Infraštruktúra verejného kľúča

Branislav Trstenský

1 Úvod

Zámer projektu je poskytnúť pohľad na infraštruktúru verejného kľúča (PKI), ktorá je základom modernej informačnej bezpečnosti na internete. Projekt je rozdelený na dve hlavné časti: analytickú a experimentálnu.

Analytická časť projektu sa zameriava na teoretické základy PKI a jeho využitie v praxi. Súčasťou analýzi sú prinípy asymetrickej kryptografie, úvod do PKI a konkrétne do štandardu X.509. Bude tiež vysvetlená funkcionalita certificate transparency logs a ako prehliadače overujú identity poskytovateľov služieb.

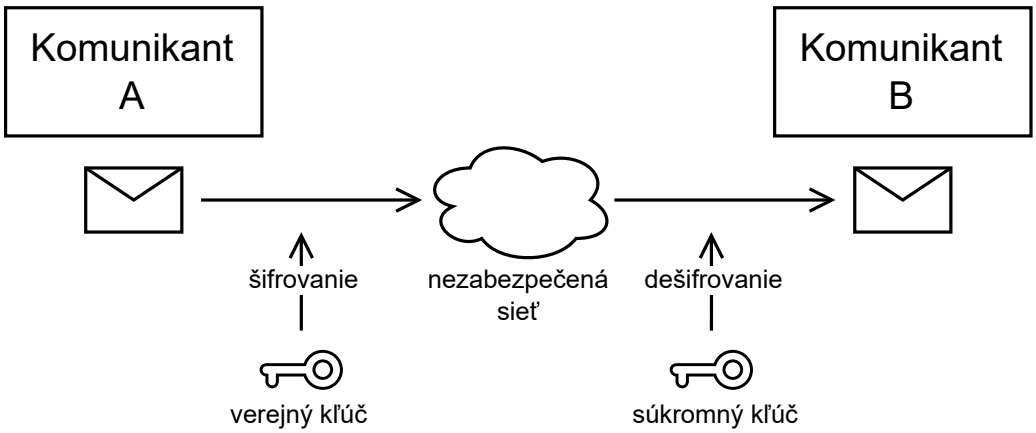
Experimentálna časť projektu sa zameriava na vytvorenie zjednodušenej infraštruktúry pre vydanie a overenie verejného kľúča a dosiahnutie zabezpečenej komunikácie medzi klientom a serverom. Na dosiahnutie tohto cieľa bude použitá sada nástrojov OpenSSL a programové prostredie Node.js. Protokol získania certifikátu bude zjednodušenou podobou ACME bez možnosti zrušenia a obnovenia platnosti certifikátu.

2 Kryptografia

Internet umožňuje ľuďom na diaľku vykonávať rôzne aktivity, ktoré vyžadujú dôvernosť, ako napríklad internetové bankovanie, e-shopping alebo aj medziľudskú komunikáciu. Dáta prechádzajúce internetom sú však čitateľné veľkým množstvom sprostredkovateľov, ktorý môžu tieto informácie využiť k svojmu benefitu.

Na riešenie tohto problému sa používa šifrovanie. Keďže pri internetovej komunikácií sa očakáva že pre každý člen komunikácie, t.j. klient a sever, nie je možná výmena údajov mimo internetu. Preto nie je možné bezpečne rozšíriť jeden kľúč potrebný pre šifrovanie, keďže by ho bolo možné zachytiť počas presunu cez sieť.

Na riešenie tohto problému sa používa asymetrické šifrovanie. Pri tejto metóde sa na požíva voľne šíriteľný verejný kľúč, ktorý následne nie je schopný dešifrovať, a súkromný kľúč, prostredníctvom ktorého je možné správy dešifrovať. Pri inicializácií šifrovanej komunikácií si komunikanti navzájom vymenia svoje verejné kľúče, čo umožní každému pred odoslaním šifrovať správy.



Obr. 1: Asymetrická kryptografia

Tento princíp umožňuje aj vytvorenie takzvaného digitálneho podpisu. Vytvorenie podpisu k správe jednoznačne potvrdzuje, že správu vytvoril držiteľ súkromného kľúča. Tento proces je opačný ako pri šifrovaní. Najprv je na správu aplikovaná kryptografická hašovacia funkcia a výsledný haš je zakódovaný súkromným kľúčom.

Pre verifikáciu podpisu klient aplikuje na správu rovnakú hašovaciu funkciu a súčasne dešifruje podpis verejným kľúčom druhého člena komunikácie. Ak sú výsledky oboch operácií rovnaké, identity autora správy je overená.

Digitálny podpis sa používa nielen na správy, ale aj uložené dáta, napríklad dokumentov, certifikátov alebo aplikácií. [9]

3 Infraštruktúra verejného kľúča

Pri inicializácií šifrovanej komunikácií si komunikanti navzájom vymenia svoje verejné kľúče.Keby ale škodlivý aktor prerušil komunikáciu medzi dvoma subjektmi, mohol by podstrčiť svoj vlastný verejný kľúč. Správy by potom bolo možné dešifrovať jeho súkromným kľúčom čiže by mal prístup k dôverným informáciám.

Vyžaduje sa teda spôsob, akým by bolo možné jednoznačne overiť či daný verejný kľúč naozaj patrí subjektu, s ktorým chceme komunikovať a taktiež zachoval možnosť šíriť verejný kľúč každému subjektu so záujmom o komunikáciu.

Jedným zo spôsobov, ako vyriešiť tento problém, je overiť verejný kľúč mimo siete. Toto pri internetovej komunikácií nie je praktické, keďže sa očakáva že jediné spojenie medzi komunikantmi je priamo internet. Napriek tomu existujú použitia, kde je tento prístup použiteľný, napríklad pri spojení SSH, kde overenie kľúča má na starosti používateľ.

Druhou možnosťou je vytvorenie registra, ktorý jednoznačne priraďuje verejný kľúč k subjektu. Klient sa teda buď pripojí k registru pri každej potrebe pre verifikáciu kľúča alebo si každý klient udržiava lokálnu kópiu registra. Tento prístup je tiež problematický, keďže klient vyžaduje overenie kľúča pri každom pripojení k akejkoľvek webovej stránke, registre by čelili obrovskému množstvu požiadaviek. Tiež, pri ich zlyhaniu, by internetová komunikácia bola znemožnená. Tiež by nebolo možné vytvoriť lokálnu kópiu registra, keďže nie je možné očakávať, aby každý klient mal záznam o každej webovej stránke na celom internete. Tento prístup je však možne použít pri menších sieťach, napríklad vpodnikovýchintranetoch.

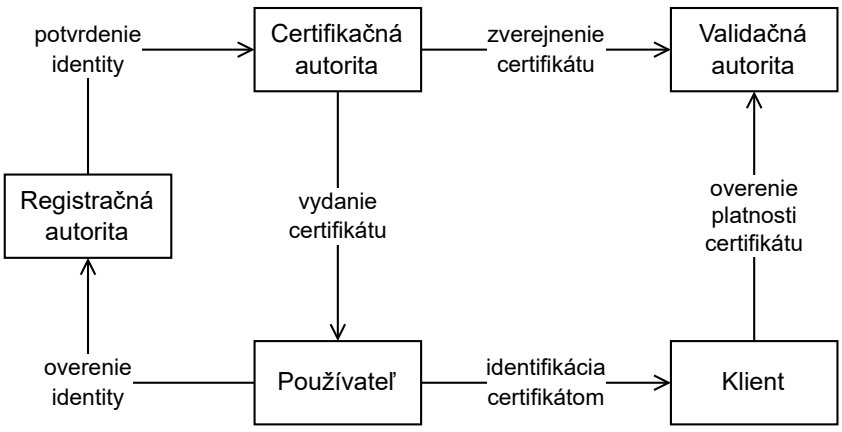
Riešenie, ktoré je dnes použité v praxi je infraštruktúra verejného kľúča, ktorá tvorí kombináciu obidvoch predošlých metód. Klient má na svojom zariadení uložený len malý počet dôveryhodných verejných kľúčov. Pri nadviazaní spojenia, poskytovateľ služby pošle svoj verejný kľúč a platnosť daného kľúča je overená prostredníctvom dôveryhodných kľúčov.

3.1 X.509

Konkrétny štandard, riadiacy infraštruktúru verejného kľúča na internete je X.509. Tento štandard je pomerne starý a rozsiahly. Obsahuje procedúry pre veľmi širokú škálu problémov, ktoré nie je relevantné pre dnešné aplikácie [8]. Verzia súčasne používaná tvorí nadstavbu ale aj podmnožinu funkcionality pôvodného štandardu.

Štandard pozostáva z nasledovných súčastí:

- Používateľ, ktorý sa chce identifikovať
- Certifikát, ktorý potvrdzuje identitu používateľa
- Certifikačná autorita, ktorá vydáva certifikát používateľovi
- Validačná autorita, ktorá overuje či certifikát nebol zneplatnený
- Klient, ktorý si chce overiť identitu používateľa



Obr. 2: Vzťahy súčastí X.509

Digitálneho certifikát obsahuje verejný kľúč spolu s informáciami o vlastníkovi daného kľúča a parametre, ktoré určujú, v akých prípadoch je možné kľúč použiť. Certifikát je prostredníctvom digitálneho podpisu jednoznačne viazaný s certifikačnou autoritou ktorá ho vydala, takže pre overenie identity stačí aby klient mal záznamy iba o certifikačných autoritách, nie o všetkých certifikátoch. [2]

3.2 Certifikačné autority

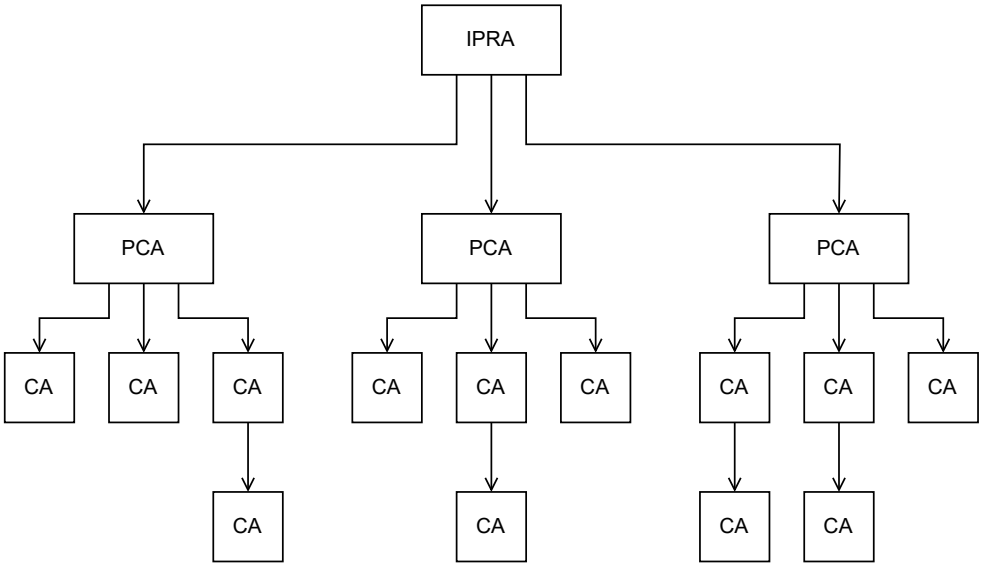
Vydávaniecertifikátupodpísaného jedným z certifikátov globálneho reťazca majú na starosti takzvané certifikačné autority. Toto sú organizácie, ktorým bol vydaný certifikát so schopnosťou podpisovania certifikátov.

Medzi tieto organizácie patria spoločnosti ako GlobalSign, Sectigo, DigiCert alebo GoDaddy, ale najväčší percentuálny podiel má nezisková organizácia Let's Encrypt. [21]

Keďže certifikáty potvrdzujú identity používateľa, certifikačné autority majú povinnosť si identitu používateľa overiť. Toto overenie buď vykonávajú samé, alebo túto úlohu delegujú na takzvanú registračnú autoritu. Overovanie identity je jedinou úlohou registračných autorít, nemajú právo vytvoriť certifikát. [2]

Všetky certifikačné autoritu majú danú hierachiu, kde nadradené autority certifikujú podradené autority a umožňujú im vydávať certifikáty. Stupne v tejto hierarchií sú nasledovné:

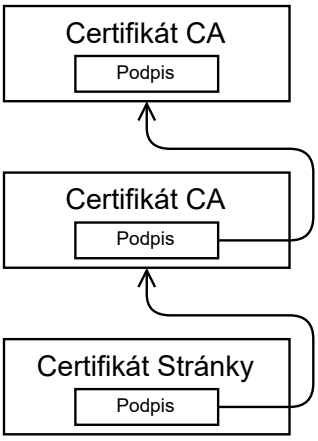
1. Internet Policy Registration Authority (IPRA) ⇒ najvyššia autorita, prevádzkovaná pod záštitou Internet Society (ISOC), certifikuje iba PCA
2. Policy Certification Authorities (PCA) ⇒ všetky autority certifikované od IPRA
3. Certification Authorities (CA) ⇒ všetky autority certifikované od PCA alebo od iných CA, tvoria tretí aj všetky nižšie stupne hierarchie



Obr. 3: Hierarchia certifikačných autorít

3.3 Reťazec dôvery

Certifikáty tvoria takzvaný reťazec dôvery. Ten tvorí hierarchickú štruktúru certifikátov, kde každý digitálny certifikát obsahuje podpis generovaný pomocou súkromného kľúča certifikátu nadradeného certifikátu. Klient si pre overenie certifikátu nájde v svojom úložisku nadradený certifikát a overí platnosť podpisu.



Obr. 4: Reťazec dôvery

Je možné nadradený certifikát nie je prítomný na zariadení klienta aj keď sú certifikáty vyššie na reťazci prítomné, čím sa vytvorí medzera, ktorá znemožňuje úspešné overenie. Preto okrem svojho certifikátu

poskytovateľ služby môže poslať aj všetky nadradené certifikáty. Pre overenie platnosti certifikátu klient overuje platnosť certifikátu postupne, pozdĺž reťazca až k dôveryhodnému certifikátu, ktorý má uložený v svojom úložisku.

Výnimkou tohto princípu je koreňový certifikát. Tento nemá nadriadený certifikát a obsahuje podpis generovaný zo svojho vlastného súkromného kľúča, takzvaný certifikát s vlastným podpisom. Pre celosvetovú infraštruktúru verejného kľúča je definovaný jeden certifikát, z ktorého sú odvodené všetky certifikáty použité na otvorenom internete.

Certifikát má takzvané extensions, ktoré určujú na čo je ho možné použiť. Na to aby certifikát mohol podpísať podradený certifikát, potrebuje mať nastavený parameter CA na true v extension X509v3 Basic Constraints. Keďže parametre certifikátu nie je možné zmeniť po podpísaní, a autoritynevydávajúcertifikáty s daným parametrom, certifikačné autority sú jediným zdrojom globálne platným certifikátom.

Pre osobné použitie alebo pre použitie vpodnikovýchintranetoch je pre každého možné vytvoriť si certifikát s vlastným podpisom. Takto je možné overovať identitu zariadení a nadviazať bezpečnú komunikáciu bez potreby tretej strany. V tomto prípade je potrebné manuálne tento certifikátnainštalovaťdo úložiska zariadenia. [2]

Tiež niektoré aplikácie alebo zariadenia, ktoré sú určené iba na komunikáciu s konkrétnou webovou službou, majú presne definovaný zoznam certifikátov, ktoré môžu akceptovať. Táto metóda je nazývaná certificate pinning. Takto sa chránia pred útokmy, keď má útočník možnosť zapísať certifikát do úložiska zariadenia, čo by mu za normálnych podmienok umožnilo odpočúvať a modifikovať komunikáciu zo zariadenia. [20]

3.4 Certificate signing request

Aby poskytovateľ webovej služby získal od CA certifikát, musí si najprv sám vytvoriť súkromný a verejný kľúč, ktorý bude používať na šifrovanie komunikácie. Následne musí vytvoriť takzvanú požiadavku o podpísanie certifikátu, (certificate signing request, CSR).

V štandarde X.509 táto požiadavka obsahuje pole SubjectName, ktoré definuje identitu vlastníka certifikátu. Toto pole pozostáva z viacero častí označeným kódmy. [10]

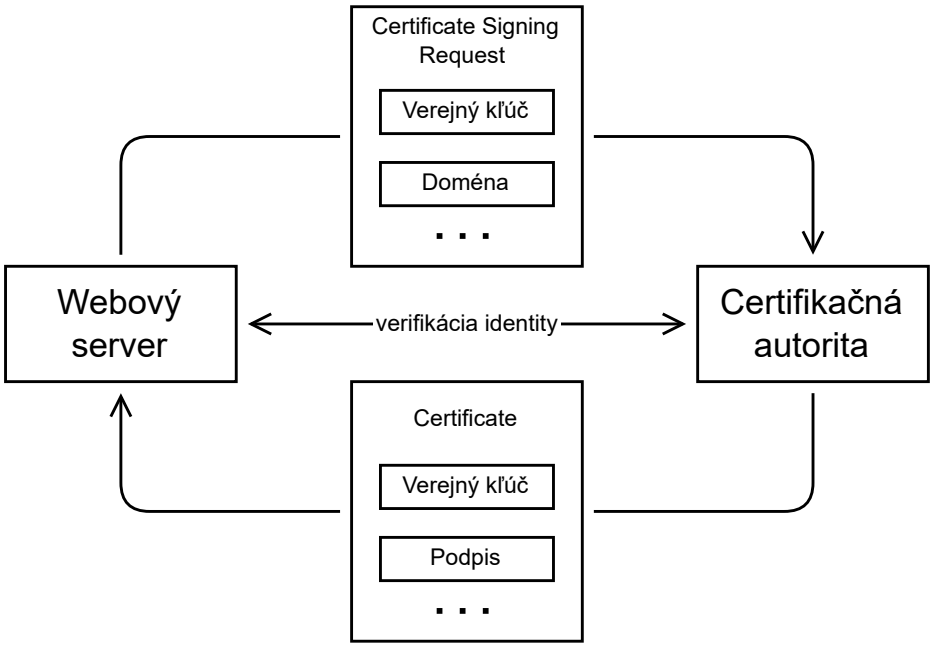
- L (lokalita)
- ST (štát alebo provincia)
- O (organizácia)
- OU (organizačná jednotka)
- CN (bežné meno, v kontexte webových stránok je to doména stránky)
- E (e-mailová adresa)

Ktoré polia sú povinné alebo vôbec povolené sa líši podľa toho aký typ certifikátu je žiadaný. V praxi existujú tri stupne certifikátov, kde každý silnejšie potvrdzuje identitu vlastníka, ale aj zvyšuje cenu o obtiažnosť získania certifikátu. [16] [5]

- DV (Domain Validation) ⇒ verifikuje iba že vlastník certifikátu vlastní danú doménu
- OV (Organizational Validation) ⇒ verifikuje aj ostatné prvky identity
- EV (Extended Validation) ⇒ z legálneho hľadiska potvrdzuje identitu subjektu

Ďalším prvkom je SAN (Subject Alternative Name). V kontexte webových stránok priraďuje DNS záznamy k vlastníkovi certifikátu. Jeden certifikát môže obsahovať viacero položiek SAN, čiže môže overovať viacero DNS domén. [17]

Po vložení týchto informácií je do CSR zapísaný verejný kľúč webovej služby a je podpísaný pomocou súkromného kľúča. CSR je potom odoslaný certifikačnej autorite.



Obr. 5: Proces vydania certifikátu

Certifikačná autorita má následne povinnosť si overiť či žiadateľ o certifikát má naozaj identitu, ktorú si žiada. Tento proces nie je štandardizovaný, líši sa medzi CA aj medzi typmi žiadaného certifikátu. Napriklad pri DV môže autorita poslať email na doménu spojenú s certifikátom alebo vyžadovať určitý zápis do DNS záznamu webovej stránky. [16] [12]

Overenie identity môže tiež certifikačná autorita delegovať na takzvanú registračnú autoritu, ktorej jediná úloha je overenie identity. [6]

Certifikačná autorita po overení identity vráti sebou podpísaný certifikát žiadateľovi.

3.5 Certificate transparency

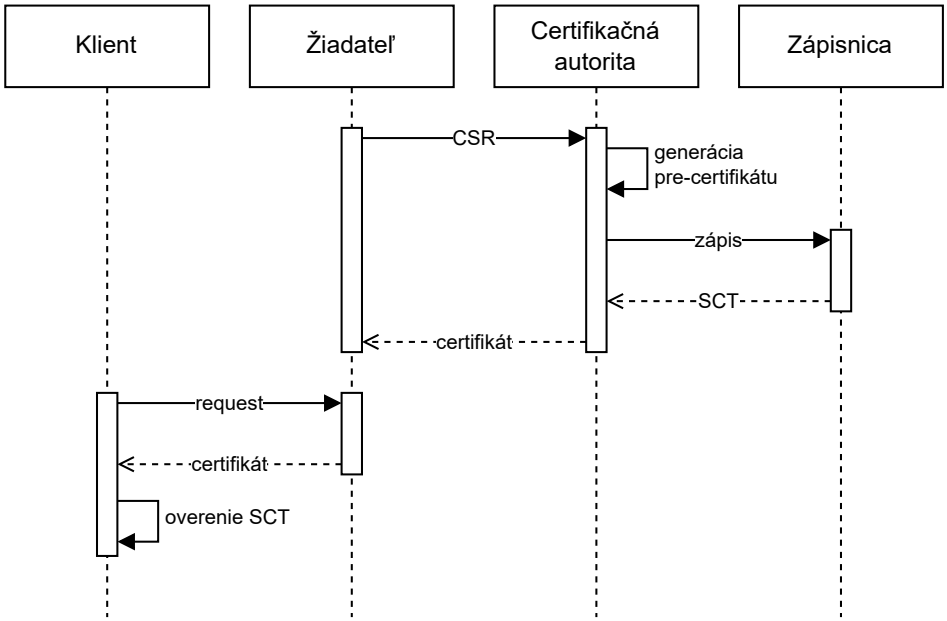
Napriek všetkým opatreniam pre overenie identity žiadateľa certifikátu stále existuje možnosť chybného vydania certifikátu alebo dokonca kompromitovanie certifikačnej autority. Chybne vydaný certifikát umožníútočníkovyobísť všetky bezpečnostné prvky PKI.

Na riešenie tohto problému sa používa štandard prehľadnosti certifikátov (Certificate Transparency). Všetky vytvorené certifikáty od oficiálnych certifikačných autorít sú nahlásené do zápisníc (logs). Zápisnice sú prevádzkované certifikačnými autoritami ale aj výrobcami prehliadačov. [3]

Priebeh procesu certificate transparency je nasledovný:

1. CA najprv vytvorí pre-certifikát, ktorý nie je platný ale je obsahuje všetky informácie ktoré vysledný certifikát bude mať
2. Pre-certifikát je zapísaný do zápisníc
3. Zápisnice počas zapisovania vytvoria podpísanú časovú pečiatku (SCT), potvrdzujúcu toto zapísanie
4. SCT je vrátené CA, ktorá túto pečiatku zahrnie do výsledného certifikátu

Keď prehliadač navštívi stránku, skontroluje či certifikát obsahuje SCT a overí si jej podpis. [11]



Obr. 6: Proces certificate transparency

Vďaka tomuto systému prevádzkovatelia webových stránok môžu kontrolovať všetky vydané certifikáty a byť teda upovedomení ak je vydaný certifikát pre ich stránku, ktorý si nevyžiadali. Túto kontrolu väčšinounevykonávajúmanuálne ale používajú kontrolné služby poskytované certifikačnými autoritami. [4]

Tento proces má tiež však nevýhody. Zverejnenie všetkých domén v jednoducho prístupnom zozname poskytuje útočníkovy zdroj potenciálnych cieľov. Zverejnené sú tiež domény, ktoré sú určené len na vnútorné použitie. [18]

3.6 Zneplatnenie certifikátov

Certifikáty sú vydané len na určitý čas, po ktorom stratia svoju platnosť a musia byť opätovne vyžiadané. Ak je však poskytovateľ webovej služby alebo certifikačná autorita kompromitovaná a útočník získa možnosť použiť certifikát na svoje účely, je potrebný proces na zrušenie platnosti certifikátu (Certificate Revocation).

Ak majiteľ certifikátu zistí že bol jeho certifikát kompromitovaný pošle na certifikačnú autoritu, ktorá daný certifikát vydala žiadosť o zneplatnenie. Tá musí distribuovať správu o zrušení všetkým klientom webovej služby.

Jedným so spôsobov informovaní klientov o zrušení sú takzvané zoznamy zneplatnených certifikátov (Certificate Revocation List, CRL). Tieto sú pravidelne vydané samotnou certifikačnou autoritou alebo dôveryhodnou treťou stranou, a majú určitú platnosť, po ktorej musí klient prevziať novú verziu CRL. [8]

Existuje tiež novšia metóda Online Certificate Status Protocol (OCSP), kde sa pri spojení klienta s webovou stránkou, klient pošle OCSP požiadavku na certifikačnú autoritu, ktorá vydala certifikát stránky, alebo iný dôveryhodný OCSP odpovedač. [14]

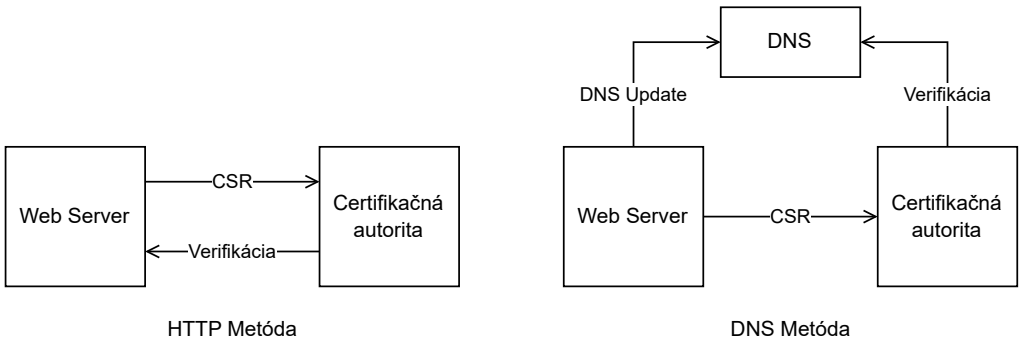
Obidva z týchto metód predstavujú problémy. Pri znemožnení komunikácie s certifikačnou autoritou alebo OCSP odpovedačom buď klienti stratia možnosť navštíviť akúkoľvek stránku alebo sa vystavia riziku navštívenia kompromitovanej stránky. Pri CRL, tiež vyžadujú veľký prenos dát, celý zoznam zneplatnených certifikátov. Pri OCSP tiež klient informuje tretiu stranu o každej navštívenej stránke.

Z týchto dôvodov prehliadače upúšťajú od týchto metód. Napríklad Firefox a Google Chrome používajú proprietárne protokoly [19] [15].

3.7 ACME Protokol

ACME (Automatic Certificate Management Environment) je moderný protokol pre automatické vydávanie certifikátov a overenie identity. Bol vytvorený pre použitie s certifikačnou autoritou Let's Encrypt [13]ale je podporovaný aj inými CA, ako napríklad Google Trust Services [7].

Tento protokol ponúka úplne automatickú operáciu, po prvotnej konfigurácií používateľom už sám manažuje verifikáciu identity a obnovenie certifikátu pre jeho vypršaním. Toto je umožnené prostredníctvom rôznych metód overenia identity, ktoré po prvotnej konfigurácií nepotrebujú zásah používateľa. ACME podporuje viacero metód:



Obr. 7: Metódy verifikácie identity

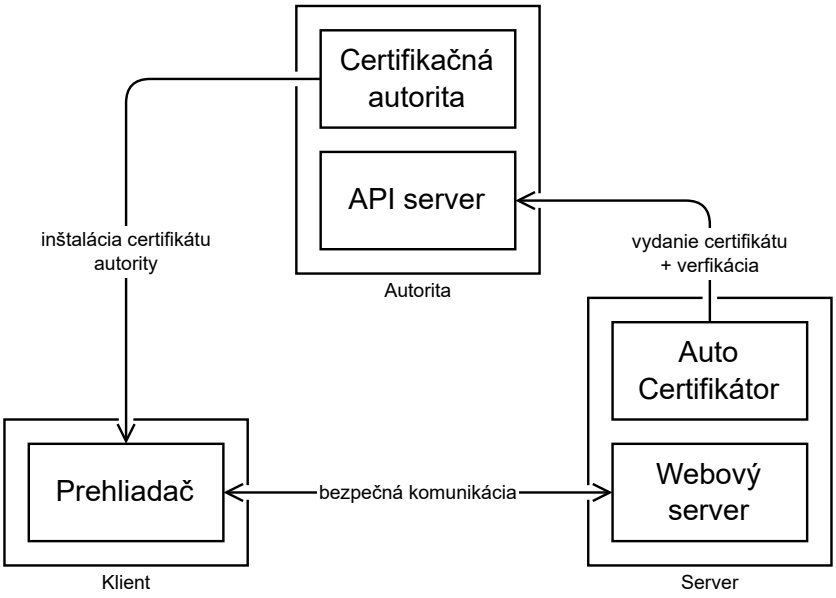
HTTP Challenge Pri tejto metóde je verifikované vlastníctvo HTTP servera. Po vyžiadaní certifikátu klientom server CA vráti kľúč (challenge key). Tento kľúč potom musí klient sprístupniť na webovej stránke na špecifikovanom endpointe. Táto metóda je najviac používaná.

DNS Challenge Pri tejto metóde je verifikované vlastníctvo DNS záznamu domény tak, že challenge key je zapísaný do záznamu TXT pod danou doménou. Nevýhodou je že pre automatické vykonanie verifikácie, je potrebné, aby mal poskytovateľ domény API, prostredníctvom ktorého je možné TXT záznamy automaticky meniť a potreba čakať kým sa upravený DNS záznam rozšíri do siete. Tiež kvôli tomu, že je potrebné na serveri uložiť autentifikačné parametre do DNS, je vytvorené riziko ich odcudzenia. [1]

4 Experimentálna časť

Experimentálna časť tohto project spočíva vo vytvorení vlastnej certifikačnej autority. Daná autorita disponuje REST API serverom, prostredníctvom ktorého je klient schopný vyžiadať si certifikát prostredníctvom CSR a overiť svoju identitu. Ďalej je vytvorený REST klient, ktorý automatizuje proces interakcie s danou certifikačnou autoritou.

Súčasťou projektu je aj testovanie vytvorených programov a overenie ich funckionality. Pre webový server bude vyžiadaný certifikát a na daný webový server bude pripojení používateľ prostredníctvom webového prehliadača. Bude overené či prehliadač príjme zo servera správny certifikát, a po nainštalovaní certifikátu autority, bude overené či prehliadač vidí danú webovú stránku ako vierohodnú.



Obr. 8: Prehľad projektu

4.1 Špecifikácia projektu

Pre vytvorené programy platia nasledovné požiadavky:

Plne automatický proces Po konfigurácií REST klienta, celý proces musí prebehnúť bez zásahu používateľa. Toto umožní obnovenie certifikátu pred jeho vypršaním, nastavením automatického spustenia klienta v danom intervale, napríklad cez cron.

Overenie identity Server certifikačnej autoritu musí jednozančne overiť že žiadateľ je vlastníkom webového servera, pre ktorého doménu je certifikát žiadaný. Overenie bude prebiehať cez zjednodušenú verziu HTTP challenge z ACME protokolu. Server autority vygeneruje kód a cestu, na ktorej musí byť sprístupnený. Po tom ako klient potvrdí, že je kód sprístupnený, server autority tento fakt overí. Pri probléme s overením identity server vráti chybovú hlášku, ktorá pomôže tento problém vyriešiť.

Nezávislý HTTP web server Program klienta musí fungovať so akýmkoľvek štandardným webovým serverom. Jediné požiadavky na webový server je podpora TLS certifikátov a čítanie statických súborov so súborového systému.

REST komunikácia Komunikácia medzi klientom a serverom autority musí prebiehať cez REST. REST je postavený nad HTTP, čo umožňuje jeho použitie v sieťových prostrediach, kde by neštandardné

protokoly mohli byť blokované. Tiež dáva možnosť použitia HTTPS, čo garantuje že komunikácia bude súkromná.

Konkrétne zodpovednosti REST klienta sú nasledovné:

- Generácia RSA kľúčov ⇒ ak nie sú predom vytvorené, klient vytvorí súkromný a verejný kľúč
- Generácia CSR ⇒ klient vytvorí CSR podľa konfigurovaného názvu domény
- Manažment súborov ⇒ klient uloží súbor s challenge kódom na správne miesto a po vytvorení uloží certifikát na konfigurované miesto a odstráni dočasné súbory

Zodpovednosti servera autority sú nasledovné:

- Generácia RSA kľúčov ⇒ server vytvorí súkromný a verejný kľúč autority
- Generácia certifikátu CA ⇒ server vytvorí sebou-podpísaný certifikát autority
- Overenie identity ⇒ server overí, že žiadateľ o certifikát vlastní doménu špecifikovanú v CSR
- Generácia certifikátu z CSR ⇒ server vytvorí podpísaný certifikát z CSR a vráti ho klinetovy

Vytvorené programy fungujú na prostredí Node.js, čo umožňuje ich využitie na väčšine operačných systémoch. Kryptografické operácie budú vykonané s pomocou knižnice OpenSSL.

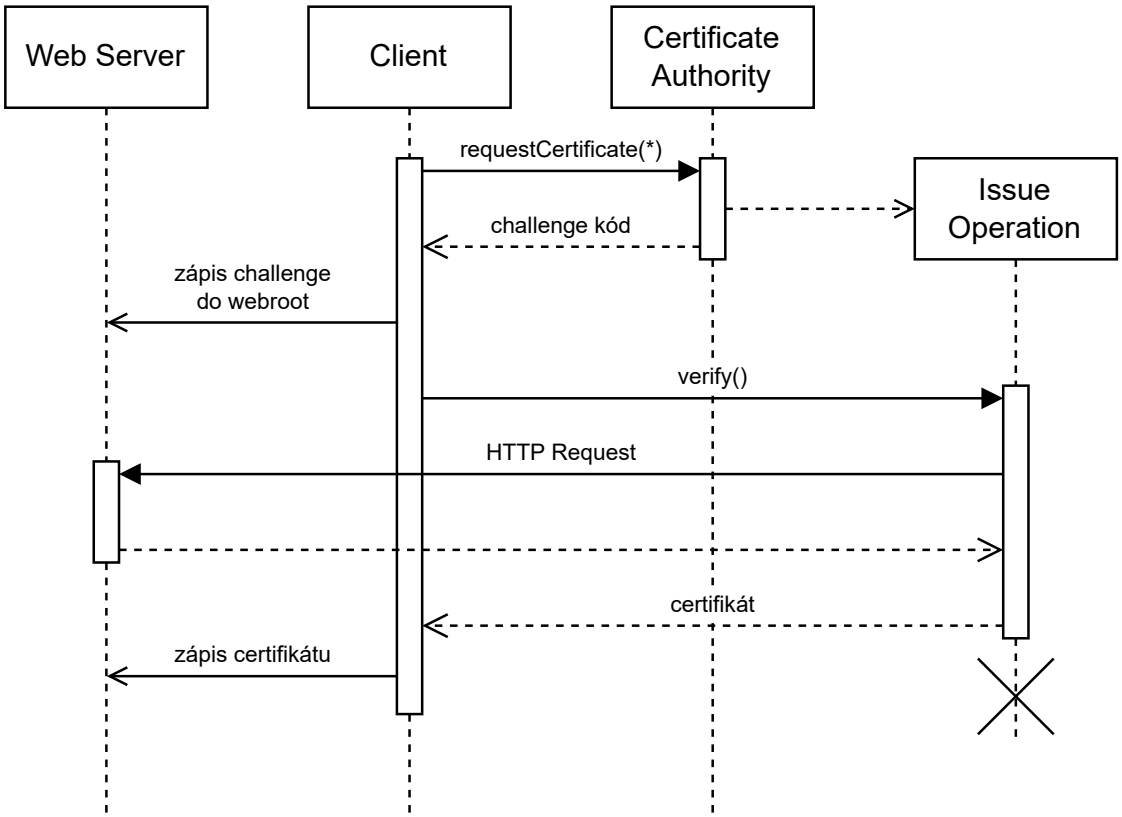
4.2 Detaily procesu vydania certifikátu

Server autority poskytuje tri klientovi tri rôzne volania:

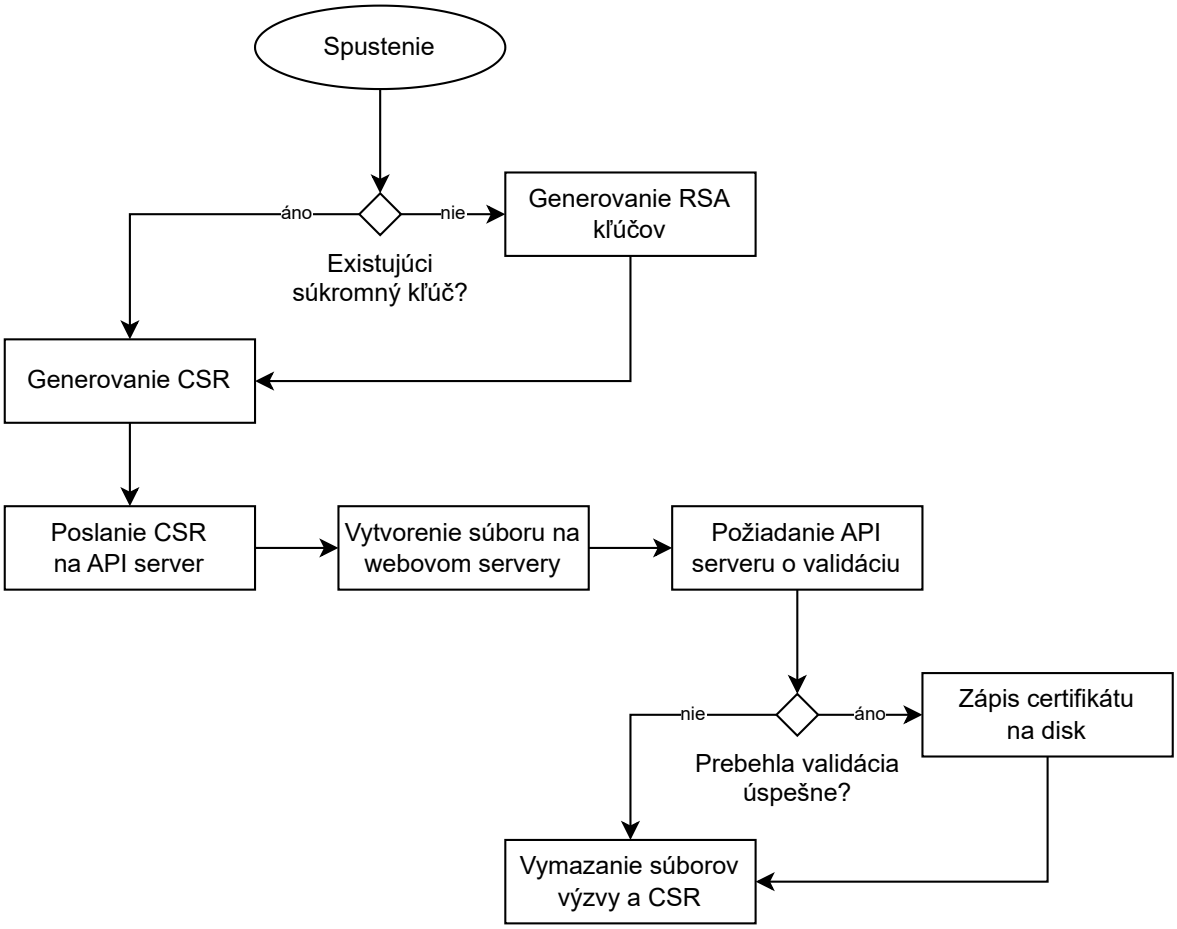
- `/certificate-authority/request-certificate`
- `/issue-operation/:id/verify`
- `/issue-operation/:id/cancel`

Prvým krokom klienta je volanie na `/certificate-authority/request-certificate`, kde pošle CSR a bude mu vrátený challenge kód a cestu, kde ho má sprístupniť a identifikačný kód, pre ďalšiu komunikáciu so serverom. Súbežne si server CSR uloží do pamäte.

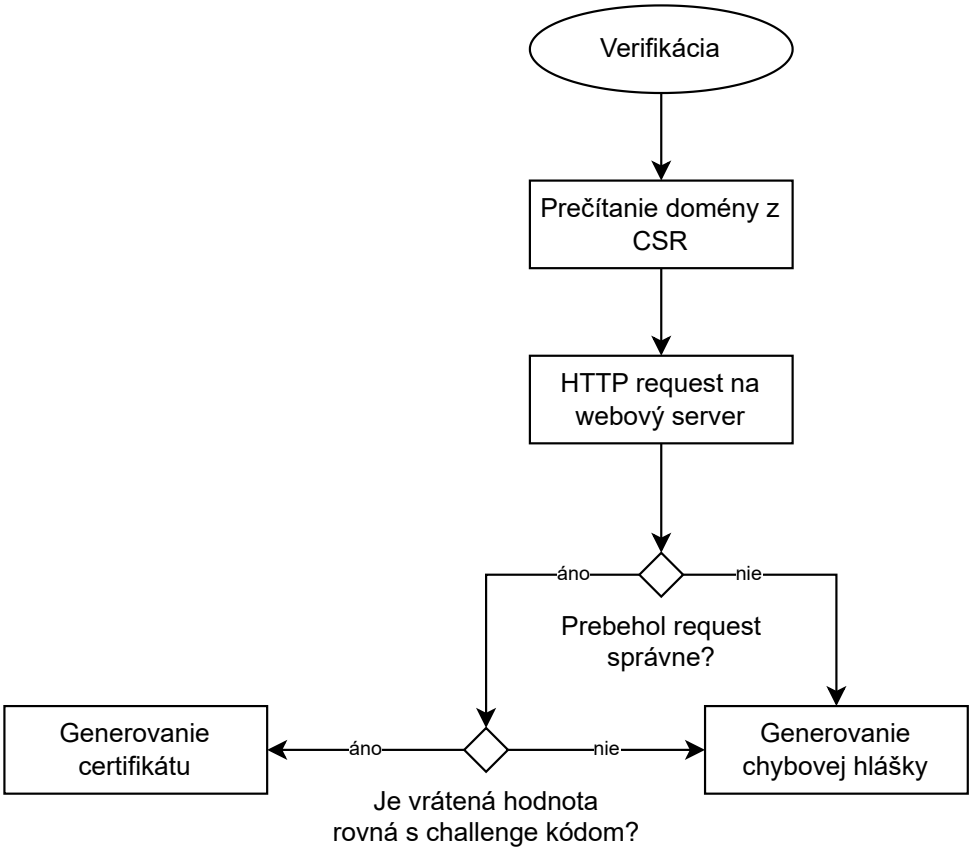
Klient následne vytvorí súbor s kódom na stanovenej ceste a podá žiadosť na overenie identity `/issue-operation/:id/verify`. Server načíta podľa identifikačného kódu uložený CSR a prečíta z neho doménu. Na doménu následne pošle HTTP request, a overí správnosť vráteného kódu. Ak je kód správny, sever vytvorí a podpíše certifikát a vráti ho klientovi.



Obr. 9: Diagram procesu vydania certifikátu



Obr. 10: Diagram funckionality klienta



Obr. 11: Diagram overenia identity

4.3 Použitie programu

Používateľ REST klienta vytvorí súbor `cert-client.json`, kde stanový:

- URL certifikačnej autority
- Doménu pre ktorý žiada certifikát
- Priečinok kde budú uložené súkromný a verejný kľúč a vytvorený certifikát
- Priečinok z ktorého webový server číta súbory na uloženie súboru s kódom výzvy

Príkladom konfigurácie je:

```
{
    "caUrl": "used-ca.local",
    "domain": "chodbau.local",
    "keyDir": "/etc/ssl",
    "webRoot": "/var/www/html"
}
```

Program následne musí byť spustený v priečinku obsahujúci tento konfiguračný súbor. Príklady výstupu programu pri úspešnej operácií je následovný:

```
[INFO] Loading config file...
[INFO] Testing connection to CA...
[INFO] Status { ready: true }
[INFO] Generating private key...
[INFO] Generating CSR...
[INFO] Sending CSR...
[INFO] IssueOperation {
  id: 'ynNC0yleT4H7eawWa2HzUg',
  challenge: '_K8zxIK84BXcCxENnZpPmQcn6JOVKHt-GYG6n3jeyEQQ',
  challengePath: '/pib-ca/challenge.txt'
}
[INFO] Preparing for challenge...
[INFO] Waiting for CA to verify...
[INFO] Saving certificate...
[INFO] Cleaning up challenge...
[INFO] Cleaning temporary files...
[INFO] Done!
```

Pri chybe program vypíše chybovú hlášku, možné chyby sú nasledovné:

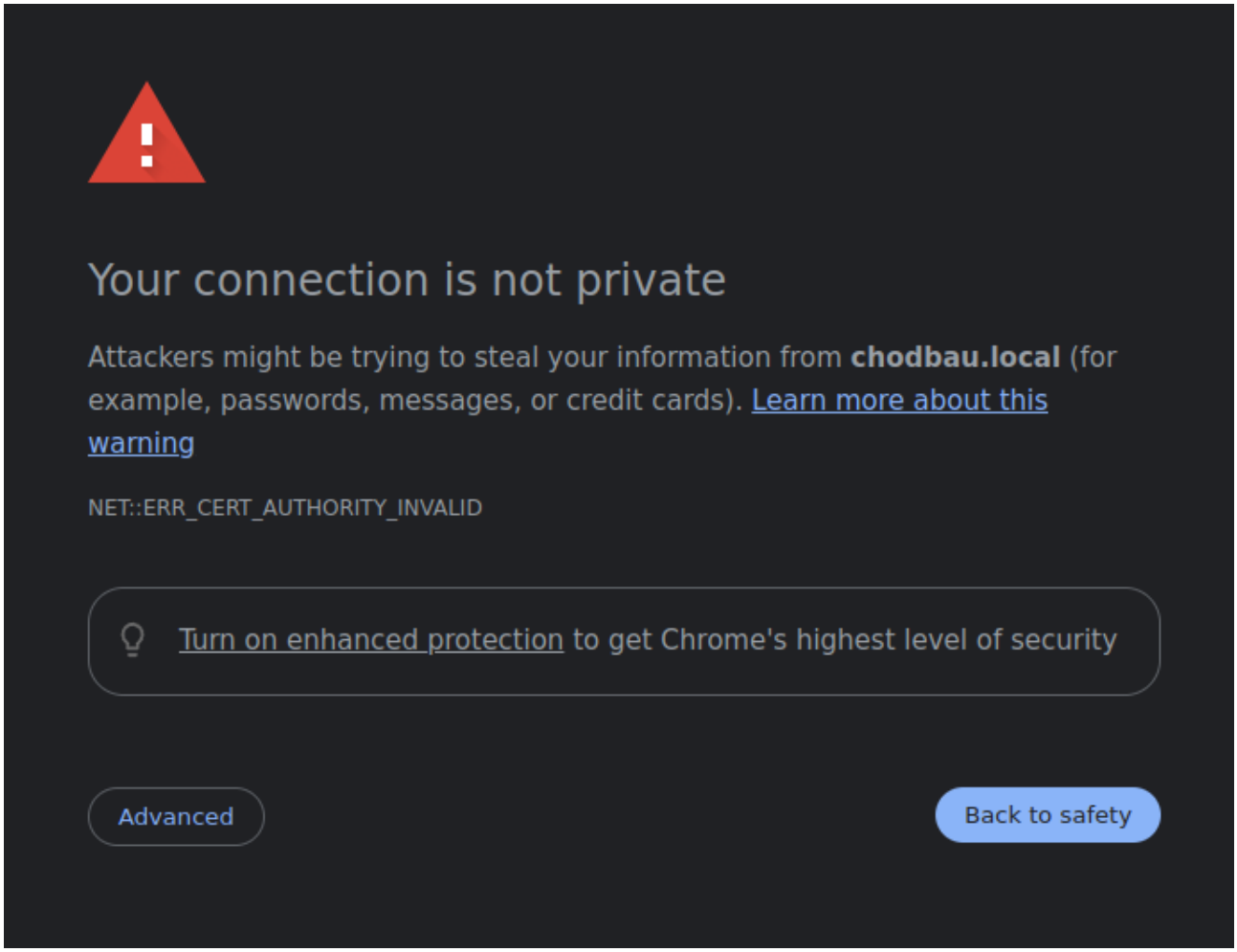
```
[ERROR] ERR_CA_CANNOT_VERIFY  Cannot verify identity, client responded with status 404

[ERROR] ERR_CA_CANNOT_VERIFY  Cannot verify identity, client did not respond with challenge code,
    got: "fxW
DcMGTtrYa_7shk0DabgAzrq16cJY47Y630wJqOg\n"

[ERROR] ERR_CA_CANNOT_VERIFY  Cannot verify identity because the CA cannot connect to the client
    domain
```

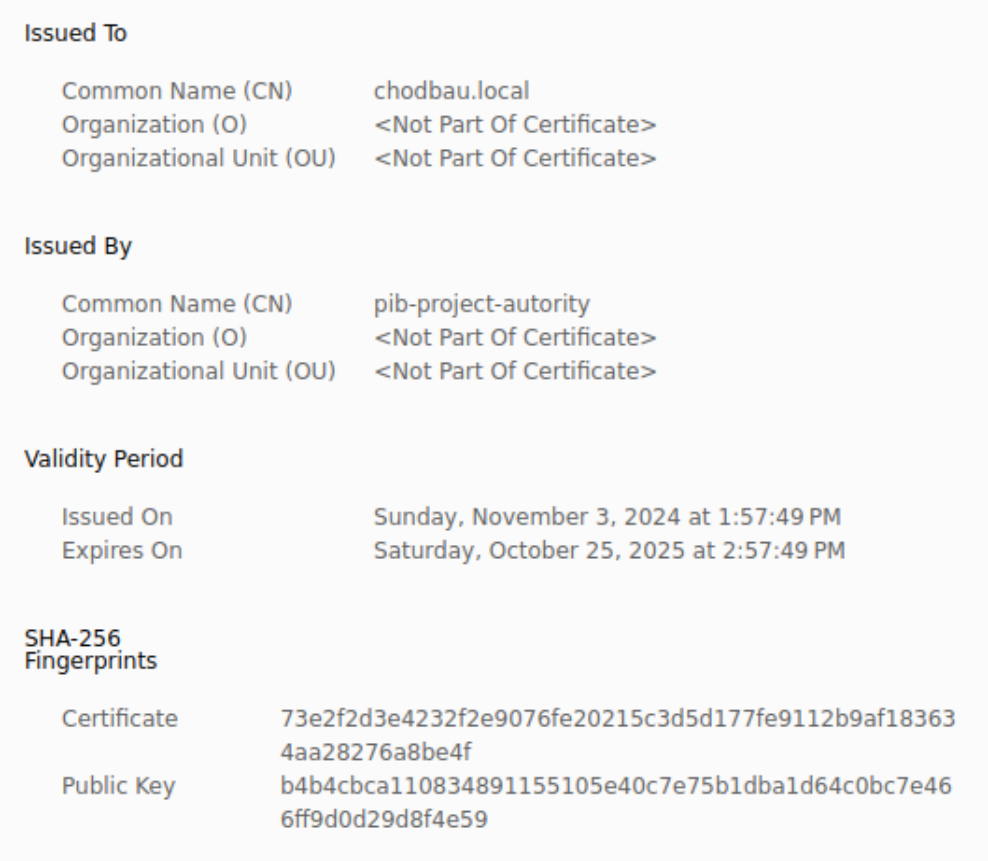
4.4 Testovanie programu

Program bol spustený a certifikát vygenerovaný. Pri prepojení s prehliadačom na webový server bola zobrazená chybová hláška. Toto je očakávané správanie, keďže certifikát je podpísaný certifikačnou autoritou, ktorú klient nespoznáva, čo udáva chybový kód `ERR_CERT_AUTHORITY_INVALID`.



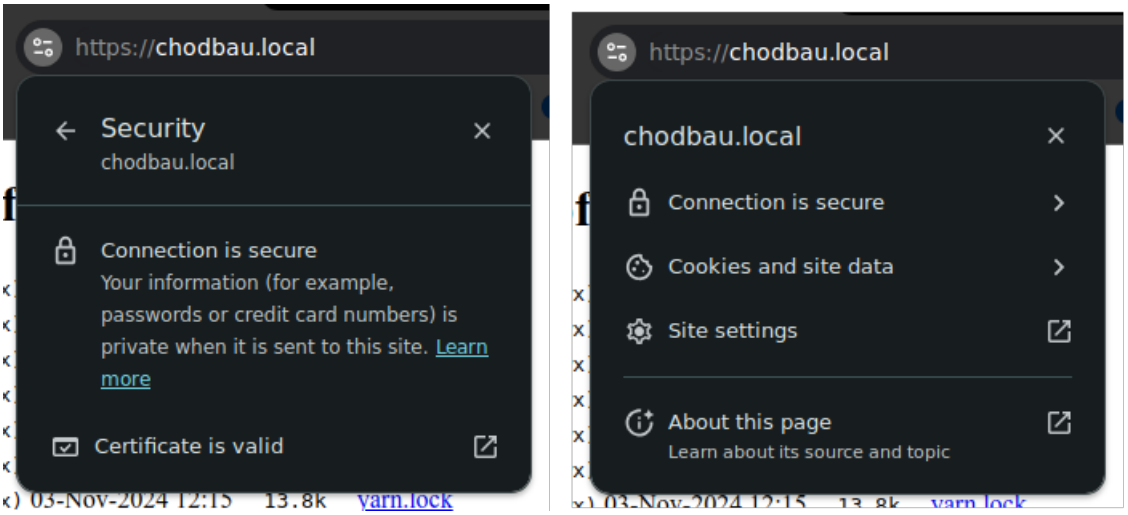
Obr. 12: Chybová hláška, prehliadač nespoznáva certifikačnú autoritu

Pri prehliadaní detailov certifikátu stránky sú zobrazené správne detaily certifikátu. Je možné vidieť správny názov domény a autority.



Obr. 13: Detaily certifikátu v prehliadači

Po inštalácií certifikátu certifikačnej autority a opätovnom načítaní stránky uz prehliadač nezobrazuje chybovú hlášku. Je možné vidieť že prehliadač považuje spojenie za bezpečné.



Obr. 14: Prehliadač považuje spojenie za bezpečné

5 Záver

Projekt predniesol princípy fungovania verejnej infraštruktúry verejného kľúča a tieto znalosti prakticky využil vytvorením vlastnej certifikačnej autority.

Literatúra

[1] BARNES, R.; HOFFMAN-ANDREWS, J.; MCCARNEY, D. et al.: Automatic Certificate Management Environment (ACME). RFC 8555, Marec 2019, doi:10.17487/RFC8555. Dostupné z: `https://www.rfc-editor.org/info/rfc8555`

[2] BOEYEN, S.; SANTESSON, S.; POLK, T. et al.: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280, Máj 2008, doi:10.17487/RFC5280. Dostupné z: `https://www.rfc-editor.org/info/rfc5280`

[3] Chromium Contributors: Recognized Logs. 2024. Dostupné z: `https://googlechrome.github.io/CertificateTransparency/log_list.html`

[4] Digicert®: CT Log Monitoring for Secure Site Pro from DigiCert. 2024. Dostupné z: `https://www.digicert.com/tls-ssl/ct-log-monitoring`

[5] Digicert®: What's the difference between DV, OV & EV SSL certificates? 2024. Dostupné z: `https://www.digicert.com/difference-between-dv-ov-and-ev-ssl-certificates`

[6] FORD, D. W. S.; CHOKHANI, D. S.; WU, S. S. et al.: Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework. RFC 3647, November 2003, doi:10.17487/RFC3647. Dostupné z: https://www.rfc-editor.org/info/rfc3647

[7] Google Inc.: Google Trust Services — Home. 2024. Dostupné z: https://pki.goog/

[8] GUTMANN, P.: Engineering Security, Apríl 2014, https://www.cs.auckland.ac.nz/~pgut001/pubs/book.pdf.

[9] JONSSON, J. KALISKI, B.: Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1. RFC 3447, Február 2003, doi:10.17487/RFC3447. Dostupné z: https://www.rfc-editor.org/info/rfc3447

[10] KEMP, D. P.; ADAMS, D. C.; MYERS, M. et al.: Internet X.509 Certificate Request Message Format. RFC 2511, Marec 1999, doi:10.17487/RFC2511. Dostupné z: https://www.rfc-editor.org/info/rfc2511

[11] LAURIE, B.; MESSERI, E. STRADLING, R.: Certificate Transparency Version 2.0. RFC 9162, December 2021, doi:10.17487/RFC9162. Dostupné z: https://www.rfc-editor.org/info/rfc9162

[12] Let's Encrypt: Challenge Types. Február 2023. Dostupné z: https://letsencrypt.org/docs/challenge-types/

[13] Let's Encrypt: How It Works - Let's Encrypt. Jún 2024. Dostupné z: https://letsencrypt.org/how-it-works/

[14] Microsoft: [MS-OCSP]: Overview — Microsoft Learn. Apríl 2024. Dostupné z: https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-ocsp/5792b4c4-c6ba-439a-9c2a-52867d12fb66

[15] MozillaWiki Contributors: CA/Revocation Checking in Firefox - MozillaWiki. Február 2024. Dostupné z: https://wiki.mozilla.org/CA/Revocation_Checking_in_Firefox

[16] PKI Consortium: What Are the Different Types of SSL Certificates? 2013. Dostupné z: https://pkic.org/2013/08/07/what-are-the-different-types-of-ssl-certificates/

[17] SANTESSON, S.: Internet X.509 Public Key Infrastructure Subject Alternative Name for Expression of Service Name. RFC 4985, August 2007, doi:10.17487/RFC4985. Dostupné z: https://www.rfc-editor.org/info/rfc4985

[18] SCHEITLE, Q.; GASSER, O.; NOLTE, T. et al.: The Rise of Certificate Transparency and Its Implications on the Internet Ecosystem. In: *Proceedings of the Internet Measurement Conference 2018*, IMC '18, New York, NY, USA: Association for Computing Machinery, 2018, ISBN 9781450356190, s. 343–349, doi:10.1145/3278532.3278562. Dostupné z: https://doi.org/10.1145/3278532.3278562

[19] SELTZER, L.: Chrome does certificate revocation better — ZDNET. Apríl 2014. Dostupné z: https://www.zdnet.com/article/chrome-does-certificate-revocation-better/

[20] SSL.com: What Is Certificate Pinning? - SSL.com. Október 2023. Dostupné z: `https://www.ssl.com/blogs/what-is-certificate-pinning/`

[21] W3Techs.com: Usage statistics and market shares of SSL certificate authorities for websites. November 2024. Dostupné z: `https://w3techs.com/technologies/overview/ssl_certificate`