Netzwerkadministration

Baumi Kram

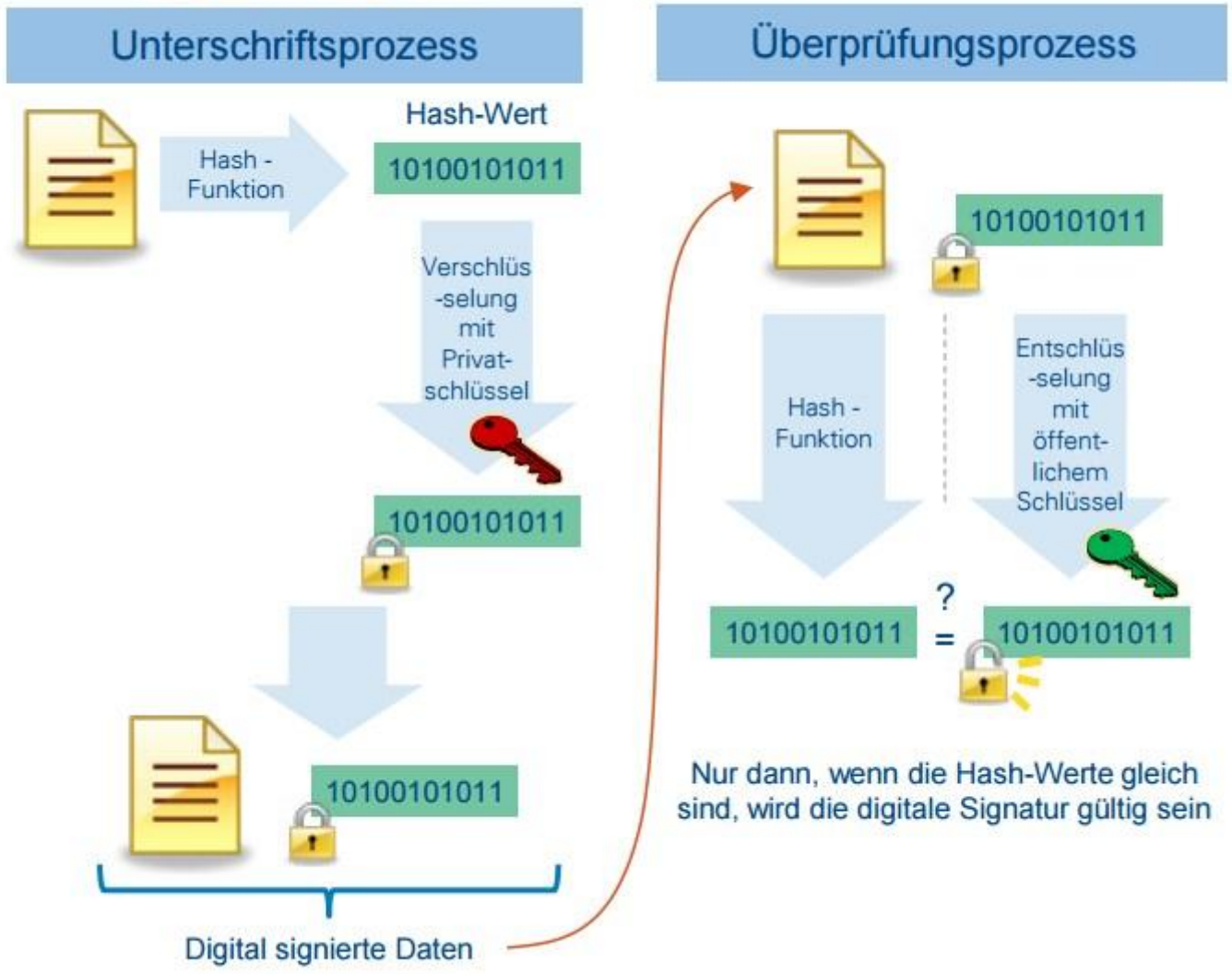
- Digitale Signatur
- Digitales Zertifikat
- Let's Encrypt / Certificate Authorities
- Klausur Themen
- Hashcat

Digitale Signatur

Durch eine Digitale / Elektronische Signatur können zwei der vier Anforderungen an die Kryptografie erfüllt werden.
Dazu wird sich die asymmetrische Kryptografie zu Nutze gemacht.
Die Digitale Signatur erfüllt die:

- Integrität
- Verbindlichkeit

Bild 1: Der Unterschrifts- und Verifikationsprozess der Digitalen Signatur



Quelle: Arthur D. Little

Für digitale Signatur braucht, muss man ein Schlüsselpärchen generieren.

Das Signieren passiert in 3 schritten

1. Hashen der Datei
2. Verschlüsseln des Hashes mit Private Key
3. Versenden der Datei mit dem verschlüsselten Hash

Zum Prüfen der Signatur braucht der Empfänger deinen Public Key. Die Authentität dieses muss durch manuellen Abgleich des Fingerabdrucks (Hash) geprüft werden.

Zum Prüfen wird die Signatur mit dem Public Key entschlüsselt, dadurch erhalt man den Hash. Dann wird die ursprüngliche Datei gehashed und die beiden Hashes verglichen.
Wenn beide gleich sind, ist die Signatur korrekt.

Digitales Zertifikat

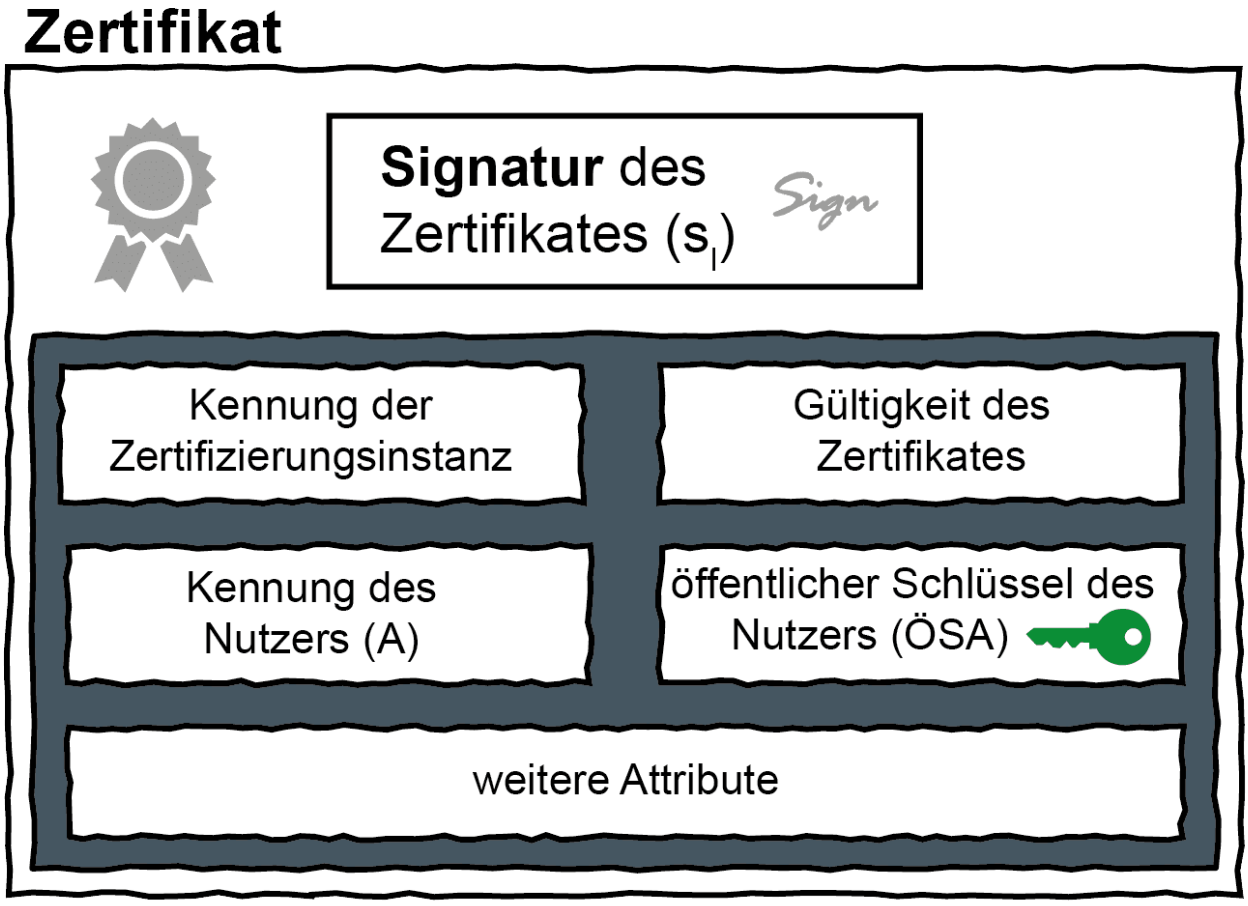
Was ist ein Zertifikat?

Ein digitales Zertifikat (auch Zertifikat oder Public-Key-Zertifikat) dient als Echtheitsnachweis für eine Website, Mailserver oder Signatur.

Durch ein Zertifikat kann geprüft werden, ob der Inhaber einer Webseite oder Versender einer E-Mail auch die Rechte hat diese Webseite / E-Mail-Adresse zu betreiben und gewährleistet somit auch die Vertraulichkeit und Authentizität.

Da im Zertifikat der Public Key des Hosts enthalten ist, kann durch ein Zertifikat auch verschlüsselte Kommunikation stattfinden.

Im Zertifikat sind einige Verwaltungsinformationen enthalten, wie Name des Inhabers, Name der CA (Certificate Authority), Gültigkeit des Zertifikats, Public Key des Inhabers, Seriennummer des Zertifikats, digitale Signatur der CA.



Zweck von Zertifikaten:

Die Zertifikate dienen dem Zweck, einen Identitätsnachweis des Inhabers einer Webseite von einer vertrauenswürdigen Stelle zu haben, auf die man sich verlassen kann.

Funktionsweise:

Der Inhaber erstellt zuerst ein Schlüsselpaar.

Mit dem Öffentlichen Schlüssel und weiteren Verwaltungsinformationen beantragt man ein Zertifikat bei einer CA.

Die CA signiert ein Zertifikat und stellt es für eure Domain aus, nachdem Sie bestätigt hat, das ihr auch der tatsächliche Inhaber der Domain seid. (Dazu wird einfach die Domain aufgelöst und die IP mit der des Antragstellers verglichen)

Das Zertifikat wird über eine "Site" im Webserver hinterlegt.

Beim Aufruf wird das Zertifikat geprüft und die Signatur der CA kann über die bereits bei Installation hinterlegten Public Keys der Root-CAs geprüft werden.

Über den im Zertifikat hinterlegten Public Key kann dann auch eine verschlüsselte Verbindung mit dem Webserver hergestellt werden.

Erstellen eines Selbst-Signierten Zertifikats:

Grundsätzlicher Ablauf:

1. Eigenen FQDN (Fully Qualified Domain Name) ermitteln
2. Apache2 installieren und SSL Module aktivieren -> Port 80 und 443 sollten jetzt offen sein
3. Schlüsselpaar generieren / Zertifikat erstellen
4. Virtuellen Server (VirtualHost) konfigurieren
5. Testen und Zertifikat prüfen

```
#1 FQDN (Fully Qualified Domain Name) herausfinden
ip -c a #x.x.x.x
host x.x.x.x #host.domain.lokal

#2 Apache2 installieren
apt install apache2 -y
lsof -i -P #Port 80 muss jz offen sein
#Im Browser http://localhost -> geht
#           https://localhost -> nope

#3 SSL Modul aktivieren:
a2enmod ssl
```

```
systemctl restart apache2

lsof -i -P #Port 443 jz offen
#Im Browser https://localhost -> Fehler "sichere" Verbindung fehlgeschlagen

#4. Schlüsselpaar generieren
apt install openssl -y
mkdir /etc/apache2/ssl
cd /etc/apache2/ssl
openssl req -new -x509 -days 365 -nodes -out apache.pem -keyout apache.pem #Hier Als Common
Name den FQDN eintragen
chmod 600 apache.pem

#Zertifikat prüfen
openssl x509 -text -noout -in apache.pem

#5. Virtuellen Server konfigurieren
cd /etc/apache2/sites-available
cat << EOF > ssl.conf
<VirtualHost *:443>
ServerName [FQDN]
SSLEngine ON
SSLCertificateFile /etc/apache2/ssl/apache.pem
DocumentRoot /var/www/html
</VirtualHost>
EOF
# [FQDN] ersetzen

a2ensite ssl
systemctl restart apache2
```

Am Ende kommt im Browser immer noch eine Sicherheitsabfrage, weil das Zertifikat selbst signiert wurde und der Browser somit die Signatur des Zertifikats nicht prüfen kann.

Let's Encrypt / Certificate Authorities

Was ist eine CA?

Eine CA ist eine Organisation oder ein Unternehmen, welche/s berechtigt ist, allgemein anerkannte digitale Zertifikate auszustellen.

Die CAs müssen sehr hohe Sicherheitsstandards erfüllen, um als CA anerkannt zu werden.

Wenn eine CA einmal geprüft und anerkannt wurde, wird ihr Public Key in den Zertifikatstores der Browser eingebunden. Dadurch kann die Integrität der Zertifikate geprüft werden, da die Signatur des Ausstellers, also der CA, geprüft werden kann.

Was ist Let's Encrypt?

Let's Encrypt ist eine CA, unter der Root-CA "ISRG Root X1" und "ISRG Root X2". Die "ISRG" "Internet Security Research Group" ist eine gemeinnützige Organisation, welche durch viele große Tech-Unternehmen wie Google, Mozilla, AWS, EFF, Cisco, SAP und viele mehr finanziert wird.

Let's Encrypt ist eine CA, bei der kostenlos Zertifikate beantragt werden können.

Zertifikate mit Let's Encrypt erstellen:

```
apt install python3-certbot-apache
certbot --apache
```

Klausur Themen

1. Apache mit TLS
2. Theorie: Digitale Signatur, Zertifikate
3. Hashcracking mit Hashcat -> Selbst einarbeiten vorher!!

Hashcat

Basic Befehle und Syntax

hashcat -m500 -a3 /etc/shadow '?l?l?l?l?l?l' (--show)

-m hash-type:

- m0 md5
- m100 sha1
- m500 md5crypt (unix) $1
- m7400 sha256crypt (unix) $5
- m1800 sha512crypt (unix) $6

-a attack-mode:

- a0 Straight (Wörterbuch angriff)
- a3 Bruteforce
- Rest unrelevant

--username zum überspringen des username blocks, für /etc/shadow/ empfohlen

Output File

Dictionary File (nur bei a0)

Bei Angegebener struktur:

- ?d eine Ziffer
- ?l lowercase Letter
- ?u uppercase Letter
- ?a Beliebiges zeichen (kann auch bei unsicherer wortlänge benutzt werden)

--show zeigt das ergebnis an, aber nur wenn der hash schon einmal gecracked wurde

Ein Standardwörterbuch kann man hierüber importieren:

```
wget https://github.com/brannondorsey/naive-hashcat/releases/download/data/rockyou.txt -O
/tmp/rockyou.txt
```