SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Ing. Štefan Počarovský

Autoreferát dizertačnej práce

Zabezpečenie prenosu šifrovacích kľúčov v nezabezpečených sieťach prostredníctvom eliptických kriviek

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

v doktorandskom študijnom programe: telekomunikácie

v študijnom odbore: informatika

Forma štúdia: externá prezenčná

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. Štefan Počarovský

Slovenská technická univerzita v Bratislave
Fakulta elektrotechniky a informatiky
Ústav multimediálnych informačných a komunikačných technológií
Ilkovičova 3, 812 19 Bratislava 1

Školiteľ: doc. Ing. Miloš Orgoň, PhD.

Slovenská technická univerzita v Bratislave
Fakulta elektrotechniky a informatiky
Ústav multimediálnych informačných a komunikačných technológií
Ilkovičova 3, 812 19 Bratislava 1

Oponenti: Prof. Ing. Dan Komosný, PhD.

Vysoké učení technické v Brně
Fakulta elektrotechniky a komunikačních technologií
Ústav telekomunikácií
Technická 3082/12, Královo Pole
616 00 Brno, Česká republika

Ing. Daniel Adamko, PhD.

Riaditeľ školy

Stredná priemyselná škola elektrotechnická
Ul. Karola Adlera č. 5
841 02 Bratislava, Slovenská republika

Autoreferát bol rozoslaný:

....

Obhajoba dizertačnej práce sa bude konať dňa .....2023 **o** .....**h**.
na .....

.....
prof. Ing. Vladimír Kutiš, PhD.

Obsah

1 Úvod

Hlavnou témou práce je oblasť kryptografie. Práca sa delí na teoretickú a praktickú časť. Tieto časti na seba logicky nadväzujú a popisujú princípy kryptografie, ktoré sa neskôr aj používajú v praktickej časti.

V teoretickej časti je postupne popísaný vývoj kryptografie. V práci sú uvedené aj základné obmedzené algoritmy. Neskôr sú popísané princípy základných monoalfabetických šifier a potom aj zložité polyalfabetické šifry (napr. Vigenereova šifra). Ďalej sú vysvetlené symetrické kryptosystémy, asymetrické autentizačné a utajovacie kryptosystémy. Sú tu uvedené základné princípy asymetrickej kryptografie, ako napr. problém faktorizácie veľmi veľkých čísel alebo problém diskrétneho logaritmu na eliptickej krivke. V tejto fáze sú spísané hlavné výhody/nevýhody používania eliptických kriviek v kryptografii, ktoré sa v súčasnosti používajú ako nástupca RSA.

Praktická časť je zameraná na vývoj programu na testovanie eliptických kriviek a ich správnu implementáciu. V úvode praktickej časti je rozobraná reálna chyba Microsoftu pri implementácii knižnice crypt32.dll. Vzhľadom k tejto chybe sa mohol útočník autentizovať ako dôveryhodná osoba, čím bolo možné preniknúť do rôznych informačných systémov. Cieľom bolo upozorniť, že okrem bezpečnosti samotného kryptografického algoritmu je veľmi dôležitá aj správna implementácia požadovaného kryptografického algoritmu.

Ďalší celok praktickej časti sa venuje vývoju aplikácie, ktorá testuje vhodnosť eliptických kriviek na kryptografické účely. Postupne posudzuje základné parametre zadanej eliptickej krivky a vyhodnocuje ich. Samotnú eliptickú krivku uvedený počítačový program aj vykreslí v grafe a pomocou ECDH vypočíta symetrický kľúč na zadanej eliptickej krivke.

V ďalšej časti nasleduje pojednanie o výsledkoch jednotlivých testov, možných vylepšeniach programu v budúcnosti a odporúčanie pre používanie jednotlivých eliptických kriviek. V závere je zhrnutý prehľad tejto práce.

2 Stručný prehľad problematiky

2.1 Rozdelenie kryptosystémov podľa typu

Účelom kryptografie nie je len utajovanie správ, ale aj dôveryhodnosť a autenticita správy. Utajením správy rozumieme, že otvorený text správy nie je voľne dostupný nikomu, okrem požadovaného príjemcu správy. Autentickosť správy znamená, že je možné jednoznačne potvrdiť odoslanie správy konkrétnym odosielateľom (v prípade aj v konkrétnom čase). Podľa toho môžeme rozdeliť kryptografiu na:

1) autentizačné kryptosystémy (cieľ je autentickosť správy)
2) utajovacie kryptosystémy (cieľ je dôvernosť správy)
3) hybridné kryptosystémy (cieľ je dôvernosť aj autentickosť správy)

2.2 Rozdelenie kryptosystémov vzhľadom na distribúciu kľúčov

Okrem rozdelenia kryptosystémov podľa účelu ich použitia môžeme kryptosystémy triviálne rozdeliť aj z hľadiska distribúcie šifrovacích/dešifrovacích kľúčov. Zatiaľ čo symetrické šifrovacie systémy používajú ten istý kľúč (prípadne odvodený kľúč) na šifrovanie/dešifrovanie správy (prípadne pečatenie/odpečatenie správy), v asymetrickej kryptografii sa používa kľúčový pár, verejný kľúč K_V a súkromný kľúč K_S.

2.2.1 Symetrické kryptosystémy

V symetrických kryptosystémoch je dešifrovací kľúč K_D rovnaký ako šifrovací kľúč K_{SI}, prípadne je odvodený od šifrovacieho kľúča K_{SI}. To znamená, že aj keď sú rozdielne, vždy sú navzájom odvoditeľné. V praxi sú však väčšinou oba kľúče zhodné, preto je možné uvažovať, že $K_{SI} = K_D = K$.

2.2.2 Asymetrické kryptosystémy

Problém s distribúciou kľúčov rieši asymetrická kryptografia (kryptografia s párom kľúčov). Prvú zmienku o úvahe použiť dva kľúče na šifrovanie a dešifrovanie uviedli vo svojej práci Diffie Whitfield a Hellman Martin. Tí publikovali v roku 1976 článok „New directions in Cryptography“ [1].Tu sa začína zmienka o použití dvoch kľúčov K_V. – verejný (šifrovací) kľúč

a K_S – súkromný (dešifrovací kľúč). Pri asymetrických kryptosystémoch je veľmi dôležité, že v reálnom čase nie je možné odvodiť súkromný kľúč K_S z verejného kľúča K_V.

2.2.3 Asymetrické systémy typu „Elliptic curve“ (EC)

Bezpečnosť asymetrických kryptosystémov typu EC zabezpečuje problematickosť výpočtu diskrétneho logaritmu na eliptickej krivke. V súčasnosti nie je známy algoritmus na výpočet diskrétneho logaritmu na eliptickej krivke, resp. sa uvažuje, že to je nemožné. Tento typ kryptosystémov sa aktuálne používa najmä na podpisovanie správ. Pri eliptických krivkách je možné použiť omnoho menší výpočtový výkon na zabezpečenie rovnakej bezpečnosti, ako napríklad oproti RSA. Z toho vyplýva, že hlavná výhoda ECC oproti RSA je použitie kľúčov menšej dĺžky pri rovnakom zabezpečení. Na obr. č. 13 a v tab. č.1 [2] je zobrazené porovnanie dĺžok kľúčov pri rovnakom zabezpečení.

Tab. č.1: Dĺžka kľúčov pri RSA a ECC pri rovnakej bezpečnosti, zdroj [2]

Security level (bits)	*RSA key lenght (bits)*	*ECC key lenght (bits)*
80	1024	160-223
112	2048	224-255
128	3072	256-283
192	7680	384-511
256	15360	512-571

2.2.4 Správna implementácia kryptografických algoritmov

Bezpečnosť je jednou z najdôležitejších vlastností, ktoré musí informačný systém spĺňať. Neustály a čoraz rýchlejší rozvoj technológií má veľký vplyv na bezpečnosť výmeny informácií, bezpečnosť operačných systémov a aplikácií a bezpečnosť počítačových sietí. Medzi najpoužívanejšie metodiky na zabezpečenie dôveryhodnej komunikácie patrí kryptografia. Existuje mnoho kryptosystémov, ktoré prešli skúškou času odolnosti voči kryptografickým útokom, no to len za predpokladu, že boli správne použité. Veľmi dôležitý parameter je správna implementácia konkrétneho kryptoalgoritmu. Pri zanedbaní akéhokoľvek parametra, výberu prvočísla, výberu správnej eliptickej krivky (pri ECC) pre správnu

implementáciu kryptografického systému, môže byť celý kryptosystém neúčinný. Príkladom týchto nesprávnych implementácií sú potom narušenia systémov, ako napríklad zlá implementácia crypt32.dll pre ECC [3] alebo v minulosti hack „Sony playstation 3“ skupinou „FailOverFlow“ v roku 2010 [4].

2.2.5 Vhodnosť eliptických kriviek

Nakoľko by mali eliptické krivky zabezpečovať dostatočnú odolnosť voči kryptoútokom, musí byť zabezpečená aj vhodnosť danej krivky pre jej praktickú implementáciu. Ak je eliptická krivka chybne navrhnutá a súčasne takto chybne navrhnutá krivka je aj vygenerovaná, útočník má omnoho väčšiu šancu uspieť pri kryptoútoku na daný kryptografický systém. Prvé vydanie kryptografického štandardu, ktorý špecifikuje eliptické krivky na použitie v praxi bolo v roku 2000 [5]. V súčasnosti sa ponúkajú mnohé šifrovacie sady v protokoloch Secure Shell (SSH), Transport Layer Security (TLS), kde kryptografické algoritmy sú založené na eliptických krivkách [6]. V poslednom čase časť kryptografickej komunity hľadá rôzne alternatívy k súčasne používaným eliptickým krivkám, ktoré by vedeli poskytnúť lepší výkon a vyššiu bezpečnosť.

Posledný takýto známy návrh nových typov kriviek pre praktickú kryptografiu bol v r. 2013 [7], niektoré z týchto eliptických kriviek sa v súčasnosti už používajú. K dôvodom na testovanie kriviek pribudli aj uniknuté skutočnosti od NSA, ktoré naznačujú aj existenciu zadných dvierok v generátore náhodných bitov [8]. Tieto podozrenia vznikli už v roku 2007 [9], následkom čoho sa urýchlila polemika, či by sa mali používané krivky NIST [10] nahradiť krivkami s overiteľne deterministickým generátorom. Napriek uvedeným bezpečnostným obavám sa od počiatku štandardizácie ECC kryptografie dosiahol významný pokrok v účinnosti a bezpečnosti. Významným príkladom sú eliptické krivky, kde sa používajú rôzne „špeciálne“ tvary prvočísel, ktoré umožňujú rýchlejšiu modulárnu aritmetiku. V roku 2007 bol objavený Haroldom Edwardsom [11] zaujímavý tvar eliptickej krivky. Do kryptografie ho implementovali Bernstein a Lange [12] a tento typ krivky sa nazýva Edwardsov skrútený model. Skrútené Edwardsove krivky však nie sú kompatibilné s Weierstrassovými krivkami, ktoré sa používajú v súčasných kryptografických štandardoch.

Krivky NIST [10] boli zahrnuté do mnohých noriem a sú aj používané v niektorých bezpečnostných protokoloch. Alternatívu ku krivkám NIST navrhla nemecká pracovná skupina BRAINPOOL [13]. Ich výber kriviek bol založený aj na ďalších bezpečnostných požiadavkách,

napríklad overiteľne pseudonáhodné generovanie kriviek. Ďalšiu krivku pre efektívny výpočet ECDH navrhol Bernstein [14], to ide o Mongomeryho krivku s názvom Curve25519.

3 Ciele dizertačnej práce

ECC kryptografia nám ponúka novú efektívnejšiu možnosť používania kryptografických kľúčov. Dôležité je však vybrať správnu eliptickú krivku pre daný účel a súčasne použiť aj správnu implementáciu zvoleného algoritmu. Ideálne je vytvoriť nástroj, ktorý by analyzoval zadané eliptické krivky podľa vopred zadaných testov. Takto vytvorený počítačový program by dokázal testovať eliptické krivky, prípadne vygenerovať dáta, ktoré by sa dali následne skopírovať a analyzovať v inom počítačovom programe (napr. Matlab, Excel, SQL). Algoritmus zapísaný do zdrojového kódu by sa však musel optimalizovať, aby bol schopný zadané krivky testovať v kvázi „reálnom čase“.

Riešenie tejto dizertačnej práce bude vyplývať z nasledovných téz:

1. Otestujte eliptické krivky a posúďte ich vhodnosť na zabezpečenie prenosu kľúčov a realizáciu digitálneho podpisu v nezabezpečených sieťach.
2. Navrhnite algoritmus testovania eliptických kriviek vhodný na posúdenie odolnosti týchto kriviek voči kryptoútokom.
3. Na základe získaných skúseností z testovania eliptických kriviek vyberte také eliptické krivky, ktoré sú vhodné na zabezpečenie prenosu kľúčov a realizáciu digitálneho podpisu v nezabezpečených sieťach vzhľadom na ich odolnosť voči kryptoanalýze.
4. Experimentálne overte navrhnuté riešenie zabezpečenia prenosu kľúčov pomocou vybraných eliptických kriviek.

4 Zvolené metódy riešenia

V tejto kapitole navrhneme a popíšeme základné metódy riešenia. Zameriame sa na správnu implementáciu kryptografického algoritmu a popisujeme a analyzujeme reálnu chybu v systémoch Microsoft, kde bol chybne implementovaný ECC algoritmus. Následne sa venujeme vývoju algoritmu a vytvoreniu počítačového programu, ktorý by testoval samotné eliptické krivky. Zdrojový kód algoritmu bol napísaný v jazyku C sharp.

4.1 CVE-2020-0601

14. januára 2020 zverejnila NSA (národná bezpečnostná agentúra) dokument [29], v ktorom identifikovala kritickú zraniteľnosť v systémoch Windows v knižnici crypt32.dll. Chyba sa dala zneužiť pri overovaní X.509 certifikátov, kde mohol útočník sfalšovať certifikáty X.509 spôsobom, že systém Windows by ich potvrdil ako legitímne podpísané dôveryhodnou entitou [30]. Dôsledky tejto chybnej implementácie algoritmu boli pre bezpečnosť systému veľmi významné.

Daná chybná implementácia sa týka nasledovných verzii systémov:

- windows_10: 1607:*:*:*:*:*:*:*
- windows_10: 1709:*:*:*:*:*:*:*
- windows_10: 1803: *: *: *: *: *: *: *
- windows_10: 1809:*:*:*:*:*:*:*
- windows_10: 1903:*:*:*:*:*:*:*
- windows_10: 1909:*:*:*:*:*:*:*
- windows_server_2016:-:*:*:*:*:*:*:*:*
- windows_server_2016: 1803:*:*:*:*:*:*:*:*
- windows_server_2016: 1903:*:*:*:*:*:*:*:*
- windows_server_2016: 1909:*:*:*:*:*:*:*:*

Porovnávané DLL knižnice:

crypt32-beforePatch.dll - (ver. 10.0.18362.476, verzia s chybou, neoveruje základný bod **P**)
crypt32-afterPatch.dll - (ver. 10.0.18362.592, verzia po oprave, overuje základný bod P)
crypt32-20H2.dll - (ver. 10.0.19041.1320, verzia z poslednej aktualizácie Win 10 Pro)

Na obr. č. 1 je porovnanie veľkostí všetkých troch.

```
10.04.2022  16:36    <DIR>          .
10.04.2022  16:36    <DIR>          ..
29.01.2022  10:50         1 383 152 crypt32-20H2.dll
01.03.2020  20:00         1 330 952 crypt32-afterPatch.dll
17.12.2019  11:49         1 327 064 crypt32beforePatch.dll
               3 File(s)      4 041 168 bytes
               2 Dir(s)  433 249 005 568 bytes free
```

Obr. č. 1: Porovnanie veľkostí súborov knižníc crypt32.dll pred a po implementovaní patchu a pri Win 10 Pro 20H2

Je vidieť, že aktualizovaná („opatchovaná“) verzia crypt32.dll je o 3 888 B väčšia, teda musela byť upravovaná, resp. musel byť do nej pridaný ďalší kód. Zmeny sa analyzovali v programe WinMerge, ktorý umožňuje porovnávať binárne súbory. Po zakompilovaní boli knižnice kvôli bezpečnosti a reverznému inžinierstvu obfuskované, čoho výsledkom bola prakticky nemožná orientácia v zdrojovom kóde programu. Na obr. č. 2 je vidieť čiastočné porovnanie časti knižníc crypt32.dll (ver. 10.0.18362.476 vs ver. 10.0.18362.592).

Obr. č. 2: Porovnanie binárnych súborov crypt32.dll (ver. 10.0.18362.476 vs ver. 10.0.18362.592)

V technickej analýze problému [30] použil autor na porovnanie súborov program nástroj BinDiff, ktorý zvládol porovnanie a čiastočnú dekompiláciu binárneho súboru úspešnejšie. Autor sa zameral na analýzu najviac zmenených metód `ChainGetSubjectStatus()` a `CertObjectCache::FindKnownStoreFlags()` a súčasne na vznik nových 5 funkcií.

4.2 Vhodnosť eliptických kriviek

Kryptografia na báze eliptických kriviek sa považuje za nástupcu kryptografie na báze problému faktorizácie veľkých čísel, pretože používa omnoho menšie kľúče (napr. oproti RSA) pri zachovaní rovnakej bezpečnosti. Vzhľadom na to poskytuje veľmi rýchle generovanie kľúčov, rýchlu dohodu kľúčov a rýchle podpisy. Tento typ kryptografie je teda vhodný na šifrovanie (ECIES), elektronické podpisy (ECDSA) a výmenu kľúčov (ECDH). Aby bola krivka použiteľná v kryptografii, musí spĺňať niektoré základné parametre. Ak by nespĺňala základné parametre, problém diskrétneho logaritmu by sa dal rozložiť, tým pádom by sa zjednodušili možné útoky na kryptografické systémy na báze eliptických kriviek. Základnými parametrami, aby bola krivka vhodná pre kryptografické systémy sú:

a) Eliptická krivka musí byť nesingulárna.
b) Musíme používať dostatočne vysokú pravdepodobnosť, že pracujeme naozaj s prvočíslom.
c) Základný bod G(x,y) musí ležať na niektorom z bodov zvolenej eliptickej krivky. Nesmie ležať mimo týchto bodov zvolenej eliptickej krivky.
d) Ideálny stav je, ak postupným sčitovaním základného bodu G(x,y) vypočítame všetky existujúce body na eliptickej krivke.
e) Prvočíslo musí byť dostatočne veľké. V eliptických krivkách, ktoré sú v súčasnosti používané v kryptografií majú tieto prvočísla v dekadickom tvare až 68 digitov (číslic).

4.3 Vytvorenie počítačového programu na analýzu parametrov eliptickej krivky

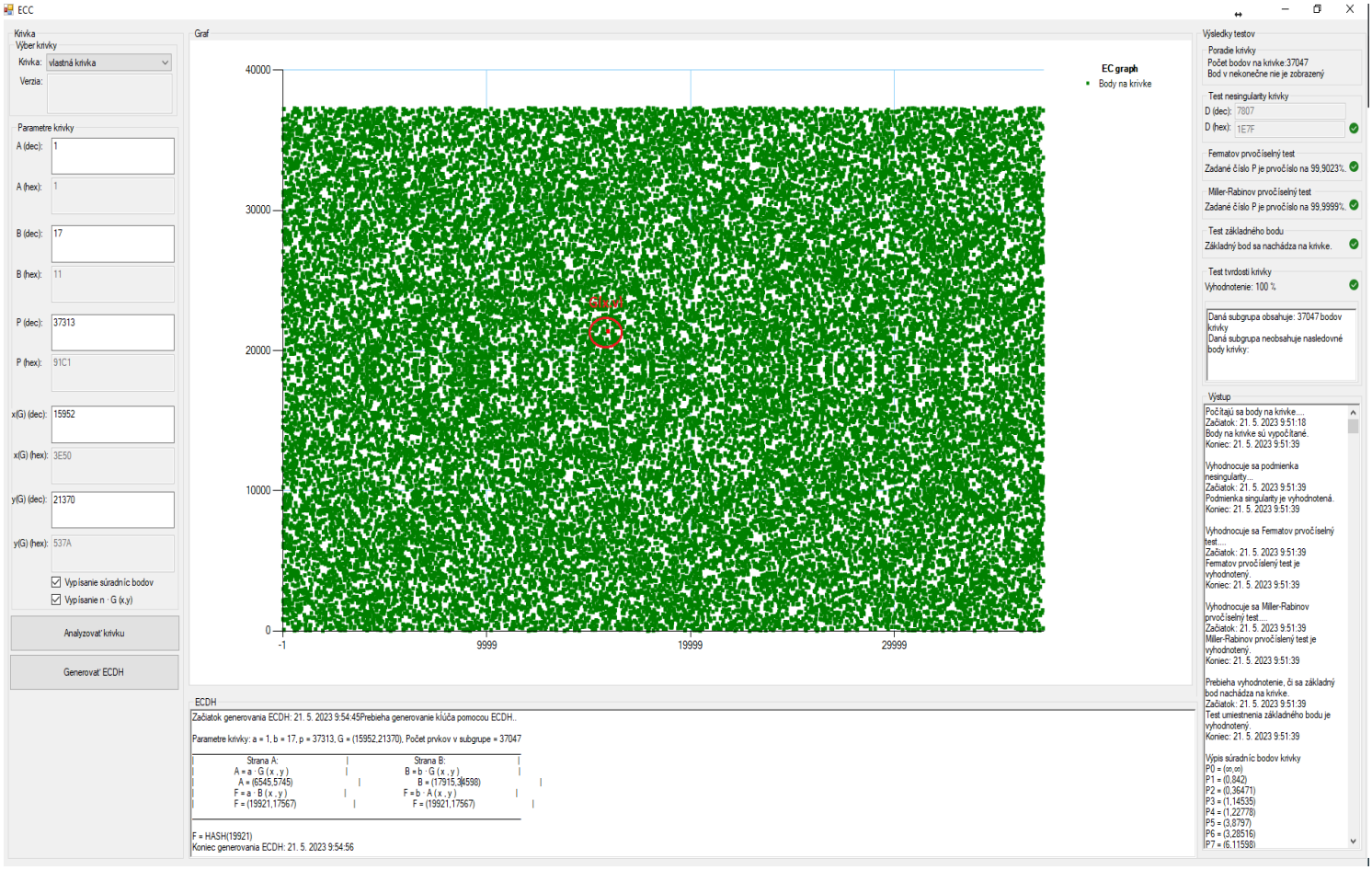
Navrhli sme počítačový program, ktorý overuje tieto parametre, teda overuje vhodnosť eliptických kriviek na kryptografické účely. Tento program môže slúžiť ako na pedagogické účely, tak aj na overovanie jednotlivých parametrov eliptických kriviek. Nakoľko sú algoritmy

výpočtu veľmi zložité, enormne sa zvyšuje aj potrebný výpočtový výkon počítača, čas výpočtu jednotlivých testov je závislý od zložitosti a veľkosti zadaného prvočísla.

Samotný počítačový program umožňuje:

a) Vykresliť zadanú eliptickú krivku nad poľom prvočísel (EC nad Fp), súčasne na nej vykreslí aj polohu základného bodu G(x,y).
b) Vypočítať počet bodov eliptickej krivky (vrátane bodu v nekonečne).
c) Vypočítať diskriminant Δ eliptickej krivky a následne určiť, či ide o krivku nesingulárnu.
d) Pomocou malej Fermatovej vety určiť, či ide o prvočíslo a s akou pravdepodobnosťou.
e) Pomocou Miller-Rabinovho testu určiť, či ide o prvočíslo a s akou pravdepodobnosťou.
f) Určiť, či sa základný bod G(x,y) nachádza v obore hodnôt eliptickej krivky.
g) Určí sa tvrdosť eliptickej krivky, teda vypočíta sa kofaktor.
h) Vypíše súradnice jednotlivých bodov eliptickej krivky (vrátane nekonečna), súčasne vypíše súradnice sčitovania základného bodu G(x,y), súčasne vypíše body, ktoré neobsahuje subgrupa (v prípade ak kofaktor > 1).
i) Po overení jednotlivých testov je program schopný implementovať ECDH na zadanej eliptickej krivke.
j) Ku všetkým získaným parametrom program zobrazuje čas, za aký sa daný algoritmus vykonal.

Na nasledujúcom obrázku č. 26 je znázornený obrázok frontendu počítačového programu. Ten uvádzame ako demonštráciu funkčnosti počítačového programu, ktorý sme naprogramovali. Prostredie pre naprogramovanie daných algoritmov je Visual Studio 2022, algoritmy boli naprogramované v programovacom jazyku C-sharp pod frameworkom .NET Framework 4.7.2.



Obr. č. 26: Frontend počítačového programu pre analýzu parametrov eliptickej krivky

Podľa obr. č. 26 môžeme daný počítačový program rozdeliť na 4 celky, z toho v jednom celku zadávame vstupné dáta a ostatné 3 celky nám poskytujú výstupné údaje po analýze zadanej eliptickej krivky, a to:

a) *Krivka/Parametre krivky*

   Zadávanie vstupných parametrov eliptickej krivky a spúšťanie analýzy zadanej eliptickej krivky

b) *Graf*

   Grafické znázornenie eliptickej krivky

c) *Výsledky testov/Výstup*

   Zobrazenie výstupov jednotlivých testov krivky

d) *ECDH*

Znázornenie experimentálne vygenerovaného kľúča pomocou Diffie – Hellmanovho algoritmu a príslušného času potrebného na vygenerovanie kľúča

4.3.1 Pojednanie o uvedených testoch

Daný počítačový program testuje základné parametre užívateľom zadaných eliptických kriviek vo weierstrassovom tvare. Samotný čas potrebný pre analýzu rastie s veľkosťou zadaných vstupných údajov, najmä s veľkosťou prvočísla p. V navrhnutom algoritme pre výpočty jednotlivých testov sú jednak najdôležitejšími krokmi a súčasne najzdĺhavejšími krokmi nasledovné celky:

a) výpočet bodov na eliptickej krivke,
b) vykreslenie bodov v grafe,
c) výpočet násobkov základného bodu G(x,y),
d) výpis súradníc bodov na eliptickej krivke a súčasne výpis súradníc násobeného bodu G(x,y) v danej cyklickej grupe.

V jednotlivých algoritmoch bol aj problém s číselnými dátovými typmi, ktoré nám obmedzovali veľkosť možných výstupov. Tým pádom sa komplikovali aj konverzie premenných z jedného dátového typu na iný. Napríklad v C-sharpe metóda `Math.Pow()` potrebuje ako vstupný parameter hodnotu typu `double`. Vzhľadom k veľkosti požadovaných vstupov sme pracovali s dátovými typmi `UInt64` a *`Int64`*. Tu nastal problém pri konverzii väčších dátových typov na menšie. Pokiaľ sme to chceli zaistiť, najprv bolo potrebné skonvertovať typ na `string` a potom opäť na iný číselný dátový typ. Pred tým však bolo potrebné zaistiť, že prekonvertovaná hodnota typu `UInt64` je v rozsahu hodnôt `double`, inak by vznikla v programe syntaktická chyba.

Pri výbere eliptickej krivky musíme taktiež zohľadniť jej účel, ktorý má vplyv na výkon pre ECDSA, ECIES a ECDH. Každý typ krivky bol navrhnutý na iný typ implementácie, ktorý sa odráža vo výkonnosti. Krivky NIST sa považujú za rýchlejšie oproti krivkám BRAINPOOL. Pri využití eliptických kriviek BRAINPOOL sa používajú náhodné prvočísla, čím sa líšia od kriviek NIST (tie používajú kvázi-Mersennove prvočísla). To má za následok aj výkonnosť uvedených kriviek. Cieľom použitia náhodných prvočísel bolo vyhnúť sa potenciálnym

bezpečnostným problémom s patentami rýchlych redukčných algoritmov. Výkon kriviek BRAINPOOL nemožno upraviť, aby bol ekvivalentný výkonu krivky NIST. Preto navrhol Bernstein a kol. krivku Curve25519 pre ECDH a Ed25519 pre ECDSA.

5 Záver

V súčasnosti je kryptografia jedna z najdôležitejších odvetví v informatike. Pomáha chrániť dáta pred nepovolanými osobami, pomáha utajovať existenciu daných informácií a súčasne pomáha dokazovať aj autenticitu správy.

Práca bola štruktúrovaná do logických celkov, ktoré sa skladali z teoretickej a praktickej časti.

Prvý hlavný celok dizertačnej práce bol zameraný na teoretickú časť. Po úvode nasledoval krátky prehľad vývoja kryptografie s prvými zmienkami utajovania správ. Od prvých obmedzených algoritmov boli v práci popísané aj klasické symetrické kryptosystémy. Nasledovali asymetrické kryptosystémy založené na probléme faktorizácie veľkých čísel. V ďalšom celku práce boli vysvetlené kryptosystémy založené na probléme diskrétneho logaritmu na eliptických krivkách. Tie boli rozpracované detailnejšie a v tejto práci boli spísané aj hlavné výhody/nevýhody kryptosystémov založených na probléme faktorizácie veľkých čísel (napr. RSA) a kryptosystémov založených na probléme diskrétneho logaritmu na eliptických krivkách.

Druhý celok bol zameraný na praktickú časť. Najprv bola zameraná na skutočnosť, že odolnosť algoritmu zabezpečuje nielen algoritmus samotný, ale aj správna implementácia algoritmu. Analyzovali sme chybu, ktorá bola implementovaná priamo do ekosystému Microsoft s názvom CVE-2020-0601. Tento výskum sme publikovali v dvoch časopisoch [11],[12].

Po pojednaní o tejto chybe sa zameranie práce nasmerovalo na algoritmizáciu a testovanie parametrov eliptických kriviek. Z toho dôvodu sme navrhli a naprogramovali počítačový program, ktorý dokázal zadané parametre eliptickej krivky vyhodnotiť. Parametre, ktoré boli testované nami naprogramovaným počítačovým programom nám môžu udať základnú predstavu o tom, či je daná eliptická krivka vhodná pre použitie v kryptografii.

Nasledovalo pojednanie o jednotlivých testoch a vzniknutých problémoch pri programovaní daného algoritmu, kde sme postupne testovali nami zadané rôzne eliptické krivky. Súčasne sme pojednali o aktuálnych reálnych eliptických krivkách a vhodnosti ich použitia. Vo vývoji tohto počítačového programu by sme chceli pokračovať aj naďalej a cieľom by bolo zrýchliť jednotlivé algoritmy zadaných testov.

Nakoľko je ECC kryptografia jasným kandidátom na nasledovníka kryptografie založenej na problematike faktorizácie veľkých čísel, stále je potrebné, aby prešla tzv. skúškou času. Súčasne je potrebné, aby sa zamedzilo chybným implementáciám tohto typu kryptografie a aby sa obmedzili typy eliptických kriviek, ktoré sú zraniteľné voči útokom.

6 Prínosy dizertačnej práce

Táto dizertačná práca sa zaoberala skúmaním eliptických kriviek vhodných pre kryptografiu a súčasne posudzovala parametre zadaných eliptických kriviek. Cieľom práce bolo navrhnúť algoritmus, ktorý by posúdil vhodnosť eliptickej krivky na použitie v kryptografii, otestovať niekoľko kriviek. Po takto navrhnutom riešení sme chceli experimentálne overiť naše riešenie, a to pomocou algoritmu pre prenos kľúčov na vybraných eliptických krivkách. Následne sme chceli vybrať eliptické krivky, ktoré by sme odporučili pre ECDH a ECDSA.

V prvej časti práce sme spísali súhrn aktuálnych poznatkov z oblasti kryptografie, ktoré boli využiteľné pre danú tému. Obrázky a tabuľky ako aj text v tejto časti boli vytvorené z nášho pohľadu pochopenia danej problematiky. Tým sme prispeli k rozšíreniu literatúry a zosumarizovaniu poznatkov v danej oblasti aj v slovenskom jazyku.

V ďalšej časti sme analyzovali a potvrdili chybu v implementácii kryptografickej knižnice crypt32.dll v operačných systémoch Microsoft Windows. Po zverejnení tejto chyby organizáciou NSA bola síce táto chyba aktualizovaná a opravená, no v reálnych firemných infraštruktúrach neustále existujú neaktualizované verzie operačných systémov. Z toho dôvodu sú tieto infraštruktúry extrémne zraniteľné. Starú aj novú verziu knižnice crypt32.dll sa nám podarilo aj čiastočne dekompilovať a dokázali sme, že v aktualizácii pribudla metóda, ktorá overuje zhodu základného bodu.

V nasledujúcej časti sme vytvorili vlastný počítačový program, ktorý testuje jednotlivé základné parametre eliptických kriviek, dané krivky aj vykreslí, súčasne všetky parametre vypíše vo forme, že sa dajú z tohto vytvoreného počítačového programu vyexportovať a ďalej analyzovať iným dostupným softvérom. Takto analyzované parametre nám dávajú základné informácie o vhodnosti výberu eliptickej krivky na kryptografické účely. Na záver dokáže program vypočítať ECDH na danej krivke, čím sa dá overiť správnosť daného riešenia a vypočítaných súradníc jednotlivých bodov. Daný program zobrazí výpočet súradníc bodov

oboch strán, následne zobrazí súradnice vypočítaného spoločného bodu (súradnice privátneho kľúča) a porovná ich. Nami vytvorený algoritmus si pri každom pokuse ECDH vždy vyberie náhodné súradnice na zostavenie verejného kľúča na rovnakej eliptickej krivke. Ak by totiž daný algoritmus vždy vyberal rovnaké súradnice verejného kľúča na tej istej eliptickej krivke, znížili by sme pravdepodobnosť správnosti výpočtov daného systému.

Pri hľadaní literatúry, používaných algoritmov a počítačových programov na výpočet parametrov na eliptickej krivke sme nenašli ucelené riešenie, ktoré by analyzovalo aspoň časť parametrov, ktoré sme vyhodnocovali my v našom programe. Preto sa domnievame, že počítačový program s rovnakými funkciami (ako sme navrhli my) neexistuje, prípadne nie je dostupný verejnosti. Súčasne sme našli len niektoré podobné aplikácie, ktoré napríklad len vykresľujú zadanú eliptickú krivku, no vedia pracovať iba s prvočíslami menšími ako 1000. Nami vytvorená aplikácia vie pracovať aj s enormne veľkými prvočíslami, napríklad krivku s prvočíslom 999 983 analyzuje za necelé 2 hodiny a s prvočíslom do 40 000 analyzuje do 1 minúty. Samozrejme daná rýchlosť závisí od výpočtového výkonu počítača.

V poslednej experimentálnej časti sme pojednali o výsledkoch jednotlivých testoch nami vytvorenej aplikácie. Zamerali sme sa aj na pojednanie o kvalite a rýchlosti kriviek NIST vs. BRAINPOOL. Cieľom použitia kriviek BRAINPOOL bola náhodnosť parametrov, čím sa autori pri návrhu chceli vyhnúť rôznym bezpečnostným rizikám. Aj keď krivky BRAINPOOL oproti krivkám NIST vyžadujú väčší počítačový výkon a viac času na ECDH, prípadne ECDSA, vzhľadom k naštudovanej problematike by sme odporučili využiť Curve25519 pre ECDH a Ed25519 pre ECDSA.

Vyplývajúc z horeuvedeného pojednania je možné zosumarizovať jednotlivé prínosy pre prax a rozvoj vednej disciplíny v nasledujúcich bodoch:

- sumarizácia poznatkov o kryptografií založenej na problematike diskrétneho logaritmu na eliptickej krivke s praktickými ukážkami výpočtov na vybraných eliptických krivkách;
- vytvorenie počítačového programu, ktorý graficky znázorňuje a vykresľuje problematiku eliptických kriviek a uľahčuje tak výber kriviek na použitie; pričom uvedený počítačový program umožňuje vypísať údaje zadanej eliptickej krivky, ktoré je možné vyexportovať a analyzovať aj v iných počítačových programoch (Matlab, SQL, Excel atď.);

- analyzovanie chyby pri implementácii priamo do ekosystému Microsoft s názvom CVE-2020-0601, ktoré bolo nami publikované v dvoch vedeckých časopisoch [11] a [12];
    - na základe nášho výskumu publikovanom vo vedeckých časopisoch [11],[12], upozorňujeme na potrebu pravidelných aktualizácií OS, nakoľko sa vyskytujú verzie OS v ekosystémoch Microsoft-u, kde sú zle implementované kryptosystémy;
- vytvorenie algoritmu na testovanie eliptických kriviek, ktorý umožňuje skvalitniť výber vhodnej krivky pre kryptografické účely, ktorý bol otestovaný na reálnych eliptických krivkách;
- systém overuje Diffie - Hellmanovu výmenu kľúčov na vlastnej zadanej eliptickej krivke v nezabezpečenej sieti.

7 Resumé

This dissertation investigated elliptic curves suitable for cryptography while simultaneously assessing the parameters of the specified elliptic curves.

The aim of the thesis was to design an algorithm that would assess the suitability of an elliptic curve for use in cryptography, to test several curves. Having thus proposed a solution, we experimentally verified our solution by using the key transfer algorithm on the selected elliptic curves. Subsequently, we wanted to select the elliptic curves that we would recommend for ECDH and ECDSA. In this work, we analyzed and confirmed a flaw in the implementation of the cryptographic library crypt32.dll in Microsoft Windows operating systems. Although this flaw was updated and patched after the NSA published this vulnerability, there are still unupdated versions of the operating systems in real corporate infrastructures. This makes these infrastructures extremely vulnerable.

Consequently, we have created a custom computer program that tests each of the basic parameters of the elliptic curves, plots the curves, and at the same time outputs all the parameters in a form that can be exported from the computer program we have created and further analyzed by other available software. Parameters analyzed in this way give us basic information about the suitability of elliptic curve selection for cryptographic purposes. Finally, the program is able to calculate the ECDH on a given curve, which allows to verify the correctness of the given solution and the calculated coordinates of the individual points. The given program displays the computation of the coordinates of the points of both sides, then displays the coordinates of the computed common point (private key coordinates) and compares them. In our search of the literature, algorithms and computer programs used to calculate parameters on an elliptic curve, we did not find a comprehensive solution that analyzed at least part of the parameters we evaluated in our program. Therefore, we believe that a computer program with the same features (as we have proposed) does not exist or is not available to the public.

In the last section, we focused and conceptualized the quality and speed of the NIST vs. BRAINPOOL curves. The goal of using BRAINPOOL curves was the randomness of the parameters, thus the authors wanted to avoid various security risks in the design. Although the BRAINPOOL curves versus the NIST curves require more computing power and more time for ECDH or ECDSA, respectively, given the issues studied, we would recommend using Curve25519 for ECDH and Ed25519 for ECDSA.

8 Publikované práce

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

[2] **Počarovský Štefan**, Orgoň Miloš: Comparison of application dynamics in two types of CLOUD solutions, In ICUMT 2020. Danvers: IEEE, 2020, s. 154--158. ISBN 978-1-7281-9281-9

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

[4] Hung Bui Duc, 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

[5] Hung Bui Duc, **Počarovský Štefan**, Orgoň Miloš: 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

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

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

[8] Róka Rastislav, Baroňák Ivan, Orgoň Miloš, Mokráň Martin, Hecl David, Köppl Martin, Letenay Jakub, **Počarovský Štefan**: Optické bezdrôtové technológie: Vybrané kapitoly, Vydavateľstvo Spektrum STU, 2022. 187 s. ISBN 978-80-227-5196-4

[9] **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

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

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

[12] **Počarovský Štefan**, Köppl Martin, Orgoň Miloš, Boháčik Antonín: Kerberos golden ticket, Z. Data Science and Algorithms in Systems. Cham: Springer, 2023, s. 677--688. ISBN 978-3-031-21437-0

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

[14] David Hecl, Martin Koppl, Viktor Szitkey, Andrej Grolmus, Matus Hozlar, Rastislav Roka, Ivan Baronak, **Štefan Počarovský**, Milos Orgon, and Petr Blazek: Measurement of transmission characteristics of LiFiMAX, príspevok bol odprezentovaný na konferencii CSOC 2023, bude publikovaný v zborníku z konferencie

[15] Martin Koppl, Matus Hozlar, Andrej Grolmus, Milos Orgon, Rastislav Róka, **Štefan Počarovský**, Peter Blazek: Prenosové charakteristiky zariadení LiFiMAX, článok je po recenznom konaní a bude publikovaný v časopise ELEKTROREVUE

9 Riešené projekty

1. 034STU-4/2021 Použitie progresívnych foriem vzdelávania pri príprave nových vzdelávacích programov v oblasti optických bezdrôtových technológií

10 Použitá literatúra

[1] Diffie W., Hellman M.E.:, New direction in cryptographie, IEEE Trans. Info. Theory (1976)

[2] Martinez V.G., Encinas L.H., Ávila C.S.: A Survey of the Elliptic Curve Integrated Encryption Scheme, Journal of computer science and engineering, Volume 2, ISSUE 2, 2010

[3] Maksim Dubyk, Rajat Ravinder Varuni: Examining CVE-2020-0601 Crypt32.dll Elliptic Curve Cryptography (ECC) Certificate Validation Vulnerability, The SANS institute, 01.03.2022

[4] Group FailOverFlow: Console Hacking 2010 – PS3 Epic fails, 27th chaos communication congress, 2010

[5] Certicom Research. Standards for efficient cryptography 2: Recommended elliptic curve domain parameters. Standard SEC2, Certicom, 2000

[6] J.W. Bos, J.A. Halderman, N. Heninger, J. Moore, M. Naehrig, and E. Wustrow. Elliptic curve cryptography in practice (to appear). In *Financial Cryptography and Data Security*, LNCS. Springer, 2014

[7] D. J. Bernstein and T. Lange. SafeCurves: choosing safe curves for elliptic-curve cryptography. http: //safecurves.cr.yp.to, accessed 16 October 2013.

[8] The New York Times. Government announces steps to restore confidence on encryption standards. http://bits.blogs.nytimes.com/2013/ 09/10/government-announces-steps-to-restoreconfidence-on-encryption-standards, 2013.

[9] D. Shumow and N. Ferguson. On the possibility of a back door in the NIST SP800-90 dual ec prng. http://rump2007.cr.yp.to/15-shumow.pdf, 2007

[10] U.S. Department of Commerce/National Institute of Standards and Technology. Digital Signature Standard (DSS). FIPS-186-4, 2013. http://nvlpubs.nist.gov/ nistpubs/FIPS/NIST.FIPS.186-4.pdf

[11] H. M. Edwards. A normal form for elliptic curves. Bulletin of the American Mathematical Society, 44:393–422, July 2007

[12] D. J. Bernstein and T. Lange. Faster addition and doubling on elliptic curves. In K. Kurosawa, editor, ASIACRYPT, volume 4833 of LNCS, pages 29–50. Springer, 2007

[13] ECC Brainpool. ECC Brainpool Standard Curves and Curve Generation. http://www.ecc-brainpool.org/ download/Domain-parameters.pdf, 2005

[14] D. J. Bernstein. Curve25519: New Diffie-Hellman speed records. In M. Yung, Y. Dodis, A. Kiayias, and T. Malkin, editors, Public Key Cryptography – PKC 2006, volume 3958 of LNCS, pages 207–228. Springer, Heidelberg, 2006