

Postkvantové certifikáty s MTC

bezpečnost bez podpisu

Petr Krčmář



4. října 2026



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

O mně

- linuxák od roku 1998
- správce serverů
- lektor a konzultant
- šéfredaktor [Root.cz](#)
- člen [vpsFree.cz](#)
- organizátor [LinuxDays](#)
- můj web je [petrkrmar.cz](#)



Prezentace už teď na webu

www.petrkrccmar.cz

Proč už teď?

Postkvantová kryptografie

- kryptografie stojí na problémech faktorizace a diskrétního logaritmu
- kvantové počítače dokáží některé úlohy řešit efektivněji
 - dostatečně výkonný může použít Shorův algoritmus k faktorizaci
 - ohroženy zejména RSA, Diffie-Hellman a eliptické křivky
- takový kvantový počítač (zřejmě) ještě neexistuje
 - nemusíme se to hned dozvědět
- riziko dešifrování starší komunikace (Harvest now, decrypt later)
- budeme mít takový počítač?
 - nevíme, ale raději se připravujeme

Harvest now, decrypt later

- útočník může zašifrovaná data zachytit a uložit už dnes
- v budoucnu je může dešifrovat pomocí kvantového počítače
- některá data jsou citlivá i po letech
 - osobní a zdravotní údaje
 - obchodní tajemství a výzkum
 - státní a vojenské informace
 - dlouhodobé smlouvy a archivovaná komunikace
- pro tato data je hrozba aktuální už dnes
- i když takový kvantový počítač zatím neexistuje

Není to v principu nový problém

- vývoj je postupný, vyvíjejí se algoritmy i hardware
- průběžně opouštíme staré a slabé mechanismy
 - protokoly: SSL, TLS 1.0, TLS 1.1...
 - šifry: DES, 3DES, RC4, RC2...
 - haše: MD5, SHA-1...
 - podpisy: DSA, RSA se SHA-1, RSA s krátkým klíčem...
 - přenosy klíčů: statický RSA key transport, slabý Diffie-Hellman...
- nemůžeme používat 512bitový klíč RSA z Netscape 4.51 (1999)
 - moderními metodami a hardwarem jej lze **prolomit za 32 hodin**
 - se soukromým klíčem je možné podepsat vlastní certifikát
 - tyto certifikáty byly z Netscape odstraněny v roce 2002

Proč firmy začínají jednat už teď?

- kryptografie není jen jedna izolovaná knihovna
- nestačí jednoduše vyměnit jednu komponentu
- RSA a ECC jsou rozšířené v:
 - TLS a VPN
 - certifikačních autoritách
 - elektronických podpisech
 - aktualizacích firmwaru
 - podepisování softwaru a dokumentů
- migrace vyžaduje:
 - inventuru algoritmů
 - změny protokolů
 - nové certifikáty
 - testování kompatibility
- u rozsáhlých systémů může přechod trvat mnoho let

Plány a termíny

Z plánů se stávají termíny

- NIST v srpnu 2024 **dokončil standardy postkvantové kryptografie:**
 - ML-KEM pro ustavení klíčů
 - ML-DSA pro digitální podpisy
 - SLH-DSA jako hašovací algoritmus
- organizace mohou začít nasazovat
- vládní a regulační plány stále častěji požadují:
 - inventuru používané kryptografie
 - pilotní nasazení
 - podporu postkvantových algoritmů
 - postupné odstranění RSA a klasického ECC

Přechod nebude „někdy v budoucnu“

- USA počítají s **postupným přechodem** v období 2030–2035
- Google **posunul plán přechodu** už na rok 2029
- **evropský koordinovaný plán** pracuje s podobnou posloupností:
 - do konce 2026: strategie, inventury a první pilotní projekty
 - do konce 2030: přechod rizikových systémů a kritické infrastruktury
 - do konce 2035: dokončení migrace v maximálním možném rozsahu
- termín dokončení je daleko, ale příprava začíná nyní
- je to spousta kroků na mnoha úrovních
 - řada změn vyžaduje úpravy na více úrovních
 - často koordinovaně zasáhne ekosystém PKI
- nemůžeme čekat do roku 2030
- změny už se připravují a dějí, pak už bude pozdě

Co z toho plyne pro certifikáty?

- certifikát není jen dokument s veřejným klíčem
- je součástí celého řetězce důvěry:
 - certifikační autorita
 - kořenové a mezilehlé certifikáty
 - podpisy certifikátů
 - validace a revokace
 - podpora v klientech a serverech
- postkvantová kryptografie proto mění nejen algoritmus
 - mění se i velikosti klíčů, podpisů a certifikátů
- zaměříme se na postkvantové certifikáty a jejich bezpečnost
 - tahle oblast nás zajímá jako správce

Problém s velkými klíči

Velké klíče

- postkvantové algoritmy mají výrazně větší veřejné klíče a podpisy
- RSA má typicky 2048 bitů, tedy 256 bajtů
 - podpis ECDSA P-256: přibližně 64 bajtů
 - podpis ML-DSA-44: přibližně **2420 bajtů**
 - veřejný klíč ML-DSA-65: přibližně **1952 bajtů**
- nejde tedy jen o výměnu jednoho algoritmu za jiný
- velikost kryptografických dat ovlivní TLS handshake
 - podpisů a klíčů se přenáší celá sada
 - všechny operace se zvětšují a prodlužují

Certifikát není v TLS sám

- nepřenášíme pouze certifikát serveru
- přenáší se obvykle také:
 - certifikační řetězec
 - veřejné klíče a podpisy jednotlivých certifikátů
 - podpis handshake
 - Signed Certificate Timestamps z Certificate Transparency
- každá vrstva PKI znamená další velký postkvantový podpis
- dnešní systém byl navržen v době, kdy byly podpisy velmi malé

Výměna PQ podpisů nestačí

- když jednoduše vyměníme podpisy za ML-DSA
- certifikační řetězec se několikanásobně zvětší
- celý TLS handshake naroste z kilobajtů na desítky kilobajtů
- plně postkvantový handshake bude znamenat přenos 15 až 30 kB
- problémem není pouze přenosová kapacita
 - zhoršení latence
 - mobilní a pomalé sítě
 - paměť a výpočetní výkon malých zařízení
- dopad závisí na konkrétních podmínkách, největší:
 - mobilních sítích
 - sítích s vysokou latencí
 - zařízeních s omezenou konektivitou
 - službách s velkým počtem krátkých TLS spojení
- bez strukturální změny postkvantová autentizace zhorší výkon

Proč nestačí jen „použít ML-DSA“

- ML-DSA řeší bezpečnost podpisu proti kvantovým počítačům
- neřeší ale spoustu dalších kroků:
 - kolikrát se podpis v handshake objeví
 - jak velký je certifikační řetězec
 - jak se přenášejí podpisy Certificate Transparency
 - jak se aktualizují důvěryhodné informace u klientů
- současná PKI postupně přidávala další podpisy a důkazní informace
- u malých klasických podpisů to nebyl zásadní problém
- u postkvantových podpisů se stejný návrh stává příliš náročným

Co můžeme nasadit už dnes: X25519MLKEM768

- X25519MLKEM768 je hybridní výměna klíčů pro TLS 1.3
- kombinuje dva mechanismy:
 - klasický mechanismus X25519
 - postkvantový mechanismus ML-KEM-768
- výsledný klíč vznikne kombinací obou částí
- pokud je bezpečný alespoň jeden mechanismus, klíč je v bezpečí
- server stále může používat certifikát s ECDSA nebo RSA
- X25519MLKEM768 chrání výměnu klíče a šifrovaný provoz
 - neřeší ještě postkvantovou autentizaci certifikátem
- můžete vyzkoušet na [Portále NÚKIB](#) – vpravo dole hláška



Connection - secure connection settings

The connection to this site is encrypted and authenticated using QUIC, X25519MLKEM768, and AES_128_GCM.

Změna perspektivy

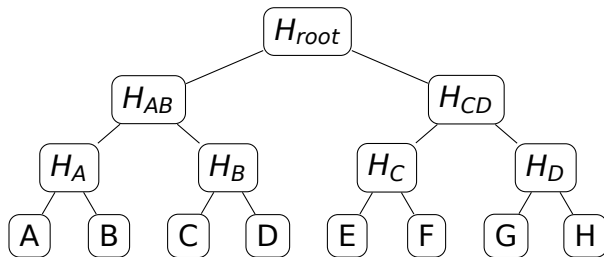
Znovu a lépe

- tradiční přístup:
 - každý certifikát je podepsán samostatně
 - každý podpis se přenáší v rámci handshaku
 - klient ověřuje celý řetězec přímo z dat v handshaku
- efektivnější přístup:
 - podepsat společně větší množství certifikátů
 - podpis distribuovat klientům jinou cestou
 - v handshaku přenášet pouze krátký důkaz příslušnosti
- právě na této myšlence jsou založeny Merkle Tree Certificates

Základní myšlenka MTC

- autorita nevytváří podpis pro každý certifikát zvlášť
- seskupí do dávky a uspořádá je do Merkleova stromu
- každý certifikát tvoří jeden list stromu
- vnitřní uzly jsou hašové hodnoty jednotlivých potomků
- kořen stromu reprezentuje celou dávku certifikátů
- certifikační autorita **podepíše pouze kořen** stromu
- certifikáty mají díky hašům pevnou vazbu
- lze prokázat, že certifikát do daného stromu patří

Merkleův strom certifikátů



- každý list je hašem konkrétního certifikátu
- hašováním dvojic vznikají vyšší úrovně stromu
- jediný kořen zastupuje všechny certifikáty v dávce

Co certifikační autorita podepisuje

- certifikační autorita podepíše pouze hodnotu kořene stromu

$$\text{Sign}_{\text{CA}}(H_{\text{root}})$$

- tento podpis říká:
 - autorita vytvořila daný strom
 - kořen stromu nebyl změněn
 - jednotlivé položky náležejí do daného stromu
 - všechny zahrnuté certifikáty jsou součástí podepsané dávky
- jeden postkvantový podpis tak pokrývá třeba **miliony certifikátů**
- náročnost velkého podpisu se rozloží mezi celou dávku

Důkaz příslušnosti certifikátu

- klient nemusí dostat celý Merkleův strom
- server na začátku komunikace pošle:
 - svůj certifikát
 - haše **sourozeneckých uzlů** na cestě ke kořeni
 - informaci o pořadí spojování uzlů
- pro certifikát A v našem příkladu stačí:

$$H_B, H_{CD}, H_{EFGH}$$

- Klient postupně dopočítá:

$$H(A \parallel H(B)) \longrightarrow H_{AB} \longrightarrow H_{ABCD} \longrightarrow H_{\text{root}}$$

- hodnota H_{AB} se nepřenáší, klient ji sám vypočítá

Jak to bude velké?

- důkaz MTC obsahuje haše sourozeneckých uzlů na cestě
 - od certifikátu ke kořeni Merkleova stromu
- počet hašů roste logaritmicky s počtem certifikátů

$$\log_2(1\,000\,000) \approx 20$$

- pokud jeden haš SHA-256 má 256 bitů = 32 bajtů
- velikost takové sady hašů je tedy přibližně: $20 \times 32 \text{ B} = 640 \text{ B}$
- strom se při každém kroku zdvojuje, roste exponenciálně
 - každý další haš pokrývá dvakrát větší část stromu
- důkaz nemusí obsahovat milion certifikátů
 - ale pouze jednu cestu dlouhou přibližně $\log_2(n)$ kroků
- k tomu se přidají už jen metadata
 - index listu, velikost stromu, údaje o kořeni

Komunikace s novými informacemi

Ověření MTC u klienta

- 1 klient spočítá haš přijatého certifikátu
- 2 pomocí sourozeneckých hašů postupně dopočítá kořen stromu
- 3 výsledný kořen porovná se známým kořenem
- 4 ověří podpis certifikační autority nad tímto kořenem
- 5 pokud podpis platí, certifikát je členem podepsané dávky

certifikát $\longrightarrow H_{\text{root}}$ $\longrightarrow \text{Verify}_{\text{CA}}$

Důkaz pravosti certifikátu

- útočník může změnit obsah certifikátu
- změní se tím však jeho haš
- tím se změní haš každého nadřazeného uzlu
- nakonec tím vznikne jiný kořen stromu
- podpis certifikační autority pak nebude odpovídat
- útočník nemá soukromý klíč CA
 - nemůže vytvořit platný podpis nového kořene
- důkaz byl bezpečně ověřen jako v klasickém schématu
 - jen nemáme individuální podpis, ale hromadný

Kde se vezme podepsaný kořen

- podepsané kořeny stromů se distribuují mimo TLS handshake
- prohlížeč nebo operační systém si je může průběžně aktualizovat
- aktualizace může probíhat například:
 - společně s aktualizací prohlížeče
 - přes mechanismus aktualizace důvěryhodných dat
 - prostřednictvím pravidelné synchronizace
- klient tak může stále vše ověřit offline
 - bez dalšího dotazu na CA (to nechceme)
- důvěra se přesune od jednotlivého podpisu
 - na aktuální stav podepsaného stromu

Co zůstává TLS handshaku

- klient sdělí serveru, které podepsané kořeny stromů zná
 - v praxi budou vznikat v obrovském stromě **podkořeny**
 - landmarks – orientační body podepsané uvnitř stromu
 - zkrátí se tím cesta k nejbližšímu důkazu
- pokud serverový certifikát patří do některého z těchto stromů:
 - server pošle certifikát a svůj podpis
 - pošle důkaz zařazení do hašového stromu
 - neposílá znovu velký podpis certifikační autority
- to je samozřejmě výrazně méně dat
 - než kdybychom opakovali klíče a podpisy

Dva režimy: landmark a standalone

- režim landmark:
 - klient již zná podepsaný kořen příslušné dávky
 - v handshaku stačí certifikát a Merkleův důkaz
 - velký podpis byl doručen dříve a mimo handshake
 - ideální stav – všechno už máme, jsme synchronní
- režim standalone:
 - klient nemá aktuální kořen stromu (landmarks)
 - server musí přibalit i údaje umožňující ověření (podpis MTCA)
 - handshake je větší, musí obsahovat chybějící informace
 - funguje ale i s méně aktuálním klientem
- server může obvykle počkat na zařazení do dávky
 - pokud potřebuje certifikát okamžitě, vyžádá si standalone
- landmark je optimalizace, standalone je záložní varianta

Vestavěná otevřenost

Dnešní model Certificate Transparency

- Certificate Transparency (CT) odděluje dvě činnosti
 - certifikační autorita vydá certifikát
 - nezávislý CT log potvrdí jeho zveřejnění
- CA pošle certifikát do jednoho nebo více CT logů
- Log vrátí Signed Certificate Timestamp (SCT)
- SCT je příslib logu, že certifikát bude vložen do veřejného stromu
- server pak předá SCT klientovi společně s certifikátem
- klient má důkaz o zveřejnění certifikátu
 - bez něj je certifikát neplatný
 - nutí autority vše zveřejňovat

Kdo v CT co dělá

- Certifikační autorita
 - vydává certifikáty
 - posílá je do logů CT
- CT log
 - provozuje veřejný log (Merkleův strom)
 - potvrzuje přijetí certifikátu pomocí SCT
- monitory
 - sledují logy
 - hledají podezřelé nebo neoprávněné certifikáty
- auditoři
 - kontrolují konzistenci logů
 - hlídají neměnnost minulosti
 - hlídají stejné pohledy z různých stran

Nevýhoda dnešního CT pro postkvantovou dobu

- CT je k vydávání certifikátů přidáno jako samostatná vrstva
- jeden certifikát může obsahovat:
 - podpis CA
 - jeden nebo více SCT
 - veřejné klíče a další údaje řetězce
- každý další podpis zvětšuje režii
- při přímém použití PQ podpisů by se zvětšil:
 - certifikát
 - TLS handshake
 - infrastruktura CT logů
 - práce klienta při ověřování

MTC: transparentnost jako součást vydávání

- u MTC provozuje certifikační autorita vlastní log
- každý vydaný certifikát se stává listem v tomto logu
- certifikát tedy nevzniká mimo transparentní log
 - kdy se do něj teprve dodatečně zapisuje
- vydání certifikátu a jeho zveřejnění jsou spojeny
- CT se tím stává základní vlastností PKI
 - nikoliv jen jeho dodatečným rozšířením

Co nahrazuje SCT

- v klasickém modelu server přenáší SCT podepsaný CT logem
- v MTC je důkazem zveřejnění:
 - certifikát jako položka issuance logu
 - Merkleův důkaz příslušnosti
 - podpis vydaný certifikační autoritou
- klient ověří, že certifikát patří do veřejného stromu CA
- samostatné SCT podpisy už není nutné přidávat do handshaku
- odpadá tak část podpisové režie
- principem MTC je vždy zařazení do stromu
 - důkaz o zařazení slouží k ověření
 - zároveň jako důkaz o zveřejnění

Experimenty a plány

Kdy se postkvantové certifikáty objeví?

- u výměny klíčů už přechod probíhá:
 - prohlížeče podporují hybridní TLS
 - typicky kombinaci X25519 a ML-KEM
 - některé velké služby jej používají v produkci
- postkvantová autentizace serveru je obtížnější
- nestačí změnit algoritmus podpisu v certifikátu
- je nutná změna Web PKI:
 - formát certifikátu
 - Certificate Transparency
 - TLS handshake
 - prohlížeče a jejich úložiště
 - infrastruktura CA a změny ACME
- první experiment můžeme očekávat v roce 2027

Slepice nebo vejce?

- obě strany musejí postupovat současně: autority i prohlížeče
- certifikační autorita může začít s experimentálními certifikáty
- prohlížeč je ale musí umět:
 - rozpoznat
 - ověřit
 - svázat s důvěryhodným landmarkem
 - bezpečně zpracovat při TLS handshaku
- bez podpory prohlížečů nemá smysl vydávat certifikáty
- bez CA zase prohlížeče nemají co testovat
- prvním krokem proto není plošné vydávání
 - potřebujeme koordinovaný experiment: CA, prohlížeče a logy

Co se děje už dnes

- MTC už není pouze akademický návrh
- Cloudflare a Chrome provádějí experiment s reálným provozem
- experiment ověřuje například:
 - velikost handshaku
 - latenci a chybovost
 - chování různých síťových cest
 - aktualizaci landmarků
 - kompatibilitu s existujícími klienty
- experimentální MTC je současně chráněno klasickým certifikátem
- zatím nejde o plošné nahrazení dnešních certifikátů

Plán Cloudflare a Chrome

- fáze 1: probíhá
 - experiment proveditelnosti – Chrome s Cloudflarem
 - MTC se testují proti reálnému provozu
 - klasický X.509 certifikát slouží jako záchranná varianta
- fáze 2: první čtvrtletí 2027
 - bootstrap veřejných MTC
 - zapojení vybraných provozovatelů CT logů
 - ověření provozní dostupnosti a konzistence infrastruktury
- fáze 3: třetí čtvrtletí 2027
 - požadavky pro zapojení dalších CA
 - možnost postupného nasazení na webech
 - vznik Chrome Quantum-resistant Root Store

Chrome Quantum-resistant Root Store

- Chrome neplánuje přidat PQ podpisy do dnešního úložiště
- důvodem je především velikost a režie tradičního řetězce:
 - veřejné klíče
 - podpisy CA
 - podpisy CT
 - mezilehlé certifikáty
- místo toho Chrome připravuje oddělený trust store pro MTC
- bude mít zřejmě jiné:
 - formáty
 - aktualizací mechanismy
 - požadavky na landmarky a logy
 - bezpečnostní politiky
- obě úložiště budou po určitou dobu fungovat vedle sebe

Plán Let's Encrypt

- Let's Encrypt považuje MTC za hlavní cestu k postkvantovému PKI
- plánované milníky
 - konec roku 2026: testovací (staging) prostředí schopné vydávat MTC
 - rok 2027: produkčně připravené prostředí
- nejde pouze o přidání nového podpisového algoritmu
- bude nutné upravit:
 - vydávací infrastrukturu
 - protokol ACME
 - revokaci
 - provozní nástroje
 - transparency logy

Co to znamená pro provozovatele webu

- v nejbližší době není potřeba měnit certifikáty
- připravte se ale na hybridní postkvantovou výměnu klíčů
- provozovatelé už teď mohou nasazovat a hlídat:
 - podporu X25519MLKEM768 na serveru
 - aktualizace šifrovacích knihoven a nástrojů
 - podporu v reverzních proxy a load balancerech
 - aktualizace ACME klientů
- MTC certifikáty budou vyžadovat změny na straně CA i klienta
- běžný web se k nim pravděpodobně dostane přes ACME
 - stejně jako dnes k certifikátům Let's Encrypt

Postkvantová Web PKI v kostce

- kvantově odolné podpisy jsou výrazně větší
- přímé nahrazení algoritmů by zvětšilo certifikáty i TLS handshake
- MTC přesouvají velký podpis z jednotlivého certifikátu na celý strom
- v handshaku se potom přenáší pouze:
 - klasický serverový certifikát
 - krátký Merkleův důkaz o zařazení
 - vazba na známý a podepsaný landmark
- Certificate Transparency zůstává zachována
- první experimenty už probíhají
- potřebujeme podporu prohlížečů, CA i serverové infrastruktury
- chceme zajistit postkvantovou bezpečnost a nerozbít web

Otázky?



Petr Krčmář
petr.krcmar@iinfo.cz