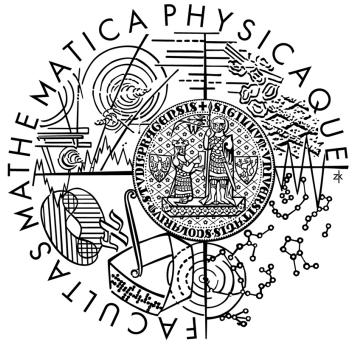
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Ondřej Väter

Triangulační algoritmus pro systémy nelineárních rovnic

Katedra algebry

Vedoucí bakalářské práce: RNDr. Michal Hojsík, Ph.D.

Studijní program: Matematika

Studijní obor: MOM

Praha 2012

Rád bych poděkoval svému vedoucímu RNDr. Michalu Hojsíkovi, Ph.D. za jeho přínosné rady k podobě této práce a postupům jak tuto práci zhotovit. Děkuji za čas, který obětoval na čtení této práce a na konzultace. Oceňuji zkušenosti s vypracováním odborné práce, které jsem díky němu získal.

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V Praze dne ..... Podpis autora

Název práce: Triangulační algoritmus pro systémy nelineárních rovnic

Autor: Ondřej Väter

Katedra: Katedra algebry

Vedoucí bakalářské práce: RNDr. Michal Hojsík, Ph.D.

Abstrakt: Tato práce se zabývá triangulačním algoritmem a jeho využitím v kryptoanalýze. Uvedeme si definici soustavy nelineárních rovnic, na kterou můžeme aplikovat trian. alg., a objasníme si co je výstupem trian. alg. Ukážeme si použití tohoto algoritmu v kryptoanalýze, konkrétně při útoku na šifru Rijndael. Tento útok si ilustrujeme při hledání kolize pro námi vytvořenou hashovací funkci v Davies-Mayerově módu za použití šifry Rijndael. Součástí této práce je implementační část, ve které si ukážeme reálné využití trian. alg. při hledání kolize pro výše zmíněnou hashovací funkci.

Klíčová slova: triangulační algoritmus, nelineární systém rovnic, kryptoanalýza

Title: Triangulation algorithm for non-linear equation systems

Author: Ondřej Väter

Department: Department of Algebra

Supervisor: RNDr. Michal Hojsík, Ph.D.

Abstract: The topic of this thesis is a triangulation algorithm and its use in cryptanalysis. First of all we will define a non-linear equation system on which we can apply triangulation algorithm and we will explain what its output is. Then we will demonstrate its application in cryptanalysis, more specificaly during the attack on the Rinjdael cifer. We will illustrate this attack by a search of collision for our hash function, created for this purpose in Davies-Mayer mode using Rijndael cipher This thesis also contains a practical part in which we will demonstrate the search of collision for our hash function mention before.

Keywords: triangulation algorithm, non-linear equation systems, cryptanalysis

Obsah

Úvod

V této práci si popíšeme triangulační algoritmus a ukážeme využití tohoto algoritmu v kryptoanalýze. Tento algoritmus je velmi jednoduchý. Jeho myšlenka je chytré přeuspořádání rovnic v nelineární soustavě rovnic. Konkrétně si ilustrujeme využití tohoto algoritmu v útoku na šifru Rijndael, která je v dnešní době nejpoužívanější blokovou šifrou a také je standardizovaná jako AES. Budeme se snažit pomocí triangulačního algoritmu hledat dva klíče, které šifrují libovolný pevný otevřený text na stejný šifrový. Díky tomuto poznatku budeme moci hledat kolize pro hashovací funkci, která vznikne použitím šifry Rijndael v Davies-Mayerově módu. Tedy díky velmi jednoduchému algoritmu dovedeme nalézt kolizi pro hashovací funkci. Součástí této práce je také implementační část, ve které pomocí triangulačního algoritmu najdeme kolizní zprávy pro tuto hashovací funkci.

K hledání dvou výše zmíněných klíčů budeme používat diferenční stopu pro šifru Rijndael s nulovou diferencí otevřeného a šifrového textu a s nenulovou diferencí klíče. Nejdříve budeme hledat takovouto diferenční stopu. Následně budeme hledat klíč, který by s pevně zvoleným otevřeným textem splnil zadanou diferenční stopu. Z hledání takovéhoto klíče vzejdou podmínky, které povedou na nelineární soustavu rovnic, kterou budeme řešit pomocí triangulačního algoritmu. Objasníme si, že takto vytvořenou nelineární soustavu rovnic lze opravdu řešit triangulačním algoritmem a ilustrujeme si toto na konkrétním příkladě.

Struktura této práce je následující. V první kapitole si definujeme pojmy, se kterými budeme pracovat. V druhé kapitole se budeme zabývat diferenční stopou, ukážeme jak vytvořit diferenční stopy s neznámou platností, jak se šíří diference skrz zobrazení, se kterými pracujeme, a vysvětlíme si co znamená ověřit platnost diferenční stopy. V třetí kapitole si popíšeme triangulační algoritmus a ukážeme, jak sestavit nelineární soustavu rovnic, která reprezentuje průběh šifrování šifrou Rijndael a na závěr třetí kapitoly využijeme již získaných poznatků k popisu postupu jak ověřit diferenční stopu. Ve čtvrté kapitole provedeme ukázku ověření platnosti diferenční stopy pro pevnou konfiguraci šifry Rijndael za použití implementace triangulační algoritmu. V páté kapitole si zadefinujeme hashovací funkci v Davies-Mayerově módu a ukážeme, jak pomocí získaních znalostí vytvořit postup k nalezení kolize pro tuto hashovací funkci. V šesté kapitole si objasníme k čemu přesně slouží programy, které jsou součástí této práce.

V této práci převážně navazujeme na článek [1] a snažíme se vysvětlit myšlenky z tohoto článku do detailů a předvést jejich důkazy.

1. Základní pojmy

V této kapitole uvedeme základní pojmy a důležité informace, které budeme potřebovat v této práci. Vycházíme z literatury [2] a [3].

1.1 Kryptosystém a hashovací funkce

Definice 1.1. *Nechť kryptosystém $\mathcal{S}$ je uspořádaná pětice $\mathcal{S} = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, kde:*

1. *$\mathcal{P}$ je konečná množina otevřených textů*
2. *$\mathcal{C}$ je konečná množina šifrových textů*
3. *$\mathcal{K}$ je konečná množina klíčů*
4. *$\mathcal{E}$ je množina zobrazení $\mathcal{E} = \{e_k : \mathcal{P} \to \mathcal{C}; k \in \mathcal{K}\}$*
5. *$\mathcal{D}$ je množina zobrazení $\mathcal{D} = \{d_k : \mathcal{C} \to \mathcal{P}; k \in \mathcal{K}\}$*
6. *platí $\forall k \in \mathcal{K}, \forall x \in \mathcal{P} : d_k(e_k(x)) = x$*

Definice 1.2. *Nechť $\mathcal{S} = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ je kryptosystém a $n \in \mathbb{N}$. $\mathcal{S}$ nazveme blokovou šifrou pokud pro $x \in \mathcal{P}^n$, $x = (x_1, \ldots, x_n)$ a $k \in \mathcal{K}$ platí, že $\overline{e_k}(x) = (e_k(x_1), \ldots, e_k(x_n)) = y \in \mathcal{C}^n$. Tedy na každý prvek z $\mathcal{P}$ použijeme stejné zobrazení e_k.*

Definice 1.3. *Nechť $\mathcal{S} = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ je bloková šifra, $r \in \mathbb{N}$, $\mathcal{P} = \mathcal{C} =: X$, $\rho : X \to X$ je zobrazení a $\delta[j] : X \to X$ je zobrazení, kde J je množina všech možných parametrů a $j \in J$ je parametr zobrazení. Nechť existuje zobrazení $\mathcal{KS} : \mathcal{K} \times \{0, \ldots, r\} \to J$. Označme $\forall k \in \mathcal{K}$ a $\forall i \in \{0, \ldots, r\}$ $k_i := \mathcal{KS}(k, i) \in J$.*

Potom pokud $\forall k \in \mathcal{K}$ platí:

$$e_k = \delta[k_r] \circ \rho \circ \delta[k_{r-1}] \circ \rho \circ \ldots \circ \rho \circ \delta[k_0]$$

říkáme, že $\mathcal{S}$ je klíč iterující bloková šifra.

Poznámka 1.4. *Pro některé objekty z definice 1.3 se obvykle používají názvy, které budeme používat také:*

- *zobrazení $\mathcal{KS}$ se nazývá Key Schedule*
- *číslu r se říká počet rundovních zobrazení*
- *parametry k_i pro $i \in \{0, \ldots, r\}$ se nazývají rundovní klíče*
- *zobrazení $\delta[k_i] \circ \rho$ se nazývá rundovní zobrazení pro $i \in \{1, \ldots, r\}$*

Nyní si zadefinujeme základní pojmy týkající se hashovacích funkcí. Některé definice by bylo možné zobecnit, ale pro naše účely budou stačit ve znění, v kterém si je definujeme.

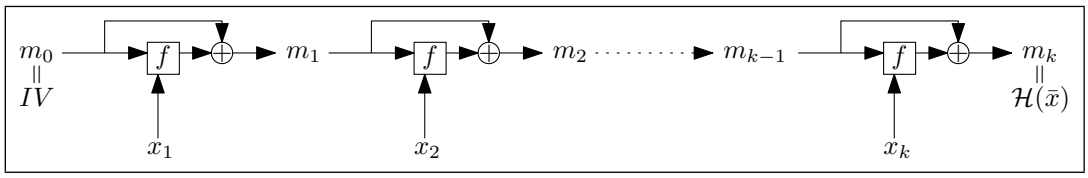
Definice 1.5. *Nechť $M \subseteq \mathbb{N}$ definujeme množinu $\mathcal{B}_M := \bigcup_{i \in M} \{0,1\}^i$, tedy $\mathcal{B}_M$ je množina všech posloupností bitů délek j, $\forall j \in M$. Definujeme ještě $\mathcal{B}_i := \{0,1\}^i$ pro $i \in \mathbb{N}$, tedy množinu všech posloupností bitů délky i.*

Definice 1.6. *Nechť $M \subseteq \mathbb{N}$ a $d \in \mathbb{N}$. Platí, že $\max M \gg d$. Potom funkci $\mathcal{H} : \mathcal{B}_M \to \mathcal{B}_d$, budeme nazývat hashovací funkce. $m \in \mathcal{B}_M$ budeme označovat jako zprávu a $\mathcal{H}(m)$ jako hodnotu hash zprávy m pro hashovací funkci $\mathcal{H}$ nebo také jako digitální otisk zprávy m.*

Definice 1.7. *Nechť $n, m \in \mathbb{N}$, f je zobrazení $f : \mathcal{B}_n \times \mathcal{B}_m \to \mathcal{B}_n$ a $IV \in \mathcal{B}_n$. Definujme množinu $M = \{m \cdot n;\, n \in \mathbb{N}\}$. Potom pro $\bar{x} = (x_1, \ldots, x_k) \in \mathcal{B}_M$, kde $k \in \mathbb{N}$, délka posloupnosti $\bar{x}$ je $k \cdot m$ bitů a $\forall i \in \{1, \ldots, k\}$ je $x_i \in \mathcal{B}_m$. Definujme rekurzivně hodnoty:*

- $m_0 := IV \in \mathcal{B}_n$
- $m_i := f(m_{i-1}, x_i) + m_{i-1} \in \mathcal{B}_n$ *pro* $i \in \{1, \ldots, k\}$, *POZN: sčítáním máme na mysli sčítání v* ${\mathbb{F}_2}^n$

Definujme hashovací funkci $\mathcal{H} : \mathcal{B}_M \to \mathcal{B}_n$ předpisem $\mathcal{H}(\bar{x}) := m_k$. Takto zadefinovanou hashovací funkci budeme nazývat hashovací funkci v Davies-Mayerově módu pro zobrazení f. Tuto definici si budeme ilustrovat na obrázku 1.1.



Obrázek 1.1: Hashovací funkce v Davies-Mayerově módu pro zobrazení f

Definice 1.8. *Nechť $m \in \mathbb{N}$. Definujme množinu $M = \{m \cdot n;\, n \in \mathbb{N}\}$. Potom algoritmus, jak libovolné $x \in \mathcal{B}_{\mathbb{N}}$ prodloužit na x', aby platilo $x' \in \mathcal{B}_M$, nazveme padding. Takovýto algoritmus například používáme, pokud chceme rozšířit definici hashovací funkce z $\mathcal{H} : \mathcal{B}_M \to \mathcal{B}_n$ na $\mathcal{H}' : \mathcal{B}_{\mathbb{N}} \to \mathcal{B}_n$, tak, že libovolnou posloupnost prodloužíme na patřičnou délku násobku m a použijeme hashovací funkci $\mathcal{H}$.*

Definice 1.9. *Nechť máme $M \subseteq \mathbb{N}$, $d \in \mathbb{N}$ a hashovaci funkci $\mathcal{H} : \mathcal{B}_M \to \mathcal{B}_d$. Definujeme kolizi pro hashovací funkci $\mathcal{H}$ jako dvojici zpráv $m_1, m_2 \in \mathcal{B}_M$, pro které platí, že $m_1 \neq m_2$ a zároveň $\mathcal{H}(m_1) = \mathcal{H}(m_2)$. Jedná se tedy o dvě zprávy, které mají stejnou hash hodnotu pro hashovací funkci $\mathcal{H}$.*

1.2 Propagace diference

Definice 1.10. *Nechť $\mathcal{G}$ a $\mathcal{H}$ jsou konečné abelovské grupy s aditivní notací, zobrazení $S : \mathcal{G} \to \mathcal{H}$, $a \in \mathcal{G}$, $b \in \mathcal{H}$. Pak definujeme:*

- *množinu vstupních hodnot vyhovujících diferenčnímu páru a, b skrz zobrazení S jako $M_S(a, b) = \{x \in \mathcal{G}; b = S(x + a) - S(x)\}$*

- *pravděpodobnost šíření diference a na b skrz zobrazení S následujícím předpisem:* $D_pS(a,b) = \dfrac{|M_S(a,b)|}{|\mathcal{G}|}$
- *a, b jako nekompatibilní diferenční pár zobrazení S, pokud platí* $D_pS(a,b) = 0$
- *a, b jako kompatibilní diferenční pár zobrazení S, pokud platí* $D_pS(a,b) > 0$
- *a, b jako jistý diferenční pár zobrazení S, pokud platí* $D_pS(a,b) = 1$
- *matici propagace diference zobrazení S* $\mathbf{D}_S \in \mathbb{R}^{|\mathcal{G}|\times|\mathcal{H}|}$. *Její prvky indexujeme prvky grup $\mathcal{G}$ a $\mathcal{H}$ s hodnotami* $\mathbf{d}_{a\in\mathcal{G},b\in\mathcal{H}} = D_pS(a,b)$.

Poznámka 1.11. *Nechť $\mathcal{G}$ a $\mathcal{H}$ jsou konečné abelovské grupy s aditivní notací, zobrazení $S: \mathcal{G} \to \mathcal{H}$, $a \in \mathcal{G}$, $b \in \mathcal{H}$.*

1. *$\forall a \in \mathcal{G}$ platí:*
$$\sum_{b\in\mathcal{H}} D_pS(a,b) = 1$$
protože
$$\sum_{b\in\mathcal{H}} D_pS(a,b) = \frac{1}{|\mathcal{G}|}\sum_{b\in\mathcal{H}} |\{x \in \mathcal{G}; b = S(x+a) - S(x)\}| = \frac{|\mathcal{G}|}{|\mathcal{G}|} = 1$$
2. *$D_pS(0,0) = 1$, protože $|\{x \in \mathcal{G}; 0 = S(x+0) - S(x)\}| = |\mathcal{G}|$. Důsledkem je, že $\forall b \in \mathcal{H}$, $b \neq 0$ platí, že $D_pS(0,b) = 0$.*
3. *S je homomorfismus grup, potom $D_pS(a,S(a)) = 1$, protože $S(x+a) - S(x) = S(x) + S(a) - S(x) = S(a)$. Tedy a, $S(a)$ je jistý diferenční pár zobrazení S.*
4. *S je zobrazení tvaru $S(x) = L(x) + c$, kde $c \in \mathcal{H}$ je konstanta a $L : \mathcal{G} \to \mathcal{H}$ je homomorfismus grup. Pak $D_pS(a,L(a)) = 1$, protože $S(x+a) - S(x) = L(x+a) + c - L(x) - c = L(x) + L(a) - L(x) = L(a)$. Tedy a, $L(a)$ je jistý diferenční pár zobrazení S.*
5. *S je bijekce, potom $\forall a \in \mathcal{G}$, $a \neq 0$ platí, že $D_pS(a,0) = 0$, protože $\forall x \in \mathcal{G}$ platí $S(x+a) \neq S(x)$.*

Úmluva 1.12. *Po zbytek této práce budeme zapisovat permutace na konečné množině v základním tvaru. Viz příklad: Nechť $n \in \mathbb{N}$ a definujme množinu $X := \{1,\ldots,n\}$ a nechť Π je permutace množiny X. Potom permutaci Π zapíšeme:*
$$\Pi = \begin{pmatrix} 1 & 2 & \ldots & n \\ \Pi(1) & \Pi(2) & \ldots & \Pi(n) \end{pmatrix}$$

Příklad 1.13. *Nechť máme zobrazení* $S = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 4 & 3 & 1 & 2 & 0 \end{pmatrix}$ *(značení dle úmluvy 1.12) definované nad tělesem $\mathbb{F}_5$, pak matici propagace diference zobrazení S je*

$$\mathbf{D}_S = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{5} & 0 & \frac{2}{5} & \frac{2}{5} \\ 0 & 0 & \frac{2}{5} & \frac{1}{5} & \frac{2}{5} \\ 0 & \frac{2}{5} & \frac{1}{5} & \frac{2}{5} & 0 \\ 0 & \frac{2}{5} & \frac{2}{5} & 0 & \frac{1}{5} \end{pmatrix} \end{array}$$

1.3 Šifra Rijndael

Rijndael je klíč iterující bloková šifra s následujícími vlastnostmi:

- množina $\mathcal{P}$ všech otevřených textů je $\mathcal{B}_n$ a množina $\mathcal{C}$ všech šifrových textů je také $\mathcal{B}_n$ pro $n \in \mathbb{N}$
- $\mathcal{K}$ množina všech klíčů je $\mathcal{B}_m$ pro $m \in \mathbb{N}$

Hodnoty m a n mohou být voleny na sobě nezávisle z této množiny $\{128, 160, 192, 224, 256\}$. Na posloupnosti bitů můžeme také nahlížet jako na posloupnosti bytů.

1.3.1 Těleso $\mathbb{F}_{256}$

Jeden byte je posloupnost 8 bitů: $a_7, a_6, \ldots, a_0$. Na tyto bity můžeme nahlížet jako na koeficienty polynomu nad $\mathbb{F}_2$ tedy: $A(x) := a_7 \cdot x^7 + a_6 \cdot x^6 + \ldots + a_0$. Tedy existuje jednoznačná korespondence mezi polynomy nad $\mathbb{F}_2$ stupně menšího než 8 a byty.

Zvolme polynom $m(x) := x^8 + x^4 + x^3 + x + 1$, tento polynom je ireducibilní nad $\mathbb{F}_2$. Tedy rozkladové rozšíření $\mathbb{F}_2$ dané polynomem $m(x)$ je 256 prvkové těleso, jehož prvky lze reprezentovat jako polynomy stupně menšího než 8. Toto těleso budeme po zbytek této práce značit jako $\mathbb{F}_{256}$. Toto těleso má charakteristiku 2, tedy $-1 \equiv 1$.

Jeden byte lze zapsat v hexadecimálním zápisu (př.: $0x5F$). Jak takovýto zápis vznikne si ukážeme na příkladu:

- jeden byte zapsaný jako posloupnost osmi bitů: 10110110
- první čtveřice bitů: $(1011)_2 = (11)_{10} = (B)_{16}$
- druhá čtveřice bitů: $(0110)_2 = (6)_{10} = (6)_{16}$
- tedy posloupnost bitů 10110110 zapíšeme v hexadecimálním zápisu $0xB6$

Na množinu všech bytů lze tedy pohlížet jako na $\mathbb{F}_{256}$ a pro prvky $\mathbb{F}_{256}$ používat hexadecimální zápis. Toto značení a zápis budeme používat po zbytek celé této práce.

Je dobré si připomenout, že rozkladové rozšíření má strukturu vektorového prostoru nad svým podtělesem a platí, že operace sčítání v rozkladovém rozšíření je isomorfní s operací sčítání v tomto vektorovém prostoru. Tedy v našem případě je operace sčítání v $\mathbb{F}_{256}$ isomorfní s operací sčítání v ${\mathbb{F}_2}^8$

1.3.2 Maticová reprezentace klíče a otevřeného a šifrového textu

Na posloupnost bytů můžeme nahlížet také jako na posloupnost prvků tělesa $\mathbb{F}_{256}$. Všimněme si, že hodnoty, kterých mohou nabývat parametry n a m jsou dělitelné 32. Tedy posloupnosti prvků $\mathbb{F}_{256}$ reprezentující otevřený a šifrový text a klíč lze zapsat do matic s rozměry $4 \times \frac{n}{32}$ pro otevřený a šifrový text a s rozměry $4 \times \frac{m}{32}$ pro klíč.

Ukážeme si jak tyto matice vzniknou. Máme $a_0, a_1, \ldots, a_{\frac{n}{8}-1}$ posloupnost prvků z $\mathbb{F}_{256}$. Do matice je poskládáme takto:

$$\begin{pmatrix} a_0 & a_4 & \ldots & a_{\frac{n}{8}-4} \\ a_1 & a_5 & \ldots & a_{\frac{n}{8}-3} \\ a_2 & a_6 & \ldots & a_{\frac{n}{8}-2} \\ a_3 & a_7 & \ldots & a_{\frac{n}{8}-1} \end{pmatrix}$$

Obdobně pro posloupnost reprezentující klíč.

Definujeme si parametry $N_b := \frac{n}{32}$ a $N_k := \frac{m}{32}$. Toto značení budeme používat po celou tuto práci. Potom na otevřený a šifrový text můžeme nahlížet jako na matici nad $\mathbb{F}_{256}$, tedy: $\mathcal{P} = \mathcal{C} = {\mathbb{F}_{256}}^{4\times N_b}$ a obdobně pro množinu všech klíčů, tedy: $\mathcal{K} = {\mathbb{F}_{256}}^{4\times N_k}$. Pro upřesnění parametry N_b a N_k mohou být voleny na sobě nezávisle z množiny $\{4, 5, 6, 8\}$.

1.3.3 Sub Bytes

Definujeme permutaci S tělesa $\mathbb{F}_{256}$. S definujeme tabulkou 1.1. Významnou vlastností permutace S je, že $max\{D_pS(a,b); a, b \in \mathbb{F}_{256} \wedge ab \neq 0\} = 2^{-6}$. Tedy pokud si zvolím libovolné nenulové $a, b \in \mathbb{F}_{256}$, potom množina $M_S(a, b)$ má maximálně 4 prvky. Po zbytek této práce, pokud budeme mluvit o permutaci S, máme vždy na mysli tuto permutaci definovanou tabulkou 1.1.

Sub Bytes je zobrazení ${\mathbb{F}_{256}}^{4\times N_b} \to {\mathbb{F}_{256}}^{4\times N_b}$ definované tak, že každý prvek vstupní matice se zobrazí permutací S. Po zbytek této práce budeme používat pro Sub Bytes značení $\mathcal{SB}$.

$$\mathcal{SB} : \begin{pmatrix} a_{0,0} & a_{0,1} & \ldots & a_{0,N_b-1} \\ a_{1,0} & a_{1,1} & \ldots & a_{1,N_b-1} \\ a_{2,0} & a_{2,1} & \ldots & a_{2,N_b-1} \\ a_{3,0} & a_{3,1} & \ldots & a_{3,N_b-1} \end{pmatrix} \mapsto \begin{pmatrix} S(a_{0,0}) & S(a_{0,1}) & \ldots & S(a_{0,N_b-1}) \\ S(a_{1,0}) & S(a_{1,1}) & \ldots & S(a_{1,N_b-1}) \\ S(a_{2,0}) & S(a_{2,1}) & \ldots & S(a_{2,N_b-1}) \\ S(a_{3,0}) & S(a_{3,1}) & \ldots & S(a_{3,N_b-1}) \end{pmatrix}$$

Definice Sub Bytes lze rozšířit na: $\mathcal{SB} : {\mathbb{F}_{256}}^{n\times m} \to {\mathbb{F}_{256}}^{n\times m}$ pro libovolné $n, m \in \mathbb{N}$ předpisem:

$$\mathcal{SB} : \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,m} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,m} \end{pmatrix} \mapsto \begin{pmatrix} S(a_{1,1}) & S(a_{1,2}) & \cdots & S(a_{1,m}) \\ s(a_{2,1}) & S(a_{2,2}) & \cdots & s(a_{2,m}) \\ \vdots & \vdots & \ddots & \vdots \\ S(a_{n,1}) & S(a_{n,2}) & \cdots & S(a_{n,m}) \end{pmatrix}$$

1.3.4 Shift Rows

Shift Rows je zobrazení ${\mathbb{F}_{256}}^{4\times N_b} \to {\mathbb{F}_{256}}^{4\times N_b}$ definované tak, že každý řádek vstupní matice se cyklicky posune doleva. Velikost posunu jednotlivých řádků je závislá na hodnotě N_b. Tato závislost je definovaná tabulkou 1.2, kde c_0, c_1, c_2, c_3

Vstup	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
Výstup	0x63	0x7C	0x77	0x7B	0xF2	0x6B	0x6F	0xC5	0x30	0x01	0x67	0x2B	0xFE	0xD7	0xAB	0x76
Vstup	0x10	0x11	0x12	0x13	0x14	0x15	0x16	0x17	0x18	0x19	0x1A	0x1B	0x1C	0x1D	0x1E	0x1F
Výstup	0xCA	0x82	0xC9	0x7D	0xFA	0x59	0x47	0xF0	0xAD	0xD4	0xA2	0xAF	0x9C	0xA4	0x72	0xC0
Vstup	0x20	0x21	0x22	0x23	0x24	0x25	0x26	0x27	0x28	0x29	0x2A	0x2B	0x2C	0x2D	0x2E	0x2F
Výstup	0xB7	0xFD	0x93	0x26	0x36	0x3F	0xF7	0xCC	0x34	0xA5	0xE5	0xF1	0x71	0xD8	0x31	0x15
Vstup	0x30	0x31	0x32	0x33	0x34	0x35	0x36	0x37	0x38	0x39	0x3A	0x3B	0x3C	0x3D	0x3E	0x3F
Výstup	0x04	0xC7	0x23	0xC3	0x18	0x96	0x05	0x9A	0x07	0x12	0x80	0xE2	0xEB	0x27	0xB2	0x75
Vstup	0x40	0x41	0x42	0x43	0x44	0x45	0x46	0x47	0x48	0x49	0x4A	0x4B	0x4C	0x4D	0x4E	0x4F
Výstup	0x09	0x83	0x2C	0x1A	0x1B	0x6E	0x5A	0xA0	0x52	0x3B	0xD6	0xB3	0x29	0xE3	0x2F	0x84
Vstup	0x50	0x51	0x52	0x53	0x54	0x55	0x56	0x57	0x58	0x59	0x5A	0x5B	0x5C	0x5D	0x5E	0x5F
Výstup	0x53	0xD1	0x00	0xED	0x20	0xFC	0xB1	0x5B	0x6A	0xCB	0xBE	0x39	0x4A	0x4C	0x58	0xCF
Vstup	0x60	0x61	0x62	0x63	0x64	0x65	0x66	0x67	0x68	0x69	0x6A	0x6B	0x6C	0x6D	0x6E	0x6F
Výstup	0xD0	0xEF	0xAA	0xFB	0x43	0x4D	0x33	0x85	0x45	0xF9	0x02	0x7F	0x50	0x3C	0x9F	0xA8
Vstup	0x70	0x71	0x72	0x73	0x74	0x75	0x76	0x77	0x78	0x79	0x7A	0x7B	0x7C	0x7D	0x7E	0x7F
Výstup	0x51	0xA3	0x40	0x8F	0x92	0x9D	0x38	0xF5	0xBC	0xB6	0xDA	0x21	0x10	0xFF	0xF3	0xD2
Vstup	0x80	0x81	0x82	0x83	0x84	0x85	0x86	0x87	0x88	0x89	0x8A	0x8B	0x8C	0x8D	0x8E	0x8F
Výstup	0xCD	0x0C	0x13	0xEC	0x5F	0x97	0x44	0x17	0xC4	0xA7	0x7E	0x3D	0x64	0x5D	0x19	0x73
Vstup	0x90	0x91	0x92	0x93	0x94	0x95	0x96	0x97	0x98	0x99	0x9A	0x9B	0x9C	0x9D	0x9E	0x9F
Výstup	0x60	0x81	0x4F	0xDC	0x22	0x2A	0x90	0x88	0x46	0xEE	0xB8	0x14	0xDE	0x5E	0x0B	0xDB
Vstup	0xA0	0xA1	0xA2	0xA3	0xA4	0xA5	0xA6	0xA7	0xA8	0xA9	0xAA	0xAB	0xAC	0xAD	0xAE	0xAF
Výstup	0xE0	0x32	0x3A	0x0A	0x49	0x06	0x24	0x5C	0xC2	0xD3	0xAC	0x62	0x91	0x95	0xE4	0x79
Vstup	0xB0	0xB1	0xB2	0xB3	0xB4	0xB5	0xB6	0xB7	0xB8	0xB9	0xBA	0xBB	0xBC	0xBD	0xBE	0xBF
Výstup	0xE7	0xC8	0x37	0x6D	0x8D	0xD5	0x4E	0xA9	0x6C	0x56	0xF4	0xEA	0x65	0x7A	0xAE	0x08
Vstup	0xC0	0xC1	0xC2	0xC3	0xC4	0xC5	0xC6	0xC7	0xC8	0xC9	0xCA	0xCB	0xCC	0xCD	0xCE	0xCF
Výstup	0xBA	0x78	0x25	0x2E	0x1C	0xA6	0xB4	0xC6	0xE8	0xDD	0x74	0x1F	0x4B	0xBD	0x8B	0x8A
Vstup	0xD0	0xD1	0xD2	0xD3	0xD4	0xD5	0xD6	0xD7	0xD8	0xD9	0xDA	0xDB	0xDC	0xDD	0xDE	0xDF
Výstup	0x70	0x3E	0xB5	0x66	0x48	0x03	0xF6	0x0E	0x61	0x35	0x57	0xB9	0x86	0xC1	0x1D	0x9E
Vstup	0xE0	0xE1	0xE2	0xE3	0xE4	0xE5	0xE6	0xE7	0xE8	0xE9	0xEA	0xEB	0xEC	0xED	0xEE	0xEF
Výstup	0xE1	0xF8	0x98	0x11	0x69	0xD9	0x8E	0x94	0x9B	0x1E	0x87	0xE9	0xCE	0x55	0x28	0xDF
Vstup	0xF0	0xF1	0xF2	0xF3	0xF4	0xF5	0xF6	0xF7	0xF8	0xF9	0xFA	0xFB	0xFC	0xFD	0xFE	0xFF
Výstup	0x8C	0xA1	0x89	0x0D	0xBF	0xE6	0x42	0x68	0x41	0x99	0x2D	0x0F	0xB0	0x54	0xBB	0x16

Tabulka 1.1: Permutace S definovaná tabulkou

je velikost posunu příslušného řádku. Pro zobrazení Shift Rows budeme po zbytek této práce používat značení $\mathcal{SR}$. Shift Rows je lineární zobrazení, tedy pro libovolné $A, B \in {\mathbb{F}_{256}}^{4\times N_b}$ a libovolné $r \in \mathbb{F}_{256}$ platí:

- $\mathcal{SR}(A + B) = \mathcal{SR}(A) + \mathcal{SR}(B)$
- $\mathcal{SR}(r \cdot A) = r \cdot \mathcal{SR}(A)$

$$\mathcal{SR} : \begin{pmatrix} m_{0,0} & \dots & m_{0,N_b-1} \\ m_{1,0} & \dots & m_{1,N_b-1} \\ m_{2,0} & \dots & m_{2,N_b-1} \\ m_{3,0} & \dots & m_{3,N_b-1} \end{pmatrix} \mapsto \begin{pmatrix} m_{0,0+c_0 mod N_b} & \dots & m_{0,(N_b-1+c_0) mod N_b} \\ m_{1,0+c_1 mod N_b} & \dots & m_{1,(N_b-1+c_1) mod N_b} \\ m_{2,0+c_2 mod N_b} & \dots & m_{2,(N_b-1+c_2) mod N_b} \\ m_{3,0+c_3 mod N_b} & \dots & m_{3,(N_b-1+c_3) mod N_b} \end{pmatrix}$$

N_b	c_0	c_1	c_2	c_3
4	0	1	2	3
5	0	1	2	3
6	0	1	2	3
7	0	1	2	4
8	0	1	3	4

Tabulka 1.2: Velikost posunů jednotlivých řádků pro zobrazení $\mathcal{SR}$

1.3.5 Mix Columns

Mix Columns je zobrazení ${\mathbb{F}_{256}}^{4\times N_b} \to {\mathbb{F}_{256}}^{4\times N_b}$ definované tak, že vynásobíme zleva vstupní matici maticí A. Matice A je definovaná takto:

$$A := \begin{pmatrix} 0x02 & 0x03 & 0x01 & 0x01 \\ 0x01 & 0x02 & 0x03 & 0x01 \\ 0x01 & 0x01 & 0x02 & 0x03 \\ 0x03 & 0x01 & 0x01 & 0x02 \end{pmatrix}$$

Tedy zobrazení Mix Columns je lineární. Pro zbytek této práce budeme Mix Columns značit $\mathcal{MC}$.

$$\mathcal{MC} : \begin{pmatrix} m_{0,0} & \dots & m_{0,N_b-1} \\ m_{1,0} & \dots & m_{1,N_b-1} \\ m_{2,0} & \dots & m_{2,N_b-1} \\ m_{3,0} & \dots & m_{3,N_b-1} \end{pmatrix} \mapsto A \cdot \begin{pmatrix} m_{0,0} & \dots & m_{0,N_b-1} \\ m_{1,0} & \dots & m_{1,N_b-1} \\ m_{2,0} & \dots & m_{2,N_b-1} \\ m_{3,0} & \dots & m_{3,N_b-1} \end{pmatrix}$$

1.3.6 Add Round Key

Add Round Key je zobrazení ${\mathbb{F}_{256}}^{4\times N_b} \times {\mathbb{F}_{256}}^{4\times N_b} \to {\mathbb{F}_{256}}^{4\times N_b}$ definované tak, že sečteme dvě vstupní matice. Po zbytek této práce budeme používat značení $\mathcal{AK}[K_i] : {\mathbb{F}_{256}}^{4\times N_b} \to {\mathbb{F}_{256}}^{4\times N_b}$, kde $K_i \in {\mathbb{F}_{256}}^{4\times N_b}$ je i-tý rundovní klíč vygenerovaný Key Schedulem (kapitola 1.3.9). Toto značení rundovních klíčů budeme používat po zbytek této práce, pokud nebude řečeno jinak.
$\mathcal{AK}[K_i] : M \mapsto K_i + M$ pro $M \in {\mathbb{F}_{256}}^{4\times N_b}$

1.3.7 Konstanta $RC(n)$

Na prvky $\mathbb{F}_{256}$ můžeme nahlížet jako na polynomy stupně menšího než 8 (viz kapitola 1.3.1). Definujeme konstanty $RC(n) := x^{n-1}$ pro $n \in \mathbb{N}$. Připomeňme, že x lze brát jako prvek $\mathbb{F}_{256}$ a tedy v definici $RC(n)$ se jedná o násobení v $\mathbb{F}_{256}$. Pro konstanty $RC(n)$ platí:

- $RC(1) = 0x01$
- $RC(2) = 0x02$
- pro $n > 2$ platí $RC(n) = x \cdot RC(n-1)$ (pozn: jedná se o násobení v $\mathbb{F}_{256}$)

Konstanty $RC(n)$ se používají v Key Schedulu 1.3.9.

1.3.8 Počet rund

Na začátku této kapitoly 1.3 jsme zadefinovali šifru Rijndael jako klíč iterující blokovou šifru. Tedy $\forall k \in \mathcal{K}$ platí, že $e_k = \delta[k_{N_r}] \circ \rho \circ \delta[k_{N_r-1}] \circ \rho \circ \ldots \circ \rho \circ \delta[k_0]$ pro nějaké zobrazení $\rho, \delta[k_i] : {\mathbb{F}_{256}}^{4\times N_b} \to {\mathbb{F}_{256}}^{4\times N_b}$. Dle poznámky 1.4 budeme po zbytek této práce hodnotou N_r označovat počet rundovních zobrazeních. Hodnota N_r je závislá na hodnotách N_b a N_k. Tato závislost je určená tabulkou 1.3.

1.3.9 Key Schedule

V průběhu šifrování je potřeba $N_r + 1$ rundovních klíčů. Jeden rundovní klíč je matice s rozměry $4 \times N_b$ nad $\mathbb{F}_{256}$ (kapitola 1.3.6). Označme matici $K_i \in {\mathbb{F}_{256}}^{4\times N_b}$ jako i-tý rundovní klíč pro $i \in \{0, \ldots, N_r\}$ (N_r viz kapitola 1.3.8).

Definuji matici $\mathcal{W} \in {\mathbb{F}_{256}}^{4\times(N_b(N_r+1))}$, tak že sloupce $i \cdot N_b$ až $(i+1) \cdot N_b - 1$ matice $\mathcal{W}$ (indexujeme od 0) jsou tvořeny sloupci matice K_i pro $i \in \{0, \ldots, N_r\}$.

$N_k \backslash N_b$	4	5	6	7	8
4	10	11	12	13	14
5	11	11	12	13	14
6	12	12	12	13	14
7	13	13	13	13	14
8	14	14	14	14	14

Tabulka 1.3: Závislost počtu rundondovních zobrazeních na hodnotách N_k a N_b

Tedy matice $\mathcal{W} = \left(K_0 \mid K_1 \mid \ldots \mid K_{N_r} \right)$ a budeme ji nazývat rozšířenou maticí klíče K pro N_r rundovních zobrazení.

Nechť máme klíč $K \in \mathcal{K} = {\mathbb{F}_{256}}^{4\times N_k}$. Matice $\mathcal{W}$ vznikne tak, že do prvních N_k sloupců matice $\mathcal{W}$ dosadíme sloupce klíče K. Zbývající sloupce nagenerujeme následujícím rekurzivním algoritmem, který je zapsán v pseudokódu:

Konvence pro pseudokód:

- $N_b := b$, $N_k := k$ a $N_r := r$
- W:= $\mathcal{W} \in {\mathbb{F}_{256}}^{4\times(N_b(N_r+1))}$
- K:= $\mathcal{K} \in {\mathbb{F}_{256}}^{4\times N_k}$
- S je permutace definovaná v kapitole 1.3.3
- RC(n) je konstanta definovaná v kapitole 1.3.7, pro $n \in \mathbb{N}$

Pokud $N_k \leq 6$**:**

```
KeyExpansion( byte K[4][k], byte W[4][b(r + 1)] ) {
for(j=0; j<k; j++) {
   for(i=0; i<4; i++){ W[i][j] = K[i][j]; }
}
for(j=k; j<b(r+1); j++) {
   if (j mod k == 0) {
      W[0][j] = W[0][j - k] + S(W[1][j-1]) + RC(j/k);
      for(i=1; i<4; i++) {
         W[i][j] = W[i][j - k] + S(W[(i+1) mod 4][j-1]);
      }
   }
   else {
      for(i=0; i<4; i++) {
         W[i][j] = W[i][j - k] + W[i][j-1];
      }
   }
}}
```

Pokud $N_k > 6$**:**

```
KeyExpansion( byte K[4][k], byte W[4][b(r + 1)] ) {
for(j=0; j<k; j++) {
   for(i=0; i<4; i++){ W[i][j] = K[i][j]; }
```

```
}
for(j=k; j<b(r+1); j++) {
   if (j mod k == 0) {
      W[0][j] = W[0][j - k] + S(W[1][j-1]) + RC(j/k);
      for(i=1; i<4; i++) {
         W[i][j] = W[i][j - k] + S(W[(i+1) mod 4][j-1]);
      }
   }
   else if (j mod k == 4) {
      for(i=0; i<4; i++) {
         W[i][j] = W[i][j - k] + S(W[i][j-1]);
      }
   }
   else {
      for(i=0; i<4; i++) {
         W[i][j] = W[i][j - k] + W[i][j-1];
      }
   }
}}
```

Definuji zobrazení (Key Schedule) $\mathcal{KS} : {\mathbb{F}_{256}}^{4\times N_k} \times \{0,\ldots,N_r\} \to {\mathbb{F}_{256}}^{4\times N_b}$ předpisem $\mathcal{KS}(K,i) = K_i$ pro $K \in {\mathbb{F}_{256}}^{4\times N_k}$ a $i \in \{0,\ldots,N_r\}$. Tedy zobrazení, které klíčí K a i přiřadí i-tý rundovní klíč. Toto značení budeme používat po zbytek této práce.

1.3.10 Popis šifrování

Na začátku této kapitoly 1.3 jsme zadefinovali šifru Rijndael jako klíč iterující blokovou šifru. Tedy $\forall K \in \mathcal{K}$ platí, že $e_K = \delta[K_{N_r}] \circ \tilde{\rho} \circ \delta[K_{N_r-1}] \circ \rho \circ \ldots \circ \rho \circ \delta[K_0]$, pro zobrazení definovaná:

- $\rho := \mathcal{MC} \circ \mathcal{SR} \circ \mathcal{SB}$
- $\tilde{\rho} := \mathcal{SR} \circ \mathcal{SB}$
- $\delta[K_i] := \mathcal{AK}[\mathcal{KS}(K,i)]$ pro $i \in \{0,\ldots,N_r\}$

Všimněme si, že poslední rundovní zobrazení je odlišné od ostatních rundovních zobrazení. I přes tuto drobnou odchylku oproti definici 1.3 budeme nazývat šifru Rijndael klíč iterující blokovou šifrou.

Poznámka 1.14. *Všimněme si, že z definice kryptosystému (definice 1.1) platí $\forall P \in \mathcal{P}$ a $\forall K \in \mathcal{K}$: $d_K(e_K(P)) = P$. Z toho plyne, že $\mathcal{SB}$, $\mathcal{SR}$, $\mathcal{MC}$ a $\mathcal{AK}[\cdot]$ jsou bijektivní zobrazení a pro každé existuje inverzní zobrazení.*

Poznámka 1.15. *Všimněme si, že definice šifry Rijndael lze intuitivně rozšířit i pro hodnoty N_b a N_k větší než 8.*

Úmluva 1.16. *V dalším textu budeme používat šifru Rijndael i s jinými parametry než byla zadefinovaná. Např.: budeme používat hodnoty parametru N_k a N_b větší než 8 (poznámka 1.15) nebo budeme měnit počet rundovních zobrazeních N_r*

oproti definici nebo v posledním rundovním zobrazení nebudeme vynechávat zobrazení Mix Columns (kapitola 1.3.5). Pro usnadnění zavedeme značení pro šifru Rijndael $\mathcal{R}(N_b, N_k, N_r)$, kde N_b je počet sloupců v matici otevřeného a šifrového textu, N_k je počet sloupců v matici klíče a N_r je počet rundovních zobrazení.

Dané značení si ilustrujeme na příkladě: $\mathcal{R}(5, 10, 5)$ je šifra Rijndael kde $\mathcal{P} = \mathcal{C} = {\mathbb{F}_{256}}^{4\times 5}$, $\mathcal{K} = {\mathbb{F}_{256}}^{4\times 10}$, počet rundovních zobrazení je pouze 5 a v posledním rundovním zobrazení se vynechává zobrazení Mix Columns.

Pokud nebudeme vynechávat v posledním rundovním zobrazení zobrazení Mix Columns, tak budeme používat značení $\mathcal{R}(N_b, N_k, N_r)^*$.

Tedy $\mathcal{R}(N_b, N_k, N_r)$ nebo $\mathcal{R}(N_b, N_k, N_r)^*$ značí konfiguraci šifry Rijndeal s příslušnými parametry pro $N_b, N_k, N_r \in \mathbb{N}$

1.4 Šifrový diagram pro šifru Rijndael

Po zbytek této práce budeme pracovat pouze s šifrou Rijndael (pozn: $\mathcal{P} = \mathcal{C} = {\mathbb{F}_{256}}^{4\times N_b}$ a $\mathcal{K} = {\mathbb{F}_{256}}^{4\times N_k}$, kapitola 1.3.2). Z kapitoly 1.3 plyne, že šifrové zobrazení $e_K : \mathcal{P} \to \mathcal{C}$ lze zapsat předpisem:
$e_K(P) = \mathcal{AK}[\mathcal{KS}(K, N_r)] \circ \mathcal{SR} \circ \mathcal{SB} \circ \mathcal{AK}[\mathcal{KS}(K, N_r - 1)] \circ \mathcal{MC} \circ \mathcal{SR} \circ \mathcal{SB} \circ \mathcal{AK}[\mathcal{KS}(K, N_r - 2)] \circ \ldots \circ \mathcal{AK}[\mathcal{KS}(K, 1)] \circ \mathcal{MC} \circ \mathcal{SR} \circ \mathcal{SB} \circ \mathcal{AK}[\mathcal{KS}(K, 0)](P)$
pro $\forall P \in \mathcal{P}$ a $\forall K \in \mathcal{K}$.

Tedy zobrazení e_K je definováno jako posloupnost zobrazení. Tuto posloupnost zobrazení můžeme rozdělit na po sobě jdoucí úseky, z kterých lze zpětně poskládat zobrazení e_K. Tuto myšlenku si ilustrujeme na příkladě. Definujme následující zobrazení:

- $\alpha_i[K] := \mathcal{AK}[\mathcal{KS}(K, i)] \circ \mathcal{MC} \circ \mathcal{SR} \circ \mathcal{SB}$ pro $i \in \{1, \ldots, N_r - 1\}$
- $\alpha_{N_r}[K] := \mathcal{AK}[\mathcal{KS}(K, N_r)] \circ \mathcal{SR} \circ \mathcal{SB}$

Potom platí $e_K = \alpha_{N_r}[K] \circ \ldots \circ \alpha_1[K] \circ \mathcal{AK}[\mathcal{KS}(K, 0)]$. Tuto úvahu využijeme v následující definici:

Definice 1.17. *Nechť máme $n \in \mathbb{N}$, $\mathcal{R}(N_b, N_k, N_r)$ nebo $\mathcal{R}(N_b, N_k, N_r)^*$ a existuje posloupnost zobrazení $\alpha_1[K], \ldots, \alpha_n[K]$ takových, že $\alpha_i[K] : {\mathbb{F}_{256}}^{4\times N_b} \to {\mathbb{F}_{256}}^{4\times N_b}$ pro $i \in \{1, \ldots, n\}$. Dále platí $e_K = \alpha_n[K] \circ \ldots \circ \alpha_1[K]$ pro $\forall K \in \mathcal{K}$. Zvolme libovolné $P \in \mathcal{P}$ a $K \in \mathcal{K}$ a definujme matice:*

- $A_0 := P$
- $A_i := (\alpha_i[K] \circ \ldots \circ \alpha_1[K])(P)$ *pro* $i \in \{1, \ldots, n-1\}$
- $A_n := (\alpha_n[K] \circ \ldots \circ \alpha_1[K])(P) = e_K(P)$

Definujeme šifrový diagram (šifry Rijndael) jako uspořádanou trojici

$$((A_0, \ldots, A_n), (K_0, \ldots, K_{N_r}), (\alpha_1[K], \ldots, \alpha_n[K]))_{P,K}$$

Šifrový diagram jsou tedy posloupnosti mezivýsledků, rundovních klíčů a dílčích zobrazení.

Poznámka 1.18. *Všimněme si několika důležitých postřehů k definici 1.17 (používáme stejné značení jako v této definici):*

- *Některá zobrazení $\alpha_i[K]$ pro $i \in \{1, \ldots, n\}$ jsou závislá na rundovním klíči tedy i na klíči K, a proto tuto závislost značíme v hranatých závorkách pro všechna zobrazení $\alpha_i[K]$. Někdy na tato zobrazení budeme nahlížet jako na zobrazení dvou proměnných $\alpha_i[\cdot]$.*

- *Rundovní klíč je generován z několika předchozích rundovních klíčů. Přesný počet závisí na dané konfiguraci šifry Rijndael. Tato vlastnost plyne z rekurzivní definice zobrazení $\mathcal{KS}$ (kapitola 1.3.9). Na zobrazení $\mathcal{KS}$ lze tedy také nahlížet jako na zobrazení mezi několika předchozími rundovními klíči a aktuálním rundovním klíčem.*

- *K určení šifrového diagramu stačí definovat uspořádanou n-tici zobrazení $(\alpha_1[\cdot], \ldots, \alpha_n[\cdot])$ s příslušnými vlastnostmi a hodnoty $P \in \mathcal{P}$ a $K \in \mathcal{K}$.*

- *Jedna z důležitých vlastností této definice je, že šifrový diagram můžeme velmi dobře graficky znázornit, což budeme ilustrovat v dalším textu.*

Úmluva 1.19. *Definujeme zobrazení:*

- $\mathcal{AC}(n) : {\mathbb{F}_{256}}^{4\times 1} \to {\mathbb{F}_{256}}^{4\times 1}$ *pro* $n \in \mathbb{N}$ *předpisem:*

$$\mathcal{AC}(n) : \begin{pmatrix} a_{0,0} \\ a_{1,0} \\ a_{2,0} \\ a_{3,0} \end{pmatrix} \mapsto \begin{pmatrix} a_{0,0} + RC(n) \\ a_{1,0} \\ a_{2,0} \\ a_{3,0} \end{pmatrix}$$

- $\mathcal{RW} : {\mathbb{F}_{256}}^{4\times 1} \to {\mathbb{F}_{256}}^{4\times 1}$ *předpisem:*

$$\mathcal{RW} : \begin{pmatrix} a_{0,0} \\ a_{1,0} \\ a_{2,0} \\ a_{3,0} \end{pmatrix} \mapsto \begin{pmatrix} a_{1,0} \\ a_{2,0} \\ a_{3,0} \\ a_{0,0} \end{pmatrix}$$

Zobrazení $\mathcal{AC}(n)$, $\mathcal{RW}$ a $\mathcal{SB}$ (kapitola 1.3.3) budeme vždy používat při popisu rekurzivní závislosti rundovních klíčů v šifrovém diagramu. Rekurzivní závislost rundovních klíčů jsme zmínili v poznámce 1.18.

Při grafickém znázornění šifrového diagramu budeme používat místo $\mathcal{AC}(n)$ pouze $\mathcal{AC}$. Parametr n bude určovat pořadí výskytu zobrazení $\mathcal{AC}$ v šifrovém diagramu. Tedy pokud se bude jednat o n-tý výskyt zobrazení $\mathcal{AC}$ v šifrovém diagramu máme na mysli $\mathcal{AC}(n)$.

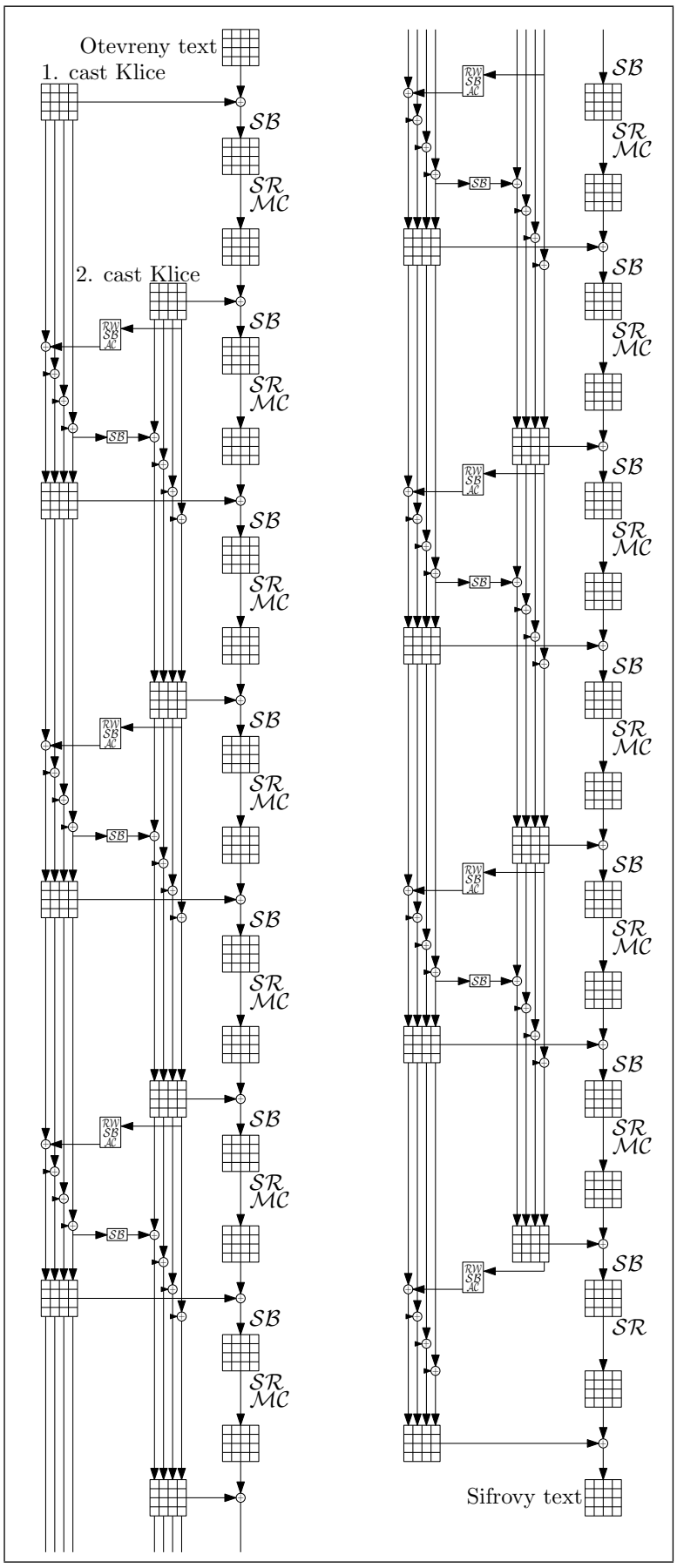
Nyní si zadefinujeme několik šifrových diagramů, se kterými budeme pracovat v dalším textu.

Šifrový diagram 1.20. *Nechť máme $\mathcal{R}(N_b, N_k, N_r)$ nebo $\mathcal{R}(N_b, N_k, N_r)^*$ a libovolné hodnoty $P \in \mathcal{P}$ a $K \in \mathcal{K}$. Definujeme zobrazení:*

- $\alpha_{2i}[K] := \mathcal{SB} \circ \mathcal{AK}[\mathcal{KS}(K, i)]$ *pro* $i \in \{0, \ldots, N_r - 1\}$

- $\alpha_{2i+1}[K] := \mathcal{MC} \circ \mathcal{SR}$ *pro* $i \in \{0, \ldots, N_r - 2\}$

- *definice $\alpha_{2N_r-1}[K]$ se liší dle volby $\mathcal{R}(N_b, N_k, N_r)$ nebo $\mathcal{R}(N_b, N_k, N_r)^*$*



Obrázek 1.2: Šifrový diagram 1.20 pro $\mathcal{R}(4, 8, 14)$

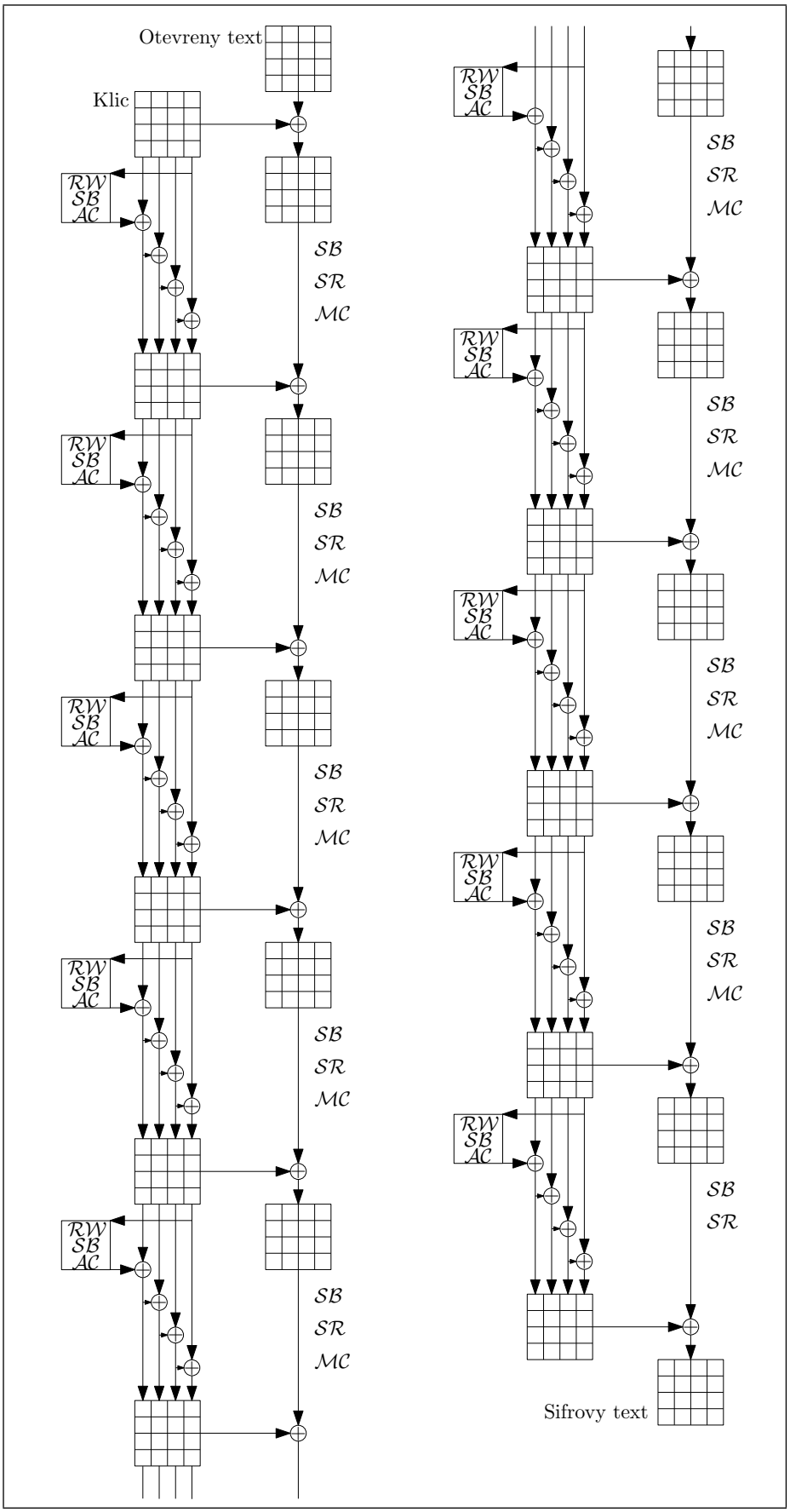
  - *pro* $\mathcal{R}(N_b, N_k, N_r)^*$ *definujeme* $\alpha_{2N_r-1}[K] := \mathcal{MC} \circ \mathcal{SR}$
  - *pro* $\mathcal{R}(N_b, N_k, N_r)$ *definujeme* $\alpha_{2N_r-1}[K] := \mathcal{SR}$
- $\alpha_{2N_r}[K] := \mathcal{AK}[\mathcal{KS}(K, N_r)]$

Potom uspořádaná $2N_r+1$*-tice* $(\alpha_0[K], \ldots, \alpha_{2N_r}[K])$ *a hodnoty* P *a* K *definují šifrový diagram. Tento šifrový diagram si ilustrujeme na obrázku 1.2 pro* $\mathcal{R}(4, 8, 14)$.

Šifrový diagram 1.21. *Nechť máme* $\mathcal{R}(N_b, N_k, N_r)$ *nebo* $\mathcal{R}(N_b, N_k, N_r)^*$ *a libovolné hodnoty* $P \in \mathcal{P}$ *a* $K \in \mathcal{K}$*. Definujeme zobrazení:*

- $\alpha_0[K] := \mathcal{AK}[\mathcal{KS}(K, 0)]$
- $\alpha_i[K] := \mathcal{AK}[\mathcal{KS}(K, i)] \circ \mathcal{MC} \circ \mathcal{SR} \circ \mathcal{SB}$ *pro* $i \in \{1, \ldots, N_r - 1\}$
- *definice* $\alpha_{N_r}[K]$ *se liší dle volby* $\mathcal{R}(N_b, N_k, N_r)$ *nebo* $\mathcal{R}(N_b, N_k, N_r)^*$
  - *pro* $\mathcal{R}(N_b, N_k, N_r)^*$ *definujeme* $\alpha_{N_r}[K] := \mathcal{AK}[\mathcal{KS}(K, N_r)] \circ \mathcal{MC} \circ \mathcal{SR} \circ \mathcal{SB}$
  - *pro* $\mathcal{R}(N_b, N_k, N_r)$ *definujeme* $\alpha_{N_r}[K] := \mathcal{AK}[\mathcal{KS}(K, N_r)] \circ \mathcal{SR} \circ \mathcal{SB}$

Potom uspořádaná $N_r + 1$*-tice* $(\alpha_0[K], \ldots, \alpha_{N_r}[K])$ *a hodnoty* P *a* K *definují šifrový diagram. Tento šifrový diagram si ilustrujeme na obrázku 1.3 pro* $\mathcal{R}(4, 4, 10)$.



Obrázek 1.3: Šifrový diagram 1.21 pro $\mathcal{R}(4, 4, 10)$

2. Diferenční stopa

V této kapitole si ukážeme vlastní pohled na pojem diferenční stopa, který byl motivován literaturou [2] a vlastní pohled na práci s tímto pojmem, který vychází z myšlenek článku [1]. Tyto myšlenky si vysvětlíme dopodrobna a některé si formulujeme jako větu a doplníme vlastními důkazy.

Definice 2.1. *Nechť máme* $\mathcal{R}(N_b, N_k, N_r)$ *nebo* $\mathcal{R}(N_b, N_k, N_r)^*$, $P_1, P_2 \in \mathcal{P}$ *a* $K^1, K^2 \in \mathcal{K}$ *(používáme horní index pro rozlišení klíčů a dolní index pro příslušný rundovní klíč). Definujeme* $\triangle P := P_1 + P_2$ *a* $\triangle K := K^1 + K^2$. *Nechť máme dva šifrové diagramy:*

- $\mathcal{D}_1 = ((A_0, \ldots, A_n), ({K^1}_1, \ldots, {K^1}_{N_r}), (\alpha_1[K^1], \ldots, \alpha_n[K^1]))_{P_1,K^1}$
- $\mathcal{D}_2 = ((B_0, \ldots, B_n), ({K^2}_1, \ldots, {K^2}_{N_r}), (\alpha_1[K^2], \ldots, \alpha_n[K^2]))_{P_2,K^2}$

Potom definujeme:

1. *platnou diferenční stopu vytvořenou z* $\mathcal{D}_1$ *a* $\mathcal{D}_2$ *jako uspořádanou trojici:*

   $$((A_0{+}B_0, \ldots, A_n{+}B_n), ({K^1}_0{+}{K^2}_0, \ldots, {K^1}_{N_r}{+}{K^2}_{N_r}), (\alpha_1[\cdot], \ldots, \alpha_n[\cdot]))_{P_1,K^1}$$

2. *neplatnou diferenční stopu jako uspořádanou trojici:*

   $$((\triangle B_0, \ldots, \triangle B_n), (\triangle L_0, \ldots, \triangle L_{N_r}), (\alpha_1[\cdot], \ldots, \alpha_n[\cdot]))$$

   kde $\triangle B_0, \ldots, \triangle B_n, \triangle L_0, \ldots, \triangle L_{N_r} \in {\mathbb{F}_{256}}^{4\times N_b}$ *a*
   $\forall P_1, P_2 \in \mathcal{P}$ *a* $\forall K^1, K^2 \in \mathcal{K}$ *platí, že šifrové diagramy*
   $\mathcal{D}_1 = ((A_0, \ldots, A_n), ({K^1}_1, \ldots, {K^1}_{N_r}), (\alpha_1[K^1], \ldots, \alpha_n[K^1]))_{P_1,K^1}$ *a*
   $\mathcal{D}_2 = ((B_0, \ldots, B_n), ({K^2}_1, \ldots, {K^2}_{N_r}), (\alpha_1[K^2], \ldots, \alpha_n[K^2]))_{P_2,K^2}$
   nevytvoří platnou diferenční stopu:
   $((\triangle B_0, \ldots, \triangle B_n), (\triangle L_0, \ldots, \triangle L_{N_r}), (\alpha_1[\cdot], \ldots, \alpha_n[\cdot]))_{P_1,K_1}$

3. *diferenční stopu s neznámou platností jako uspořádanou trojici:*

   $$((\triangle B_0, \ldots, \triangle B_n), (\triangle L_0, \ldots, \triangle L_{N_r}), (\alpha_1[\cdot], \ldots, \alpha_n[\cdot]))$$

   kde $\triangle B_0, \ldots, \triangle B_n, \triangle L_0, \ldots, \triangle L_{N_r} \in {\mathbb{F}_{256}}^{4\times N_b}$

Poznámka 2.2. *Několik postřehů k definici 2.1 (používáme stejné značení jako v této definici):*

- *Na zobrazení* $\alpha_i[\cdot] : {\mathbb{F}_{256}}^{4\times N_b} \times {\mathbb{F}_{256}}^{4\times N_k} \to {\mathbb{F}_{256}}^{4\times N_b}$ *pro* $i \in \{1, \ldots, n\}$ *nahlížíme jako na zobrazení dvou proměnných.*
- *Všimněme si, že hodnoty* $\triangle P$ *a* $\triangle K$ *jsou obsažené v platné diferenční stopě,* $\triangle P = A_0 + B_0$ *a* $\triangle K$ *je obsažená v několika prvních maticích* ${K^1}_i + {K^2}_i$. *Toto plyne z definice Key Schedulu (kapitola 1.3.9).*

- *Ze znalosti platné diferenční stopy jsme schopni znovu získat šifrové diagramy $\mathcal{D}_1$ a $\mathcal{D}_2$. K jejich definici potřebujeme znát: $(\alpha_1[\cdot], \ldots, \alpha_n[\cdot])$, P_1, K_1, $\triangle P$ a $\triangle K$ a všechny tyto informace jsou obsažené v platné diferenční stopě.*

- *Pokud ověřujeme platnost diferenční stopy, tak hledáme hodnoty $P_1, P_2 \in \mathcal{P}$ a $K^1, K^2 \in \mathcal{K}$, z kterých lze vytvořit šifrové diagramy pro $(\alpha_1[\cdot], \ldots, \alpha_n[\cdot])$ a následně z těchto šifrových diagramů lze vytvořit diferenční stopu, kterou ověřujeme. Vzhledem k tomu, že z diferenční stopy známe hodnoty $\triangle P$ a $\triangle K$ (dle 2. bodu), stačí nalézt hodnoty P_1 a K^1. Tyto hodnoty budeme nazývat hodnoty splňující diferenční stopu. Celkově tedy pokud ověřujeme platnost diferenční stopy, hledáme P_1 a K^1 takové, že splňují diferenční stopu.*

- *Diferenční stopu budeme graficky znázorňovat stejně jako šifrový diagram, z kterého byla odvozena. (př.: obrázky 1.2 a 1.3)*

Věta 2.3. *Nechť máme $\mathcal{R}(N_b, N_k, N_r)$ nebo $\mathcal{R}(N_b, N_k, N_r)^*$,*

$$((\triangle B_0, \ldots, \triangle B_n), (\triangle L_0, \ldots, \triangle L_{N_r}), (\alpha_1[\cdot], \ldots, \alpha_n[\cdot]))$$

je diferenční stopa s neznámou platností pro zadanou konfiguraci šifry Rijndael a $\triangle L$ je příslušná diference klíčů a $\triangle P$ je příslušná diference otevřených textů (znalost těchto hodnot plyne z poznámky 2.2). Potom nutná podmínka pro platnost této stopy je:

- *$\forall i \in \{1, \ldots, n\}$ je $(\triangle B_{i-1}, \triangle L)$ a $\triangle B_i$ kompatibilní diferenční pár zobrazení $\alpha_i[\cdot]$ a zároveň*

- *$\forall i \in \{0, \ldots, N_r\}$ je $\triangle L$ a $\triangle L_i$ kompatibilní diferenční pár zobrazení $\mathcal{KS}(\cdot, i)$.*

Důkaz. Chceme dokázat, že platnost diferenční stopy implikuje platnost výše zmíněné podmínky. Tuto implikaci budeme dokazovat sporem, tedy $\exists P \in \mathcal{P}$ a $\exists K \in \mathcal{K}$, které splní zadanou diferenční stopu (poznámka 2.2) a zároveň daná podmínka neplatí. V tuto chvíli se důkaz rozpadá na dvě části:

1. $\exists i \in \{0, \ldots, N_r\}$, že $\triangle L$ a $\triangle L_i$ je nekompatibilní diferenční pár skrz zobrazení $\mathcal{KS}(\cdot, i)$. Tedy $\forall L \in {\mathbb{F}_{256}}^{4 \times N_k} = \mathcal{K}$ platí, že $\triangle L_i \neq \mathcal{KS}(L + \triangle L, i) + \mathcal{KS}(L, i)$. Ale z předpokladu a definice diferenční stopy víme $\triangle L_i = \mathcal{KS}(K + \triangle L, i) + \mathcal{KS}(K, i)$, což je spor.

2. Předpokládáme, že $\forall i \in \{0, \ldots, N_r\}$ platí, že $\triangle L$ a $\triangle L_i$ je kompatibilní diferenční pár pro zobrazení $\mathcal{KS}(\cdot, i)$. Pokud by toto neplatilo použijeme bod 1. Zvolme nejmenší $i \in \{1, \ldots, n\}$ takové, že $(\triangle B_{i-1}, \triangle L)$ a $\triangle B_i$ je nekompatibilní diferenční pár pro zobrazení $\alpha_i[\cdot]$. Potom $\forall M \in {\mathbb{F}_{256}}^{4 \times N_b}$ platí, že $\triangle B_i \neq \alpha_i[K + \triangle L](M + \triangle B_{i-1}) + \alpha_i[K](M)$. Dosaďme za M výraz $(\alpha_{i-1}[K] \circ \ldots \circ \alpha_1[K])(P)$:

   $$\triangle B_i \neq \alpha_i[K+\triangle L]\Big((\alpha_{i-1}[K]\circ\ldots\circ\alpha_1[K])(P)+\triangle B_{i-1}\Big)+\alpha_i[K]\Big((\alpha_{i-1}[K]\circ \ldots \circ \alpha_1[K])(P)\Big)$$

Je dobré si uvědomit, že $\triangle B_{i-1} = \Big(\alpha_{i-1}[K+\triangle L]\circ\ldots\circ\alpha_1[K+\triangle L]\Big)(P+\triangle P)+\Big(\alpha_{i-1}[K]\circ\ldots\circ\alpha_1[K]\Big)(P)$, protože předpokládáme, že P a K splňují zadanou diferenční stopu a zároveň platí, že $(\triangle B_{i-2}, \triangle L)$ a $\triangle B_{i-1}$ je kompatibilní diferenční pár pro zobrazení $\alpha_{i-1}[\cdot]$.

$$\triangle B_i \neq \alpha_i[K+\triangle L]\Big(\Big(\alpha_{i-1}[K]\circ\ldots\circ\alpha_1[K]\Big)(P)+\Big(\alpha_{i-1}[K+\triangle L]\circ\ldots\circ\alpha_1[K+\triangle L]\Big)(P+\triangle P)+\Big(\alpha_{i-1}[K]\circ\ldots\circ\alpha_1[K]\Big)(P)\Big)+\Big(\alpha_i[K]\circ\ldots\circ\alpha_1[K]\Big)(P)$$

$$\triangle B_i \neq \alpha_i[K+\triangle L]\Big((\alpha_{i-1}[K+\triangle L]\circ\ldots\circ\alpha_1[K+\triangle L])(P+\triangle P)\Big)+\Big(\alpha_i[K]\circ\ldots\circ\alpha_1[K]\Big)(P)$$

$$\triangle B_i \neq (\alpha_i[K+\triangle L]\circ\ldots\circ\alpha_1[K+\triangle L])(P+\triangle P)+(\alpha_i[K]\circ\ldots\circ\alpha_1[K])(P)$$

Což je spor s tím, že P a K splní zadanou diferenční stopu.

□

Jak jsme si řekli v úvodu, naší snahou bude pro libovolný otevřený text $P \in \mathcal{P}$ nalézt klíče $K^1, K^2 \in \mathcal{K}$ (používáme horní indexy k indexaci klíčů) takové, že $e_{K^1}(P) = e_{K^2}(P)$. Konkrétní hodnota $e_{K^1}(P)$ je pro nás nepodstatná. Začneme tím, že budeme hledat diferenční stopu s neznámou platností vytvořenou od šifrových diagramů 1.20, která splňuje nutnou podmínku pro platnost (viz věta 2.3) a také bude splňovat $\triangle P = A_n + B_n = 0$ a $\triangle K \neq 0$ (značení stejné jako v definice 2.1). Po celou tuto kapitolu budeme pracovat pouze s diferenčními stopami vytvořenými od šifrových diagramu 1.20.

2.1 Chování diference skrz známé zobrazení

Připomeňme si poznámku 1.11 bod 3 a 4 a fakt, že na vektorové prostory můžeme nahlížet jako na grupy a na lineární zobrazení vektorových prostorů tedy jako na homomorfismus grup. Jak jsme se zmínili, zobrazení Shif Rows a Mix Columns (kapitoly 1.3.4 a 1.3.5) jsou lineární. Šíření diference skrz tato zobrazení se tedy řídí poznámkou 1.11 bod 3. Příklad, pokud máme $\mathcal{R}(N_b, N_k, N_r)$ a vstupní diference do zobrazení Sift Rows je $a \in {\mathbb{F}_{256}}^{4\times N_b}$, potom výstupní diference je $\mathcal{SR}(a) \in {\mathbb{F}_{256}}^{4\times N_b}$, tedy a, $\mathcal{SR}(a)$ tvoří jistý diferenční pár (definice 1.10).

Je dobré si všimnout, že zobrazení $\mathcal{RW}$ je lineární a zobrazení $\mathcal{AC}(n)$ pro $n \in \mathbb{N}$ je afinní (úmluva 1.19). Šíření diference skrz tato zobrazení se tedy opět řídí poznámkou 1.11.

Věta 2.4. *Nechť máme $\mathcal{R}(N_b, N_k, N_r)$ nebo $\mathcal{R}(N_b, N_k, N_r)^*$ a na zobrazení Add Round Key (kapitola 1.3.6) nahlížíme jako na zobrazení dvou proměnných tedy $\mathcal{AK} : {\mathbb{F}_{256}}^{4\times N_b} \times {\mathbb{F}_{256}}^{4\times N_b} \to {\mathbb{F}_{256}}^{4\times N_b}$. Nechť jsou $\triangle x, \triangle y \in {\mathbb{F}_{256}}^{4\times N_b}$ libovolné. Potom $(\triangle x, \triangle y)$ a $\triangle x + \triangle y$ tvoří jistý diferenční pár (definice 1.10).*

Důkaz. Vstupní diference je $(\triangle x, \triangle y)$. Zvolme libovolné $x, y \in {\mathbb{F}_{256}}^{4\times N_b}$, potom výstupní diference je $\mathcal{AK}(x+\triangle x, y+\triangle y)+\mathcal{AK}(x,y) = x+\triangle x+y+\triangle y+x+y = \triangle x + \triangle y$ □

Celkem pro výše zmíněné zobrazení umíme pro libovolnou vstupní diferenci určit výstupní diferenci tak, aby tyto dvě hodnoty tvořili jistý diferenční pár (definice 1.10). Také víme, že kompatibilní diferenční pár pro tato zobrazení je vždy jistý diferenční pár.

Nyní jsme popsali šíření diference skrz všechna zobrazení vyskytující se v šifře Rijndael krom zobrazení Sub Bytes (kapitola 1.3.3). Šíření diference skrz Sub Bytes je závislé na šíření diference skrz permutaci S, o které víme pouze to, že pro libovolné nenulové $a, b \in \mathbb{F}_{256}$, kde a je vstupní diference a b je výstupní diference, existují maximálně 4 prvky $x \in \mathbb{F}_{256}$, které tuto diferenci splní. To jest platí: $b = S(x+a)+S(x)$. Průchod nenulové diference skrz permutaci S budeme označovat jako aktivní S-box.

Pokud máme diferenční stopu s neznámou platností, která splňuje nutnou podmínku pro platnost (věta 2.3), potom pro ověření této diferenční stopy hledáme $P \in \mathcal{P}$ a $K \in \mathcal{K}$, které by splnily tuto diferenční stopu (poznámka 2.2). Všimněme si, že pro všechna zobrazení v šifře Rijndael krom zobrazení Sub Bytes jsou v této diferenční stopě všechny diferenční páry jisté. Toto plyne z předchozí úvahy. Hledáme tedy $P \in \mathcal{P}$ a $K \in \mathcal{K}$, které vytváří šifrový diagram, v němž vstupy do aktivních S-boxů splňují podmínky dané ze vstupní a výstupní diference daného aktivního S-boxu.

2.2 Jedno rundovní diferenční stopa

V této kapitole si ukážeme jak vytvořit platnou diferenční stopu pro $\mathcal{R}(5,10,1)^*$. V této konfiguraci potřebujeme pouze dva rundovní klíče, které vzniknou rozdělením matice klíče $K \in {\mathbb{F}_{256}}^{4\times 10}$ na dvě matice typu ${\mathbb{F}_{256}}^{4\times 5}$ dle Key Schedule (kapitola 1.3.9).

Matice $\triangle K^1, \triangle K^2 \in {\mathbb{F}_{256}}^{4\times 5}$ (používáme horní index, dolními budeme indexovat prvky) reprezentují diference rundovních klíčů. Dále matice $\triangle P, \triangle C \in {\mathbb{F}_{256}}^{4\times 5}$ reprezentují diference otevřeného a šifrového textu. O těchto maticích předpokládáme $\triangle P = \triangle C = 0$. Dále budeme pracovat s maticemi $A^1, A^2 \in {\mathbb{F}_{256}}^{4\times 5}$ (používáme horní index, dolními budeme indexovat prvky). Zadefinujeme diferenční stopu:

$$((\triangle P, A^1, A^2, \triangle C), (\triangle K^1, \triangle K^2), (\mathcal{SB} \circ \mathcal{AK}[\mathcal{KS}(\cdot,0)], \mathcal{MC} \circ \mathcal{SR}, \mathcal{AK}[\mathcal{KS}(\cdot,1)])$$

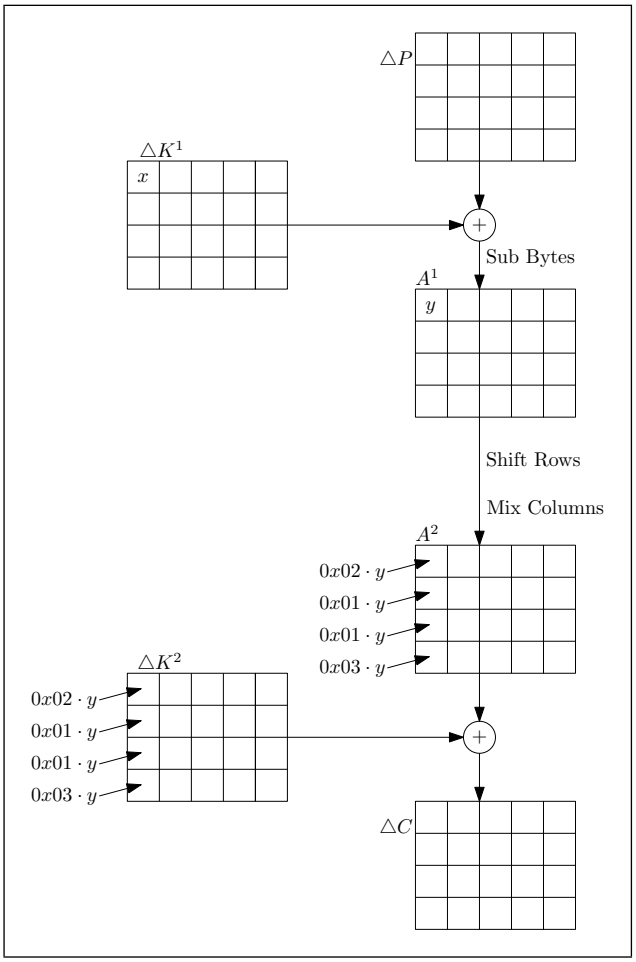
Jaké konkrétní hodnoty mají výše zmíněné matice si ukážeme nyní: Matici $\triangle K^1$ položme rovnou nule až na $\triangle {k^1}_{0,0} := x \in \mathbb{F}_{256}$. Položit jediný nenulový prvek na tuto pozici má tyto důvody:

- 1. řádek jsme zvolili z toho důvodu, že zobrazení Shift Rows (kapitola 1.3.4) nemění 1. řádek.
- 1. sloupec jsme zvolili proto, že to bude výhodné při vytváření diferenční stopy pro více rundovních zobrazení, což si ukážeme v následujícím textu.

Z kapitoly 2.1 víme, jak se šíří diference skrz jednotlivá zobrazení a z poznámky 1.11 víme, že nulová diference se vždy šíří na nulovou diferenci. Předpokládali jsme, že matice $\triangle P = 0$. Tedy matice A^1 bude rovna nule až na $a^1{}_{0,0} := y \in \mathbb{F}_{256}$. Zde volme x a y tak, aby $D_p S(x, y) = 2^{-6}$ to jest $|M_S(x, y)| = 4$.

Vzhledem k tomu, že zobrazení Shift Rows a Mix Columns jsou lineární, víme, jak se diference bude šířit dál. Tedy matice

$$A^2 = \begin{pmatrix} 0x02 \cdot y & 0 & 0 & 0 & 0 \\ 0x01 \cdot y & 0 & 0 & 0 & 0 \\ 0x01 \cdot y & 0 & 0 & 0 & 0 \\ 0x03 \cdot y & 0 & 0 & 0 & 0 \end{pmatrix}$$



Obrázek 2.1: 1 rundovní diferenční stopa

Dále víme jak se šíří diference skrz zobrazení Add Round Key. Odtud platí vztah $A^2 + \triangle K^2 = \triangle C$. Víme, že matice $\triangle C = 0$, tedy $A^2 = \triangle K^2$. Nyní známe hodnoty všech matic vystupujících v naší diferenční stopě. Tuto stopu si ilustrujeme na obrázku 2.1. Prázdná políčka značí nulovou hodnotu.

Zvolme hodnoty $x := 0x03$ a $y := 0x18$, potom $M_S(x, y) = \{0xDF, 0xDC, 0x03, 0x00\}$. Tato fakta plynou z kapitoly 6, konkrétně z tabulky 6.1. V tuto chvíli máme diferenční stopu, která splňuje nutnou podmínku pro platnost (věta 2.3). Všimněme si, že v této diferenční stopě je pouze jeden aktivní S-box. Nyní

předpokládejme, že máme libovolný pevně zvolený otevřený text $P \in \mathcal{P}$. Tedy hledáme $K \in \mathcal{K}$ takové, že P a K splní diferenční stopu, to jest v šifrovém diagramu vytvořeném pro konfiguraci $\mathcal{R}(5,10,1)^*$ od P a K bude vstup do aktivního S-boxu z množiny $M_S(x,y) = \{0xDF, 0xDC, 0x03, 0x00\}$. Jediná podmínka tedy je, aby platilo $k_{0,0} + p_{0,0} \in \{0xDF, 0xDC, 0x03, 0x00\}$, kde $k_{0,0}$ je prvek matice K a $p_{0,0}$ je prvek matice P. Ostatní prvky K mohou být voleny libovolně. Tímto postupem jsme ukázali, že existují $P \in \mathcal{P}$ a $K \in \mathcal{K}$, které splňují diferenční stopu. Takto vytvořená diferenční stopa je platná.

Tato stopa nemá moc velký praktický význam. Ilustruje myšlenku hledání diferenčních stop. Tuto diferenční stopu můžeme hlavně rozšířit pro konfiguraci šifry Rijndael s větším počtem rundovních zobrazení. Ukážeme si to v následující kapitole.

2.3 Natažení diferenční stopy pro $\mathcal{R}(5,10,1)^*$

Objasníme si pojem natahovat diferenční stopu. Znamená to najít diferenční stopu pro konfiguraci šifry Rijndael s větším počtem rundovních zobrazení, která v sobě obsahuje původní diferenční stopu. Pokud natahujeme diferenční stopu dolů, přidáváme za stávající diferenční stopu další rundovní zobrazení a studujeme šíření diference. Obráceně, pokud natahujeme diferenční stopu nahoru, přidáváme rundovní zobrazení před stávající diferenční stopu a opět studujeme, jak se šíří diference.

Úmluva 2.5. *Ve všech obrázcích kapitoly 2.3 bílá políčka značí nulovou diferenci a šedivá políčka značí nenulovou diferenci. U šedivých políček velmi často budeme zaznamenávat hodnoty diference.*

Nyní si ukážeme, že není moc chytré se snažit natahovat diferenční stopu z kapitoly 2.2 dolů. Zkusme přidat dvě rundovní zobrazení a sledovat jak se šíří diference. Ilustrujeme si to na obrázku 2.2.

Jak jsme zmínili dříve, hledáme diferenční stopy, pro které platí $\triangle P = \triangle C = 0$ (značení dle obrázku 2.2). V tomto případě je komplikované splnit podmínku, aby matice $\triangle C = 0$. Tedy natahovat tuto diferenční stopu dolů není konstruktivní, budeme hledat jiný postup.

Zkusíme diferenční stopu z kapitoly 2.2 natáhnout o jedno rundovní zobrazení nahoru, tedy vytvořit diferenční stopu pro konfiguraci $\mathcal{R}(5,10,2)^*$. Tuto nataženou diferenční stopu ilustruje obrázek 2.3.

Nyní si objasněme, jak tato diferenční stopa vznikla. Budeme používat značení z obrázku 2.3. Matice A_2 A_3, A_4, $\triangle C$, K_2 a K_3 tvoří původní diferenční stopu z kapitoly 2.2, jejich hodnoty jsou tedy určené touto diferenční stopou. Prvky $x, y \in \mathbb{F}_{256}$ mají stejnou vlastnost jako v kapitole 2.2, tj. $D_p S(x,y) = 2^{-6}$. Matice A_1 musí být nulová, protože pouze nulová diference se pro bijektivní zobrazení šíří na nulovou diferenci dle poznámek 1.11 a 1.14. První sloupec matice K_1 a první sloupec matice K_3 se rovnají, protože poslední sloupec matice K_2 je nulový. První a druhý sloupec matice K_1 se rovnají, protože druhý sloupec matice K_3 a poslední sloupec matice K_2 jsou nulové. Takže první a druhý sloupec matice K_1 a první sloupec matice K_3 se rovnají. Protože matice A_1 je nulová, tedy matice $\triangle P = K_1$.

Tato diferenční stopa splnila nutnou podmínku pro platnost (věta 2.3), ale neplatí požadavek $\triangle P = 0$. Tedy tuto diferenční stopu nemůžeme použít pro naše účely.

Zkusme nyní diferenční stopu z kapitoly 2.2 natáhnout o dvě rundovní zobrazení nahoru, tedy vytvořit diferenční stopu pro konfiguraci $\mathcal{R}(5, 10, 3)^*$. Tuto nataženou diferenční stopu ilustrujeme na obrázku 2.4.

Myšlenka, jak tato diferenční stopa vznikla je velmi podobná jako pro diferenční stopu vzniklou natažením nahoru o jedno rundovní zobrazení. Proto si ukážeme jen zkrácený popis jejího vzniku. Budeme používat značení z obrázku 2.4. Matice A_4, A_5, A_6, $\triangle C$, K_3 a K_4 tvoří původní diferenční stopu z kapitoly 2.2 a tedy známe jejich hodnoty. Pro x a y opět platí $D_pS(x, y) = 2^{-6}$. První sloupec matice K_4 a první a druhý sloupec matice K_2 se rovnají, protože druhý sloupec matice K_4 a poslední sloupec K_3 jsou nulové. První sloupec matice K_3 a první a druhý sloupec matice K_1 se rovnají, protože druhý sloupec K_3 a poslední sloupec K_2 jsou nulové. Matice A_3 je nulová, protože pouze nulová diference se pro bijektivní zobrazení šíří na nulovou diferenci. Toto plyne z poznámek 1.11 a 1.14. Matice A_2 se rovná matici K_2, protože matice A_3 je nulová. Pro matice A_1 a A_2 musí platit následující vztah: $A_2 = \mathcal{MC}(\mathcal{SR}(A_1))$ (kapitola 2.1). Protože pro x a y platí $D_pS(x, y) = 2^{-6}$, tedy x a y je kompatibilní diferenční pár, a proto můžeme $\triangle P$ položit rovno nule.

Takto vytvořená diferenční stopa splňuje nutnou podmínku pro platnost (věta 2.3) a také platí $\triangle P = \triangle C = 0$. Tato diferenční stopa obsahuje tři aktivní S-boxy. Našli jsme diferenční stopu pro $\mathcal{R}(5, 10, 3)^*$, která splňuje všechny požadavky, které jsme si stanovili v této kapitole.

Ukázali jsme si základní myšlenku, jak natáhnout diferenční stopu z kapitoly 2.2 nahoru. Pokud bychom se snažili natáhnout tuto diferenční stopu o tři nebo pět rundovních zobrazení nahoru, tak bychom se setkali se stejným problémem jako při natažení této diferenční stopy o jedno rundovní zobrazení nahoru, tedy $\triangle P \neq 0$, kde $\triangle P$ je rozdíl otevřených textů. Natažení diferenční stopy z kapitoly 2.2 o čtyři nebo šest rundovních zobrazení nahoru nám dá diferenční stopy, které splňují nutnou podmínku pro platnost (věta 2.3) a požadovanou podmínku $\triangle P = \triangle C = 0$, kde $\triangle P$ a $\triangle C$ jsou rozdíly otevřených a šifrových textů. Úvahy, jak tyto diferenční stopy vznikly, jsou velmi podobné úvahám vzniku diferenční stopy pro konfiguraci $\mathcal{R}(5, 10, 3)^*$. Na obrázku 2.5 si ilustrujeme diferenční stopu pro konfiguraci $\mathcal{R}(5, 10, 7)^*$. Diferenční stopa pro konfiguraci $\mathcal{R}(5, 10, 5)^*$ vznikne odebráním prvních dvou rundovních zobrazení z diferenční stopy pro konfiguraci $\mathcal{R}(5, 10, 7)^*$.

Celkem jsme našli diferenční stopy splňující nutnou podmínku pro platnost (věta 2.3) a podmínku o nulové diferenci otevřených a šifrových textů pro různé konfigurace šifry Rijndael. V dalším textu budem ověřovat platnost diferenční stopy pro konfiguraci $\mathcal{R}(5, 10, 5)^*$. Tabulka 2.1 zaznamenává konfigurace, pro které jsme našli tyto diferenční stopy a počet aktivních S-boxů pro dané diferenční stopy.

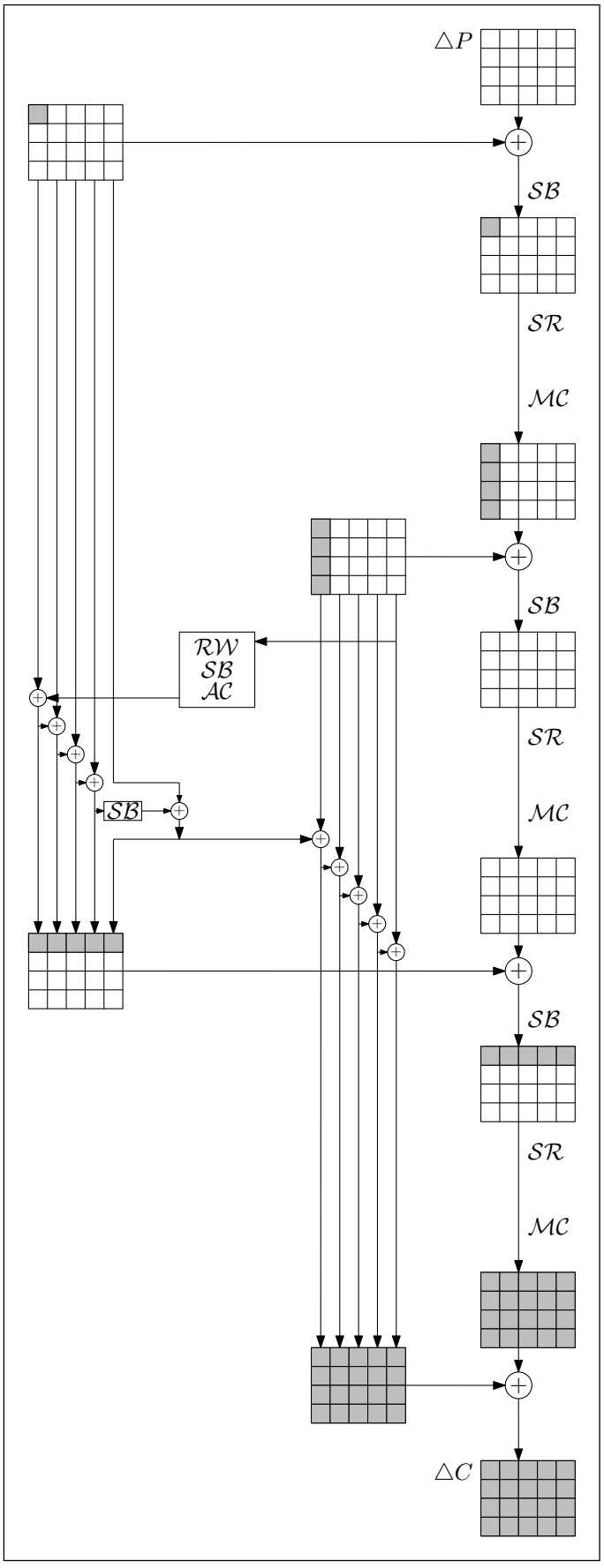
Kdybychom se snažili tímto postupem vytvořit diferenční stopu pro konfiguraci $\mathcal{R}(5, 10, 9)^*$, museli bychom řešit průchod nenulové diference skrz zobrazení $\mathcal{RW}$, $\mathcal{SB}$ a $\mathcal{AC}$ při generování rundovních klíčů. Toto by silně zkomplikovalo hledání $P \in \mathcal{P}$ a $K \in \mathcal{K}$, které by splňovaly tuto diferenční stopu.

Ještě si všimněme, že volba dát nenulovou diferenci do prvního sloupce, při

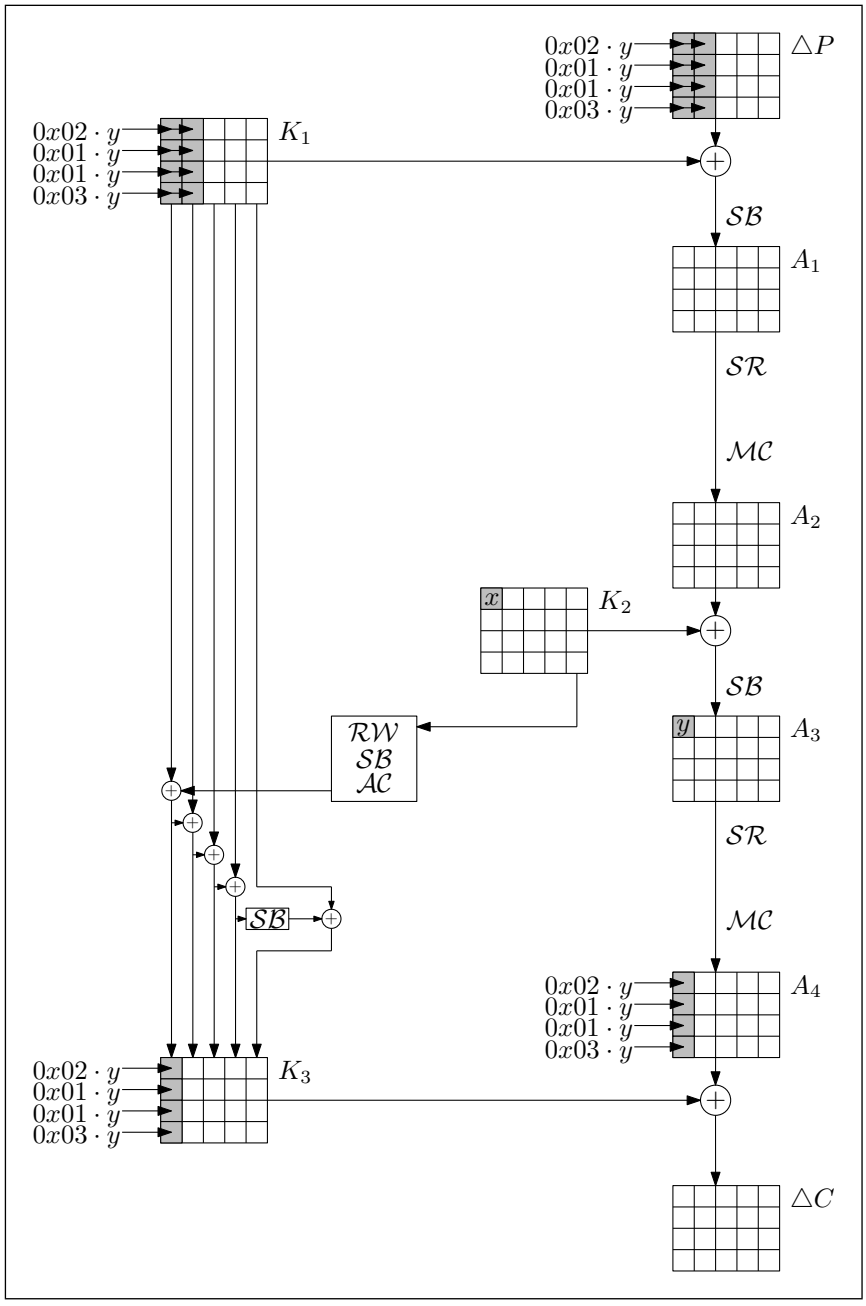
$\mathcal{R}(5, 10, 1)^*$	1
$\mathcal{R}(5, 10, 3)^*$	3
$\mathcal{R}(5, 10, 5)^*$	5
$\mathcal{R}(5, 10, 7)^*$	9

Tabulka 2.1: Nalezené diferenční stopy s počtem aktivních S-boxů

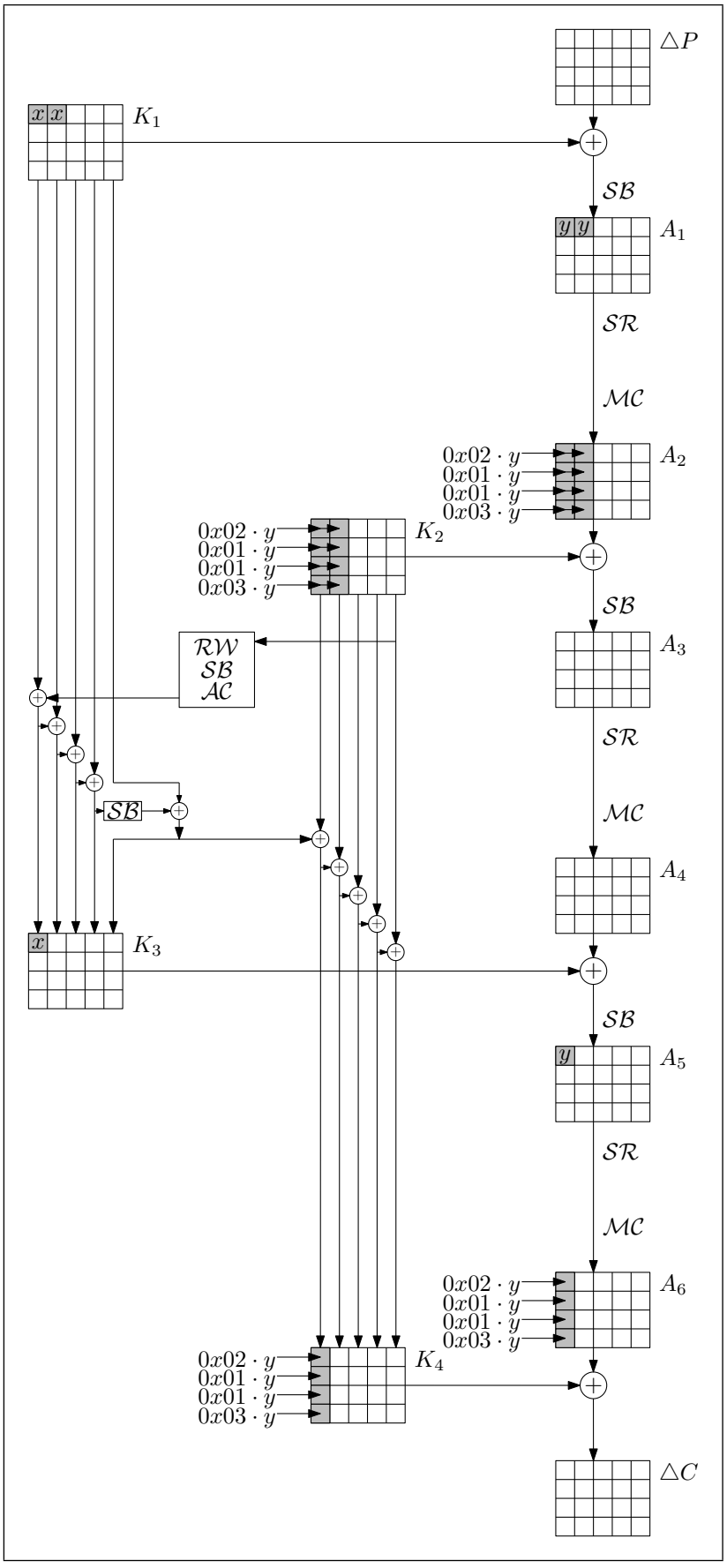
vytváření diferenční stopy pro konfiguraci $\mathcal{R}(5, 10, 1)^*$ (kapitola 2.2) měla význam při natahování této diferenční stopy nahoru. Takto jsme nemuseli řešit průchod nenulové diference skrz zobrazení $\mathcal{RW}$, $\mathcal{SB}$ a $\mathcal{AC}$ při generování rundovních klíčů.



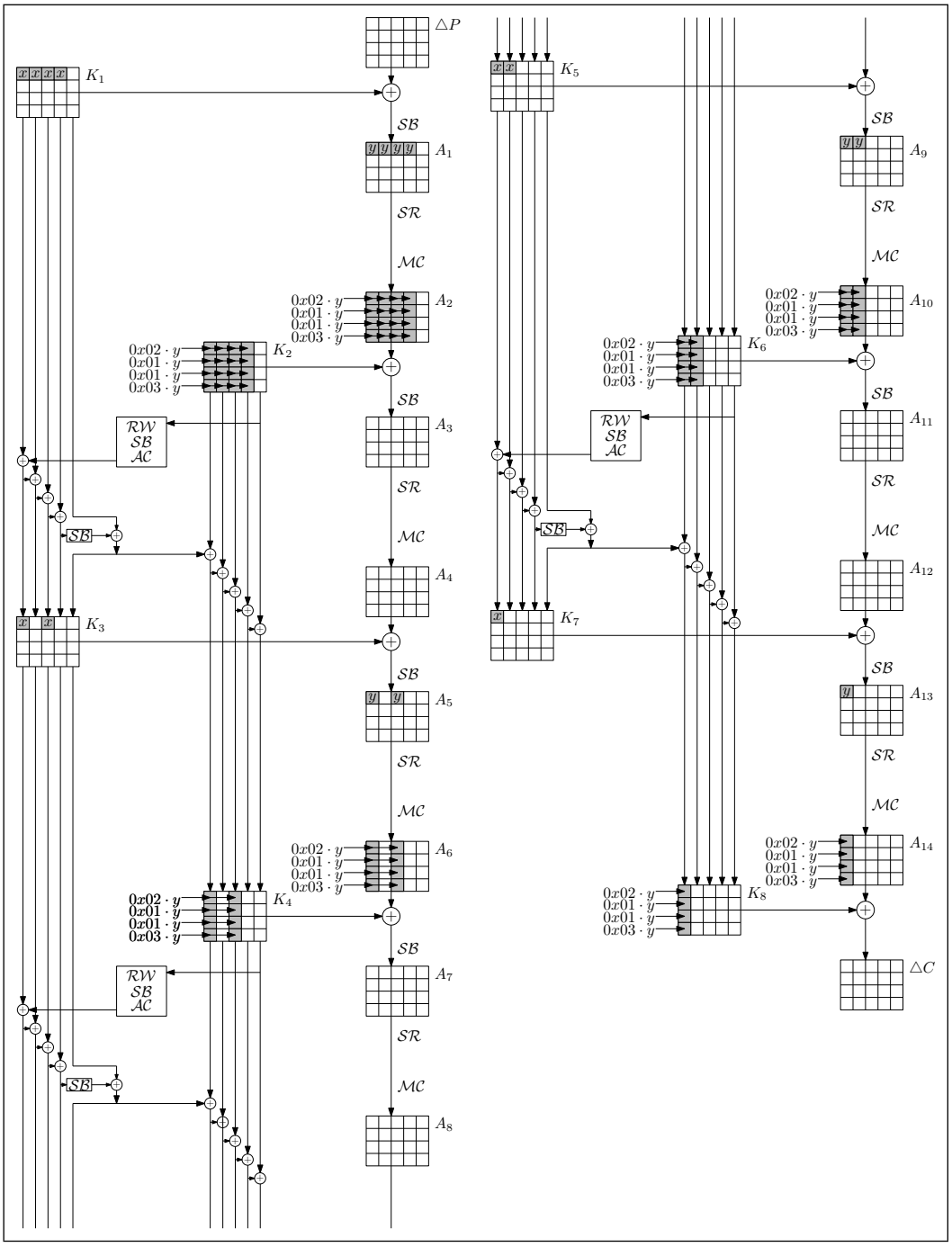
Obrázek 2.2: Diferenční stopa natažená o dvě rundovní zobrazení dolů



Obrázek 2.3: Diferenční stopa natažená o jedno rundovní zobrazení nahoru



Obrázek 2.4: Diferenční stopa natažená o dvě rundovní zobrazení nahoru



Obrázek 2.5: Diferenční stopa natažená o šest rundovních zobrazení nahoru

3. Triangulační algoritmus

V této kapitole si přesně definujeme triangulační algoritmus, jehož nástin byl uveden v článku [1]. Dále si pro myšlenky týkající se tohoto algoritmu z článku [1] zadefinujeme vlastní pojmy. Následně pomocí těchto pojmů vyslovíme zmíněné myšlenky v matematických větách a doplníme je o vlastní důkazy. V podkapitole 3.1 si uvedeme nejnutnější pojmy k definici trian. alg. a samotný algoritmus. V podkapitole 3.2 si definujeme pojmy, na kterých vysvětlíme vlastnosti nelineárních soustav rovnic, na nichž úspěšně proběhl trian. alg. Ověříme správnost tohoto algoritmu, ukážeme souvislosti mezi existencí řešení nelineární soustavy rovnic a úspěšnou triangulací této soustavy. Uvedeme a dokážeme podmínky za jakých můžeme provést triangulaci na nelineární soustavě rovnic. V podkapitole 3.3 si ukážeme, jak lze využít nejednoznačnosti trian. alg. Uvedeme si teoretický základ jak kombinovat trian. alg. s jinými metodami. V podkapitole 3.4 si ukážeme jak sestavit nelineární soustavu rovnic reprezentující průběh šifrování šifrou Rijndeal a základní vlastnosti takto vytvořené nelineární soustavy rovnic. V podkapitole 3.5 si ukážeme postup ověřování diferenční stopu s neznámou platností pomocí již získaných poznatků.

3.1 Základní popis

Definice 3.1. *Nechť $n, m, k \in \mathbb{N}$, $\mathbb{F}$ těleso, $H_1, \ldots, H_n$ jsou bijekce $\mathbb{F}$ a $X = \{x_1, \ldots, x_k\}$ množinu proměnných. Následně definujeme $\forall i \in \{1, \ldots, m\}$ podmnožiny $X_i \subseteq X$ a hodnotu $k_i := |X_i|$ a funkce $f_i(X_i)$ z $\mathbb{F}^{k_i}$ do $\mathbb{F}$ s následujícími vlastnostmi:*

1. *Pokud na operace a na prvky tělesa nahlížíme jako na funkce, potom funkci f_i jsme schopni zapsat jako složení těchto funkcí a bijekcí $H_1, \ldots, H_n$.*
2. *Pro libovolné $l \in \{1, \ldots, k_i\}$ a pro libovolní hodnoty $a_1, \ldots, a_{k_i} \in \mathbb{F}$ takové, že platí $f_i(a_1, \ldots, a_{k_i}) = 0$, lze hodnota a_l určit jednoznačně z hodnot a_j $\forall j \in \{1, \ldots, k_i\} \setminus \{l\}$*

Potom definujeme:

1. *$\forall i \in \{1, \ldots, m\}$ rovnost $f_i(X_i) = 0$ jako nelineární rovnici nad tělesem $\mathbb{F}$ o k_i proměnných obsahující bijekce $H_1, \ldots, H_n$.*
2. *množinu rovností $\rho(m, k, n, \mathbb{F}) = \{f_1(X_1) = 0, \ldots, f_m(X_k) = 0\}$ jako soustavu nelineárních rovnic nad tělesem $\mathbb{F}$ o m nelineárních rovnicích a k proměnných obsahující bijekce $H_1, \ldots, H_n$.*

Příklad 3.2. *Nyní si ukažme jednoduchý příklad rovnice, která je v rozporu s vlastností 2 definice 3.1. Nechť máme $\mathbb{F}_5$ a funkci $f(x_1, x_2) = x_1^2 + x_2$. Dosaďme hodnoty $x_1 = 2$ a $x_2 = 1$, potom platí, že $f(2, 1) = 2^2 + 1 = 0$ a ze znalosti*

hodnoty $x_2 = 1$ jednoznačně neurčíme hodnotu x_1:

$$\begin{aligned} x_1^2 + 1 &= 0 \\ x_1^2 &= 4 \\ x_1 &= 2 \\ x_1 &= 3 \end{aligned}$$

Příklad 3.3. *Máme zadanou soustavu nelineárních rovnic $\rho(4, 7, 3, \mathbb{F}_3)$ následujícím předpisem:*
množina proměnných: $X = \{x_1, \ldots, x_7\}$
bijektivní funkce (značení dle úmluvy 1.12):
$H_1 = \begin{pmatrix} 0 & 1 & 2 \\ 2 & 1 & 0 \end{pmatrix}$, $H_2 = \begin{pmatrix} 0 & 1 & 2 \\ 0 & 2 & 1 \end{pmatrix}$, $H_3 = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \end{pmatrix}$
rovnice:

$$2x_1 + 2x_3 + x_4 + x_6 + x_7 = 0 \tag{3.1}$$

$$H_2(x_1) + H_1(x_2 + x_5) + x_6 + 2 = 0 \tag{3.2}$$

$$2H_3(x_2 + x_4) + 2H_2(x_3) + x_7 + 2 = 0 \tag{3.3}$$

$$H_2(x_1) + 2H_1(x_3 + x_6) + 1 = 0 \tag{3.4}$$

Úmluva 3.4. *Pro zbytek této kapitoly je $n, m, k \in \mathbb{N}$ a $\mathbb{F}$ je těleso.*

Definice 3.5. *Nechť $\rho(m, k, n, \mathbb{F})$ je soustava nelineárních rovnic, X ($|X| = k$) je množina proměnných soustavy $\rho(m, k, n, \mathbb{F})$. Nechť $V \subseteq X$ taková, že po dosazení hodnot proměnných odpovídajících řešení soustavy $\rho(m, k, n, \mathbb{F})$ z V do soustavy $\rho(m, k, n, \mathbb{F})$, lze jednoznačně určit hodnoty ostatních proměnných odpovídajících danému řešení soustavy $\rho(m, k, n, \mathbb{F})$ z množiny X. Pak V nazýváme množinu volných proměnných soustavy $\rho(m, k, n, \mathbb{F})$ a $v \in V$ nazýváme volnou proměnnou soustavy $\rho(m, k, n, \mathbb{F})$.*

Definice 3.6. (Popis triangulačního algoritmu) *Nechť $\rho(m, k, n, \mathbb{F})$ je soustava nelineárních rovnic a X je množina proměnných $\rho(m, k, n, \mathbb{F})$. Vstupem trian. alg. je soustava nelineárních rovnic $\rho(m, k, n, \mathbb{F})$ a výstupem je $V \subseteq X$ množina volných proměnných soustavy $\rho(m, k, n, \mathbb{F})$. Popis trian. alg.:*

1. *Všechny rovnice a proměnné označíme jako nezpracované*
2. *Najdeme nezpracovanou proměnou, které je obsažená pouze v jedné nezpracované rovnici. Danou rovnici a proměnnou označíme za zpracované a přiřadíme jim pořadí, v kterém byly zpracovány.*
3. *Bod 2 opakujeme dokud je to možné. Konec opakování nastane, jestliže:*
   (a) *Všechny rovnice jsou zpracovány, potom triangulační algoritmus proběhl úspěšně a množina všech nezpracovaných proměnných tvoří množinu volných proměnných soustavy $\rho(m, k, n, \mathbb{F})$*
   (b) *Pokud existuje nezpracovaná rovnice a již nemůžeme najít nezpracovanou proměnou obsaženou pouze v jedné nezpracované rovnici. V tomto případě triangulační algoritmus byl neúspěšný.*

Poznámka 3.7. *Nechť $\rho(m, k, n, \mathbb{F})$ je soustava nelineárních rovnic, na které úspěšně proběhl trian. alg. Zaznamenávání pořadí zpracování rovnic a proměnných v průběhu trian. alg. (definice 3.6) je užitečné, když známe hodnoty volných proměnných odpovídající řešení soustavy a chceme dopočítat hodnoty ostatních proměnných. Postupně dosazujeme hodnoty proměnných do rovnic v opačném pořadí, než byly zpracovány. Tímto postupem získáme řešení soustavy $\rho(m, k, n, \mathbb{F})$. Toto je možné z vlastnosti 2 definice 3.1 nelineární soustavy rovnic a za předpokladu, že hodnoty volných proměnných odpovídají řešení soustavy $\rho(m, k, n, \mathbb{F})$.*

Příklad 3.8. *Máme zadanou soustavu $\rho(4, 7, 3, \mathbb{F}_3)$ z příkladu 3.3. Nyní na ní ukážeme použití trian. alg.*
triangulační algoritmus použitý na zadanou soustavu rovnic:

1. *Proměnná x_5 je pouze v rovnici 3.2.*
2. *Proměnná x_2 je obsažená v rovnicích 3.2 a 3.3, ale rovnice 3.2 je již zpracována, tedy v kroku 2. zpracujeme proměnnou x_2 a rovnici 3.3.*
3. *Zpracujeme proměnnou x_4 a rovnici 3.1.*
4. *Zpracujeme proměnnou x_1 a rovnici 3.4.*

Množina volných proměnných je $V = \{x_3, x_6, x_7\}$. Za hodnoty volných proměnných zvolme: $x_3 = x_6 = x_7 = 0$ Začněme dosazovat do rovnic v opačném pořadí než jsme zpracovali rovnice v trian. alg.:

1. *dosazení do rovnice 3.4:*
   $H_2(x_1) + 2H_1(0 + 0) + 1 = 0$
   $H_2(x_1) + 2 \cdot 2 + 1 = 0$
   $H_2(x_1) = 1$
   $x_1 = 2$
2. *dosazení do rovnice 3.1:*
   $2 \cdot 2 + 2 \cdot 0 + x_4 + 0 + 0 = 0$
   $x_4 + 1 = 0$
   $x_4 = 2$
3. *dosazení do rovnice 3.3:*
   $2H_3(x_2 + 2) + 2H_2(0) + 0 + 2 = 0$
   $2H_3(x_2 + 2) + 2 \cdot 0 + 0 + 2 = 0$
   $H_3(x_2 + 2) = 2$
   $x_2 = 2$
4. *dosazení do rovnice 3.2:*
   $H_2(2) + H_1(2 + x_5) + 0 + 2 = 0$
   $H_1(x_5 + 2) + 1 + 2 + 0 = 0$
   $H_1(x_5 + 2) = 0$
   $x_5 = 0$

Tedy řešením soustavy $\rho(4,7,3,\mathbb{F}_3)$ *je* $\bar{x} = \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \end{matrix}$
$$\begin{pmatrix} 2 & 2 & 0 & 2 & 0 & 0 & 0 \end{pmatrix}$$

3.2 Vlastnosti

Definice 3.9. *Nechť* $\rho(m,k,n,\mathbb{F})$ *je soustava nelineárních rovnic. Pak definujeme matici závislosti příslušející soustavě* $\rho(m,k,n,\mathbb{F})$ $A_\rho \in {\mathbb{F}_2}^{m\times k}$ *s hodnotami* $a_{ij} = 0$*, pokud* j*-tá proměnná není obsažená v* i*-té rovnici a naopak* $a_{ij} = 1$*, pokud* j*-tá proměnná je obsažená v* i*-té rovnici pro* $i \in \{1,\dots,m\}$ *a* $j \in \{1,\dots,k\}$*. Tedy* i*-tý řádek obsahuje informaci o tom, která proměnná je obsažená v* i*-té rovnici a* j*-tý sloupec obsahuje informaci o tom, v které rovnici je obsažená* j*-tá proměnná.*

Pokud na soustavě $\rho(m,k,n,\mathbb{F})$ *proběhl úspěšně trian. alg.(definice 3.6), potom definujeme matici závislosti po triangulaci příslušející soustavě* $\rho(m,k,n,\mathbb{F})$ $\tilde{A}_\rho \in {\mathbb{F}_2}^{m\times k}$*, která vznikne permutováváním řádků a sloupců matice* A_ρ *dle pořadí, v kterém byly zpracovány odpovídající rovnice a proměnné v trian. alg. Tedy v* i*-tém řádku pro* $i \in \{1,\dots,m\}$ *je zaznamenána rovnice, která byla zpracována jako* i*-tá v trian. alg. a v* j*-tém sloupci pro* $j \in \{1,\dots,m\}$ *je zaznamenána proměnná, která byla zpracována jako* j*-tá v trian. alg. V* j*-tém sloupci pro* $j \in \{m+1,\dots,k\}$ *jsou zaznamenány proměnné, které nebyly zpracovány v trian. alg., v libovolném pořadí.*

Příklad 3.10. *Máme zadanou soustavu* $\rho(4,7,3,\mathbb{F}_3)$ *z příkladu 3.3. Na této soustavě úspěšně proběhl trian. alg. (příklad 3.8), existují tedy pro zadanou soustavu* $\rho(4,7,3,\mathbb{F}_3)$ *matice závislosti* A_ρ *a matice závislosti po triangulaci* $\tilde{A}_\rho$*:*

$$A_\rho = \begin{matrix} \\ 3.1 \\ 3.2 \\ 3.3 \\ 3.4 \end{matrix} \begin{matrix} \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \end{matrix} \\ \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix} \end{matrix}$$

$$\tilde{A}_\rho = \begin{matrix} \\ 3.2 \\ 3.3 \\ 3.1 \\ 3.4 \end{matrix} \begin{matrix} \begin{matrix} x_5 & x_2 & x_4 & x_1 & x_3 & x_6 & x_7 \end{matrix} \\ \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} \end{matrix}$$

Věta 3.11. *Nechť* $\rho(m,k,n,\mathbb{F})$ *je soustava nelineárních rovnic. Pokud úspěšně proběhl trian. alg.(definice 3.6), potom* $m \le k$*.*

Důkaz. sporem předpokládejme, že trian. alg. proběhl úspěšně a zároveň $m > k$. Po k krocích trian. alg. již neexistuje nezpracovaná proměnná a existuje $m-k > 0$ nezpracovaných rovnic. Tedy trian. alg. skončí v bodě 3b z definice trian. alg. (definice 3.6): existuje nezpracovaná rovnice a zároveň neexistuje nezpracovaná proměnná obsažená pouze v jedné nezpracované rovnici, což je spor s tím, že trian. alg. proběhl úspěšně. □

Věta 3.12. *Nechť* $\rho(m,k,n,\mathbb{F})$ *je soustava nelineárních rovnic, na které úspěšně proběhl trian. alg.(definice 3.6), potom matice závislosti po triangulaci* $\tilde{A}_\rho \in {\mathbb{F}_2}^{m\times k}$ *má tyto vlastnosti:*

1. *$\tilde{A}_\rho$ je horní trojúhelníková matice*
2. *$\tilde{A}_\rho$ má na diagonále jedničky*
3. *proměnné zaznamenané v sloupcích $m+1$ až k matice $\tilde{A}_\rho$ tvoří množinu volných proměnných soustavy $\rho(m,k,n,\mathbb{F})$.*

Důkaz.

1. Chceme, aby pro $\forall j \in \{1,\ldots,m-1\}$ a $\forall i \in \{j+1,\ldots,m\}$ platilo $a_{ij}=0$, kde a_{ij} jsou prvky $\tilde{A}_\rho$. Pro libovolné $j \in \{1,\ldots,m-1\}$ proměnná zaznamenaná v j-tém sloupci $\tilde{A}_\rho$ byla zpracována v j-tém kroku trian. alg. V j-tém kroku trian. alg. byly nezpracovány právě ty rovnice, které jsou zaznamenány v řádcích $j+1$ až m v matici $\tilde{A}_\rho$ a z bodu 2 trian. alg. (definice 3.6) plyne, že v rovnicích zaznamenaných na $j+1$ až m řádku v matici $\tilde{A}_\rho$ nebyla obsažena j-tá proměnná, tedy $a_{ij}=0$ pro $\forall j \in \{1,\ldots,m-1\}$ a $\forall i \in \{j+1,\ldots,m\}$.
2. Chceme aby platilo $a_{ii}=1$ pro $\forall i \in \{1,\ldots,m\}$. Pro libovolné $i \in \{1,\ldots,m\}$ víme, že proměnná zaznamenaná v i-tém sloupci matice $\tilde{A}_\rho$ byla zpracována v i-tém kroku trian. alg. a z trian. alg. (definice 3.6) plyne, že byla obsažena v rovnici zaznamenané v i-tém řádku $\tilde{A}_\rho$ a tedy $a_{ii}=1$.
3. Proměnné zaznamenané ve sloupcích $m+1$ až k v matici $\tilde{A}_\rho$ byly nezpracované trian. alg. tedy jsou volné.

□

Poznámka 3.13. *Ve Větě 3.12 bod 3 předpokládáme, že trian. alg. funguje správně, tedy jeho výstupem je opravdu množina volných proměnných. Správnost trian. alg. bude dokázaná později.*

Věta 3.14. (Správnost triangulačního algoritmu) *Nechť $\rho(m,k,n,\mathbb{F})$ je soustava nelineárních rovnic, na které úspěšně proběhl trian. alg.(definice 3.6) Pak výstupem trian. alg. je množina volných proměnných.*

Důkaz. Nechť $\bar{x}$ je libovolné řešení soustavy $\rho(m,k,n,\mathbb{F})$. Bez újmy na obecnosti předpokládejme, že hodnoty v $\bar{x}$ jsou stejně uspořádané jako proměnné v matici závislosti po triangulaci $\tilde{A}_\rho$. Chceme dokázat, že ze znalosti hodnot x_{m+1} až x_k vektoru $\bar{x}$ lze získat zbývající hodnoty x_1 až x_m vektoru $\bar{x}$. Pokud pro libovolné $i \in \{1,\ldots,m\}$ znám hodnoty x_{i+1} až x_k z vektoru $\bar{x}$, tak hodnotu x_i můžeme dopočítat dosazením hodnot x_{i+1} až x_k z vektoru $\bar{x}$ do rovnice zaznamenané v i-tém řádku matice $\tilde{A}_\rho$. Toto můžeme provést, protože $\bar{x}$ je řešení soustavy $\rho(m,k,n,\mathbb{F})$ a díky vlastnosti 2 definice 3.1 nelineární soustavy rovnic a také proto, že vždy v rovnici zaznamenané na i-tém řádku matice $\tilde{A}_\rho$ jsou obsaženy proměnné i až k dle uspořádání proměnných v $\tilde{A}_\rho$. Tímto induktivním postupem můžeme dopočítat z hodnot x_{m+1} až x_k vektoru $\bar{x}$ zbývající hodnoty vektoru $\bar{x}$. Tedy trian. alg. vrací správný výsledek. □

Poznámka 3.15. *Nechť $\rho(m,k,n,\mathbb{F})$ je soustava nelineárních rovnic, na které úspěšně proběhl trian. alg.(definice 3.6). Je dobré si uvědomit, že úspěšnost trian. alg. nic nevypovídá o existenci řešení soustavy $\rho(m,k,n,\mathbb{F})$, viz následující příklad.*

Příklad 3.16. *Máme zadanou soustavu nelineárních rovnic $\rho(2,2,0,\mathbb{F}_3)$ následujícím předpisem:*
množina proměnných: $X = \{x_1, x_2\}$
rovnice:

$$x_1 \cdot x_2 + 1 = 0 \tag{3.5}$$

$$x_2 = 0 \tag{3.6}$$

Množina řešení rovnice 3.5 je $M_1 = \{(1,2),(2,1)\}$ a množina řešení rovnice 3.6 je $M_1 = \{(0,0),(1,0),(2,0)\}$. Všimněme si, že na této nelineární soustavě rovnic úspěšně proběhne trian. alg., ale $M_1 \cap M_2 = \emptyset$. Tedy neexistuje řešení této soustavy. Tato soustav nelineárních rovnic splňuje definice 3.1. Vlastnost 1 z této definice je splněna zřejmě a vlastnost 2 je splněna zřejmě pro rovnici 3.6 a pro rovnici 3.5:

- *$x_1 = 1$ potom $1 \cdot x_2 + 1 = 0$, tedy $x_2 = 2$*
- *$x_1 = 2$ potom $2 \cdot x_2 + 1 = 0$, tedy $x_2 = 1$*

Pro proměnou x_2 je to obdobné, tedy rovnice 3.6 splnila vlastnost 2 z definice 3.1.

Definice 3.17. *Nechť $k \in \mathbb{N}$, $X = \{x_1, \ldots, x_k\}$ je množina proměnných a $f(X)$ je funkce z $\mathbb{F}^k$ do $\mathbb{F}$. Řekněme, že funkce f je bijekce v každé proměnné pokud pro libovolné $l \in \{1, \ldots, k\}$ a pro libovolné hodnoty $a_1, \ldots, a_{l-1}, a_{l+1}, \ldots, a_k \in \mathbb{F}$ platí , že $f(a_1, \ldots, a_{l-1}, \cdot, a_{l+1}, \ldots, a_k)$ je bijekce $\mathbb{F}$*

Poznámka 3.18. *Nechť $k \in \mathbb{N}$, $X = \{x_1, \ldots, x_k\}$ je množina proměnných a $f(X)$ je funkce z $\mathbb{F}^k$ do $\mathbb{F}$. Pokud f je bijekce v každé proměnné, potom f splňuje vlastnost 2 z definice 3.1.*

Důkaz. Nechť máme libovolné hodnoty $a_1, \ldots, a_k \in \mathbb{F}$ takové, že $f(a_1, \ldots, a_k) = 0$. Zvolme libovolné $l \in \{1, \ldots, k\}$, potom dle předpokladu $f(a_1, \ldots, a_{l-1}, \cdot, a_{l+1}, \ldots, a_k)$ je bijekce $\mathbb{F}$, tedy můžeme jednoznačně určit hodnotu a_l z rovnosti: $f(a_1, \ldots, a_{l-1}, \cdot, a_{l+1}, \ldots, a_k) = 0$. □

Věta 3.19. *Nechť $\rho(m,k,n,\mathbb{F})$ je nelineární soustava rovnic a X je množina proměnných $\rho(m,k,n,\mathbb{F})$. Platí, že na $\rho(m,k,n,\mathbb{F})$ úspěšně proběhl trian. alg. a zároveň $\forall i \in \{1, \ldots, m\}$ platí, že funkce f_i, která určuje i-tou rovnici v $\rho(m,k,n,\mathbb{F})$ je bijekce v každé proměnné (definice 3.17). Potom dosazení libovolných hodnot za volné proměnné dává řešení soustavy $\rho(m,k,n,\mathbb{F})$.*

Důkaz. Předpokládejme bez újmy na obecnosti, že proměnné a rovnice jsou uspořádaný jako v matici závislosti po triangulaci $\tilde{A}_\rho$. Nyní $\forall i \in \{1, \ldots, m\}$ f_i je funkce, která určuje rovnici zpracovanou v i-tém kroku trian. alg. a X_i je množina proměnných funkce f_i a platí z definice trian. alg., že $x_i \in X_i$ a zároveň $X_i \subseteq \{x_i, \ldots, x_k\}$. Nyní zvolme zcela libovolně $a_{i+1}, \ldots, a_k \in \mathbb{F}$, dosaďme patřičné hodnoty do funkce $f_i(\cdot, a_{i+1}, \ldots, a_k)$. O této funkci víme z předpokladu, že je to bijekce tělesa $\mathbb{F}$, tedy umíme určit jednoznačně hodnotu $a_i \in \mathbb{F}$, tak aby platilo $f_i(a_i, a_{i+1}, \ldots, a_k) = 0$.

Nyní zvolme libovolné hodnoty $b_{m+1}, \ldots, b_k \in \mathbb{F}$, dosaďme je do rovnice zpracované v m-tém kroku trian. alg. a dle předchozí úvahy jsme schopni jednoznačně určit hodnotu proměnné x_m. Následně opakováním tohoto induktivního postupu získáme celé řešení $\rho(m,k,n,\mathbb{F})$. □

Poznámka 3.20. *Trian. alg. (definice 3.6) nemůžeme použít na všechny soustavy nelineárních rovnic. Například pokud by byla zadaná soustava $\rho(m, k, n, \mathbb{F})$ a každá proměnná byla obsažena v každé rovnici, tak zřejmě nemůžeme použít trian. alg. na $\rho(m, k, n, \mathbb{F})$.*

Věta 3.21. *Nechť $\rho(m, k, n, \mathbb{F})$ je soustava nelineárních rovnic. Pak následující tvrzení jsou ekvivalentní:*

(i) *Na $\rho(m, k, n, \mathbb{F})$ lze úspěšně použít trian. alg.*

(ii) *$\exists \tilde{A}_\rho$ matice závislosti po triangulaci, tedy lze permutovat řádky a sloupce matice A_ρ tak, aby vznikla horní trojúhelníková matice s jedničkami na diagonále.*

(iii) *$\forall i \in \{0, \ldots, m-1\}$ $\exists i$ rovnic, které když odebereme z $\rho(m, k, n, \mathbb{F})$ tak, že na nově vzniklé soustavě rovnic můžeme úspěšně aplikovat trian. alg.*

Důkaz.

(i) $\Rightarrow$ (ii) Plyne z definice 3.9 matice závislosti po triangulaci a věty 3.12.

(ii) $\Rightarrow$ (i) Předpokládejme, že v matici závislosti A_ρ příslušející soustavě $\rho(m, k, n, \mathbb{F})$ umíme permutovat řádky a sloupce, aby vznikla námi požadovaná matice $\tilde{A}_\rho$. Potom v i-tém kroku trian. alg. zpracujeme rovnici zapsanou na i-tém řádku a proměnnou zapsanou v i-tém sloupci v matici $\tilde{A}_\rho$ pro $i \in \{1, \ldots, m\}$. Takto vytvořený postup odpovídá trian. alg. (definice 3.6), protože daná proměnná je nezpracovaná a je obsažená pouze v jedné nezpracované rovnici.

(ii) $\Rightarrow$ (iii) Pro libovolné $i \in \{0, \ldots, m-1\}$ definujeme soustavu ρ_i, která vznikne odebráním i rovnic zapsaných v prvních i řádcích matice závislosti po triangulaci $\tilde{A}_\rho$ příslušející soustavě ρ. Matice závislosti po triangulaci $\tilde{A}_{\rho_i}$ vznikne odebráním prvních i řádků a sloupců z matice $\tilde{A}_\rho$. Tedy pro soustavu ρ_i existuje matice závislosti po triangulaci, tedy můžeme úspěšně použít trian. alg. na ρ_i.

(iii) $\Rightarrow$ (i) zřejmé □

3.3 Variabilita triangulačního algoritmu

Věta 3.22. (o velikosti množiny volných proměnných) *Nechť $\rho(m, k, n, \mathbb{F})$ je soustava nelineárních rovnic, na které úspěšně proběhl trian. alg.(definice 3.6) Pokud existují dvě různé triangulace soustavy $\rho(m, k, n, \mathbb{F})$, pak označme V_1 a V_2 množiny volných proměnných odpovídající příslušným triangulacím. Potom $|V_1| = |V_2|$.*

Důkaz. Počet volných proměnných je roven počtu nezpracovaných proměnných v trian. alg. a to jsou proměnné zaznamenané v matici závislosti po triangulaci $\tilde{A}_\rho$ v $m+1$ až k-tém sloupci, tedy $|V_1| = k - m = |V_2|$. □

Poznámka 3.23. *Nechť $\rho(m, k, n, \mathbb{F})$ je soustava nelineárních rovnic, na které lze provést triangulaci více způsoby. Existuje tedy krok v trian. alg.(definice 3.6), v kterém můžeme vybrat více nezpracovaných proměnných, které jsou obsaženy pouze v jedné nezpracované rovnici. Tato situace může nastat pouze ve dvou případech a jejich kombinacích:*

1. *Existují dvě a více nezpracovaných proměnných, které jsou obsaženy právě v jedné nezpracované rovnici. Všimneme si, že tyto proměnné se navzájem určují. Tedy můžeme vybrat jakoukoliv z těchto proměnných a zpracovat ji a zbývající proměnné se tedy stanou volnými, protože určují právě zpracovanou proměnou. V tomto případě můžeme ovlivnit volbu volných proměnných, pro případ, že by to bylo výhodné pro další práci s volnými proměnnými soustavy $\rho(m, k, n, \mathbb{F})$.*

2. *Existují dvě a více nezpracovaných proměnných, každá obsažená právě v jedné a jiné nezpracované rovnici. Všimneme si, že pokud zpracujeme libovolnou z těchto proměnných a příslušnou nezpracovanou rovnici, tak zbytek výše zmíněných proměnných lze zpracovat v dalších krocích trian. alg., tedy nezáleží na pořadí zpracování rovnic a proměnných v tomto případě. Pokud chceme jednu z výše zmíněných proměnných jako volnou, tak se v tomto případě můžeme pokusit vybírat vždy jinou proměnnou pro zpracování.*

Závěrem lze říci, že nemůžeme ovlivnit počet volných proměnných, ale za určitých okolností můžeme ovlivnit výběr volných proměnných.

Definice 3.24. *Nechť $\mathbb{F}$ je těleso, $n, m, k, l, o \in \mathbb{N}$, $\rho(m, k, n, \mathbb{F})$, $\sigma(l, k, o, \mathbb{F})$ jsou dvě soustavy nelineárních rovnic a M_ρ, M_σ jsou příslušné množiny řešení daných soustav. Pak nazveme soustavy ρ a σ ekvivalentní právě tehdy když $M_\rho = M_\sigma$. Tuto relaci ekvivalence budeme značit $\rho(m, k, n, \mathbb{F}) \sim \sigma(l, k, o, \mathbb{F})$*

Poznámka 3.25. *Relace ekvivalence z definice 3.24 je opravdu ekvivalence, jelikož je reflexivní, symetrická, tranzitivní.*

Definice 3.26. *Nechť $\rho(m, k, n, \mathbb{F})$ je soustava nelineárních rovnic. Pak řekněme, že σ je podsoustavou ρ, pokud $\sigma \subseteq \rho$.*

Věta 3.27. *Nechť ρ, σ a σ' jsou soustavy nelineárních rovnic nad tělesem $\mathbb{F}$ a platí $\sigma \subseteq \rho$ a zároveň $\sigma \sim \sigma'$. Definujeme $\rho' := (\rho \setminus \sigma) \cup \sigma'$. Potom platí $\rho \sim \rho'$.*

Důkaz. Soustavy σ a σ' mají l proměnných a soustavy ρ a ρ' mají k proměnných a platí $l \leq k$, kde $k, l \in \mathbb{N}$.
Značení:

1. M_ρ, $M_{\rho'}$, M_σ a $M_{\sigma'}$ jsou množiny řešení příslušných soustav

2. $\tilde{M}_\sigma$ a $\tilde{M}_{\sigma'}$ jsou rozšířené množiny řešení soustav σ a σ'

Rozšíření ve smyslu:
M_σ je množina uspořádaných l-tic a $\tilde{M}_\sigma$ je množina uspořádaných $k - tic$. $\tilde{M}_\sigma$ vznikne tak, že každé $x \in M_\sigma$ doplníme o $k - l$ souřadnic na k-tici. Na původních pozicích x necháme původní hodnoty a na nové pozice doplníme všechny možné kombinace hodnot, tedy za každé $x \in M_\sigma$ bude v $\tilde{M}_\sigma$ $|\mathbb{F}|^{k-l}$ k-tic. Hodnoty prvků $\tilde{M}_\sigma$ jsou uspořádány (pořadí proměnných v zápisu) totožně s uspořádáním hodnot prvku M_ρ.
dále platí:

1. z předpokladu víme: $M_\sigma = M_{\sigma'} \Rightarrow \tilde{M}_\sigma = \tilde{M}_{\sigma'}$

2. $M_\rho \subseteq \tilde{M}_\sigma = \tilde{M}_{\sigma'} \supseteq M_{\rho'}$

3. $\rho \setminus \sigma = \rho' \setminus \sigma'$

Chceme dokázat $M_\rho = M_{\rho'}$.

"$\subseteq$"Zvolme libovolné $x \in M_\rho$ a předpokládejme, že $x \notin M_{\rho'}$. Tedy existuje $f \in \rho'$, pro kterou platí $f(x) \neq 0$. Mohou nastat dvě možnosti:

1. $f \in \sigma' \Rightarrow x \notin \tilde{M}_{\sigma'} = \tilde{M}_\sigma \supseteq M_\rho$ a zároveň $x \in M_\rho$ SPOR
2. $f \in \rho' \setminus \sigma' = \rho \setminus \sigma$ a zároveň $f(x) \neq 0$, ale $f \in \rho$ a x je řešením soustavy ρ tedy platí $f(x) = 0$ SPOR

"$\supseteq$"obdobně □

Důsledek 3.28.

1. *Pokud máme zadanou nelineární soustavu rovnic $\rho(m, k, n, \mathbb{F})$, která nelze triangulovat, lze se pokusit vzít podsoustavu σ, která zabraňuje triangulaci a najít k ní ekvivalentní soustavu σ', která umožní, že ρ' (značení jako ve větě 3.27) lze již triangulovat. Příklad této situace je soustava ρ, ve které se nachází podsoustava σ skládající se z lineárních rovnic. Tedy na σ lze aplikovat Gaussovu eliminační metodu a vytvořit σ' v Jordanově tvaru.*
2. *Nechť máme zadanou nelineární soustavu rovnic $\rho(m, k, n, \mathbb{F})$ a její podsoustavu σ, kterou umíme upravit na jinou soustavu σ' ekvivalentní se σ a pro další účely více vyhovující. Takto za předpokladu, že nepokazíme možnost triangulace, můžeme použít jiné postupy k upravení soustavy ρ na ekvivalentní ρ'. Zde se znovu nabízí použití Gaussovy eliminační metody na lineární podsoustavy soustavy ρ.*

Příklad 3.29. *Nechť $\rho(6, 10, n, \mathbb{F})$ soustava nelineárních rovnic a A_ρ je její matice závislost:*

$$A_\rho = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Na ρ nelze provést triangulaci. Za předpokladu, že rovnice zaznamenané v posledních dvou řádcích matice A_ρ jsou lineární, tak na ně můžeme použít Gaussovu eliminační metodu a takto upravit soustavu ρ, aby šla provést triangulace.

3.4 Triangulační algoritmus a šifra Rijndael

Nyní si ukážeme, jak sestavit soustavu rovnic, která reprezentuje průběh šifrování šifrou Rijndael. Rovnice budeme sestavovat pro šifrový diagram 1.21.

Definice 3.30. *Nechť máme $\mathcal{R}(N_b, N_k, N_r)$ nebo $\mathcal{R}(N_b, N_k, N_r)^*$ a šifrový diagram 1.21 pro tuto konfiguraci*

$$\mathcal{D} = ((P, A^0, \ldots, A^{N_r}), (K^0, \ldots, K^{N_r}), (\alpha_0[K], \ldots, \alpha_{N_r}[K]))_{P,K}$$

horní indexy používáme proto, že spodní indexy budeme používat ke značení prvků matic. Definujme nelineární soustavu rovnic:

- *Proměnné jsou prvky matic* $P, A^0, \ldots, A^n, K^0, \ldots, K^{N_r}$. *Zde na tyto prvky nenahlížíme jako na prvky* $\mathbb{F}_{256}$, *ale jako na proměnné soustavy nelineárních rovnic, které nabývají hodnot z* $\mathbb{F}_{256}$. *Není tedy potřeba znát hodnoty* $P \in \mathcal{P}$ *a* $K \in \mathcal{K}$

- *Vztahy mezi maticemi* P, A^i *pro* $i \in \{0, \ldots, n\}$ *a maticemi* K^j *pro* $j \in \{0, \ldots, N_r\}$, *jsou dány zobrazeními* $\alpha_0[\cdot], \ldots, \alpha_{N_r}[\cdot]$. *Z toho plyne, že* $\forall i \in \{0, \ldots, N_r\}$ *a* $\forall j \in \{0, 1, 2, 3\}$ *a* $\forall k \in \{0, \ldots, N_b - 1\}$ $\exists f_{i,j,k}$ *vztah daný zobrazením* $\alpha_i[\cdot]$ *určující proměnnou* ${a^i}_{j,k}$. *Platí tedy* ${a^i}_{j,k} = f_{i,j,k}$. *Proto část rovnic této soustavy nelineárních rovnic bude mít tvar* ${a^i}_{j,k} + f_{i,j,k} = 0$ *pro* $i \in \{0, \ldots, N_r\}$ *a* $j \in \{0, 1, 2, 3\}$ *a* $k \in \{0, \ldots, N_b - 1\}$.

  Nechť $\mathcal{W} \in {\mathbb{F}_{256}}^{4\times(N_b(N_r+1))}$ *je rozšířená matice klíče pro* N_r *rundovních zobrazení (kapitola 1.3.9). Matici* $\mathcal{W}$ *nevztahujeme ke konkrétnímu klíči* K, *protože nás nezajímají konkrétní hodnoty, ale pouze vztahy mezi prvky této matice. Pro tento odstavec proměnné rundovních klíčů značme* $w_{i,j}$, *kde* i, j *jsou příslušné indexy prvků matice* $\mathcal{W}$. *Dále víme, že* $\forall i \in \{0, 1, 2, 3\}$ *a* $\forall j \in \{N_k, \ldots, N_b(N_r+1)-1\}$ *(prvky matice indexujeme od nuly)* $\exists g_{i,j}$ *vztah daný rekurzivním generováním matice* $\mathcal{W}$ *určující proměnou* $w_{i,j}$. *Platí tedy, že* $w_{i,j} = g_{i,j}$. *Proto druhá část rovnic této soustavy nelineárních rovnic bude mít tvar:* $w_{i,j} + g_{i,j} = 0$ *pro* $i \in \{0, 1, 2, 3\}$ *a* $j \in \{N_k, \ldots, N_b(N_r + 1) - 1\}$. *POZN: Dle definice matice* $\mathcal{W}$ *v kapitole 1.3.9 existuje jednoznačná korespondence mezi proměnnými matice* $\mathcal{W}$ *a proměnnými matic* $K^0, \ldots, K^{N_r}$. *Pro tento odstavec jsme provedli přeznačení proměnných rundovních klíčů.*

- *V soustavě se bude vyskytovat pouze jedna bijekce a to permutace* S *(kapitola 1.3.3).*

- *Tato soustava je nad tělesem* $\mathbb{F}_{256}$

Takto vytvořenou nelineární soustavu rovnic budeme nazývat soustava nelineárních rovnic odvozená od šifrového diagramu $\mathcal{D}$. *Budeme ji značit* $\rho_{\mathcal{D}}$ *a platí* $\rho_{\mathcal{D}} = \rho_{\mathcal{D}}(2 \cdot (N_r + 1) \cdot 4 \cdot N_b - 4 \cdot N_k, (2 \cdot N_r + 3) \cdot 4 \cdot N_b, 1, \mathbb{F}_{256})$

Poznámka 3.31.

- *V definici 3.1 jsme požadovali, aby funkce určující nelineární rovnice měly vlastnosti 1 a 2. Nyní si ukážeme, že tyto vlastnosti splňují funkce určující rovnice z definice 3.30. Vlastnost 1 je splněna, protože zobrazení v šifře Rijndael (kapitola 1.3) jsou lineární, afinní nebo obsahují permutaci* S *(kapitola 1.3.3). Funkce určující rovnice z definice 3.30 můžeme tedy zapsat jako složení permutace* S *a funkcí reprezentujících operace a prvky tělesa* $\mathbb{F}_{256}$. *To, že platí vlastnost 2, plyne z věty 3.32, která tvrdí, že všechny funkce, které určují rovnice v soustavě nelineárních rovnic z definice 3.30, jsou bijekce v každé proměnné (definice 3.17). To implikuje vlastnost 2 z definice 3.1 dle poznámky 3.18. Celkem nelineární soustava rovnic z definice 3.30 je definovaná správně.*

- *Budeme používat stejné značení jako v definici 3.30. Všimněme si, že vztahy* $g_{i,j}$ *pro* $i \in \{0, 1, 2, 3\}$ *a* $j \in \{N_k, \ldots, N_b(N_r + 1) - 1\}$ *nejsou určeny jednoznačně. Ukážeme si dva významné tvary těchto vztahů. První*

tvar vychází přesně z algoritmu (kapitola 1.3.9), kterým se generuje rozšířená matice klíče $\mathcal{W}$. Konkrétně pokud generujeme i-tý sloupce pro $i \in \{N_k, \ldots, N_b(N_r+1)-1\}$, tak tento sloupec je závislý na $(i-1)$-ním sloupci a $(i-N_k)$-tém sloupci. Druhý tvar je tvořený tak, aby všechny proměnné libovolného rundovního klíče byly závislé pouze na proměnných předchozích rundovních klíčů. Tento tvar vznikne rozepsáním rekurzivní závislosti dané algoritmem generujícím rundovní klíče (kapitola 1.3.9). Ilustrujme si oba tvary na konkrétním příkladě. Nechť máme $\mathcal{R}(5,10,5)^*$, potom vztah $g_{1,14}$ může vypadat takto:

- $g_{1,14} := w_{1,4} + S(w_{1,13})$
- $g_{1,14} := w_{1,4} + S(w_{1,3} + w_{1,2} + w_{1,1} + w_{1,0} + S(w_{2,9}))$

Potom daná rovnice může mít jeden z následujících tvarů:

- $w_{1,14} + w_{1,4} + S(w_{1,13}) = 0$
- $w_{1,14} + w_{1,4} + S(w_{1,3} + w_{1,2} + w_{1,1} + w_{1,0} + S(w_{2,9})) = 0$

Ve zbytku této práce, pokud nebudeme zmiňovat, jaký tvar rovnic generující rundovní klíče používáme, tak vždy používáme první tvar z výše uvedených. Následující větu 3.32 vyslovíme v takovém znění, aby platila pro oba zmíněné tvary rovnic.

Věta 3.32. *Nechť máme $\mathcal{R}(N_b, N_k, N_r)$ nebo $\mathcal{R}(N_b, N_k, N_r)^*$, $\mathcal{D}$ šifrový diagram 1.21 pro tuto konfiguraci a nelineární soustava rovnic $\rho_\mathcal{D}$ odvozenou od šifrového diagramu $\mathcal{D}$ (definice 3.30). Rovnice v $\rho_\mathcal{D}$ generující rundovní klíče mají jeden ze dvou tvarů popsaných v poznámce 3.31. Potom všechny funkce, které určují rovnice v této soustavě nelineárních rovnic, jsou bijekce v každé proměnné (definice 3.17).*

Důkaz. Definujme hodnoty:

- $m := 2 \cdot (N_r + 1) \cdot 4 \cdot N_b - 4 \cdot N_k$, což je počet rovnic v $\rho_\mathcal{D}$
- $k := (2 \cdot N_r + 3) \cdot 4 \cdot N_b$, což je počet proměnných v $\rho_\mathcal{D}$

Zvolme libovolné $i \in \{1, \ldots, m\}$, následně funkce f_i určuje i-tou rovnici v $\rho_\mathcal{D}$ a $X_i = \{x_1, \ldots, x_{k_i}\}$, kde $k_i \in \mathbb{N}$, je množina proměnných obsažených v f_i. Protože v šifře Rijndael (kapitola 1.3) jsou všechna zobrazení lineární nebo afinní nebo se jedná o aplikaci permutace S (kapitola 1.3.3) na všechny prvky matice, tak funkce f_i má následující tvar:

$$f_i(X_i) = \sum_{i \in R} g_j + C_0 + S(\sum_{i \in T} g_j + C_1)$$

kde:

- g_j je výraz závislý na proměnné x_j pro $j \in \{1, \ldots, k_i\}$, který má jeden z následujících tvarů:
  - $g_j(x_j) = c \cdot x_j$ kde $c \in \mathbb{F}_{256}$ a $c \neq 0$
  - $g_j(x_j) = c_2 \cdot S(c_1 \cdot x_j)$ kde $c_1, c_2 \in \mathbb{F}_{256}$ a $c_1, c_2 \neq 0$

- $R, T \subseteq \{1, \ldots, k_i\}$
  - $j \in \{1, \ldots, k_i\}$ je prvkem R, pokud g_j není v součtu, na který se aplikuje permutace S
  - $j \in \{1, \ldots, k_i\}$ je prvkem T, pokud g_j je v součtu, na který se aplikuje permutace S
- $C_0 \in \mathbb{F}_{256}$ je součet všech konstant takových, že na jejich součet se neaplikuje permutace S
- $C_1 \in \mathbb{F}_{256}$ je součet všech konstant takových, že na jejich součet se aplikuje permutace S

Všimněme si, že $\forall j \in \{1, \ldots, k_i\}$ je g_j bijekce $\mathbb{F}_{256}$.

Nyní zvolme libovolné $l \in \{1, \ldots, k_i\}$ a libovolné hodnoty $a_1, \ldots, a_{l-1}, a_{l+1}, \ldots, a_{k_i} \in \mathbb{F}_{256}$. Definujeme funkci $H := f_i(a_1, \ldots, a_{l-1}, \cdot, a_{l+1}, \ldots, a_{k_i})$. Chceme dokázat, že H je bijekce $\mathbb{F}_{256}$. Dále se postup rozpadne na dvě části:

1. pokud $l \in R$ definuji součet

$$s := \sum_{i \in R \setminus \{l\}} g_j(a_j) + C_0 + S(\sum_{i \in T} g_j(a_j) + C_1)$$

   Potom funkce $H(x) = g_l(x) + s$ je bijekce, protože g_l je bijekce.

2. pokud $l \in T$ definuji součty:

$$s_1 := \sum_{i \in R} g_j(a_j) + C_0$$

$$s_2 := \sum_{i \in T \setminus \{l\}} g_j(a_j) + C_1$$

   Potom funkce $H(x) = s_1 + S(g_l(x) + s_2)$ je bijekce, protože H můžeme zapsat $H = h_2 \circ S \circ h_1 \circ g_l$, kde g_l, h_1, S, h_2 jsou bijekce:
   - $h_1(x) = x + s_2$
   - $h_2(x) = x + s_1$

H je bijekce, proto $f_i(a_1, \ldots, a_{l-1}, \cdot, a_{l+1}, \ldots, a_{k_i})$ je bijekce, a protože nezáleželo na volbě l, i a volbě hodnot $a_1, \ldots, a_{l-1}, a_{l+1}, \ldots, a_{k_i}$, tedy $\forall i \in \{1, \ldots, m\}$ platí, že funkce f_i jsou bijekce v každé proměnné (definice 3.17). □

Poznámka 3.33. *Všimněme si, že kdybychom soustavu $\rho_{\mathcal{D}}$ z definice 3.30 mohli triangulovat, tak počet volných proměnných dle věty 3.22 pro tuto soustavu nelineárních rovnic je roven* $(2 \cdot N_r + 3) \cdot 4 \cdot N_b - (2 \cdot (N_r + 1) \cdot 4 \cdot N_b - 4 \cdot N_k) = 8 \cdot N_b \cdot N_r + 12 \cdot N_b - 8 \cdot N_b \cdot N_r - 8 \cdot N_b + 4 \cdot N_k = 4 \cdot N_b + 4 \cdot N_k$, *což je počet proměnných v klíči a otevřeném textu (značení dle definice 3.30).*

Podrobně si ukážeme, jak vypadají všechny typy rovnic soustavy $\rho_{\mathcal{D}}$ z definice 3.30 pro konfiguraci $\mathcal{R}(5, 10, 5)^*$. Máme šifrový diagram 1.21 pro tuto konfiguraci:

$$\mathcal{D} = ((A^0, \ldots, A^6), (K^0, \ldots, K^5), (\alpha_1[K], \ldots, \alpha_6[K]))_{P,K}$$

Rovnice:

- ${a^1}_{i,j} + {a^0}_{i,j} + {k^0}_{i,j} = 0$ pro $i \in \{0,1,2,3\}$ a $i \in \{0,1,2,3,4\}$
- ${a^i}_{0,0} + {k^{i-1}}_{0,0} + 0x02 \cdot S({a^{i-1}}_{0,0}) + 0x03 \cdot S({a^{i-1}}_{1,1}) + 0x01 \cdot S({a^{i-1}}_{2,2}) + 0x01 \cdot S({a^{i-1}}_{3,3}) = 0$ pro $i \in \{2,3,4,5,6\}$
- ${a^i}_{0,1} + {k^{i-1}}_{0,1} + 0x02 \cdot S({a^{i-1}}_{0,1}) + 0x03 \cdot S({a^{i-1}}_{1,2}) + 0x01 \cdot S({a^{i-1}}_{2,3}) + 0x01 \cdot S({a^{i-1}}_{3,4}) = 0$ pro $i \in \{2,3,4,5,6\}$
- ${a^i}_{0,2} + {k^{i-1}}_{0,2} + 0x02 \cdot S({a^{i-1}}_{0,2}) + 0x03 \cdot S({a^{i-1}}_{1,3}) + 0x01 \cdot S({a^{i-1}}_{2,4}) + 0x01 \cdot S({a^{i-1}}_{3,0}) = 0$ pro $i \in \{2,3,4,5,6\}$
- ${a^i}_{0,3} + {k^{i-1}}_{0,3} + 0x02 \cdot S({a^{i-1}}_{0,3}) + 0x03 \cdot S({a^{i-1}}_{1,4}) + 0x01 \cdot S({a^{i-1}}_{2,0}) + 0x01 \cdot S({a^{i-1}}_{3,1}) = 0$ pro $i \in \{2,3,4,5,6\}$
- ${a^i}_{0,4} + {k^{i-1}}_{0,4} + 0x02 \cdot S({a^{i-1}}_{0,4}) + 0x03 \cdot S({a^{i-1}}_{1,0}) + 0x01 \cdot S({a^{i-1}}_{2,1}) + 0x01 \cdot S({a^{i-1}}_{3,2}) = 0$ pro $i \in \{2,3,4,5,6\}$
- ${a^i}_{1,0} + {k^{i-1}}_{1,0} + 0x01 \cdot S({a^{i-1}}_{0,0}) + 0x02 \cdot S({a^{i-1}}_{1,1}) + 0x03 \cdot S({a^{i-1}}_{2,2}) + 0x01 \cdot S({a^{i-1}}_{3,3}) = 0$ pro $i \in \{2,3,4,5,6\}$
- ${a^i}_{1,1} + {k^{i-1}}_{1,1} + 0x01 \cdot S({a^{i-1}}_{0,1}) + 0x02 \cdot S({a^{i-1}}_{1,2}) + 0x03 \cdot S({a^{i-1}}_{2,3}) + 0x01 \cdot S({a^{i-1}}_{3,4}) = 0$ pro $i \in \{2,3,4,5,6\}$
- ${a^i}_{1,2} + {k^{i-1}}_{1,2} + 0x01 \cdot S({a^{i-1}}_{0,2}) + 0x02 \cdot S({a^{i-1}}_{1,3}) + 0x03 \cdot S({a^{i-1}}_{2,4}) + 0x01 \cdot S({a^{i-1}}_{3,0}) = 0$ pro $i \in \{2,3,4,5,6\}$
- ${a^i}_{1,3} + {k^{i-1}}_{1,3} + 0x01 \cdot S({a^{i-1}}_{0,3}) + 0x02 \cdot S({a^{i-1}}_{1,4}) + 0x03 \cdot S({a^{i-1}}_{2,0}) + 0x01 \cdot S({a^{i-1}}_{3,1}) = 0$ pro $i \in \{2,3,4,5,6\}$
- ${a^i}_{1,4} + {k^{i-1}}_{1,4} + 0x01 \cdot S({a^{i-1}}_{0,4}) + 0x02 \cdot S({a^{i-1}}_{1,0}) + 0x03 \cdot S({a^{i-1}}_{2,1}) + 0x01 \cdot S({a^{i-1}}_{3,2}) = 0$ pro $i \in \{2,3,4,5,6\}$
- ${a^i}_{2,0} + {k^{i-1}}_{2,0} + 0x01 \cdot S({a^{i-1}}_{0,0}) + 0x01 \cdot S({a^{i-1}}_{1,1}) + 0x02 \cdot S({a^{i-1}}_{2,2}) + 0x03 \cdot S({a^{i-1}}_{3,3}) = 0$ pro $i \in \{2,3,4,5,6\}$
- ${a^i}_{2,1} + {k^{i-1}}_{2,1} + 0x01 \cdot S({a^{i-1}}_{0,1}) + 0x01 \cdot S({a^{i-1}}_{1,2}) + 0x02 \cdot S({a^{i-1}}_{2,3}) + 0x03 \cdot S({a^{i-1}}_{3,4}) = 0$ pro $i \in \{2,3,4,5,6\}$
- ${a^i}_{2,2} + {k^{i-1}}_{2,2} + 0x01 \cdot S({a^{i-1}}_{0,2}) + 0x01 \cdot S({a^{i-1}}_{1,3}) + 0x02 \cdot S({a^{i-1}}_{2,4}) + 0x03 \cdot S({a^{i-1}}_{3,0}) = 0$ pro $i \in \{2,3,4,5,6\}$
- ${a^i}_{2,3} + {k^{i-1}}_{2,3} + 0x01 \cdot S({a^{i-1}}_{0,3}) + 0x01 \cdot S({a^{i-1}}_{1,4}) + 0x02 \cdot S({a^{i-1}}_{2,0}) + 0x03 \cdot S({a^{i-1}}_{3,1}) = 0$ pro $i \in \{2,3,4,5,6\}$
- ${a^i}_{2,4} + {k^{i-1}}_{2,4} + 0x01 \cdot S({a^{i-1}}_{0,4}) + 0x01 \cdot S({a^{i-1}}_{1,0}) + 0x02 \cdot S({a^{i-1}}_{2,1}) + 0x03 \cdot S({a^{i-1}}_{3,2}) = 0$ pro $i \in \{2,3,4,5,6\}$
- ${a^i}_{3,0} + {k^{i-1}}_{3,0} + 0x03 \cdot S({a^{i-1}}_{0,0}) + 0x01 \cdot S({a^{i-1}}_{1,1}) + 0x01 \cdot S({a^{i-1}}_{2,2}) + 0x02 \cdot S({a^{i-1}}_{3,3}) = 0$ pro $i \in \{2,3,4,5,6\}$
- ${a^i}_{3,1} + {k^{i-1}}_{3,1} + 0x03 \cdot S({a^{i-1}}_{0,1}) + 0x01 \cdot S({a^{i-1}}_{1,2}) + 0x01 \cdot S({a^{i-1}}_{2,3}) + 0x02 \cdot S({a^{i-1}}_{3,4}) = 0$ pro $i \in \{2,3,4,5,6\}$
- ${a^i}_{3,2} + {k^{i-1}}_{3,2} + 0x03 \cdot S({a^{i-1}}_{0,2}) + 0x01 \cdot S({a^{i-1}}_{1,3}) + 0x01 \cdot S({a^{i-1}}_{2,4}) + 0x02 \cdot S({a^{i-1}}_{3,0}) = 0$ pro $i \in \{2,3,4,5,6\}$

- ${a^i}_{3,3} + {k^{i-1}}_{3,3} + 0x03 \cdot S({a^{i-1}}_{0,3}) + 0x01 \cdot S({a^{i-1}}_{1,4}) + 0x01 \cdot S({a^{i-1}}_{2,0}) + 0x02 \cdot S({a^{i-1}}_{3,1}) = 0$ pro $i \in \{2,3,4,5,6\}$
- ${a^i}_{3,4} + {k^{i-1}}_{3,4} + 0x03 \cdot S({a^{i-1}}_{0,4}) + 0x01 \cdot S({a^{i-1}}_{1,0}) + 0x01 \cdot S({a^{i-1}}_{2,1}) + 0x02 \cdot S({a^{i-1}}_{3,2}) = 0$ pro $i \in \{2,3,4,5,6\}$
- ${k^i}_{0,0} + {k^{i-2}}_{0,0} + {k^{i-1}}_{0,4} = 0$ pro $i \in \{3,5\}$
- ${k^i}_{1,0} + {k^{i-2}}_{1,0} + {k^{i-1}}_{1,4} = 0$ pro $i \in \{3,5\}$
- ${k^i}_{2,0} + {k^{i-2}}_{2,0} + {k^{i-1}}_{2,4} = 0$ pro $i \in \{3,5\}$
- ${k^i}_{3,0} + {k^{i-2}}_{3,0} + {k^{i-1}}_{3,4} = 0$ pro $i \in \{3,5\}$
- ${k^i}_{0,1} + {k^{i-2}}_{0,1} + {k^i}_{0,0} = 0$ pro $i \in \{3,5\}$
- ${k^i}_{1,1} + {k^{i-2}}_{1,1} + {k^i}_{1,0} = 0$ pro $i \in \{3,5\}$
- ${k^i}_{2,1} + {k^{i-2}}_{2,1} + {k^i}_{2,0} = 0$ pro $i \in \{3,5\}$
- ${k^i}_{3,1} + {k^{i-2}}_{3,1} + {k^i}_{3,0} = 0$ pro $i \in \{3,5\}$
- ${k^i}_{0,2} + {k^{i-2}}_{0,2} + {k^i}_{0,1} = 0$ pro $i \in \{3,5\}$
- ${k^i}_{1,2} + {k^{i-2}}_{1,2} + {k^i}_{1,1} = 0$ pro $i \in \{3,5\}$
- ${k^i}_{2,2} + {k^{i-2}}_{2,2} + {k^i}_{2,1} = 0$ pro $i \in \{3,5\}$
- ${k^i}_{3,2} + {k^{i-2}}_{3,2} + {k^i}_{3,1} = 0$ pro $i \in \{3,5\}$
- ${k^i}_{0,3} + {k^{i-2}}_{0,3} + {k^i}_{0,2} = 0$ pro $i \in \{3,5\}$
- ${k^i}_{1,3} + {k^{i-2}}_{1,3} + {k^i}_{1,2} = 0$ pro $i \in \{3,5\}$
- ${k^i}_{2,3} + {k^{i-2}}_{2,3} + {k^i}_{2,2} = 0$ pro $i \in \{3,5\}$
- ${k^i}_{3,3} + {k^{i-2}}_{3,3} + {k^i}_{3,2} = 0$ pro $i \in \{3,5\}$
- ${k^i}_{0,4} + {k^{i-2}}_{0,4} + {k^i}_{0,3} = 0$ pro $i \in \{3,5\}$
- ${k^i}_{1,4} + {k^{i-2}}_{1,4} + {k^i}_{1,3} = 0$ pro $i \in \{3,5\}$
- ${k^i}_{2,4} + {k^{i-2}}_{2,4} + {k^i}_{2,3} = 0$ pro $i \in \{3,5\}$
- ${k^i}_{3,4} + {k^{i-2}}_{3,4} + {k^i}_{3,3} = 0$ pro $i \in \{3,5\}$
- ${k^i}_{0,0} + {k^{i-2}}_{0,0} + S({k^{i-1}}_{1,4}) + C = 0$ pro $i \in \{2,4\}$, pro $i = 2$ definujme $C := 0x01$ a pro $i = 4$ definujme $C := 0x02$
- ${k^i}_{1,0} + {k^{i-2}}_{1,0} + S({k^{i-1}}_{2,4}) = 0$ pro $i \in \{2,4\}$
- ${k^i}_{2,0} + {k^{i-2}}_{2,0} + S({k^{i-1}}_{3,4}) = 0$ pro $i \in \{2,4\}$
- ${k^i}_{3,0} + {k^{i-2}}_{3,0} + S({k^{i-1}}_{0,4}) = 0$ pro $i \in \{2,4\}$
- ${k^i}_{0,1} + {k^{i-2}}_{0,1} + {k^i}_{0,0} = 0$ pro $i \in \{2,4\}$

- $k^i{}_{1,1} + k^{i-2}{}_{1,1} + k^i{}_{1,0} = 0$ pro $i \in \{2, 4\}$
- $k^i{}_{2,1} + k^{i-2}{}_{2,1} + k^i{}_{2,0} = 0$ pro $i \in \{2, 4\}$
- $k^i{}_{3,1} + k^{i-2}{}_{3,1} + k^i{}_{3,0} = 0$ pro $i \in \{2, 4\}$
- $k^i{}_{0,2} + k^{i-2}{}_{0,2} + k^i{}_{0,1} = 0$ pro $i \in \{2, 4\}$
- $k^i{}_{1,2} + k^{i-2}{}_{1,2} + k^i{}_{1,1} = 0$ pro $i \in \{2, 4\}$
- $k^i{}_{2,2} + k^{i-2}{}_{2,2} + k^i{}_{2,1} = 0$ pro $i \in \{2, 4\}$
- $k^i{}_{3,2} + k^{i-2}{}_{3,2} + k^i{}_{3,1} = 0$ pro $i \in \{2, 4\}$
- $k^i{}_{0,3} + k^{i-2}{}_{0,3} + k^i{}_{0,2} = 0$ pro $i \in \{2, 4\}$
- $k^i{}_{1,3} + k^{i-2}{}_{1,3} + k^i{}_{1,2} = 0$ pro $i \in \{2, 4\}$
- $k^i{}_{2,3} + k^{i-2}{}_{2,3} + k^i{}_{2,2} = 0$ pro $i \in \{2, 4\}$
- $k^i{}_{3,3} + k^{i-2}{}_{3,3} + k^i{}_{3,2} = 0$ pro $i \in \{2, 4\}$
- $k^i{}_{0,4} + k^{i-2}{}_{0,4} + S(k^2{}_{0,3}) = 0$ pro $i \in \{2, 4\}$
- $k^i{}_{1,4} + k^{i-2}{}_{1,4} + S(k^i{}_{1,3}) = 0$ pro $i \in \{2, 4\}$
- $k^i{}_{2,4} + k^{i-2}{}_{2,4} + S(k^i{}_{2,3}) = 0$ pro $i \in \{2, 4\}$
- $k^i{}_{3,4} + k^{i-2}{}_{3,4} + S(k^i{}_{3,3}) = 0$ pro $i \in \{2, 4\}$

Věta 3.34. *Nechť máme $\mathcal{R}(N_b, N_k, N_r)$ nebo $\mathcal{R}(N_b, N_k, N_r)^*$ a*

$$\mathcal{D} = ((P, A_0, \ldots, A_{N_r}), (K_0, \ldots, K_{N_r}), (\alpha_0[K], \ldots, \alpha_{N_r}[K]))_{P,K}$$

šifrový diagram 1.21 pro tuto konfiguraci. Potom soustavu nelineárních rovnic odvozenou od $\mathcal{D}$ $\rho_{\mathcal{D}}$ lze triangulovat a množina volných proměnných obsahuje pouze proměnné klíče a proměnné otevřeného textu.

Důkaz. Proměnné, které jsou obsaženy pouze v jedné rovnici, jsou proměnné v šifrovém textu A_{N_r}. Když tyto proměnné a příslušné rovnice zpracujeme, potom proměnné v matici A_{N_r-1} jsou nezpracované a jsou obsažené pouze v jedné nezpracované rovnici, a proto je zpracujeme s příslušnými rovnicemi. Opakováním tohoto postupu zpracujeme všechny rovnice a proměnné, které se nevyskytují v generování rundovních klíčů, krom proměnných otevřeného textu. Nyní proměnné rundovních klíčů jsou nezpracované a jsou obsažené pouze v rovnicích generující rundovní klíče. Proměnné posledního rundovního klíče K_{N_r} jsou nezpracované a jsou obsažené pouze v rovnicích, které generují tento klíč, tedy tyto proměnné a příslušné rovnice můžeme zpracovat. Následně proměnné předposledního rundovního klíče K_{N_r-1} jsou nezpracované a jsou obsažené pouze v rovnicích, které generují tento klíč, tedy opět je můžeme zpracovat s příslušnými rovnicemi. Tento postup můžeme opakovat a postupně zpracovávat proměnné jednotlivých rundovních klíčů. Jediné proměnné, které zůstanou nezpracované jsou proměnné klíče, které jsou obsaženy v několika prvních rundovních klíčích. Celkem jsme zpracovali všechny rovnice a proto trian. alg. (definice 3.6) proběhl úspěšně a nezpracované

proměnné jsou obsažené pouze v otevřeném textu a klíči. Z toho plyne, že proměnné klíče a proměnné otevřeného textu jsou volné proměnné této nelineární soustavy rovnic. □

Poznámka 3.35. *Věta 3.34 říká, že průběh šifrování je závislý na volbě otevřeného textu a klíče, což je vcelku logické.*

3.5 Ověření platnosti diferenční stopy pomocí triangulačního algoritmu

V kapitole 2.1 jsme si vysvětlili, že ověření diferenční stopy s neznámou platností (definice 2.1) spočívá v nalezení $P \in \mathcal{P}$ a $K \in \mathcal{K}$, které vytvoří šifrový diagram, v němž vstupy do aktivních S-boxů budou nabývat požadovaných hodnot. Tyto hodnoty plynou z podmínek daných vstupními a výstupními diferencemi u daných aktivních S-boxů. Také jsme si určili podmínku, že otevřený text je libovolný a pevný (kapitola 2).

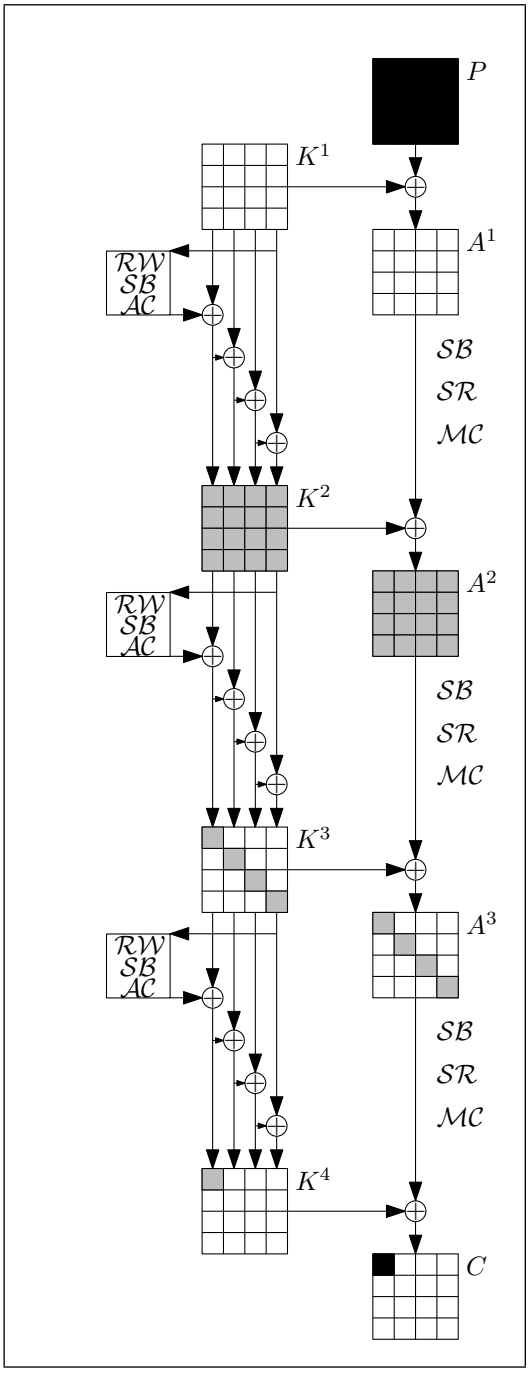
Jedna z možností, jak ověřit platnost diferenční stopy pro konfiguraci $\mathcal{R}(N_b, N_k, N_r)$ nebo $\mathcal{R}(N_b, N_k, N_r)^*$ je vytvořit nelineární soustavu rovnic odvozenou od šifrového diagramu 1.21 pro příslušnou konfiguraci šifry Rijndael. Tvar rovnic generující rundovní klíče (poznámka 3.31) volíme tak, aby výslednou nelineární soustavu rovnic bylo možné triangulovat. Následně vstupy do aktivních S-boxů a otevřený text nepovažujeme za proměnné, ale za konstanty rovné příslušným hodnotám. Je dobré si uvědomit, že proměnné, které neovlivní vstup do aktivních S-boxů jsou pro nás nepodstatné a nemusíme je s příslušnými rovnicemi, kde jsou obsažené, zahrnovat do soustavy nelineárních rovnic. Většina těchto proměnných je graficky znázorněna za vstupem do posledního aktivního S-boxu v příslušném šifrovém diagramu. Takto zmenšíme počet rovnic a proměnných v nelineární soustavě rovnic. Vstupy do aktivních S-boxů mohou nabývat více než jednu hodnotu, zde můžeme zvolit libovolnou hodnotu z možných vstupů do aktivního S-boxu dle věty 3.37. Takto vytvořenou soustavu nelineárních rovnic triangulujeme. Bohužel ne vždy můžeme takto vytvořenou soustavu triangulovat (viz příklad 3.36). V takovýchto případech musíme nalézt specifický postup pro danou situaci. Pokud se podaří daná soustava triangulovat, tak existuje triangulace, pro kterou všechny volné proměnné se nachází v klíči, protože hodnoty klíče jednoznačně určují průběh šifrování pro pevný otevřený text. Aby všechny volné proměnné byly v klíči, tak proměnnou klíče v triangulačním algoritmu zpracováváme (bod 2 trian. alg.) pouze pokud neexistuje žádná jiná proměnná, kterou bychom v danou chvíli mohli zpracovat. Následně můžeme zvolit hodnoty volných proměnných libovolně a takto získat řešení dané soustavy dle věty 3.37. Tímto způsobem můžeme ověřit platnost diferenční stopy. Pro praktické účely stačí dopočítat hodnoty proměnných obsažených v klíči.

Příklad 3.36. *Nechť máme $\mathcal{R}(4,4,3)^*$ a šifrový diagram 1.21 pro tuto konfiguraci, který je znázorněn na obrázku 3.1. Po celý tento příklad budeme používat značení z obrázku 3.1. Horní index používáme k indexaci matic, spodní index budeme používat k indexaci prvků matice. Vytvořme nelineární soustavu rovnic $\rho_{\mathcal{D}}$ odvozenou od tohoto šifrového diagramu. Fixujme proměnou $c_{0,0}$ a všechny*

proměnné otevřeného textu jako konstanty, v obrázku je značíme černě. Nyní postupujme dle triangulačního algoritmu (definice 3.6). Proměnné v matici C jsou obsaženy pouze v jedné rovnici, tedy tyto proměnné a příslušné rovnice zpracujme. Následně si všimněme, že proměnné ${a^3}_{0,0}$, ${a^3}_{1,1}$, ${a^3}_{2,2}$ a ${a^3}_{3,3}$ jsou v tuto chvíli obsaženy v rovnici s konstantou $c_{0,0}$ a v rovnicích, které určují vztah vůči matici A^2. Všechny tyto rovnice jsou v tuto chvíli nezpracované. Následně všechny proměnné matice A^3 krom výše zmíněných jsou nezpracované a jsou obsaženy každá právě v jedné nezpracované rovnici určující vztah vůči proměnným matice A^2. Bohužel všechny proměnné matice A^2 jsou obsažené v nezpracovaných rovnicích určujících vztahy s proměnnými ${a^3}_{0,0}$, ${a^3}_{1,1}$, ${a^3}_{2,2}$, ${a^3}_{3,3}$ a zároveň v nezpracovaných rovnicích určující vztahy vůči matici A^1. S tímto postupem souběžně zpracováváme rovnice generující rundovní klíče, které můžeme zpracovat. V tuto chvíli všechny nezpracované proměnné jsou vždy obsažené alespoň ve dvou nezpracovaných rovnicích. Tedy triangulace je neúspěšná. Pro lepší pochopení tohoto příkladu je dobré si podrobněji rozepsat průběh triangulačního algoritmu a podrobněji prostudovat matici závislosti této soustavy nelineárních rovnic (definice 3.9).

Věta 3.37. *Nechť máme $\mathcal{R}(N_b, N_k, N_r)$ nebo $\mathcal{R}(N_b, N_k, N_r)^*$ a $\mathcal{D}$ šifrový diagram 1.21 pro zadanou konfiguraci. A dále nechť $\rho_{\mathcal{D}}$ je nelineární soustava rovnic odvozená od šifrového diagramu $\mathcal{D}$ (definice 3.30). Rovnice v $\rho_{\mathcal{D}}$ generující rundovní klíče jsou jednoho z dvou tvarů popsaných v poznámce 3.31. V $\rho_{\mathcal{D}}$ fixujeme libovolný počet proměnných jako konstanty a jejich hodnoty zvolme libovolně. Takto vytvořenou nelineární soustavu rovnic označme $\tilde{\rho_{\mathcal{D}}}$. Pokud $\tilde{\rho_{\mathcal{D}}}$ lze triangulovat, potom libovolné dosazení za volné proměnné vede k řešení $\tilde{\rho_{\mathcal{D}}}$.*

Důkaz. Víme, že dle věty 3.32 každá funkce, která určuje rovnici v $\rho_{\mathcal{D}}$, je bijekce v každé proměnné (definice 3.17). Všimněme si, že pokud v každé takovéto funkci fixujeme libovolný počet proměnných jako konstanty s libovolnou hodnotou, tak nově vzniklé funkce jsou stále bijekce v každé proměnné, tedy každá funkce, která určuje rovnici v $\tilde{\rho_{\mathcal{D}}}$, je bijekce v každé proměnné. Z předpokladu, že můžeme $\tilde{\rho_{\mathcal{D}}}$ triangulovat, pak dle věty 3.19 můžeme za volné proměnné dosadit libovolné hodnoty a takto získat řešení $\tilde{\rho_{\mathcal{D}}}$. □



Obrázek 3.1: Šifrový diagram pro $\mathcal{R}(4,4,3)^*$ s náznakem problému triangulace

4. Ověření diferenční stopy pro $\mathcal{R}(5,10,5)^*$

V kapitole 2.3 jsme vytvořili diferenční stopu s neznámou platnosti pro konfiguraci $\mathcal{R}(5,10,5)^*$ s pěti aktivními S-boxy. V kapitole 3.5 jsme si ukázali postup, kterým je možné ověřit platnost diferenční stopy. Nyní si ukážeme na výše zmíněné diferenční stopě, jak přesně tento postup funguje. Protože nelineární soustavu rovnic vytvořenou dle postupu v kapitole 3.5 nemůžeme triangulovat, vysvětlíme si vlastní specifický postup jak umožnit triangulaci a získat kýžený výsledek.

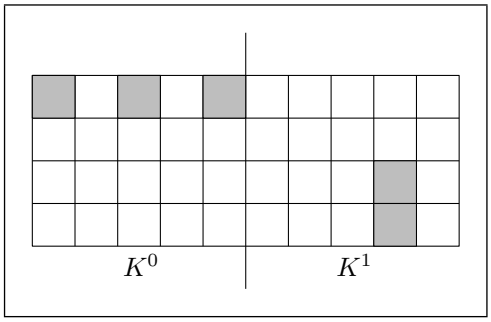
Po zbytek této kapitoly budeme používat značení z obrázku 4.2, používáme horní indexy k indexování matic, protože spodními indexy budeme indexovat prvky matic. Nyní si přesně vysvětlíme, jak sestavit nelineární soustavu rovnic. Jak jsme popsali v kapitole 3.5, $\rho_{\mathcal{D}}$ je soustava nelineárních rovnic odvozená od šifrového diagramu 1.21 pro konfiguraci $\mathcal{R}(5,10,5)^*$, ve které rovnice generující rundovní klíče mají první tvar popsaný v poznámce 3.31. Takový tvar těchto rovnic jsme zvolili proto, že se s ním lépe pracuje a v našem případě nemá žádný vliv na možnost triangulovat výslednou soustavu nelineárních rovnic. Dále v $\rho_{\mathcal{D}}$ fixujeme jako konstanty vstupy do aktivních S-boxů a proměnné otevřeného textu. Náš specifický postup spočívá v tom, že proměnou $p_{0,4}$ nebudeme fixovat jako konstantu, i když je to proměnná otevřeného textu. Tím pádem budeme moci danou nelineární soustavu rovnic triangulovat. Poznatek, že nelineární soustava rovnic, ve které fixujeme všechny proměnné otevřeného textu a vstupy do aktivních S-boxů jako konstanty, nelze triangulovat, jsme získali díky implementaci trian. alg. pro takovouto soustavu. Myšlenku nefixovat proměnné $p_{0,4}$ jako konstantu jsme získali díky této implementaci trian. alg. Celou situaci si ilustrujeme na obrázku 4.2, kde šedivá políčka značí konstanty, což je otevřený text krom proměnné $p_{0,4}$ a vstupy do aktivních S-boxů, a bílá políčka značí proměnné.

Dále odebereme z $\rho_{\mathcal{D}}$ ty proměnné, které neovlivní vstupy do aktivních S-boxů, a všechny rovnice, ve kterých se tyto proměnné nachází. Tj. odebereme proměnné, které se v obrázku 4.2 vyskytují za vstupem do posledního aktivního S-boxu, to jest celé matice C a K^5 a všechny proměnné v matici A^5. Pozor, ${a^5}_{0,0}$ je konstanta, tedy zůstává v $\rho_{\mathcal{D}}$. Nakonec odebereme všechny proměnné matice K^4, krom proměnné ${k^4}_{0,0}$, která je obsažena v rovnici s konstantou ${a^5}_{0,0}$. Dále odebereme všechny rovnice, ve kterých se výše zmíněné proměnné vyskytují. Bylo by možné odebrat i jiné proměnné a s nimi příslušné rovnice z $\rho_{\mathcal{D}}$. Ověření, že na těchto proměnných nezávisí žádný vstup do aktivního S-boxu je složitější než je nechat v $\rho_{\mathcal{D}}$ a tyto proměnné zpracovat triangulačním algoritmem. Jinak řečeno odebrali jsme proměnné, které na první pohled neovlivní vstupy do aktivních S-boxů.

Takto jsme vytvořili soustavu nelineárních rovnic $\rho_{\mathcal{D}}$ o 158 proměnných, 122 rovnicích a 24 konstantách nad tělesem $\mathbb{F}_{256}$ s jedním bijektivním zobrazením S (kapitola 1.3.3). Pomocí implementace trian. alg. (kapitola 6) jsme ověřili, že $\rho_{\mathcal{D}}$ lze triangulovat. Dle věty 3.22 má tato soustava rovnic $36 = 158 - 122$ volných proměnných. Na obrázku 4.3 je matice závislosti pro soustavu $\rho_{\mathcal{D}}$ a na obrázku 4.4 je matice závislosti po triangulaci pro soustavu $\rho_{\mathcal{D}}$, černá políčka značí jedničku a bílá nulu. Nyní si uvědomme, že pokud zvolíme konkrétní hodnoty vstupů do

aktivních S-boxů, které jsou určené vstupní a výstupní diferencí těchto aktivních S-boxů, a dále libovolně zvolíme hodnoty zbývajících konstant a volných proměnných, tak dle věty 3.37 získáme řešení soustavy $\rho_{\mathcal{D}}$. Takto získáme otevřený text a klíč splňující danou diferenční stopu. Tedy diferenční stopa je platná.

Protože jsme použili specifický postup, nefixovali jsme $p_{0,4}$ jako konstantu, proto nejsou všechny volné proměnné obsažené v klíči. Proměnná ${a^1}_{0,4}$ je volná proměnná. Zbývající volné proměnné jsou obsaženy v klíči. Na obrázku 4.1 je znázorněna matice klíče K, což je konkatenace matic K^1 a K^2, tedy $K = (K^1|K^2)$, kde bílá políčka značí volné proměnné a šedivá políčka zbývající proměnné. V triangulačním algoritmu jsme použili myšlenku, že zpracováváme proměnnou klíče, jen pokud neexistuje jiná proměnná, kterou bychom mohli v danou chvíli zpracovat.



Obrázek 4.1: Volné proměnné v matici klíče

Nyní si uvedeme postup jak nalézt dva klíče, které zašifrují libovolný pevný otevřený text na stejný šifrový, pro konfiguraci $\mathcal{R}(5, 10, 5)^*$. V diferenční stopě pro konfiguraci $\mathcal{R}(5, 10, 5)^*$ se vyskytují hodnoty $x, y \in \mathbb{F}_{256}$ (kapitola 2.3) určující vstupní a výstupní diferenci aktivních S-boxů. Zvolme tyto hodnoty $x := 0x03$ a $y := 0x18$, potom $M_S(x, y) = \{0xDF, 0xDC, 0x03, 0x00\}$. Víme, že hodnoty vstupů do aktivních S-boxů musí nabývat jednu z hodnot z množiny $M_S(x, y)$. Dle věty 3.37 můžeme zvolit hodnoty vstupů do aktivních S-boxů libovolně, a proto zvolme hodnotu $0x03$. Opět dle věty 3.37 hodnoty volných proměnných můžeme volit libovolně, a proto za hodnoty volných proměnných krom proměnných ${a^1}_{0,4}$ a ${k^1}_{0,0}$ dosaďme hodnotu $0x00$.

Zvolme libovolný otevřený text $R \in \mathcal{P} = {\mathbb{F}_{256}}^{4\times 5}$. Hodnoty prvků matice R dosadíme jako hodnoty patřičných konstant reprezentující otevřený text v soustavě $\rho_{\mathcal{D}}$. Všimněme si, že proměnná $p_{0,4}$ není fixovaná jako konstanta a také není volná proměnná, a proto nelze dosadit hodnotu patřičného prvku matice R za tuto proměnnou. Vysvětleme si druhou část našeho specifického postupu. V tuto chvíli známe všechny hodnoty konstant a volných proměnných krom proměnných ${a^1}_{0,4}$ a ${k^1}_{0,0}$. Dle věty 3.37 můžeme dosadit za hodnoty volných proměnných libovolné hodnoty. Nyní zkusme dosadit za hodnoty volných proměnných ${a^1}_{0,4}$ a ${k^1}_{0,0}$ všechny možné hodnoty, kterých je 2^{16}, a dopočítat hodnotu proměnné $p_{0,4}$. Pokud tímto postupem najdeme dosazení za proměnné ${a^1}_{0,4}$ a ${k^1}_{0,0}$, které dává hodnotu proměnné $p_{0,4}$ rovnající se patřičnému prvku z matice R, tak dopočítáme ostatní hodnoty proměnných klíče. Takto získáme matici klíče $K_1 \in \mathcal{K} = {\mathbb{F}_{256}}^{4\times 10}$. Za předpokladu, že se nám podařilo najít vhodné hodnoty proměnných ${a^1}_{0,4}$ a ${k^1}_{0,0}$, tak otevřený text R a klíč K_1 splňují diferenční stopu z kapitoly 2.3 pro

konfiguraci $\mathcal{R}(5, 10, 5)^*$.

Dosaďme výše zmíněné hodnoty x a y do matice $\triangle K$ diference klíčů, kterou známe z diferenční stopy (kapitola 2.3), konkrétně se jedná o konkatenaci matic K_3 a K_4 z obrázku 2.5 ($\triangle K = (K_3|K_4)$):

$$\triangle K = \left(\begin{array}{ccccc|ccccc} x & 0 & x & 0 & 0 & 0x02 \cdot y & 0 & 0x02 \cdot y & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0x01 \cdot y & 0 & 0x01 \cdot y & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0x01 \cdot y & 0 & 0x01 \cdot y & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0x03 \cdot y & 0 & 0x03 \cdot y & 0 & 0 \end{array}\right)$$

$$\triangle K = \left(\begin{array}{ccccc|ccccc} 0x03 & 0 & 0x03 & 0 & 0 & 0x30 & 0 & 0x30 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0x18 & 0 & 0x18 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0x18 & 0 & 0x18 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0x28 & 0 & 0x28 & 0 & 0 \end{array}\right)$$

Definujme klíč $K_2 := K_1 + \triangle K$. Za předpokladu, že otevřený text R a klíč K_1 splňují danou diferenční stopu, tak jsme tímto postupem získali dva klíče K_1 a K_2, které zašifrují libovolně zvolený otevřený text R na stejný šifrový text, tedy $e_{K_1}(R) = e_{K_2}(R)$.

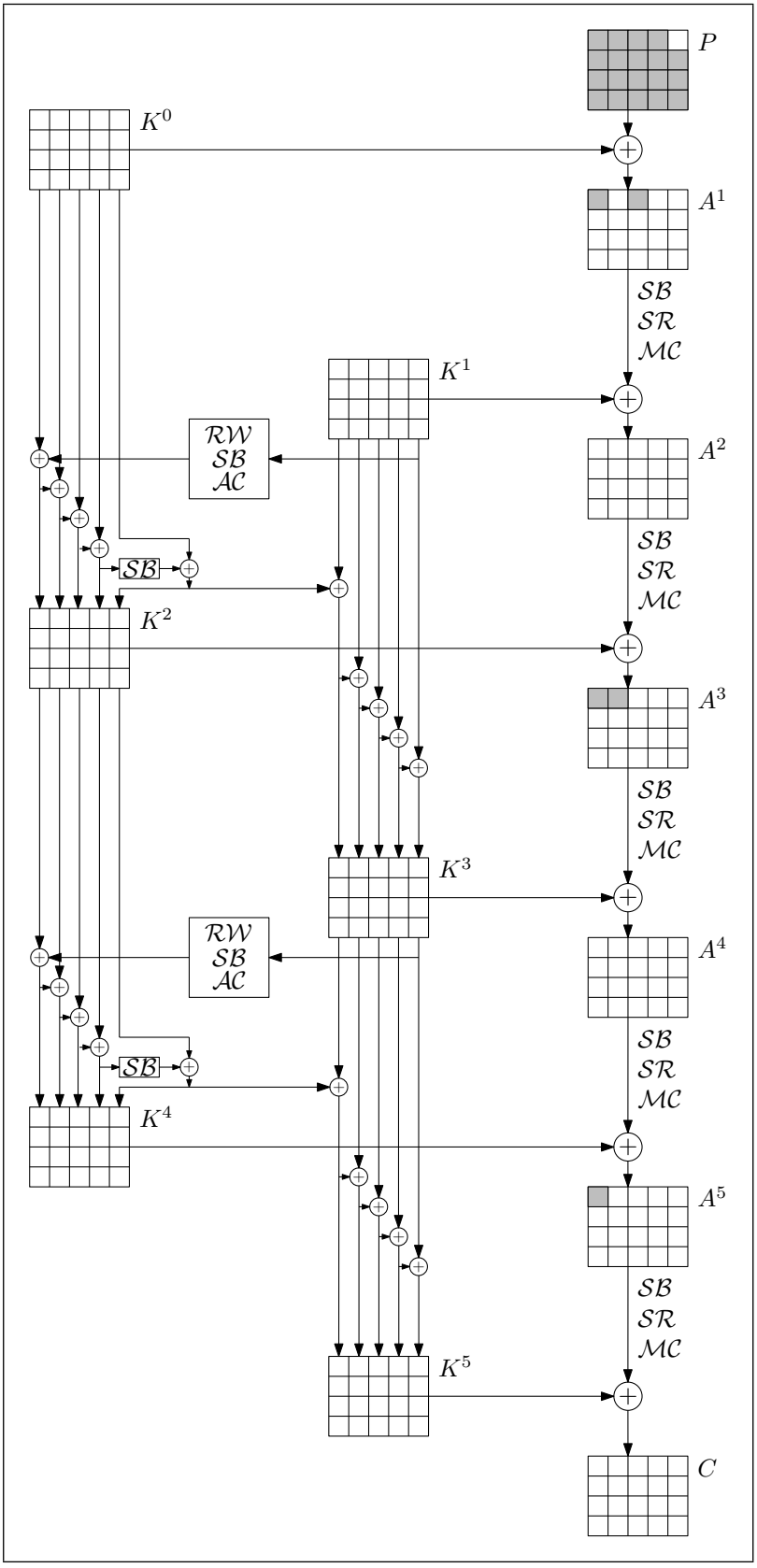
Všimněme si, že úspěch výše popsaného postupu závisí na existenci vhodných hodnot proměnných ${a^1}_{0,4}$ a ${k^1}_{0,0}$. Formulujme si hypotézu:

Hypotéza 4.1. *Nechť máme nelineární soustavu rovnic $\rho_{\mathcal{D}}$, která je popsaná na začátku této kapitoly. Hodnoty konstant vstupů do aktivních S-boxů zvolme* $0x03$. *Hodnoty volných proměnných krom proměnných ${a^1}_{0,4}$ a ${k^1}_{0,0}$ zvolme* $0x00$. *Zvolme libovolný otevřený text $R \in \mathcal{P}$ a dosaďme hodnoty prvků matice R jako hodnoty patřičných konstant reprezentující otevřený text v soustavě $\rho_{\mathcal{D}}$. Potom vždy existuje dosazení hodnot za proměnné ${a^1}_{0,4}$ a ${k^1}_{0,0}$, které dává hodnotu proměnné $p_{0,4}$ rovnou příslušnému prvku matice R.*

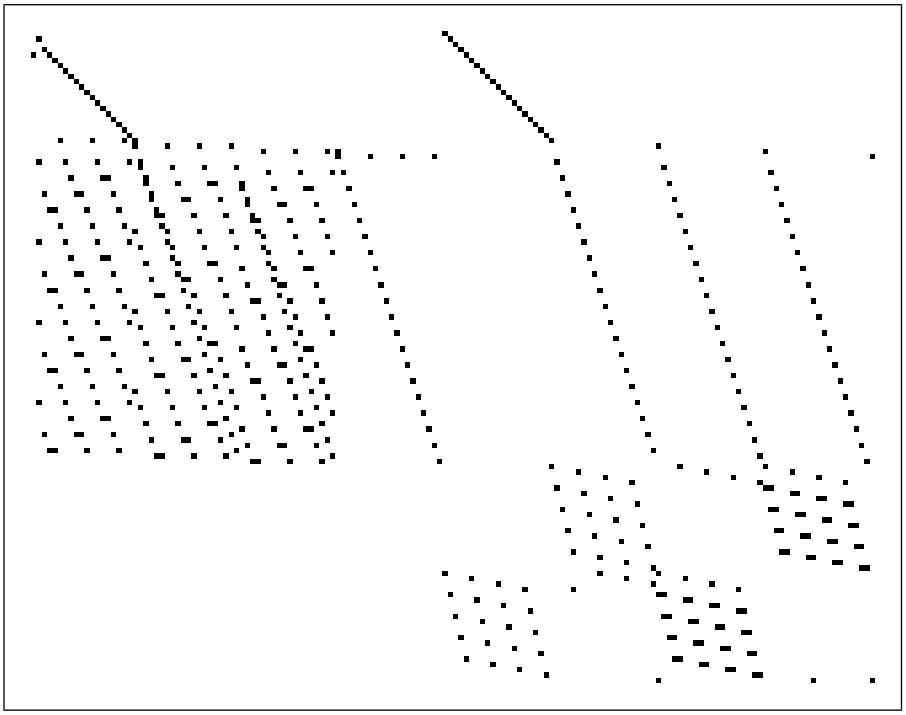
Zvolme libovolný otevřený text $R \in \mathcal{P}$ a proveďme dosazení hodnot konstant a volných proměnných do soustavy $\rho_{\mathcal{D}}$ jako v hypotéze 4.1. Definujme funkci předpisem: $F_R({a^1}_{0,4}, {k^1}_{0,0}) := p_{0,4}$, která dvojici hodnot proměnných ${a^1}_{0,4}$ a ${k^1}_{0,0}$ přiřadí hodnotu proměnné $p_{0,4}$. Tuto hodnotu získáme dopočítáním řešení soustavy $\rho_{\mathcal{D}}$ odpovídající výše zmíněné volbě dosazení hodnot za konstanty a volné proměnné a za dosazení hodnot za volné proměnné ${a^1}_{0,4}$ a ${k^1}_{0,0}$ odpovídající vstupům do funkce F_R. Funkce F_R závisí na parametru R, což je otevřený text. Potom hypotéza 4.1 říká, že funkce $F_R({a^1}_{0,4}, {k^1}_{0,0})$ je na $\mathbb{F}_{256}$ pro libovolnou volbu parametru R.

Při implementaci trian. alg. pro soustavu $\rho_{\mathcal{D}}$ jsme zjistili, že pokud jsme provedli dosazení hodnot za konstanty a volné proměnné dle hypotézy 4.1 a za jednu z volných proměnných ${a^1}_{0,4}$ nebo ${k^1}_{0,0}$ jsme dosadili konkrétní hodnotu a následně dosazovali všechny možné hodnoty do druhé proměnné, tak jsme velmi rychle našli otevřený text $R \in \mathcal{P}$, pro který výše zmíněný postup nevedl k řešení soustavy $\rho_{\mathcal{D}}$, ve které by proměnná $p_{0,4}$ měla hodnotu odpovídající příslušnému prvku matice R. Pokud jsme provedli výše popsané dosazení a dosazovali jsme všechny možné hodnoty do proměnných ${a^1}_{0,4}$ a ${k^1}_{0,0}$, tak se nám nepodařilo najít otevřený text vyvracející hypotézu 4.1. Konkrétně jsme provedli test pro $27 \cdot 10^6$ voleb otevřeného textu a vždy jsme našli dosazení za proměnné ${a^1}_{0,4}$ a ${k^1}_{0,0}$ dávající řešení, ve kterém hodnota proměnné $p_{0,4}$ byla rovna příslušnému prvku matice R.

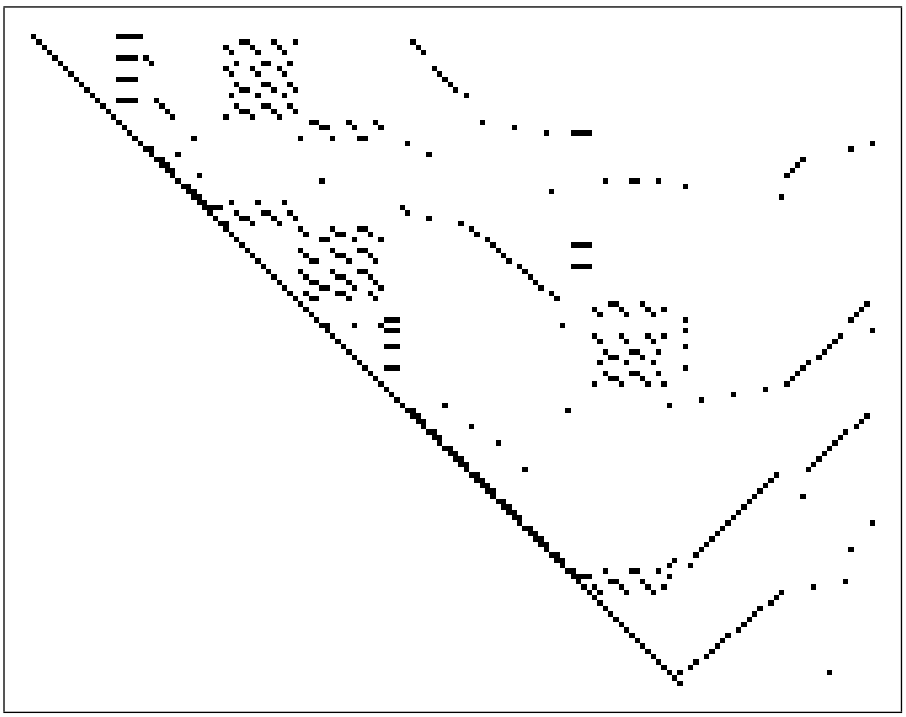
Celkem za předpokladu platnosti hypotézy 4.1 máme konstruktivní postup jak nalézt pro libovolně zvolený otevřený text dva klíče, které tento otevřený text zašifrují na stejný šifrový text pro konfiguraci $\mathcal{R}(5, 10, 5)^*$.



Obrázek 4.2: Šifrový diagram 1.21 pro konfiguraci $\mathcal{R}(5, 10, 5)^*$



Obrázek 4.3: Matice závislosti pro soustavu $\rho_{\mathcal{D}}$



Obrázek 4.4: Matice závislosti po triangulaci pro soustavu $\rho_{\mathcal{D}}$

5. Vytvoření hashovací funkce a nalezení kolize

V této kapitole si zadefinuje hashovací funkci v Davies-Mayerově módu (definice 1.7) pro šifru Rijndeal s konfigurací $\mathcal{R}(5,10,5)^*$ a ukážeme, jak nalézt kolizi (definice 1.9) pro námi vytvořenou hashovací funkci pomocí již získaných poznatků.

Definice 5.1. *Nechť máme množinu $M := \{320 \cdot n;\ n \in \mathbb{N}\}$. Zvolme libovolnou zprávu $m \in \mathcal{B}_{\mathbb{N}}$ (značení dle definice 1.5). Definujme padding (definice 1.8) následujícím způsobem. Pokud $m \in \mathcal{B}_M$, tak m prodloužíme o 320 nul na zprávu $\bar{m}$, pokud $m \notin \mathcal{B}_M$, tak m prodloužíme o nejmenší možný počet nul na zprávu $\bar{m}$ tak, aby $\bar{m} \in \mathcal{B}_M$. Tedy $\bar{m} = (m_1, \ldots, m_k) \in \mathcal{B}_M$, kde $k \in \mathbb{N}$, $\forall i \in \{1, \ldots, k\}$ je $m_i \in \mathcal{B}_{320}$ a $320 \cdot k$ je délka zprávy $\bar{m}$ v bitech. Hodnotě k budeme říkat délka zprávy $\bar{m}$ v blocích.*

Nechť máme $\mathcal{R}(5,10,5)^*$ a definujeme matici

$$IV := \begin{pmatrix} 0xAD & 0x1B & 0xE1 & 0x57 & 0x7D \\ 0x78 & 0xE5 & 0xB4 & 0x28 & 0xEF \\ 0xCB & 0xD2 & 0xA4 & 0x5C & 0x99 \\ 0x4C & 0xA3 & 0x0F & 0x27 & 0xFF \end{pmatrix} \in \mathcal{P} = {\mathbb{F}_{256}}^{4\times 5}$$

POZN: Matice IV byla volena zcela náhodně.
Definujeme rekurzivně následující hodnoty:

- $M_0 := IV \in {\mathbb{F}_{256}}^{4\times 5}$
- $M_i := e_{m_i}(M_{i-1}) + M_{i-1} \in {\mathbb{F}_{256}}^{4\times 5}$ *pro* $i \in \{1, \ldots, k\}$

Definujeme hashovací funkci v Davies-Mayerově módu (definice 1.7) pro konfiguraci $\mathcal{R}(5,10,5)^*$ $\mathcal{RH} : \mathcal{B}_{\mathbb{N}} \to \mathcal{B}_{160}$, předpisem: $\mathcal{RH}(\bar{m}) := M_k$.

Poznámka 5.2. *Všimněme si, že v definici 1.7 hashovací funkce v Davies-Mayerově módu pracujeme s posloupnostmi bitů a v definici 5.1 hashovací funkce $\mathcal{RH}$ pracujeme s maticemi nad tělesem $\mathbb{F}_{256}$ (kapitola 1.3.1). Dle kapitol 1.3.1 a 1.3.2 můžeme na posloupnosti bitů patřičné délky nahlížet jako na matice nad tělesem $\mathbb{F}_{256}$ a operace sčítání těchto posloupnosti bitů je isomorfní se sčítáním těchto matic nad $\mathbb{F}_{256}$. Tuto korespondenci mezi maticemi nad $\mathbb{F}_{256}$ a posloupnostmi bitů využíváme v definici 5.1 a budeme jí brát za samozřejmou v další úvaze.*

Poznámka 5.3. *Hashovací funkci z definice 5.1 jsme zadefinovali z toho důvodů, abychom na této hashovací funkci mohli ilustrovat použití triangulačního algoritmu v kryptoanalýze. Pro reálné použití by bylo vhodné zvětšit počet rundovních zobrazení v konfiguraci šifry Rijndael a zvolit jiný padding.*

Ukážeme, jak ze znalostí, které již máme, jsme schopni nalézt kolizi pro hashovací funkci $\mathcal{RH}$ (definice 5.1). V kapitole 4 jsme si ukázali postup jak pro $\mathcal{R}(5,10,5)^*$ nalézt dva klíče, které libovolný pevný otevřený text zašifrují na stejný šifrový text. Zvolme za otevřený text hodnotu IV z definice 5.1 a dle postupu z kapitoly 4 najděme dva klíče M_1 a M_2, které takto zvolený otevřený

text zašifrují na stejný šifrový text. V tomto konkrétním případě, i když nemáme ověřenou platnost hypotézy 4.1, lze najít takovéto dva klíče M_1 a M_2. Jejich konkrétní hodnoty jsme našli pomocí implementace trian. alg. (kapitola 6). Nyní na M_1 a M_2 nahlížejme jako na zprávy v délce 320 bitů a spočítejme hodnotu hash pro tyto zprávy: $\mathcal{RH}(M_1) = e_{\mathbf{0}}\Big(e_{M_1}(IV) + IV\Big) + e_{M_1}(IV) + IV = e_{\mathbf{0}}\Big(e_{M_2}(IV) + IV\Big) + e_{M_2}(IV) + IV = \mathcal{RH}(M_2) =: h$, kde $\mathbf{0} \in {\mathbb{F}_{256}}^{4\times 10}$ je nulová matice. Takto jsme našli kolizi pro hashovací funkce $\mathcal{RH}$. Konkrétní hodnoty zpráv M_1, M_2 a jejich hash hodnota h jsou zaznamenány v následujících maticích:

$$h = \begin{pmatrix} 0xC3 & 0xBB & 0xFF & 0x1C & 0xE0 \\ 0x91 & 0x40 & 0x45 & 0xB9 & 0x49 \\ 0x55 & 0x78 & 0x8A & 0x27 & 0x42 \\ 0x71 & 0x2F & 0x02 & 0xD7 & 0x07 \end{pmatrix}$$

$$M_1 = \begin{pmatrix} 0xAE & 0x00 & 0xE2 & 0x00 & 0xEA & 0x00 & 0x00 & 0x00 & 0x00 & 0x00 \\ 0x00 & 0x00 & 0x00 & 0x00 & 0x00 & 0x00 & 0x00 & 0x00 & 0x00 & 0x00 \\ 0x00 & 0x00 & 0x00 & 0x00 & 0x00 & 0x00 & 0x00 & 0x00 & 0xEA & 0x00 \\ 0x00 & 0x00 & 0x00 & 0x00 & 0x00 & 0x00 & 0x00 & 0x00 & 0xE1 & 0x00 \end{pmatrix}$$

$$M_2 = \begin{pmatrix} 0xAD & 0x00 & 0xE1 & 0x00 & 0xEA & 0x30 & 0x00 & 0x30 & 0x00 & 0x00 \\ 0x00 & 0x00 & 0x00 & 0x00 & 0x00 & 0x18 & 0x00 & 0x18 & 0x00 & 0x00 \\ 0x00 & 0x00 & 0x00 & 0x00 & 0x00 & 0x18 & 0x00 & 0x18 & 0xEA & 0x00 \\ 0x00 & 0x00 & 0x00 & 0x00 & 0x00 & 0x28 & 0x00 & 0x28 & 0xE1 & 0x00 \end{pmatrix}$$

Nyní si ukážeme, jak najít delší zprávy, které dávají kolizi pro hashovací funkci $\mathcal{RH}$ (definice 5.1) za předpokladu, že platí hypotéza 4.1. Konkrétně si ukážeme jak vytvořit dvě zprávy v délce n bloků pro libovolné $n \in \mathbb{N}$. Tedy hledáme dvě zprávy $M^1, M^2 \in \mathcal{B}_{320\cdot n}$ (značení dle definice 1.5):

- $M^1 = ({m^1}_1, \ldots, {m^1}_n)$
- $M^2 = ({m^2}_1, \ldots, {m^2}_n)$

kde ${m^1}_1, \ldots, {m^1}_n, {m^2}_1, \ldots, {m^2}_n \in \mathcal{B}_{320}$ a platí $\mathcal{RH}(M^1) = \mathcal{RH}(M^2)$. Horní indexy používáme k indexování zpráv a spodní indexy používáme k indexování jednotlivých bloků zpráv. Ukážeme si induktivní postup, kterým vytvoříme zprávy M^1, M^2. Opět využijeme poznatků z kapitoly 4. Za předpokladu platnosti hypotézy 4.1 pro konfiguraci $\mathcal{R}(5, 10, 5)^*$ a libovolný pevný otevřený text $P \in \mathcal{P} = {\mathbb{F}_{256}}^{4\times 5}$ umíme nalézt dva klíče K_1 a K_2, pro které platí $e_{K_1}(P) = e_{K_2}(P)$. Definujme $A_0 := IV$, kde IV je konstanta z definice 5.1. Postupujme indukcí, zvolme libovolné $i \in \{0, \ldots, n-1\}$ a předpokládejme, že známe hodnotu $A_i \in {\mathbb{F}_{256}}^{4\times 5}$. Potom dle postupu z kapitoly 4 najdeme dva klíče $K_1, K_2 \in \mathcal{K} = {\mathbb{F}_{256}}^{4\times 10}$, pro které platí, že $e_{K_1}(A_i) = e_{K_2}(A_i)$. Definujme hodnotu $A_{i+1} := e_{K_1}(A_i) + A_i$ a bloky zpráv ${m^1}_{i+1} := K_1$ a ${m^2}_{i+1} := K_2$. Tímto induktivním postupem vytvoříme zprávy M^1 a M^2, pro která platí $\mathcal{RH}(M^1) = \mathcal{RH}(M^2) = e_{\mathbf{0}}(A_n) + A_n$, kde $\mathbf{0} \in {\mathbb{F}_{256}}^{4\times 10}$ je nulová matice. To, že M^1 a M^2 jsou kolizní zprávy pro hashovací funkci $\mathcal{RH}$ plyne z rekurzivní definice $\mathcal{RH}$ a z vlastnosti $e_{{m^1}_{i+1}}(A_i) = e_{{m^2}_{i+1}}(A_i)$ $\forall i \in \{0, \ldots, n-1\}$. Za předpokladu platnosti hypotézy 4.1 jsme našli postup

jak vytvořit dvě kolizní zprávy délky n bloku pro hashovací funkci $\mathcal{RH}$ a pro libovolné $n \in \mathbb{N}$.

Pokud se nám podaří najít výše popsaným postupem dvě kolizní zprávy, tak pokud tyto zprávy zapíšeme jako posloupnost bytů, budou obsahovat velké množství nulových bytů. Tento fakt plyne z toho, že diference klíčů příslušející použité diferenční stopě (kapitola 2.3) obsahuje velké množství nulových prvků a také z toho, že v kapitole 4 dosazujeme za většinu hodnot volných proměnných nuly. Pokud bychom výše zmíněným postupem chtěli generovat kolizní zprávy bez velkého množství nul, tak můžeme upravit postup popsaný v kapitole 4. Tedy budeme za hodnoty volných proměnných, za které dosazujeme nulu, dosazovat jinou hodnotu než nulu a adekvátně k tomu upravíme zbytek tohoto postupu.

Všimněme si, že pokud dle výše popsaného postupu nalezneme dvě kolizní zprávy v délce $n \in \mathbb{N}$ bloků pro hashovací funkci $\mathcal{RH}$, tak potom umíme také nalézt dvě kolizní zprávy v délce $k \in \mathbb{N}$ pro $k \leq n$. Toto plyne z faktu, že postup pro hledání kolizních zpráv je induktivní. Pomocí implementace (kapitola 6) trian. alg. a výše popsaného postupu pro hledání kolizních zpráv pro hashovací funkci $\mathcal{RH}$ jsme našli kolizní zprávy v délce $27 \cdot 10^6$ bloků, což je přibližně 1GB. Dle předchozí úvahy tedy můžeme díky této implementaci najít kolizní zprávy v délce k bloků pro hashovací funkci $\mathcal{RH}$ a pro $k \in \{1, \ldots, 27 \cdot 10^6\}$. Při dostatečně velkém strojovém času se můžeme tedy pokusit pomocí zmíněné implementace najít kolizní zprávy v libovolné délce bloků. Pokud by platila hypotéza 4.1, tak bychom takto vždy našli dvě kolizní zprávy v libovolné délce bloků.

6. Popis implementační části

V této kapitole si popíše programy, které jsou součástí této práce na přiloženém DVD. Celkem se jedná o tři programy.

```
permutace_S_hodnoty_vyhovujici_diferenci
hash_RH
kolize_pro_RH
```

Tyto programy jsou napsány v programovacím jazyce C++ za použití vývojového prostředí Microsoft Visual Studio 2010 Professional. Na DVD se nachází celé projekty se zdrojovými kódy, testovací data a Microsoft Visual C++ Runtime Library, která je potřebná k běhu programů.

První program nalezne $\forall a \in \mathbb{F}_{256}$ a $\forall b \in \mathbb{F}_{256}$ množinu $M_S(a,b) = \{x \in G; b = S(x+a) - S(x)\}$ (definice 1.10) pro permutaci S (kapitola 1.3.3). Program má za výstup šest souborů. Tři mají formát csv a ve zbývajících třech je LaTeXový kód k nagenerování stejných tabulek jako je v souborech csv. V těchto souborech je v prvním sloupci zaznamenaná vstupní diference a, v druhém sloupci výstupní diference b a ve třetím prvky množiny $M_S(a,b)$. V prvním typu souboru jsou zaznamenány všechny kombinace vstupních a výstupních diferencí, tyto soubory mají název:

```
output_full.csv
output_tex_full.txt
```

V druhém typu souboru jsou zaznamenány kombinace vstupní diference a a výstupní diference b, pro které platí, že $a \cdot b \neq 0$ a zároveň $|M_S(a,b)| \neq 0$. Tyto soubory mají název:

```
output_medium.csv
output_tex_medium.txt
```

Ve třetím typu souboru jsou zaznamenány kombinace vstupní diference a a výstupní diference b, pro které platí, že $|M_S(a,b)| = 4$. Tyto soubory mají název:

```
output_small.csv
output_tex_small.txt
```

Ve výstupních souborech používáme hexadecimální zápis dle kapitoly 1.3.1, s výjimkou, že používáme malá písmena ke značení cifer. Ukázka z výstupu tohoto programu je v tabulce 6.1.

Druhý program načte textový soubor, kde na každém řádku je adresa souboru, a k těmto souborům spočítá hodnotu hash pro hashovací funkci $\mathcal{RH}$ (definice 5.1).

Třetí program implementuje triangulační algoritmus (definice 3.6) pro soustavu $\rho_{\mathcal{D}}$ z kapitoly 4 a postup pro hledání kolizních zpráv popsaný v kapitole 5. Tento program slouží k hledání kolize pro hashovací funkci $\mathcal{RH}$ (definice 5.1) v zadané délce bloků a nebo k vykreslení matice závislosti a matice závislosti po triangulaci pro soustavu $\rho_{\mathcal{D}}$ z kapitoly 4 ve formátu tga.

Vs. dif.	Výs. dif.	Vs. hodnoty vyhovujících vs. a výs. dif.
0x01	0x1f	0xbd, 0xbc, 0x01, 0x00
0x02	0x14	0x63, 0x61, 0x02, 0x00
0x03	0x18	0xdf, 0xdc, 0x03, 0x00
0x04	0x91	0xc6, 0xc2, 0x04, 0x00
0x05	0x08	0x7f, 0x7a, 0x05, 0x00
0x06	0x0c	0xa5, 0xa3, 0x06, 0x00
0x07	0xa6	0x1e, 0x19, 0x07, 0x00
0x08	0x53	0x9f, 0x97, 0x08, 0x00
0x09	0x62	0x2b, 0x22, 0x09, 0x00
0x0a	0x04	0xfe, 0xf4, 0x0a, 0x00
0x0b	0x48	0x48, 0x43, 0x0b, 0x00
0x0c	0x9d	0x5d, 0x51, 0x0c, 0x00
0x0d	0xb4	0xed, 0xe0, 0x0d, 0x00
0x0e	0xc8	0x3c, 0x32, 0x0e, 0x00
0x0f	0x15	0x8e, 0x81, 0x0f, 0x00
0x10	0xa9	0x35, 0x25, 0x10, 0x00
0x11	0xe1	0x98, 0x89, 0x11, 0x00
0x12	0xaa	0x56, 0x44, 0x12, 0x00
0x13	0x1e	0xf9, 0xea, 0x13, 0x00
0x14	0x99	0xf3, 0xe7, 0x14, 0x00
0x15	0x3a	0x5a, 0x4f, 0x15, 0x00
0x16	0x24	0x90, 0x86, 0x16, 0x00
0x17	0x93	0x3b, 0x2c, 0x17, 0x00
0x18	0xce	0xba, 0xa2, 0x18, 0x00
0x19	0xb7	0x1e, 0x07, 0x19, 0x00
0x1a	0xc1	0xdb, 0xc1, 0x1a, 0x00
0x1b	0xcc	0x7d, 0x66, 0x1b, 0x00
0x1c	0xff	0x78, 0x64, 0x1c, 0x00
0x1d	0xc7	0xd8, 0xc5, 0x1d, 0x00
0x1e	0x11	0x19, 0x07, 0x1e, 0x00
0x1f	0xa3	0xbb, 0xa4, 0x1f, 0x00
0x20	0xd4	0x6a, 0x4a, 0x20, 0x00
0x21	0x9e	0xf7, 0xd6, 0x21, 0x00
0x22	0xf0	0x2b, 0x09, 0x22, 0x00
0x23	0x45	0xb5, 0x96, 0x23, 0x00
0x24	0x55	0xac, 0x88, 0x24, 0x00
0x25	0x5c	0x35, 0x10, 0x25, 0x00
0x26	0x94	0xe9, 0xcf, 0x26, 0x00
0x27	0xaf	0x73, 0x54, 0x27, 0x00
0x28	0x57	0xfd, 0xd5, 0x28, 0x00
0x29	0xc6	0x68, 0x41, 0x29, 0x00
0x2a	0x86	0xb4, 0x9e, 0x2a, 0x00
0x2b	0x92	0x22, 0x09, 0x2b, 0x00

Tabulka 6.1: Ukázka hodnot vyhovující vstupní a výstupní diferenci pro S

Závěr

Cíl, který jsme si dali, se nám podařilo splnit. Ilustrovali jsme si využití triangulačního algoritmu v kryptoanalýze. Podařilo se nám nalézt kolizi pro námi zadefinovanou hashovací funkci $\mathcal{RH}$ (definice 5.1).

Ještě si shrňme možnosti, jak na tuto práci navázat. Ukázali jsme postup jak nalézt dvě zprávy, které dávají stejnou hodnotu hash pro hashovací funkci $\mathcal{RH}$. Zajímavé by bylo využít triangulační algoritmus pro hledání zprávy, která s libovolně zvolenou jinou zprávou tvoří kolizi pro hashovací funkci $\mathcal{RH}$. V kapitole 4 jsme si vysvětlili vlastní specifický postup jak upravit nelineární soustavu rovnic $\rho_{\mathcal{D}}$ zadefinovanou v této kapitole, abychom mohli danou soustavu triangulovat. Tento postup zkouší 2^{16} možností a hledá tu správnou. Zde se nabízí prostor pro nalezení lepšího specifického postupu, který by řešil daný problém.

V definici hashovací funkce $\mathcal{RH}$ jsme použili konfiguraci $\mathcal{R}(5, 10, 5)^*$, abychom mohli ilustrovat využití trian. alg. Pět rundovních zobrazení není zcela dostačujících, aby šifra Rijndael dosahovala patřičných kvalit. Na tuto práci lze navázat tím, že se pokusíme najít diferenční stopu s neznámou platností pro větší počet rundovních zobrazení, celý proces pro tuto diferenční stopu zopakovat a vytvořit algoritmus pro hledání dvou klíčů, které libovolný pevný otevřený text zašifrují na stejný šifrový text. Asi nejtěžší na tomto úkolu by bylo najít specifický postup, abychom mohli získanou nelineární soustavu rovnic triangulovat. Pokud bychom takto konstruovali hashovací funkci v Davies-Mayerově módu, bylo by lepší zvoli jiný druh paddingu.

Seznam použité literatury

[1] Khovratovich, Dmitry - Biryukov, Alex - Nikolic, Ivica. Speeding up Collision Search for Byte-Oriented Hash Functions. In *Topics in Cryptology – CT-RSA 2009 The Cryptographers' Track at the RSA Conference 2009, San Francisco, CA, USA, April 20-24, 2009. Proceedings.* Marc Fischlin. Berlin Heidelberg: Springer-Verlag, 2009. s. 164-181. (Lecture Notes in Computer Science; vol. 5473, ISSN: 0302-9743).

[2] Daemen, Joan - Rijmen, Vincent. *The Design of Rijndael.* 1. vydání. Berlin Heidelberg: Springer-Verlag, 2002. ISBN 3-540-42580-2.

[3] Stinson, Douglas R. *Cryptography: Theory and Practice.* 3. vydání. Chapman and Hall/CRC, 2005. ISBN 1-584-88508-4.

Seznam tabulek

Seznam obrázků

Seznam použitých zkratek a značení

- S ... permutace definovaná v kapitole 1.3.3
- $\mathcal{SB}$ ... zobrazení definované v kapitole 1.3.3
- $\mathcal{SR}$ ... zobrazení definované v kapitole 1.3.4
- $\mathcal{MC}$ ... zobrazení definované v kapitole 1.3.5
- $\mathcal{AK}[\cdot]$ ... zobrazení definované v kapitole 1.3.6
- $\mathcal{RC}(n)$ ... konstanta definovaná v kapitole 1.3.7
- $\mathcal{KS}$ ... zobrazení definované v kapitole 1.3.9
- $\mathcal{W}$ ... rozšířená matice klíče definovaná v kapitole 1.3.9
- $\mathcal{R}(N_b, N_k, N_r)$ ... značení šifry Rijndeal s příslušnými parametry definováno v úmluvě 1.16
- $\mathcal{R}(N_b, N_k, N_r)^*$ ... značení šifry Rijndeal s příslušnými parametry definováno v úmluvě 1.16
- N_b ... počet sloupců v matici otevřeného a šifrového textu šifry Rijndael definováno v kapitole 1.3.2
- N_k ... počet sloupců v matici klíče šifry Rijndael definováno v kapitole 1.3.2
- N_r ... počet rundovních zobrazení v šifře Rijndael definováno v kapitole 1.3.8
- $\mathcal{AC}(n), \mathcal{AC}$ ... zobrazení definované v úmluvě 1.19
- $\mathcal{RW}$ ... zobrazení definované v úmluvě 1.19
- $\mathcal{RH}$ ... hashovací funkce definovaná v definici 5.1