VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

POSTRANNÍ KANÁLY – VYTVOŘENÍ LABORATORNÍ ÚLOHY

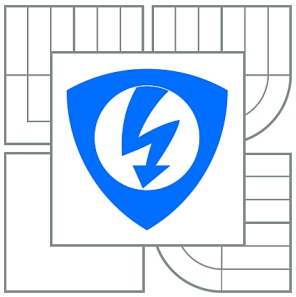
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE JAN HOLEMÁŘ
AUTHOR

BRNO 2013

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF TELECOMMUNICATIONS

POSTRANNÍ KANÁLY – VYTVOŘENÍ LABORATORNÍ ÚLOHY

SIDE CHANNELS - PREPARATION OF LAB TASK

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

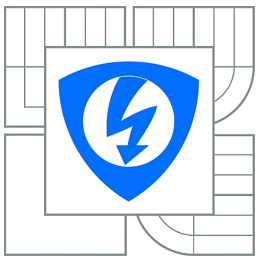
AUTOR PRÁCE — JAN HOLEMÁŘ

AUTHOR

VEDOUCÍ PRÁCE — Ing. ZDENĚK MARTINÁSEK

SUPERVISOR

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student:	Jan Holemář	***ID:***	134494
Ročník:	3	***Akademický rok:***	2012/2013

NÁZEV TÉMATU:

Postranní kanály – vytvoření laboratorní úlohy

POKYNY PRO VYPRACOVÁNÍ:

V rámci bakalářské práce prostudujte problematiku proudového postranního kanálu. Pozorně prostudujte jednoduchou a diferenční proudovou analýzu například na šifrovacích algoritmech RSA a AES. Realizujte experimentální pracoviště určené k analýze proudovým postranním kanálem. Na pracovišti realizujte jednoduchou proudovou analýzu algoritmu RSA a diferenční analýzu algoritmu AES. Navrhněte a realizujte laboratorní úlohu seznamující studenty s útoky proudovým postranním kanálem. Do laboratorní úlohy zahrňte i možná protiopatření proti proudové analýze.

DOPORUČENÁ LITERATURA:

[1] Mangard, S.; Oswald, E.; Popp, T.: Power Analysis Attacks: Revealing the Secrets of Smart Cards (Advances in Information Security). Secaucus, NJ, USA:Springer-Verlag New York, Inc., 2007, ISBN 0387308571.

[2] Kocher, P. C.; Jaffe, J.; Jun, B.: Differential Power Analysis. In CRYPTO '99: Proceedings of the 19th Annual International Cryptology Conference onAdvances in Cryptology, London, UK: Springer-Verlag, 1999, ISBN 3-540-66347-9, s. 388–397.

Termín zadání:	11.2.2013	***Termín odevzdání:***	5.6.2013

Vedoucí práce: Ing. Zdeněk Martinásek
Konzultanti bakalářské práce:

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

ABSTRAKT

Bakalářka práce se věnuje kryptoanalýze zaměřená na postranní kanály. Je zaměřena na útok proudovým postranním kanálem na kryptografické zařízení. Jako kryptografické zařízení je použita čipová karta Gemalto .NET vykonávající šifrovací algoritmy RSA, DES a AES. Analýza proudové spotřeby čipové karty byla provedena proudovou sondou Tektronix CT-6. Data získaná měřením byla zpracována na počítači s příslušným programovým vybavením k nalezení důležité informace o použitém šifrovacím klíči.

KLÍČOVÁ SLOVA

Kryptoanalýza, proudový postranní kanál, proudová analýza, RSA, DES, AES, čipová karta Gemalto .NET

ABSTRACT

This thesis deals with side-channel cryptoanalysis. It is focused on power side-channel attack on cryptographic device. The smart card Gemalto .NET is used as the cryptographic device. This smart card performs encryption through algorithm RSA, DES, AES. The power consumption of the smart card was scanned by a Tektronix CT-6 current probe. Data obtained by measuring were processed on the computer with relevant software and provided important information about the encryption key that was used.

KEYWORDS

Cryptanalysis, power side-channel, power analysis, RSA, DES, AES, Gemalto .NET smart card

HOLEMÁŘ, Jan *Postranní kanály – vytvoření laboratorní úlohy*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2012. 44 s. Vedoucí práce byl Ing. Zdeněk Martinásek

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Postranní kanály – vytvoření laboratorní úlohy“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Zdeňku Martináskovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno ....

(podpis autora)

Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic

http://www.six.feec.vutbr.cz

PODĚKOVÁNÍ

Výzkum popsaný v této bakalářské práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno ....

(podpis autora)

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

ÚVOD

V 90.letech minulého století se došlo k závěru, že moderní kryptografie je postavena na pevných matematických základech a tím pádem je bezpečná. Kryptoanalytici proto přestali hledat chyby v matematickém algoritmu, ale zaměřili se na konkrétní implementace v kryptografických zařízeních. Zde se ukázalo, že obsahují spoustu slabin a poskytují informace, s nimiž se v matematickém modelu nepočítalo. Každý nežádoucí způsob výměny informací mezi zabezpečeným systémem a okolním prostředím je postranní kanál. V současnosti je problematika postranních kanálu v popředí zájmu moderní kryptografie, neboť představují vážné nebezpečí úniku tajných informací v kryptografickém systému.

Bakalářská práce se zabývá kryptoanalýzou útoků na kryptografické zařízení se zaměřením na proudový postranní kanál. Proudová analýza postranního kanálu byla představena v roce 1998 panem Kocherem. Průběh proudové spotřeby elektronického zařízení není konstantní v čase a průběh se mění v závislosti na zpracovávaných datech a probíhajících operacích.

Teoretická část bakalářské práce se zabývá úvodem do kryptologie, objasněním kryptografického algoritmu, kryptografického zařízení, druhů, možností a principů jednotlivých útoků na něj. Příčinou vzniku proudového postranního kanálu, útoky jednoduchou a diferenciální proudovou analýzou s typickými příklady útoků na známé šifrovací algoritmy RSA a AES.

V praktické části bakalářské práce je popsán vlastní návrh a realizace měřícího pracoviště určeného pro měření proudovým postranním kanálem. Seznámení s důležitými zařízeními měřícího pracoviště a použité měřící proudové sondy. Závěrečná část bakalářské práce je věnována dosaženým výsledkům jednoduché a diferenciální proudové analýzy šifrovacích algoritmů, zhodnocením a analýzou výsledků.

1 ÚVOD DO KRYPTOLOGIE

V této kapitole definujeme základní pojmy kryptologie, které jsou spojeny s postranními kanály. Kryptologie je věda, zahrnující kryptografii a kryptoanalýzu. Kryptografie je věda, která se zabývá vytvářením šifrovacích systémů. Kryptoanalýza je věda, zabývající se metodami získávání obsahu šifrovaných informací bez přístupu k tajnému šifrovacímu klíči. Kryptoanalýza je opak kryptografie [1].

1.1 Kryptografický algoritmus

Moderní bezpečnostní systémy používají kryptografické algoritmy k zabezpečení přenosu informací mezi uživateli. Systémy musí zajistit:

- důvěrnost dat – utajení informací před neoprávněnými uživateli,
- integritu dat – informace nesmí být během přenosu změněna,
- nepopíratelnost – schopnost prokázat totožnost uživatele, který zprávu odeslal (např. digitální popis),
- autentičnost – ověření identity dat nebo entity,
- autorizaci – pouze oprávněným subjektům je umožněný přístup k vykonávání činnosti.

Kryptografické algoritmy jsou matematické funkce, které obvykle berou dva vstupní parametry:

- zpráva,
- šifrovací klíč.

Kryptografický algoritmus pracuje s těmito parametry a na výstupu je šifrovaný text. Tento proces se nazývá šifrování. U kryptografického algoritmu ve většině případů známe všechny jeho operace a údaje, pouze šifrovací klíč je držen v tajnosti. Rozlišujeme dva typy kryptografií.

- Symetrická – subjekty které mezi sebou komunikují sdílejí společný tajný klíč. Nejznámější symetrický šifrovací algoritmus je Advanced Encryption Standart (AES).
- Asymetrická – každý subjekt má jeden pár klíčů. Pár klíčů se skládá z veřejného parametru, který se nazývá veřejný klíč a tajného parametru, který se nazývá soukromý klíč. Nejznámější asymetrický šifrovací algoritmus je Rivest Shamir Adleman (RSA).

Prolomení šifrovacího algoritmu obvykle znamená najít tajný klíč, který může být například otevřený nebo šifrovaný text. Mnoho algoritmů je navrženo tak, aby

snaha prolomit šifru rostla exponenciálně s počtem bitů klíče. V důsledku toho je délka klíče důležitým faktorem bezpečnosti kryptografického algoritmu.

1.2 Kryptografické zařízení

Kryptografické zařízení jsou elektronické přístroje, které implementují šifrovací algoritmy a jsou zde uloženy šifrovací klíče. Dále provádí šifrovací a dešifrovací operace a zajišťuje autentizaci a autorizaci. Prolomení šifrovacího zařízení znamená zjistit šifrovací klíč zařízení. Osoba, která se snaží získat klíč kryptografického zařízení neautorizovaným způsobem se nazývá útočník. Bezpečnost kryptografického zařízení by se neměla spoléhat na utajení implementovaného šifrovacího algoritmu.

1.2.1 Druhy útoků na kryptografické zařízení

Cílem útoků na kryptografické zařízení je odhalit tajný šifrovací klíč. Nicméně techniky používané k dosažení tohoto cíle jsou rozmanité. Pro úspěšný útok na kryptografické zařízení a získání tajné informace jsou potřeba finanční prostředky, čas, přístup k danému zařízení a nutná úroveň znalostí a zkušeností útočníka. Základní dělení útoků na kryptografická zařízení je rozděleno z hlediska ovlivňování chodu zařízení. Jedná se o dva druhy útoků – aktivní a pasivní.

Aktivní

Hlavním cílem útočníka je ovlivnit zařízení tak, aby se zařízení chovalo nestandardně. Útočník se snaží ovlivnit vstupy zařízení nebo prostředí, ve kterém se nachází. Například změnou napájecího napětí, hodinového signálu, atd. Tato manipulace způsobuje nestandardní chování zařízení a jejím zkoumáním je možné odhalit šifrovací klíč.

Pasivní

Útočník při tomto druhu útoku nijak neovlivňuje chod zařízení. Útočník sleduje fyzikální vlastnosti zařízení, tj. doba šifrování či dešifrování, proudová spotřeba, atd. Na základě vypozorovaných vlastností je možné odhalit šifrovací klíč.

Dalším možným dělením je zaměření útoku na rozhraní kryptografického prostředku. Kryptografické zařízení mají několik fyzických a logických rozhraní. Některé z těchto rozhraní mohou být jednoduše přístupné, zatímco jiný mohou být přístupní pouze se speciálním vybavením. Na základě rozhraní, které se používá k útoku, lze rozlišit útoky na invazivní, semi-invazivní a neinvazivní [1]. Každý z těchto útoků může být jak aktivní, tak i pasivní.

Invazivní útoky

Při invazivním útoku dochází k trvalému poškození ochranné vrstvy zařízení. Účelem je získat přímý přístup k vnitřním komponentům zařízení. Při pasivním útoku pomocí měřící sondy pouze sledujeme datové signály po sběrnici (např. signály na sběrnici procesoru). Při aktivním útoku jsou signály v zařízení ovlivněny vnějším zařízením. Mezi tato zařízení patří např. laserový nůž či iontový paprsek. Invazivní útoky jsou extrémně silné, naopak nevýhodou je vyžadující drahé vybavení.

Semi-invazivní útoky

Při semi-invazivním útoku nedochází k poškození ochranné vrstvy zařízení. Účelem je získat přístup k vnitřním částem zařízení, není však nutné vytvořit přímý elektrický kontakt. Cílem semi-invazivního útoku je přečíst obsah paměťových buněk. Aktivní útok lze provést např. rentgenovými paprsky, působením elektromagnetického pole, nebo světelnými paprsky. Cílem je vyvolat poruchy v zařízení. Semi-invazivní útok nevyžaduje tak drahé vybavení jak u invazivního. Útok je ale i tak velmi náročný a vyžaduje dostatek času.

Ne-invazivní útoky

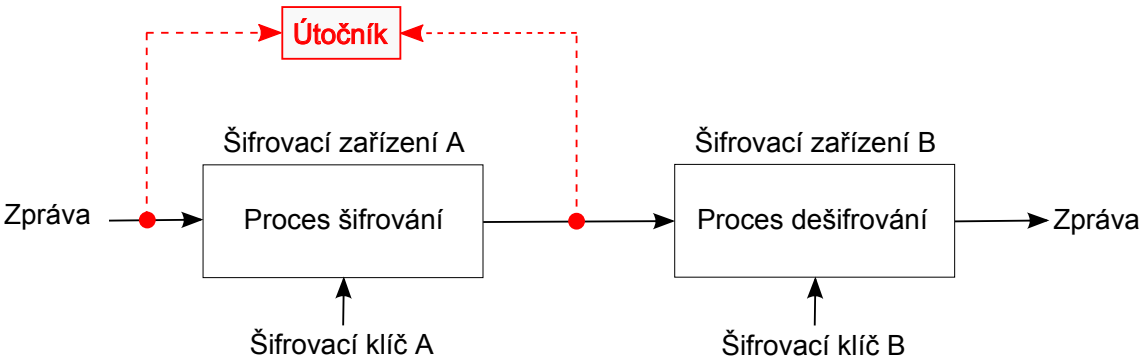
Při ne-invazivním útoku nedochází k poškození zařízení, útočník za sebou nenechá žádné stopy. Hlavní výhodou je poměrně levné vybavení. Tyto útoky představují závažné praktické ohrožení z hlediska bezpečnosti kryptografických zařízení. V současnosti největší pozornost získali pasivní ne-invazivní útoky, tzv. útoky postranními kanály. Nejčastěji používané jsou útoky postranním kanálem časovým, proudovým, elektromagnetickým, optickým a akustickým. Existují také aktivní ne-invazivní útoky. Využívají změn napájecího napětí nebo změn teploty okolního prostředí zařízení.

1.2.2 Možnosti útoku na kryptografické zařízení

V praxi se využívají k prolomení šifrovacího algoritmu uvnitř šifrovacího zařízení dva základní typy útoků. Útok konvenčním způsobem, postupem času se tento útok rozšířil o postranní kanály.

- Konvenční způsob útoku.

Konvenční model je zobrazen na obr.1.1. Útočník systematicky testuje všechny možné kombinace určitých znaků a útok se stává úspěšným, pokud nalezne správnou kombinaci znaků, která se shoduje se šifrovacím klíčem. Šifrovací klíč vstupující do šifrovacího algoritmu jako vnitřní parametr, může dosahovat velké bitové délky. Čím větší je bitová délka šifrovacího klíče, tím jsou kladeny větší nároky na čas, početní výkony a tím se zmenšuje šance úspěchu prolomení šifrovacího algoritmu. V současnosti je prolomení šifry pomocí konvenčního způsobu považováno jako metoda velmi neefektivní.



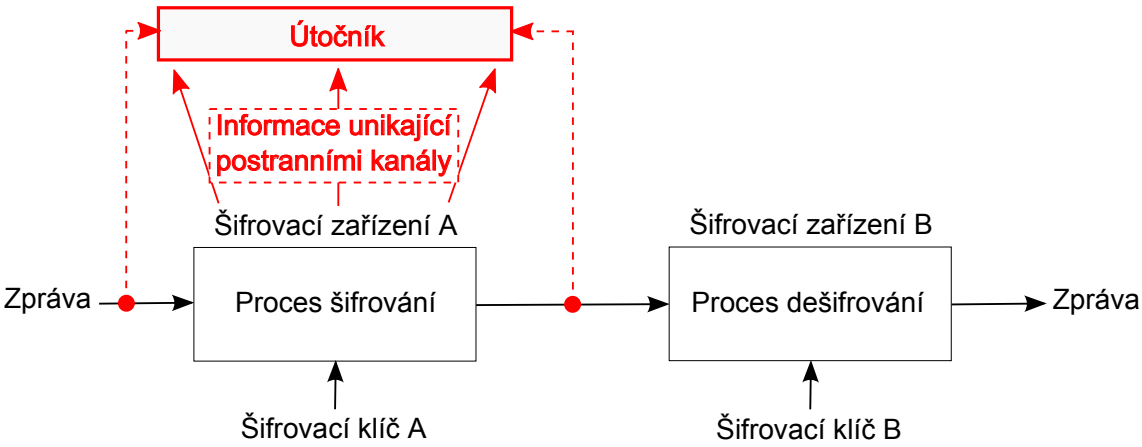
Obr. 1.1: Konvenční model útoku na kryptografické zařízení.

- Rozšířený konvenční způsob útoku.

Postupem času se útok přestal zaměřovat pouze na šifrovací algoritmus, ale obrátila se pozornost na samotné šifrovací zařízení. Každé zařízení určitým způsobem ovlivňuje své okolí. Šifrovací algoritmus je sice založen na matematické metodě, který samo o sobě nemusí být prolomen, ale musíme ho implementovat do šifrovacího zařízení, aby mohl být využit v reálném světě. Šifrovací zařízení při různých operacích šifrování/dešifrování uniká do okolního prostředí určité informace. Mezi tyto unikající informace patří například tepelné, elektromagnetické nebo jiné záření. Každé zařízení při vykonávání různých operací odebírá určitý proud z napájecího zdroje, každá jeho operace má určité časové zpoždění. Na konkrétní situace zařízení reaguje určitými stavovými a chybovými hlášením. Šifrovací/dešifrovací klíč algoritmu může být určitým způsobem závislý na některým z těchto unikajících

informací. Útočník analyzuje tyto informace a zkoumá jejich závislost s klíčem algoritmu a mohou mu ukázat nebo upřesnit podobu klíče. Tento druh útoku na šifrovací zařízení se nazývá útok tzv. postranním kanálem.

Postranní kanál označuje vysílání nežádoucích informací do okolního prostředí. Postranní kanály jsou časové, proudové, elektromagnetické, optické, akustické, popis je probrán v následující části práce. Tento rozšířený model útoku s postranními kanály je zobrazen na obr.1.2.



Obr. 1.2: Rozšířený konvenční model útoku na kryptografické zařízení.

Analýzou postranního kanálu je označován proces, při kterém je možné získat užitečné informace, které lze odvodit analýzou signálu přicházejícím po tomto kanálu.

Útok postranními kanály

Útok postranními kanály na kryptografická zařízení je označován zkratkou SCA (Side-Channel Attacks). Jsou tak označovány všechny útoky na kryptografické zařízení, které se oproti konvenčnímu modelu útoku zaměřují na informace, unikající z fyzické implementace do okolí při vykonávání různých operací algoritmu. Stručně je to proces využití postranní informace k napadení kryptografického zařízení. Tento druh útoku patří do skupiny pasivních neinvazivních útoků na kryptografické zařízení, tzn. že útočník neovlivňuje chod zařízení ani nedochází k jeho poškození. Cílem útočníka je získat potřebné informace k odhalení tajného šifrovacího klíče. Touto informací je buď zašifrovaný text nebo otevřený text, který má být zašifrován. Každý typ postranního kanálu je založen na jedné konkrétní měřitelné informaci. Obvykle mají podobu fyzikální veličiny, kterou útočník má možnost nějakým způsobem změřit. Potřebný čas a výše finančních nároků závisí na typu útoku postranního kanálu.

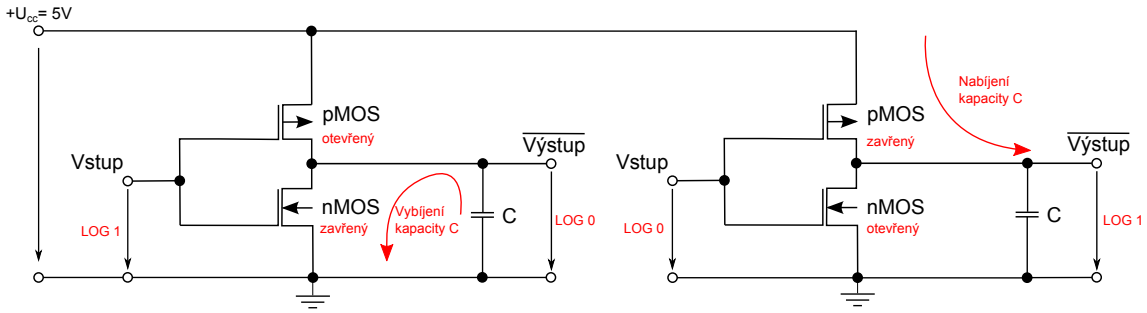
Za hlavní druhy postranních kanálů se považují ty, které lze využít s vysokou efektivitou při útoku na kryptografické zařízení. Druhy analýz postranních kanálů a jejich stručný popis.

- Elektromagnetická analýza – všechna elektronická zařízení vytvářejí během své činnosti elektromagnetická záření. Pokud je záření dostatečně silné, je možné ho zachytit a také analyzovat [2].
- Časová analýza – každá operace implementovaného algoritmu v kryptografickém zařízení trvá určitou dobu. Princip je založen na měření času nutného pro provedené určité operace. Tato informace může vést k odhalení tajného klíče [3].
- Proudová analýza – průběh proudové spotřeby kryptografického zařízení není konstantní v čase a průběh se mění v závislosti na zpracovávaných datech a probíhajících operacích. Tento typ útoku je podrobně popsán v následující kapitole 2 [1].
- Chybová analýza – útok tzv. zavedením chyby. Útočník se snaží uvést kryptografické zařízení do neobvyklého stavu pomocí vnější určité techniky. V takovémto chybovém stavu kryptografické zařízení může poskytnout útočníkovi důležité postranní informace, které mohou pomoci prolomit šifru. Mezi techniky zavedení chyb patří například změna napájecího napětí, okolní teploty, hodinového signálu [4].
- Optická analýza – vychází ze stejného principu jako proudová analýza. Při určitých operacích, kryptografické zařízení uvolňuje do okolního prostředí energii v podobě fotonů. Speciálním zařízením PICA je možné toto záření zachytit, analyzovat a informace mohou vést k odhalení tajného klíče [2].
- Akustická analýza – je úzce spjata se vstupním zařízením do kryptografického systému (např. klávesnice). Zmáčknutí každé klávesy vydává určitý zvuk, který je možný zachytit, analyzovat a může např. umožnit přístup do daného kryptografického systému [2].

2 ÚTOK PROUDOVÝM POSTRANNÍM KANÁLEM

Hlavní charakteristickou vlastností proudové analýzy je analyzování proudové spotřeby kryptografického zařízení. Průběh proudové spotřeby není konstantní v čase a průběh se mění v závislosti na zpracovávaných datech a probíhajících operacích. Integrované obvody kryptografických zařízení jsou převážně založeny na technologii CMOS (Complementary Metal Oxide Semiconductor). Základní elementární součástkou (buňkou) je invertující člen (invertor). Invertor je složen ze dvou typů tranzistorů. Typů pMOS (p-kanál) a nMOS (n-kanál), které jsou zapojeny jako spínače řízené napětím. Oba tranzistory pracují v tzv. obohaceném módu, tzn. že jsou samouzavíratelné.

Pokud je na vstupu invertoru logická 1, bude tranzistor pMOS otevřený a tranzistor nMOS zavřený. Na výstupu bude logická 0. Pokud je na vstupu logická 0, tranzistor pMOS otevřený a tranzistor nMOS zavřený. Na výstupu bude logická 1. Vždy je jeden z tranzistorů otevřený a jeden zavřený. Schématické zapojení obvodu se zobrazením vybíjení/nabíjení parazitní kapacity na obr.2.1.



Obr. 2.1: Schématické zapojení CMOS invertoru a princip jeho činnosti.

Celková proudová spotřeba CMOS obvodu je dána součtem proudových spotřeb jednotlivých buněk, ze kterých se skládá obvod. Proudovou spotřebu invertoru lze rozdělit do dvou druhů [1]. Na statickou a dynamickou proudovou spotřebu. Statická spotřeba je u invertoru, který je v ustáleném stavu. Nazývá se tzv. zbytkovým proudem. Zbytkový proud MOS tranzistoru je typicky v jednotkách pA. Dynamická spotřeba je u invertoru, kde dochází k přechodu mezi logickými stavy 0 a 1. V zásadě jde o nabíjení a vybíjení parazitní kapacity a o svodový proud. Dynamická spotřeba je obvykle dominantní faktor v celkové proudové spotřebě CMOS obvodu. Závisí na

datech, které jsou zpracovávány v CMOS obvodu. Druhy proudové spotřeby CMOS obvodu při přechodu mezi svými stavy jsou zobrazeny v tabulce 2.1.

Počáteční stav	Konečný stav	Typ proudové spotřeby
0	0	statická
0	1	statická+dynamická
1	0	statická+dynamická
1	1	statická

Tab. 2.1: Druhy proudové spotřeby CMOS obvodu a přechody mezi svými stavy [1].

Pokud je invertor v ustáleném stavu, tak hodnota proudové spotřeby je malá (statická). Naopak při přechodu mezi svými stavy je proudová spotřeba velká (dynamická). Tuto změnu stavu je možné sledovat na postranním kanále a útočník může zajistit informace o aktuálním dění uvnitř kryptografického zařízení. Tento typ útoku se nazývá jednoduchá proudová analýza SPA (Simple Power Analysis). Další typ útoku je diferenciální proudová analýza DPA (Differential Power Analysis). Oba typy útoku jsou podrobněji popsány níže v dalších kapitolách.

2.1 Jednoduchá proudová analýza SPA

Jednoduchá proudová analýza byla definována P. Kocherem v roce 1998 následujícím způsobem: jednoduchá proudová analýza je technika, která zahrnuje přímé interpretování proudové spotřeby měřené během provozu kryptografických operací. Je tedy založena na principu přímého měření proudové spotřeby kryptografického zařízení z napájecího zdroje.

Kromě křivky proudové spotřeby musí mít útočník přesné informace o tom, jaký kryptografický algoritmus je v daném zařízení implementován. Dále v napadeném zařízení musí mít klíč (přímo nebo nepřímo) významný dopad na proudovou spotřebu ze zdroje napájení. SPA útoky využívají různou spotřebu proudu mezi operacemi v kryptografickém zařízení. I když rozsahy změn v proudové spotřebě jsou malé, standardními digitálními osciloskopy lze snadno zobrazit tyto průběhy. Kmitočtové filtry a průměrování funkce (nastavení osciloskopu) jsou často používány k odfiltrování vysokofrekvenční složky. Hlavní výhodou této metody je velmi malý počet naměřených dat.

Jedním z typických příkladů je útok na implementaci šifrovacího algoritmu RSA (Rivest Shamir Adleman), který je popsán v následující části práce.

Algoritmus RSA

Algoritmus RSA je jedním z prvních asymetrických algoritmů [6]. Byl zveřejněn v roce 1977 tvůrci Rivest, Shamir, Adleman. Algoritmus je založen na principu faktorizace součinu dvou velkých prvočísel. Bezpečnost klíče závisí na jeho délce. V praxi se používá nejčastěji k šifrování dat nebo elektronickému podpisu. Nevýhoda algoritmu RSA je jeho nízká rychlost šifrování. Níže je vysvětlen princip vytvoření klíčů pro šifrování a princi šifrování/dešifrování zprávy.

Vytvoření klíčů pro šifrování

1. Vygenerujeme dvě velká prvočísla p a q tak, aby měli přibližně stejnou bitovou délku.
2. Vypočítáme jejich součin n a Eulerovu funkci $\varphi(n)$, která určuje počet přirozených čísel nesoudělných a menších než n,

$$n = pq, \tag{2.1}$$

$$\varphi(n) = (p-1)(q-1). \tag{2.2}$$

3. Zvolíme si náhodné celé číslo e, které splňuje podmínku:

$$1 < e < \varphi(n), \tag{2.3}$$

   kde $\varphi(n)$ a e jsou nesoudělná čísla.
4. Zvolíme si exponent d, musí splňovat podmínky:

$$1 < d < \varphi(n), \tag{2.4}$$

$$de = 1(mod\varphi(n)). \tag{2.5}$$

5. Veřejný klíč je dvojice čísel (n, e), kde n je modul a e je šifrovací exponent. Soukromý klíč je tvořen dvojicí (n, d), kde n je modul a d je dešifrovací exponent.

Zašifrování zprávy

1. Otevřená zpráva M se převede na číslo m v intervalu $[0, n-1]$ podle předem dané abecedy.
2. Vypočteme šifrovaný text aplikováním vzorce:

$$c = m^e \, mod \, n. \tag{2.6}$$

Dešifrování zprávy

1. Původní zprávu m získáme aplikováním vzorce:

$$m = c^d \, mod \, n. \tag{2.7}$$

Square and Multiply

Umocňování takto velkých čísel představuje velmi náročnou operaci. Proto se využívají metody, které tuto náročnou operací zjednoduší. Operace *square and multiply* je jedním z typických příkladů útoku proudové postranní analýzy. Při operaci *square and multiply* lze pozorovat rozdíl proudové spotřeby v operaci *square* a operaci *multiply* z napájecího zařízení a umožňuje útočníkovi vypočítat tajný klíč. Příklad průběhu zobrazující tento rozdíl spotřeb je zobrazen na obr.2.2.

Popis algoritmu

Vstupní hodnoty:

- exponent H (například šifrovací exponent e),
- základní prvek x (například zpráva v číselné podobě m),
- modul n.

Výstupní hodnota:

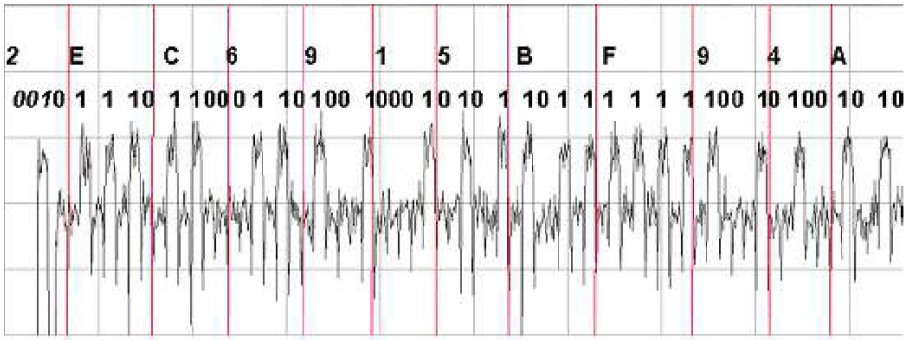
$$y = x^H \, mod \, n. \tag{2.8}$$

Postup:

1. exponent H převedeme na binární číslo $H = (h_t, h_{t-1}, \ldots, h_0)_2$,
2. dále postupujeme podle algoritmu:

   for $i = t - 1$ *to* 0
   $y = y^2 \, mod \, n$ /*square*/
   If $h_i = 1$ *then*
   $y = yx \, mod \, n$ /*multiply*/
   return y.

Postupně násobíme všechny prvky dle první rovnice (*square*) a pokud se u prvku $h_i = 1$, pak dosadíme do druhé rovnice (*multiply*).



Obr. 2.2: Rozdíl proudové spotřeby operace square (0) a operace multiply (1) [7].

2.2 Diferenciální proudová analýza DPA

Diferenciální proudová analýza je nejvíce používaným typem útoku proudové analýzy. Narozdíl od SPA útoku, DPA útoky nevyžadují podrobné znalosti o napadeném zařízení, postačí pouze jaký šifrovací algoritmus je v zařízení implementován. Dále oproti SPA, DPA analyzuje závislost proudové spotřeby v určitém konstantním časovém okamžiku na zpracovávaných datech. Můžeme odhalit klíč i z průběhů obsahující velký šum. Naopak potřebuje velké množství naměřených dat (řádově až tisíce) a tudíž je útok časově velmi náročný. Hlavním cílem je naměření velkého počtu průběhů spotřeby kryptografického zařízení při šifrování či dešifrování pro různá vstupní data. Útok je proveden podle následujících pěti kroků [1].

Krok 1: Volba vnitřní hodnoty šifrovacího algoritmu

Prvním krokem je zvolení tzv. vnitřní hodnoty šifrovacího algoritmu napadeného šifrovacího zařízení. Tato vnitřní hodnota musí mít funkcí $f(d, k)$. Kde d jsou známá vstupní hodnota dat a k je malá část použitého klíče jenž lze odhadnout (např. první bajt). Ve většině případů je d buď otevřený či zašifrovaný text.

Krok 2: Měření proudové spotřeby

Druhým krokem je měření proudové spotřeby kryptografického zařízení při šifrování nebo dešifrování různých datových bloků D. Útočník potřebuje znát odpovídající datovou hodnotu d, která se podílí na výpočtu mezivýsledku z zvoleného v kroku 1. Z hodnot známých dat vytvoříme vektor $d = (d_1, \ldots, d_D)$, kde d označuje hodnotu dat v i-tého šifrovacího nebo dešifrovacího bloku.

Během vykonávání těchto operací útočník zaznamenává proudovou spotřebu zařízení. Každému průběhu spotřeby $t_i = (t_{i,1}, \ldots, t_{i,T})$, kde T označuje dobu trvání průběhu, odpovídá jedna hodnota zpracovávaných dat d_i. Můžeme sestavit matice T o velikosti $D \times T$. Pro útok je důležité, aby naměřené průběhy byli správně zarovnané. Hodnoty proudové spotřeby v sloupci t_j matice T musí odpovídat stejné operaci. Tohoto lze dosáhnou správným nastavením synchronizace osciloskopu.

Krok 3: Sestavení matice hypotéz vnitřních hodnot

Dalším krokem útoku je vypočítat hypotetické hodnoty pro všechny klíče k. Klíče k lze zapsat jako vektor $k = (k_1, \ldots, k_K)$, kde K označuje celkový počet možných klíčů. Útočník je schopen z vektoru známých dat d a vektoru hypotéz všech klíčů snadno vypočítat hypotetické střední hodnoty $f(d, k)$ pro všechny šifrovací operace D a pro všechny hypotetické klíče K. Výsledkem je matice V o velikosti $D \times K$.

$$v_{i,j} = f(d_i, k_j) \ ; \ i = 1, \ldots, D \ ; \ j = 1, \ldots, K. \tag{2.9}$$

Sloupec j matice obsahuje vnitřní hodnoty, který byly vypočteny na základě hypotézy klíče k_j. Jeden sloupec V obsahuje ty vnitřní hodnoty, které byli vypočteny v zařízení v průběhu D operací šifrování a dešifrování. Prvek vektoru k je hodnota klíče uvnitř zařízení. Označujeme ho ck. Klíč používaný zařízením poté odpovídá prvku k_{ck}. Cílem DPA útoku je zjistit, který sloupec V byl zpracováván během D operací šifrování či dešifrování a získat tak k_{kc}.

Krok 4: Mapování vnitřních hodnot v závislosti na proudové spotřebě

Dalším krokem je mapování hypotetické vnitřní hodnoty matice V do matice hypotetických hodnot proudové spotřeby H. Útočník používá simulaci proudové spotřeby kryptografického zařízení. Pomocí této simulace přiřadí každé hypotetické vnitřní hodnotě $v_{i,j}$ hypotetickou hodnotu proudové spotřeby $h_{i,j}$.

Kvalita simulace a tím i útoku DPA je silně závislá na znalostech a zkušenostech útočníka o analyzovaném zařízení. Nejčastěji používané modely spotřeby patří model Hammingovy vzdálenosti a Hammingovy váhy.

Krok 5: Porovnání hodnot hypotetických spotřeb se změřenými průběhy

V posledním kroku DPA útoku se každá hodnota sloupce h_i hypotetické matice H porovnává se změřenými průběhy t_j matice T. Výsledkem tohoto srovnání je

matice R o velikost $K \times T$., kde každý prvek $r_{i,j}$ obsahuje porovnání mezi sloupci h_i a t_j.

Naměřené průběhy odpovídají proudové spotřebě zařízení při vykonávání šifrovacího algoritmu pro různá vstupní data. Zvolená vnitřní hodnota z kroku 1 je součástí tohoto algoritmu. Proto zařízení potřebuje vypočítat vnitřní hodnoty v_{ck} během různých operací algoritmu. Tzn., že naměřené průběhy jsou v určitých polohách na těchto vnitřních hodnotách závislé. Tuto polohu označíme jako ct a platí, že sloupec t_{ct} proudové spotřeby závisí na vnitřních hodnotách v_{ck}.

Hypotetická proudová spotřeba hodnot h_{ck} byla útočníkem nasimulována na základě vnitřních hodnot v_{ck}. Proto sloupce h_{ck} a t_{ct} jsou na sobě závislé. Tyto dva sloupce vedou k nejvyšší hodnotě R, tj. nejvyšší hodnota matice R je hodnota $r_{ck,ct}$. Všechny ostatní hodnoty R jsou nízké, protože další sloupce H a T neprokazují takovou závislost. Útočník může tedy odhalit správný klíč ck a okamžik času ct pouhým nalezením nejvyšší hodnoty v matici R. Výsledkem DPA útoku je správný klíč ck.

V praktickém provedení DPA útoku se může stát, že hodnoty R jsou přibližně stejně velké. V tomto případě útočník obvykle neměří dostatečné množství proudové spotřeby ke stanovení závislosti mezi sloupci matic H a T. Čím více hodnot bude naměřeno, tím budou sloupce matic H a T obsahovat více prvků a tím lze lépe určit vztah mezi sloupci. Blokový diagram DPA útoku znázorňující kroky 3 až 5 jsou graficky znázorněny na obr.2.3. Jedna z nejznámějších používaných statistických metod je korelační koeficient.

Korelační koeficient

Patří k nejznámější metodě k určení lineární závislosti mezi dvěma náhodnými proměnnými. Korelační koeficient je definován pomocí kovariance vztahem:

$$\rho(X,Y) = \frac{Cov(X,Y)}{\sqrt{\rho^2(X) \cdot \rho^2(Y)}}, \tag{2.10}$$

kde ρ je bezrozměrná veličina nabývajících hodnot od -1 do 1. Hodnota -1 značí nepřímou závislost (změna v jedné skupině je provázena opačnou změnou ve skupině druhé). Hodnota 0 značí, že mezi hodnotami obou skupin neexistuje žádná statisticky zjistitelná závislost. Jestliže je roven 1, značí to přímou závislost, dokonalou korelaci mezi hodnotami obou skupin.

Veličina ρ je většinou neznámá a je nutné tuto hodnotu odhadnout. Tento odhad r je definován vztahem:

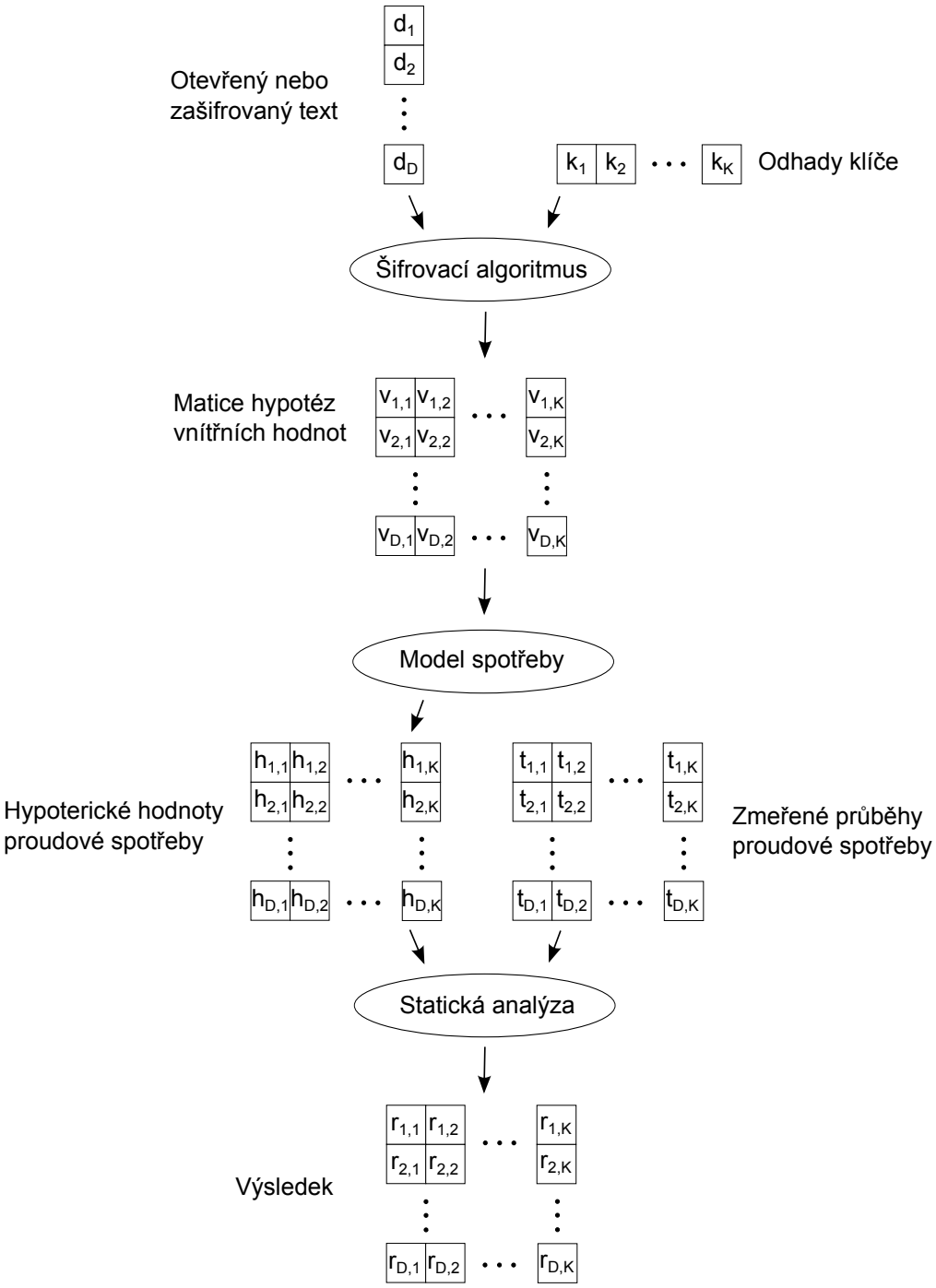
$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \cdot \sum_{i=1}^{n}(y_i - \bar{y})^2}}. \tag{2.11}$$

V DPA je korelační koeficient použit k určení lineární závislosti mezi sloupci h_i a t_j pro $i = 1, \ldots, K$ a $j = 1, \ldots, T$. Výsledkem je matice R obsahující korelační koeficienty. Označíme každou hodnotu jako $r_{i,j}$ na základě elementů D ze sloupců h_i a t_j. Použijeme-li předchozí definici korelačního koeficientu můžeme vztah 2.11 vyjádřit:

$$r_{i,j} = \frac{\sum_{d=1}^{D}(h_{d,i} - \bar{h_i}) \cdot (t_{d,j} - \bar{t_j})}{\sqrt{\sum_{d=1}^{D}(h_{d,i} - \bar{h_i})^2 \cdot \sum_{d=1}^{D}(t_{d,j} - \bar{t_j})^2}}, \tag{2.12}$$

kde $\bar{h_i}$ a $\bar{t_j}$ označují průměrné hodnoty sloupců h_i a t_j.

Jedním z typických příkladů DPA útoku je na implementaci šifrovacího algoritmu AES. Útokem diferenciální proudovou analýzou na inicializační fáze `AddRoundKey` a operace `SubBytes` umožňuje útočníkovi zjistit hodnotu šifrovacího klíče. Pro úspěšné provedení útoku je nutná znalost algoritmu vzebrubně. Šifrovací algoritmus AES je popsán v následující části práce.



Obr. 2.3: Blokový diagram znázorňující kroky 3 až 5 DPA útoku [1].

Advanced Encryption Standard AES

V roce 1997 Americký národní institut pro standardy a technologie (NIST) zahájil iniciativu vyvinout nový šifrovací standart. Dosavadní zastaralý Data Encryption Standart (DES [8]) měl nahradit nový Advanced Encryption Standart (AES). Po zdlouhavém výběru byl 2.října 2000 zvolen algoritmus Rijndael, pojmenovaný podle tvůrců Joana Daemena a Vincenta Rijmena. Celé znění standardu AES je možné nalézt v literatuře [9].

AES šifruje data pevné délky bloků (128 bitů) podle šifrovacího klíče, který může mít délku 128, 192 nebo 256 bitů. Klíče se od sebe liší počtem tzv. rund (kol). Podle délky šifrovacího klíče existují různé varianty AES shrnuty v tab.2.2.

Varianta AES	Délka vstupního boxu (bit)	Délka klíče (bit)	Počet rund
AES-128	128	128	10
AES-192	128	192	12
AES-256	128	256	14

Tab. 2.2: Počet rund pro různé varianty AES.

Varianta AES-128 zašifruje datový blok o velikosti 128 bitů datovým klíčem o velikosti 128 bitů. Tyto data jsou ve tvaru dvou matic o velikosti 4×4. Pole dat se nazývají *stav*. U AES na každou z 10 rund je aplikován *stav*. Round klíče jsou generovány pomocí klíče algoritmu. Dešifrování funguje podobně jako šifrováním. S jediným rozdílem, round klíče musí být aplikovány v opačném pořadí a musí být použita inverzní operace.

AES rundy se skládají ze čtyř různých operací, které se nazývají `AddRound- Key`, `SubBytes`, `ShiftRows` *a* `MixColumns`. Celý proces šifrování a dešifrování vstupního bloku dat znázorňuje obr.2.8. Prve proběhne devět rund obsahující všechny tyto čtyři operace a poté závěrečná desátá, která provede pouze tři operace (vynechá `MixColumns` operaci).

Operace AES

KeyExpansion

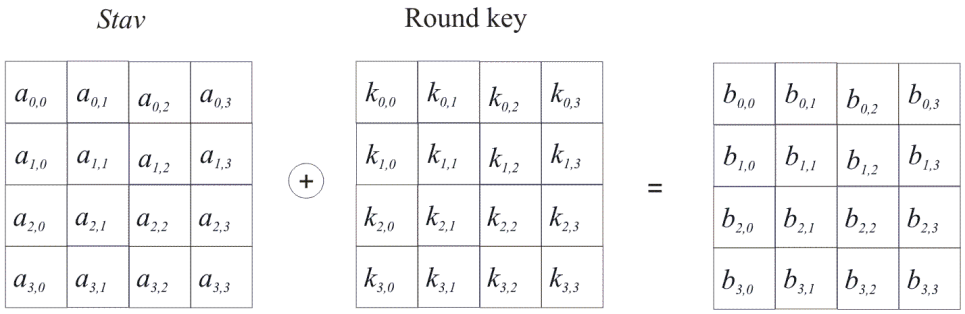
Algoritmus AES vytvoří z šifrovacího klíče jednotlivé rundovní klíče, podrobněji popsáno v literatuře [9].

AddRoudKey

V operaci `AddRoundKey` je rundovní klíč přidán do pole *Stav* prostřednictvím bitové operace exklusive-or (XOR), jejíž pravdivostní hodnoty jsou uvedeny v tab.2.3.

Vstup A	Vstup B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

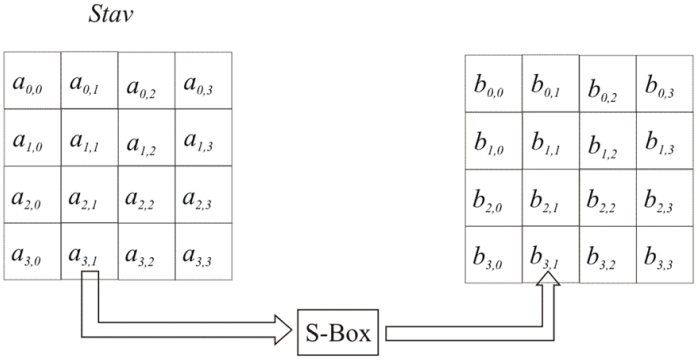
Tab. 2.3: Tabulka stavů bitové operace XOR.



Obr. 2.4: Operace `AddRoundKey`.

SubBytes

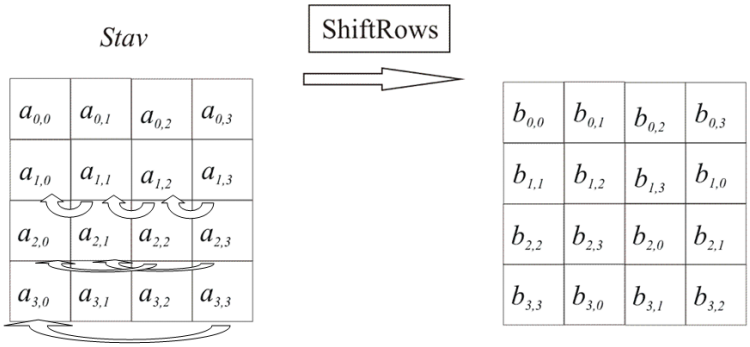
Tato nelineární operace provede substituci (nahrazení) každého bajtu vytvořeného v inicializační fázi (*stav*) hodnotou v substituční tabulce, tzv. S-boxu [10].



Obr. 2.5: Operace `SubBytes`.

ShiftRows

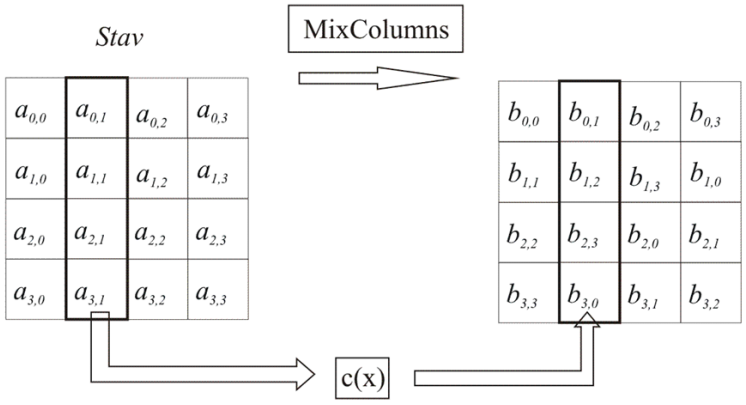
Bajty v řádcích *stav* jsou cyklicky posunuty doleva o tolik pozic, kolik udává číslo řádku matice. Počet kroků, o které je bajt v řádku matice posunut udává číslo řádku matice. První řádek se označuje jako nultý.



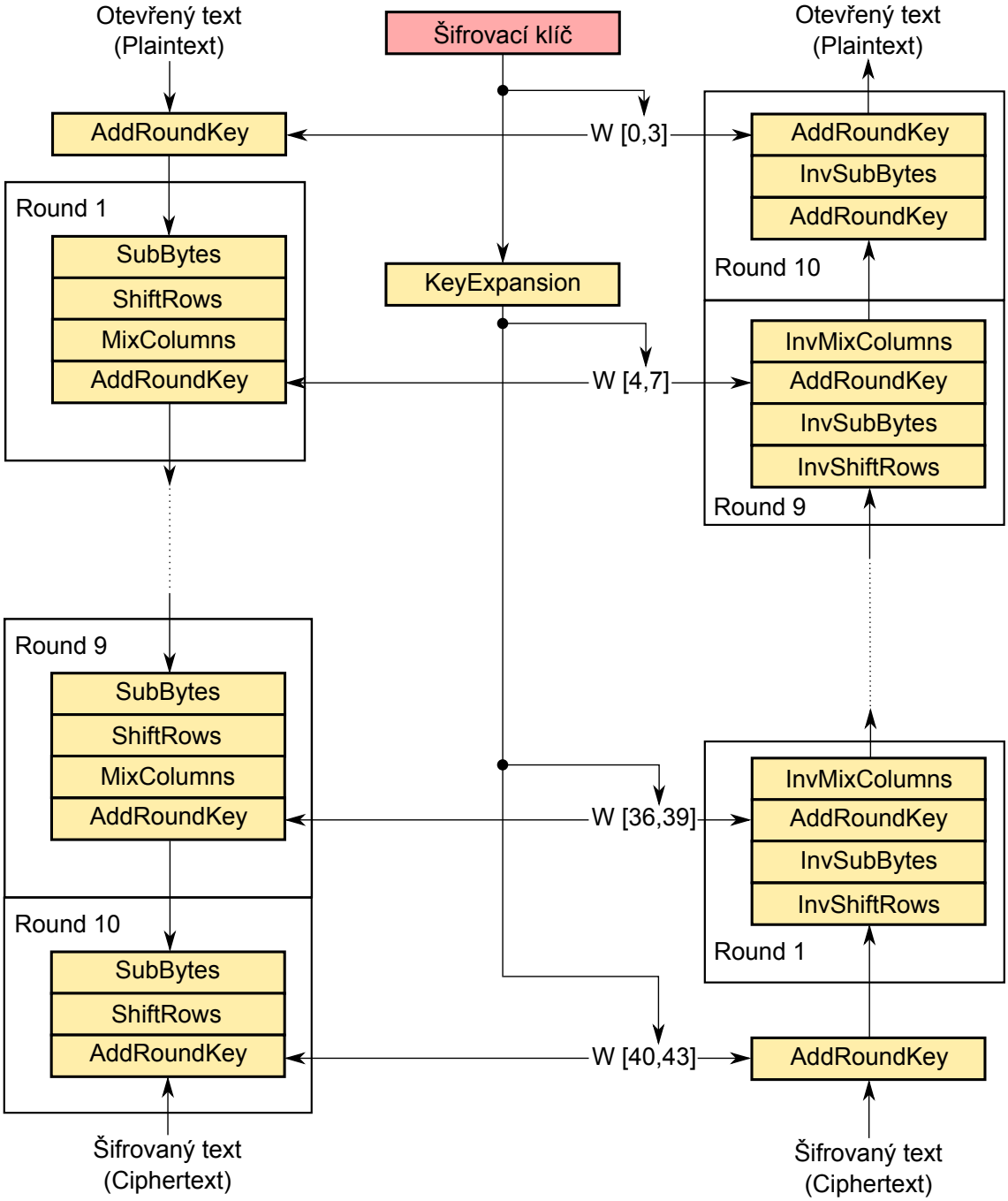
Obr. 2.6: Operace `ShiftRows`.

MixColumns

Každý sloupec *stav* je vynásoben pevně daným polynomem $c(x)$.



Obr. 2.7: Operace `MixColumns`.



Obr. 2.8: Princip šifrování a dešifrování algoritmem AES-128 [11].

2.3 Ochrana proti proudové analýze

Jak již bylo zmíněno, průběh proudové spotřeby není konstantní v čase a průběh se mění v závislosti na zpracovávaných datech a probíhajících operacích. Cílem protiopatření je tuto závislost přerušit, změnit, nebo alespoň snížit na minimum. Dle literatury [1], opatření rozdělujeme na dva typy. Na skrývání, nebo maskování závislostí tajných dat.

Skrývání (Hiding)

Principem skrývacích technik je zastavit přímé ovlivňování proudové spotřeby zpracovávaných dat a operací. Toho lze dosáhnout dle následujících dvou způsobů. Kryptografické zařízení spotřebuje v každém hodinovém cyklu buď konstantní, nebo náhodnou velikost závislosti proudové spotřeby. Dále lze skrývací techniky rozdělit do dvou skupin. Na ty, které zasahují do časové oblasti proudové spotřeby, nebo ovlivňují okamžitou velikost proudové spotřeby.

Ovlivnění časové oblasti

Tento typ ochrany je obvykle založen na principu snahy co nejvíce provádět náhodné výpočty kryptografických operací. Nejčastěji jsou použity následující dvě techniky.

- Náhodné vkládání fiktivních operací – do průběhu provádění kryptografických operací jsou náhodně vkládány fiktivní, nebo-li falešné operace. Jejich hlavním účelem je prodloužit měřený průběh o útočníkovi neznámou délku. Dále je důležité zajistit, aby během každé části algoritmu bylo použito stejné množství fiktivních operací, aby byly tyto operace od ostatních útočníkem nerozlišitelné.
- Náhodné přehazování pořadí operací – kryptografické operace, které mohou být provedeny v libovolné posloupnosti proběhnou v náhodném pořadí. Množství operací, jejichž pořadí může být zaměřeno se liší podle druhu použitého algoritmu.

Tato ochrana mnohonásobně ztěžuje útočníkovi dosáhnout úspěšného útoku diferenciální proudovou analýzou. U diferenciální proudové analýzy musí být splněna podmínka naměření průběhů proudové spotřeby určité operace pokaždé na stejné pozici. Útok by vyžadoval mnohonásobně vyšší počet měření pokud by tato podmínka nebyla splněna.

Ovlivnění okamžité velikosti

Ochrana se zabývá způsoby, jak přímo ovlivnit měření operací proudové analýzy. Jedním z možných řešení je vhodná volba programových instrukcí při softwarové implementaci kryptografických algoritmů. Různé instrukce operací mají různý průběh proudové spotřeby. Princip této ochrany spočívá ve výběru instrukcí, při jejichž vykonávání dochází k minimálnímu průběhu proudové spotřeby a útočníkovi znesnadňuje rozeznání různých kryptografických operací z proudové analýzy.

Maskování (Masking)

Principem maskovacích technik je zamaskovat (překrýt) zpracovávanou hodnotu x (obvykle klíč nebo text) náhodnou hodnotou m zvanou maska. Maska je generována kryptografickým zařízením a jeho hodnota se každým novým výpočtem mění. Proudová spotřeba operací kryptografického zařízení poté není skutečná, ale je maskovacího obrazu. Podle typu maskování je dělíme maskování booleovské a aritmetické. U booleovského maskování je hodnota x maskována maskou m pomocí logické operace XOR: $x_m = XOR(x, m)$. Aritmetické maskování založeno na aritmetické operaci modulárního sčítání: $x_m = x + m (mod\, n)$ nebo násobení: $x_m = x \times m (mod\, n)$. Modulo n je vždy voleno v závislosti na vlastnostech použitého algoritmu.

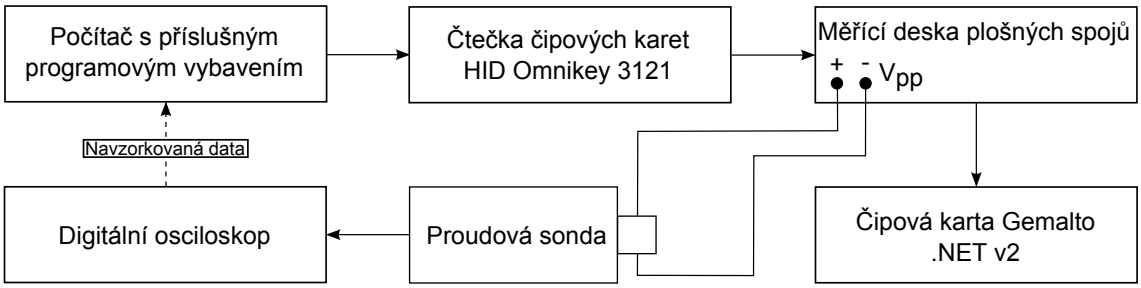
3 VLASTNÍ ŘEŠENÍ PROUDOVÉ ANALÝZY

V této kapitole se budeme zabývat praktickou částí bakalářské práce. Navrženým měřícím pracovištěm, určeného pro měření proudovým postranním kanálem. Nejprve bude popsáno použité vybavení pracoviště a následně popis a funkce důležitých částí pracoviště. Navržená jednoduchá deska plošných spojů umožňující měření průběhů proudové spotřeby, použitá čipová karta Gemalto .NET v2 představující kryptografické zařízení a měřící proudová sonda. V závěru kapitoly jsou popsány dosažené výsledky proudových analýz, zhodnocení a analýza výsledků.

3.1 Měřící pracoviště

Měřící pracoviště bylo sestaveno v laboratoři datových přenosů SC 5.34. Na obr.3.1 je zobrazeno blokové schéma zapojení měřícího pracoviště. Jednotlivé vybavení je stručně popsáno níže, důležité části měřícího pracoviště jsou popsány podrobněji v následujících částech.

- **Počítač s příslušným programovým vybavením:** počítač s operačním systémem Windows 7, nainstalovaným vývojovým prostředím Microsoft Visual Studio 2008 a ovladači Gemalto .NET SDK 2.2 potřebný pro práci s čipovými kartami.
- **Čtečka čipových karet Omnikey 3121 [15]:** slouží k programování a následné spouštění operací na uživatelem zvolené a podporované čipové kartě.
- **Jednoduchá deska plošných spojů:** pomocí DPS lze zapojit měřící sondu Tektronix CT-6 a změřit proudovou spotřebu při vykonávání různých operací.
- **Čipová karta Gemalto .NET v2 [12]:** čipová karta, na které jsou implementovány kryptografické operace a na které je prováděna proudová analýza.
- **Digitální osciloskop Tektronix DPO-4032 [16]:** dvoukanálový digitální osciloskop. Měřené průběhy ukládá na externí paměťové zařízení.
- **Proudová sonda Tektronix CT-6 [17]:** proudová sonda pro měření průběhů proudové spotřeby čipové karty.



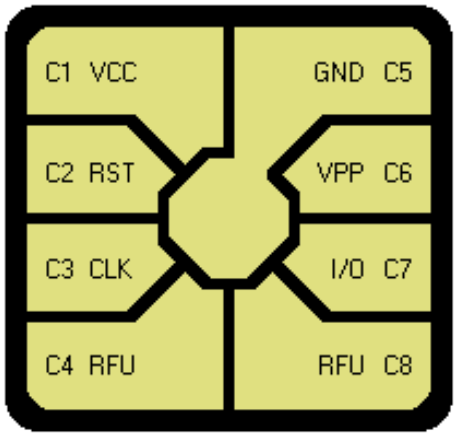
Obr. 3.1: Blokové schéma zapojení měřícího pracoviště.

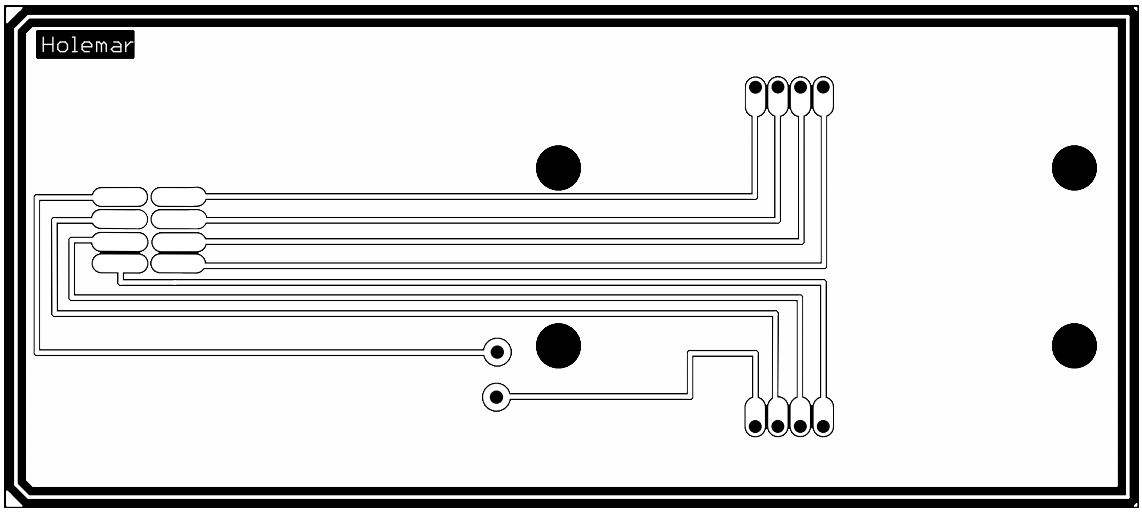
3.2 Měřící jednoduchá deska plošných spojů

K realizaci měření proudové spotřeby čipové karty při probíhajících operacích bylo nutné navrhnout jednoduchou desku plošných spojů (dále DPS) tak, aby bylo možné vytvořit drátovou propojku a zapojit měřící proudovou sondu. Kontakty čipové karty a jejich funkce je popsána na obr.3.2 dle standardu ISO 7816 [18]. Schéma navrhované měřící DPS s přerušeným obvodem pro napájení čipové karty je zobrazeno na obr.3.3. DPS je dále osazena slotem na čipové karty a dvěmi přerušenými vodiči (drátová propojka) ke které je připojena sonda a vodiče jsou spojeny kabelovou spojkou. DPS je vložena do čtečky čipových karet HID Omnikey 3121 normálním způsobem jako běžná čipová karta. Popsaný způsob zapojení je zobrazen na obr.3.4.

Kontakt	Funkce
Vcc	Napájení
RST	Reset
CLK	Hodinový signál
GRD	Zem
Vpp	Programové napájení
I/O	Sériový vstup a výstup
RFU	Není používán

Obr. 3.2: Funkce kontaktů čipové karty.





Obr. 3.3: Schéma navrhované měřící desky plošných spojů.

Obr. 3.4: Zapojení proudové sondy a měřící DPS do čtečky karet.

3.3 Čipová karta Gemalto .NET v2

V této bakalářské práci čipová karta Gemalto představuje kryptografické zařízení, na které budou implementovány kryptografické algoritmy a operace, na kterých bude prováděna proudová analýza. V této kapitole je popsána čipová karta, její princip a způsob implementace aplikací, podrobně jsou popsány všechny funkce čipové karty v literatuře [12].

Obsah čipové karty

Čipová karta má zabudovaný mikroprocesorový čip, který umožňuje zpracovávat různá data. Karta je schopná přijmout data a vrátit požadované informace. Gemalto .NET karta je kontaktní typ karty a má na svém těle umístěnou plochu s osmi kontakty, zajišťující komunikaci včetně napájení. Z hlediska softwaru obsahuje .NET Framework pro čipové karty s běhovým prostředím CLR[1], jenž je popsána následující kapitole 3.3. V tab.3.1 je seznam kryptografických algoritmů s délkami šifrovacích klíčů, které Gemalto .NET karta verze 2 podporuje.

Karta obsahuje i souborový systém na který lze nahrávat uživatelské aplikace a je popsán v kapitole 3.3. Do obsahu čipové karty lze nahlédnout pomocí rozšířeného ovladače .NET Smart Card Framework SDK. Lze ho spustit samostatně nebo jako zásuvný modul nazývaný Card Explorer v námi používaném vývojovém prostředí Microsoft Visual Studio 2008.

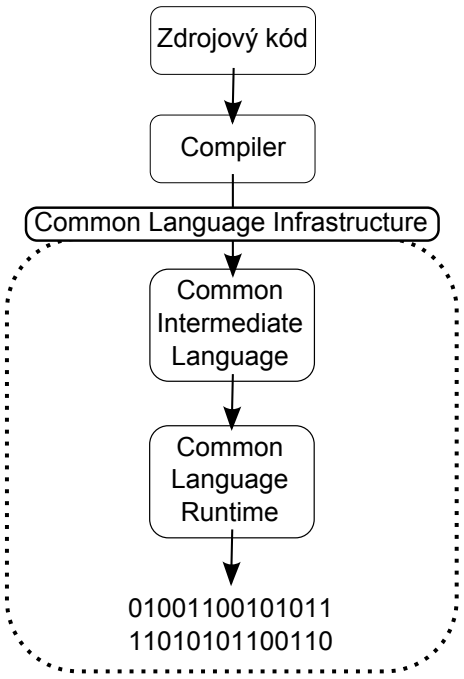
Kryptografický algoritmus	Délka šifrovacího klíče (bit)	
	Minimální	Maximální
DES	64	64
TripleDES	128	192
Rijndael	128	256
RSA	256	2048
SHA1	hash	

Tab. 3.1: Kryptografické algoritmy na čipové kartě Gemalto .NET v2 [12].

[1] Common Language Runtime

Správa a řízení aplikací

Gemalto .NET karta představuje první implementaci .NET Frameworku určeného pro čipové karty. .NET Framework pro čipové karty je ve většině vlastností velmi podobný standardnímu .NET Frameworku. Byl speciálně navržen pro práci s paměťovým modelem nacházejícím se na čipové kartě. Obsahuje speciální běhové prostředí optimalizované přímo pro řízení aplikací na čipové kartě nazývané CLR. CLR je postaveno na implementaci CLI[2] dle standardu ECMA-335 [14]. Zajišťuje správný průběh aplikací na čipové kartě v podobě bytekódu. Aplikace na .NET karty mohou být napsány v libovolném programovacím jazyce, jenž lze pomocí kompileru přeložit do CLI. V našem případě byl zvolen programovací jazyk *C#*. Jak lze vidět na obr.3.5 schématu CLR, zdrojový kód je pomocí .NET kompiléru přeložen do tzv. neutrálního jazyka nazývaného CIL[3]. CLI je dále zkompilováno CLR do tzv. mechanického kódu, který lze spustit na zvolené platformě (v našem případě .NET čipová karta).

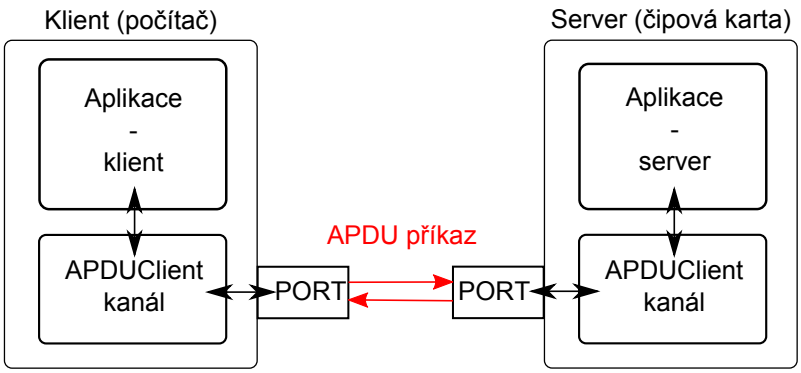


Obr. 3.5: Schéma Common Language Runtime systému.

[2]Common Language Infrastructure
[3]Common Intermediate Language

Komunikace

Komunikace (nazýváno také remoting) u klasického .NET Frameworku umožňuje jedné operaci nebo procesu komunikovat s jinou spuštěnu na stejném počítači, nebo na dvou počítačích propojených pomocí lokální sítě nebo sítě internet. .NET Framework pro čipové karty je rozšířen o APDU[4] komunikaci, kdy aplikace spuštěná na počítači může komunikovat s aplikací na čipové kartě a jsou schopni navzájem přistupovat ke svým datům. APDU je protokol aplikační vrstvy, který je popsán v standardu ISO/IEC 7816-4 [13]. Aplikace na čipové kartě představuje server a aplikace v počítači klienta. Komunikace probíhá pomoci kanálu APDUClient na straně klienta a kanálu APDUServer na straně serveru. Každý z těchto kanálů má registrovaný port, na kterém zjišťuje, zda nepřišel APDU příkaz pro určitou část aplikace (viz obr.3.6). Každá část aplikace je reprezentována unikátním identifikátorem URI[5], pomocí něhož voláme funkce dané aplikace.



Obr. 3.6: Komunikace klienta a serveru prostřednictvím APDU kanálu a portu.

Souborový systém

Souborový systém karty je rozdělen na dvě „diskové jednotky“ C a D. Disk C obsahuje následující složky.

- Gemalto – zde jsou uloženy knihovny a aplikace nahrané výrobcem.
- Pub – jedna z veřejných (public) složek, kam uživatel může nahrávat své soubory a aplikace. V našem případě budem nahrávat aplikace do této složky.
- System – úložiště knihoven .NET Frameworku pro čipové karty.

Disk D obsahuje následující složky.

- Pub – opět veřejná složka, obsahuje základní informace o dané čipové kartě.

[4] Application protocol data unit
[5] Uniform Resource Identifier

Implementace aplikace

Implementace aplikací, následné spuštění a ukončení je prováděno dle následujících kroků.

- **Nahrání aplikace na kartu.** Aplikace je vytvořena v programovacím jazyce, který podporuje .NET (v našem případě *C#*). Následně je aplikace převedena do binárního formátu (viz 3.3). Poté se aplikace nahraje na čipovou kartu. Prakticky jde o pouhé nahrání souboru(aplikace) do složky na čipové kartě.
- **Instalace aplikace.** Aplikace kterou jsme nahráli na kartu se zaregistruje jako server.
- **Spuštění aplikace.** Aplikace se spustí a probíhají interakce mezi aplikací na kartě (server) a aplikací na počítači (klient). Komunikace je vždy zahájena klientem užitím protokolu dotaz/odpověď.
- **Ukončení aplikace.** Ukončení spuštěné aplikace a odregistrování aplikace.
- **Odstranění aplikace z karty.**

3.4 Proudová sonda

Proudová sonda Tektronix CT-6 je navržena tak, aby vyhovovala potřebám vysokorychlostních obvodů a testování aplikací v nízkopaměťových vysokofrekvenčních obvodech. Mezi její hlavní vlastnosti a přednosti patří:

- velká šířka pásma – 250kHZ až 2 GHz,
- velmi nízká indukčnost (<3 nH) – nezatěžuje měřený okruh,
- charakterizovat průběhy proudu s dobou náběhu menší než 200ps,
- nízká vstupní impedance,
- citlivost 5mV/mA,
- umožňuje diferenciální měření proudu.

Proudová sonda musí být zapojena ve správném směru měřeného protékajícího proudu dle instrukcí v přiloženém manuálu [17]. Pomocí SMA-BNC konektoru se připojí k měřícímu zařízení (osciloskopu) a druhý konec sondy k měřícímu obvodu. Měřený obvod se odpojí od napájecího zdroje, přeruší se vodivá cesta a nahradí se drátovou propojkou, například krátký vodič s malým průměrem. Sondou je provlečena drátová propojka a propojena zpět s měřeným obvodem. Způsob zapojení proudové sondy při proudové spotřebě čipové karty byl popsán v kapitole 3.2.

3.5 Dosažené výsledky

Realizace jednoduché proudové analýzy SPA

Implementovaný program byl vytvořen ve vývojovém prostředí Microsoft Visual Studio 2008 se zásuvným modulem Card Explorer, který umožňuje správu souborového systému čipových karet. Ve vývojovém prostředí bylo vytvořen prostředí se dvěma projekty. Hlavní, obsahující aplikaci `MyClient` a vedlejší `MyService`. Aplikace `MyClient` je spuštěna v počítači a přestavuje klienta. Aplikace `MyService` je implementována, spuštěna a zaregistrována jako server na čipové kartě (popsáno v kap.3.3).

SPA RSA algoritmu

V klientské části se definuje délka šifrovacího klíče – `RSAsize` a data v podobě textu převedeného pomocí *C#* konvertoru do byte kódu, která se mají zašifrovat – `DataToEncrypt`. Poté zavoláme funkci šifrování `service.RSAEncrypt` na čipové kartě se vstupními parametry. Vstupní parametry jsou data k šifrování, délka šifrovacího klíče a funkce `RSA.ExportParameters`, která exportuje z čipové karty šifrovací klíče nutné k následnému dešifrování zprávy.

```
encryptedData = service.RSAEncrypt(DataToEncrypt, RSAsize,
                                   RSA.ExportParameters(false);
```

V serverové části jsou vstupní parametry poslané prostřednictvím APDU kanálu deklarovány v hlavičce volané funkce `RSAEncrypt(...)`. Prvním krokem šifrování je vytvoření instance RSA třídy nazývaného `RSACryptoServiceProvider` se zvolenou bitovou délkou klíče `RSAsize`. V dalším kroku importujeme šifrovací parametry (klíče) vygenerované v hlavičce této funkce (`RSAparam`) do vytvořené třídy RSA. Nyní lze požadovaná data zašifrovat funkcí `RSA.Encrypt`. V posledním kroku se vyexportují (`return`) zašifrovaná data zpět do klientské části.

```
public byte[] RSAEncrypt(byte[] DataToEncrypt, int RSAsize,
                                               RSAParameters RSAparam)
   RSACryptoServiceProvider RSA = new RSACryptoServiceProvider(RSAsize);
   RSA.ImportParameters(RSAparam);
   encryptedData = RSA.Encrypt(DataToEncrypt, false);
 return encryptedData;
```

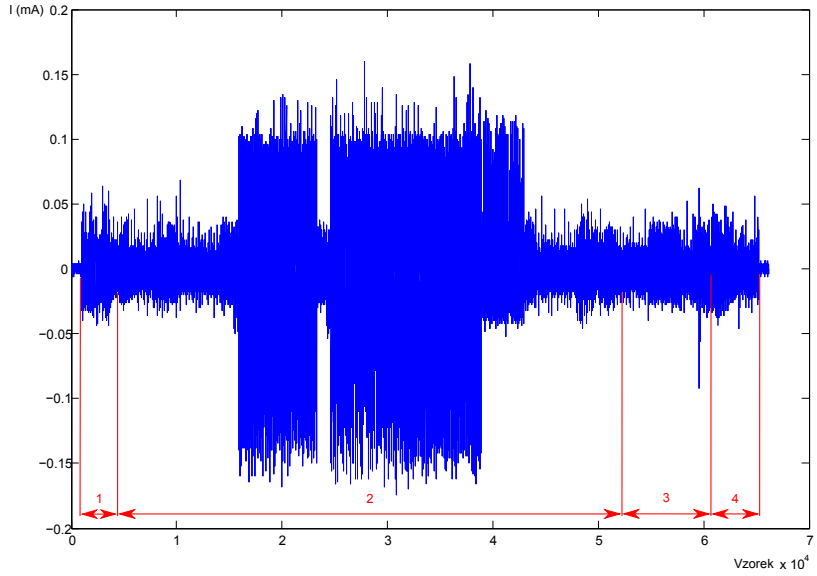
Následně je provedena kontrola šifrování. Šifrovaná zpráva je dešifrována, převedena z byte kódu nazpět do formy textu a musí se shodovat se vstupními daty určenými k šifrování. Proudová analýza RSA algoritmu s délkou klíče 1024 bitů je zobrazen na obr.3.7 a s délkou klíče 512 bitů na obr.3.8.

Popis průběhu obr.3.7:

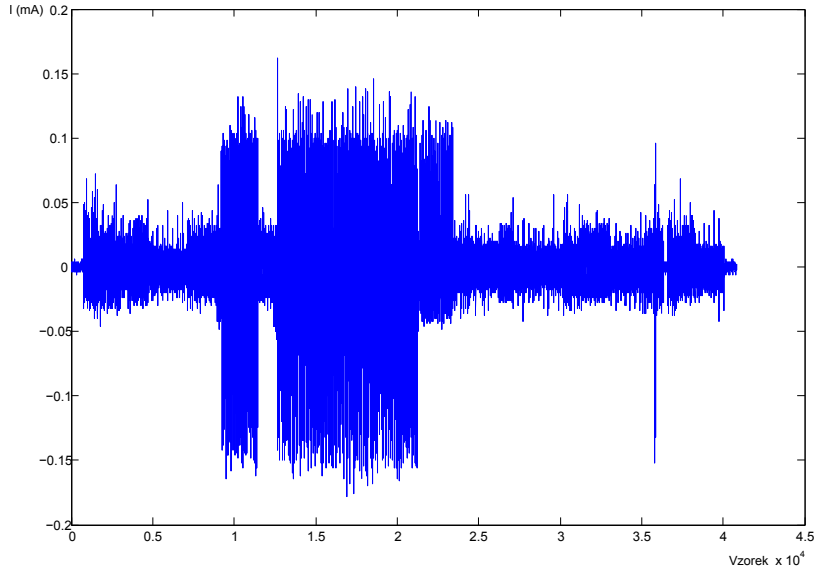
1. nahrání vstupních hodnot,
2. vytvoření instance RSA třídy,
3. šifrování dat,
4. export výsledku šifrování.

Celkový průběh proudové spotřeby čipové karty při volání kryptografické operace uložené na čipové kartě lze rozdělit do tří částí. První část průběhu proudové spotřeby jsou vstupní hodnoty (parametry) nahrány z počítače (klient) na čipovou kartu (server), v druhé částí dochází k samotnému provádění operací a algoritmů implementovaných na čipové kartě a v třetí části průběhu vrací (`return`) zpět do počítače požadovaný výsledek.

Podrobnější analýza nebyla možná z důvodu nenalezení způsobu pro správnou synchronizaci a zarovnání neměřených průběhů. Průběhy pro hodnoty různých klíčů nebyli možné navzájem porovnávat, tudíž nebylo možné pozorovat například rozdíl proudové spotřeby při operacích šifrování/dešifrování, jak lze vidět na obr.2.2. Provedené způsoby a možnosti synchronizace osciloskopu jsou popsány v kapitole 3.6 a možná řešení v závěru.



Obr. 3.7: Jednoduchá proudová analýza RSA algoritmu s délkou klíče 1024 bitů.



Obr. 3.8: Jednoduchá proudová analýza RSA algoritmu s délkou klíče 512 bitů.

SPA DES algoritmu

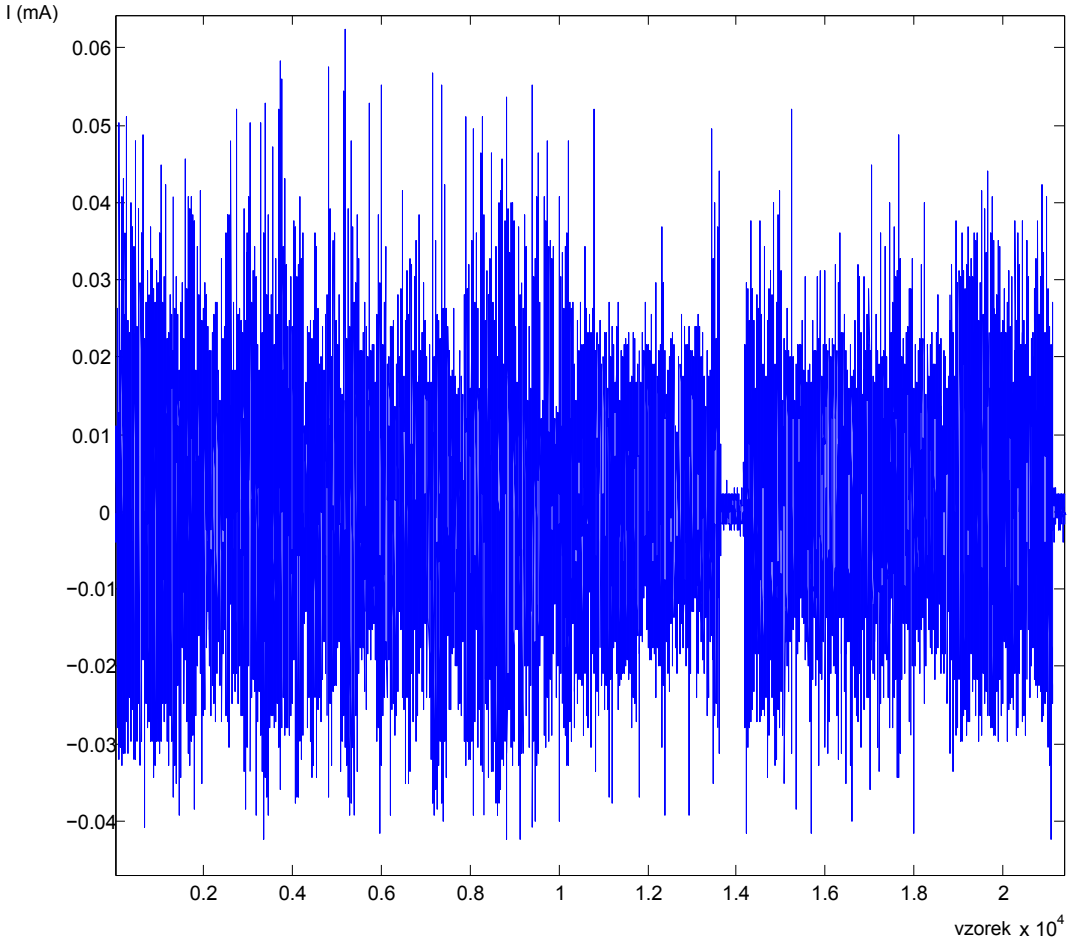
V klientské části se definují šifrovací klíče DES algoritmu. Šifrovací klíč `Key` a inicializační vektor `IV` a data, v podobě textu převedeného pomocí *C#* konvertoru do byte kódu, která se mají zašifrovat – `DataToEncrypt`. Poté zavoláme funkci šifrování service.DESEncrypt na čipové kartě se vstupními parametry. Vstupní parametry jsou data k šifrování, šifrovací klíč a inicializační vektor IV.

```
byte[] res = service.DESEncrypt(DataToEncrypt, DESd.Key, DESd.IV);
```

V serverové části jsou vstupní parametry poslané prostřednictvím APDU kanálu deklarovány v hlavičce volané funkce `DESEncrypt(...)`. Prvním krokem šifrování je vytvoření instance DES šifrování funkcí `ICryptoTransform` s požadovanými vstupními parametry. Nyní lze požadovaná data zašifrovat. V posledním kroku se vyexportují (`return`) zašifrovaná data zpět do klientské části.

```
public byte[] DESEncrypt(byte[] DataToEncrypt, byte[] key, byte[] IV)
  ICryptoTransform encrypt = DESd.CreateEncryptor(key,IV);
  byte[] res = encrypt.TransformFinalBlock(DataToEncrypt, 0,
                                                DataToEncrypt.Length);
return res;
```

Proudová analýza DES algoritmu je zobrazena na obr.3.9.

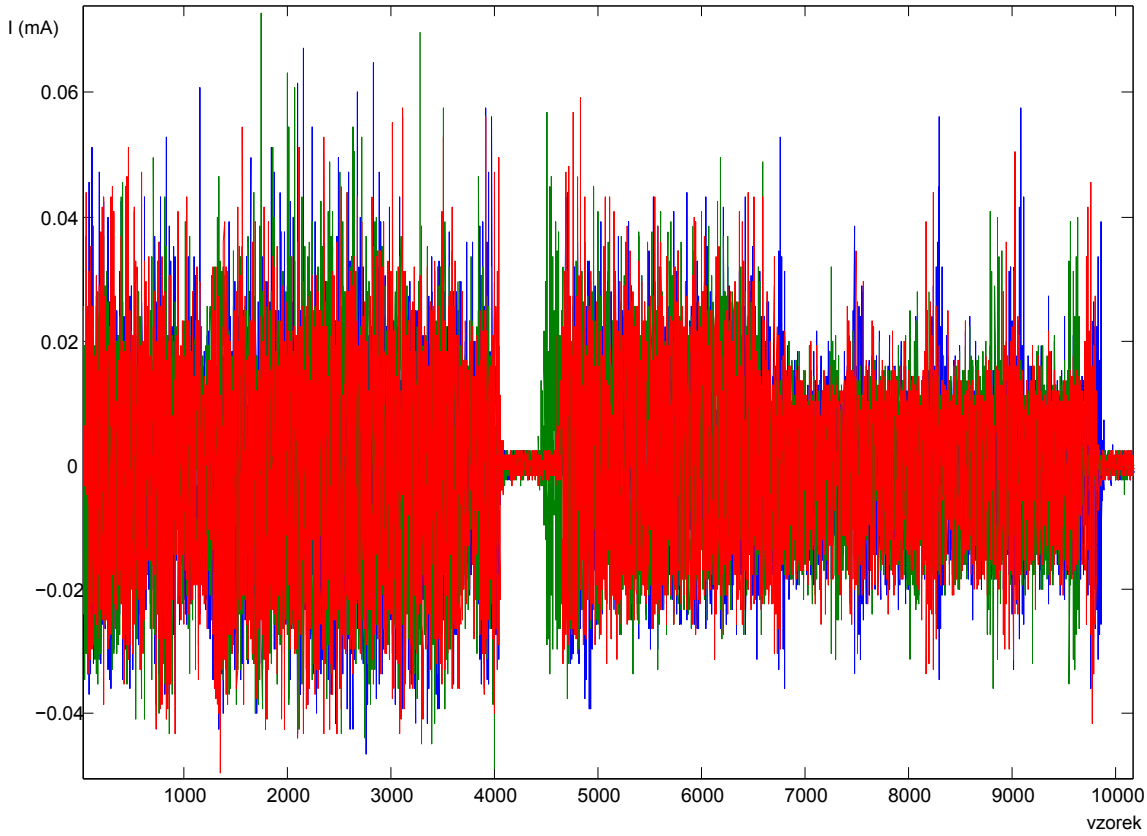


Obr. 3.9: Jednoduchá proudová analýza DES algoritmu.

Realizace diferenciální proudové analýzy

Čipová karta Gemalto byla pro diferenciální proudovou analýzu implementována upravenou částí šifrovacího algoritmu AES. Inicializační fází `AddRoundKey` a operací `SubBytes`. Cílem této analýzy je nalézt hodnotu prvního bajtu šifrovacího klíče.

Diferenciální analýza byla provedena podle obecně známých kroků uvedených v kapitole 2.2. V prvním kroku byla zvolena vnitřní hodnota algoritmu AES, která závisí na známých vstupních datech a námi zvoleném šifrovacím klíči. Dále byly změřeny průběhy proudové spotřeby čipové karty při procesu šifrování 200 bloků vstupních dat s měnící se námi známou hodnotou prvního bajtu. Průběhy byly zarovnány metodou oříznutí podle určité hodnoty první špičky průběhu. Obr.3.10 znázorňuje průběhy spotřeby čipové karty při šifrování 3 náhodných vstupních dat.



Obr. 3.10: Šifrování 3 náhodných vstupních dat.

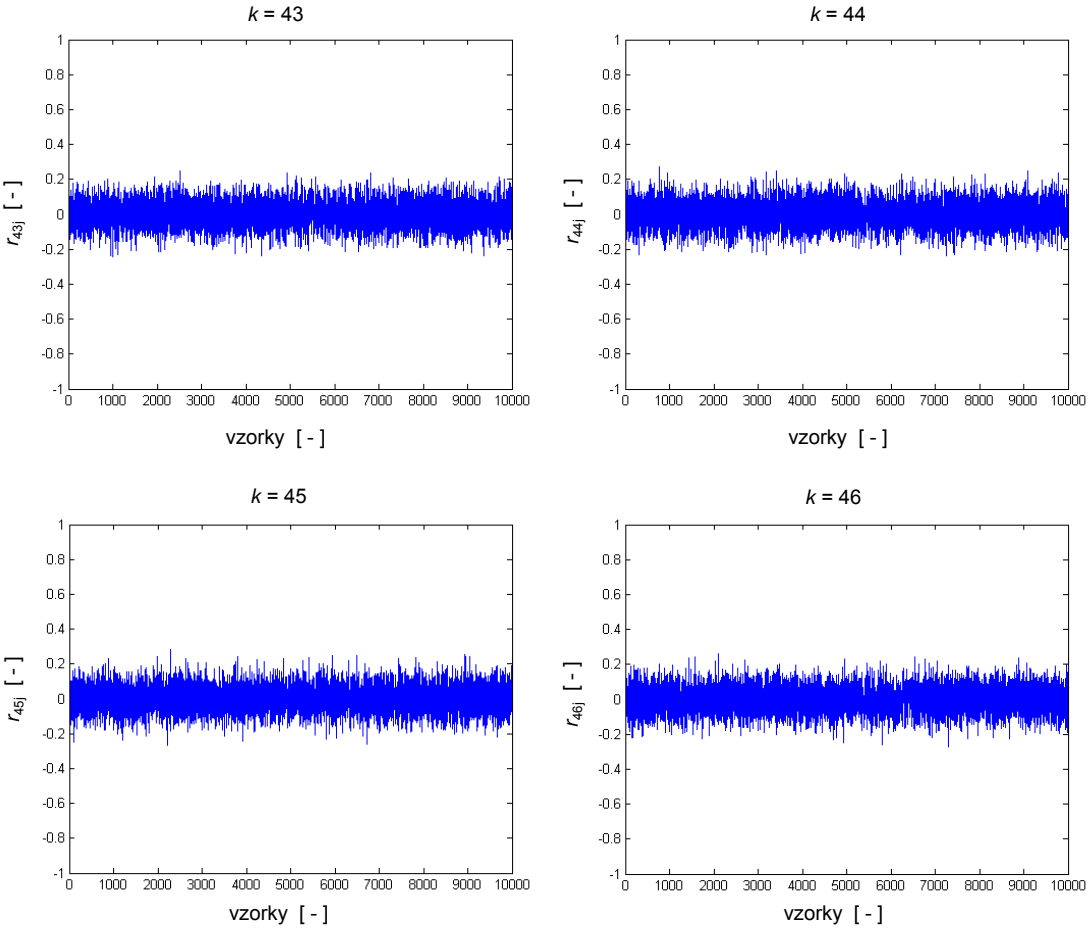
Další kroky DPA útoku byli simulovány v programu MATLAB[6]. Byla sestavena matice hypotéz vnitřních hodnot pro 256 bloků vstupních dat. Hledanému prvnímu bajtu šifrovacího klíče odpovídá první bajt každého bloku. Získáme 256×256 hypotéz vnitřních hodnot.

[6]Data k demonstraci DPA útoku získána z http://www.cs.bris.ac.uk/home/eoswald/opensca.html

V předposledním kroku DPA útoku se vytvoří simulace proudové spotřeby čipové karty. V našem případě byl zvolen model Hammingovy váhy. Aplikací Hammingovy váhy na matici hypotéz vnitřních hodnot získáme matici hypotéz o spotřebě.

V posledním kroku dojde k vyhodnocení míry lineární závislosti naměřených průběhů a hypotéz proudové spotřeby pro všechna vstupní data a klíče pomocí použitého výpočtu korelačního koeficientu. Z výsledné matice korelačních koeficientů jsou zaznamenány průběhy pro hypotézy klíčů 1 až 256. Dle teoretických předpokladů by se u hodnoty hypotézy klíče shodujícím s hodnotou manuálně zvoleného prvního bajtu šifrovacího klíče hodnoty 44 měli objevit v průběhu hypotézy klíče pozorovatelné špičky.

Na obr.3.11 jsou zobrazeny průběhy pro hypotézy klíče 43 až 46. V průběhu se žádné výrazné špičky nevyskytují. Z toho vyplývá, že mezi naměřenými průběhy a hypotetickou spotřebou čipové karty pro daný klíč není žádná závislost. Důvodem neúspěšného DPA útoku je pravděpodobně nesprávná synchronizace osciloskopu a zarovnání průběhů, jak lze vidět na obr.3.10, která je jednou z podmínek úspěšného DPA útoku. Provedené způsoby a možnosti synchronizace osciloskopu jsou popsány v následující kapitole 3.6 a možná řešení v závěru.



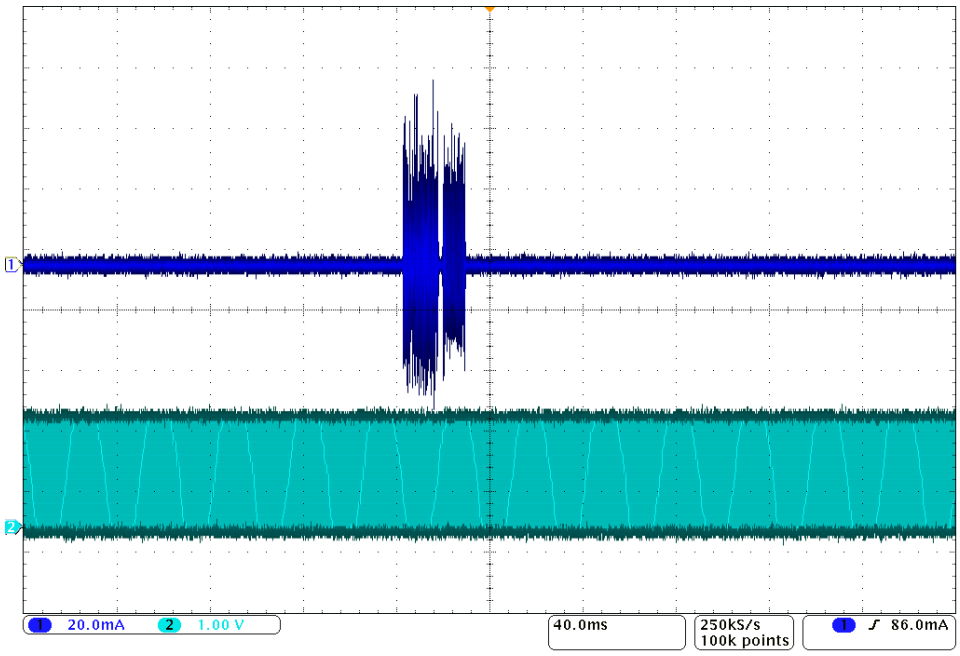
Obr. 3.11: Průběhy pro hypotézy klíče 43 až 46.

3.6 Zhodnocení a analýza

U realizace jednoduché a diferenciální analýzy nebylo možné správně definovat synchronizaci osciloskopu s kryptografickým zařízením. Při analýze naměřených průběhů musí být splněna podmínka zarovnání měřených průběhů, aby bylo možné průběhy navzájem porovnávat a dojít k požadovaným výsledkům měření. Níže jsou uvedeny a popsány možnosti, na které bylo zkoušeno zasynchronizovat osciloskop.

Hodinový signál

Synchronizační napěťová sonda se zapojí na pin CLK a GND (obr.3.2) čipové karty a připojí se na druhý kanál osciloskopu. Zobrazený hodinový signál signál obr.3.12 má velmi malou periodu kmitů a nebylo možné průběh vyexportovat tak, aby se s ním dalo dále pracovat.



Obr. 3.12: Synchronizace na hodinový signál čipové karty.

Funkce ostatních pinů čipové karty

Byli prostudovány funkce ostatních pinů obr.3.2 čipové karty. Jednou z dalších možností bylo poslat data (například logická 1) na výstup pinu I/O (sériový vstup a výstup) nebo RFU (nepoužíván). Dostupná čipová karta Gemalto .NET v2 nepodporuje programování a posílání dat na piny čipové karty.

LED dioda čtečky čipových karet HID Omnikey 3121

Čtečka čipových karet HID Omnikey 3121 [15] má LED diodu, která při provádění operací na čipové kartě bliká. Synchronizační sonda byla zapojena na LED diodu. Měřením bylo zjištěno že dioda nezačne blikat pokaždé ve stejném místě, ale s proměnlivým zpožděním při provádění operací na čipové kartě.

Vložení prázdných instrukcí

Dále byly do implementovaného programu na čipovou kartu mezi instrukce RSA algoritmu vloženy prázdné instrukce a námi zvolená operace vložena mezi prázdné instrukce. Prázdná instrukce v *C#* se deklaruje středníkem `;` nebo prázdnými závorkami `{}`. Předpoklad byl, že prázdné instrukce budou mít velmi malou proudovou spotřebu a vložená operace vytvoří proudovou špičku, podle níž průběhy zarovnáme. Měřením bylo zjištěno, že čipová karta při provádění prázdných instrukcí má průběh stejnou úroveň jak při provádění jakýchkoliv ostatních operací.

Oříznutí podle první špičky průběhu

Průběhy proudových spotřeb byly naměřeny funkcí $SINGLE$ osciloskopu a uloženy do počítače. V programu MATLAB byl vytvořen skript, který signál ořízne od první proudové špičky námi zvolené hodnoty průběhu. První proudová špička při provádění operací na čipové kartě ale nemá pokaždé konstantní velikost a průběh. Po analýze a porovnání oříznutých signálů nebyli všechny průběhy správně zarovnány.

4 ZÁVĚR

Cílem bakalářské práce bylo prostudovat problematiku proudového postranního kanálu, realizovat měřící pracoviště určené k analýze proudovým postranním kanálem a vytvořit laboratorní úlohu seznamující studenty s útoky proudovým postranním kanálem. Potřebné znalosti k problematice postranních kanálů se zaměřením na proudový postranní kanál a útoky na něj obsahuje teoretická část této práce. Teoretické základy byli použity ke správnému návrhu a sestavení měřícího pracoviště a realizace jednoduché a diferenciální proudové analýzy.

Jako kryptografické zařízení, na které byl prováděn útok proudovým postranním kanálem, byla použita čipová karta Gemalto .NET v2. K realizaci měření proudové spotřeby čipové karty při probíhajících kryptografických operacích, bylo nutné navrhnout a vyrobit jednoduchou desku plošných spojů. DPS byla navržena tak, aby bylo možné zapojit měřící proudovou sondu Tektronix CT-6 a realizovat tak útok proudovým postranním kanálem. Následně bylo sestaveno měřící pracoviště určeného k analýze proudovým postranním kanálem a mohla být provedena měření.

V první části jednoduché proudové analýzy byla na čipovou kartu Gemalto implementována aplikace s šifrováním RSA algoritmem. Aplikace obsahovala vytvoření RSA algoritmu s námi manuálně zvolenou délkou a hodnotami šifrovacího klíče. Poté následovalo zašifrování námi zvolených dat. Při této analýze byl zkoumán celkový průběh proudové spotřeby čipové karty při volání funkcí na čipové kartě. Vliv délky šifrovacího klíče a manuálně nastavené různé hodnoty šifrovacích parametrů na proudové spotřebě čipové karty. Analýzou RSA šifrování klíči o délce 512 a 1024 bitů bylo zjištěno, že se zvětšující se délkou šifrovacího klíče narůstá celkový čas průběhu proudové spotřeby šifrování. Manuální změny hodnot šifrovacích parametrů nebylo možné viditelně pozorovat na průběhu proudové spotřeby, podrobnější analýza nebyla možná z důvodu nenalezení způsobu pro správnou synchronizaci a zarovnání naměřených průběhů.

V druhé části jednoduché proudové analýzy byla na čipovou kartu implementována aplikace s šifrováním DES algoritmem. Aplikace obsahovala opět vytvoření DES algoritmu s námi manuálně zvolenými hodnotami šifrovacího klíče. Poté následovalo zašifrování námi zvolených dat. Při operaci šifrování nebyli pozorovány opakující se části 16-ti rund na celkovém průběhu šifrování DES algoritmem.

Při diferenciální proudové analýze byla na čipovou kartu implementována aplikace vykonávající funkce `AddRoundKey` a `SubBytes` šifrovacího algoritmu AES. Diferenciální analýza měla za úkol odhalit hodnotu prvního bajtu šifrovacího klíče, byla cílena na první výstupní bajt šifrovaných dat operace `SubBytes`. Výsledný průběh pro hypotézu klíče 44 (hledaná hodnota) neobsahoval žádné špičky. Z toho vyplývá, že mezi naměřenými průběhy a hypotetickou spotřebou čipové karty pro daný klíč není žádná závislost.

Důvodem nenalezení proudových špiček u diferenciální proudové analýzy a podrobnější analýzy u jednoduché proudové analýzy byla nesprávně definovaná synchronizace osciloskopu s kryptografickým zařízením. Všechny naměřené průběhy nebyly správně zarovnány, tudíž nebylo možné porovnávat změny hodnot klíčů a dojít k požadovaným výsledkům. Naměřené průběhy spotřeb kvůli synchronizaci nebylo možné pomocí osciloskopu průměrovat (snímací mód Average), tudíž mohlo docházet k nárůstům šumu, a tím i docházet k překrytí částí průběhů.

Jako možné řešení se jeví implementace na jinou čipovou kartu, která umožňuje uživateli komunikaci s piny, na který by bylo možné osciloskop zasynchronizovat. Vzhledem k dosaženým výsledkům měření, nelze požadovanou laboratorní úlohu pro studenty vytvořit tak, aby je seznámila s útoky proudovým postranním kanálem.

LITERATURA

[1] MANGARD, Stefan; OSWALD, Elisabeth; POPP, Thomas. *Power Analysis Attacks: Revealing the Secrets of Smart Cards.* New York: Springer, 2007. 338 s. ISBN 978-0-387-30857-9.

[2] ZHOU, Yong Bin; FENG, Deng Guo. *Side-Channel Attacks: Ten Years After Its Publication and the Impacts on Cryptographic Module Security Testing).* In [online]. 2005, [cit. 4. 12. 2012]. Dostupné z URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.100.8856&rep=rep1&type=pdf>.

[3] KOCHER, Paul C. *Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems* [online]. 1996 [cit. 2012-12-04]. Dostupné z URL: <http://www.cryptography.com/public/pdf/TimingAttacks.pdf>.

[4] HAGAI, Bar-El. *The Sorcerer's Apprentice Guide to Fault Attacks* [online]. 2004 [cit. 2012-12-04]. Dostupné z URL: <http://www.hbarel.com/publications/Sorcerers_Apprentice_Guide.pdf>.

[5] ZAPLETAL, Ondřej. *Proudový postranní kanál:* bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2012. 60 s. Vedoucí práce byl Ing. Zdeněk Martinásek

[6] MENEZES, Alfred J. *Handbook of applied cryptography.* Vyd. 1. Boca Raton: CRC Press, 1997, 780 s. ISBN 08-493-8523-7.

[7] Joye, M.; Oliver, F.: Side-Channel Analysis. In *Encyklopedia of Cryptography and security (2nd E.)* ,2011, s. 1198-1204.

[8] FIPS PUB 46-3. *Data Encryption Standard (DES)).* [online]. National Institute of Standards and Technology, 25. 10. 1999 [cit. 2012-05-21]. 26 s. Dostupné z URL: <http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf>.

[9] FIPS PUB 197. *Advanced Encryption Standard (AES)* [online]. National Institute of Standards and Technology, 26. 11. 2001 [cit. 2012-05-21]. 51 s. Dostupné z URL: <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>.

[10] AES S-box. [online]. [cit. 2013-05-11]. Dostupné z URL: <http://web.math.pmf.unizg.hr/~duje/kript/tablice/aessbox.html>.

[11] BUČEK, Jiří. *Útok postrannímí kanály.* Praha: CryptoFest, 11.6. 2011

[12] Gemalto *.NET Smart Card v2* [online]. [cit. 2013-04-24]. Dostupné z URL: <http://www.gemalto.com/dwnld/5042_070520_WP_Gemalto_.NET_Certificate_Enrollment_using_MSFT_Certificate_Services.pdf>.

[13] Standard ISO/IEC 7816-4. *Identifikační karty, karty s integrovanými obvody – část 4 – organizace, bezpečnost a příkazy pro vzájemnou výměnu* [online]. [cit. 2013-05-11]. Dostupné z URL: <http://webstore.iec.ch/preview/info_isoiec7816-4%7Bed3.0%7Den.pdf>.

[14] Standard ECMA-335. *Common Language Infrastructure (CLI)* [online]. [cit. 2013-05-11]. Dostupné z URL: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-335.pdf>.

[15] HID *Omnikey 3121 USB desktop smart card reader* [online]. [cit. 2013-04-24]. Dostupné z URL: <https://www.hidglobal.com/sites/hidglobal.com/files/resource_files/omnikey-3121-usb-desktop-reader-ds-en.pdf>.

[16] Tektronix. *DPO4000 Series Digital Phosphor Oscilloscopes: User manual.* [cit. 2012-12-07].

[17] Tektronix. *CT-6 High Frequency AC Current Probe: Instruction manual.* [online]. [cit. 2012-12-07]. Dostupné z URL: <http://www.tek.com/sites/tek.com/files/media/media/resources/60W_12572_2.pdf>.

[18] Standard ISO 7816. *Mezinárodní norma elektronických identifikačních karet s kontakty.* [online]. [cit. 2013-05-11]. Dostupné z URL: <http://www.cardwerk.com/smartcards/smartcard_standard_ISO7816.aspx>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

AES Advanced Encryption Standard – pokročilý standard pro šifrování dat

APDU Application protocol data unit – protokol aplikační vrstvy

CIL Common Intermediate Language – tzv. neutrální jazyk čipové karty

CLI Common Language Infrastructure – zajišťuje správný průběh aplikací na čipové kartě

CLR Common Language Runtime – běhové prostředí na čipové kartě

CMOS Complementary Metal-Oxide Semiconductor – technologie pro výrobu integrovaných obvodů

DES Data Encryption Standard – standard pro šifrování dat

DPA Differential Power Analysis – Diferenciální proudová analýza

DPS Deska plošných spojů

PA Power Analysis – proudová analýza

RSA Rivest-Shamir-Adleman – asymetrický šifrovací algoritmus

SCA Side-Channel Attack – útok postranním kanálem

SPA Simple Power Analysis – jednoduchá proudová analýza

URI Uniform Resource Identifier – unikátním identifikátor

XOR Exclusive disjunction – exkluzivní disjunkce

A OBSAH PŘILOŽENÉHO CD

- Elektronická verze bakalářské práce.
- Program v jazyce *C#* implementující RSA šifrování (spustitelný ve vývojovém prostředí Microsoft Visual Studio 2008).
- Program v jazyce *C#* implementující DES šifrování (spustitelný ve vývojovém prostředí Microsoft Visual Studio 2008).
- Program v jazyce *C#* implementující operaci `AddRoundKey` *a* `SubBytes` (spustitelný ve vývojovém prostředí Microsoft Visual Studio 2008).
- Veškerá data potřebná pro diferenciální proudovou analýzu ve formě matic s příponou *.mat (spustitelné ve vývojovém prostředí Matlab 7.12.0).
- Skripty použité pro zpracování naměřených proudových průběhů při diferenciální proudové analýze (spustitelné ve vývojovém prostředí Matlab 7.12.0).