SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Ing. Martin Köppl

Autoreferát dizertačnej práce

Zabezpečenie prenosu kľúčov v sieti prostredníctvom eliptických kriviek

na získanie akademického titulu „doktor“ („philosophiae doctor“, v skratke „PhD.“)

v doktorandskom študijnom programe: telekomunikácie

v študijnom odbore: informatika

Forma štúdia: denná

Miesto a dátum: Bratislava, máj 2023

STU

Dizertačná práca bola vypracovaná na:
Ústav multimediálnych informačných a komunikačných technológií
Fakulta elektrotechniky a informatiky
Slovenská technická univerzita v Bratislave

Predkladateľ: Ing. Martin Köppl
Slovenská technická univerzita v Bratislave
Fakulta elektrotechniky a informatiky
Ústav multimediálnych informačných a komunikačných technológií

Školiteľ: doc. Ing. Miloš Orgoň, PhD.
Slovenská technická univerzita v Bratislave
Fakulta elektrotechniky a informatiky
Ústav multimediálnych informačných a komunikačných technológií

Oponenti: prof. Ing. Jiří Mišurec, CSc.
Vysoké učení technické v Brně
Fakulta elektrotechniky a komunikačních technologií
Ústav telekomunikací

Ing. Juraj Taraba, PhD.
Slovak Telekom, a.s.

Autoreferát bol rozoslaný .....

Obhajoba dizertačnej práce sa bude konať dňa:.....o.....**h**

na.....

.....
prof. Ing. Vladimír Kutiš, PhD.

Obsah

1. Úvod

História kryptografie siaha do dávnych čias, kedy ľudia potrebovali odosielať tajné správy. Úlohou bolo doručiť správu príjemcovi tak, aby ju bol schopný rozlúštiť práve on. Tento proces pozostával z jednoduchých krokov, najskôr bolo treba správu napísať, následne ju zašifrovať a odoslať. Príjemca ju po prijatí dešifroval a získal informáciu. Ak by sa počas cesty aj nejaký iný človek pokúsil informáciu získať, nepodarilo by sa mu to práve vďaka zašifrovanému textu. Tento jednoduchý princíp sa využíval v minulosti, ale používa sa aj teraz. Časom vynikla nutnosť vylepšovať spôsoby šifrovania a tak sa rozvíjala aj veda nazývaná kryptografia.

V dnešnej dobe je šifrovanie komunikácie nevyhnutné najmä v online priestore a je to každodennou súčasťou takmer každého človeka. Spoločnosti pracujú s množstvom citlivých údajov, ktoré prenášajú cez zabezpečené aj nezabezpečené prostredia, ľudia nakupujú online, kde zadávajú bankové údaje, atď. Človek denne vytvára tisícky šifrovaných spojení napríklad pri surfovaní na Internete a ani si to nemusí uvedomovať. Týmto všetkým sa zaoberá práve kryptografia, ktorá pomocou protokolov a štandardov aplikuje bezpečnostné mechanizmy na tieto spojenia a udržuje ich bezpečnými.

Spôsobov šifrovania existuje v dnešnej dobe niekoľko. Jedno z efektívnych, avšak nie najpopulárnejších riešení, je kryptografia pomocou eliptických kriviek. Toto riešenie prináša niekoľko výhod, najväčší potenciál však predstavuje nízka energetická a hardvérová náročnosť, čo môže byť obrovským prínosom pre rozvíjajúcu sa sféru IoT zariadení. Aj preto sa rozpráva o eliptických krivkách viac a viac.

Dizertačná práca sa skladá z niekoľkých častí. V prvých častiach je opísaná kryptografia eliptických kriviek a základné algoritmy využívajúce tieto krivky ECDSA a ECDH. Takisto tu nájdeme popísané bližšie krivky, ktoré odporúča Americká bezpečnostná inštitúcia NIST a aj krivky typu Brainpool. V druhej časti je diskutovaná téma dôveryhodnosti eliptických kriviek. V ďalších častiach práce je popísaný výber eliptických kriviek pre zabezpečenie prenosu dát, návrh optimalizácie algoritmu ECDH pre efektívnejšie využitie prenosového pásma a výpočtového výkonu v podnikových sieťach a návrh modelu implementácie eliptických kriviek. V poslednej časti je popísaná samotná implementácia navrhnutého modelu na reálnych zariadeniach a zhodnotenie praktických výsledkov.

2. Ciele dizertačnej práce

Ciele dizertačnej práce sa budú odvádzať z nasledovných téz:

1. Na základe výsledkov testovania vyberte vhodné eliptické krivky na zabezpečenie šifrovaného prenosu dát.

2. Otestujte a porovnajte efektivitu zabezpečenia prenosu kľúčov v sieti prostredníctvom eliptických kriviek.

3. Navrhnite model implementácie eliptických kriviek na ochranu podnikových sietí pre efektívne využívanie výkonu na výpočet aritmetických operácií potrebných pri kryptografických funkciách.

4. Overte navrhnuté riešenie implementácie eliptických kriviek experimentom.

3. Súčasný stav riešenia problematiky

Kryptografia pomocou eliptických kriviek bola predstavená v roku 1985 pánmi Victorom Millerom a Nealom Koblitzom. Tento typ kryptografie ponúka alternatívu k typom asymetrického šifrovania pomocou privátnych a verejných kľúčov. Hlavným dôvodom atraktivity eliptických kriviek je takzvaný problém diskrétneho logaritmu pri eliptických krivkách, ktorý sa dodnes úspešne nedarí efektívne vyriešiť. [1]

1, Máme eliptickú krivku E definovanú nad konečným poľom F_q.
2, $E: y^2 = x^3 + Ax + B$; $A,B \in F_q$.
3, Body S a $T \in$ krivke $E(F_q)$. $S,T \in E(F_q)$.
4, Bod T je násobkom bodu S skalárom m. $T = m*S$.
5, Nájdenie čísla $m \in N$ sa nazýva problém diskrétneho logaritmu na eliptickej krivke E.

ECDSA je verzia DSA algoritmu využívajúca eliptické krivky. Tento algoritmus používa tzv. signatár, ktorý slúži na vygenerovanie digitálneho podpisu, a overovateľ, ktorý overuje pravosť podpisu. Každý signatár má verejný a súkromný kľúč. Súkromný kľúč je použitý pri procese vytvorenia podpisu a verejný kľúč je použitý pri overovacom procese. Signatár je jediný, kto pozná súkromný kľúč, a teda jediný, kto vie vygenerovať správny podpis. Naopak overiť podpis vie ktokoľvek, pretože verejný kľúč je dostupný pre všetkých.

Každá eliptická krivka má zadefinované doménové parametre. Doménové parametre eliptických kriviek môžu byť verejné, bezpečnosť systému nezávisí od tajnosti ich parametrov. [2]
Pri ECDSA rozlišujeme dva scenáre:

1. Doménové parametre eliptickej krivky cez pole F(p),

2. Doménové parametre eliptickej krivky cez pole F(2^m)

Diffie-Hellman protokol slúži na výmenu kľúčov cez nezabezpečené prostredie. Pôvodne bol definovaný pre operácie v poliach s vysokými prvočíslami. Neal Koblitz a Victor Miller neskôr analyzovali tento protokol na využitie s eliptickými krivkami. [3]

ECDH je veľmi podobný klasickému algoritmu DHKE (Diffie-Hellman Key Exchange). Namiesto modulárnych umocnení však využíva násobenie bodov eliptickej krivky. ECDH je založený na nasledovnej vlastnosti bodov eliptickej krivky: (a * G) * b = (b * G) * a. [4]

Ak máme dve tajné čísla *a* a *b* (dva súkromné kľúče patriace účastníkovi 1 a účastníkovi 2, a eliptickú krivku ECC s generátorom bodu G, vieme vymeniť cez nezabezpečený kanál hodnoty (a * G) a (b * G) (verejné kľúče účastníka 1 a účastníka 2), a z toho odvodiť spoločné tajomstvo: *secret = (a * G) * b = (b * G) * a*. Rovnica získa nasledovnú formu:

A-PublicKey*B-PrivateKey = B-PublicKey*A-PrivateKey = spoločné tajomstvo

ECDH algoritmus:

1. Účastník A vygeneruje náhodný ECC pár kľúčov:

   (A-PrivateKey, A-PublicKey = A-PrivateKey*G).

2. Účastník B vygeneruje náhodný ECC pár kľúčov:

   (B-PrivateKey, B-PublicKey = B-PrivateKey*G).

3. Účastník A a účastník B si vymenia ich verejné kľúče cez nezabezpečený kanál (napr. Internet).

4. Účastník A vypočíta spoločné tajomstvo:

   secret = B-PublicKey * A-PrivateKey.

5. Účastník B vypočíta spoločné tajomstvo:

secret = A-PublicKey * B-PrivateKey.

6. Účastník A a účastník B teraz majú rovnaké spoločné tajomstvo:

secret = B-PublicKey * A-PrivateKey = A-PublicKey * B-PrivateKey [4]

4. Výber eliptických kriviek na zabezpečenie prenosu dát

Na správny výber eliptických kriviek pre zabezpečenie prenosu dát v praxi treba zvážiť niekoľko parametrov, a to najmä:

- dĺžku kľúča,
- bezpečnosť
- výkonovú náročnosť spracovania danej krivky,
- podporu danej krivky rôznymi systémami tretích strán.

Dĺžka kľúča a bezpečnosť spolu priamo korelujú. Hoci dĺžka kľúča nie je jediným faktorom bezpečnosti danej krivky, je známe, že čím dlhší kľúč daná krivka má, tým je úroveň bezpečnosti vyššia. Samotná bezpečnosť v praxi zahŕňa aj nasledovné faktory: typ danej krivky, dôveryhodnosť krivky, schopnosť implementácie krivky do prostredia a iné.

Výkonová náročnosť spracovania danej krivky je kľúčová pri implementáciách so zariadeniami s nižším výpočtovým výkonom. V dnešnej dobe sú zariadenia, ktoré disponujú fyzicky malými rozmermi, dobre vybavené na spracovanie rôznych kriviek. Treba však zvážiť aj scenáre, kedy zariadenie šifruje niekoľko dátových tokov súčasne, a teda aj generuje kľúče častejšie. Preto nie je výkonová náročnosť spracovania danej krivky zanedbateľná.

V laboratórnych podmienkach je možné prispôsobiť prostredie akejkoľvek krivke. V praxi je však dôležité, aby bola implementácia danej krivky podporovaná a dobre zabezpečená. Nakoľko firmy často využívajú produkty od viacerých výrobcov, je vhodné použiť známe populárne krivky, ktoré sú implementované v čo najväčšom počte produktov na trhu. Široká implementácia takisto podporuje skúmanie daných nasadení pre rôzne tretie strany a teda pomáha riešiť potenciálne bezpečnostné hrozby samotných implementácií.

4.1 Výber vhodných eliptických kriviek na použitie v praxi

Na základe publikácie amerického inštitútu NIST, bola prvým kritériom zvolenia krivky práve dĺžka kľúča aspoň 256 bitov. Tabuľka 8 uvádza dané odporúčanie, a hoci hodnota bezpečnostnej sily 112 je do roku 2030 prípustná, neskôr bude nedostatočne bezpečná. Naopak bezpečnostná sila 128, ktorá odpovedá dĺžke kľúča pri eliptických krivkách 256 bitov, bude akceptovateľná aj po roku 2030.

Tab. 1 Odporúčané bezpečnostné úrovne do a po roku 2030

Bezpečnostná úroveň (bit)	Do roku 2030	Po roku 2030
< 112	Nepovolené	Nepovolené
112	Akceptovateľné	Nepovolené
128	Akceptovateľné	Akceptovateľné
192	Akceptovateľné	Akceptovateľné
256	Akceptovateľné	Akceptovateľné

Z hľadiska bezpečnosti, výkonovej náročnosti spracovania a podpory systémami tretích strán bol výber zúžený na tieto štyri eliptické krivky:

- BrainpoolP256r1,
- BrainpoolP384r1,
- NIST256p,
- NIST384p.

Skupina Brainpool bola vybraná ako najdôveryhodnejšia, nakoľko je matematický proces generovania dokázateľne náhodný. Skupina NIST bola vybraná kvôli verejne známej efektivite pri spracovaní, ktorá bola podložená niekoľkými vedeckými článkami a aj samotným našim experimentom. Z hľadiska výkonovej náročnosti boli určené krivky s minimálnou bitovou dĺžkou kľúča 256 bitov, nakoľko tieto krivky spĺňajú bezpečnostné odporúčanie aj po roku 2030 a súčasne tieto krivky vyžadujú najnižší výkon v porovnaní s krivkami s vyššími dĺžkami kľúča, čo je odrazené na rýchlosti generovania samotných kľúčov. Vybraté boli aj krivky s dĺžkou kľúča 384 bitov, čo odpovedá bezpečnostnej sile 192, a teda poskytujú vyššiu úroveň bezpečnosti. Skupiny kriviek NIST a Brainpool sú široko rozšírené v systémoch tretích strán, ako napríklad softvér OpenSSL, ktorý bude využitý pre experimentálnu časť.

Tabuľka 9 uvádza porovnanie časov pre dané krivky pri algoritmoch ECDH a ECDSA podľa nášho experimentu publikovaného v [5]:

Tab. 2 Porovnanie časov ECDH a ECDSA pre vybrané krivky [5]

Názov krivky	Bezpečnostná sila krivky (security bit)	Čas vytvorenia algoritmu ECDH (s)	Čas generovania kľúčov pri ECDSA (s)	Čas podpisu správy pri ECDSA (s)	Čas overenia správy pri ECDSA (s)
BrainpoolP256r1	128	0.00309	0.00085	0.00088	0.00172
BrainpoolP384r1	192	0.00633	0.00184	0.00183	0.00334
NIST256p	128	0.00296	0.00084	0.00092	0.00172
NIST384p	192	0.00609	0.00179	0.00181	0.0035

4.2 Testovanie AES algoritmu s možnosťou jeho vynechania pri prenosoch malých súborov

AES alebo Advanced Encryption Standard algoritmus je typ symetrického šifrovania, ktorý využíva rovnaký kľúč na kryptovanie a dekryptovanie dát. Používa takzvaný SPN (Substitution Permutation Network) algoritmus, ktorý aplikuje niekoľko kôl kryptovania. Existujú tri dĺžky AES kryptovacích kľúčov:

- 128 bitový kľúč, celkovo 3.4x1038 možných kľúčových kombinácií,
- 192 bitový kľúč, celkovo 6.2x1057 možných kľúčových kombinácií,
- 256 bitový kľúč, celkovo 1.1x1077 možných kľúčových kombinácií.

Hoci sa dĺžka kľúča líši, veľkosť bloku zostáva vždy 128 bitov. V súčasnosti je AES jedným z najlepších dostupných šifrovacích protokolov, pretože kombinuje rýchlosť a bezpečnosť. [6]

AES je teda šifrovací protokol, ktorý nešifruje metódou „bit po bite“, ale „blok po bloku“. V každom kole kryptovania používa niekoľko procesov, kedy šifruje nešifrovaný text do šifrovaného. Celkovo používa až 10 kôl (rúnd) pre 128 bitové kľúče, 12 kôl pre 192 bitové kľúče a 14 kôl pre 256 bitové kľúče. [7]

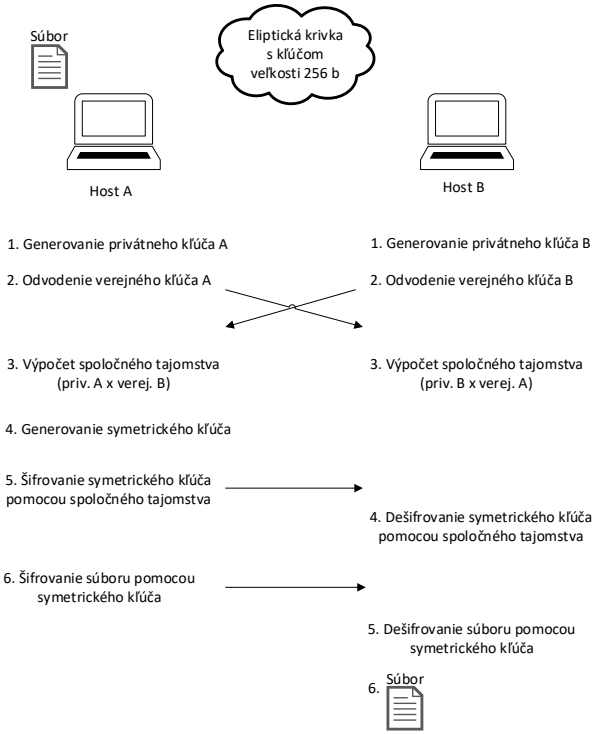
Ak sa vezme do úvahy, že existuje súbor určitej veľkosti, ktorý treba zašifrovať, v skutočnosti je celý proces načrtnutý na Obr. 20. Z hľadiska potrebnej šírky pásma sa prenáša medzi odosielateľom a prijímateľom verejný kľúč, napríklad s veľkosťou 256 bitov. Pri zachovaní rovnakej sily šifry

potom odosielateľ generuje AES kľúč dĺžky 128 bitov, ktorý následne šifruje pomocou zdieľaného tajomstva – výsledná dĺžka šifry je 128 bitov, ktorá je odoslaná prijímateľovi. Po doručení ju prijímateľ dešifruje, a keď už obaja disponujú spoločným kľúčom, odosielateľ šifruje týmto kľúčom originálnu správu, ktorú následne odošle prijímateľovi. Celková suma prenesených bitov je teda:

- 256 bitový verejný kľúč od odosielateľa,
- 256 bitový verejný kľúč od prijímateľa,
- 128 bitový symetrický kľúč od odosielateľa,
- zašifrovaný súbor, ktorého veľkosť záleží od veľkosti originálneho súboru.

Z hľadiska náročnosti výpočtového výkonu treba vykonať tieto kroky:

- vygenerovať privátny kľúč z dohodnutej eliptickej krivky (obaja),
- odvodiť verejný kľúč (obaja),
- vymeniť si verejné kľúče (obaja),
- vygenerovať spoločné tajomstvo (obaja),
- odosielateľ musí vygenerovať symetrický kľúč,
- odosielateľ šifruje symetrický kľúč pomocou spoločného tajomstva,
- odosielateľ odošle šifrovaný symetrický kľúč prijímateľovi,
- prijímateľ dešifruje symetrický kľúč pomocou spoločného tajomstva,
- odosielateľ šifruje súbor pomocou symetrického kľúča,
- odosielateľ odosiela šifrovaný súbor prijímateľovi,
- prijímateľ dešifruje šifrovaný súbor.



Obr. 1 Postup šifrovania s využitím ECDH

Pri určitých podmienkach by mohol však fungovať aj proces bez generovania nového symetrického kľúča. Vhodné prostredia by mohli byť práve siete, v ktorých treba prenášať veľkosťou malé súbory, avšak veľmi často. Na zachovanie vyššej úrovne bezpečnosti by každá takáto správa využívala nový pár kľúčov, avšak znížil by sa potrebný výkon pre matematické operácie a aj samotná potreba šírky pásma. V tomto scenári by odosielateľ negeneroval nový symetrický kľúč, ale odderivoval by reťazec priamo zo spoločného tajomstva, vytvoreného pomocou asymetrického šifrovania eliptickými krivkami. Na zachovanie bezpečnostnej úrovne by odosielateľ použil prvých 128 bitov z 256-bitového spoločného tajomstva ako symetrický kľúč. Súbor by teda šifroval pomocou nového odvodeného kľúča a následne ho odoslal prijímateľovi, ktorý by ho rovnakým spôsobom dešifroval. Ušetril by sa výkon a potrebná šírka pásma, keďže by sa vynechal proces generovania

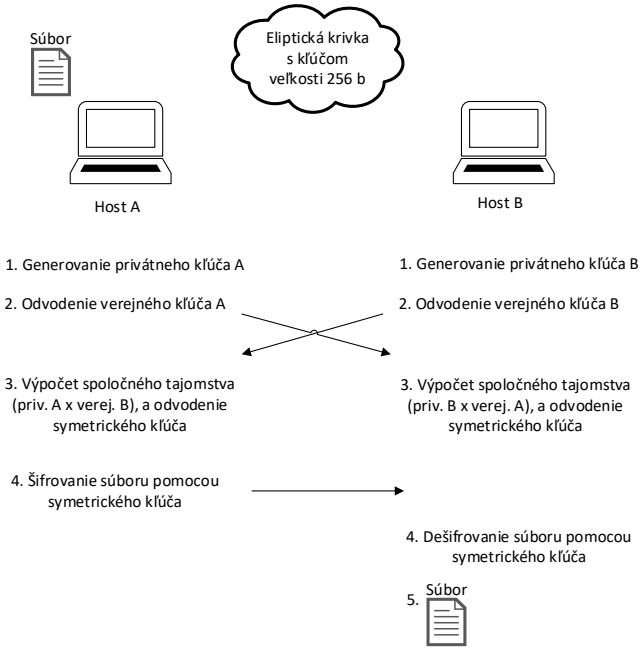
symetrickej šifry u odosielateľa, jeho šifrovanie, odoslanie a nakoniec dešifrovanie u prijímateľa. Proces je znázornený na Obr. 21.

Celková suma prenesených bitov je teda:

- 256 bitový verejný kľúč od odosielateľa,
- 256 bitový verejný kľúč od prijímateľa,
- zašifrovaný súbor, ktorého veľkosť záleží na veľkosti originálneho súboru

Z hľadiska náročnosti výpočtového výkonu by bolo treba vykonať tieto kroky:

- vygenerovať privátny kľúč z dohodnutej eliptickej krivky (obaja),
- odvodiť verejný kľúč (obaja),
- vymeniť si verejné kľúče (obaja),
- vygenerovať spoločné tajomstvo (obaja),
- odvodiť symetrický kľúč zo spoločného tajomstva – prvých 128 bitov,
- odosielateľ šifruje súbor pomocou symetrického kľúča,
- odosielateľ odosiela šifrovaný súbor prijímateľovi,
- prijímateľ dešifruje šifrovaný súbor.



Obr. 2 Postup šifrovania s elimináciou kroku generovania symetrickej šifry

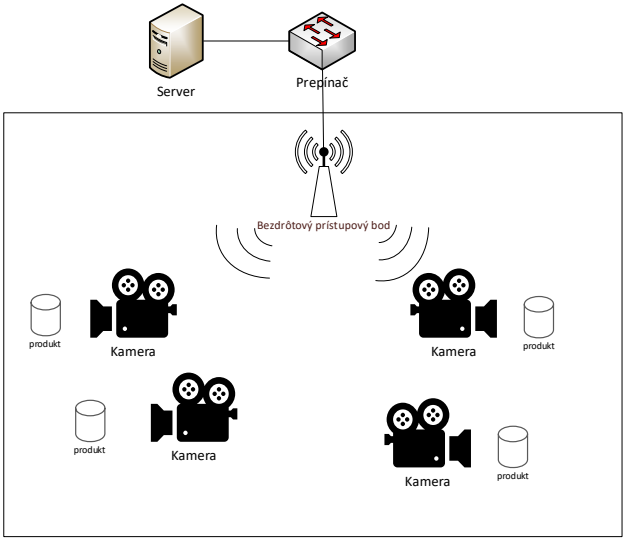
Na prvý pohľad je vidieť, že celkový výpočtový výkon a potrebná šírka pásma bude nižšia, ako pri prvom scenári, ktorý sa v súčasnosti reálne využíva. Prenesené bity sú však len teoretické, pretože treba uvažovať aj podporné bity celej správy, napríklad v IP sieťach pridanie hlavičiek všetkých vrstiev OSI modelu. Takisto sa kľúče neprenášajú v čistom texte, ale častokrát napríklad v pem súboroch, ktoré majú pridané parametre na jednoduchú identifikáciu v digitálnom svete.

5. Návrh modelu implementácie eliptických kriviek

Na uľahčenie nasadenia eliptických kriviek na ochranu sietí bol vytvorený model implementácie eliptických kriviek. Vytvorený model bol zameraný hlavne na nasledovné faktory:

- sieť využiteľná v podnikovom prostredí ,
- prostredie vyžadujúce vysokú úroveň bezpečnosti prenosu cez sieť,
- obmedzená šírka prenosového pásma,
- obmedzený výpočtový výkon koncových zariadení siete.

Na Obr. 22 je možné vidieť navrhnutý model, ktorý vyhovuje požiadavkám. Tento model berie do úvahy všetky vyššie uvedené podmienky. Ide o podnikovú sieť nachádzajúcu sa v budove alebo výrobnej hale, ktorá je založená na kombinácii bezdrôtovej a káblovej technológie, pomocou ktorej komunikujú koncové zariadenia so serverom. Podnik je zameraný na kontrolu kvality vyrobených produktov, kedy koncové zariadenia pomocou kamery snímajú dané produkty a vyhodnocujú ich kvalitu, chyby a odchýlky.



Obr. 3 Model implementácie eliptických kriviek

5.1 Elementy siete

- **Koncové zariadenia**

Koncové zariadenia disponujú malým výpočtovým výkonom, konkrétne mikropočítačom typu Raspberry Pi. Mikropočítač spracováva obraz pomocou kamery, a takisto vyhodnocuje kvalitu produktov pomocou nainštalovaného softvéru. Následne vytvára metadáta, ktoré odosiela cez sieť na server. Koncových zariadení je umiestnených v miestnosti niekoľko, a frekvencia vyhodnocovania a odosielania dát je vysoká.

- **Sieť**

Sieť sa skladá z bezdrôtovej a káblovej časti. Bezdrôtová časť pozostáva z technológie LiFi, kedy sú v miestnosti umiestnené vysielače, ktoré využívajú ako prenosové médium svetlo. Koncové zariadenia disponujú LiFi prijímačmi, ktoré zabezpečujú komunikáciu s vysielačom a následne aj so zvyškom siete. Prenosové rýchlosti v LiFi prostredí sú obmedzené, avšak oproti klasickému WiFi prinášajú aj množstvo výhod. Káblová časť pozostáva z prepínača, na ktorom sú priamo napojené vysielač a server.

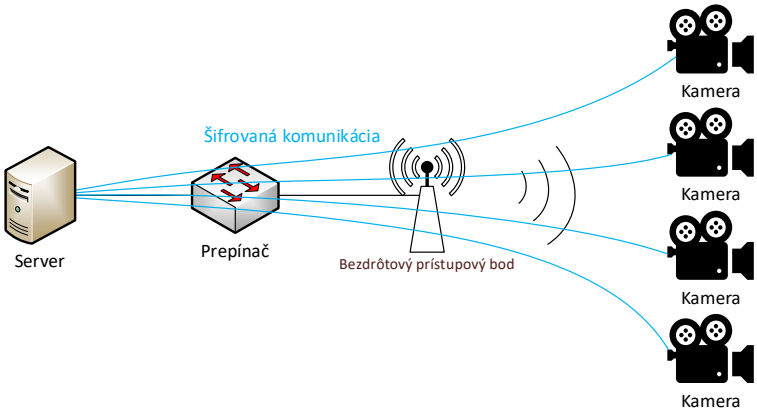
Reálne podnikové siete obsahujú väčšie množstvá sieťových zariadení, závisí to najmä od veľkosti podniku, počte pripojených zariadení, bezpečnostnej úrovne alebo redundancie. Náš model je elementárny, pozostávajúci z malého množstva zariadení, avšak princípy zabezpečenia sú rovnaké ako pri väčších sieťach.

- **Server**

Typ samotného serveru nie je vôbec podstatný, jedinou podmienkou je predpoklad, že bude dostatočne výkonný na to, aby spracoval správy od viacerých koncových zariadení naraz, a teda sa nestal najslabším článkom siete, ktorý by prenos spomaľoval. Do LiFi pásma nijako nezasahuje, pretože je pripojený pomocou kábla priamo do prepínača. Takisto musí podporovať eliptické krivky a softvéry kompatibilné s koncovými zariadeniami.

5.2 Zabezpečenie komunikačného toku

Z pohľadu bezpečnosti je nutné kryptovať komunikáciu po celej ceste cez sieť, teda od koncového zariadenia až po server. Nakoľko je v sieti viac koncových zariadení, každé z nich bude komunikovať so serverom oddelene. Obr. 23 znázorňuje oddelené komunikačné toky jednotlivých koncových zariadení.



Obr. 4 Komunikačné toky medzi zariadeniami a serverom

Celkový cyklus pozostáva z troch krokov:

1, Koncové zariadenie nasníma produkt,

2, koncové zariadenie vyhodnotí kvalitu a vytvorí metadáta,

3, koncové zariadenie zašifruje dáta a odošle ich serveru. Tento krok predstavuje bezpečnostnú fázu cyklu, kedy prebieha:

- výmena kľúčov pomocou asymetrického šifrovania,
- šifrovanie metadát pomocou symetrickej šifry,
- odosielanie a dešifrovanie šifrovanej správy a spracovanie informácie.

5.2.1 Výmena kľúčov pomocou asymetrického šifrovania

Prvým potrebným krokom je zvoliť krivku, ktorá sa bude používať. Je potrebné, aby túto krivku obe strany poznali. V našom návrhu bola zvolená krivka BrainpoolP256r1, pretože sa ukázala ako najlepšia voľba, kvôli vlastnostiam:

- dostatočná úroveň ochrany (256 bitový kľúč),

- dôveryhodnosť,

- rýchlejší čas operácií v porovnaní s krivkami s väčšími bitovými dĺžkami,

- menšia náročnosť na výpočtový výkon a šírku pásma,

- priama kompatibilita so symetrickým šifrovaním AES,

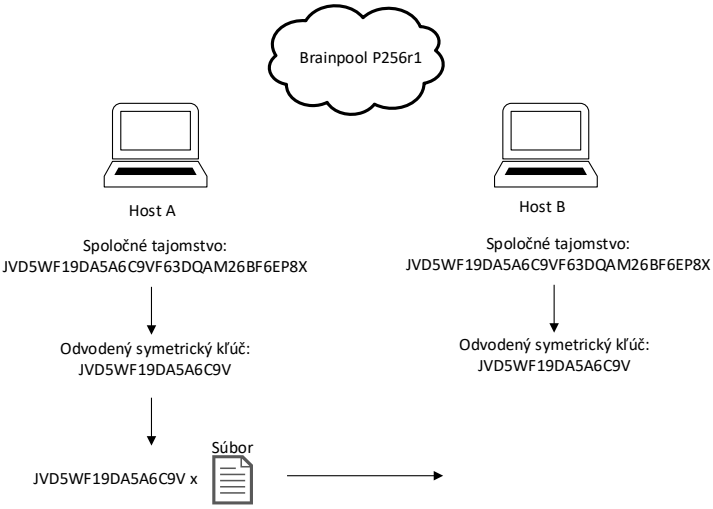
- štandardná podpora u mnohých výrobcov hardvéru.

Iniciátorom spojenia je v našom prípade koncové zariadenie, ktoré si generuje privátny a verejný kľúč pomocou krivky BrainpoolP256r1. Verejný kľúč je následne odoslaný serveru, ktorý ho prijíma, a generuje si svoj pár kľúčov podľa danej krivky. Verejný kľúč servera je potom odoslaný koncovému zariadeniu. Obe strany si teda vedia vypočítať spoločné tajomstvo podľa definície ECDH.

5.2.2 Šifrovanie metadát pomocou symetrickej šifry

Podľa zaužívaných techník by malo nasledovať generovanie symetrického kľúča na strane koncového zariadenia, šifrovanie kľúča pomocou spoločného tajomstva a následné doručenie kľúča serveru. V našom návrhu sú však tieto kroky vynechané, a symetrický kľúč sa priamo derivuje zo spoločného tajomstva.

Krivka BrainpoolP256r1 je priamo kompatibilná so symetrickým šifrovaním, a teda vypočítané 256-bitové spoločné tajomstvo by mohlo byť priamo použité pre symetrické šifrovanie AES256. AES256 však poskytuje vyššiu úroveň ochrany ako 256-bitové eliptické krivky, a teda 256-bitové tajomstvo môže byť oklieštené o polovicu - na 128 bitov. 128 bitový kľúč vie byť teda použitý na symetrické šifrovanie AES128, čo zodpovedá bezpečnostnej úrovni eliptických kriviek s dĺžkou kľúča 256 bitov. Bezpečnostná úroveň teda ostáva zachovaná, a eliminuje sa potreba generácie symetrického tajomstva, šifrovania symetrického tajomstva, odoslania šifrovaného symetrického tajomstva a jeho následné dešifrovanie na opačnej strane. Obr. 24 znázorňuje postup.



Obr. 5 Postup derivácie symetrického kľúča

5.2.3 Odosielanie a dešifrovanie šifrovanej správy

Odosielanie správy prebieha štandardne, bez špeciálnych praktík. Koncové zariadenie a server spolu komunikujú pomocou IP adries. Dešifrovanie správy vykonáva server pomocou symetrického kľúča, ktorý si sám odvodil zo spoločného tajomstva.

5.2.4 Veľkosť metadát

V našom návrhu je riešenie, v ktorom je prenášané veľké množstvo šifrovaných správ s malým obsahom dát. Softvér na tvorbu metadát je preto vhodné prispôsobiť sieťovému riešeniu. Bez ohľadu na tvorbu kľúčov a ďalších procesov, podstatná je veľkosť kryptovanej správy. Metadáta budú kryptované pomocou AES protokolu s veľkosťou kľúča 128 bitov. AES protokol všetkých úrovní rozdeľuje nešifrovaný text na 128-bitové bloky, ktoré následne šifruje v niekoľkých krokoch pomocou kľúča. V prípade, že daný blok nemá 128 bitov (posledný blok reťazca, alebo reťazec s menšou dĺžkou), pridáva AES doplnkové bity, aby bola konečná dĺžka bloku práve 128 bitov.

V prvom kroku bol vytvorený súbor symkey.txt, ktorý reprezentuje 128-bitový symetrický kľúč, ktorý bude použitý na AES šifrovanie. V druhom príkaze boli pridané 8-bitové znaky do súboru. V Linuxovom prostredí po zadaní príkazu echo je pridaný na koniec znak nového riadku, a teda konečná veľkosť súboru nebola 16 B ale 17 B. Tretí a štvrtý príkaz odstraňujú znak nového riadku a teda vytvárajú kľúč veľkosti 16 B alebo 128 bitov.

Následne boli vytvorené testovacie súbory s veľkosťami 0 B, 15 B a 16 B. Na vytvorenie bolo treba takisto použiť metódu odstránenia znaku nového riadku.

V poslednej časti kódu sme použili príkaz na AES šifrovanie daného textu pomocou 128-bitového kľúča sym_key. Linux prostredie vypísalo upozornenie použitia zastaranej derivácie kľúča, to bolo spôsobené verziou softvéru OpenSSL. V novších verziách je pri šifrovaní vyžadované doplnenie viacerých parametrov, ako napríklad pridanie hashu alebo funkcie SALT, ktorá pridáva ďalšiu vrstvu ochrany a opäť šifruje už šifrovaný text. Tieto parametre však pridávajú reťazcu nami nechcenú hlavičku, a teda veľkosti šifrovaného textu sú väčšie. Keďže našim cieľom bolo overiť skutočnú dĺžku šifrovaného textu, tieto parametre neboli použité.

Veľkosť šifrovaného textu je zobrazená v Tab. 10.

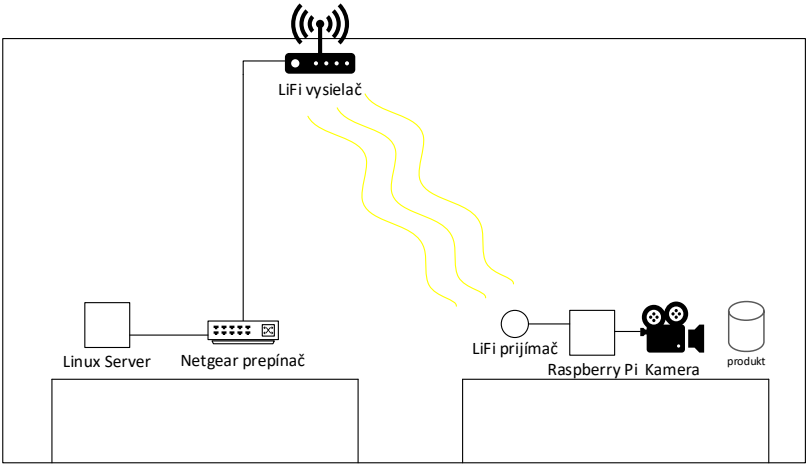
Tab. 3 Porovnanie pôvodného textu so šifrovaným pri AES

Originálny text [B]	**AES Šifrovaný text [B]**
0-15	16
16-31	32
32-47	48
48-63	64
64-79	80
80-95	96
96-111	112
...	...
(x*16) - (x*16+15)	(x+1)*16
,kde x ∈ N	

Na najefektívnejšie využitie prenosového pásma je pri našom návrhu teda najvhodnejšie zvoliť metadáta s veľkosťami x*15 , kde x je prirodzené číslo. Súbory s týmito veľkosťami budú mať v šifrovanej forme rovnakú veľkosť, ako súbory s originálnou veľkosťou o 15 B menšou.

6. Modelové riešenie implementácie eliptických kriviek

Na priblíženie využitia eliptických kriviek v praxi bolo potrebné zvoliť čo najpresnejší model, a teda zvážiť miesto s obmedzenou šírkou pásma, zariadeniami s nízkym výpočtovým výkonom a samotné prostredie vyžadujúce vysokú úroveň bezpečnosti. V našom prípade išlo o miestnosť, v ktorej by mohli byť umiestnené, prípadne pohybujúce sa kamery. Tie sú pripojené k mikropočítaču, ktorý slúži aj ako médium na prenos metadát. Na strope je umiestnený LiFi vysielač, prijímače sú pripojené k mikropočítačom. Vysielač je pripojený ethernetovým káblom aj k pevnej časti siete, konkrétne do prepínača Netgear. Do prepínača je káblom pripojený aj server. Obr. 28 znázorňuje nami zvolený model.



Obr. 6 Modelové riešenie implementácie eliptických kriviek

6.1 Popis zariadení

Raspberry Pi 400

Raspberry Pi 400 je kompletný osobný počítač vbudovaný do kompaktnej klávesnice. Využíva operačný systém Raspberry Pi OS, ktorý je založený na systéme Linux. Je to skvelé riešenie na surfovanie na webe, vytváranie a upravovanie dokumentov alebo sledovanie videa. [8]
Známy je najmä kvôli kompatibilite s rôznymi periférnymi zariadeniami, ako napríklad monitory, mikrofóny alebo displeje, a preto sa využíva ako centrálna programovacia jednotka pre tieto zariadenia.

V návrhu bolo toto zariadenie použité na oboch koncoch, teda aj ako zariadenie na natáčanie videa a spracovanie metadát, a aj ako samotný linuxový server.

Oledcomm LiFi Max kit

LiFiMax systém sa skladá z LiFi vysielača a USB dongle prijímača, ktorý môže byť pripojený ku koncovému zariadeniu ako napríklad počítaču alebo tabletu. Prijímač a vysielač spolu komunikujú pomocou infračerveného žiarenia, čím sa eliminuje rušenie so štandardne používanou WiFi. [9]
Medzi vlastnosti patrí okrem iného:

- prenosová rýchlosť dosahujúca 100 Mbit/s,
- bezpečnosť, keďže je prenos založený na neviditeľnom svetle, ktoré nepresiahne steny miestnosti, sieť nie je dostupná mimo miestnosti kde sa vysielač nachádza,
- každý LiFi vysielač podporuje súčasné pripojenie až 16 užívateľov [10].

Zariadenie USB dongle bolo pripojené priamo do Raspberry Pi pomocou microUSB kábla, vysielač bol pomocou ethernet kábla pripojený do Netgear prepínača. Keďže vysielač podporuje funkciu PoE, ethernet kábel bol využívaný súčasne ako médium pre dátový, ale aj energetický prenos.

Netgear PoE prepínač

Netgear GS305EPP je ľahko manažovateľný prepínač poskytujúci základné funkcie, ale aj funkciu PoE s celkovým výkonom 120 W na pripojenie zariadení ako sú VoIP telefóny, WiFi prístupové body alebo IP kamery. Celkovo disponuje piatimi gigabitovými portmi, podporou VLAN tagovania, QoS, prehľadným grafickým užívateľským rozhraním a inými užitočnými funkciami. [11]

V návrhu bola najužitočnejšia práve funkcia PoE, ktorá bola využitá na priame napojenie LiFi vysielača. Prepínač okrem toho aj priamo pripájal linuxový server pomocou ethernet kábla.

Predprípravou implementácie modelu bolo vytvorenie fungujúcej siete a správne zapojenie všetkých zariadení. Fyzické pripojenia boli vytvorené pomocou klasických ethernet káblov CAT6E, kamera a LiFi USB dongle boli pripojené pomocou USB-C káblov. Na správne fungovanie bolo potrebné zabezpečiť vzájomnú viditeľnosť LiFi vysielača a USB dongle, čo indikovali farebné LED diódy na daných zariadeniach. Všetky zariadenia návrhu boli umiestnené v jednej broadcastovej doméne vo VLAN 1.

Konektivita medzi Raspberry Pi a serverom bola zabezpečená pomocou IP protokolu, kedy obidve zariadenia využívali IP adresu z privátneho rozsahu 192.168.9.0/24, konkrétne 192.168.9.245 pre linuxový server a 192.168.9.247 pre Raspberry Pi. Konektivita bola overená pomocou ICMP protokolu, konkrétne príkazom ping, ktorý potvrdil, že sieť fungovala a zariadenia spolu vedeli komunikovať.

6.2 Skript a jeho optimalizácia

Vytvorenie skriptu bolo nutnou podmienkou pre automatizáciu procesu tvorby kľúčov, ich výmeny, šifrovania a dešifrovania metadát a aj samotného prenosu. Jednou z podmienok tvorby skriptu bolo jeho navrhnutie tak, aby obsahoval čo najmenej riadkov kódu a teda, aby bol procesor zariadení čo najmenej zaťažený. Skript tvorí niekoľko častí:

1. nahratie videa a spracovanie metadát,

2. generovanie párov kľúčov,

3. výpočet spoločného tajomstva a odvodenie symetrického kľúča,

4. prenos a šifrovanie/dešifrovanie metadát,

5. prečistenie vytvorených súborov a znovu spustenie skriptu.

6.2.1 Nahratie videa a spracovanie metadát

V prvom kroku bol použitý predinštalovaný program guvcview, ktorý sa po spustení skriptu automaticky spúšťa na Raspberry Pi (host A). Príkaz spúšťa aj nahrávanie samotného videa po dobu dvoch sekúnd s rozlíšením 640x480 pixelov. Po dvoch sekundách sa nahrávanie prerušuje a video sa ukladá do zvoleného priečinka.

Z daného videa sú následne extrahované metadáta, v našom prípade sme zvolili hodnotu pomeru bitov / pixel. Táto hodnota je následne pomocou predinštalovaného softvéru mediainfo uložená do súboru metadata.txt. Keďže výstup obsahuje veľké množstvo medzier, ktoré nepridávajú žiadnu hodnotu, sú tieto medzery odstránené, aby bola zmenšená celková veľkosť súboru metadata.txt. Výsledná veľkosť metadát je v našom prípade štandardne 22-24 B. Obr. 29 ukazuje implementáciu v praxi na hostovi A.

```
#recording video
guvcview --resolution=640x480 --fps=30 --video=/home/martin/diz/videjko.mp4 --video_timer=2 --exit_on_term

#extracting Format info into metadata file
mediainfo -f /home/martin/diz/videjko-1.mp4 | grep "Bits/(Pixel" | tr -d '[:space:]' >> metadata.txt;
```

Obr. 7 Príkaz na nahranie videa a spracovanie metadát

6.2.2 Generovanie párov kľúčov

Po spracovaní metadát začína host A generovať pár kľúčov. V prvom kroku získava súbor, ktorý obsahuje parametre zvolenej eliptickej krivky, teda BrainpoolP256. Následne si generuje privátny kľúč, z ktorého odvodzuje aj verejný kľúč. Verejný kľúč je následne odosielaný serveru (host B).

Medzičasom host B čaká na prijatie verejného kľúča od hosta A a akonáhle ho dostáva, začína si generovať svoj pár kľúčov. Používa identický postup, a teda generovanie súboru s parametrami eliptickej krivky, generovanie privátneho kľúča a následné odvodenie verejného kľúča. Po vygenerovaní kľúčov odosiela host B svoj verejný kľúč hostovi A. Obr. 30 znázorňuje generovanie kľúčov a odosielanie na hostovi A, Obr. 31 na hostovi B.

```
openssl ecparam -name brainpoolP256r1 -out bp256.pem;
openssl ecparam -in bp256.pem -genkey -noout -out Aprivkey.pem;
openssl ec -in Aprivkey.pem -pubout -out Apubkey.pem;

scp Apubkey.pem server@192.168.9.245:/home/server/Desktop/diz;
```

Obr. 8 Príkaz na generovanie kľúčov na hostovi A

```
#wait until host A's public key is received
while [ $(ls $diz | grep -c "Apubkey") -lt 1 ];
do
        sleep 1
done

openssl ecparam -name brainpoolP256r1 -out bp256.pem;
openssl ecparam -in bp256.pem -genkey -noout -out Bprivkey.pem;
openssl ec -in Bprivkey.pem -pubout -out Bpubkey.pem;

#send your public key to the host A
scp Bpubkey.pem martin@192.168.9.247:/home/martin/diz
```

Obr. 9 Príkaz na generovanie kľúčov na hostovi B

6.2.3 Výpočet spoločného tajomstva a odvodenie symetrického kľúča

Po prijatí verejného kľúča od hosta B začína host A počítať spoločné tajomstvo, kedy násobí svoj privátny kľúč s verejným kľúčom hosta B. Rovnakú operáciu vykoná aj host B, a teda násobí svoj privátny kľúč s verejným kľúčom hosta A. Po vypočítaní získavajú obaja hostovia rovnaké spoločné tajomstvo s veľkosťou 256 bitov, alebo 32 bajtov. Z neho si následne vyberajú prvých 16 znakov reťazca, aby bola veľkosť symetrického kľúča rovná 128 bitov, a teda bola zachovaná rovnaká úroveň ochrany – symetrické šifrovanie AES 128-bitové má rovnakú úroveň ochrany ako šifrovania pomocou eliptických kriviek s kľúčom veľkosti 256 bitov. Problémom v linuxovom prostredí je fakt, že na koniec reťazca pridáva medzeru, a tak má symetrický kľúč veľkosť 17 B. Pomocou jednoduchého príkazu je medzera odstránená a dostávame 128 bitové tajomstvo, rovnaké na oboch stranách. Obr. 32 a Obr. 33 znázorňujú použité príkazy.

```
#wait until host B's public key is received
while [ $(ls $diz | grep -c "Bpubkey.pem") -lt 1 ];
do
        sleep 1
done

openssl pkeyutl -derive -inkey Aprivkey.pem -peerkey Bpubkey.pem -out Ashared.txt;


cut -c1-16 Ashared.txt >> symkey.txt;
tr -d "\n" < symkey.txt > sym_key.txt;
rm symkey.txt;
```

Obr. 10 Príkaz na deriváciu symetrického kľúča na hostovi A

```
#create a common secret
openssl pkeyutl -derive -inkey Bprivkey.pem -peerkey Apubkey.pem -out Bshared.txt;

cut -c1-16 Bshared.txt >> symkey.txt;
tr -d "\n" < symkey.txt > sym_key.txt;
rm symkey.txt;
```

Obr. 11 Príkaz na deriváciu symetrického kľúča na hostovi B

6.2.4 Prenos a šifrovanie/dešifrovanie metadát

Po vytvorení rovnakých symetrických kľúčov začína host A šifrovať metadáta, a následne ich odosiela hostovi B. Šifrovanie je vykonané pomocou AES 128, bez pridanej bezpečnostnej funkcie salt, ktorá by inak pridala 20 B hlavičku, čím by sa výrazne zvýšila veľkosť šifrovaných metadát.

Host B čaká na prijatie šifrovaných metadát, a následne ich pomocou symetrického kľúča dešifruje.

Obr. 34 a Obr. 35 znázorňujú príkazy použité pre dané operácie.

```
#encrypt metadata.txt file
openssl enc -aes128 -nosalt -k sym_key.txt -e -in metadata.txt -out enc_metadata.txt

#sending metadata
scp enc_metadata.txt server@192.168.9.245:/home/server/Desktop/diz/enc_metadata.txt
```

Obr. 12 Príkaz na šifrovanie metadát pomocou symetrického kľúča na hostovi A

```
#wait until encrypted metadata file is received
while [ $(ls $diz | grep -c "enc_metadata") -lt 1 ];
do
        sleep 1
done

#decrypt metadata
openssl enc -aes128 -nosalt -k sym_key.txt -d -in enc_metadata.txt -out metadata.txt;
```

Obr. 13 Príkaz na dešifrovanie metadát pomocou symetrického kľúča na hostovi B

6.2.5 Prečistenie vytvorených súborov a znovuspustenie skriptu

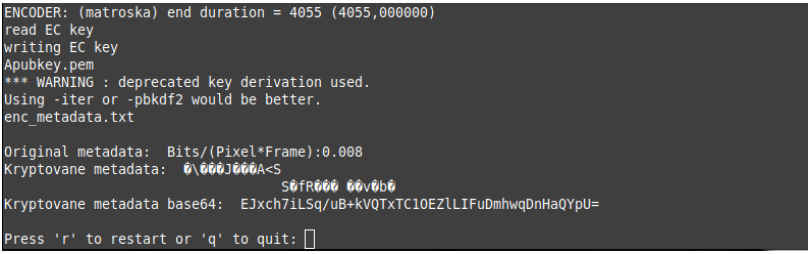
Po vykonaní celého procesu sú na oboch hostoch prečistené vytvorené súbory a objavuje sa výzva na znovuspustenie skriptu. Jednoduchá funkcia wait_restart vyzýva na zadanie klávesu q na ukončenie skriptu, r na znovuspustenie skriptu. Akýkoľvek iný kláves vypíše chybové hlásenie a opäť sa vyžaduje zadanie klávesu „r“ alebo „q“. Obr. 36 znázorňuje riadky kódu pre hosta A, časť skriptu pre hosta B je veľmi podobná.

```
#remove created files
rm -f videjko-1.mp4;
rm -f metadata.txt;
rm -f bp256.pem;
rm -f Aprivkey.pem;
rm -f Apubkey.pem;
rm -f Bpubkey.pem;
rm -f Ashared.txt
rm -f sym_key.txt;
rm -f enc_metadata.txt;

#function to start the script again or to close it

function wait_restart {

        #output for user to determine next action
        read -p "Press 'r' to restart or 'q' to quit: " input

        #if to determine action based on user input
        if [ "$input" == "r" ]; then
                bash "$0"
        elif [ "$input" == "q" ]; then
                exit

        #invalid input
        else
                echo "Invalid input, try again."
                wait_for_input
        fi
}

wait_restart
```

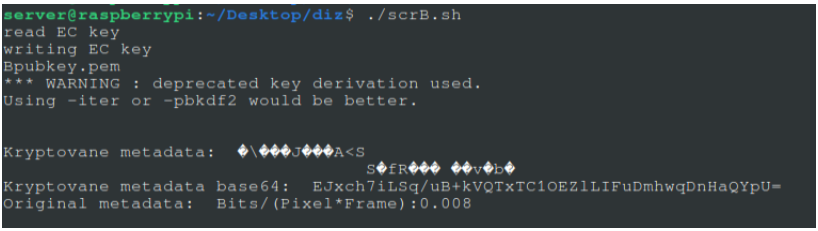
Obr. 14 Príkaz na prečistenie súborov a reštartovanie skriptu na hostovi A

6.3 Overenie funkčnosti skriptu

Na overenie správneho fungovania skriptu boli medzi jednotlivé riadky kódu pridané výpisy originálnych metadát, kryptovaných metadát a kryptovaných metadát vo formáte base64 (pre lepšiu čitateľnosť) na oboch zariadeniach. Ako je možno vidieť na Obr. 37 a Obr. 38, skript funguje správne a host B získal po dešifrovaní rovnaký textový reťazec, aký bol pôvodne zašifrovaný hostom A.



Obr. 15 Overenie originálnych a šifrovaných dát na hostovi A



Obr. 16 Overenie originálnych a šifrovaných dát na hostovi B

Obr. 37 je kvôli prehľadnosti zobrazený až po dokončení natáčania videa, nakoľko tento proces vypísal veľké množstvo riadkov (spúšťanie guvcview, atď.). Na oboch Obr. 37 a Obr. 38 je možné vidieť upozorňujúce hlásenie, že bola použitá zastaraná metóda odvodenia kľúča, čo je spôsobené aj tým, že pri symetrickom šifrovaní AES 128 nebola použitá funkcia Salt.

7. Celkové zhodnotenie výsledkov

V dizertačnej práci bolo vykonaných niekoľko praktických experimentov.

Prvý bol zameraný na dôveryhodnosť eliptických kriviek, konkrétne na porovnanie efektivity kriviek typu Brainpool a NIST. Nakoľko je v širokej verejnosti vedených niekoľko diskusií o dôveryhodnosti eliptických kriviek odporúčaných americkým inštitútom NIST, avšak súčasne sa pýšia najlepšími vlastnosťami a správaním v digitálnom prostredí, predmetom bolo porovnať tieto krivky s krivkami iného typu. Konkurentom sa stali krivky typu Brainpool, ktoré sú založené na matematických odvodeniach a teda nie je možné pochybovať o vytvorení takzvaných „zadných vrátok".

Boli preskúmané algoritmy ECDH a ECDSA. Pri ECDH boli porovnané časy potrebné pre vytvorenie spoločného kľúča. Pri ECDSA to boli časy generovania jednotlivých kľúčov, vytvorenie podpisu správy a čas potrebný na verifikáciu správy. Ukázalo sa, že tieto časy pre porovnateľné krivky boli len mierne vyššie, niekedy dokonca zhodné alebo nižšie pre krivky typu Brainpool. Vzhľadom na dôveryhodnosť je teda možné konštatovať, že je možné používať krivky typu Brainpool a zachovať pocit bezpečia

V druhom praktickom experimente bolo skúmané symetrické šifrovanie AES. Tento symetrický protokol je štandardne využívaný po ECDH, kedy je po vytvorení spoločného ECDH tajomstva používané na zašifrovanie symetrického kľúča, ktorý je prenesený na druhú stranu, a následne sú dáta prenášané a kryptované kvôli efektivite pomocou AES. Zámerom bolo potvrdiť správanie sa AES pri takzvanom dopĺňaní bitov v 16 B blokoch. Jednou z vlastností AES je, že za určitých podmienok nepridáva originálnemu textu pri šifrovaní žiadne bity navyše. Keďže je v prostredí eliptických kriviek častokrát kritickým faktorom aj veľkosť prenesených dát, táto vlastnosť je veľmi užitočná.

Skúmané boli textové reťazce v prostredí Linux, kódované podľa ASCII – teda 8 bitov na jeden znak. Potvrdila sa vlastnosť AES, a teda všetky reťazce dĺžky 0-15 znakov mali po zašifrovaní veľkosť 16 bajtov. Tá istá matematická postupnosť platila aj pre ďalšie 16 bajtové bloky, a teda napríklad reťazce dĺžky 16-31 bajtov mali po zašifrovaní 32 bajtov.

V návrhu modelu využitia eliptických kriviek bola uplatnená teória, ktorá by vedela zefektívniť prenos dát pre určité prostredia- najmä siete, ktoré disponujú malými prenosovými rýchlosťami s nízkym energetickým výkonom

koncových zariadení a vyžadujú vysokú úroveň bezpečnosti. Teória hovorí o vynechaní generovania symetrického kľúča po dokončení ECDH, ale jeho odvodení zo spoločného tajomstva vypočítaného pri ECDH z verejného kľúča opačnej strany a vlastného súkromného kľúča. Tento proces mal ušetriť niekoľko krokov pri každom generovaní nových kľúčov a následnej výmene, čo by priniesol výrazný nárast efektivity prenosu pre zariadenia s malým výpočtovým výkonom, ale aj pre samotnú sieť. Pri procese by bol vynechaný krok generovania symetrického tajomstva, jeho šifrovanie na jednej strane, prenos cez sieť a následné dešifrovanie na druhej strane. V modeli bola kvôli najefektívnejším vlastnostiam použitá eliptická krivka Brainpool P256r1.

V šiestej kapitole bol popísaný tento model implementácie eliptických kriviek do reálneho prostredia. Prostredie sa skladalo z niekoľkých častí. Ako koncové zariadenia boli použité mikropočítače Raspberry Pi 400, na jednej strane ako reprezentácia zariadenia využívaného v podniku na monitorovanie a vyhodnocovanie prostredia, na druhej strane ako server na ukladanie dát. Sieťovými zariadeniami boli LiFi vysielač a prijímač od firmy Oledcomm a prepínač netgear. Na oboch koncových zariadeniach boli vytvorené linuxové bash skripty, ktoré automatizovali všetky procesy na bezpečný prenos dát z jednej strany na druhú pomocou eliptických kriviek. Konkrétne išlo o nahratie videa a spracovanie metadát, následné generovanie párov kľúčov a výmenu pomocou ECDH, odvodenie symetrického kľúča zo spoločného kľúča použitého na šifrovanie a samotného prenosu metadát cez sieť. Overenie funkčnosti skriptov bolo vykonané pomocou vypísania originálneho reťazca metadát a šifrovanej podoby na oboch koncových zariadeniach.

8. Prínosy pre prax a rozvoj vednej disciplíny

Dizertačná práca s názvom **Zabezpečenie prenosu kľúčov v sieti prostredníctvom eliptických kriviek** sa zaoberala skúmaním vlastností eliptických kriviek určených na použitie v podnikovom sektore, efektívnou optimalizáciou pre podnikový sektor a návrhom riešenia implementácie eliptických kriviek do siete.

Cieľom práce bolo skúmať schopnosť eliptických kriviek nahradiť aktuálne používané algoritmy lepším, efektívnejším riešením s dôrazom na zachovanie alebo zvýšenie bezpečnostnej úrovne prenosu. Veľký dôraz sa kládol na aktuálny rozmach IoT zariadení, ktoré sú častokrát rozmerovo malé

a nedisponujú tak vysokým výpočtovým výkonom, aby zvládali množstvá operácií výmeny kľúčov a šifrovania. Ďalším kritériom bol súčasný trend implementácie nových prenosových bezdrôtových riešení, ktoré majú oproti štandardným WiFi protokolom častokrát výhody aj v bezpečnosti ako takej, avšak neponúkajú vysoké prenosové rýchlosti.

Jednotlivé skupiny eliptických kriviek majú rôzne správanie a efektivitu v sieťach. V širokej verejnosti je známe, že krivky typu NIST patria medzi najlepšie. Polemizuje sa však o ich dôveryhodnosti, a preto boli vykonané merania ich výkonu oproti skupine kriviek Brainpool. Za veľký prínos sa dá považovať zistenie, ktoré ukázalo porovnateľné výsledky pri oboch typoch kriviek pri operáciách ECDH a ECDSA. Pri implementácii v praxi teda môžu byť použité krivky typu Brainpool ako alternatíva, ktorá je odbremenená od všeobecnej nedôvery.

V práci bol navrhnutý model implementácie eliptických kriviek do siete. Tento model je základná sieťová topológia, ktorá vie byť priamo implementovaná do podniku. Pozostáva z mikropočítačov ako koncových zariadení, bezdrôtovej časti siete určenej na komunikáciu meracích koncových zariadení s bezdrôtovým vysielačom a drôtovej časti pre následnú komunikáciu vysielača so serverom a zvyškom siete. Návrh je škálovateľný, a je veľmi jednoduché rozšíriť ho a ďalšie komponenty v prípade potreby pokrytia väčších plôch v podniku. Model je zameraný na efektívnu implementáciu eliptických kriviek so zachovaním vysokej úrovne bezpečnosti a šifrovania dát. Samotný proces šifrovania bol odvodený zo štandardného algoritmu ECDH, ktorý je však prispôsobený tak, aby ešte menej zaťažoval koncové zariadenia a samotnú sieť.

Model bol prakticky implementovaný aj v reálnom prostredí, keď boli ako koncové zariadenia použité mikropočítače Raspberry Pi 400, bezdrôtová časť siete pozostávala zo systému LiFiMax Oledcomm a drôtová časť bola založená na klasickom netgear prepínači a ethernetových kábloch. Boli vytvorené linuxové skripty, ktoré automatizovali proces vytvorenia videa a spracovania metadát, párov kľúčov a spoločného tajomstva, prenosu kľúčov a odvodenie symetrickej šifry zo spoločného tajomstva a aj samotné šifrovanie a prenos metadát. Na základe získaných skúseností boli navrhnuté skripty optimalizované na čo najmenší počet operácií a najnižšie zaťaženie procesorov mikropočítačov. Funkčnosť riešenia bola následne aj overená prakticky.

V neposlednom rade bolo prínosom pre prax a vednú disciplínu aj teoretické zhrnutie problematiky eliptických kriviek, ktoré môže slúžiť ako študijný materiál. V práci sa takisto nachádza niekoľko praktických ukážok manipulácií s eliptickými krivkami v softvérovom nástroji openssl v linuxovom prostredí, ktorý v súčasnosti podporuje širokú škálu možností operácií s rôznymi typmi eliptických kriviek.

Prínosy pre prax a rozvoj vednej disciplíny sa dá zhrnúť v nasledovných bodoch:

- boli zhrnuté teoretické poznatky z oblasti využitia eliptických kriviek na ochranu sietí
- boli preskúmané vlastnosti rôznych typov a skupín eliptických kriviek,
- bola dokázaná konkurencieschopnosť kriviek typu Brainpool oproti krivkám NIST,
- boli preskúmané možnosti rôznych operácií s eliptickými krivkami v softvérovom nástroji openssl pracujúcom v linuxovom prostredí,
- bol navrhnutý model siete na použitie v podnikovom prostredí,
- bolo navrhnuté riešenie efektívneho využitia eliptických kriviek v sieti so zreteľom na zníženie zaťaženia potrebného výpočtového výkonu a operačnej pamäte,
- bolo prakticky implementované a overené riešenie využitia eliptických kriviek na ochranu sietí na reálnych zariadeniach,
- návrh, optimalizácia a overenie navrhnutých skriptov.

Záver

Kryptografia pomocou eliptických kriviek má veľký potenciál do budúcnosti, najmä kvôli jej vlastnostiam. Je to vhodná technológia, ktorá konkuruje aktuálne najrozšírenejším asymetrickým systémom. Dá sa konštatovať, že kryptografia pomocou eliptických kriviek má veľký potenciál stať sa viac používanou šifrovacou technológiou, najmä kvôli výhodám oproti aktuálne najpoužívanejším asymetrickým šifrovacím systémom kvôli:

- nižšej energetickej náročnosti,
- nižšieho potrebného výpočtového výkonu,
- výrazne kratších šifrovacích kľúčov pre rovnakú úroveň bezpečnosti,
- rýchlejšiemu generovaniu samotných kľúčov.

Tieto fakty podporujú najmä možnosť využívať eliptické krivky na šifrovanie komunikácie medzi IoT zariadeniami, ktoré majú v súčasnej dobe čoraz väčšiu popularitu. Nakoľko medzi IoT zariadenia patria aj rozmerovo malé zariadenia, môže byť nesmierne ťažké až nemožné implementovať do nich procesor, ktorý by zvládal efektívne a bezpečné šifrovanie pomocou aktuálne rozšírených systémov, ako napríklad RSA. Na druhej strane tieto zariadenia môžu vyžadovať maximálnu úroveň ochrany prenosu dát, keďže prenášané dáta môžu obsahovať citlivé informácie.

Dizertačná práca s názvom **Zabezpečenie prenosu kľúčov v sieti prostredníctvom eliptických kriviek** skúma vlastnosti praktického použitia eliptických kriviek. Zameriava sa na faktory ako je dôveryhodnosť, úroveň bezpečnosti a možnosť efektívnej implementácie do podnikových sietí. Práca takisto obsahuje škálovateľný návrh, ktorý môže byť prispôsobený podľa potrieb a možností firiem. Tento návrh je aj prakticky implementovaný na reálnych zariadeniach a detailne popísaný v práci.

Práca poukazuje na výhody využitia eliptických kriviek v podnikových sieťach a vyzýva na ďalšie skúmanie v tejto oblasti pre ešte lepšie a efektívnejšie využitie v praxi.

Conclusion

Elliptic curve cryptography has great potential for the future, mainly due to its properties. It is a suitable technology that competes with the currently most widespread asymmetric systems. It can be concluded that elliptic curve cryptography has great potential to become a more widely used encryption technology, mainly due to its advantages over the currently most used asymmetric encryption systems:

- lower energy consumption,
- lower required computing power,
- significantly shorter encryption keys for the same level of security,
- faster generation of the keys themselves.

These facts mainly support the possibility of using elliptic curves to encrypt communication between IoT devices, which are currently gaining more and more popularity. Since IoT devices also include small devices, it can be extremely difficult or even impossible to implement a processor in them that would handle efficient and secure encryption using currently widespread systems, such as RSA. On the other hand, these devices may require the maximum level of data transmission protection, as the transmitted data may contain sensitive information.

The dissertation, entitled Securing the Transmission of Keys in the Network using Elliptic Curves, examines the properties of the practical use of elliptic curves. It focuses on factors such as trustworthiness, level of security and the possibility of effective implementation in corporate networks. The work also includes a scalable design that can be adapted according to the needs and capabilities of companies. This proposal is also practically implemented on real devices and described in detail in the work.

The work points out the advantages of using elliptic curves in corporate networks and calls for further research in this area for even better and more effective use in practice.

Použitá literatúra

[1] U.S. Government (USG): Mathematical routines for the NIST prime elliptic curves, 2010, https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.204.9073&rep=rep1&type=pdf

[2] American National Standards Institute, Digital signatures using reversible public key cryptography for the financial services industry (ANSI X9.31), 1998

[3] McGrew D., Igoe K., Salter M., RFC6090 – Fundamental Elliptic Curve Cryptography Algorithm, 2011

[4] Svetlin Nakov, Practical Cryptography for Developers, 2018, https://cryptobook.nakov.com/asymmetric-key-ciphers/ecdh-key-exchange

[5] Elaine Barker, Recommendation for Key Management: Part 1 – General, Máj 2020, dostupné online 20.11.2022, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r5.pdf

[6] Ruta Rimkiene, What is AES encryption and how does it work?, dostupné online 20.11.2022, https://cybernews.com/resources/what-is-aes-encryption/

[7] Corinne Bernstein, Advanced Encryption Standard (AES), dostupné online 20.11.2022, https://www.techtarget.com/searchsecurity/definition/Advanced-Encryption-Standard

[8] Raspberry Pi Ltd, Raspberry Pi 400, dostupné online 20.2.2023, https://datasheets.raspberrypi.com/rpi400/raspberry-pi-400-product-brief.pdf

[9] Chukwuemeka Livinus, Oledcomm LIFIMax Kit Review, dostupné online 20.2.2023, https://www.lifitn.com/blog/lifimaxreview

[10] Oledcomm, LiFiMax- High speed internet through invisible light, dostupné online 20.2.2023, https://lifi.co/wp-content/uploads/2021/08/Brochure-LiFiMax-pdf-HD-24062018.pdf

[11] Netgear, 300 Series SOHO Plus (GS305EPP), dostupné online 20.2.2023,
https://www.netgear.com/business/wired/switches/plus/gs305epp/

Publikované práce

[1] Ďuriga, Roman -- **Köppl, Martin** -- Počarovský, Štefan -- Orgoň, Miloš - Common noise sources and their impact on OFDM highspeed HomePlug PLC networks. In ICUMT 2020. Danvers: IEEE, 2020, s. 163--167. ISBN 978-1-7281-9281-9

[2] Ďuriga, Roman -- **Köppl, Martin** -- Počarovský, Štefan -- Orgoň, Miloš - Impact of household electrical appliances on transmission speed in PLC networks. In ICUMT 2020. Danvers: IEEE, 2020, s. 168--172. ISBN 978-1-7281-9281-9.

[3] Bui Duc, Hung -- **Köppl, Martin** -- Orgoň, Miloš -- Počarovský, Štefan - Measurement of transmission speed in a wireless network with encrypted communication using various security protocols. In ŠILHAVÝ, R. Artificial Intelligence in Intelligent Systems. Cham: Springer, 2021, s. 650--657. ISBN 978-3-030-77444-8.

[4] Bui Duc, Hung -- Počarovský, Štefan -- Orgoň, Miloš -- **Köppl, Martin** - Penetration testing of WiFi networks secured by WEP and WPA/WPA2 protocols. In ŠILHAVÝ, R. Informatics and Cybernetics in Intelligent Systems. Cham: Springer, 2021, s. 571--585. ISBN 978-3-030-77447-9.

[5] **Köppl, Martin** -- Siroshtan, Dmytro -- Orgoň, Miloš -- Počarovský, Štefan -- Bohačík, Antonín -- Kuchár, Pavel -- Holásová, Eva -Performance comparison of ECDH and ECDSA. In CECIT 2021. Piscataway: CPS, 2021, s. 825--829. ISBN 978-1-6654-3757-8.

[6] **Köppl, Martin** -- Paulovič, Matúš -- Orgoň, Miloš -- Počarovský, Štefan -- Bohačík, Antonín -- Kuchař, Karel -- Holásová, Eva -Application of cryptography based on elliptic curves. In CECIT 2021. Piscataway: CPS, 2021, s. 268--272. ISBN 978-1-6654-3757-8.

[7] Róka, Rastislav -- Baroňák, Ivan -- Orgoň, Miloš -- Mokráň, Martin -- Hecl, David -- **Köppl, Martin** -- Letenay, Jakub -- Počarovský, Štefan -Optické

bezdrôtové technológie: Vybrané kapitoly. Bratislava : Vydavateľstvo Spektrum STU, 2022. 187 s. ISBN 978-80-227-5196-4.

[8] Počarovský, Štefan -- Orgoň, Miloš -- **Köppl, Martin** -- Bohačík, Antonín - Comparison of different cloud solutions. In ŠILHAVÝ, R. Software Engineering Perspectives in Systems. Springer Nature ; Cham,, 2022: 2022, s. 611--621. ISBN 978-3-031-09069-1.

[9] **Köppl, Martin** -- Ďuriga, Roman -- Hallon, Jozef -- Bohačík, Antonín -- Orgoň, Miloš -- Počarovský, Štefan -Testing EMC properties of high-speed PLC adapters. In ŠILHAVÝ, R. Cybernetics Perspectives in Systems. Cham: Springer Nature, 2022, s. 582--592. ISBN 978-3-031-09072-1.

[10] Počarovský, Štefan -- **Köppl, Martin** -- Orgoň, Miloš -Flawed implemented cryptographic algorithm in the Microsoft ecosystem. Journal of Electrical Engineering, 73. s. 190--196.

[11] Počarovský, Štefan -- **Köppl, Martin** -- Orgoň, Miloš -- Bohačík, Antonín -Kerberos golden ticket attack. In ŠILHAVÝ, R. -- ŠILHAVÝ, P. -- PROKOPOVÁ, Z. Data Science and Algorithms in Systems. Cham: Springer, 2023, s. 677--688. ISBN 978-3-031-21437-0.

[12] Počarovský, Štefan -- **Köppl, Martin** -- Orgoň, Miloš -Security test of active directory domain services. Research and Development in Material Science, 18. s. 2097--2102.

[13] Hecl, David -- **Köppl, Martin** – Szitkey, Viktor – Grolmus, Andrej – Hozlár, Matúš – Róka, Rastislav – Baroňák, Ivan – Počarovský, Štefan – Orgoň, Miloš – Blažek, Petr -Measurement of transmission characteristics of LiFiMAX, 28.4.2023 prednesený na konferencii CSOC 2023, bude publikovaný.

[14] **Köppl, Martin** – Hozlár, Matúš -- Grolmus, Andrej -- Orgoň, Miloš -- Róka, Rastislav –– Počarovský, Štefan – Blažek, Petr -Prenosové charakteristiky zariadení LiFiMAX, časopis Elektrorevue, bude publikovaný.

Účasť autora na výskumných projektoch

KEGA - 034STU-4/2021 - Použitie progresívnych foriem vzdelávania pri príprave nových vzdelávacích programov v oblasti optických bezdrôtových technológií (2021-2023) Garant: doc. Ing. Rastislav Róka, PhD.