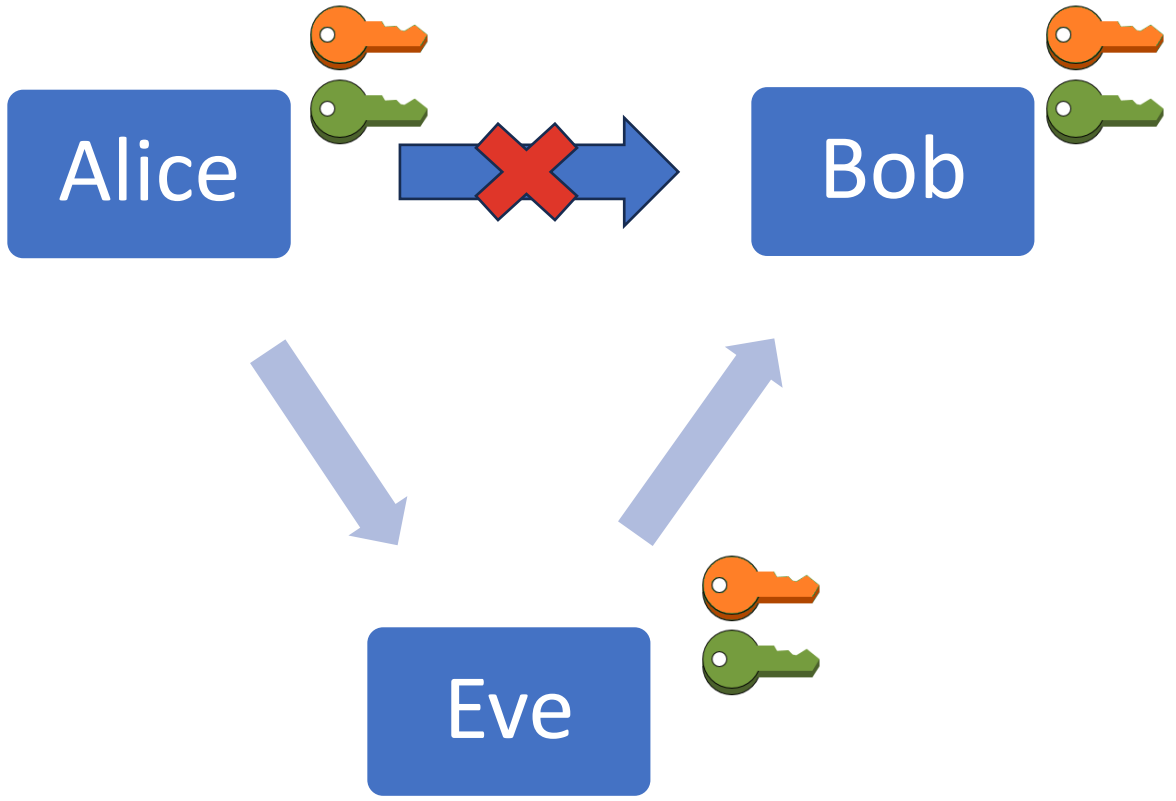
IT Sicherheit

01 PKI – Public Key Infrastructure

Einführung / Probleme mit asymmetrischer Verschlüsselung

- Asymmetrische Verschlüsselung: Public Key / Private Key

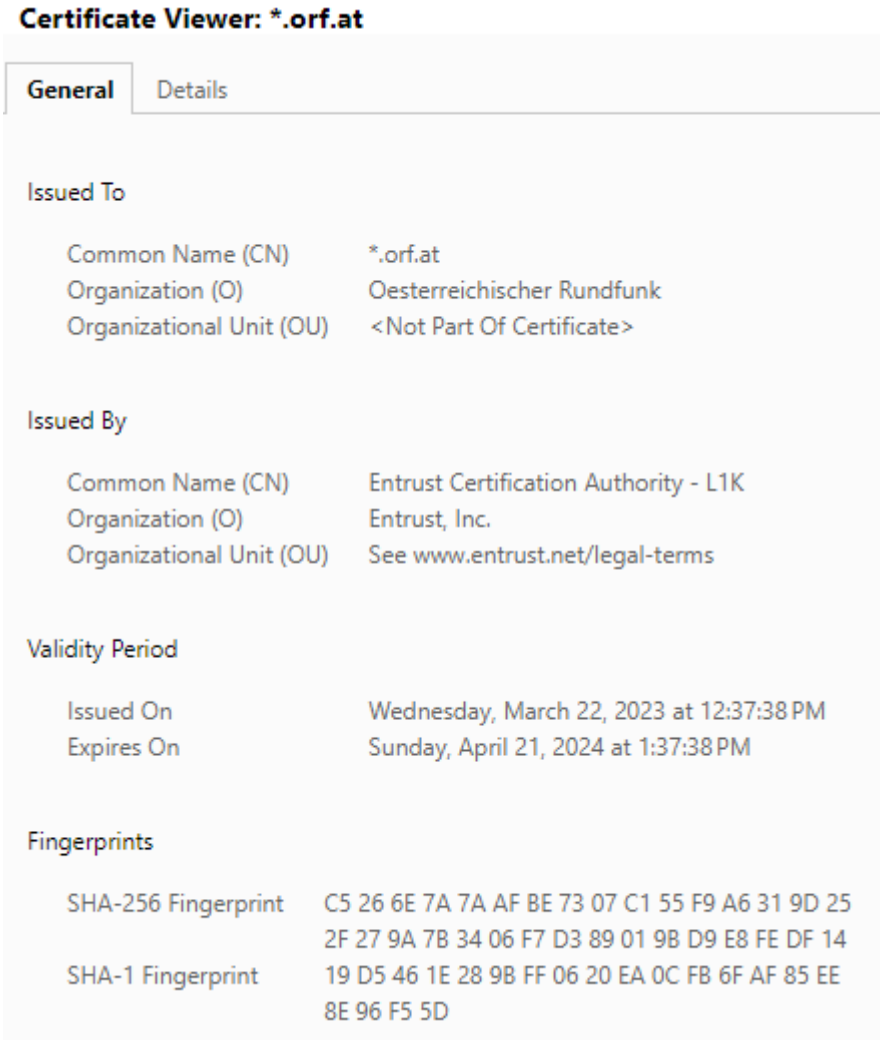
- Problem: Wie bekommt Kommunikationspartner B den Public Key des Kommunikationspartners A?



Einführung / Probleme mit asymmetrischer Verschlüsselung

- Lösung: Zertifikate

- Zertifikat: Digitaler Datensatz, mit dessen Hilfe bestimmte Eigenschaften bzw. Identitäten nachgewiesen werden können



Einführung / Probleme mit asymmetrischer Verschlüsselung

- PKI (Public-Key Infrastructure): Struktur zur Ausstellung, Verteilung und Prüfung von digitalen Zertifikaten

- Zwei gängige Varianten:
  - OpenPGP – Web of Trust
  - X.509 – Hirarchische Zertifizierungsstellen

OpenPGP – Web Of Trust

- Netz an Teilnehmern
- Teilnehmer erstellen Zertifikate selbst
- Teilnehmer teilen Zertifikate mit anderen Teilnehmern (z.B. Keyserver, Email, USB Stick, Messenger, …)
- Zertifikate werden von anderen Teilnehmern signiert
- Teilnehmer können selbst entscheiden, welchen Zertifikaten Sie vertrauen:
  - Mindestens X andere haben das Zertifikat signiert
  - Bestimmte andere Personen haben das Zertifikat signiert

OpenPGP – Web Of Trust

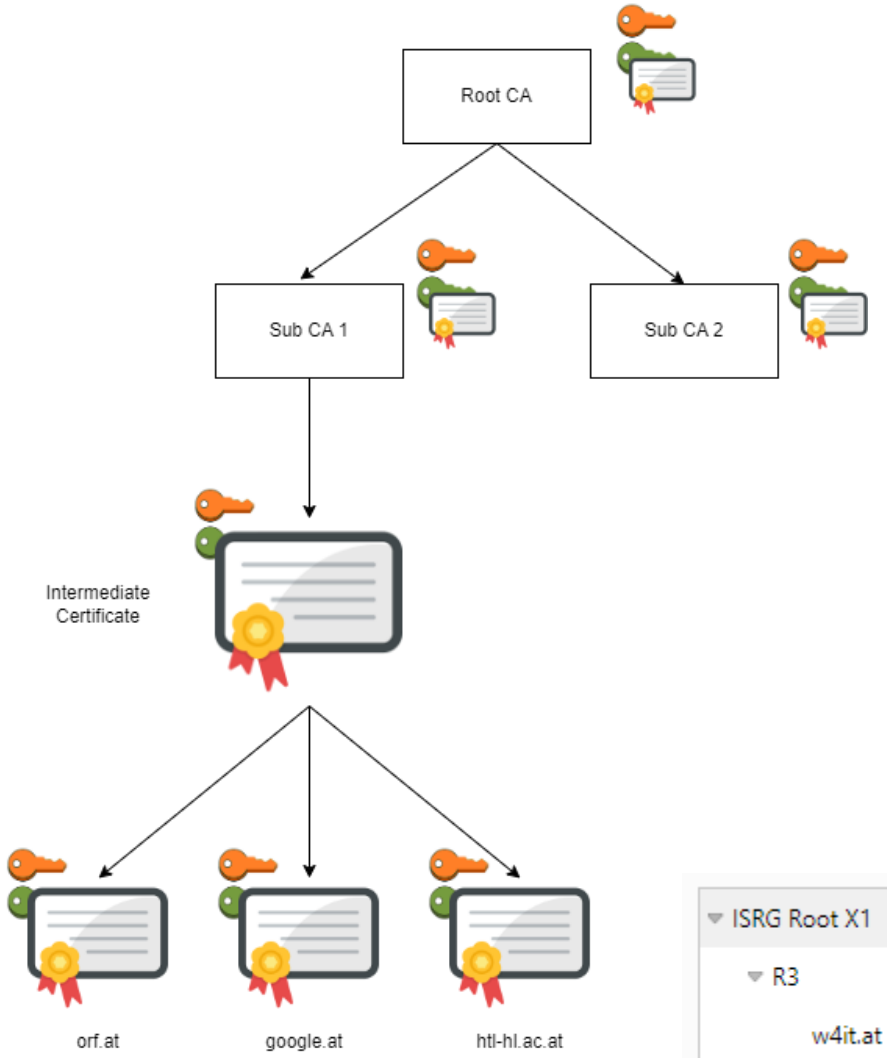
- Übung

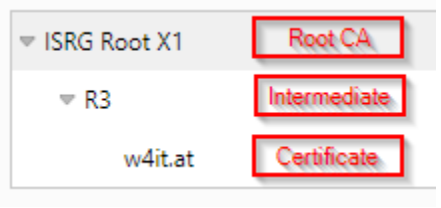
X.509 – Hierarchische Zertifizierungsstellen

- Streng hierarchisch aufgebaut – an der Spitze steht die Root-CA
- Unterschied zu “Web of Trust”: Nur eine CA signiert Zertifikate – nicht jeder
- Root-CA kann auch Sub-CAs ermöglichen – diese können dann auch Zertifikate signieren, welchen jeder vertraut der der Root-CA vertraut

X.509 – Hierarchische Zertifizierungsstellen

- Root CA
  - Private Key (stark geschützt – offline only)
  - Public Key und Zertifikat (wird von gängigen OS/Browsern vertraut)
- Sub CA 1/2
  - Private Key (stark geschützt – offline only
  - Public Key und Zertifikat (Optional: wird von gängigen OS/Browsern vertraut), wird von Root CA signiert
- Intermediate Certificate
  - Private Key: wird für Signierung von Zertifikaten genutzt
  - Public Key und Zertifikat
- Zertifikat
  - Private Key: Sollte von Administrator erzeugt werden (nicht von der CA)
  - Public Key und Zertifikat (wird von Intermediate Public Key signiert)
  - Damit Client dem Zertifikat vertraut, muss mit Zertifikat auch das Intermediate (und optional das Sub CA) Certificate ausgeliefert werden





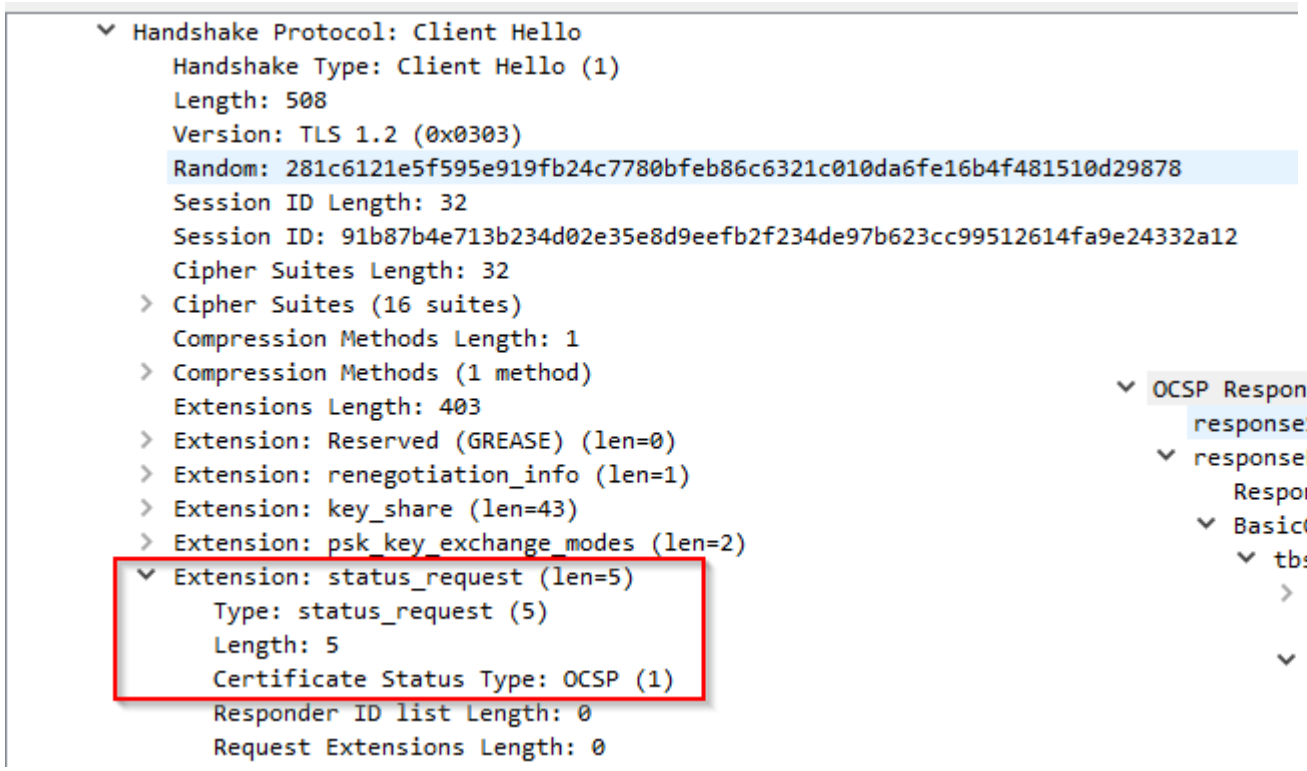
X.509 – Hierarchische Zertifizierungsstellen

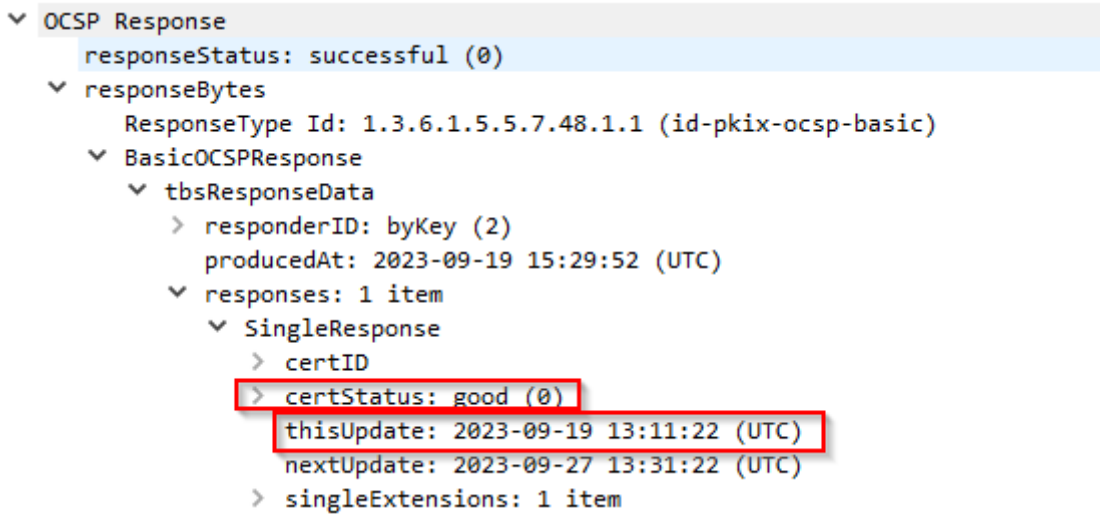
- Einmal signierte Zertifikate können nicht zurückgenommen warden – Lösung: CRL
- CRL gibt Auskunft über gesperrte Zertifikate – Sollte vor dem akzeptieren eines Zertifikats überprüft werden

- Weiterentwicklung: OCSP
- Online Service, welcher über den Status eines Zertifikats Auskunft gibt

- Weiterentwicklung: OCSP Stapling
- Zertifikatstatus wird vom Server in regelmäßigen Abständen bei der CA angefragt und dann bei Bedarf an den TLS Handshake angehängt

X.509 – Hierarchische Zertifizierungsstellen





X.509 – Hierarchische Zertifizierungsstellen

- Typische Bestandteile eines Zertifikats
  - Subject / Common Name: Domain/Email – mittlerweile eher unwichtig
  - Subject Alternative Name: Domains – sehr wichtig
  - Validity: Not Before, Not After – sehr wichtig
  - Issuer – informativ
  - Serial Number – wichtig zur Überprüfung, ob Zertifikat gefälscht wurde

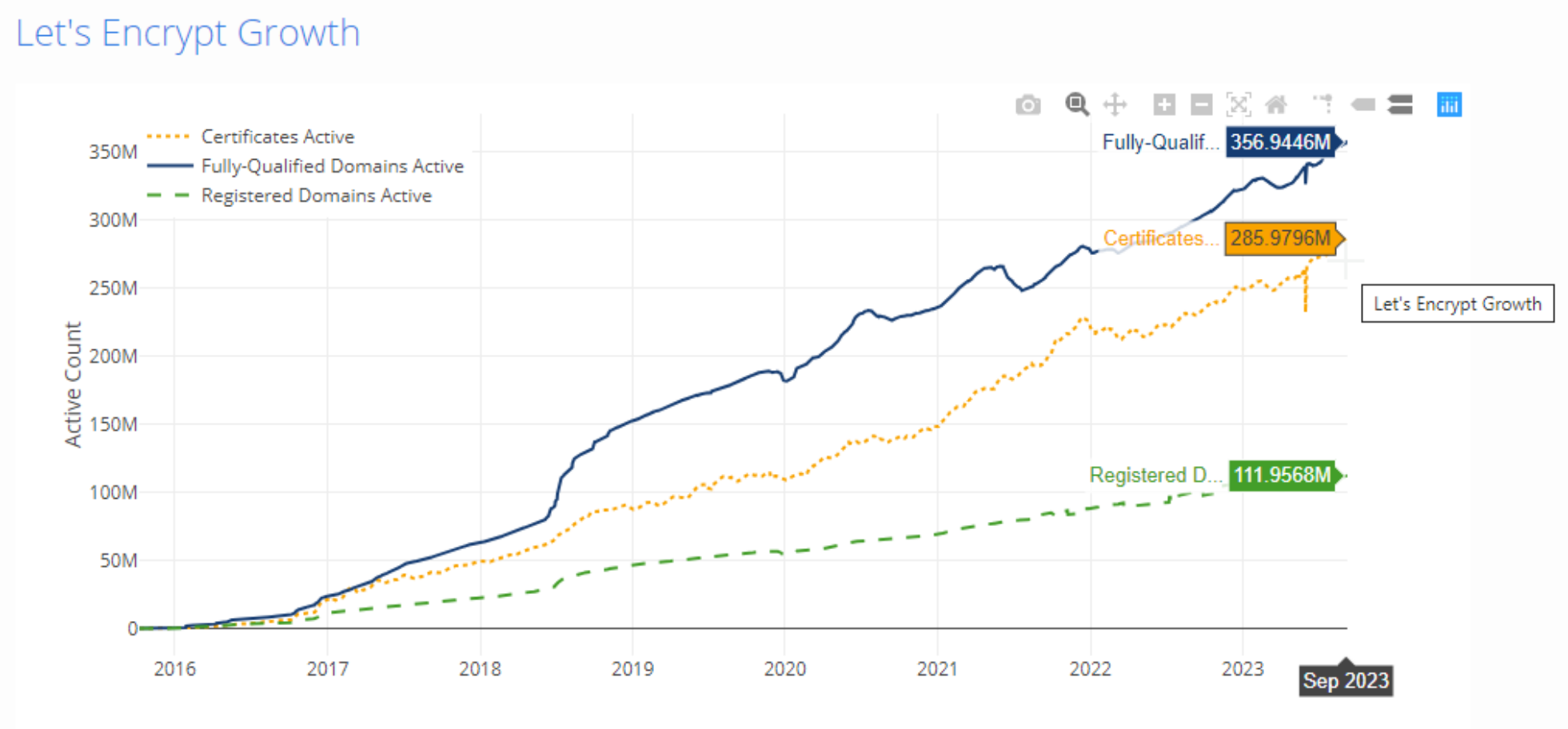
X.509 – Hierarchische Zertifizierungsstellen

- Übung

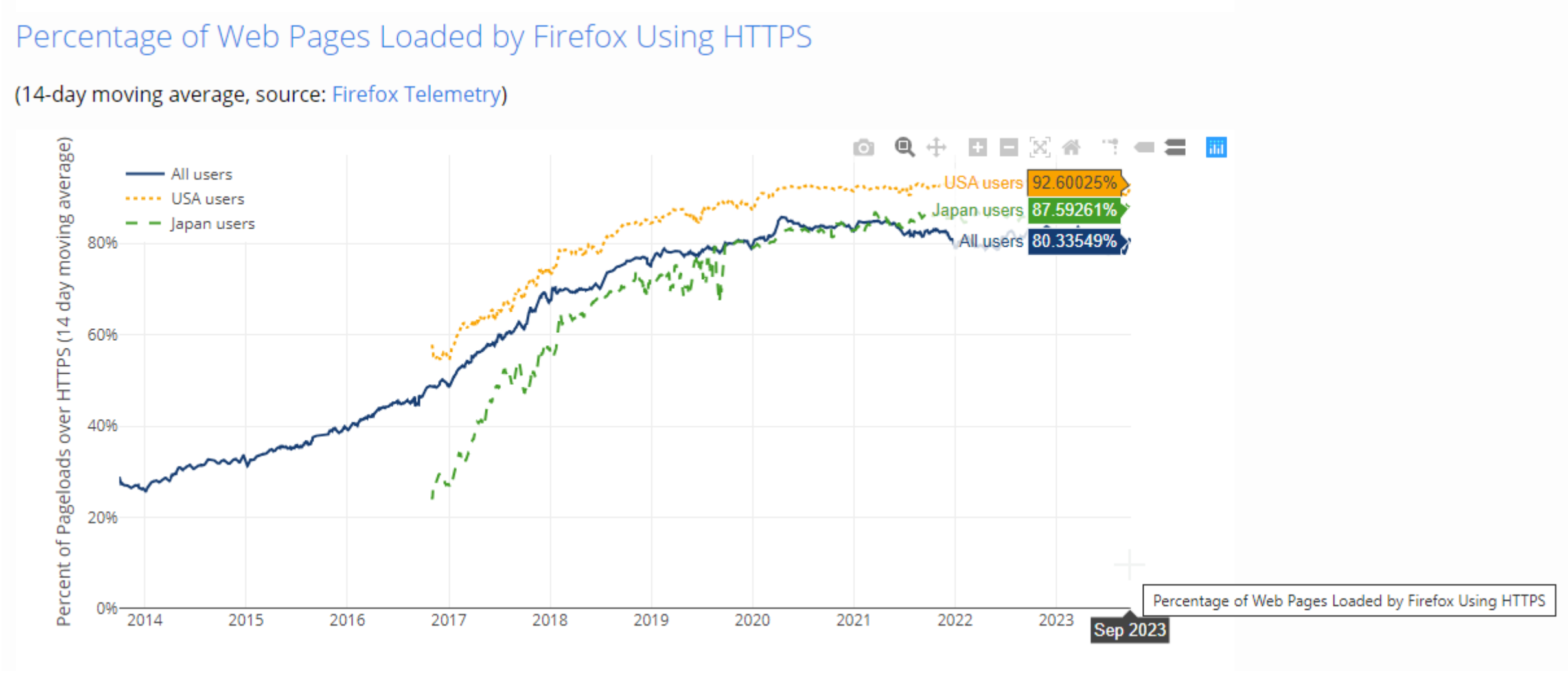
Freie CA - Let’s Encrypt

- Freie / Gemeinnützige CA aus den USA
- Aktiv seit 2014
- Ziel: HTTPS Verbindungen zum Standard machen
- Gültigkeit der Zertifikate: 90 Tage
  - Reduziert den Schaden bei Kompromittierung des Zertifikats / Keypairs
  - Motiviert Anwender den Prozess der Zertifikatsaustellung zu automatisieren
- Größte CA nach Anzahl Domains / ausgestellten Zertifikaten

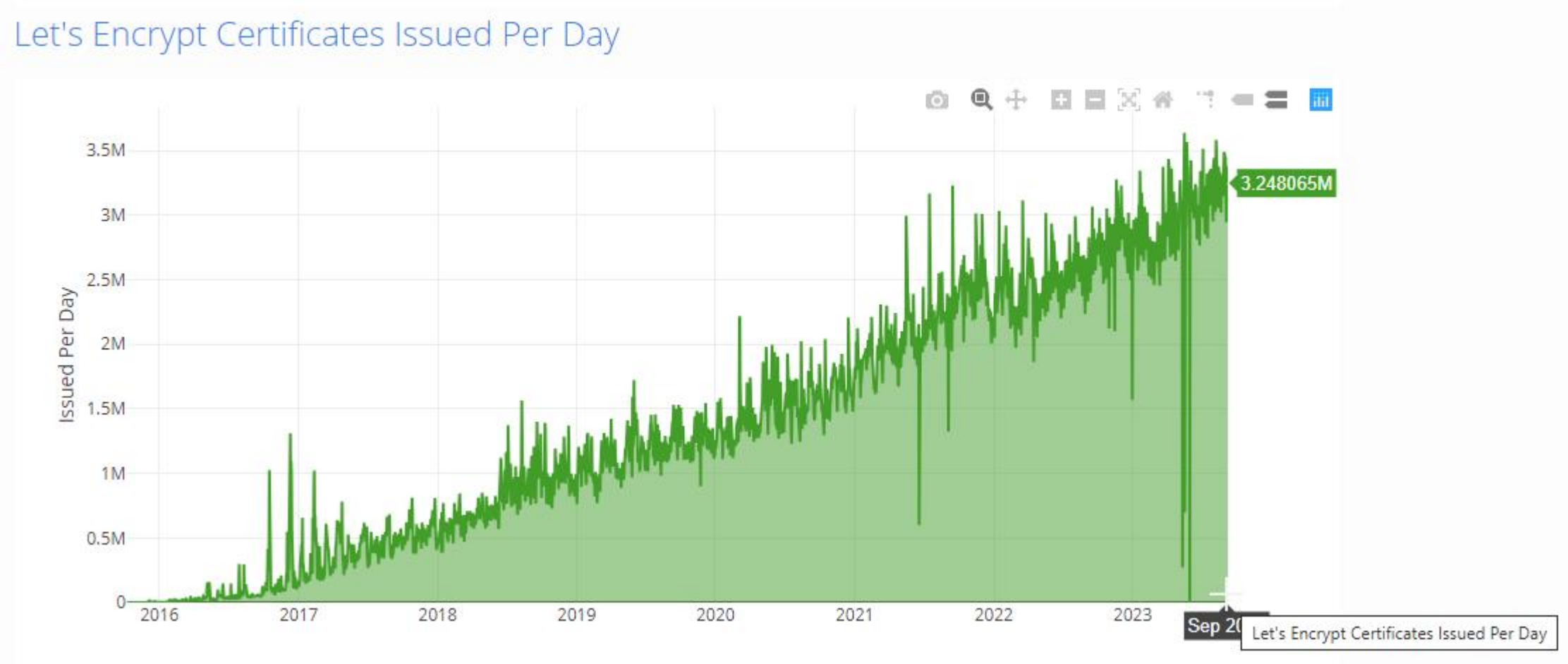
Freie CA - Let’s Encrypt



Freie CA - Let’s Encrypt



Freie CA - Let’s Encrypt



Freie CA - Let’s Encrypt

- Protokoll zur Ausstellung von Zertifikaten: ACME
  - Aktuelle Version: v2
  - Mögliche Zertifikate:
    - Domain Validated Certificates
    - Domain Validated Wildcard Certificates
  - Beschreibt die Kommunikation zwischen den Servern der CA und den Servern der Anwender
    - Basis: Austausch von JSON Nachrichten über HTTPS
    - Basis sind so genannte Challenges:
      - HTTP-01 challenge
      - DNS-01 challenge
      - TLS-ALPN-01 challenge

Freie CA - Let’s Encrypt

- HTTP-01 challenge
  - Anfrage an Let’s Encrypt ACME API
  - Let’s Encrypt ACME API liefert einen Token zurück
  - Url muss via Port 80 erreichbar sein (Datei mit Token + Fingerprint von Account-Key):

    `http://<YOUR_DOMAIN>/.well-known/acme-challenge/<TOKEN>`
  - Rückmeldung an Let’s Encrypt API
  - Let’s Encrypt überprüft die Verfügbarkeit von mehreren Standorten weltweit
  - Zertifikat wird ausgestellt
  - Pro: Einfach zu automatisieren/Ohne Zugriff auf DNS verwendbar
  - Contra: Keine Wildcard Zertifikate / Port 80 erforderlich / Schwieriger bei lastverteilten Webservern

Freie CA - Let’s Encrypt

- DNS-01 challenge
  - Anfrage an Let’s Encrypt ACME API
  - Let’s Encrypt ACME API liefert einen Token zurück
  - TXT Eintrag für `_acme-challenge.<YOUR_DOMAIN>` mit dem Token als Wert muss angelegt warden
  - Rückmeldung an Let’s Encrypt API
  - Let’s Encrypt überprüft die Verfügbarkeit von mehreren Standorten weltweit
  - Zertifikat wird ausgestellt
  - Pro: Ermöglicht Wildcard Zertifikate / Einfacher bei lastverteilten Webservern
  - Contra: Nur Sinnvoll, wenn DNS Anbieter eine API anbietet und Aktualisierungen schnell genug ausgerollt werden

Freie CA - Let’s Encrypt

- TLS-ALPN-01 challenge
  - Nachfolger der TLS-SNI-01 challenge
  - Operiert auf Port 443 im Zuge des TLS Handhakes
  - Aktuell nicht stark verbreitet
  - Pro: Kann verwendet werden, wenn Port 80 nicht verfügbar ist
  - Contra: Aktuell nicht mit Apache/Nginx/Certbot unterstützt / Keine Wildcard Zertifikate möglich

Freie CA - Let’s Encrypt

- Übung

Certificate Transparency Logs

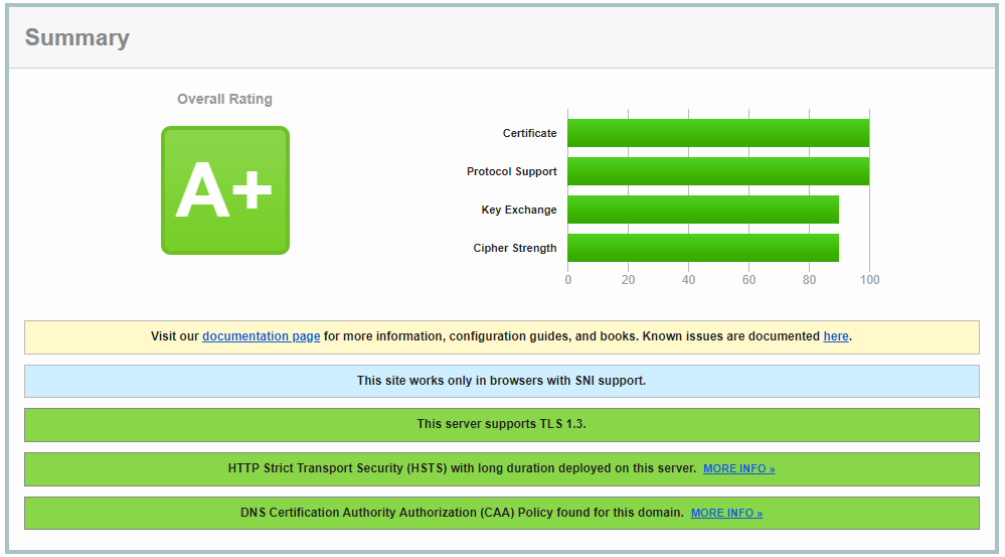
- CAs können bzw. müssen Informationen zu ausgestellten Zertifikaten in CTL ablegen
- CTLs warden von verschiedensten Anbietern bereitgestellt (Letsencrypt, Cloudflare, Google, …)
- Browser prüfen, ob Zertifikat in genügend “passed” CTLs hinterlegt sind (Certificate Transparency Policy)
  - Google Chrome
    - Gültigkeitsdauer < 180Tage: Zertifikat muss in min. 2 CTL vorhanden sein
    - Gültigkeitsdauer > 180Tage: Zertifikat muss in min. 3 CTL vorhanden sein

Certificate Transparency Logs

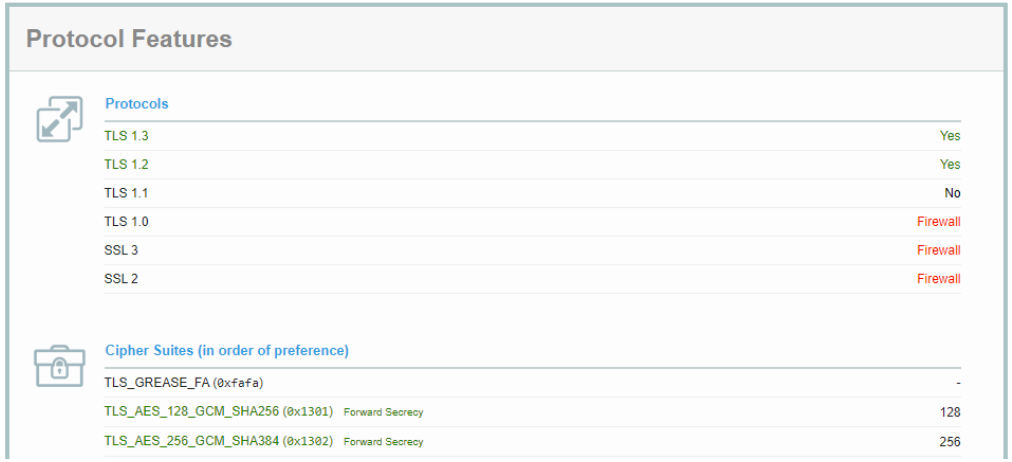
- CTL manuell überprüfen
  - https://crt.sh/
  - https://sslmate.com/labs/ct_policy_analyzer/
  - https://ct.cloudflare.com/

Qualys SSL Server/Client Test

- https://www.ssllabs.com/ssltest/
- Einfache Überprüfung der Konfiguration eines Webservers
- SNI / HSTS / CAA



- https://clienttest.ssllabs.com:8443/ssltest/viewMyClient.html
- Einfache Überprüfung des verwendeten Browsers



Qualys SSL Server/Client Test

- Demo