

HPS / GWDG



Jonathan Decker

## **Certificates and PKI**

Simple in Principle - Complicated in Praxis

#### Table of contents

- 1 What are Certificates and PKI
- 2 Terminology
- 3 Certificates in the Wild
- 4 Certificate File Formats
- 5 Public Key Infrastructure

#### Overview

- We use complex math to build a cryptographic system
- Enables TLS (and HTTPS)
- Universally applicable in networks
- Identify devices and code
- Build on top of public-private-key cryptography
- Simple concept, complicated standards

## Public-Private-Key Cryptography

- Also called Asymmetric Cryptography
- Use math to generate key pairs
- Encrypt with one
  - Decrypt with the other
- Encrypted messages give no hint on the key itself
- Public key can be shared
- Private key must stay private
- Use someone's public on a message
  - It can only be read with the private key

## Notes on Asymmetric Cryptography

#### It is slow compared to symmetric crypto

- It makes sense if you see the math
- Commonly used to establish a shared secret between client and server
  - Shared secret enables symmetric cryptograhpy
- Verifying the receiver across a network is taken for granted nowadays
- Thanks number theory

Public Key Infrastructure

## Who makes the Terminology - The IETF

#### IETF (Internet Engineering Task Force)

Founded in 1986

- Develop voluntary standards
- Publishes technical documents RFCs
- RFC (Request For Comments)
  - Historical acronym now RFC means RFC

#### RFC 4949 - 1/3

#### Entity

- "An active part of a system a person, a set of persons (e.g., some kind of organization), an automated process, or a set of processes that has a specific set of capabilities."
- Anything that exists logically or conceptually
  - Your computer, your code, you yourself
- Can be a server, software package, email contact

#### Identity

- "The collective aspect of a set of attribute values (i.e., a set of characteristics) by which a system user or other system entity is recognizable or known."
- Do not confuse with Identifier
- Identifier is a unique key representing an identity

### RFC 4949 - 2/3

#### Claim

- An entity may declare an attribute
- "My name is Max Mustermann"
- Another entity may Authenticate a claim

#### Authentication

- Process of confirming the truth of a claim
- When you login, you claim to be a user
  - And verify the claim by giving your password

#### End Entity or End User

"A system entity that is the subject of a public-key certificate and that is using, or is permitted and able to use, the matching private key only for purposes other than signing a digital certificate; i.e., an entity that is not a CA."

#### RFC 4949 - 3/3

#### Certificate Authority (CA)

An entity that issues certificates to subscribers

#### Subscriber

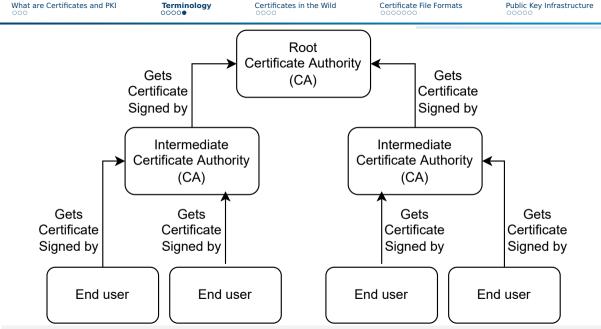
"A user that is registered in a PKI and, therefore, can be named in the "subject" field of a certificate issued by a CA in that PKI."

#### Public-Key Certificate

- "A digital certificate that binds a system entity's identifier to a public key value, and possibly to additional, secondary data items."
- A public and metadata (hostname, expiry date, etc.)

#### Root Certificate

- "A certificate for which the subject is a root."
- "The self-signed public-key certificate at the top of a certification hierarchy."



Jonathan Decker

What are Certificates and PKI	Terminology	Certificates in the Wild ●○○○	Certificate File Formats	Public Key Infrastructure

#### The gwdg.de Certificate

#### shell

# echo | openssl s\_client -connect gwdg.de:443 2>/dev/null | \ openssl x509 --text

Public Key Infrastructure

## The gwdg.de Certificate

Find from top to bottom:

- Unique Serial Number
- Issuer metadata
- Validity range
- Subject metadata
  - Including encoded public key
- X509v3 extensions
  - Extended Key Usage
  - Basic Constraints
  - CRL Distribution Points
  - Authority Information Access
  - CT Precertificate SCTs
  - Subject Alternative Name
- Signature Algorithm

e echo	1 spensel subtlevi -connect gody.der443 2v2dev2md1 ( ) stddtext
Gertifi	
	Hersion: 3 (4e2) Serial Mander:
	6114a1821501981661e21b718517e1d51d81FE162168180
ar er	— 05 for (2:16)-09-06-12:167-165 - New 55-000 FTF-067-08-100 Vipandura Jipan Jipan Jipan Hamilton Arytino Tanwer: C = GA, ST = Graviar Machaster, L = Salfard, 8 = Sectige Limited, OI = Sectiga MSA Organization Validation Secur
	nanosis Baforer (m. 2. 2004)00 2023 000 Bat Affer (m. 44. 2.23)000 2020 000 Sobjett C. 2 R. 57 = Niedersachsen, 0 = Gentlickoft fVCJMCr wissenschaftliche Dotenverscheitung mbH, Di = polg.de Sobjett TALLis Ry IFGG
	Subject: C = 00, ST = Niedersachsen, O = Gesellschaft fic3UDCr wissenscheftliche Onterverarbeitung mbH, OI = gelg.de Bubject Public Key Info:
	Public Rey Algertine: Fielderrygtion Mile Public-Devi (1972) 6511
	Hodylass
	e2rc2redt52r72r78rf8r6f158x154t52r82r78rf8r6f158x154t55r6ct
	14+59+28+72+1e++++19+45+81+81+78+42+48+ 68+61+65+45+44+19++10+19+42+10+19+42+68+1
	2 million 2 million 7 Million (Million 2 2 million 2 1 million 4 m
	an i Millicho (Milli Millicho) (2017) Millio (Millio (2012)) An (Millio) an i Anto (Millio) Colomba (1910) An (Anto Anto Anto Anto Anto Anto Anto Anto
	97 (cb 84) 29(104) 041 041 041 041 000 000 000 000 041 041
	48 (18) (Fill) Fill (18) (46) (18) (18) (28) (28) (Fill (22) (Fill (22) (Fill)) 45 (18) (45 (18) (16) (26) (26) (26) (26) (26) (26) (26) (2
	Bit and the first of a control of the first of an end of the first
	00153120010/70281106304106101 Exponenti 05337 10120003
	NEMPLY ANTINITION / May Identifiant Nordel Totel Revise (Internet (Internet) Navid Totel Revise (Internet) (Internet) (Internet) (Internet) (Internet)
	SMBH ANJEL WE SHAFTERE AL-SHOT SHAFTERE SHAFTERE SHAFTERE SHAFTERE SHAFTERE SHAFTERE SHAFTERE SHAFTERE SHAFTERE SHAFTERE SHAFTERE SHAFTERE SHAFTERE SHAFTERE SHAFTERE
	91gttal Signature, Kay Enclyberment 3309v3 Basic Canatratits: critical
	33990-3 Ernement Ray Damper Tha Web Ammer Arabititation, Tax Web (liset Authoritation 33990-3 Certificate Polyona) Polyon (Jack) Laka Laka Pata
	Policy 13.2.6.16.3.668.12.1.2.4 (25): 53.16.1/am/140.000/25
	313983 CARTCOLOUR LANGETUND 1000 CARTCOLOUR LANGETUND 100 CARTCOLOUR 2020 CARTCOLOUR CARTON CART 100 CARTCOLOUR 2020 CARTON CARTON CARTON CARTON 100 CARTON CARTON CARTON CARTON CARTON CARTON CARTON CARTON CARTON 100 CARTON CARTON CARTON CARTON CARTON CARTON CARTON 100 CARTON CARTON CARTON CARTON CARTON CARTON CARTON 100 CARTON CARTON CARTON CARTON CARTON CARTON CARTON CARTON 100 CARTON CARTON CARTON CARTON CARTON CARTON CARTON CARTON 100 CARTON CA
	04744701887297001887897801801901901001001001001041 0474470188729700188142768439718818714015645845841
	Supartice 1 and a section 444300 Supartice 2 and 2 and 2 and 2 and 4 and 2 and 3 and 4 and 4 and 4 and memory and 4 and a section 4 and 4 and a section 4 and 4 and a section 4 and 4 and 1 and 4 and 1 and 4 and 1 and 4 and 1 and 4 a
	F0:54:737#E:55:FE:25 Signed Certificate Timestamp:
	Veritor 1 v2 (468) Leg ID - 04-04-07-08-37-85-06-22-97-85-06-22-97-85-06-28-38- 93 77-06-083 73-06-083-34-083-34-083-09-98-07-98 (48
Γ.,	91/71/66/88/51/84/85/24/88/24/20/20/88/07/F8/A8 Tungstamp - Feb 2: 87/27/50.165 2023 0FT
Γ.,	
	STOPPOTETE T POCUSENY (19-54A356 DECALOR: 201381.001081.001081.001081.001081.001081.001081.001081.001081.001081.001081.001081.001081.001081.0010
	5 (2017) 11 (1) (1) (1) (1) (2) (1) (2) (1) (2) (2) (2) (2) (2) (2) (2) (2) (2) (2
	011A714314611531801461454144188169131131184461131 8013814814531801851818441581481
Γ.	
Γ	Version v V (1868) Leg ID i Bircholenies/Birch/ClirCS/SC182180184/CDIISTA21 331-672-445-721-86-722-692-632-934-939-46-721-17-051-98
	1.5pn.11mm - 1.40.41m-45122130330 301-851422121100130344103135168103130147717 2014273977143440434813130179197218214138814814981921

	What are Certificates and PKI	Terminology	<b>Certificates in the Wild</b> ○○●○	Certificate File Formats	Public Key Infrastructure
--	-------------------------------	-------------	---	--------------------------	---------------------------

#### The hps.vi4io.org Certificate

#### shell

# echo | openssl s\_client -connect hps.vi4io.org:443 2>/dev/null|\ openssl x509 --text

Public Key Infrastructure

## The hps.vi4io.org Certificate

Find differences top to bottom:

- Issuer metadata
- Validity range
- Subject metadata
- X509v3 extensions
  - CRL Distribution Points
  - Authority Information Access
  - CT Precertificate SCTs
  - Subject Alternative Name

#### At the end, you will understand most fields

<ul> <li>cole   openail a_client -connect hps.wi4to.org/443 2&gt;<u>/dev/mdl</u>   \ series] ab80 r=test</li> </ul>
Certificate:
Both ( Arthough a Charles )
Writer 1 (8:2) Serial Number:
84:20:54:45:certe:u0:0027154:17:180:75:540:05:54 Signature Algorithm: shol544:146AEccryption Isiawari C = 45,
Issuer: C = US, C = LAC's Socrypt, CH = R3
Velifity Bod mefore: Peb 11 22:00:00 2023 UPT
Bod Burfare: Palo 11 22/04/158 2023 UHT Bod After = Nay 12 22/04/155 2023 UHT Budjet: UH = Noi-Ville.org
Subject: DR = Aprivilia.arg Subject Public Key Infa:
Subject Public Kep Infa: Public Rey Algorithm: rodencrystian Kin Public Rey (1940 011)
Tendo Com 10 (1997) - 10 (199
La contra con con contra contra Contra contra cont
5e-55-63-58-17-85-16-13-74-68-73-12-66-62-24-
041-051-051-051-051-051-051-051-051-051-05
221561-0419015-16-080172128151-0214755-081-041
0510010310510513013010010011110010017110130100170101 0210001101011110101001001001001701200000101101
22 - 26 (dots all of respective), the limit of the limit of the respective (1) - 26 - 26 (dots all of respective), the limit of the l
311-31-314512441-100-101154144-1472-145724-044-091
accels 24 at 124 bit 50 at 122 cd 131 ad accel 122 cl
and the state of t
and constraints and a second second and a second se
Exposet: 65537 (0x30001)
NUMER and examine 1899/D-Reg Vanger critical 2899/D-Created Ang Heady Th We know Anthenization, Thi Nek flant Anthenication This We have controllering 2890/D-Reg Controllering critical
Digital Signature, Koy Kacipherment
KSEPV3 Extended Key Esege: TEX Web Berver Asthentication, TEX Web Client Asthentication
8509v3 Basic Constraints: critical
CRIFALSH RSMDV3 Subject Way Identifian
TF 104 (2) 000 (2000) (2) 103 (2000) 1 (40) (2) 29 (20) (20) (20) (20) (20) (20)
La constanti 1999 - Sono Constanti a Seno Constanti 1999 - Seno Constanti a Constanti a Constanti a Constanti a Constanti 1999 - Seno Constanti a Constanti a Constanti a Constanti a Constanti Negativa a Constanti a Constanti a Constanti a Constanti a Constanti a Constanti Negativa a Constanti a Constanti a Constanti a Constanti a Constanti a Constanti Negativa a Constanti a Constanti a Constanti a Constanti a Constanti a Constanti Negativa a Constanti a Constanti a Constanti a Constanti a Constanti a Constanti Negativa a Constanti a Constanti a Constanti a Constanti a Constanti a Constanti Negativa a Constanti a Constanti a Constanti a Constanti a Constanti a Constanti Negativa a Constanti a Constanti Negativa a Constanti a Constanti Negativa a Constanti a Constanti Negativa a Constanti
Rotherity Information Access: ECSP - Amelistari/Vele.inscr.arg CElainer - AMElitari/Vele.inscr.arg/
ASABY3 Subject Alternative Name:
1941/561/14/10.070 HSPV/ Centificate Pulicies
Patity 2.2.2.100.11.2.1 Patity 1.2.5.1.61.1400.1.1.1.1 (Patity 1.2.5.1.61.1400.1.1.1.1 (Patity 1.2.5.1.61.1400.1.1.1.1
CPS: http://sps.latsencrypt.org
ET Presentificate SCI.
(1 Protect (1 Goods (55)) 59 and Cert (1 Goods Thank Lang) 59 and 50 (1 H Goods Thank Lang) 59 and 50 (1 H Goods (1 H
144 E8 1 R7 (80 (80 (94 (94 (94 (94 (94 (94 (94 (94 (94 (94
50+PC+42+CP+74+9E+13+C4+9E+12+69+81+25+E0+04+99
Eligitational L. Acceleration-Melline Texastrophysical and texastrophysical and texastrophysical and texastrophysical and texastrophysical and texastroph
Vignatura: 1 (2000-2011) (2013) (2
AA1181150F060F041002730F7XF1064F53F002351F01146566555 805080F045F085F087F047F087F085F085F085F085F085F085F085F085F085F085
Signed Certificate Timutampi Version 1 al (Aud)
50000001111111111111111111111111111111
#31#31/08/09/111/08/00/1117/19160/06/00/06/00/06
Extent Land   Porte
513/16CUTH - RC030-VELC9-SERV230 301461821201301721871761831961801801801801801801801
28+89:171-172+54+64+99:180+64+181:58+C8+72+71+28+C8+
5) (1971/07) 4 (4153-41/07) 1992/5 (1) 31 (4) (77 - 73) (1) (7 - 77) (7 - 73) (1) (7 - 77) (7 - 73) (1) (1) (1) (1) (1) (1) (1) (1) (1) (1
In PATER TO A
Standard Algorithm - daaladeettikkikovyn ton # Alexand 70034 de 1077-4086 och 100-000-000-000-000-000-000-000- etti 10071033 000 1000 - 000-000-000-000-000-000-000-
#a18077116315911a18519916a1561041651201831841641481
dan 81 oder 2er nam San 380 obd de2 ober 13 obd x83 o 31 obt 12 fob3 x88 o Str. Fernands tudf van 181 oan obdie 20 officielle 191 obsider 191 obsider 191
e4+6c+6b+27+66+3e+e8+Fa+ae+ee+e2+68+2c+2E+de+18+92+Fe+
f2:20:01:01:04:04:00:00:00:00:00:01:00:01:00:01:00:01:00:00
3216-20-3216(10017100-30-000000-00120100-00100100-001001
der 3er 36-65 refit Ser 8e refit 33r66/48 r 15152 r 57 r F f r 14 r 68 r 86 r
6a 1381 9a 1261 93 157 174 181 1561 741 20 162 182 1a21 591 351 681 191
20111100-40 BEGIN (ENTIFICATE
HEEFRICERINGINGEN ENDERGE LEERNEN + 660CCD/EMMOCSIOSEDBOOGRICKUN NEECCASUMMAATTA/TRIMINATYVOREALISEDBOC/EMMOCSIOSEDBOOGRICKUN NEEMMAAN/NEEMMONTERISETUN + 900000000000000000000000000000000000
Ew21812AAF WBYRIZA (MENING Y JAINET ZAFWR YRIANET I YW JAINET YWRAD YN GAND ARWANNET

## Certificate Standard: X.509

#### Most common format

- Separate from SSH and PGP
- From the "PKIX standards"
- Works out-of-the-box for TLS and HTTPS clients and servers
  - Good ecosystem support
- Arose from X.500 project in 1988
  - ITU-T (International Telecommunication Union Standards Body) tried to build a phone book

## Certificate Encoding

- X.509 builds on ASN.1. another ITU-T standard
- ASN.1 (Abstract Syntax Notation One) is an encoding standard

Certificates in the Wild

- Similar to ISON and Protobuf
- ASN.1 does not define binary representation
  - Its encoding rules can lead to incompatible binaries
- DER (Distinguished Encoding Rules) is an ASN.1 encoding format
  - Most commonly used for X.509
  - Sometimes also BER (Basic Encoding Rules) another ASN.1 encoding format
  - DER encoding/decoding handled by libraries

https://www.rfc-editor.org/rfc/rfc5280 https://cabforum.org/baseline-requirementsdocuments/

## **Problematic Standards**

- Certificates may be DER format or something "fancier"
  - Raw DER is binary data
- PEM (Privacy Enhanced EMail) encapsulates certificates
  - Adds header and footer
  - May hold any DER certificate, private key, public key
  - Similar to PGP and S/MIME
  - Don't ask why it's called "EMail" that's another 4 RFC document
- PEM headers hold labels to describe payload
  - Attempt at standardization via RFC 7468
  - Labels can be inconsistent
- PEM file "typically" stored as .pem, .crt or .cer
  - Raw DER is commonly stored as .der

#### Another Layer

- PEM might be included in an "envelope" with additional data
  - A "Certificate" might be a PEM file or an envelope with a PEM
- Envelopes are part of PKCS (Public Key Cryptography Standards)
  - Published by RSA (the company, not the algorithm)
- Most relevant are PKCS#7 and PKCS#12
  - Both can contain certificate chains
  - May hold PEM and raw DER, BER
  - May be stored as raw DER or as PEM
- PKCS#7 was rebranded as CMS (Cryptographic Message Syntax) by the IETF
  - Commonly stored as .p7b or .p7c and used in Java
- PKCS#12 commonly used with Microsoft products
  - Commonly stored as .pfx or .p12

https://www.rfc-editor.org/rfc/rfc3447 https://www.rfc-editor.org/rfc/rfc2315 https://www.rfc-editor.org/rfc/rfc7292

## Key Encoding

#### General pattern:

- ASN.1 data structure
- DER binary encoding
- PEM headers
- Private keys commonly use PKCS#8
  - Can be encrypted using a passphrase
  - Uses another encoding standard within a PEM



hpcsa-course-vm-key.pem was created for your cloud setup by OpenStack
 Try reading it with

shell

openssl x509 --text -noout -in hpcsa-course-vm-key.pem

This fails, its a PEM but not a certificate

shell

openssl rsa -check -in hpcsa-course-vm-key.pem

#### This prints your RSA private key

## Summary

- X.509 is the standard for certificates
- DER is a common encoding used with it
- PEM encapsulates binary DER certificates for better usability on the web
- PKCS#7 and PKCS#12 encapsulate certs with additional certs and data
- Private keys are commonly PEM encoded

## Overview

- By definition: Needs no certificates!
- Only goal: Bind names to public keys
  - ~/.ssh/authorized\_keys is a PKI
- Web PKI builds on top of certificates
  - Enables secure communication via the internet
- Anyone can create a public key and publish it
  - PGP (Pretty Good Privacy) concept
- On the web, what public keys and certificates to trust?

 What are Certificates and PKI
 Terminology
 Certificates in the Wild
 Certificate File Formats
 Public Key Infrastructure

 000
 0000
 0000
 0000
 0000
 0000

#### **Trust Stores**

Trust Store contains preconfigured trusted root certificates

- Usually preinstalled in your OS
- Root certificates are self-signed
- Who to trust to pick these?
- Microsoft, Apple, Mozilla (and partially Google) have trust store programs
- Over 100 CAs in most trust stores
  - Not all of them are morally "trustworthy"
  - Some governments operate CAs to spy on their citizens or impersonate websites
  - Some CAs were compromised in the past
- Trust stores are only as secure as the least secure CA

#### Attack Vector: Headset

2018 Sennheiser headset software installed a root certificate

- Into the trust store of users
- With the belonging private key hidden in the application
- Attackers would extract the private key
  - Use it to sign new web certificates
  - Systems with the Sennheiser software installed would accept them
- This is one example, there are more cases

https://medium.com/asecuritysite-when-bob-metalice/your-headphones-might-break-the-security-ofyour-computer-4f304ed86611

## Improvements to Security

- Baseline for issuing CAs was made more strict
- RFC 6962 introduced CT (Certificate Transparency)
  - Impartial observers check for fraudulent certificates
  - Checks are recorded in the certificates
- Root CA certificates are not used for automatically singing certificates
  - Only kept on special air-gaped machines
  - Intermediate certs used for automatic signing (certificate chain)
- Do not disable certificate path validation
  - ▶ e.g., curl -k
  - Unless you know you can trust the server

Public Key Infrastructure

### Inspecting Your Trust Store

- On Linux installed via ca-certificates package
- In /etc/ssl/certs
- Extract and inspect certs

```
shell
trust extract --format=x509-directory x509-certs & cd $_
openssl x509 -noout -text -inform=der -in NAME.pem
```

- Look for
  - Issuer
  - Subject
  - Validity
  - x509 extensions

## Table of contents

6 Practical Problems

#### 7 Let's Encrypt

## Getting a Certificate from a CA - In Theory

- 1 You create a key pair
- 2 You submit your key with your data to a CA
- 3 CA validates your data
- 4 CA signs and returns a certificate
- After some time your certificate expires
  - Replace it using above process

Practical	Problems
000000	00000

#### Getting a Certificate from a CA - In Practice

#### Solve two hard problems in computer science

- Cache invalidation
- Naming things

## DNs (Distinguished Names)

#### Certificate issuer/subject metadata field

- Subject: YOUR\_NAME
- YOUR\_NAME could be my.domain.de
- Historical artifact
- Instead only use CN (Common Name)
- List domains under SANs (Subject Alternative Names)
  - DNS:my.domain.de
  - Wildcards are possible DNS:\*.pages.gwdguser.de
  - Be careful with wildcards

## Generating a Key Pair

- Optimally, generate the key pair yourself
- Ensure security of the private key
- From key pair generate a CSR (Certificate Signing Request)
  - CSR is signed, can be freely shared
- Send CSR to a CA

Practical Problems	Let's Encrypt
000000000	00

Exercise

## **Certificate Signing Request Validation**

#### CSR Validation has 3 Options:

- DV (Domain Validation)
  - Check DNS name in WHOIS record
  - Either send an email
  - Or use ACME (Automatic Certificate Management Environment)
    - For a HTTP challenge
    - For a DNS TXT record check
- OV (Organization Validation) and EV (Extended Validation)
  - Builds on top of DV
  - Associate legal entity with certificate
  - Takes days or weeks to process
- DV is good enough usually

https://ietf-wg-acme.github.io/acme/draft-ietfacme-acme.html

## **Domain Validation Security**

#### What does successful DV mean?

#### Some entity was at one point in time able to either

- Read email
- Configure DNS
- Serve an HTTP challenge
- Relies on the security of these layers
- Fraudulent certificates were issued
  - e.g., on an Amazon domain in 2018

https://doublepulsar.com/hijack-of-amazonsinternet-domain-service-used-to-reroute-webtraffic-for-two-hours-unnoticed-3a6f0dda6a6f

## **Certificate Expiration**

- Validity range on certs is very common
- Not a hard requirement
- Incorrect system clocks are a problem
  - Out of sync system clock might reject a valid cert
  - Systems without internal time reset to UNIX epoch 0 or 01.01.1970
- Private keys should have proper life cycle
  - Separate keys for certificate singing and encryption

## **Certificate Renewal**

- No standard process
- Just replace the old cert with a new one
- Short lived certs are better for security
- Let's Encrypt defaults to 90 days

## **Certificate Revocation**

- Certs might be unneeded
- Private keys can be compromised
- $\Rightarrow$  Revoke the certificate
- Revoking X.509 certs is a mess
- Revocation status cannot be encoded in a certificate
- 2 systems in use
  - CRLs (Certificate Revocation Lists)
  - OCSP (Online Certificate Singing Protocol)

## CRLs (Certificate Revocation Lists)

#### Defined in RFC 5280

- Long list of serial numbers of revoked certs
- Can be linked in a certificate
  - "Look for my serial number here, if I am on this list, I am revoked"
- What if the CRL endpoint is unreachable?
  - The certificate is accepted
- Lists are often cached
  - Propagating new revocations takes time

## OCSP (Online Certificate Signing Protocol)

- Defined in RFC 2560
- Similar to CRL, endpoints can be queried for status of a cert
- Queries are for a specific certificate
- Privacy issue: The OCSP responder knows the websites you are visiting
- Solved by OCSP stapling
  - Owner of cert gets short-lived token from OCSP to validate cert
  - Might as well use short-lived certificates

https://www.ietf.org/rfc/rfc2560

## Using a Certificate

- 1 Set up a HTTPs server
- 2 Point it to certificate file and public key
- 3 Done

#### Overview

#### Non-profit company

- https://letsencrypt.org/
- Founded in 2012
- Provides a free service for acquiring certificates
  - Only supports automatic domain-validation
  - Over 300 mil (unexpired) certificates November 2022

#### Certbot

#### Open-source tool

- Allows obtaining certificates from Let's Encrypt
- Can also install them via plugins
  - Support Apache, NGINX, etc.
- Automatically renew certificates
- Uses ACME for DV

## Exercise

Please work on the exercise now.