UNIVERZITET CRNE GORE

Prirodno-matematički fakultet Podgorica

Aleksandar Zeković

Kriptografski napadi podataka na internetu

SPECIJALISTIČKI RAD

Podgorica, 2018.

UNIVERZITET CRNE GORE

Prirodno-matematički fakultet Podgorica

Kriptografski napadi podataka na internetu

SPECIJALISTIČKI RAD

Matematika

Mentor: Vladimir Božović

Aleksandar Zeković

Studijski program Matematika

Podgorica, Septembar 2018.

Posveta

Zahvaljujem se porodici, prijateljima i mentoru.

Apstrakt

Sigurnost razmjene podataka je izuzetno opširna tema koja sadrži definisane probleme, metode, sredstva i ciljeve. Cilj ovog rada jeste objašnjenje šifri, sa akcentom na moderne šifre, klasifikovanje kriptografskih napada i izlaganje osnovnih termina neophodnih za razumijevanje istih.

Abstract

Security data exchange is an extremely comprehensive topic that contains defined problems, methods, resources and goals. The aim of this paper is to explain the codes, with the emphasis on modern codes, classification of cryptographic attacks and exposure of basic terms necessary to understand them.

Sadržaj

Glava 1

Uvod

Kriptografija je nauka o sigurnim metodama komunikacije. Predmet proučavanja moderne kriptografije su različiti aspekti informacione bezbjednosti kao što su integritet i vjerodostojnost podataka, tajnost podataka, autentifikacija, nemogućnost izbjegavanja odgovornosti. Kriptografski napadi su dio kriptoanalize, nauke koja, suprotno kriptografiji, za predmet proučavanja ima dešifrovanje šifrovanih podataka.

U prvom poglavlju ćemo objasniti šifre, dati akcenat na moderne šifre, klasifikovanje kriptografskih napada i izlaganje osnovnih termina neophodnih za razumijevanje istih.

U drugom poglavlju je data podjela šifri.

U trećem poglavlju je izložena podjela kriptografskih napada i data su objašnjenja značajnijih napada.

U četvrtom poglavlju bavimo se sa RSA šifrom koja spada u jednu od često upotrebljenih savremenom šifara, prikazani su neki od napada na ovu šifru i date su procjene o njegovoj sigurnosti i budućem korišćenju u savremenim tehnologijama.

U petom poglavlju je pokazana implementacija Vinerovog napada.

Poslednje poglavlje je ujedno i zaključak, i odnosi se na trenutno stanje i predviđanja daljeg razvoja.

Glava 2

Kriptografske šifre

Šifra (engl. cipher), u kontekstu kriptografije, označava algoritam za šifrovanje i dešifrovanje, tj. šifra predstavlja jedan kriptografski sistem. ***Otvoreni tekst*** (engl. plaintext) je originalna poruka koju treba zaštititi. ***Šifrovani tekst*** (engl. ciphertext) je dobijen šifrovanjem otvorenog teksta. U kriptografiji ***ključem*** nazivamo informaciju tj. parametar koji određuje izlaz šifre. Ključ određuje bijektivno preslikavanje otvorenog teksta u šifrovani tekst u slučaju šifrovanja kao i inverzno preslikavanje šifrovanog u otvoreni tekst u slučaju dešifrovanja [3]. Danski kriptograf *August Kirkofs* izložio je princip koga se i danas dizajneri šifri pridržavaju. Po tom principu bezbjednost šifre u potpunosti zavisi od tajnosti ključa koji se koristi, a ne od tajnosti algoritma za šifrovanje. *Klod Šanon* je ovaj princip preformulisao u maksimu *"Neprijatelj poznaje sistem"*, i definisao je dvije operacije, difuziju i konfuziju, koje kada su zadovoljene garantuju sigurnost date šifre.

Konfuzija je operacija koja obezbjeđuje da relacija između ključa i šifrovanog teksta bude nejasna, tj. neshvatljiva napadaču. Konfuzija se postiže supstitucijom.

Difuzija je operacija kojom se uticaj jednog simbola (bita) otvorenog teksta širi na više simbola (bita) šifrovanog teksta, sa ciljem prikrivanja statističkih karakteristika otvorenog teksta. Većina algoritama za šifrovanje je objavljena i poznata

javnosti.

Šifre se dijele na klasične i moderne.

Klasične šifre dijele se na:

- ***Šifre transpozicije*** (engl. transposition cipher) - šifrovani tekst dobija se mijenjanjem redosleda slova, simbola ili bita u otvorenom tekstu.

- ***Šifre substitucije*** (engl. substitution cipher) - šifrovani tekst nastaje tako što se slova, simboli ili biti iz otvorenog teksta mijenjaju drugim slovima, simbolima ili bitima, respektivno, po nekoj utvrđenoj logici. Mogu biti polialfabetske, kod kojih se za substituciju koristi više alfabeta, i monoalfabetske, kod kojih se substitucija obavlja u domenu jednog alfabeta.

Podjela modernih šifara može se izvršiti u odnosu na ključ koji koriste i u odnosu na tip ulaznih podataka.

Prema ključu šifre mogu biti:

- ***Simetrične*** (engl. symmetric key cipher) - šifrovanje i dešifrovanje se vrše istim, tajnim ključem. Kriptografija od antičkih vremena pa do 1976. godine je bila isključivo zasnovana na ovoj metodi. Simetrične šifre i danas imaju široku primjenu naročito za šifrovanje podataka i provjeru integriteta poruke. Postoje dva tipa simetričnih šifri:

  - ***Šifre niza*** (engl. stream cipher) - šifrovanje otvorenog teksta se vrši bit po bit (slovo po slovo)

  - ***Šifre bloka*** (engl. block cipher) - otvoreni tekst se dijeli u blokove fiksne dužine i vrši se šifrovanje na nivou bloka.

- ***Asimetrične*** (engl. asymmetric key cipher) – 1976. godine ovaj tip šifre su

uveli *Vitfild Difi*, *Martin Helman* i *Ralf Merkle*. Šifrovanje se vrši pomoću javnog ključa, a dešifrovanje pomoću privatnog tj. tajnog ključa. Asimetrično šifrovanje se tipično koristi kod autentifikacije i digit pravnih dokumenata u elektronskoj formi), kao i za razmjenu kljuceva (engl. key establishment) simetričnih šifri [4].

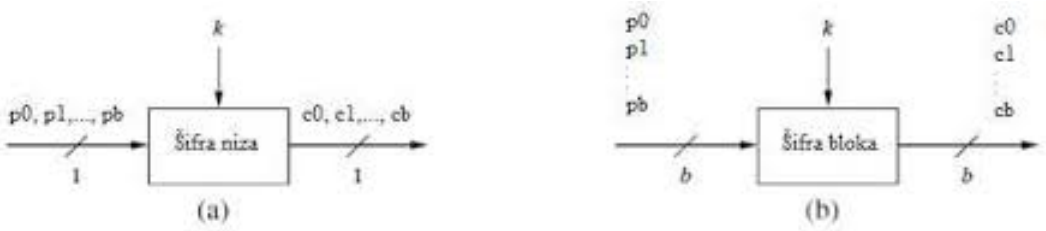
2.1 Simetrične šifre

Princip funkcionisanja simetričnih šifri je najlakše objasniti na primjeru. Dva korisnika A i B razmjenjuju podatke preko kanala koji nije zaštićen. Napadač pokušava da pristupi kanalu i na taj način vrši neautorizovano slušanje. Da bi se napadač u tom onemogućio koristi se simetrična šifra. Korisnik A šifruje svoju poruku P i prosleđuje je korisniku B. Korisnik B po prijemu poruke vrši dešifrovanje. Ako korisnici A i B imaju jak sistem za šifrovanje napadač neće biti u mogućnosti da protumači poruke koje se razmijenjuju iako je ostvario pristup kanalu. Ono što je neophodno jeste nalaženje sigurnog načina da korisnici A i B razmijene ključ K koji će koristiti za šifrovanje i dešifrovanje. Primjer protokola za sigurnu razmjenu ključeva je *WPA* (engl. Wi-Fi Protected Access) u bežičnim LAN mrežama. Korisnici A i B samo jednom vrše razmjenu ključa nakon čega se sva naredna komunikacija šifruje i dešifruje tim ključem. Ukoliko napadač ima pristup ključu komunikacija prestaje da bude sigurna pa je bezbjedna razmjena i čuvanje ključeva nužna.

Navedenim primjerom je pokazano kako se sadržaj poruke može sakriti od napadača. Kriptografija omogućava ne samo povjerljivost podataka već i čuvanje integriteta poruke (napadač ne može izmijeniti sadržaj poruke), i autentifikaciju poruke (garantuje se korisniku B da je poruku poslao korisnik A, i obratno).

Šifre niza i šifre bloka se mogu lako razlikovati. Na slici 2.1 je predstavljena

operativna razlika između šifre niza (slika 2.1 a) i šifre bloka (slika 2.1 b), kada se vrši šifrovanje b bita, gdje je b širina blok šifre (odnosno veličina bloka).



Slika 2.1: a) Šifra niza b) Šifra bloka

Šifre niza vrše šifrovanje bit po bit. To se postiže dodavanjem jednog bita ključa jednom bitu otvorenog teksta. Blok šifre šifruju blokove bita otvorenog teksta istim ključem. To znači da šifrovanje jednog bita otvorenog teksta zavisi od svih ostalih bita otvorenog teksta istog bloka. Velika većina blok šifri ima dužinu bloka 128 bita (AES) ili 64 bita (DES, 3DES).

U praksi se više koriste blok šifre, naročito za šifrovanje komunikacije na Internetu. Šifre niza su relevantne za aplikacije koje imaju malu *računarsku moć* (engl. Computing power) kao što su mobilni telefoni. Primjer takve šifre je *A5/1* koja je dio GSM standarda i koristi se za voice šifrovanje. Šifre niza se mogu koristiti i za šifrovanje Internet saobraćaja, naročito *RC4* šifra.

2.1.1 Šifre niza

Otvoreni tekst, šifrovani tekst i ključ označavamo redom sa $P_i, C_i, K_i \in \{0, 1\}$. Funkcije šifrovanja i dešifrovanja označavamo redom sa E_k i D_k. Šifrovanje se vrši prema formuli:

$$C_i = E_{K_i}(P_i) \equiv P_i + K_i \pmod 2,$$

a dešifrovanje:

$$P_i = D_{K_i}(C_i) \equiv C_i + K_i \pmod 2.$$

Iz prethodnih formula se vidi da su funkcije šifrovanja i dešifrovanja iste.

One-Time Pad je simetrična šifra čija je bezbjednost dokazana, ali je veoma nepraktična za upotrebu jer dužina ključa mora biti jednaka dužini otvorenog teksta, tj. jedan bit ključa se koristi za šifrovanje tačno jednog bita otvorenog teksta.

2.1.2 Šifre bloka

Kao što je pomenuto ranije šifre bloka tretiraju blok otvorenog teksta, dužine koja je karakteristična za datu šifru bloka. Postoji nekoliko režima šifrovanja, tj. načina upotrebe blok šifri za šifrovanje velikih otvorenih tekstova čija dužina prelazi dužinu bloka date šifre. Neki od osnovnih režima šifrovanja su: *ECB* (engl. Electronic Code Book), *CBC* (engl. Cipher Block Chaining), *CFB* (engl. Cipher Feedback mode), *OFB* (engl. Output feedback mode), *CTR* (engl. Counter). *ECB* i *CFB* zahtijevaju da dužina otvorenog teksta bude cjelobrojni proizvod dužine bloka šifre koju koriste. Ukoliko to nije slučaj primjenjuje se padding na otvoreni tekst.

Padding je tehnika dopunjavanja otvorenog teksta bitima do određene dužine sa ciljem sakrivanja statističkih karakteristika otvorenog teksta, postizanja željene dužine otvorenog teksta ili da bi se sakrila od napadača stvarna dužina otvorenog teksta. Jedna padding metoda je da se na otvoreni tekst doda jedna jedinica i onoliko nula koliko je potrebno da otvoreni tekst bude proizvod dužine bloka.

ECB je najjednostavniji režim šifrovanja. Otvoreni tekst se izdijeli na blokove čija je dužina ista kao dužina bloka šifre koju koristimo, zatim se svaki blok otvorenog teksta šifruje pojedinačno. Prednost ovakvog šifrovanja je što ne zahtijeva sinhronizaciju između predajne i prijemne strane. Ukoliko dođe do problema u prenosu i neki blokovi šifrovanog teksta ne stignu do prijemnika biće moguće dešifrovati samo

pristigle blokove.

CBC režim šifrovanja nije deterministički, šifrovani tekst se randomizuje tako da ukoliko jedan isti otvoreni tekst šifrujemo više puta dobićemo različite šifrovane tekstove. Šifrovanje svih blokova je povezano tako da šifrovani blok C_i zavisi ne samo od otvorenog bloka C_i već i od svih prethodno šifrovanih blokova otvorenog teksta.

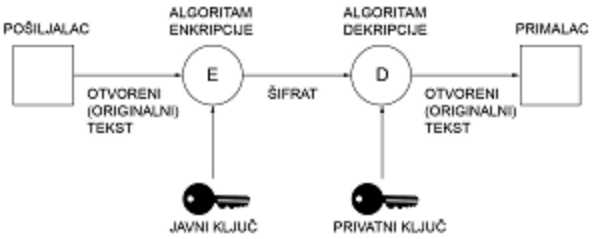
OFB se koristi za izgradnju šifri niza, gdje se ključ šifre ne generiše na nivou bita, kao kod šifri niza, već na nivou bloka. I ovaj režim kao rezultat daje nedeterministički šifrovani tekst.

CTR se takođe koristi za izgradnju šifri niza. Ključ šifre se generiše na nivou bloka.

2.2 Asimetrične šifre

Difi, Helman i Merkle su 1976. godine predložili novu vrstu šifrovanja koja se zasniva na ideji da tajnost ključa za šifrovanje nije neophodna već da je dovoljno osigurati tajnost ključa za dešifrovanje. Kod asimetričnih šifri šifrovanje se vrši javnim ključem k_{jav}, koji je dostupan svima na uvid, a dešifrovanje se vrši tajnim ključem k_{pr}, koji je poznat samo ovlašćenim korisnicima.

Tu je potrebno samo prvi put na siguran način razmeniti privatni ključ koji će onda imati samo lica koja primaju poruke, a javni ključ je praktično poznat i dostupan svima, pa čak i napadacima. Baš iz tog razloga u ovim situacijama nije moguće do ključa dekripcije doći samo sa znanjem i informacijama o ključu enkripcije, makar ne u praksi, tj. u razumnom vremenu.



Slika 2.2: Ilustracija kriptografskog procesa sa javnim ključem (asimetričnog procesa)

Kao interesantana simulacija takvog događaja može se razmotriti sledeća situacija: zamislite da imate standardan telefonski imenik (klasifikovan abecedno po imenima stanovnika) višemilionskog grada, te imate prvi zadatak: naći broj telefona po imenu i prezimenu čovjeka. Vrlo lako za uraditi, čak i ukoliko ima više od jednog čovjeka pod istim imenom, za nekoliko trenutaka izbor se svodi na nekoliko ljudi. Međutim, drugi zadatak se sastoji u tome da za dati telefonski broj trebate identifikovati kome zapravo pripada. Taj zadatak, ukoliko je u pitanju višemilionski grad, gdje trebate "ručno" pretražiti milione ljudi da biste došli do vlasnika broja čini se neizvodiv, ili makar neizvodiv u razumnom vremenu. Ovakav mehanizam asimetricnih metoda se matematički može opisati jednosmjernim *one-way* funkcijama.

Definicija 2.1. *Funkciju $f : X \to Y$ nazivamo* ***jednosmjernom funkcijom*** *ako ona može biti, za proizvoljan elemenat x iz domena X, lako izračunata (u polinomnom vremenu), ali za proizvoljan elemenat $y \in f(X)$ je mnogo teže, praktično nemoguće u polinomnom (razumnom) vremenu, pronaći $x \in X$, takvo da važi $y = f(X)$*

Definicija 2.2. *Jednosmernu funkciju $f : X \to Y$ nazivamo* ***funkcijom sa skrivenim vratima*** *(engl: trapdoor function) ako, uz neku dodatnu informaciju, postaje računski izvodljivo $\forall y \in f(X)$ naći elemenat iz domena $x \in X$ takav da je $f(X) = y$.*

Uz pomoć ove matematičke terminologije kratak opis mehanizma metoda sa javnim ključem bio bi sledeći: svaki korisnik javno podeli svoj ključ za kriptovanje kako

bi svako ko želi mogao bezbedno komunicirati sa njim. Ključ za kriptovanje predstavlja jednu *trapdoor* funkciju, a upravo samo on poseduje dodatne informacije, pomoću kojih će samo on moći naći inverz funkcije sa skrivenim vratima, tjst. funkciju dekripcije, da bi dekriptovao poruke koje dobija.

Glava 3

Kriptografski napadi

Uspješan napad na kriptografski sistem odnosno šifru podrazumijeva nalaženje praktičnog načina da napadač od šifrovanog teksta dobije otvoreni tekst [2]. Cilj napada nije nužno dekriptovanje samo jedne šifrovane poruke već sticanje informacije o ključu koji se koristi u datom sistemu i na taj način kompromitovanje cjelokupne prošle i buduće komunikacije.

Kriptografski napadi se mogu klasifikovati na više načina. Radi bolje preglednosti navodimo napade koji se baziraju na otvorenom/šifrovanom tekstu, kao i poznate napade na klasične, simetrične i asimetrične šifre.

U odnosu na tip informacije koju napadač posjeduje, kriptografski napadi mogu biti podijeljeni na [1]:

- Napadi zasnovani na poznatom otvorenom tekstu:
  - ***Poznati otvoreni tekst*** (engl. known plaintext)
  - ***Odabrani otvoreni tekst*** (engl. chosen plaintext)
  - ***Adaptivni odabrani otvoreni tekst*** (engl. adaptive chosen plaintext)
- Napadi zasnovani na poznatom šifrovanom tekstu:

  - ***Šifrovani tekst*** (engl. ciphertext only)
  - ***Odabrani šifrovani tekst*** (engl.chosen ciphertext)
  - ***Adaptivni odabrani šifrovani tekst*** (engl.adaptive chosen ciphertext)

Neki od poznatijih napada na klasične šifre, koji ujedno pripadaju grupi napada šifrovani tekst, su [5]:

- ***Analiza učestanosti*** (engl. Frequency analysis)
- ***Računanje podudaranja*** (engl. Coincidence counting)
- ***Kasiski ispitivanje*** (engl. Kasiski examination)

Poznati napadi na simetrične šifre su:

- ***Diferencijalna kriptoanaliza*** (engl. Differential cryptanalysis)
- ***Linearna kriptoanaliza*** (engl. Linear cryptanalysis)
- ***Dejvisov napad***
- ***Related key napad***
- ***Meet-in-the-middle napad***
- ***Brute force napad***
- ***Standardni ASCII napad*** (engl. Standard ASCII attack)

Standardni ASCII i Brute force napad pripadaju grupi napada šifrovani tekst. Diferencijalna i linearna kriptoanaliza, Dejvisov napad, Meet-in-the-middle napad pripadaju grupi napada poznat otvoren tekst.

3.1 Napadi zasnovani na poznatom otvorenom/šifrovanom tekstu

3.1.1 Poznati otvoreni tekst

Kod napada poznat otvoren tekst napadač ima pristup i otvorenom tekstu i njemu odgovarajućem šifrovanom, i sprovodi analizu datih podataka sa ciljem pronalaženja ključa koji se koristi za šifrovanje. Klasične šifre su podložne ovakvoj vrsti napada dok su moderne šifre otporne. U Drugom svjetskom ratu njemačka vojska je koristila ENIGMA mašinu za šifrovanje vojnih poruka, i dok su najviši redovi u vojsci znali za opasnost napada poznat otvoreni tekst operateri na terenu nisu vodili računa o tome pa su britanski kriptolozi bili u stanju da pretpostave značenje pojedinih djelova šifrovanog teksta. Na primjer, svakog dana u isto vrijeme su razmjenjivane poruke o vremenskim uslovima i riječ *vrijeme* (germ. Wetter) se pojavljivala svakog dana u svakoj poruci na tačno određenom mjestu, što je kriptografima dalo osnova za na pad poznat otvoren tekst. Najpoznatiji napad otvoren tekst na savremenu šifru je bio na PKZIP šifru niza. Ako napadač ima zip fajl šifrovan PZKIP šifrom potreban mu je samo jedan otvoreni tekst koji pripada zip arhivi da bi uspješno dešifrovao cijeli zip fajl. Napad neće biti uspješan ako su PKZIP fajlovi šifrovani AES šifrom.

3.1.2 Odabrani otvoreni tekst

Kod odabranog otvorenog teksta napadač može sam da bira otvorene tekstove i može da vidi njima odgovarajuće šifrovane tekstove. Ovaj tip napada se najčešće koristi za napade na asimetrične šifre kod kojih napadač, pošto zna javni ključ, može da šifruje otvorene tekstove po svom izboru. Ukoliko je šifra ranjiva na napad poznat otvoreni tekst onda je nužno ranjiva i na ovaj napad, ali ne mora važiti obrnuto. U

savremenoj kriptoanalizi primjer ovakvog napada je diferencijalna kriptoanaliza.

3.1.3 Adaptivni odabrani otvoreni tekst

Napadač ima pristup šifri tako da može da zada jedan otvoren tekst, dobije šifrovani rezultat, a zatim bira sledeći otvoreni tekst koji će šifrovati ali tako da postoji veza između dva otvorena teksta sa ciljem nalaženja veze između dva rezultujuća šifrovana teksta. Slična tehnika je korišćena u Drugom svjetskom ratu kada bi britanska vojska napadala dobro poznate lokacije u Njemačkoj a analitičari pratili šifrovane izvještaje o tim napadima.

3.1.4 Šifrovani tekst

Napadač ima šifrovani tekst, ali ne i njemu odgovarajući otvoreni tekst, tj. napadač može pasivno da "sluša" šifrovanu komunikaciju. Napadač može da pretpostavi neke karakteristike otvorenog teksta, npr. da je otvoreni tekst kodiran ASCII kodom i da je napisan na engleskom jeziku. U tom se slučaju pristup otvorenom tekstu, bez otkrivanja ključa, smatra uspješno ostvarenim napadom. Ovo je najteži tip napada jer napadač raspolaže sa malo informacija.

3.1.5 Odabrani šifrovani tekst

Napad je isti kao odabrani otvoreni tekst samo što sada umjesto funkcije šifrovanja posmatramo dešifrovanje. Napadač prikuplja informacije tako što odabere šifrovani tekst i posmatra dešifrovani, otvoreni, tekst, pri čemu nema pristup ključu već samo rezultatu dešifrovanja. Relevantan je u slučaju asimetrične kriptografije kada su napadaču zbog javnog ključa za šifrovanje na raspolaganju i poznati otvoreni tekstovi i odabrani poznati tekstovi.

3.1.6 Adaptivni odabrani šifrovani tekst

Ovaj napad je varijacija napada odabrani šifrovani tekst. Napadač bira određeni broj šifrovanih tekstova koje dešifruje, pri čemu ima pristup samo funkciji šifrovanja, nema pristup ključu, i to tako da svaki sledeći šifrovani tekst koji dešifruje bira na osnovu prethodno dešifrovanih tekstova. To jest, postoji povratna sprega u odlučivanju koji naredni šifrovani tekst se dešifruje. Na taj način se postepeno otkrivaju informacije o ključu. U slučaju asimetrične kriptografije ovaj napad se može primijeniti na šifre koje su *malleable*, tj. šifre kod kojih napadač može presresti i izmijeniti šifrovani tekst, i to na takav način izmijeniti da je promjena u otvorenom tekstu predvidljiva.

Rane verzije RSA padding korišćene u SSL protokolu su bile podložne napadu adaptivni odabrani šifrovani tekst, kojim su napadači otkrivali ključeve SSL sesije.

3.2 Napadi na klasične šifre

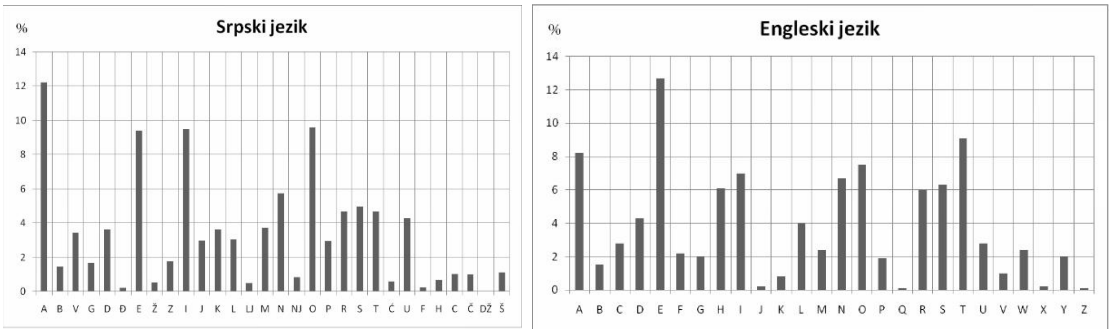
Klasična kriptoanaliza je najstariji oblik analize kriptografskih šifri i u obzir uzima pravilnosti jednog jezika.

3.2.1 Analiza učestanosti

Za svaki jezik, na dovoljno velikom uzorku teksta, mogu se uočiti pravilnosti po pitanju učestanosti pojavljivanja pojedinačnih slova, kombinacije dva odnosno tri slova, itd. Tačno određena distribucija pojave slova postoji za svaki jezik. Nekada je moguće navedene karakteristike otvorenog teksta uočiti i u šifrovanom tekstu i iskoristiti ih u napadu "samo šifrovani tekst" [5].

Osnovna olakšica koja se koristi pri napadu na klasične šifre jeste unilateralna frekvencijska distribucija koja se dobija prebrojavanjem slova u tekstu, slovo po slovo,

kao što je to prikazano na slici 3.1.



Slika 3.1: Najčešća slova u Srpskom i Engleskom jeziku

Na velikom uzorku teksta, kao što se vidi na prethodnim slikama, uviđaju se pravilnosti unutar jezika, tako da je u srpskom jeziku najčešće slovo A (u engleskom slovo E).

U šiframa transpozicije frekvencijska distribucija slova u šifrovanom tekstu će biti ista kao kod otvorenog teksta. U najjednostavnijoj šifri substitucije svako slovo iz otvorenog teksta odgovara jednom slovu iz šifrovanog teksta. To znači da frekvencijska distribucija slova u šifrovanom tekstu ne će biti ista kao u otvorenom tekstu, ali će u konačnom rezultatu figurirati isti brojevi - ako smo u otvorenom tekstu imali 33 slova A i ako se slovo A nakon šifrovanja zapisuje slovom G, onda ćemo u šifrovanom tekstu imati 33 slova G. Kompleksnije šifre substitucije, kao što su polialfabetske, jedno isto slovo će pri svakom sledećem pojavljivanju biti šifrovano drugim slovom, i na taj način šifrovani tekst ima druga čiju frekvencijsku distribuciju od otvorenog teksta, pa napad analizom učestanosti može biti neuspješan [8].

3.2.2 Računanje podudaranja

Računanje podudaranja je metoda koja se koristi uz analizu učestanosti. ***Indeks podudaranja*** (engl. Index of coincidence) je mjera vjerovatnoće da pri nasumičnom

odabiru dva slova iz jednog teksta, ta dva slova budu ista. Indeks podudaranja se koristi u napadima na šifre substitucije, i može se iskoristiti za otkrivanje dužine ključa. To se ostvaruje tako što poredimo slovo po slovo (bajt po bajt) šifrovani tekst sa istim tim tekstom pomjerenim za određeni broj karaktera, pri čemu pomjeraj odgovara dužini ključa koja se testira. Za svaku dužinu ključa koja se testira napadač računa indeks podudaranja i čuva rezultate. Kada se dobije indeks podudaranja koji je blizak očekivanom za dati jezik kojim je napisan tekst, znači da smo pronašli dužinu ključa.

Ako imamo tekst dužine n napisan na jeziku čiji alfabet ima N slova, i u tom tekstu imamo redom n_1, n_2, ..., n_N pojedinačnih slova, gdje je

$$n = \sum_{n=1}^{N} n_i$$

Ukupan broj parova istih slova u tekstu U_1 je:

$$U_1 = \sum_{n=1}^{N} \frac{n_i(n_i - 1)}{2}$$

Ukupan broj parova slova u tekstu U je:

$$U = \frac{n(n-1)}{2}$$

Vjerovatnoća da su dva slova jednog para ista predstavlja indeks podudarnosti i iznosi:

$$IC = \frac{U_1}{U} = \frac{\sum_{n=1}^{N} \frac{n_i(n_i - 1)}{2}}{\frac{n(n-1)}{2}} = \frac{\sum_{n=1}^{N} n_i(n_i - 1)}{n(n-1)}$$

Za različite jezike IC se razlikuje jer svaki jezik ima sebi svojstvenu frekvencijsku distribuciju slova.

3.2.3 Kasiski ispitivanje

Kasiski ispitivanje je metod napada na polialfabetske šifre substitucije i razvijeno je kao metod razbijanja *Viženerove šifre*. Kod šifrovanja *Viženerovom* šifrom bira se *ključna riječ* (engl. Key word), zatim se ta ključna riječ napiše onoliko puta koliko je potrebno da dužina tako nastalog *niza* (engl. Keystream) dostigne dužinu otvorenog teksta, nakon čega se vrši supstitucija svakog slova otvorenog teksta jednim slovom šifrovanog teksta. Šifrovanje se vrši uz pomoć Tabula recta, tabelom sa abecedom jezika na kom je napisan otvoreni tekst, kod koje je svaki red nastao pomjeranjem prethodnog reda za jedno mjesto ulijevo. Kada jedno slovo šifrujemo, posmatramo koje slovo niza ključne riječi mu odgovara, a zatim u Tabula recta tražimo šifrovano slovo u presjeku reda koji počinje datim slovom otvorenog teksta i kolone koja počinje odgovarajućim slovom niza ključne riječi. Kasiski ispitivanjem se dolazi do saznjanja o dužini ključne riječi koja se koristi tako što se uočavaju ponavljajuće strukture u šifrovanom tekstu i njihova rastojanja. Ako se na primjer u šifrovanom tekstu uoči niz od 4 uzastopna slova koji se pojavljuje bar još jednom u tekstu, i ako je njihovo rastojanje 15 slova onda je dužina ključne riječi broj koji je činilac od 15 (1, 3, 5, 15). Ukoliko imamo više ponavljajućih struktura u tekstu onda pretpostavljamo da je dužina ključne riječi najveći broj koji se pojavljuje kao činilac svih nađenih rastojanja. Nakon pretpostavljanja ključne riječi šifrovani tekst će se raspisati u obliku matrice kod koje je broj kolona jednak dužini ključne riječi. Jedna kolona matrice je šifrovana jednim slovom ključne riječi. Izvršimo frekvencijsku analizu svake kolone ovako napisanog šifrovanog teksta, i svaku kolonu pojedinačno dešifrujemo korišćenjem frekvencijske analize.

3.3 Napadi na simetrične šifre

3.3.1 Diferencijalna kriptoanaliza

Definicija 3.1 (***XOR profili***). *Funkcija koja preslikava ulazni string S-bloka*

$$f : \sum^n \to \sum^m$$

gdje su s_1,$s_2 \in \sum^n$, *tada su* $f(s_1)$,$f(s_2) \in \sum^m$ *odnosno* s_1^*,$s_2^* \in \sum^m$, *gdje su*

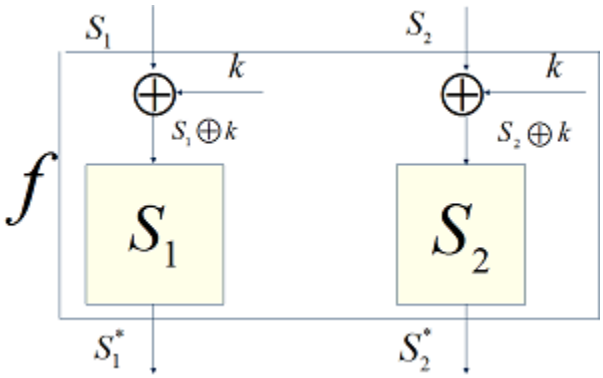
$$s_1^* = f(s_1),\ s_2^* = f(s_2)$$

Definišimo $\delta = s_1 \oplus s_2$, $\Delta = s_1^* \oplus s_2^*$ i četvorku

$$S_\Delta^\delta = \{(s_1, s_2; s_1^*, s_2^*), |\forall(s_1 \oplus s_2 = \delta, s_1^* \oplus s_2^* = \Delta)\}$$

gdje $|S_\Delta^\delta|$ označava broj četvorki u skupu.

Primjer 3.1. $S_2^{3C} = \{(3, 3F, F, D), (17, 2B, B, 9), (2B, 17, 9, B), (3F, 3, D, F)\}$, *tada je* $|S_2^{3C}| = 4$



XOR profil S-bloka definisan sa $f : \sum^n \to \sum^m$ je tabela koja ima 2^n redova i 2^m kolona. Svaki red i kolona indeksiraju se sa δ i Δ respektivno. Svaki unos (δ, Δ) u tabelu prikazuje broj elemenata u skupu S_Δ^δ

Primjer 3.2. *Neka je skup*

$$S_{1_x}^{19_x} = \{(2_x, 1B_x; 4_x, 5_x), (1B_x, 2_x; 5_x, 4_x),$$
$$(22_x, 3B_x; 1_x, 0_x), (2C_x, 35_x, 2_x, 3_x), (35_x, 2C_x, 3_x, 2_x), (3B_x, 22_x, 0_x, 1_x)\}.$$

Tada je unos $(1, 19)$ *u tabeli* $|S_\Delta^\delta| = 6$

Svojstva XOR profila:

- Svi unosi u tabeli su nule ili pozitivni čak i cijeli brojevi
- Red za $\delta = 0$ ima samo jedan nulti unos koji je jednak 2^n (n je broj ulaznih bitova S-bloka)
- Suma stavki u svakom redu je jednaka 2^n
- Ulazna razlika δ može izazvati razliku u izlazu Δ sa vjerovatnoćom $p = \frac{\alpha}{2^n}$
- Ako je unos (δ, Δ) nula, onda razlika ulaza δ ne može izazvati razliku Δ na izlazu

Šta možemo reći o vrijednosti unosa?

$$(s_1 \oplus k) \oplus (s_2 \oplus k) = s_1 \oplus s_2$$

Šta možemo reći o ključu?

$$k \in \{s_{i_1} \oplus s_1 \wedge s_{i_2} \oplus s_1 \wedge, ..., \wedge s_{i_j} \oplus s_1\}$$

Primjer 3.3. *Neka ulaz* $(s_1, s_2) = (21_x, 38_x)$ *ima razliku na izlazu* $\Delta = 1_x$.

$$\delta = 100001 \oplus 111000 = 19$$

Skup:

$$S_{1_x}^{19_x} = \{(2_x, 1B_x; 4_x, 5_x), (1B_x, 2_x; 5_x, 4_x),$$
$$(22_x, 3B_x; 1_x, 0_x), (2C_x, 35_x, 2_x, 3_x), (35_x, 2C_x, 3_x, 2_x), (3B_x, 22_x, 0_x, 1_x)\}.$$

Ulaz je:

$$\gamma = \{2_x, 1B_x, 22_x, 3B_x, 2C_x, 35_x\}$$

Primjenjeni ključ mora biti u skupu:

$$k_1 = \gamma \oplus s_1 = \gamma \oplus s_2$$

što je:

$$k_1 = \{23_x, 3A_x, 3_x, 1A_x, D_x, 14_x\}$$

Neka je drugi ulaz $(s_1, s_2) = (14_x, 23_x)$*,* $\delta = 37_x$ *i* $\Delta = 2_x$*. Tada je*

$$S_{14_x}^{23_x} = \{(E_x, 38_x; 8_x, A_x), (F_x, 38_x; 1_x, 3_x), (11_x, 26_x; A_x, 8_x),$$
$$(12_x, 25_x, A_x, 8_x), (18_x, 2F_x, 5_x, 7_x), (19_x, 2E_x, 9_x, B_x), (25_x, 12_x, 8_x, A_x),$$
$$(26_x, 11_x, 8_x, A_x), (2E_x, 19_x, B_x, 9_x), (2F_x, 18_x, 7_x, 5_x), (38_x, F_x, 3_x, 1_x), (39_x, E_x, A_x, 8_x)\}.$$

Skup ulaza je:

$$\gamma = \{E_x, 39_x, F_x, 38_x, 11_x, 26_x, 12_x, 25_x, 18_x, 2F_x, 19_x, 2E_x\}$$

Ključ skupa je:

$$k_2 = \{1A_x, 2D_x, 2C_x, 1B_x, 32_x, 5_x, 31_x, 6_x, 3B_x, C_x, 3A_x, D_x\}$$

Uzmimo još jedan ulaz $(s_1, s_2) = (14_x, 1C_x)$*,* $\Delta = 9$*, tada je*

$$\gamma = \{6_x, E_x, 20_x, 28_x, 25_x, 2D_x\} \ ,$$

$$k_3 = \{12_x, 1A_x, 34_x, 3C_x, 31_x, 39_x\}$$

Ključ se mora sadržati u sva tri skupa pa dobijamo da je ključ:

$$k_1 \cap k_2 \cap k_3 = \{1A_x\}$$

XOR profil S-bloka sa tajnim ključem sa ulazom je identičan XOR profilu S-bloka bez ključa. Svako ulazno posmatranje (s_1, s_2) i odgovarajuca razlika u izlazu Δ omogućavaju kriptoanalizatoru da pronađe set K ključnih kandidata. Analiza razlika za jedan S-blok dozvoljava korisniku da preuzme ključ koji je XOR na ulazu S-bloka.

3.3.2 Linearna kriptoanaliza

Linearna kriptoanaliza uključuje postupke pronalaženja približnih vrijednosti šifri, a sastoji se od dva koraka. Prvi predstavlja izgradnju linearnih jednačina povezanih sa nekriptovanim tekstom, šifrovanim tekstom i ključem čije je odstupanje "visoko"(tj. vjerovatnosti zadržavanja preko prostora svih mogućih vrijednosti promjenjivih je što bliža nuli ili jedinici). Drugi korak je upotreba linearnih jednačina u konjunkciji sa poznatim parovima šifrovanog i/ili nešifrovanog teksta kako bi se odredio ključ.

Za potrebe linearne kriptoanalize, linearna jednačina izražava jednakost dva izraza koji se sastoje od binarnih promjenjivih kombinovanih sa XOR operacijom. Na primjer, sledeća jednačina sadrži XOR sumu prvog i trećeg bita nešifrovanog teksta te prvog bita šifrovanog teksta što je upravo jednako drugom bitu ključa:

$$P_1 \oplus P_3 \oplus C_1 = K_2$$

U idealnim šiframa, svaka linearna jednačina povezana sa nešifrovanim tekstom, šifrovanim tekstom i ključem bi bila uspješna sa vjerovatnoćom od $\frac{1}{2}$. Budući da jednačine u realnim linearnim sistemima variraju sa vjerovatnoćama, često se primjenjuju linearne aproksimacije. Procedure za konstrukciju aproksimacija su različite za svaku šifru. U osnovom tipu blok šifre, mreže supstitucija i permutacija, analiza je skoncentrisana na S-tablice kao jedinom nelinearnom dijelu šifre. Za dovoljno

male S-tablice moguće je odrediti svaku moguću linearnu jednačinu povezanu sa ulazima i izlazima S-tablice i izračunati osnovne bitove i odabrati najbolje. Linearna aproksimacija S-tablica mora biti kombinovana sa drugim akcijama šifre, kao što su permutacije i miješanje ključa, kako bi se postigla linearna aproksimacija cijele šifre. Postoje mnoge tehnike za poboljšanje linearne aproksimacije (npr. „piling-up“ postupak).

Nakon otkrivanja linearne aproksimacije oblika:

$$P_{i_1} \oplus P_{i_2} \oplus ... \oplus C_{j_1} \oplus C_{j_2} \oplus ... = K_{k_1} \oplus K_{k_2} \oplus ...$$

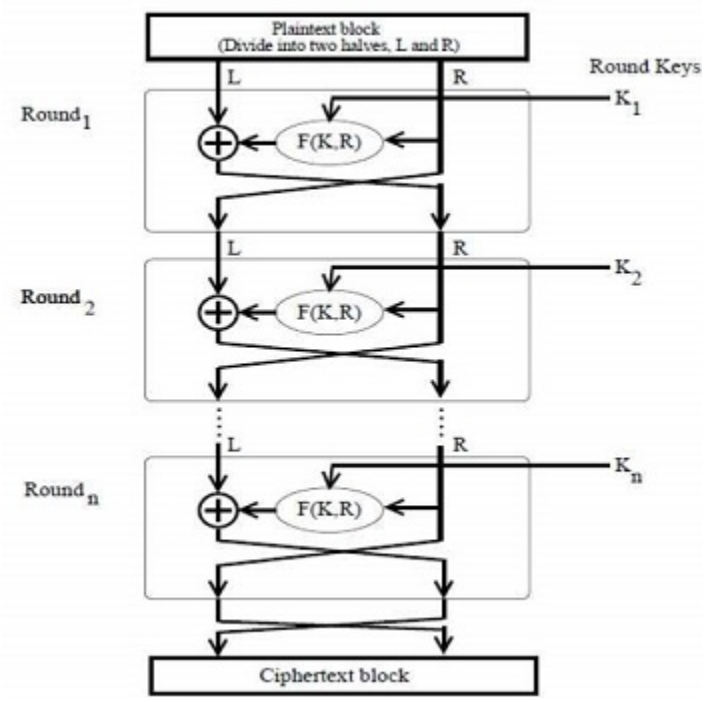
moguće je primijeniti razne algoritme koji korišćenjem poznatih šifrovanih/nešifrovanih parova teksta mogu otkriti vrijednosti bitova ključeva u aproksimacijama. Za svaki skup vrijednosti ključa sa desne strane potrebno je odrediti koliko je puta aproksimacija istinita preko svih poznatih šifriranih/nešifriranih parova (vrijednost T). Ključ koji ima najveću apsolutnu razliku vrijednosti T od polovine šifrovanih/nešifrovanih parova uzima se kao najvjerovatniji skup vrijednosti za bitove ključa. Postupak je moguće ponoviti s drugim linearnim aproksimacijama dok se broj nepoznatih bitova ključa ne smanji dovoljno da se može primijeniti ”brute force” napad.

3.3.3 Dejvisov napad

Najprije ćemo reći nešto o Fistel šifri. ***Fistel šifra*** nije specifična šema blok šifre. To je model dizajna iz kojeg dobijaju mnoge različite blok šifre. DES je samo jedan primer Fistel šifre.

Kriptografski sistem baziran na strukturi Fistel šifre koristi isti algoritam kako za enkripciju tako i za dekripciju.

Fistel struktura prikazana je na sledećoj ilustraciji:



Slika 3.2: Fistel struktura

Proces enkripcije koristi Fistel strukturu koja se sastoji od više krugova obrađivanja otvorenog teksta, svaki krug koji se sastoji od koraka "supstitucije", praćen korakom permutacije. Ulazni blok svakog kruga podijeljen je na dvije polovine koje se mogu označiti kao L (lijeva polovina) i R (desna polovina). U svakom krugu, desna polovina bloka, R, ostaje nepromijenjena, dok lijeva polovina, L, prolazi kroz operaciju koja zavisi od R i ključa za šifrovanje. Prvo, primenjujemo funkciju šifrovanja f koja uzima dva ulaza - ključ K i R. Rezultat funkcije je $f(R, K)$. Zatim, XOR izlaz matematičke funkcije sa L.

U stvarnoj implementaciji Fistelovog algoritma šifriranja, kao što je DES, umesto korišćenja cijelog ključa za šifrovanje tokom svakog kruga, kružni zavisni ključ (potključ) se izvodi iz ključa za šifrovanje. To znači da svaka runda koristi drugi ključ, iako su svi ovi potključevi povezani sa originalnim ključem. Postupak permutacije na

kraju svakog kruga zamenjuje modifikovano L i nemodifikovano R. Stoga, L za sledeći krug biće R trenutnog kruga, dok će R za sledeću rundu biti izlaz L trenutnog kruga. Broj krugova je specificiran dizajnom algoritma. Kada se završi poslednji krug, dva pod-bloka, R i L se spajaju da bi se formirala blok šifra. Teži dio dizajniranja Fistel šifre je izbor funkcije *f*.

Proces dekripcije u Fistelovoj šifri je gotovo sličan. Umjesto da se započne sa blokom otvorenog teksta, blok šifrovanja se unosi u početak strukture Fistela, a potom proces nakon toga je potpuno isti kao što je opisano na datoj ilustraciji. Rečeno je da je proces sličan i ne baš isti. U slučaju dešifrovanja, jedina razlika je u tome što su *subkeys* koje se koriste u šifrovanju koriste u obrnutom redosledu. Konačna zamjena L i R u poslednjem koraku Fistel šifre je od suštinskog značaja. Ako se one ne zamjene onda rezultat šifrovanja ne može biti dešifrovan koristeći isti algoritam.

Dejvisov napad je osmišljen za DES šifru ali se može primijeniti i na druge Fistel šifre (Fistel šifre su Blowfish, Twofish, 3DES, RC5, FEAL, Lucifer, ...). Pripada grupi napada poznat otvoren tekst, i bazira se na neuniformnoj distribuciji izlaza parova susjednih S-kutija. Napadač vrši sakupljanje velikog broja parova otvoreni/šifrovani tekst i računa empirijsku distribuciju pojedinih karakteristika. Biti ključa mogu biti izvedeni ako je dat veliki broj otvorenih tekstova, a preostale bite ključa nalazimo brute-force napadom. Postoji korelacija između broja zahtijevanih otvorenih tekstova, broja bita ključa koje možemo pronaći napadom i stope uspjeha i to tako da se napadom može pronaći 24 bita ključa (od ukupno 56 bita) poznavanjem 2^{57} otvorenih tekstova sa stopom uspjeha od 53%.

3.3.4 Related key napad

Ideja koja stoji iza ovog napada je da napadač poznaje, ili bira, relaciju koja postoji između više ključeva, i ima pristup šifri tj. može vršiti šifrovanje otvorenog

teksta datim ključevima. Cilj napada je nalaženje vrijednosti datih ključeva. Ukoliko je veza između grupe ključeva poznata, ali ne može biti promijenjena napad se zove poznat related key, ako napadač može sam odabrati grupu ključeva međusobno povezanih na način na koji to on želi onda se napad zove odabran related key. Kao što se može zaključiti iz prethodnog, pretpostavlja se da napadač ima velike mogućnosti pa je samim tim napad nerealističan u praksi. Ipak se ovaj tip napada može koristiti za pokazivanje slabosti algoritma za generisanje ključeva kod nekih šifri. Više šifri su pokazane slabima korišćenjem ovog napada a neki od njih su IDEA, GOST, SAFER, CAST, DES-X, RC2 i TEA. Related key se tipično smatra moćnim ali strogo teoretskim napadom. Međutim, određene implementacije šifri su podložne related key kriptoanalizi. Protokoli za sigurnu komunikaciju ponekad koriste K za šifrovanje a $\bar{\text{K}}$ za dešifrovanje. Bar jedan program za šifrovanje poruka niz uzastopnih poruka šifruje ključevima $K, K+1, K+2, \ldots$ (Ove implementacije su zaštićene autorskim pravima, pa nema referenci na raspolaganju) Takve implementacije ostavljaju šifru ranjivu na related key napad. Najočigledniji metod kojim napadač sprovodi related key je nepraktičan, napadač mora imati mogućnost da mijenja ključ na neki predefinisani način, a da pritom ne zna njegovu vrijednost [7].

3.3.5 Brute force napad

Brute force napad podrazumijeva da napadač pokušava sve moguće ključeve na šifrovanom tekstu dok ne nađe pravi [5]. Najčešće se primjenjuje kod napada šifrovani tekst. Za konačnu dužinu ključa i dovoljno vremena na raspolaganju Brute force napad je uvjek uspješan. Jedan od izazova sa kojima se susreće ova metoda jeste u slučaju napada šifrovani tekst, kada se može dobiti više smislenih otvorenih tekstova i treba zaključiti koji je pravi (Ako se otvoreni tekst dužine 15b koji glasi "This is

secure"šifruje jednokratnim ključem (iste dužine kao i otvoreni tekst), onda se primjenom Brute force napada može uspješno otkriti originalna poruka, ali i druge poruke kao što je "This is purple".). Računanje kompleksnosti ovog napada je jednostavno, ako je ključ dužine 8 bita postoji ukupno 2^8=256 ključeva. Dakle, u najgorem slučaju pravi ključ se pronalazi nakon 256 pokušaja, ali je vjerovatnoća pronalaska $\frac{1}{2}$ nakon pola pokušaja (128). Ako imamo ključ dužine 56 bita (Single DES koristi ključ dužine 56 bita), imamo 256 mogućih ključeva, računaru koji bi mogao u jednoj sekundi da isproba milion ključeva treba ukupno 2285 godina da obavi pretragu.

Pametni Brute force napadi ne pokušavaju sve moguće ključeve po numeričkom redosledu već prvo pokušavaju najočiglednije ključeve. Ovaj metod se naziva *dictionary* napad, jer napadač koristi rječnik najčešćih ključeva [6].

3.3.6 Meet-in-the-middle napad

Meet-in-the-middle napad su osmislili Difi i Helman, i spada u grupu napada "poznat otvorenom tekstu". To je napad na blok šifre kod kojih se šifrovanje vrši dva puta sa dva različita ključa, s ciljem povećanja sigurnosti šifre. Neka blok šifra koristi ključ dužine k bita. Kod dvostrukog šifrovanja otvoreni tekst P se prvo šifruje ključem K_l i rezultujući šifrovani tekst se zatim šifruje drugim ključem K_r.

$$C = E_{K_r}(E_{K_l}(P_1))$$

Napadač mora da poznaje bar dva para otvoreni/šifrovani tekst. Ako bismo pokušali bruteforce napad na ovako šifrovan tekst trebalo bi pretražiti sve moguće kombinacije oba ključa tj. efektivna dužina ključa bi bila 2k, pa bi pretraga ključeva zahtijevala $2^k \times 2^k = 2^{2k}$ šifrovanja (ili dešifrovanja). Korišćenjem meet-in-the-middle napada broj koraka se drastično smanjuje. Napad se izvodi na sljedeći način:

Za dati otvoreni tekst P_1 kreira se lukap tabela za sve parove $K_{l,1}$, $M_{l,1}$ gdje je

$$E_{K_{l,i}}(P_1) = M_{l,1},$$

a indeks i uzima vrijednosti $\{1,2,\ldots,\ 2^k\}$. $M_{l,1}$ su šifrovani tekstovi dobijeni nakon prvog šifrovanja. Lukap tabela koju napadač generiše treba da bude organizovana po vrijednosti $M_{l,1}$. Broj unosa u tabelu je 2^k a svaki unos je dužine $n+k$ bita, gdje je n dužina šifrovanog teksta. Sledeći korak je dešifrovanje šifrovanog teksta C_1, dobijenog kao rezultat dvostrukog šifrovanja. Biramo prvi mogući ključ $K_{r,1}$ (npr. sve nule), dešifrujemo tim ključem C_1. Dobijeni rezultat dešifrovanja, $M_{r,1}$ poredimo sa lukap tabelom i gledamo da li postoji takvo $M_{l,i}$, da važi

$$M_{r,1} = M_{l,i}$$

Ukoliko nismo dobili podudaranje prelazimo na sledeći ključ i ponavljamo proces sve dok ne dobijemo $M_{r,i} = M_{l,j}$. Tako dobijamo dva ključa: vrijednost $M_{l,j}$ je povezana sa ključem $K_{l,j}$ a $M_{r,i}$ sa ključem $K_{r,i}$. To znači da smo pronašli par ključeva ($K_{l,j}$,$K_{r,i}$) kojima se vrši dvostruko šifrovanje. Sada se vrši provjera nađenih ključeva tako što ih koristimo za šifrovanje nekoliko drugih parova otvoreni/šifrovani tekst. Ukoliko šifrovanjem otvorenih tekstova ključevima ($K_{l,j}$,$K_{r,i}$) dobijemo tačne šifrovane tekstove znači da smo pronašli prave ključeve, ako ne nastavljamo sa pretragom tako što krenemo od $M_{r,i+1}$ i ponavljamo proces. U prvoj fazi napada treba izvršiti 2^k šifrovanja i sačuvati ih u 2^k lokacija u memoriji. U drugoj fazi napada treba izvršiti maksimalno 2^k dešifrovanja. Pa je ukupan broj šifrovanja i dešifrovanja $2 \times 2^k = 2^{k+1}$, što je uporedivo sa brojem šifrovanja koje treba izvršiti u slučaju brute-force napada na jednostruko šifrovanje (2^k). Iz toga proizilazi da dvostruko šifrovanje značajno ne povećava sigurnost šifre, pa se umjesto dvostrukog koristi trostruko šifrovanje (3DES).

Primjer uspješnog meet-in-the-middle napada je napad na Double DES (2DES).

3.3.7 Standardni ASCII napad

Standardni ASCII napad pripada grupi napada "samošifrovani tekst", i primjenjuje se na šifre kod kojih se otvoreni tekst predstavlja standardnim ASCII kodom, u ovom primjeru je to Single DES (64 bita otvoreni tekst, 64 bita šifrovani tekst, 56 bita ključ). Kada se otvoreni tekst predstavlja standardnim ASCII kodom znamo da je prvi bit svakog bajta jednak 0, i tu informaciju koristimo u ovom napadu. Prvi šifrovani tekst C_1 se dekriptuje sa 2^{56} ključeva i čuvaju se oni ključevi koji za rezultat daju otvoreni tekst zapisan Standardnim ASCII kodom, takvih ključeva će biti 2^{48}, odnosno 2^{-8} od ukupnog broja. Drugi šifrovani tekst dekriptujemo sa izdvojenih 2^{48} ključeva i na isti način izdvajamo one ključeve kojima se dobija otvoreni tekst zapisan standardim ASCII kodom. Dakle, C_1 se dekriptuje sa 2^{56}, C_2 sa 2^{48}, C_3 sa 2^{40} ključeva, itd. Na kraju se C_7 dekriptuje sa 2^8 ključeva i od rezultujućih 2^8 otvorenih tekstova najvjerovatnije će samo jedan biti predstavljen Standardnim ASCII kodom što znači da smo uspješno otkrili ključ. Ovaj napad zahtijeva $2^{56}+2^{48}+...+2^8$ koraka (dekripcija), što je približno jednako 2^{56}, tj. duplo više koraka nego da smo na istu šifru primijenili Brute force napad.

Glava 4

RSA - napad na savremenu šifru

RSA algoritam razvili su Rivest, Šamir i Adelman 1977. godine. Sigurnost RSA algoritma zavisi od sposobnosti hakera da razloži brojeve na činioce. Novi, brži i bolji metod za faktorizaciju brojeva se konstantno razvijaju. Očigledno je da što je veći broj to je teže razložiti ga, a time što je broj veći, time je i bezbjednost samog algoritma bolja. Kako se teorija i računari poboljšavaju, veći i veći ključevi moraće da se upotrebljavaju. Prednost u korišćenju izuzetno velikih ključeva je računarski trošak koji je uključen u šifrovanju/dešifrovanju. Ovo će samo postati problem ako se pojavi nova tehnika za faktorizaciju tj. razlaganje brojeva koja će zahtevati korišćenje ključeva takvih veličina da će se potrebna dužina ključa povećavati mnogo brže od povećanja prosječne brzine računara koji koriste RSA algoritam. Buduća sigurnost RSA će se oslanjati isključivo na napredak u tehnikama faktorizacije tj. razlaganja brojeva.

4.1 Matematička osnova RSA algoritma

Matematička osnova RSA algoritma je delimično zasnovana na sledećim teoremama.

Teorema 4.1. *Neka su p i q prosti brojevi i neka je*

$$g = (p-1, q-1)$$

Tada je

$$a^{\frac{(p-1,q-1)}{g}} \equiv 1 \pmod{pq}$$

za svako a koje zadovoljava $(a, pq) = 1$

RSA algoritam počiva na težini problema tipa

$$x^e \equiv c \pmod{N}$$

gdje su e, c, N poznate, a x nepoznata veličina.

Teorema 4.2. *Neka je p prost broj, a $e \geq 1$ cio broj za koji važi $(e, p-1) = 1$. Tada kongruencija*

$$x^e \equiv c \pmod{N}$$

ima jedinstveno rješenje

$$x \equiv c^d \pmod{N}$$

gdje je $de \equiv 1 \pmod{(p-1)}$.

Sledeća teorema predstavlja srce RSA algoritma, i u direktnoj je vezi sa prethodnom teoremom.

Teorema 4.3. *Neka su p i q prosti brojevi i neka je $e > 1$ cio broj za koji važi $(e, (p-1)(q-1)) = 1$. Tada kongruencija*

$$x^e \equiv c \pmod{N}$$

ima jedinstveno rješenje

$$x \equiv c^d \pmod{N}$$

gdje je $de \equiv 1 \pmod{(p-1)(q-1)}$.

Neki od nedostataka RSA algoritma su:

- Koriste se prosti brojevi sa nekoliko stotina cifara
- Koriste se posebni algoritmi za množenje
- Sporiji su u odnosu na simetrične algoritme
- Algoritmi za napad tj. faktorizaciju postaju sve bolji, pa RSA nema nikakvih šansi
- 512 bitni RSA je nedovoljan za bezbedno šifrovanje
- 1024 bitni RSA prema NIST-u (National Institute of Standards and Technology) je bio siguran do 2010 godine, a takođe prema njihovoj pretpostavci 2048-bitni RSA bi trebao ostati siguran (bezbjedan) do 2030 godine.

Prvi napadi na RSA bazirali su se na ranim protokolarnim greškama kriptosistema [9].

4.2 Rani napadi na RSA

4.2.1 Korišćenje istog modula

Kada više korisnika koristi isti modul svaka se poruka koju šifrujemo sa dva različita javna ključa može lako dešifrovati. Neka su (e_1, N) i (e_2, N) dva različita javna ključa koji imaju relativno proste javne eksponente. Tada se, koristeći Euklidov algoritam, lako mogu izračunati brojevi a_1 i a_2 takvi da $a_1e_1 + a_2e_2 = 1$. Za svaku

poruku m dobijamo šifrate $c_1 = m^{e_1} \pmod N$ i $c_2 = m^{e_2} \pmod N$. Sada otvoreni tekst poruke možemo lako izračunati množenjem $c^{a_1}c^{a_2}$ jer važi:

$$c^{a_1}c^{a_2} = m^{a_1e_1}m^{a_2e_2} = m^{a_1e_1+a_2e_2} = m$$

Pokazano je i da se poznavajući samo jedan par javnog i odgovarajućeg tajnog ključa može za svaki drugi javni ključ sa istim modulom efikasno izračunati odgovarajući tajni ključ.

Teorema 4.4. *Neka je (e, N) RSA javni ključ sa odgovarajućim tajnim ključem d i neka je e_1 neki drugi javni ključ takav da je $e1 \neq e$. Za date e, d, N i e_1, tajni ključ koji odgovara ključu e_1 možemo izračunati kao:*

$$d_1 = e_1^{-1} \pmod{\frac{ed-1}{\mathbf{nzd}(e_1, ed-1)}}$$

Dokaz: Jednačinu ključa možemo napisati kao $ed-1 = k\lambda(N)$, gdje je $k \in N$. Znamo da vazi $\mathbf{nzd}(e_1, \lambda(N)) = 1$, pa $\mathbf{nzd}(e_1, \lambda(N)) = k'$, za neki k' takav da $k' \mid k$. Neka je $\check{k} = k \mid k'$. Imamo:

$$\frac{ed-1}{\mathbf{nzd}(e_1, ed-1)} = \frac{k\lambda(N)}{k'} = \check{k}\lambda(N)$$

pa tajni eksponent $d_1 = e_1^{-1} \pmod{\frac{ed-1}{\mathbf{nzd}(e_1, ed-1)}}$ zadovoljava:

$$e_1d_1 = 1 + k_1(\check{k}\lambda(N)),$$

za neki prirodni broj k_1. Slijedi da $e_1d_1 \equiv 1(\pmod \lambda)(N))m$ pa je d_1 ispravan tajni eksponent za javni ključ (e_1, N).

Dakle zaključujemo da bi svaki RSA modul trebao biti poznat samo jednom korisniku.

4.2.2 Napad korišćenjem bliskih prostih brojeva

Kada koristimo bliske proste brojeve (faktore), tačnije, ako je $N = pq$ i $|p - q| < N^{1/4}$ tada N možemo efikasno faktorisati. Neka je $N = pq$, definisimo $x = \frac{1}{2}(p+q)$ i $y = \frac{1}{2}(p-q)$.

Važi $N = x^2 - y^2$ odnosno $y^2 = x^2 - N$, sada inicijalizujemo $x = \lfloor\sqrt{N}\rfloor + 1$, kvadriramo x, oduzmemo od njega N i provjeravamo jesmo li dobili kvadrat cijelog broja, iterativno povećavamo x za jedan i ponovno vršimo provjeru dok ne dođemo do kvadrata.

Primjer 4.1. *Neka je $N = 20648509087$. Dakle $x = \lfloor\sqrt{N}\rfloor + 1 = 143696$.*

$143696^2 - 20648509087 = 31329 = 177^2$

Kvadrat smo dobili iz prvog pokušaja. Za $y = 177$ vrijedi $x^2 = y^2 + N = 31329 + 20648509087 = 20648540416 = 143696^2$, pa je $x = 143696$.

Rješavamo sistem jednačina:

$143696 = \frac{1}{2}(p+q)$

$177 = \frac{1}{2}(p-q)$

Dobijamo rješenja $p = 126481$ i $q = 126839$.

4.3 Brute Force napad

Ovaj napad podrazumijeva da napadač pokušava sve moguće ključeve na šifrovanom tekstu dok ne nađe onaj pravi koji mu je potreban (d – tajni ključ). Brute force se temelji na isprobavanju svih mogućih kombinacija. Ovakvi napadi ne mogu biti veoma efikasni iz razloga što zahtevaju veliku količinu resursa za isprobavanje velikog broja različitih kombinacija, ali ukoliko je poznata dužina ključa i imamo dovoljno vremena na raspolaganju ova vrsta napada će biti uspešna. Npr. kompleksnost ovog napada se

računa vrlo jednostavno. Ako je ključ dužine 8 bita postoji ukupno $2^8 = 256$ ključeva.

Najefikasniji napad protiv RSA je zapravo napad faktorizacijom broja n, ali takođe postoje i napadi koji se zasnivaju na izračunavanju $(p-1)(q-1)$. Međutim, za ovakve napade treba mnogo vremena.

Neki od algoritama za faktorizaciju su sledeći:

- Deljenje – najneefikasnija metoda
- Kvadratni sieve algorithm (algoritam kvadratnog sita) – najbrži algoritam za brojeve koji imaju manje od 100 znakova
- SNFS (Specific Number Field Sieve)
- GNFS (General Number Field Sieve)

4.3.1 Eratostenovo sito

Eratostenovo sito predstavlja postupak nalaženja prostih brojeva koji su manji od nekog zadatog broja n. Ovo sito je konstruisao starogrčki matematičar Eratosten. Kako algoritam funkcioniše?

Primjer 4.2. *Ovaj algoritam nalazi sve proste brojeve do broja n. Uzmimo da je n = 120.*

Korak 1: Napišemo sve brojeve od 2 do 120.

Korak 2: Počevši od broja 2 koji je prvi na spisku, precrtaćemo sve brojeve koji su deljivi sa 2 i broj 2 upisujemo kao prost broj.

Korak 3: Nakon toga, ponavlja se postupak, krećući od početka, sa sledećim neprecrtanim brojem m. Precrtaćemo sve brojeve deljive sa m, a njega samog obilježavamo kao prost broj. Prosti brojevi u ovom slučaju su: 2, 3, 5, 7, 11, 13, 17, 19 23, 29, 31 37, 41, 43, 47, 53, 59 61, 67, 71, 73, 79, 83, 89, 97 , 101, 103, 107, 109, 113.

Algoritmi kvadratnog sita spadaju u široku grupu algoritama koja traže dva broja x i y. Šema za faktorizaciju n je zapravo rješavanje jednačine $x^2 = y^2 \pmod{n}$. *Ukoliko pronađemo jedan ovakav par za koji važi* $x^2 = y^2 \pmod{n}$, *tada je* $\mathbf{nzd}(x - y, n)$ *netrivijalni činilac od n. Trivijalna rešenja su ona gdje je* $y = z \pmod{n}$ *ili* $y = -z \pmod{n}$.

Trivijalna rešenja su ona rešenja gdje je homogeni sistem određen, a neodređeni homogeni sistem ima netrivijalna rešenja.

4.3.2 SNFS

SNFS je algoritam za faktorizaciju sa posebnim namjenama. SNFS je jako efikasan za cele brojeve u formi $r^e \pm s$ gdje su r i s prosti brojevi (na primer Mersenovi prosti brojevi). U principu SNFS je sličan algoritmu kvadratnog sita, samo što on radi nad algebarskim poljima brojeva koji su definisani pomoću r, e i b.

Napomena: Mersenovi prosti brojevi su oblika $2^n - 1$. Naziv su dobili po matematičaru Marinu Mersenu, koji je tvrdio da je broj $2^n - 1$ za $n = 2$, 3, 5, 7, 13, 17 , 19, 31, 67, 127, 257 prost, a za sve druge prirodne brojeve manje od 257 da je složen Međutim, kasnije provjere su pokazale da brojevi 67 i 257 nisu prosti, a prosti brojevi su 61, 89 i 107.

4.3.3 GNFS

GNFS je samo nastavak SNFS-a. Ovde se zahtevaju dva polinoma $g(x)$ i $f(x)$ malih stepena d i e, koji imaju celobrojne koeficijente koji su nesvodljivi preko razlomka, i koji se tumače kao mod n, imaju zajednički celobrojni korijen m. Optimalna strategija za odabir ovih polinoma nije poznata.

4.4 Napad korišćenjem malog enkripcijskog eksponenta

Teorema 4.5 (***Kineska teorema o ostacima***)**.** *Neka su m_1, m_2,..., m_r u parovima relativno prosti prirodni brojevi, i neka su a_1, a_2,..., a_r cijeli brojevi. Tada sistem kongruencija*

$$x \equiv a_1 \pmod{m_1},\ x \equiv a_2 \pmod{m_2},\ldots,\ x \equiv a_r \pmod{m_r}$$

ima rješenja. Ako je x_0 jedino rješenje, onda su sva rješenja sistema data sa $x \equiv x_0 \pmod{m_1 m_2 \ldots m_r}$

4.4.1 Vrlo mali enkripcijski eksponent za jednake poruke

Posmatrajmo sledeći napad za vrlo mali enkripcijski eksponent, uzmimo za primjer $e = 3$. Ako istu poruku m pošaljemo tri osobe koristeći $\{(e, N_1), (e, N_2), (e_N 3)\}$, gdje je $\mathbf{nzd}(N_i, N_j) = 1$, $\forall i,\ j \in \{1, 2, 3\}$, $i \neq j$ i $m < Ni$ tada lako dolazimo do originalne poruke rješavajući sistem kongruencija:

$$c_1 \equiv m^3 \pmod{N_1};\ c_2 \equiv m^3 \pmod{N_2};\ c_3 \equiv m^3 \pmod{N_3}$$

Koristimo Kinesku teoremu o ostacima i dolazimo do x sa svojstvom $x \equiv m^3 \pmod{N_1 N_2 N_3}$. Nego kako je $m^3 < N_1 N_2 N_3$, vazi $x = m^3$, pa računamo poruku tako da izvadimo treći korijen od x.

Primijetimo da kad bi m bio manji od $N^{1/3}$ dovoljno bi bilo izvaditi treći korijen od originalne poruke. Uopšteno, za $m < N^{1/e}$, za šifrat važi $c = m^e \pmod{N} = m^e$, pa je za dekripciju bilo dovoljno izvaditi e-ti korijen od šifrata. U praksi se takve male poruke izbjegavaju nadopunjavanjem originalne poruke nasumičnim bitovima kako bi postigli željenu, sigurnu veličinu poruke.

4.5 Napad korišćenjem malog dekripcijskog eksponenta

Definicija 4.1. *Neka su a_0,a_1,...,a_n realni brojevi. Izraz oblika:*

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cdots}}}$$

nazivamo konačni verižni razlomak. Ako je a_o cijeli, a a_i, $i = 0, ..., n$ prirodni brojevi, kažemo da je verižni razlomak jednostavan.

4.5.1 Vinerov napad

Vinerov napad je nazvan po kriptologu Mišaelu J. Vineru i predstavlja kriptografski napad na RSA algoritam. Napad koristi metodu verižnog razlomka da otkrije tajni ključ d. Vinerov napad, će otkriti d, kada je d veličine do jedne četvrtine n i e je manje od n. Ovo se retko javlja ako su e i d izabrani nasumično.

Teorema 4.6. *Neka je $n = pq$, $n < q < 2p$, $e < \varphi(n)$ i neka je $d < \frac{1}{3}n^{1/4}$. Tada postoji polinomijalni algoritam koji iz poznavanja n i e izračunava d.*

Dokaz: Iz $ed \equiv 1 \pmod{\varpi}(n)$ slijedi da postoji prirodan broj k takav da je $ed - k\varphi(n) = 1$. Odavde je:

$$\left|\frac{e}{\varphi(n)} - \frac{k}{d}\right| = \frac{1}{d\varphi(n)} \tag{4.5.1}$$

Dakle, $\frac{k}{d}$ je dobra aproksimacija od $\frac{e}{\varphi(n)}$. Međutim, $\varphi(n)$ je nepoznat. Stoga cemo $\varphi(n)$ aproksimirati sa n. Iz $\varphi(n) = n-p-q+1$ i $p+q-1 < 3\sqrt{n}$ slijedi $|n-\varphi(n)| <$

$3\sqrt{n}$. *Zamijenimo* $\varphi(n)$ *sa* n *u* (4.4.1), *i dobijamo:*

$$\Big|\frac{e}{n}-\frac{k}{d}\Big| = \Big|\frac{ed-k\varphi(n)-kn+k\varphi(n)}{nd}\Big| = \Big|\frac{1-k(n-\varphi(n))}{nd}\Big| \leq \frac{3k\sqrt{n}}{nd} = \frac{3k}{d\sqrt{n}}$$

Sada je $k\varphi(n) = ed-1 < ed$ *pa iz* $e < \varphi(n)$ *(standardna pretpostavka u RSA), slijedi* $k < d < \frac{1}{3}n^{1/4}$, *pa dobijamo:*

$$\Big|\frac{e}{\varphi(n)}-\frac{k}{d}\Big| \leq \frac{1}{d\sqrt[4]{x}} < \frac{1}{2d^2} \tag{4.5.2}$$

Nadalje će nam u dokazu trebati sledeći rezultat.

Teorema 4.7 (***Ledžender***)**.** *Neka su* p, q *cijeli brojevi takvi da je* $pq \geq 1$ *i*

$$\Big|\,\alpha-\frac{p}{q}\,\Big| \;< \frac{1}{2q^2} \tag{4.5.3}$$

Tada je $\frac{p}{q}$ *neka konvergenta od* α.

Iz Ledženderove teoreme slijedi da relacija (4.5.2) povlači da je $\frac{k}{d}$ neka konvergenta razvoja u verižni razlomak od $\frac{e}{n}$.

Iz rekurzije za konvergente $\frac{p_k}{q_k}$ verižnog razlomka, slijedi da je $q_k \geq F_k$, gdje je F_k k-ti Fibonačijev broj, što znači da konvergente rastu eksponencijalno. U našem slučaju slijedi da imamo $O(\log n)$ konvergenti od $\frac{e}{n}$. Jedna od njih je $\frac{k}{d}$. Dakle, izračunamo sve konvergente od $\frac{e}{n}$ i testiramo koja od njih zadovoljava uslov $(x^e)^d \equiv x \pmod{n}$ za slučajno odabran broj x. To daje polinomijalni algoritam za otkrivanje tajnog ključa d.

Drugi način za testiranje tačnosti pretpostavke da je neka konkretna konvergenta jednaka $\frac{k}{d}$, jeste da se, uz tu pretpostavku, izračuna $\varphi(n) = (p-1)(q-1) = \frac{ed-1}{k}$ Tada se moze izračunati $\frac{p+q}{2}$ iz identiteta:

$$\frac{pq-(p-1)(q-1)+1}{2}=\frac{p+q}{2}$$

pa $\frac{q-p}{2}$ iz identiteta:

$$\left(\frac{p+q}{2}\right)^2-pq=\left(\frac{q-p}{2}\right)^2$$

Ako se na ovaj način dobije da su brojevi $\frac{p+q}{2}$ i $\frac{q-p}{2}$ cijeli, onda zaključujemo da je posmatrana konvergenta stvarno jednaka $\frac{k}{d}$. Tada iz $\frac{p+q}{2}$ i $\frac{q-p}{2}$ možemo lako dobiti faktorizaciju modula $n=pq$.

Primjer 4.3 (***Numerički primjeri Vinerovog napada***)**.** *Neka je u RSA kriptosistemu zadan modul* $n=2989234739$ *i javni eksponent* $e=2501103889$*, i neka je poznato da tajni eksponent* d *zadovoljava* $d<\frac{1}{3}n^{1/4}<80$*. Razvoj od* $\frac{e}{n}$ *u verižni razlomak je:*

[0, 1, 5, 8, 13, 3, 2505, 1, 7, 3, 1, 5, 3]

Zatim računamo pripadne konvergente:

$$0,\ 1,\ \frac{5}{6},\ \frac{41}{49},\ \frac{538}{643},\ldots$$

Sada ćemo provjeravati za koje konvergente $\frac{k_i}{d_i}$*,* $i\geq 2$*, vrijedi da je* $\varphi(n)=(p-1)(q-1)=\frac{ed-1}{k}$ *cijeli broj i da li se* n *može faktorisati iz* $\varphi(n)$*.*

Za $k_3=5$ *i* $d_3=6$ *slijedi:*

$$\varphi(n)=\frac{ed_3-1}{k_3}=\frac{2501103889*6-1}{5}=\frac{15006623333}{5}=3001324666,6.$$

Za $k_4=41$ *i* $d_4=49$ *slijedi:*

$$\varphi(n)=\frac{ed_4-1}{k_4}=\frac{2501103889*49-1}{41}=\frac{122554090560}{41}=2989124160.$$

Iz $\varphi(n)=2989124160$ *dobijamo i odgovarajuce faktore od* n *(*$p=47059$ *i* $q=63521$*), što ce značiti da je traženi tajni kljuc* $d=49$*.*

Primjer 4.4. *Neka je* $n = 7978886869909$*,* $e = 4603830998027$*, i pretpostavimo da je* $d < 10000000$*. Razvoj u verižni razlomak broja* $\frac{e}{n}$ *je*

[0, 1, 1, 2, 1, 2, 1, 18, 10, 1, 3, 3, 1, 6, 57, 2, 1, 2, 14, 7, 1, 2, 1, 4, 6, 2].

a prvih nekoliko konvergenti je

$$0,\ 1,\ \frac{1}{2},\ \frac{3}{5},\ \frac{4}{7},\ \frac{11}{9},\ \frac{15}{26},\ \frac{281}{487},\ \frac{2825}{4896},\ldots$$

Tražimo dvije susjedne neparne konvergente izmedu kojih se nalazi $\frac{e}{n} + \frac{2.122e}{n\sqrt{n}}$*. Dobijamo:*

$$\frac{281}{487} < \frac{e}{n} + \frac{2.122e}{n\sqrt{n}} < \frac{11}{19}$$

Tajni eksponent tražimo među brojevima nekog od oblika $26r + 19s$ *ili* $487s - 26t$ *ili* $4896r' + 487s'$*. Primjenjujući kriterijum za testiranje kandidata za razlomak* $\frac{k}{d}$*, nalazimo da je* $d = 5936963$*, što se dobija za* $s = 12195$*,* $t = 77$*.*

Glava 5

Implementacija Vinerovog napada

```
from decimal import *
import math
getcontext().prec = 40

def euklidov_algoritam(b,a):
    r0 = a
    r1 = b
    r2 = 1
    niz = []
    while(r2 > 0):
        q = r0 / r1
        niz.append(q)
        r2 = r0 - q * r1
        r0 = r1
        r1 = r2
    return niz

def main():
    n = int(raw_input())
    b = int(raw_input())
    q = euklidov_algoritam(n,b)
    c = [1,q[0]]
    d = [0,1]
    n1 = 0
    p1 = 0
    p2 = 0
    for j in range(1,len(q)):
        if c[j] != 0:
            n1 = (d[j] * b - 1) / c[j]
            if n1 % 1 == 0:
                prod = n - n1 + 1
                descrim = Decimal(prod * prod - 4 * n)
                if descrim > -0:
                    sqrtdesc = descrim.sqrt()
                    if sqrtdesc._isinteger():
                        p1 = (prod + sqrtdesc)
                        p2 = (prod - sqrtdesc)
                        if p1 % 2 == 0 and p2 % 2 == 0:
                            p1 = p1 / 2
                            p2 = p2 / 2
                        if p1 > 0 and p2 > 0:
                            if p1 < n and p2 < n:
                                print "p: ",p1," q: ",p2
                                return 1
        c.append(q[j] * c[j] + c[j-1])
        d.append(q[j] * d[j] + d[j-1])
    print "Wienerov napad je neuspješan"
    return 0
main()
```

Slika 5.1: Implementacija Vinerovog napada

Glava 6

Zaključak

U radu je data klasifikacija šifri kao i pregled napada na kriptografske šifre. Raspravljano o mehanizmu šifrovanja asimetrične šifre RSA, a posebno se obratila pažnja na bezbjednost ove šifre. Nekoliko zaključaka se može izvesti iz prethodnog rada. Bezbjednost šifre leži u tajnosti ključa kao i u njegovoj dužini i to tako da sigurnost šifre raste sa porastom broja bita ključa, ali samo ukoliko je najbolji poznati napad na datu šifru brute-force. Može se reći da je velika dužina ključa potreban, ali ne i dovoljan uslov bezbjednosti jedne šifre. Osobine koje jedna sigurna šifra mora imati su difuzija i konfuzija. Konfuzijom se postiže da jedan bit šifrovanog teksta zavisi od više bita ključa, tako da je veza između ključa i šifrovanog teksta što kompleksnija. Difuzija garantuje da će promjena jednog bita otvorenog teksta uzrokovati promjenu više bita šifrovanog teksta, tako da statističke karakteristike otvorenog teksta nijesu uočljive u šifrovanom.

Onog trenutka kada nalaženje prostih činioca velikih cijelih brojeva postane računarski izvodljivo, asimetrična šifra RSA neće biti sigurna.

Razvoj kriptografije je usko spregnut sa razvojem kriptoanalize, nauke koja ima suprotan cilj – razbijanje šifre. Upravo zbog te veze realno je za očekivati razvoj novih napada na postojeće šifre, ali i razvoj novih šifri.

Bibliografija

[1] Eric Conrad, Types of Cryptographic Attacks

[2] Bruce Schneier, A self-study course in block-cipher cryptanalysis

[3] Fauzan Mirza, Block ciphers and cryptanalysis

[4] James Price, Cryptanalysis and Brute Force Attacks

[5] http://www.cert.hr/sites/default/files/CCERT-PUBDOC-2009-08-275.pdf

[6] Bruce Schneier, Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C, Wiley Publishing, 1996.

[7] John Kelsey, Bruce Schneier, David Wagner, Key-Schedule Cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES

[8] Basic cryptanalysis, Field manual 34-40-2, Headquarters Department of the Army, 1990.

[9] M. Jason Hinek, Cryptanalysis of RSA and its variants, CRC press, 2009.