

Role certifikátu při zabezpečení šifrovaného spojení

Petr Krčmář



26. května 2018



Uvedené dílo (s výjimkou obrázků) podléhá licenci Creative Commons Uveďte autora 3.0 Česko.

<https://www.petrkrcmar.cz>

Co je to HTTPS?

- přenos protokolu HTTP v šifrovaném TLS tunelu
- provoz běží po TCP portu 443
- data jsou přenášena šifrovaně a autentizovaně
- pro sestavení kanálu se používá asymetrická kryptografie
- k potvrzení identity se využívají **certifikáty**
- pro šifrování se používá symetrická kryptografie

Před čím chrání HTTPS?

- před síťovými útoky (věříte zdejší Wi-Fi?)
- před odposlechem komunikace ([RFC 7258](#) – odposlech je útok)
- před man-in-the-middle – víme, s kým komunikujeme
- před blokováním části obsahu (Wikipedie)
- před pozměňováním dat během přenosu
- před vkládáním supercookies, malware, trackovacích kódů
- před únikem osobních údajů – nechcete e-shop bez HTTPS
- před únosem session cookie – při přihlašování

Před čím chrání HTTPS?

- před síťovými útoky (věříte zdejší Wi-Fi?)
- před odposlechem komunikace ([RFC 7258](#) – odposlech je útok)
- před man-in-the-middle – víme, s kým komunikujeme
- před blokováním části obsahu (Wikipedie)
- před pozměňováním dat během přenosu
- před vkládáním supercookies, malware, trackovacích kódů
- před únikem osobních údajů – nechcete e-shop bez HTTPS
- před únosem session cookie – při přihlašování
- příjemný bonus: možnost nasazení HTTP/2 – výkon ([http2demo.io](#))
- používá 25 % webů (podle [W3Techs](#))



Zařízení značky Sandvine (dnes Procera)
(Root.cz: [Vlády přeměrovávají uživatele na software doplněný o malware](#))

Před čím HTTPS nechrání?

- před únikem informací o doméně (DNS, SNI)
- před sběrem metainformací (netflow, IP...)
- před phishingovými weby
- před blokováním provozu na síťové vrstvě
- před útokem na legitimního provozovatele webu (XSS, SQL...)
- před zlým úmyslem autora webu

Apple ID

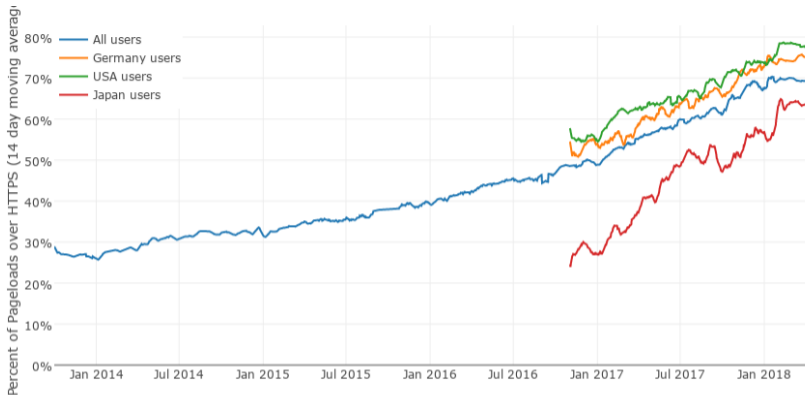
Manage your Apple account

Apple ID

Password

Remember me

[Forgot Apple ID or password?](#)



- 70 % stránek načteno po HTTPS, v USA dokonce 78 %
- 81 ze 100 největších webů má HTTPS (před rokem jen 37!)

Problém: důvěryhodné předání klíče

- šifrovat bezpečně asymetrickou šifrou umíme
- protistrana je pro nás ale neznámá
- autentizace stejně důležitá jako silná šifra
- bez ní se kdokoliv může vydávat za kohokoliv
- problém důvěryhodného předání veřejného klíče
- nastupují authority: důvěryhodní prostředníci
- ověří žadatele, vystaví certifikát



Certifikát je internetový pas


- pas propojuje fotografii obličeje se jménem
- certifikát propojuje doménové jméno s veřejným klíčem
- server se prokazuje: toto je potvrzení o mém klíči
- pas vystavují jen důvěryhodné státy
- certifikáty vystavují důvěryhodné authority
- celník ověří totožnost na základě dokumentu známého státu
- prohlížeč ověří identitu na základě dokumentu známé authority
- důvěryhodné authority a jejich klíče jsou předinstalované v software
- to celé ↑ se jmenuje PKI (Public Key Infrastructure)

Co je to certifikát?


- **veřejný** dokument, který obsahuje hlavně:
 - jméno autority
 - doménová jména
 - veřejný klíč žadatele
 - datum platnosti
 - podpis autority
 - a další
- elektronický podpis = nezfalšovatelné
- ověříme pomocí známého veřejného klíče v software
- certifikát **není tajemstvím**, chráníme privátní klíč

Tři druhy certifikátů


- **EV** – Extended Validation
 - velmi drahý, důkladné ověření identity žadatele
 - signalizace zeleným názvem vedle adresy
 - nesmí obsahovat wildcard (hvězdičku)
 - nemůžou vydávat všechny authority
- **OV** – Organization Validation
 - levnější varianta, zběžné prověření identity žadatele
 - v prohlížeči nemá speciální označení
- **DV** – Domain control Validation
 - velmi levné vystavení, lze automatizovat
 - jen kontrola doménového jména
 - není nijak svázán s identitou žadatele
 - nejběžnější typ certifikátu
- max. platnost všech je 825 dnů (přibližně 27 měsíců)

 Zabezpečeno | <https://www.root.cz>

 Fio banka, a.s. [CZ] | <https://www.fio.cz>

 Nezabezpečeno | www.policie.cz

 Nezabezpečeno | <https://www.sejf.cz>

 <https://very.badssl.com>

Řetězec důvěry

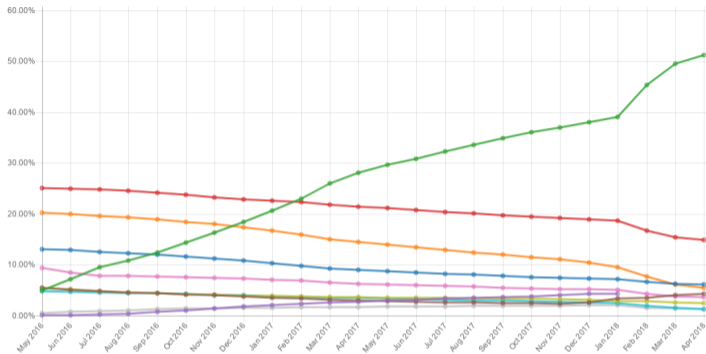
- software zná kořenové certifikáty s veřejným klíčem autority
- od serveru dostane řetězec – delegace pravomocí autority
- certifikáty se musí vzájemně potvrzovat
- konečný certifikát potvrzuje identitu žadatele (serveru)
- zároveň obsahuje veřejný klíč
- identita je potvrzena, klíč předán protistraně
- komunikace může začít
- existuje asi 1000 důvěryhodných CA
- ve skutečnosti je to asi 60 firem
- nedodání mezilehlých → velmi častá chyba

Let's Encrypt

- projekt EFF, Mozilla Foundation, Akamai a Cisco Systems
- představena v listopadu 2014, beta od prosince 2015
- od dubna 2016 v ostrém provozu



Top SSL Issuers



#	SSL Issuer	Percentage
1	Let's Encrypt	51.21%
2	COMODO CA Limited	14.82%
3	GoDaddy.com	6.14%
4	GeoTrust Inc.	5.5%

(NetTrack.info)

Vlastnosti Let's Encrypt

Let's Encrypt to dělá jinak:

- **zdarma** – stačí vlastnit doménu/ovládat server
- **automaticky** – vše vyřídí stroje mezi sebou
- **průhledně** – od začátku všechny certifikáty zveřejňuje
- **otevřeně** – protokol i software jsou otevřené

Vlastnosti Let's Encrypt

Let's Encrypt to dělá jinak:

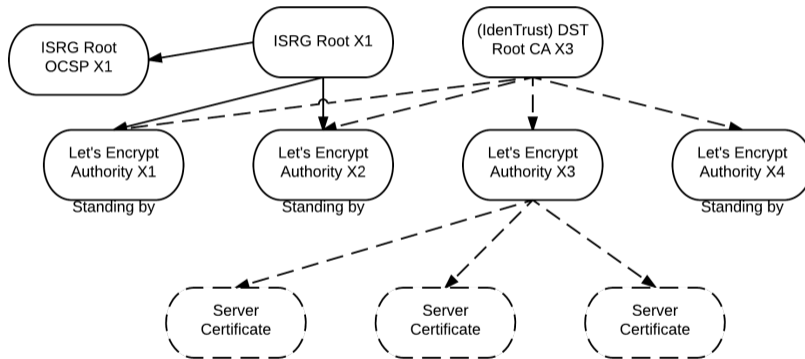
- **zdarma** – stačí vlastnit doménu/ovládat server
- **automaticky** – vše vyřídí stroje mezi sebou
- **průhledně** – od začátku všechny certifikáty zveřejňuje
- **otevřeně** – protokol i software jsou otevřené

- provoz stojí **3 miliony dolarů ročně**
- přispějte na provoz, pokud můžete

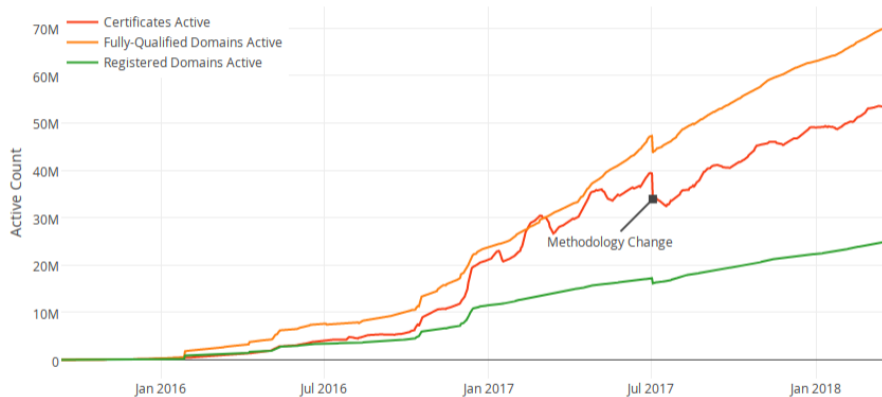
Vlastnosti certifikátů Let's Encrypt

- pouze DV certifikáty
- ověření pomocí HTTP nebo DNS
- platnost certifikátu 90 dnů
- možno až 100 doménových jmen v SAN
- nově také wildcard (ale jen DNS ověření)
- kořenový certifikát zatím jen ve Firefoxu (50+)
- možnost certifikáty revokovat
- všechny vystavené certifikáty jsou veřejné
- využívá kořenové autority IdenTrust

Cross-signing



Počet vystavených certifikátů



(Let's Encrypt stats)

- protokol ACME
 - Automated Certificate Management Environment
 - JSON nad HTTPS
- automatické utility
- ověření pomocí výzev v `/.well-known/`
- nebo DNS `_acme-challenge.<doménové jméno>` TXT "hex řetězec"
- vygenerujete klíč, dostanete certifikát a chain
- cross-sign IdenTrust („DST Root CA X3“ Root CA)
- výchozí utilita Certbot konfiguruje web server
- existuje celá řada dalších implementací (ACME.sh, Dehydrated...)
- část software integruje (Caddy, mod_md pro Apache...)

Pozor na rate limiting

- 20 žádostí v jedné doméně (SLD) za týden
- na obnovení certifikátu je výjimka
- 5 duplicitních (se stejnými doménami) certifikátů za týden
- **revokace neresetuje limity**
- 300 nedokončených žádostí za týden – pro vývojáře
- 100 doménových jmen v certifikátu
- existuje testovací (staging) prostředí s mnohem vyššími limity
- viz [Let's Encrypt rate limits](#)

Čím to otestovat?

- [SSL Labs Test](#) - velmi podrobný test
- [SSL Decoder](#) - vypíše všechny detaily o certifikátech
- [Symantec CryptoReport](#) - protokoly, chyby, díry
- [GeoCerts SSL Checker](#) - ukazuje řetězec
- [COMODO SSL Analyzer](#) - a ještě jeden
- gcr-viewer v balíčku gnome-keyring

```
openssl s_client -showcerts -connect www.root.cz:443 < \  
/dev/null | openssl x509 -outform DER > cert.der
```

Jak to ještě vylepšit: hlavička HSTS

- HTTP Strict Transport Security (HSTS)
- hlavička v HTTP odpovědi ([RFC 6797](#))
- tento web musí mít vždy důvěryhodný certifikát
- prohlížeč sám přepíše http:// na https://
- TOFU = Trust On First Use

Jak to ještě vylepšit: hlavička HSTS

- HTTP Strict Transport Security (HSTS)
- hlavička v HTTP odpovědi ([RFC 6797](#))
- tento web musí mít vždy důvěryhodný certifikát
- prohlížeč sám přepíše http:// na https://
- TOFU = Trust On First Use

```
Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
```

Jak to ještě vylepšit: hlavička HSTS

- HTTP Strict Transport Security (HSTS)
- hlavička v HTTP odpovědi ([RFC 6797](#))
- tento web musí mít vždy důvěryhodný certifikát
- prohlížeč sám přepíše http:// na https://
- TOFU = Trust On First Use

```
Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
```

- možno i HSTS preload
- <chrome://net-internals/#hsts>
- rozšíření HTTPS Everywhere

Funguje to celé (?)

- tohle celé funguje skvěle

Funguje to celé (?)

- tohle celé funguje skvěle
- až na případy, kdy to selhává

Funguje to celé (?)

- tohle celé funguje skvěle
- až na případy, kdy to selhává
- authority jsou „univerzálně důvěryhodné“
- kdokoliv vystavuje cokoliv
- DigiNotar, Thawte, Symantec, WoSign...
- technická chyba, omyl, útok, státní zájmy
- řetěz je silný jako nejslabší článek
 - bezpečnost neurčuje nejlepší, ale nejhorší
 - jedno shnilé jablko zničí celý košík

Provozovatelé se bojí

- provozovatelé služeb se bojí
- PKI je jedinou ochranou
- robustní, ale stojí na bezpečnosti autorit
- pokud selže, může se kdokoliv vydávat za kohokoliv
- typicky Google, Microsoft, Apple, GitHub...
- ukradení přihlašovacích údajů, odposlech
- vkládání vlastních informací
- platný certifikát = internetová identita

Řešení = transparentnost

- donutit autority zveřejňovat **všechny** certifikáty
- možnost monitoringu i zpětného auditu
- pokud někdo vydá neoprávněně, můžu reagovat
- spustím poplach, můžu revokovat
- mám přehled o všech vydaných certifikátech
- autority se dostávají pod veřejnou kontrolu
- hlídám své domény, kdokoliv hlídá cokoliv

Certificate Transparency

- veřejné logy pro ukládání certifikátů
- lze do nich jen přidávat (Merklov hashový strom)
- kdokoli je může mirrorovat a prohledávat v nich
- kdokoli může přidávat certifikáty
- kvůli ochraně ale pouze od uznávaných CA
- odstranění certifikátu je detekovatelné
- není možné antedatovat certifikáty
- monitor – kontroluje logy a hledá problémy
- auditor – kontroluje konkrétní certifikát (prohlížeč)
- definováno v [RFC 6962](#)

- první log spustil Google v březnu 2013
- v září 2013 začala první CA vkládat (DigiCert)
- od 1. ledna 2015 vyžaduje Chrome pro EV
 - přítomnost alespoň ve dvou lozích
- od 1. června 2016 vyžaduje u všech od Symantec
- od 30. dubna 2018 je **vyžadováno u všech**
- původně to měl být už říjen 2017
- Firefox oznámil podporu, ale bez termínů

Jak authority donutit

- bude fungovat, jen když to budou dělat všichni
- musí existovat donucovací mechanismus
- je zabudován do prohlížeče
- prohlížeč zkontroluje že je certifikát v logu
- (vedle data platnosti, domény a podobně)
- jen takový certifikát bude důvěryhodný
- technicky se vynutí zveřejňování certifikátů

- klienti se sami ptát nebudou
- to by neškálovalo a unikaly by informace
- důkazní břemeno je na serveru, který certifikát předává
- ten musí doložit, že je certifikát v logu
- ideálně ho vloží už CA, ale může i sám
- log vydává Signed Certificate Timestamp (SCT)
 - příslib budoucího zařazení certifikátu do stromu
 - server musí klientovi doručit i SCT

Tři způsoby doručení

1 OCSP stapling

- složité a nespolehlivé (umí prohlížeč OCSP?)
- server i autorita musí spolupracovat
- autorita získá SCT a vloží do OCSP responderů
- server musí podporovat stapling
- s OCSP zprávou předává i SCT

Tři způsoby doručení

1 OCSP stapling

- složité a nespolehlivé (umí prohlížeč OCSP?)
- server i autorita musí spolupracovat
- autorita získá SCT a vloží do OCSP responderů
- server musí podporovat stapling
- s OCSP zprávou předává i SCT

2 rozšíření TLS

- server pošle certifikát a získá SCT
- změní konfiguraci web serveru (podpora?)
- rozšíření TLS `signed_certificate_timestamp`

Tři způsoby doručení

1 OCSP stapling

- složité a nespolehlivé (umí prohlížeč OCSP?)
- server i autorita musí spolupracovat
- autorita získá SCT a vloží do OCSP responderů
- server musí podporovat stapling
- s OCSP zprávou předává i SCT

2 rozšíření TLS

- server pošle certifikát a získá SCT
- změní konfiguraci web serveru (podpora?)
- rozšíření TLS `signed_certificate_timestamp`

3 rozšíření certifikátu

- nulová zátěž na provozovatele serveru
- vše zařídí CA, pošle do logu, získá SCT
- SCT je pak přímo součástí certifikátu

Certificate

Subject `www.nebezi.cz`
SAN `cas.dev.nebezi.cz`
`cas.nebezi.cz`

[Show more \(14 total\)](#)

Valid from `Thu, 29 Mar 2018 18:02:58 GMT`
Valid until `Wed, 27 Jun 2018 18:02:58 GMT`
Issuer `Let's Encrypt Authority X3`

[Open full certificate details](#)

Certificate Transparency

SCT `Cloudflare 'Nimbus2018' Log (Embedded in certificate, Verified)`
SCT `Google 'Icarus' log (Embedded in certificate, Verified)`

[Show full details](#)

Současný stav logů

- v tuto chvíli je v Chrome uznáváno 27 logů od 10 firem
- Certly, DigiCert, Comodo, Google, Cloudflare...
- jsou různé velké (statisíce až desetimiliony certů)
- Google uvádí, že je v nich přes 1,3 miliardy záznamů
- infrastruktura se bude časem zahušťovat
- Google nabízí i [webové rozhraní](#)
- případně vyhledávače třetích stran jako [crt.sh](#)
- další info na www.certificate-transparency.org

Kde to uvidím?

- Chrome devtools → Security → Main origin
- chrome://net-internals „signed_cert...”
- na webu crt.sh
- Certspotter [služba](#), [GitHub](#)
- pomocí řady nástrojů a knihoven
- OpenSSL má od verze 1.0.2 podporu pro SCT
- Facebook má vlastní monitoring posílající maily

Pohled dovnitř certifikátu

Co je certifikát

- strojově čitelný dokument ve formátu ASN.1

```
$ openssl asn1parse -in cert.crt  
$ echo "3045820D706574726B72..." | xxd -r -p
```

- datová struktura klíč – hodnota
- každá položka má číselný identifikátor
- a příznak o kritičnosti – nekritičnosti
- kritická nerozpoznaná položka – **musí** být zamítnuta
- nekritická nerozpoznána – **může** být ignorována
- každá rozpoznaná položka **musí** být validována
- certifikát kódován pomocí DER nebo base64 (PEM)

Získání certifikátu

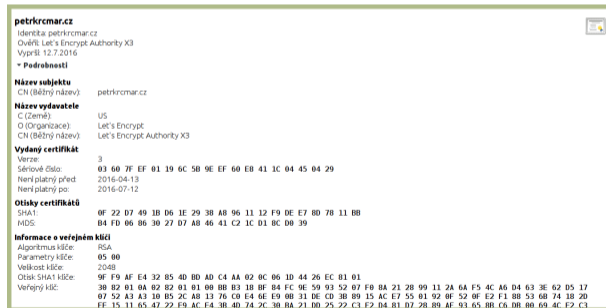
```
$ openssl s_client -showcerts -connect www.petrkrcomar.cz:443 \\  
-servername petrkrcomar.cz < /dev/null | openssl x509 -outform PEM  
  
-----BEGIN CERTIFICATE-----  
MIIFIjCCBAqgAwIBAgISA2B/7wEZbFue72DoQRwERQQpMA0GCSqGSIb3DQEBCwUA  
MEoxCzAJBgNVBAYTAlVTMRYwFAYDVQKQEW1MZXQncyBFbmNyeXB0MSMwIQYDVQQD  
ExpMZXQncyBFbmNyeXB0IEF1dGhvcml0eSBYMTZsMzYwLjEwLjEwLjEwLjEwLjEw  
NjA3MTIxMTA3MDBaMBGxGjAUBG9NVBAMTDXBldHJrcmNtYXJ3Y3owggEiMA0GCSqG  
SIb3DQEBAQUAA4IBDwAwggEKAoIBAQC7sxi/hPyeWZNSB/CKISiZESpq9Uym1GM+  
YtUXB1KjoxC1LKgTdsDkbukLMd7N04kVr0dVAZIPUg/i8YhTa3QYLf4VEWVHIums  
5DtNdCwwuiHdJSLD8tSB1yiJr5Nli8bbAGLM4sM9QrL3BGP1N5SD3h3e12J0Q0r2  
920xoT3vDCUWnKvViuggLLSs2sogKrlpKBwCUmvsG2mmgMwbBEIjapg0lUAs7bNa  
dn61a00eN3Y5L2Z5PTusR0e8Ib7VISibG4Awo8hjaIppVivQyFXT/32AZbRxhnFK  
arLzk/eFpt5XgfkMeCiGJ/UZ5TWFuxBh9+08QN4+vdvnh14V2KD1AgMBAAGjggIy  
...  
  
WIXNxiCXe/EBWMB04Aek19I+ovnCFnkfsD05vxDBZYZfumg2YiTwGY2sG/EJNfUx  
Nsjkxbd7VZuZhJUY40IRm6Iko2QPmQ==  
-----END CERTIFICATE-----
```


Prohlížení certifikátu

- ve webovém prohlížeči
- přímo pomocí OpenSSL

```
$ openssl x509 -in certificate.crt -text -noout
```

- gcr-viewer v balíčku gnome-keyring



petrkrcmar.cz
Identita: petrkrcmar.cz
Ověřil: Let's Encrypt Authority X3
Vyprší: 12.7.2016

▼ **Podrobnosti**

Název subjektu
CN (Běžný název): petrkrcmar.cz

Název vydavatele
C (Země): US
O (Organizace): Let's Encrypt
CN (Běžný název): Let's Encrypt Authority X3

Vydán certifikát
Verze: 3
Sériové číslo: 03 60 7F EF 01 19 6C 5B 9E EF 60 EB 41 1C 04 45 04 29
Není platný před: 2016-04-13
Není platný po: 2016-07-12

Otisky certifikátů
SHA1: 0F 22 D7 49 1B D6 1E 29 38 A8 96 11 12 F9 DE E7 8D 7B 11 BB
MD5: B4 FD 06 86 30 27 D7 A8 46 41 C2 1C 01 8C D0 39

Informace o veřejném klíči
Algoritmus klíče: RSA
Parametry klíče: 05 00
Velikost klíče: 2048
Otisk SHA1 klíče: 9F F9 AF E4 32 85 4D BD AD C4 AA 02 0C 06 1D 44 26 EC 01 01
Veřejný klíč: 30 82 01 0A 02 82 01 01 00 BB B3 18 BF 84 FC 9E 59 93 52 07 F9 8A 21 28 99 11 2A 6A F5 4C A6 D4 63 3E 62 D5 17 07 52 A3 A3 10 B5 2C A8 13 76 C8 E4 6E E9 0B 31 DE CD 3B 89 15 AC E7 55 81 92 0F 52 0F E2 F1 8B 53 68 74 18 2D FF 15 11 65 47 22 F9 AC E4 38 4D 74 2C 38 BA 21 00 25 22 C3 E2 04 81 07 28 89 AF 93 65 8B 06 DB 08 69 4C E2 C3

Co certifikát obsahuje

- Číslo verze
- Sériové číslo
- ID algoritmu podpisu
- Jméno vystavovatele
- Platnost: od do
- Jméno subjektu
- Klíč subjektu
 - algoritmus veřejného klíče
 - veřejný klíč samotný
- Rozšíření a volitelné položky
- Podpis certifikátu
 - algoritmus podpisu
 - podpis samotný

- verze standardu X.509
- verze formátu certifikátu
- první v RFC 1422
- verze 2 přidala možnost ID autority
- pro případ recyklace jména - nedoporučuje se
- verze 3 - RFC 2459
- přidala volitelná rozšíření
- dnes nejpoužívanější je verze 3

Sériové číslo

- unikátní číslo certifikátu
- v rámci autority se nesmí opakovat
- jméno autority a sériové číslo jednoznačně identifikuje certifikát
- používá se pro identifikaci při revokaci
- jedná se o celé číslo
- není definováno, jak je číslo tvořeno
- klienti musí být schopni zpracovat alespoň 128 bitů
- třeba 03:60:7f:ef:01:19:6c:5b:9e:ef

Algoritmus podpisu

- algoritmus, kterým je podepsán certifikát
- uložen jako identifikátor (OID) dané kombinace
- vždy dvojice kryptografické a hashovací funkce
- historicky RSA a DSA s MD5 a SHA-1
- dnes nepodporované kombinace - nebezpečné
- dnes pouze SHA - až do 512
- podporu ECDSA zavádí RFC 5758
- třeba sha256WithRSASignature

Jméno vystavovatele

- neprázdné popisné jméno autority
- obvykle v UTF8
- může obsahovat další položky
 - stát, organizace a další (viz další slide)
 - totéž pak u jména subjektu
- musí odpovídat důvěryhodné autoritě
- přes jména vede řetězec důvěry (!)
- třeba C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3

... formát jména (RFC 5280)

- stát (countryName, C)
- organizace (organizationName, O)
- organizační jednotka (organizationalUnitName, OU)
- stát či oblast (stateOrProvinceName, ST)
- běžné jméno (commonName, CN)

- datum a čas - začátek a konec platnosti
- časová zóna Greenwich Mean Time (Zulu)
- včetně sekund, i kdyby byly rovny nula
- do roku 2050 se používá formát UTCTime
 - YYMMDDHHMMSSZ
 - znak Z označuje formát zápisu
 - YY < 50 = 19xx; YY > 50 = 20xx
- od roku 2050 povinně GeneralizedTime
 - YYYYMMDDHHMMSSZ

Jméno subjektu

- subjekt, kterému patří veřejný klíč
- může být tady nebo v rozšíření subjectAltName
- dnešní prohlížeče CN ignorují, rozhodující je SAN
- pokud je subjekt CA, pod tímto jménem vydává certifikáty
- třeba CN=petrkrcmar.cz
- u mailu
CN=petr.krcmar@iinfo.cz/emailAddress=petr.krcmar@iinfo.cz

Klíč subjektu

- samotný „náklad“ v certifikátu
- nejprve algoritmus klíče (už jsme viděli)
- třeba rsaEncryption (2048 bit)
- poté klíč samotný

```
00:e6:6f:18:0b:2a:c8:c1:e0:cb:9c:dc:67:5c:be:
37:1c:42:96:ab:0b:85:9a:cf:45:5e:0e:f1:39:f5:
a4:c2:92:b4:39:05:c1:25:07:64:a8:77:6d:01:ae:
e0:01:03:76:f3:d4:75:30:f1:b8:8f:e7:7d:5f:9d:
cb:25:df:8f:44:c1:7d:6c:c4:b8:b3:1b:23:75:70:
1f:d9:ef:62:3f:2f:f7:91:dc:e3:07:9f:67:37:f7:
0c:51:85:f7:3e:e3:17:04:8a:31:21:11:c5:64:c8:
36:15:5a:a1:57:41:c0:4f:26:f1:71:16:5d:39:25:
4a:a1:85:fc:48:45:0c:83:c8:ae:b1:aa:67:3f:81:
f6:e6:ff:98:2b:34:73:ae:e9:07:01:4b:ac:18:e0:
```

...

Rozšíření dle RFC 5280

- součástí certifikátu jsou volitelná rozšíření
- vždy mají označení kritičnosti (výchozí je FALSE)
- umožňují doplnit informace o užití certifikátu
- každé rozšíření se může objevit jen jednou
- CA musí některá rozšíření povinně uvést
 - key identifiers
 - basic constraints
 - key usage
 - certificate policies

Důležitá rozšíření

- Basic Constraints – patří certifikát CA?
- Subject Alternative Name – další identity subjektu
- CRL Distribution Points – odkazy pro CRL (viz revokace)
- Authority Information Access – odkaz pro OCSP (viz revokace)
- Authority Key Identifier – identifikace klíče pro podpis revokací
- Subject Key Identifier – (obvykle) hash klíče
 - slouží aplikacím pro rychlé hledání

Další používaná rozšíření

- Key Usage – bitová mapa o možnostech využití klíče
 - šifrování, podpis, revokace...
- Extended Key Usage – účel použití klíče
 - pouze koncové certifikáty
 - serverové TLS, pošta
 - časová razítka, podpis kódu
 - podpisy OCSP a další
- Name Constraints – omezení pro určité domény
 - pouze u certifikátů CA
- Certificate Policies – strojově čitelná pravidla vydání a použití
 - 2.23.140.1.2.1 = DV; 2.23.140.1.2.2 = OV; EV různě

Podpis certifikátu

- celý certifikát je podepsaný autoritou
- nejprve ID funkce a algoritmu
- sha256WithRSAEncryption
- poté následuje samotný podpis

```
71:84:5e:01:a2:b3:e6:2f:c0:00:19:fa:49:46:42:7e:f9:6d:  
c9:b9:b3:fa:f9:d3:1c:a6:5a:1f:20:f3:84:9a:8c:a0:c5:f2:  
09:9d:89:b9:a5:b9:a0:a5:d8:3c:99:07:a8:d9:8d:87:32:c0:  
0d:32:64:8e:f5:c1:c5:13:6f:09:7b:d7:69:4c:2d:62:bf:db:  
8b:d5:86:fc:ad:8d:ab:45:31:49:0b:a7:29:62:82:b8:6f:68:  
41:61:e8:bb:df:22:58:96:cb:f9:60:19:ec:18:3b:eb:33:e2:  
c8:60:46:f6:21:94:58:d7:f6:50:02:77:27:bf:80:d4:11:7e:  
dc:19:c3:08:23:76:9b:9c:60:c1:4d:22:b0:d2:c8:7b:2f:9b:  
84:1f:73:f0:07:51:8c:2b:a4:71:84:f4:6f:83:56:39:a5:9b:  
...
```

Revokace certifikátů

- chceme možnost revokovat před vypršením
- měníme CA, končí služba, zmizely klíče...
- čím delší platnost, tím větší riziko
- CA poskytují rozhraní k ověření platnosti
 - CRL - seznamy neplatných certifikátů
 - OCSP - razítka potvrzující platnost
- kořenový certifikát nelze revokovat

CRL (Certificate Revocation List)

- veřejné seznamy revokovaných klíčů
- stahuje se po HTTP
- odkaz v rozšíření CRL Distribution Points
- identifikátorem je sériové číslo certifikátu
- dále datum revokace a důvod
- vše podepsáno klíčem CA
- revokovaný cert musí být v CRL až do konce platnosti

```
$ openssl crl -inform DER -text -noout -in DSTROOTCAX3CRL.crl
```


Nevýhody CRL

- nutnost pravidelného stahování a obnovování
- poměrně velké objemy
- síťová zátěž na klienta i autoritu
 - problém na mobilních datech
 - neškáluje to ani u autority
- čím víc certifikátů, tím delší seznamy
- klient musí mít komplexní knihovny a databázi
- prohlížeče od CRL upouštějí

OCSP (Online Certificate Status Protocol)

- jednoduchý protokol pro revokační dotazy
- definuje ho RFC 6960
- dotazy probíhají po HTTP
- klient se ptá OCSP responderu
- odpověď je krátká - týká se jednoho certifikátu
- tři možné stavy odpovědi
 - good
 - revoked
 - unknown
- zpráva je digitálně podepsaná autoritou
- stejnou nebo může CA delegovat na OCSP autoritu
 - ta může mít jiný klíč a nemůže vydávat certifikáty

Výhody a nevýhody OCSP

Výhody

- malý dotaz a odpověď
- odpověď má dobu platnosti
- dá se cachovat

Nevýhody

- klienti zasypávají autoritu dotazy
- autorita ví, kdo kdy kam přistupuje
- neškáluje to

Řešení: OCSP stapling

- díky podpisu je validační zprávu možné předat
- je platná sama o sobě
- stapling = přicvaknutí k certifikátu
- responderu se ptá server a odpověď přicvakne
- klient se nemusí ptát sám
- škáluje to, neunikají osobní údaje
- původní stapling neumí požádat o více razítek najednou
- problém pro mezilehlé – neumíme se serveru zeptat
- musíme zjišťovat revokace jinou cestou
- řeší RFC 6961 – umí poslat dotaz na víc certifikátů

Podpora staplingu v serverech a klientech

- Apache od 2.3.3
- nginx od 1.3.7
- Microsoft IIS od 2008
- Firefox od 3, od 26 stapling
- Internet Explorer od Vista
- Safari od OS X 10.7
- Opera od 8.0
- Exim v server i klient módu
- Chrome nepodporuje – šíří si seznamy sám

Praktická ukázka OCSP stapling

- stačí se připojit ke klientovi, který stapling umí

```
$ openssl s_client -connect petrkrmar.cz:443 -status < /dev/null
```

- na začátku komunikace je info o OCSP

```
CONNECTED(00000003)
OCSP response:
=====
OCSP Response Data:
  OCSP Response Status: successful (0x0)
  Response Type: Basic OCSP Response
  Version: 1 (0x0)
  Responder Id: C = US, O = Let's Encrypt, CN = Let's Encrypt Authority X3
  Produced At: Apr 28 09:12:00 2018 GMT
  Responses:
  Certificate ID:
    Hash Algorithm: sha1
    Issuer Name Hash: 7EE66AE7729AB3FCF8A220646C16A12D6071085D
    Issuer Key Hash: A84A6A63047DDDBAE6D139B7A64565EFF3A8ECA1
    Serial Number: 0304B9F9427D8E6EDFE48B0343C7EFB2653C
  Cert Status: good
  This Update: Apr 28 09:00:00 2018 GMT
  Next Update: May 5 09:00:00 2018 GMT
```

Otázky?



Petr Krčmář
petr.krcmar@iinfo.cz