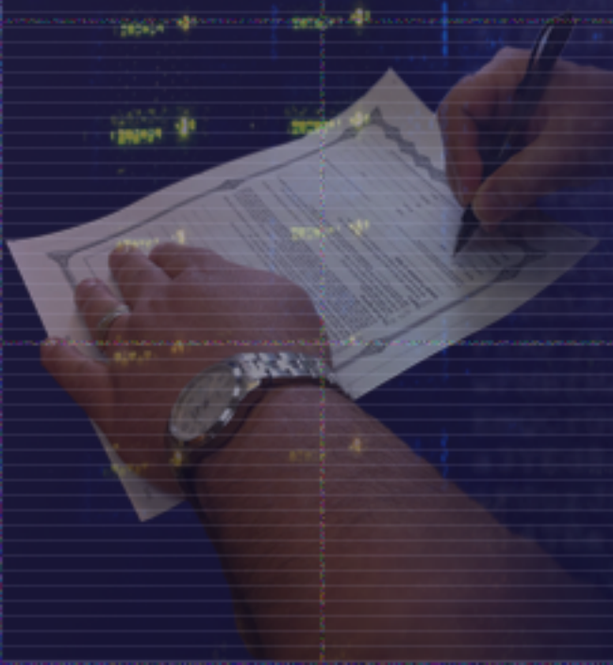
Zertifikat enthält Referenz(en) auf Widerrufsliste der CA

- Link zu statischem File (CRL)
- Referenz zu OCSP

X.509 Zertifikat

Zertifikat enthält Information über erlaubte Nutzungen:

- Basic Constraints:
  - CA
  - nicht-CA
- Key Usage:
  - Signatur
  - Verschlüsselung
  - CRL-Signatur
  - …
- Extended Key Usage:
  - TLS WWW server authentication
  - TLS WWW client authentication
  - Code Signing
  - Email Protection
  - OCSP Signing
  - …

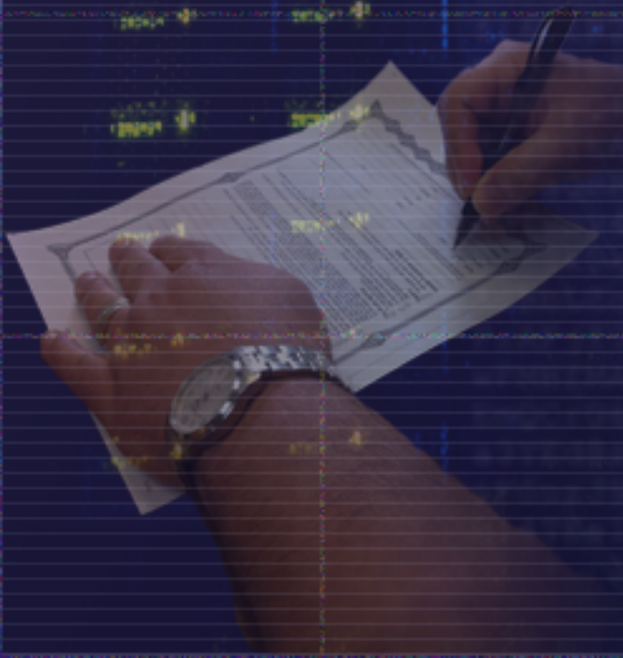


X.509 Certificate Revocation

Ein Zertifikat ist für einen Gültigkeitszeitraum ausgestellt

Ein Zertifikat kann einen „Link“ zu einer Revocation List (Sperrliste) enthalten.

- Ein Zertifikat kann nur von dem Zertifikats-Aussteller gesperrt werden (nicht Inhaber)
- Gängig waren Listen, in denen alle gesperrten Zertifikate enthalten sind
- Heutzutage setzt sich OCSP (Online Certificate Status Protokoll) durch
  - via HTTP
  - Positiv-Auskünfte (nicht spezifiziert, aber vom deutschen SIG-Gesetz gefordert)
  - Negativ-Auskünfte
  - Implementierungen meist sehr buggy



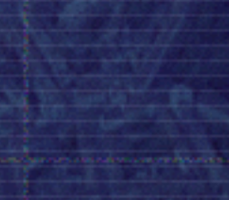
X.509 Online Certificate Status Protocol

- anders als File-basierende Listen müssen lokal keine großen Datenbanken gehalten werden
  - via HTTP
  - Positiv-Auskünfte (nicht spezifiziert, aber vom deutschen SIG-Gesetz gefordert)
  - Negativ-Auskünfte

- Implementierungen meist sehr buggy
- Clients reagieren auf fehlende Antworten mit Akzeptanz des zu verifizierenden Zertifikats
- Google Chrome hat OCSP ausgeschalten, wegen Performance und Privatsphären-Problemen

X.509 Zertifizierung

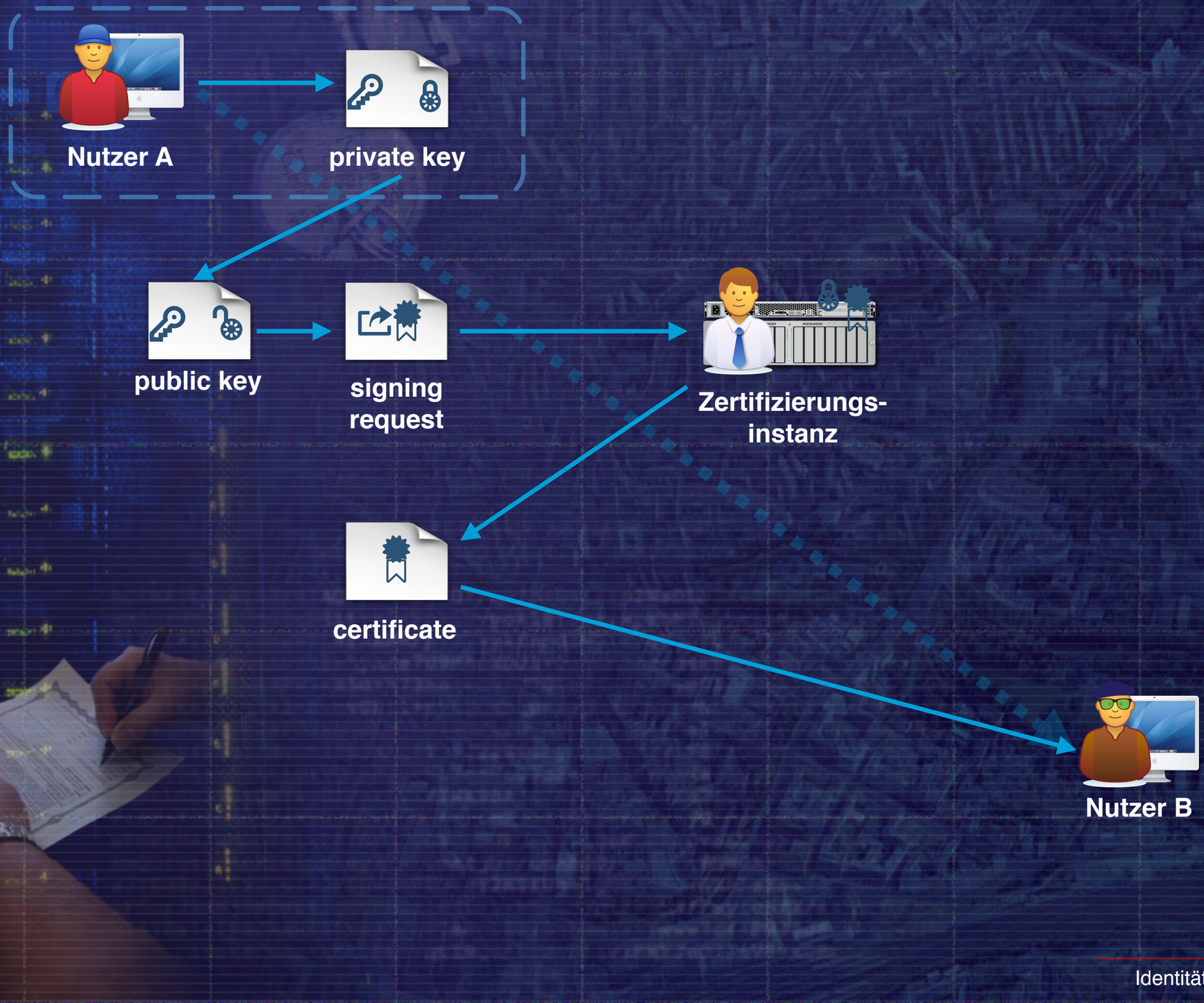
Nutzer A



Nutzer B

Identitätsnachweis

X.509 Zertifizierung



X.509 Zertifizierung

Nutzer A

private key

signing request

=

request info

+

public key

certificate

=

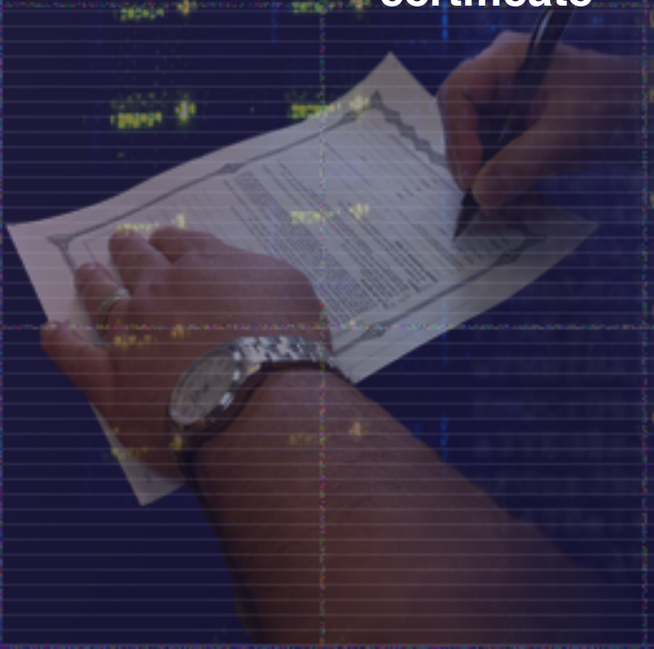
public key

+

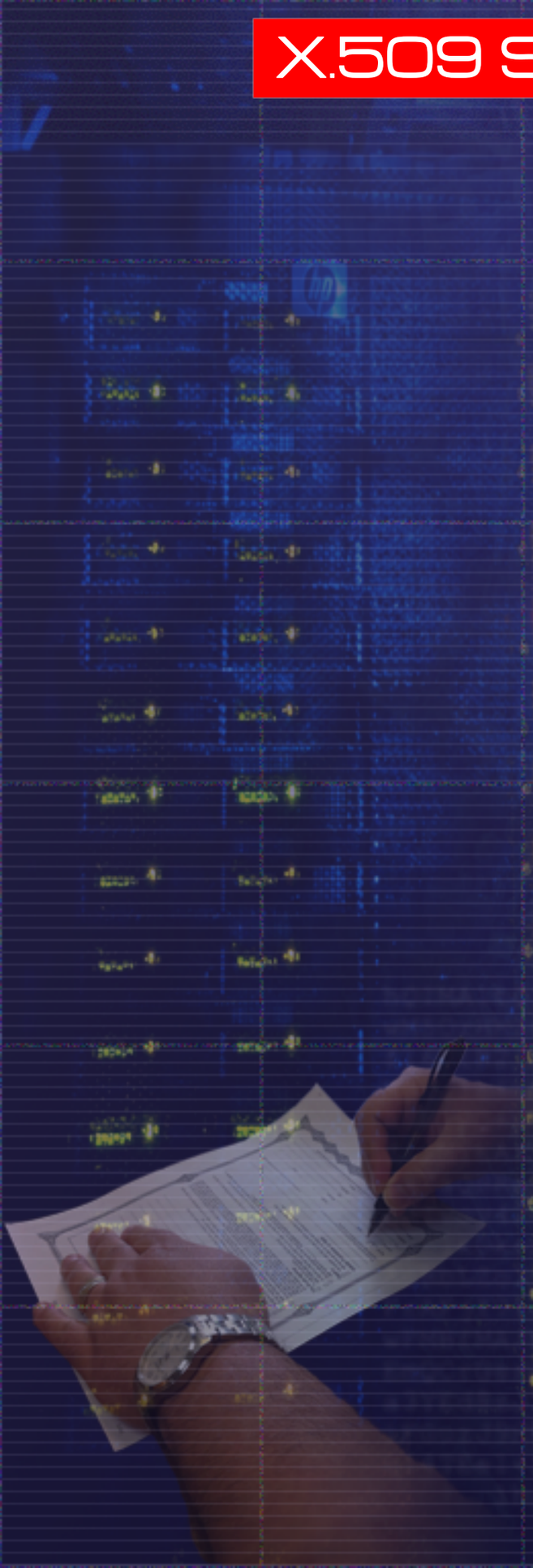
info

+

signature



X.509 Selbstsignierung



Root-CA

signiert durch
???

Sign-CA

signiert durch
Root-CA

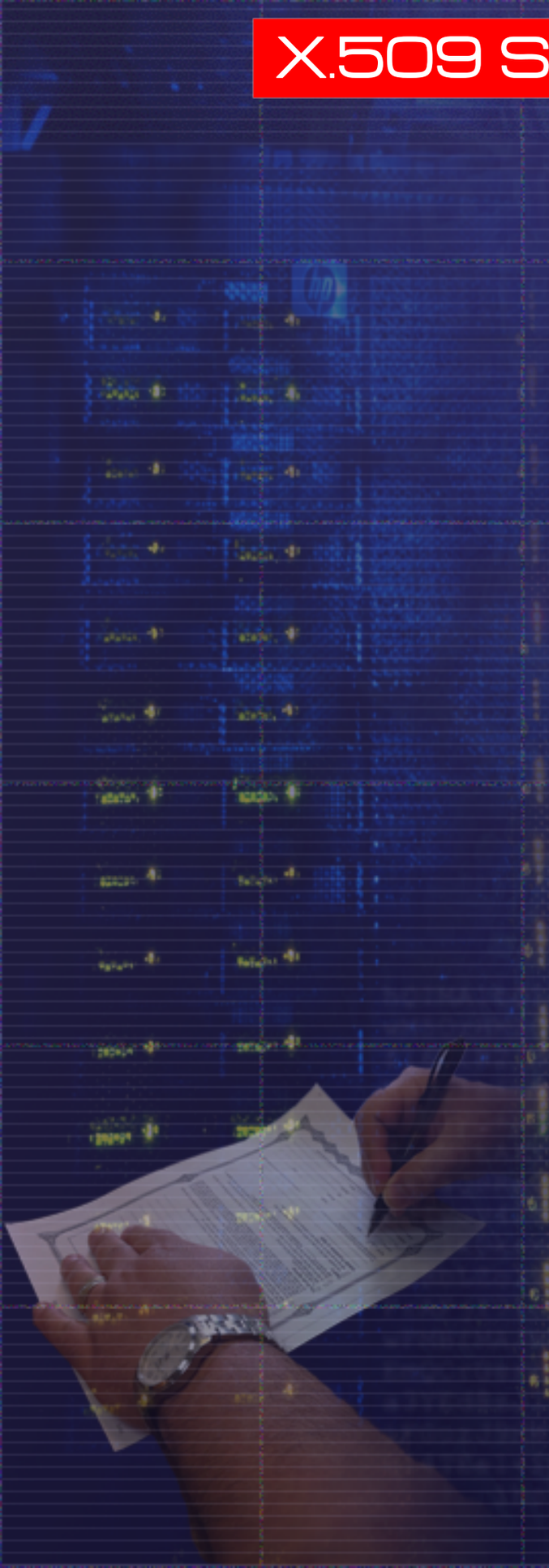
User1-Cert

signiert durch
Sign-CA

Certificate Authority

self sign

X.509 Selbstsignierung



Root-CA

signiert durch sich selbst

Sign-CA

signiert durch Root-CA

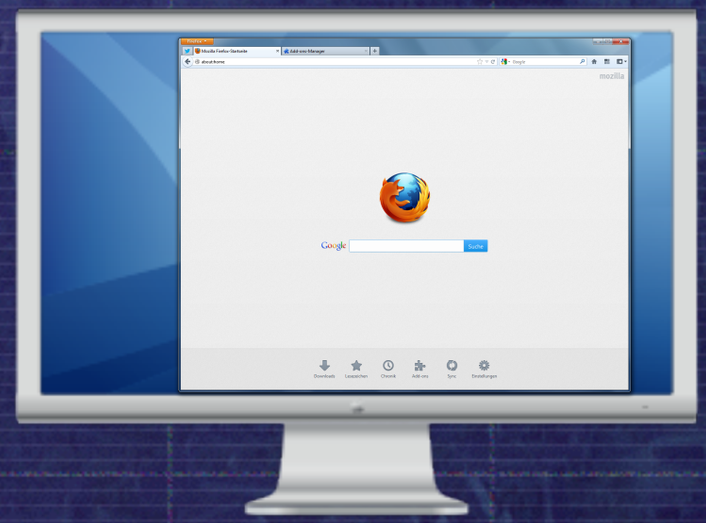
User1-Cert

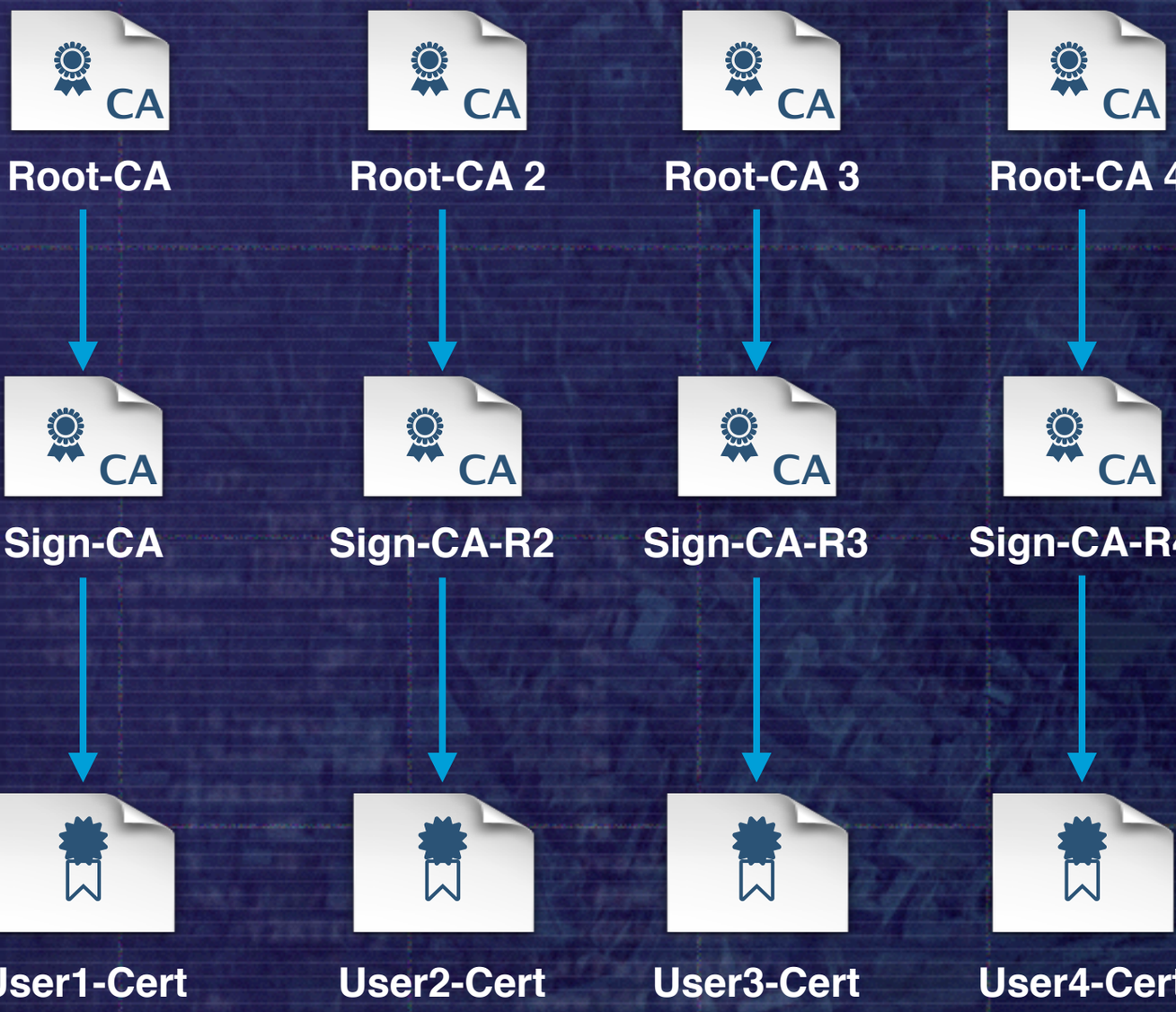
signiert durch Sign-CA

Certificate Authority

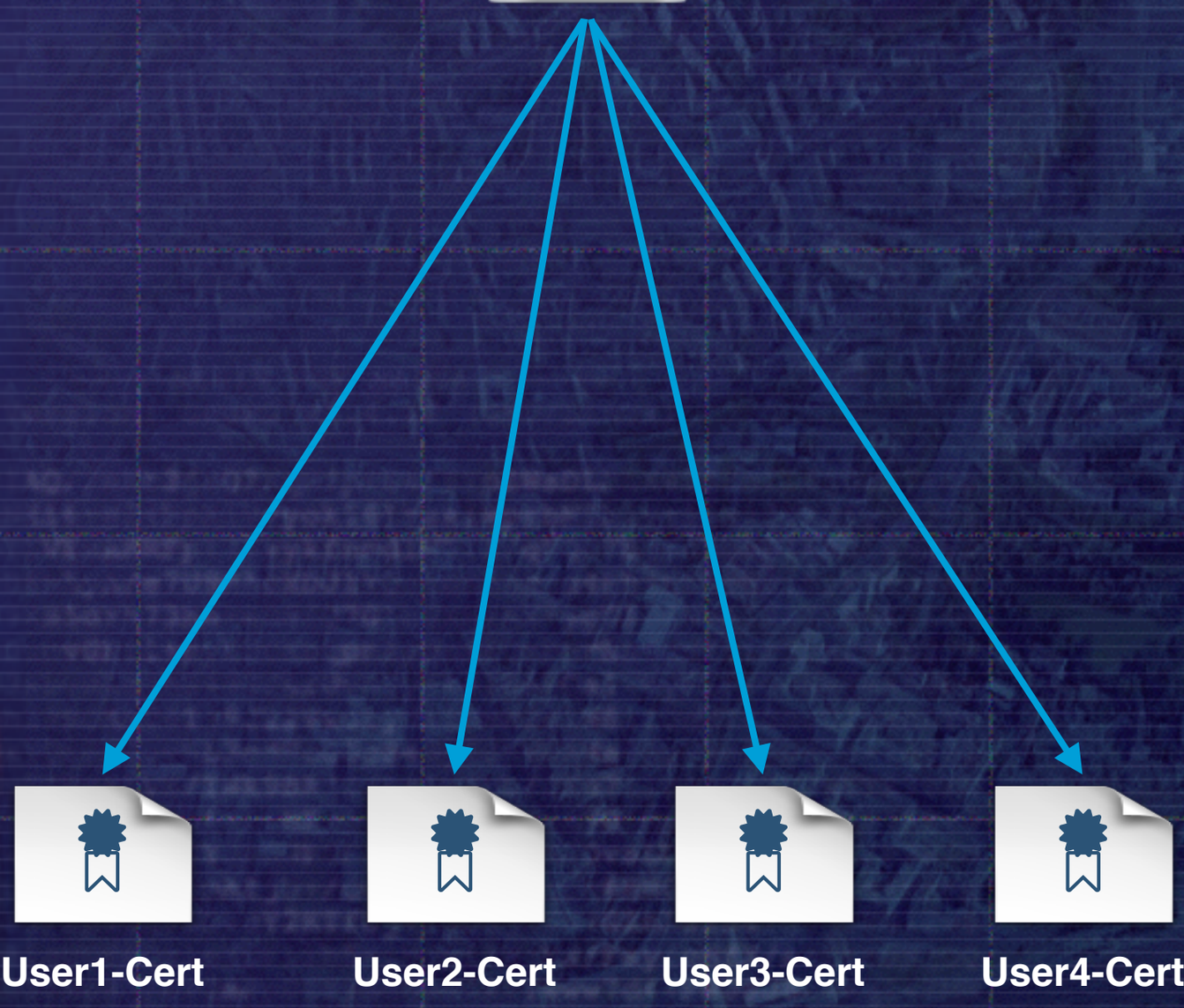
self sign

X.509 Root Store





X.509 Root Store



CA list

X.509 Root Store

CA “A”

CA “B”

User1-Cert

www.google.de

User2-Cert

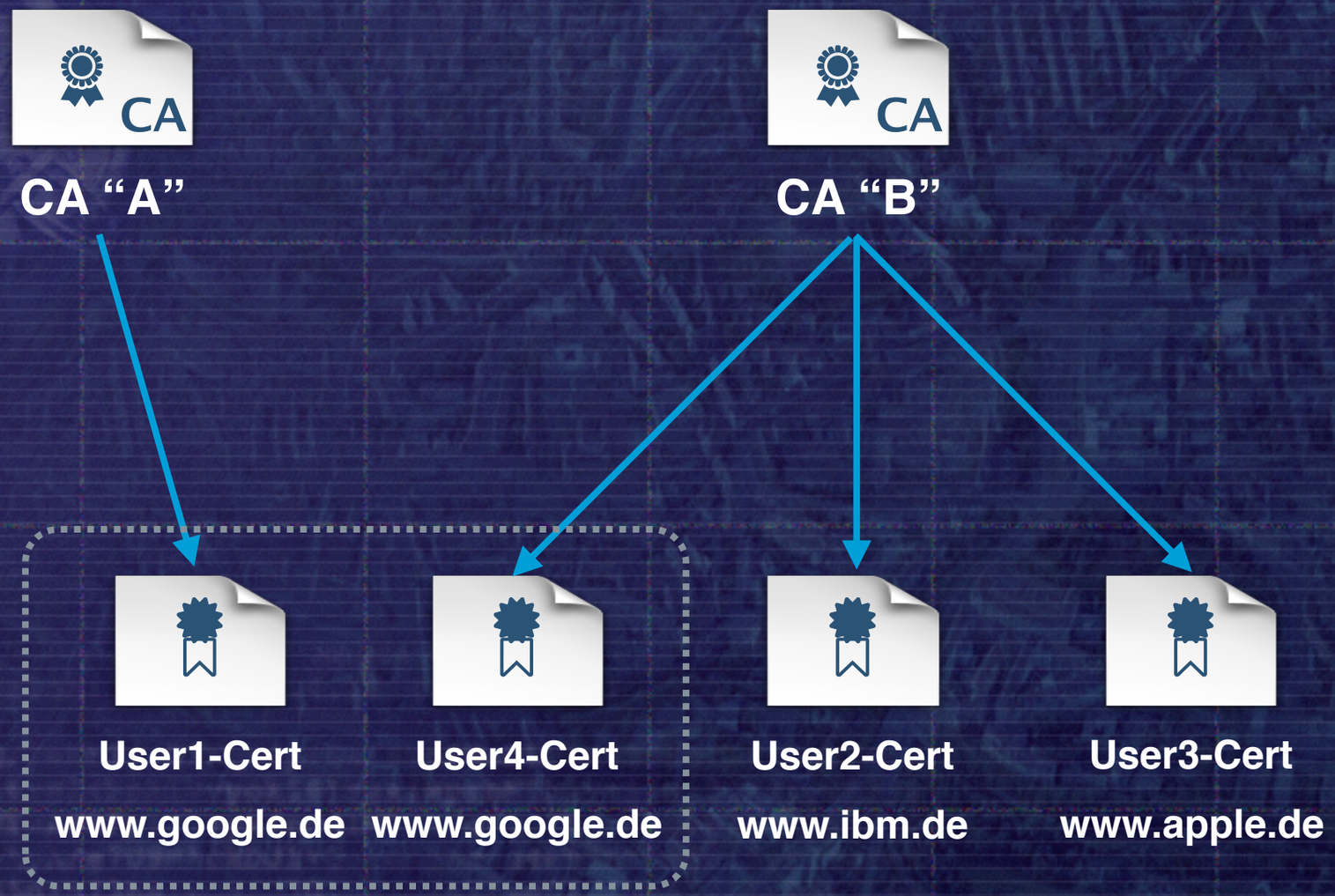
www.ibm.de

User3-Cert

www.apple.de

CA list

X.509 Root Store



ein Browser verifiziert den Zielserver einer HTTPS-Verbindung durch Vergleich von Hostnamen aus der URL mit dem CN des Zertifikats

stimmen Hostname und Zertifikat überein und ist das Zertifikat Teil einer akzteptierten Certificate-Chain gilt die Verbindung als “sicher”

X.509 Root Store

Software enthält eine Liste von CAs, deren Zertifikate erkannt werden

- Jede Software hat potentiell eigene Liste
  - Apple: Keychain
  - Windows: Windows Certificate Store
  - Linux: oft /etc/ssl/
- Mozilla-Software enthält Liste von 153 Zertifikaten (16.11.2014)
- Ein Zertifikat kann im Normalfall nur von einer CA abstammen, aber ein Identifier kann von mehreren Zertifizierungsinstanzen signiert worden sein, d.h. es gibt potentiell mehr als ein Zertifikat für ein Objekt.

X.509 Zertifizierungsstelle

Es gibt eine Vielzahl von öffentlichen Zertifizierungsstellen, die von gängiger Software akzeptiert werden

Preise variieren von Aussteller zu Aussteller und Produkt zu Produkt

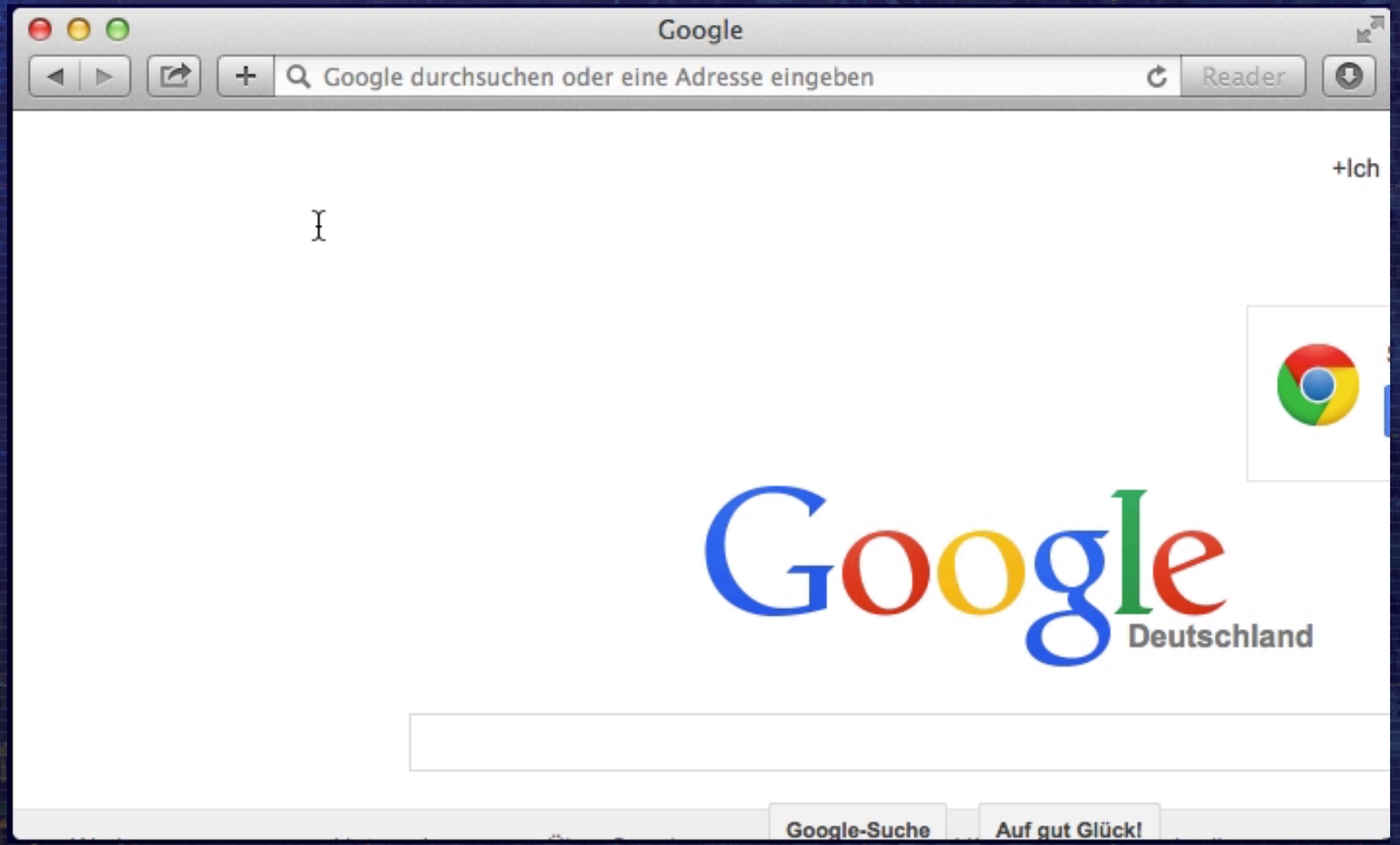
- Domainvalidiert
- Organisationsvalidiert
- Extended Validation

Browser zeigen für alle Zertifikate das “Sicherheitsicon”, bei Extended Validation aber eine besonders auffällige Sicherheitsbestätigung

X.509 Zertifikat

Domain-Validiert

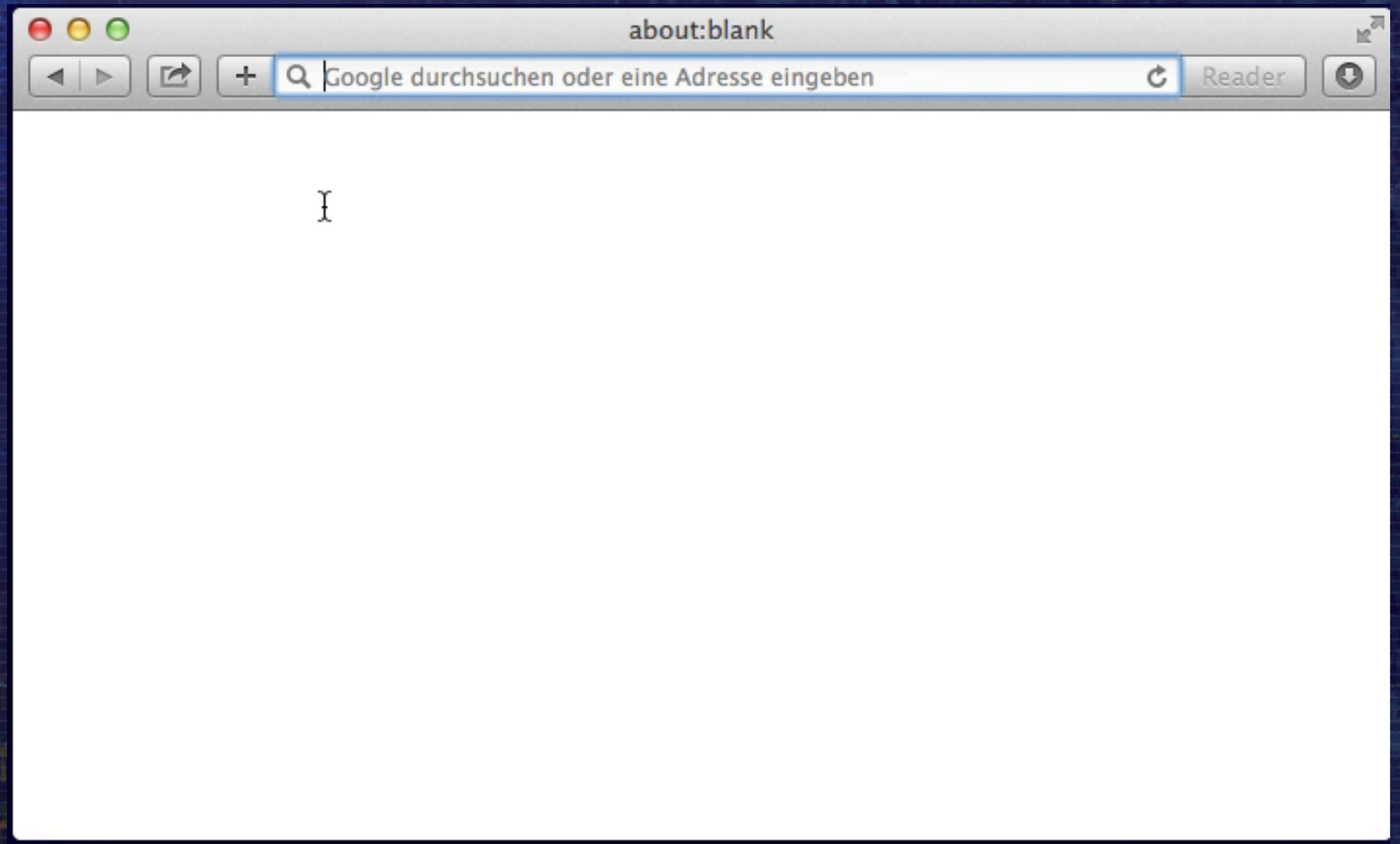
- Bestätigung des Empfangs einer E-Mail an die Domain



X.509 Zertifikat

Organisations-Validiert

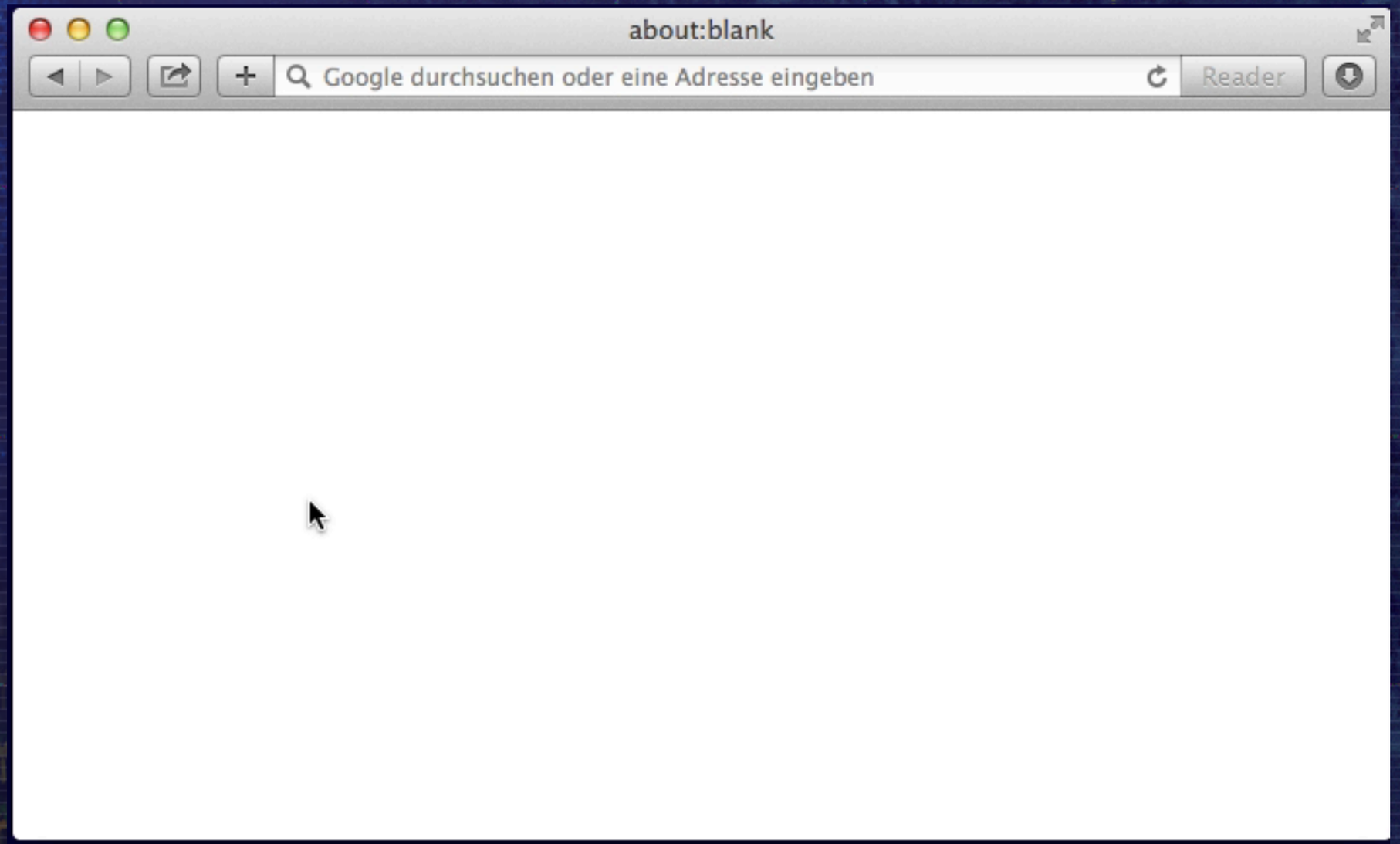
- Kommunikation via E-Mail und/oder Telefon
- schriftliche Dokumentation über Besitz der Domain



X.509 Zertifikat

EV - Zertifikat

- schriftliche Dokumentation über Besitz der Domain
- erweiterte Überprüfung des Antragstellers



X.509 Sicherheit

Die Sicherheit der X.509 PKI hängt nicht nur vom Schutz des privaten Schlüssels ab

Die Public-Key-Komponente besteht fast ausschliesslich aus RSA

- historische Wahl von zu wenigen Bits
- erzeugt zu einem Zeitpunkt, als die Software vorhersehbaren Zufall verwendete
- Mindestlänge heute 2048 Bits
- DSA und EC sind nahezu nicht in Verwendung

In der Vergangenheit wurden Keys mit MD5 signiert, dies gilt heute nicht mehr als ausreichend

- MD5-Kollision gegen RapidSSL CA berechnet in ~2 Tage auf einem Cluster aus 200 Sony PS3 (2008)
- “flame”-Virus hat sich durch erfolgreiche MD5-Kollision als Update-Server in Microsoft-Netzwerken ausgegeben (2012)
- sha1 als Ersatz ist bereits wieder in der Ablösephase
- sha2 neuer Standard als Hash-Algorithmus für X.509 Zertifikate

SSL und TLS

SSL

- Secure Sockets Layer
- Verschlüsselung über Verbindungs-Schicht

TLS

- Transport Layer Security
- TLS v1.0 = SSL v3.1

SSL und TLS

SSL wurde durch Netscape entwickelt/erfunden um Daten auf Verbindungsebene zu verschlüsseln

SSLv2 hat viele Schwachstellen, in fast allen Applikationen ausgeschalten

SSLv3 (1996) gilt heutzutage als unsicher und wird von vielen Applikationen abgelehnt

TLS1.0 entspricht SSL v3.1 (1999)

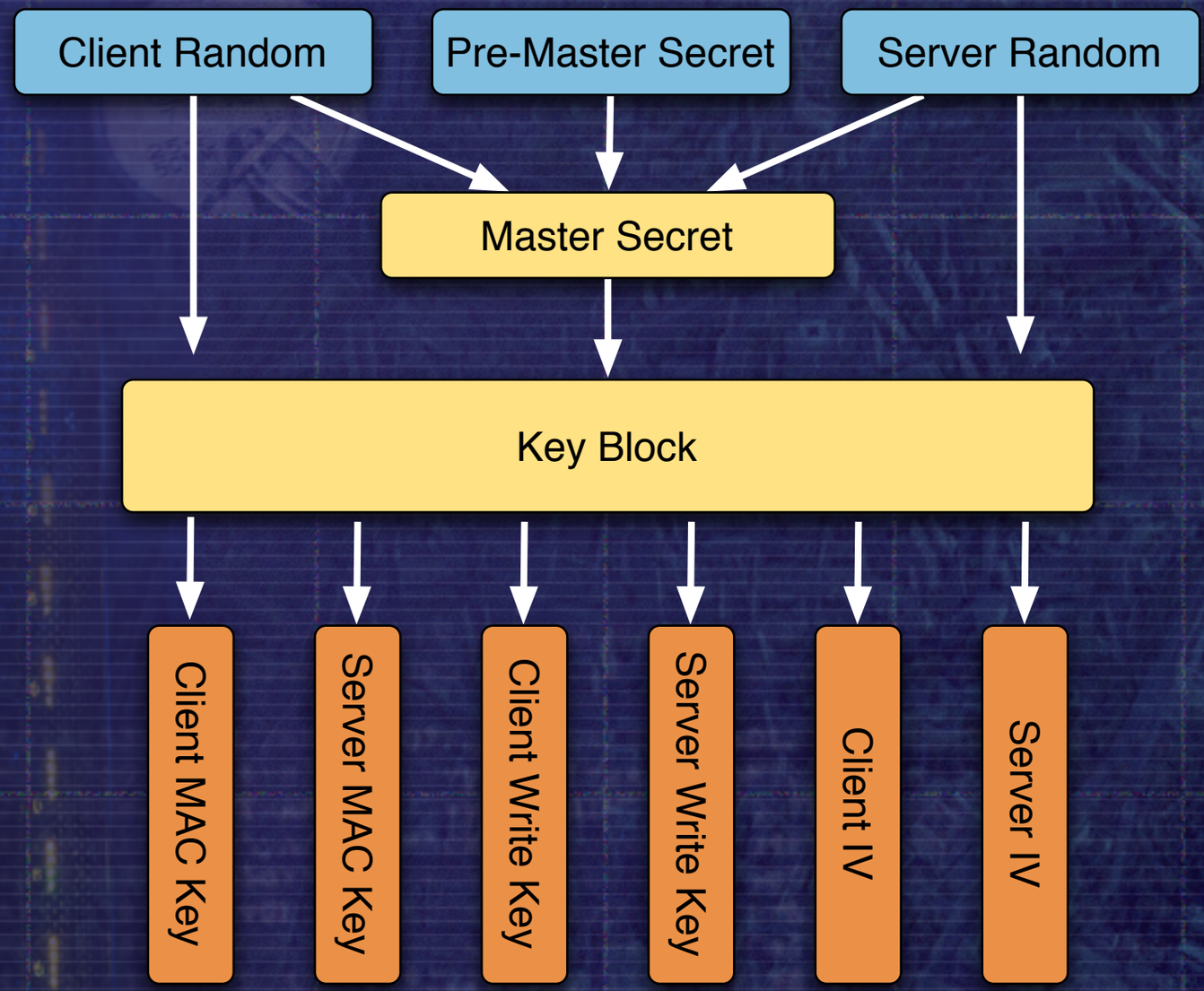
Um die Verwendung von SHA1 auszuschliessen muss TLS1.2 verwendet werden, verbreitete Software unterstützt aber nur TLS1.0

SSL/TLS Session Key

Client und Server einigen sich nach Verbindungsaufbau auf Verschlüsselungsalgorithmus

- Client und Server tauschen “Random” aus
- Client schickt einen “Pre-Master-Key” verschlüsselt mit dem Public-Key des Servers
- Mit dem Random und dem Pre-Master-Key werden Session-Keys berechnet:
  - Client MAC (Message Authentication Code)
  - Server MAC (Message Authentication Code)
  - Client Write
  - Server Write
  - Client IV
  - Server IV

SSL/TLS Session Key



SSL/TLS Session Key

Problem:

- Die assymetrische Verschlüsselung (RSA) ist teuer
- Anwendungen bauen zum Nachrichtenaustausch oft viele Verbindungen auf (HTTP/POP3S/IMAPS)

Neue Verbindungen zu bekanntem Partner recyceln daher das Master-Secret

- vermeidet unnötige Public-Key-Kryptographie

Gespeichertes Master-Secret wird mit neuem Random kombiniert um neue Session-Keys zu berechnen

Server verwendet zur Zuordnung die Session-ID, Clients verwenden den Hostnamen, damit kann das alte Master-Secret auch über IP-Adresswechsel hinaus recyclet werden

SSL/TLS Konzept

SSL & TLS basieren auf einem Multi-Layer Protokollkonzept:

- Handshake Protocol
  Authenticated Key Exchange (AKE) Protokoll um kryptographische Secrets und Algorithmen auszuhandeln

- Record and Application Data Protocol
  Eine zwischen Applikation und TCP-Layer geschaltene MAC-then-PAD-then-Encrypt Ebene

- Alert Protocol
  Zur Übertragung von Fehlermeldungen

- ChangeCipherSpec Protocol
  Besteht nur aus einer Nachricht, die einen Wechsel des Verschlüsselungsalgorithmus einleitet

SSL/TLS Security

Heartbleed

- Bug in der „Heartbeat“ Implementation von OpenSSL konzipiert um ein Datenpaket an den Kommunikationspartner zu schicken, das dieser unverändert zurücksendet
- ein Implementierungs-Fehler führte dazu, dass bis zu 64kByte des von dem auf der Gegenseite laufenden Programms genutzten Speichers ausgelesen werden konnte

Poodle

- Angreifer muss Man-in-the-Middle sein (Traffic mitlesen können)
- Client und Server müssen sich auf SSLv3 einigen
- nach aktuellem Kenntnisstand muss der Angreifer dem Client Code unterschieben können (z.B. Java-Script auf einer Webseite)
- Client wird dann bekannte Pakete an den Server schicken und der MitM kann die verschlüsselten Pakete dann analysieren und ist am Ende in der Lage alle übertragenen Pakete zu dekodieren

SSL/TLS Security

- Komplexität bringt Probleme

- Ciper Suite Rollback (Rückfall auf schwächere Verschlüsselung)
- Version Rollback (Rückfall auf anfällige Version von SSL)
- Timing-Attacken (Antwortzeiten auf spezielle Anfragen lassen Rückschlüsse über RSA-Parameter zu)
- Struktur des Protokolls erlaubte vereinfachte Berechnung des Pre-Master-Keys
- Padding von Messages nicht in MAC enthalten
- Angriff auf CBC ohne Padding Länge, so daß Folgeblocks beeinflusst werden können
- Timing plus Padding plus Known Plain-Text Angriff auf IMAP-Logins
- SSL Initialization Vector nur bei erster Message Random, danach basierend auf vorherigem Block (fixed in TLS 1.1)
- Teile von TLS-Messages sind nicht verschlüsselt, dies erlaubt Rückschlüsse auf Message-Inhalte (z.B. Länge einer URL bei HTTPS)
- Alert-Messages nicht geschützt und daher fälschbar
- Mangel bei TLS-Neuaushandlung erlaubt einschieben von Daten ohne die Session zu zerstören

SSL/TLS Security: OpenSSL 2012-2014

- CVE-2014-3513: 15th October 2014
- CVE-2014-3567: 15th October 2014
- CVE-2014-3568: 15th October 2014
- CVE-2014-3508: 6th August 2014
- CVE-2014-5139: 6th August 2014
- CVE-2014-3509: 6th August 2014
- CVE-2014-3505: 6th August 2014
- CVE-2014-3506: 6th August 2014
- CVE-2014-3507: 6th August 2014
- CVE-2014-3510: 6th August 2014
- CVE-2014-3511: 6th August 2014
- CVE-2014-3512: 6th August 2014
- CVE-2014-0224: 5th June 2014
- CVE-2014-0221: 5th June 2014
- CVE-2014-0195: 5th June 2014
- CVE-2014-3470: 30th May 2014
- CVE-2014-0198: 21st April 2014
- CVE-2010-5298: 8th April 2014
- CVE-2014-0160: 7th April 2014
- CVE-2014-0076: 14th February 2014
- CVE-2013-4353: 6th January 2014
- CVE-2013-6449: 14th December 2013
- CVE-2013-6450: 13th December 2013
- CVE-2012-2686: 5th February 2013
- CVE-2013-0166: 5th February 2013
- CVE-2013-0169: 4th February 2013
- CVE-2012-2333: 10th May 2012
- CVE-2012-2131: 24th April 2012
- CVE-2012-2110: 19th April 2012
- CVE-2012-0884: 12th March 2012
- CVE-2011-4108: 4th January 2012
- CVE-2011-4109: 4th January 2012
- CVE-2011-4576: 4th January 2012
- CVE-2011-4577: 4th January 2012
- CVE-2011-4619: 4th January 2012
- CVE-2012-0027: 4th January 2012
- CVE-2012-0050: 4th January 2012

SSL/TLS: Perfect Forward Secrecy

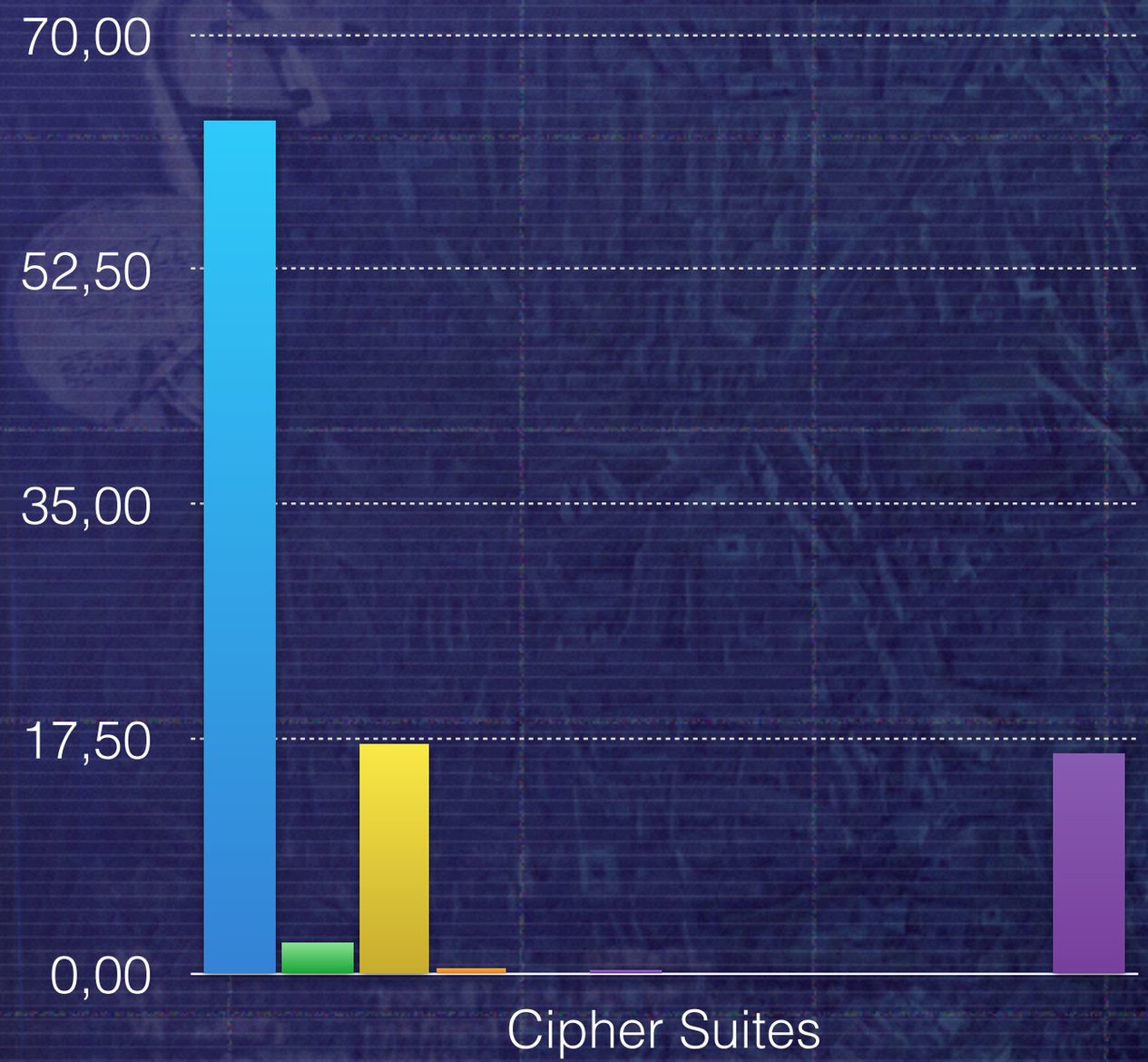
Bei Perfect Forward Secrecy ermöglicht die Kenntnis über den Langzeit-Schlüssel nicht das decodieren einer mit einem temporären Schlüssel geschützter Übertragung

- Nachrichten können nicht nachträglich dekodiert werden, nachdem die Private-Keys der Kommunikation bekannt wurden
- Das Konzept ist bereits seit 1992 bekannt, ist aber erst in relativ neuen Crypto-Implementationen enthalten
- Für jede Session werden neue Public-Keys ausgehandelt
- Führt zu deutlich erhöhter Berechnungslast der Kommunikationspartner
- Schützt nur die Keys, Angriffe auf die Verschlüsselungs-Algorithmen und -Implementierungen werden von PFS nicht unbedingt erschwert
- In TLS:
  - Diffie-Hellman basierende PFS DHE-RSA, DHE-DSS
  - Elliptische Kurfen Diffie–Hellman basierende PFSs ECDHE-RSA, ECDHE-ECDSA

SSL/TLS Security: Clients/Server

- API sehr komplex, viele dringend notwendige Dinge müssen händisch erledigt werden
  - Match von Partner und DN
  - Gültigkeit von Zertifikaten (Valid From/To)
  - Länge der Zertifikatskette
  - X.509 Extensions
  - Certificate Revocation Check

SSL/TLS: Cipher Suites



- AES128-SHA
- AES256-SHA
- RC4-SHA
- DES-CBC3-SHA
- ECDHE-RSA-AES128-SHA
- ECDHE-RSA-AES256-SHA
- ECDHE-ECDSA-AES128-SHA
- ECDHE-ECDSA-AES128-SHA
- DHE-DSS-AES128-SHA
- DHE-DSS-AES256-SHA
- EDH-DSS-DES-CBC3-SHA
- RC4-MD5

Key Pinning

Hintergrund:

viele Skandale um unberechtigt ausgestellte Zertifikate von verschienen Zertifizierungsstellen:

- India CCA (Indien)
- Diginotar (Niederlande)
- Turktrust (Turkey)
- Comodo (England, USA)
- …

Key Pinning

Problem:

- mehr als eine Zertifizierungsstelle kann ein Zertifkat auf einen Distinguished Name ausstellen
- es langt ein Fehler oder Court-Order bei einer Zertifizierungsstelle um das X.509 Trust-System zu kompromittieren

Lösung:

- bekanntes Zertifikat für ein Ziel verankern (pinnen)

Sobald Informationen über ein Zertifikat einer Site bekannt ist, wird dieses als einziges Zertifikat für dieses Ziel akzeptiert; andere Zertifikate, auch wenn sie von einer als vertrauenswürdig eingestuften Instanz ausgestellt wurden, werden nicht anerkannt

- SneakerNet
- technische Systeme

Key Pinning

Systeme:

- In-Browser-Liste
  - Mozilla hat eine Liste von bekannten „großen“ Sites und den verwendeten Certificate Authorities, die für diese Zertifikate ausstellen
    - hilft nur bei den „großen Websites“
    - hilft nicht, wenn rogue-certificate von gleicher CA ausgestellt wurde
- Plugins
  - Beim ersten Besuch einer „SSL-Site“ wird deren Zertifikats-Hash gespeichert, wird bei einem späteren Besuch ein anderes Zertifikat präsentiert, wird der Nutzer gewarnt
    - hilft nur, wenn die Verbindung nicht bereits beim ersten Verbindungsaufbau gehackt war
    - mit der Zeit große Datenbank und Warnungen werden unvermeidbar, so dass User sich an das „wegklicken“ gewöhnen
- HPKP (Header Public Key Pinning)
- DANE (DNS-based Authentication of Named Entities)
- Souvereign Keys

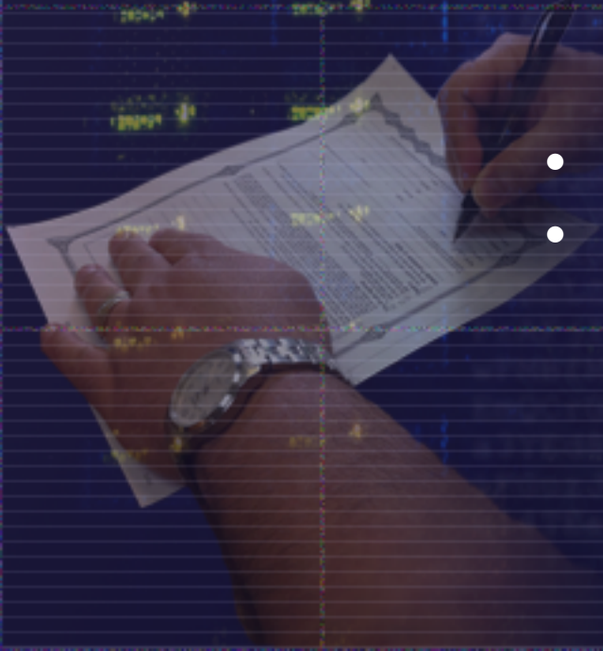
Header Public Key Pinning (HPKP)

Header Public Key Pinning (HPKP):

- IETF Standard kurz vor der Verabschiedung
- Webseite teilt Clients mittels Header mit, welchen Zertifikaten von nun an für sie zu vertrauen sind

```
Public-Key-Pins: pin-sha256="base64=="; max-age=expireTime [; includeSubdomains]
```

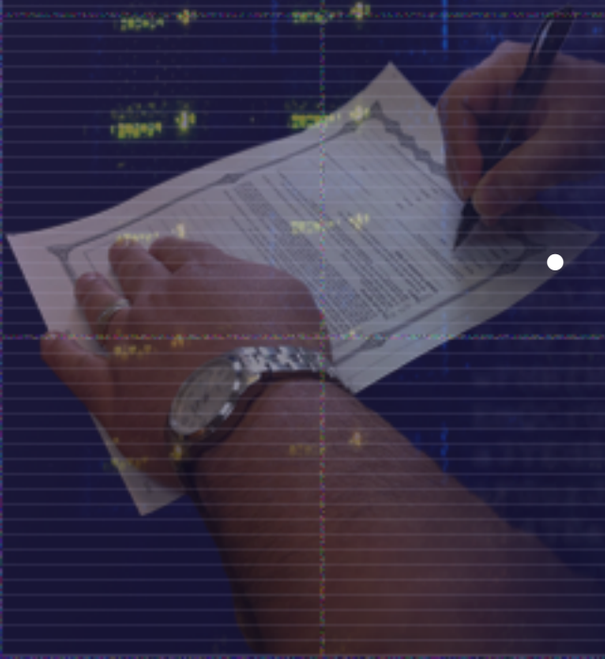
- bisher nur in Firefox
- Pin-Header kann alternative Notfall-Zertifikate enthalten
- kann eine Report-URI enhalten, über die der Browser einen Security-Vorfall melden kann



- benötigt „initial safe contact“ um das Pinning aufzusetzen
- funktioniert nur für HTTPS-Verbindungen

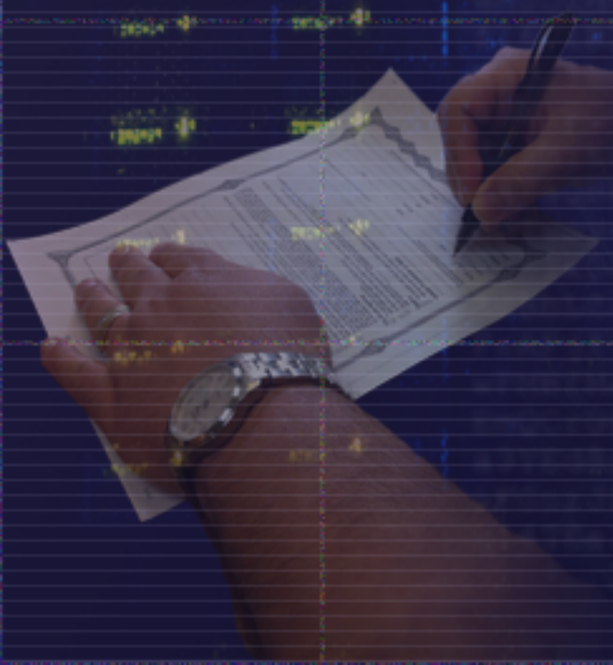
DNS-based Authentication of Named Entities (DANE)

- IETF-Standard: RFC 6698
- Information über Zertifikat(e) zur Domain wird über DNS übertragen
  - Service Certificate Constraints: definiertes Zertifikat, das zudem über die Zertifizierungskette zu einem vertrauenswürdigen Root-Zertifikat gehört
  - Domain-Issued Certificate: definiertes Zertifikat, das nicht weiter verifiziert wird
  - Trust Anchor Assertion: ein Zertifikat muss das spezifizierte Zertifikat in der Zertifizierungskette haben
- unterstützt (via Plugins) von Chrome und Firefox, sowie von Postfix (SMTP-Server), und irssi (IRC client)
- benötigt DNSSEC



Souvereign Keys

- Electronic Frontiers Foundation (EFF) ~2011
- Informationen über Zertifikat(e) zur Domain werden über DNS (DNSSEC) übertragen
- Informationen werden in einer „append only“ Datenbank zentral gehalten, c.a. 10-20 Server
- Einige komplexe Probleme:
  - Widerruf von Zertifikaten
  - Übertragung von Domain an neuen Eigentümer
  - Denial-Of-Service Atacken auf zentrale Server



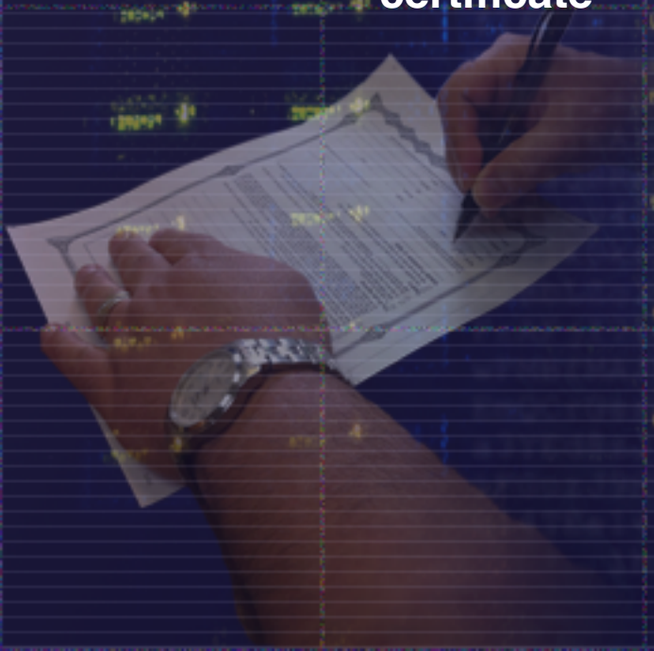
X.509 Certificate Hacking

Nutzer A

private key + public key

signing request = request info + public key

certificate = public key + info + signature



X.509 Certificate Hacking

Nutzer A

private key

+

public key

certificate

=

public key

+

info

+

signature

hash über key und info

signierter hash

c82a4

private key X

public key X

info X

c82a4

X.509 Certificate Hacking

Nutzer A

private key

public key

certificate

=

public key

info

hash über key und info

c82a4

private key X

public key X

info X

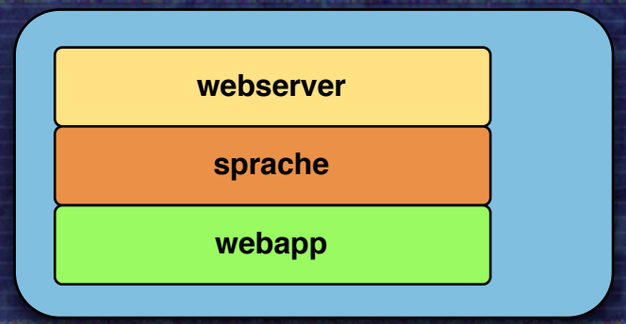
signature

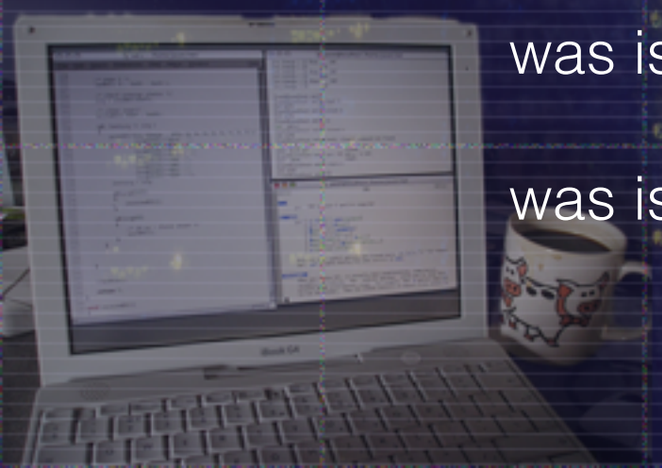
c82a4

signierter hash

Zertifikat

X.509/SSL/TLS und darunter ?





Eine sichere Übertragung ist nur „die halbe Miete“

was ist mit anderen Einfallstoren für Kompromittierung der Kommunikation?

was ist mit Sniffen auf dem Client?

Vielen Dank für die Aufmerksamkeit ...

Bildnachweise

- “Old Keys” by Jake Liefer, CC by 2.0 (modified by removing background)
- “Signing Paperwork” by Dan Moyle, CC by 2.0 (modified by removing background)
- “Software Freedom Days Singapore 09” by Ruiwen Chua, CC by-SA 2.0 (modified by removing background)
- “Code Bug” by Guilherme Tavares, CC by 2.0 (modified by adding alpha)
- “Tree Tutorial” by Katie Walker, CC by-SA 2.0 (modified by removing background)
- “Manufactured security” by Kris Krüg, CC by-SA 2.0 (modified by adding alpha)