МАТЕМАТИЧКИ ФАКУЛТЕТ
УНИВЕРЗИТЕТ У БЕОГРАДУ

МАСТЕР РАД

Корелациони напади на проточне шифре

Студент: **Душан Ристовски**

Број индекса: **1154/2013**

Ментор: **Миодраг Живковић**

Београд, 2017.

Универзитет у Београду – Математички факултет

Мастер рад

Аутор: Душан Ристовски

Наслов: Корелациони напади на проточне шифре

Ментор: Миодраг Живковић

Чланови комисије: Предраг Јаничић, Младен Николић

Садржај

1 Увод

Овај рад се бави проучавањем проточних шифара заснованих на померачком регистру са линеарном повратном спрегом (ПРЛПС) и њихове подложности корелационим нападима. Корелациони напади су класа познатих напада са познатим отвореним текстом за разбијање проточних шифара чији се низ кључа генерише комбиновањем излаза неколико ПРЛПС користећи Булове функције. Корелациони напади користе статистичку слабост која произилази из сиромашног избора Булових функција и ово је једна од ситуација када је могућ корелациони напад. Ипак, могуће је изабрати функцију која у некој мери онемогућује корелационе нападе, тако да ова врста шифара није инхерентно несигурна. Приликом дизајнирања проточних шифара, непоходно је узети у обзир рањивост проточних шифара на корелационе нападе.

У поглављу 2 дефинишу се основни појмови везани за област криптографије. Наводе се основни појмови, криптосистеми и неке од техника криптоанализе које се користе у самом раду. Поред тога, описују се проточне шифаре, генератори случајних бројева, комбинациони и филтерски генератора. У поглављу 3 описују се сами корелациони напади и примена. У поглављу 4 описују се једна посебна врста корелационих напада, тзв. брзи корелациони напади. У поглављу 5 приказује се програмска реализација корелационих напда и резултати напада на неке проточне шифре.

2 Основни појмови

У овом поглављу дефинишу се основни појмови везани за област криптографије. Такође, ово поглавље описује неке системе шифровања као што су симетрични системи шифровања, односно једну такву потклasu симетричних шифарских система звану проточне шифре. Честа компонента генератора низа кључа код проточних шифра, јесу померачки регистри са линеарном повратном спрегом (ПРЛПС). Ово поглавље детаљно се бави карактеристикама таквих ПРЛПС.

2.1 Увод у криптографију

Криптографија (енгл. *cryptography*) је наука која се бави методама очувања тајности информација. Назив потиче од грчке речи *kriptos*, што значи скривен. Она омогућава да субјекат А, безбедно пошаље трансформисану поруку ка субјекту Б, тако да субјекат Ц не дође до садржаја послате поруке на основу трансформисане поруке.

Порука која се шаље зове се *отворени текст* (енгл. *plaintext*). *Шифровање* (енгл. *encryption*) је процедура која трансформише отворени текст у шифроване податке. Шифрована порука назива се шифрат (енгл. *ciphertext*). Обрнути процес, *дешифровање* (енгл. *decryption*), реконструише отворени текст помоћу шифрата. *Шифарски систем* (енгл. *cryptosystem*) је пар чији су елементи алгоритам шифровања и алгоритам дешифровања. Ови алгоритми скоро увек зависе од једног посебног параметра, који се зове *кључ* (енгл. *key*). Кључ је параметар којим се бира конкретна шифарска трансформација у оквиру изабраног система. *Симетрични систем* подразумева употребу истог тајног кључа за шифровање и дешифровање. *Асиметрични систем* (системи са јавним кључем) подразумева да субјекат публикује своје кључеве за шифровање, а да у највећој тајности чува своје кључеве за дешифровање. *Криптоанализа* (енгл. *cryptanalysis*) је наука која се бави разбијањем шифара, односно откривањем отвореног текста на основу шифрата, без познавања кључа. Различите технике криптоанализе називају се напади. Успешна криптоанализа назива се декриптирање.

Криптоанализа за циљ има проналажење рањивости система који користи одређени шифарски систем. Криптоанализа се предузима од стране самих дизајнера система ради провере сигурности система или злонамерних нападача који хоће да сруше систем или да дођу до одређених информација.

Криптографија има за циљ:

1. Очување интегритета информација које се шифрују
2. Чување тајности информација тиме што је садржај доступан само особама које знају кључ
3. Проверу идентитета, корисници који комуницирају представљају се један другом на поуздан начин

Формално се процес шифровања може представити на следећи начин. Нека је P – отворени текст, C – шифрат, E – шифровање, D – дешифровање. Тада се процес шифровања опсује са

$$E(P) = C$$

а процес дешифровања

$$D(C) = P.$$

Пошто шифровање и дешифровање за циљ има преношење оригиналне информације треба да важи

$$D(E(P)) = P.$$

Криптологија (енгл. *Cryptology)* је област математике која обухвата криптографију и криптоанализу.

2.2 Коначна поља

Поље је алгебарска структура у којој се могу вршити операције сабирања, одузимања, множења и дељења (осим нулом) и важе иста правила која су позната из стандардне аритметике. За сваки прост број p, коначно поље остатака целих бројева по модулу p означава се са Z/pZ, F_p или $GF(p)$. Ако је p прост број, $F_p = Z/pZ$ је поље са p елемената $\{0,1, \dots, p-1\}$ са операцијама сабирања, одузимања и множења. Запажа се да за све елементе $\alpha \neq 0$ важи $NZD(\alpha, p) = 1$, па се може одредити α^{-1}, па се због тога може делити било којим ненула елементом. Овде $NZD(a, b)$означава највећи заједнички делилац бројева a и b. Највећи заједнички делилац два цела броја различита од нуле је највећи позитиван цео број који дели оба броја без остатка.

У теорији бројева, Ојлерова функција $\boldsymbol{\varphi(n)}$, за позитивне целе бројеве n, је дефинисана као број позитивних целих бројева мањих или једнаких n, који су узајамно прости са n. У скупу $F_p^* = \{1, \dots, p-1\}$ могу се користити операције множења и дељења. Група F_p^* је циклична, то јест садржи бар један елемент g такав да је $\{1, g, g^2, g^3, \dots, g^{p-1}\} = F_p^*$. Прецизније, група F_p^* има $\varphi(p-1)$ генератора. Скупови $\{1, g, g^2, g^3, \dots, g^{p-1}\}$ и $\{1, \dots, p-1\}$ су једнаки.

Лако се доказује да ако је g генератор групе F_p^*, тада је g^k генератор ове групе ако и само ако је $NZD(k, p-1) = 1$.

У раду се користи и друга врста коначних поља. Нека је $F_2[x]$ скуп полинома са коефицијентима из $F_2 = Z/2Z = \{0,1\}$. Запажа се да је $-1 = 1$, па је одузимање исто што и сабирање. Полиноми из овог скупа су

$$0,1, x, x+1, x^2, x^2+1, x^2+x, x^2+x+1 \dots$$

Број полинома степена мањег или једнаког n је 2^{n+1}. То су полиноми $a_n x^n + \dots + a_1 x + a_0$, где су a_i елементи који припадају скупу $\{0,1\}$. Полиноми се множе на уобичајени начин

при чему се коефицијенти рачунају у F_2. Неки полином је несводљив над пољем ако се не може раставити у производ полинома нижег степена са коефицијентима из тог поља. На пример, над пољем F_2 полином $x^2 + x + 1$ је несводљив.

Посматрају се сада полиноми из $F_2[x]$ и њихови остаци по модулу несводљивог полинома $x^3 + x + 1$. Добијају се полиноми мањег степена и важи $x^3 + x + 1 \equiv 0$, то јест $x^3 \equiv x + 1$. Према томе, на скупу $F_2[x]/(x^3 + x + 1) = \{0,1, x, x + 1, x^2, x^2 + 1, x^2 + x, x^2 + x + 1\}$ дефинисане су уобичајне операције сабирања, одузимања, множења, при чему је $x^3 \equiv x + 1$. Уопште, ако је $p(x)$ несводљив полином степена d, онда је $F_2[\mathrm{x}]/(\mathrm{p(x)})$ поље, које се означава са F_{2^d} и има 2^d елемената.

Булова функција од n променљивих дефинисана је као $f: F_2^n \to F_2$. Скуп свих Булових функција од n променљивих обележава се са B_n. Скуп B_n са операцијом сабирања је поље над F_2 са 2^n елемената. Било која Булова функција f, може се представити као полином од n променљивих над пољем F_2

$$f(x_1, \dots, x_n) = a_0 + \sum_{1 \le i \le n} a_i x_i + \sum_{1 \le i < j \le n} a_{i,j} x_i x_j + \cdots + a_{1,2,\dots,n} x_1 x_2 \dots x_n,$$

где су коефицијенти $a_0, a_i, a_{i,j}, \dots, a_{1,2,\dots,n}$ из скупа $\{0,1\}$. Овако представљена функција f, зове се алгебарска нормална форма (АНФ) функције f. Алгебарски степен функције f, $\deg(f)$, је степен највећег члана функције f која је у АНФ.

Нека су $X_0, \dots, X_{n-1}$ бинарне случајне променљиве. Нека је f Булова функција дефинисана са $f: F_2^n \to F_2$. Булова функција f је корелационо имуна k-тог реда, ако је случајна променљива $Z = f(X_0, \dots, X_{n-1})$ независна од било ког вектора $(X_{i_1}, \dots, X_{i_k}), 0 \le i_1 < \cdots < i_k < n$. У [XM88] може се видети да случајна променљива Z не зависи од m бинарних случајних променљивих ако и само ако Z не зависи од збира $\lambda_1 X_1 + \cdots + \lambda_m X_m$ за било које $\lambda = (\lambda_1, \dots, \lambda_m)$.

2.3 Проточне шифре

Проточна шифра (енгл. *Stream cipher)* трансформише отворени текст симбол по симбол, односно најчешће бит по бит. У пракси, обично се користе шифровање бит по бит у комбинацији са операцијом ексклузивно ИЛИ (XOR). Проточне шифре су важна класа алгоритама за шифровање.
Проточна шифра је симетрична шифра која рукује трансформацијама на појединачним симболима, или најчешће битовима, отвореног текста. Прецизније, низ битова отвореног текста p_0p_1 ... трансформише се у низ c_0c_1 ..., при том се користи псеудослучајни низ s_0s_1 ..., тзв. низ кључа који се добија из коначног аутомата чије се почетно стање добија уз помоћ тајног кључа.

Постоји огромно теоријско знање о проточним шифрама. Међутим, постоји мали број тотално специфицираних проточних алгоритама у отвореној литератури. Ово се може објаснити чињеницом да већина проточних шифара имају тенденцију да буду поверљиве. За разлику од проточних шифара, бројни блоковски шифарски системи су објављени, стандардизовани и стављени у јавну употребу. Ипак, због својих предности, проточне шифре су данас у широкој употреби.

2.4 Генератор случајних бројева

Прави генератор случајних бројева карактерише чињеница да излаз не може да се репродукује два пута на исти начин. Ако бацимо новчић 100 пута и забележимо секвенцу резултата, фактички немогуће је за било кога на Земљи да репродукује исту секвенцу од 100 битова. Вероватноћа успеха је $\frac{1}{2}^{100}$, што је веома мало. Прави генератори случајних бројева су базирани на физичким процесима, као што су на пример бацање новчића, бацање коцкица, полупроводници са шумом.

Псеудослучајни генератори случајних бројева генеришу секвенцу чији се чланови рачунају полазећи од неког почетног стања, које се обично означава са $seed$. Често се рачунају рекурзивно на следећи начин

$$S_0 = seed$$

$$S_{i+1} = f(S_i), i = 0,1, \dots$$

Генерализација овог генератора је $S_{i+1} = f(S_i, S_{i-1}, \dots, S_{i-t})$, где је t фиксиран цео број. Популаран пример је линеарани конгруентни генератор

$$S_0 = seed$$

$$S_{i+1} = aS_i + b \bmod m, i = 0,1, \dots$$

Псеудослучајни генератори нису случајни у буквалном смислу, јер је њихов излаз предвидљив. Један пример је функција *rand()* у програмском језику C.

2.5 Особине померачких регистара

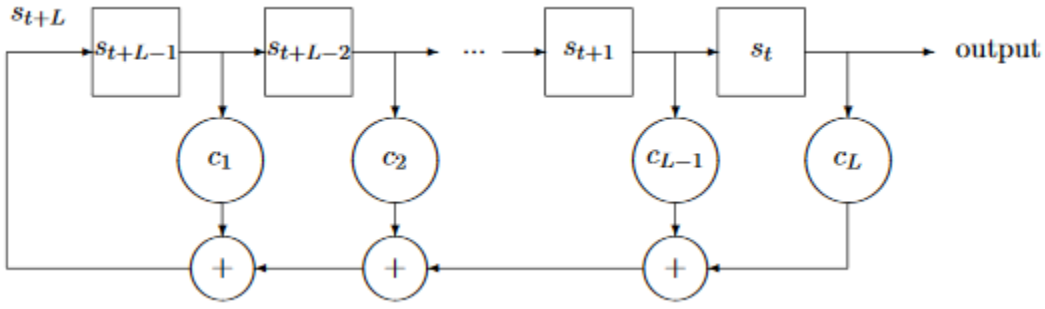
Померачки регистар са линеарном повратном спрегом (ПРЛПС, *енгл. linear feedback shift register, LFSR*), су основне компоненте за многе генераторе низа кључа који се користе за апликације базиране на проточним шифрама, јер су прикладни за хардверску имплементацију и производе низове битова са добрим статистичким својствима. Један од основних разлога је зато што имају дуг период, нарочито када је повратни полином примитиван, период ПРЛПС са n ћелија је $2^n - 1$. Померачки регистар са линеарном повратном спрегом се често користи као основа за псеудослучајне генераторе. Добро је користити ПРЛПС као псеудослучајну основу у видео играма. Међутим, није баш пожељно користити ПРЛПС као генератор за низове кључа. Заиста, посматрањем $2n$ излазних битова, нападач може комплетно да реконструише стање и повратни полином, користећи линеарну алгебру или специфичан Берлекамп-Месијев (Berlekamp-Massey) алгоритам. Међутим, својства дугог периода заједно са добрим статистичким својствима излазне секвенце су веома пожељни и веома је примамљиво користити ПРЛПС као основну компоненту за генераторе низ кључа.

ПРЛПС дужине L над F_q је коначни аутомат који производи полу-бесконачни низ елемената из F_q, $s = (s_t)_{t\geq 0} = s_0 s_1 \ldots$, који задовољава линеарну повратну релацију реда L над пољем F_q

$$S_{t+L} = \sum_{i=1}^{L} c_i S_{t+L-i}, t \geq 0.$$

Сви коефицијенти $c_1, \ldots, c_L$ су елементи из поља F_q, и називају се повратни коефицијенти ПРЛПС.

ПРЛПС дужине L над F_q може се представити моделом на следећој слици:



Слика 2.1 Померачки регистар са линеарном повратном спрегом

Регистар се састоји од L ћелија и свака садржи један елемент из F_q. Садржај L ћелија $s_t, \ldots, s_{t+L-1}$ формира стање ПРЛПС. У почетку су L ћелија попуњене са L елемената $s_0, \ldots, s_{L-1}$, који се могу произвољно изабрати из F_q и образују почетно стање регистра. Померачки регистар је контролисан екстерним сатом. У сваком кораку, садржај сваке ћелије

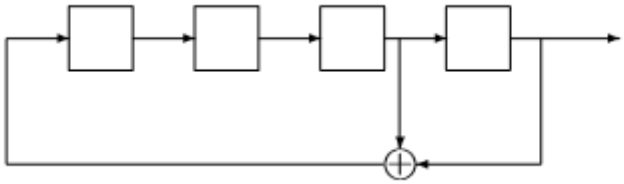
се помера за једну ћелију удесно. Садржај најдеснијe ћелије s_t је излаз а садржај најлевије ћелије постаје повратни бит, s_{t+L}. Садржај најлевије ћелије добија се као линеарна комбинација садржаја ћелија регистара.

$$S_{t+L} = \sum_{i=1}^{L} c_i S_{t+L-i}, t \geq 0.$$

Табела 2.1 даје стања ПРЛПС дужине 4 са повратним коефицијентима $c_1 = c_2 = 0$, $c_3 = c_4 = 1$ и са почетним стањем $(s_0, s_1, s_2, s_3) = (1,0,1,1)$ над F_2. Овај ПРЛПС је представљен на слици 2.2 и одговара линеарној рекурентној релацији

$$s_{t+4} = s_{t+1} + s_t \bmod 2.$$

Излазни низ $s_0 s_1$ ..., генерисан овим ПРЛПС је 1011100...



Слика 2.2 Бинарни ПРЛПС са повратним коефицијентима $(c_0, c_1, c_2, c_3) = (0,0,1,1)$

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s_t	1	0	1	1	1	1	0	0	0	1	0	0	1	1	0	1
s_{t+1}	0	1	1	1	1	0	0	0	1	0	0	1	1	0	1	0
s_{t+2}	1	1	1	1	0	0	0	1	0	0	1	1	0	1	0	1
s_{t+3}	1	1	1	0	0	0	1	0	0	1	1	0	1	0	1	1

Табела 2.1 Стања ПРЛПС са повратним коефицијентима $(c_0, c_1, c_2, c_3) = (0,0,1,1)$ и почетним стањем $(s_0, s_1, s_2, s_3) = (1,0,1,1)$.

2.5.1 Повратни и карактеристични полином

Излазни низ из ПРЛПС је јединствено одређен повратним коефицијентима и почетним стањем. Повратни коефицијенти $c_1, \dots, c_L$ ПРЛПС дужине L се обично представљају преко повратног полинома дефинисаног на следећи начин:

$$P(x) = 1 - \sum_{i=1}^{L} c_i x^i.$$

Карактеристичан полином се добија као реципрочан полином повратног полинома

$$P^*(x) = x^L P\left(\frac{1}{x}\right) = x^L - \sum_{i=1}^{L} c_i x^{L-i}.$$

За ПРЛПС се каже да је несингуларан ако је степен његовог повратног полинома једнак дужини ПРЛПС. Било који низ генерисан несингуларним ПРЛПС дужине L је периодичан, и тај период не прелази $q^L - 1$.

За дати ПРЛПС дужине L над F_q може се генерисати q^L различитих низова који одговарају различитим почетним стањима и ови низови припадају векторском простору над F_q. Скуп свих низова генерисаних помоћу ПРЛПС са повратним полиномом P је карактерисан следећим својствима: низ $(s_t)_{t\geq 0}$ је генерисан са ПРЛПС дужине L над F_q са повратним полиномом P ако и само ако постоји полином Q који припада $F_q[X]$ где је степен полинома Q мањи од L, тако да генератриса низа $(s_t)_{t\geq 0}$ задовољава

$$\sum_{t\geq 0} s_t X^t = \frac{Q(X)}{P(X)}. \qquad \textbf{(2.1)}$$

Штавише, полином Q је комплетно одређен коефицијентима полинома P и почетним стањем ПРЛПС:

$$Q(X) = -\sum_{i=0}^{L-1} X^i \left(\sum_{j=0}^{i} c_{i-j} s_j \right),$$

где је $P(X) = -\sum_{i=0}^{L} c_i x^i$. Овај израз значи да постоји један на један кореспонденција између низова генерисаних помоћу ПРЛПС дужине L са повратним полиномом P и разломка $\frac{Q(X)}{P(X)}$, где је степена полинома Q мањи од L. Ово за последицу има две ствари. Прво, било који низ генерисан уз помоћ ПРЛПС са повратним полиномом P такође се може генерисати било којим ПРЛПС чији повратни полином је умножак полинома P. Ово својство се користи приликом напада на генераторе низа кључа базираних на ПРЛПС, тзв. брзи корелациони напади. Друго, низ генерисан помоћу ПРЛПС са повратним полиномом P такође се може добити помоћу краћег ПРЛПС са повратним полиномом P' ако је одговарајући разломак

$\frac{Q(X)}{P(X)}$ такав да је $NZD(P,Q) \neq 1$. Међу свим низовима генерисаним помоћу ПРЛПС са повратним полиномом P, постоји један који се може добити помоћу краћег ПРЛПС ако и само ако P није несводљив над F_q.

Коефицијент a_L, полинома P степена L, назива се најстарији коефицијент. Ако је тај најстарији коефицијент 1, каже се да је полином $P(x)$ моничан. Штавише, за било који линеарни рекурентни низ $(s_t)_{t\geq 0}$, постоји јединствен полином P_0 чији је најстарији коефицијент једнак 1, такав да функција генерисања низа $(s_t)_{t\geq 0}$ добија се од $\frac{Q_0(x)}{P_0(x)}$, где су Q_0 и P_0 узајамно прости. Нека је $\deg(P)$ степен полинома P. Најкраћи ПРЛПС који генерише $(s_t)_{t\geq 0}$ има дужину $L = \max(\deg(P_0), \deg(Q_0) + 1)$, и његов повратни полином једнак је P_0. Реципрочан полином полинома P_0, $x^L P_0\left(\frac{1}{x}\right)$, је карактеристичан полином најкраћег ПРЛПС који генерише $(s_t)_{t\geq 0}$, и назива се минимални полином низа. Минимални полином одређује линеарну рекурентну релацију најмањег реда који задовољава низ. Степен минималног полинома линеарног рекурентног низа је линеарна сложеност низа, и одговара дужини најкраћег ПРЛПС који га генерише. $L(s)$ је ознака за линеарну сложеност (линеарна сложеност детаљније се описује у поглављу 2.5.5).

2.5.2 Минимални полином

Минимални полином линеарног рекурентног низа $s = (s_t)_{t\geq 0}$ елемената из F_q је полином P најмањег степена из $F_q[X]$ такав да је $(s_t)_{t\geq 0}$ генерисан помоћу ПРЛПС са карактеристичним полиномом P. Другим речима, $P = \sum_{i=0}^{L-1} p_i X^i + X^L$ је карактеристичан полином линеарне рекурентне релације најмањег реда који задовољава низ

$$S_{t+L} + \sum_{i=1}^{L-1} p_i S_{t+i} = 0, t \geq 0.$$

Минимални полином линеарног рекурентног низа s је моничан и јединствен; дели карактеристични полином било ког ПРЛПС који генерише низ s. Минимални полином линеарног рекурентног низа који има линеарну сложеност L може се открити из било која $2L$ узастопних битова низа уз помоћ Берлекамп-Месијевог алгоритма.

Пример 2.1 Нека ПРЛПС дужине 10 има повратни полином

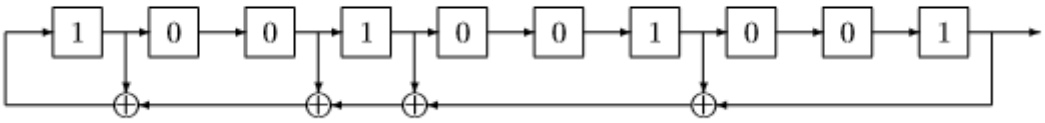
$$P(X) = 1 + X + X^3 + X^4 + X^7 + X^{10},$$

и почетно стање $s_0 s_1 \ldots s_9$ је 1001001001.
Функција која генерише низ уз помоћ датог ПРЛПС је дата са **(2.1),**

$$\sum_{t\geq 0} s_t X^t = \frac{Q(X)}{P(X)}$$

где је полином Q изведен из коефицијената полинома P и из почтеног стања:

$$Q(X) = 1 + X + X^7.$$

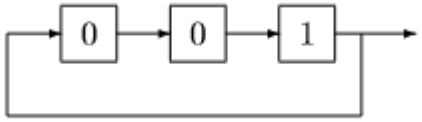


Слика 2.3 Пример ПРЛПС дужине 10

Дакле, имамо

$$\sum_{t\geq 0} s_t X^t = \frac{1+X+X^7}{1+X+X^3+X^4+X^7+X^{10}} = \frac{1}{1+X^3}$$

јер је $1+X+X^3+X^4+X^7+X^{10} = (1+X+X^7)(1+X^3)$ у пољу $F_2[X]$. Према томе, низ $(s_t)_{t\geq 0}$ може се такође генерисати помоћу ПРЛПС са повратним полиномом $P_0(X) = 1+X^3$ као на слици 2.4. Минимални полином низа је дакле $1+X^3$ и линеарна сложеност је 3.



Слика 2.4 ПРЛПС дужине три који генерише исти низ као и *ПРЛПС* на слици 2.3.

2.5.3 Период низа ПРЛПС

Минимални полином линеарног рекурентног низа игра главну улогу јер одређује линеарну сложеност и најмањи период низа. Заправо, најмањи период линеарног рекурентног низа једнак је периоду минималног полинома. Период или ред полинома P у $F_q[X]$, где је $P(0) \neq 0$, је најмањи позитиван цео број e за који $P(X)$ дели $X^e - 1$. Низ s има максималан период $q^{L(s)} - 1$ ако и само ако је минимални полином примитиван (тј. ако је период минималног полинома максималан).

На пример, низ генерисан уз помоћ ПРЛПС из примера 2.1, има период 3, зато што његов минимални полином $1+X^3$ има период 3.Тај низ игледа овако, 100100100….

У другу руку, било који не-нула низ генерисан уз помоћ ПРЛПС дужине 4 на слици 2.2 има период $2^4 - 1 = 15$. Заправо, минимални полином било којег таквог низа одговара карактеристичном полиному $P^*(X) = 1+X+X^4$ јер је P^* несводљив. Штавише, P^* је примитиван полином.

Било који низ $s = (s_t)_{t\geq 0}$ генерисан помоћу ПРЛПС дужине L који има примитиван повратни полином, има највећу могућу сложеност $L(s)$ и највећи могући период $q^{L(s)} - 1$. Такви низови се називају „Максимално дуги линеарни низови“, тј. m низови. Због претходних оптималних својстава, линеарни рекурентни низ који се користи у криптографији увек се бира да буде m низ. Штавише, поседује добра статистичка својства. Другим речима, повратни полином ПРЛПС би требало увек бити изабран тако да буде примитиван.

2.5.4 Генератор низа кључа базиран на ПРЛПС

Јасно је да не би никада требало користити ПРЛПС као генератор низа кључа ако су повратни коефицијенти ПРЛПС јавни. Цео низ се може открити познавањем L повезаних битова било ког низа кључа, где је L линеарна сложеност кључа. Ако су повратни коефицијенти тајни, цео низ кључа се може открити из било која $2L$ повезаних битова низа кључа уз помоћ Берлекамп-Месијевог алгоритма. Дакле, уобичајна техника за псеудослучајне генераторе који се могу користити за низове кључа јесте комбиновање више ПРЛПС на различите начине у циљу генерисања линеарног рекурентног низа који има велику линеарну сложеност.

2.5.5 Линеарна сложеност

Линеарна сложеност полу-бесконачног низа $s = (s_t)_{t\geq 0}$ елемената из F_q, $L(s)$, је најмањи цео број такав да се s може добити из *ПРЛПС* дужине L над F_q, ако *ПРЛПС* не постоји онда је линеарна сложеност бесконачна. Конвенционално, линеарна сложеност нула низа је 0. Линеарна сложеност линеарног рекурентног низа одговара степену минималног полинома. Линеарна сложеност $L(s^n)$ коначног низа $s^n = s_0 s_1 \dots s_{n-1}$ од n елемената из F_q је дужина најкраћег *ПРЛПС* који производи s^n као првих n излазних битова за неко почетно стање. Линеарна сложеност било ког коначног низа може се утврдити помоћу Берлекамп-Месијевог алгоритма. Једна важна тврдња по Месију [Mas69] је та да за било који коначни низ s^n дужине n, *ПРЛПС* дужине $L(s^n)$ који генерише s^n је јединствен ако и само ако је $n \geq 2L(s^n)$.

Линеарна сложеност бесконачног рекурентног низа s и линеарна сложеност коначног низа s^n конструисан од првих n битова низа s, су повезани следећим својствима: ако је s бесконачан линеаран рекурентни низ са линеарном сложеношћу L, онда коначан низ s^n има линеарну сложеност L за било које $n \geq 2L$. Штавише, јединствен *ПРЛПС* дужине L који генерише s је јединствен *ПРЛПС* дужине L који генерише s^n за било које $n \geq 2L$.

Очекивана линеарна сложеност бинарног низа $s^n = s_0 s_1 \dots s_{n-1}$ од n независних, бинарних и униформно дистрибуираних случајно променљивих је

$$E\big(L(s^n)\big) = \frac{n}{2} + \frac{\big(4 + \varepsilon(n)\big)}{18} + 2^{-n}\left(\frac{n}{3} + \frac{2}{9}\right)$$

где је $\varepsilon(n) = n \bmod 2$. Ако је s бесконачан бинарни низ са периодом 2^n који се добија понављањем низа $s_0 s_1 \dots s_{2^n-1}$ од 2^n независних, бинарних и униформно дистрибуираних случајних променљивих, тада очекивана линеарна сложеност је $E\big(L(s)\big) = 2^n - 1 + 2^{-2n}$ [Rue86]. Због обимности рада, докази тврђења који се наводе у овом поглављу, неће се представљати. Докази истих могу се пронаћи у [Mas69], [Rue86].

2.6 Алгоритам за генерисање излазног низа ПРЛПС

Пре самог алгоритма за корелациони напад, потребно је приказати алгоритам за генерисање излазног низ из ПРЛПС. Нека је

$$f = c_0 + c_1 x + c_2 x^2 + \cdots + c_{L-1} x^{L-1} + x^L,$$

карактеристичан полином једног ПРЛПС. Нека је $R_0 = (r_0, \dots, r_{L-1})$ почетно стање ПРЛПС. Излазни бит у кораку t је $r_t, t \geq l$, може се израчунати на следећи начин

$$r_т = c_0 r_{t-L} + c_1 r_{t-L+1} + c_2 r_{t-L+2} + \cdots + c_{L-1} r_{t-1} \text{ за } t \geq L.$$

Нека је w број ненула коефицијената у скупу $\{c_0, \dots, c_{L-1}\}$. Број ненула коефицијената полинома f је $w + 1$ (надаље за број ненула коефицијената користи се назив тежина). Овај параметар се користи приликом рачунања сложености алгоритма.

Алгоритам 1

Улаз: карактеристичан полином f степена L, почетно стање R_0
Излаз: низ $r = (r_t)_{0 \geq t \geq N-1}$ дужине N генерисан помоћу једног ПРЛПС са карактеристичинм полином f и почетним стањем R_0

- $L = \deg(f)$
- $r = R_0$
- **for** $t = L, \dots, N-1$ **do**

  $$r[t] = \sum_{0 \leq i \leq l-1} c_i r[t - l + i] \bmod 2$$
- **return** r

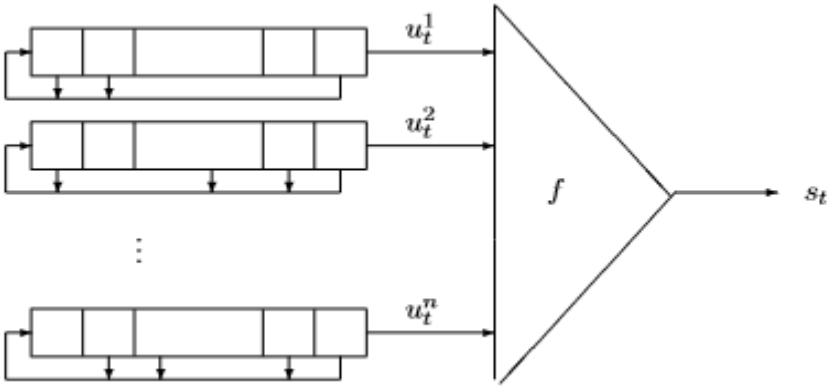
Нека је $f \in F_2[x]$ полином степена L и тежине $w + 1$. Низ дужине N добијен из ПРЛПС са карактеристичним полиномом f може се генерисати помоћу претходног алгоритма користећи $(N-L)w$ сабирања у F_2.

2.7 Комбинациони генератори

Комбинациони генератори су један од типова генератора низа кључа за проточне шифре. Комбинациони генератор је конструисан од неколико ПРЛПС чији се излаз комбинује Буловом функцијом која производи низ кључа. Излазни низ $(s_t)_{t\geq 0}$ комбинационог генератора конструисан од n ПРЛПС, $R_1, R_2, \dots R_n$, је

$$s_t = f(u_t^1, u_t^2, \dots, u_t^n), t \geq 0.$$

Низ генерисан регистром R_i, i-тог саставног *ПРЛПС* означава се са $\left(u_t^i\right)_{t\geq 0}$ а f је функција од n променљивих. Када посматрамо комбинациони генератор конструисан од n *ПРЛПС* над F_2, комбинациона функција је функција из F_{2^n} у F_2.



Слика 2.5 Комбинациони генератор од n ПРЛПС комбинованих Буловом функицијом f

Комбинациона функција f треба бити балансирана, да би њен излаз био 0, односно 1 са вероватноћом $\frac{1}{2}$. Саставни ПРЛПС треба изабрати да има примитивни повратни полином да би се осигурала добра статистичка својства излазног низа. Тајни параметари су почетна стања ПРЛПС. Стога, највише напада на комбинационе генераторе састоје се од проналажења почетног стања свих ПРЛПС познавајући неке битове низа кључа. Када су непознати повратни полиноми ПРЛПС и комбинациона функција, конструкциони напад може да реконструише комплетан опис генератора познавањем великог сегмента низа шифрата и одговарајућег отвореног текста [CF00].

Својства излазног низа

Низ добијен помоћу комбинационог генератора је линеарни рекурентни низ. Његов период и линеарна сложеност могу се извести из низова генерисаних помоћу саставних *ПРЛПС* и из алгебарске нормалне форме комбинационе функције. Заиста, ако посматрамо два линеарна рекурентна низа u и v над F_2 са линеарном сложеношћу $L(u)$ и $L(v)$ имамо следећа својства:

- Линеарна сложеност низа $u + v = (u_t + v_t)_{t\geq 0}$ задовољава $L(u+v) \leq L(u) + L(v)$ са једнакошћу ако и само ако су минимални полином низова u и v узајамно прости.

Штавише, у случају једнакости, период низа $u + v$ је најмањи заједнички садржалац периода од u и v.

- Линеарна сложеност низа $uv = (u_t v_t)_{t\geq 0}$ задовољава $L(uv) \leq L(u)\, L(v)$, где једнакост важи ако су минимални полиноми низова u и v примитивни и ако су $L(u)$ и $L(v)$ различити и већи од два [Her85], [RS87], [GN95].

Низ кључа добијен помоћу комбинационог генератора, који се састоји од n бинарних ПРЛПС са примитивним полиномима и комбинационом Буловом функцијом f, задовољава следећа својства која се могу пронаћи у [RS87]. Ако су сви ПРЛПС дужине $L_1, \dots, L_n$ различити и већи од 2 (и немају почетно стање са свим 0), онда је линеарна сложеност излазног низа s једнака

$$f(L_1, \dots, L_n)$$

где се алгебарска нормална форма рачуна у скупу целих бројева. На пример, ако су дата четири ПРЛПС дужине $L_1, \dots, L_4$ која задовољавају претходне услове, са комбинационом функцијом $x_1x_2 + x_2x_3 + x_4$, линеарна сложеност излазног низа је $L_1L_2 + L_2L_3 + L_4$. Слични резлтати над пољем F_q могу се пронаћи у [RS87] и [Bry95]. Велика линеарна сложеност је пожељно својство за низове кључа како оно осигурава да Берлекамп-Месијев алгоритам постаје неупотребљив. Стога, комбинациона функција f требало би имати велики алгебарски степен.

2.7.1 Познати напади и одговарајући захтеви пројектовања

Комбинациони генератори су рањиви на корелационе нападе и варијанту корелационих напада звану брзи корелациони напади. Да би ови напади били неизводљивим, повратни полином ПРЛПС не би требало бити редак. Комбинациона функција би требало да има велики ред (корелациони имунитет) такође назван ред еластичности када је функција балансирана. Ипак, постоји компромис између корелационог имунитета и алгебарског степена Булове функције. Пре свега, корелациони имунитет балансиране Булове функције од n променљивих не прелази $n - 1 - \deg(f)$, када је алгебарски степен функције f, $\deg(f)$, већи од један. Штавише, сложеност корелационог напада и брзог корелационог напада се повећава са нелинеарношћу комбинационом функцијом. Тражење компромиса између алгебарског степена, корелационог имунитета и нелинеарности се може избећи тако што се комбинациона функција замени коначним аутоматом са меморијом. Пример комбинационог генератора са меморијом имамо код генератора који користе операцију целобројног сабирања и проточних шифара E_0 коришћених код блутута [LV04].

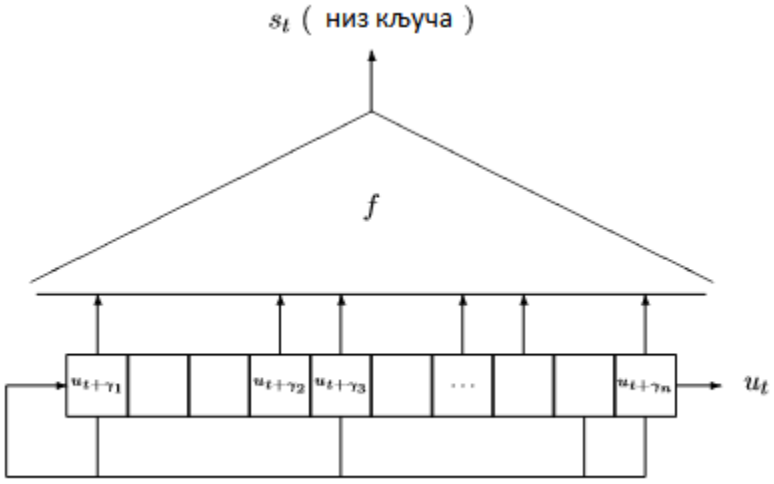
2.8 Филтерски генератори

Филтерски генератор је генератор за генерисање низа кључа за проточне шифре. Састоји се од једног ПРЛПС који се филтрира нелинеарном функцијом. Прецизније, излазни низ филтерског генератора одговара излазу нелинеарне функције чији улаз су стања неких ћелија ПРЛПС.

Ако $(u_t)_{t\geq 0}$ представља низ генерисан уз помоћ ПРЛПС, излазни низ $(s_t)_{t\geq 0}$ филтерског генератора је

$$s_t = f\left(u_{\gamma_1+t}, u_{\gamma_2+t}, \dots, u_{\gamma_n+t}\right), \forall t \geq 0,$$

где је f функција од n променљивих, n је мање или једнако дужини ПРЛПС и $(\gamma_i)_{1\leq i\leq n}$ представља опадајући низ ненегативних целих бројева назван низ позиција извода са регистра.



Слика 2.6 Филтерски генератор

Да би низ кључа имао добра статистичка својства, филтерска функција треба бити балансирана, а повратни полином ПРЛПС треба изабрати да буде примитиван.

Код филтерских генератора, повратни полином ПРЛПС, филтерска функција и низ позиција извода са регистра су углавном јавни. Почетно стање ПРЛПС је тајни параметар. Стога, највећи број напада на филтерске генераторе састоји се у откривању почетног стања познавањем неколико битова добијених из генератора (напад са познатим отвореним текстом), или неколико битова низа шифрата (када се зна само шифрат).

Било који филтерски генератор еквивалентан је специфичном комбинационом генератору, у смислу да оба генератора производе исти излазни низ. Еквивалентан комбинациони генератор састоји се од n копија ПРЛПС коришћених у филтерском генератору са помереним почетним стањем, и још треба напоменути да комбинациона функција одговара филтерској функцији.

Својства излазног низа

Излазни низ s филтерског генератора је линеарно рекурентни низ. Линеарна сложеност низа, $L(s)$, је повезана са дужином ПРЛПС и алгебарским степеном филтерске функције f. За бинарни ПРЛПС са примитивним повратним полиномом, важи

$$L(s) \leq \sum_{i=0}^{d} \binom{L}{i}$$

где L означава дужину ПРЛПС, а d означава алгебарски степен функције f [Key76, Mas01]. Период низа s дели $2^L - 1$. Ако је L велики прост број, $L(s)$ је најмање $\binom{L}{d}$ за већину филтерских функција са алгебарским степеном d. Да би се постигла нека велика линеарна сложеност, дужина L ПРЛПС и алгебарски степен филтерске функције требало би да буду довољно велики. Прецизније, дужина низа кључа доступна нападачу требало би бити увек знатно мања од $\binom{L}{\deg(f)}$.

2.8.1 Познати напади

Филтерски генератори су рањиви на брзе корелационе нападе јер је излазни низ корелисан са неком линеарном комбинацијом стања ћелија ПРЛПС. Познато је да је линеарна комбинација ћелија низ, чија је линеарна сложеност једнака дужини ПРЛПС. Да би корелациони напади били рачунски неизводљиви, филтерска функција треба имати велику нелинеарност. Још један критеријум за дизајн филтерских генератора је да повратни полином не треба бити редак.

2.9 Пример: Гефеов генератор

Гефеов генератор је добро познати нелинеарни комбинациони генератор. Састоји се од три ПРЛПС са различитим полиномима. У сваком кораку, излаз из генератора једнак је првом или другом ПРЛПС, у зависности од вредости излаза трећег ПРЛПС.

Нека су дата три ПРЛПС која су саставни део Гефеовог генератора са следећим повратним полиномима

$$\text{ПРЛПС}_a: \quad f_a(x) = x^{21} + x^2 + 1 \in F_2[x],$$

$$\text{ПРЛПС}_b: \quad f_b(x) = x^{23} + x^5 + 1 \in F_2[x],$$

$$\text{ПРЛПС}_c: \quad f_c(x) = x^{17} + x^3 + 1 \in F_2[x],$$

и почетним стањем

$$A_0 = (a_0, \dots, a_{20}),$$

$$B_0 = (b_0, \dots, b_{22}),$$

$$C_0 = (c_0, \dots, c_{16}),$$

Излазни низ s_t Гефеовог генератора је дефинисан на следећи начин

$$s_t = \begin{cases} a_t, c_t = 0 \\ b_t, c_t = 1 \end{cases}. \qquad \textbf{(2.2)}$$

Потребно је израчунати период низа $s = (s_t)_{t \geq 0}$ који се добија као излаз из Гефеовог генератора. Повратни полиноми су несодљиви и примитивни над $F_2[x]$, што даље своди на претходна тврђења да повратни полиноми имају максималан период. Периоди од ПРЛПС_a, ПРЛПС_b, ПРЛПС_c су редом $2^{21} - 1$, $2^{23} - 1$, $2^{17} - 1$. Правило **(2.2)** алгебарски се може написати на следећи начин

$$s_t = F(a_t, b_t, c_t) = (c_t + 1)a_t + c_t b_t = c_t a_t + c_t b_t + a_t \in F_2$$

Више ПРПЛС различитих дужина са повратним примитивним полиномима су комбиновани са нелинеарном функцијом F. Стога, резултујући низ има линеарну сложеност једнаку функцији F, под условом да сви ПРЛПС имају ненулто почетно стање. Добија се линеарна сложеност Гефеовог низа под условом да нема ненула почетно стање,

$$F(21, 23, 17) = 17 * 21 + 17 * 23 + 21 = 769.$$

Нека Ω представља скуп низова који се добијају помоћу Гефеовог генератора где су A_0, B_0, C_0 ненула почетна стања. Ови низови имају линеарну сложеност 769. За сваки низ може се утврдити одговарајући повратни полином степена 769. Како Гефеов генератор пролази кроз сва могућа стања три улазна ПРЛПС, последица је то да елементе из Ω разликује само померање у неком времену t и сви ПРЛПС имају минималне полиноме.

Означимо f_z повратни полином степена 769. Како сви низови из Ω имају исти период који је једнак периоду функције f_z, и како период функције f_z дели најмањи заједнички садржалац периода ПРЛПС$_а$, ПРЛПС$_б$ и ПРЛПС$_ц$, добија се да је период функције f_z $(2^{21} - 1) * (2^{23} - 1) * (2^{17} - 1)$, па тако сви низови из Ω имају период једнак $(2^{21} - 1) * (2^{23} - 1) * (2^{17} - 1) \approx 2^{61}$.

Гефеов генератор генерише низове кључа са великим периодом, и Гефеов генератор је веома ефикасан. Међутим Гефеов генератор није отпоран на корелациони напад. Ако је излазни низ Гефеовог генератора у корелацији са једним ПРЛПС, могуће је одредити почетно стање једног ПРЛПС независно од остатка. Онда је то битно брже од претраге по почетним стањима свих регистара.

3 Корелациони напад

Корелациони напад представио је Siegenthaler 1985. године [Sie85]. Напад се односи на генератор низа кључа који се састоји од померачких регистара са линеарном повратом спрегом (ПРЛПС). Корелациони напад користи технику завади па владај: циљ је откривање почетног стања сваког саставног ПРЛПС у ситуацији када се зна одређени број битова низа кључа. Такође, могућ је напад када се зна само шифрат ако постоји редунданса у отвореном тексту.

Оригинални корелациони напад односи се на неке комбинационе генераторе од n ПРЛПС дужина $L_1, \dots, L_n$. Напад у најједноставнијем случају омогућава откривање комплетног почетног стања генератора са само $\sum_{i=1}^{n}(2^{L_i}-1)$ провера уместо $\prod_{i=1}^{n}(2^{L_i}-1)$ провера које захтева потпуна претрага. Ефикасни корелациони напад може се применити и на друге генераторе низа кључа базираних на ПРЛПС, који су на неки начин специјалан случај филтерских генератора.

3.1 Основна идеја

Разматра се нелинеарни комбинациони генератор чији су делови n померачких регистара ПРЈПС, $ПРЛПС_1$,…,$ПРЛПС_n$ дужине $L_1, \dots, L_n$ који су део нелинеарног комбинационог генератора. Нека су $R_0^{(1)}, \dots, R_0^{(n)}$ почетна стања тих ПРЛПС. Нека је $r_t^{(i)}$ излаз из $ПРЛСП_i$ и нека је s_t излаз из генератора у кораку t. Претпоставка је да нападач има приступ низу кључа $s = (s_i)_{i\geq 0}$, тј. у питању је напад са познатим отвореним текстом. За проналазак почетног стања n-тог ПРЛПС, требало би проверити свако могуће почетно стање да се види које доводи до низа кључа s. Један ПРЛПС дужине L има $2^L - 1$ могућих ненула почетних стања. Стога, потребно је проверити сваку могућу комбинацију свих могућих почетних стања, а то значи проверити

$$\prod_{\mathrm{i}=1,\dots,\mathrm{n}} 2^{l_i} - 1 \approx 2^{\Sigma_i l_i},$$

различитих комбинација почетних стања.
Нека важи следећа корелациона веорватноћа

$$P_r\left(r_l^{(i)} = s_t\right) = \frac{1}{2} + \varepsilon_i, \text{где је } \varepsilon_i > 0 \text{ за све i} = 1, \dots, \mathrm{n}.$$

Претпоставља се да је ε_i позитивно пошто је потребно да је корелациона вероватноћа различита од $\frac{1}{2}$. Ознака ε_i зове се корелација.

За дати вектор $y \in F_2^N$, Хемингова тежина $wt(y)$ је број ненула коефицијената вектора y. За $x,y \in F_2^N$ Хемингово растојање између x и y, је број коефицијената у којима се x и y разликују, односно $wt(x+y)$. Инверзно Хемингово растојање је број коефицијената у којима се x и y подударају.

За вектор $x \in F_2^N$, ознака $x \oplus 1$ представља сабирање свих координата вектора x са 1 по модулу 2. Односно, $x \oplus 1$ је уствари комплемент x. У даљем тексту, за векторе $x, y \in F_2^N$, ознака $wt(x \oplus y \oplus 1)$ је инверзно Хемингово растојање. Постојање корелације излаза са излазом неког ПРЛПС може се употребити за проналажење почетног стања тог ПРЛПС независно од осталих ПРЛПС. Нека је s излазни низ дужине N и нека је $r^{(i)} = \left(r_t^i\right)_{1 \le i \le N}$ одговарајући улазни низ за ПРЛПС$_i$. Очекивано инверзно Хемингovo растојање између s и $r^{(i)}$ је

$$E\left(wt\left(r^{(i)} \oplus s \oplus 1\right)\right) = \mathrm{N}\left(\frac{1}{2} + \varepsilon_i\right) = \frac{N}{2} + \varepsilon_i N.$$

С друге стране, за било коју случајни низ $z = (z_t)_{1 \le t \le N}$ важи

$$E(wt(z \oplus s \oplus 1)) = \frac{N}{2}.$$

За одређивање почетног стања i-тог ПРЛПС потребно је генерисати низ z помоћу i-тог ПРЛПС, за сва могућа почетна стања тог регистра, који се потом пореди са низом кључа s. У ту сврху рачуна се инверзно Хемингово растојање

$$h_z^{(i)} := wt(z \oplus s \oplus 1).$$

Ако је z излазни низ регистра добијен са погођеним почетним стањем, онда је $h_z^{(i)}$ случајна променљива са биномном расподелом $B(N, \frac{1}{2} + \varepsilon_i)$, а ако је z излазни низ регистра добијен за било које погрешно почетно стање, онда је $h_z^{(i)}$ случајна променљива са биномном расподелом $B(N, \frac{1}{2})$. То значи, ако је $r^{(i)}$ погођено почетно стање, онда је очекивање $\frac{N}{2} + \varepsilon_i N$. Ако није погођено почетно стање, онда се $r^{(i)}$ разликује од претпостављеног излаза из ПРЛПС$_i$ на око $\frac{N}{2}$ позиција, па је ефекат као да је s сабран са случајним низом.

За проналазак почетног стања, у случају да је ε_i мало потребан је велики узорак, у овом случају број N. Претпоставља се да постоје услови за проналазак исправног почетног стања, па се могу пронаћи почетна стања различитих ПРЛПС независно од осталих ПРЛПС што води ка сложености

$$\sum_{i=1,\dots,n} 2^{L_i} - 1.$$

Сложеност оваквог поступка је знатно мања од потпуне претраге где је потребно искомбиновати сва почетна стања различитих ПРЛПС што води ка сложености $\prod_{\mathrm{i}=1,\dots,\mathrm{n}}(2^{L_i} - 1)$. Чак иако се корелациони напад примени само на један ПРЛПС, сложеност напада потпуном претрагом се смањује за фактор 2^{L_i}

$$2^{L_1} + \prod_{\mathrm{i}=2,\dots,\mathrm{n}} 2^{L_i} - 1 \approx \frac{2^{\sum_{i=1,\dots,n} L_i}}{2^{L_i}}.$$

3.2 Основна варијанта корелационог напада

Излазни низ ПРЛПС израчунава се за сва могућа почетна стања $R_0 \in F_2^L$. За свако почетно стање R_0 добија се излазни низ r дужине N. За генерисање низа r користи се **Алгоритам** 1 (поглавље 2.6). Ови низови се пореде са низом s и рачуна се инверзно Хемингово растојање. Низ са највећим инверзним Хеминговим растојањем је највероватније онај који се тражи. Испоставља се да је то низ са највећом вероватноћом да буде исправно почетно стање.

Алгоритам 2

Улаз: бинарни низ кључа s дужине N, ПРЛПС дужине L и корелација $\varepsilon > 0$
Излаз: претпоставка о почетном стању R_0 дужине L ПРЛПС и одговарајуће инвернзно Хемингово растојање

- $\text{maxInvDistance} = 0$
- **for** $i = 1, \dots, 2^L - 1$ **do**

  - $R_0 = i$

  - генерише се низ $z = (z_i)_{0 \le i \le N}$ са почетним стањем R_0 помоћу ПРЛПС користећи **Алгоритам 1**

  - рачуна се инверзно Хемингово растојање $h_z = wt(z \oplus s \oplus 1)$

  - Ако је $h_z > \text{maxInvDistance}$ онда $\text{maxInvDistance} = h_z$; $\text{tempClosest} = R_0$

- **return** $\text{tempClosest}, \text{maxInvDistance}$

Нека је L дужина циљаног ПРЛПС који има корелацију $\varepsilon > 0$ са низом кључа s, и нека је $w > 0$ број ненула коефицијената карактеристичног полинома. **Алгоритам 2** доноси претпоставку о почетном стању циљаног ПРЛПС користећи мање од $2^L N(w + 2)$ сабирања. Почетно стање добијено **Алгоритмом 2** није увек тачно, али може се израчунати вероватноћа тачности.

Нека је R_0 почетно стање низа добијеног помоћу ПРЛПС, и нека је $\varepsilon > 0$ корелација између низа добијеног помоћу ПРЛПС и низа кључа s дужине N. Вероватноћа да почетно стање Z_0 добијено **Алгоритмом 2** даје одговарајуће инверзно Хемингово растојање h је

$$P_r(S_0 = R_0 | H.d. = h) = \frac{\left(\frac{1}{2} + \varepsilon\right)^h (\frac{1}{2} - \varepsilon)^{N-h} (\frac{1}{2})^l}{(\frac{1}{2} + \varepsilon)^h (\frac{1}{2} - \varepsilon)^{N-h} \left(\frac{1}{2}\right)^l + (\frac{1}{2})^N (1 - (\frac{1}{2})^l)}. \quad (\mathbf{3.1})$$

За дати излазни низ алгоритма, може се утврдити вероватноћа исправности. Међутим, не зна се тачна вероватноћа да ли алгоритам враћа исправно почетно стање. Сам одабир

дужине низа N игра веома битну улогу у исправном проналажењу почетног стања. Следеће поглавље се бави исправним одабиром дужине низа кључа N.

Јасно је да сложеност **Алгоритма 2** расте експоненцијално са дужином L ПРЛПС. Наравно, за веће L потребно је проверити више почетних стања. Истовремено, L не утиче значајно на сложеност **Алгоритма 1**. Може се закључити да већи број повратних коефицијената игра кључну улогу, стога било би исправно одабрати карактеристичне полиноме на следећи начин:

- Одабрати примитивне карактеристичне полиноме са малим бројем повратних коефицијената и великог степена, да би генератор низа кључа био ефикасан а у исто време да има велику комплексност корелационог напада.

Овакви примитивни карактеристични полиноми су веома интеренсантни код нелинеарних комбинационих генератора. Међутим, у поглављу 4 показује се да су генератори са карактеристичним полиномима који имају мали број повратних коефицијената ипак рањиви на једну врсту корелационих напада, брзи корелациони напади. Ипак, главна ствар код одабира комбинационог генератора није у самим карактеристичним полиномима, већ у одабиру нелинеарне комбинационе функције F. Главну улогу код корелационих напада игра корелација ε. Ако је ε мало, онда је потребан велики низ кључа, тј. N треба да је велико да би напад био успешан. Уствари, показаће се да је

$$N = O\left(\frac{1}{\varepsilon^2}\right).$$

Параметар ε се одређује помоћу комбинационе функције F, па је тако потребно изабрати F такво да се дешава мала корелација. Типичан пример нелинеарног комбинационог генератора је Гефеов генератор, представљен у поглављу 2.9.

3.2.1 Корелациони напад на Гефеов генератор

У овом поглављу примењују се **Алгоритм 1** и **Алгоритам 2** на Гефеов генератор представљен у поглављу 2.9. Три улазна ПРЛПС имају ознаку $ПРЛПС_a$, $ПРЛПС_b$ и $ПРЛПС_c$, њихови излази у кораку t са a_t,b_t и c_t, и одговарајући излазни низ Гефеовог генератора са s_t.

Прво, потребно је увидети корелацију. Табела 3.1 показује излани низ s_t Гефеовог генератора за одређене улазне вредности a_t,b_t и c_t.

a_t	b_t	c_t	s_t
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Табела 3.1: Излаз из Гефеовг генератора

Корелационе вероватноће се могу директно очитати са ове таблице. Нека се посматра корелација између $ПРЛПС_a$ и низа кључа s. Имамо

$$a_t = s_t \Leftrightarrow (a_t, b_t, c_t) = \begin{cases} (0,0,0) \\ (0,0,1) \\ (0,1,0) \\ (1,0,0) \\ (1,1,0) \\ (1,1,1) \end{cases}$$

и

$$a_t \neq s_t \Leftrightarrow (a_t, b_t, c_t) = \begin{cases} (0,1,1) \\ (1,0,1) \end{cases}.$$

Како су сви карактеристични полиноми три улазна ПРЛПС примитивни, излаз $ПРЛПС_a$, $ПРЛПС_b$ и $ПРЛПС_c$ може се разматрати уравнотеженим, тј. $P_r(a_t = 0) = P_r(b_t = 0) = P_r(c_t = 0) = \frac{1}{2}$. Како су сва три $ПРЛПС_a$, $ПРЛПС_b$ и $ПРЛПС_c$ линеарно независна, јер су њихове дужине, а самим тим и дужине периода, узајамно прости бројеви, добија се

$$P_r((a_t, b_t, c_t) = (\alpha, \beta, \gamma)) = \frac{1}{8} \text{ за све } (\alpha, \beta, \gamma) \in F_2^3.$$

На основу претходног добија се

$$P_r(a_t = s_t) = \frac{3}{4} = \frac{1}{2} + \frac{1}{4}.$$

и

$$P_r(b_t = s_t) = \frac{1}{2} + \frac{1}{4},$$

и

$$P_r(c_t = a_t \oplus s_t) = \frac{1}{2} + \frac{1}{4}.$$

На основу претходних корелација, може се очекивати да описан напад ради исправно за низове кључа довољно велике дужине N. Уствари, показаће се вероватноћа проналажења тачног почетног стања расте упоредо са растом дужине низа кључа N.

3.3 Статистичка анализа корелационог напада

Нека важе ознаке из претходних поглавља, намеће се питање:
Под којим условима је могуће одредити да ли је h_s случајна променљива са биномном расподелом $B\left(N,\frac{1}{2}\right)$ или $B\left(N,\frac{1}{2}-\varepsilon\right)$?

3.3.1 Тест максималне веродостојности

Нека је $X=(X_1,X_2,\dots,X_N)$ случајна променљива, где су $X_1,X_2,\dots,X_N$ независне случајне променљиве. Нека је $D(y)=P_r[X_i=y]$ расподела променљивих $X_1,X_2,\dots,X_N$ и нека је $D^N(\bar{x})=P_r^N[X=\bar{x}]$ расподела променљиве X. Нека је дат узорак $\bar{x}=(x_1,x_2,\dots,x_N)$ случајне променљиве X који припада простору Ω, потребно је одлучити која од ове две хипотезе

- $H_0: D=P_0$, случајна променљива X_i има расподелу P_0
- $H_1: D=P_1$, случајна променљива X_i има расподелу P_1

је тачна. Са $\phi(x):\Omega\to\{0,1\}$ може се означити одлучујућа функција, где ако је $\phi(x)=0$ значи да важи хипотеза H_0, а ако је $\phi(x)=1$ значи да важи хипотеза H_1. Могу се десити два типа грешке:

- $E_F: \phi(\bar{x})=1$, иако важи H_0,
- $E_M: \phi(\bar{x})=0$, иако важи H_1.

Пожељно је изабрати функцију $\phi(x)$ на начин тако да су вероватноће грешке

- $P_F=P_0^N[\phi(X)=1]$,
- $P_M=P_1^N[\phi(X)=0]$,

минималне. Природно, постоји неки компромис између ове две вероватноће. То се лако може видети ако се изабере да је $\phi=1$ константа. Онда је $P_M=0$, па је наравно P_F максимално. Класичан приступ је да се уведе граница α за P_M и P_F, и да се изабере такво ϕ да су вероватноће минималне, $P_F\le\alpha$ односно $P_M\le\alpha$.

Дефиниција 3.1 За дати узорак *x*, веродостојност од H_0 односно H_1 једнако је

$$p_0(x) := \text{вероватноћа од } x \text{ када је } H_0 \text{ тачно},$$
$$p_1(x) := \text{вероватноћа од } x \text{ када је } H_1 \text{ тачно}.$$

Однос ових вероватноћа означава се

$$\Lambda(x) := \frac{p_0(x)}{p_1(x)}.$$

Дефиниција 3.2 Укупна вероватноћа грешке је

$$P_e = \pi_0 P_F + \pi_1 P_M$$

где су π_0 и π_1 претходне вероватноће хипотеза H_0 и H_1 где важи $\pi_0 + \pi_1 = 1$.

Следећа лема, Њуман-Пирсонова лема, говори како треба изабрати ϕ да би се добила оптимална одлучујућа функција у скупу свих функција где је $P_M \leq \alpha$. Ставове и леме које се наводе у наредном делу рада, се узимају за важеће без претходног доказа. Сами докази ових тврђења могу се наћи у литератури која се наводи на крају рада.

Лема (Њуман-Пирсонова лема) 3.1 Оптимална одлучујућа функција има следећу форму:

$$\phi(\bar{x}) = \begin{cases} 0, \Lambda(\bar{x}) \geq k, \\ 1, \Lambda(\bar{x}) \leq k, \end{cases}$$

Вредност прага k је одређена условом $P_M = \alpha$.

У случају да су вероватноће грешке једнаке $P_M = P_F = \alpha$, онда би требало изабрати $k = 1$. Јасно је да ефикасност корелационог напада зависи од вероватноће грешака и оптималне одлучујуће функције ϕ. Код корелационог напад, рачуна се инверзно Хемингово растојање h_s. Потребно је одредити да ли је h_s узорак $B\left(N, \frac{1}{2} + \varepsilon\right)$ или $B\left(N, \frac{1}{2}\right)$ случајне променљиве.

Став 3.1 Нека је s бинарни низ дужине N, и нека h представља Хемингову тежину. Коефицијент веродостојности у одлучујућој функцији која треба да каже да ли је низ узорак случајне променљиве Y са расподелом $B\left(N, \frac{1}{2} + \varepsilon\right)$ (Хипотеза H_0) или случајне променљиве X са расподелом $B\left(N, \frac{1}{2}\right)$ (Хипотеза H_1), једнак је

$$\Lambda(h, N) = \frac{P_0^N(h)}{P_1^N(h)} = \frac{\binom{N}{h}(\frac{1}{2} + \varepsilon)^h(\frac{1}{2} - \varepsilon)^{N-h}}{\binom{N}{h}(\frac{1}{2})^N} = \frac{(\frac{1}{2} + \varepsilon)^h(\frac{1}{2} - \varepsilon)^{N-h}}{(\frac{1}{2})^N}.$$

Овај коефицијент расте заједно са порастом h.

Из претходног, може се утврдити да за дату дужину низа N, коефицијент веродостојности зависи само од h. Па тако, услов $\Lambda \geq k$ у одлучујућој функцији може се заменити са $H \geq k$ па се добија нови праг $0 \leq H \leq N$

$$\frac{(\frac{1}{2} + \varepsilon)^H(\frac{1}{2} - \varepsilon)^{N-H}}{(\frac{1}{2})^N} = k,$$

и добија се

$$H = \frac{\log k - N \log\left(\frac{\frac{1}{2}}{\frac{1}{2}-\varepsilon}\right)}{\log\left(\frac{1}{2}+\varepsilon\right) - \log\left(\frac{1}{2}-\varepsilon\right)}.$$

Став 3.2 Нека су H_0 и H_1 хипотезе дефинисане у ставу 3.1, и нека је H праг Хемингове тежине у одлучујућој функцији ϕ. Вероватноће грешке су

$$P_F = P_0^N[\phi(X) = 1] = \sum_{h \geq H} \binom{N}{h} \left(\frac{1}{2}\right)^N$$

$$P_M = P_1^N[\phi(X) = 0] = \sum_{h < H} \binom{N}{h} \left(\frac{1}{2}+\varepsilon\right)^h \left(\frac{1}{2}-\varepsilon\right)^{N-h}.$$

Може се закључити да дату корелацију ε, вероватноће грешке зависе од дужине кључа N и прага H.

3.3.2 Чебишевљева неједнакост

Став 3.4 (Чебишевљева неједнакост) Нека је X случајан променљива са очекиваном вредношћу μ и коначном варијансом σ^2. Важи следећа неједнакост

$$P_r[|X - \mu| \geq k\sigma] \leq \frac{1}{k^2}.$$

Другим речима, за $k > 1$ вероватноћа да је узорак унутар радијуса од k пута стандардно одступање је већа од $1 - \frac{1}{k^2}$.

Став 3.5 Нека је X случајан променљива са биномна расподела $B\left(N, \frac{1}{2}\right)$ и Y случајна променљива са биномном расподелом $B\left(N, \frac{1}{2}+\varepsilon\right)$ са очекивањем

$$\mu = N\frac{1}{2} \qquad \text{и} \qquad \mu_\varepsilon = N\left(\frac{1}{2}+\varepsilon\right)$$

и стандардним одступањем

$$\sigma = \sqrt{\frac{1}{4}N} \qquad \text{и} \qquad \sigma_\varepsilon = \sqrt{N\left(\frac{1}{4}-\varepsilon^2\right)}.$$

Нека је H праг Хемингове тежине h у одлучујућој функцији ϕ. Добијају се следеће границе за вероватноће грешке

$$P_F(H) \leq \frac{1}{2}\frac{1}{k^2} \qquad \text{и} \qquad P_M(H) \leq \frac{1}{2}\frac{1}{k_\varepsilon^2},$$

где су

$$k = \frac{H-\mu}{\sigma} \qquad \text{и} \qquad k_\varepsilon = \frac{\mu_\varepsilon - H}{\sigma_\varepsilon}.$$

Став 3.6 Нека су дати услови из става 3.5. Повећањем дужине низа N и одабиром одговарајућег прага H, добија се да су вероватноће грешке произвољно мале.

Став 3.7 Нека су F и M горење границе за вероватноће грешке P_F и P_M, и нека је

$$k \leq \sqrt{\frac{1}{2P_F}} \qquad \text{и} \qquad k_\varepsilon \leq \sqrt{\frac{1}{2P_M}}.$$

Ако су

$$N > \frac{\left(\frac{1}{2}k + k_\varepsilon\sqrt{\frac{1}{4} - \varepsilon^2}\right)^2}{\varepsilon^2},$$

и

$$\frac{1}{2}N + k\sqrt{\frac{1}{4}N} < H < N\left(\frac{1}{2} + \varepsilon\right) - k_\varepsilon\sqrt{\frac{1}{4} - \varepsilon^2},$$

Одлучујућа функција ϕ има вероватноће грешке

$$P_F \leq F \qquad \text{и} \qquad P_M \leq M.$$

Став 3.8 Вероватноћа успеха **Алгоритма 2** за

$$N > \frac{\left(\frac{1}{2}k + k_\varepsilon\sqrt{\frac{1}{4} - \varepsilon^2}\right)^2}{\varepsilon^2},$$

већа је од

$$\left(1 - \frac{1}{k^2}\right)^{2^{L-2}}\left(1 - \frac{1}{k_\varepsilon^2}\right)$$

где је L дужина циљаног ПРЛПС. Са порастом дужине низа кључа расте и вероватноћа успеха **Алгоритма 2** и тежи ка 1.

3.4 Уопштење корелационог напада

Из става 3.8 показује се да **Алгоритам 2** враћа исправно почетно стање ако је дужина кључа N довољно велика. На основу свега виђеног до сада, може се дати уопштенији алгоритам корелационог напада у односу на **Алгоритам 2**. Нека је дат комбинациони генератор чији излаз је у корелацији са једним ПРЛПС, низ кључа s, корелација ε и број почетнних стања M које треба тестирати.

1. За сва могућа почетна стања генерисати одговарајуће низове z^i дужине N.
2. Израчунати инверзно Хемингово растојање h_z^i низа кључа s.
3. Применити оптималну одлучћућу функцију из Њуман-Пирсонове леме за све h_s^i да би се одредило да ли важи хипотеза H_0 (расподела $B\left(N, \frac{1}{2}+\varepsilon\right)$) или хипотеза H_1 или (расподела $B\left(N, \frac{1}{2}\right)$). Нека је Z_a скуп почених стања за које важи хипотеза H_0 и Z_d скуп почетних стања за које важи хипотеза H_1.
4. Елиминише се скуп Z_d. Увећава се N и рекурзивно се примењује алгоритам на скуп Z_a.

Вероватноћа да је исправно почетно стање у скупу Z_a једнака је

$$P_r(\text{исправно почетно стање у } Z_a) = 1 - P_M.$$

Очекивани број неисправних почетних стања у скупу Z_a је

$$MP_{F.}$$

На основу става 3.6, закључује се да повећавањем дужине кључа N може се изабрати произвољно мало P_M и P_F. У сучају када скуп Z_a садржи само исправно почетно стање, тада би и **Алгоритам 2** вратио исправан резултат. Међутим, ово захтева велики низ кључа N. Предност уопштенијег алгоритма јесте тај што се немогућа почетна стања елиминишу са низом кључа релативно мале дужине, где код **Алгоритма 2** морамо проверити сва почетна стања са истим низом кључа.

4 Брзи корелациони напад

До сада је показано како се може употребити корелациони напад на нелинеарне комбинационе генераторе за проналазак почетног стања. Ако постоји нека корелација између ПРЛПС низа и низа кључа, почетно стање се може одредити одвојено, што смањује линеарну сложеност напада на $O(2^L)$ где је L дужина ПРЛПС. Ако сви саставни ПРЛПС имају велику дужину L, упркос смањењу линеарне сложености у поређењу са нападом потпуном претрагом, корелациони напад може бити неизбежан. Код брзих корелациони напада, користи се линеарна зависност међу битовима низа кључа за проналажење почетног стања. Криптоаналитички проблем проналажења почетног стања ПРЛПС може се посматрати као проблем декодирања линеарног блока. За оне који нису упознати са линеарним блок кодовима, детаљан опис може се пронаћи у [МЖ90]. Овај приступ се може користи приликом напада на филтерске генераторе.

4.1 Основни појмови о кодовима за исправљање грешака

Пре самог представљања брзог корелационог напада, описују се кодови за исправљање грешака. Уводе се ознаке и основни појмови кодова за исправљање грешака.

Нека је $B_n = \{0,1\}^n$, $n \geq 1$ и $B = B_1 = \{0,1\}$. Елементи скупа B_n су n-димензионални бинарни вектори, односно матрица димензије $n \times 1$. Ако је M матрица, а $j = (j_1, j_2, \dots)$ уређени скуп индекса, онда M_j означава матрицу формирану од колона матрице M са индексним редом $(j_1, j_2, \dots)$. Вектор $M_{\{j\}}$ означава се једноставније са M_j. Ако је M вектор-колона, онда је M_j ознака за његову j-ту кординату.

При преносу порука кроз канал везе долази до њиховог изобличења. Ако се на улаз канала доведе бинарни вектор $x \in B_n$, под дејством шума се на другом крају канала добија случањи бинарни n-димензиони вектор Y. Реализација y случајне променљиве Y у општем случају није једнака вектору x. Под претпоставком да су грешке појединих координата независне и једнако вероватне, са вероватноћом p, каже се да је вектор Y настао порпуштањем вектора x кроз бинарни симетрични канал (БСК) са вероватноћом прелаза p.

Дефиниција 4.1 Нека је E случајна n-димензиона векторска променљива са расподелом вероватноћа

$$P\{E = e\} = p^{\omega(e)}(1-p)^{n-\omega(e)}, e \in B_n,$$

при чему је $0 \leq p \leq 1$, а са $\omega(e)$ за $e \in B_n$ је означена тежина вектора e, односно број његових кордината различитих од нуле. Пропуштањем вектора $x \in B_n$ кроз бинарни симетрични канал (БСК) са вероватноћом прелаза p добија се векторска случајна променљива $Y = x \oplus E$, односно примљена порука, где је са $\oplus$ означена покоординатно сабирање вектора по модулу два.

У даљем тексту се под каналом подразумева БСК. Да би било могуће исправљати грешке у примиљеној поруци, на улаз канала се доводе вектори из неког подскупа скупа B_n. Једна класа оваквих подскупова су *линеарни кодови*.

Дефиниција 4.2 Нека су n, k природни бројеви, $0 \leq k \leq n$, и нека је G матрица димензије $k \times n$ чији је ранг једнак k. Скуп вектора

$$C = \{x \mid x = u^T G, u \in B_k\}$$

је линеарни код (n, k) (или $[n, k]$) код. Овде је са T означена операција транспоновања матрице. Елемент кода $x \in C$ је кодна реч. Параметар n је дужина кодних речи. Матрица G је генеришућа матрица кода C. Линеарни код C је системски ако има генеришућу матрицу G такву да је $G_{i,j} = \delta_{i,j}$, $1 \leq i, j \leq k$, где $\delta_{i,j}$ означава Кронкеров симбол

$$\delta_{i,j} = \begin{cases} 1, i = j \\ 0, i \neq j \end{cases}.$$

Линеарни код C је линеарни k-димензиони потпростор линеарног простора који чини скуп B_n над пољем $GF(2)$. База овог простора је скуп вектор-врста матрице G. Параметар k је *димензија линеарног кода* C. Код системског линеарног кода све координате кодне речи могу се очигледно изразити као линеарна комбинација првих k координата кодне речи. Ова чињеница може се уопштити увођењем појма *информационог скупа*.

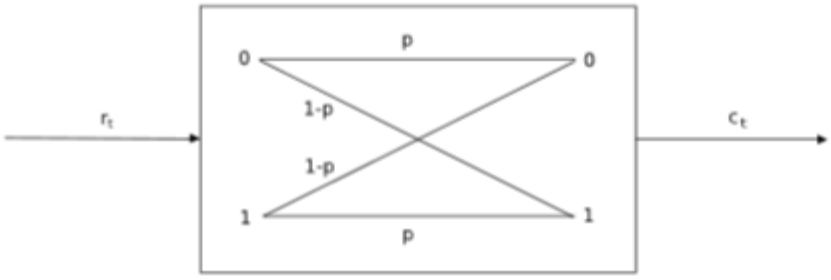
Дефиниција 4.3 Нека је C линеарни (n, k) код са генеришућом матрицом G. Дуални код линеарног кода C је линеарни $(n, n-k)$ код једнак ортогоналном комплементу линеарног потпросотра C. Димензија дуалног кода $n-k$ обично се означава са r. Матрица H је контролна матрица линеарног кода C ако је дуални код C' генеришућа матрица. Елементи дуалног кода $a \in C'$, односно линеарне форме $a^T x$, $x \in B_n$, су контроле парности линеарног кода C, с обзиром да је $a^T x = \mathbf{0}$ за свако $x \in C$. Овде је са $\mathbf{0}$ означен нула вектор.

Јасно је да се линеарни код може еквивалентно дефинисати једнакошћу

$$C = \{x \in B_k \mid Hx = \mathbf{0}\}.$$

Ако је j' комплемент информационог скупа j линеарног кода C, односно $j' = \{1,2, \dots, n\} \backslash j$, онда је j' информациони скуп дуалног кода C'. Заиста, ако се координате кодне речи са индексима из скупа j' напишу као линеарна комбинације координата са индексима из скупа j, добије се контролна матрица H кода C (генеришућа матрица кода C') таква да матрица $H_{j'}$ има у свакој колони тачно једну јединицу, што значи да је несингуларна.

Исти бит x_i кодне речи, $1 \leq i \leq n$, може да буде обухваћен са више контрола парности $a^{(j)T} x = \mathbf{0}$, при чему су $a^{(j)}$ неке кодне речи дуалног кода, $1 \leq i \leq l$. Овакав систем контрола је ортогоналан у односу на бит x_i ако у свакој колони матрице чије су врсте вектори $a^{(j)T}$ (сем i-те) постоји највише једна јединица. Наравно, сви елементи i-те колоне ове матрице су јединице.



Слика 4.1 Бинарни симетрични канал са вероватноћом прелаза $1-p$

4.2 Блок код модел

Нека Π означава скуп свих могућих ПРЛПС низова добијених помоћу ПРЛПС дужине L. Посматрањем само првих $N \geq L$ битова сваког низа из Π, скуп Π_N представља скуп свих скраћених низова. Јасно је да је $|\Pi_N| = 2^L$. Нека је

$$G = \begin{pmatrix} 0 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & \cdots & \cdots & 0 \\ \vdots & & & \ddots & & \vdots \\ \vdots & & & & 1 & 0 \\ 0 & 0 & \cdots & \cdots & 0 & 1 \\ c_0 & c_1 & c_2 & \cdots & \cdots & c_{N-1} \end{pmatrix},$$

матрица транзиционог стања. Па је тако i-то стање ПРЛПС једнако $G^i k$, где $k = (k_1, \dots, k_L)$ означава почетно стање ПРЛПС. Стога сваки бит r_i, у низу добијеном помоћу ПРЛПС, је линеарна комбинација L-тог почетног бита, па је $r_i = g_i k^T = k g_i^T * g_i$ први ред матрице G^{i-1}. Скуп Π_N се може посматрати као линеарни (N, L) код са одговарајућом генеришућом матирцом

$$G_{\text{ПРЛПС}} = (g_1 \quad g_2 \quad \cdots \quad g_N).$$

Јасно је да су $g_1, \dots, g_L$ L димензиони јединични вектори, и да је g_{L+1} вектор који садржи повратне коефицијенте карактеристичног полинома. Па тако $G_{\text{ПРЛПС}}$ има стандардну форму

$$G_{\text{ПРЛПС}} = (I_L \quad P)$$

где је I_L јединична матрица димензије $L \times L$ и P је матрица димензије $L \times (N-L)$. Онда је почетно стање једнако информационом скупу одговарајућег низа дужине N који је једнак кодној речи.

Нека је $r = (r_t)_{1 \leq t \leq N}$ представља скраћени низ добијен помћу ПРЛПС и нека је $s = (s_t)_{1 \leq t \leq N}$ одговарајући излаз из генератора низа кључа са корелационом вероватноћом $p = P_r(r_t = s_t) = \frac{1}{2} + \varepsilon$. Низ r се може посматрати као кодна реч линеарног (N, L) кода. У [Sieg85] се може наћи да се под одређеним условима (који се своди на то да се низ вектора

на које се примењују Булова функција приближно равномерно пролази кроз скуп B_L) може сматра да је низ битова s добијен пропуштањем низа r кроз БСК са вероватноћом прелаза p, $1 \leq i \leq p$, при чему вероватноћа p зависи само од Булове функције и редног броја i низа r. Из претходног следи, низ битова s је реч добијена преко бинарног симетричног канала са вероватноћом $1 - p$ кад се на улаз доведе низ r. Ако се може декодирати низ s, може се пронаћи и почетно стање ПРЛПС. То значи, да се проналажење почетног стања ПРЛПС своди на декодирање кодних речи.

Применом ML-декодирања пореде се све кодне речи са добијеном речи и тражи се најприближнија. Техника ML-декодирања је оптимална када је грешка вероватноће минимална, али је и спора. То је приступ који је одговарајући стандардном корелационом нападу и његова сложеност је $O(2^L)$. Међутим, постоје линеарни блок кодови, где се субоптималане итеративне методе декодирања веома добро изводе. Класа таквих блокова је $LDPC$ код, ту класу је представио Галагер [Gal63]. За велико N, линеарни блок код у нашемо случају може се посматрати као $LDPC$ код. Стога, није изненађење да је алгоритам представљен од стране Мејера и Штафелбека [MS88], који су у неком смислу измислили брзи корелациони напад, личи на итеративну технику декодирања или $LDPC$ кодове. У овом раду неће бити посебног објашњавања $LDPC$ кодова.

4.3 Оригинални брзи корелациони напад

У овом поглављу представља се брзи корелациони напад представљен од стране Мејера и Штафелбека [MS88]. За брзи корелациони напад користи се линеарни блок код модел. Такође, представљају се **Алгоритам А** и **Алгоритам В** за декодирање речи. **Алгоритам А** представља декодирање у једном кораку, чије перформансе се могу предвидети за дату ситуацију. **Алгоритам В** је итеративан алгоритам веома сличан алгоритму који се користи код $LDPC$ кодова. Оба алгоритма су заснована на оцени контроле парности датих линеарном стуктуром кодне речи које произилазе из ПРЛПС. Нека је

$$f = a_0 + a_1 x + \cdots + a_{L-1} x^{L-1} + a_L x^L$$

повратни полином ПРЛПС дужине L. Нека је ω број повратних коефицијената ПРЛПС, број коефицијената различитих од нуле у повратном полиному је $\omega + 1$.

За низ добијен помоћу ПРЛПС важи

$$a_0 r_i \oplus a_1 r_{i-1} \oplus \ldots \oplus a_L r_{i-L} = 0.$$

Па се тако добија регуларна контролна матрица H димензије $(N - L) \times N$ за кодне речи дужине N, добијене помоћу ПРЛПС.

$$H = \begin{pmatrix} a_L & a_{L-1} & \cdots & a_0 & 0 & \cdots & \cdots & 0 \\ 0 & a_L & a_{L-1} & \ddots & a_0 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & a_L & a_{L-1} & \cdots & a_0 \end{pmatrix}$$

За брзи корелациони напад пожељно је имати велики број конторла парности. Због тога су конструисане додатне котроле парности са матрицом H. То значи да свако множење полинома f дефинише једначину контроле парности низа r. Како у коначном пољу са два елемента F_2 важи $f^{2^i}(x) = f(x^{2^i})$, подизањем повратног полинома f на квадрат, добија се више једначина за конторлу парности тежине $\omega + 1$. Дужина овако добијених контрола парности је $2^i L$. Просечан број конторла парности за сваки бит $(r_i)_{1 \le i \le N}$ означава се са J и добија се на следећи начин [MS88]

$$J = \log_2 \left(\frac{N}{2L}\right)(\omega + 1).$$

Контрола парности јасно даје информацију о исправности бита. У зависности од броја контрола парности које један бит примљеног низа задовољава, може се израчунати вероватноћа тачности. Број контрола парности говори о исправности једног бита.

Контрола парности тежине ω на битовима са једнаким вероватноћама тачности p

Нека је дата једначина контроле парности тежине $\omega + 1$ на биту u_1

$$u_1 \oplus u_{i_1} \oplus \ldots \oplus u_{i_\omega} = 0, 1 < i_1 < \cdots < i_\omega \qquad \textbf{(4.1)}$$

и нека је $r = (r_i)_{i\geq 1}$ низ који задовољава контроле парности. Нека је $s = (s_i)_{i\geq 1}$ и нека важи $P_r(r_t = s_t) = \frac{1}{2} + \varepsilon$. Са

$$c := s_1 \oplus s_{i_2} \oplus \ldots \oplus s_{i_\omega}$$

дефинишемо контроле парности **(4.1)** примењене на низу s. Може се израчунати вероватноћа да контрола парности пружа исправне информације о биту s_1. Коришћењем контрола парности **(4.1)** може се изразити апостериорна вероватноћа грешке.

Ако је дефинисано n независних бинарних случајних променљивих $X_1, \ldots, X_n$ са вероватноћом $P_r(X_i = 0) = \frac{1}{2} + \varepsilon_i$ онда се добије [Vau06]

$$P_r(X_1 \oplus \ldots \oplus X_n = 0) = \frac{1}{2} + 2^{n-1} \prod_{i=1,\ldots,n} \varepsilon_i.$$

На основу претходног, добија се

$$p_\omega := P_r(c = 0 | s_1 = r_1) = P_r(c \neq 0 | s_1 \neq r_1) = \frac{1}{2} + 2^{\omega-1}\varepsilon^\omega.$$

Еквивалентно имамо

$$p_\omega := P_r(c = 0 | s_1 \neq r_1) = P_r(c \neq 0 | s_1 = r_1) = 1 - p_\omega = \frac{1}{2} - 2^{\omega-1}\varepsilon^\omega.$$

Нека је

$$\rho := 2^{\omega-1}\varepsilon^\omega$$

како је $|\varepsilon| < \frac{1}{2}$ лако се уочава да се повећањем ω смањује $|\rho|$ па је стога

$$p_\omega \xrightarrow{\omega\to\infty} \frac{1}{2}.$$

Претпоставља се да је дато J контрола парности првог бита и да су сви они тежине $\omega + 1$, и да две контроле парности не укључују исти бит осим првог. Догађај да једна контрола парности даје исправну информацију о првом биту је независан о догађају да друга контрола парности даје исправну информацију о првом биту. Стога, ако је $s_1 = r_1$ онда број контрола парности које задовољавају S има биномну расподелу $B\left(J, \frac{1}{2} + \rho\right)$, ако је $s_1 \neq r_1$ онда има биномну расподелу $B\left(J, \frac{1}{2} - \rho\right)$. За веома мало ρ очекује се да вредности ових расподела буду сличне. Број битова који задовољавају конторле парности се не разликују

много од битова који незадовољавају. Генерално, што је веће ρ то је потребно мање контрола парности да би се одлучило да ли је бит исправан или погрешан.

Заправно, може се израчунати вероватноћа исправности бита за дати број конторла парности које су задовољене. Добија се

$$P_r(c = 0) = P_r(s_1 = r_1)p_\omega + P_r(s_1 \neq r_1)(1 - p_\omega) = pp_\omega + (1 - p)(1 - p_\omega). \quad (\mathbf{4.2})$$

Коришћењем Бајесове формуле следи

$$P_r(s_1 = r_1 | c = 0) = \frac{P_r(c = 0 \wedge s_1 = r_1)}{P_r(c = 0)}. \quad (\mathbf{4.3})$$

Стога, на основу **(4.2)** и **(4.3)** и како је

$$P_r(c = 0 \wedge s_1 = r_1) = pp_\omega$$

добија се

$$P_r(s_1 = r_1 | c = 0) = \frac{pp_\omega}{pp_\omega + (1 - p)(1 - p_\omega)}. \quad (\mathbf{4.4})$$

Како је претпоставка да су све контроле парности независне, и нека је задовољено укупно h од J контрола парности, онда се може лако утврдити вероватноћу да је први бит исправан. Нека је S број задовољених конторла парности, онда

$$p^* = P_r(s_1 = r_1 | S = h) = \frac{\binom{J}{h} pp_\omega^h (1 - p_\omega)^{J-h}}{\binom{J}{h}\left(pp_\omega^h (1 - p_\omega)^{J-h} + (1 - p)(1 - p_\omega)^h p_\omega^{J-h}\right)} = \frac{pp_\omega^h (1 - p_\omega)^{J-h}}{\left(pp_\omega^h (1 - p_\omega)^{J-h} + (1 - p)(1 - p_\omega)^h p_\omega^{J-h}\right)} \quad (\mathbf{4.5})$$

Ову вероватноћа p^* се зове апостериорна вероватноћа грешке.

Сада се може израчунати вероватноћа да бит задовољава најмање Λ контрола парности

$$\sum_{\Lambda \leq h \leq J} \binom{J}{h} \left(pp_\omega^h (1 - p_\omega)^{J-h} + (1 - p)(1 - p_\omega)^h p_\omega^{J-h}\right), \quad (\mathbf{4.6})$$

и вероватноћа да је $s = r$, под условом да је задовољено бар h од J контрола парности

$$p_\Lambda = \frac{\sum_{\Lambda \leq h \leq J} \binom{J}{h} pp_\omega^h (1 - p_\omega)^{J-h}}{\sum_{\Lambda \leq h \leq J} \binom{J}{h} \left(pp_\omega^h (1 - p_\omega)^{J-h} + (1 - p)(1 - p_\omega)^h p_\omega^{J-h}\right)}. \quad (\mathbf{4.7})$$

Контрола парности тежине ω на битовима са различитим вероватноћама тачности p_i

Претпоставља се да је дата једначина контроле парности **(4.1),** и нека је $r = (r_i)_{i\geq 1}$ низ који задовољава контроле парности. Нека је $y = (y_i)_{i\geq 1}$ и нека важи $P_r(r_i = y_i) = \frac{1}{2} + \varepsilon_i$. Са

$$c' := y_1 \oplus y_{i_2} \oplus \ldots \oplus y_{i_\omega},$$

дефинишемо контроле парности **(4.1)** примењене на низу y. Опет постоји вероватноћа да контрола парности производи исправне информације о биту y_1

$$b(p_{j_1}, \ldots, c_{j_\omega}) := P_r(c = 0 | y_1 = r_1) = \frac{1}{2} + \prod_{i=j_1,\ldots,j_\omega} \varepsilon_i. \qquad \mathbf{(4.8)}$$

Стога, за сваку контролу парности може се израчунати вероватноћа b. Нека је задовољено h од укупно J контрола парности. Без губитка општости, нека $c_1, \ldots, c_h$ задовољавају контролу парности и нека $c_{h+1}, \ldots, c_J$ не задовољавају контролу парности. Нека су $b_1, \ldots, b_J$ одговарајуће вероватноће. Онда, под претпоставком да су контроле парности независне, може се израчунати вероватноћа да је y_1тачно.

$$P_r(s_1 = u_1 | c_1 = \cdots = c_h = 0, c_{h+1} = \cdots = c_J = 1) \qquad \mathbf{(4.9)}$$
$$= \frac{p_1 \prod_{t-1,\ldots,h} b_t \prod_{t-h+1,\ldots,J}(1-b_t)}{p_1 \prod_{t-1,\ldots,h} b_t \prod_{t-h+1,\ldots,J}(1-b_t) + (1-p_1)\prod_{t-1,\ldots,h}(1-b_t)\prod_{t-h+1,\ldots,J} b_t}.$$

4.3.1 Алгоритам А

Нека важе тврђења из претходних поглавља, **Алгоритам А** ради на следећи начин. Прво, процене се све контроле парности. На основу тога, за сваки бит $s_1, \ldots, s_N$ рачуна се апостериорна вероватноћа p^*. За оне апостериорне вероватноће p^* које достигну праг λ претпоставља се да су тачне. Ако имамо довољно исправних битова, почетно стање се може пронаћи решавањем линеарних једначина.

Главна ствар код овог алгортма јесте правилан одабир прага λ. Ако је λ превише велико, не добија се довољно битова за проналажење почетног стања. Ако је λ превише мало, добија се превише нетачних битова у скупу битова код којих је апостерирона вероватноћа већа од λ. Онда је потребно урадити много исправки у скупу ових битова да би се пронашло почетно стање, а то онемогућава ефикасно декодирање.

Као што се може видети у **(4.5)**, апостериона вероватноћа p^* зависи само од броја контрола парности које су задовољене. Стога се може утврдити праг Λ за број h задовољених контрола парности

$$p^* = \frac{p p_\omega^h (1-p_\omega)^{J-h}}{\left(p p_\omega^h (1-p_\omega)^{J-h} + (1-p)(1-p_\omega)^h p_\omega^{J-h}\right)} \geq \lambda \Leftrightarrow h \geq \Lambda.$$

Нека Ω_Λ означава скуп битова који задовољавају најмање Λ контрола праности. Очекивана кардиналност $N_{>\Lambda}$ скупа Ω_Λ може се израчунати једначином **(4.6)**

$$N_{>\Lambda} = N \sum_{\Lambda \le h \le J} \binom{J}{h} \left(p p_\omega^h (1-p_\omega)^{J-h} + (1-p)(1-p_\omega)^h p_\omega^{J-h} \right)$$

док једначина **(4.7)** даје вероватноћу да је бит из Ω_Λ тачан.

$$p_\Lambda = \frac{\sum_{\Lambda \le h \le J} \binom{J}{h} p p_\omega^h (1-p_\omega)^{J-h}}{\sum_{\Lambda \le h \le J} \binom{J}{h} \left(p p_\omega^h (1-p_\omega)^{J-h} + (1-p)(1-p_\omega)^h p_\omega^{J-h} \right)}.$$

Сада се могу навести захтеви који су потребни да би напад био успешан. Подсећања ради, нека L представља дужину ПРЛПС, стога је потребно најмање L исправних битова да би се пронашло почетно стање ПРЛПС.

- Одредити средњи број контрола парности J
- Треба изабрати Λ тако да је $N_{>\Lambda}$ веће од L.
- У исто време, број очекиваних нетачних битова у Ω_Λ, добијених помоћу

  $(1-p_\Lambda)N_{>\Lambda}$, треба бити што мање, пожељно мање од 1. Што је овај број већи, то је потребно проверити више битова из Ω_Λ на исправност.

Избор одговарајућег прага λ није увек могућ. У ставри, Мајер и Штафелбек су показали да се комплексност корелационог напада $O(2^L)$ може смањити на $O\left(2^{\beta L}\right)$ где β зависи од ω, p и $\frac{N}{L}$. За велико ω ($\omega \geq 16$), процељује се да β треба бити близу функције $H(p) = -p \log_2 p - (1-p)(1-p)$, и зато се не може очекивати много побољшања у сложености.

Пример 4.3.1 Нека је $p = 0.55$, онда је $H(p) = -0.55 \log_2 5.5 - 0.45 \log_2 4.5 \approx 0.99$.

4.3.2 Алгоритам Б

Сваки бит $s_i, i = 1, \dots, N$ има вероватноћу тачности p. Проценом контрола парности, може се израчунати апостериорна вероватноћа тачности p^*. Стога сваки бит $s_i, i = 1, \dots, N$ има нову вероватноћу тачности. Коришћењем ових вероватноћа као нове априорне вероватноће, поново се може израчунати апостерироне вероватноће. Ово рачунање се може итеративно применити. Нека је $p_i^{(j)}$ вероватноћа тачности бита s_i у итерацији j. Ако је $p_i^{(1)} = p$ за све $i = 1, \dots, N$, онда итерација изгледа на следећи начин:

- Итерација 0: Ради се процена свих контрола парности. Рачунају се апостериорне вероватноће $p_i^{(1)}$ применом **(4.5)**
- Итерација $j > 0$: Израчунати $p_i^{(j*1)}$ применом **(4.9)**

После одређеног броја итерација, они битови са вероватноћом испод прага λ се комплементирају и њихова вероватноћа тачности се ресетује на p.

У наставку се анализара очекивани ефекат корекције. Тачније, процењује се ефекат корекције после прве итерације. У итерацијама већим од 0, вероватноће различитих битова више нису независне. Тада, вероватноће израчунате у алгоритму нису тачне. Стога може се заменити праг λ са прагом Λ на броју битова контрола парности које задовољавају. Очекивани број битова за које је број задовољних контрола парности мањи од прага може се израчунати преко

$$N_{>\Lambda} = N \sum_{\Lambda \le h \le J} \binom{J}{h} \left(p p_\omega^h (1 - p_\omega)^{J-h} + (1 - p)(1 - p_\omega)^h p_\omega^{J-h} \right).$$

Ако су сви ови битови комплементирани, добијамо повећање тачних битова I,

$$I = N \sum_{0 \le h \le \Lambda} \binom{J}{h} p_\omega^h (1 - p_\omega)^{J-h} - N \sum_{0 \le h \le \Lambda} \binom{J}{h} (1 - p)(1 - p_\omega)^h p_\omega^{J-h}.$$

Оцена границе овог алгоритма већ се може видети посматрањем нулте итерације. За p_ω које има вредност око $\frac{1}{2}$, очекиван број битова за исправљање је мали. Међутим, у току наредних итерација може се појавити већи број исправљених битова. У ствари, Мајер и Штафелбек су показали да у случајевима када је корелациона вероватноћа мала $(p \le 0.75)$ и када је број повратних коефицијената већи $(t \ge 10)$ Алгоритам Б (као и Алгоритам А) је неупотребљиви. Постоје побољшања ових алгоритама А и Б, међутим у овом раду се не наводе. Примери неких побољшаних алгоритама могу се наћи у [CT00] и [LH04].

5 Програмска реализација напада

Прогарам који имплементира алгоритам за корелациони напад написан је на програмском језику C++, у развојном окружењу Qt Creator 3.5.1.

Састоји се из 7 класа. Може се рећи да је програм написан из два дела, један део је графичко корисничко окружење (GUI), док други део представља класе које се користе за формирање генератора низа кључа и самог напада на те исте генераторе. Сам напад се одвија у посебној нити програма, и независан је од графичког корисничког окружења. Покретањем функције main(), покреће се GUI. GUI класа је изведена из класе QMainWindow. Класа QMainWindow пружа подршку за рад у главном прозору програма. Класе које се користе у овом програму а почињу са словом Q, су класе које се могу користити уз развојно окружење Qt Creator. Овај рад даје само основна значења тих класа, док се детаљан опис ових класа може наћи на следећем линку http://doc.qt.io/qt-4.8/.

5.1 Генерисање низа битова

Класа LFSR представља померачки регистар са линеарном повратном спрегом. Класа садржи следеће чланове приватног типа:

- QSharedPointer<Polynomial> m_oPolynomial;
- uint32_t m_uiInitialState;
- uint32_t m_uiCurrentState;
- uint32_t m_uiLength;

QSharedPointer<Polynomial> m_oPolynomial је приватан члан, који садржи показивач на објекат типа Polynomial и који представља повратни полином ПРЛПС дефинисан класом LFSR. QSharedPointer је делељни показивач, који се понаша као нормални показивач с тим што дељени показивач аутоматски ослобађа алоцирану меморију. m_uiInitialState представља почетно стање LFSR. m_uiCurrentState представља тренутно стање у коме се налази регистар, док m_uiLength представља дужину регистра. Класа LFSR садржи следећу методу:

- unsigned short ShiftRegister();

Метода ShiftRegister() помера регистар за један бит удесно, рачуна повратни бит у зависности од повратног полинома и повратни бит смешта на најлевију позицију у регистру. Док се бит који је шифтован удесно, враћа као повратна вредност методе. Такође, треба поменути да ова класа, и неке класе које се описјују у раду, имају имплементиране тзв. гетере и сетере, методе које служе за читање односно за сетовање вредности чланова класе.

Класа Polynomial је класа која представља полином. Класа Polynomial садржи следеће чланове приватног типа:

- uShort *m_usPolynomialCoefficient;

- uShort m_usDegree;

usPolynomialCoefficient представља низ коефицијената полинома, док usDegree представља степен полинома.

Генератор класа

Представљена класа LFSR описује један ПРЛПС за генерисање низа битова, међутим потребно је неколико излаза из ПРЛПС комбиновати једном функцијом. Па се тако ствара потреба за нову класу, класу Generator. Класа Generator комбинује један или више излаза ПРЛПС Буловом функцијом и као излаз добија се један бит. Класа Generator је базна класа за неке типове генератора. Класа Generator садржи следеће приватне чланове:

- QList<QSharedPointer<LFSR> > m_qShpList;

m_qShpList представља листу дељених показивача на LFSR. Методе ове класе су следеће:

- virtual uShort *GenerateOneBit*();
- virtual QString *GenerateNBits*(unsigned int N);

Метода *GenerateOneBit*() је виртуелна метода, у изведеним класама ова метода се може предефинисати. Служи за генерисање једног излазног бита комбиновањем више излазних битова саставних ПРЛПС, једном Буловом функцијом. Метода враћа излазни бит. Друга метода ове класе је *GenerateNBits*(unsigned int N), која је такође виртуелна. Служи за генерисање одређеног броја битова. Аргумент ове методе је број битова који треба генерисати. Метода враћа низ N битова у текстуалном формату.

Следеће класа CombinationGenerator, је класа која је изведена из класе Generator. Ова класа има предефинисане методе *GenerateOneBit*() и *GenerateNBits*(unsigned int N).

5.2 Корелациони напад

Сада, представљене класе из претходног поглавља могу се искористити за имплементацију напада. Класа LFSRProcessor представља главну класу за корелациони напад. Сам напад се извршава у посебној нити програма, а комуникација са GUI класом MainWindow се одвија преко сигнала и слотова. Класа садржи следеће приватне чланове:

- QSharedPointer<Generator> m_shpGenerator;
- QVector<int> m_qiVector;

m_shpGenerator је дељени показивач на генератор низа кључа који се напада. m_qiVector је вектор низа кључ. Класа садржи следеће методе:

- void SetKeyStream(QString strKeyStream);
- float ProbabilityCalculation(float fProbability, int iMatches, int iNumberOfBits, int iLFSRLength);
- void process();

Метода SetKeyStream(QString strKeyStream) као аргумент има низ кључа у текстуалном формату. Метода SetKeyStream узима бит по бит из прослеђеног низа кључа, и конвертује у цео број и чува у вектору m_qiVector. Метода ProbabilityCalculation рачуна вероватноћу да је одређено почетно стање исправно у зависности од инверзног Хеминговог растојања. Метода има четири аргумента, float fProbability, int iMatches, int iNumberOfBits, int iLFSRLength. fProbability представља корелациону вероватноћу, iMatches инверзно Хемингово растојање, тј. број битова у којима се кандидат за исправно почетно стање и низ кључа подударају. iNumberOfBits је број битова са којим се напада регистар. iLFSRLength је дужина циљаног ПРЛПС. Метода process() служи за конкретан напад на одређени ПРЛПС. Метода напада ПРЛПС са одређеном корелационом вероватноћом. За свако могуће почетно стање, у опсегу од 1 до $2^L - 1$, где је L дужина ПРЛПС, генерише се низ битова једнак дужини низа кључа који се чува у вектору m_qiVector. Затим се рачуна инверзно Хемингово растојање и узима се почетно стање са највећим инверзним Хеминговим растојањем. Затим, помоћу методе ProbabilityCalculation рачуна се вероватноћа да је добијено почетно стање исправно и емитује се сигнал за завршетак напада, који се после хвата у класи MainWindow. Сигнали који се могу емитовати из класе LFSRProcessor су:

- void finished();
- void emitResults(QString strResulst);

Сигнал finished() обавештава класу MainWindow да је напад завршен и да може да се изађе из нити, тј. да се исправно обрише нит из програма. Сигнал emitResults(QString strResulst) обавештава класу MainWindow да ажурира податке у графичком корисничком окружењу.

5.3 Графичко корисничко окружење (GUI)

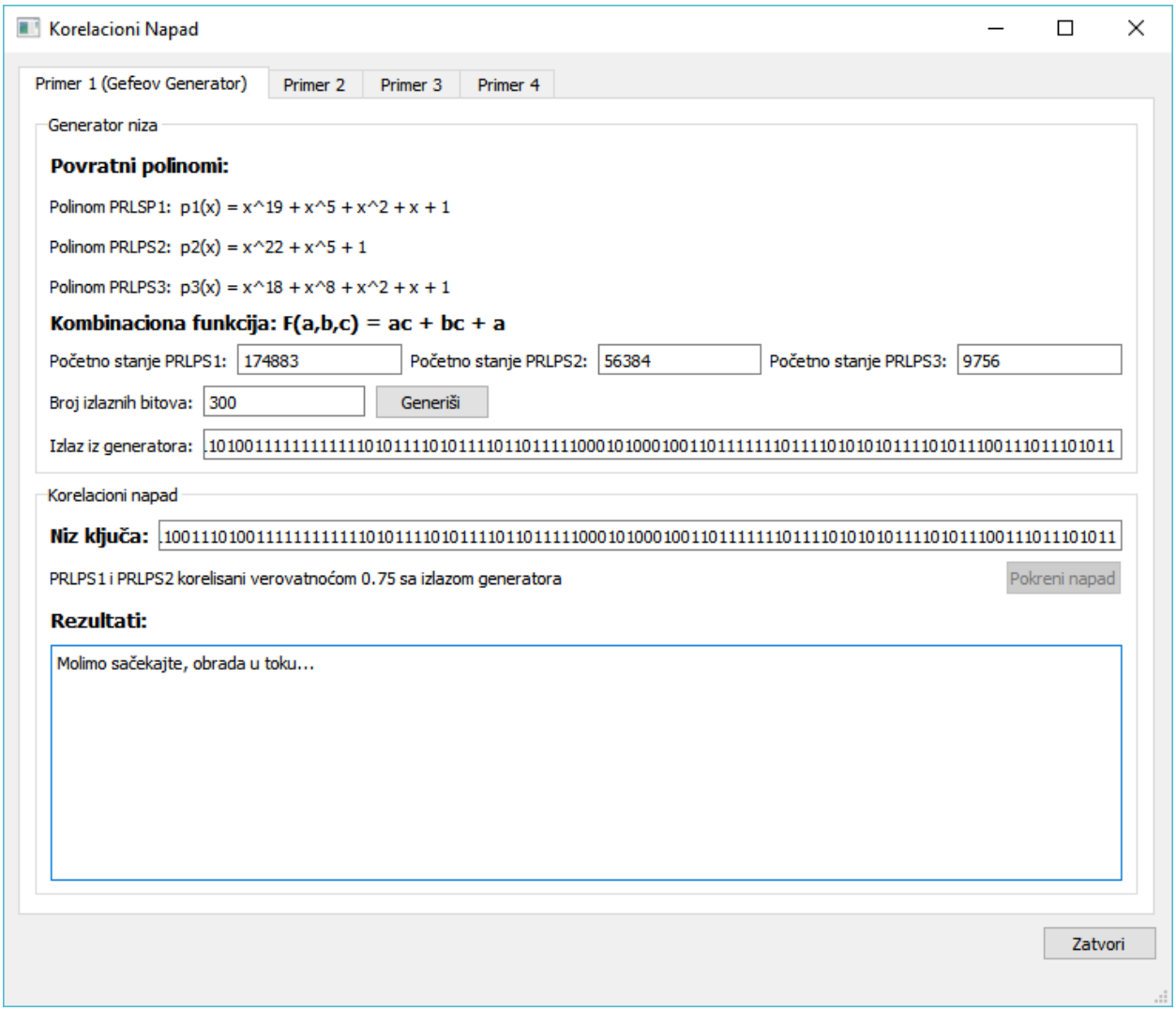
Графички део програма покреће се позивом функције main(). Рад у графичко-корисничком окружењу подразумева приказивање прозора апликације и омогућава једноставно коришћење функционалности програма. Основни прозор апликације састоји се од четири одвојена таба. Сваки таб је пример једног генератора. Први таб је пример Гефеовог генератора, док остала три таба представљају комбинациони генератор са истим повратним полиномима али са различитим комбинационим функцијама.

Класа MainWindow пружа подршку за рад у главном прозору програма. Класа садржи следеће приватне чланове:

- Ui::MainWindow *ui;
- QSharedPointer<Generator> m_shpGenerator;
- QSharedPointer<Generator> m_shpGenerator1;
- QSharedPointer<Generator> m_shpGenerator2;
- QSharedPointer<Generator> m_shpGenerator3;

ui представља класу за рад са графичким компонентама. m_shpGenerator, m_shpGenerator1, m_shpGenerator2 и m_shpGenerator3 прдстављају дељене показиваче на објекте класе

Generator. Класа MainWindow садржи слотове који се позивају на одређене догађаје унутар главног прозора, као што су дугме притиснуто, унет текст у неко поље за унос текста и друго. Такође, поред ових постоји и слот који на сигнал из нити, која врши напад на регистар, ажурира податке о успешности напада.



Сва четири таба имају исти изглед, само различите комбинационе функције. Састоје се од два панела, један на горњој страни и један на доњој страни. Горњи део прозора односи се на генерисање низа битова, доњи на напад и резултате напада. У делу који се односи на генерисање низа битова постоји опис повратних полинома и комбинационе функције. За успешно генерисање низа битова потребно је унети вредности у поља која означавају почетна стања генератора. То стање истовремено представља и тренутно стање ПРЛПС и од тог стања се генеришу следећи битови. Вредност која се уноси у поље за почетно стање је целобројна и у опсегу од 1 до $2^L - 1$, где L представља дужину ПРЛПС. Такође, потребно је унети број битова који треба генерисати. Кликом на дугме „Generiši“ генерише се одређен број битова и ажурира се поље „Izlaz iz generatora“.

Доњи део прозора односи се на корелациони напад и резултате напада. У зависности од таба на коме се корисник налази, врши се корелациони напад на одређен ПРЛПС или више њих. У поље „Niz ključa“ уноси се низ кључа. За тај низ кључа примењује се корелациони напад и после завршеног напад, ажурира се поље за испис резултат. У пољу за испис резултата исписују се: пронађено почетно стање, вероватноћу да је добијено почетно стање исправно у зависности од вредности инверзног Хемингового растојања, очекивани број погођених битова, стандардно одступање, инверзно Хемингово растојање и време трајања напада у секундама.

5.4 Добијени резултати

Корелациони напад се врши на неке типове генератора низа кључа и приказују се добијени резултати. За пример се узимају генератори низа кључа који су у корелацији са неким ПРЛПС. За тако изабране генераторе показују се добијени резултати напада. Корелационим нападом се нападају генератори низа кључа који се састоје од три саставна ПРЛПС али са различитим комбинационим функцијама. Нека μ означава очекиван број погођених битова, σ^2 стандардно одступање, $wt(x \oplus y \oplus 1)$ инверзно Хемингово растојање. Сви напади који се врше на одређене генераторе низа кључа тестирају се на процесору Intel® Core™ i3-4000M CPU @ 2400GHz 2400GHz.

Пример 1 (Гефеов Генератор)

Комбинациона функција: $F(a_t, b_t, c_t) = c_t a_t + c_t b_t + a_t$

Корелационе вероватноће се могу очитати директно са табеле 5.1.

a_t	b_t	c_t	s_t
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Табела 5.1 Гефеов генератор низа кључа корелисан са два регистра

Корелационе вероватноће имају следеће вредности

$$P_r(a_t = s_t) = \frac{3}{4}, P_r(b_t = s_t) = \frac{3}{4}, P_r(c_t = s_t) = \frac{1}{2}.$$

N	μ	σ^2	$wt(x \oplus y \oplus 1)$	Погођено почетно стање	Време
100	75	4.3	88	Да	3s
150	112.5	5.3	124	Да	4s
200	150	6.1	166	Да	6s
300	225	7.5	250	Да	9s
400	300	8.6	338	Да	13s
600	450	10.6	513	Да	21s
800	600	12.2	684	Да	26s

Табела 5.2 Резултати напада на први ПРЛПС

N	μ	σ^2	$wt(x \oplus y \oplus 1)$	Погођено почетно стање	Време
100	75	4.3	75	Не	38s
150	112.5	5.3	100	Не	55s
200	150	6.1	150	Не	71s
300	225	7.5	190	Не	106s
400	300	8.6	250	Не	145s
600	450	10.6	362	Да	229s
800	600	12.2	488	Да	290s

Табела 5.3 Резултати напада на други ПРЛПС

Пример 2

Комбинациона функција: $F(a_t, b_t, c_t) = \neg(a_t c_t) + \neg(a_t b_t) + a_t c_t$

Корелационе вероватноће се могу очитати директно са табеле 5.4.

a_t	b_t	c_t	s_t
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Табела 5.4 Генератор низа кључа корелисан са три регистра

Корелационе вероватноће имају следеће вредности

$$P_r(a_t = s_t) = \frac{3}{8}, P_r(b_t = s_t) = \frac{3}{8}, P_r(c_t = s_t) = \frac{5}{8}.$$

N	μ	σ^2	$wt(x \oplus y \oplus 1)$	Погођено почетно стање	Време
100	62.5	2.5	86	Не	18s
150	93.75	3	122	Не	27s
200	125	3.5	166	Не	36s
300	187.5	4.3	234	Не	54s
400	250	5	294	Не	75s
600	375	6.1	415	Не	112s
800	500	7	521	Не	145s

Табела 5.5 Резултати напада на први ПРЛПС

N	μ	σ^2	$wt(x \oplus y \oplus 1)$	Погођено почетно стање	Време
100	62.5	2.5	83	Не	6s
150	93.75	3	119	Не	10s
200	125	3.5	158	Не	14s
300	187.5	4.3	223	Не	20s
400	250	5	287	Не	28s
600	375	6.1	404	Не	41s
800	500	7	530	Не	55s

Табела 5.6 Резултати напада на други ПРЛПС

N	μ	σ^2	$wt(x \oplus y \oplus 1)$	Погођено почетно стање	Време
100	62.5	2.5	70	Не	1s
150	93.75	3	99	Не	1s
200	125	3.5	127	Не	1s
300	187.5	4.3	184	Да	2s
400	250	5	246	Да	3s
600	375	6.1	360	Да	4s
800	500	7	491	Да	5s

Табела 5.7 Резултати напада на трећи ПРЛПС

Пример 3

Комбинациона функција: $F(a_t, b_t, c_t) = \neg(a_t b_t) + \neg b_t c_t + a_t c_t$.

Корелационе вероватноће се могу очитати директно са табеле 5.8.

a_t	b_t	c_t	s_t
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Табела 5.8 Генератор низа кључа корелисан са два регистра

Корелационе вероватноће имају следеће вредности

$$P_r(a_t = s_t) = \frac{1}{2}, P_r(b_t = s_t) = \frac{1}{4}, P_r(c_t = s_t) = \frac{5}{8}.$$

N	μ	σ^2	$wt(x \oplus y \oplus 1)$	Погођено почетно стање	Време
100	75	4.33	73	Не	6s
150	112.5	5.3	102	Не	10s
200	150	6.1	135	Не	13s
300	225	7.5	193	Не	21s
400	300	8.6	247	Не	28s
600	450	10.6	373	Да	40s
800	600	12.2	489	Да	53s

Табела 5.9 Резултати напада на други ПРЛПС

N	μ	σ^2	$wt(x \oplus y \oplus 1)$	Погођено почетно стање	Време
100	62.5	2.5	72	Не	0s
150	93.75	3	101	Не	1s
200	125	3.5	130	Не	1s
300	187.5	4.3	190	Да	2s
400	250	5	257	Да	2s
600	375	6.1	374	Да	4s
800	500	7	502	Да	5s

Табела 5.10 Резултати напада на трећи ПРЛПС

Пример 4

Комбинациона функција: $F(a_t, b_t, c_t) = a_t c_t + \neg b_t c_t + \neg a_t b_t \neg c_t$.

Корелационе вероватноће се могу очитати директно са табеле 5.11.

a_t	b_t	c_t	s_t
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Табела 5.11 Генератор низа кључа корелисан са једним регистром

Корелационе вероватноће имају следеће вредности

$$P_r(a_t = s_t) = \frac{1}{2}, P_r(b_t = s_t) = \frac{1}{2}, P_r(c_t = s_t) = \frac{5}{8}.$$

N	μ	σ^2	$wt(x \oplus y \oplus 1)$	Погођено почетно стање	Време
40	25	1.5	34	Не	0s
60	37.5	1.9	45	Не	0s
80	50	2.2	60	Да	0s
100	62.5	2.5	75	Да	0s
200	125	3.5	154	Да	1s

Табела 5.12 Резултати напада на трећи ПРЛПС

6 Закључак

Приказан је корелациони напад на проточне шифре са генератором који као свој део садржи померачке регистре са линеарном повратном спрегом. Приказана је основна верзија корелационог напада коју је представио Siegenthaler [Sie85], и како се сложеност напада може смањити иако је само један од саставних ПРЛПС у корелацији са излазом генератора.

Поред основне верзије корелационог напада размотрени су и тзв. брзи корелациони напади.

Програмски је реализован корелациони напад на Гефеов генератор и на неке комбинационе генераторе. Имплементација алгоритма садржи кориснички интерфејс за генерисање низа битова генератора и напада на исти. Програм даје и могућност интеракције корисника током рада, избором почетних стања за генерисање низова и уноса самог низа кључа који се напада.

Интересантан правац за наставак рада у овој области је реализација варијанти брзог корелационог напада.

7 Литература

[Ber67] E.R. Berlekamp. Algebraic coding theory. McGraw-Hill, 1967.

[Bry85] L. Brynielsson. On the linear complexity of combined shift register sequences. In: Advances in Cryptology - EUROCRYPT ’85, Lecture Notes in Computer Science, number 219, pages 156–160. Springer-Verlag, 1986.

[CF00] A. Canteaut and E. Filiol. Ciphertext only reconstruction of stream ciphers based on combination generators. In: Fast Software Encryption 2000, Lecture Notes in Computer Science, number 1978, pages 165–180. Springer-Verlag, 2001.

[CT00] A. Canteaut and M. Trabbia. Improved fast correlation attacks using parity-check equations of weight 4 and 5. In Advances in cryptology - EUROCRYPT '00, volume 1807 of Lecture Notes in Comput. Sci., pages 573-588. Springer, Berlin, 2000.

[FMI07] M. Fossorier, M. Mihaljevic, and H. Imai. Modeling block decoding approaches for the fast correlation attack. IEEE Trans. Inform. Theory, 54(12):4728-4737, 2007.

[GN95] R. G¨ottfert and H. Niederreiter. On the minimal polynomial of the product of linear recurring sequences. Finite Fields and Their Applications, 1(2):204–218, 1995.

[Her85] T. Herlestam. On functions of linear shift register sequences. In: Advances in Cryptology - EUROCRYPT ’85, Lecture Notes in Computer Science, number 219, pages 119–129. Springer-Verlag, 1986.

[LH04] P. Lu and L. Huang. A new correlation attack on lfsr sequences with high error tolerance. In Coding, Cryptography and Combinatorics, volume 23 of Progress in Computer Science and Applied Logic, pages 67-83. Birkhauser, Basel, 2004.

[LN83] R. Lidl and H. Niederreiter. Finite fields. Cambridge University Press, 1983.

[LV04] Yi Lu and Serge Vaudenay. Faster correlation attack on Bluetooth keystream generator E0. In Advances in Cryptology - CRYPTO 2004, volume 3152 of Lecture Notes in Computer Science, pages 407–425. Springer-Verlag, 2004.

[Mas69] J.L. Massey. Shift-register synthesis and BCH decoding. IEEE Transactions on Information Theory, vol. 15, pp. 122–127, 1969.

[MS88] W. Meier and O. Staffelbach. Fast correlation attacks on stream ciphers. In Advances in Cryptology - EUROCRYPT’88, LectureNotesinComputerSciencenumber 330, pages 301–314. Springer-Verlag, 1988.

[MS89] W. Meier and O. Staffelbach. Fast correlation attack on certain stream ciphers. J. Cryptology, pages 159–176, 1989.

[Sie85] T. Siegenthaler. Decrypting a class of stream ciphers using ciphertext only. IEEE Trans. Computers, C-34(1):81–84, 1985.

[RS87] R.A. Rueppel and O.J. Staffelbach. Products of linear recurring sequences with maximum complexity. IEEE Transactions on Information Theory, 33(1):124–131, 1987.

[Rue86] R.A. Rueppel. Analysis and design of stream ciphers. Springer-Verlag, 1986.

[XM88] G.Z. Xiao and J.L. Massey. A spectral characterization of correlation - immune combining functions. IEEE Trans. Inform. Theory, 34(3):569-571, 1988.

[Živk90] Miodrag Živković, Prilog analizi linearnog rekurentnih nizova u polju GF(2), 1990.