VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

KRYPTOANALÝZA POSTRANNÍMI KANÁLY
SIDE-CHANNEL ANALYSIS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Tatiana Novosadová**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Tomáš Gerlich**
SUPERVISOR

BRNO 2019

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní obor **Informační bezpečnost**
Ústav telekomunikací

Studentka: Tatiana Novosadová ***ID:*** 195163
Ročník: 3 ***Akademický rok:*** 2018/19

NÁZEV TÉMATU:

Kryptoanalýza postranními kanály

POKYNY PRO VYPRACOVÁNÍ:

Cílem bakalářské práce je realizace útoku proudovým postranním kanálem na kryptografický systém založený na eliptických křivkách implementovaný na čipové kartě. Seznamte se s experimentálním pracovištěm využívající SAKURA G a W vývojové desky. Teoretická část bude zaměřena na kryptoanalýzu postranními kanály a kryptografii založenou na eliptických křivkách. Navazující praktická část bude rozebírat postup implementace, měření a realizaci útoku proudovou analýzou na nejméně jeden vybraný kryptografický systém. Výsledkem práce budou naměřené průběhy a případně i úspěšně zjištěný šifrovací klíč.

DOPORUČENÁ LITERATURA:

[1] KOCHER, Paul, Joshua JAFFE a Benjamin JUN. Differential Power Analysis. Advances in Cryptology — CRYPTO' 99. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, 1999-12-16, , 388-397. Lecture Notes in Computer Science. DOI: 10.1007/3-540-48405-1_25. ISBN 978-3-540-66347-8.

[2] BRIER, Eric, Christophe CLAVIER a Francis OLIVIER. Correlation Power Analysis with a Leakage Model. Cryptographic Hardware and Embedded Systems - CHES 2004. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, 2004, , 16-29. Lecture Notes in Computer Science. DOI: 10.1007/978-3-540-28632-5_2. ISBN 978-3-5-0-22666-6. ISSN 03029743.

Termín zadání: 1.2.2019 ***Termín odevzdání:*** 27.5.2019

Vedoucí práce: Ing. Tomáš Gerlich
Konzultant:

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

Prehlásenie

Vyhlasujem, že som svoju bakalársku prácu na tému „Kryptoanalýza postranními kanály“ vypracovala samostatne pod vedením vedúceho bakalárskej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autorka uvedenej bakalárskej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušila autorské práva tretích osôb, najmä som nezasiahla nedovoleným spôsobom do cudzích autorských práv osobnostných a som si plne vedomá následkov porušenia ustanovenia §11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákonníka Českej republiky č. 40/2009 Sb.

Brno ....

Podpis autorky

Poďakovanie

Rada by som poďakovala vedúcemu bakalárskej práce Ing. Tomášovi Gerlichovi za odborné vedenie, konzultácie, ochotu a najmä trpezlivosť.

Brno ....

Podpis autorky

Abstrakt

Táto práca sa zaoberá problematikou postranných kanálov, eliptickými krivkami a ich implementáciou v algoritmoch. Teoretická časť je zameraná na bližšie popísanie základných pojmov a rôznych možností útokov postrannými kanálmi, s detailnejším zameraním na prúdovú analýzu. V druhej časti teoretického rozboru je popísaný princíp eliptických kriviek a rôznych protokolov na nich založených. Praktická časť rozoberá implementáciu kryptosystému využívajúceho eliptické krivky na čipovú kartu, využitie experimentálneho prostredia na nameranie prúdových priebehov a ich analýzu.

Kľúčové slová

Kryptoanalýza, postranný kanál, diferenciálna prúdová analýza, eliptické krivky, čipová karta

Abstract

This thesis deals with an issue of side channels, elliptic curves and their implementation in algorithms. Theoretical part is aimed at describing the basic concepts and various possibilities of side-channel attacks, with a more detailed focus on power analysis. The second theoretical part is focused on describing methods of elliptic curves and different elliptic curves-based protocols. The practical part deals with implementation of such cryptosystem on smart card, using the experimental environment to measure power consumption and analysis of this consumption.

Keywords

Cryptanalysis, side channel, differential power analysis, elliptic curves, smart card

NOVOSADOVÁ, T. *Kryptoanalýza postrannými kanály*. Brno 2019. Dostupné také z: https://www.vutbr.cz/studenti/zav-prace/detail/118097. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce Ing. Tomáš Gerlich.

Zoznam obrázkov

Zoznam tabuliek

Obsah

ÚVOD

S rýchlym vývojom spoločnosti sú na bezpečnosť dnes používaných kryptografických protokolov neustále kladené väčšie nároky. Z tohto dôvodu musela byť v mnohých algoritmoch zväčšená dĺžka kľúčov, čo je samozrejme náročnejšie na pamäť zariadenia v ktorom je daný protokol implementovaný a môže tak spomaliť jeho funkciu. Tento problém je však možné vyriešiť implementovaním eliptických kriviek. Eliptické kryptosystémy umožňujú oveľa menšiu dĺžku kľúčov než súčasné kryptosystémy pri zachovaní rovnakej úrovne bezpečnosti. Podľa množstva a komplikovanosti parametrov použitých na vygenerovanie eliptickej krivky je možné určiť si úroveň potrebnej bezpečnosti.

Aj napriek dostatočnej úrovni ochrany, ani kryptosystémy založené na eliptických krivkách nie sú dokonale bezpečné. Ako každý algoritmus, aj tieto majú svoje slabiny a informácie unikajúce z fyzickej implementácie. Za takéto úniky sa považuje napríklad čas spracovania požiadavky, prúdový odber, dokonca aj zvuk. Práve tieto informácie sa využívajú pri útokoch postrannými kanálmi a vedia útočníkovi pomôcť získať údaje o používanom algoritme, či dokonca umožniť získanie kryptografického kľúča.

Cieľom tejto práce je zoznámiť sa so spôsobmi útokov postrannými kanálmi a kryptografiou založenou na eliptických krivkách. V praktickej časti prebehne implementácia vybraného eliptického kryptosystému na čipovú kartu a následne prúdová analýza s cieľom získania citlivých informácií, ideálne aj odchytenie tajného kľúča.

Teória je rozdelená do viacerých častí. V rámci prvej kapitoly sú popísané základné pojmy a rôzne možnosti útokov postrannými kanálmi. V druhej kapitole sa pojednáva o téme kryptografie eliptických kriviek, predstaví sa samotný termín a rozoberú sa rôzne algoritmy založené na eliptických krivkách. Následne je spomenutá kryptografia založená na supersingulárnych eliptických krivkách. Ide o kryptografiu budúcnosti, ktorá by mala byť odolná voči kvantovým útokom.

1. KRYPTOANALÝZA

Slovo kryptoanalýza pochádza z gréckych slov *kryptós* – schovaný a *analýein* – uvoľniť alebo odviazať. Ide o štúdiu zameranú na analýzu informačných systémov s cieľom odhaliť skryté aspekty systémov. Je využívaná na narušenie kryptografických bezpečnostných systémov a získanie prístupu k obsahom šifrovaných správ aj napriek neznalosti kľúčov. Existujú rôzne druhy kryptoanalýzy. Napriek tomu, že počiatočný cieľ tejto štúdie bol rovnaký, metódy a techniky sa počas histórie drasticky menili pôsobením zvyšujúcej sa náročnosti rôznych kryptografických systémov. Metódy zamerané na porušenie bezpečnosti moderných kryptosystémov prevažne používajú komplexné matematické problémy, medzi ktoré patrí aj napríklad prvočíslený rozklad [1].

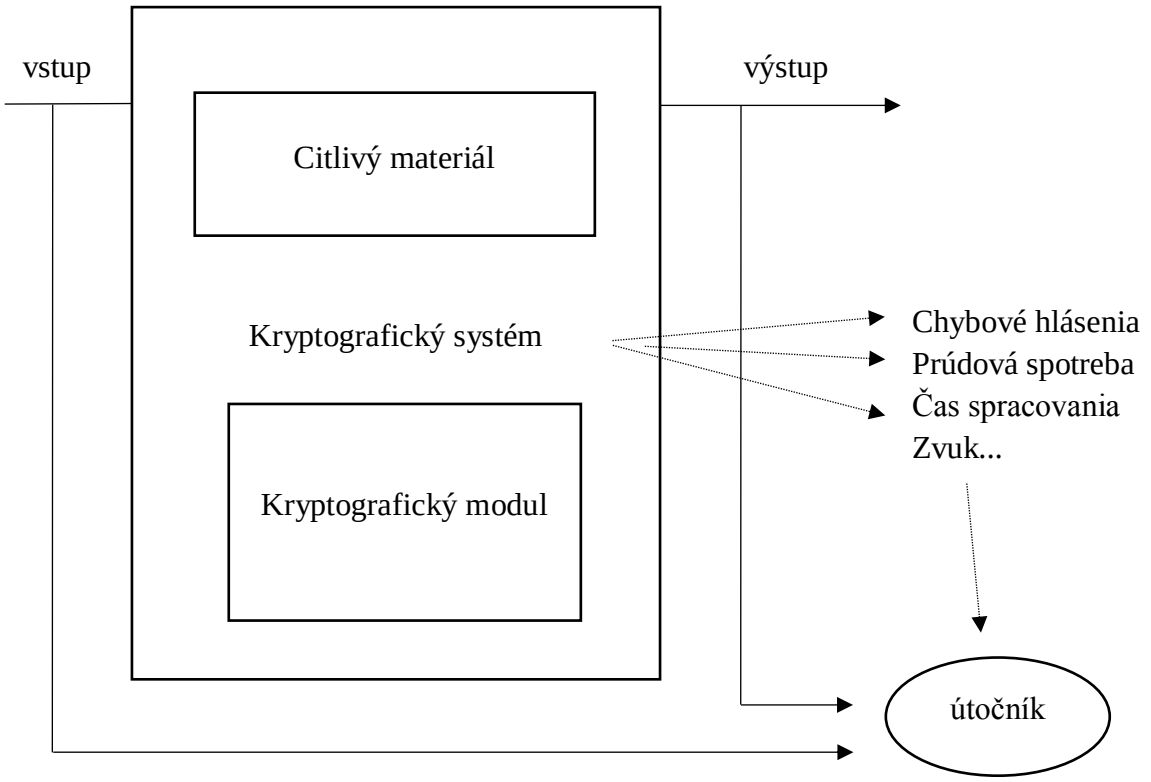
Okrem matematickej analýzy kryptografických algoritmov zahŕňa kryptoanalýza aj útoky vedľajšími kanálmi. Tie sa nezaoberajú samotnými slabými stránkami algoritmov, ale miesto toho z týchto nedostatkov „ťažia“ a využívajú ich vo svoj prospech. Presne na tento typ kryptoanalýzy budú zamerané nasledujúce kapitoly.

Kryptoanalýza postrannými kanálmi

Ako už bolo vyššie spomínané, útok postrannými kanálmi je akýkoľvek útok, ktorý sa na rozdiel od klasickej kryptoanalýzy nesnaží nájsť teoretické slabiny v matematickej štruktúre algoritmu, ale pokúša sa o zneužitie informácií, ktoré unikajú priamo z fyzickej implementácie systému počas behu kryptografického algoritmu.

Niektoré útoky vedľajšími kanálmi vyžadujú technické znalosti systému, iné sú však účinné aj ako útoky na čierne skrinky (napríklad diferenciálna prúdová analýza, kapitola 1.7.2).

Pri útokoch postrannými kanálmi sú sledované rôzne faktory, ako napríklad spotreba elektrickej energie, elektromagnetické úniky, či dokonca zvuk. Tieto informácie môžu útočníkovi pomôcť získať informácie o tom, o aký šifrovací algoritmus sa jedná, odhaliť PIN alebo získať kryptografický kľúč a odchytiť posielanú správu. Podľa sledovaného úniku sa kryptoanalýza vedľajšími kanálmi delí na rôzne typy, ktoré sú podrobnejšie rozobraté na ďalších stranách. Na nasledujúcej strane sa tiež nachádza grafické znázornenie útoku postranným kanálom (obrázok 1).



Obr. 1 Princíp útokov postrannými kanálmi

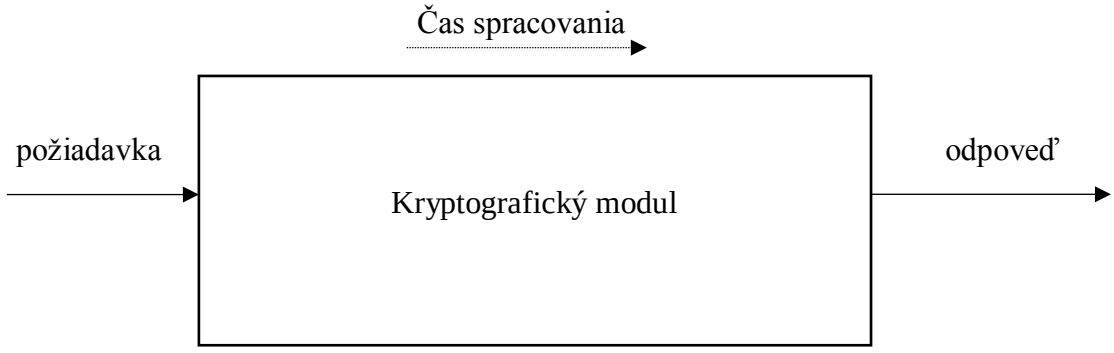
1.1 Časová analýza

Komerčná kryptografia sa už dlho zaoberá tým, ako dlho trvá realizácia určitej kryptografickej implementácie. Čas použitý na zašifrovanie správy, či vytvorenie digitálneho podpisu je často využívaný ako referenčná hodnota pri porovnávaní rôznych kryptografických schém. V prípade, že všetky ostatné faktory sú rovnaké, rýchlosť realizácie je veľmi dôležitá. Čas potrebný na výpočet kryptografickej funkcie závisí nielen na tom čo daná funkcia robí, ale aj na tom, aké vstupy do algoritmu prichádzajú. Napríklad šifrovacia funkcia založená na násobení celých čísiel môže byť rýchlo zrealizovaná. Avšak okrem správ, mnohé kryptografické algoritmy využívajú tajné kľúče ako vstup a tým pádom hodnota kľúča môže ovplyvniť verejne viditeľný čas spracovania [2].

Táto teória bola prvý krát popísaná Paulom Kocherom v roku 1995 [24]. Tiež načrtol spôsoby, akými môže útočník analyzovať časové údaje a z nich vypočítať napríklad podpisy RSA a odvodiť tajný kľúč podpisujúcej entity. Využíva sa doba trvania, ktorá je potrebná k vypočítaniu modulárnej odmocniny pomocou algoritmu „square and multiply“. Tento algoritmus pracuje s jednotlivými bitmi súkromného kľúča. Keď je bit rovný nule, výpočet trvá kratšiu dobu. V prípade, že sa rovná jednej, výpočet trvá dlhšie.

Časový postranný kanál a útok pomocou neho možno realizovať pri algoritmoch, ktoré používajú modulárne mocnenie, napríklad [3]:

- AES
- DES
- DSA
- Diffie-Hellman protokol
- IDEA
- RC5



Obr. 2 Časový postranný kanál

1.2 Chybová analýza

U väčšiny zariadení využívaných na kryptografické operácie sa predpokladá, že spoľahlivo fungujú počas ich využívania, preto sa nepozastavujeme nad tým, či prebiehajúce operácie závisia na zariadeniach v ktorých sú tieto kryptografické moduly implementované. V skutočnosti však hardvérové chyby vzniknuté počas chodu kryptografickej operácie vážne ovplyvňujú bezpečnosť. Tieto chyby, či chybové výstupy vytvárajú nebezpečné postranné kanály a zvyšujú zraniteľnosť šifry. Chybové útoky predstavujú efektívne útoky na hardvérové zariadenia ako napríklad čipové karty. Spôsob akým sa dá zneužiť chybový kanál silno závisí od používaného algoritmu a zariadenia, na ktorom je algoritmus implementovaný. Taktiež samozrejme zaváži o akú chybu ide [5].

Vo všeobecnosti by chybový model mal špecifikovať minimálne tieto aspekty:
- Presnosť, ktorú môže útočník dosiahnuť pri výbere času a miesta na ktorom sa chyba vyskytne počas vykonávania algoritmu
- Dĺžka dát ovplyvnená danou chybou
- Pretrvávanie poruchy – či je chyba dočasná alebo trvalá
- Typ poruchy

1.3 Útok zavádzaním chýb

Ako už bolo spomínané v predchádzajúcej kapitole, chybový vedľajší kanál patrí k najviac nebezpečným kanálom. Okrem chybovej analýzy, kde sú využívané už existujúce nedostatky hardvéru, je tiež možné umelo pridať do kryptografického modulu chyby vytvorené útočníkom. Tie vyvolajú chybové hlásenie, poprípade zlyhanie celého modulu, čo spôsobí nutnosť tohto modulu komunikovať s okolím. Postupným vkladaním chýb a skúmaním chybových hlásení a chovania sa modulu je možné získať niektoré citlivé informácie.

Útočník môže vyvolať chybové hlásenie napríklad týmito spôsobmi:
- Krátkym zvýšením alebo znížením napájacieho napätia
- Ožiarením intenzívnym svetlom
- Extrémnymi teplotami

Týmto spôsobom môže útočník napríklad zmeniť podmienené vetvenie tak, aby bola zvolená zlá vetva, čo mu môže umožniť načítať väčšiu časť pamäte RAM, než by mal normálne povolené. Ak by táto časť obsahovala kľúč, útočník má k nemu prístup [5].

1.4 Elektromagnetická analýza

Pri elektromagnetickej analýze sa využíva skutočnosť, že v elektrickom obvode v dôsledku prechodu elektrického prúdu vzniká striedavé elektromagnetické pole. Takmer každý kryptografický modul obsahuje elektronickú časť, ktoré vytvára elektromagnetické pole. Veľkosť poľa je priamo úmerná aktuálne vykonávanej informácii.

Tento typ útoku je pasívny, čo znamená, že útoky môžu byť vykonané sledovaním bežnej prevádzky cieľového zariadenia bez toho, aby spôsobil fyzické poškodenie. Elektromagnetická analýza sa často vykonáva v spojení s inými útokmi vedľajších kanálov, väčšinou spolu s prúdovou analýzou.

Elektromagnetickú kryptoanalýzu možno rozdeliť na dva druhy – jednoduchú elektromagnetickú analýzu a diferenciálnu prúdovú analýzu.

Simple Electromagnetic Analysis (SEMA), využíva priame sledovanie elektromagnetickej emisie kryptografického modulu. Pri využívaní tejto metódy sa predpokladá znalosť použitého šifrovacieho algoritmu. Za tohto predpokladu je možné zistiť napríklad závislosť elektromagnetických emisií na vstupných dátach. Tento typ elektromagnetickej analýzy nepoužíva žiadne matematické postupy a z toho dôvodu nemôže byť zautomatizovaná.

Druhým spôsobom elektromagnetickej analýzy je DEMA (Differencial Electromagnetic Analysis). Táto metóda, na rozdiel od analýzy SEMA už matematické operácie využíva. Diferenciálna elektromagnetická analýza je založená na zbere dát elektromagnetických emisií emitovaných kryptografickým modulom a ich štatistickou úpravou, kde sú emisie oddelené od nežiadúcich vplyvov okolia, napríklad šumu. Základným spôsobom spracovania získaných štatistických údajov je vytváranie diferenčných priebehov pre signály s rôznymi vstupnými hodnotami. Analýzu DEMA je možné automatizovať a je využívaná pri podrobnejších analýzach [5].

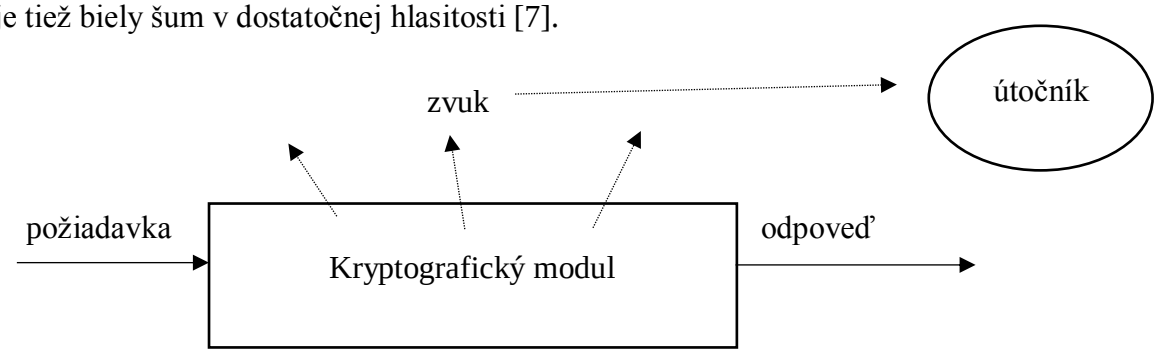
1.5 Akustická analýza

Princíp tohto typu analýzy spočíva v zneužití zvukov vytváraných počítačmi a inými zariadeniami. V dnešnej dobe sa táto kryptoanalýza zameriava na zvuky vyprodukované

klávesnicou využívanou ako vstupné zariadenie do kryptografického modulu (môže ísť o počítačové klávesnice, klávesnice na mobilnom telefóne či bankomate). Každé tlačidlo po stlačení vydáva špecifický zvuk a na základe tejto informácie je možné zistiť napríklad heslo na počítač, mail, či k inému kontu alebo aj bankomatový PIN. Útoky takéhoto typu sa začali objavovať už v roku 2004 [6].

Táto analýza však nie je obmedzená len na použitie klávesnice. Akustické emisie sa vyskytujú aj v cievkach a kondenzátoroch, kvôli drobným pohybom pri prechode prúdu. Tento jav sa prevažne vyskytuje pri kondenzátoroch práve v momentoch, keď v ich mnohých vrstvách dôjde napríklad k piezoelektrickému javu.

Ochranou voči takémuto útoku je vytváranie zvukov, ktoré sa nachádzajú v rovnakom spektre ako sú napríklad zvuky stlačenia klávesov. Zvuky jednotlivých kláves tiež môžu byť náhodne prehrané a tým sa znemožní takýto spôsob útoku. Alternatívnym odporúčaním je tiež biely šum v dostatočnej hlasitosti [7].



Obr. 3 Akustický postranný kanál

1.6 Optická analýza

Únik informácie optickým postranným kanálom je založený na súvislosti zmien logických úrovní a emitácie fotónov. Pri zmene logickej úrovne z „0“ na logickú úroveň „1“ a opačne, tranzistor uvoľňuje časť využitej energie do okolia v podobe fotónov. Špeciálne zaradenie, nazývané PICA, umožňuje toto žiarenie zachytiť a následne analyzovať a poprípade aj rozlúštiť tajný kľúč. Toto zariadenie je však dostupné len v pár vybraných laboratóriách vo svete a z toho dôvodu je analýza optickým postranným kanálom považovaná za jednu z najdrahších metód [4].

1.7 Prúdová analýza

Okrem času vykonania kryptografickej operácie, chybových hlásení, či elektromagnetických emisií patrí aj spotreba energie medzi nebezpečné postranné kanály. Odber prúdu kryptografickým modulom môže poskytnúť veľa informácií o uskutočňovaných operáciách a parametroch používaných pri ich behu. Analýza je mimoriadne efektívna a jej úspešnosť je osvedčená najmä pri útokoch na čipové karty alebo iné zabudované systémy uchovávajúce kryptografický kľúč. Útoky prúdovým postranným kanálom sa pri testovaniach ukázali ako veľmi silné a účinné najmä voči väčšine jednoduchých implementácií symetrických šifier a šifier s verejnými kľúčmi.

Väčšina prebiehajúcich výskumov postranných kanálov je zameraná na prúdovú, respektíve odberovú analýzu. Útok týmto postranným kanálom patrí medzi najznámejší a najprepracovanejší a z toho dôvodu sú zaobstarané príslušné protiopatrenia.

Prúdovú kryptoanalýzu možno rozdeliť na dva druhy – jednoduchú prúdovú analýzu (Simple Power Analysis – SPA) a diferenciálnu prúdovú analýzu (Differencial Power Analysis – DPA). Pri SPA útokoch je cieľom v podstate hádať aká časť kryptografického algoritmu prebieha a hodnoty vstupu a výstupu na základe stopy prúdového odberu. Pri tomto type útoku je teda nutné vedieť aká implementácia kryptografického modulu je používaná. Na druhej strane, útok diferenciálnou prúdovou analýzou túto znalosť nevyžaduje. Útok DPA sleduje a skúma drobné štatistické korelácie medzi tajnými bitmi a spotrebou energie [2].

Paul Kocher, Joshua Jaffe a Benjamin Jun uviedli oba druhy kryptoanalýzy do povedomia už v roku 1999 [9]. Vykonali útok prúdovou analýzou na hardvérovú implementáciu kryptosystému DES. O niečo neskôr po prvý krát využil Jean-Sébastien Coron prúdovú kryptoanalýzu pri útoku na systémy využívajúce eliptické krivky a navrhol metódy ochrany voči jednoduchej aj diferenciálnej prúdovej analýze [25].

1.7.1 Jednoduchá prúdová analýza

Jednoduchá prúdová analýza je druh útoku postranným kanálom, ktorý spočíva v priamej interpretácií prúdovej spotreby, najčastejšie tento proces zahŕňa vizuálne skúmanie grafov (priama interpretácia) vzniknutých počas behu kryptografického modulu na základe jeho prúdového odberu. Rozdiely v spotrebe energie nastávajú pri vykonávaní rôznych operácií, napríklad rôzne úlohy vykonávané mikroprocesorom budú mať rôzne profily spotreby energie. Aj napriek tomu, že veľkosti odchýlok v spotrebe nie sú veľké, štandardné digitálne osciloskopy sú schopné tieto zmeny zaznamenať a jednoducho zobraziť, zistenie súkromného kľúča z týchto údajov však v praxi také jednoduché nie je [8]. Jednoduchú prúdovú analýzu možno rozdeliť na dva druhy: single-shot a multi-shot. Jediným rozdielom medzi týmito dvoma metódami je, že pri sigle-shote sa nameria len jeden prúdový priebeh a pri multi-shot metóde sa zaznamenáva niekoľko priebehov pre rovnakú, opakovane zasielanú správu, poprípade viacero správ. Tento útok je samozrejme možný len za predpokladu, že generovanie alebo prenos súkromného kľúča nejakým spôsobom vplýva na prúdovú spotrebu fyzickej implementácie kryptosystému.

Okrem priamej interpretácie môže útok prúdovou analýzou prebehnúť napríklad ako útok pomocou šablón.

Šablónový útok sa skladá z dvoch častí. Prvým krokom je charakterizovanie zariadenia na základe vopred vytvorených šablón. Druhá časť šablónového útoku spočíva využitie už spomínaných šablón pri útoku na zariadenie. Tento typ útoku je vhodné použiť napríklad, ak vlastníme zariadenie, ktoré je zhodné s tým, na ktoré chceme útočiť. Na našom zariadení

teda môžeme zaznamenať prúdový priebeh v jednotlivých častiach prebiehajúceho algoritmu a na základe toho vytvoriť vlastnú šablónu. Ako už bolo vyššie spomínané, pri jednoduchej prúdovej analýze je nutné vedieť na aký konkrétny algoritmus útočím, čo je teda v prípade, že vlastním rovnaké zariadenie jednoduché obísť, ak na našom zariadení nameriame priebehy pre rôzne protokoly. Následne na základe porovnania priebehov môžeme jednoducho zistiť o aký protokol ide a pokúsiť sa získať súkromný kľúč.

Existujú rôzne druhy ochrany voči jednoduchej prúdovej analýze:
- Pridanie šumu k signálu (šum je však možné odstrániť priemerovaním – tento poznatok využíva diferenciálna prúdová analýza)
- Desynchronizácia, falošné inštrukcie.

1.7.2 Diferenciálna prúdová analýza

Zatiaľ čo útoky SPA využívajú primárne vizuálnu analýzu, DPA používa štatistickú analýzu a korekciu chýb na získanie informácií súvisiacich s tajnými kľúčmi. Výhodou je, že pri útoku na zariadenie nepotrebujeme vedieť aký konkrétny kryptosystém je na ňom implementovaný. Použitie diferenciálnej prúdovej analýzy sa skladá z dvoch častí – zbieranie veľkého množstva dát a následne ich analýza. Kvôli potrebe zozbierania dostatočného množstva prúdových priebehov je nutný prístup k napádanému zariadeniu na dlhší časový úsek (niekedy aj viacero dní) aby analýza prebehla úspešne [8].

Na vykonanie útoku prúdovou analýzou je potrebné postupovať podľa nasledujúcich piatich krokov [9], [15].

Krok 1: Voľba medzivýsledku algoritmu
Na začiatok je potrebné zvoliť si medzivýsledok kryptografického algoritmu, ktorý je vykonávaný zariadením. Táto hodnota musí byť funkciou *f(d,k),* kde *d* sú známe vstupné dáta (spravidla otvorený, či zašifrovaný text) a *k* predstavuje malú časť šifrovacieho kľúča, ktorú je možno odhadnúť (napríklad prvý bajt).

Krok 2: Meranie prúdovej spotreby
Druhým krokom je meranie prúdovej spotreby zariadenia, na ktorom je implementovaný kryptografický modul, počas vykonávania rôznych operácií ako napríklad šifrovanie alebo dešifrovanie rôznych blokov dát *D*. Pre všetky operácie šifrovania a dešifrovania je pri útoku potrebné poznať hodnoty spracovávaných dát d, ktoré sa podieľajú na výpočte medzivýsledku spomínaného v prvom kroku. Hodnoty známych dát vytvoria vektor $\boldsymbol{d} = (d_1, \ldots, d_D)'$, kde d_i označuje výsledok *i*-teho spracovania bloku vstupných dát.

Pri vykonávaní týchto operácií je zaznamenávaný prúdový odber zariadenia. Každému priebehu spotreby $\boldsymbol{t_i'} = (t_{i,1}, \ldots, t_{i,T})$, zodpovedá jedna hodnota spracovávaných dát d_i. Prúdová spotreba je meraná pre každý blok dát *D*, tým pádom je možné priebehy zapísať ako maticu **T** veľkosti $D \times T$. Pre úspešný výsledok po aplikovaný diferenciálnej prúdovej analýzy je potrebné, aby namerané prúdové priebehy boli správne zarovnané – hodnoty

spotreby v ľubovoľnom stĺpci $\mathbf{t}_j$ matice **T** musia zodpovedať rovnakej operácii. Tento stav je možné dosiahnuť správnou synchronizáciou používaného osciloskopu, poprípade je možné dáta zarovnať pomocou softwaru.

Krok 3: Zostavenie matice hypotéz medzivýsledkov

Následne je potrebné vypočítať hypotetické medzivýsledky pre všetky možné hodnoty šifrovacieho kľúča *k*. Tieto hodnoty je možné zapísať ako vektor $\boldsymbol{k} = (k_1, \ldots, k_K)$. *K* označuje celkový počet možných kľúčov. Jednotlivé prvky vektorov sú nazývané hypotézy alebo odhady kľúča.

Z vektoru dát **d** a vektoru hypotéz všetkých kľúčov je možné vypočítať hypotetické medzivýsledky $f = (d, k)$ pre všetky šifrovacie operácie *D* a pre všetky hypotézy kľúča *K*. Výsledkom bude matica **V** veľkosti $D \times T$, ktorú je možné vypočítať pomocou nasledujúceho vzťahu:

$$v_{i,j} = f(d_i, k_j) \quad i = 1, \ldots, D \;\; j = 1, \ldots, K \qquad (1.1)$$

Stĺpec matice **V** obsahuje medzivýsledky, ktoré boli vypočítané podľa hypotéz kľúča k_j. Hodnota kľúča je prvkom vektoru *k*. Index tohto prvku sa označuje *ck*. Kľúč používaný zariadením zodpovedá prvku k_{ck}. Jeden stĺpec vzniknutej matice obsahuje hodnoty namerané počas procesu šifrovania alebo dešifrovania. Cieľom tohto kroku je teda zistiť, ktorý stĺpec matice **V** bol spracovávaný behom *D* operácií šifrovania a dešifrovania a získať tak k_{ck}.

Krok 4: Určenie závislosti prúdovej spotreby na hypotetických medzivýsledkoch

Predposledným krokom diferenciálnej prúdovej analýzy je namapovanie matice hypotetických medzivýsledkov **V** na maticu **H**, ktorá reprezentuje predpokladané hodnoty prúdovej spotreby. Pri tomto kroku sa využíva simulácia prúdovej spotreby kryptografického zariadenia. Vytvorený model priradí každej hypotetickej hodnote medzivýsledku $v_{i,j}$ predpokladanú hodnotu prúdovej spotreby $h_{i,j}$. Čím viac znalostí má útočník o analyzovanom zariadení, tým lepšiu simuláciu spotreby je schopný vytvoriť a tým zefektívniť útok diferenciálnou prúdovou analýzou.

Krok 5: Porovnanie hypotetických hodnôt s nameranými

Posledný krok spočíva v porovnávaní predpokladaných hodnôt prúdovej spotreby závislej na odhade kľúča (hodnoty v stĺpci $\mathbf{h}_i$ matice **H**) s nameranými priebehmi (hodnoty v stĺpci $\mathbf{t}_j$ matice **T**). Výsledkom je matica **R** veľkosti $D \times T$. Každý prvok matice R je výsledkom porovnania medzi stĺpcami $\mathbf{h}_i$ a $\mathbf{t}_j$. Čím väčšia je hodnota prvku $r_{i,j}$, tým je miera lineárnej závislosti (korelácie) medzi stĺpcami $\mathbf{h}_i$ a $\mathbf{t}_j$ väčšia. Porovnanie môže prebehnúť použitím rôznych metód (napríklad metóda rozdielu stredných hodnôt, metóda založená na korelačnom koeficiente...).

Namerané priebehy vyjadrujú prúdovú spotrebu zariadenia na ktorom je kryptosystém implementovaný, pri vykonávaní šifrovacieho algoritmu pre rôzne vstupné dáta. Medzivýsledok, ktorý bol vybraný v prvok kroku je súčasťou tohto algoritmu. Namerané hodnoty sú teda v určitých časových okamihoch logicky závislé na hodnotách medzivýsledkov. Miesto nameraných priebehov sa označuje ako *ct* (stĺpec t_{ct} obsahuje hodnoty prúdovej spotreby, ktoré závisia na medzivýsledku v_{ck}). Na základe hodnôt v_{ck}boli nasimulované hodnoty prúdovej spotreby h_{ck}. Stĺpce h_{ck} a t_{ct} sú na sebe silno závislé. Ich koreláciou vznikne hodnota $r_{ck,ct}$ v matici **R**, ktorá bude najvyššia v tejto matici. Keďže hodnoty v matici **H** a **T** nepreukazujú podobnú závislosť, všetky ostatné hodnoty v matici **R** budú menšie. Pri útoku teda vieme získať tajný kľúč k_{ck} jednoducho, vďaka nájdeniu najvyššej hodnoty v matici **R**.

V praxi sa však môže stať, že hodnoty v matici **R** dosahujú približne rovnaké hodnoty. To sa najčastejšie stáva v prípadoch, keď sa nenameria dostatočné množstvo priebehov prúdovej spotreby. Čím viac priebehov je nameraných, tým viac prvkov budú matice **H** a **T** obsahovať, na základe čoho bude jednoduchšie charakterizovať vzťah medzi stĺpcami týchto matíc a najvyššia hodnota bude jasnejšie viditeľná.

Spôsoby ochrany proti DPA

Ochrana proti diferenciálnej prúdovej analýze je náročná. Prvým spôsobom ochrany je zníženie veľkosti vysielaného signálu, napríklad výberom operácií pri ktorých dochádza k menšiemu úniku informácií o ich energetickej spotrebe. Takýto spôsob ochrany však len oslabí signál viditeľný útočníkovi, neeliminuje ho. Útočník je teda stále schopný útok vykonať aj na silne oslabenom signáli. V praxi je táto ochrana pomerne úspešná, ale zvyšuje náročnosť implementácie kryptosystému a často aj veľkosť zariadenia [9].

2. KRYPTOGRAFIA ELIPTICKÝCH KRIVIEK

Eliptická krivka je matematický objekt, ktorý je možné popísať rovnicou:

$$E: y^2 = x^3 + ax + b \quad (2.1),$$

kde a a b sú konštanty. Pomocou tejto rovnice možno aplikovať rôzne operácie ako napríklad sčitovanie či odčitovanie. Ak sa pri sčítaní dvoch bodov *P* a *Q* na rovinnej krivke vytvorí priamka, ktorá sa s krivkou opäť pretne až v nekonečne, priamka má s krivkou spoločný bod *O*, tiež nazývaný „nulový bod". Všetky body (*x ,y, a, b*) sú z oboru reálnych čísel, pre využitie v praxi ide o celé čísla, nazývaných „teleso" $\mathbb{F}$ (tiež nazývané pole). Najčastejšie využívanými telesami je $\mathbb{F}(P)$ – obsahuje len prvočísla a operácie modulo *p* a $\mathbb{F}(2^m)$ – binárne teleso s operáciami modulo 2^m, kde $m \in \mathbb{N}$ [13]. Zhrnuté v skratke - eliptická krivka *E*, nad telesom $\mathbb{F}(P)$ s množinou *P(x,y)*, kde *a, b, x, y* sú prvky sú prvky telesa $\mathbb{F}(P)$ má rovnicu

$$y^2 \equiv x^3 + ax + b \ (mod\ p) \quad (2.2).$$

V tomto prípade išlo o všeobecné, najčastejšie používané vyjadrenie eliptickej krivky, známe aj ako skrátená Weisrtassova forma. Existujú však rozličné spôsoby vyjadrenia eliptických kriviek, ako napríklad Montgomeryho ($M_{A,B}: By^2 = x^3 + Ax + x$), Hessianova ($by^2 + axy + by = x^3$), Edwardsova ($y^2 + x^2 = 1 + dx^2y^2$), Jacobianova krivka ($y^2 = cy^4 + 2ax^2 + 1$) a mnohé ďalšie. Každá je využívaná pri rôznych špecifických výpočtoch a rôznych kryptografických operáciách.

Kryptografia eliptických kriviek (ECC) je druh asymetrickej kryptografie s verejným kľúčom založený na algebraickej štruktúre eliptických kriviek nad konečnými poľami [10]. Vyžaduje menšie kľúče v porovnaní s klasickou kryptografiou pri zachovaní rovnakej úrovne bezpečnosti [11].

Tab. 1 Porovnanie odporúčanej veľkosti kľúčov

Ochrana	**Symetrická kryptografia**	**Asymetrická kryptografia**	**Kryptografia eliptických kriviek**
Slabá úroveň	80	1024	160
Momentálne postačujúca	128	3072	256
Dlhodobo postačujúca	256	15360	512

Mimo veľkosti kľúčov, bezpečnosť algoritmov založených na eliptických krivkách závisí na náročnosti riešenia ECDLP (problém diskrétneho logaritmu eliptických kriviek), teda na schopnosti vypočítať násobky bodov a neschopnosti nájdenia násobku pomocou ktorého tieto body vznikli v prípade prístupu k pôvodným bodom. To znamená, že je jednoduché vyrátať $Q = d * P$ ale pri znalosti bodov Q a P by malo byť komplikované vypočítať d, keďže aritmetika eliptických kriviek v konečnom poli sa radikálne líši od klasickej [12].

V praxi sú eliptické krivky využívané najčastejšie v protokoloch na dohodnutie kľúčov, pri digitálnych podpisoch alebo ako pseudonáhodné generátory.

2.1 Algoritmy eliptických kriviek

Než prejdeme k samotným protokolom založeným na eliptických krivkách, je potrebné sa oboznámiť so spôsobom generovania náhodnej eliptickej krivky, s princípom generovania kľúčových párov a samozrejme so šifrovaním a dešifrovaním textu za použitia eliptických kriviek [12].

2.1.1 Generovanie náhodnej eliptickej krivky

Generovanie eliptickej krivky je prvým krokom v každom protokole, ktorý je založený na EC. Najčastejšie ide o generáciu náhodnej krivky, najmä z dôvodu zvýšenia bezpečnosti, keďže pri každom šifrovaní je vygenerovaná nová náhodná eliptická krivka. Výpočty teda nikdy nebudú rovnaké aj pre rovnaký kľúč. Z dôvodu zvýšenia bezpečnosti sa tiež do generačného algoritmu vkladá takzvané *seed,* aby naozaj išlo o náhodnú a nie len pseudonáhodnú krivku.

Eliptické krivky môžu byť vygenerované nad rôznym druhom polí, najčastejšie ide o prvočíselné alebo Galoisove pole, ktoré využíva binárnu reprezentáciu čísel. Keďže v dnešnej dobe takmer všetky výpočty tohto typu prebiehajú na počítači, Galoisovo pole je teda pre dnes využívané protokoly eliptických kriviek ideálne. Postup generovania náhodnej eliptickej krivky je teda v nasledujúcej časti popísaný pre Galoisovo pole. Využitie tohto poľa je tiež priaznivé na náročnosť na pamäť. Body sú už reprezentované v binárnej sústave, takže nie je nutný prepočet z dekadickej sústavy a taktiež je možnosť nahradiť modulo n operáciou XOR pri niektorých aritmetických operáciách, čím sa zníži počet potrebných výpočtových operácií.

Postup generovanie eliptickej krivky nad Galoisovým polom:

1.) Prvým krokom je vygenerovanie seedu g tak, aby $g \geq 160 bitov$

2.) Následne sa vypočíta $H = SHA\text{-}1(g)$

3.) b_0 je bitový reťazec obsahujúci y LSB bitov z H

4.) z je celé číslo, ktorého binárny rozvoj je daný g-bitovým reťazcom seed

5.) Pre všetky $i \in [1, s]$:

a. s_i, je g-bitový reťazec, ktorý zodpovedá binárnemu rozvoju celého čísla $(z + i) mod\ 2^g$
b. spočítame $b_i = SHA\text{-}1(s_i)$

6.) b je element poľa $\mathbb{F}_{2^m}$ získaný spojením reťazcov $b_1, b_2, b_3, \dots, b_s$

7.) Ak $b = 0$ opakujeme celý algoritmus od bodu 1

8.) a je ľubovoľný prvok z poľa $\mathbb{F}_{2^m}$

9.) Eliptická krivka nad poľom $\mathbb{F}_{2^m}$ zodpovedá rovnici $E: y^2 + xy = x^3 + ax^2 + b$

2.1.2 Generovanie kľúčových párov

Keďže kryptografia eliptických kriviek spadá pod asymetrickú kryptografiu, je pre správnu funkciu potrebné, aby si každá komunikujúca strana bola schopná vygenerovať platnú dvojicu kľúčov ešte pred samotným šifrovaním, dešifrovaním, či podpisovaním.

Na generovanie kľúčov sú potrebné tieto parametre:
- Charakteristika poľa q
- Definujúce parametre eliptickej krivky a a b
- Bod G a jeho rád n
- Kofaktor h

Postup generovania kľúčov:

1.) Vyberieme náhodné číslo $d \in [1, n-1]$

2.) Vypočítame bod $Q = d * G$

3.) Bod Q je verejným kľúčom, celé číslo d je súkromný kľúč

4.) Kľúčovou dvojicou je teda *(Q, d)*

2.1.3 Šifrovanie

Základným princípom šifrovania je prevod textu, respektíve nejakých dát na číselné bloky, ktoré sú následne pomocou problému diskrétneho logaritmu prevedené na bod eliptickej krivky.

Vstupom pre šifrovanie je verejný kľúč Q, plaintext (nešifrovaný, otvorený text), respektíve správa m a náhodný bod $P \in E(\mathbb{F}_{p^m})$. Obsah správy m je prevedený do ASCII formátu. Toto vytvorené číslo sa následne rozdelí do jednotlivých blokov, ktorých veľkosť nemôže presiahnuť veľkosť rádu bodu P.

Ďalším krokom je náhodný výber súkromného kľúču d z intervalu $(1, n-1)$ a dopočíta sa bod $A = d * Q$ a bod $B = d * P$ a číslo $C = A_x * b_1\ mod\ p$. Tento postup sa zopakuje pre všetky bloky, ktoré boli prevedené do ASCII formátu.

2.1.4 Dešifrovanie

Proces dešifrovania je zrkadlový proces šifrovania. Na získanie pôvodnej správy zo šifrovaného textu je potrebné vykonať rovnaké kroky, len v opačnom poradí, poprípade inverznými operáciami. Najprv sa dopočíta pomocný bod $D = d * R$. Následne sa postupne dešifrujú všetky bloky b_i a $b_1 = C * B_x^{-1} \; mod \; p$.

2.2 Protokoly založené na eliptických krivkách

Niektoré protokoly klasickej kryptografie založené na probléme diskrétneho logaritmu (DLP) boli upravené na použitie eliptických kriviek (multiplikatívna grupa $\mathbb{Z}_p^*$ je nahradená konečným poľom $\mathbb{F}_p$). Prakticky každý algoritmus, ktorého bezpečnosť spočíva v DLP, je možné adaptovať na algoritmus využívajúci eliptické krivky.

Problém diskrétneho logaritmu
Ide o NP-úplný problém, čo znamená, že jeho riešenie je nemožné nájsť v nedeterministicky polynomiálnom čase. Problémy patriace do skupiny NP-úplné sú považované za najzložitejšie a mimo diskrétneho logaritmu sem patrí napríklad aj rozklad veľkých čísel na prvočísla (faktorizácia), či výber najvhodnejšieho ťahu v šachu.

Na pochopenie tohto problému je nutné najprv definovať samotný logaritmus $y = log_a x$, pre ktorý v obore reálnych čísel $\mathbb{R}$ platí, že $x = a^y$. Ďalším dôležitým pojmom je konečná cyklická grupa G $(\mathbb{Z}_n)^x$. Grupa je matematický pojem definujúci skupinu čísel spĺňajúcich určité vlastnosti (existujú rôzne druhy, ako napríklad už spomínané multiplikatívne grupy, aditívne grupy bodov rovinnej eliptickej krivky a podobne). Cyklická multiplikatívna grupa je skupina, ktorej všetky prvky sú tvorené základným prvkom – generátorom g – a všetky prvky v tejto grupe sú jeho mocninou.

Diskrétny logaritmus je teda definovaný tak, že ak n je prvočíslo a $b, c \in (\mathbb{Z}_n)^x$, potom diskrétnym logaritmom čísla c je k a môžeme to zapísať ako $log_b c = k$, kde k je celé číslo, ktoré spĺňa podmienku $b^k = c$. Nájdenie takého čísla k pre zadané $b, c \in (\mathbb{Z}_n)^x$ (v prípade, že $b \neq 1$) je komplikované a tento problém sa definuje ako problém diskrétneho logaritmu.

Problém diskrétneho logaritmu pre eliptické krivky má podobnú definíciu, keďže sa však pohybujeme v inej grupe, nastanú určité rozdiely (multiplikatívna cyklická grupa $(\mathbb{Z}_n)^x$ je nahradená aditívnou grupou bodov eliptickej krivky, ktorá je častejšie nazývaná pole). Pre daný bod $P \in E(\mathbb{F}_p)$ a $Q = d * P$ je komplikované nájsť d [17].

2.2.1 Diffie-Hellman

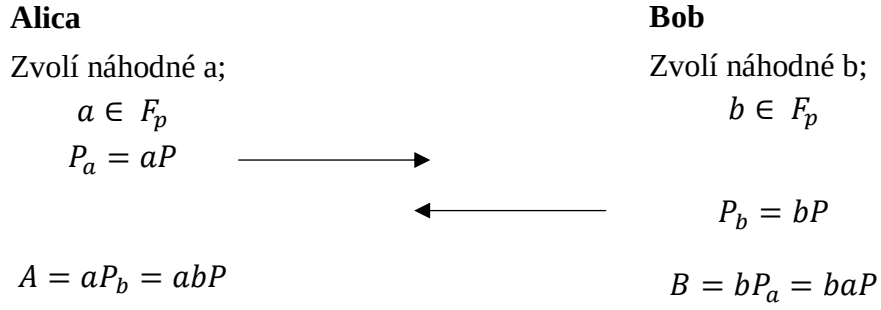
Protokol Diffie-Hellman, skrátene DH alebo D-H, je asymetrický kryptografický protokol, ktorý umožňuje vytvoriť šifrovanú komunikáciu medzi dvoma alebo viacerými stranami cez verejný kanál. Výsledkom spojenia je vytvorenie tajných symetrických kľúčov. Vygenerované kľúče môžu byť priamo použité ako súkromné kľúče, poprípade je možné

použiť ich na odvodenie ďalších kľúčov. Jeho bezpečnosť je založená na probléme diskrétneho algoritmu [16].

Diffie-Hellman založený na eliptických krivkách (označovaný skratkou ECDH – Elliptic Curve Diffie-Hellman) je bezpečnou adaptáciou klasického protokolu Diffie-Hellman s menšou náročnosťou na pamäť.

Generovanie kľúčov pre ECDH
Máme zadanú eliptickú krivku E v poli F_p (p je ideálne prvočíslo) a bod $P \in E$.

1. Alica aj Bob si náhodne zvolia parametre $a, b \in F_p$ (a,b sú prvky z všeobecnej rovnice eliptickej krivky $y^2 \equiv x^3 + ax + b \ (mod\ p)$)
2. Každá strana si následne vypočíta svoje verejné kľúče P_a, P_b - tie môžu byť buď nemenné (a označené za dôveryhodné napríklad pomocou certifikátov), alebo dočasné (takéto kľúče nie sú nutne overené, takže ak je overenie vyžadované, je nutné ho zaistiť inými spôsobmi)
3. Nastane výmena verejných kľúčov cez nezabezpečený kanál
4. Na základe verejných kľúčov sa Alica aj Bob dopočítajú súkromné kľúče A a B podľa rovnice na obrázku 4, pre ktoré musí následne platiť, že $A = B$



Obr. 4 Schéma ECDH – generovanie kľúčov

[14].

2.2.2 ECMQV

MQV (Menezes-Qu-Vanstone) je autentizačný protokol určený na ustanovenie kľúča založený na schéme Diffie-Hellman. Táto schéma vznikla už v roku 1995 a je používaná v štandarde verejných kľúčov IEEE P1363 a ide o protokol štandardizovaný spoločnosťou NIST. Niektoré variácie tohto protokolu môžu podliehať patentom [29].

Potrebné všeobecné informácie:
- Alica má kľúčový pár *(A, a)*, kde *A* je verejný kľúč a *a* je súkromný kľúč
- Tak isto aj Bob má kľúčový pár *(B, b)*, kde *B* je jeho verejný kľúč a *b* je súkromný kľúč.

Následne je potrebné vysvetliť, čo znamená *R'*:
- Bod *R* leží na eliptickej krivke; $R = (x, y)$
- $R' = (x \ mod \ n \ 2^L) + 2^L$
- $L = \frac{(log_2 n)+1}{2}$
- *n* je rád použitého generátoru na vytvorenie eliptickej krivky
- *R'* a *L* prvými bitmi premennej *R*

Ustanovenie kľúčov:

1. Alica si zvolí kľúčový pár *(X, x)*
2. Bob si zvolí kľúčový pár *(Y, y)*
3. Alica dopočíta $S_a = x + X'a \ mod \ n$ a pošle *X* Bobovi
4. Bob vypočíta $S_b = y + Y'b \ mod \ n$ a pošle *Y* Alici
5. Alica aj Bob vypočítajú *K* pomocou kofaktoru *h; (*$h \in [1, n-1]$)
6. Spoločné tajomstvo K bolo stanovené a následne je možné z neho odvodzovať ďalšie kľúče pre protokoly využívajúce symetrické kľúč

Alica **(už má vygenerovaný kľúčový pár A, a)**		**Bob** **(už má vygenerovaný kľúčový pár B, b)**
Zvolí náhodné x; $x \in [1, n-1]$ dopočíta $X = xP$; *P* je bod na eliptickej krivke		Zvolí náhodné y; $y \in [1, n-1]$ dopočíta $Y = yP$; *P* je bod na eliptickej krivke
	X →	
	← Y	
$S_a = x + X'a \ mod \ n$		$S_b = y + Y'b \ mod \ n$
$K = h * S_a(Y + Y'A)$		$K = h * S_b(X + X'A)$
	K=K	

Obr. 5 ECMQV – ustanovenie kľúčov

2.2.3 Digital Singnature Algorithm

Digitálny podpis je identifikačný údaj odosielateľa, ktorým sa zaisťuje integrita a nepopierateľnosť pôvodu dát, autenticitu odosielateľa, poprípade aj časový údaj vzniknutia elektronického podpisu.

Algoritmus digitálneho podpisu (DSA) je možné popísať nasledujúcimi piatimi krokmi:

1. Na začiatok je nutné pomocou hashovacej funkcie vytvoriť otlačok podpisovaného dokumentu (otlačkom je číselná hodnota fixnej dĺžky vypočítaná z binárnej formy dokumentu)
2. Digitálny podpis sa vytvorí tak, že sa otlačok zašifruje pomocou súkromného kľúča odosielateľa
3. Druhá strana nezávisle na prijatom podpise vypočíta otlačok dokumentu rovnakou hashovacou funkciou akou bol digitálny podpis vytvorený
4. Druhá strana následne dešifruje prijatý podpis verejným kľúčom odosielateľa
5. Posledným krokom je porovnanie prijatého otlačku s vypočítaným - v prípade, že sú zhodné, je potvrdené, že dokument bol podpísaný vlastníkom súkromného kľúča.

Keďže bezpečnosť elektronického podpisu závisí na bezpečnom uschovaní súkromného kľúča, mimo dosah neoprávnených osôb) je dôležité po jeho odcudzení okamžite vygenerovať nový súkromný kľúč.

Digitálny podpis založený na eliptických krivkách (ECDSA) je algoritmus odvodený z DSA [18].

Tab. 2 Porovnanie DSA a ECDSA

	DSA	**ECDSA**
Grupa	$\mathbb{Z}_p^*$	$E(Z_p\ )$
Prvky grupy	Celé čísla {1, 2, 3, ..., $p-1$}	Body eliptickej krivky *E*; *P(x,y)*, *Q(x,y)* a bod *O* v nekonečne
Matematické operácie	- Násobenie $g * h$ - Inverzia g^{-1} - Delenie g/h - Mocnenie g^a	- Násobenie c - Negácia $-P$ - Sčitovanie $P+Q, P+P, P+O$ - Odčitovanie $P-Q$
DLP	Pre dané $g \in \mathbb{Z}_p^*$ a $h = g^a \ mod\ p$ je cieľom nájsť a	Pre daný bod $P \in E(\mathbb{Z}_p\ )$ a $Q = d * P$ je cieľom nájsť d

Generovanie digitálneho podpisu pre ECDSA

Máme danú eliptickú krivku $E(F_p)$. Na úspešné generovanie elektronického podpisu potrebujeme poznať nasledujúce parametre:

- q – veľkosť poľa, ideálne $q = p$
- bod $G \in E$; jeho súradnice $x_g, y_g \in F_p$
- n – rád bodu G; $n > 2^{160}$ a $n > 4\sqrt{q}$
- m – dokument (na výpočet otlačku a následné zašifrovanie)
- d – súkromný kľúč
- Q – verejný kľúč

Postup:

1.) Vyberieme náhodné číslo k z intervalu $< 1, n-1 >$

2.) Vypočítame $Q = k * G = (x_1, y_1)$

3.) $r = x_1 mod\ n$ (v prípade, že $r = 0$, je nutné zopakovať prvý krok, podpis by nebol bezpečný, keďže by nebol vôbec závislý na tajnom kľúči ako môžeme vidieť v šiestom bode)

4.) $k^{-1}\ mod\ n = l$

5.) Použijeme secure hash algorithm verziu 1 a výsledný binárny hash prevedieme na celé číslo e; $SHA\text{-}1(m) \rightarrow e$

6.) $s = l\ (e + dr) mod\ n$ (ak $s = 0$, je potrebné začať od znovu)

7.) digitálny podpis → $A = (r, s)$

Overovanie digitálneho podpisu pre ECDSA

1.) Prvým krokom na overenie podpisu je overenie, či premenné prijatého podpisu r a s ležia v intervale $< 1, n-1 >$. V prípade, že táto podmienka neplatí, podpis je neplatný

2.) Následne sa použije hashovacia funkcia, ktorá bola použitá aj na podpisovanie $m = SHA - 1(e)$

3.) Nepoužité bity z podpísanej správy e označíme ako z

4.) Vypočítame $u_1 = zs^{-1}\ mod\ n$ a $u_2 = rs^{-1}\ mod\ n$

5.) Ďalej je nutné vypočítať $(x_1, y_1) = u_1 \times G + u_2 \times Q$; ak $(x_1, y_1) = O$, podpis je neplatný (O je element identity, miesto, kde krivka má nulové hodnoty)

6.) Nakoniec je potrebné zistiť, či $r = x_1$

2.2.4 Edwards Curve DSA

Tento algoritmus kryptografie s verejnými kľúčmi je protokol slúžiaci na vytvorenie digitálnych podpisov a je založený na skrútených Edwardsových krivkách a príncípe Schnorrovho podpisu (jeden z prvých digitálnych podpisov založených na DLP) [21].

Edwardsova krivka je biracionálne ekvivalentná k eliptickej krivke vyjadrenej vo Weistrassovej forme. Rovnice číslo (2.3, 2.4) znázorňujú zápis tejto krivky nad poľom *K*, kde $c, d \in K$ a $cd(1 - c^4.d) \neq 0$:

$$y^2 + x^2 = 1 + dx^2y^2 \qquad (2.3)$$
$$y^2 + x^2 = c^2(1 + dx^2y^2) \quad (2.4)$$

V prípade, že pole *K* je konečné, značná časť všetkých eliptických kriviek nad K môže byť zapísaná vo forme Edwardsovej krivky. Vo väčšine prípadov sa predpokladá, že premenná $c = 1$ [19].

Protokol je vytvorený tak, aby pracoval rýchlejšie ako ostatné protokoly určené na digitálne podpisovanie dokumentov, bez toho aby bola oslabená bezpečnosť. Protokol bol vyvinutý Danielom Bersteinom a jeho tímom v roku 2017, takže sa stále jedná o pomerne nový koncept [20].

Najsilnejšou verziou tohto protokolu je Ed25519. Ide o EdDSA (DSA založené na Edwardsovej krivke) používajúce SHA-512 a krivku 25519 (eliptická krivka ponúkajúca 128 bitovú ochranu) [26]. Tento typ protokolu nevyužíva operácie vetvenia ani indexovanie poľa, teda operácie, ktoré nejakým spôsobom závisia od tajných údajov a vďaka tomuto je odolná voči mnohým útokom postrannými kanálmi.

2.2.5 Integrovaná šifrovacia schéma s eliptickými krivkami

IES (integrated encryption scheme) je hybridná šifrovacia schéma, ktorej bezpečnosť je založená na výpočtovom probléme protokolu Diffie-Hellman, ktorým je problém diskrétneho logaritmu. Štandardizované sú dva druhy tejto šifrovacej schémy – integrovaná šifrovacia schéma diskrétneho logaritmu (DLIES) a Integrovaná šifrovacia schéma s eliptickými krivkami (ECIES), tiež známa ako rozšírená schéma šifrovania, ktorej sa táto kapitola bude venovať.

Potrebné informácie:
- Aký kryptografický balíček *E* sa bude používať, vrátane funkcie na odvodenie kľúča
- Parametre eliptickej krivky (*p, a, b, G, n*) - $a, b \in F_p$ (a,b sú prvky z všeobecnej rovnice eliptickej krivky $y^2 \equiv x^3 + ax + b \ (mod\ p)$), *G* – bod na eliptickej krivke, *n* – rád bodu *G*
- Zdieľané informácie S_1, S_2
- *O* – bod v nekonečne

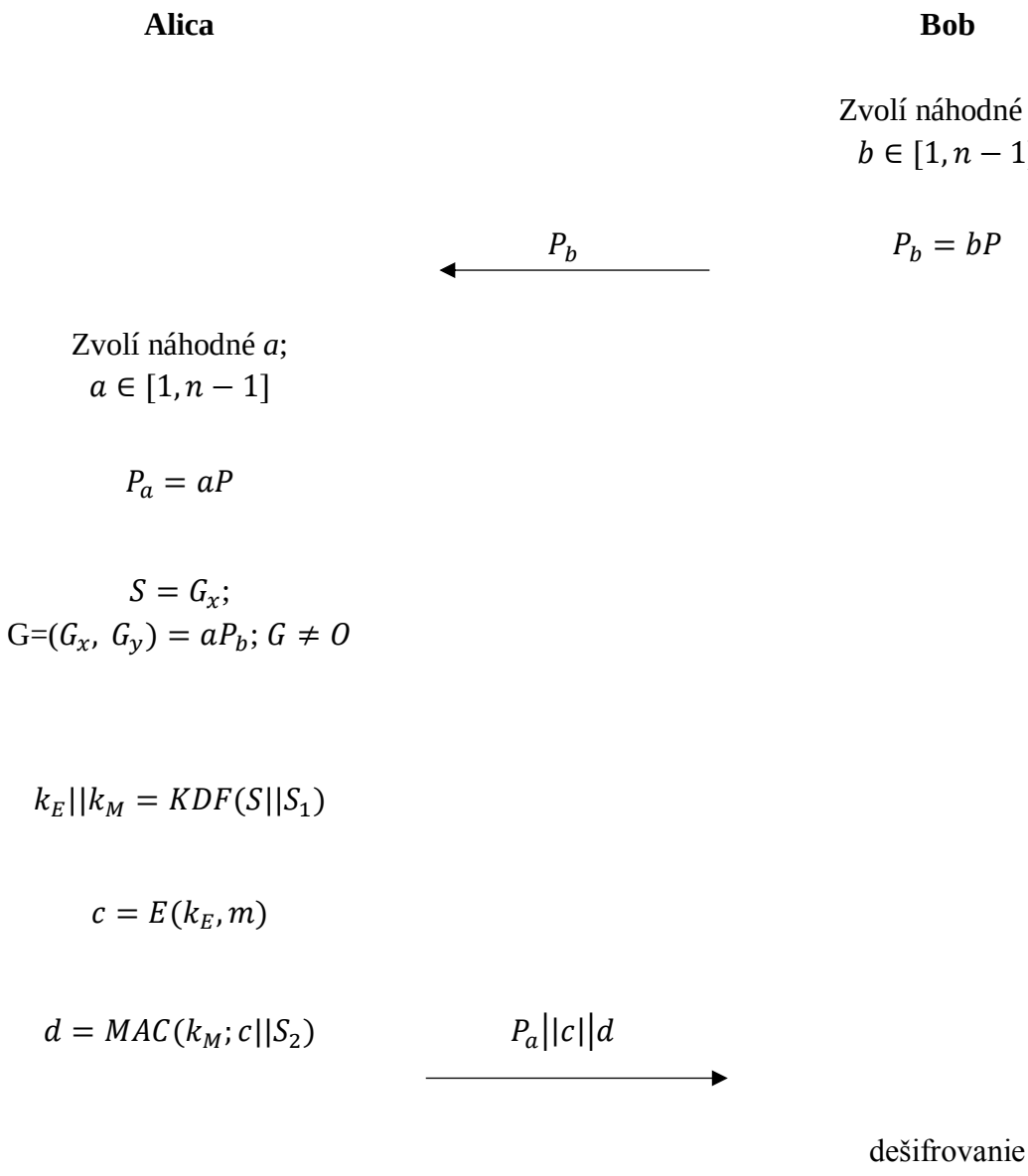
Šifrovanie (zobrazené na obrázku 6)

- Bob si zvolí súkromný kľúč b a dopočíta z neho verejný kľúč P_b
- Alica si zvolí súkromný kľúč a a vypočíta z neho verejný kľúč P_a
- Následne dopočíta zdieľané tajomstvo S
- Použije zvolený mechanizmus na derivovanie kľúčov KDF (key derivation function) a MAC kľúče (message authetication code) na vygenerovanie symetrických šifrovacích kľúčov $k_E||k_M$
- Alica zašifruje správu m
- Dopočíta tag zašifrovanej správy d a zdieľané tajomstvo S_2
- Pošle Bobovi zašifrovanú správu a údaje potrebné na jej dešifrovanie

Dešifrovanie (zobrazené na obrázku 7)

Na dešifrovanie textu $P_a||c||d$ je potrebné spraviť nasledovné:

- Bob dopočíta zdieľané tajomstvo $S = G_x$, kde G = $(G_x, G_y) = bP * P_a$, malo by sa rovnať tajomstvu, ktoré vygenerovala Alica, v prípade, že $G = O$ niekde nastala chyba
- Bob dopočíta kľúče $k_E||k_M$ rovnakým spôsobom ako Alica pomocou mechanizmu na derivovanie kľúčov
- Bob použije MAC na skontrolovanie tagu, niekde nastala chyba ak $d \neq MAC(k_M; c||S_2)$
- Následne Bob použije schému symetrického šifrovania na dešifrovanie správy [14].



Obr. 6 Schéma ECIES – šifrovanie

Alica **Bob**

$$P_a||c||d \longrightarrow$$

$$S = G_x$$
$$\mathrm{G} = (G_x,\ G_y) = \ bP * P_a$$
$$G \neq O$$

$$k_E||k_M = KDF(S||S_1)$$

$$d \neq MAC(k_M; c||S_2)$$

$$m = E^{-1}(k_E; c)$$

Obr. 7 Schéma ECIES – dešifrovanie

2.2.6 Supersingulárna kryptografia eliptických kriviek

Tento druh už spadá do kategórie post-kvantovej kryptografie, čo znamená že jej protokoly by mali byť odolné voči útokom silných kvantových počítačov. Keďže takéto počítače zatiaľ nie je možné vyrobiť, ide skôr o kryptografiu budúcnosti. Dnes používané algoritmy využívajúce eliptické krivky (napríklad už spomínaný protokol ECDH) zatiaľ nie sú voči kvantovým útokom odolné, no pozmenením klasických eliptických kriviek na supersingulárne je možné tento stupeň bezpečnosti dosiahnuť. Protokol ECDH využíva body na jednej krivke, zatiaľ čo supersingulárne krivky sú skupina minimálne piatich eliptických kriviek využívajúcich nezvyčajne veľkých endomorfných okruhov. Ďalším rozdielom je fakt, že súkromné kľúče v post-kvantovej kryptografii sú izogénne. Je to funkcia zobrazujúca body jednej eliptickej krivky do druhej so zachovaním vrcholov. Tajné kľúče sú teda izogenity, ktoré vedú z jednej eliptickej krivky na druhú. Verejným kľúčom je samotná supersingulárna eliptická krivka [22].

Supersingular isogeny Diffie-Hellman and Supersingular isogeny Key Encapsulation

Supersingular isogeny Diffie-Helman key exchange (SIDH) je protokol využívaný na ustanovenie tajného kľúča medzi dvoma stranami na inak nezabezpečenej komunikačnej platforme. Ide o analógiu ku klasickému protokolu Diffie-Helman, či ECDH. Rozdielom je však vyšší stupeň ochrany aj voči kvantovým útokom a to vďaka využitiu supersingulárnych kriviek.

Supersingular isogeny Key Encapsulation (SIKE) je tiež protokol využívajúci supersingulárne eliptické krivky. Je určený na bezpečné zapúzdrenie kľúčov. SIKE je založený na rovnakom princípe ako SIDH, len s tým rozdielom, že navyše využíva Hofheinzovu transformáciu.

Generovanie kľúčov

Pri vytváraní kľúčov sa využívajú verejné parametre *p, E, P₂, Q₂, P₃, Q₃*. Parameter *p* je prvočíslo, pre ktoré platí $p = 2^{e2}3^{e3} - 1$. Supersingulárna eliptická krivka je označená parametrom *E* o ráde $(p+1^2)$. Parametre *P₂, Q₂* sú z množiny $E[2^{e2}]$ a parametre *P₃, Q₃* z množiny $E[3^{e3}]$. Parametre *P₂, Q₂, P₃, Q₃* sú z množiny $\mathbb{Z}$. Najprv je potrebné ustanovenie verejného kľúča na jednej strane (Alica). Tá vygeneruje $sk_2 \in \mathbb{Z}_q$. Pomocou tohto parametra dopočíta verejný kľúč *S₂* rádu 2^{e2} (rovnica 2.5) a ϕ_2 (2.6).

$$S_2 = P_2 + sk_2 Q_2 \quad (2.5)$$

$$\phi_2 : \mathrm{E} \rightarrow \mathrm{E}/[S_2] \quad (2.6)$$

Alica pošle Bobovi $\mathrm{E}/[S_2], \phi_2(P_3), \phi_2(Q_3)$. Bob rovnakým spôsobom ako Alica dopočíta *S₂* a ϕ_2. Zdieľaný kľúč je potom zderivovaný z:

$$\mathrm{E}\,[S_2, S_3] = (\mathrm{E}/[S_2])/(\,\phi_2(P_3) + sk_3\phi_2(Q_3)) \quad (2.7)$$
$$(\mathrm{E}/[S_3])/(\,\phi_3(P_2) + sk_2\phi_3(Q_2))$$

Výhodou je, že sa jedná o pomerne jednoduché protokoly, využívajúce malé kľúče. Aj napriek tomu sú však časovo náročné. Zatiaľ sú odolné voči negenerickým (nenáhodným) útokom [23].

3. PRAKTICKÁ ČASŤ

Cieľom praktickej časti tejto bakalárskej práce bolo implementovať vybraný kryptografický protokol založený na eliptických krivkách na čipovú kartu, následné nameranie prúdových odberov počas chodu aplikácie a vykonanie prúdovej analýzy na vybraný kryptografický systém, ktorej výsledkom bude zobrazenie priebehov a ideálne úspešne zaistený tajný šifrovací kľúč.

3.1 Virtuálne prostredie, čipová karta a implementácia protokolu

Prvým krokom bolo implementovať vybraný kryptografický systém na čipovú kartu, konkrétne na BasicCard. Táto čipová karta je založená na programovacom jazyku Basic, ktorý bol vytvorený už dávno a je dnes pomerne málo využívaný, aj keď ho stále môžeme nájsť napríklad na obmedzených, avšak výkonných zariadeniach ako Raspberry Pi. Konkrétnym typom tohto jazyka využívaným na čipovej karte je ZC-Basic, ide o procedúrovo orientovaný jazyk, podobný QBasic so špeciálnymi úpravami pre procesor a kartové prostredie.

Ako protokol vhodný pre toto zadanie som si zvolila digitálny podpisy založený na eliptických krivkách. Ide o pomerne jednoduchý a na pamäť nenáročný protokol, ktorý je bližšie popísaný v kapitole 2.2.3. Vybraný protokol využíva Secure Hash Algortihm verziu 2, konkrétne SHA-256 (jednosmerná funkcia s výstupom o veľkosti 256 bitov). Tento algoritmus bol následne implementovaný na BasicCard čipovú kartu - verzia 7.6, revízia D.

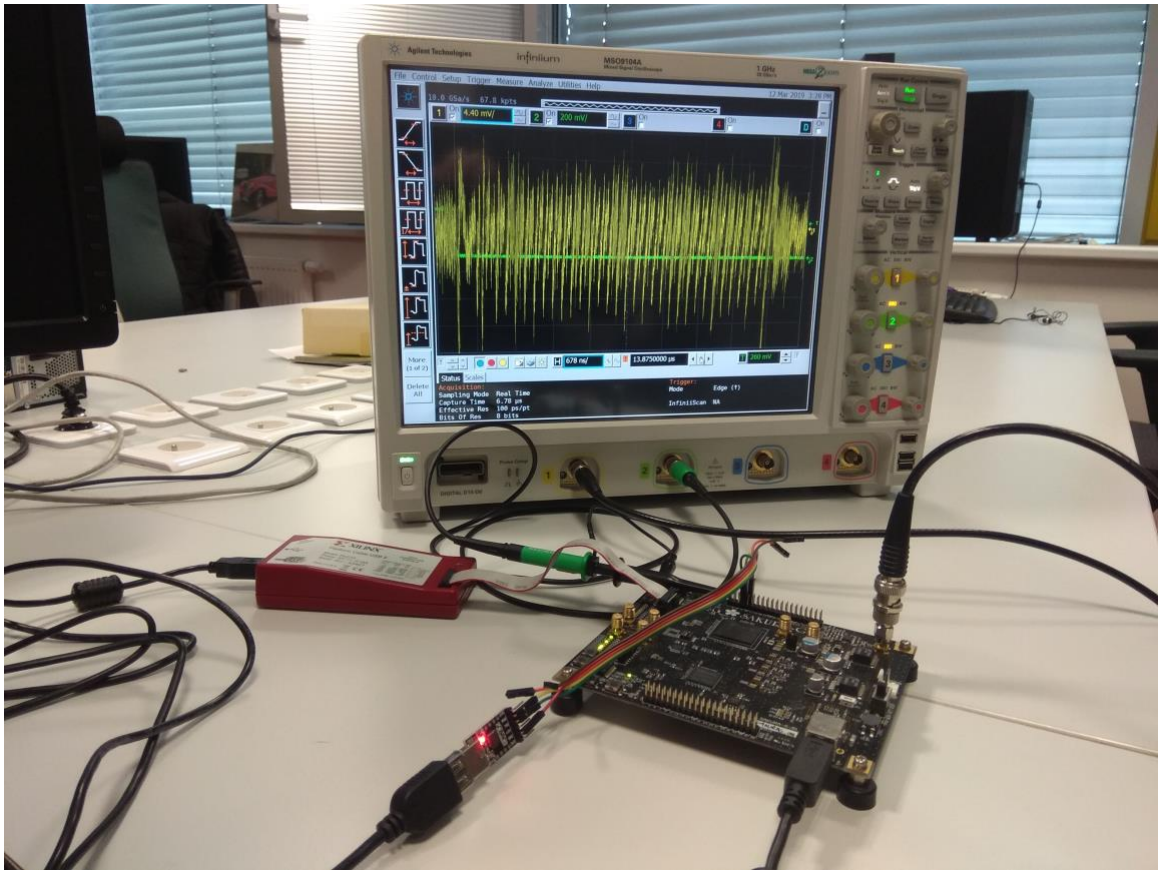
Hlavným problémom pri spúšťaní tohto programu boli neúplné cesty k potrebným súborom a knižniciam. Po neúspešnom zadávaní týchto ciest manuálne bolo použité virtuálne prostredie využívajúce Windows 7 s nainštalovaným developerským prostredím pre Basic karty, ktoré už malo tieto cesty prednastavené.

Prvé pokusy o implementáciu prebiehali za použitia čipovej karty multiapplication Basic Card verzie 8.6 rev. D. Keďže karta nepodporovala generovanie kľúčov, ktoré program používal. Problém bol v tom, že neobsahovala potrebnú knižnicu a preto bolo potrebné program upraviť. Kľúče boli miesto generovania napevno priradené. Aj napriek tomu, že program už takto pridané kľúče podporoval, nastali touto zmenou chyby aj v ďalších častiach kódu a program nebolo možné zapísať na čipovú kartu. Neustále sa vyskytoval problém pri komunikácii s kartou, program nebol schopný nadviazať s ňou spojenie a preto nebolo možné algoritmus implementovať. Z tohto dôvodu bola teda karta zamenená za iný typ, konkrétne išlo o čipovú karu Basic Card verzia 7.6, revízia D. Táto verzia karty už podporovala všetky potrebné funkcie a knižnice a bolo teda možné použiť aj pôvodný program bez úprav. Implementácia na túto kartu prebehla úspešne.

3.2 Experimentálne pracovisko s vývojovými doskami SAKURA G a W

Doska SAKURA-G FPGA je určená na výskum a vývoj v oblasti hardvérovej bezpečnosti, vhodná na testovanie útokov postrannými kanálmi, či útokov vkladaním chýb. Na doske sú integrované dve Spartan™ -6 FPGA (polia logických členov programovateľné užívateľom), ktoré slúžia ako ovládač a hlavné bezpečnostné obvody. Extrémne nízka hladina hluku a integrovaný zosilňovač uľahčujú analýzu výkonu. Konfigurácia a ovládanie dosky prebieha pomocou špecializovaného softwaru [27].

SAKURA-W je kontaktná čítačka čipových kariet navrhnutá ako nadstavba k doske SAKURA-G. Dodáva sa s kartou Atmel ATMega-8515, kde sú už aplikované protokoly AES a DES [28].



Obr. 8 Doska SAKURA-G pripojená k osciloskopu

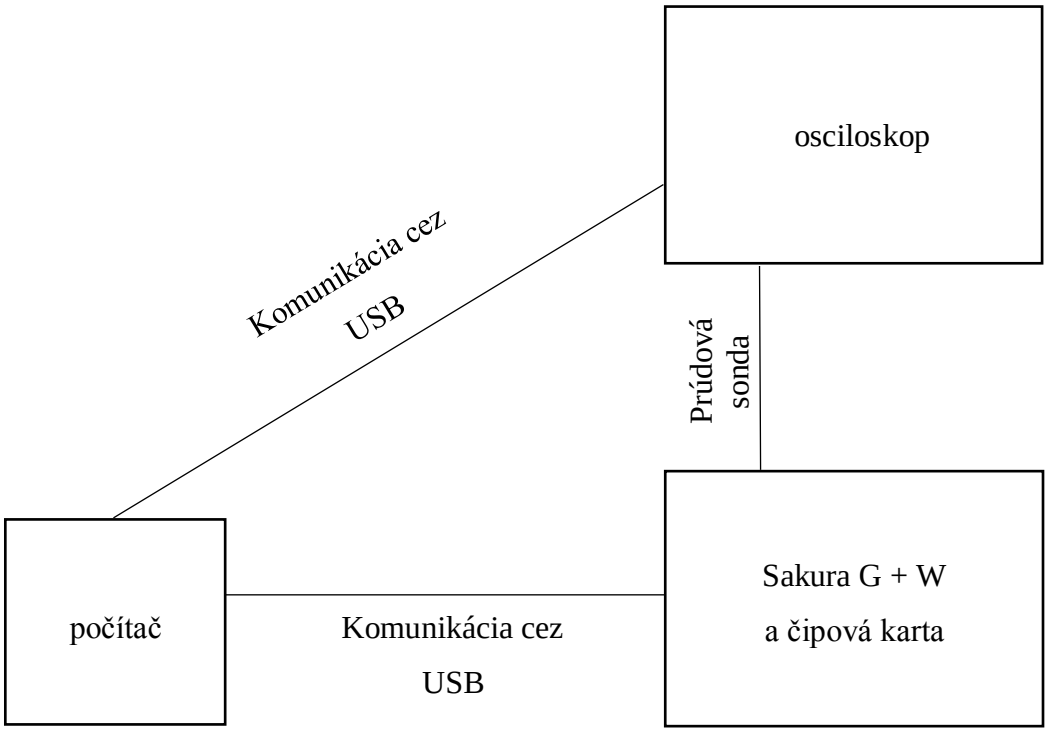
Po úspešnom nahraní protokolu ECDSA na čipovú kartu bolo potrebné upraviť už existujúci program určený na komunikáciu s vývojovými doskami. Pri prvom kroku bolo nutné zistiť aké parametre je treba odoslať čipovej karte aby volala potrebné príkazy a aby teda prebehlo podpisovanie. Tento krok je potrebný z toho dôvodu, že karta sama o sebe síce program implementovaný má, ale nevie ho bez vonkajšieho impulzu spustiť. Protokol elektronického podpisu založený na eliptických krivkách mal pri každej funkcii zadané určité parametre. Tie bolo neskôr treba správne zadať do ovládacieho programu tak, aby

karta začala podpisovanie a aby bolo možné zavolať ďalšie funkcie potrebné na úspešný priebeh algoritmu.

Pri vhodnom zadaní parametrov CLA, INS, L_e a L_c (parametre hlavičiek príkazov definovaných v protokolovom programe na čipovej karte) je možné volať dané príkazy na karte aj bez priameho prístupu k jej vývojovému prostrediu.

Prvým volaným príkazom bolo generovanie súkromného kľúča a následné dopočítanie verejného. Táto funkcia bola zavolaná len raz, dvojica kľúčov sa uložila do EEPROM pamäte a počas merania bola táto informácia nezmenená. Pred tým, než program zavolal samotnú funkciu podpisovania, bolo potrebné pre každé merania vygenerovať správu a následne dopočítať jej hash. Z toho dôvodu bolo nutné v komunikujúcom programe dopísať funkciu, ktorá by presne toto vykonávala. Na kartu sa teda neposielali dlhé správy ale už len ich zhashované verzie. Z celého programu implementovaného na čipovej karte sa teda využívala nakoniec len funkcia, ktorá správu podpísala, dúfajúc, že tento fakt by mohol nejakým spôsobom uľahčiť následnú analýzu.

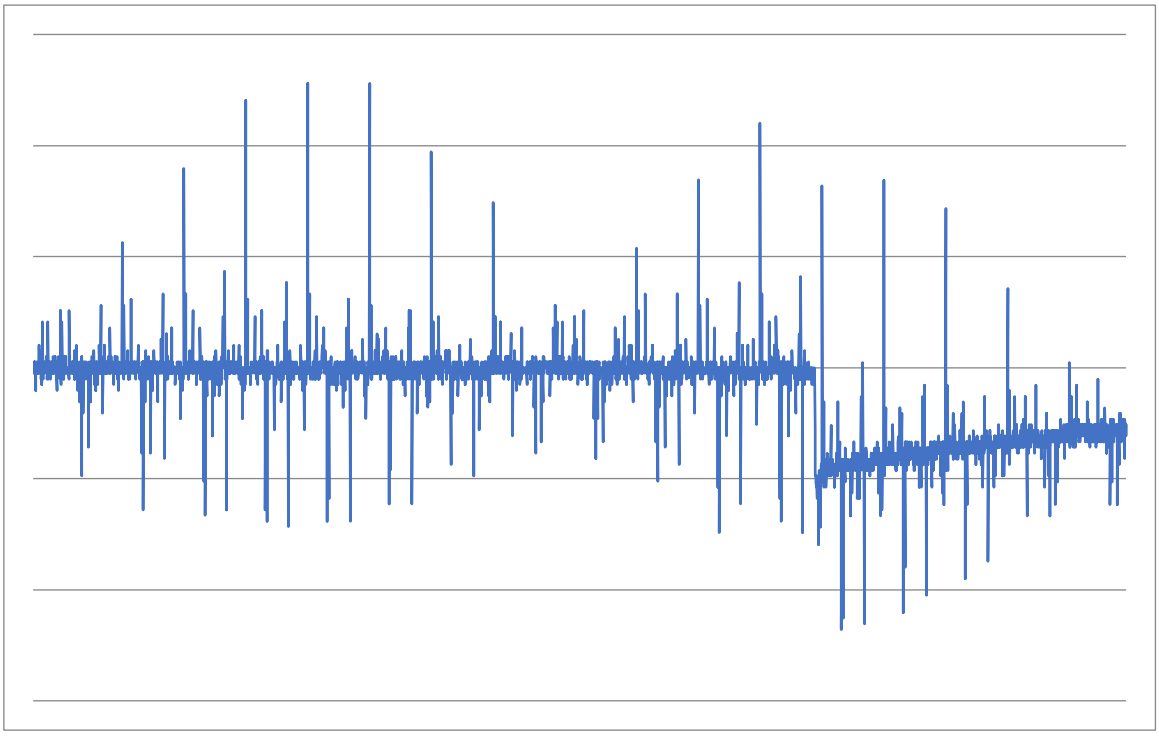
Všetky komponenty bolo nutné správne pozapájať (tak ako ukazuje schéma na obrázku 7). Pri bežnom meraní by medzi osciloskopom a čipovou kartou vloženou do vývojovej dosky bol privedený synchronizačný signál, ktorý je určený na vyrovnanie výkyvov pri meraní a uľahčenie následnej analýzy. Čipová karta BasicCard verzia 7.6, rev. D však takúto synchronizáciu nepodporuje.



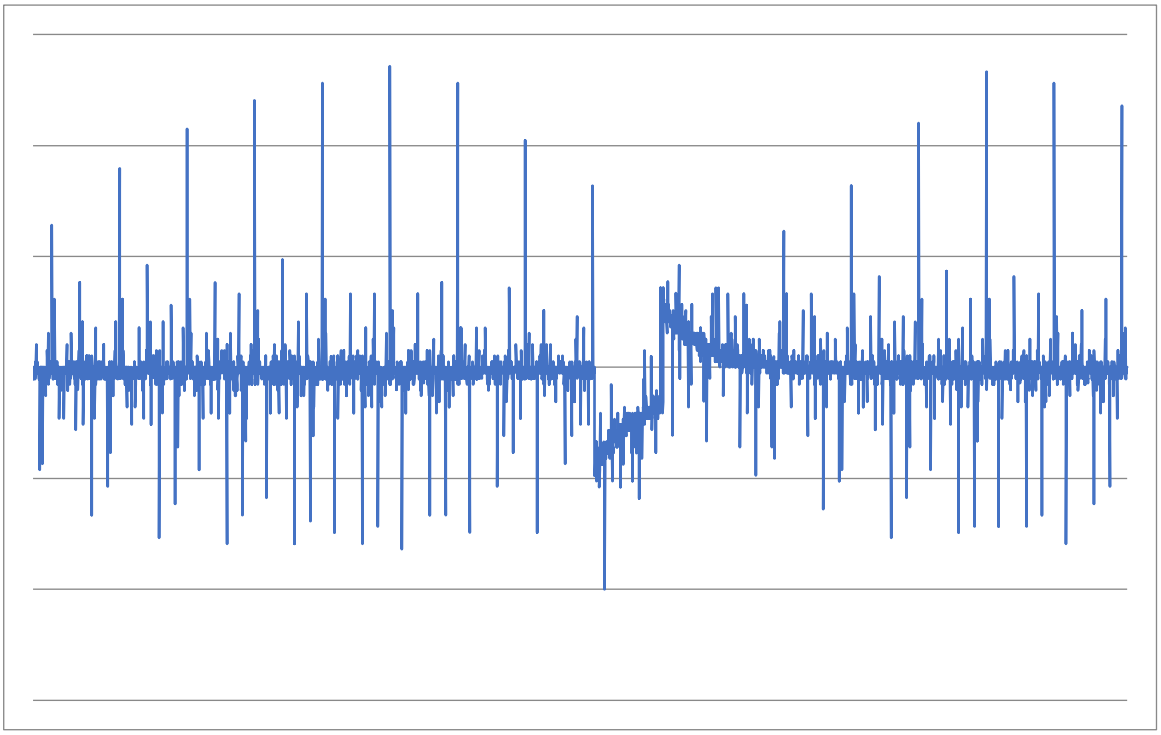
Obr. 9 Schéma zapojenia komponentov pri meraní

3.3 Namerané priebehy a analýza

Pri meraní bolo zaznamenaných 1000 priebehov. Ako bolo vyššie spomínané, tieto priebehy neboli synchronizované, keďže to daná čipová karta neumožňuje.



Obr. 10 Namerané priebehy (1)



Obr. 11 Namerané priebehy (2)

Namiesto synchronizačného signálu však bolo možné priviesť na pin karty iné signály, ktoré nám mohli prezradiť aspoň minimálne informácie o nameraných priebehoch. Prvým signálom, ktorý bol privedený bol časový signál, ktorý však mimo rýchlosti tisícich priebehov žiadne iné informácie prezradiť nedokázal.

Ďalším privedeným signálom bol signál ukazujúci príjem a odosielanie dát. Jeho priebeh na obrázkoch zobrazený nie je, pri meraní však pomocou neho bolo možné odvodiť kedy približne podpisovanie prebiehalo.

Predpokladáme, že fáza podpisovania nastane hneď po prijatí dát (zhashovanej správy) posielaných riadiacim programom. Dáta boli prijímané v čase, keď je hladina prúdového odberu konštantná. Ak podpisovanie naozaj prebiehalo hneď po prijatí týchto dát, zmena v prúdovej spotrebe v nameranom priebehu by mohla znázorňovať túto fázu. Následný nárast odberu na druhom priebehu môže znázorňovať odoslanie vypočítaného podpisu, po ktorom sa opäť odber vrátil do konštantného stavu pri ktorom karta pravdepodobne čaká na prijatie novej správy na podpísanie.

Všetky tieto úsudky sú však čisto len hypotetické. S istotou by to bolo možné povedať až po vykonaní prúdovej analýzy programom na to určeným. Keďže však tieto priebehy synchronizované neboli, nie je možné spraviť prúdovú analýzu a tým pádom ani získať súkromný kľúč.

V prípade, že by sa využil iný typ karty, ktorý synchronizáciu podporuje, prúdovú analýzu by bolo jednoduché spraviť pomocou už existujúceho programu napísaného v Matlabe. Opäť by však pravdepodobne nastali problémy pri implementácii protokolu na čipovú karu, keďže nie všetky karty podporujúce synchronizáciu podporujú aj knižnice eliptických kriviek (napríklad čipová karta, ktorá bola pôvodnou súčasťou experimentálneho vývojového prostredia s doskami Sakura G a W synchronizáciu síce umožňuje ale nepodporuje žiadne knižnice eliptických kriviek).

Ak by sa však pri každom podpisovaní generovali nové kľúče, tak ako bol pôvodný program napísaný, prúdová analýza by pravdepodobne nebola úspešná aj v prípade, že by implementácia na inú kartu a následná synchronizácia pri meraní bola možná. Na úspešnú analýzu a odhalenie tajného kľúča je totiž potrebných viacero zosynchronizovaných priebehov pracujúcich s rovnakým kľúčom.

Cieľom praktickej časti tejto bakalárskej práce bolo implementovať vybraný kryptografický protokol založený na eliptických krivkách na čipovú kartu, zmeranie prúdových odberov počas chodu aplikácie a následné vykonanie prúdovej analýzy, ktorej výsledkom malo byť zobrazenie priebehov a ideálne zaistený tajný šifrovací kľúč. Implementácia na čipovú kartu po menších problémoch prebehla úspešne, meranie prúdovej spotreby počas podpisovania správ čipovou kartou bolo tiež úspešné, výsledkom bolo veľké množstvo prúdových priebehov, avšak šifrovací kľúč sa z dôvodu desynchornizácie získať nepodarilo.

ZÁVER

Bakalárska práca rozoberá analýzu postrannými kanálmi a modernú kryptografiu eliptických kriviek. Zameranie tejto práce je kladené na pochopenie problematiky, vzhľadom na jej obšírnosť a fakt, že obe témy sú pomerne nové v oblasti kryptografie a k určitým častiam boli zdroje veľmi obmedzené. Sú tu popísané rôzne možnosti útokov pomocou bočných kanálov, s detailnejším zameraním na prúdový postranný kanál a spôsoby útokov pomocou tohto kanálu. Taktiež sú vysvetlené základné informácie o eliptických krivkách, všeobecné algoritmy a princípy práce s eliptickými krivkami. Popísané sú aj vybrané protokoly založené na probléme diskrétneho algoritmu v eliptických krivkách.

Kryptografia eliptických kriviek je momentálne dostatočne bezpečná a rýchla a je ideálnym základom na ktorom by mohli byť postavené ďalšie nové a bezpečnejšie protokoly do budúcnosti. Supersingulárne eliptické krivky a ich protokoly umožňujú teoretickú ochranu voči kvantovým počítačom a mnohí vidia budúcnosť práve v post-kvantových protokoloch. Ich reálne praktické využitie v plnom rozsahu však zatiaľ nie je najideálnejšie, keďže sú pamäťovo aj výpočtovo náročnejšie ako protokoly klasickej kryptografie.

V rámci praktickej časti bakalárskej práce bolo cieľom implementovať vybraný kryptografický protokol založený na eliptických krivkách na čipovú kartu, následné nameranie prúdových odberov počas chodu aplikácie a vykonanie prúdovej analýzy na vybraný kryptografický systém, ktorej výsledkom bude zobrazenie priebehov a ideálne úspešne zaistený tajný šifrovací kľúč.

Na čipovú kartu bol implementovaný protokol digitálneho podpisu, ktorý je založený na eliptických krivkách. Po zapojení komponentov na meranie a správnej úprave programu riadiaceho vývojové prostredie s doskami SAKURA-G a SAKURA-W bolo namerané množstvo prúdových priebehov. Tieto priebehy však boli desynchronizované, keďže čipová karta synchronizáciu neumožňovala. Aj napriek dostatočnému počtu meraní však nebolo možné diferenciálnou prúdovou analýzou tajný kľúč odhaliť.

Ak by sa pri každom podpisovaní počas merania používal rovnaký kľúčový pár a synchronizácia bola možná (či už pri meraní alebo by sa nejakým spôsobom po odmeraní dali priebehy manuálne synchronizovať), tajný kľúč by bolo pravdepodobne možné získať pomerne jednoducho.

Aj napriek tomu, že v tomto prípade útok postranným kanálom nebol úspešný, kryptoanalýza postrannými kanálmi sa dnes javí ako jeden z najjednoduchších a najúčinnejších útokov na kryptosystémy, keďže žiadny nie je dostatočne dokonalý natoľko, aby z neho neunikali absolútne žiadne informácie. Útočník je schopný využiť informácie ako napríklad zvuk či prúdový odber a získať informácie z ktorých je teoreticky možné získať šifrovacie tajomstvo.

LITERATÚRA

[1] FOUCHÉ GAINES, Helen. *Cryptanalysis*. Dover Publications, 1989. ISBN 0486200973.

[2] MUIR, James Alexander. *Techniques of Side Channel Cryptanalysis* [online]. Waterloo,Ontario,Canada, 2001 [cit. 2018-10-21]. Dostupné z: https://www.collectionscanada.gc.ca/obj/s4/f2/dsk3/OWTU/TC-OWTU-53.pdf Masters thesis. University of Waterloo

[3] BUDÍK, Lukáš. *POSTRANNÍ KANÁLY V KRYPTOGRAFII* [online]. Brno, 2009 [cit. 2018-10-22]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=15130 Bakalárska práca. VUT Brno, FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ, ÚSTAV TELEKOMUNIKACÍ. Vedúci práce Ing. Zdeněk Martinásek.

[4] KOLAŘÍK, Jan. *POSTRANNÍ KANÁLY* [online]. Brno, 2012 [cit. 2018-10-22]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=51870 Diplomová práca. VUT Brno, FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ, ÚSTAV TELEKOMUNIKACÍ. Vedúci práce Ing. Zdeněk Martinásek.

[5] ZHOU, JongBin a DengGuo FENG. *Side-Channel Attacks: Ten Years After Its Publication and the Impacts on Cryptographic Module Security Testing* [online]. Chinese Academy of Sciences, Beijing [cit. 2018-10-30]. Dostupné z: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.100.8856&rep=rep1&type=pdf State Key Laboratory of Information Security, Institute of Software.

[6] YANG, Sarah. *Researchers recover typed text using audio recording of keystrokes* [online]. 2005 [cit. 2018-11-09]. Dostupné z: https://www.berkeley.edu/news/media/releases/2005/09/14_key.shtml

[7] LAPS, Mark a Roy GRACE. *Capacitors for Reduced Micro phonics and Sound Emission* [online]. 2007, , 8 [cit. 2018-11-09]. Dostupné z: http://www.kemet.com/Lists/TechnicalArticles/Attachments/62/2007%20CARTS%20-%20Reduced%20Microphonics%20and%20Sound%20Emissions.pdf

[8] KOCHER, Paul, Joshua JAFFE a Benjamin JUN. *Introduction to Diferential Power Analysis*[online]. San Francisco, 1998 [cit. 2018-12-9]. Dostupné z: https://42xtjqm0qj0382ac91ye9exr-wpengine.netdna-ssl.com/wp-content/uploads/2015/08/DPATechInfo.pdf Cryptography Research.

[9] KOCHER, Paul, Joshua JAFFE a Benjamin JUN. *Diferential Power Analysis* [online]. San Francisco [cit. 2018-12-9]. Dostupné z: https://www.paulkocher.com/doc/DifferentialPowerAnalysis.pdf

[10] U.S. NATIONAL SECURITY AGENCY. *Commercial National Security Algorithm Suite and Quantum Computing FAQ* [online]. Január 2016, s. 11 [cit. 2019-03-21]. Dostupné z: https://cryptome.org/2016/01/CNSA-Suite-and-Quantum-Computing-FAQ.pdf

[11] GIRY, Damien. *Cryptography Key Length Recommendation* [online]. 2018 [cit. 2019-03-21]. Dostupné z: https://www.keylength.com/en/3/

[12] HANKERSON, D. MENEZES, A. J. VANSTONE, S.: Guide to Elliptic Curve Cryptography. Springer, 2004. 311 s. ISBN 978-0387952734.

[13] KOBLITZ, Neal *Elliptic Curve Cryptosystems* [online]. American Mathematical Society, január 1987, [cit. 2019-03-26]. Dostupné z: http://pages.cs.wisc.edu/~cs812-1/koblitz87.pdf

[14] BROWN, Daniel R.L. *Elliptic Curve Cryptography: Certicom Research* [online]. 2009 [cit. 2019-03-26]. Dostupné z: http://www.secg.org/sec1-v2.pdf 2009 Certicom Corp

[15] MARTINÁSEK, Zdeněk. *Kryptoanalýza postranními kanály* [online]. Brno, 2013 [cit. 2019-03-28]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=57268 Dizertační práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Vedoucí práce Ing. Václav Zeman, Ph.D.

[16] DIFFIE, Whitfield a Martin E. HELLMAN. *New Directions In Cryptography* [online]. 1976 [cit. 2019-03-29]. Dostupné z: https://ee.stanford.edu/~hellman/publications/24.pdf

[17] ODLYZKO, A.M. *Discrete logarithms in finite fields and their cryptographic significance* [online]. Murray Hill, New Jersey 07974 [cit. 2019-03-29]. Dostupné z: http://www.dtc.umn.edu/~odlyzko/doc/arch/discrete.logs.pdf AT&T Bell Laboratories

[18] JOHNSON, Don a Alfred MENEZES. *The Elliptic Curve Digital Signature Algorithm ECDSA* [online]. Waterloo, 1999 [cit. 2019-03-29]. Dostupné z: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.472.9475&rep=rep1&type=pdf Certicom Resear. University of Waterloo.

[19] EDWARDS, Harold M. *A NORMAL FORM FOR ELLIPTIC CURVE*[online]. 2007 [cit. 2019-03-30]. Dostupné z: http://www.ams.org/journals/bull/2007-44-03/S0273-0979-07-01153-6/S0273-0979-07-01153-6.pdf BULLETIN (New Series) OF THE AMERICAN MATHEMATICAL SOCIETY.

[20] JOSEFSSON, S. a I. LIUSVAARA. *Edwards-Curve Digital Signature Algorithm (EdDSA)* [online]. 2017 [cit. 2019-03-30]. Dostupné z: https://tools.ietf.org/html/rfc8032 Internet Research Task Force (IRTF)

[21] SEURIN, Yannick. *On the Exact Security of Schnorr-Type Signatures in the Random Oracle Model* [online]. Paris, France [cit. 2019-03-30]. Dostupné z: https://eprint.iacr.org/2012/029.pdf

[22] HUYNH, Anh. *An Introduction to Supersingular Elliptic Curves and Supersingular Primes* [online]. [cit. 2018-11-26]. Dostupné z: https://wstein.org/edu/2011/581g/final/anh-supersingular_elliptic_curves.pdf

[23] LONGA, Patrick. *A Note on Post-Quantum Authenticated Key Exchange from Supersingular Isogenies* [online]. Chicago, 2008 [cit. 2018-11-29]. Dostupné z: https://eprint.iacr.org/2018/267.pdf Microsoft Research, USA.

[24] KOCHER, Paul C. *Timing Attacks on Implementations of Diffe-Hellman, RSA, DSS, and Other Systems* [online]. [cit. 2019-05-12]. Dostupné z: https://www.paulkocher.com/doc/TimingAttacks.pdf

[25] CORON, Jean-Sebastien. *Resistance against Differential Power Analysis for elliptic curves cryptosystems* [online]. [cit. 2019-05-12]. Dostupné z: http://www.crypto-uni.lu/jscoron/publications.html#dpaecc

[26] BERNSTEIN, D.J. *A state-of-the-art Diffie-Hellman function* [online]. [cit. 2019-05-12]. Dostupné z: https://cr.yp.to/ecdh.html

[27] *Informácie z oficiálnej stránky Sakura Hardware Security Project (Sakura G)* [online]. [cit. 2019-05-12]. Dostupné z: http://satoh.cs.uec.ac.jp/SAKURA/hardware/SAKURA-G.html

[28] *Informácie z oficiálnej stránky Sakura Hardware Security Project (Sakura W)* [online]. [cit. 2019-05-12]. Dostupné z: http://satoh.cs.uec.ac.jp/SAKURA/hardware/SAKURA-W.html

[29] ROGAWAY, Phillip, Mihir BELLARE a Dan BONEH. *Evaluation of Security Level of Cryptography ECMQVS* [online]. 2001 [cit. 2019-05-16]. Dostupné z: https://www.ipa.go.jp/security/enc/CRYPTREC/fy15/doc/1069_ks-ecmqv.pdf

ZOZNAM SYMBOLOV A SKRATIEK

Skratky:

AES	Advanced Encryption Standard
DEMA	Differential Electromagnetic Analysis –diferenciálna EM analýza
DES	Data Encryption Standard
DH	Diffie-Helman protokol
DLIES	Discrete Logarithm Integrated Encryption Scheme – integrovaná šifrovacia schéma diskrétneho logaritmu
DLP	Discrete Logarithm Problem – problém diskrétneho logaritmu
DPA	Differential Power Analysis – diferenciálna prúdová analýza
DSA	Digital Signature Algorithm – algoritmus digitálneho podpisu
ECC	Elliptic Curve Cryptography – kryptografia eliptický kriviek
ECDH	Elliptic Curve Diffie-Hellman – protokol DH založený na eliptických krivkách
ECDSA	Elliptic Curve Digital Signature Algorithm - algoritmus digitálneho podpisu založený na eliptických krivkách
ECIES	Elliptic Curve Integrated Encryption Schceme – integrovaná schéma šifrovania založená na eliptických krivkách
ECDLP	Elliptic Curve Discrete Logarithm Problem – problém diskrétneho logaritmu pri eliptických krivkách
EdDSA	Edwards Curve Digital Signature Algorithm - algoritmus digitálneho podpisu založený na Edwardsových eliptických krivkách
EEPROM	Electrically Erasable Programmable Read-Only Memory - elektricky mazateľná pamäť
ECMQV	protokol Menezes-Qu-Vanstone založený na eliptikckých krivkách
IDEA	Intenational Data Encryption Algorithm
IES	Integrated Encryption Scheme
LSB	Least significant bit – najmenej významný bit
KDF	Key Derivation Function – funkcie na odvodenie kľúčov
MAC	Message Authentication Code
RC5	bloková šifra so symetrickým kľúčom
RSA	Rivest-Shamir-Adleman (kryptosystém)
SEMA	Simple Electromagnetic Analysis – jednoduchá EM analýza
SHA	Secure Hash Algorithm
SIDH	Supersingular isogeny Diffie-Hellman key exchange – protokol výmeny kľúčov pomocou supersingulárnych eliptických kriviek
SIKE	Supersingular isogeny Key Encapsulation – protokol na zapúzdrenie kľúčov pomocou supersingulárnych eliptických kriviek
SPA	Simple Power Analysis – jednoduchá prúdová analýza

ZOZNAM PRÍLOH

Príloha 1 – Namerané priebehy

- vybrané namerané priebehy sú priložené na CD