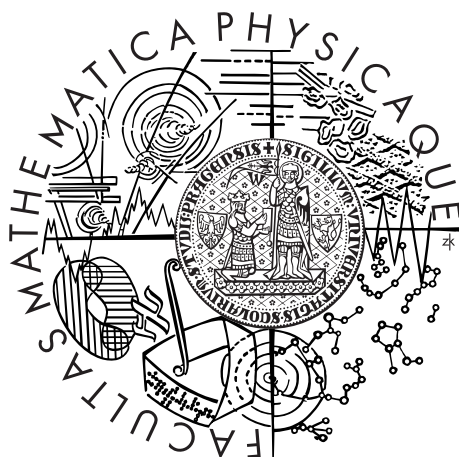


Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



David Mařák

Korelační útoky

Katedra algebry

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

Studijní program: Matematika

Studijní obor: Matematické metody informační bezpečnosti

Praha 2015

Děkuji svému vedoucímu RNDr. Michalu Hojsíkovi, Ph.D. za četné konzultace, pomoc a trpělivost, které mi věnoval při vytváření této práce. Dále děkuji mé rodině, přítelkyni a přátelům za jejich podporu a trpělivost.

Prohlašuji, že jsem tuto bakalářskou práci vypracoval(a) 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 dne

Podpis autora

Název práce: Korelační útoky

Autor: David Mařák

Katedra: Katedra algebry

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

Abstrakt: Tato bakalářská práce se zabývá popisem Korelačních útoků na proudové šifry typu kombinačního generátoru. Čtenář je nejdříve zběžně seznámen se základními definicemi z kryptografické teorie potřebné k porozumění textu. V práci je pak popsán původní článek, který tento druh útoku představuje, a který byl inspirací pro další vylepšení a modifikace. Podrobněji je pak popsána skupina vylepšených útoků nazývaná Rychlé korelační útoky, které jsou mnohem efektivnější, a již plně nahradily původní útok. Závěrem jsou pak popsány některé modifikace Rychlých korelačních útoků.

Klíčová slova: korelační útok, proudová šifra, kombinační generátor, kryptografie

Title: Correlation attacks

Author: David Mařák

Department: Department of Algebra

Supervisor: RNDr. Michal Hojsík, Ph.D., Department of Algebra

Abstract: This bachelor thesis describes Correlation attacks on combination generator type stream ciphers. The reader is briefly acquainted with the basic definitions of cryptographic theory which is necessary to understand the text. Afterwards, the thesis describes the original article that presented this type of attack, and which was the inspiration for further improvements and modifications. These improvements are then described in more details, namely the group called "Fast correlation attacks" which are more efficient and fully replaced the original attack. Finally, the thesis describes some modifications of Fast correlation attacks.

Keywords: correlation attack, stream cipher, combination generator, cryptography

Obsah

| | | |
|----------|--|-----------|
| 1 | Definice a terminologie | 3 |
| 2 | Korelační útok | 8 |
| 2.1 | Siegenthalerův útok | 8 |
| 2.2 | Vztah k dekódovacímu problému | 10 |
| 3 | Rychlé korelační útoky | 12 |
| 3.1 | Základní princip a motivace | 12 |
| 3.2 | Výpočetní a statistický model | 13 |
| 3.2.1 | Statistický model | 15 |
| 3.3 | Algoritmus A | 17 |
| 3.4 | Algoritmus B | 19 |
| 3.5 | Shrnutí | 22 |
| 4 | Modifikace rychlých korelačních útoků | 24 |
| 4.1 | Vylepšené hledání rovnic | 24 |
| 4.2 | Modifikace interpretací | 24 |
| | Literatura | 27 |
| | Seznam obrázků | 28 |
| | Seznam tabulek | 29 |

Úvod

Denně se po světě díky pokročilým komunikačním technologiím přenesou miliony terabytů dat a závislost lidské populace na těchto technologiích je čím dál tím větší. Proto se klade velký důraz na rychlost, spolehlivost a také bezpečnost přenosu těchto dat. V posledních letech bylo vynaloženo mnoho úsilí na lepší porozumění návrhu a zabezpečení proudových šifer. Mezi jejich velké výhody patří zejména malá hardwarová náročnost a rychlost šifrování.

V této práci se budeme zabývat jednou z nejvýznamnějších tříd útoků, která se používá k prolamování proudových šifer a také se zohledňuje při jejich návrhu. Jedná se o korelační útoky. Především se zaměříme na proudové šifry, které k vytváření proudu klíče využívají posuvné registry s lineární zpětnou vazbou (takzvané LFSR, z angl. Linear Feedback Shift Register), kombinované pomocí nelineární Booleovské funkce.

V kapitole jedna uvedeme základní definice a tvrzení, která budou použita v této práci. Ve druhé kapitole bude představen původní korelační útok a jeho vztah k dekódovacímu problému. Ve třetí kapitole pak popíšeme rychlé korelační útoky. V závěrečné kapitole pak budou prezentovány některé modifikace a vylepšení rychlých korelačních útoků.

1. Definice a terminologie

V této kapitole se budeme zabývat základními definicemi a tvrzeními z teoretické kryptografie a teorie kódů, které budou použity v této práci. Čerpáno bylo především z obsahu přednášek předmětů Teoretická kryptografie a Samoopravné kódy. Od čtenáře se očekávají základní znalosti algebry (Stanovský, 2010), pravděpodobnosti a statistiky (pravděpodobnost, střední hodnota, pravděpodobnostní rozdělení...) (Zvára a Štěpán J., 2002). Některé z nich jsou zde explicitně rozepsány.

Konečné těleso s q prvky budeme značit \mathbb{F}_q . Okruh polynomů nad konečným tělesem pak budeme značit $\mathbb{F}_q[x]$.

Definice 1. *Nechť A je konečná množina, pak definujeme $V(A) = \{\{a_i\}_{i \geq 0} | a_i \in A\}$.*

Definice 2 (q -ární rekurentní posloupnost). *Posloupnost $\{a_i\}_{i \geq 0} \in V(\mathbb{F}_q)$ se nazývá q -ární rekurentní posloupnost právě tehdy, když $\exists l \in \mathbb{N}$ a $\exists g : \mathbb{F}_q^l \rightarrow \mathbb{F}_q$ takové, že $\forall k \geq 0$ platí*

$$a_{k+l} = g(a_k, a_{k+1}, \dots, a_{k+l-1})$$

Definice 3 (Lineární rekurentní posloupnost). *Q -ární rekurentní posloupnost se nazývá lineární rekurentní posloupnost (LRP), jestliže funkce g , která tuto posloupnost definuje, je lineární zobrazení tj. jestliže existují $c_0, c_1, \dots, c_{l-1} \in \mathbb{F}_q$ takové, že $\forall k \geq 0$ platí*

$$a_{k+l} = \sum_{i=0}^{l-1} c_i a_{i+k}.$$

Posloupnost $(a_0, a_1, \dots, a_{l-1})$ nazveme inicializační vektor LRP a polynomu $g(x) = \sum_{i=0}^{l-1} c_i x^i$, kde $c_l = 1$, říkáme charakteristický polynom LRP.

Poznámka. Lineární rekurentní posloupnosti nad \mathbb{F}_2 říkáme *binární lineární rekurentní posloupnost*.

Poznámka. Pokud nebude řečeno jinak, budeme v této práci považovat všechny LRP za binární.

Tvrzení 1. *Mějme LRP $\{a_i\}_{i \geq 0}$ s charakteristickým polynomem $g(x) = \sum_{i=0}^{l-1} c_i x^i$. Pak platí, že $\forall k \geq l$ lze prvek a_k vyjádřit jako lineární kombinaci prvků inicializačního vektoru $(a_0, a_1, \dots, a_{l-1})$*

Důkaz. Víme, že libovolný prvek posloupnosti můžeme vyjádřit na n předchozích prvcích takto:

$$a_{l+k} = \sum_{i=0}^{l-1} c_i a_{i+k}.$$

Tedy prvek a_k máme v závislosti na prvcích $(a_{k-l}, a_{k-l+1}, \dots, a_{k-1})$, každý z těchto prvků můžeme opět vyjádřit na jeho k předchůdcích. Tento proces opakujeme dokud nevyjádříme potřebné prvky v závislosti na prvcích inicializačního vektoru. \square

Důsledek. Mějme LRP $\{a_i\}_{i \geq 0}$ s charakteristickým polynomem $g(x) = \sum_{i=0}^n c_i x^i$, a l libovolných prvků této posloupnosti $a_{i_1}, a_{i_2}, \dots, a_{i_l}$ společně s indexem pořadí. Zároveň předpokládejme, že pro každý z těchto prvků existuje lineární vztah

$$a_{i_j} = b_0 a_{q_1} + \dots + b_c a_{q_c},$$

kde $b_i \in \mathbb{F}_2$ a a_{q_i} jsou libovolné znaky posloupnosti $\{a_i\}_{i \geq 0}$. Pokud systém rovnic sestavený z těchto l rovnic má plnou hodnotu, pak můžeme z těchto l prvků jednoznačně rekonstruovat inicializační vektor $(a_0, a_1, \dots, a_{l-1})$.

Pro praktické využití v kryptografii se lineární rekurentní posloupnosti vytváří pomocí struktury zvané *Posuvný registr s lineární zpětnou vazbou* nebo také LFSR (z angl. Linear Feedback Shift Registr). Jedná se o mechanický popis LRP a můžeme ho definovat následovně.

Definice 4 (Posuvný registr s lineární zpětnou vazbou). *Mějme LRP $\{a_i\}_{i \geq 0}$ s charakteristickým polynomem $g(x) = \sum_{i=0}^l c_i x^i$. Pak LFSR popisující tuto LRP se skládá z l vnitřních registrů $(R_0, R_1, \dots, R_{l-1})$, časovacího signálu a je určen pomocí charakteristického polynomu $g(x) = \sum_{i=0}^l c_i x^i$. Za každý takt časovacího signálu se pak provede následující:*

Je spočítána hodnota $T = -\sum_{i=0}^{l-1} R_i c_i$,

obsah registru R_0 je předán na výstup,

obsah registru R_i se přesune do registru R_{i-1} pro $\forall i \in \{1, \dots, l-1\}$ a

obsah registru R_{l-1} je nastaven na hodnotu T .

Obsahy registrů na začátku procesu odpovídají inicializačnímu vektoru LRP a nazýváme je počátečním stavem LFSR.

Definice 5 (Proudová šifra). Proudovou šifru definujeme jako uspořádanou devítici $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{L}, \mathcal{S}, r, h, \mathcal{E}, \mathcal{D})$, kde

\mathcal{P} je konečná množina prvků otevřeného textu,

\mathcal{C} je konečná množina prvků šifrovaného textu,

\mathcal{K} je konečná množina klíčů,

\mathcal{L} je konečná množina prvků proudu klíče,

\mathcal{S} je konečná množina stavů,

$r : (\mathcal{S}, \mathcal{K}) \rightarrow \mathcal{S}$ je obnovovací funkce,

$h : (\mathcal{S}, \mathcal{K}) \rightarrow \mathcal{L}$ je výstupní funkce,

$\mathcal{E} = \{e_z : \mathcal{P} \rightarrow \mathcal{C} | z \in \mathcal{L}\}$ je množina všech šifrovacích transformací,

$\mathcal{D} = \{d_z : \mathcal{C} \rightarrow \mathcal{P} | z \in \mathcal{L}\}$ je množina všech dešifrovacích transformací,

a platí, že pro $\forall p \in \mathcal{P} \forall z \in \mathcal{L} d_z(e_z(p)) = p$.

Na obrázku (1.1) je popsáno šifrovací a dešifrovací schéma proudové šifry.

Definice 6 (Korelace). Mějme náhodné veličiny U a V . Jejich vzájemnou korelaci ϵ pak definujeme jako $\epsilon(U, V) = P(U = V) - \frac{1}{2}$. Pravděpodobnost $p = P(U = V) = \frac{1}{2} + \epsilon$ nazýváme korelační pravděpodobností.

Pro $n \in \mathbb{N}$ budeme množinu všech Booleovských funkcí značit jako B_n

$$B_n = \{f : \{0,1\}^n \rightarrow \{0,1\}\}.$$

Definice 7 (Korelace funkcí). Nechť $f, g \in B_n$, pak definujeme korelaci funkcí f a g jako $\epsilon(f, g) = P(f = g) - \frac{1}{2}$.

Poznámka. $P(f = g)$ máme na mysli pravděpodobnost jevu, že funkce f má stejnou hodnotu jako funkce g , tedy $P(f = g) = \frac{|\{x \in \mathbb{F}_2^n \mid f(x) = g(x)\}|}{2^n}$.

Definice 8 (Lineární Booleovská funkce). Pro $w \in \mathbb{F}_2^n$ definujeme Lineární Booleovskou funkci $l^{(w)}$ takto:

$$l^{(w)}(x) = x * w,$$

kde $x * w = \sum_{i=1}^n x_i w_i$ v tělese \mathbb{F}_2 , kde $x = (x_1, x_2, \dots, x_n)$ a $w = (w_1, w_2, \dots, w_n)$.

Definice 9 (Balancovaná Booleovská funkce). Mějme funkci $f \in B_n$, řekneme, že f je balancovaná právě tehdy, když $|f^{-1}(0)| = |f^{-1}(1)| = 2^{n-1}$.

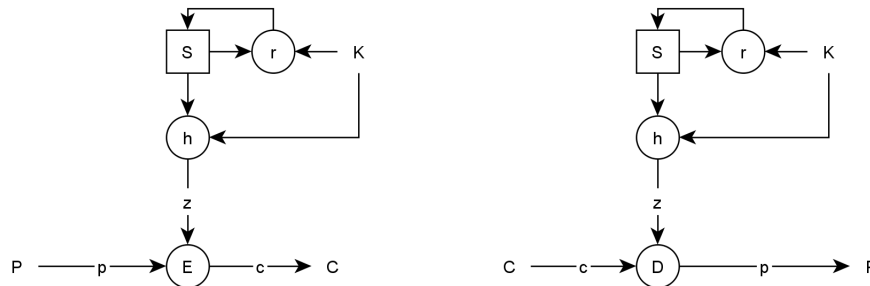
Definice 10. Necht' $f \in B_n$ a $m \in \mathbb{N}$, $m < n$. Pak $f^{(m)}$ definujeme jako množinu funkcí $(n - m)$ proměnných, které vzniknou dosazením libovolných hodnot do libovolných m proměnných f .

Definice 11 (Korelační odolnost funkce). Necht' $f \in B_n$ a $m \in \mathbb{N}$, $m < n$. Funkci f nazveme m -odolnou, pokud platí, že $\forall f' \in f^{(m)}$ je f' balancovaná.

Tvrzení 2. Uvažujme funkci $f \in B_n$, která je m -odolná, pak f je i ($m - 1$)-odolná.

Důkaz. f je m -odolná tedy $\forall f' \in f^{(m)}$ jsou balancované. Stačí ukázat, že $\forall f'' \in f^{(m-1)}$ jsou také balancované. Pro spor uvažujme, že existuje funkce $f'' \in f^{(m-1)}$, která není balancovaná. BÚNO $f'' = f(l_0, l_1, \dots, l_{m-2}, x_{m-1}, \dots, x_{n-1})$, kde $l_i \in \mathbb{F}_2$ jsou pevně dosazené hodnoty. Pro každé $j \in \{m - 1, m, \dots, n - 1\}$ a pro $\forall t \in \mathbb{F}_2$ však existuje funkce $f' \in f^{(m)}$ taková, že $f' = f(l_0, l_1, \dots, l_{m-2}, x_{m-1}, \dots, t, \dots, x_{n-1})$, kde t je pevně dosazená hodnota na j -té pozici. Tedy f'' můžeme rozdělit podle j -té pozice do dvou balancovaných funkcí, tedy f'' je balancovaná, což je spor. \square

Definice 12 (Kombinační generátor). Kombinační generátor je proudová šifra, která ke generování proudu klíče využívá soubor n LFSR, jejichž výstupy kombinuje nelineární Booleovskou funkcí h . Platí, že $\mathcal{P} = \mathcal{C} = \mathcal{L} = \{0, 1\}$, \mathcal{S} je množina stavů všech LFSR, obnovovací funkci r zastupuje princip posunů LFSR a výstupní funkce h je definována jako Booleovská funkce $h(u_i^{(1)}, u_i^{(2)}, \dots, u_i^{(n)}) = z_i$, kde u_i^j je výstup j -tého LFSR v čase i , a platí $e_z(p) = p \oplus z$, $d_z(c) = c \oplus z$ pro $\forall p \in P$, $\forall c \in C$.



Obrázek 1.1: Šifrovací a dešifrovací schéma proudové šifry.

Na obrázku (1.2) je znázorněno šifrovací a dešifrovací schéma kombinačního generátoru.

Definice 13 (Lineární kód). Lineární q -ární kód C délky n a dimenze k (značení $[n,k]_q$) je lineární podprostor dimenze k vektorového prostoru \mathbb{F}_q^n .

$c = (c_0, c_1, \dots, c_{n-1}) \in C$ nazýváme kódovým slovem kódu C . V dalším textu budeme pracovat jen s binárními lineárními kódy, to jsou $[n,k]_q$ s $q = 2$. Ty značíme $[n,k]$.

Definice 14 (Duální kód). Lineární kód H nazveme duálním kódem lineárnímu $[n,k]$ kódu C , pokud platí, že

$$H = \{h \in \mathbb{F}_2^n \mid \sum_{i=0}^{n-1} h_i c_i = 0 \ \forall c \in C\}.$$

Definice 15 (Hammingova vzdálenost). Mějme vektory $u, v \in \mathbb{F}_q^n$, pak Hammingovu vzdálenost u a v definujeme jako počet pozic, ve kterých se u a v liší. Značíme ji $d_H(u,v)$. Speciálně pro binární případ, kdy $q=2$ platí $d_H(u,v) = \sum_{i=1}^n (u_i \oplus v_i)$.

Definice 16 (Hammingova váha). Mějme vektor $u \in \mathbb{F}_q^n$, pak Hammingovu váhu u definujeme jako počet nenulových prvků u . Hammingovu váhu značíme $w_H(u)$.

Poznámka. Dá se snadno nahlédnout, že $w_H(u) = d_H(u,0)$, kde $0 \in \mathbb{F}_q^n$ představuje nulový vektor stejné délky jako u .

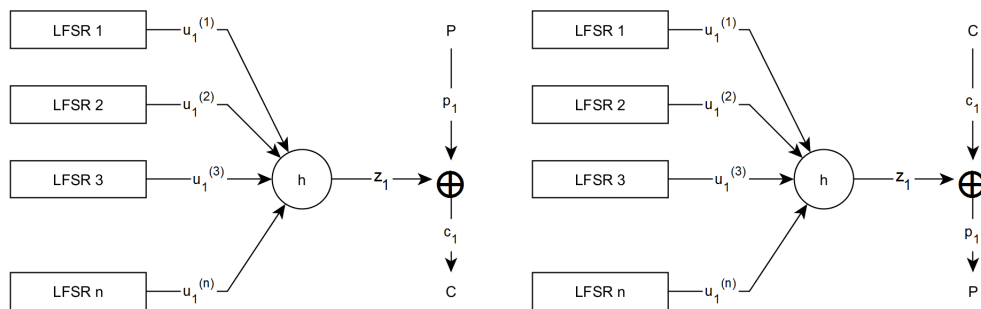
Definice 17 (Binární symetrický kanál). Binární symetrický kanál s chybovosť $p \in (0,1)$ definujeme jako $f : \mathbb{F}_2 \rightarrow \mathbb{F}_2$,

$$f(u) = u \oplus b,$$

kde $u, b \in \mathbb{F}_2$, $P(b = 0) = 1 - p$ a $P(b = 1) = p$. Takový kanál značíme $BSC[p]$ (z angl. Binary symmetric channel).

Definice 18 (Informační entropie). Mějme konečný prostor stavů $S = \{s_1, s_2, \dots, s_n\}$, kde $n \in \mathbb{N}$ a pravděpodobnosti jednotlivých stavů $P(s_i)$ takové, že $\sum_{i=1}^n P(s_i) = 1$ a $\forall i \in \{1, \dots, n\} P(s_i) > 0$. Pak informační entropii S definujeme jako

$$H(S) = - \sum_{i=1}^n P(s_i) \log_2(P(s_i)).$$



Obrázek 1.2: Šifrovací a dešifrovací schéma kombinačního generátoru.

Definice 19 (Entropická funkce). Entropickou funkci $H(a)$ *definujeme pro* $a \in (0,1)$ *následovně*

$$\begin{aligned} H(a) &= -x \log_2(a) - (1-a) \log_2(1-a) && 0 < a < 1 \\ H(a) &= 0 && a \in \{0,1\}. \end{aligned}$$

2. Korelační útok

Původní zmínka o korelačních útocích se datuje k roku 1985, kdy T. Siegenthaler tento druh útoku představil ve svém článku *Decrypting a Class of Stream Cipher Using Ciphertext only* (Siegenthaler, 1985). V něm ukazuje, že při útoku na proudové šifry typu kombinačního generátoru lze využít existence nenulové korelace mezi výstupem a jedním či skupinou vstupů do kombinační funkce a za pomoci statistického modelu značně snížit počet pokusů potřebných k prolomení šifry hrubou silou.

Motivací k tomuto útoku je fakt, že k úplnému prolomení kombinačního generátoru potřebujeme znát všechny počáteční stavy všech registrů kombinačního generátoru. Najít tyto stavy hrubou silou znamená odhadnout 1 z $2^{l_0+l_1+\dots+l_{n-1}}$ stavů, kde l_i je délka registru i a n počet registrů samotných. Pokud bychom však dokázali najít způsob, jak odhadnout stavy jednotlivých registrů nezávisle na ostatních, můžeme úlohu zjednodušit na odhad i stavů každý z 2^{l_i} možností.

Korelační útok využívá k separování odhadů počátečních stavů znalosti korelačních závislostí následovně. Předpokládáme, že máme proud klíče z délky N , kde $z = (z_0, z_1, \dots, z_{N-1})$ a zároveň známe charakteristické polynomy všech LFSR. Pokud existuje nenulová korelace $\epsilon = p - \frac{1}{2}$, kde $p = P(u_i = z_i)$, mezi výstupní posloupností u vybraného LFSR a proudem klíče z pak můžeme nalézt počáteční stav tohoto LFSR nezávisle na ostatních.

Poznámka. Pokud je $\epsilon = 0$, pak mezi proudem klíče a výstupní lineární rekurentní posloupností neexistuje žádná statisticky zjistitelná lineární závislost. V takovém případě nepřinese použití korelačního útoku žádné zjednodušení.

2.1 Siegenthalerův útok

Princip originálního útoku na kombinační generátor představeného Siegenthalerem je následující. Předpokládejme, že mezi sekvencí u generovanou jedním z LFSR použitých v kombinačním generátoru délky l a proudem klíče z existuje nenulová korelace $\epsilon = p - \frac{1}{2}$ a také předpokládáme, že charakteristický polynom LFSR $g(x) = \sum_{k=0}^l c_k x^k$ je znám. Tento předpoklad v původním útoku nebyl, místo toho se ještě prohledávalo přes všechny primitivní polynomy stupně l . Pro LFSR délky l existuje 2^l možných počátečních stavů, které se musí otestovat a najít z nich ten, který generuje správnou posloupnost, která má korelaci s proudem klíče rovnou ϵ .

Z každého počátečního stavu $(u_0, u_1, \dots, u_{l-1})$ je nagenеровána posloupnost délky N

$$u' = (u'_0, u'_1, \dots, u'_{N-1}),$$

kde

$$\begin{aligned} u'_i &= u_i & \forall i \in \{0, 1, \dots, l-1\} \\ u'_{i+l} &= - \sum_{k=0}^{l-1} c_k u'_{i+k} & \forall i \geq 0. \end{aligned} \tag{2.1}$$

Vezmeme dále proud klíče délky N $z = (z_0, z_1, \dots, z_{N-1})$ a označíme

$$\omega = N - d_H(u', z). \tag{2.2}$$

Nyní musíme zvážit dvě hypotézy:

H_1 : Daný inicializační stav je správný.

H_0 : Daný inicializační stav je špatný.

ω má za těchto hypotéz binomické rozdělení se střední hodnotou $m_{\omega|H_i}$ a rozptylem $\sigma_{\omega|H_i}^2$, kde

$$\begin{aligned} m_{\omega|H_1} &= N \cdot (1 - p) \\ \sigma_{\omega|H_1}^2 &= N \cdot p \cdot (1 - p) \\ m_{\omega|H_0} &= \frac{N}{2} \\ \sigma_{\omega|H_0}^2 &= \frac{N}{4}. \end{aligned}$$

Tedy vidíme, že pokud bude délka proudu klíče N dostatečně dlouhá, s vysokou pravděpodobností se pak ω bude pro správně zvolený inicializační stav u' blížit očekávané hodnotě $m_{\omega|H_1}$ a pro špatný inicializační stav bude blízko $m_{\omega|H_0}$. Čím větší bude ϵ , tím více budou tato binomická rozdělení rozdílná a lépe bude rozeznat správný stav od špatného.

Pokud je proud klíče korelován se skupinou registrů velikosti s , postup je následující. Řekněme, že mezi součtem sekvencí generovaných $LFSR_1, LFSR_2, \dots, LFSR_s$ existuje nenulová korelace s proudem klíče

$$\epsilon = P(u_i^{(1)} + u_i^{(2)} + \dots + u_i^{(s)} = z_i) - \frac{1}{2} \neq 0. \quad (2.3)$$

Pak existuje 2^{l_j} možných počátečních stavů pro každý $LFSR_j$ $j \in \{1, 2, \dots, s\}$ a tedy celkově $K = \prod_{k=1}^s 2^{l_k}$ kombinací možných počátečních stavů těchto LFSR. Následně stejně jako v minulém případě nagenerujeme posloupnosti délky N z každého $LFSR_j$. Dále spočítáme ω pro všech K možností

$$\omega = N - d_H(u'_1 + u'_2 + \dots + u'_s, z), \quad (2.4)$$

kde u'_j značí jednu z 2^{l_j} možných posloupností délky N nagenerované z počátečního stavu délky l_j .

Jelikož ale není jisté, že ω pro správné počáteční stavy bude to nejbližší očekávané hodnotě $m_{\omega|H_1}$, musí být provedeno ještě testování a následné korekce.

Algoritmus 1: Algoritmus Sigenthalerova útoku

Krok 1: Nastav počáteční stavy všech LFSR se známou korelací na hodnoty, při kterých byly jednotlivé ω nejbližše očekávané hodnotě $m_{\omega|H_1}$.

Krok 2: Projdi hrubou silou všechny možné stavy LFSR, pro které neexistuje korelace. A zkoušej, zdali některá z kombinací počátečních stavů negeneruje proud klíče z .

Krok 3: Pokud žádná taková kombinace stavů neexistuje, změň jeden počáteční stav v odhadu LFSR se známou korelací, pak přejdi na Krok 2.

Krok 4: Ukonči algoritmus s n počátečními stavy, které po kombinaci funkcí h tvoří proud klíče z .

V Kroku 3 jsou počáteční stavy měněny od nejvíc pravděpodobných a postupně se zkoušejí všechny kombinace.

Věta 3. *Nechť $f \in B_n$, potom f je m -odolná právě tehdy, když $\epsilon(f, l^{(u)}) = 0$ pro $u \in \mathbb{F}_2^n : w_h(u) \leq m$.*

Důkaz. viz. (Carlet, 2007) □

Důsledek. V kombinačním generátoru s m -odolnou kombinační funkcí je nejmenší skupina LFSR, se kterou můžeme najít nenulovou korelaci ϵ , velikosti $m + 1$.

Důsledek. Pokud může být korelace nalezena pro každé $LFSR_j$ samostatně (funkce h je 0-odolná vůči korelaci), pak složitost nalezení počátečních stavů všech $LFSR_j$ je redukována z $\prod_{i=1}^n (2^{l_i} - 1)$ na $\sum_{i=1}^n (2^{l_i} - 1)$.

Tvrzení 4. *Výpočetní složitost útoku na jeden LFSR, se známou korelací s proudem klíče je u Siegenthalerova útoku $O(2^l)$, kde l je délka LFSR.*

Důkaz. Jedná se o prohledávání všech možných počátečních stavů, kterých je 2^l . □

Důsledek. Mějme kombinační generátor tvořený n LFSR $_j$ délky l_j , které generují posloupnosti u_j . Dále uvažujme k disjunktních množin indexů $N_i \subset \{1, 2, \dots, n\}$ $i = 1, 2, \dots, k$ takových, že platí $P(\sum_{j \in N_i} u_j = z) = \frac{1}{2} + \epsilon_i$, $\epsilon_i > 0$. A označme množinu $N_{k+1} = \{1, 2, \dots, n\} \setminus \cup_{i=1}^k N_i$. Pak složitost Siegenthalerova útoku na tento kombinační generátor je řádu

$$O\left(\sum_{i=1}^{k+1} 2^{\sum_{l \in N_i} l_j}\right).$$

Důkaz. Nejdříve nalezneme nejpravděpodobnější kandidáty na počáteční stavy LFSR ve skupině N_i , pro kterou existuje nenulová korelace ϵ_i s proudem klíče, ty najdeme projitím všech možných stavů, kterých je $2^{\sum_{j \in N_i} l_j}$. Takto projdeme všechny skupiny. Nakonec provedeme korekce těchto kandidátů pomocí vygenerování domnělého proudu klíče tak, že hrubou silou projedeme všechny počáteční stavy skupiny N_{k+1} , kterých je $2^{\sum_{j \in N_{k+1}} l_j}$ a vybereme kombinaci kandidátů na počáteční stavy s velkou pravděpodobností v každé skupině tak, aby byl generován správný proud klíče. □

Tento útok byl v původním článku (Siegenthaler, 1985) prokázán jako efektivní nástroj, pro útok na kombinační generátor, pro maximální délku LFSR $l < 50$ s korelační pravděpodobností $p \geq 0,75$. Naopak pro dlouhé LFSR a pravděpodobnost blízko 0,5 byly výsledky srovnatelné s útokem hrubou silou.

2.2 Vztah k dekódovacímu problému

Korelační útoky můžeme také interpretovat pomocí poznatků z teorie kódů. Uvažujme stejně jako u Siegenthalerova útoku, že známe strukturu LFSR, korelaci ϵ proudu klíče z a výstupní posloupnosti u , a charakteristický polynom $g(x) =$

$\sum_{k=0}^l c_k x^k$. Opět nageneryjme dle (2.1) ze všech 2^l možných počátečních stavů posloupnosti délky N .

$$\begin{aligned} u'_1 &= (u'_{1,0}, u'_{1,1}, \dots + u'_{1,N-1}) \\ u'_2 &= (u'_{2,0}, u'_{2,1}, \dots + u'_{2,N-1}) \\ &\vdots \\ u'_{2^l} &= (u'_{2^l,0}, u'_{2^l,1}, \dots + u'_{2^l,N-1}) \end{aligned} \tag{2.5}$$

Označme dále $A = \{u'_1, u'_2, \dots, u'_{2^l}\}$. Dá se snadno nahlédnout, že A je lineární podprostorem dimenze l vektorového prostoru \mathbb{F}_2^N a tedy tvoří lineární binární $[N, l]$ kód C . Posloupnosti $u'_i \in A$ pak představují kódová slova C .

Proud klíče z délky N pak můžeme chápat jako kódové slovo kódu C , ke kterému byla přičtena neznámá sekvenční šuma $e = (e_0, e_1, \dots, e_{N-1})$, kde $e_i \in \{0, 1\}$ jsou náhodné nezávislé proměnné a $P(e_i = 0) = \frac{1}{2} + \epsilon$. Šum e můžeme generovat pomocí Binárního symetrického kanálu. Dekódovací úloha pak může být formulována následovně.

Obdrželi jsme slovo z $\text{BSC}[\frac{1}{2} - \epsilon]$, rozhodněte které $u'_i \in A$ bylo zasláno.

Hledaný počáteční stav je pak roven prvním l prvkům nalezeného kódového slova.

Tvrzení 5. *Uvažujme dekodovací problém definovaný výše. Pro dekodování proudu klíče z na správné kódové slovo musí být délka kódu C alespoň N_0 , kde*

$$N_0 = \frac{l}{1 - H(\frac{1}{2} - \epsilon)}.$$

Tento odhad vychází z definice BSC a kapacity kanálu, která je pro $\text{BSC}[\frac{1}{2} - \epsilon]$ definována jako $c = 1 - H(\frac{1}{2} - \epsilon)$. Což znamená, že jeden zasláný prvek přeneše v průměru $\frac{1}{c}$ původní informace. Každé kódové slovo kódu C obsahuje informaci o l prvcích (počátečním stavu, ze kterého bylo slovo vygenerováno), zbylých $N - l$ prvků je zde redundantních tak abychom dokázali rekonstruovat informaci v případě šumu. Tedy potřebujeme přenést alespoň l prvků původní informace. Tedy musí být zasláno minimálně $\frac{l}{c} = \frac{l}{1 - H(\frac{1}{2} - \epsilon)}$ prvků. Tato mez je však teoretická a v praxi musí být $N \gg N_0$.

3. Rychlé korelační útoky

V této kapitole se budeme zabývat rychlými korelačními útoky, které prvně představili v roce 1989 Willy Meier a Othmar Staffelbach ve článku *Fast Correlation Attacks on Certain Stream Ciphers* (Meier a Staffelbach, 1989), dále vycházíme z článků, které na tuto práci navázaly (Meier, 2011) a (Jönsson, 2002). Tyto práce představují podstatné vylepšení korelačních útoků. Nejdříve bude zhruba vysvětlen princip rychlých korelačních útoků, poté bude prezentován statistický model, na kterém jsou tyto útoky založeny, a nakonec představíme i dva algoritmy, které je aplikují.

3.1 Základní princip a motivace

V této kapitole se budeme zaměřovat na případ, kdy existuje nenulová korelace ϵ mezi proudem klíče z a výstupem u z jednoho LFSR délky l (budeme předpokládat, že kombinační funkce je 0–korelačně odolná). Příklad, kdy je funkce k –korelačně odolná $k \in \mathbb{N}$ a tedy existuje nenulová korelace se skupinou $k + 1$ LFSR, se řeší pomocí kombinací jednotlivých počátečních stavů LFSR tak, jak je popsáno v kapitole 2. V minulé kapitole jsme také ukázali interpretaci korelačních útoků pomocí dekódovacího problému. Tato reprezentace se ukázala v praxi vhodná i pro rychlé korelační útoky, a proto ji budeme používat v této kapitole po celou dobu. Pro rychlé korelační útoky popsané v této kapitole tedy definujeme zadání: Mějme známý proud klíče z délky N , který představuje výstup z BSC[$1 - p$], kde $p = P(z_i = u_i) = \frac{1}{2} + \epsilon$, po zaslání neznámé posloupnosti u délky N . Najděte správnou posloupnost u , která byla zaslána, když víme, že se jedná o výstupní posloupnost z LFSR délky l se známým charakteristickým polynomem $g(x)$ váhy t . Což je ekvivalentní nalezení počátečního stavu daného LFSR tak, aby generoval posloupnost u .

Za rychlé korelační útoky považujeme všechny, které jsou výrazně rychlejší než Siegenthalerův útok procházející všechny možné počáteční stavy LFSR. Konkrétně všechny útoky s výpočetní složitostí pro útok na samostatné LFSR $O(2^{cl})$, kde l je délka LFSR a $c < 1$ konstanta. Na rozdíl od Siegenthalerova útoku potřebují rychlé korelační útoky mnohem větší poměr $d = \frac{N}{l}$ velikosti známého proudu klíče ku délce LFSR, na druhou stranu ale dokáží útočit na LFSR s délkou $l > 100$ a korelační pravděpodobností p blízkou 0,5.

Základní myšlenkou rychlých korelačních útoků je využití faktu, že každý prvek u_n LFSR posloupnosti u splňuje několik lineárních vztahů odvozených z charakteristického polynomu, které obsahují další prvky u . Pokud stejné vztahy zkusíme aplikovat na korespondující prvky posloupnosti proudu klíče z , dostaneme pro každý prvek z_n rovnice, které buď budou platit, či ne. Počty rovnic, které budou platit i na posloupnost z , nám pak budou nápomocné k rozhodnutí, které $z_n = u_n$. Intuitivně, čím větší počet rovnic, které platí, tím větší je pravděpodobnost, že daný prvek je správně.

Rychlé korelační útoky se dělí do dvou fází. První fáze je přípravná, hledáme v ní lineární rovnice odvozené z charakteristického polynomu. Kládem je důraz na hledání co možná největšího systému lineárně nezávislých rovnic. Každá další rovnice, která není lineárně nezávislá, totiž již popisuje vztah, který známe a

nepřispívá k přesnějšímu vyhodnocení. Ve druhé fázi jsou pak tyto rovnice využity v rychlém dekódovacím algoritmu pro nalezení správného počátečního stavu LFSR. Dva základní dekódovací algoritmy využívané v rychlých korelačních útocích se nazývají jednoduše Algoritmus A a Algoritmus B.

Princip Algoritmu A je takový, že nejdříve vybereme množinu prvků I_0 posloupnosti z , které splňují dostatek rovnic a jsou tedy s velkou pravděpodobností správně. Z těchto prvků pak dopočítáme pomocí rekurentních vztahů prvky na pozicích 0 až $l - 1$. Těchto prvních l prvků pak využijeme pro odhad počátečního stavu posloupnosti u , který za příznivých podmínek generuje posloupnost blízkou správné posloupnosti. Dále pak budou potřeba pouze malé korekce I_0 . Tyto korekce jsou provedeny velmi redukovaným prohledáváním všech možností.

Algoritmus B také využívá vztahy derivované z charakteristického polynomu, nehledáme v něm však nejvíce spolehlivé prvky. Místo toho vezmeme všechny prvky z společně s jejich pravděpodobnostmi, že jsou správné. To nás vede k zavedení nové pomocné pravděpodobnosti p^* , která pro každý prvek z_n posloupnosti z definuje podmíněnou pravděpodobnost, že $z_n = u_n$ v závislosti na počtu splněných rovnic. Tato pravděpodobnost se nejdříve několikrát iterativně přepočítá pro každý prvek a po několika opakováních jsou pak všechny prvky z_n s pravděpodobností p^* menší než daná, dobře zvolená mez, invertovány. Za příznivých podmínek můžeme očekávat, že počet špatných prvků se tímto procesem zmenší. Tento proces restartujeme několikrát s nově upravenou posloupností z , dokud nedostaneme originální posloupnost u .

3.2 Výpočetní a statistický model

Prvek u_n lineárně rekurentní posloupnosti u s charakteristickým polynomem $g(x) = c_0 + c_1x + c_2x^2 + \dots + c_lx^l \in \mathbb{F}_2[x]$ váhy t , kde $c_l = 1$ můžeme zapsat takto

$$u_n = c_{l-1}u_{n-1} + c_{l-2}u_{n-2} + \dots + c_0u_{n-l}. \quad (3.1)$$

Z tohoto vyjádření můžeme pozorovat, že při vyjadřování dalších prvků se u_n objeví v l pozicích, z nichž v $t - 1$ má nenulový koeficient c_i .

$$\begin{aligned} u_n &= c_{l-1}u_{n-1} + c_{l-2}u_{n-2} + \dots + c_0u_{n-l} \\ u_{n+1} &= c_{l-1}u_n + c_{l-2}u_{n-1} + \dots + c_0u_{n-l} \\ &\vdots \\ u_{n+l-1} &= c_{l-1}u_{n+l-2} + \dots + c_1u_n + c_0u_{n-l} \\ u_{n+l} &= c_{l-1}u_{n+l-1} + c_{l-2}u_{n+l-2} \dots + c_0u_n \end{aligned} \quad (3.2)$$

Tímto způsobem lze nalézt t lineárních rovnic obsahující daný prvek u_n v závislosti na dalších $t - 1$ znacích stejné posloupnosti. Každý násobek charakteristického polynomu definuje také lineární vztahy pro u . Jelikož pracujeme nad tělesem charakteristiky 2, můžeme využít vztahu $c(x)^j = c(x^j)$ pro všechna j tvaru $j = 2^i$ $i \in \mathbb{N}$, touto cestou získáme další rovnice. Všechny tyto rovnice mají t nenulových koeficientů. Těmito dvěma metodami jsou primárně získávány rovnice, které se využívají v dekódovacích algoritmech Rychlých korelačních útoků. V kapitole 4 jsou popsány pokročilejší metody pro získávání těchto rovnic.

Ke správnému simulování dekódovacích algoritmů potřebujeme znát odhad počtu lineárních rovnic pro prvek u_n vytvořených těmito dvěma metodami. Tento odhad slouží pouze k určování řádové složitosti jednotlivých algoritmů a při praktickém využívání víme přesný počet, tedy můžeme v určování odhadu některé věci zanedbat.

Nejdříve vyjádříme počet všech rovnic, které můžeme odvodit pro LRP u délky N . Rovnice vzniklé z j -krát opakovaného mocnění ($j \geq 0$) mají délku (rozdíl indexu prvního a posledního nemulového členu) $2^j l$, je jich tedy $N - 2^j l + 1$, toto číslo musí být však kladné, takže j může být nanejvýše $\lfloor \log_2(\frac{N+1}{l}) \rfloor$. Tedy celkový počet všech rovnic může být vyjádřen jako

$$\begin{aligned}
M &= \sum_{j=0}^{\lfloor \log_2(\frac{N+1}{l}) \rfloor} (N - 2^j l + 1) = \\
&= (N + 1)(\lfloor \log_2(\frac{N+1}{l}) \rfloor + 1) - \sum_{j=0}^{\lfloor \log_2(\frac{N+1}{l}) \rfloor} 2^j l = \\
&= (N + 1)(\lfloor \log_2(\frac{N+1}{l}) \rfloor + 1) - (2^{\lfloor \log_2(\frac{N+1}{l}) \rfloor + 1} - 1)l \simeq \\
&\simeq (N + 1)\lfloor \log_2(\frac{N+1}{l}) \rfloor + N + 1 - (\frac{2(N+1)}{l} - 1)l = \\
&= (N + 1)\lfloor \log_2(\frac{N+1}{l}) \rfloor + -N - 1 + l = \\
&= (N + 1)(\lfloor \log_2(\frac{N+1}{l}) \rfloor - \log_2 2) + l \simeq \\
&\simeq (N + 1)\lfloor \log_2(\frac{N+1}{2l}) \rfloor + l.
\end{aligned}$$

Každá z těchto rovnic obsahuje t prvků u . Tedy průměrný počet rovnic m na prvek je roven

$$m = M \frac{t}{N} = (\lfloor \log_2(\frac{N+1}{2l}) \rfloor + \frac{l + \lfloor \log_2(\frac{N+1}{2l}) \rfloor}{N})t.$$

V našich aplikacích však platí, že $\frac{l + \lfloor \log_2(\frac{N+1}{2l}) \rfloor}{N}t \ll 1$ tedy pro účely odhadů můžeme zjednodušit na

$$m = \lfloor \log_2(\frac{N+1}{2l}) \rfloor t. \quad (3.3)$$

Pro dané u_n pak mohou být rovnice zapsány takto

$$\begin{aligned}
u_n + b_{1,1} + b_{1,2} + \dots + b_{1,t-1} &= 0 \\
u_n + b_{2,1} + b_{2,2} + \dots + b_{2,t-1} &= 0 \\
&\vdots \\
u_n + b_{m,1} + b_{m,2} + \dots + b_{m,t-1} &= 0,
\end{aligned} \quad (3.4)$$

kde $b_{i,j}$ značí prvek u obsažený v i -té rovnici společně s u_n . V algoritmech pro dekódování nás však zajímá, kolik těchto rovnic bude platit, když je aplikujeme

na posloupnost proudu klíče z . Dané rovnice můžeme vyjádřit takto

$$\begin{aligned} z_n + y_{1,1} + y_{1,2} + \dots + y_{1,t-1} &= L_1 \\ z_n + y_{2,1} + y_{2,2} + \dots + y_{2,t-1} &= L_2 \\ &\vdots \\ z_n + y_{m,1} + y_{m,2} + \dots + y_{m,t-1} &= L_m, \end{aligned} \tag{3.5}$$

kde $y_{i,j}$ jsou prvky z na odpovídajících indexových pozicích s $b_{i,j}$ a $L_i \in \{0,1\}$.

3.2.1 Statistický model

Nyní již máme dostatek informací k tomu, abychom mohli definovat statistický model, o který se budou opírat Rychlé korelační útoky.

Nejdříve nahradíme prvky posloupnosti u v rovnicích (3.4) množinou binárních náhodných proměnných $U = \{u_n, b_{1,1}, b_{1,2}, \dots, b_{1,t-1}, b_{2,1}, \dots, b_{m,t-1}\}$, splňujících rovnice

$$u_n + b_{i,1} + b_{i,2} + \dots + b_{i,t-1} = 0, \tag{3.6}$$

kde $i \in 1, 2, \dots, m$. Obdobně definujeme i množinu binárních náhodných proměnných $Z = \{z_n, y_{1,1}, y_{1,2}, \dots, y_{1,t-1}, y_{2,1}, \dots, y_{m,t-1}\}$, kterými nahradíme prvky v rovnici (3.5). Pro množiny U a Z platí vzájemné vztahy definované v (3.7).

$$\begin{aligned} P(u_n = z_n) &= p \\ P(b_{i,j} = y_{i,j}) &= p \end{aligned} \tag{3.7}$$

Dále definujeme náhodné proměnné b_i a y_i

$$\begin{aligned} b_i &= b_{i,1} + b_{i,2} + \dots + b_{i,t-1} \\ y_i &= y_{i,1} + y_{i,2} + \dots + y_{i,t-1} \end{aligned} \tag{3.8}$$

a také L_i

$$L_i = z_n + y_i. \tag{3.9}$$

Můžeme nahlédnout, že stav $L_i = 0$ nastane ve dvou případech:

- $z_n = u_n \wedge b_i = y_i$
- $z_n \neq u_n \wedge b_i \neq y_i$

obdobně $L_i = 1$ nastane v případech:

- $z_n \neq u_n \wedge b_i = y_i$
- $z_n = u_n \wedge b_i \neq y_i$.

Tedy potřebujeme vyjádřit pravděpodobnost $P(b_i = y_i)$, s jakou se budou sumy prvků b_i a y_i rovnat. Z rovnic (3.7) a (3.8) plyne, že pravděpodobnost $P(b_i = y_i)$ je nezávislá na indexu i a dá se tedy vyjádřit jen v závislosti na p a t , $P(b_i = y_i) = s(p, t)$.

Tvrzení 6. Uvažujme náhodné proměnné definované výše. Pak pravděpodobnost $s(p,t) = P(b_i = y_i)$ se dá vyjádřit rekurzivně jako

$$\begin{aligned} s(p,t) &= p \cdot s(p,t-1) + (1-p)(1-s(p,t-1)), t > 1 \\ s(p,1) &= p. \end{aligned}$$

Důkaz. Pro $t = 1$ platí, že $b_i = b_{i,1}$ a $y_i = y_{i,1}$, tedy nastane rovnost právě tehdy, když se $b_{i,1} = y_{i,1}$, což dle (3.7) je rovno p , tedy $s(p,1) = p$. Při vyšších t platí, že $b_i = y_i$ právě tehdy, když se liší v sudém počtu prvků. Sudý počet prvků při t znacích může nastat, pokud při $t-1$ znacích byl sudý počet odlišných prvků a t -tý prvek byl stejný, což nastane s pravděpodobností $p \cdot s(p,t-1)$, anebo při $t-1$ znacích byl lichý počet odlišných prvků a t -tý prvek se lišil také, což nastane s pravděpodobností $(1-p) \cdot (1-s(p,t-1))$. Tedy $s(p,t) = ps(p,t-1) + (1-p)(1-s(p,t-1))$. □

Poznámka. Pro $s(p,t)$ existuje řešení rekurence.

$$s(p,t) = \frac{1}{2} + 2^{t-1} \cdot \epsilon^t \quad (3.10)$$

(Chepyzhov a kol., 2000)

V závislosti na pravděpodobnosti s pak můžeme vyjádřit pravděpodobnosti pro případy, kdy $L_i = 0$ a $L_i = 1$

$$\begin{aligned} P(L_i = 0) &= p \cdot s(p,t) + (1-p) \cdot (1-s(p,t)) \\ P(L_i = 1) &= p \cdot (1-s(p,t)) + (1-p) \cdot s(p,t). \end{aligned}$$

Nyní tedy již můžeme vyjádřit, jaká je pravděpodobnost, že prvek z_n je správně za podmínky, že přesně h jeho rovnic je splněno. Uvažujme případ $S : L_1 = L_2 = \dots = L_h = 0 \wedge L_{h+1} = L_{h+2} = \dots = L_m = 1$. Pak pravděpodobnost jevu S je

$$\begin{aligned} P(S) &= P(L_1 = \dots = L_h = 0 \wedge L_{h+1} = \dots = L_m = 1 | z_n = u_n) \cdot P(z_n = u_n) + \\ &+ P(L_1 = \dots = L_h = 0 \wedge L_{h+1} = \dots = L_m = 1 | z_n \neq u_n) \cdot P(z_n \neq u_n) = \\ &= p \cdot (s(p,t))^h \cdot (1-s(p,t))^{m-h} + (1-p) \cdot (1-s(p,t))^h \cdot (s(p,t))^{m-h}. \end{aligned} \quad (3.11)$$

Jev S totiž nastane ve dvou případech:

- $z_n = u_n \wedge (b_i = y_i \text{ pro } i \in \{1, 2, \dots, h\}, b_i \neq y_i \text{ pro } i \in \{h+1, h+2, \dots, m\})$. Tento jev má pravděpodobnost $p \cdot (s(p,t))^h (1-s(p,t))^{m-h}$.
- $z_n \neq u_n \wedge (b_i \neq y_i \text{ pro } i \in \{1, 2, \dots, h\}, b_i = y_i \text{ pro } i \in \{h+1, h+2, \dots, m\})$. Tento jev má pravděpodobnost $(1-p) \cdot (s(p,t))^{m-h} (1-s(p,t))^h$.

A tedy můžeme dle Bayesova vzorce odvodit pravděpodobnost $z_n = u_n$ za podmínky S , kterou budeme značit $p^*(p,m,h)$

$$\begin{aligned} p^*(p,m,h) &= P(z_n = u_n | S) = \frac{P(S | z_n = u_n) \cdot P(z_n = u_n)}{P(S)} = \\ &= \frac{p \cdot (s(p,t))^h \cdot (1-s(p,t))^{m-h}}{p \cdot (s(p,t))^h \cdot (1-s(p,t))^{m-h} + (1-p) \cdot (s(p,t))^{m-h} \cdot (1-s(p,t))^h}. \end{aligned} \quad (3.12)$$

Obdobně vyjádříme pravděpodobnost $z_n \neq u_n$ za podmínky S

$$P(z_n \neq u_n|S) = \frac{(1-p) \cdot (1-s(p,t))^h \cdot (s(p,t))^{m-h}}{p \cdot (s(p,t))^h \cdot (1-s(p,t))^{m-h} + (1-p) \cdot (s(p,t))^{m-h} \cdot (1-s(p,t))^h}. \quad (3.13)$$

Dále označme S^* množinu všech možností pro hodnoty L_i $i = 1, 2, \dots, m$. Pro úspěšnost algoritmu potřebujeme, aby pro velká h platilo, že

$$z_n = u_n \Rightarrow p^*(p, m, h) > p$$

a

$$z_n \neq u_n \Rightarrow p^*(p, m, h) < p.$$

Toto ověříme spočítáním středních hodnot pro $p^*(p, m, h)$ za jednotlivých případů.

$$\begin{aligned} E_0[p^*(p, m, h)] &= E[p^*(p, m, h)|z_n = u_n] = \\ &= \sum_{r \in (0,1)} [r \cdot P(r = P(z_n = u_n|S_i \in S^*)|z_n = u_n)] = \\ &= \sum_{(r_1, r_2, \dots, r_m) \in \mathbb{Z}_2^m} [P(z_n = u_n|L_1 = r_1, L_2 = r_2, \dots, L_m = r_m) \cdot \\ &\cdot P(L_1 = r_1, L_2 = r_2, \dots, L_m = r_m|z_n = u_n)] = \\ &= \sum_{h=0}^m \binom{m}{h} \frac{p \cdot (s(p,t))^h (1-s(p,t))^{m-h} \cdot (s(p,t))^h \cdot (1-s(p,t))^{m-h}}{p \cdot (s(p,t))^h \cdot (1-s(p,t))^{m-h} + (1-p) \cdot (1-s(p,t))^h \cdot (s(p,t))^{m-h}} \end{aligned} \quad (3.14)$$

Jelikož $S_i \in S^*$ nabývá konečně mnoho hodnot, můžeme přejít na sumu v závislosti na vektoru $(r_1, r_2, \dots, r_m) \in \mathbb{Z}_2^m$. Obdobně pak můžeme vyjádřit

$$\begin{aligned} E_1[p^*(p, m, h)] &= E[p^*(p, m, h)|z_n \neq u_n] = \\ &= \sum_{h=0}^m \binom{m}{h} \frac{p \cdot (s(p,t))^h (1-s(p,t))^{m-h} \cdot (s(p,t))^{m-h} \cdot (1-s(p,t))^h}{p \cdot (s(p,t))^h \cdot (1-s(p,t))^{m-h} + (1-p) \cdot (1-s(p,t))^h \cdot (s(p,t))^{m-h}}. \end{aligned} \quad (3.15)$$

Příklad. Pro případ, kdy $p = 0,75$, $t = 2$ a $m = 20$ platí $E_0[p^*(p, m, h)] = 0,9$ a $E_1[p^*(p, m, h)] = 0,3$. Převzato z (Meier a Staffelbach, 1989).

3.3 Algoritmus A

Předpokládáme, že známe posloupnost proudu klíče z délky N , charakteristický polynom $g(x)$ LFSR délky l a že výstupní posloupnost LFSR u je korelována s posloupností z s pravděpodobností $p = \frac{1}{2} + \epsilon$, $\epsilon > 0$. Náš úkol je nalézt správnou posloupnost u při znalostech z , p , l , $g(x)$. Tato posloupnost může být rekonstruována ze správně určených prvků pomocí řešení lineárních rovnic pro počáteční stav. Abychom dostali přibližnou posloupnost u , vybereme alespoň l prvků s největší pravděpodobností $p^*(p, m, h)$ nebo ekvivalentně ty, které splňují nejvíce vztahů (3.4).

Pro potřeby algoritmu a pozdějších výpočtů označíme některé pravděpodobnosti. Označme jev $H := |\{i|L_i = 0\}| \geq h$, tedy že je splněno alespoň h rovnic. Definujme pak $Q(p, m, h)$ jako pravděpodobnost, že fixní prvek posloupnosti z splňuje nejméně h z m rovnic,

$$Q(p,m,h) = P(H) = \sum_{i=h}^m \left[\binom{m}{i} (p(s(p,t)))^i (1-s(p,t))^{m-i} + (1-p)(1-s(p,t))^i (s(p,t))^{m-i} \right]. \quad (3.16)$$

Dále označme $R(p,m,h)$ jako pravděpodobnost, že $z_i = u_i$ a nejméně h z m rovnic je splněno,

$$R(p,m,h) = P(z_n = u_n \wedge H) = \sum_{i=h}^m \left[\binom{m}{i} p(s(p,t))^i (1-s(p,t))^{m-i} \right]. \quad (3.17)$$

A tedy stav, kdy $z_i = u_i$ za podmínky, že nejméně h z m rovnic je splněno, je vyjádřen jako $T(p,m,h)$,

$$T(p,m,h) = P(z_n = u_n | H) = \frac{P(z_n = u_n \wedge H)}{P(H)} = \frac{R(p,m,h)}{Q(p,m,h)}. \quad (3.18)$$

Tvrzení 7. $T(p,m,h)$ je pro fixní p a m rostoucí v závislosti na h .

Důkaz. Platí, že

$$\binom{m}{i} p(s(p,t))^i (1-s(p,t))^{m-i} > 0, 0 \leq i \leq m$$

$$\binom{m}{i} (p(s(p,t)))^i (1-s(p,t))^{m-i} + (1-p)(1-s(p,t))^i (s(p,t))^{m-i} > 0, 0 \leq i \leq m$$

$$\binom{m}{i} (1-p)(1-s(p,t))^i (s(p,t))^{m-i} > 0, 0 \leq i \leq m,$$

a tedy platí, že $Q(p,m,h)$ i $R(p,m,h)$ jsou klesající s rostoucím h . Jelikož také platí, že

$$Q(p,m,h) = R(p,m,h) + \sum_{i=h}^m \left[\binom{m}{i} (1-p)(1-s(p,t))^i (s(p,t))^{m-i} \right]$$

a $s(p,t) \geq \frac{1}{2}$ (3.10) tak platí, že

$$\frac{1}{T(p,m,h)} = \frac{Q(p,m,h)}{R(p,m,h)}$$

je také klesající. Tedy $T(p,m,h)$ je rostoucí v závislosti na h . □

Očekávaný počet prvků posloupnosti z , které splní H , je pak $Q(p,m,h) \cdot N$. Každý prvek posloupnosti z je s pravděpodobností $T(p,m,h)$ správně. Jelikož pro rekonstruování inicializačního vektoru LRP délky l potřebujeme alespoň l prvků, které splňují H , tak hledáme maximální h takové, že platí

$$Q(p,m,h) \cdot N > l. \quad (3.19)$$

I_0 pak označíme množinu všech prvků, pro které platí H a zároveň $|I_0| \geq l$. Z prvků I_0 pak vytvoříme soustavu lineárních rovnic pro inicializační vektor LRP.

Řešením těchto rovnic bude odhad inicializačního vektoru LRP. Jediný problém nastává, pokud soustava těchto rovnic je lineárně závislá a neexistuje jednoznačné řešení. Tento problém se dá řešit přidáváním dalších prvků, které splňují hodně rovnic, dokud nedostaneme systém rovnic maximální hodnoty. V našem modelu však předpokládáme, že počet prvků bude dostatečný tak, aby měl systém těchto rovnic maximální hodnotu a aby existovalo jednoznačné řešení. Očekávaný počet chybných prvků v I_0 je pak

$$r_{avg} = (1 - T(p,m,h))|I_0|. \quad (3.20)$$

Pokud je tento počet malý, pak inicializační vektor LRP může být nalezen pomocí zkoušení malých úprav I_0 .

Algoritmus 2: Algoritmus A

Krok 1: Spočítej odhad m pomocí vzorce (3.3).

Krok 2: Najdi maximální hodnotu h takovou, že $Q(p,m,h)N \geq l$.

Krok 3: Vytvoř pak z prvků I_0 pomocí řešení soustavy rovnic odhad pro inicializační vektor LRP.

Krok 4: Najdi správný inicializační vektor LRP pomocí testování modifikací I_0 s Hammingovou vzdáleností $0,1,\dots$

Věta 8. *Výpočetní složitost Algoritmu A je $O(2^{cl})$, kde $0 < c = H(\frac{r_{avg}}{l}) < 1$.*

Důkaz. Výpočetní složitost kroků 1 – 3 je zanedbatelná, jedná se pouze o dosazení do vzorců. Stačí tedy určit průměrný počet pokusů v kroku 4. Předpokládáme, že přesně r prvků z kroku 3 je nesprávných. Maximální počet pokusů v kroku 4 je pak

$$A(l,r) = \sum_{i=0}^r \binom{l}{i}, \quad (3.21)$$

pro tuto formuli existuje odhad využívající funkci binární entropie (van Lint, 1982)

$$A(l,r) = \sum_{i=0}^r \binom{l}{i} \leq 2^{H(\frac{r}{l})l}. \quad (3.22)$$

Pokud místo r dosadíme do (3.22) r_{avg} , dostaneme odhad počtu pokusů potřebných v kroku 4. Z čehož plyne, že výpočetní složitost Algoritmu A je $O(2^{cl})$, kde $c = H(\frac{r_{avg}}{l})$ a $0 < c < 1$. □

Za příznivých podmínek je $c \ll 1$, což znamená, že útok je mnohem rychlejší než prohledávání všech možností.

Příklad. Nechť $p = 0,75$, $t = 2$ a poměr známého proudu klíče ku délce LFSR $d = \frac{N}{t} = 100$. Pak $c = 0,012$ (viz. Tabulka 3.1). Tedy výpočetní složitost pro tento případ je $O(2^{0,012l})$, což je mnohem rychlejší v porovnání s $O(2^l)$ při prohledávání všech možností. Převzato z (Meier a Staffelbach, 1989)

3.4 Algoritmus B

Na rozdíl od Algoritmu A, který je jednorůchodový, Algoritmus B funguje na iteračním principu. Využívá faktu, že podmíněná pravděpodobnost $p^*(p,m,h)$ je

| p | t | | | | | | | | |
|------|-------|-------|-------|-------|-------|-------|-------|-------|----------|
| | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | ∞ |
| 0,51 | 1,000 | 1,000 | 1,000 | 1,000 | 1,000 | 1,000 | 1,000 | 1,000 | 1,000 |
| 0,53 | 0,994 | 0,997 | 0,997 | 0,997 | 0,997 | 0,997 | 0,997 | 0,997 | 0,997 |
| 0,55 | 0,973 | 0,993 | 0,993 | 0,993 | 0,993 | 0,993 | 0,993 | 0,993 | 0,993 |
| 0,57 | 0,927 | 0,985 | 0,986 | 0,986 | 0,986 | 0,986 | 0,986 | 0,986 | 0,986 |
| 0,59 | 0,846 | 0,973 | 0,976 | 0,976 | 0,977 | 0,977 | 0,977 | 0,977 | 0,977 |
| 0,61 | 0,729 | 0,956 | 0,964 | 0,965 | 0,965 | 0,965 | 0,965 | 0,965 | 0,965 |
| 0,63 | 0,584 | 0,930 | 0,949 | 0,951 | 0,951 | 0,951 | 0,951 | 0,951 | 0,951 |
| 0,65 | 0,432 | 0,890 | 0,930 | 0,934 | 0,934 | 0,934 | 0,934 | 0,934 | 0,934 |
| 0,67 | 0,293 | 0,832 | 0,905 | 0,914 | 0,915 | 0,915 | 0,915 | 0,915 | 0,915 |
| 0,69 | 0,122 | 0,750 | 0,871 | 0,890 | 0,893 | 0,893 | 0,893 | 0,893 | 0,893 |
| 0,71 | 0,062 | 0,641 | 0,825 | 0,860 | 0,867 | 0,868 | 0,869 | 0,869 | 0,869 |
| 0,73 | 0,028 | 0,462 | 0,761 | 0,822 | 0,837 | 0,840 | 0,841 | 0,841 | 0,841 |
| 0,75 | 0,012 | 0,314 | 0,671 | 0,772 | 0,800 | 0,808 | 0,810 | 0,811 | 0,811 |

Tabulka 3.1: Koefficient výpočetní složitosti Algoritmu A $c(p,t,\frac{N}{t})$ pro $\frac{N}{t} = 100$. Převzato z (Meier a Staffelbach, 1989).

malá, pokud prvek z_n splňuje pouze několik rovnic. Původní idea tohoto algoritmu byla, že vhodně zvolíme mez počtu splněných rovnic a všechny prvky posloupnosti z pod touto mezí invertujeme (jelikož pracujeme nad \mathbb{F}_2 stačí přičíst 1). Za příznivých podmínek pak po této úpravě bude počet prvků, které se ze špatných změnilo na správné, větší než těch, co se změnilo ze správných na špatné. A tedy po několika iteracích tohoto procesu budeme schopni rekonstruovat původní posloupnost u . Z praxe se však ukázalo vhodné používání alternativní metody, kdy nejdříve několikrát iterativně přepočítáme pravděpodobnosti $p^*(p,m,h)$ pro každý prvek posloupnosti z a invertujeme, až když máme dostatečný počet prvků s $p^*(p,m,h)$ pod zvolenou mezí. Tento proces se několikrát restartuje, dokud nedostaneme správnou původní posloupnost.

Pro přehlednost výpočtů si stejně jako v předcházejícím algoritmu označíme jednotlivé pravděpodobnosti, které jsou použity v samotném algoritmu. Označme jev $G := |\{i | L_i = 0\}| \leq h$, tedy že je splněno nejvýše h rovnic. Pravděpodobnost, že nejvýše h z m vztahů bude splněno, pak bude $U(p,m,h)$,

$$U(p,m,h) = P(G) = \sum_{i=0}^h \binom{m}{i} p s^i(p,t) (1-s(p,t))^{m-i} + (1-p) (1-s(p,t))^i s^{m-i}(p,t). \quad (3.23)$$

Dále označme pravděpodobnost, že $z_n = u_n$ a nejvýše h z m vztahů je splněno jako $V(p,m,h)$. Obdobně můžeme odvodit, že

$$V(p,m,h) = P(z_n = u_n \wedge G) = \sum_{i=0}^h \binom{m}{i} p s^i(p,t) (1-s(p,t))^{m-i}, \quad (3.24)$$

nakonec i pravděpodobnost, že $z_n \neq u_n$ a maximálně h z m vztahů je splněno,

označíme jako $W(p,m,h)$,

$$W(p,m,h) = P(z_n \neq u_n \wedge G) = \sum_{i=0}^h \binom{m}{i} (1-p)(1-s(p,t))^i s^{m-i}(p,t). \quad (3.25)$$

Tedy $U(p,m,h) \cdot N$ je očekávaný počet prvků z , které splňují nanejvýš h vztahů. Pokud jsou tyto prvky invertovány, tak očekávaný počet správně upravených prvků je $W(p,m,h) \cdot N$ a $V(p,m,h) \cdot N$ je počet špatně změněných prvků. Tedy zvýšení správných prvků je rozdíl

$$W(p,m,h) \cdot N - V(p,m,h) \cdot N.$$

A relativní zvýšení správných prvků je

$$I(p,m,h) = W(p,m,h) - V(p,m,h). \quad (3.26)$$

Optimální korekce tedy nastane, když vybereme $h = h_{max}$ takové, že $I(p,m,h)$ je maximální pro dané p a m . Jelikož budeme v algoritmu rozhodovat v závislosti na hodnotách $p^*(p,m,h)$, použijeme mez pro $p^*(p,m,h)$, která byla v původním článku experimentálně nalezena optimální pro

$$p_{mez} = \frac{1}{2} [(p^*(p,m,h_{max}) + p^*(p,m,h_{max} + 1))] \quad (3.27)$$

tak, aby byl korekční efekt co největší. Poté očekávaný počet prvků s $p^*(p,m,h)$ pod p_{mez} je

$$N_{mez} = U(p,m,h) \cdot N. \quad (3.28)$$

Pokud je pouze několik prvků s $p^*(p,m,h)$ pod p_{mez} , tak zopakujeme přidělení nových pravděpodobností $p^*(p,m,h)$. Pro iterování pravděpodobností $p^*(p,m,h)$ potřebujeme zobecnění vzorce z Tvzení (6) pro situaci, kdy každý z t prvků může mít různou pravděpodobnost p_1, p_2, \dots, p_t .

$$\begin{aligned} s(p_1, p_2, \dots, p_t, t) &= p_t \cdot s(p_1, \dots, p_{t-1}, t-1) + (1-p_t)(1-s(p_1, \dots, p_{t-1}, t-1)) \\ s(p_1, 1) &= p_1. \end{aligned} \quad (3.29)$$

Toto zobecnění platí pro všechny další formule, především pro (3.12), která vyjadřuje $p^*(p,m,h)$.

Samotné iterování pravděpodobnosti $p^*(p,m,h)$ pak funguje následovně. Nejdříve vypočítáme počáteční pravděpodobnost $p^*(p,m,h)$ v závislosti na původní pravděpodobnosti p dle (3.12) a (3.29) $p^*(p_1, p_2, \dots, p_{t-1}, m, h)$, kde $p_1 = p_2 = \dots = p_{t-1} = p$. Každé další $p^*(p_1, p_2, \dots, p_{t-1}, m, h)$ se již počítá v závislosti na pravděpodobnostech pro jednotlivé prvky, kde p_i jsou pravděpodobnosti $p^*(p_1, p_2, \dots, p_{t-1}, m, h)$ pro odpovídající prvky v minulém kroku.

Samotný algoritmus pak vypadá takto

Algoritmus 3: Algoritmus B

Krok 1: Vypočítej odhad m pomocí rovnice z (3.3).

Krok 2: Najdi hodnotu $h = h_{max}$, pro kterou je $I(p, m, h)$ maximální, a spočítej p_{mez} a N_{mez} pomocí formulí (3.27) a (3.28).

Krok 3: Nastav čítač $i = 0$.

Krok 4: Pro každý prvek posloupnosti z spočítej novou pravděpodobnost p^* pomocí formulí (3.12) a (3.29), s ohledem na jednotlivé počty vztahů, které jsou splněny. Vypočítej počet N_w prvků s $p^*(p_1, p_2, \dots, p_{t-1}, m, h) < p_{mez}$.

Krok 5: Pokud $N_w < N_{mez}$ nebo $i < \alpha$, zvyš hodnotu i o jedna a přejdi na krok 4.

Krok 6: Invertuj ty prvky posloupnosti z , které mají

$p^*(p_1, p_2, \dots, p_{t-1}, m, h) < p_{mez}$, a resetuj pravděpodobnosti pro každý prvek na originální hodnotu p .

Krok 7: Pokud zde jsou prvky, které nesplňují (3.1), přejdi do kroku 3.

Krok 8: Ukonči s $u = z$.

Iterace kroků 4 – 5 nazýváme vnitřní smyčka a kroků 3 – 7 smyčka vnější. Protože, jak mnoho experimentů ukázalo, po několika opakování vnitřních smyček ztrácí tento proces účinnost, byl zaveden koeficient α , který se většinou nastavuje jako $\alpha = 5$.

K ohodnocení korekčního efektu Algoritmu B potřebujeme vypočítat $I_{max} = I(p, m, h_{max})$, pro dané p, t, N a l . Ze znalosti vzorců pro výpočet m a h_{max} však víme, že můžeme I_{max} vyjádřit jako $I_{max} = I(p, t, d)$ a očekávaný počet prvků opravených v jedné vnitřní smyčce je pak

$$N_0 = I_{max}(p, t, d) \cdot N. \quad (3.30)$$

Pro přehlednost je lepší N_0 vyjádřit jako $N_0 = F(p, t, d) \cdot l$, kde

$$F(p, t, d) = I_{max}(p, t, d) \cdot d \quad (3.31)$$

je opravný faktor nezávislý na l .

Jak ukázalo mnoho pokusů, Algoritmus B bude s velkou pravděpodobností úspěšný, pokud $F(p, t, d) \geq 0,5$. Naopak pokud $F(p, t, d) \leq 0$, tak nenastane žádný opravný efekt a útok zklame. V tabulce (3.2) je pro ilustraci ukázáno, jak velká musí být korelační pravděpodobnost p , aby $F(p, t, d) \geq 0,5$ a aby s největší pravděpodobností útok uspěl. Hodnoty jsou vyjádřeny pro dané váhy t charakteristického polynomu a poměry d délky známého proudu klíče ku délce LFSR.

Výpočetní složitost Algoritmu B roste lineárně v závislosti na délce LFSR l tedy $O(l)$. (Meier a Staffelbach, 1989).

3.5 Shrnutí

Algoritmy prezentované v této sekci se v praxi ukázaly velmi efektivní, což vedlo k zavedení restrikcí pro konstrukci kombinačních generátorů. Neměla by existovat žádná netriviální korelace mezi proudem klíče a LFSR s délkou $l < 100$ a také s LFSR, které má charakteristický polynom malé váhy $t < 10$. Pro Booleovské funkce by zpravidla také neměla existovat žádná netriviální korelace mezi proudem klíče a malou skupinou LFSR. Bohužel, jak ukázal T. Sigenthaier, zvyšování korelační odolnosti Booleovské funkce způsobuje snižování jejího

| d | t | | | | | | | | |
|------------------|----------|----------|----------|----------|-----------|-----------|-----------|-----------|-----------|
| | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 |
| 10 | 0,761 | 0,880 | 0,980 | 0,980 | 0,980 | 0,980 | 0,980 | 0,980 | 0,980 |
| 10 ² | 0,595 | 0,754 | 0,824 | 0,863 | 0,889 | 0,905 | 0,917 | 0,926 | 0,934 |
| 10 ³ | 0,553 | 0,708 | 0,787 | 0,832 | 0,861 | 0,882 | 0,897 | 0,908 | 0,918 |
| 10 ⁴ | 0,533 | 0,679 | 0,763 | 0,812 | 0,844 | 0,867 | 0,883 | 0,896 | 0,906 |
| 10 ⁵ | 0,525 | 0,663 | 0,748 | 0,800 | 0,833 | 0,857 | 0,875 | 0,889 | 0,900 |
| 10 ⁶ | 0,519 | 0,650 | 0,737 | 0,789 | 0,825 | 0,849 | 0,868 | 0,883 | 0,894 |
| 10 ⁷ | 0,515 | 0,641 | 0,727 | 0,781 | 0,817 | 0,843 | 0,862 | 0,877 | 0,890 |
| 10 ⁸ | 0,514 | 0,634 | 0,720 | 0,774 | 0,812 | 0,838 | 0,858 | 0,874 | 0,886 |
| 10 ⁹ | 0,512 | 0,628 | 0,714 | 0,770 | 0,807 | 0,833 | 0,854 | 0,870 | 0,882 |
| 10 ¹⁰ | 0,510 | 0,621 | 0,709 | 0,764 | 0,802 | 0,830 | 0,850 | 0,866 | 0,879 |

Tabulka 3.2: Pravděpodobnosti p pro $F(p,t,d) = 0,5$. Převzato z (Meier a Staffelbach, 1989).

algebraického stupně (Siegenthaler, 1985), což vede k náchylnosti na jiné útoky, například pomocí Berlekamp-Massey algoritmu (Massey, 1969). Proto se musí při návrhu Booleovské funkce v tomto ohledu dospět ke kompromisu.

4. Modifikace rychlých korelačních útoků

V této kapitole se budeme zabývat modifikacemi rychlých korelačních útoků, které byly prezentovány jako reakce na původní rychlé korelační útoky prezentované v (Meier a Staffelbach, 1989). Budou představeny základní principy a myšlenky těchto metod. Zaměříme se především na metody hledání více rovnic z charakteristického polynomu a hledání rovnic s malou vahou. Také představíme alternativní reprezentaci korelačních útoků.

4.1 Vylepšené hledání rovnic

Jedna z metod hledání rovnic s malou vahou byla představena Chepyzhovem a Smeetssem (Chepyzhov a Smeets, 1991). Metoda je založena na známém problému z teorie kódu. Z kapitoly 2 víme, že výstupní sekvence u LFSR může být interpretována jako kódové slovo lineárního kódu C . Pro takový kód však existuje duální kód H , pro který platí $H = \{h \in \mathbb{F}_2^n \mid \sum_{i=0}^{n-1} h_i c_i = 0 \forall c \in C\}$. V (Chepyzhov a Smeets, 1991) je ukázáno, že hledání rovnic s malou vahou je ekvivalentní hledání kódového slova kódu H s malou vahou.

V (Penzhorn, 1996) Penzhorn představil metodu hledání rovnic s malou vahou pomocí dělení polynomů. Máme $g(x) = 1 + c_1x + \dots + c_lx^l$ charakteristický polynom LFSR a hledáme polynomy $h(x)$ takové, že $h(x) \equiv 0 \pmod{(g(x))}$. Penzhorn ukázal, že hledání dostatečně mnoha rovnic s vahou 3 je velmi složité až neproveditelné, proto musíme hledat rovnice s vahou 4. V (Penzhorn, 1996) je složitost algoritmu hledání rovnic váhy 4 dokázána řádu $O(2^{\frac{2l}{3}})$.

4.2 Modifikace interpretací

V článku Thomase Johanssona a Fredrika Jönssona Fast Correlation Attacks through Reconstruction of Linear Polynomials (Johansson a Jönsson, 2000) bylo ukázáno efektivní použití reprezentace pomocí polynomů více proměnných a následně představeny dva vylepšené dekódovací algoritmy využívající tuto reprezentaci. Reprezentace pomocí polynomů více proměnných vypadá takto. Mějme inicializační stav $LFSR$ délky l

$$u = (u_0, u_1, \dots, u_{l-1}), \quad (4.1)$$

tedy jak jsme již ukázali dříve, dají se další prvky posloupnosti vyjádřit v závislosti na těchto l znacích.

$$u_i = \sum_{j=0}^{l-1} w_{ij} u_j, \quad (4.2)$$

kde w_{ij} jsou konstanty vypočítané pomocí charakteristického polynomu. Pak definujeme polynom inicializačního stavu $U(x)$

$$U(x) = U(x_0, x_1, \dots, x_{l-1}) = u_0x_0 + u_1x_1 + \dots + u_{l-1}x_{l-1}. \quad (4.3)$$

Takto můžeme vyjádřit každé u_i jako výsledek polynomu inicializačního stavu v daném známém bodu $x^{(i)} = (w_{i1}, w_{i2}, \dots, w_{ij})$, tedy

$$u_i = U(x^{(i)}). \quad (4.4)$$

Tato reprezentace interpretuje korelační útok jako dekódovací problém. Tedy jak již bylo ukázáno v kapitole (2.2), proud klíče chápeme jako sekvenci, ke kterému byla přičtena sekvence šumu $e = (e_0, e_1, \dots, e_{N-1})$, kde $e_i \in \{0, 1\}$ jsou náhodné nezávislé proměnné a $P(e_i = 0) = \frac{1}{2} + \epsilon$. Vyjádření pomocí polynomu $U(x)$ pak vypadá takto

$$z = (U(x^{(0)}) + e_0, U(x^{(1)}) + e_1, \dots, U(x^{(N-1)}) + e_{N-1}). \quad (4.5)$$

Naším úkolem je pak rekonstruovat polynom $U(x)$ ze znalosti výstupů z a známé korelační pravděpodobnosti. Tento problém se nazývá Learning Parity with Noise a je považován za těžký. Řešení zjednodušení tohoto problému, kdy si můžeme vybírat body, ve kterých známe hodnotu polynomu, je pak použito jako základ pro dekódovací algoritmus.

Johansson a Jönsson se také zabývali reprezentací pomocí konvolučních kódů (Johansson a Jönsson, 1999a). Motivací této reprezentace byly především poznatky z rychlých korelačních útoků, a to že dekódovací algoritmy jsou mnohem efektivnější, pokud máme dostatek rovnic derivovaných z charakteristického polynomu s malou vahou. Pokud použijeme reprezentaci pomocí konvolučních kódů, můžeme pak takové rovnice nalézt, i když je charakteristický polynom velké váhy. Tato metoda využívá algoritmus s pamětí a právě velká paměťová složitost je nevýhodou tohoto způsobu.

Jako řešení problému s paměťovou složitostí se však ukázalo používání reprezentace pomocí turbo kódů (Johansson a Jönsson, 1999b), které fungují na bázi použití více paralelních konvolučních kódů.

Závěr

Rychlé korelační útoky jsou stále aktuální téma, kterým se zabývá mnoho kryptologů, a přibývají nové práce snažící se tento druh útoku vylepšit. Většina těchto vylepšení však vyžaduje k porozumění již hlubší znalosti teorie kódů. V této práci jsme se zabývali především základy. Představili jsme původní průkopnickou práci na téma Korelačních útoků. Zabývali jsme se vztahem korelačních útoků k dekodovacímu problému. Rychlé korelační útoky byly dále představeny a podrobněji rozebrány. V neposlední řadě bylo ukázáno několik příkladů modifikací rychlých korelačních útoků.

Literatura

- CARLET, C. (2007). Boolean functions for cryptography and error correcting codes.
- CHEPYZHOV, V. a SMEETS, B. (1991). On a fast correlation attack on certain stream ciphers. **547**, 176–185.
- CHEPYZHOV, V., JOHANSON, T. a SMEETS, B. (2000). A simple algorithm for fast correlation attack on stream ciphers. *Springer-Verlag*, **LNCS 1978**, 181–195.
- JOHANSSON, T. a JÖNSSON, F. (1999a). Improved fast correlation attacks on stream ciphers via convolutional codes. In *EUROCRYPT*, volume 1592 of *Lecture Notes in Computer Science*, pages 347–362. Springer. ISBN 3-540-65889-0.
- JOHANSSON, T. a JÖNSSON, F. (1999b). Fast correlation attacks based on turbo code techniques. In *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 181–197. Springer. ISBN 3-540-66347-9.
- JOHANSSON, T. a JÖNSSON, F. (2000). Fast correlation attacks through reconstruction of linear polynomials. In *CRYPTO*, volume 1880 of *Lecture Notes in Computer Science*, pages 300–315. Springer. ISBN 3-540-67907-3.
- JÖNSSON, F. (2002). *Some results on fast correlation attacks*. PhD thesis, Lund University.
- MASSEY, J. L. (1969). Shift-register synthesis and bch decoding. *IEEE Transactions on Information Theory*, **15**, 122–127.
- MEIER, W. a STAFFELBACH, O. (1989). Fast correlation attacks on certain stream ciphers. *Journal of Cryptology*, **1**(3), 159–176.
- MEIER, W. (2011). Fast correlation attacks: Methods and countermeasures. pages 55–67.
- PENZHORN, W. T. (1996). Correlation attacks on stream ciphers: Computing low-weight parity checks based on error-correcting codes. In *Fast Software Encryption*, pages 159–172.
- SIEGENTHALER, T. (1985). Decrypting a class of stream ciphers using ciphertext only. *IEEE Transactions of Information Theory*, **C-34**(1), 81–85.
- STANOVSKÝ, D. (2010). *Základy Algebry*. Matfyzpress, Praha. ISBN 9788073781057.
- VAN LINT, J. (1982). Introduction to coding theory. *Springer-Verlag*, **86**.
- ZVÁRA, K. a ŠTĚPÁN J. (2002). *Pravděpodobnost a matematická statistika*. Matfyzpress, Praha. ISBN 80-85863-24-3.

Seznam obrázků

| | | |
|-----|---|---|
| 1.1 | Šifrovací a dešifrovací schéma proudové šifry. | 5 |
| 1.2 | Šifrovací a dešifrovací schéma kombinačního generátoru. | 6 |

Seznam tabulek

| | | |
|-----|---|----|
| 3.1 | Koeficient výpočetní složitosti Algoritmu A $c(p, t, \frac{N}{l})$ pro $\frac{N}{l} = 100$. Převzato z (Meier a Staffelbach, 1989). | 20 |
| 3.2 | Pravděpodobnosti p pro $F(p, t, d) = 0,5$. Převzato z (Meier a Staffelbach, 1989). | 23 |