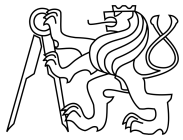
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



ZADÁNÍ DIPLOMOVÉ PRÁCE

Název:	Diferenciální kryptoanalýza šifry Baby Rijndael
Student:	Bc. Jakub Tomanek
Vedoucí:	Ing. Josef Kokeš
Studijní program:	Informatika
Studijní obor:	Počítačová bezpečnost
Katedra:	Katedra počítačových systémů
Platnost zadání:	Do konce letního semestru 2017/18

Pokyny pro vypracování

Seznamte se s technikou diferenciální kryptoanalýzy včetně souvisejících technik (využití nemožných diferenciálů apod.). Zpracujte rešerši aktuálního stavu kryptoanalýzy Baby Rijndael a Rijndael. Proveďte diferenciální kryptoanalýzu šifry Baby Rijndael a implementujte program, který provede útok touto technikou. Diskutujte dosažené výsledky ve vztahu k ostatním technikám a pokuste se odhadnout jejich dopad na šifru Rijndael (AES).

Seznam odborné literatury

Dodá vedoucí práce.

prof. Ing. Róbert Lórencz, CSc.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
děkan

V Praze dne 24. ledna 2017

České vysoké učení technické v Praze
Fakulta informačních technologií
Katedra počítačových systémů

Diplomová práce

Diferenciální kryptoanalýza šifry Baby Rijndael

Bc. Jakub Tomanek

Vedoucí práce: Ing. Josef Kokeš

3. května 2017

Poděkování

Na tomto místě bych rád poděkoval panu Ing. Josefu Kokešovi za jeho pravidelné schůzky a cenné rady, kterými mi pomohl a inspiroval k sepsání této práce. Dále bych chtěl poděkovat panu prof. Róbertu Lórenczovi, CSc. za jeho podněty a materiály, které mi poskytl a které mě v mnohých směrech inspirovaly. V neposlední řadě bych chtěl poděkovat svým rodičům, celé své rodině a přátelům, od kterých jsem cítil podporu po celou dobu svého studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 3. května 2017 .....

Odkaz na tuto práci

Tomanek, Jakub. *Diferenciální kryptoanalýza šifry Baby Rijndael.* Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.

Abstrakt

V této práci se zabýváme metodou diferenciální kryptoanalýzy aplikovanou na šifru Baby Rijndael. V prvních dvou kapitolách se přesvědčíme o podobnosti designového návrhu šifer Rijndael a Baby Rijndael. Dále si uvedeme základní principy diferenciální kryptoanalýzy, kterými jsou předvýpočet diferenciálních charakteristik a jejich následné využití během útoku a extrakce šifrovacího klíče. U předvýpočtu dif. charakteristik se soustředíme na jejich možné sjednocování a shlukování. U metod útoků na šifru zkoumáme jednotlivé parametry, které mají vliv na výslednou úspěšnost získání šifrovacího klíče. V závěru odhadujeme paměťovou a výpočetní složitost jednotlivých variant útoků a porovnáváme naše výsledky s jinými doposud publikovanými pracemi. V provedených útocích se nám povedlo v průměru extrahovat šifrovací klíč ve 26 % v kratším čase oproti hrubé síle.

Klíčová slova Rijndael, Baby Rijndael, diferenciální kryptoanalýza, diferenciální charakteristika

Abstract

In this thesis we deal with the methods of differential cryptanalsis applied to the Baby Rijndael cipher. In the first two chapters, we demonstrate the similarness of the Rijndael and Baby Rijndael ciphers. Then we discuss the basic principles of differential cryptanalysis. These are the precomputation of differential characteristics and their later usage for key extraction. We focus on the possibility of merging and clustering of the differential characteristics. We discuss the parameters which have an impact on the overall success of key extraction. Finally we estimate the memory and time complexity of our attack in comparison with the brute force approach and we compare our results to results of other works. In our attack we were able to extract secret key in 26 % cases on average in better time than the brute force attack.

Keywords Rijndael, Baby Rijndael, differential cryptanalysis, differential characteristics

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Kryptografie je součástí lidstva už od nepaměti a je obecně znám fenomén nikdy nekončícího soupeření mezi kryptology[1] a kryptoanalytiky[2]. Za celou tu dobu se situace převahy jedné či druhé strany několikrát vystřídala. Kryptologové měli vždy převahu v dobách, kdy přišli s jakýmkoliv novým a inovativním kryptosystémem. Příklady převah kryptoanalytiků můžou být například objevení frekvenční analýzy, prolomení Vigenerovy šifry, mechanizace útoků hrubou silou či poměrně novou technikou útoku postranními kanály[3].

V této práci prověříme slabá místa šifry Baby Rijndael, která byla dle autora navržena se stejnými principy jako originální šifra Rijndael. Tyto principy si v této práci uvedeme a zároveň ukážeme, že Baby Rijndael je dobře navrženou redukovanou šifrou. Výhodou této redukované šifry je fakt, že díky menší velikosti klíče lze prakticky ověřit její vlastnosti a chování vzhledem k teoretickým útokům, publikovaným na originální šifru Rijndael. Mimo stručné historie a okolnosti vzniku šifry Rijndael si v úvodní kapitole představíme ty nejslibnější výsledky dosud publikovaných kryptoanalýz šifry Rijndael a také Baby Rijndael. V následujících kapitolách si představíme zmíněnou strukturu a designové prvky obou šifer. Následně si ukážeme obecný postup diferenciální kryptoanalýzy. Tyto poznatky využijeme při konstrukci této techniky na šifru Baby Rijndael. Dvě samostatné kapitoly budeme věnovat několika metodám nutných předvýpočtu pro útoky a následně budeme zkoumat vliv parametrů jednotlivých útoků na zkoumané šifře. V poslední kapitole vyhodnotíme paměťovou a výpočetní náročnost útoku a porovnáme dosažené výsledky s ostatními publikacemi.

[1] Kryptolog je člověk, který se zabývá tvorbou nových šifer.

[2] Kryptoanalytik je člověk, který se zabývá jakýmkoliv typem luštění šifer.

[3] Tento útok neprolamuje šifru obecně jako takovou, nýbrž její nedokonalou implementaci.

Kapitola 1

Historie a evoluce diferenciální kryptoanalýzy

Tato práce stojí na kryptoanalytických technikách představených Elim Bihamem a Adim Shamirem. Tito kryptoanalytici vydali v roce 1993 knihu *"Differential Cryptanalysis of the Data Encryption Standard"*, ve které představují novou techniku diferenciální kryptoanalýzy (DK) [1].

V 70. letech minulého století byla v laboratořích IBM konstruována šifra DES. Tato šifra byla předložena do výběrového řízení NBA (National Bureau of Standards) jako kandidát na šifrovací algoritmus určený pro ochranu neutajovaných vládních dat v elektronické podobě. Po několika úpravách konzultovaných s NSA (National Security Agency) byla tato šifra oficiálně uznána jako americký standard pro ochranu elektronických dat pro úřady a komerční sféru. Mezi kontroverzní úpravy odchylující se od originálního návrhu patřilo zkrácení délky klíče na 56 bitů a úprava S-boxů. Mezi kryptoanalytiky tak vzniklo podezření, že těmito úpravami byla do šifry propašována zadní vrátka. Do začátku 90. let minulého století bylo zveřejněno hned několik typů útoků na tuto šifru. Mezi nimi byla také diferenciální kryptoanalýza zveřejněná Bihamem a Shamirem. Po zveřejnění této metody vyšlo najevo, že designéři DESu znali tuto techniku útoku již při návrhu šifry a z důvodů národní bezpečnosti byli požádáni tuto techniku a designové návrhy držet v tajnosti. Tyto techniky a designové návrhy pak v roce 1994 zveřejnil spoludesignér DESu Don Coppersmith v [2].

Kombinací stále novějších typů útoků na DES společně se vzrůstajícím výpočetním výkonem počítačů šifra ztrácela atribut bezpečného standardu. Z tohoto důvodu Národní institut standardů a technologie (NIST) vyhlásil v roce 1997 veřejnou soutěž na nový kryptosystém, který nahradí dosluhující DES. Soutěže se zúčastnilo 15 kandidátů a do užšího finále postoupilo pět nejlepších. Testována nebyla pouze odolnost vůči známým útokům, ale také možnosti paralelizace nebo cena implementace v hardware (např. FPGA, čipové karty, 8-bitové procesory atd.) či software. Po více než roce a několika

finále	1. kolo	šifra	autoři
✓	✓	Rijndael	Vincent Rijmen, Joan Daemen
✓	✓	MARS	Carolynn Burwick, Don Coppersmith et al.
✓	✓	RC6	Ron Rivest, Matt Robshaw et al.
✓	✓	Serpent	Ross Anderson, Eli Biham, Lars Knudsen
✓	✓	Twofish	Bruce Schneier, John Kelsey et al.
	✓	CAST-256	C. Adams, S. Tavares, H. Heys, M. Wiener
	✓	CRYPTON	Chae Hoon Lim
	✓	DEAL	Lars Knudsen
	✓	DFC	Henri Gilbert, Marc Girault et al.
	✓	E2	Masayuki Kanda, Shiho Moriai et al.
	✓	FROG	Dianelos Georgoudis, Damian Leroux et al.
	✓	HPC	Richard Schroeppel
	✓	LOKI97	L. Brown, J. Pieprzyk, J. Seberry
	✓	MAGENTA	M. Jacobson Jr., K. Huber
	✓	SAFER+	J. Massey, G. Khachatrian, M. Kuregian

Tabulka 1.1: Kandidáti a finalisté veřejné soutěže na AES

konferencích, kde byly prezentovány kryptoanalytické pokusy prolomení pětice finalistů, byla za vítěznou šifru vybrána šifra Rijndael. V listopadu roku 2001 pak byl Rijndael oficiálně uznán jako národní standard v USA pro šifrování neutajovaných dat pro úřady a komerční sféru [3].

1.1 Diferenciální kryptoanalýzy Rijndaelu

Klasickou diferenciální kryptoanalýzou se inspirovalo do dnešního dne velké množství prací. Vznikly tak nové typy útoků, které vylepšují schopnosti kryptoanalytiků prolamovat i nově vznikající šifry.

1.1.1 Boomerang attack

Prním příkladem útoku odvozeného od klasické diferenciální kryptoanalýzy je útok nazvaný *boomerang attack*. Autorem tohoto útoku je David Wagner a jedná se o typ útoku s voleným otevřeným textem a voleným šifrovým textem. Ve svém článku [4] z roku 1999 popisuje boomerang attack aplikovaný na několik druhů šifer. Jednou z prolomených šifer byla COCONUT98. Navzdory konstrukci této šifry, která byla designována s prokazatelnou odolností vůči lineární a diferenciální kryptoanalýze, se povedlo s využitím boomerang attack získat šifrovací klíč v reálném čase s reálnými prostředky.

1.1.2 Nemožné diferenciály

Dalším příkladem odvozeného útoku je *útok pomocí nemožných diferenciálů* (impossible differentials). Poprvé jej dle data publikování práce použil Lars Knudsen v roce 1998 v rámci ověření bezpečnosti jeho vlastního kryptosystému DEAL, se kterým se hlásil do soutěže na AES. Název tomuto novému kryptoanalytickému útoku však dali Biham, Biryukov a Shamir. Nezvyklou sílu této metody představili na konferenci CRYPTO'98, kde předvedli útok na šifry IDEA a Skipjack[4]. Tyto útoky a uvažovaná metoda byla autory sepsána v [5]. Bližší popis této metody si uvedeme později, protože tato metoda byla aplikována také na šifru Baby Rijndael se zajímavými výsledky.

1.1.3 Zkrácené diferenciály

Lars Knudsen popsal zkrácené diferenciály v „Truncated and higher order differentials". Knudsen představil tento koncept na šifře DES. V běžné diferenciální kryptoanalýze se snažíme predikovat n-bitovou hodnotu šifrového textu po určitém množství rund. Knudsen však navrhuje, že se můžeme zaměřit jen na několik bitů uvažované diference. Diferenci, která predikuje jen jistou část z n-bitové hodnoty, nazýváme zkrácený diferenciál [6]. Tento typ útoku, založený na zkrácených diferencích, hezky shrnuje Jan Říha v práci [7]. V ní uvádí, že tento typ útoků využívá faktu, že v některých šifrách dochází ke shlukování diferenciálních charakteristik. Pod pojmem shlukování si můžeme představit situaci, kdy pro danou vstupní a výstupní diferenci existuje mnoho různých diferenciálních charakteristik. Jednotlivé diferenciální charakteristiky jsou navzájem nezávislé a lze tak spočítat očekávanou pravděpodobnost takového shluku. Tento typ útoku je úspěšný u šifer, které pracují se skupinou bitů namísto jednotlivých bitů a Rijndael je příkladem takové šifry, protože pracuje s celými byty. Knudsen tento typ útoku demonstroval na šifře DES, jehož 6-tirundovou verzi dokázal prolomit se 46 volenými otevřenými texty a výpočetní náročností 3500-násobku doby šifrování.

1.2 Kryptoanalýzy Baby Rijndaelu

Kryptoanalýzou Baby Rijndaelu se zabývalo do dnešní doby hned několik prací. Uvedu tedy jejich stručný přehled, protože z jisté části má práce na nich staví a čerpá.

Diferenciální kryptoanalýza

Diferenciální kryptoanalýzou Baby Rijndaelu se již v roce 2009 zabýval student Iowa State University Jonathan Wrolstad. V práci [8] popisuje po-

[4]Skipjack je 32-rundová šifra navržena NSA. S pomocí útoku *impossible differentials* se na zmiňované konferenci povedlo prolomit 31-rundovou verzi tohoto kryptosystému

drobně techniku klasické diferenciální kryptoanalýzy na šifře Baby Rijndael. Ve své práci odvozuje teoretické meze a nastiňuje principy útoku na 2, 3 a 4-rundovou verzi šifry. Má práce svým způsobem navazuje na tu jeho a tyto útoky testuje v mnoha obměnách.

Lineární kryptoanalýza

Lineární kryptoanalýzou šifry Baby Rijndael se v roce 2013 zabýval Josef Kokeš. Se svou prací vyhrál prestižní soutěž ACM SPY 2013 a společně vedoucím této práce, panem prof. Róbertem Lórenczem, CSc., zpopularizovali na FIT ČVUT vědeckou činnost této disciplíny. V následujících letech vznikly 2 práce z oblasti kryptoanalýzy a 3. v pořadí zrovna čtenář drží v ruce.

Lineární kryptoanalýza (LK) spolu s diferenciální kryptoanalýzou tvoří základní dvojici kryptoanalytických metod. LK poprvé popsal Mitsuru Matsui v roce 1993 jako teoretický útok na šifru DES. LK patří do kategorie *known plaintext attack*, tedy útok se znalostí otevřeného textu. Tento způsob útoku se snaží vyjádřit určitou část analyzované šifry v podobě lineární funkce, která dává do souvislosti bity otevřeného textu a bity některého vnitřního stavu šifry. Kvůli nelineární operaci, která se vyskytuje v každé šifře, to nelze udělat dokonale. Pokud je ale nelineární operace navržená špatně, může být možné ji s větší pravděpodobností lineární funkcí aproximovat. Techniky lineární kryptoanalýzy pomáhají najít takové aproximace, které budou mít co největší pravděpodobnost, a následně je využít k nalezení části šifrovacího klíče [9].

Kokeš ve své práci dokázal, že zkoumaná šifra splňuje všechny požadavky, principy a designová rozhodnutí stanovená při návrhu šifry Rijndael. Dle autora je jedinou návrhovou slabinou zkoumané šifry operace *ShiftRows*, která by mohla představovat menší odolnost vůči diferenciální kryptoanalýze a *Square útoku*. Mimo tento nedostatek však autor ukázal, že dává smysl analyzovat odolnost Rijndaelu pomocí redukované šifry Baby Rijndael [9]. Autor ve své práci také zavádí zajímavé metody ekvivalentních úprav struktury šifry Baby Rijndael pro zjednodušení implementace kryptoanalýzy. Tyto techniky využijeme i v této práci. Jejich bližší popis si uvedeme v kapitole 4

Algebraická kryptoanalýza

V roce 2016 vypracovala Lenka Vábková diplomovou práci [10] na téma algebraické kryptoanalýzy námi zkoumané šifry. Algebraická kryptoanalýza je druh útoku se znalostí otevřeného a šifrového textu, kde se zkoumaný kryptosystém pokoušíme popsat rovnicemi do stupně maximálně 2 a jejich soustavy pak řešíme pomocí známých algoritmů. Vábková ve své práci útočila na jednorundovou, dvourundovou a čtyřrundovou variantu Baby Rijndaelu. K řešení soustav nelineárních rovnic využila algoritmy XL, XLS a T'. Úspěšnost jejího útoku na čtyřrundovou variantu se dostala na hodnotu 85,1929 %.

Nemožné diferenciály

Speciálním druhem diferenciální kryptoanalýzy se v roce 2017 v [11] zabýval Peter Poljak. Zabýval se metodou *nemožných diferenciálů* (*impossible differentials*), která problematiku extrakce klíče zjednodušuje laicky řečeno na vylučovací metodu. Pro každou vstupní diferenci a daný počet rund je předpočítaná množina nemožných diferenciálů. Pokud se během fáze extrakce klíče ukáže, že pro zkoumaný klíč se diference šifrových textů objeví v předpočítané množině nemožných diferencí, pak je možné zkoumaný klíč se 100% jistotou označit za špatný. Tato metoda má podle výsledků pana Pojlaka doposud nejlepší výsledky. Pro extrakci správného klíče si vystačil v průměru s 51 páry otevřených a šifrových textů a časová složitost oproti útoku hrubou silou byla znatelně menší. V této práci byly uvedeny dva způsoby útoku na klíč.

V prvním způsobu se autor pokoušel zjistit obě poloviny klíčů pomocí dvou po sobě jdoucích útoků na jednu polovinu klíče. Časová složitost tohoto útoku byla odhadnuta na 27671 časových jednotek a prostorová složitost na 65536 jednotek. Z pohledu časové složitosti je výhoda tohoto útoku proti hrubé síle $\frac{27671}{32768}$.

Ve druhém způsobu se autor pokoušel zjistit diferenciální kryptoanalýzou pouze první polovinu klíče a druhou polovinu pak dopočítal hrubou silou. V tomto případě pak časová složitost byla odhadnuta na 13436 časových jednotek a prostorová složitost na 256 jednotek. Z pohledu časové složitosti je výhoda tohoto útoku $\frac{13436}{32768}$ oproti útoku hrubou silou.

Kapitola 2

Rijndael a Baby Rijndael

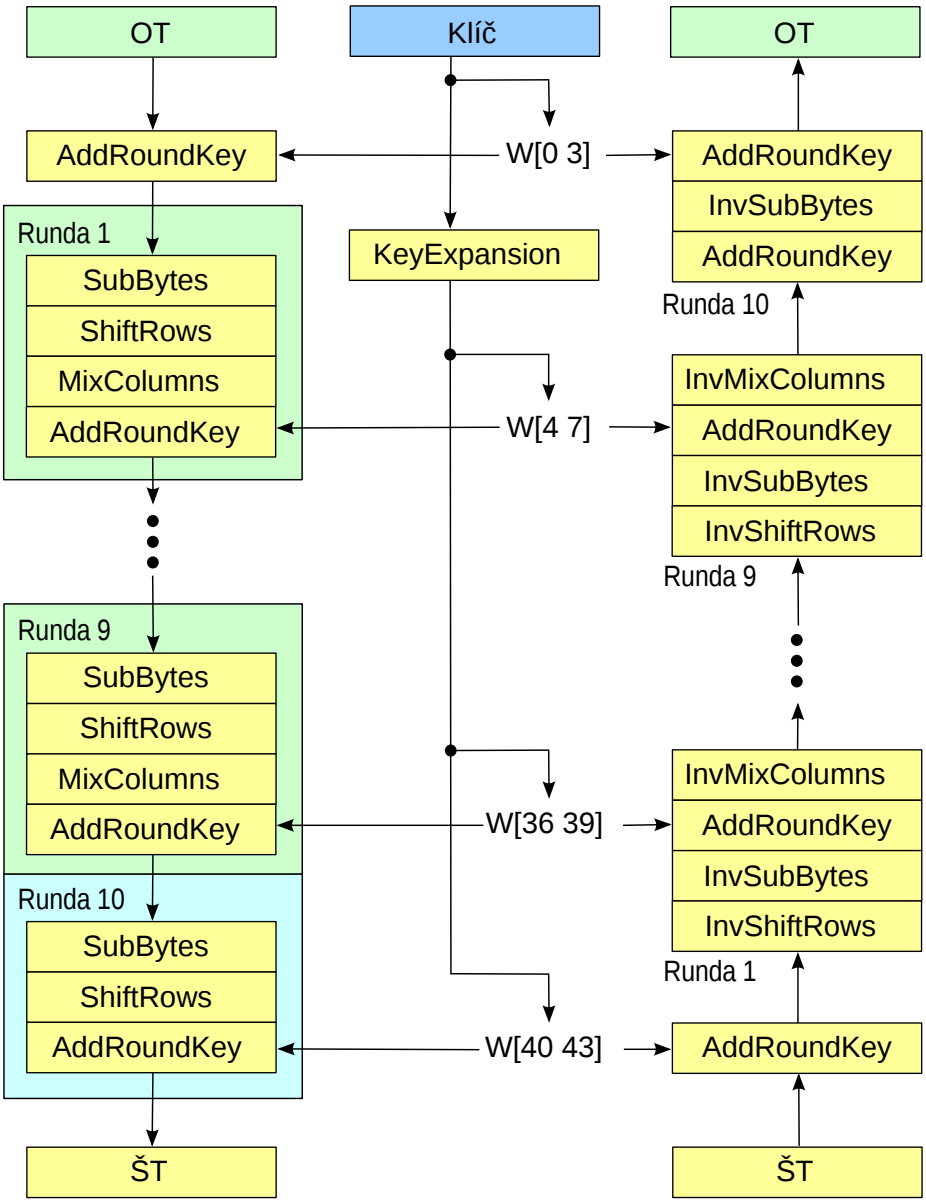
Rijndael je iterativní bloková šifra s parametrizovanou délkou bloku a velikosti klíče. Dle originálního návrhu [12] mohou být velikosti bloku a velikosti klíče voleny mezi 128 a 256 bity se 32-bitovými rozestupy. NIST ve svém standardu AES uvažuje pouze variantu se 128 bitovou velikostí bloku. Velikosti klíčů ponechal na hodnotách 128, 192 a 256 bitů. Tyto varianty se od sebe liší počtem iterací, které se také běžně nazývají rundy. Popořadě mají tyto varianty 10, 12 a 14 rund. Každá runda se sestává z operací *SubBytes*, *ShiftRows*, *MixColumns* a *AddRoundKey*. V poslední rundě šifry se vypouští operace *MixColumns*. Strukturu šifry znázorňuje obrázek 2.1

Šifru Baby Rijndael navrhl v roce 2005 profesor Cliff Bergman na univerzitě Iowa State University[14]. Díky podrobné specifikaci designu Rijndaelu bylo možné navrhnout zmenšenou verzi originálu, na kterém by bylo možné prakticky realizovat známé útoky na Rijndael. Baby Rijndael si v porovnání s Rijndaelem zachovává všechny rundovní operace s pouhým rozdílem velikosti bloku na 16 bitů a počtu rund na 4 rundy. Upravena byla ale tak, aby maximálně kopírovala chování originálního Rijndaelu. Úspěšně provedené útoky a analýzy však bohužel nelze vždy s jistotou aplikovat na originální verzi šifry Rijndael.

2.1 Struktury šifry Rijndael a Baby Rijndael

Nejmenším prvkem, se kterým pracuje šifra Rijndael je byte[5]. Nejmenší prvky zpracovávané Baby Rijndaelem jsou nibly[6]. Sekvence bytů (niblů) jsou uspořádány do čtvercové matice postupně shora dolů a zleva doprava.

[5] 8 bitů.
[6] 4 bity.



Obrázek 2.1: Struktura šifry Rijndael [13]

$$Blok_{Rijndael} = \begin{bmatrix} a_0 & a_4 & a_8 & a_{12} \\ a_1 & a_5 & a_9 & a_{13} \\ a_2 & a_6 & a_{10} & a_{14} \\ a_3 & a_7 & a_{11} & a_{15} \end{bmatrix} \qquad Blok_{BabyRijndael} = \begin{bmatrix} b_0 & b_2 \\ b_1 & b_3 \end{bmatrix}$$

$$\forall i \in \{0, \ldots, 15\} : \quad a_i \in \{0, \ldots, 255\} \tag{2.1}$$

$$\forall j \in \{0, \ldots, 3\} : \quad b_j \in \{0, \ldots, 15\} \tag{2.2}$$

Takto uspořádané sekvence bytů označujeme jako stav[7].

2.2 SubBytes

Operace *SubBytes* je jedinou nelineární operací v rámci šifry. Tato operace je v symetrických šifrách obecněji nazývána jako *S-box* (Substitution-box). *SubBytes* pracuje s jednotlivými byty stavu nezávisle a běžně se tato operace implementuje jako překladová tabulka[8]. Pro porovnání obou šifer si ale uvedeme příslušné matematické operace.

Rijndael

Z matematického hlediska se jedná o operace s prvky Galoisova tělesa $GF(2^8)$ vzhledem k ireducibilnímu polynomu

$$m_{Rijndael}(x) = x^8 + x^4 + x^3 + x + 1$$

Dle [12], tento polynom je první (nejmenší) ireducibilní polynom řádu 8. Podrobnější důvody pro volbu tohoto polynomu však nejsou ve specifikaci uvedeny. Transformovaný bajt $B = (b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0)_2$ reprezentujeme v $GF(2^8)$ jako polynom

$$b_7 x^7 + b_6 x^6 + b_5 x^5 + b_4 x^4 + b_3 x^3 + b_2 x^2 + b_1 x + b_0$$

kde $b \in GF(2)$ a x chápeme jako formální proměnnou.

Mějme tedy konkrétní byte reprezentovaný jako prvek z $GF(2^8)$. Pro tuto reprezentaci bytu najdeme její multiplikativní inverzi modulo $m_{Rijndael}(x)$. Následně s vypočtenou inverzí provedeme afinní transformaci:

[7] U Rijndaelu budeme v následujícím textu uvažovat pouze standardizovanou variantu se 128 bitovými stavy.

[8] Tuto tabulku lze nalézt v příloze B této práce.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} inv\_b_0 \\ inv\_b_1 \\ inv\_b_2 \\ inv\_b_3 \\ inv\_b_4 \\ inv\_b_5 \\ inv\_b_6 \\ inv\_b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} SubBytes(b_0) \\ SubBytes(b_1) \\ SubBytes(b_2) \\ SubBytes(b_3) \\ SubBytes(b_4) \\ SubBytes(b_5) \\ SubBytes(b_6) \\ SubBytes(b_7) \end{bmatrix}$$

Při návrhu S-boxu ([15]) byla kladena následující kritéria:

1. Invertovatelnost
2. Minimalizace největší netriviální korelace mezi lineárními kombinacemi vstupních bitů a lineární kombinací výstupních bitů
3. Složitost algebraického vyjádření v $GF(2^8)$
4. Jednoduchý slovní popis operace

Návrhová kritéria pro konstrukci operace *SubBytes* vycházela ze zkušeností s lineární, diferenciální a algebraickou kryptoanalýzou. Operace výpočtu inverze by byla dostačující proti lineární a diferenciální kryptoanalýze. Operace afinní transformace byla přidána, aby se zamezilo také útokům, jako je uveden např. *Interpolační útok* (viz. [15]).

Baby Rijndael

Matematická konstrukce *SubBytes* u Baby Rijndael se musela uzpůsobit 16 bitovému bloku. Druh a posloupnost operací však zůstal zachován. Podrobný popis operací je popsán v [8]. Na rozdíl od Rijndaelu se pracuje s polynomy z Galoisového tělesa $GF(2^4)$ vzhledem k ireducibilnímu polynomu

$$m_{BabyRijndael}(x) = x^4 + x + 1$$

Transformovaný nibl $B = (b_3b_2b_1b_0)_2$ reprezentujeme v $GF(2^4)$ jako polynom

$$b_3x^3 + b_2x^2 + b_1x + b_0$$

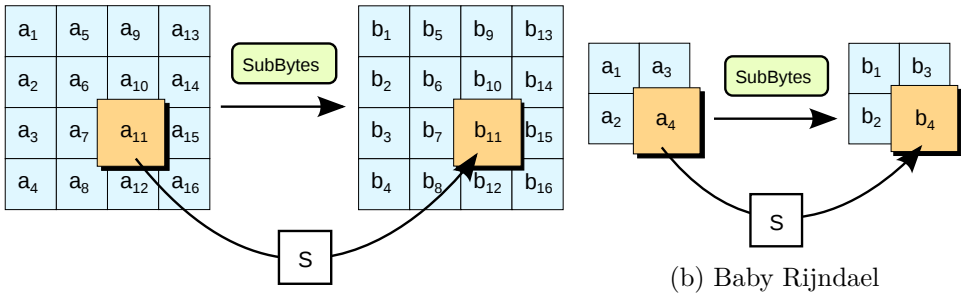
kde $b \in GF(2)$ a x chápeme jako formální proměnnou.

Mějme tedy konkrétní byte reprezentovaný jako prvek z $GF(2^4)$. Pro tuto reprezentaci bytu najdeme její multiplikativní inverzi modulo $m_{BabyRijndael}(x)$. Následně s vypočtenou inverzí provedeme afinní transformaci:

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} inv\_b_0 \\ inv\_b_1 \\ inv\_b_2 \\ inv\_b_3 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} SubBytes(b_0) \\ SubBytes(b_1) \\ SubBytes(b_2) \\ SubBytes(b_3) \end{bmatrix}$$

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
s(x)	a	4	3	b	8	e	2	c	5	7	6	f	0	1	9	d

Tabulka 2.1: SubBytes překladová tabulka pro Baby Rijndael



(a) Rijndael

(b) Baby Rijndael

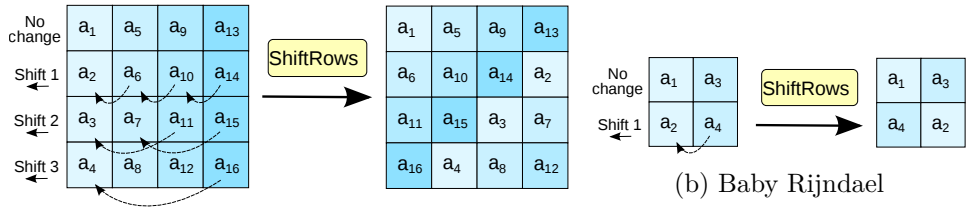
Obrázek 2.2: Operace *SubBytes*. Převzato z [13]

2.3 ShiftRows

Operace *ShiftRows* cyklicky levotočivě rotuje jednotlivé řádky stavu šifry o různý počet pozic. Informace o počtu rotovaných bytů řádku je určena indexovým pořadím řádku při indexaci začínající nulou. Při návrhu této operace ([15]) byla brána v úvahu následující kritéria:

1. Každá z rotací má jiný posun
2. Odolnost vůči útokům *Truncated differentials*
3. Odolnost vůči útoku *Square*
4. Jednoduchost

Tato operace je ve své jednoduchosti také dobře škálovatelná pro různé velikosti bloků. Otázkou ale je, zda ve zmenšené verzi Rijndaelu tato operace poskytuje dostatečnou odolnost vůči vyjmenovaným útokům. Schéma operace *ShiftRows* pro obě varianty šifer můžeme pozorovat na obrázku 2.3.



(a) Rijndael

(b) Baby Rijndael

Obrázek 2.3: Operace *ShiftRows*. Převzato z [13]

2.4 MixColums

Operace *MixColumns* promíchává bajty stavu šifry v rámci jednotlivých sloupců. Při návrhu této operace v [15] byla zvažována tato kritéria:

1. Invertovatelnost
2. Linearita v GF(2)
3. Relevantní síla difúze
4. Rychlost operace na 8-bitových procesorech
5. Symetrie
6. Jednoduchý slovní popis operace

Rijndael

Na hodnoty sloupců stavu šifry je nahlíženo jako na koeficienty polynomu $b(x) = b_0 + b_1x + b_2x^2 + b_3x^3$. Koeficienty jsou chápány jako polynomy nad $GF(2^8)$. Každý sloupec reprezentovaný polynomem $b(x)$ je vynásoben konstantním polynomem

$$c(x) = \mathtt{02} + \mathtt{01}x + \mathtt{02}x^2 + \mathtt{03}x^3$$
$$b(x)c(x) = d(x) \mod(x^4 + 1)$$

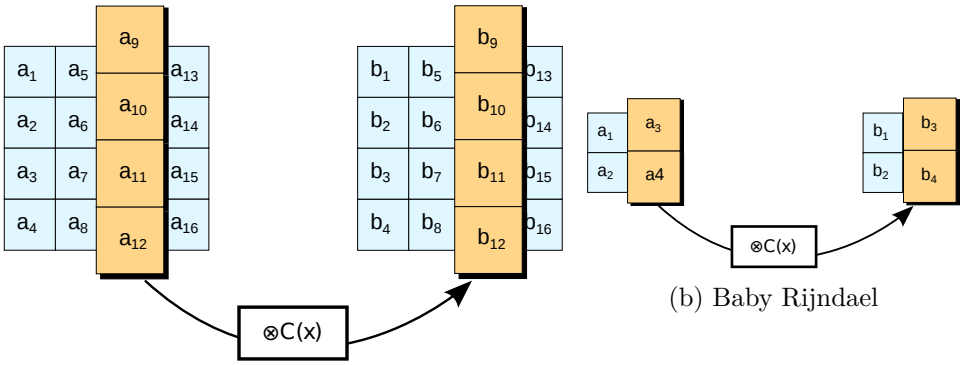
Polynom $d(x)$ reprezentuje sloupec nového stavu na odpovídající pozici.

Baby Rijndael

Návrh operace *MixColumns* v Baby Rijndaelu je na první pohled odlišná od té v Rijndaelu. Zde je operace definována jako násobení stavu binární maticí 8×8. Stav je reprezentován jako matice o rozměrech 8×2.

$$MixColumns(A) = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}_2 \times A \qquad \text{kde } A \in GF(2^8)$$

Důkaz správnosti a podobnosti operace *MixColumns* v Rijndaelu a BabyRijndaelu uvedl Kokeš v [9].



(a) Rijndael

(b) Baby Rijndael

Obrázek 2.4: Operace *MixColumns*. Převzato z [13]

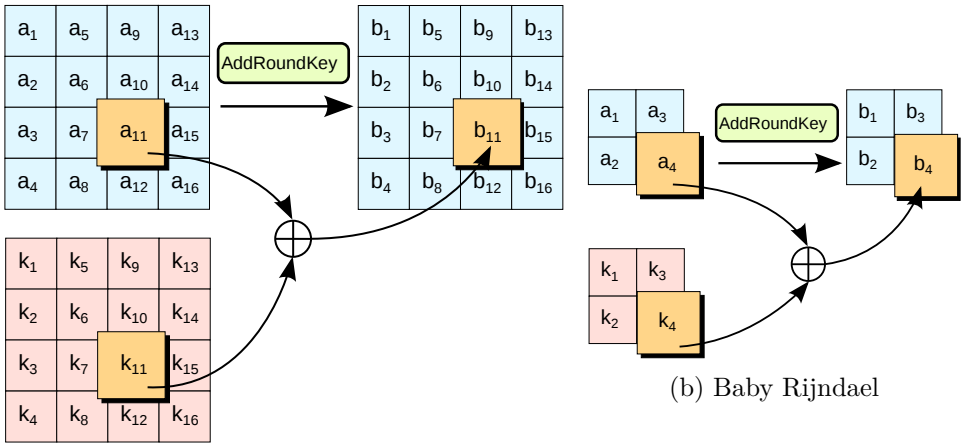
2.5 AddRoundKey

Tato operace v i-té rundě přičte (operací xor) i-tý rundovní klíč ke stavu šifry. *AddRoundKey* je totožná pro obě varianty šifer, až na velikosti stavu.

2.6 Generování rundovních klíčů

Důvodem pro zavedení rundovních klíčů v Rijndaelu byla hlavně obrana proti množině známých útoků. Příkladem mohou být *Related key attack* nebo útok, při němž kryptoanalytik zná část klíče. Důraz při návrhu algoritmu ([15]) byl kladen mimo jiné na následující aspekty:

1. Invertovatelnost, kde při znalosti některého rundovního klíče je možné zrekonstruovat všechny ostatní
2. Rychlost na široké škále procesorů



(a) Rijndael

(b) Baby Rijndael

Obrázek 2.5: Operace *AddRoundKey*. Převzato z [13]

3. Použití prvku rundovní konstanty pro eliminaci symetrie v rundovních klíčích
4. Difúze šifrového klíče do rundovních klíčů
5. Znalost části hlavního nebo rundovního klíče by nemělo dovolit dopočítání libovolného klíče
6. Jednoduchý slovní popis operace

Rijndael

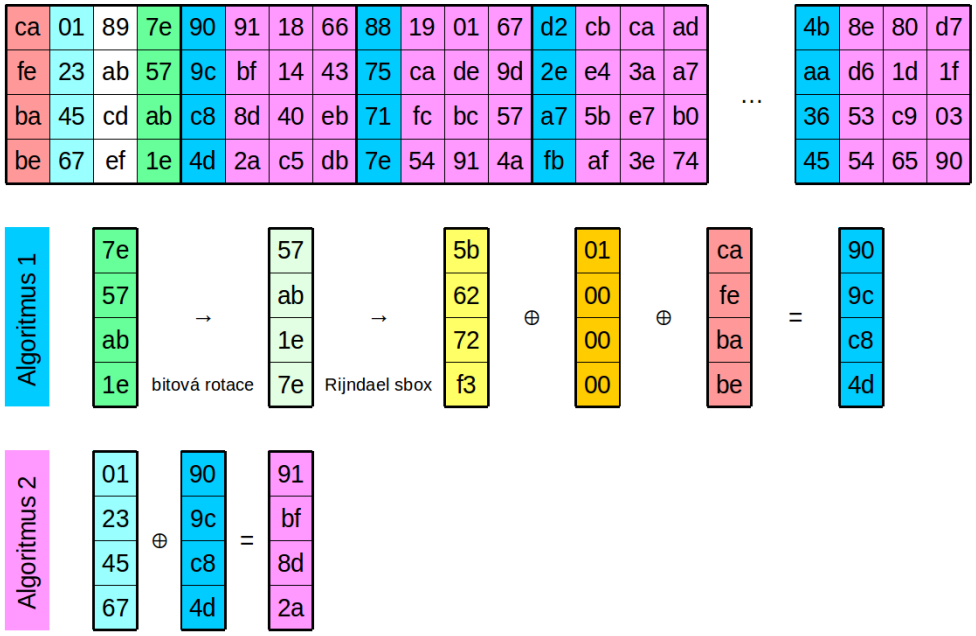
V rámci algoritmu generování rundovních klíčů se pracuje se 4-bajtovými hodnotami. Tyto hodnoty nazývejme slova. První 4 slova[9] w_0, w_1, w_2 a w_3 jsou tvořena zřetězenými bajty hlavního klíče $(k_0, \ldots, k_{15})$. Uspořádané bajty jednotlivých slov budeme reprezentovat v hranatých závorkách (viz. níže). Další čtveřice w_{4i}, w_{4i+1}, w_{4i+2} a w_{4i+3} tvoří rundovní klíče. Operace pro získání prvního slova rundovního klíče je odlišná od generování ostatních slov. Uveďme si tedy předpis pro získání libovolného slova:

$$w_i = \begin{cases} [k_{4i}, k_{4i+1}, k_{4i+2}, k_{4i+3}] & \text{pro } i = 0, 1, 2, 3 \\ w_{i-4} \oplus w_{i-1} & \text{pro } i \mod 4 \neq 0, i \geq 4 \\ SubByte(RotByte(w_{i-1})) \oplus c_i \oplus w_{i-4} & \text{pro } i \mod 4 = 0, i \geq 4 \end{cases}$$

kde *RotByte* je funkce, která rotuje slovo o jeden byte doleva[10] a c_i je slovo $(2^{\frac{i}{4}-1}, 0, 0, 0)$. Použití operace *SubBytes* a konstanty c_i při konstrukci prvního

[9] prvních 6 resp. 8 slov pro 192-bitovou resp. 256-bitovou variantu Rijndaelu
[10] $RotByte((a, b, c, d)) = (b, c, d, a)$

slova rundovních klíčů uspokojuje 3. kritérium návrhu a neklade nové požadavky na 8-bitové procesory[12].

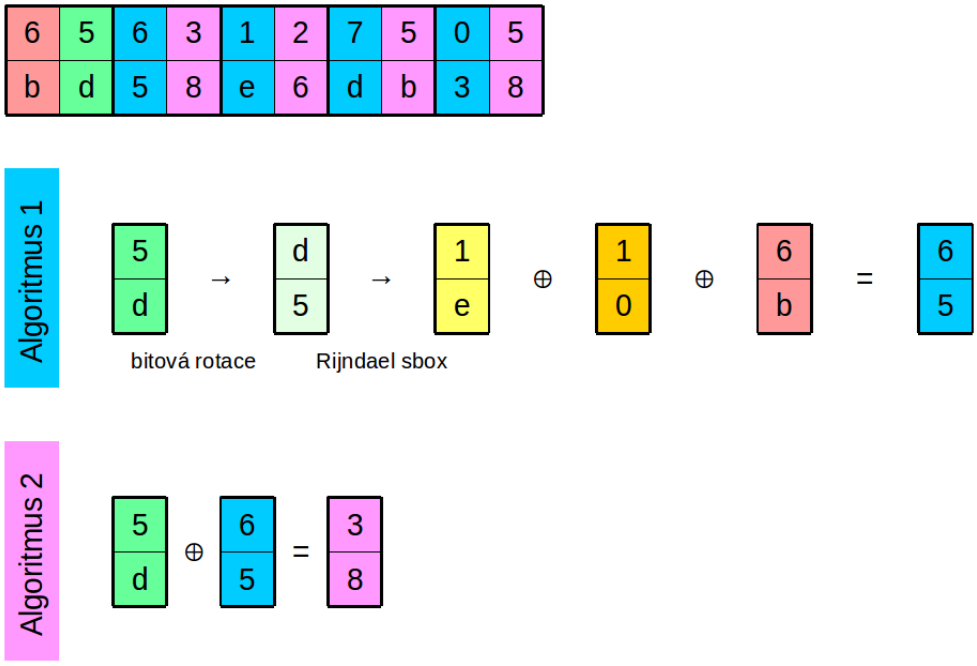


Obrázek 2.6: Generování rundovních klíčů pro Rijndael

Baby Rijndael

Předpis pro generování rundovních klíčů u Baby Rijndaelu je analogický. Pracuje se zde s jednobajtovými celky.

$$w_i = \begin{cases} [k_{2i}, k_{2i+1}] & \text{pro } i = 0, 1 \\ w_{i-2} \oplus w_{i-1} & \text{pro } i \mod 2 \neq 0, i \geq 2 \\ SubByte(RotByte(w_{i-1})) \oplus c_i \oplus w_{i-2} & \text{pro } i \mod 2 = 0, i \geq 2 \end{cases}$$



Obrázek 2.7: Generování rundovních klíčů pro Baby Rijndael

Kapitola 3

Diferenciální kryptoanalýza

Diferenciální kryptoanalýza (DK) je druh útoku s voleným otevřeným textem (OT). Tento typ útoku pracuje se specificky připravenými rozdíly[11] OT a s rozdíly jim odpovídajím šifrovým textům (ŠT). Jedním z cílů DK je nalézt správný vzor pro rozdíly otevřených textů. Vzor je následně využit pro statistické vyhodnocení omezené části prostoru klíčů. Druhým cílem DK je určit, který z klíčů byl nejpravděpodobněji použit při šifrování využívaných OT a ŠT.

3.1 Definice pojmů

Pro přesné zavedení diferenciální kryptoanalýzy si v následující sekci zadefinujeme některé klíčové pojmy:

- **Vstupy** kryptosystému budeme označovat $X = [x_0, x_1, ..., x_n]$, kde x_i reprezentuje i-tý bit X
- **Výstupy** kryptosystému budeme označovat $Y = [y_0, y_1, ..., y_n]$, kde y_i reprezentuje i-tý bit Y

Mějme dva vstupy kryptosystému X', X" a dva jim odpovídající výstupy Y' a Y"

- **Vstupní diferencí** rozumíme $\triangle X = X' \oplus X'' = [\triangle x_0, \triangle x_1, ..., \triangle x_n]$, $\triangle x_i = x'_i \oplus x''_i$, kde i reprezentuje i-tý bit diference
- **Výstupní diferencí** rozumíme $\triangle Y = Y' \oplus Y'' = [\triangle y_0, \triangle y_1, ..., \triangle y_n]$, $\triangle y_j = y'_j \oplus y''_j$, kde j reprezentuje j-tý bit diference
- Dvojici $(\triangle X, \triangle Y)$ nazýváme **rozdíl** nebo **diferenciál**

[11] Slovem rozdíly chápejme bitový XOR

- **Diferenciální charakteristikou** rozumíme posloupnost diferencí takových, že výstupní diference jedné rundy se rovná vstupní diferenci následující rundy. V literatuře bývá také nazývána jako trasy (*traces*).

3.2 Struktura diferenciální kryptoanalýzy

Schéma DK by se dalo rozdělit do dvou pomyslných podrutin. První z nich je analýza chování šifry a nalezení vhodných diferenčních charakteristik. Druhou z nich je provedení útoku a získání nějaké strategické informace z dvojic otevřených a šifrových textů. Příkladem takové informace může být v nejlepším případě část klíče použitého k šifrování zpráv. Za úspěch se však považuje i rozeznání druhu použité šifry od náhodných dat[12]. My se budeme zaměřovat na útok, který se snaží zjistit část použitého šifrovacího klíče.

Tvorba diferenčních charakteristik

Běžně se postupuje tak, že se kryptosystém rozdělí na jednotlivé atomické funkcionality. Tyto funkcionality se pak podrobí samostatné analýze chování vzhledem ke všem možným diferencím. Například u Baby Rijndaelu budeme zkoumat, jak se chovají operace *SubBytes*, *ShiftRows*, *MixColumns* a *AddRoundKey* vzhledem ke všem možným vstupním diferencím. Cílem této analýzy je stanovit, které vstupní diference mají na výstupu zkoumané operace v pravděpodobnosti nerovnoměrné rozdělení. Výstupem tohoto zkoumání může být tzv. *diferenční tabulka*, která obsahuje všechny naměřené pravděpodobnostní rozdělení. Konkrétní příklad analýzy a diferenční tabulky je uveden při popisu analýzy operace *SubBytes* 4.1.1. Z výsledků zkoumání pak přistupujeme ke konstrukci diferenčních charakteristik, které obsahují více na sebe navazujících operací. Někdy to mohou být charakteristiky odpovídající podmnožině zkoumaných operací pro zjednodušení konstrukce diferenčních charakteristik celé šifry. My si takové zjednodušení ukážeme na příkladě sjednocení operací *SubBytes* a *MixColumns*[13]. Pokud se nenabízejí další možná zjednodušení, pak se přistoupí ke konstrukci diferenčních charakteristik pro celou šifru. Ty nám pak udávají, jaké vstupní diference volit pro reálný útok a které jim odpovídající výstupní diference máme očekávat v předposlední[14] rundě šifry. Další kladenou podmínkou na charakteristiku je, aby její výstupní diference měla co nejvíce nulových atomických podbloků[15]. Důvody všech kla-

[12]Tento typ útoku je nazýván *Distinguishing attack*. Pokud by byl útok tohoto typu úspěšný, pak z otevřených a šifrových textů přece jen uniká jistá nepatrná informace, která umožňuje kryptoanalytikům se později zaměřit jen na konkrétně zjištěný kryptosystém.

[13]Sjednocení *SubBytes* a *MixColumns* s sebou nese ještě další úkol, protože tyto operace na sebe přímo nenavazují. Je potřeba dokázat, že je možné ekvivalentně vyměnit pořadí operace *ShiftRows* s operací *SubBytes*.

[14]Pro obecný útok může být zvolena výstupní diference na jiné hladině šifry (viz. Biclique attack).

[15]Například u Rijndaelu je to Byte, u Baby Rijndaelu 4 bity.

dených podmínek si vyvětlíme v následující části. Důležité je si uvědomit, že popisované trasy nedemonstrují deterministický průchod otevřeného textu šifrou, ale pouze její statisticky pravděpodobnější schéma. Příklad vizuální reprezentace takové trasy pro šifru Baby Rijndael můžeme pozorovat na obrázku 3.1.

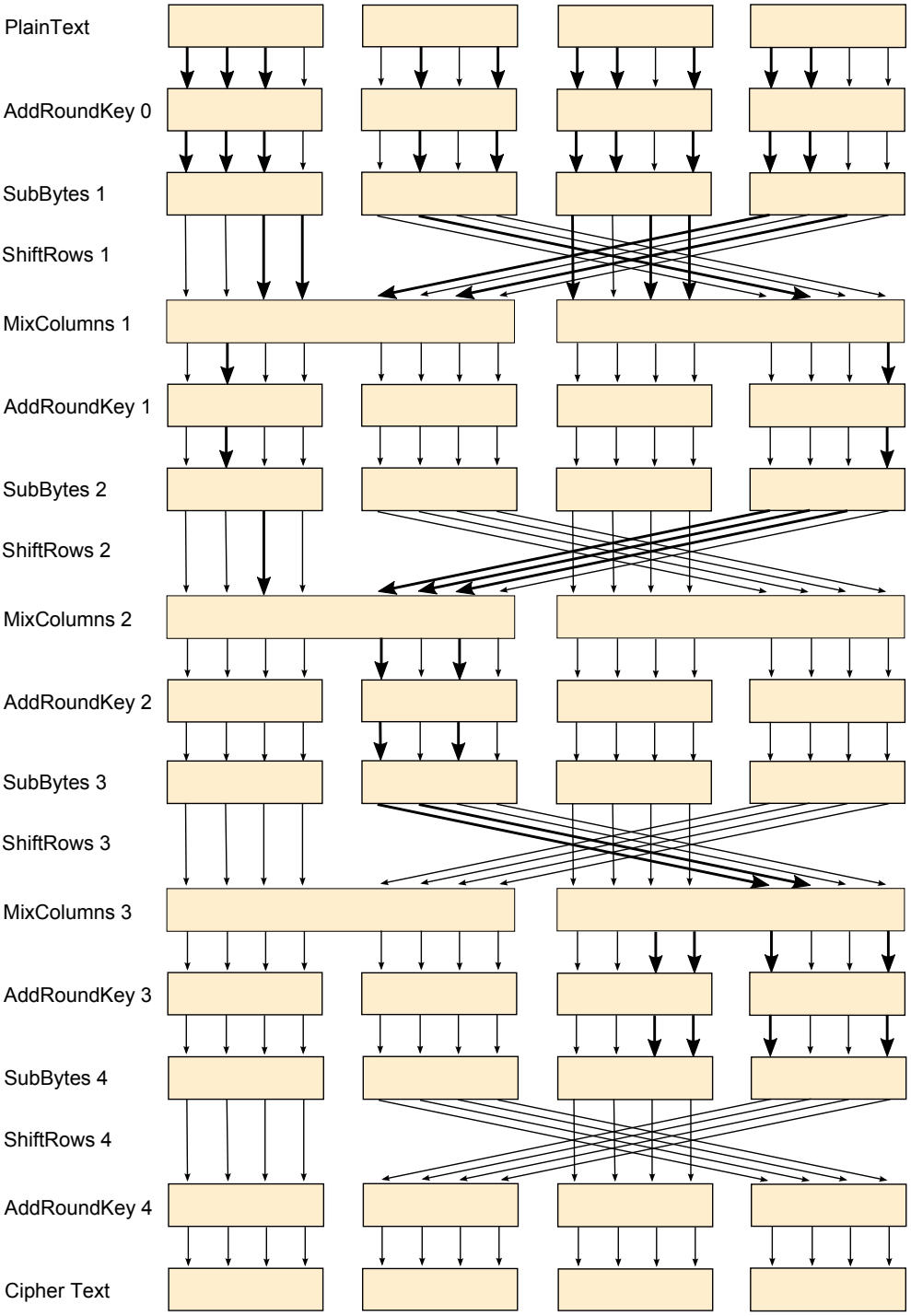
Útok a extrakce klíče

V předchozí části jsme si zmínili, že diferenciální charakteristika šifry udává svoji výstupní diferenci v předposlední rundě a samotná výstupní diference by měla mít co nejméně aktivních atomických podbloků.

Mějme tedy diferenciální charakteristiku se vstupní diferencí $\triangle X$ a výstupní diferencí $\triangle Y$. Dále mějme množinu dvojic otevřených textů a jim odpovídající dvojice šifrových textů, kde rozdíly dvojic otevřených textů jsou rovny $\triangle X$. Samotné otevřené texty jsou pro nás v následujících operacích irelevantní. Uvažovanou množinu dvojic OT a ŠT budeme v matematických zápisech zobrazovat jako množinu dvojic ŠT. Pro přehlednost si uveďme matematický zápis.

$$
\begin{aligned}
c_{1,i} &= \mathrm{encrypt}_{key}(p_{1,i}) \\
c_{2,i} &= \mathrm{encrypt}_{key}(p_{2,i}) \\
p_{1,i} \oplus p_{2,i} &= \triangle X \qquad \text{kde} \quad i \in \{0, \ldots, n\} \qquad (3.1) \\
c\_diff &= \{\{c_{1,i}, c_{2,i}\}, i \in \{0, \ldots, n\}\} \qquad (3.2)
\end{aligned}
$$

Dále víme, že na $\triangle Y$ je kladená podmínka co nejmenšího počtu aktivních atomických podbloků. Tento fakt nám umožňuje se zaměřit na prohledávání pouze jisté části klíče, používaného k šifrování OT. Pozice aktivních atomických podbloků v $\triangle Y$ přesně určují pozice částí klíčů, které se budeme snažit uhádnout/zjistit/vydedukovat. V rámci pozic těchto částí klíčů si vygenerujeme všechny možné klíče. Tuto množinu si označme K. Počet těchto klíčů je roven 2^x, kde x je počet bitů části klíče, který chceme extrahovat. Uvažovanou množinu klíčů pak postupně aplikujeme na množinu dvojic šifrových textů c_diff a sestavíme si **žebříček kandidátních klíčů**, který si označíme R. Ten udává, který klíč z K je nejlepším kandidátem skutečného klíče použitého při šifrování nám známých dvojic OT a ŠT. V naší implementaci sestavování žebříčku kandidátů klíčů budeme pracovat s několika obměnami, nicméně koncept jeho sestavování bude vždy stejný. Představme si tedy koncept sestavování R. Mějme libovolný klíč $k_\alpha \in K$ a libovolnou dvojici šifrových textů $c_{1,\beta}$ a $c_{2,\beta}$. Na oba šifrové texty samostatně aplikujeme jednu rundu dešifrovacího algoritmu. Tato operace pracuje s konkrétními hodnotami šifrových textů a části klíče. Na pozicích aktivních atomických podbloků Y tak můžeme porovnat předpovídanou diferenci s diferencí reálných šifrových textů, do kterých vstoupila hádaná část klíče k_α. Pokud byl hádaný klíč k_α pravým klíčem,



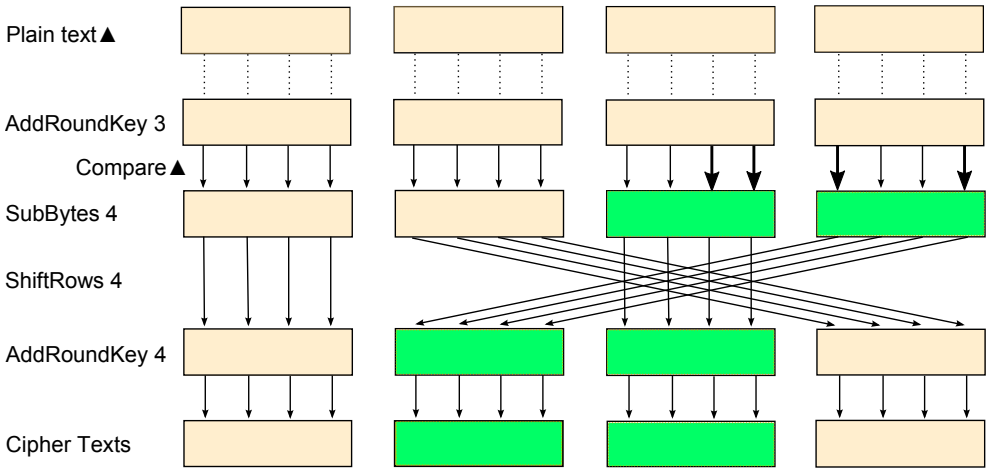
Obrázek 3.1: Jedna z diferenciálních charakteristik Baby Rijndaelu

pak logicky diference jednorundového dešifrování ŠT by měla statisticky významně odpovídat diferenci Y. Pokud tedy nastane rovnost $Y = c'_{1,\beta} \oplus c'_{2,\beta}$, pak si zaznamenáme tento fakt v žebříčku R u klíče k_α. Takové zaznamenání si můžeme představit tak, že klíči k_α zvýšíme *skóre*, které popisuje jeho kvalitu a na základě kterého může být prováděno srovnání s ostatními kandidátními klíči. Tento obecný algoritmus můžeme přepsat pro Baby Rijndael do následujícího pseudokódu:

Algorithm 1 Algoritmus sestavování žebříčku podklíčů – pseudokód

Require: c_diff $= \{[c_{1,i}, c_{2,i}], i \in \{0, \ldots, n\}\}$, outputDiffCharacteristic
1: K$[2^x] = [k_1, k_2, \ldots, k_{2^x}]$
2: RANKING$[2^x] = [0, 0, \ldots, 0]$
3: **for** $k_\alpha \in$ K **do**
4: **for** $[c_{1,\beta}, c_{2,\beta}] \in$ c_diff **do**
5: $c'_{1,\beta} = SubBytes^{-1}(ShiftRows^{-1}(k_\alpha \oplus c_{1,\beta}))$
6: $c'_{2,\beta} = SubBytes^{-1}(ShiftRows^{-1}(k_\alpha \oplus c_{2,\beta}))$
7: **end for**
8: **if** $c'_{1,\beta} \oplus c'_{2,\beta} =$ outputDiffCharacteristic **then**
9: RANKING$[\alpha]$ += 1
10: **end if**
11: **end for**

Důvod volby předposlední rundy pro konstrukci diferenciální charakteristiky je úzce spjat s technikou extrakce klíče. Speciálně v Rijndaelu a Baby Rijndaelu je vynechaná operace *MixColumns*, která by během extrakce klíče (dešifrování jedné rundy) zvětšila velikost atomických podbloků. U Baby Rijndaelu bychom se museli zaměřovat pouze na větve, které využívají společnou operaci *MixColumns*.



Obrázek 3.2: Ilustrace zpětného dešifrování během extrakce klíče

Kapitola 4

Konstrukce diferenciálních charakteristik

Tato kapitola úzce navazuje na předchozí kapitolu a zaměříme se v ní na detailní analýzu operací šifry Baby Rijndael. Dále si ukážeme způsoby, jak sestavovat diferenciální charakteristiky. Důležitým kritériem algoritmů pro sestavování charakteristik je také jejich pravděpodobnost, na kterou se také zaměříme. V závěru kapitoly si představíme způsoby konstrukce tzv. spojovaných a několikanásobných charakteristik.

4.1 Kryptoanalýza operací šifry Baby Rijndael

V úvodu provedeme kryptoanalýzu operací *SubBytes*, *ShiftRows*, *MixColumns* a *AddRoundKey*. Záhy uvidíme, že některé operace můžeme ignorovat a s některými můžeme i pozičně manipulovat.

4.1.1 Kryptoanalýza SubBytes

V sekci 2.2 jsme si představili operaci *SubBytes* a víme, že tato operace pracuje se 4-bitovými vstupy a výstupy. Analýzu provedeme tak, že aplikujeme operaci na všechny kombinace 4-bitových dvojic a tím získáme odpovídající výstupní dvojice. Takto získaná data zaneseme do tabulky následujícím způsobem:

- Vyjdeme z tabulky o rozměrech 16×16 naplněné nulami
- Rozdíl vstupní dvojice použijeme jako řádkový index do tabulky
- Rozdíl odpovídající výstupní dvojice použijeme jako sloupcový index do tabulky
- V tabulce na souřadnicích určených vstupním a výstupním rozdílem přičteme jedničku

Tabulku zkonstruovanou výše uvedeným algoritmem budeme nazývat **diferenční tabulka**. Diferenční tabulku Baby Rijndaelu můžeme vidět v tabulce 4.2.

Pro názornost konstrukce diferenční tabulky si uveďme příklad analýzy jednoho vstupu operace *SubBytes*. Nechť vstupní diference je rovna $\triangle$`0x9`. Tato diference může být výsledkem 16 kombinací dvojic vstupů. Pro každý jednotlivý vstup určíme jeho funkční obraz. Z jednotlivých odpovídajících dvojic obrazů vypočítáme jejich diferenci. Z počtu zastoupení jednotlivých výstupních diferencí se pak dá vypočítat s jakou pravděpodobností se vstupní diference $\triangle$`0x9` zobrazí na danou konkrétní diferenci. Uveďme si popisovaný příklad v přehledné tabulce 4.1. Existuje 16 možných výstupních diferencí a například výstupní diference $\triangle$`0xd` se vyskytuje v tabulce přesně 4-krát. Pravděpodobnost této výstupní diference je tedy $\frac{4}{16}$. Některé z výstupních diferencí si obarvíme v tabulce 4.1 a stejnou barvou obarvíme odpovídající záznamy v tabulce 4.2. Z tabulky 4.1 si můžeme také povšimnout, že pro zkoumanou vstupní diferenci $\triangle$`0x9` ve výstupních diferencích nejsou $\triangle$`0x0`, $\triangle$`0x2`, $\triangle$`0x3`, $\triangle$`0x4`, $\triangle$`0x6`, $\triangle$`0x7`, $\triangle$`0x8`, $\triangle$`0xa`, $\triangle$`0xb`. Tyto diference nikdy nenastanou a jejich pravděpodobnost je tedy rovna 0.

x_1	x_2	$SubBytes(x_1)$	$SubBytes(x_2)$	$SubBytes(x_1) \oplus SubBytes(x_2)$
0000	1001	1010	0111	1101 = $\triangle$0xd
0001	1000	0100	0101	0001 = $\triangle$0x1
0010	1011	0011	1111	1100 = $\triangle$0xc
0011	1010	1011	0110	1101 = $\triangle$0xd
0100	1101	1000	0001	1001 = $\triangle$0x9
0101	1100	1110	0000	1110 = $\triangle$0xe
0110	1111	0010	1101	1111 = $\triangle$0xf
0111	1110	1100	1001	0101 = $\triangle$0x5
1000	0001	0101	0100	0001 = $\triangle$0x1
1001	0000	0111	1010	1101 = $\triangle$0xd
1010	0011	0110	1011	1101 = $\triangle$0xd
1011	0010	1111	0011	1100 = $\triangle$0xc
1100	0101	0000	1110	1110 = $\triangle$0xe
1101	0100	0001	1000	1001 = $\triangle$0x9
1110	0111	1001	1100	0101 = $\triangle$0x5
1111	0110	1101	0010	1111 = $\triangle$0xf

Tabulka 4.1: Tvorba diferenřní tabulky *SubBytes* pro vstupní diferenci $\triangle$`0x9`

V diferenční tabulce 4.2 si můžeme povšimnout několik zajímavých vlastností. Předně všechny vyskytující se hodnoty jsou sudé. Tuto vlastnost lze vysvětlit komutativitou operace *xor*.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**a**	**b**	**c**	**d**	**e**	**f**
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	2	2	0	2	0	2	0	2	2	0	0	0	0	4	0
2	0	0	2	2	0	0	0	0	2	4	2	0	2	0	0	2
3	0	4	0	0	2	0	0	2	2	0	2	0	2	2	0	0
4	0	2	4	0	0	2	2	2	0	0	2	0	0	0	0	2
5	0	0	0	0	4	0	2	2	0	2	0	2	2	0	0	2
6	0	0	0	0	0	2	2	0	4	0	2	2	2	0	2	0
7	0	0	0	2	0	0	4	2	2	0	0	0	0	2	2	2
8	0	2	0	2	2	2	0	0	2	0	0	2	0	0	0	4
9	0	2	0	0	0	2	0	0	0	2	0	0	2	4	2	2
a	0	2	2	2	0	0	2	0	0	0	0	2	4	2	0	0
b	0	0	2	2	2	4	0	2	0	0	0	0	2	0	2	0
c	0	0	0	2	2	2	2	0	0	2	4	0	0	2	0	0
d	0	0	2	0	2	0	0	0	0	0	2	4	0	2	2	2
e	0	2	0	4	0	0	0	2	0	2	2	2	0	0	2	0
f	0	0	2	0	0	2	0	4	2	2	0	2	0	2	0	0

Tabulka 4.2: Diferenční tabulka *SubBytes*

$$a \oplus b = b \oplus a \tag{4.1}$$

$$SubBytes(a) \oplus SubBytes(b) = SubBytes(b) \oplus SubBytes(a) \tag{4.2}$$

Další vlastností diferenční tabulky (nejen) BabyRijndaelu je, že součet libovolného řádku a sloupce dává stejnou hodnotu 2^4. Pro tuto vlastnost si uvedeme krátký důkaz. Popis vlastnosti a důkaz si pro přehlednost rozdělíme zvlášť pro řádky a zvlášť pro sloupce.

Tvrzení 1. Mějme vstupní diferenci $\triangle X$ s bitovou délkou m. Pak součet hodnot v diferenční tabulce v řádku s indexem $\triangle X$ je roven 2^m.

Důkaz. Zadanou diferenci $\triangle X$ si rozepišme do následujících dvou ekvivalentních tvarů

$$\begin{aligned} \triangle X &= X_i \oplus X_j \qquad \text{kde} \quad X_i \text{ a } X_j \text{ jsou hodnoty s bitovou délkou } m \\ X_j &= X_i \oplus \triangle X \qquad \text{kde} \quad i, j \in \{0, ..., 2^m - 1\} \end{aligned} \tag{4.3}$$

Z rovnice 4.3 si můžeme všimnout, že pro zadanou diferenci a libovolnou hodnotu X_i je hodnota X_j určena jednoznačně. Počet všech možných rovnic pro zadanou diferenci $\triangle X$ je tedy 2^m. Z toho vyplývá, že i počet jedničkových

příspěvků do patřičného řádku diferenční tabulky je 2^m a stejný je také součet hodnot v $\triangle X$-tém řádku uvažované tabulky. □

Tvrzení 2. Mějme výstupní diferenci $\triangle Y$ s bitovou délkou m a nechť zobrazení reprezentující S-box je bijektivní. Pak součet hodnot v diferenční tabulce ve sloupci s indexem $\triangle Y$ je roven 2^m.

Důkaz. Zadanou diferenci $\triangle Y$ si rozepišme do následujících dvou ekvivalentních tvarů

$$\begin{aligned} \triangle Y &= Y_i \oplus Y_j \qquad \text{kde} \quad Y_i \text{ a } Y_j \text{ jsou hodnoty s bitovou délkou } m \\ Y_j &= Y_i \oplus \triangle Y \qquad \text{kde } i, j \in \{0, ..., 2^m - 1\} \end{aligned} \tag{4.4}$$

Z rovnice 4.4 si můžeme všimnout, že pro zadanou diferenci a libovolnou hodnotu Y_i je hodnota Y_j určena jednoznačně. Počet všech možných rovnic pro zadanou diferenci $\triangle Y$ je tedy 2^m. Díky předpokladu invertovatelnosti víme, že pro každou uspořádanou dvojici $[Y_i, Y_j]$ existuje právě jedna dvojice vstupů. Z toho vyplývá, že počet jedničkových příspěvků do patřičného sloupce diferenční tabulky je stejný, jako počet všech možných rovnic 4.4. Tedy také součet hodnot v $\triangle Y$-tém sloupci uvažované tabulky je roven 2^m. □

Předpoklad bijektivního zobrazení funkce *SubBytes* je zřejmý z překladové tabulky 2.1 a z faktu, že jedno z návrhových kritérií uvažované funkce byla invertovatelnost. Obě výše uvedená tvrzení platí pro šifru Baby Rijndael.

4.1.2 Kryptoanalýza ShiftRows

Analýza této funkce z pohledu diferencí je triviální. *ShiftRows* prohodí vždy stejné čtveřice bitů.

$$ShiftRows([a, b, c, d]) = [a, d, c, b] \tag{4.5}$$

$$\begin{aligned} ShiftRows([a, b, c, d]) \oplus ShiftRows([a', b', c', d']) &= \\ = [a, d, c, b] \oplus [a', d', c', b'] &= \\ = [\triangle a, \triangle d, \triangle c, \triangle b] & \end{aligned} \tag{4.6}$$

$$\begin{aligned} ShiftRows([a, b, c, d] \oplus [a', b', c', d']) &= \\ = ShiftRows([\triangle a, \triangle b, \triangle c, \triangle d]) &= \\ = [\triangle a, \triangle d, \triangle c, \triangle b] & \end{aligned} \tag{4.7}$$

Je tedy lineární.

4.1.3 Kryptoanalýza MixColumns

Zde bychom mohli postupovat jako v analýze *SubBytes*. Pokud si ale vzpomeneme, v sekci 2.4 jsme si uvedli, že operace *MixColumns* je realizována maticovým násobením, kde příslušná matice má svou inverzi. Tato operace je tedy lineárním a bijektivním zobrazením. Definovaná matice je aplikovaná na celý 16tibitový stav šifry. Obecný zápis operace by se dal vyjádřit následovně:

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}_2 \begin{bmatrix} a_0 & a_8 \\ a_1 & a_9 \\ a_2 & a_{10} \\ a_3 & a_{11} \\ a_4 & a_{12} \\ a_5 & a_{13} \\ a_6 & a_{14} \\ a_7 & a_{15} \end{bmatrix}_2 \quad \text{kde } a_i \in \{0,1\}, i \in \{0,\ldots,15\}$$

Tato operace se však dá logicky nezávisle rozdělit na dva 8-bitové řetězce, kde čtvercová matice působí nejprve na první sloupec stavu a pak na druhý sloupec stavu. Analýzu můžeme tedy zaměřit na zkoumání operace na 8-bitových řetězcích. Z pohledu sestavování diferenční tabuky má délka zpracovávaného vstupu vliv na její rozměry. Ta bude mít pro 8-bitové řetězce rozměry 256 krát 256. Dále se pokusíme ukázat, že každý řádek a sloupec této tabulky obsahuje pouze jednu hodnotu 256. Tento fakt se pokusíme odvodit z matematického vyjádření uvažované operace[16]. Pro prokázání linearity operace *MixColumns* vůči diferenciálním charakteristikám musíme ukázat následující:

$$MixColumns(A) \oplus MixColumns(A') = MixColumns(A \oplus A')$$

$$MC(\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}_2) = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}_2 \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}_2 = \begin{bmatrix} a_0 \oplus a_2 \oplus b_2 \oplus b_3 \\ a_0 \oplus a_1 \oplus a_3 \oplus b_3 \\ a_0 \oplus a_1 \oplus a_2 \oplus b_0 \\ a_1 \oplus a_3 \oplus b_1 \oplus b_2 \oplus b_3 \\ a_2 \oplus a_3 \oplus b_0 \oplus b_2 \\ a_3 \oplus b_0 \oplus b_1 \oplus b_3 \\ a_0 \oplus b_0 \oplus b_1 \oplus b_2 \\ a_1 \oplus a_2 \oplus a_3 \oplus b_1 \oplus b_3 \end{bmatrix}_2$$

[16]Pro kompaktnější zápis budeme v jistých částech substituovat *MixColumns* za *MC*

$$MC(\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}) \oplus MC(\begin{bmatrix} a'_0 \\ a'_1 \\ a'_2 \\ a'_3 \\ b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \end{bmatrix}) = \begin{bmatrix} a_0 \oplus a_2 \oplus b_2 \oplus b_3 \oplus a'_0 \oplus a'_2 \oplus b'_2 \oplus b'_3 \\ a_0 \oplus a_1 \oplus a_3 \oplus b_3 \oplus a'_0 \oplus a'_1 \oplus a'_3 \oplus b'_3 \\ a_0 \oplus a_1 \oplus a_2 \oplus b_0 \oplus a'_0 \oplus a'_1 \oplus a'_2 \oplus b'_0 \\ a_1 \oplus a_3 \oplus b_1 \oplus b_2 \oplus b_3 \oplus a'_1 \oplus a'_3 \oplus b'_1 \oplus b'_2 \oplus b'_3 \\ a_2 \oplus a_3 \oplus b_0 \oplus b_2 \oplus a'_2 \oplus a'_3 \oplus b'_0 \oplus b'_2 \\ a_3 \oplus b_0 \oplus b_1 \oplus b_3 \oplus a'_3 \oplus b'_0 \oplus b'_1 \oplus b'_3 \\ a_0 \oplus b_0 \oplus b_1 \oplus b_2 \oplus a'_0 \oplus b'_0 \oplus b'_1 \oplus b'_2 \\ a_1 \oplus a_2 \oplus a_3 \oplus b_1 \oplus b_3 \oplus a'_1 \oplus a'_2 \oplus a'_3 \oplus b'_1 \oplus b'_3 \end{bmatrix}$$

$$MC(\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} \oplus \begin{bmatrix} a'_0 \\ a'_1 \\ a'_2 \\ a'_3 \\ b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \end{bmatrix}) = MC(\begin{bmatrix} \triangle a_0 \\ \triangle a_1 \\ \triangle a_2 \\ \triangle a_3 \\ \triangle b_0 \\ \triangle b_1 \\ \triangle b_2 \\ \triangle b_3 \end{bmatrix}) = \begin{bmatrix} \triangle a_0 \oplus \triangle a_2 \oplus \triangle b_2 \oplus \triangle b_3 \\ \triangle a_0 \oplus \triangle a_1 \oplus \triangle a_3 \oplus \triangle b_3 \\ \triangle a_0 \oplus \triangle a_1 \oplus \triangle a_2 \oplus \triangle b_0 \\ \triangle a_1 \oplus \triangle a_3 \oplus \triangle b_1 \oplus \triangle b_2 \oplus \triangle b_3 \\ \triangle a_2 \oplus \triangle a_3 \oplus \triangle b_0 \oplus \triangle b_2 \\ \triangle a_3 \oplus \triangle b_0 \oplus \triangle b_1 \oplus \triangle b_3 \\ \triangle a_0 \oplus \triangle b_0 \oplus \triangle b_1 \oplus \triangle b_2 \\ \triangle a_1 \oplus \triangle a_2 \oplus \triangle a_3 \oplus \triangle b_1 \oplus \triangle b_3 \end{bmatrix} =$$

$$= \begin{bmatrix} a_0 \oplus a'_0 \oplus a_2 \oplus a'_2 \oplus b_2 \oplus b'_2 \oplus b_3 \oplus b'_3 \\ a_0 \oplus a'_0 \oplus a_1 \oplus a'_1 \oplus a_3 \oplus a'_3 \oplus b_3 \oplus b'_3 \\ a_0 \oplus a'_0 \oplus a_1 \oplus a'_1 \oplus a_2 \oplus a'_2 \oplus b_0 \oplus b'_0 \\ a_1 \oplus a'_1 \oplus a_3 \oplus a'_3 \oplus b_1 \oplus b'_1 \oplus b_2 \oplus b'_2 \oplus b_3 \oplus b'_3 \\ a_2 \oplus a'_2 \oplus a_3 \oplus a'_3 \oplus b_0 \oplus b'_0 \oplus b_2 \oplus b'_2 \\ a_3 \oplus a'_3 \oplus b_0 \oplus b'_0 \oplus b_1 \oplus b'_1 \oplus b_3 \oplus b'_3 \\ a_0 \oplus a'_0 \oplus b_0 \oplus b'_0 \oplus b_1 \oplus b'_1 \oplus b_2 \oplus b'_2 \\ a_1 \oplus a'_1 \oplus a_2 \oplus a'_2 \oplus a_3 \oplus a'_3 \oplus b_1 \oplus b'_1 \oplus b_3 \oplus b'_3 \end{bmatrix}$$

MixColumns se tedy chová k 8-bitovým diferencím lineárně. Je tedy jedno, zda provedeme maticové násobení dvou 8-bitových vstupů skládajících se na uvažovanou diferenci, nebo provedeme maticové násobení na 8-bitovém řetězci diference dvou stavů. Zafixujme si konkrétní diferenci $\triangle x = x \oplus x'$. Výsledek operace *MixColumns* nezávisí na jednotlivých stavech x a x', které tvoří zafixovanou diferenci. Pro $\triangle x$ opět existuje $2^8 = 256$ dvojic x a x', pro které platí $\triangle x = x \oplus x'$. Tedy každý řádek diferenční tabulky operace *MixColumns* má pouze jednu nenulovou hodnotu 256.

4.1.4 Kryptoanalýza AddRoundKey

Analýza přičtení klíče demonstruje zajímavou vlastnost diferenciální analýzy. Mějme hodnoty $x \oplus x' = \triangle x$. Pak pro funkci *AddRoundKey* platí následující vztah

$$AddRoundKey(x) \oplus AddRoundKey(x') = x \oplus k \oplus x' \oplus k = x \oplus x' = \triangle x$$

Operace *AddRoundKey* tedy diference nemění a můžeme ji z návrhu konstrukce diferenciálních charakteristik vyjmout.

4.2 Výpočet pravděpodobnosti diferenciální charakteristiky

Doposud jsme provedli analýzu všech elementárních operací zkoumané šifry. Jediná nelineární operace z pohledu diferencí je *SubBytes*, a ta jako jediná bude přispívat k výpočtu pravděpodobnosti složitějších diferenčních charakteristik. Na jednotlivé S-boxy budeme nahlížet jako na nezávislé operace, tudíž jejich jednotlivé pravděpodobnosti budeme násobit. Pokud diferenční charakteristika bude obsahovat operaci *SubBytes*, která bude mít jinou než nulovou diferenci, budeme říkat, že daný S-box je aktivní. Pravděpodobnost diferenciální charakteristiky je tedy vynásobení pravděpodobností všech aktivních S-boxů. Pravděpodobnosti jednotlivých vstupně-výstupních hodnot pro operaci *SubBytes* je vyjádřena v diferenční tabulce 4.2. Hodnoty této tabulky vyjadřují čitatel zlomku pravděpodobnosti, kde jmenovatel je roven 16. Za povšimnutí stojí zmínit, že pro *neaktivní* S-box je jeho pravděpodobnost vyčíslena jako $16/16 = 1$. Pro názornost si vypočteme pravděpodobnost diferenční charakteristiky z obrázku 3.1 z minulé kapitoly. V této charakteristice máme tyto aktivní S-boxy s následujícími pravděpodobnostmi:

△ vstup → △ výstup	pravděpodobnost
△`e` → △`3`	4/16
△`5` → △`4`	4/16
△`d` → △`b`	4/16
△`c` → △`a`	4/16
△`4` → △`2`	4/16
△`1` → △`e`	4/16
△`a` → △`c`	4/16

Tabulka 4.3: Aktivní sboxy 3.1 včetně jejich pravděpodobností

Pravděpodobnost této charakteristiky je tedy

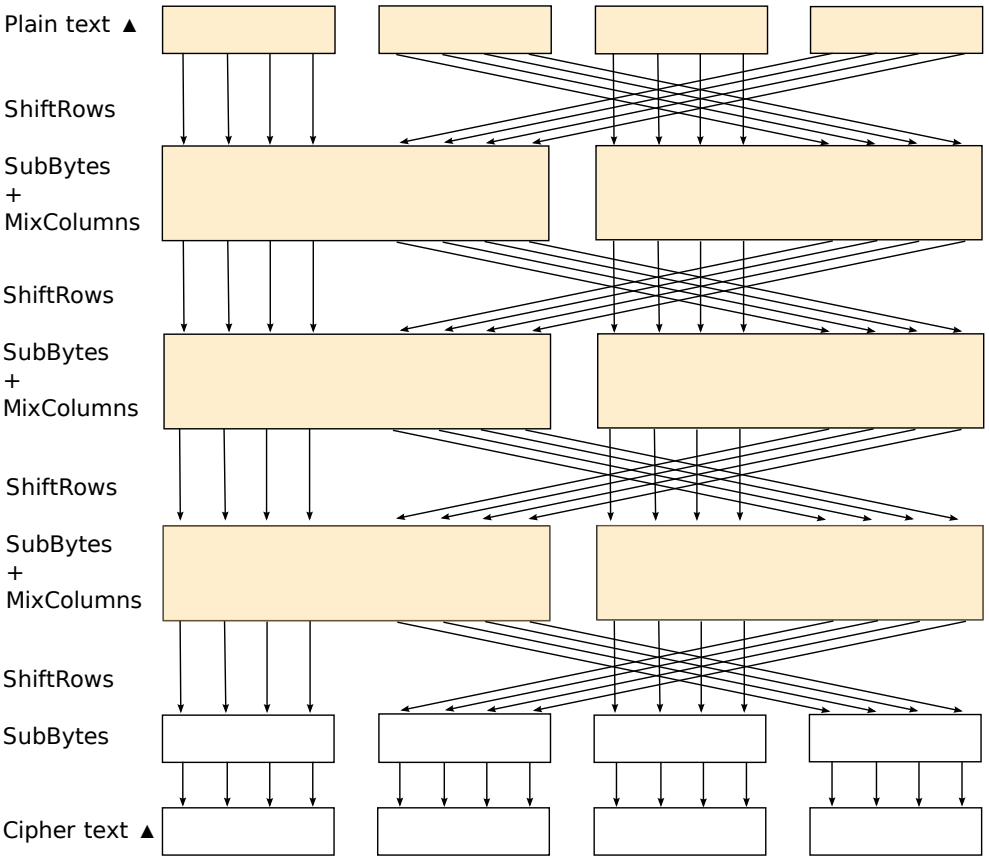
$$(\frac{4}{16})^7 = (\frac{1}{4})^7 = (2^{-2})^7 = 2^{-14}$$

4.3 Ekvivalentní úpravy šifry

Z pohledu konstrukce diferenciálních charakteristik je výhodné si některé zkoumané operace zjednodušit. V naší zkoumané šifře se jeví jako dobrý nápad sjednocení operace *SubBytes* a *MixColumns*, které můžeme chápat jako jeden

velký S-box. Tento nápad popisuje Kokeš v [9]. V této substituci však existuje jeden problém. Operace *SubBytes* a *MixColumns* jsou odděleny operací *ShiftRows*. *SubBytes* a *ShiftRows* však shodně pracují se 4-bitovými celky, kde *SubBytes* nezávisí na pořadí celku a *ShiftRows* je odpovědná pouze za změnu jejich pořadí. Je tedy lhostejné, zda nejdříve provedeme operace *SubBytes* a následně provedeme záměnu pořadí, či naopak. Uvažované operace tedy můžeme prohodit.

Dalším zjednodušením je vyjmutí operace *AddRoundKey* ze struktury šifry. Ta totiž nemá na změnu charakteristiky vliv, protože dvojnásobné přičtení stejné hodnoty se vzájemně vyruší. Zjednodušenou strukturu šifry Baby Rijndael z pohledu konstrukce diferenciálních charakteristik můžeme pozorovat na obrázku 4.1.



Obrázek 4.1: Zjednodušená struktura Baby Rijndaelu z pohledu konstrukce diferenciálních charakteristik

4.4 Konstrukce diferenčních charakteristik šifry

Zatím jsme provedli analýzu všech elementárních operací šifry a zjednodušili si strukturu Baby Rijndaelu pro účely konstrukce diferenciálních charakteristik. Charakteristiky je možné konstruovat hned několika způsoby a našim cílem bude nalézt takovou metodu, která by byla dostatečně efektivní i pro větší variantu šifry.

4.4.1 Hrubá síla

První variantou bylo hledání diferenčních charakteristik hrubou silou. Nagenerování všech možných charakteristik včetně těch nejméně pravděpodobných by bylo časově velmi náročné. Proto jsem při implementaci přistoupil k následující optimalizaci:

- Pamatoval jsem si pouze takové výstupní diference, které měly aktivní pouze 1 nebo 2 čtveřice bitů
- V paměti jsem udržoval pouze diferenční charakteristiky s doposud nejlepší pravděpodobností

Konstruování diferenčních charakteristik jsem tak mohl ořezávat dle dosud nejlepších řešení.

4.4.2 Hledání diferenčních charakteristik inverzní úlohou

Přístup pomocí hrubé síly často naráží na problém prohledávání tras, které neodpovídají námi kladeným podmínkám. V našem případě se jedná o nejvýše 2 aktivní 4-bitové podbloky ve výstupní diferenci. Tento problém se dá vyřešit hledáním charakteristik pomocí řešení inverzní úlohy ke hrubé síle. To spočívá v tom, že si stanovíme všechny výstupní diference, které odpovídají našim požadavkům, a diferenciální charakteristiky konstruujeme pomocí inverzních operací *ShiftRows*, *SubBytes*, *MixColumns* nebo sjednocené *SubBytes+MixColumns*.

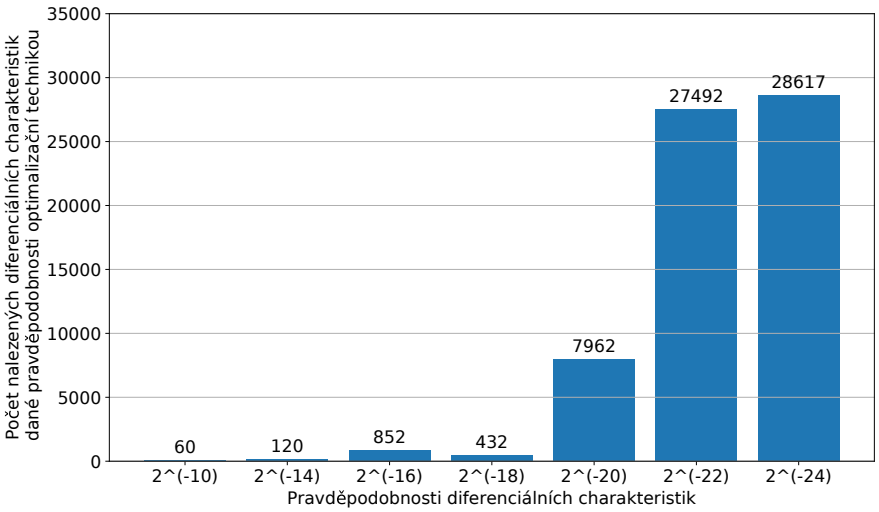
4.4.3 Optimalizační technika s více diferenčními tabulkami

Po úspěšném nalezení charakteristik s největšími pravděpodobnostmi jsem ověřil následující logickou hypotézu. Všechny průchody operací *SubBytes* byly pro nejlepší charakteristiky realizovány pouze s využitím hodnot 4 diferenční tabulky 4.2. Jinými slovy tyto charakteristiky nikde nevyužívají zhoršující strategii. S touto logikou jsem upravil algoritmus hledání charakteristik s použitím parametrizované diferenční tabulky S-boxu. Myšlenka spočívá v tom, že si připravíme několik diferenčních tabulek a v rámci prohledávání budeme povolovat parametrizovaný počet použití méně striktních diferenčních tabulek. V případě Baby Rijndaelu tedy pouze 2, protože v diferenční tabulce

máme pouze hodnoty 16, 4 a 2, kde 16 reprezentuje nulovou diferenci. Jedna tabulka obsahuje pouze hodnoty 16 a 4 a druhá je samotná originální tabulka. Další zajímavostí je fakt, že originální diferenční tabulka *SubBytes* obsahuje hodnotu 4 pro každou vstupní diferenci.

Tento postup nám může pomoci při hledání všech diferenčních charakteristik až do nějaké meze. Nutno ale zmínit, že u tohoto postupu není obecně zajištěno, zda nalezne diferenční charakteristiky šifry v sestupném pořadí dle pravděpodobnosti. Pravděpodobnost diferenční charakteristiky šifry snižuje mnohem více parametr počtu aktivovaných S-boxů. Hypoteticky se tedy může stát, že diferenční charakteristika s menším počtem aktivních S-boxů a několika zhoršujícími volbami má ve skutečnosti vyšší pravděpodobnost než charakteristika, která aktivuje víc S-boxů bez zhoršujících voleb. Tento fenomén se ale u Baby Rijndaelu v této podobě hledání diferenčních charakteristik neukázal.

Vyzkoušel jsem však popisovanou metodu s použitím pouze diferenční tabulky S-boxu s hodnotami 16 a 4 a zároveň jsem nekladl žádné podmínky na výstupní diference. Tímto pokusem se mi povedlo zjistit, že v Baby Rijndaelu existují diferenciální charakteristiky s pravděpodostmi 2^{-10}. V grafu 4.2 můžeme pozorovat jejich počty. Pro klasickou diferenciální kryptoanalýzu však klademe omezení, které jsme si výše popsali. Grafem si tak pouze ilustrujeme fakt, že ve zkoumané šifře existuje několik samostatných diferenčních charakteristik s vysokou pravděpodobností, které pro nás nejsou použitelné.



Obrázek 4.2: Počty diferenciálních charakteristik dle jejich pravděpodobností bez ohledu na omezení výstupních diferencí

Nejlepší diferenční charakteristiky s našimi omezeními dosahují pravděpodobnosti 2^{-14} a je jich přesně 60. Dají se rozdělit do dvou skupin, kde první

z nich má ve výstupní diferenci aktivní bity mezi 0. až 7. bitem[17]. Druhá skupina pak mezi 8. až 15. bitem. Zajímavé také je, že obě dvě skupiny koncových diferencí tvoří stejnou množinu hodnot. Tato skutečnost je ve skutečnosti způsobena symetrií struktury šifry. V příloze B.2 můžeme pozorovat tabulku se všemi popisovanými diferencemi. Zde si uvedeme jen několik z nich.

vstupní diference	koncová diference
e5dc → 4001 → 0a00 → 0039	
dce5 → 0140 → 000a → 3900	
d415 → 01a0 → 0030 → 005d	
15d4 → a001 → 3000 → 5d00	
e582 → 0f70 → 0080 → 0067	
82e5 → 700f → 8000 → 6700	
1ec4 → a004 → 0900 → 00ec	
c41e → 04a0 → 0009 → ec00	

Tabulka 4.4: Výběr několika významných diferenčních charakteristik

Záměrně jsem v ukázce uvedl korespondující diferenční charakteristiky z obou výše popsaných skupin. Na jejich příkladu si můžeme všimnout symetrie, která kopíruje operaci *ShiftRows*.

4.5 Metody spojování diferenčních charakteristik

Pravděpodobnosti nějaké konkrétní diferenční charakteristiky počítáme ze zřetězených elementárních diferenčních charakteristik jednotlivých rund. Důležité je si uvědomit, že při samotném útoku využíváme ze zvolené diferenční charakteristiky pouze informaci o vstupní diferenci a výstupní diferenci. Vstupní diference nám slouží k určení množiny správných dvojic otevřených textů. Výstupní diference nám pak slouží k extrakci správného klíče. Ukázalo se, že ve všech nalezených diferenčních charakteristikách existuje mnoho takových, které mají společnou vstupní a koncovou diferenci. Tento pohled například popisuje Knudsen v práci o *zkrácených diferenciálech* [16]. Autor ve svých slidech demonstruje konstrukci zkrácených diferenciálů v rámci šifry *CipherFOUR*. Důležitým aspektem je poznatek, že v konstruovaných diferenciálních charakteristikách nás nezajímají jednotlivé diference vnitřních rund.

4.5.1 Sjednocené diferenciální charakteristiky

Mějme následující diferenciální charakteristiku:

[17] Aktivní 4bitové skupiny výstupní diference budeme značit symbolem $\star$. 4bitové skupiny výstupní diference, které nejsou aktivní značíme symbolem 0. Aktivní bity mezi 0. a 7. bitem tedy značíme "$\star\star 00$".

$$\texttt{e5dc} \rightarrow \texttt{4001} \rightarrow \texttt{0a00} \rightarrow \texttt{0039}$$

K využití této charakteristiky v rámci extrakce klíče však využíváme pouze vstupní diferenci `e5dc` k volbě dvojic otevřených textů a výstupní diferenci `0039` k sestavování rankingu kandidátních klíčů. Naši diferenciální charakteristiku tak můžeme přepsat následovně:

$$\texttt{e5dc} \rightarrow \texttt{?} \rightarrow \texttt{?} \rightarrow \texttt{0039}$$

Všechny charakteristiky odpovídající výše uvedené šabloně jsou dle Knudsena v rámci diferenciální kryptoanalýzy na sobě nezávislé a pravděpodobnosti jednotlivých dif. charakteristik můžeme sčítat. V tabulce 4.5 můžeme vidět všechny existující charakteristiky odpovídající výše uvedené šabloně.

Diferenční charakteristika	$P(\triangle)$
e5dc → 4001 → 0a00 → 0039	2^{-14}
e5dc → 4001 → 0200 → 0039	2^{-17}
e5dc → 4006 → 0a00 → 0039	2^{-17}
e5dc → 4007 → 0a00 → 0039	2^{-17}
e5dc → b001 → 0a00 → 0039	2^{-17}
e5dc → d001 → 0a00 → 0039	2^{-17}
e5dc → f001 → 0a00 → 0039	2^{-17}
e5dc → 4006 → 0200 → 0039	2^{-18}
e5dc → b001 → 0200 → 0039	2^{-18}
e5dc → 4003 → 0200 → 0039	2^{-19}
e5dc → 4007 → 0200 → 0039	2^{-19}
e5dc → b006 → 0200 → 0039	2^{-19}
e5dc → f001 → 0200 → 0039	2^{-19}
e5dc → 4003 → 0500 → 0039	2^{-19}
e5dc → 4007 → 0500 → 0039	2^{-19}
e5dc → 4006 → 0600 → 0039	2^{-19}
e5dc → b001 → 0900 → 0039	2^{-19}
e5dc → b003 → 0200 → 0039	2^{-20}

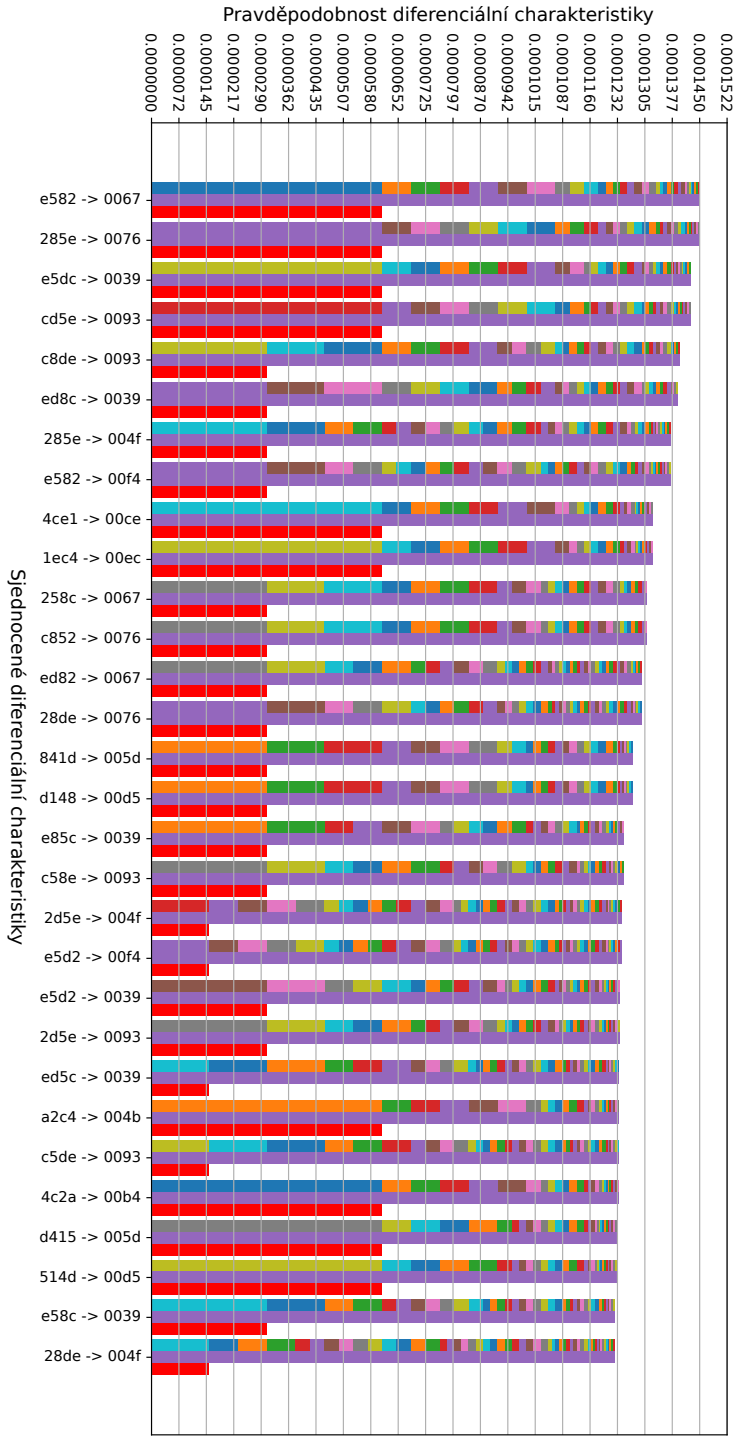
Diferenční charakteristika	$P(\triangle)$
e5dc → b007 → 0200 → 0039	2^{-20}
e5dc → f006 → 0200 → 0039	2^{-20}
e5dc → b006 → 0a00 → 0039	2^{-20}
e5dc → b007 → 0a00 → 0039	2^{-20}
e5dc → d006 → 0a00 → 0039	2^{-20}
e5dc → d007 → 0a00 → 0039	2^{-20}
e5dc → f006 → 0a00 → 0039	2^{-20}
e5dc → f007 → 0a00 → 0039	2^{-20}
e5dc → f003 → 0200 → 0039	2^{-21}
e5dc → f007 → 0200 → 0039	2^{-21}
e5dc → b003 → 0300 → 0039	2^{-21}
e5dc → b006 → 0300 → 0039	2^{-21}
e5dc → d003 → 0300 → 0039	2^{-21}
e5dc → d006 → 0300 → 0039	2^{-21}
e5dc → d006 → 0600 → 0039	2^{-21}
e5dc → f003 → 0b00 → 0039	2^{-21}
e5dc → f006 → 0b00 → 0039	2^{-21}

Tabulka 4.5: Diferenční charakteristiky pro 4-rundový Baby Rijndael se společnou počáteční a koncovou diferencí

$$P(\texttt{c8de} \rightarrow \texttt{?} \rightarrow \texttt{?} \rightarrow \texttt{0093}) =$$

$$= 2^{-14} + 6 \cdot 2^{-17} + 2 \cdot 2^{-18} + 8 \cdot 2^{-19} + 9 \cdot 2^{-20} + 9 \cdot 2^{-21} = \frac{299}{2^{21}}$$

Uvedený graf 4.3 demonstruje pravděpodobnostní přírůstek spojovaných diferenciálních charakteristik oproti těm s přesně definovanými vnitřními diferencemi. Každý sloupec grafu reprezentuje jednu diferenční charakteristiku. Levý sloupec charakteristiky reprezentuje seřazený seznam pravděpodobností jednotlivých diff. charakteristik, které mají společnou vstupní a koncovou diferenci. Prostřední sloupec reprezentuje pouze součet pravděpodobností jednotlivých diferenciálních charakteristik. Slouží hlavně jako vizuální oddělovač



Obrázek 4.3: Pravděpodobnosti sjednocených diferenciálních charakteristik

levého a pravého sloupce. Pravý sloupec představuje největší pravděpodobnost ze sjednocené charakteristiky. Tento pravý sloupec demonstruje další zajímavost sjednocených charakteristik. Pokud bychom se spoléhali na hledání plně definovaných charakteristik a jejich pravděpodobností, pak bychom některé charakteristiky mylně označovali za více pravděpodobné, nežli ve skutečnosti jsou. Příkladem mohou být

$$\texttt{c8de} \rightarrow \texttt{?} \rightarrow \texttt{?} \rightarrow \texttt{0093} \qquad (2^{-15} < \frac{299}{2^{21}} \approx 0.0001426 \approx 2^{-12.776})$$

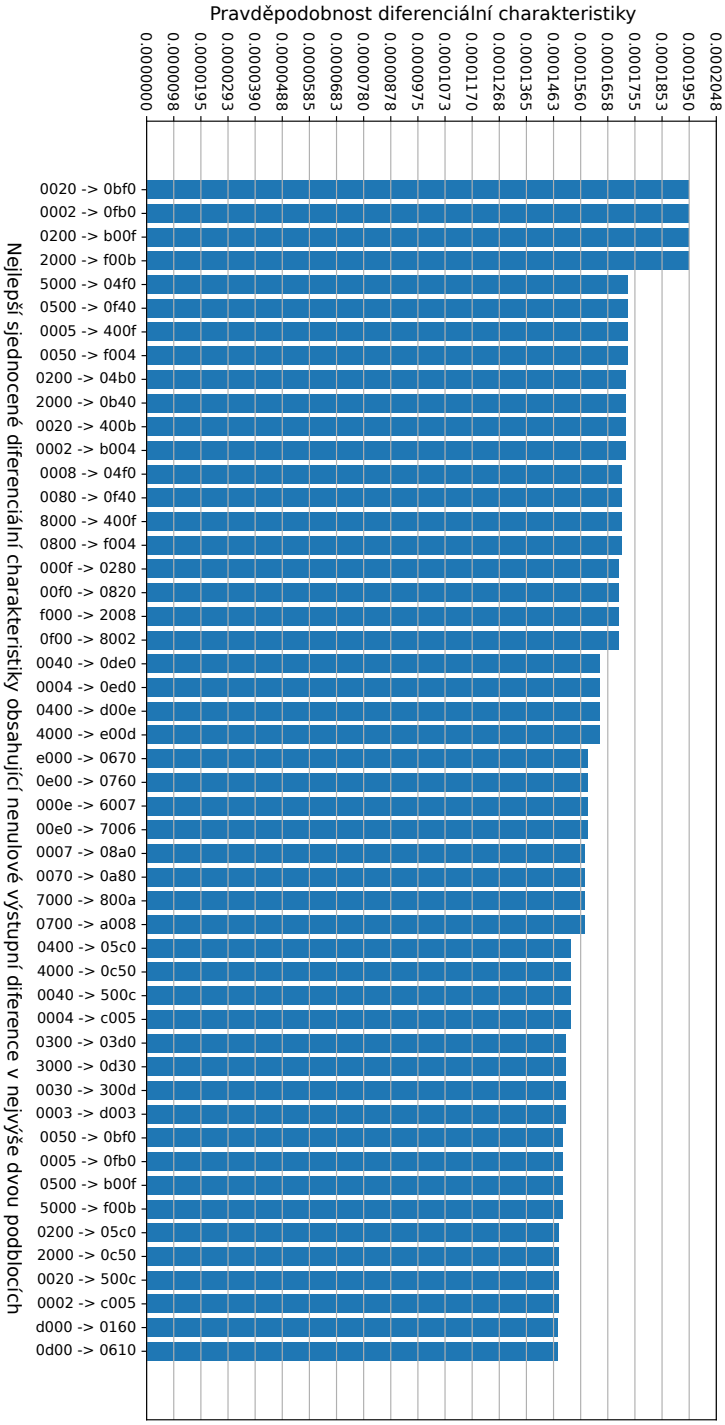
$$\texttt{514d} \rightarrow \texttt{?} \rightarrow \texttt{?} \rightarrow \texttt{00d5} \qquad (2^{-14} < \frac{258}{2^{21}} \approx 0,0001230 \approx 2^{-12,989})$$

Graf 4.3 demonstruje pouze diferenciální charakteristiky, které mají aktivní bity koncové diference na 8. až 15. pozici. Symetricky jsem našel i diferenciální charakteristiky, jejichž aktivní koncové diference jsou na pozicích 0 až 7. Pro nalezení pouze těchto charakteristik jsem byl motivován faktem, že významné charakteristiky nalezené hrubou silou měly koncové diference pouze z pozic typu "$00\star\star$"a "$\star\star 00$". Pro tyto druhy koncových diferencí bylo také zajímavé, že množství existujících diferenciálních charakteristik bylo o několik řádů menší než pro ostatní charakteristiky, které měly koncové diference na jiných místech. Nicméně bez prozkoumání všech diferenciálních charakteristik, které mají v koncových diferencích aktivní nejvýše 2 4-bitové podbloky, nemohu tvrdit, že nalezené charakteristiky z grafu 4.3 jsou ty nejlepší možné. Z výpočetního hlediska jsem pro překonání řádově většího počtu existujících diferenciálních charakteristik přistoupil ke kompromisu a sjednocoval jsem hledané charakteristiky a jejich pravděpodobnosti za běhu. Tím jsem ztratil informaci o jednotlivých charakteristikách (vnitřní aktivní bity rundy a jejich pravděpodobnosti), které přispívají na jednotlivé sjednocené charakteristiky. Výsledek je však překvapující a můžeme jej pozorovat na grafu 4.4. Nejlepší diferenciální charakteristiky dle našich omezení ve skutečnosti dosahují pravděpodobnosti $\approx 0.000195 \approx 2^{12.324}$.

4.5.2 Několikanásobné diferenciální charakteristiky

Následující způsob seskupování diferenciálních charakteristik by se dal chápat spíše jako speciální metoda útoku a extrakce klíče. Pro její použití je však nutná pečlivá příprava množin diferenciálních charakteristik. Podrobnější případy využití těchto charakteristik v rámci útoku budeme však diskutovat v následující kapitole.

Sjednocování diferenciálních charakteristik se společnou vstupní a výstupní diferencí je vcelku intuitivní. Z grafů 4.3 a 4.4 je patrné, že mezi nejlepšími charakteristikami se vyskytují takové, které mají shodnou vstupní diferenci a shodné 4-bitové pozice aktivních bitů ve výstupní diferenci. Napadlo mě využít této vlastnosti pro nový druh diferenciálních charakteristik. Důležité je si



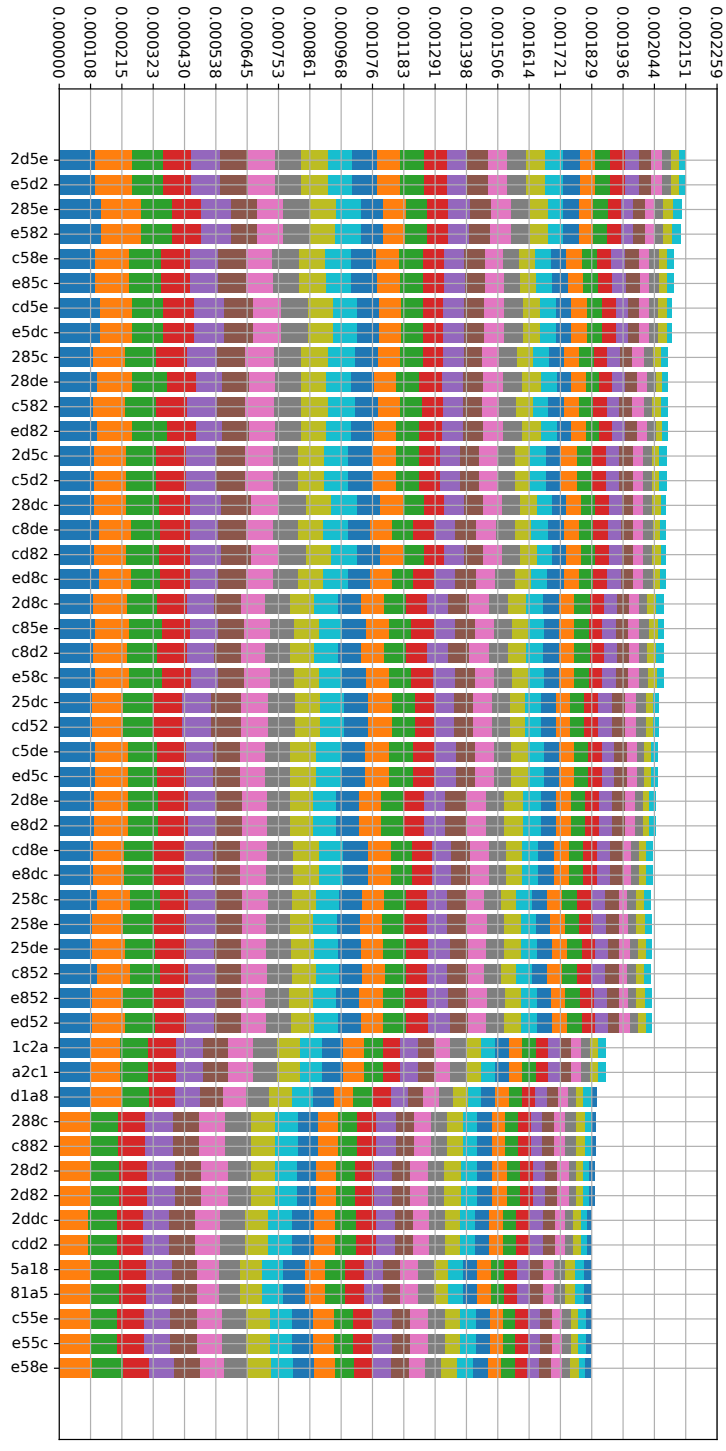
Obrázek 4.4: Pravděpodobnosti nejlepších sjednocených diferenciálních charakteristik

připomenout, jakou roli hrají při extrakci klíče vstupní a výstupní diference. Vstupní diference charakteristiky nám říkají, které dvojice otevřených textů chceme zašifrovat pro nás neznámým klíčem. Tím získáme množinu dvojic otevřených textů a jim odpovídající dvojice šifrových textů. Informaci o použitém šifrovacím klíči se dále snažíme zjistit pouze z dvojic šifrových textů. Mějme dvě následující obecné diferenciální charakteristiky, které odpovídají popisované situaci:

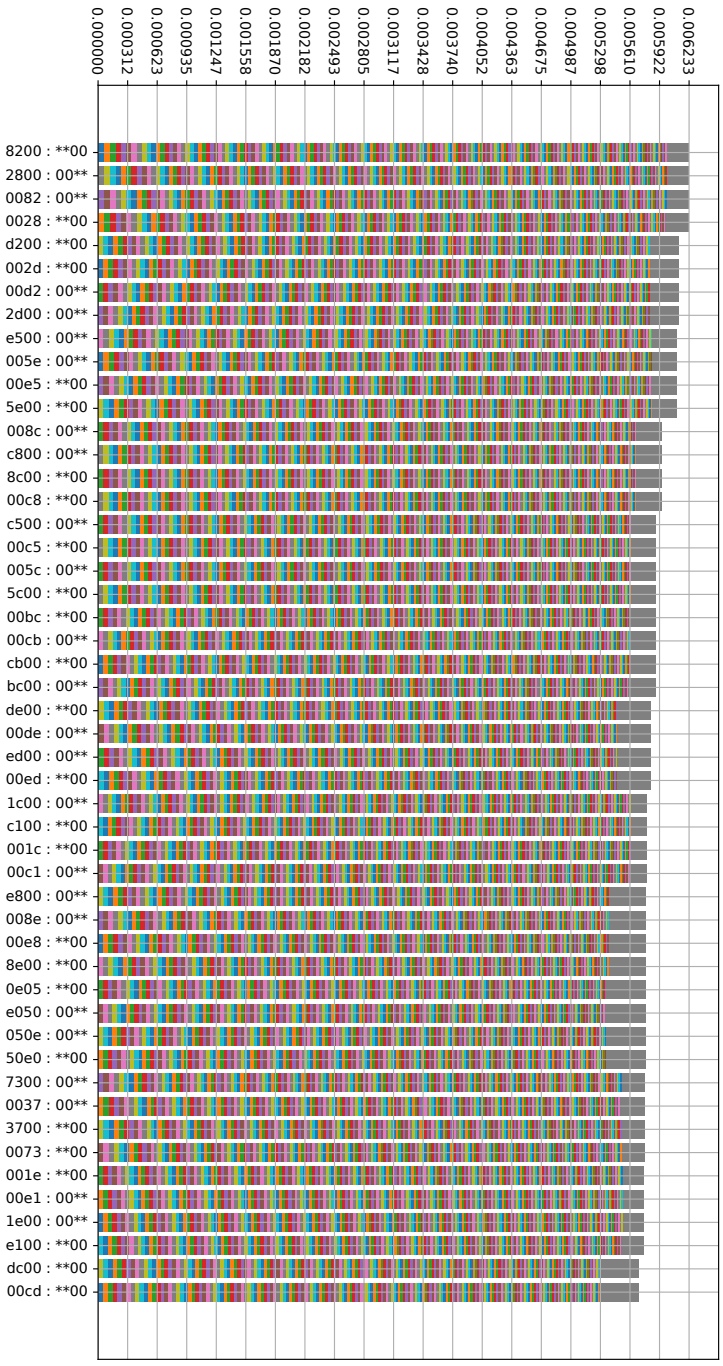
$$\texttt{pqrs} \rightarrow \texttt{?} \rightarrow \texttt{?} \rightarrow \texttt{00xy}$$
$$\texttt{pqrs} \rightarrow \texttt{?} \rightarrow \texttt{?} \rightarrow \texttt{00wz}$$

V klasické metodě extrakce klíče bychom si zvolili pouze jednu z těchto charakteristik (například první z nich). Dále bychom pro známé dvojice šifrových textů analyzovali všechny klíče z pozic "$0\star\star0$". Daný klíč by se v žebříčku umístil dle toho, kolikrát se diference jednorundového dešifrování známých dvojic šifrových textů shoduje s diferencí $\star\star$xy.

V metodě, kterou navrhuji, se žebříček úspěšnosti klíče sestavuje z pohledu všech diferenčních charakteristik uvažované šablony (společná vstupní diference a společná pozice koncové diference). Bližší popis a experimenty této metody probereme v následující kapitole. Na tomto místě si už jen uvedeme graf, který demonstruje seřazené *několikanásobné diferenciální charakteristiky*. Jako metriku řazení jsem zvolil součet pravděpodobností jednotlivých diferenciálních charakteristik. Nechci tím ale tvrdit, že tento součet představuje očekávanou pravděpodobnost nalezení klíče popisovanou metodou. V grafu 4.5 jsou zobrazeny pouze jednotlivé několikanásobné dif. charakteristiky s výstupními diferencemi na pozicích "$00\star\star$". Výstupní diference jsou zároveň omezeny na množinu výstupních diferencí všech nalezených sjednocených charakteristik s pravděpodobnostmi 2^{-14}. Tyto charakteristiky jsou tak omezeny pouze na 30 výstupních diferencí, avšak máme jistotu, že jsou v nich zahrnuty nejlepší sjednocené dif. charakteristiky. Na vodorovné ose uvažovaného grafu jsou pak označeny pouze vstupní diference. Graf stejného typu bez žádných omezení výstupních diferencí a jejich pozic, který zobrazuje nejlepší několikanásobné diferenciální charakteristiky z pohledu součtu pravděpodobností jednotlivých dif. charakteristik můžeme pozorovat na grafu 4.6. Pro každou několikanásobnou dif. charakteristiku jsou její jednotlivé dif. charakteristiky seřazeny dle jejich pravděpodobností a dif. charakteristiky s pravděpodobnostmi menšími, než 2^{-16} jsou vykresleny šedou barvou.



Obrázek 4.5: Pravděpodobnosti několikanásobných sjednocených diferenciálních charakteristik s koncovou diferencí na pozicích $00\star\star$



Obrázek 4.6: Pravděpodobnosti nejlepších několikanásobných sjednocených diferenciálních charakteristik

Kapitola 5

Metody útoků a extrakce klíčů

V předchozí kapitole jsme se zabývali způsoby vytváření diferenciálních charakteristik. Nyní se budeme soustředit na měření jejich úspěšnosti při získávání tajného šifrovacího klíče. Obecnou techniku extrakce klíče jsme si uvedli v závěru kapitoly 3. Úspěšnost nalezení správného šifrovacího klíče ovlivňuje několik faktorů. Mimo zvolenou diferenciální charakteristiku je to také počet volených dvojic otevřených textů a způsob sestavování žebříčku kandidátních klíčů. V průběhu provádění testů však zjistíme, že úspěšnost je také ovlivněna samotným šifrovacím klíčem.

Mnohé práce zabývající se kryptoanalýzou libovolné šifry zpravidla provádí také útoky na jejich redukované verze. V našem případě by to byl 2 a 3-rundový Baby Rijndael. Tyto verze šifry jsem podrobil útoku diferenciální kryptoanalýzy a úspěšnost nalezení šifrovacího klíče byla i při použití malého počtu dvojic OT téměř stoprocentní. U těchto redukovaných verzí Baby Rijndaelu by se dalo zkoumat, jaká je spodní hranice počtu volených dvojic OT/ŠT pro zachování uvedené úspěšnosti. 4-rundová verze šifry však představovala větší výzvu pro průzkum parametrů, které zásadně ovlivňují nalezení správného šifrovacího klíče.

5.1 Způsoby konstrukce žebříčku kandidátních klíčů

Mezi faktory, které ovlivňují úspěšnost nalezení správného šifrovacího klíče, patří zcela jistě volba diferenciální charakteristiky a počet použitých dvojic OT a jim odpovídajících ŠT. Do úspěšnosti se však zapojují i další faktory bez ohledu na způsoby útoků, které si budeme v této kapitole popisovat. Jedním z takových faktorů je způsob přiřazování pořadí jednotlivým kandidátním klíčům v sestavovaném žebříčku. Z kapitoly 3 víme, že během provádění útoku je každému kandidátnímu klíči k_c přiřazeno skóre, na základě toho, v kolika případech se diference dešifrovaných ŠT s využitím k_c rovná výstupní diferenci

použité diferenciální charakteristiky. Věrohodnost kandidátního klíče je pak přímo úměrná jeho získanému skóre. Empiricky jsme však zjistili, že v mnoha případech nastávají situace, kdy několik kandidátních klíčů dosahuje shodného skóre. Řazení takových žebříčků pak vnáší datovou citlivost do statistik úspěšností útoků, protože pořadí testovaných kandidátních klíčů je vždy nějak uspořádáno a druhým řadícím kritériem je ve výchozím stavu stabilních řadících algoritmů hodnota kandidátního klíče. Nabízelo se několik východisek:

I Množině klíčů se shodným skóre přiřadit nejnižší pozici z původního řazení této množiny klíčů

II Množině klíčů se shodným skóre přiřadit nejvyšší pozici z původního řazení této množiny klíčů

III Množině klíčů se shodným skóre přiřadit aritmetický průměr pozic z původního řazení této množiny klíčů

klíč	skóre	defaultní pořadí	I	II	III
...	...	...	...	...	...
...	s_α	...	p_y	p_x	p_q
...	...	...	...	...	...
k_n	s_α	p_x	p_y	p_x	p_q
k_{n+1}	s_β	p_{x+1}	p_{x+1}	p_{x+m}	$p_{x+1} + \frac{p_{x+m} - p_{x+1}}{m}$
...	...	...	...	...	...
k_{n+m}	s_β	p_{x+m}	p_{x+1}	p_{x+m}	$p_{x+1} + \frac{p_{x+m} - p_{x+1}}{m}$
k_{n+m+1}	s_γ	p_{x+m+1}	p_{x+m+1}	p_z	p_w
...	...	...	...	...	...
...	s_γ	...	p_{x+m+1}	p_z	p_w
...	...	...	...	...	...

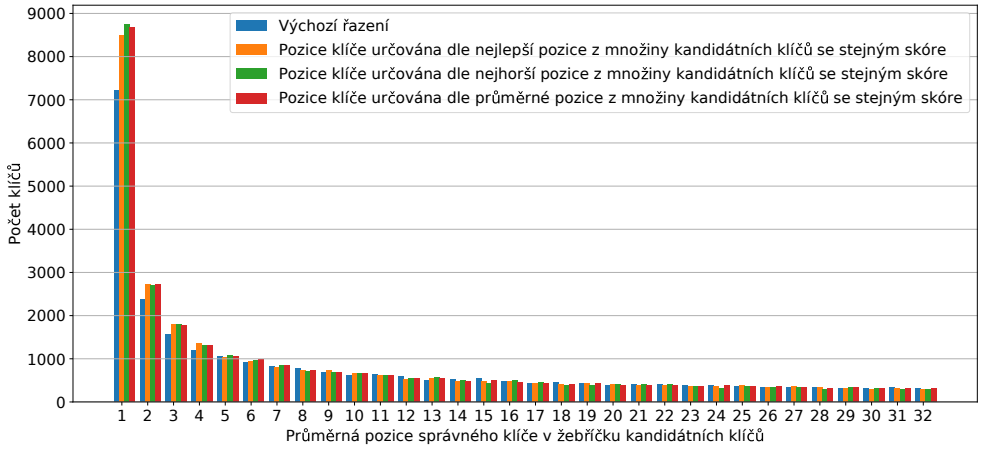
Tabulka 5.1: Ukázka možných řazení žebříčku kandidátních klíčů s ohledem na řazení klíčů se stejným skóre

Tyto metody přiřazování pořadí kandidátním klíčům jsem podrobil následujícímu testu. Zvolil jsem si diferenciální charakteristiku

$$\texttt{0020} \rightarrow \texttt{?} \rightarrow \texttt{?} \rightarrow \texttt{0bf0}$$

která má nejvyšší vyčíslenou pravděpodobnost sjednocené dif. charakteristiky (viz 4.4). Dále jsem zvolil metodu sestavování žebříčku kandidátních klíčů a iteroval jsem přes všechny existující šifrovací klíče. Pro každý šifrovací klíč jsem provedl klasický útok diferenciální kryptoanalýzou s ohledem na metodu řazení žebříčku kandidátních klíčů. Pro útoky jsem z důvodu výpočetní náročnosti nastavil počet náhodně volených diferencí OT na hodnotu 2^{10}. Tento

útok jsem pro každou takovou konfiguraci opakoval 50-krát. Z 50 získaných žebříčků kandidátních klíčů jsem pro každý kandidátní klíč spočítal jeho průměrnou pozici a na základě těchto průměrných pozic jsem sestavil nový žebříček kandidátních klíčů. Z tohoto žebříčku jsem si zjistil pozici správného šifrovacího klíče. Pro každou metodu řazení jsem tak získal 2^{16} průměrných pozic správných klíčů z žebříčků kandidátních klíčů klasické diferenciální kryptoanalýzy. Tyto průměry pozic můžeme pozorovat na sloupcovém grafu 5.1. Indexy jednotlivých sloupců reprezentují celočíselný průměr pozice v žebříčku kandidátních klíčů a výška sloupce reprezentuje počet klíčů, které dosáhly tohoto odpovídajícího průměru. Na tomto grafu můžeme pozorovat, že všechny 3 navrhované metody přiřazování nových pozic kandidátním klíčům vykazují lepší vlastnosti, než řazení pozic dle hodnoty klíče. Na grafu však můžeme pozorovat i zdánlivý paradox, kde přiřazování pozic dle metody II vykazuje lepší výsledky než metoda I. To může být dáno tím, že jednotlivé metody byly testovány odděleně a výběr diferencí OT byl volen náhodně. Pro experimenty, které si představíme v této kapitole, jsem se však rozhodl využívat metodu III.



Obrázek 5.1: Index sloupce představuje průměrnou pozici správného klíče v klasické diferenciální kryptoanalýze vzhledem k metodě určování pozic kandidátů klíčů se shodným skóre. Výška sloupce pak určuje počet unikátních klíčů, které pro testovanou diferenciální charakteristiku dosáhly uvažované průměrné pozice klíče

5.2 Sjednocené diferenciální charakteristiky

V této sekci si uvedeme extrakci klíče s využitím sjednocených diferenciálních charakteristik. Útok jsem experimentálně ověřil na mnoha charakteristikách, nicméně zde si vybereme 5 zástupců s nejlepší vyčíslenou pravděpo-

dobností. U těchto charakteristik prověříme jejich úspěšnost, se kterou jsme schopni získat pravou část šifrovacího klíče pro zafixovaný parametr počtu volených dvojic OT. Tento parametr jsem nastavil na hodnotu $2^{10} = 1024$.

Pro zachování podobnosti všech provedených experimentů jsem zvolil jednotný způsob grafického zobrazení úspěšnosti zvolených diferenciálních charakteristik. Protože úspěšnost nalezení správného klíče v žebříčcích kandidátních klíčů nebyla vysoká, přistoupil jsem k následující metodice.

- Zvolíme si diferenciální charakteristiky a nastavíme si fixní parametry prováděného testu
- Provedeme útok se zaznamenáním pozice správného klíče v sestaveném žebříčku. Pro statistickou významnost budeme tento útok několikanásobně[18] opakovat.
- Ze získané množiny pozic správných klíčů sestavíme graf, který udává pravděpodobnost nalezení správného klíče v závislosti na hloubce, do které bychom potenciálně správný klíč hledali. Kontrastní křivkou tohoto grafu bude vždy křivka funkce $f(x) = \frac{x}{2^m}$, kde m je počet bitů části klíče. Tato křivka vystihuje nalezení správného klíče, pokud by byla pozice správného klíče náhodná veličina s rovnoměrným rozdělením.

Test úspěšnosti jsem pro každou diferenciální charakteristiku prováděl 150krát. Výsledek tohoto testu pro 5 vybraných charakteristik můžeme pozorovat na grafu 5.2. Za povšimnutí stojí, že výsledky úspěšnosti nekopírují pořadí charakteristik dle jejich pravděpodobnosti. Například charakteristika `5000` $\rightarrow$ `04f0` v našem testu dopadla lépe než charakteristika `0020` $\rightarrow$ `0bf0`, která má vyšší teoretickou pravděpodobnost. V průběhu experimentů s tímto druhem testu jsem zjistil, že úspěšnost nalezení šifrovacího klíče závisí také na použitém tajném klíči. Této problematice se budeme věnovat v 5.5.

5.3 Několikanásobné diferenciální charakteristiky

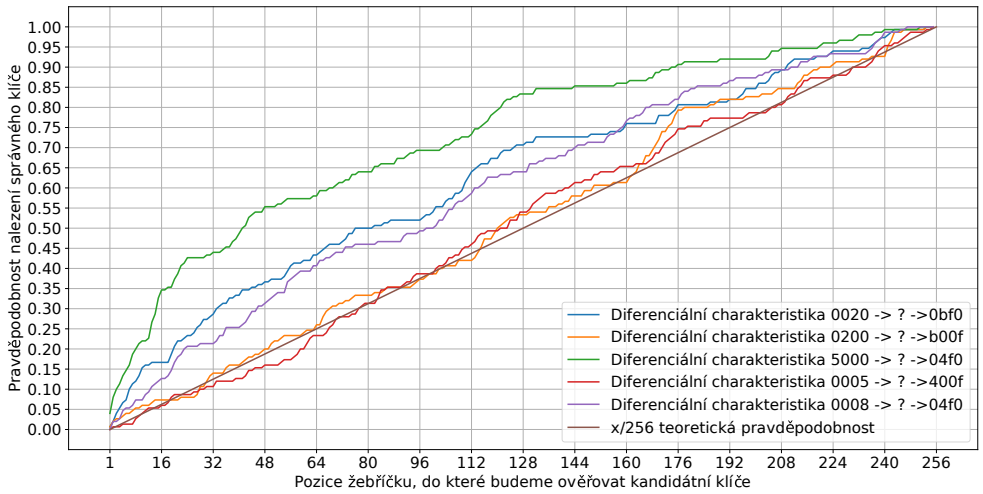
Nyní si uvedeme extrakci klíče s druhým typem diferenciálních charakteristik. Každá několikanásobná charakteristika se vyznačuje tím, že pro jednu vstupní diferenci $\triangle in$ obsahuje množinu výstupních diferencí[19] OUT. Označme si mohutnost množiny výstupních diferencí

$$|OUT| = n$$

Mějme nyní množinu dvojic OT se vstupními diferencemi $\triangle in$ a jim odpovídající ŠT. Tyto diference můžeme použít pro n samostatných útoků. Oproti

[18] Tento násobek budeme u každého testu vždy uvádět

[19] Přesnější charakterizaci výstupních diferencí nalezneme v kapitole 4.5.2



Obrázek 5.2: Graf zobrazující pravděpodobnost nalezení správného klíče v žebříčku kandidátních klíčů jednotlivých sjednocených diferenciálních charakteristik v závislosti na hloubce, do které bychom kandidátní klíče ověřovali na jejich správnost

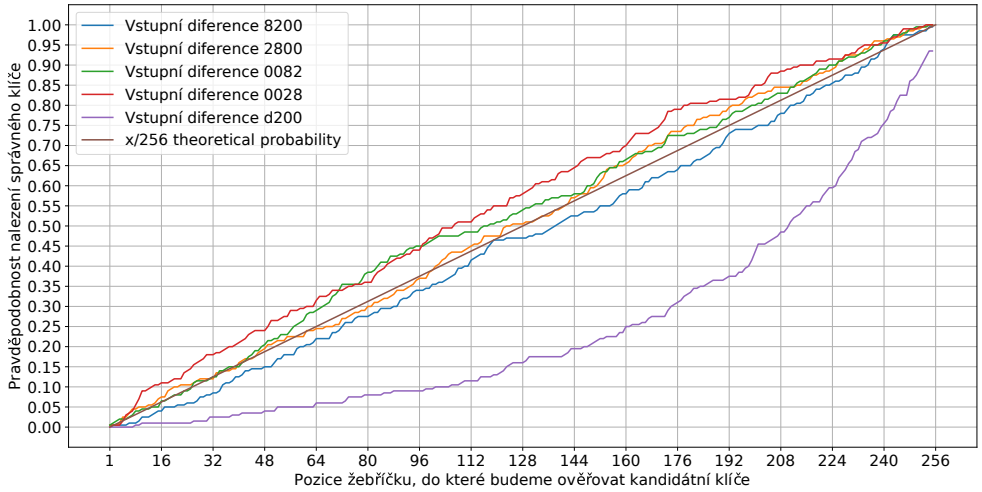
klasickému přístupu tak získáváme n žebříčků kandidátních klíčů a všechny žebříčky klíčů navíc obsahují stejnou podmnožinu klíčů. Z uvažované sady n žebříčků je potřeba sestavit jeden společný. V následujících podsekcích se tak budeme věnovat metodám, jak uvažované sady žebříčků sjednocovat pro vylepšení pozice správného šifrovacího klíče.

5.3.1 Dle (ne)váženého součtu skóre daného klíče v několika žebříčcích

Prvním naivním způsobem vytváření společného žebříčku je sečíst dosažená skóre jednotlivých kandidátních klíčů. Každý žebříček z této množiny představuje jeden provedený útok z pohledu jedné diferenciální charakteristiky. Každá taková diferenciální charakteristika má však svoji pravděpodobnost, s jakou očekáváme pro danou vstupní diferenci konkrétní výstupní diferenci. Jednotlivé útoky jsou však provedeny s charakteristikami různých pravděpodobností a tím pádem výsledné žebříčky mají v pravděpodobnosti jiné váhy. Navrhl jsem tedy test, který srovnává přístup metody součtu skóre jednotlivých klíčů s použitím vážení a bez něj. Vážení skóre pro danou diferenciální charakteristiku jsem realizoval následujícím způsobem
Test jsem provedl pro několikanásobnou charakteristiku

$$\texttt{2d5e} \rightarrow ? \rightarrow ? \rightarrow 00\star\star$$

Abych eliminoval vliv náhody při volbě dvojic OT s diferencí $\triangle\texttt{2d5e}$, využil jsem pro každý útok všechny existující dvojice OT (2^{15}). Počet útoků



Obrázek 5.3: Graf zobrazující pravděpodobnost nalezení správného klíče v žebříčku kandidátních klíčů jednotlivých několikanásobných diferenciálních charakteristik v závislosti na hloubce, do které bychom kandidátní klíče ověřovali na jejich správnost

dif. char.	pravděpodobnost
$\triangle in \rightarrow \triangle out_1$	p_1
$\vdots$	$\vdots$
$\triangle in \rightarrow \triangle out_n$	p_n

Tabulka 5.2: Obecná několikanásobná diferenciální charakteristika

klíč	skóre	vážené skóre
k_{i_1}	s_{i_1}	$\frac{p_q}{P_{diff}} s_{i_1}$
$\vdots$	$\vdots$	$\vdots$
k_{i_m}	s_{i_m}	$\frac{p_q}{P_{diff}} s_{i_m}$

(Ne)vážený žebříček kandidátních klíčů ($\triangle in \rightarrow \triangle out_q$)

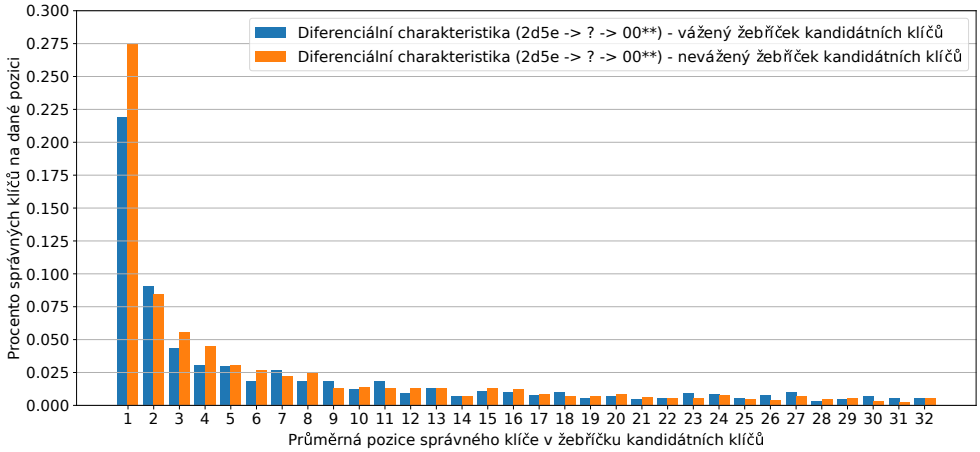
$$P_{diff} = \sum_{\alpha=1}^{n} p_\alpha$$

jsem stanovil na 2048 s náhodně voleným klíčem. Každý sloupec výsledného grafu 5.4 představuje pozici správného klíče v žebříčku kandidátních klíčů. Výška sloupce představuje procentuální zastoupení pozice ze všech provedených testů.

Z provedeného testu vyplývá, že pro statisticky větší množinu klíčů je diferenciální kryptoanalýza několikanásobných diferenciálních charakteristik účinnější bez použití vyvažování skóre žebříčků jednotlivých útoků.

5.3.2 Dle minimálního průměru pozic daného klíče v několika žebříčcích

Další metoda sjednocující několik žebříčků je založená na manipulaci s pozicemi kandidátních klíčů. Uvažujme kandidátní klíč k_c, který se nachází v každém žebříčku na jiné pozici. Novým měřítkem kvality klíče k_c se stává aritme-



Obrázek 5.4: Graf porovnávající váženou a neváženou metodu sestavování sjednoceného žebříčku kandidátních klíčů. Každý sloupec představuje pozici správného klíče ve výsledném žebříčku. Výška sloupce představuje procentuální zastoupení pozice ze všech provedených testů.

tický průměr jeho pozic z jednotlivých žebříčků. Nový společný žebříček pak sestavíme dle průměrů pozic od nejmenšího po největší. Důležité je uvědomit si, že tento postup pracuje s jednotlivými pozicemi kandidátních klíčů a nikoliv s jejich dosaženými skóre. Můžeme jej tedy aplikovat nezávisle na předchozí metodě (ne)vážení skóre jednotlivých žebříčků. Výsledky měření si uvedeme a popíšeme níže, společně s následující metodou.

5.3.3 Dle minimálního rozptylu pozic daného klíče v několika žebříčcích

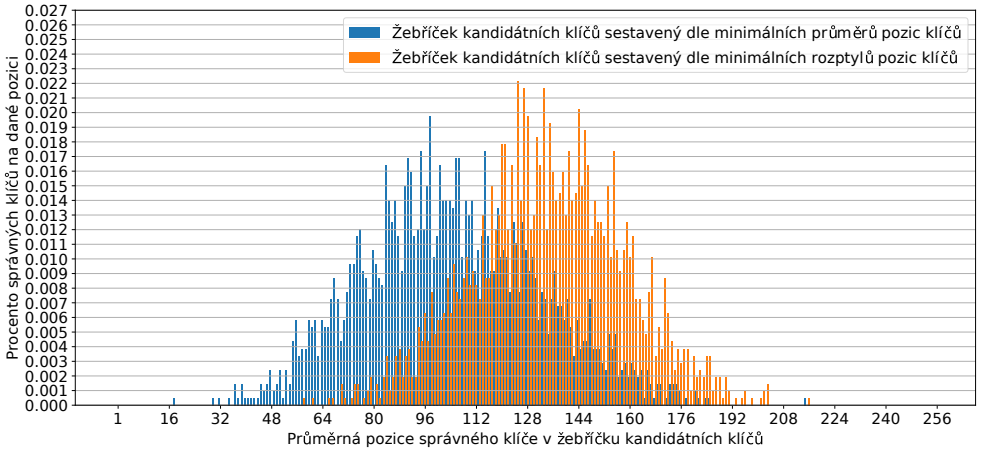
Velmi podobným způsobem, jako v předchozí podsekci, je možné nový žebříček kandidátních klíčů sestavit dle minimálních rozptylů pozic každého klíče ze všech sestavených žebříčků.

Graf 5.5 je výsledkem měření metod 5.3.2 a 5.3.3. Pro teto test jsem zvolil několikanásobnou diferenciální charakteristiku

$$\texttt{2d5e} \rightarrow ? \rightarrow ? \rightarrow 00\star\star$$

Dále jsem stanovil počet náhodně volených dvojic OT s diferencí $\triangle\texttt{2d5e}$ na 2^{10}. Počet útoků jsem stanovil na $10 \cdot 2048$, s náhodně voleným klíčem po každém 10. pokusu. Pro každý provedený útok s daným klíčem jsem si vypočítal průměrnou pozici správného klíče ze sjednoceného žebříčku sestaveného dle metod 5.3.2 a 5.3.3. Průměrné pozice jsem zaokrouhlil na celočíselnou hodnotu a v grafu 5.5 můžeme pozorovat procentuální zastoupení těchto pozic správných klíčů. Obě varianty vytvářejí diskrétní variantu gaussovy křivky a správný klíč se v obou případech nachází s největší pravděpodobností blízko

středu žebříčku. S metodou ověřování kandidátních klíčů od nejnižší pozice k těm nejvyšším lze však označit metodu 5.3.2 za úspěšnější.



Obrázek 5.5: Graf porovnává metody sestavování sjednoceného žebříčku kandidátních klíčů dle minimálních průměrů a dle minimálních rozptylů pozic kandidátních klíčů. Každý sloupec představuje pozici správného klíče ve výsledném žebříčku. Výška sloupce představuje procentuální zastoupení pozice ze všech provedených testů

5.3.4 Přístup paní Marty Sartori

S originálním nápadem sestavování žebříčku kandidátních klíčů přišla paní Marta Sartori[17]. Její přístup sází na fakt, že nejlepší kandidátní klíče nalezneme na předních příčkách ve všech sestavených žebříčcích. Postup je následující:

I Vytvoříme nový prázdný žebříček

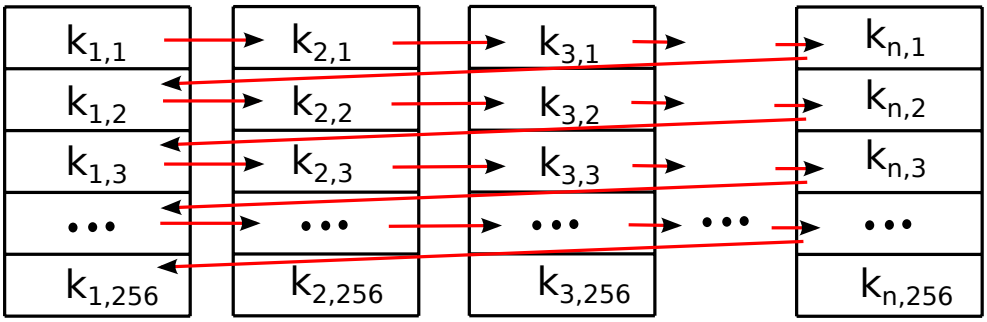
II Iterujeme postupně přes všechny pozice a všechny žebříčky.

III Během průchodu žebříčky si zaznamenáváme klíče, které jsme průchodem přečetli

IV Pokud narazíme na klíč, který jsme již dříve přečetli v jiném žebříčku, je tento klíč přidán na konec nového žebříčku

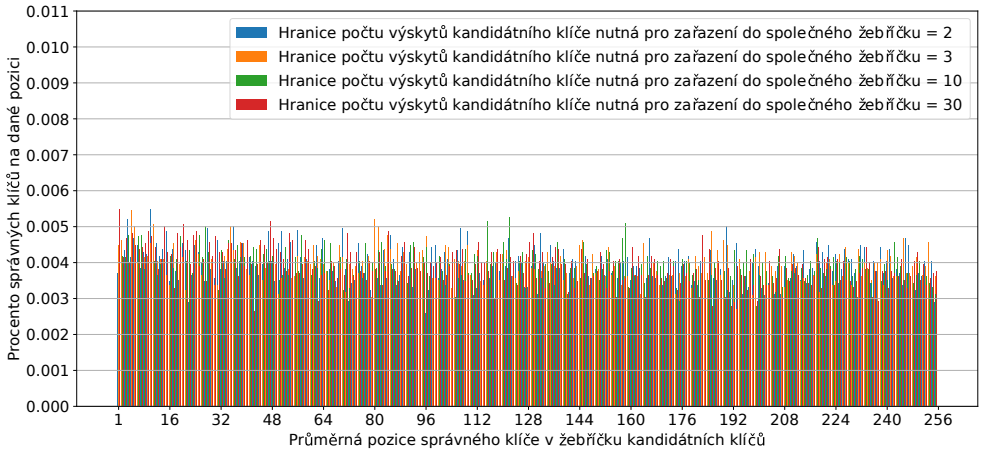
Přidání nového klíče do vznikajícího žebříčku lze také podmínit větším počtem výskytů (nejvýše však počtem procházených žebříčků). Schéma průchodu žebříčky můžeme pozorovat na obrázku 5.6.

Měření probíhalo ve stejném nastavení jako měření metody sestavování sjednoceného žebříčku dle minimálních průměrů a rozptylů pozic kandidátních klíčů. Výsledky měření můžeme vidět na grafu 5.7. Vykreslil jsem několik



Obrázek 5.6: Schéma průchodu jednotlivými žebříčky během hledání opakujících se kandidátních klíčů v několika žebříčcích

sloupců, kde každý představuje jinou variantu přidávání kandidátního klíče do nově vznikajícího sjednoceného žebříčku (viz bod IV). Z grafu můžeme pozorovat, že statistické rozdělení pozice správného klíče je pro všechny vykreslené[20] metody rovnoměrné. Tato metoda je tedy pro sestavování sjednoceného žebříčku kandidátních klíčů a následnou extrakci klíče nepoužitelná.

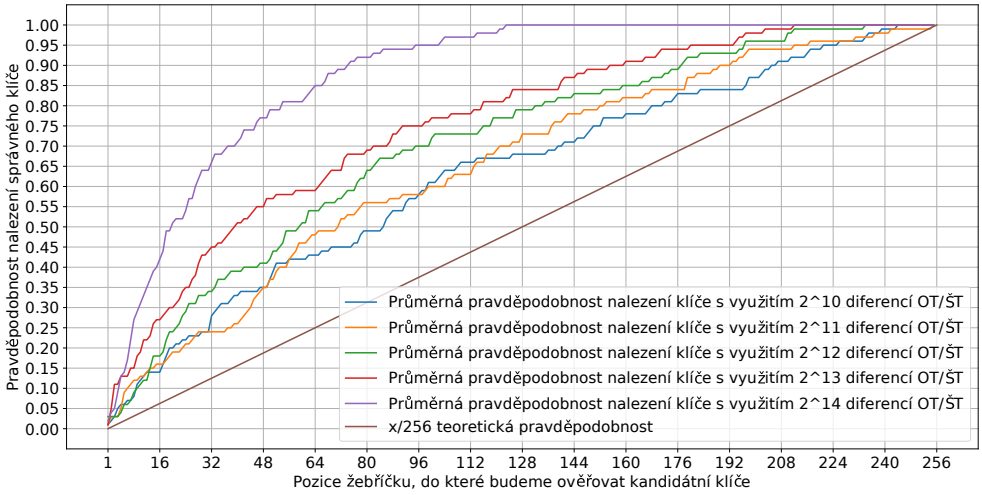


Obrázek 5.7: Graf porovnává metody sestavování sjednoceného žebříčku kandidátních klíčů dle metody paní Sartori. Každý sloupec představuje pozici správného klíče ve výsledném žebříčku. Výška sloupce představuje procentuální zastoupení pozice ze všech provedených testů

[20] Rovnoměrné rozdělení klíčů mají všechny varianty

5.4 Úspěšnost nalezení klíče dle počtu použitých dvojic OT/ŠT

Počet použitých dvojic OT/ŠT je bezesporu významný parametr, který ovlivňuje úspěšnost nalezení tajného šifrovacího klíče. Otázkou pouze je, do jaké míry. Experiment této sekce je přímočarý. Pro pevně zvolené diferenciální charakteristiky a pevně zvolený klíč budeme provádět útok a zaznamenávat úspěšnost. Měřítkem úspěšnosti zde bude průměrná pozice správného klíče v sestavovaném žebříčku kandidátních klíčů. Pro každou diferenciální charakteristiku budeme experiment opakovat 100krát a parametrem, který budeme měnit, bude počet použitých dvojic OT. Počty dvojic OT jsem nastavil na hodnoty $2^{10}, 2^{11}, 2^{12}, 2^{13}$ a 2^{14}. Protože výsledky tohoto testu jsou pro všechny druhy diferenciálních charakteristik podobné, ukážeme si výstup pouze na následující charakteristice `2d5e` $\rightarrow$ `?` $\rightarrow$ `?` $\rightarrow 00\star\star$. Pro test byl použit šifrovací klíč `0x1234`. Z výsledku testu, který můžeme pozorovat na grafu 5.8, je patrné, že mezi úspěšností diferenciální kryptoanalýzy a počtem použitých OT/ŠT existuje přímá úměra.



Obrázek 5.8: Graf zobrazující pravděpodobnost nalezení správného klíče v žebříčku kandidátních klíčů, do které bychom kandidátní klíče ověřovali na jejich správnost. Měření probíhalo na několikanásobné diferenciální charakteristice `2d5e` $\rightarrow$ `?` $\rightarrow$ `?` $\rightarrow 00\star\star$

5.5 Datová závislost úspěchu kryptoanalýzy na tajném klíči

V této sekci si ukážeme, do jaké míry je úspěšnost kryptoanalýzy závislá na použitém tajném šifrovacím klíči. Podezření na tuto závislost vznikla během

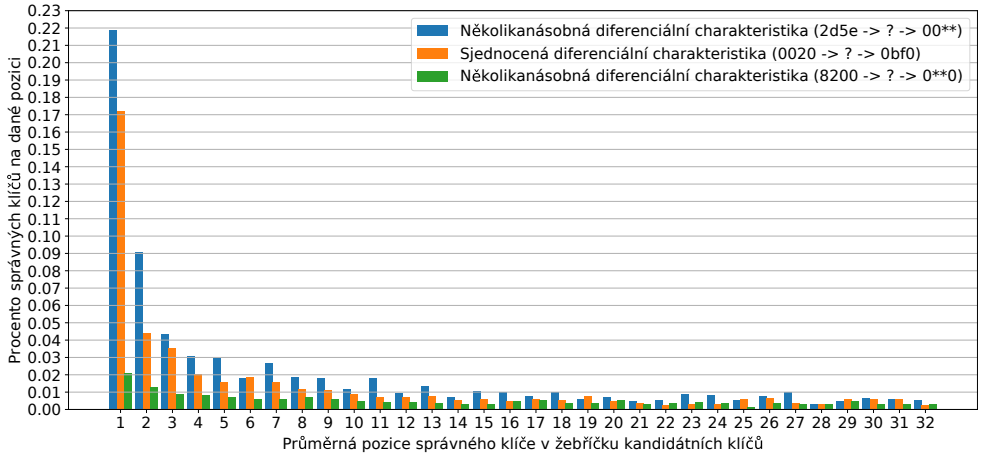
testování úspěšnosti kryptoanalýzy diferenciálních charakteristik s různými pravděpodobnostmi a s použitím konstantního tajného klíče. I přes velkou podobnost vypočtených pravděpodobností použitých charakteristik byly úspěšnosti získání správného klíče ve velkém nepoměru. Navíc byl tento nepoměr pro daný použitý tajný klíč vždy velmi podobný. Pro stejné charakteristiky a jiný tajný klíč pak mezi úspěšností existoval jiný stabilní nepoměr. Zajímalo mě tedy, pro jak velký počet klíčů je daná charakteristika „použitelná“. Navrhl jsem tedy experiment, při kterém jsem pro jednu konkrétní diferenciální charakteristiku provedl útok pro statisticky významný počet náhodných tajných klíčů a zaznamenal si jednotlivé úspěšnosti. Měřítkem úspěšnosti byla pozice reálného klíče v sestaveném žebříčku kandidátních klíčů. Abych vyloučil co nejvíce náhod kryptoanalýzy, použil jsem v tomto testu plný počet dvojic OT/ŠT (2^{15}). Důvod pro použití pouze podmnožiny klíčů byla výpočetní náročnost. Tento test jsem provedl pro sjednocenou i několikanásobnou diferenciální charakteristiku s nejlepší zjištěnou pravděpodobností. Počet použitých klíču testu jsem zvolil 2048, což představuje $\frac{1}{32}$ prostoru klíčů. Množina náhodných testovaných klíčů byla pro obě charakteristiky stejná. Výsledek testu můžeme pozorovat na grafu 5.9. Každý sloupec představuje jednu pozici správného klíče v žebříčku kandidátních klíčů. Výška sloupce představuje procentuální zastoupení pozice ze všech provedených testů. Pro přehlednost grafu jsem počet pozic omezil pouze na prvních 32. Test jsem prováděl na těchto diferenciálních charakteristikách

$$\texttt{2d5e} \rightarrow \texttt{?} \rightarrow \texttt{?} \rightarrow \texttt{00}\star\star$$
$$\texttt{0020} \rightarrow \texttt{?} \rightarrow \texttt{?} \rightarrow \texttt{0bf0}$$
$$\texttt{8200} \rightarrow \texttt{?} \rightarrow \texttt{?} \rightarrow \star\texttt{00}\star$$

V dalším zkoumání této problematiky bych doporučoval ověřit, zda jsou zjištěná rozdělení úspěšnosti kryptoanalýzy podobná pro další diferenciální charakteristiky. Zajímavé by bylo také prozkoumat rozdělení klíčů v provedené analýze a zda je toto rozdělení klíčů podobné, či jiné pro jinou diferenciální charakteristiku. Pokud by se ukázalo, že tato rozdělení jsou podobná, pak by šifra byla zranitelná z pohledu slabých klíčů.

5.6 Útok na algoritmus generování rundovních klíčů

Dosud jsme útok prováděli způsobem výběru jisté části klíče. Pro každou jednotlivou část klíče a pro nám známé dvojice ŠT jsme adekvátně dešifrovali jednu rundu šifry. Výsledek dešifrování jsme pak porovnávali s výstupními diferencemi použitých diferenciálních charakteristik. Pokud jsme byli úspěšní a dokázali zjistit všechny části posledního rundovního klíče, pak bylo velice



Obrázek 5.9: Graf zobrazuje procentuální zastoupení průměrných pozic správných klíčů v žebříčcích kandidátních klíčů dle použitého typu diferenciální charakteristiky

snadné postupně vypočítat všechny ostatní rundovní klíče. Napadlo mě zaměřit svou pozornost na samotný algoritmus generování rundovních klíčů. Jeden z požadavků tohoto algoritmu je, že z neúplné znalosti rundovního klíče by nemělo být možné vypočítat ostatní rundovní klíče. To v plné míře platí, avšak nyní si zodpovíme otázku, zda ze znalosti části klíče lze odvodit část jiného rundovního klíče. Algoritmus generování rundovních klíčů jsme si ukázali v 2. kapitole. Připomeňme obecné odvození 5. rundovního klíče z předchozího 4. rundovního klíče.

...	ka_4	kc_4	ka_5	kc_5
...	kb_4	kd_4	kb_5	kd_5

, kde $ka_i, kb_i, kc_i, kd_i \in GF(2^4)$

Vztah 4. a 5. rundovního klíče můžeme popsat následujícími rovnicemi:

$$ka_5 = SubBytes(kd_4) \oplus ka_4 \oplus \texttt{0x8} \tag{5.1}$$
$$kb_5 = SubBytes(kc_4) \oplus kb_4 \tag{5.2}$$
$$kc_5 = kc_4 \oplus ka_5 \tag{5.3}$$
$$kd_5 = kd_4 \oplus kb_5 \tag{5.4}$$

Během fáze extrakce klíče, kde provádíme měření pro jistou konkrétní část klíče, chápeme tuto část klíče jako známou součást kryptosystému. S využitím rovnice 5.3 však zjistíme, že pokud si zafixujeme části klíče ka_5 a kc_5, pak část klíče kc_4 je přesně definována. Obdobně je tomu tak s volbou částí klíčů kb_5 a kd_5. Na tomto místě je vhodné poukázat na důležitost operace rotace částí klíčů kd_4 a kc_4 při odvozování následujícího rundovního klíče. Bez této operace

by bylo možné dopočítat všechny části rundovních klíčů na pozicích ka_i a kc_i. Tato vlastnost a drobná slabina generování klíčů platí i pro originální šifru Rijndael.

V této kapitole jsme ověřili mnoho parametrů diferenciální kryptoanalýzy, a proto si uvedeme krátké shrnutí. Zjistili jsme, jakými způsoby volit některé parametry (přiřazení pozic kandidátním klíčům se shodným skóre). Otestovali jsme několik variant sjednocených diferenciálních charakteristik a zjistili jsme, že jejich pravděpodobnosti neodpovídají jejím reálným výsledkům v DK a úspěšnost nalezení správného klíče mnohem více závisí na parametrech počtu volených OT/ŠT a samotném šifrovacím klíči. Z pohledu několikanásobných diferenciálních charakteristik jsme otestovali mimo jiné způsoby vytváření sjednocených žebříčků kandidátních klíčů. Za úspěšné se dají považovat metody sestavování dle minimálních průměrů a rozptylů.

V následující kapitole se budeme zabývat vlivem parametrů výpočetní a prostorové složitosti diferenciální kryptoanalýzy.

Kapitola 6

Odhad paměťové a výpočetní složitosti kryptoanalýzy

V předchozí kapitole jsme si předvedli několik typů útoků spolu s jejími výsledky. Pro srovnání s ostatními druhy kryptoanalýz však musíme vyčíslit složitosti jednotlivých útoků. V úvodu provedeme hrubou analýzu složitosti diferenciální kryptoanalýzy v závislosti na několika parametrech. Následně pak srovnáme dosažené výsledky s doposud publikovanými výsledky ostatních kryptoanalytických útoků.

6.1 Paměťová a výpočetní složitost

U paměťové složitosti se omezíme pouze na počty dvojic OT a ŠT. Výpočetní složitost by šlo odvozovat od složitostí a počtu jednotlivých operací použitých pro získání diferencí OT/ŠT a samotný útok. Složitosti jednotlivých operací však odvozovat nebudeme, protože implementačně lze operace *SubBytes* a *MixColumns* převést na předpočítané tabulky. Náhodný přístup do paměti je z pohledu dnešních počítačů časově nejnáročnější atomická operace. Naopak operace *rotace bitů*[21] nebo operace *xor*[22] je v porovnání s náhodným přístupem do paměti časově zanedbatelná operace. Proto jsem se rozhodl odvozovat časovou složitost diferenciální kryptoanalýzy z pohledu počtu přístupů do paměti (tj. z pohledu počtu využitých operací *SubBytes*, *MixColumns* a $SubBytes^{-1}$).

6.1.1 Výpočetní složitost

Šifra Baby Rijndael obsahuje v každé ze 4 rund 4 operace *SubBytes*. Dále pak obsahuje v každé ze 3 rund 2 operace *MixColumns*. Výpočetní složitost

[21] využívaná v operaci *ShiftRows*
[22] využívaná v operaci *AddRoundKey*

jednoho šifrování je rovna 22 přístupům do paměti. Složitost atomické operace v rámci útoku, kdy využijeme dvojici ŠT pro dešifrování poslední rundy s využitím části hádaného klíče, využívá dle této práce 1 až 2 inverzní operace $SubBytes^{-1}$. Pro následující analýzu si označme tyto parametry:

- $\#OT$ - počet použitých OT/ŠT v rámci útoku
- kb - počet bitů hádané části klíče
- h - hloubka, do které budeme zkoušet klíč v žebříčku kandidátních klíčů

Pro stanovení celkové výpočetní složitosti si potřebujeme vysvětlit vliv definovaných parametrů kb a h. Parametr počtu bitů hádané části klíče souvisí nejen s počtem použitých inverzních operací $SubBytes^{-1}$, ale také s tím, kolik útoků budeme muset provést, abychom byli schopni určit správnost získaného klíče. Pokud budeme útočit na nepřekrývající se 8-bitové klíče, pak stačí útok provést 2 krát. V případě útoku na nepřekrývající se 4-bitové klíče bychom museli provést 4 útoky[23]. Provedení celého útoku tak vyžaduje jednorundové dešifrování s využitím všech možných částí klíčů (2^{kb}) na všech dvojicích ŠT (#OT). Parametr hloubky h pak souvisí s faktem, že výsledky, kterých jsme dosáhli v minulé kapitole, nezaručují uspokojivou úspěšnost, pokud bychom se spoléhali pouze na kandidátní klíč z vrcholu sestaveného žebříčku. Pokud se tedy rozhodneme ve variantě 8-bitových hádaných klíčů hledat správný klíč do hloubky h, pak počet kombinací dvou částí klíčů je roven h^2.

$$T_{attack} = (22\#OT + \frac{kb}{4}\#OT \cdot 2^{kb}) \cdot \frac{16}{kb} \tag{6.1}$$

$$T_{key\_verify} = 22 \cdot h^{\frac{16}{kb}} \tag{6.2}$$

$$T_{dif\_cryptanalysis} = T_{attack} + T_{key\_verify} \tag{6.3}$$

$$T_{brute\_force} = 22 \cdot 2^{16} = 1441792 \tag{6.4}$$

6.1.2 Alternativní extrakce šifrovacího klíče

Celý šifrovací klíč můžeme však získat i jinak než několikanásobným provedením útoku na unikátní části klíče. Můžeme se pokusit extrahovat pouze část klíče a zbylou chybějící část klíče odvodit pomocí hrubé síly. Výpočetní složitost se změní následovně:

$$T_{attack\_alt} = (22\#OT + \frac{kb}{4}\#OT \cdot 2^{kb}) \tag{6.5}$$

$$T_{key\_verify\_alt} = 22 \cdot h \cdot 2^{(16-kb)} \tag{6.6}$$

$$T_{dif\_cryptanalysis\_alt} = T_{attack\_alt} + T_{key\_verify\_alt} \tag{6.7}$$

$$\tag{6.8}$$

[23] V následující analýze nebudeme uvažovat útoky se smíšenými velikostmi hádaných klíčů

Do složitosti útoku by se dalo také započítat využití operace *SubBytes* v rámci expanze klíče. Konkrétně je tato funkce využita 8-krát. V naší práci se však zabýváme kryptoanalýzou s využitím pouze jednoho klíče, takže implementačně lze expanzi klíče vypočítat pouze jednou a to je vzhledem k výpočetní náročnosti útoku a ověření klíče zanedbatelná položka. Výpočetní složitost popsaná v 6.3 vyjadřuje složitost útoku s využitím pouze jedné diferenciální charakteristiky. Složitost DK *několikanásobné* diferenciální charakteristiky dostaneme tak, že rovnici 6.1 přenásobíme počtem výstupních diferencí, které charakteristika obsahuje[24].

Výsledky diferenciální kryptoanalýzy představené v minulé kapitole bohužel nedosahují takové výsledky, kde bychom mohli s jistotou tvrdit, že s použitím daných parametrů útoku jsme schopni získat šifrovací klíč se 100% úspěšností. Dále představované výsledky si proto uvedeme z pohledu předpokládané úspěšnosti nalezení šifrovacího klíče. Zcela jistě nemůžeme využít typy útoků, kde využíváme všechny existující dvojice OT dané diference. Tyto útoky by překračovaly složitost útoku hrubou silou. Ukážeme si analýzy časové složitosti útoku s využitím sjednocené i několikanásobné diferenciální charakteristiky. Z důvodů výpočetní náročnosti se mi nepovedlo naměřit útoky s více parametry pro nalezení optimálních nastavení parametrů.

6.1.3 Složitost diferenciální kryptoanalýzy sjednocené diferenciální charakteristiky

Úspěšnost diferenciální kryptoanalýzy s využitím sjednocených diferenciálních charakteristik si demonstrujeme na diferenciální charakteristice

$$\texttt{e582} \rightarrow \texttt{?} \rightarrow \texttt{?} \rightarrow \texttt{0067}$$

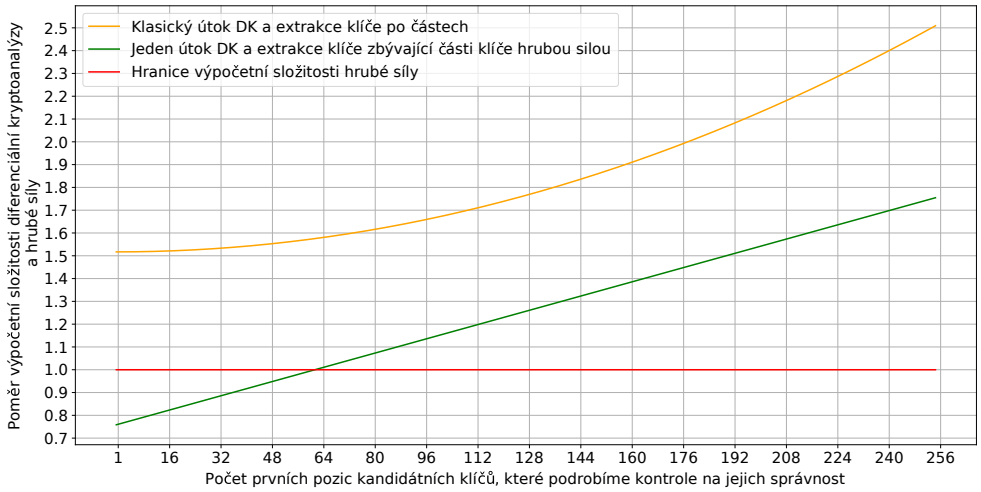
Parametry testu kopírují parametry z kapitoly 5.3.2 s tím rozdílem, že je zde použitá sjednocená diferenciální charakteristika. V testu jsem použil 2048 náhodných klíčů a pro každý klíč jsem provedl 5-násobné opakování. Výpočetní náročnost provedeného testu si demonstrujeme na poměru složitosti útoku diferenciální kryptoanalýzy a hrubé síly. Tento poměr si na grafu demonstrujeme ve dvou variantách útoku diferenciální kryptoanalýzy, které jsme si popsali výše v této kapitole při odvozování výpočetní složitosti útoku.

$$r = \frac{T_{dif\_cryptanalysis}}{T_{brute\_force}} \tag{6.9}$$

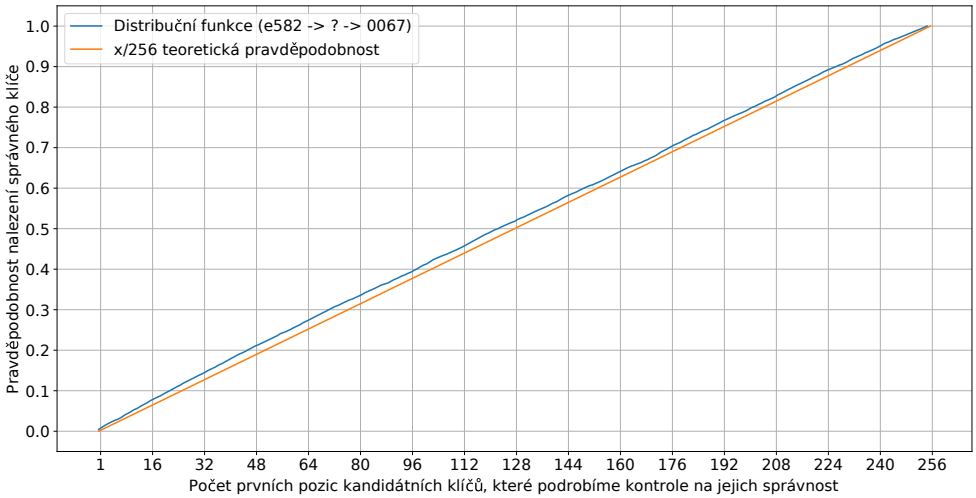
$$r_{alt} = \frac{T_{dif\_cryptanalysis\_alt}}{T_{brute\_force}} \tag{6.10}$$

V grafu 6.1 můžeme pozorovat poměry r a r_{alt} v závislosti na parametru h (hloubky prohledávání žebříčku kandidátních klíčů).

[24]Na tento útok se díváme jako na několik samostatných útoků



Obrázek 6.1: Graf poměrů výpočetní složitosti DK s využitím sjednocených diferenciálních charakteristik a hrubé síly



Obrázek 6.2: Distribuční funkce průměrné pravděpodobnosti nalezení správného klíče pro útok DK s využitím sjednocené diferenciální charakteristiky e582 $\rightarrow$? $\rightarrow$? $\rightarrow$ 0067

Úspěšnost provedeného testu můžeme pozorovat na grafu 6.2. Z grafů 6.1 a 6.2 můžeme odvodit následující závěry. Několikanásobný útok diferenciální kryptoanalýzy na všechny části klíče je při použití 2^{10} dvojic[25] OT s danou vstupní diferencí výpočetně mnohem náročnější než hrubá síla. Dále však kombinovaný útok diferenciální kryptoanalýzou, který je méně náročný než hrubá síla, nám zaručuje pravděpodobnost nalezení správného klíče v nejvýše 26 %

[25]Detaily použitých parametrů nalezneme v kapitole 5.3.2

případů.

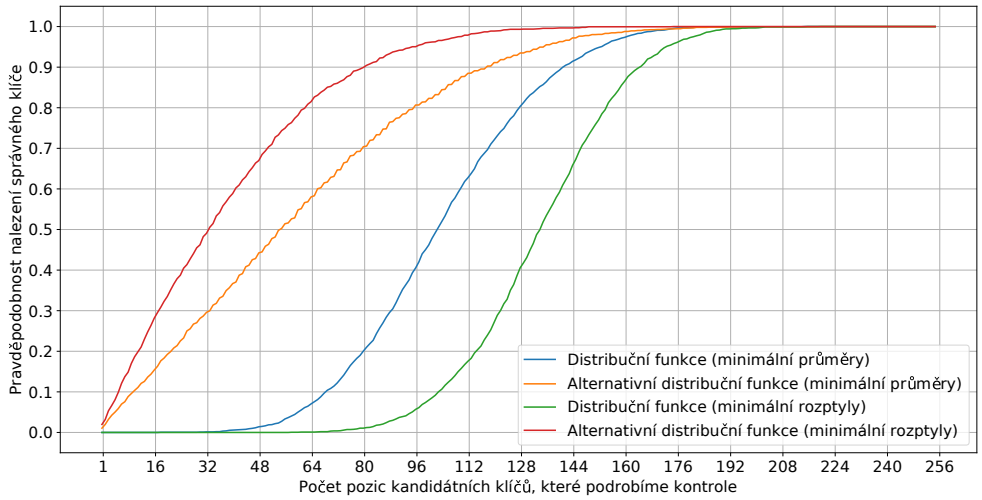
6.1.4 Složitost diferenciální kryptoanalýzy několikanásobné diferenciální charakteristiky

Úspěšnost diferenciální kryptoanalýzy s využitím několikanásobných diferenciálních charakteristik si demonstrujeme na výsledcích testů 5.3.2 a 5.3.3. V těchto testech jsme se zaměřovali na dva druhy sestavování sjednocených žebříčků. Buď to bylo z pohledu minimálních průměrů pozic kandidátních klíčů, nebo dle minimálních rozptylů jejich pozic. Na grafu 6.3 můžeme pozorovat obě varianty. Dále můžeme na tomto grafu pozorovat alternativní křivky úspěšnosti. Pokud se podíváme na graf 5.5, který znázorňoval průměrné pozice správných klíčů v sestaveném žebříčku, pak obě varianty (dle minimálních průměrů, či rozptylů) mají nejvyšší pravděpodobnosti kolem středu žebříčku. Toho by se dalo využít pro modifikaci výběrů kandidátních klíčů k ověřování na jejich správnost. Běžný přístup výběru je sekvenční od nejnižší pozice k těm vyšším. Alternativní způsob by postupoval od středu žebříčku a pak střídavě od středu ke krajním pozicím. Pro žebříček s 256 kandidátními klíči by se kontrolovaly pozice v následujícím pořadí:

$$128, 127, 129, 126, 130, \ldots, 1, 256$$

Tento způsob průchodu žebříčku jsme v této práci netestovali a je to jedna z cest, kterou by stálo za to prozkoumat detailněji. Na grafu 6.3 můžeme však pozorovat, že alternativní přístup průchodu žebříčku nám dává daleko lepší pravděpodobnost nalezení správného klíče. Z alternativních přístupů je lepší ten se žebříčkem sestaveným dle minimálních rozptylů. Naopak z přirozených přístupů průchodu žebříčku je lepší ten, u kterého využíváme žebříček sestavený dle minimálních průměrů.

Než se však podíváme na poměry výpočetní náročnosti útoku s využitím několikanásobných diferenciálních charakteristik, musíme se podívat, jak se z obecného hlediska změní složitost oproti klasickému přístupu. V předchozí kapitole jsme si řekli, že na útok s několikanásobnými dif. charakteristikami se můžeme dívat jako na sérii n klasických útoků, kde n je počet výstupních diferencí definovaných diferenciální charakteristikou. Tento přímočarý přístup by byl však velmi neefektivní, protože by n-krát opakoval jednorundové dešifrování známé množiny ŠT. Implementačně lze tento krok provést pouze jednou a zaměřit se na efektivní sestavování n žebříčků kandidátních klíčů současně. Hrubým odhadem nám oproti klasickému útoku přibudou 2 operace čtení/zápisu do paměti pro každou dvojici ŠT a hádanou část klíče. Výpočetní náročnost uvažovaného útoku můžeme zapsat následovně:



Obrázek 6.3: Distribuční funkce průměrných pravděpodobností nalezení správného klíče pro útok DK s využitím několikanásobné diferenciální charakteristiky `2d5e` $\rightarrow$ `?` $\rightarrow$ `?` $\rightarrow$ `00` $\star\star$

$$T_{attack\_multiple} = (22\#OT + 2 \cdot \frac{kb}{4}\#OT \cdot 2^{kb}) \cdot \frac{16}{kb} \tag{6.11}$$

$$T_{key\_verify} = 22 \cdot h^{\frac{16}{kb}} \tag{6.12}$$

$$T_{dif\_cryptanalysis\_multiple} = T_{attack\_multiple} + T_{key\_verify} \tag{6.13}$$

Stejným způsobem se změní i předpis výpočetní složitosti útoku, kde útok provádíme jen na jednu část klíče a na zbylou část útočíme hrubou silou.

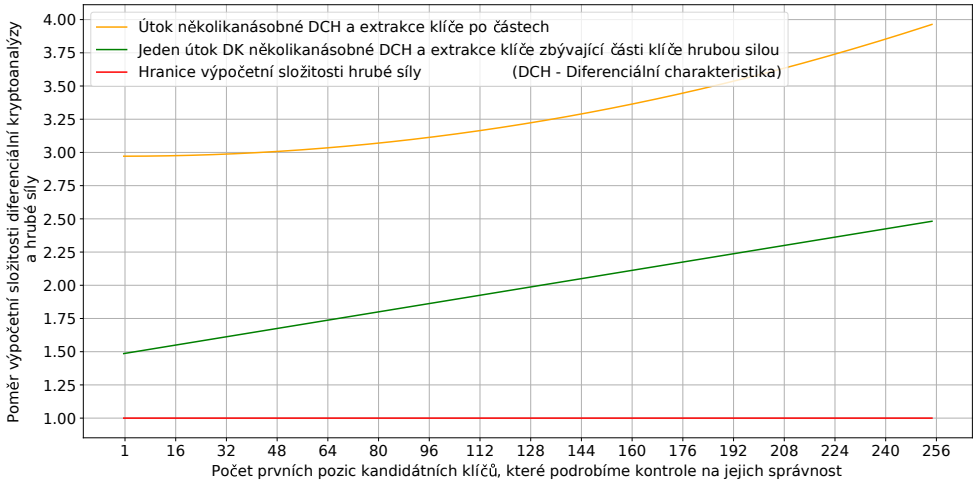
$$T_{attack\_alt\_multiple} = (22\#OT + 2 \cdot \frac{kb}{4}\#OT \cdot 2^{kb}) \tag{6.14}$$

$$T_{key\_verify\_alt} = 22 \cdot h \cdot 2^{(16-kb)} \tag{6.15}$$

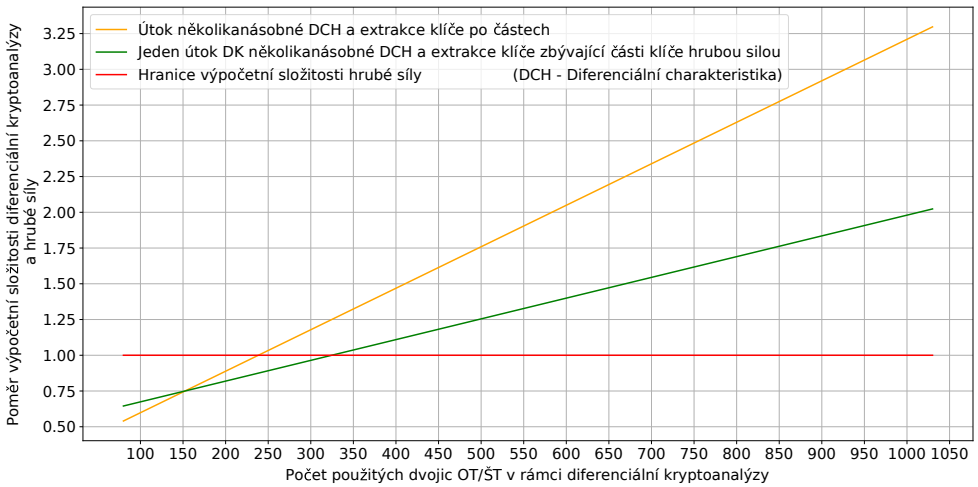
$$T_{dif\_cryptanalysis\_alt\_multiple} = T_{attack\_alt\_multiple} + T_{key\_verify\_alt} \tag{6.16}$$

Oba tyto přístupy výpočetní náročnosti si demonstrujeme na poměru složitosti uvažovaných útoků a hrubé síly v závislosti na hloubce prohledávaného žebříčku kandidátních klíčů. Křivky těchto poměrů můžeme sledovat na grafu 6.4. Z tohoto grafu můžeme vyčíst fakt, že uvažovaný útok s využitím 1024 dvojic OT/ŠT je v nejlepším případě 1.5-krát výpočetně náročnější, než hrubá síla.

Výpočetní složitost závisí na dvou parametrech: na počtu volených dvojic OT/ŠT a hloubce prohledávaného žebříčku kandidátních klíčů. Uveďme si tedy graf poměru výpočetní náročnosti DK a hrubé síly v závislosti na počtu



Obrázek 6.4: Graf poměrů výpočetní složitosti DK s využitím několikanásobných diferenciálních charakteristik a hrubé síly v závislosti na hloubce ověřování kandidátních klíčů sestaveného žebříčku



Obrázek 6.5: Graf poměrů výpočetní složitosti DK s využitím několikanásobných diferenciálních charakteristik a hrubé síly v závislosti na počtu použitých dvojic OT/ŠT

použitých dvojic OT/ŠT. Pro křivky grafu 6.5 byl parametr hloubky prohledávaného žebříčku kandidátních klíčů zafixován na hodnotu 128.

Na grafu 6.5 můžeme pozorovat skutečnost, že útok diferenciální kryptoanalýzou s využitím několikanásobných diferenciálních charakteristik je výpočetně méně náročný s využitím nejvýše 320 dvojic OT/ŠT.

Nutno také poznamenat, že při využití několikanásobných charakteristik nám nezanedbatelně vzrůstá paměťová složitost v souvislosti s mnoha žebříčky

kandidátních klíčů. U sjednocené charakteristiky to byl pouze jeden žebříček, a ten zabíral 2^{kb} paměťových míst. U několikanásobné charakteristiky to je $m \cdot 2^{kb}$ paměťových míst, kde m chápeme počet výstupních diferencí charakteristiky.

6.2 Srovnání dosažených výsledků s ostatními pracemi

Pokud se pokusíme srovnat dosažené výsledky s výsledky ostatních prací, které se věnují útoky na šifru Baby Rijndael, budeme se muset nejdříve vypořádat s několika nesnázemi. Každá práce interpretuje své výsledky svým unikátním způsobem. Některé práce dokonce neudávají analýzu výpočetní složitosti provedených útoků.

6.2.1 Srovnání s lineární kryptoanalýzou

Kokeš v [9] uvádí, že dokáže najít šifrovací klíč dvourundového Baby Rijndaelu se 100 % úspěšností, u 3-rundové varianty s úspěšností 36,6 % a u plné varianty s úspěšností 4,4 %. V naši práci jsme se věnovali pouze plné variantě šifry. Pokud se omezíme pouze na varianty útoků, které jsou rychlejší, než hrubá síla, pak je úspěšnost našeho útoku, se 26 %, násobně lepší, než v případě lineární kryptoanalýzy (viz. 6.1.3).

6.2.2 Srovnání s algebraickou kryptoanalýzou

Vábková prováděla v [10] útok algebraickou kryptoanalýzou s využitím komerčního software Mathematica. Ve své práci z pochopitelných důvodů neodhadovala výpočetní náročnost svého útoku. I přes tento fakt musíme přiznat, že její úspěšnost 85 % je násobně lepší, než jakou jsme dosáhli v naši práci.

6.2.3 Srovnání s diferenciální kryptoanalýzou pomocí nemožných diferenciálů

Poljak v [11] jako jediný vztahuje úspěšnost své kryptoanalýzy vzhledem ke složitosti útoku hrubou silou. Jeho výsledky jsou však řádově lepší, než naše, protože jemu se povedlo extrahovat klíč se 100 % úspěšností v čase menším, než který bychom použili na hrubou sílu.

Závěr

V úvodu této práce jsme si uvedli historii a evoluci šifrovacích kryptosystémů a kryptoanalytických metod souvisejících s diferenciální kryptoanalýzou. Dále jsme si uvedli porovnání standardizovaného kryptosystému Rijndael a její redukované verze. Ukázali jsme si, že tato redukovaná verze je navržená se stejnými principy, jako její originální vzor.

Ve třetí kapitole jsme si představili obecné principy diferenciální kryptoanalýzy. Diferenciální kryptoanalýzu jsme si rozdělili do dvou částí. První z nich je hledání diferenciálních charakteristik. Druhou částí DK je samotný útok s využitím nalezené charakteristiky. Obecně jsme si popsali přípravu hádaných klíčů, jednorundové dešifrování a sestavování žebříčků kandidátních klíčů. Oběma částem jsme následně věnovali samostatné kapitoly.

V kapitole o tvorbě diferenciálních charakteristik jsme si uvedli jejich způsob hledání a jisté optimalizační metody pro jejich hledání. Dále jsme si ukázali, jak se dají charakteristiky porovnávat z pohledu přidělovaných pravděpodobností a uvedli jsme si, jakými způsoby se dají diferenciální charakteristiky sjednocovat a shlukovat. Z těchto poznatků nám vznikly dva klíčové druhy diferenciálních charakteristik. Sjednocené diferenciální charakteristiky se vyznačují pevnou vstupní a výstupní diferencí. Několikanásobné diferenciální charakteristiky jsou pak definované pevnou vstupní diferencí a množinou výstupních diferencí, které mají aktivní bity výstupní diference ve stejných podblocích.

V následující kapitole jsme se věnovali metodám extrakce klíče s využitím dříve sestavených diferenciálních charakteristik. Útok DK je mnohoparametrická úloha, a proto jsme se pokusili jednotlivé parametry od sebe separovat a otestovat jejich vliv na úspěšnost DK. Podrobili jsme testování oba druhy diferenciálních charakteristik, otestovali jsme úspěšnost DK v závislosti na počtu použitých dvojicích OT/ŠT. Dále jsme otestovali úspěšnost DK v závislosti na způsobech tvorby žebříčků kandidátních klíčů. Za nejzajímavější zjištění se dá považovat míra vlivu použitého tajného šifrovacího klíče. V závěru kapitoly jsme si nastínili možnost útoku na generování rundovních klíčů. Budoucí

pokračování práce by jistě měl tento útok blíže prozkoumat.

V poslední kapitole jsme si odvodili paměťovou a výpočetní složitost diferenciální kryptoanalýzy pro obě varianty diferenciálních charakteristik. Z pohledu sjednocených diferenciálních charakteristik jsme zjistili, že v čase lepším, než hrubá síla dosahujeme průměrné úspěšnosti extrakce šifrovacího klíče ve 26 %. U několikanásobných diferenciálních charakteristik jsme bohužel zjistili, že útok s jejich využitím trvá pro otestované varianty v nejlepším případě 1.5-násobek času hrubé síly. Do budoucna by bylo dobré otestovat využití těchto diferenciálních charakteristik pro menší počet použitých dvojic OT/ŠT a využít popisované alternativní procházení žebříčku kandidátních klíčů (od středu do okrajů).

Závěrem by se dalo říci, že se nám nepovedlo šifru Baby Rijndael prolomit pomocí diferenciální kryptoanalýzy do takové míry, jako se to povedlo panu Poljdakovi v [11]. Práce však prověřila šifru z mnoha úhlů a nabízí několik cest, jak na ní navázat.

Literatura

[1] Biham, E.; Shamir, A.: *Differential Cryptanalysis of the Data Encryption Standard*. London, UK, UK: Springer-Verlag, 1993, ISBN 0-387-97930-1.

[2] Coppersmith, D.: *The Data Encryption Standard (DES) and its strength against attacks.*, ročník 38. 1994, 243-250 s. Dostupné z: http://dx.doi.org/10.1147/rd.383.0243

[3] Federal Information Processing: *Announcing the ADVANCED ENCRYPTION STANDARD*. FIPS PUBS, 2001, [cit. 1. 5. 2017]. Dostupné z: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf

[4] Wagner, D.: *The Boomerang Attack*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, ISBN 978-3-540-48519-3, 156–170 s., [cit. 1. 5. 2017]. Dostupné z: http://dx.doi.org/10.1007/3-540-48519-8_12

[5] Biham, E.; Biryukov, A.; Shamir, A.: *Miss in the middle attacks on IDEA and Khufu*. Berlin: Springer-Verlag, 1999, 124–138 s., [cit. 1. 5. 2017]. Dostupné z: https://pdfs.semanticscholar.org/e4ec/f122cf08dabbd93a9abaca461a4a2fec8f90.pdf

[6] Knudsen, L. R.: *Truncated and higher order differentials*. Springer Berlin Heidelberg, 1995, ISBN 978-3-540-47809-6, 196–211 s., [cit. 1. 5. 2017]. Dostupné z: http://dx.doi.org/10.1007/3-540-60590-8_16

[7] Říha, J.: *Konstrukce a kryptoanalýza AES (Advanced Encryption Standard)*. Diplomová práce, MFF UK, 2006, [cit. 1. 5. 2017]. Dostupné z: https://is.cuni.cz/webapps/zzp/download/130011791/?lang=cs

[8] Wrolstad, J.: *A differential cryptanalysis of Baby Rijndael*. Iowa State University, 2009, [cit. 1. 5. 2017]. Dostupné z: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.155.1547&rep=rep1&type=pdf

[9] Kokeš, J.: *Kryptoanalýza šifry Baby Rijndael*. Diplomová práce, FIT ČVUT, 2013, [cit. 1. 5. 2017]. Dostupné z: https://dip.felk.cvut.cz/browse/pdfcache/kokesjo1_2013dipl.pdf

[10] Vábková, L.: *Algebraická kryptoanalýza Baby Rijndael*. Diplomová práce, FIT ČVUT, 2016, [cit. 1. 5. 2017]. Dostupné z: https://dspace.cvut.cz/bitstream/handle/10467/65071/F8-DP-2016-Vabkova-Lenka-thesis.pdf

[11] Poljak, P.: *The Impossible Differential Cryptanalysis*. Diplomová práce, FIT ČVUT, 2017, [cit. 1. 5. 2017]. Dostupné z: https://is.fit.cvut.cz/group/intranet/zp/list#u_poljape1

[12] Daemen, J.; Rijmen, V.: *The design of Rijndael: AES — the Advanced Encryption Standard*. Springer-Verlag, 2002, ISBN 3-540-42580-2, 238 s.

[13] Lórencz, R.: *Symetrická kryptografie*. FIT ČVUT, 2013, [cit. 1. 5. 2017]. Dostupné z: https://edux.fit.cvut.cz/courses/MI-KRY

[14] Bergman, C.: *A Description of Baby Rijndael*. Iowa State University, 2005, [cit. 1. 5. 2017]. Dostupné z: http://www.math.iastate.edu/cbergman/crypto/homework/babyr/babyr.pdf

[15] Daemen, J.; Rijmen, V.: *AES Proposal: Rijndael*. 2003, [cit. 1. 5. 2017]. Dostupné z: http://csrc.nist.gov/archive/aes/rijndael/Rijndael-ammended.pdf

[16] Knudsen, L. R.: *Truncated Differentials (presentation)*. DTU Mathematics, 2011, [cit. 1. 5. 2017]. Dostupné z: https://www.cosic.esat.kuleuven.be/ecrypt/courses/albena11/slides/LRK-truncated_differentials.pdf

[17] Sartori, M.; Lórencz, R.: *Emailová korespondence*. FIT ČVUT, 2013.

Příloha A

Seznam použitých zkratek

LK Lineární kryptoanalýza

DK Diferenciální kryptoanalýza

OT Otevřený text

ŠT Šifrový text

DK Diferenciální kryptoanalýza

AES Advanced Encryption Standard

DES Data Encryption Standard

NIST National Institute of Standards and Technology

NSA National Security Agency

NBA National Bureau of Standards

FPGA Field-programmable gate array, programovatelné hradlové pole

Příloha **B**

Vybrané struktury šifry Rijndael

SubBytes překladová tabulka

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Tabulka B.1: SubBytes překladová tabulka pro 128-bitový Rijndael

a851 → 3007 → d000 → 1600	51a8 → 0730 → 00d0 → 0016
e1a2 → 0f30 → 0005 → 1700	a2e1 → 300f → 0500 → 0017
b6b9 → c002 → b000 → 2c00	b9b6 → 02c0 → 00b0 → 002c
7f73 → 0d80 → 0006 → 2e00	737f → 800d → 0600 → 002e
f373 → 500d → 1000 → 3a00	e5dc → 4001 → 0a00 → 0039
dce5 → 0140 → 000a → 3900	73f3 → 0d50 → 0010 → 003a
c4a2 → b009 → c000 → 4b00	a2c4 → 09b0 → 00c0 → 004b
a8d4 → 0960 → 000e → 4f00	d4a8 → 6009 → 0e00 → 004f
b696 → 02e0 → 000f → 5800	96b6 → e002 → 0f00 → 0058
15d4 → a001 → 3000 → 5d00	d415 → 01a0 → 0030 → 005d
158a → 0370 → 000d → 6100	8a15 → 7003 → 0d00 → 0061
82e5 → 700f → 8000 → 6700	e582 → 0f70 → 0080 → 0067
2a1e → f003 → 5000 → 7100	1e2a → 03f0 → 0050 → 0071
5e28 → 07f0 → 0008 → 7600	285e → f007 → 0800 → 0076
696b → 200e → f000 → 8500	6b69 → 0e20 → 00f0 → 0085
f3f7 → 0850 → 0007 → 8d00	f7f3 → 5008 → 0700 → 008d
5ecd → 1004 → a000 → 9300	b969 → c00e → 0400 → 009a
69b9 → 0ec0 → 0004 → 9a00	cd5e → 0410 → 00a0 → 0093
373f → 05d0 → 0001 → a300	3f37 → d005 → 0100 → 00a3
9b96 → e00c → 4000 → a900	969b → 0ce0 → 0040 → 00a9
2a4c → 0b90 → 000c → b400	4c2a → 900b → 0c00 → 00b4
dc28 → 600b → 2000 → bf00	28dc → 0b60 → 0020 → 00bf
9b6b → 0c20 → 000b → c200	6b9b → 200c → 0b00 → 00c2
e14c → 400a → 9000 → ce00	4ce1 → 0a40 → 0090 → 00ce
4d51 → 0a10 → 0003 → d500	514d → 100a → 0300 → 00d5
7f3f → 8005 → 7000 → d800	3f7f → 0580 → 0070 → 00d8
37f7 → d008 → 6000 → e200	f737 → 08d0 → 0060 → 00e2
c41e → 04a0 → 0009 → ec00	1ec4 → a004 → 0900 → 00ec
4d8a → 9006 → e000 → f400	8a4d → 0690 → 00e0 → 00f4
82cd → 06b0 → 0002 → fb00	cd82 → b006 → 0200 → 00fb

Tabulka B.2: Základní diferenční charakteristiky pro 4-rundový Baby Rijndael. Pravděpodobnost každé z nich je 2^{-14}

Příloha C

Obsah přiloženého CD

- `readme.txt` ..... stručný popis obsahu CD
- `src`
  - `impl` ..... zdrojové kódy
  - `thesis` ..... zdrojová forma práce ve formátu LaTeX
- `text` ..... text práce
  - `DP_Jakub_Tomanek_2017.pdf` ..... text práce ve formátu PDF
  - `ZZP_DP_Jakub_Tomanek_2017.pdf` ..... zadání práce ve formátu PDF.