Univerzitet u Beogradu

Matematički fakultet

Rotacioni napad na šifre ARX

Teodora Macanović

Master rad

Beograd, 2022.

Mentor:

dr Miodrag Živković, redovni profesor u penziji
Univerzitet u Beogradu, Matematički fakultet

Članovi komisije:

dr Filip Marić, vanredni profesor
Univerzitet u Beogradu, Matematički fakultet

dr Saša Malkov, vanredni profesor
Univerzitet u Beogradu, Matematički fakultet

Datum odbrane:

Mojim roditeljima, baki Koviljki i bratu Nemanji

Rezime: U ovom radu analizira se bezbednost sistema koji se baziraju samo na operacijama modularnog sabiranja, rotacije i XOR-a (ARX sistemi). Nudi se kako teorijska podrška njihovoj bezbednosti tako i praktična kriptoanaliza pravih ARX primitiva. Koristi se tehnika rotacione kriptoanalize, koja je univerzalna za ARX sisteme i prilično efikasna. Rotaciona kriptoanaliza je probabilistički napad, a veruje se da se verovatnoća uspeha rotacione kriptoanalize u odnosu na ARX šifre i funkcije, može izračunati samo brojanjem broja sabiranja. U ovom radu, pokazala sam da je ova jednostavna formula netačna zbog nevažeće pretpostavke Markovljeve šifre koja se koristi za izračunavanje verovatnoće. Tačnije, pokazala sam da ulančana modularna sabiranja koja se koriste u ARX šiframa ne formiraju Markovljev lanac, tako da se rotaciona verovatnoća ne može izračunati kao jednostavan proizvod rotacionih verovatnoća pojedinačnih sabiranja. Dala sam preciznu vrednost verovatnoće takvih lanaca i novi algoritam za izračunavanje rotacione verovatnoće ARX šifara, na osnovu [4]. Metodu sam ilustrovala najpoznatijim napadom na originalnu verziju blok šifre Threefish (jezgro Skeina), kao i napadima na Skein i BLAKE.

Ključne reči: rotaciona kriptoanaliza, Markovljevi lanci, rotaciona verovatnoća, rotaciono svojstvo, rotacioni parovi, Skein, Threefish, BLAKE, SPECK

Sadržaj

1 Uvod

Informaciona bezbednost je praksa zaštite informacija i obezbeđivanja komunikacija u prisustvu ”neprijatelja”. Kriptografija je osnovni alat koji se koristi za dizajniranje bezbednosnih protokola u komunikacionom sistemu. Primenjuje se za postizanje poverljivosti, integriteta ili autentifikacije. Na primer, kriptografija sprečava neovlašćeno otkivanje i korišćenje informacija šifrovanjem istih, odnosno, transformacijom informacija u ono što prividno deluje kao besmislica. Da bi se povratila originalna informacija, neophodno je posedovati ključ – tajnu vrednost koju poznaje samo ovlašćeno lice. Sistemi šifrovanja su klasifikovani u dve glavne grupe: simetrični i asimetrični. U simetričnim kriptosistemima, isti tajni ključ se koristi za transformaciju (šifrovanje) i rekonstrukciju originalne informacije (dešifrovanje). Kod asimetričnih sistema koristi se javni ključ za šifrovanje i privatni za dešifrovanje informacija. Privatni ključ je jednoznačno određen javnim ključem, ali algoritam njegovog određivanja na osnovu javnog ključa zahteva izvršavanje neprihvatljivo kompilovanog računanja.

Kriptografija se široko koristi u našem svakodnevnom životu – kriptografski algoritmi se koriste svaki put kad koristimo veb-pregledač da bismo poslali elektronsku poštu, proverili bankovni račun ili kupili nešto onlajn.

Zbog napretka tehnologije, elektronski uređaji su postali toliko jeftini i mali da su ugrađeni u svakodnevne predmete. Kao rezultat toga, internet se razvija u mrežu „pametnih objekata"koji međusobno komuniciraju. Smatra se da je na internet priključeno preko 20 milijardi uređaja. Osnovna funkcija ovih uređaja je prikupljanje i prenošenje informacija. U nekim slučajevima, prikupljaju se osetljive informacije kao što su zdravstveni ili biometrijski podaci, te stoga postoji velika potražnja za implementacijom kriptografskih algoritama u ove uređaje. Međutim, neki od ovih novih uređaja imaju tako ekstremna ograničenja u računarskoj snazi, površini čipa ili memoriji da nisu dovoljno moćni da koriste iste kriptografske algoritme kao standardni računari. Primer ovih tipova uređaja su RFID čipovi i senzorske mreže. Iz tog razloga, mnogi kriptografski algoritmi prilagođeni takvim ograničenim okruženjima su implementirani i objavljeni tek relativno skoro. Neki od algoritama koriste samo tri vrste operacija: modularno sabiranje, ciklično rotiranje bitova i bitovsko ekskluzivno ili (u daljem tekstu XOR).

Posebna vrsta simetričnih kriptosistema su blok šifre. Blok šifra sa n-bitnim blokom i k-bitnim ključem je simetrični kriptosistem gde šifrovanje uzima n-bitni blok kao otvoreni tekst i k-bitni blok kao ključ i proizvodi n-bitni blok kao šifrat. Blok šifre su među najviše proučavanim kriptografskim algoritmima. Blok šifra treba da postigne visoki stepen konfuzije (složenost odnosa između šifrovanog i otvorenog teksta) i visok stepen difuzije (uticaj svakog bita otvorenog teksta i svakog bita ključa na šifrovani tekst). Više modernih blok šifri su iterirane šifre, gde se šifrovanje i dešifrovanje zasnivaju na iteraciji fiksne funkcije.

Blok šifre koje koriste samo modularno sabiranje (sabiranja po modulu 2^t, gde je t veličina bloka), cikličnu rotaciju i XOR (sabiranje po modulu 2, bit po bit) nazivamo ARX blok šiframa. Neka od važnijih svojstava šifara ARX[1]:

1. Bilo koja funkcija može biti izražena kao kompozicija modularnog sabiranja, rotacije, XOR-a i jedne konstante

2. Šifre ARX se mogu efikasno softverski realizovati, što je posebno važno kada se realizuju na malim procesorima

3. Šifre ARX je lako opisati, što rezultira u implementacijama sa kratkim kodovima

Popularnost strukture proizilazi iz činjenice da se korišćenjem samo tri operacije može postići dobra konfuzija i difuzija. Primeri kriptografskih primitiva zasnovanih na ARX-u [3] uključuju protočne šifre ChaCha i Salsa 20 i blok šifre kao što je TEA, XTEA i Speck.

Kriptografija je naučna disciplina takve prirode da njeni mehanizmi često zahtevaju funkcije koje za dati elemenat domena lako izračunavaju element kodomena, ali ne dozvoljavaju da se lako uradi obratno. Heš funkcije predstavljaju neki vid jednosmernih funkcija.

Definicija 1: Heš funkcija je funkcija koja za ulaz uzima proizvoljno dugačak dokument D, i za njega kao izlaz vraća niz simbola S. Ukoliko se radi na binarnom skupu heš funkciju bismo zapisali kao $H : \{0,1\}^m \rightarrow \{0,1\}^t$, gde je $m \geq t$.

Ulazni podaci nad kojima se primenjuje heš funkcija se nazivaju poruka, a vrednost koja se dobija nakon primene funkcije se zove heš vrednost. Kriptografske heš funkcije se primenjuju u savremenim telekomunikacijama gde je zaštita podataka od zlonamernih napadača od ključne važnosti. Cilj je na neki način osigurati komunikaciju učesnika ako se zna da napadač može imati pristup podacima koji se razmenjuju tokom komunikacije. U polju kriptografije, heš ima primenu u procesu autentifikacije korisnika, provere integriteta poruke, potvrdi lozinke, digitalnom potpisivanju dokumenata i generisanju statistički slučajnih nizova podataka. Takođe se koriste da bi se ubrzalo traženje stavki u bazama podataka, detektovanje dupliranih ili sličnih vrednosti u velikom fajlu i pronalaženje sličnih segmenata u DNK sekvenci. Heš funkcija treba da bude deterministički odredjena, tj. kada se dva puta pozove nad identičnim podacima (npr. dve niske koje sadrže potpuno iste karaktere), funkcija treba da proizvede istu vrednost. To je od ključnog značaja za ispravnost gotovo svih algoritama na osnovu heširanja. Heš funkcije nisu 1-1, pa inverz nije jednoznačno određen, što znači da nije moguće rekonstruisati ulaznu vrednost x samo iz njene heš vrednosti $h(x)$. Jak heš algoritam može brzo da generiše heš iz ulaznih podataka i da pritom bude praktično nemoguće napadačima da otkriju originalne ulazne podatke na osnovu poznavanja vrednosti izlaznog heša. Verovatnoća kolizije, gde se za dve ili više različitih poruka generiše identičan heš na izlazu, mora biti zanemarljivo mala. Svaka promena ulazne poruke, ma koliko bila mala, treba da izazove drastičnu promenu izlaznog heša. Za primenu u kriptografiji, glavne osobine koje bi idealna heš funkcija trebalo da poseduje su sledeće [8]:

- izračunavanje heš funkcije od proizvoljnog dokumenta D treba biti jednostavno i brzo (u linearnom vremenu)

- izlaz konstantne dužine za svaki ulaz proizvoljne dužine

- inverz heš funkcije treba biti praktično nemoguće izračunati (moguće u eksponencijalnom vremenu). Precizno rečeno, za dati niz S kao rezultat heš funkcije H teško je pronaći dokument D za koji važi da je $H(D) = S$.

- Najčešće je neophodno da heš funkcija bude otporna na sudaranje (collision resistant), tj. da je "teško" pronaći dva različita dokumenta D1 i D2 takva da vrednosti njihovih odgovarajućih heš funkcija budu jednake (u teoriji moguće jer je skup domena kod heš funckija mnogo veći od skupa kodomena jer su elementi domena nizovi karaktera mnogo veće dužine nego nizovi karaktera kodomena, mada u praksi se heš funkcije mogu birati

tako da je drugi original kome odgovara slika koju heš funkcija daje praktično nemoguće naći)

- I najmanja izmena dokumenta nad kojim se izvršava heš funkcija, pod time se podrazumeva čak i izmena samo jednog bita, daje značajno drugačiji rezultat nakon heširanja.

Stoga, nije lako napraviti novu kriptografsku heš funkciju. Ona bi trebalo da se ponaša što je više moguće kao random funkcija, ali da i dalje bude deterministička. Primer kriptografskih heš funkcija su Skein i BLAKE.

Kriptoanaliza je disciplina koja analizira bezbednost kriptografskih metoda pronalaženjem slabosti i napada. Glavni cilj napada na blok šifru je da se dobije tajni ključ. Kriptoanalitičke tehnike se mogu klasifikovati prema mogućnostima napadača. U ovom radu se razmatra napad sa izabranim otvorenim tekstom, gde napadač može izabrati veći broj otvorenih tekstova i zahtevati njihove odgovarajuće šifrate.

Kao i kod mnogih drugih struktura, diferencijalna kriptoanaliza i linearna kriptoanaliza predstavljaju dva glavna pristupa analizi šifara ARX. Pored diferencijalne kriptoanalize, koja je jedan od najmoćnijih napada na blok šifre, a čiji će pregled biti dat u narednoj glavi, sve je popularnija i rotaciona kriptoanaliza.

Rotaciona kriptoanaliza predstavlja probabilistički napad koji prati evoluciju takozvanih rotacionih parova kroz runde blokovske šifre. To je kriptoanalitička metoda koja ima za cilj pronalaženje prepoznatljivih statističkih svojstava u ARX šiframa. Osnovna ideja je da i operacije rotacije bita i operacije XOR-ovanja čuvaju korelacije između parova ulaza povezanih bitskom cikličnom rotacijom, a da modularno sabiranje rotiranih ulaza delimično čuva korelacije bitova.

Ipak, i ova metoda ima svoje nedostatke. Iako se smatralo da se verovatnoća uspeha rotacione kriptoanalize protiv šifara i funkcija zasnovanih na modularnom sabiranju, rotacijama i XOR-ovima računa prebrojavanjem broja sabiranja, ispostavlja se da je korišćena pogrešna pretpostavka o nezavisnosti *(Markov cipher assumption)* prilikom izračunavanja verovatnoće. Pretpostavka nezavisnosti ne važi kada je izlaz modularnog sabiranja direktno vodi na ulaz drugog modularnog sabiranja. Ovakav dogadjaj naziva se „lančano modularno sabiranje “ *(chained modular additions)*. Pored toga, analiza se komplikuje ako se u šifri ARX koriste konstante. Na primer, dizajneri blok šifre SEA tvrde da je njihova konstrukcija otporna na rotacionu kriptoanalizu zbog nelinearnog algoritma za proširivanje ključa i korišćenja pseudo-slučajnih konstanti.

Kada struktura ARX uključuje konstante, ona se zove ARX-C. Dokazano [3] je da je ova struktura potpuna, tj. da se bilo koja funkcija može implementirati kroz ARX-C konstrukciju. Dokazano [3] čak i da su AR sistemi, oni sistemi koji ne koriste XOR, teorijski ekvivalentni ARX sistemima, no da su ipak manje sigurni. Kako je XOR može realizovati kombinacijom operacija sabiranja i rotacija, AR sistemima je moguće predstaviti isti skup funkcija kao ARX. Kao što je u [3] pokazano, AR šema može se aproksimirati linearnom funkcijom sa verovatnoćom 2^{-q}, gde je q broj sabiranja u šemi. Takođe, zna se [3] i da su XR i AX sistemi nestabilni, odnosno neotporni na napad.

U ovom radu se bezbednost sistema ARX ispituje upravo tehnikom rotacione analize, praćenjem propagiranja rotacionog para $(X, X \ggg_r)$, gde $X \ggg_r$ označava rezultat rotacije X za r pozicija udesno (ka bitima niže vrednosti). Operacije XOR i rotacija „čuvaju"rotacioni par sa verovatnoćom 1 - sve konstante korišćene u ARX konstrukciji zadržavaju svoje vrednosti nakon

primene ovih operacija, dok modularno sabiranje čini to sa verovatnoćom do $\frac{3}{8}$, u zavisnosti od veličine r – broja rotacija. Zbog toga se ulazni rotacioni par transformiše u izlazni rotacioni par sa verovatnoćom koja zavisi isključivo od broja modularnih sabiranja.

U drugoj glavi ovog rada, razmatra se diferencijalna kriptoanaliza, zbog sličnosti pristupa sa rotacionom. Opis rotacione kriptoanalize dat je u trećoj glavi, a njene primene na neke poznatije algoritme date su u četvrtoj. U petoj glavi, opisan je algoritam SPECK i ArxPy tool. U šestoj glavi, dat je zaključak.

2 Teorijske osnove diferencijalne kriptoanalize

Diferencijalna kriptoanaliza, koju su uveli Biham i Shamir, je probabilistički napad otvorenog teksta. Primenjuje se protiv šifara čija diferencijalna svojstva odstupaju od onih koje se očekuju za slučajnu šifru. Za proizvoljnu šifru, možemo odrediti kolekciju ulaznih razlika (u odnosu na tekst koji se šifruje), i odgovarajućih izlaznih razlika (šifrovan tekst).

Neka je dat sistem sa ulazom $X = [X_1 X_2 \dots X_n]$ i izlazom $Y = [Y_1 Y_2 \dots Y_n]$ i neka su dva ulaza u sistem X' i X'', a odgovarajući izlazi Y' i Y''. Razlika između ulaza X' i X'' je bitski vektor $\alpha = X' \oplus X''$, koji ima jedinice tačno na pozicijama na kojima se razlikuju X' i X'':

$$\alpha = [\Delta X_1 \Delta X_2 \dots \Delta X_n],$$

gde je $\Delta X_i = X'_i \oplus X''_i$, a X'_i i X''_i su i-ti bitovi X', odnosno X''.
Slično je i $\beta = Y' \oplus Y''$ izlazna razlika:

$$\beta = [\Delta Y_1 \Delta Y_2 \dots \Delta Y_n],$$

gde je $\Delta Y_i = Y'_i \oplus \Delta Y''_i$.

Ako se šifra ponaša savršeno statistički nepredviljivo, verovatnoća da se za zadatu razliku ulaza α pojavi fiksirana razlika izlaza β je $\frac{1}{2^n}$, gde je n broj bitova bloka X. Ideja diferencijalne kriptoanalize je da pokušamo da identifikujemo par bitovskih vektora (α, β) koji se pojavljuje sa dosta većom verovatnoćom od $\frac{1}{2^n}$. Ovaj par nazivamo diferencijalom.

Ovo svojstvo omogućava razlikovanje šifre od neke nasumične permutacije i u mnogim slučajevima može primenom posebnog postupka dovesti do rekonstrukcije tajnog ključa (ili potpunog razbijanja šifre).

Nalaženje diferencijala koji maksimizira verovatnoću je zapravo glavni cilj diferencijalne kriptoanalize, a ujedno i najteži zadatak.

Za izabranu ulaznu razliku, razlika šifrovanog teksta se može naći propagiranjem razlike otvorenog teksta kroz funkciju runde. Mnoge šifre su iterirane, odnosno sastoje se od ponavljajuće primene neke nelinearne funkcije koja se izvršava u rundama Y = f(X, Z_i), gde je X stanje na početku runde, Z_i je ključ koji se koristi u rundi i, a Y je izlazno stanje. Početna razlika otvorenog teksta, α, se propagira rundu po rundu, i nakon r rundi (za r-rundnu šifru), može se dobiti izlazna razlika šifrovanog teksta, β. Evolucija razlika nastalih nakon svake runde naziva se diferencijalna karakteristika, a može se reprezentovati nizom:

$$\alpha = \Delta Y(0), \Delta Y(1), \dots \Delta Y(r) = \beta,$$

gde je $\Delta Y(i)$ razlika na izlazu iz i-te runde.

Efikasnost diferencijalne kriptoanalize je usko povezana sa verovatnoćom diferencijala (diferencijalne karakteristike) – što je veća verovatnoća, to je manja složenost napada. Laj i Masi (Lai i Massei) [11], stavljaju fokus na verovatnoću diferencijalnih karakteristika i proučavaju uslove koji neophodno moraju biti ispunjeni, da bi diferencijalne karakteristike formirale Markovljev lanac.

2.1 Markovljev lanac

Procesi Markova predstavljaju jednostavnu i izuzetno korisnu klasu slučajnih procesa, koja nam pomaže da opišemo i modeliramo i najkomplikovanije procese u realnom životu. Jednostavna struktura ovih procesa omogućava nam da uz minimalnu količinu početno poznatih informacija modeliramo i predvidimo uslovne raspodele i buduće ponašanje nekih sistema. Ime potiče od Andreja Markova, ruskog matematičara, koji se bavio istraživanjima u ovoj grani matematike. Zadovoljavati svojstvo Markova znači da sem od trenutnog stanja, buduće stanje sistema ne zavisi od prošlih. To tehnički znači da na buduće stanje procesa utiče informacija, u potpunosti sadržana u sadašnjem stanju – pored date sadašnjosti, budućnost ne zavisi od prošlosti.

Navedimo primer – slučajna šetnja po brojevnoj osi gde se, pri svakom koraku, pozicija menja za 1 (jednako verovatno i u jednom i u drugom smeru). Sa svake pozicije postoje dva moguća i jednako verovatna prelaza: na sledeći ili na prethodni ceo broj. Verovatnoće prelaza tada zavise samo od trenutne pozicije, a ne od načina kako se do njega došlo. Recimo, ako je trenutna pozicija -5, prelaz u -4 ima verovatnoću $\frac{1}{2}$ bez obzira na prethodne pozicije. U svakom trenutku sistem, na osnovu date raspolodele slučajne promenljive može promeniti stanje ili ostati u istom. Promene stanja nazivamo prelazima, a verovatnoće, koje se odnose na različite promene stanja, nazivamo verovatnoćama prelaza.

Markovljevi lanci primenjuju se u stvarnom svetu – koriste se za izučavanje kontrolnih sistema tempomata u motornim vozilima. Osnovna funkcije tempomata je u suštini kontrolisanje brzine kretanja. Pored toga, Markovljevi lanci koriste se za utvrđivanje trendova redova ili linija putnika koji dolaze na aerodrom, kao i za varijacije deviznih kurseva i populacione dinamike životinja [15], [16], [17], [18].

Definicija 2: Skup svih mogućih ishoda ili rezultata nekog eksperimenta označava se sa Ω i naziva se prostor elementarnih događaja ili prostor ishoda. Elementi ovog skupa označavaju se sa ω_i, gde je $i = 1, 2, ...$ i nazivaju se elementarni događaji.

Definicija 3: Podskup F partitivnog skupa $P(\Omega)$ je σ-polje (*sigma*-algebra) nad Ω ako važe sledeći uslovi:

1. $\Omega \in F$
2. $A \in F \implies \overline{A} \in F$
3. $A_{i,i \in N} \subseteq F \implies \bigcup_{i=1}^{\infty} A_i \in F$

Jedno od svojstava σ-algebre jeste da nju ine merljivi skupovi. Takodje par (Ω, F) se naziva merljiv prostor.

Definicija 4: Ako je Ω skup svih elementarnih događaja i ako je F σ-polje nad skupom Ω, tada se funkcija $\Pr : F \to [0, 1]$ zove verovatnoća na prostoru (Ω, F) ako su ispunjeni sledeći uslovi:

1. $\Pr(\Omega) = 1$

2. $\Pr(A) \geq 0,\ \forall A \in F$

3. $A_{i,i\in N} \subseteq F, A_i \cap A_j = \varnothing,\ i \neq j,\ i = 1, 2... \implies \Pr\left(\sum_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty}\left(P(A_i)\right)$

Definicija 5: Uređena trojka $(\Omega, F, \Pr)$ naziva se prostor verovatnoća.

Pretpostavimo da analiziramo neki slučajni proces $X_t, t \in T$, gde je X_t slučajna veličina definisana na nekom prostoru verovatnoća $(\Omega, \mathcal{F}, P)$, parametar t označava vreme, a T podskup realne ose. Tada, za ovaj proces kažemo da poseduje Markovljevo svojstvo, ukoliko verovatnosna struktura procesa u budućnosti ne zavisi od njegove predistorije, već zavisi samo od sadašnjeg trenutka (u kom se proces trenutno nalazi). U nastavku dodatno pretpostavljamo da sistem posmatramo u vremenskim trenucima $n \in N_0$, i u skladu sa tim slučajni proces označavamo sa $X_n, n \geq 0$. Takođe, ubuduće ćemo uvek pretpostaviti da X ne zavisi od potključeva $Z^{(1)}, \ldots Z^{(n)}$.

Definicija 6: Diskretan slučajni proces $X_n, n \geq 0$ je Markovljev lanac ako za svaki trenutak $n \geq 0$ i za sva stanja $i_0, i_1, \ldots . i, j \in S$ važi sledeća jednakost:

$$\Pr(X_{n+1} = j | X_n = i, X_{n-1} = i_{n-1} \ldots X_0 = i_0 = \Pr X_{n+1} = j | X_n = i).$$

Pri fiksiranom stanju i u momentu n dalje ponašnanje procesa ne zavisi od toga na koji način je proces došao u stanje i i od toga kroz koja stanja je proces prolazio do trenutka n. Neka

$$p_{ij}(n) = \Pr(X_{n+1} = j | X_n = i),$$

označava verovatnoću prelaska sistema iz stanja i u stanje j. U skladu sa tim matricu svih verovatnoća prelaska označavamo sa P, gde je $P = [p_{ij}]_{|Z|\times|Z|}$, gde je Z prostor stanja - dimenzije matrice verovatnoće prelaska zavise od broja svih mogućih stanja u kojem sistem može da se nađe. U opštem slučaju matrica svih verovatnoća prelaska $P = [p_{ij}]_{|N_0|\times|N_0|}$ ima sledeći oblik:

$$P = \begin{pmatrix} p_{00} & p_{01} & \ldots \\ p_{10} & p_{11} & \ldots \\ \vdots & \ddots & \ldots \end{pmatrix}$$

Za svaki homogeni Markovljev lanac $X_n, n \geq 0$ važi da im je matrica verovatnoća prelaska P stohastička, tačnije da za nju važe svojstva:

- $p_{ij} \geq 0$, za sve $i, j \in Z$,
- $\sum_j p_{ij} = 1$, za sve $i \in Z$.

Prethodno opisane verovatnoće prelaska, odnose se na verovatnoće prelaska iz stanja i u stanje j u jednom koraku.

Definicija 7: Ako je dat homogen slučajni proces $\{X_n, n \geq 0\}$, sa prostorom stanja Z, tada se matrica $P_n = [p_{ij}^{(n)}]_{|Z|\times|Z|}$, čiji su članovi određeni jednakostima:

$$p_{ij}^{(n)} = \Pr\{X_{m+n} = j | X_m = i\}, i, j \in Z,$$

naziva matrica verovatnoće prelaska iz stanja i u stanje j za n koraka.
Dodatno primećujemo da važi $P_1 = P$.

Definicija 8: Markovljev lanac je homogen ako je $\Pr(X_{i+1} = \beta | X_i = \alpha)$ nezavisno od i za sve izbore α i β.

Definicija 9: Markovljev lanac se naziva erodičnim, ako je moguće preći iz bilo kog stanja u bilo koje stanje (ne obavezno u jednom koraku).

Definicija 10: Markovljeva šifra je iterirana šifra sa rundnom funkcijom $Y = f(X, Z)$, ako za sve n-torke α i β različite od nula vektora važi da je $\Pr(\Delta Y = \beta | \Delta X = \alpha, X = \gamma)$ nezavisno od vrednosti γ kada je potključ Z uniformno slučajan.

Teorema 1: Ako je sa r rundi iterirana šifra Markovljeva šifra i potključevi u r-toj rundi su nezavisni i uniformno nasumični, tad je niz razlika $\Delta Y(0), \Delta Y(1), \ldots \Delta Y(r)$ homogeni Markovljev lanac.

Dokaz: Da se dokaže da je niz $\Delta Y(0), \Delta Y(1), \ldots \Delta Y(r)$ Markovljev homogeni lanac, dovoljno je dokazati za drugu rundu da je:

$$\Pr(\Delta Y(2) = \beta_2 | \Delta Y(1) = \beta_1, \Delta Y(0) = \alpha) = \Pr(\Delta Y(2) = \beta_2 | \Delta Y(1) = \beta_1).$$

Da bismo ovo dokazali, primećujemo da:

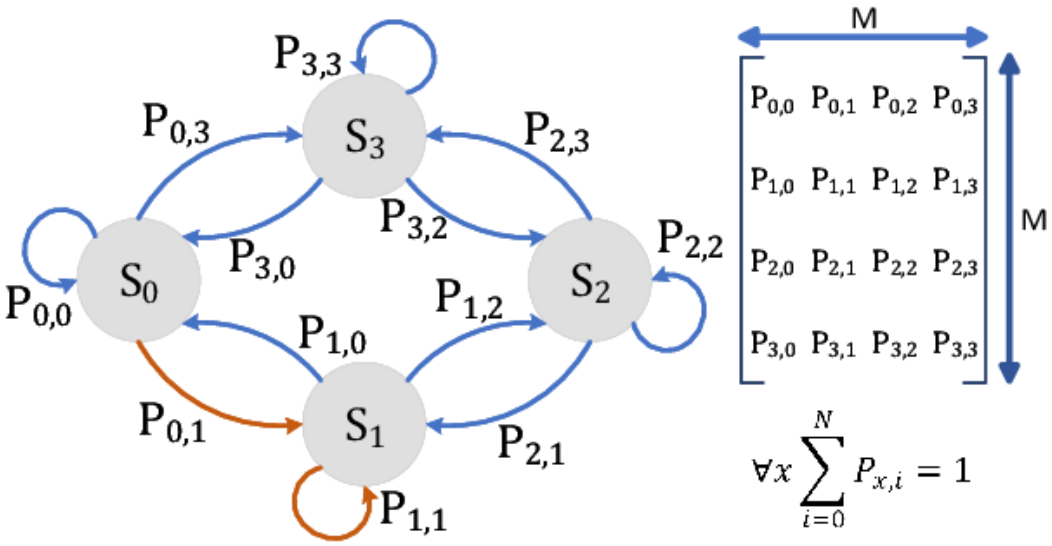
$$\begin{aligned}
&\Pr(\Delta Y(2) = \beta_2 | \Delta Y(1) = \beta_1, \Delta Y(0) = \alpha) \\
&= \sum_{\gamma} \Pr(Y(1) = \gamma, \Delta Y(2) = \beta_2 | \Delta Y(1) = \beta_1, \Delta Y(0) = \alpha) \\
&= \sum_{\gamma} \Pr(Y(1) = \gamma | \Delta Y(1) = \beta_1, \Delta Y(0) = \alpha) \Pr(\Delta Y(2) = \beta_2 | \Delta Y(1) = \beta_1, Y(1) = \gamma, \Delta Y(0) = \alpha) \\
&= \sum_{\gamma} \Pr(Y(1) = \gamma | \Delta Y(1) = \beta_1, \Delta Y(0) = \alpha) \Pr(\Delta Y(2) = \beta_2 | \Delta Y(1) = \beta_1, Y(1) = \gamma) \\
&= \sum_{\gamma} \Pr(Y(1) = \gamma | \Delta Y(1) = \beta_1, \Delta Y(0) = \alpha) \Pr(\Delta Y(2) = \beta_2 | \Delta Y(1) = \beta_1) \\
&= \Pr(\Delta Y(2) = \beta_2 | \Delta Y(1) = \beta_1),
\end{aligned}$$

gde treća jednakost proizilazi iz činjenice da $Y(1)$ i $\Delta Y(1)$ zajedno određuju $Y(1)$ i $Y(1)'$, pa $\Delta Y(2)$ nema dalje uticaj na $Y(0)$, kada su $Y(1)$ i $\Delta Y(1)$ određeni. S obzirom na to da se za svaku rundu koristi ista funkcija runde, Markovljev lanac je homogen.

■

Drugim rečima, ako je šifra Markovljeva, onda je verovatnoća diferencijalne karakteristike jednaka proizvodu verovatnoća pojedinačnih diferencijala dobijenih po rundama (jer one formiraju Markovljev lanac), pod uslovom da verovatnoće diferencijala ne zavise od vrednosti ulaznog stanja; pretpostavlja se da su rundni ključevi nezavisni i uniformno raspodeljeni.

Markovljeve lance možemo prikazati i usmerenim grafovima, gde su čvorovi pojedinačna stanja, grane odgovaraju prelazima iz jednog stanja u drugo, a vrednosti napisane na granama predstavljaju verovatnoće prelaza (u određenom smeru), kao na slici 1. Data su 4 stanja S_0, S_1, S_2, S_3 i prelazi između njih. Vrednost $P_{i,j}$ predstavlja verovatnoću prelaza iz stanja i u stanje j. Sa strane je data matrica verovatnoća prelaska P, iz koje možemo dobiti svaku verovatnoću.



Slika 1: Markovljev lanac i matrica verovatnoće prelaska.

3 Rotaciona kriptoanaliza

Definicija 11: Neka je X n-bitni vektor. Operacije rotacije u okviru reči označavamo sa $\lll_r$ i $\ggg_r$, pa se onda rotirana reč oznacava sa $X_{\lll_r}$ i $X_{\ggg_r}$, gde je $X_{\lll_r}$ rotacija X-a za r bitova ulevo, a $X_{\ggg_r}$ rotacija X-a za r bitova udesno.

Definicija 12: Neka su X i $X_{\lll_r}$ dva ulazna vektora. Tada par vektora $(X, X_{\lll_r})$ nazivamo rotacionim parom sa rotacijom r.

Definicija 13: Rotaciona kriptoanaliza je probabilistički napad koji koristi rotacione "pomeraje", tj. rotacione parove reči $(X, X_{\lll_r})$, gde je X proizvoljno.

Definicija 14: Rotaciona verovatnoća je verovatnoća da će rotacioni par nakon primene neke operacije sačuvati odnos bitske rotiranosti sa nekom verovatnoćom; količina rotacije pri tome ne mora da ostane ista.

Par $(X, X_{\lll r})$ za $X \in \{0,1\}^{m\times n}$, gde je $X = (X_1, X_2, \dots X_m)$ naziva se rotacionim parom sa rotacijom r, gde je $X_{\lll r} = (X_1 \lll r, \dots X_m \lll r)$. Neka je data funkcija $F : \{0,1\}^n \to \{0,1\}^n$ i ulazni rotacioni par $(X_1, X_1 \lll r), X_1 \in \{0,1\}^n$. Kaže se da F čuva svojstvo rotacije ako je zadovoljeno da važi $F(X_1)_{\lll r} = F(X_1 \lll r)$. Stoga, F čuva svojsto rotacije ako je za $(X_1, X_1 \lll r)$ par $(F(X_1), F(X_1 \lll r))$ rotacion. Sistem $\Phi(X) = \{0,1\}^{m\times n} \to \{0,1\}^{k\times n}$ koji se sastoji od niza transformacija $F_1, \dots, F_k : \{0,1\}^{m\times n} \to \{0,1\}^n$, na primer $\Phi = (F_1, \dots, F_k)$ čuva rotaciono svojstvo ako proizvodi rotacioni izlazni par za ulazni rotacioni par. Postoji samo nekoliko transformacija koje čuvaju rotaciono svojstvo za bilo koji ulazni par i u većini slučajeva, za slučajni ulaz X, uslov $F(X)_{\lll r'} = F(X_{\lll r})$ važi sa verovatnoćom p_F. Ovu verovatnoća se naziva rotaciona verovarnoća funkcije F i zavisi od r. Ako se pretpostavi da su izlazi iz ovih transformacija nezavisni, tada sistem Φ, koji je kompozicija transformacija $F_1, \dots F_k$ čuva rotaciono svojstvo sa verovatnoćom $p_\Phi = p_{F_1} \cdot p_{F_2} \cdot \dots \cdot p_{F_k}$. Stoga, da bi se našla verovatnoća da sistem čuva rotaciono svojstvo, potrebno je naći verovatnoće čuvanja rotacionog svojstva pojedinačnih transformacija tog sistema. Za slučajni sistem sa n-bitnim izlazom, verovatnoća da će rotacioni ulaz proizvesti rotacioni izlaz je 2^{-n}. Ako sistem Φ sa n-bitnim izlazom ima rotacionu verovatnoću $p_\Phi > 2^{-n}$, tada se ovaj sistem može statistički razlikovati od slučajnog sistema.

Rotaciona kriptoanaliza prati propagaciju rotacionih parova u izlazima pojedinih rundi kriptografske šeme za bilo koje date rotacione ulazne parove. U diferencijalnoj analizi, za par ulaza (x, y), prati se propagacija razlika $x \oplus y$. Nasuprot tome, kod rotacione kriptoanalize, ispituje se propagacija rotacione (za r, gde je r fiksirana količina rotacije) veze niza parova $(X, X_{\lll_r})$. Dakle, slično diferencijalnoj kriptoanalizi, rotaciona kriptoanaliza koristi prednost velike verovatnoće propragacije para $(X, X_{\lll_r})$ kroz ARX operacije.

Kriptografski algoritmi koji koriste operacije sa n-bitnim rečima: sabiranja po modulu 2^n (u daljem tekstu +), rotacije (u daljem teskstu $\lll_r$) i XOR-ova (u daljem tekstu $\oplus$) nazivaju se ARX primitivama. ARX primitive se sastoje od niza transformacija koje jedno stanje prevode u naredno stanje. Stanje se sastoji od nekoliko reči. Transformacija se izvršava primenom jedne od osnovnih operacija (ARX) na neke dve reči stanja. Da bi rotaciona kriptoanaliza bila efikasna, zahtev koji mora biti zadovoljen je da sve konstante korištene u ARX primitivama zadržavaju svoje vrednosti nakon rotacije za r. Drugim rečima, lako je obezbediti da ARX primitiva bude

otporna na rotacionu kriptoanalizu.

U rotacionoj kriptoanalizi, napadač unosi rotacioni par $(X, X_{\lll_r})$ u osnovni kriptografski algoritam i posmatra ponašanje izlaznih parova. Neka Z i Z' označavaju rezultujući izlazni par za ulaz $(X, X_{\lll_r})$ redom. Ako važi $Z' = Z_{\lll_r}$, onda operacije uključene u kriptografski algoritam sa nekom nezanemarljivom verovatnoćom ne utiču na rotacione odnose u izlaznom paru (Z, Z'). To jest, u tom slučaju, rotacioni odnos je očuvan sa tom verovatnoćom u izlaznom paru (Z, Z') za odgovarajući ulazni par $(X, X_{\lll_r})$. Očigledno, ukoliko se rotacioni odnosi sačuvaju, napadač može da taj odnos uoči u izlaznom paru i na osnovu njega napravi prepoznavač za taj kriptografski algoritam.

U okviru rotacionog napada, polazi se od ulaznog rotacionog para, odnosno dva stanja, takva da u su u prvom stanju sve reči slučajno odabrane, dok se drugo stanje formira rotacijom reči prvog stanja, za r pozicija. Ako su za takve ulazne parove odgovarajući izlazni parovi ARX primitiva takođe rotacioni, sa rotacionom verovatnoćom primitive većom nego za slučajnu funkciju, onda se ovo svojstvo može koristiti za statističko razlikovanje ARX primitive od slučajne funkcije.

Ranije je rečeno da je ideja rotacione kriptoanalize da i operacije rotacije bita i operacije XOR čuvaju veze između parova ulaza koji su bitski rotirani, ali i da sabiranje rotiranih ulaza takođe delimično čuva odnos bitske rotiranosti. Ulazni rotacioni parovi se stoga mogu koristiti da se „previde“ kaskadne operacije šifre ARX u većem stepenu nego što bi se moglo očekivati. Ova sposobnost da se „vide“ veze kroz runde može se zatim iskoristiti za razbijanje šifre na način koji je sličan diferencijalnoj kriptoanalizi - ako se pretpostavi potključ pretposlednje runde, onda se može izračunati par izlaza iz pretposlednje runde i tako prepoznati statistički da li je pogođen pravi potključ.

Slično diferencijalnoj i linearnoj kriptoanalizi, rotaciona kriptoanaliza zahteva da rotaciona verovatnoća ostaje nezanemarljiva kroz runde primitive. Očigledno, verovatnoća ovog događaja zavisi od broja operacija koje mogu da ugroze rotacionu verovatnoću, poput modularnog sabiranja. Rotacione verovatnoće ARX primitiva mogu se naći množenjem pojedinačnih rotacionih verovatnoća svih operacija koje se koriste u primitivi. Kako se ARX sastoji samo od tri različite operacije, polazi se od izračunavanja rotacione verovatnoće sabiranja, rotacije i XOR-a.

Neka je $F_2 = \{0, 1\}$ polje sa dva elementa. Sa x_i označavamo i-ti bit vektora x $\in F_2^n$. Vektor x od n-bitova predstavljamo pomoću sheme x= $(x_{n-1}, \dots x_1, x_0)$. Za vektorsku funkciju $F : F_2^n \to F_2^m$, gde je $y = F(x) \in F_2^m$, njen i-ti izlazni bit y_i označavamo sa $(F(x))_i$. Dodatno, konkretne vrednosti u F_2^n su date u heksadekandom zapisu. Na primer, koristimo zapis 1111 da bismo predstavili binarni string $(0001000100010001)_2$.
Neka je n dužina reči, a x neka n-bitna reč iz skupa reči W. Skup F označava skup svih funkcija u W:

$$F = \{f : W^m \to W | m \in N\}.$$

Kažemo da je skup Q baza u F ako svaka funkcija u F može biti predstavljena kao kompozicija elemenata iz Q.

Teorema 2: Skup funkcija $\{+, \oplus, \lll_r, 1\}$ je baza u F.
Dokaz: Pokazujemo za proizvoljnu funkciju $f \in F$ i X, gde je $X = x_1 x_2 \dots x_n$ da se može realizovati kao kompozicija funkcija $+, \oplus$ i $\lll_r$. Za dokaz ćemo koristiti jednostavne funkcije, za koje se lako pokazuje da se mogu dobiti kompozicijom funkcija ARX.

1. Realizacija funkcije $s_i(X) = 00\ldots0x_i$: Prvo, rotiramo X za $n-i-1$ bit udesno, koristeći $n-i$ rotaciju. Potom, pomonožimo tu vrednost sa dva. Postupak ponovimo $n-1$ put i tako dobijemo $x_i0\ldots0$. Konačno, rotiramo rezultat još jednom i dobijamo $00\ldots0x_i$.

2. Realizacija funkcije $M_k(X,Y) = 00\ldots0(x_ky_k)$: Prvo, odredimo $s_k(X) + s_k(Y)$ i dobijamo $00\ldots0(x_ky_k)(x_k \oplus y_k)$. Potom, rotiramo udesno za jedan bit, a zatim pomnožimo sa 2 i time uklonimo linearni izraz. Konačno, rotiramo udesno, za jedan bit.

3. Realizacija funkcije $j_C(X) = \begin{cases} 00\ldots01, & \text{if } X = C \\ 0, & \text{inače} \end{cases}$.
   Neka je $C = (c_1, c_2 \ldots c_n)$. Tada je $j_C(X) = \prod(x_i \oplus c_i \oplus 1)$, što može da se izračuna funkcijom $\{M_k(X,Y)\}$ i konstantom 1.

4. Realizacija funkcije $J_{C_1,C_2}(X) = \begin{cases} C_2, & \text{if } X = C_1 \\ 0, & \text{inače} \end{cases}$.
   Možemo ih predstaviti kao $C_2 * j_{C_1}(X)$.

5. Realizacije funkcije f kao $\oplus_{X \epsilon W^m} J_{X,f(X)}$.

■

Teorema 3: Bitovsko XOR koje se primenjuje na binarni string X, čuva rotacione parove za svako r, pa sa verovatnoćom 1 važi:

$$X_{\lll r} \oplus Y_{\lll r} = (X \oplus Y)_{\lll r}$$

Dokaz: Neka je $X = [X_L|_i X_R]$ n-bitni string $X = x_{n-1}\ldots x_0$, gde je $X_L = x_{n-1}\ldots x_i$ i $X_R = x_{i-1}\ldots x_0$. Slično, definišemo i string $Y = [Y_L|_i Y_R]$ za n-bitni string $Y = y_{n-1}\ldots y_0$, gde je $Y_L = y_{n-1}\ldots y_i$ i $Y_R = y_{i-1}\ldots y_0$. Tada, za proizvoljnu r-bitnu rotaciju ulevo, gde je $0 \le r \le n$, mozemo da predstavimo X i Y kao $X = [X_L|_{n-r} X_R]$ i $Y = [Y_L|_{n-r} Y_L]$ redom. Sada,

$$\begin{aligned} X_{\lll r} \oplus Y_{\lll r} &= [X_L|_{n-r} X_R]_{r\lll r} \oplus [Y_L|_{n-r} Y_L]_{\lll r} \\ &= [X_R|_r X_L]] \oplus [Y_R|_r Y_L]] \\ &= [(X_R \oplus Y_R)|_r X_L \oplus Y_L] \end{aligned}$$

i

$$\begin{aligned} (X \oplus Y)_{\lll r} &= [X_L|_{n-r} X_R] \oplus [Y_L|_{n-r} Y_R]_{\lll r} \\ &= [(X_L \oplus Y_L|_{n-r}(X_R \oplus Y_R)]_{\lll r} \\ &= [(X_R \oplus Y_R|_r X_L \oplus Y_L)]. \end{aligned}$$

Stoga sa verovatnoćom 1 važi,

$$X_{\lll r} \oplus Y_{\lll r} = (X \oplus Y)_{\lll r}$$

■

Teorema 4: Operacija bitske rotacije, koja se primenjuje na binarni string x, čuva rotacione parove, za svako proizvoljno r_1, r_2, odnosno sa verovatnoćom 1 važi:

$$(x \lll r_1) \lll r_2 = (x \lll r_2) \lll r_1$$

Sledeća lema daje opšti način za izračunavanje verovatnoće propagacije rotacionog para kroz sabiranje po modulu 2^n.

Lema 1: Za $x, y \in F_2^n$, i $0 < r < n$,

$$\Pr((x+y)_{\lll_r} = x_{\lll_r} + y_{\lll_r}) = \frac{1}{4}(1 + 2^{r-n} + 2^{-r} + 2^{-n})$$

Dokaz: Lema će biti dokazana za rotaciju ulevo [14], slučaj za rotaciju udesno se dokazuje analogno. Za n-bitovsku reč x, rotaciju definišemo kao:

$$x_{\lll_r} = (x_{\lll_r}) + (x_{\ggg_{n-r}}).$$

Tada,

$$(x_{\ggg_s})_{\lll_r} = [(x_{\ggg_s})_{\lll_r}] + [(x_{\ggg_s})_{\ggg_{n-r}}],$$
$$(x_{\lll_r})_{\ggg_s} = [(x_{\lll_r}) + (x_{\ggg_{(n-r)}}]_{\ggg_s} = [(x_{\lll_r})_{\ggg_s}] + [(x_{\ggg_{(n-r)}})_{\ggg_s}].$$

Stoga, treba naći verovatnoću da važi

$$L \equiv (x_{\ggg_s})_{\lll_r} = (x_{\lll_r})_{\ggg_s} \equiv R.$$

Neka je x= $x_{n-1} \dots x_0$, gde su x_i redom bitovi od x. Mogući su sledeći slučajevi:

1. Ako je $s \geq r$, tada,

   $$L \equiv (x_{\ggg_s})_{\lll_r} = 0 \dots 0 x_{n-1} \dots x_s 0 \dots 0$$

   (pre bita x_{n-1} pojavljuje se $s-r$ bitova, a posle bita x_s, r bitova)

   $$R \equiv (x_{\lll_r})_{\ggg_s} = 0 \dots 0 x_{n-1-r} \dots x_{s-r}$$

   (pre bita x_{n-1} pojavljuje se s bitova)

2. Ako je $0 \leq r \leq s \leq \frac{n}{2}$, tada je $L = R$ sa verovatnoćom 2^{-2r}: bitovi $x_{n-1}, \dots, x_{n-r}$ (sve zajedno r bitova) i bitovi $x_{s-1}, \dots, x_{s-r}$ (ponovo r bitova) moraju biti nule, a $r = \min(r, s, n-r, n-s)$.

3. Ako je $0 \leq r \leq \frac{n}{2} \leq s \leq n$, tada je $L = R$ sa verovatnoćom $2^{-2\min(r,n-s)}$: ako je $r \leq n-s$ tada imamo situaciju analognu prethodnoj, dok ako važi $n - s < r$, tada bitovi $x_{n-1}, \dots, x_{n-s}$ ($n-s$ bitova ukupno) i bitovi $x_{n-1-r}, \dots, x_{s-r}$ ($n-s$ bitova ukupno) moraju biti nule, a $n - s = \min(r, s, n-r, n-s)$.

4. Ako je $\frac{n}{2} < r \leq s \leq n$, tada je $n - 1 - r < \frac{n}{2}$, pa je zato verovatnoća da je $L = R$ $2^{-2(n-s)}$: bitovi x,... x (ukupno $n-s$ bitova) i bitovi x,... ,x (ukupno $n-s$ bitova) moraju biti nule, a $n - s = \min(r, s, n-r, n-s)$.

5. Slučaj kada je $s < r$ se očigledno svodi na prethodni slučaj.

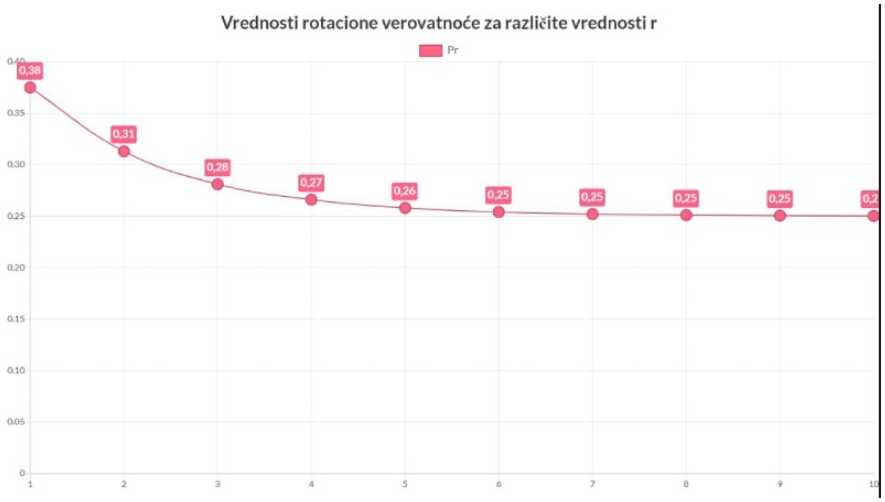
■

Za $r = \frac{n}{2}$, verovatnoća je blizu $\frac{1}{4}$. Isto važi za rotaciju udesno.

Za veliko n i malo r dobijamo sledeću tabelu i prateći grafik:

r	p_r	$\log_2 p_r$
1	0,375	-1,415
2	0,313	-1,676
3	0,281	-1,831
4	0,266	-1,911
5	0,258	-1,955
6	0,254	-1,977
7	0,252	-1,989
8	0,251	-1,994
9	0,2505	-1,997
10	0,2502	-1,999

Tabela 1: Vrednosti rotacione verovatnoće modularnog sabiranja za različite vrednosti r.



Grafik 1: Vrednosti rotacione verovatnoće modularnog sabiranja za različite vrednosti r.

Lema 2: Za date n-bitne reči x, y i za pozitivne brojeve r, s važi:

$$\Pr(x_{\lll_r} \cdot y_{\lll_s} = (x \cdot y)_{\lll_{r+s}}) \geq 2^{-2(r+s)}$$

Dokaz: Neka je $x = x_1 \cdot 2^{n-r} + x_2$ i $y = y_1 \cdot 2^{n-s} + y_2$. Tada:

$$(x \cdot y)_{\lll_{r+s}} = (x_1 \cdot y_1 2^{2n-r-s} + x_1 \cdot y_2 2^{n-r} + x_2 \cdot y_1 2^{n-s} + x_2 \cdot y_2)_{\lll_{r+s}},$$
$$x_{\lll_r} \cdot y_{\lll_s} = (x_2 \cdot 2^r + x_1)(y_2 \cdot 2^s + y_1) = x_2 \cdot y_2 \cdot 2^{r+s} + x_2 y_1 \cdot 2^r + x_1 \cdot y_2 \cdot 2^s + x_1 \cdot y_1.$$

Ako je $x_1 = 0$ i $y_1 = 0$ tada se gornja jednačine mogu pojednostaviti do:

$$(x \cdot y)_{\lll_{r+s}} = (x_2 \cdot y_2)_{\lll_{r+s}},$$
$$x_{\lll_r} \cdot y_{\lll_s} = x_2 \cdot y_2 \cdot 2^{r+s}.$$

Kako je $(x_2 \cdot y_2)_{\lll_{r+s}} = x_2 \cdot y_2 \cdot 2^{2r+s}$ sa verovatnoćom 2^{-r-s} i kako važi da je $x_1 = 0$ i $y_1 = 0$ sa istom verovatnoćom, onda je tvrđenje leme dokazano.

■

Razmotrimo sad proizvoljnu šemu S sa sabiranjima, rotacijama i XOR-ovima n-bitne reči. Sledeća teorema važi pod pretpostakom nezavisnosti.

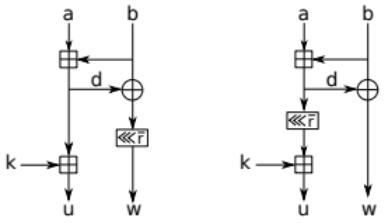
Teorema 5: Neka je q broj sabiranja po modulu u ARX primitivi koji ima proizvoljan broj rotacija i XOR-ova. Tada je rotaciona verovatnoća ARX primitive p_+^q , gde je p_+ rotaciona verovatnoća modularnog sabiranja (koja zavisi od parametra rotacije r i veličine reči n).
Dokaz se izvodi indukcijom po veličini šeme.

Lema 1 sugeriše nam da se rotaciona verovatnoća ARX-a može tačnije i preciznije izračunati ako uzmemo u obzir ne samo broj sabiranja po modulu koji se koristi u toj ARX strukturi, već i njihove pozicije. Grupišemo sabiranja u lance i izračunavamo rotacionu verovatnoću na sledeći način:

1. Pronađemo sve lance sabiranja po modulu (uključujući i one koji su sastavljeni od pojedinačnih sabiranja) u ARX primitivi
2. Za svaki lanac, izračunavamo rotacionu verovatnoću prema lemi 1
3. Za celu primitivu izračunavamo rotacionu verovatnoću kao proizvod rotacionih verovatnoća lanaca

Na osnovu teoreme 5 zaključujemo da da bi se pronašla rotaciona verovatnoća ARX potrebno je izbrojati samo broj sabiranja q. Ako je $p_+^q > 2^{-m}$, gde je m veličina stanja, onda je primitiva podložna rotacionoj kriptoanalizi. Teorema je tačna pod pretpostavkom da je ARX šifra Markovljeva i da se rundni ključevi biraju nezavisno i ravnomerno nasumično. Rotacije i XOR-ovi imaju rotacionu verovatnoću 1 i stoga su nezavisni od ulaza. Kod sabiranja po modulu, situacija je nešto drugačija. Rotaciona verovatnoća modularnog sabiranja je određena lemom 1, dok god su ulazi nasumični. Ako se izlaz nekog sabiranja po modulu uzme kao ulaz sabiranja, onda tada rotaciona verovatnoća drugog sabiranja možda neće poštovati lemu 1.

Problem ilustruju ARX primitive prikazane na slici 2. Svaka od njih ima tri ulaza, a, b, k, dva izlaza u i w, i koristi dva sabiranja po modulu. Ako je veličina rotacije r 1, a veličina reči je 64 bita, onda je prema lemi 1, rotaciona verovatnoća modularnog sabiranja $2^{-1,415}$ i prema teoremi 4, rotaciona verovatnoća obe primitive iznosi $2^{-2,83}$. Imajmo na umu da za iznos rotacije 1 rotaciona verovatnoća modularnog sabiranja jako zavisi od vrednosti najznačajnijih bitova ulaza. Preciznije, zbir najznačajnijih bitova ulaza ne bi trebalo da bude veći od 1.



Slika 2: Dve ARX primitive, sa različito povezanim sabiranjima

U ARX konstrukciji na levoj strani slike, dva modularna sabiranja su realizovana lančano, tj. izlaz prvog je ulaz u drugo.
Najznačajniji bit reči $d = a + b$, kada je

$$(a + b)_{\lll_1} = a_{\lll_1} + b_{\lll_1},$$

je pristrasan prema 1. Stoga, drugo sabiranje po modulu $u = k + d$ ima rotacionu verovatnoću različitu od onog datog lemom 1. Kao rezultat, teorema 5 nam ne daje tačnu verovatnoću.

U ARX konstrukciji na desnoj strani slike, dva modularna sabiranja sabiranja su razdvojena rotacijom. Sada, iako je najznačajniji bit reči d i dalje pristrasan ka 1, rotacijom se ovaj bit pomera na drugu poziciju, gde mu gorepomenuta pristrasnost ne utiče bitno na rotacionu verovatnoću. Šta vise, najniži bit reči $d_{\lll_r}$ postaje potpuno slučajan bit, pa odatle drugo sabiranje po modulu $d_{\lll_r} + k$ ima verovatnoću datu lemom 1. Stoga, u ovom slučaju, teorema 5 daje tačan rezultat.

Ova dva primera sugerišu da rotaciona verovatnoća ARX primitive ne može biti izračunata jednostavnim brojanjem sabiranja po modulu. Umesto toga, treba uzeti u obzir relativne položaje sabiranja, odnosno da li su sabiranja ulančana ili su razdvojena rotacijom. Što je duži lanac sabiranja, to je manja rotaciona verovatnoća za svako uzastopno sabiranje. Rotaciona verovatnoća ulančanih sabiranja po modulu data je sledećom teoremom.

Teorema 6: Lančano modularno sabiranje

Neka su $a_1, a_2, \ldots a_k$ nezavisne slučajne n-bitne promenljive sa ravnomernom raspodelom verovatnoća i neka je r pozitivan ceo broj takav da je $0 < r < n$. Tada:

$$\Pr([a_1 + a_2)_{\lll_r} = a_{1\lll_r} + a_{2\lll_r}])$$

i

$$[(a_1 + a_2 + a_3)_{\lll_r} = a_{1\lll_r} + a_{2\lll_r} + a_{3\lll_r}]$$

i . . .

$$[(a_1 + \cdots + a_k)_{\lll_r} = a_{1\lll_r} + \cdots + a_{k\lll_r}]$$
$$= \frac{1}{2^{nk}}\binom{k+2^r-1}{2^r-1}\binom{k+2^{n-r}-1}{2^{n-r}-1}$$

Dokaz: Razmotrimo rotacionu verovatnoću sabiranja l-sabiraka jednakosti:

$$(a_1 + a_2 + ... + a_k)_{\lll_r} = a_{1\lll_r} + a_{2\lll_r} + \cdots + a_{l\lll_r}. \tag{1}$$

Svaku od n-bitnih reči a_i možemo da tretiramo kao izraz dobijen konkatenacijom dve reči: r-bitne reči x_i i $(n-r)$-bitne reči y_i, takvih da važi:

$$a_i = x_i||y_i,\ |x_i| = r,\ |y_i| = n - r,$$

gde je $||$ oznaka za konkatenaciju.

Tada (1) postaje:

$$(x_1||y_1 + \cdots + x_l||y_l)_{\lll_r} = (x_i||y_i)_{\lll_r} + \ldots (x_l||y_l)_{\lll_r}. \tag{2}$$

S obzirom na to da izraz $(x_i||y_i)_{\lll_r}$ sa desne strane izraza (2) nakon rotacije r bitova postaje

$$(x_i||y_i)_{\lll_r} = y_i||x_i,$$

izraz (2) možemo da zapišemo kao:

$$(x_i||y_i + \cdots + x_l||y_l)_{\lll_r} = y_1||x_1 + \cdots + y_l||x_l. \tag{3}$$

Suma $x_1||y_1 + \cdots + x_l||y_l$ sa leve strane (3) može da se izrazi kao

$$(x_1 + \cdots + x_l + C_{y_1,\ldots y_l})||(y_1 + \cdots + y_l),$$

gde je $C_{y_1,\ldots y_k}$ prenos zbira $y_1 + y_2 + \cdots + y_l$. Slično, suma sa desne strane (3) se može izraziti kao

$$(x_1 + \cdots + x_l + C_{x_1,\ldots x_l})||(x_1 + \cdots + x_l).$$

Stoga, nakon rotacije leve sume dobijamo:

$$(y_1 + \cdots + y_l)||(x_1 + \cdots + x_l + C_{y_1,\ldots y_l}) = (y_1 + \cdots + y_l + C_{x_1,\ldots x_l})||(x_1 + \cdots + x_l). \tag{4}$$

Ako uzmemo u obzir veličinu reči x_i i y_i, iz (4) dobijamo:

$$y_1 + \cdots + y_l \equiv y_1 + \cdots + y_l + C_{x_1,\ldots x_l} \pmod{2^{n-r}},$$

$$x_1 + \cdots + x_l + C_{y_1,\ldots,y_n} \equiv x_1 + \cdots + x_l \pmod{2^r},$$

Tada:

$$C_{x_1,\ldots x_l} \equiv 0 \pmod{2^{n-r}} C_{y_1,\ldots y_l} \equiv 0 \pmod{2^r} \tag{5}$$

Rotaciona verovatnoća l-sume je jednaka verovatnoći da (5) važi i za slučajne vrednosti x_1, y_1, i = 1,... l.

Verovatnoća lančanog modularnog sabiranja datog lemom je odatle jednako verovatnoći sledećeg sistema:

$$C_{x_1,x_2} \equiv 0 \pmod{2^{n-r}}, C_{y_1,y_2} \equiv 0 \pmod{2^r}$$
$$C_{x_1,x_2,x_3} \equiv 0 \pmod{2^{n-r}}, C_{y_1,y_2,y_3} \equiv 0 \pmod{2^r}$$

...

$$C_{x_1,\ldots,x_k} \equiv 0 \pmod{2^{n-r}}), C_{y_1,\ldots y_k} \equiv 0 \pmod{2^r}$$

.

Dalje se ispostavlja da je ceo sistem ekvivalentan:

$$C_{x_1,\ldots x_k} \equiv 0, C_{y_1,\ldots y_k} \equiv 0. \tag{6}$$

Da to dokažemo, indukcijom po k pokazujemo da je sistem sistem kongruencija sa leve strane, odnosno

$$C_{x_1,\ldots x_i} \equiv 0 \,(\mathrm{mod}\ 2^{n-r})$$

za sve $2 \le i \le k$, ekvivalentan jednakosti $C_{x_1,\ldots x_k} \equiv 0$.

Slično zaključivanje pokazuje da je desna strana gornjeg sistema $C_{y_1,\ldots y_i} \equiv 0 \,(\mathrm{mod}\ 2^r)$ za sve $2 \le i \le k$ ekvivalentna jednačini $C_{y_1,\ldots y_k} \equiv 0$.

Razmotrimo najpre obrnuti smer ekvivalencije: naime, ako je $C_{x_1,\ldots x_k} = [\frac{x_1+\cdots+x_k}{2^r}] = 0$, odatle je $x_1 + x_2 + \cdots + x_k < 2^r$, odakle sledi da je $x_1 + x_2 + \cdots + x_i < 2^r$ i stoga $Cx_1, \ldots x_i = 0$, pa odatle i $C_{x_1,\ldots x_i} \equiv 0 \,(\mathrm{mod}\ 2^{n-r})$ za sve $2 \le i \le k$, kao što je potrebno za obrnuti pravac.

Drugi smer ekvivalencije dokazuje se indukcijom po k:

Za bazu indukcije uzimamo slučaj $k = 2$. Kongruencija $C_{x_1,x_2} \equiv 0 \pmod{2^{n-r}}$ je ekvivalentna $C_{x_1,x_2} = t \cdot 2^{n-r}$ za neki nenegativan celi broj t. Kako prenos sabiranja dve reči ne može biti veći od 1, to znači da C_{x_1,x_2} uzima neku od vrednosti iz skupa {0,1}. Ako je prenos 1, onda je $1 = t \cdot 2^{n-r}$, pa sledi da je $t = 1$ i $2n-r = 1$. Međutim, $r < n$, pa stoga $2n-r > 1$. Prema tome, prenos C_{x_1,x_2} može biti samo jednak nuli, pa je stoga $C_{x_1,x_2} \equiv 0 \,(\mathrm{mod}\ 2^{n-r})$ ekvivalentno $C_{x_1,x_2} \equiv 0$, čime smo dokazali bazu indukcije za $k = 2$.

Za induktivni korak, pretpostavimo da za neko $k \ge 2$ sistem kongruencija $C_{x_1,\ldots x_i} \equiv 0 \,(\mathrm{mod}\ 2^{n-r})$ za sve $2 \le i \le k$ implicira da važi jednakost $Cx_1, \ldots x_k = 0$. Pokazujemo da sistem konguencija

$$C_{x_1,\ldots x_i} \equiv 0 \,(mod\ 2^{n-r})$$

za sve $2 \le i \le k+1$ implicira da važi jednakost $C_{x_1,\dots x_{k+1}} = 0$.

Stvarno, prema induktivnoj hipotezi, imamo da prvih k kongruencija sistema implicira da je $C_{x_1,\dots x_k} = 0$, odakle $x_1 + \dots + x_k < 2^r$, dok $x_{k+1} < 2^r$, pa $x_1 + \dots + x_k + x_{k+1} < 2^r + 2^r = 2 \cdot 2^r$ i stoga, konačno,

$$C_{x_1,\dots x_{k+1}} = \left[\frac{x_1 + \dots + x_{k+1}}{2^r} \right]$$

uzima vrednost iz skupa $\{0,1\}$.

Sada, slično kao i u baznom slučaju, kongruencija

$$C_{x_1,\dots x_{k+1}} \equiv 0 \pmod{2^{n-r}}$$

i činjenica da je $r < n$ impliciraju dalje da vrednost prenosa $C_{x_1,\dots x_{k+1}}$ mora biti nula, čime završavamo dokaz induktivnog koraka. Kao rezultat ovoga, sveli smo čitav sistem na dve jednačnine, obe date u (6):

$$C_{x_1,\dots x_k} \equiv 0,$$
$$C_{y_1,\dots y_k} \equiv 0.$$

Na kraju, izračunajmo verovatnoću da (6) važi, kada su x_i, y_i, za $i = 1, 2, \dots, k$ slučajne r-bitne i $(n-r)$-bitne reči, redom. Posmatramo:

$$\Pr(x_1 + \dots x_k < 2^r \wedge 0 \le x_i < 2^r) \cdot \Pr(y_1 + \dots y_k < 2^{n-r} \wedge 0 \le y_i < 2^{n-r}) \tag{7}$$

Primetimo da

$$\Pr(x_1 + \dots x_k < 2^r \wedge 0 \le x_i < 2^r) = \sum_{j=0}^{2^{r-1}} \Pr(x_1 + \dots x_k = j \wedge 0 \le x_i < 2^r). \tag{8}$$

Dalje, članovi sa desne strane (8) mogu biti procenjeni na osnovu dobro poznate kombinatorne formule:

$$\#(z_1 + \dots + z_k = j \wedge 0 \le z_i) = \binom{j+k-1}{j}, \tag{9}$$

gde je $\#$ broj particija. Primetimo da u (9) uslov $0 \le z_i$ može biti zamenjen sa $0 \le z_i < t$ kada je $t > j$, jer se broj k-torki ne povećava kad je $z_i \ge t$ (suma je uvek veća od j). Stoga,

$$\Pr(z_1 + \dots z_k = j \wedge 0 \le z_i < t \wedge t > j) = \binom{j+k-1}{j} t^{-k},$$

a (7) može biti izraženo kao:

$$\Pr(x_1 + \dots x_k < 2^r \wedge 0 \le x_i < 2^r) \cdot \Pr(y_1 + \dots y_k < 2^{n-r} \wedge 0 \le y_i < 2^{n-r}) =$$
$$= \sum_{j=0}^{2^{r-1}} \binom{j+k-1}{j} \cdot 2^{-rk} \sum_{j=0}^{2^{n-r}-1} \binom{j+k-1}{j} \cdot 2^{-(n-r)k} =$$
$$= \frac{1}{2^{nk}} \sum_{j=0}^{2^{r-1}} \binom{j+k-1}{j} \sum_{j=0}^{2^{n-r}-1} \binom{j+k-1}{j}$$

Na kraju, koristimo formulu za sumiranje binomnih koeficijenta (za prirodne brojeve m, n)

$$\sum_{j=0}^{m} \binom{n+j}{j} = \binom{n+m+1}{m}.$$

i završavamo dokaz:

$$\begin{aligned}
&\Pr([(a_1 + a_2)_{\lll_r} = a_{1\lll_r} + a_{2\lll_r}] \wedge \\
&\wedge [(a_1 + a_2 + a_3)_{\lll_r} = a_{1\lll_r} + a_{2\lll_r} + a_{3\lll_r}] \wedge \\
&\wedge \dots \\
&\wedge [(a_1 + \dots + a_k)_{\lll_r} = a_{1\lll_r} + \dots + a_{k\lll_r}]) = \\
&= \frac{1}{2^{nk}} \sum_{j=0}^{2^{r-1}} \binom{j+k-1}{j} \sum_{j=0}^{2^{n-r}-1} \binom{j+k-1}{j} \\
&= \frac{1}{2^{nk}} \binom{k+2^r-1}{2^{r-1}} \binom{k+2^{n-r}-1}{2^{n-r}-1}
\end{aligned}$$

■

Iz gornje teoreme, sledi važna činjenica: lančana modularna sabiranja ne formiraju Markovljev lanac. Rotaciona verovatnoća ulančanih sabiranja po modulu ne može se izračunati kao proizvod verovatnoća pojedinačnih sabirnja po modulu.

Korišćenjem GNU MPFR biblioteku (GNU Multiple Precision Floating-Point Reliability) su izračunate rotacione verovatnoće ulančanih sabiranja po modulu prema rezultatima teoreme 6. Verovatnoće za 64-bitnu reč, rotacionih parametara 1 i 2 i preciznosti od 10000 cifara, date su u tabeli 2. Na primer, kad je parametar rotacije 1 i postoji 25 ulančanih sabiranja po modulu, verovatnoća da će izlazi ovih 25 sabiranja biti rotacioni je $2^{-109,6}$.
Na osnovu leme 1, ova verovatnoća bi bila $2^{-1,418 \cdot 25}$ što je približno $2^{-35,4}$.
Primetimo i da smo izračunali rotacionu verovatnoću kada je količina rotacije veća od 2. Neslaganje ima tendenciju da raste – što je količina rotacija bliža $\frac{n}{2}$, to je veće neslaganje između tvrdnji teoreme 6 i leme 1.

Lema 1 se koristi kad izlazi svih ulančanih modularnih sabiranja treba da budu rotacioni. Ovo je bitan zahtev jer u ARX-u izlazi srednjih sabiranja po modulu se koriste kao ulazi u druge operacije i pretpostavlja se da su rotacioni. Na primer, na slici 2, potrebno nam je da d bude rotaciono, kako je dalje ono korišćeno za računanje vrednosti w. Nasuprot tome, ako samo konačni izlaz više ulančanih sabiranja po modulu treba da bude rotacion, onda se ta rotaciona verovatnoća računa po drugoj formuli.

Tabela 2: Poređenje rotacionih verovatnoća izračunatih teoremom 5 i lemom 1

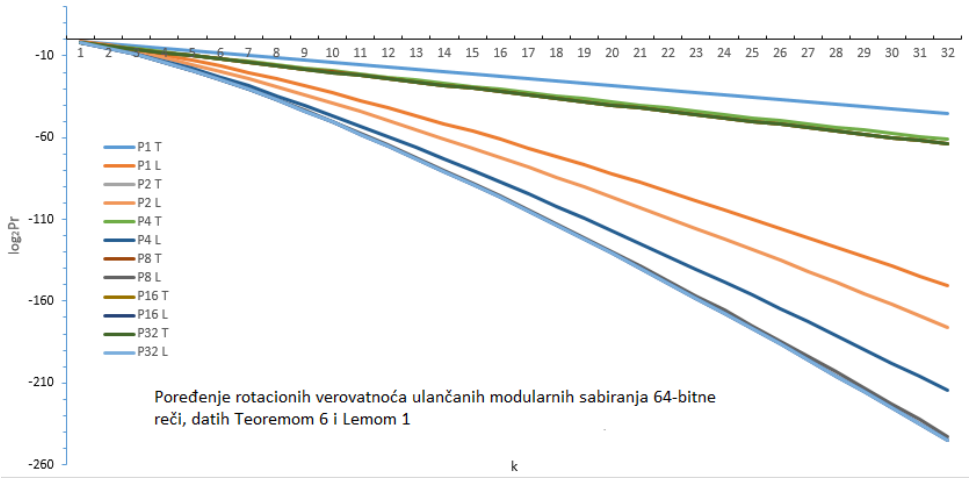
r = 1								
broj sabiranja	1	2	3	4	5	6	7	8
Teorema 6	-1,4	-2,8	-4,2	-5,7	-7,1	-8,5	-9,9	-11,3
Lema 1	-1,4	-3,6	-6,3	-9,3	-12,7	-16,3	-20,1	-24,1
broj sabiranja	9	10	11	12	13	14	15	16
Teorema 6	-12,7	-14,1	-15,6	-17,0	-18,4	-19,8	-21,2	-22,6
Lema 1	-28,3	-32,7	-37,1	-41,7	-46,4	-51,3	-56,2	-61,2
broj sabiranja	17	18	19	20	21	22	23	24
Teorema 6	-24,1	-25,5	-26,9	-28,3	-29,7	-31,1	-32,5	-34,0
Lema 1	-66,3	-71,4	-76,7	-82,0	-87,4	-92,9	-98,4	-104,0
broj sabiranja	25	26	27	28	29	30	31	32
Teorema 6	-35,4	-36,8	-38,2	-39,6	-41,0	-42,4	-43,9	-45,3
Lema 1	-109,6	-115,3	-121,1	-126,9	-132,8	-138,7	-144,6	-150,6
r = 2								
broj sabiranja	1	2	3	4	5	6	7	8
Teorema 6	-1,7	-3,4	-5,0	-6,7	-8,4	-10,1	-11,7	-13,4
Lema 1	-1,7	-4,3	-7,5	-11,1	-15,1	-19,4	-23,9	-28,7
broj sabiranja	9	10	11	12	13	14	15	16
Teorema 6	-15,1	-16,8	-18,4	-20,1	-21,8	-23,5	-25,1	-26,8
Lema 1	-33,6	-38,7	-44,0	-49,4	-54,9	-60,6	-66,3	-72,2
broj sabiranja	17	18	19	20	21	22	23	24
Teorema 6	-28,5	-30,2	-31,8	-33,5	-35,2	-36,9	-38,5	-40,2
Lema 1	-78,1	-84,2	-90,3	-96,5	-102,8	-109,1	-115,5	-122,0
broj sabiranja	25	26	27	28	29	30	31	32
Teorema 6	-41,9	-43,6	-45,3	-46,9	-48,6	-50,3	-52,0	-53,6
Lema 1	-128,5	-135,1	-141,8	-148,5	-155,3	-162,1	-169,0	-175,9
r = 4								
broj sabiranja	1	2	3	4	5	6	7	8
Teorema 6	-1,911	-3,8	-5,7	-7,6	-9,6	-11,5	-13,4	-15,3
Lema 1	-1,911	-4,9	-8,7	-12,99	-17,8	-22,9	-28,4	-34,2
broj sabiranja	9	10	11	12	13	14	15	16
Teorema 6	-17,2	-19,1	-21,0	-22,9	-24,8	-26,8	-28,7	-30,6
Lema 1	-40,1	-46,4	-52,8	-59,4	-66,1	-73,0	-80,1	-87,3
broj sabiranja	17	18	19	20	21	22	23	24
Teorema 6	-32,5	-34,4	-36,3	-38,2	-40,1	-42,0	-44,0	-45,9
Lema 1	-94,6	-102,0	-109,5	-117,1	-124,8	-132,6	-140,5	-148,5
broj sabiranja	25	26	27	28	29	30	31	32
Teorema 6	-47,8	-49,7	-51,6	-53,5	-55,4	-57,3	-59,2	-61,1
Lema 1	-156,5	-164,6	-172,8	-181,1	-189,4	-197,8	-206,2	-214,7

r = 8								
broj sabiranja	1	2	3	4	5	6	7	8
Teorema 6	-1,994	-3,99	-5,98	-7,98	-9,97	-11,96	-13,96	-15,95
Lema 1	-1,994	-5,2	-9,1	-13,76	-18,9	-24,5	-30,4	-36,7
broj sabiranja	9	10	11	12	13	14	15	16
Teorema 6	-17,95	-19,94	-21,93	-23,93	-25,92	-27,92	-29,91	-31,90
Lema 1	-43,3	-50,2	-57,3	-64,6	-72,2	-79,9	-87,8	-95,9
broj sabiranja	17	18	19	20	21	22	23	24
Teorema 6	-33,9	-35,89	-37,89	-39,88	-41,87	-43,87	-45,86	-47,86
Lema 1	-104,2	-112,6	-121,1	-129,8	-138,6	-147,5	-156,6	-165,7
broj sabiranja	25	26	27	28	29	30	31	32
Teorema 6	-49,85	-51,84	-53,84	-55,83	-57,83	-59,82	-61,81	-63,81
Lema 1	-175,0	-184,4	-193,8	-203,4	-213,1	-222,8	-232,6	-242,6

r = 16								
broj sabiranja	1	2	3	4	5	6	7	8
Teorema 6	-1,9	-3,9	-5,9	-7,9	-9,9	-11,9	-13,9	-15,9
Lema 1	-1,9	-5,2	-9,2	-13,8	-19,0	-24,6	-30,6	-36,9
broj sabiranja	9	10	11	12	13	14	15	16
Teorema 6	-17,9	-20,0	-21,9	-23,9	-25,9	-27,9	-29,9	-31,9
Lema 1	-43,6	-50,4	-57,7	-65,1	-72,7	-80,5	-88,5	-96,7
broj sabiranja	17	18	19	20	21	22	23	24
Teorema 6	-33,9	-35,9	-37,9	-39,9	-41,9	-43,9	-45,9	-47,9
Lema 1	-105,0	-113,5	-122,2	-130,9	-139,8	-148,9	-158,1	-167,4
broj sabiranja	25	26	27	28	29	30	31	32
Teorema 6	-49,9	-51,9	-53,9	-55,9	-57,9	-59,9	-61,9	-63,9
Lema 1	-176,8	-186,3	-195,9	-205,6	-215,4	-225,3	-235,3	-245,4

r = 32								
broj sabiranja	1	2	3	4	5	6	7	8
Teorema 6	-1,9	-3,9	-5,9	-7,9	-9,9	-11,9	-13,9	-15,9
Lema 1	-1,9	-5,2	-9,2	-13,8	-19,0	-24,6	-30,6	-36,9
broj sabiranja	9	10	11	12	13	14	15	16
Teorema 6	-17,9	-20,0	-21,9	-23,9	-25,9	-27,9	-29,9	-31,9
Lema 1	-43,6	-50,5	-57,7	-65,1	-72,7	-80,5	-88,5	-96,7
broj sabiranja	17	18	19	20	21	22	23	24
Teorema 6	-33,9	-35,9	-37,9	-39,9	-41,9	-43,9	-45,9	-47,9
Lema 1	-105,0	-113,5	-122,1	-130,9	-139,9	-148,9	-158,1	-167,4
broj sabiranja	25	26	27	28	29	30	31	32
Teorema 6	-49,9	-51,9	-53,9	-55,9	-57,9	-59,9	-61,9	-63,9
Lema 1	-176,7	-186,3	-195,9	-205,6	-215,4	-225,3	-235,3	-245,4

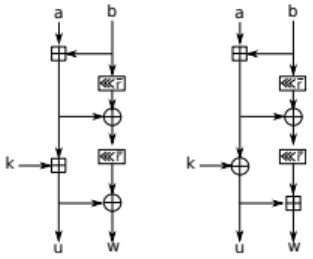
Sadržaj tabele 2 preglednije prikazuje dijagram na slici 3:



Slika 3: Poređenje rotacionih verovatnoća izračunatih teoremom 6 i lemom 1

Oznaka P1 T sa slike 3 predstavlja rotacionu verovatnoću ulančanih modularnih sabiranja datu teoremom 6, gde je količina rotacije 1, a P1 L istu tu verovatnoću i sa istom količinom rotacije određuje na osnovu leme 1. Analogno se objašnjavaju i ostale oznake.

Pretpostavlja se da samo rotacije mogu prekinuti lanac sabiranja. Moramo istaći i da XOR-ovi takođe prekidaju takve lance, svaki put kada je drugi argument XOR-a slučajna vrednost. U praksi, za ARX algoritme, lanci se prekidaju i rotacijama i XOR-ovima. Štavise, zbog mogućnosti XOR-a da prekine lanac sabiranja po modulu, rotaciona verovatnoća ARX primitiva u velikoj meri zavisi od toga na koji su način potključevi menjaju stanje, odnosno veoma je bitno da li su potključevi XOR-ovani ili sabrani po modulu u stanje. Za ilustraciju videti sliku 4.

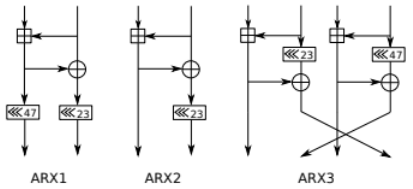


Slika 4: Dve ARX strukture sa različitim načinom delovanja potključeva.

Na slici 4 su prikazane dve ARX primitive, svaka sa po dva sabiranja po modulu. Razlika je u tome što se u ARX strukturi sa leve strane slike, potključ sabira sa stanjem, a u ARX strukturi sa desne strane, potključ se XOR-ovuje sa stanjem. Sa slike mozemo da vidimo da u levoj ARX strukturi, potključ ne prekida lanac sabiranja po modulu, dok u desnoj ARX strukturi to jeste slučaj. Dakle, za levu ARX strukturu koristićemo lemu 1 da bismo izračunali rotacionu verovatnoću. Zadatak je naći rotacione verovatnoće ovih dveju primitiva, kada je rotacioni parametar 1 (a reči 64-bitne). ARX šema sa leve strane slike 4 ima samo jedan lanac sa 2 sabiranja po modulu. Prema tome, na osnovu leme 1 i njoj pridružene tabele, rotaciona verovatnoća ovog lanca je $2^{-3,6}$.

Sa druge strane, ARX sa desne strane slike 4 ima dva lanca sastavljena od jednog sabiranja po modulu, pa je samim tim rotaciona verovatnoća ove šeme $2^{-1,4} \cdot 2^{-1,4} = 2^{-2,8}$.

Ispravnost naše analize testirana je kompjuterskom simulacijom – testovi su potvrdili predskazanu rotacionu verovatnoću. Testirali smo rotacionu verovatnoću predviđenu lemom 1 i novim algoritmom na slučaju tri ARX primitive sa kružnom funkcijom, kao što je dato na slici 5.



Slika 5: Kružne funkcije tri ARX primitive, korišćene za eksperimentalno testiranje rotacione verovatnoće

Prvi ARX nema lance sabiranja po modulu, to jest, svi lanci su prekinuti rotacijom i/ili XOR-om. Drugi ARX ima jedan lanac sabiranja po modulu kroz runde. Treći ARX ima dva lanca i zasniva se na šifri Threefish-256. Za svaku od ove tri ARX primitive dobili smo eksperimentalnim putem rotacione verovatnoće uzimajući 223 slučajno odabranih ulaza rotacionih parova (sa količinom rotacije 1 ili 2) i brojeći koliko izlaznih parova su takođe rotacioni parovi. Broj potrebnih ulaznih parova za eksperimentalnu procenu rotacione verovatnoće određuje se procenom parametra p binomne raspodele - binomna raspodela sa parametrima n i p je diskretna raspodela verovatnoće broja uspeha u sekvenci od n nezavisnih eksperimenata, svaki od kojih daje odgovor na da-ne pitanje i svaki ima svoj bulov rezultat - uspeh (sa verovatnoćom p) ili neuspeh (sa verovatnoćom $q = 1 - p$). Rezultati su sumirani u tabeli 3. Iz tabele uočavamo da se eksperimentalne verovatnoće dobro slažu sa predviđenim.

količina rotacije =1						
broj rundi	3	4	5	6	7	8
ARX1 teorijska	-4,25	-5,66	-7,08	-8,49	-9,91	-11,32
ARX1 eksperimentalna	-4,25	-5,66	-7,08	-8,49	-9,91	-11,32
ARX2 teorijska	-6,26	-9,32	-12,68	-16,30	-20,13	-24,15
ARX2 eksperimentalna	-6,26	-9,32	-12,69	-16,30	-20,15	-24,15
ARX3 teorijska	-12,53	-18,64	-25,37	-32,6	-40,26	-48,29
ARX3 eksperimentalna	-12,53	-18,64	-25,37	np	np	np
količina rotacije =2						
broj rundi	3	4	5	6	7	8
ARX1 teorijska	-1,68	-3,35	-5,03	-6,7	-8,38	-10,06
ARX1 eksperimentalna	-1,68	-3,35	-5,03	-6,71	-8,39	-10,07
ARX2 teorijska	-1,68	-4,26	-7,46	-11,1	-15,1	-19,39
ARX2 eksperimentalna	-1,68	-4,26	-7,46	-11,1	-15,11	-19,39
ARX3 teorijska	-3,36	-8,53	-14,91	-22,2	-30,2	-38,78
ARX3 eksperimentalna	-3,36	-8,53	-14,91	-22,25	-29,68	np

Tabela 3: Upoređivanje teorijske i eksperimentalne rotacione verovatnoće tri ARX primitive sa slike 5. Verovatnoće su date eksperimentalno procenjene.

np – nije izvođeno – eksperiment nije izvođen kada je teorijska verovatnoća ispod 2^{-32}.

4 Primene

Razmotrićemo primenu opisanog postupka analize na šifre ARX, odnosno na kriptografske heš algoritme.

Heš funkcija je matematička funkcija koja se primenjuje na poruku promenljive dužine, dajući izlaz fiksne dužine, koji se naziva heš vrednost. Važna je sigurnost da primenjena poruka nije izmenjena, odnosno da joj je sačuvan integritet.

Heš funkcije koriste iterativnu konstrukciju, što podrazumeva deljenje poruka na blokove, a potom iterativnu obradu svakog bloka. Jedna od najzastupljenijih konstrukcija za iterativno heširanje je Merkle-Damgard konstrukcija. Srž ove konstrukcije predstavlja funkcija kompresije $f : \{0,1\}^b \times \{0,1\}^n \rightarrow \{0,1\}^n$. Na samom početku, poruka M se deli na blokove $M = M_1, M_2, \ldots, M_l$ jednake dužine, b. Ako poslednji blok nije iste dužine kao i prethodni blokovi, dopunjuje se. Jedan od načina za dopunjavanje je sledeći: na bitove poslednjeg bloka se dodaje najpre jedinica, zatim nule bitova i na kraju dužina poruke izražena u bitovima. Dopunjavanje je potrebno kako bi se obezbedilo da dve različite poruke daju drugačije sekvence blokova, samim tim i različite heš vrednosti. Funkcija kompresije izvršava se l puta. Merkle-Damgard konstrukcija može da koristi blokovsku šifru kao funkciju kompresije. Sigurnost Merkle-Damgard konstrukcije zavisi od sigurnosti funkcije kompresije.

Heš funkcija ilustrovana je sledećim primerom. Na slici 6 data je heš vrednost praznog stringa.

```
Skein-256-256("")
c8877087da56e072870daa843f176e9453115929094c3a40c463a196c29bf7ba
Skein-512-256("")
39ccc4554a8b31853b9de7a1fe638a24cce6b35a55f2431009e18780335d2621
Skein-512-512("")
bc5b4c50925519c290cc634277ae3d6257212395cba733bbad37a4af0fa06af41fca7903d0
6564fea7a2d3730dbdb80c1f85562dfcc070334ea4d1d9e72cba7a
```

Slika 6: Heš vrednosti praznog stringa.

Na slici 7 se vidi kako se heš vrednost menja dodavanjem tačke na kraj rečenice:

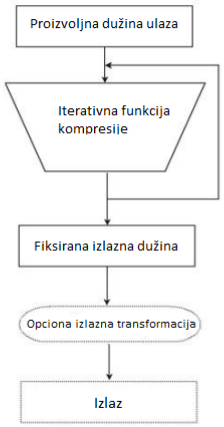
```
Skein-512-256("The quick brown fox jumps over the lazy dog")
b3250457e05d3060b1a4bbc1428bc75a3f525ca389aeab96cfa34638d96e492a
Skein-512-256("The quick brown fox jumps over the lazy dog.")
41e829d7fca71c7d7154ed8fc8a069f274dd664ae0ed29d365d919f4e575eebb
Skein-512-512("The quick brown fox jumps over the lazy dog")
94c2ae036dba8783d0b3f7d6cc111ff810702f5c77707999be7e1c9486ff238a7044de7342
93147359b4ac7e1d09cd247c351d69826b78dcddd951f0ef912713
Skein-512-512("The quick brown fox jumps over the lazy dog.")
658223cb3d69b5e76e3588ca63feffba0dc2ead38a95d0650564f2a39da8e83fbb42c9d6ad
9e03fbfde8a25a880357d457dbd6f74cbcb5e728979577dbce5436
```

Slika 7: Menjanje heš vrednosti dodavanjem tačke na kraj rečenice.

4.1 Primena rotacione kriptoanalize na BLAKE2

BLAKE2 je heš funkcija koja podržava 256-bitne i 512-bitne izlaze. Nadalje analizira samo verziju sa 512-bitnim izlazom, ali analiza se slično odnosi i na drugu verziju.

Prilikom prvog izvršavanja funkcije kompresije BLAKE2, argumenti su inicijalizovani vektor $z_0 = IV$ i prvi blok poruke M_1. Heš vrednost je vrednost koja se dobija posle poslednjeg izvršavanja funkcije kompresije F. Konstrukcija je prikazana dijagramom na slici 8, a pseudokod opisan u algoritmu.



Slika 8: Konstrukcija BLAKE2

Algoritam 1: Funkcija kompresije BLAKE2

1: $z_0 = IV$
2: for $i = 0$ to m
3: $z_i = f(z_{i-1}, M_i)$
4: return $h(M) = z_M$

BLAKE koristi samo operacije sabiranja u polju Z_2^{64} i ekskluzivnog ili (u oznaci $\oplus$) nad poljem Z_2^{64}. Početno stanje funkcije kompresije BLAKE2 se sastoji od šesnaest 64-bitnih reči. Funkcija kompresije uzima kao ulaz 864–bitnih reči $h_0, h_1, \ldots h_7$, 864–bitnih konstantnih vektora $IV_0, \ldots IV_7$, brojač t_0 t_1 od 264–bitne reči, koji broji broj bajtova koji su heširani do tog trenutka i dva bitna flega, f_0 i f_1.

Fleg f_0 je postavljen na vrednost $ff \ldots f$(šesnaest f) kad je tekući blok poruke poslednji, a $00 \ldots 00$ inače (šesnaest 0); f_1 ima sličnu ulogu u tree-hashingu.

Ulaz u funkciju kompresije se incijalizuje kao:

$$\begin{pmatrix} v_0 & v_1 & v_2 & v_3 \\ v_4 & v_5 & v_6 & v_7 \\ v_8 & v_9 & v_{10} & v_{11} \\ v_{12} & v_{13} & v_{14} & v_{15} \end{pmatrix} \leftarrow \begin{pmatrix} h_0 & h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 & h_7 \\ IV_0 & IV_1 & IV_2 & IV_3 \\ t_0 \oplus IV_4 & t_1 \oplus IV_5 & f_0 \oplus IV_6 & f_1 \oplus IV_7 \end{pmatrix}$$

Početno stanje je ulaz za 12 rundi funkcije $G = (G_0, G_1, G_2, G_3, G_4, G_5, G_6, G_7)$. Jedna runda funkcije G se sastoji od dva koraka. Prvi korak:

$$G_0(v_0, v_4, v_8, v_{12}),\ G_1(v_1, v_5, v_9, v_{13}),\ G_2(v_2, v_6, v_{10}, v_{14}),\ G_3(v_3, v_7, v_{11}, v_{15}),$$

Drugi korak:

$$G_4(v_0, v_5, v_{10}, v_{15}),\ G_5(v_1, v_6, v_{11}, v_{12}),\ G_6(v_2, v_7, v_8, v_{13}),\ G_7(v_3, v_4, v_9, v_{14}).$$

Funkcija G uzima kao ulaz reč od četiri 64-bitne reči stanja (a, b, c, d) i 2 reči poruke, M_i i M_j, koje su definisane pozicijom indeksa i u funkciji: u rundi r, M_i je dato kao $\sigma_{r \mod 10}(2i)$, a M_j kao $\sigma_{r \mod 10}(2i+1)$, gde je $\sigma_{r \mod 10}$ jedna od deset permutacija datih u tabeli 4.

σ_0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
σ_1	14	10	4	8	9	15	13	6	1	12	0	2	11	7	5	3
σ_2	11	8	12	0	5	2	15	13	10	13	3	6	7	1	9	4
σ_3	7	9	3	1	13	12	11	14	2	6	5	10	4	0	15	8
σ_4	9	0	5	7	2	4	10	15	14	1	11	12	6	8	3	13
σ_5	2	12	6	10	0	11	8	3	4	13	7	5	15	4	1	9
σ_6	12	5	1	15	14	13	4	10	0	7	6	3	9	2	8	11
σ_7	13	11	7	14	12	1	3	9	5	0	15	4	8	6	2	10
σ_8	6	15	14	9	11	3	0	8	12	2	13	7	1	4	10	5
σ_9	10	2	8	4	7	6	1	5	15	11	9	14	3	12	13	0

Tabela 4: Permutacije $\sigma_{r \mod 10}$

Sama funkcija $G(a, b, c, d, M_i, M_j)$ funkcioniše na sledeći način:

1: $a \leftarrow a + b + M_i$

2: $d \leftarrow (d \oplus a)_{\ggg_{32}}$

3: $c \leftarrow c + d$

4: $b \leftarrow (b \oplus c)_{\ggg_{25}}$

5: $a \leftarrow a + b + M_j$

6: $d \leftarrow (d \oplus a)_{\ggg_{16}}$

7: $c \leftarrow c + d$

8: $b \leftarrow (b \oplus c)_{\ggg_{11}}$

Konačno, izlaz funkcije kompresije $h'_0, h'_1, \ldots h'_7$ dobija se kombinovanjem ulaznih reči i završnih stanja $v_0, v_1, \ldots v_{15}$ izračunavanjem:

$$h'_0 \leftarrow h_0 \oplus v_0 \oplus v_8$$

$$h'_1 \leftarrow h_1 \oplus v_1 \oplus v_9$$

$$h'_2 \leftarrow h_2 \oplus v_2 \oplus v_{10}$$

$$h'_3 \leftarrow h_3 \oplus v_3 \oplus v_{11}$$

$$h'_4 \leftarrow h_4 \oplus v_4 \oplus v_{12}$$

$$h'_5 \leftarrow h_5 \oplus v_5 \oplus v_{13}$$

$$h'_6 \leftarrow h_6 \oplus v_6 \oplus v_{14}$$

$$h'_7 \leftarrow h_7 \oplus v_7 \oplus v_{15}$$

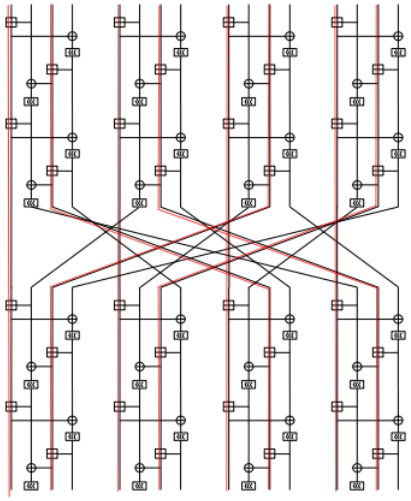
BLAKE2 je ARX primitiva, jer su jedine operacije koje se koriste modularno sabiranje, XOR i rotacija bitova. Funkcija G ne sadrži konstante, što BLAKE2 čini pogodnim za rotacioni napad. Dakle, krećemo sa ulaznim rotacionim parom $(x, x_{\lll_r})$ i proveravamo da li je izlaz primitive F takođe rotacioni, odnosno da li važi $F(x)_{\lll_r} = F(x_{\lll_r})$. Gruba procena je da na osnovu teoreme verovatnoća rotacionog izlaza za ARX primitivu (verovatnoća da prethodna jednakost važi za proizvoljno x) zavisi samo od broja sabiranja po modulu, korišćenih u F.

Funkcija G u BLAKE2 sadrži 6 sabiranja. Da bismo maksimizovali verovatnoću, postavljamo količinu rotacije na 1, pa je stoga rotaciona verovatnoća sabiranja po modulu oko $2^{-1,4}$. Stoga, za celu funkciju G imamo rotacionu verovatnoću: $2^{-1,4 \cdot 6} \approx 2^{-8,4}$. Eksperimentalnim putem je ipak utvrđeno [10] da je ta rotaciona verovatnoća nešto niža, odnosno oko $2^{-9,1}$. Koristeći ovu vrednost kao vrednost rotacione verovatnoće jedne primene G, a s obzirom na to da cela funkcija F ima 12 rundi, svaka sa 8 poziva G, zaključujemo da je rotaciona verovatnoća u funkciji kompresije BLAKE2 oko $2^{-9,1 \cdot 12 \cdot 8} = 2^{-873,6}$. Otuda, kada je drugi ulaz dobijen rotacijom prvog za 1 bit, teorijski je moguće statistički razlikovati izvršavanje funkcije F od slučajne funkcije. Pošto ova permutacija radi na skupu 1024-bitnih poruka, zahteva se rotacioni statistički prepoznavac za punu permutaciju od 12 rundi. Preciznija analiza na osnovu teoreme 6 pokazuje da je stvarna

verovatnoća bila daleko niža zbog ulančavanja sabiranja po modulu.

Postavimo sve reči poruke na 0, kako ovo daje rotacioni par poruka koji je isporučen sa najvećom rotacionom verovatnoćom. Zatim, identifikovaćemo sva ulančana sabiranja po modulu. Na slici 9 pokazana je runda permutacije, gde se može videti tačno 8 lanaca od po 4 sabiranja po modulu. Možemo pretpostaviti da su nelančani ulazi sabiranja po modulu nezavisni, jer uvek prolaze kroz rotacije, stoga se može primeniti lema 1.

Primetimo da se 8 lanaca sabiranja po modulu propagira kroz sledeće runde, čineći ukupno $4R$ sabiranja u svakom lancu tokom R rundi. Shodno tome, varijanta algoritma od 7 rundi (sa 8 lanaca od po 28 sabiranja po modulu) ima rotacionu verovatnoću jednaku $2^{126,9} \cdot 8 = 2^{-1015,2}$ kada je količina rotacije jednaka 1. Uzimanje više rundi bi rezultiralo rotacionom verovatnoćom manjom od 2^{-1024}. Zaista, lema 1 daje verovatnoću od $2^{-132,8}$ za 29 ulančanih sabiranja po modulu manjih ili jednakih od $2^{-1062,4}$. Stoga, rotacioni prepoznavac za permutaciju BLAKE2 radi samo za do 7 rundi, što je manje od 12 rundi dobijenih na osnovu teoreme 6.



Slika 9: Prikaz 8 lanaca (označenih crvenom bojom) od 4 sabiranja po modulu u jednoj rundi permutacije BLAKE2.

4.2 Primena rotacione kriptoanalize na Threefish

U ovom odeljku analizira se otpornost blok šifre Threefish na rotacionu kriptoanalizu [12,13,14]. Demonstrirano je da verovatnoća rotacionog par Threefish šifrata veća nego za slučajnu permutaciju.

Threefish blok šifra funkcioniše po principu da je sigurnije imati veliki broj jednostavnih rundi, nego nekoliko kompleksnijih. Koristi tri operacije: sabiranje, XOR i rotacije ulevo za r bita. Postoje tri različite verzije Threefish algoritma: Threefish-256 - 256-bitnu blok šifru sa 256-bitnim ključem; Threefish-512 - 512-bitni blok i ključ; Threefish-1024 - 1024-bitni blok i ključ. Za različite veličine bloka i ključa, potrebno je izvršiti različit broj rundi: na primer, kada je veličina ključa i bloka 256 bita, potrebno je izvšiti 72 runde, a u slučaju da je ključ veličine 1024 bita, potrebno je 80 rundi. Neka je N_w broj reči. I početno stanje I i ključ K se sastoje od N_w 64-bitnih reči ($N_w = 4, 8, 16$ za Threefish-256,-512,-1024 respektivno). Reči N_w s-tog potključa K^s desfinišu se na sledeći način:

$$\begin{aligned}
K^s_j &= K_{s+j} \pmod{N_w+1},\ 0 \le j \le N_w - 4;\\
K^s_{N_w-3} &= K_{(s+N_w-3)} \pmod{(N_w+1)+t_s} \pmod 3;\\
K^s_{N_w-2} &= K_{(s+N_w-2)} \pmod{(N_w+1)+t_{s+1}} \pmod 3;\\
K^s_{N_w-1} &= K_{(s+N_w-1)} \pmod{(N_w+1)+s},
\end{aligned}$$

gde je s brojač rundi, a t_0 i t_1 su reči od 128 bita čija se vrednost može izabrati, i važi:

$$t_2 = t_0 + t_1,\ K_{N_w} = \left[\frac{2^{64}}{3}\right] \oplus \oplus_{j=0}^{N_w-1} K_j$$

Za dalju analizu, fiksiraćemo $t_0 = t_1 = 0$.

Neka je N_r broj rundi. Tada, za svako $1 \le d \le N_r$:

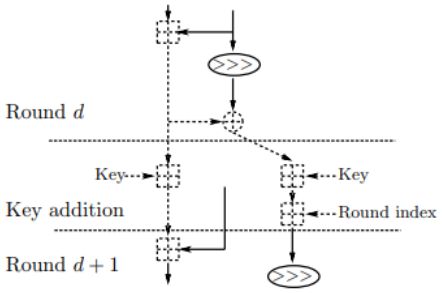
- ako je $d\ mod 4 = 1$ dodati potključ: $I_j \leftarrow I_j + K_j^{d/4}$ - podključ se dodaje u svakoj četvrtoj rundi;
- za $0 \le j < N_w/2$ postaviti: $(I_{2j}, I_{2j+1}) \leftarrow \text{MIX}\ ((I_{2j}, I_{2j+1}))$;
- primeniti permutaciju π na stanje reči.

Na kraju, dodaje se potključ $K^{N_r/4}$. Operacija MIX ima dva ulaza (x_0, x_1) i proizvodi dva izlaza (y_0, y_1) sledećim ARX transformacijama:

$$\begin{aligned}
y_0 &= x_0 + x_1\\
y_1 &= (x_1 \lll R_{(d \bmod 8)+1,j}) \oplus y_0
\end{aligned}$$

4.2.1 Napad na originalni Threefish

U ovom odeljku biće prikazana primena rotacione analize na originalnu verziju Threefish-a. Takav vid napada sa sobom povlači i obavezno bavljenje brojačima rundi - konstantama male težine. Kako bi se prevazišao ovaj problem, uvode se korekcije u paru ključeva. Neka je K prvi tajni ključ. Tada se drugi ključ K' definiše kao: $K'_i = \overleftarrow{K}_i \oplus e_i$. Primena rotacionih parova sa greškom je prikazana na slici 10:



Slika 10: Rotacione greške u sloju dodavanja ključeva u Threefishu. Isprekidane linije sadrže rotacione parove sa greškom.

Eksperimentalno je pokazano da vrednosti korekcija e_i ne bi trebalo da budu veće od 16 bita, inače ne poništavaju brojače rundi. Za Threefish-256 i Threefish-512, lako se (grubom silom) nalaze tačne vrednosti korekcija koje poništavaju brojače sa najvećom verovatnoćom. Za Threefish-1024, uzete su vrednosti koje su bile dobre za verziju 512.

Korekcije ne dozvoljavaju dobijanje precizne formule za izračunavanje verovatnoće sabiranja za rotacioni par. Stoga, ove verovatnoće se moraju naći empirijski. Grupišemo dve runde sa dodavanjem potključa (runda - dodavanje potključa - runda), pa Monte Carlo metodom nalazimo verovatnoću da rotacioni par stanja na ulazu ove dve runde i rotacioni par potključeva sa korekcijama proizvode rotacioni par stanja na izlazu. Na osnovu ovih vrednosti, dobijaju se verovatnoće najboljeg rotacionog para. Procena logaritama verovatnoća po rundama su date u sledećoj tabeli 5.

Kada rotacioni par ne sadži korekcije, verovatnoća sabiranja definisana je lemom 1, za količinu rotacije $r = 1$, pa iznosi $2^{-1,415}$. Dve uzastopne runde Threefish-256 imaju 4 MIX operacije, a svaka od njih ima jedno sabiranje. Stoga dve rudne bez dodavanja potključa imaju verovatnoću $2^{-6,6}$. Analogno, za Threefish-512 i Threefish-1024, dobijamo $2^{-13,3}$ i $2^{-26,6}$ respektivno.
Kada rotacioni par sadrži korekcije, nalazimo verovatnoću za dve runde (runda + dodavanje potključa + runda) eksperimentalno. Verovatnoća za prvu rundu (dodavanje ključa + runda) se takođe računa eksperimentalno.

Napad je moguć sa 39, 42 i 43,5 runde originalne verzije Threefish-256,-512,-1024 pri čemu je složenost $2^{252,4}, 2^{507}, 2^{1014,5}$ šifrovanja respektivno. Procedura napada prati sledeći algoritam:

1. Generisati slučajni tekst P i njegov šifrat ključem K.

runda	Threefish-256	Threefish-512	Threefish-1024
1	-13,1	-22,8	-45,6
2-3	-6,6	-13,3	-26,7
4-5	-17,57	-29,47	61,48
6-7	-6,6	-13,3	-26,7
8-9	-15,33	-31,33	- 63,32
10-11	-6,6	-13,3	-26,7
12-13	-15,6	-29,73	-61,68
14-15	-6,6	-13,3	-26,7
16-17	-21,08	- 34,35	-66,44
18-19	-6,6	-13,3	-26,7
20-21	-18,46	-37,25	-68,82
22-23	-6,6	-13,3	-26,7
24-25	-21,47	-34,38	-66,34
26-27	-6,6	-13,3	- 26,7
28-29	-21,55	-36	-67,31
30-31	-6,6	-13,3	-26,7
32-33	-21,74	-37,63	-69,28
34-35	-6,6	-13,3	-26,7
36-37	-22,96	-38,17	-67,79
38-39	-6,6	-13,3	-26,7
40-41		-36,4	-69,64
42-43		-6,6	-26,7
44-45			-13,3
ukupno rundi	39	42	43,5
ukupna verovatnoća	-254.1	-507	-1014,5

Tabela 5: Verovatnoće za rotacione parove za različite verzije Threefish

2. Izračunati P' po pravilu: $P'_I = P_i \oplus d_i$ i njegov šifrat ključem K', gde je d_i korekcija teksta.

3. Proveriti da li je $(E_K(P), E'_K(P'))$ rotacioni par.

Rotacioni par otkriva informaciju o najlevljim bitovima ključa svake reči ključa.
Korekcije ključa i teksta su definisane posebno za svaku od tri verzije Threefish u tabeli 6:

.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Threefish-256																
d_i	3	10	3	15												
e_i	6	10	6	15												
Threefish-512																
d_i	0	6	3	6	3	6	3	6								
e_i	5	6	6	6	6	6	6	6								
Threefish-1024																
d_i	0	6	3	6	3	6	3	6	3	6	3	6	3	6	3	6
e_i	5	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6

Tabela 6: Korekcije ključa i teksta za različite verzije algoritma Threefish

U tabeli 7 vidimo složenosti napada u odnosu na broj rundi za različite verzije algoritma Threefish.

šifra	indeks runde	konst $[2^64/3]$	runda	složenost
Threefish-256	ne	ne	59	2^{252}
Threefish-256	ne	da	50	$2^{253,8}$
Threefish-256	da	ne	44	$2^{251,4}$
Threefish-256	da	da	39	$2^{254,1}$
Threefish-256	ne	ne	70	2^{224}
Threefish-512	ne	ne	59	2^{504}
Threefish-512	ne	da	50	$2^{507,6}$
Threefish-512	da	ne	51.5	$2^{505,5}$
Threefish-512	da	da	42	2^{507}
Threefish-512	ne	ne	72	2^{448}
Threefish-1024	ne	ne	59	2^{1008}
Threefish-1024	ne	da	50	$2^{1015,2}$
Threefish-1024	da	da	43.5	$2^{1014,5}$
Threefish-1024	ne	ne	80	2^{950}

Tabela 7: Složenosti napada za različite verzije algoritma Threefish

4.3 Primena rotacione kriptoanalize na Skein

Skein je heš funkcija predložena za takmičenje NIST SHA-3, koja je stigla do poslednjeg kruga takmičenja. U svakom krugu su autori predlagali neka podešavanja i poboljšanja prethodne verzije. Naziv potiče od načina na koji funkcija Skein prepliće ulaz, "slično kao pletenica prediva". Njegov dizajn kombinuje brzinu, bezbednost, jednostavnost i fleksibilnost.

Razmatraćemo verziju Skein sa 512-bitnim unutrašnjim stanjem, a označavamo je kao Skein-512. Ista analiza važi i za druge verzije. Glavne komponente Skein-a su Threefish Block Cipher, Unique Block Iteration i Optional Argument system. Heš funkcija Skein može da se izračuna na osnovu blok šifre Threefish. Početni korak je generisanje potključeva ključa i reči za podešavanje koje su ulaz. Za Skein-512 tekst i ključ imaju 512 bitova, a reči za podešavanje 128. Sledeći korak je proces šifrovanja. Funkcija kompresije Skein-512 je zasnovana na blok šifri Threefish, koja funkcioniše po sistemu da je sigurnije imati veći broj jednostavnih rundi, nego mali broj kompleksnih. S tim u vezi, funkcija kompresije izvršava se u 72 runde.

Razmatramo slučaj kad su veličine ulaznih blokova 512-bita, a na njih možemo da gledamo kao na osam 64-bitnih reči.
Funkcija kompresije definise se kao:

$$F(CV, T, M) = E_{CV,T}(M) \oplus M,$$

gde je $E_{K,T}(P)$ šifra Threefish, CV je prethodna ulančana vrednost, T je reč za podešavanje, koja se sastoji od dve 64-bitne reči, a M je blok poruke. Vrednost reči za podešavanje je funkcija nekoliko parametara, uključujući i indeks poslednjeg bita poruke.

Threefish-512 transformiše tekst tekući blok M u 72 runde po sledećem principu:
$P \rightarrow$ dodaj potključ $K^0 \rightarrow 4$ runde $\rightarrow$ dodaj $K^1 \rightarrow \ \ldots \ \rightarrow$ dodaj $K^{18} \rightarrow$ C.

Kao što vidimo, potključuevi se dodaju nakon svake četvrte runde. Potključ $K^s = (K_0^s, K_1^s, \ldots K_7^s)$ je dobijen od ključa $K = (K[0], K[1], \ldots K[7])$ na sledeći način:

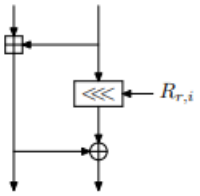
$$\begin{aligned} K_j^s &= K[(s+j) \bmod 9], \ \ 0 \le j \le 4; \\ K_5^s &= K[(s+5) \bmod 9] + T[s \bmod 3]; \\ K_6^s &= K[(s+6) \bmod 9] + T[(s+1) \bmod 3]; \\ K_7^s &= K[(s+7) \bmod 9] + s, \end{aligned}$$

gde je s brojač rundi, $T[0]$ i $T[1]$ su reči za podešavanje, $T[2] = T[0] + T[1]$, a $K[8] = C_{240} \oplus \oplus_{j=0}^{7} K[j]$, sa konstantom C_{240} optimizovanom da otežа rotacioni napad.

Jedna runda transformiše početno stanje na sledeći način. Osam reči $I^0, I^1, \ldots I^7$ su grupisane u parove i svaki par se obrađuje 128-bitnom funkcijom MIX.

Operacija MIX je nelinearna funkcija, ima dva ulaza, (x_0, x_1) i proizvodi dva izlaza, (y_0, y_1) sledećom transforamcijom:

$$y_0 = (x_0 + x_1) \pmod{2^{64}}$$
$$y_1 = (x_1 \lll_{R_{(d \bmod 8),j}}) \oplus y_0$$



Slika 11: MIX funkcija za Skein

Na slici vidimo funkciju MIX.
Vrednosti rotacionih konstanti $R_{i,j}$ kao i permutacije π, koje su različite za svaku verziju Threefish-a su date u tabeli 8 i tabeli 9:

j	0	1	2	3
$d = 0$	46	36	19	37
$d = 1$	33	27	14	42
$d = 2$	17	49	36	39
$d = 3$	44	9	54	56
$d = 4$	39	30	34	24
$d = 5$	13	50	10	17
$d = 6$	25	29	39	43
$d = 7$	8	35	56	22

Tabela 8: Vrednosti rotacionih konstanti $R_{d,j}$.

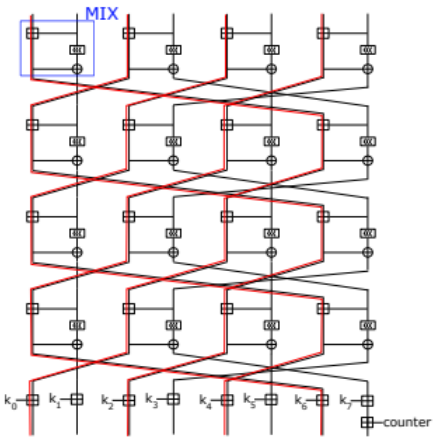
0	1	2	3	4	5	6	7
2	1	4	7	6	5	0	3

Tabela 9: Vrednosti funkcije $\pi(i)$ koja permutuje reči.

Potom, sve reči su permutovane operacijom PERM.Svaka runda primenjuje četiri paralelne MIX funkcije, kao što se vidi na slici 11, a potključevi se dodaju na svake četiri runde.

Skein (i prateći Threefish) je korišćen kao algoritam za proveru za rotacionu kriptoanalizu, zbog konstantne nepromenljive rotacije 0x555...555 koja se koristi u toku proširivanja ključeva i kod brojača male težine. Postavljanjem rotacione količine na 2, 42 runde su napadnute u radu [4]. Potom je napad kombinovan sa metodom refleksije i proširen na 55 rundi u [9]. Autori su uočili razliku izmedju teorijske rotacione verovatnoće (dobijene lemom 1) i eksperimentalne vrednosti, što možemo primetiti u tabeli 2.

Pokazuje se [12] da pojednostavljena verzija permutacije, gde su svi potključevi i brojači postavljeni na 0, ima daleko nižu rotacionu verovatnoću od onoga što bismo očekivali teoremom 4. Uočavaju se četiri paralelna lanca sabiranja po modulu na slici 12, koji pokrivaju reči stanja S[0], S[2], S[4], S[6], sa jednim sabiranjem po lancu po rundi. Primećujemo da ulazi u sabiranja koji dolaze iz drugih reči stanja podležu rotaciji, pa se stroga mogu smatrati nezavisnim. Dakle,



Slika 12: Četiri runde Threefish praćene sa sabiranjem podključa. Ukupno, Threefish koristi 18 takvih. Runde koriste različite količine rotacije.

za R rundi Threefish, dobijamo četiri lanca, svaki sa R sabiranja po modulu. Pošto nema konstantne, možemo podesiti količinu rotacije na 1. Tabela 5 jasno implicira da lanac dužine 28 ima rotacionu verovatnoću manju ili jednaku $2^{-126,9}$. Stoga, četiri lanca u 28 rundi imaju rotacionu verovatnoću oko 2^{-508}. Postavljajući rotacioni parametar na 2 (kao u prethodnoj kriptoanalizi Skein-a), smanjuje se broj rundi koje je moguće napasti na $24\,((2^{-122,0})^4 = 2^{-488})$.

5 Analiza algoritma SPECK alatkom ARXPy

U ovom poglavlju prikazuje se demonstracija rotacione analize algoritma SPECK alatkom ARXPy tool.

5.1 Algoritam SPECK

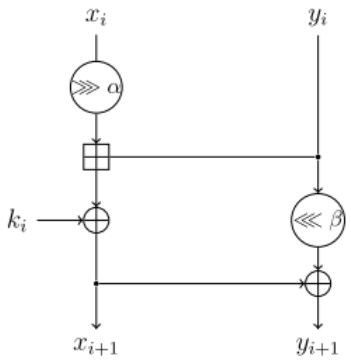
SPECK familija blok šifri je kolekcija 10 blok šifri dizajnirana od strane NSA u 2013. godini sa ciljem da zadovolji potrebu za efikasnošću i fleksibilnošću. Parametri za svaku verziju SPECK familije su dati u tabeli 7.

veličina bloka	veličina ključa	rot α	rot β	broj rundi T
32	64	7	2	22
48	72 ili 96	8	3	22 ili 23
64	96 ili 128	8	3	28 ili 29
96	96 ili 144	8	3	28 ili 29
128	128 ili 192 ili 256	8	3	32 ili 33 ili 34

Tabela 10: Parametri za deset verzija familije SPECK.

Izlaz $(x_{i+1}\|y_{i+1})$ jedne runde može da se izračuna na osnovu ulaza $(x_i\|y_i)$ po sledećem principu:

$$x_{i+1} = ((x_{i \ggg \alpha}) + y_i) \oplus k_i$$
$$y_{i+1} = (y_{i \lll \beta}) \oplus x_{i+1}$$



Slika 13: Jedna runda algoritma SPECK

5.2 Analiza algoritma SPECK

Definicija 15: RX karakteristika za par promenljivih (X, X') je par (α, γ) ako važi da je: $X \oplus (X' \leq_\gamma) = \alpha$.

Alat ArxPy je razvijen kako bi olakšao i ubrzao procenu bezbednosti ARX blok šifara. ArxPy je oruđe-instrument za pronalaženje optimalnih RX karakteristika ARX blok šifara. ArxPy koristi Python implementaciju ARX blok šifri kao ulaz i primenjuje SAT-zasnovanu metodu za pronalaženje tih karakteristika. Kada nam je data Python implementacija ARX blok šifre, ArxPy se izršava jednostavnom shell komandom, te je stoga jedino potrebno uložiti trud za implementaciju ARX šifri u Python-u. S obzirom na to da ArxPy ima otvoreni izvorni kod i modularnu arhitekturu, moguće je lako ga prilagoditi za neku specifičnu potrebu.

ArxPy očekuje određenu strukturu u Python implementaciji ARX blok šifri. Python implementacija ARX blok šifre koja zadovoljava traženu strukturu naziva se ARX implementacija. Minimalna ARX implementacija se sastoji od globalne promenljive `wordsize`, i dve funkcije: `key_schedule` i `encryption`. Globalna promenljiva `wordsize` sadrži veličinu reči (u bitovima) ARX blok šifre. Osim promenljive `wordsize` i funkcija `key_schedule` i `encryption`, moguće je definisati dodatne promenljive i funkcije, kako bi se poboljšala preglednost i modularnost implementacije.

Funkcija `key_schedule` implementira algoritam proširivanja ključa ARX blok šifre. Ova funkcija ima m argumenata, koji reprezentuju m reči ključa. Ova funkcija nema povratnu vrednost – potključevi se čuvaju u objektu nalik na listu, `round_keys`.

Funkcija `encryption` implementira algoritam šifrovanja ARX blok šifre. Argumenti ove funkcije predstavljaju reči teksta. Izlaz svake runde se čuva u objektu nalik na listu, rounds, osim poslednjeg izlaza, koji se koristi kao povratna vrednost. Ova funkcija može da sadrži rundne ključeve iz objekta `round_keys`.

Kako bi se implementirali algoritmi šifrovanja i proširivanja ključa, Python operatori +, $\ggg$, $\lll$ i $\oplus$ se koriste kao sabiranje po modulu, desna i leva rotacija i XOR, tim redosledom.

Postoji nekoliko ograničenja u vezi sa objektima rounds i round_keys:

- Oni nisu kreirani niti deklarisani u ARX implementaciji, nego su kreirani od strane parsera ArxPy
- Podržavaju jedino operator [ ] za pristup svojim elementima (negativni indeksi nisu dozvoljeni)
- Mogu da čuvaju ili jednu vrednost ili listu vrednosti, ali svaka pozicija može biti dodeljena samo jednom.

Kod 1: ARX implementacija SPECK32

```
wordsize = 16
number_of_rounds = 22
alpha = 7
beta = 2

//kruzna funkcija
def f(x, y, k):
    x = ((x >> alpha) + y) ^ k
    y =(y << beta) ^ x

def key_schedule (l2, l1, l0, k0):
    l = [None for i in range (number_of_rounds + 3)]

    round_keys[0] = k0
    l[0:3]=[l0,l1,l2]

    for i in range (number_of_rounds-1):
        l[i+3], round_keys[i+1] = f(l[i], round_keys[i], i)

def encryption (x0, y0):
    rounds[0] = f(x0, y0, round_keys[0])

    for i in range (1, number_of_rounds):
        x, y = rounds[i-1]
        round_output = f(x,y, round_keys[i])

        if i < number_of_rounds -1:
            rounds[i]= round_output
        else:
            return round_output
```

Sem funkcija encryption i key_schedule, ARX implementacija sadrži još dve specijalne funkcije: funkciju test i funkciju fix_differences. Test vektori se mogu dodati u ARX implementaciju korišćenjem Python naredbe assert unutar test funkcije. S druge strane, moguće je popraviti RX razlike potključeva i izlaza svake runde korišćenjem metode fix_difference na round_keys. Kod 2 je primer ovih funkcija za SPECK32.

Kod 2: Funkcije test i fixx_differences za SPECK32

```
def test():
    key = (0x1918, 0x1110, 0x0908, 0x0100)
    plaintext = (0x6574, 0x694c)
    ciphertext = (0xa868, 0x42f2)

    key_schedule(*key)
    assert ciphertext == encryption(*plaintext)

def fix_differences():
    for i in range(number_of_rounds):
        round_keys.fix.difference(i,0)
```

Pokretanje programa:

ArxPy koristi Python biblioteku SymPy i SMT rešavač STP. Oni, zajedno sa Python3, moraju biti instalirani, kako bi se izvršavao ArxPy.

Shell komanda za izvršavanje ArxPy je sledeća:

python3 arxpy.py <ARX_implementation> <output>

Ovde je <ARX_implementation> naziv fajla koji sadrži ARX implementaciju i <output> ime fajla gde se izlaz zapisuje. Nakon što je izvršavanje završeno, u izlaznom fajlu ostaje zapisana optimalna RX karakteristika. ArxPy zapravo traži par karakteristika: karakteristiku šifrovanja, koja sadrži RX razlike od izlaza svake runde, i karakteristiku proširivanja ključa, koja sadrži RX razlike rundnih ključeva. Kada ArxPy nađe par karakteristika, njihove RX razlike se zapisuju u izlazni fajl, zajedno sa informacijom i njihovoj verovatnoći.

6 Zaključak

Razvojem tehnologije i računarstva došlo je do poboljšavanja kriptografskih algoritama kako bi se povećala sigurnost sistema. Međutim, isti taj napredak doveo je do usavršavanja metoda otkrivanja nedostataka u kriptografskim shemama. S vremenom, metode postaju sve sofisticiranije i složenije, a usmeravaju se prema novim kriptografskim algoritmima. Za veliki broj algoritama slabosti se pronalaze ubrzo nakon predstavljanja.

Metode kriptografije obezbeđuju sigurnost i tajnost važnih informacija. Njihovo uvođenje i upotreba obezbedili su nesmetanu razmenu podataka na velike udaljenosti jer je dešifrovanje takvih podataka moguće samo uz poznavanje tajnih ključeva. Ipak, razvijeni su postupci koji su, zahvaljujući određenim nedostacima u kriptografskim shemama, omogućili otkrivanje tajnih ključeva za dešifrovanje. Takvi postupci su metodi kriptoanalize usmereni ka probijanju kriptografskih algoritma. Kako napredak tehnologije traje i dalje, može se očekivati razvoj novih metoda i usavršavanje starih postupaka za provođenje kriptoanalize. Kako su metode kriptoanalize usmerene prema sistemina koji još nisu probijeni, moguće je očekivati više napada na standard AES.

U radu je pokazano da je rotaciona analiza potencijalno efikasna za ARX sisteme. Kompleksnost rotacionog napada zavisi samo od broja sabiranja po modulu, a ne zavisi od broja operacija XOR i rotacija, kao ni količine rotacije. Pokazala sam da rotaciona verovatnoća ARX-a zavisi ne samo od broja sabiranja po modulu, već i od načina na koji su ta sabiranja povezana. Rotaciona verovatnoća lanca sabiranja po modulu ne može se izračunati kao proizvod verovatnoća pojedničanih sabiranja, jer pretpostavka Markovljeve šifre (koja je implicitno korišćena za računanje verovatnoće na taj način), ne važi. Stoga, lanac sabiranja ne može biti Markovljeva šifra, te je njegova verovatnoća manja i definisana *lemom 2*. Analiza koja je prikazana ovde sugeriše da način na koji su podključevi uključeni u šifru može da utiče na rotacionu verovatnoću ne samo zbog činjenice da modularno sabrani podključevi jednostavno uvećavaju broj sabiranja, već zbog toga sto XOR-ovani podključevi mogu da prekinu lanac sabiranja i stoga mogu da povećaju rotacionu verovatnoću.

Da sumiramo, rotaciona verovatnoća kriptografskih primitiva ne mora obavezno biti jednaka proizvodu rotacionih verovatnoća individualnih operacija korišćenih u primitivi. Takvo skraćivanje pri proceni verovatnoće ne daje ni gornju ni donju granicu stvarne verovatnoće. Ova procena se jedino može koristiti nakon sto utvrdimo da je zadovoljena Markovljeva pretpostavka primenjena na primitive, inače, rotacionu verovatnoću moramo da računamo ad-hoc. Sa druge strane, pokazala sam i da svaka funkcija može biti implementirana koristeći operacije ARX i konstante.

7 Reference

[1] Alfred Menezes, Paul C. van Oorschot, and Scott A. Vanstone. Handbook of Applied Cryptography. CRC Press, 1996.

[2] Jian Guo, Pierre Karpman, Ivica Nikolić, Lei Wang, and Shuang Wu. Analysis of BLAKE2. LNCS 8366, 2014, pp. 402 - 423.

[3] Dmitry Khovratovich, Ivica Nikolić, Josef Pieprzyk, Przemyslaw Sokolowski and Ron Steinfeld. Rotational Cryptanalysis of ARX Revisited. LNCS 9054, 2015, pp. 519 - 539.

[4] Dmitry Khovratovich and Ivica Nikolić. Rotational Cryptanalysis of ARX. LNCS 6147, 2010, pp. 333 - 346. Dmitry Khovratovich, Ivica Nikolić, and Christian Rechberger. Rotational Rebound Attacks on Reduced Skein. LNCS 6477, 2010, pp. 1 - 19.

[5] Bart Preneel, Ren Govaerts, and Joos Vandewalle. Hash Functions Based on Block Ciphers: A Synthetic Approach. LNCS 773, 1994, pp. 368 - 378.

[6] Przemyslaw Sokolowski. Contributions to Cryptanalysis: Design and Analysis of Cryptographic Hash Functions. PhD thesis, Macquarie University, 2013. Available online at http://hdl.handle.net/1959.14/307348

[7] Jean-Philippe Aumasson, Samuel Neves, Zooko Wilcox-O'Hearn, and Christian Winnerlein. BLAKE2: Simpler, Smaller, Fast as MD5 LNCS 7954, 2013, pp. 119 - 135.

[8] Przemyslaw Sokolowski. Design and Analysis of Cryptographic Hash Function. Adam Mickiewicz University in Poznan, Faculty of Mathematics and Computer Science

[9] Khovratovich D., Nikolic I., Rechberger C.: Rotational rebound attacks on reduced Skein. J.Cryptology 27(3), 452-479 (2014.)

[10] Jian Guo, Pierre Karpman, Ivica Nikolic, Lei Wang, Shuang Wu. Analysis of BLAKE, Nanyang Technological University Singapore, 2013,

[11] Lai X., Massey J.L.: Markov cipher and differential cryptoanalysis. In: Davies, D.W.(ed.), EUROCRYPT 1991. LNCS, vol. 547, pp 17-38., Springer, Heidelberg (2012)

[12] N. At, J. Beuchat, and I. San, Ćompact Implementation of Threefish and Skein on FPGA,"in 5th IFIP International Conference on New Technologies, Mobility and Security, IEEE Press, 2012.

[12] L. P. Oommen, and Anas A. S., Škein and Threefish Implementation on FPGA,"International Journal of Science and Research (IJSR), vol. 4, no. 5, pp.1493-1496, 2015.

[13] P. Gayathri, K. Sateesh, and C. Navya, "High-Throughput Hardware Implementation of Three Fish Block Cipher Encryption and Decryption on FPGA,"International Journal of VLSI System Design and Communication Systems, vol. 3, no. 8, pp. 1325-1329, 2015.

[14] Daum, M.: Cryptanalysis of Hash Functions of the MD4-Family. PhD thesis, Ruhr-Universität Bochum (May 2005)

[15] Matthew A. Kupinski, John W. Hoppin, Eric Clarkson, and Harrison H. Barrett: Ideal-observer computation in medical imaging with use of Markov-chain Monte Carlo techniques, 2003.

[16] D.S.Mayers, L. Wallin, P. Wikstrom: An introduction to Markov chains and their applications within finance, 2017.

[17] B. Sericola: Markov Chains - Theory, Algorithms and Applications. London: ISTE Ltd and John Wiley & Sons Inc, 2013.

[18] S. Mehta: Five real-world uses of the Markov chains, Developers corner, 2022.

Biografija autora:

Teodora Macanović rođena je 2.8.1996. u Beogradu. Osnovnu školu završila je 2011.godine kao vukovac i đak generacije. Iste godine upisuje Devetu gimnaziju „Mihailo Petrović Alas", društveno-jezički smer. Gimnaziju završava 2015.godine i iste godine upisuje Matematički fakultet Univerziteta u Beogradu. Osnovne akademske studije završava 2020.godine, a potom na istom fakultetu upisuje i master studije, na smeru Matematika, modul Računarstvo i informatika. Oblast interesovanja: kriptografija.