CARNet
HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

Kriptoanaliza

CCERT-PUBDOC-2009-09-275

+CERT.hr u suradnji s LSS

Sigurnosni problemi u računalnim programima i operativnim sustavima područje je na kojem CARNet CERT kontinuirano radi.

Rezultat toga rada je i ovaj dokument, koji je nastao suradnjom CARNet CERT–a i LS&S–a, a za koji se nadamo se da će Vam koristiti u poboljšanju sigurnosti Vašeg sustava.

CARNet CERT, www.cert.hr

Nacionalno središte za **sigurnost računalnih mreža** i sustava.

LS&S, www.LSS.hr

Laboratorij za sustave i signale pri Zavodu za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu.

Sadržaj

1. Uvod

Jedno od ključnih sredstava poslovanja, trgovanja i komunikacije u današnje vrijeme čine informacije koje se svakodnevno izmjenjuju, pohranjuju i obrađuju putem Interneta. Veliki postotak tih informacija predstavlja vrlo važne ili tajne podatke neke tvrtke ili organizacije. Kako bi se osigurala njihova sigurnosti razvijaju se razne metode kriptiranja podataka prije njihova slanja putem mreže. Sve metode koje proučavaju skrivanje informacija spadaju u skupinu postupaka kriptografije.

Za razliku od spomenutih metoda, cilj kriptoanalize je pronalaženje ranjivosti u kriptografskim shemama kako bi se otkrili tajni ključevi za dekriptiranje informacija. Tijekom razvoja kriptografskih tehnika, a time i njihovog napretka, razvijeni su razni napadi na kriptografske sustave. Jednostavniji napadi (npr. analiza učestalosti) odnose se na analiziranje šifriranih i/ili nešifriranih tekstova kako bi se uočila neka svojstva koja je moguće iskoristiti za dekriptiranje. Sofisticiraniji napadi (npr. diferencijalna kriptoanaliza) često uključuju provođenje određenih složenih matematičkih izračuna. Gotovo svaki kriptografski sustav moguće je probiti s dovoljno napora uporabom pretraživanja svih mogućnosti (tzv. *brute force* napad). Ipak, spomenuti napad često zahtjeva velike količine memorije i vremena koje u praksi nisu raspoložive napadačima.

U nastavku dokumenta dan je uvod u kriptoanalizu te povijesti njenog razvoja. Zatim su opisane metode kriptoanalize klasificirane u nekoliko osnovnih skupina. Nakon upoznavanja s osnovnim postupcima, dan je prikaz rada tih postupaka preko praktičnih primjera.

2. Kriptoanaliza

Kriptoanaliza je znanstvena disciplina koja proučava metode otkrivanja značenja kriptiranih informacija bez pristupa tajnim informacijama za dekriptiranje. Obično se takve metode zasnivaju na poznavanju rada sustava te pronalaženju tajnog ključa. Sama riječ nastala je od dvije grčke riječi: kryptós – skriven i analýein – odriješiti.

Izraz kriptoanaliza ponekad se odnosi na pokušaj zaobilaženja sigurnosti kriptografskih algoritama ili protokola, a ne samo kriptografske zaštite. Ipak, kriptoanaliza obično uključuje metode napada koje ne ciljaju na ranjivosti poput socijalnog inženjeringa ili krađe i zapisivanja lozinki unesenih preko tipkovnice.

Iako imaju isti cilj, metode i tehnike kriptoanalize drastično su se promijenile tijekom povijesti (zbog rasta složenosti kriptografije). Rezultat kriptonalize je također izmijenjen u smislu da više nije moguće imati gotovo neograničen uspjeh u probijanju kodova. Probijanje kriptografskog algoritma označava da je poznat postupak otkrivanja tajnih informacija (ključeva) koje se mogu iskoristiti za dešifriranje kriptiranog teksta. Sredinom 70-ih godina 20. stoljeća uvedena je nova klasa kriptografije, asimetrična kriptografija. Metode proboja takvih kriptografskih sustava su obično uključivale rješavanje složenih matematičkih problema, od kojih je najpoznatija faktorizacija cjelobrojnih brojeva.

2.1. Povijest kriptoanalize

Kriptoanaliza je evoluirala zajedno s kriptografijom koja je nastojala zamijeniti stare probijene postupke šifriranja. Nove tehnike nastale su kako bi omogućile proboj poboljšanih kriptografskih shema. U praksi, kriptografija i kriptoanaliza odnose se kao dvije strane istog novčića: kako bi se kreirala sigurna kriptografija, potrebno je dizajn prilagoditi mogućim kriptoanalizama.

2.1.1. Klasična kriptoanaliza

Iako je izraz kriptoanaliza relativno nov (uveo ga je William Friedman 1920. godine), prve metode probijanja kodova su puno starije. Prvo poznato zapisano objašnjenje kriptoanalize dao je u 9. stoljeću arapski matematičar Al-Kindi (u Europi poznat kao Alkindus) u djelu „A Manuscript on Deciphering Cryptographic Messages“ („Rukopis o dešifriranju kriptografskih poruka“). Početna stranica spomenutog djela prikazana je na Slika 1., a između ostalog rasprava uključuje opis metoda analize učestalosti (Ibrahim Al-Kadi, 1992- ref-3).

Slika 1 „A Manuscript on Deciphering Cryptographic Messages“
Izvor: Wikipedia.

Analiza učestalosti je najbolji alat za probijanje većine klasičnih šifri. Općenito, radi se o analizi učestalosti pojavljivanja određenih slova nekog jezika. Na primjer, u engleskom jeziku najčešće se pojavljuje slovo „E“. Analiza učestalosti oslanja se na činjenicu da su šifre obično ne skrivaju takvu statistiku. Na primjer, u običnom kriptiranom tekstu, slovu koje se najčešće pojavljuje pridjeljuje se značenje znaka „E“. Ovakva analiza je poprilično jednostavna ako je kriptirani tekst dovoljno dug da se može provesti analiza.

U Europi, tijekom 15. i 16. stoljeća, razvijena je ideja o abecednoj zamjeni znakova(eng. polyalphabetic substitution cipher), a jedan od pokretača bio je Blaise de Vigenère. Sljedeća tri stoljeća Vigenère-ova šifra, koja koristi ključ (proizvoljnu riječ) za odabir različitih kriptirajućih abeceda (svako slovo nekriptiranog teksta zamijeni se nekim drugim), smatrana je u potpunosti sigurnom. Međutim, Charles Babbage i kasnije Friedrich Kasiski uspjeli su probiti spomenutu šifru. Tijekom Prvog svjetskog rata, izumitelji u nekoliko država razvili su kružne mehanizme šifriranja (eng. rotor cipher machines) u nastojanju da minimiziraju ponavljanje koje je dovelo do proboja Vigenère-ova sustava.

Kako je šifriranje postajalo složenije, matematika je postajala sve važnija u kriptoanalizi. Ova je promjena postala sve očitija prije i tijekom Drugog svjetskog rata zbog napora za probijanjem šifriranja Axis saveza (http://en.wikipedia.org/wiki/Axis_Powers). Tijekom razvoja uređaja za kontrolu bomba i računala Colossus, prvog digitalnog računala kojim je upravljao program, prvi put je primijenjena automatizacija.

2.1.2. Moderna kriptoanaliza

Moderna kriptografija postala je neprobojnija za metode kriptoanalize. David Kahn je u svom djelu o kriptoanalizi (http://www.fas.org/irp/eprint/kahn.html) naveo mnoge mogućnosti zamjene tradicionalne kriptoanalize poput presretanja poruka. Navodi kako mnogi proizvođači nude kriptografske sustave koje nije moguće probiti poznatim metodama kriptoanalize. U takvim sustavima ključ nije moguće otkriti usporedbom nekriptiranog teksta s kriptiranim.

Asimetrična kriptografija oslanja se na uporabu dva ključa, jednog privatnog i drugog javnog. Probijanje takvih šifri zasniva se na rješavanju kompleksnih matematičkih problema. Na primjer, sigurnost sheme za razmjenu ključeva Diffe-Hellman ovisi o složenosti računanja diskretnih logaritama. Godine 1983. Don Coppersmith pronašao je brži način određivanja diskretnih algoritama (u određenoj grupi) što je dovelo do potrebe za uporabom većih grupa u kriptografiji. Sigurnost algoritma RSA ovisi o složenosti faktorizacije cjelobrojnih brojeva.

Godine 1980. bilo je moguće faktorizirati broj od 50 digitalnih znamenki na računalu sa 1012 osnovnih računalnih operacija. Do 1984. godine s istim utroškom računalnih resursa bilo je moguće faktorizirati broj s 75 digitalnih znamenki. Napredak u računarskoj tehnologiji također je značio brže obavljanje operacija na računalima. Na brzim, modernim računalima stručnjaci su uspjeli faktorizirati brojeve s 150 digitalnih znamenki pa se takva duljina ključa, od početka 21. stoljeća, ne smatra dovoljnom za sigurnost algoritma RSA.

2.2. Provođenje kriptoanalize

Kriptoanaliza se može provesti nagađanjem ključa (ovisno o tome koliko se informacija može otkriti) ili korištenjem informacija o sustavu koji se napada. Kao osnovna točka početka kriptoanalize obično se uzima pretpostavka kako je neprijatelju sustav poznat (Kerckhoffov princip). Ovo je razumna pretpostavka u praksi jer postoje brojni „tajni“ algoritmi koji su probijeni kroz povijest.

Osnovne vrste kriptoanalize uključuju:

- **Samo šifrirani tekst** (eng. Ciphertext-only) - napadač ima pristup samo skupini šifriranih tekstova.
- **Poznati nešifrirani tekst** (eng. Known-plaintext) – napadač ima skupinu šifriranih tekstova za koji poznaje odgovarajući nešifrirani tekst.
- **Izabrani (ne)šifrirani tekst** (eng. Chosen-plaintext/ciphertext) – napadač može otkriti (ne)šifrirani tekst koji odgovara skupini (ne)šifriranog teksta po njegovom vlastitom odabiru.

- **Prilagodljivi izabrani nešifrirani tekst** (eng. Adaptive chosen-plaintext) – poput prethodnog, osim što napadač može izabrati sljedeći nešifrirani tekst na temelju informacija koje je prikupio u prethodno opisanom načinu dešifriranja.
- **Napad odgovarajućim ključem** (eng. Related-key attack) – poput izabranog (ne)šifriranog teksta, osim što napadač može otkriti šifrirani tekst kriptiran s dva različita ključa. Pri tome, ključevi nisu poznati napadaču, ali poznat je odnos među njima (npr. kakva je razlika).

2.3. Rezultati i složenost kriptoanalize

Lars Knudsen napravio je podjelu rezultata kripoanalize dijela podataka prema količini i kvaliteti otkrivenih tajnih informacija na:

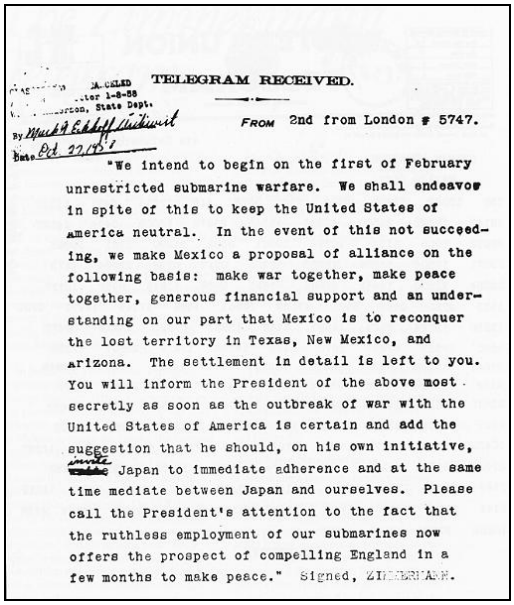
1. **Potpuno probijanje** (eng. Total break) - napadač je otkrio tajni ključ.
2. **Globalna dedukcija** (eng. Global deduction) - napadač je otkrio funkcijski ekvivalent algoritma za kriptiranje i dekriptiranje, ali ne i ključ.
3. **Lokalna dedukcija** (eng. instance or local deduction) - napadač je otkrio dodatne otvorene tekstove (ili kriptirane tekstove) koji ranije nisu bili poznati.
4. **Informacijska dedukcija** (eng. Information deduction) - napadač dobiva Shannon-ove informacije (Shannon-ova entropija – mjera informacija sadržanih u određenoj poruci) o otvorenim tekstovima (ili kriptirane tekstovima) koji ranije nisu bili poznati.
5. **Algoritam koji omogućuje razlikovanje** (eng. Distinguishing algorithm) - napadač može razlikovati kriptirani tekst od slučajne permutacije.

Uspješna kriptoanaliza donosi mogućnost pregleda tajnih poruka te može igrati važnu ulogu u povijesti. Na primjer, u drugom svjetskom ratu proboj Zimmermann-ovog telegrama (Slika 2) bio je ključni korak koji je doveo do uključenja SAD-a u rat.

Vlade raznih država davno su prepoznale prednosti kriptoanalize pa su osnovale organizacije za proboje kodova koje koriste druge nacije, poput organizacija NSA i GCHQ.

Prema složenosti postupaka, kriptoanaliza se može klasificirati u kategorije u odnosu na količinu resursa koje zahtjeva. Kategorije čini količina:

- **vremena** - broj potrebnih primitivnih operacija (osnovne instrukcije ili cijela metoda kriptiranja),
- **memorije** - količina memorijskog prostora potrebnog za pohranu za vrijeme napada,
- **podataka** - količina potrebnih otvorenih i šifriranih tekstova.



TELEGRAM RECEIVED.

FROM 2nd from London # 5747.

"We intend to begin on the first of February unrestricted submarine warfare. We shall endeavor in spite of this to keep the United States of America neutral. In the event of this not succeeding, we make Mexico a proposal of alliance on the following basis: make war together, make peace together, generous financial support and an understanding on our part that Mexico is to reconquer the lost territory in Texas, New Mexico, and Arizona. The settlement in detail is left to you. You will inform the President of the above most secretly as soon as the outbreak of war with the United States of America is certain and add the suggestion that he should, on his own initiative, ~~invite~~ Japan to immediate adherence and at the same time mediate between Japan and ourselves. Please call the President's attention to the fact that the ruthless employment of our submarines now offers the prospect of compelling England in a few months to make peace." Signed, ZIMMERMANN.

Slika 2 Zimmermann-ov telegram

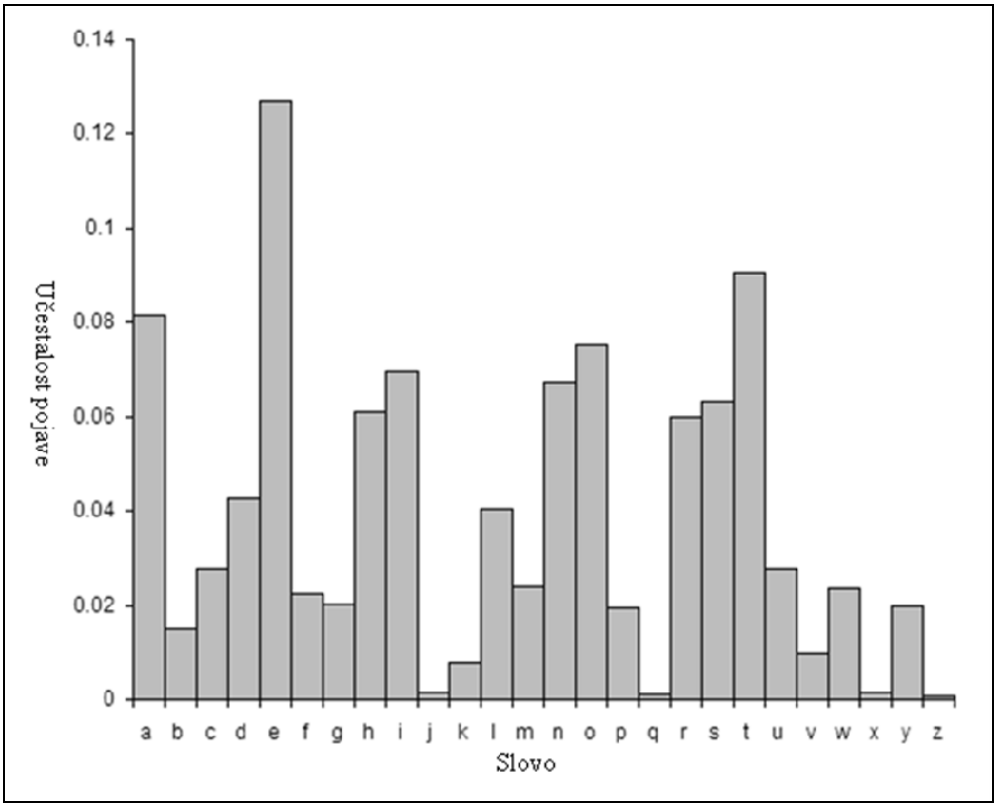
Izvor: Wikipedia

3. Vrste kriptoanalize

U nastavku dokumenta prikazane su osnove metode kriptoanalize podijeljene u pet osnovnih grupa.

3.1. Klasična kriptoanaliza

Klasična kriptoanaliza predstavlja najstariji oblik analize kriptografskih šifri, a uključuje tri metode. Prvu od metoda čini **analiza učestalosti** (eng. frequency analysis), tj. proučavanje učestalosti pojave pojedinih slova ili grupe slova u šifriranom tekstu. Zasniva se na činjenici da se u svakom dijelu šifriranog teksta određena slova i kombinacije slova pojavljuju s provjerenom vjerojatnošću. Zapravo za svaki jezik postoji određena distribucija pojave slova, a primjer za engleski jezik dan je na slici 3. U nekim slučajevima ovakva obilježja nešifriranog jezika su prisutna u šifriranom tekstu pa ih je moguće iskoristit u napadu koji koristi pretpostavku samo šifriranog teksta.



Slika 3 Distribucija pojave pojedinih slova u engleskom jeziku
Izvor: Wikipedia.

Druga metoda je **računanje podudaranja** (eng. *coincidence counting*), tehnika usporedbe dva teksta uz računanje koliko se puta isto slovo pojavljuje na istom mjestu u oba teksta. Rezultat, koji se može pokazati kao ukupni broj ili normalizirati dijeljenjem s očekivanim rezultatom, naziva se indeksom podudaranja.

Posljednju metodu predstavlja „**Kasiski ispitivanje“** (eng. Kasiski examination), također poznato pod nazivima Kasiski ispit ili metoda. Riječ je o metodi napada na šifru abecedne zamjene, poput Vigenère-ove šifre. Zasniva se na određivanju duljine ključne riječi te podijeli šifriranog teksta u *n* stupaca (gdje *n* predstavlja duljinu ključne riječi). Tada je svaki stupac moguće promatrati zasebno te primijeniti analizu učestalosti.

U nastavku dokumenta detaljno je objašnjena metoda računanja podudarnosti kao predstavnica klasičnih metoda kriptoanalize.

3.1.1. Računanje podudaranja

Računanje podudaranja zasniva se na određivanju indeksa podudaranja prilikom usporedbe dva teksta. Metoda je jednako značajna u analizi nešifriranog i šifriranog teksta, a daje dobre rezultate prilikom analize Vigenère-ove šifre. Za abecednu šifru s ponavljajućim ključem raspoređenim u matrici, učestalost podudarnosti u svakom stupcu će biti najveća kada je širina matrice višekratnik duljine ključa. Ova se činjenica koristi se za određivanje duljine ključa, što je prvi korak u probijanju sustava.

Spomenuta metoda računanja podudarnosti može pomoći u određivanju da li su dva teksta pisana u istom jeziku koristeći istu abecedu. Računanje podudarnosti za takve tekstove bit će izrazito različito od računanja podudarnosti za tekstove pisane u različitim jezicima. Na primjer, ako se zamisli abeceda koja ima samo dva slova: „A“ i „B“ pretpostavi se da se u jeziku slovo „A“ koristi 75% vremena, a slovo „B“ 25% vremena. Ako se dva teksta usporede, mogu se očekivati sljedeći parovi:

Par	Vjerojatnost
AA	56.25%
BB	6.25%
AB	18.75%
BA	18.75%

Sveukupno, vjerojatnost podudaranja je 62.5% (56.25% za „AA“ + 6.25% za „BB“).

Ako se promatra slučaj kada su oba teksta šifrirana uporabom šifre zamijene, gdje se slovo „A“ prikazuje slovom „B“ i obrnuto vjerojatnost pojave parova tada je sljedeća:

Par	Vjerojatnost
AA	6.25%
BB	56.25%
AB	18.75%
BA	18.75%

Cjelokupna vjerojatnost podudaranja je 62.5% (6.25% za „AA“ + 56.25% za „BB“), što je upravo jednako u slučaju nešifriranog teksta.

Poruka koja je kriptirana koristeći zamjenu (A,B) → (B,A) omogućuje pojavu sljedećih parova:

Par	Vjerojatnost
AA	18.75%
BB	18.75%
AB	56.25%
BA	6.25%

Vjerojatnost podudaranja je 37.5% (18.75% za „AA“ + 18.75% za „BB“).

Isti princip provodi se u prirodnim jezicima (poput engleskog jezika) gdje je moguće odrediti koji se simboli pojavljuju češće. Ako se uspoređuju dva teksta pisana istim jezikom, dobit će se visoko podudaranje. Također, teško je generirati slučajni tekst koji će imati razdiobu

vjerojatnosti kao stvarni tekst. Unatoč tomu, ovu je metodu moguće je koristiti za identifikaciju tekstova koji bi mogli sadržavati značajne informacije, određivanje duljine ključeva i sl.

Jednaka ideja može se primijeniti na jedan test gdje se uzorak uspoređuje sam sa sobom. Indeks podudaranja moguće je odrediti matematički kao:

$$\mathrm{IC} = \frac{\sum_{i=1}^{c} n_i(n_i - 1)}{N(N-1)/c}$$

gdje je *N* duljina teksta, n_1 i n_2 frekvencije slova *c* u abecedi (*c*=26 za engleski). Zbroj vrijednosti n_i mora biti *N*. Umnožak *n(n-1)* određuje broj kombinacija parova za *n* elemenata, što je potrebno podijeliti s vrijednošću *c* kako bi se dobila vjerojatnost pojave svakog para.

Očekivana srednja vrijednost IC može se izračunati preko relativne frekvencije slova:

$$\mathrm{IC}_{expected} = \frac{\sum_{i=1}^{c} f_i^{\,2}}{1/c}$$

Pri jednolikoj raspodijeli slova, očekivana vrijednost indeksa bila bi oko 1.0, ali u većini slučajeva raspodjela nije jednolika pa su očekivane vrijednosti dosta različite (tablica 1).

Jezik	Indeks IC
Engleski	1.73
Francuski	2.02
Njemački	2.05
Talijanski	1.94
Portugalski	1.94
Ruski	1.76
Španjolski	1.94

Tablica 1 Vrijednosti indeksa IC

3.2. Simetrični algoritmi

Simetrični algoritmi koriste se za napad na blokovske kriptografske sustave koji nekriptirani tekst dijele u blokove te koriste posebne funkcije za miješanje bloka teksta sa ključem. Na opisani način dobije se šifrirani tekst. Česta primjena simetričnih algoritama kriptoanalize su šifre nizova, gdje se nekriptirani tekst miješa s nizom bitova ključa preko određenih operacija (obično xor funkcija).

Najčešći oblici napada na ovakve sustave uključuju uporabu rječnika. Kako bi se to izbjeglo, u kriptografskim sustavima često se koriste veliki blokovi te permutacije i transformacije. U nastavku je dan kratki pregled svih algoritama koji spadaju u spomenutu skupinu.

Diferencijalna kriptoanaliza je opći oblik kriptoanalize koja se primjenjuje na blokovske šifre, ali i na hash funkcije i šifre nizova. U širem pogledu, radi se o proučavanju kako razlike u ulaznim vrijednostima mogu utjecati na rezultantne razlike u izlaznim vrijednostima. Ako se radi o blokovskim šiframa, metoda se odnosi na skup tehnika za praćenje razlika kroz mrežu tranformacija. Tako se pokušava otkriti gdje šifra pokazuje ponašanje koje nije pseudoslučajno kako bi se to ponašanje iskoristilo za otkrivanje tajnog ključa.

Bumerang napad je metoda kriptoanalize bloka šifriranih podataka koja se temelji na diferencijalnoj kriptoanalizi. Objavio ju je 1999. godine David Wagner, koji ju je iskoristio za proboj šifre COCONUT98 (eng. Cipher Organized with Cute Operations and N-Universal Transformation). Radi se o šifri koja je dizajnirana kao obrana protiv diferencijalne i linearne

kriptoanalize. Bumerang napad omogućio je načine napada za mnoge šifre koje su prethodno smatrane sigurnim.

Još jedan napad ove skupine čini **„brute force“ napad**, metoda proboja kriptografske sheme sistematskim pokušavanjem velikog broja mogućih ključeva (unos velikog broja ključeva s ciljem dekriptiranja poruke). U većini shema, teoretska vjerojatnost uspješnosti ovog napada je poznata pa se sustavi implementiraju na način da bude nemoguće izvesti napad. Praktična ostvarivost uspješnog izvođenja ovog napada uvelike ovisi o odabiru duljine ključa.

Daviesov napad predstavlja metodu statističke kriptoanalize, a koristi se kao napad na DES (eng. Data Encryption Standard) algoritam. Spomenuti napad je izumio Donald Davies 1987. godine, a 7 godina kasnije Eli Biham i Alex Biryukov su poboljšali tehniku napada. Spada u skupinu napada s poznatim nekriptiranim tekstom, a temelji se na neuniformnoj distribuciji izlaznih vrijednosti parova susjednih S-tablica (osnovni element algoritma). Funkcionira na način da se skupljaj mnogo poznatih parova šifriranih i nešifriranih znakova te računa empirijska distribucija pojedinog znaka.

Integralna kriptoanaliza je napad koji se primjenjuje na blokovske šifre, a temelji se na supstitucijsko-permutacijskim (eng. substitution-permutation) mrežama. Originalno ju je dizajnirao Lars Knudsen za napad na šifru Square (prethodnik AES standarda) pa je poznata i pod nazivom „Square napad“. Nakon uspješno izvedenog napada na spomenuti algoritam, iskorišten je za napad na razne druge algoritme poput Rijndaela, IDEA, Camellia i dr. Napad se obavlja uporabom skupine ili više skupina izabranih nekriptiranih tekstova kod kojih se dio zadržava konstantnim, a drugi dio prolazi kroz sve mogućnosti supstitucija i permutacija.

Sljedeći napad ove skupine predstavlja **linearna kriptoanaliza**. Radi se o općem obliku kriptoanalize koji se temelji na pronalaženju srodnih približnih vrijednosti šifri. Napad je razvijen za blokovske šifre i šifre nizova, a zajedno sa diferencijalnom kriptoanalizom čini dva najčešće korištena napada na blokovske šifre. Otkrio ga je Mitsuru Matusui te ga prvotno primijenio na šifru FEAL (šifra koja je predstavljana kao alternativa DES standardu).

Napad „meet-in-the-midle“ je napad koji nalikuje „birthday“ napadu jer koristi prostorno-vremenske odnose (eng. space-time tradeoff). Cilj napada je pronaći posebnu vrijednost u svakom rasponu i domenama kompozicije dvaju funkcija. Napad funkcionira na način da napadač kriptira sve primjere nekriptiranog teksta svim mogućim ključevima te rezultat pohrani u memoriju. Drugi korak uključuje dekriptiranje šifriranih parova svakim mogućim ključem, a bilo koje slaganje rezultata može otkriti ključ. Ovaj napad razvili su Diffie i Hellman 1977. godine kao napad na blokovske šifre.

„Mod n“ kriptoanaliza je napad na blokovske šifre i šifre nizova. Radi se o obliku kriptoanalize razdjeljivanjem koja iskorištava promjenjivost u tome kako šifra funkcionira preko ekvivalentnih klasa (skup svih elemenata u nekom skupu koji su ekvivalentni s danim elementom) modulo *n*. Metodu su predložili 1999. godine John Kelsey, Bruce Schneier i David Wagner, a korištena je za napad na šifre RC5P (inačicu šifre RC5) i M6 (blokovske šifre standarda FireWire).

Česti oblik kriptoanalize simetričnih sustava je **napad povezanim ključevima**. Riječ je o metodi gdje napadač može otkriti operacije šifre kroz nekoliko različitih ključeva čije su vrijednosti inicijalno nepoznate. Ipak napadač mora znati neku matematičku vezu koja povezuje ključeve. Moderna kriptografija donijela je napredak ove metode jer je čovjeku gotovo nemoguće odrediti nekriptirani tekst preko brojnih tajnih ključeva koji su povezani na neki način. Tek je razvoj složenih računalnih protokola omogućio uspješno obavljanje ovakvog napada.

Napad klizanjem (eng. slide attack) je oblik kriptoanalize koji je dizajniran kako bi se pobila ideja da se slabe šifre mogu poboljšati povećanjem broja runda kriptiranja. Ovaj napad funkcionira tako da broj runda učini beznačajnim analizirajući grafikon ključa (algoritam kojim se računaju podključevi u rundama). Time se omogućuje iskorištavanje ranjivosti u grafikonu (npr. ponavljanje istog ključa u rundama) kako bi se probila šifra. Napad su prvi opisali David Wagner i Alex Biryukov, a Bruce Schneier mu je dao trenutni naziv 1999. godine.

XSL (eng. eXtended Sparse Linearization) napad je metoda kriptoanalize za blokovske šifre, a objavili su ju 2002. godine Nicolas Courtois i Josef Pieprzyk. Autori su tvrdili kako spomenuta metoda može probiti AES standard puno brže od „brute force“ napada. Ipak, metoda zahtjeva puno rada pa ne smanjuje napore u probijanju AES šifre. Napad se oslanja na analiziranje postupka šifranja te rješavanje sustava kvadratnih simultanih jednadžbi. Obično se ovakvi sustavi sastoje od velikog broja jednadžbi s puno varijabli (8000 jednadžbi s 1600 varijabli za AES). XSL napad predstavlja specijalizirani algoritam koji se koristi za rješavanje sustava i obnovu ključa.

3.2.1. Diferencijalna kriptoanaliza

Otkriće diferencijalne kriptoanalize je pripisano Eliu Bihamu i Adiu Shamiru u kasnim 80-im godinama 20. stoljeća. Isti su autori objavili opis teoretskih ranjivosti u DES algoritmu te mogućnost njegovog proboja putem diferencijalne kriptoanalize. Godine 1994. član IBM DES tima - Don Coppersmith objavio je kako je diferencijalna kriptoanaliza već bila poznata njihovom timu 1974. godine kao „Tickle“ napad. Prema izvješću Stevena Lavya organizacije IBM i NSA su odvojeno otkrile spomenutu tehniku.

Dok je standard DES dizajniran kako bi bio otporan na diferencijalnu kriptoanalizu, mnoge druge šifre su ranjive. Jedan od prvih ciljeva napada bila je blokovska šifra FEAL. Ovom tehnikom moguće je probiti FEAL-4 (šifra koja uključuje četiri runde) uz uporabu samo osam odabranih nekriptiranih tekstova, a čak i FEAL s 31 rundom pokazuje ranjivost na ovaj napad.

Diferencijalna kriptoanaliza obično spada pod napade uz odabrane nekriptirane tekstove. Osnovna tehnika koristi nekriptirani tekst povezan s konstantnom razlike (eng. difference) između tog teksta i zadnje runde kriptiranja. Razlika se može definirati na nekoliko načina, ali obično se koristi XOR operacija. Ako postoje ulazni nizovi X' i X'' gdje je $X=[X_1X_2...X_n]$, i izlazne vrijednosti Y' i Y'', razlika između ulaza je:

$DX = X' \text{ xor } X''$

$DX = [DX_1\ DX_2\ ...\ DX_n]$

gdje je $DX_i = X'_i \text{ xor } X''_i$, X'_i i X''_i su i-ti bitovi od X' i X''.

Slično je $DY = Y' \text{ xor } Y''$ izlazna razlika:

$DY = [DY_1\ DY_2\ ...\ DY_n]$

gdje je $DY_i = Y'_i \text{ xor } Y''_i$.

U kriptografskom sustavu s idealnim generatorom slučajnih brojeva, vjerojatnost da se nad danom razlikom ulaza DX pojavi određena razlika izlaza DY je $1/(2)^n$ gdje je *n* broj bitova od X. Diferencijalna kriptoanaliza traži slučaj u kojem se određeni DY nad danim DX pojavljuje sa vrlo velikom vjerojatnošću p_d (puno većom od $1/2^n$). Par (DX, DY) se naziva diferencija.

Za bilo koju šifru, razlika ulaznih podataka mora biti pažljivo izabrana (cilj je dobiti razliku DY koja se pojavljuje s velikom vjerojatnošću) kako bi napadač uspješno otkrio ključ. Standardna metoda uključuje praćenje puta najvjerojatnijih razlika kroz različite korake kriptiranja.

Postoje tri podvrste diferencijalne kriptoanalize:

1. **Diferencijalna kriptoanaliza višeg reda** (eng. higher-order differential cryptanalysis) – generalizacija diferencijalne kriptoanalize za napad na blokovske šifre, a temelji se na usporedbi razlika među razlikama.
2. **Odrezana diferencijalna kriptoanaliza** (eng. truncated differential cryptanalysis) – generalizacija diferencijalne kriptoanalize za napad na blokovske šifre, a temelji se na usporedbi razlika koje su samo djelomično određene.
3. **Nemoguća diferencijalna kriptoanaliza** (eng. impossible differential cryptanalysis) – generalizacija diferencijalne kriptoanalize za napad na blokovske šifre, a temelji se na iskorištavanju razlika koje su nemoguće u nekom unutarnjem stanju algoritma.

Diferencijalna kriptoanaliza prikazana je preko 4x4 S-tabllice koja ima ulaz $X = [X_1\ X_2\ X_3\ X_4]$ i izlaz $Y = [Y_1\ Y_2\ Y_3\ Y_4]$. Promatraju se svi parovi razlika S-tablice (DX, DY), a vjerojatnost za DY uz dani DX se može dobiti razmatrajući parove (X', X'') takvi da je $DX = X' \text{ xor } X''$. Potrebno je računati sa svih 16 vrijednosti za X', dok DX ograničava vrijednost X'' na $X'' = X' \text{ xor } DX$.

Ako se pogledaju S-tablice ovog primjera kriptosustava, za svaki par ulaza (X', X'' = X' xor DX) mogu se dobiti izlazi DY. Primjer su binarne vrijednosti od X i Y za parove (X, X xor DX) i odgovarajuće vrijednosti DY prikazane u Tablica 2. Vrijednost razlike DX je 1011 (heksidecimalno B), 1000 (heksidecimalno 8) i 0100 (heksadecimalno 4). Zadnja tri stupca predstavljaju vrijednost DY za vrijednosti X i određeni DX za svaki stupac.

X	Y	ΔY		
		$\Delta X = 1011$	$\Delta X = 1000$	$\Delta X = 0100$
0000	1110	0010	1101	1100
0001	0100	0010	1110	1011
0010	1101	0111	0101	0110
0011	0001	0010	1011	1001
0100	0010	0101	0111	1100
0101	1111	1111	0110	1011
0110	1011	0010	1011	0110
0111	1000	1101	1111	1001
1000	0011	0010	1101	0110
1001	1010	0111	1110	0011
1010	0110	0010	0101	0110
1011	1100	0010	1011	1011
1100	0101	1101	0111	0110
1101	1001	0010	0110	0011
1110	0000	1111	1011	0110
1111	0111	0101	1111	1011

Tablica 2 Primjer razlike parova za S-tablicu

U tablici se može vidjeti da je broj pojavljivanja DY=0010 za DX=1011 8 od 16 mogućih vrijednosti (vjerojatnost 1/2), zatim broj pojavljivanja DY=1011 za DX=1000 4 od 16, te DY=1010 za DX=0100 je 0 od 16. Podaci za S-tablicu se mogu prikazati u tablici distribucije diferencija u kojoj retci predstavljaju vrijednosti DX, a stupci DY (Tablica 3).

Svaki element tablice predstavlja broj pojavljivanja odgovarajuće izlazne razlike DY uz danu ulaznu razliku DX. Osim posebnog slučaja gdje je DX=0 i DY=0, najveća vrijednost u tablici je 8, što odgovara DX=B i DY=2 (vjerojatnost 8/16). Najmanja vrijednost u tablici je 0, a pojavljuje se za više parova.

Iz tablice diferencijala mogu se izvesti sljedeća svojstva:

1. suma svih elemenata u retku je 2^n=16,
2. suma elemenata u stupcu je 2^n=16,
3. sve vrijednosti elemenata su parne,
4. ulazna razlika DX=0 daje izlaznu diferenciju DY=0 za preslikavanje S-tablice jedan na jedan pa element u gornjem desnom uglu u tablici ima vrijednost 2^n=16, a sve ostale vrijednosti u prvom retku i prvom stupcu su 0 i
5. kada bi bilo moguće konstruirati idealnu S-tablicu, koja ne daje informacije o razlikama izlaza uz dane ulaze, svi bi elementi u tablici bili jednaki i vjerojatnost pojavljivanja odgovarajuće vrijednosti DY uz dani DX bila bi $1/2^n$=1/16.

Ulazna razlika \ Izlazna razlika	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	2	0	0	0	2	0	2	4	0	4	2	0	0
2	0	0	0	2	0	6	2	2	0	2	0	0	0	0	2	0
3	0	0	2	0	2	0	0	0	0	4	2	0	2	0	0	4
4	0	0	0	2	0	0	6	0	0	2	0	4	2	0	0	0
5	0	4	0	0	0	2	2	0	0	0	4	0	2	0	0	2
6	0	0	0	4	0	4	0	0	0	0	0	0	2	2	2	2
7	0	0	2	2	2	0	2	0	0	2	2	0	0	0	0	4
8	0	0	0	0	0	0	2	2	0	0	0	4	0	4	2	2
9	0	2	0	0	2	0	0	4	2	0	2	2	2	0	0	0
A	0	2	2	0	0	0	0	0	6	0	0	2	0	0	4	0
B	0	0	8	0	0	2	0	2	0	0	0	0	0	2	0	2
C	0	2	0	0	2	2	2	0	0	0	0	2	0	6	0	0
D	0	4	0	0	0	0	0	4	2	0	2	0	2	0	2	0
E	0	0	2	4	2	0	0	0	6	0	0	0	0	0	2	0
F	0	2	0	0	6	0	0	0	0	4	0	2	0	0	2	0

Tablica 3 Tablica distribucije diferencijala

3.2.2. Linearna kriptoanaliza

Linearna kriptoanaliza uključuje postupke pronalaženja srodnih približnih vrijednosti šifri, a sastoji se od dva koraka. Prvi predstavlja izgradnju linearnih jednadžbi povezanih s nekriptiranim tekstom, šifriranim tekstom i ključem čija je devijacija „visoka“ (tj. vjerojatnosti zadržavanja preko prostora svih mogućih vrijednosti varijabli je što bliža nuli ili jedinici). Drugi korak je uporaba linearnih jednadžbi u konjunkciji s poznatim parovima šifriranog i/ili nešifriranog teksta kako bi se odredio ključ.

Za potrebe linearne kriptoanalize, linearna jednadžba izražava jednakost dvaju izraza koji se sastoje od binarnih varijabli kombiniranih sa „xor“ operacijom. Na primjer, sljedeća jednadžba sadrži „xor“ sumu prvog i trećeg bita nešifriranog teksta te prvog bita šifriranog teksta što je upravo jednako drugom bitu ključa:

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

U idealnim šiframa, svaka linearna jednadžba povezana s nešifriranim tekstom, šifriranim tekstom i ključem bi bila uspješna s vjerojatnošću od 1/2. Budući da jednadžbe u realnim linearnim sustavima variraju s vjerojatnostima, često se primjenjuju linearne aproksimacije. Procedure za konstrukciju aproksimacija su različite za svaku šifru. U osnovom tipu blokovske šifre, mreži supstitucija i permutacija, analiza je koncentrirana na S-tablice kao jedinom nelinearnom dijelu šifre. Za dovoljno male S-tablice moguće je odrediti svaku moguću linearnu jednadžbu povezanu s ulazima i izlazima S-tablice te izračunati osnovne bitove i odabrati najbolje. Linearna aproksimacija S-tablica mora biti kombinirana s 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:

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

moguće je primijeniti razne algoritme koji korištenjem poznatih šifriranih/nešifriranih parova teksta mogu otkriti vrijednosti bitova ključeva u aproksimacijama. Za svaku skupinu vrijednosti ključa s 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 polovice šifriranih/nešifriranih parova uzima se kao najvjerojatnija skupina 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.

Budući da se rukuje S-tablicama, prikazana je osnovna ranjivost njihovog korištenja. Ako je zadana S-tablica s ulazom X = [X_1 X_2 X_3 X_4] i odgovarajućim izlazom Y = [Y_1 Y_2 Y_3 Y_4], mogu se ispitati sve linearne aproksimacije tako da se izračunaju devijacije vjerojatnosti za svaku.

Neka je za danu S-tablicu, linearni izraz slijedeći:

X_2 xor X_3 xor Y_1 xor Y_3 xor Y_4 = 0
X_2 xor X_3 = Y_1 xor Y_3 xor Y_4.

Primjenom svih 16 mogućih vrijednosti ulaza X i ispitivanjem odgovarajuće izlazne vrijednosti Y, može se vidjeti da je za 12 od 16 slučajeva, gornji izraz istinit. Dakle, devijacija vjerojatnosti je 12/16 – 1/2 = 1/4, a rezultati su prikazani u Tablica 4

X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4	$X_2 \oplus X_3$	$Y_1 \oplus Y_3 \oplus Y_4$	$X_1 \oplus X_4$	Y_2	$X_3 \oplus X_4$	$Y_1 \oplus Y_4$
0	0	0	0	1	1	1	0	0	0	0	1	0	1
0	0	0	1	0	1	0	0	0	0	1	1	1	0
0	0	1	0	1	1	0	1	1	0	0	1	1	0
0	0	1	1	0	0	0	1	1	1	1	0	0	1
0	1	0	0	0	0	1	0	1	1	0	0	0	0
0	1	0	1	1	1	1	1	1	1	1	1	1	0
0	1	1	0	1	0	1	1	0	1	0	0	1	0
0	1	1	1	1	0	0	0	0	1	1	0	0	1
1	0	0	0	0	0	1	1	0	0	1	0	0	1
1	0	0	1	1	0	1	0	0	0	0	0	1	1
1	0	1	0	0	1	1	0	1	1	1	1	1	0
1	0	1	1	1	1	0	0	1	1	0	1	0	1
1	1	0	0	0	1	0	1	1	1	1	1	0	1
1	1	0	1	1	0	0	1	1	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1	0	1	0
1	1	1	1	0	1	1	1	0	0	0	1	0	1

Tablica 4 Linearne aproksimacije S-kutije

Slično, za jednadžbu X_1 xor X_4 = Y_2 devijacija vjerojatnosti je 0, a za jednadžbu X_3 xor X_4 = Y_1 xor Y4, 2/16 – 1/2 = -3/8. U posljednjem slučaju, najbolja aproksimacija je afina aproksimacija, što se može vidjeti po negativnom predznaku. Uspješnost napada se temelji na veličini devijacije, a afine aproksimacije se mogu koristiti ekvivalentno linearnim aproksimacijama.

Sve linearne aproksimacije S-tablice za dani primjer kriptografskog sustava su prikazane u Tablica 5.

Zbroj ulaza \ Zbroj izlaza	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	+8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	−2	−2	0	0	−2	+6	+2	+2	0	0	+2	+2	0	0
2	0	0	−2	−2	0	0	−2	−2	0	0	+2	+2	0	0	−6	+2
3	0	0	0	0	0	0	0	0	+2	−6	−2	−2	+2	+2	−2	−2
4	0	+2	0	−2	−2	−4	−2	0	0	−2	0	+2	+2	−4	+2	0
5	0	−2	−2	0	−2	0	+4	+2	−2	0	−4	+2	0	−2	−2	0
6	0	+2	−2	+4	+2	0	0	+2	0	−2	+2	+4	−2	0	0	−2
7	0	−2	0	+2	+2	−4	+2	0	−2	0	+2	0	+4	+2	0	+2
8	0	0	0	0	0	0	0	0	−2	+2	+2	−2	+2	−2	−2	−6
9	0	0	−2	−2	0	0	−2	−2	−4	0	−2	+2	0	+4	+2	−2
A	0	+4	−2	+2	−4	0	+2	−2	+2	+2	0	0	+2	+2	0	0
B	0	+4	0	−4	+4	0	+4	0	0	0	0	0	0	0	0	0
C	0	−2	+4	−2	−2	0	+2	0	+2	0	+2	+4	0	+2	0	−2
D	0	+2	+2	0	−2	+4	0	+2	−4	−2	+2	0	+2	0	0	+2
E	0	+2	+2	0	−2	−4	0	+2	−2	0	0	−2	−4	+2	−2	0
F	0	−2	−4	−2	−2	0	+2	0	0	−2	+4	−2	−2	0	+2	0

Tablica 5 Linearna aproksimacija

Svaki element tablice predstavlja broj podudaranja između linearne jednadžbe predstavljene heksadecimalno kao “Zbroj ulaza“ i zbroja bitova izlaza “Zbroj izlaza“ umanjen za osam. Dakle, dijeljenjem vrijednosti elementa sa 16 dobiva se devijacija vjerojatnosti za određenu linearnu kombinaciju bitova ulaza i izlaza. Heksadecimalna vrijednost koja predstavlja zbroj, kada se prikaže kao binarna vrijednost, pokazuje varijable koje su uključene u zbroj. Za linearnu kombinaciju ulaznih varijabli predstavljenih kao

$$a_1 \text{ and } X_1 \text{ xor } a_2 \text{ and } X_2 \text{ xor } a_3 \text{ and } X_3 \text{ xor } a_4 \text{ and } X_4$$

gdje su a_i Î{0, 1}, a heksadecimalna vrijednost predstavlja binarnu vrijednost a1a2a3a4 (a1 najznačajniji bit).

Slično tome, za linearnu kombinaciju izlaznih bitova:

$$b1 \text{ and } Y1 \text{ xor } b2 \text{ and } Y2 \text{ xor } b3 \text{ and } Y3 \text{ xor } b4 \text{ and } Y4$$

gdje su bi Î{0, 1}, a heksadecimalne vrijednosti predstavljaju binarni vektor $b_1b_2b_3b_4$.
Dakle, devijacija linearne jednadžbe X_3 xor X_4 = Y_1 xor Y_4 (hex ulaz 3 i hex izlaz 9) je -6/16 = -3/8, a vjerojatnost da je linearna jednadžba istinita je 1/2 – 3/8 = 1/8.

U tablici linearnih aproksimacija mogu se uočiti sljedeća svojstva:

1. Vjerojatnost da je bilo koji zbroj nepraznog skupa bitova izlaza jednak zbroju koji ne uključuje ulazne bitove je točno 1/2,
2. Linearna kombinacija koje ne uključuje bitove izlaza će uvijek biti jednaka linearnoj kombinaciji bez bitova ulaza s konačnom devijacijom od +1/2 i vrijednošću tablice od +8 u gornjem lijevom kutu.
3. Suma bilo kojeg retka ili stupca mora biti +8 ili -8.

3.3. „Hash“ funkcije

„Hash“ funkcija je svaka dobro definirana procedura ili matematička funkcija koja pretvara veliku količinu podataka u mali podatak, obično jedinstveni cjelobrojni broj koji može poslužiti kao indeks u nekoj listi. Vrijednosti koje se dobiju kao izlaz „hash“ funkcije nazivaju se „hash“ vrijednosti ili kodovi.

Sljedeća svojstva određuju sigurnost „hash“ funkcije H():

- jednosmjernost – za bilo koji izlaz h, računski nije moguće naći ulaz x takav da je H(x)=h.
- slaba otpornost na koliziju – za bilo koji ulaz x, računski nije moguće naći drugi ulaz y¹x takav da je H(x)=H(y).
- jaka otpornost na koliziju – računski je nemoguće naći par (x, y) takve da je H(x)=H(y).
- domenska otpornost (eng. pre-image resistance) – za dani izlaz y=h(x) (kada nije dan odgovarajući ulaz x) praktično je nemoguće pronaći x.
- druga domenska otpornost (eng. 2nd pre-image resistance) – za dani izlaz y=H(x) i odgovarajući ulaz x praktično je nemoguće pronaći drugi ulaz z¹x takav da je H(z)=H(x).
- svojstvo slučajnog predviđanja – funkcija H() se ponaša kao slučajno odabrana funkcija.

Postupak kriptoanalize koji se primjenjuje na opisane funkcije naziva se **„birthday“ napad**. Riječ je o metodi koja se zasniva na rješavanju matematičkog problema koji se nalazi iza „birthday“ problema u teoriji vjerojatnosti. Cilj napada je pronaći dvije različite ulazne vrijednosti za x_1, x_2 za funkciju f takve da vrijedi $f(x_1) = f(x_2)$, tj. pronaći koliziju.

3.3.1. „Birthday“ napad

„Birthday“ napad je vrsta kriptoanalitičkog napada koji se temelji na pronalaženju kolizije, tj. parova različitih vrijednosti ulaza koji daju jednak izlaz. Metoda koja se koristi za pronalaženje kolizije je jednostavno izračunavanje funkcije f za različite ulazne vrijednosti koje mogu biti slučajno odabrane. Zahvaljujući „birthday“ problemu ova metoda može biti vrlo učinkovita.

Vjerojatnost da se ne nađe kolizija za k različitih ulaza sa „hash“ funkcijom f koja može izračunati n različitih vrijednosti je:

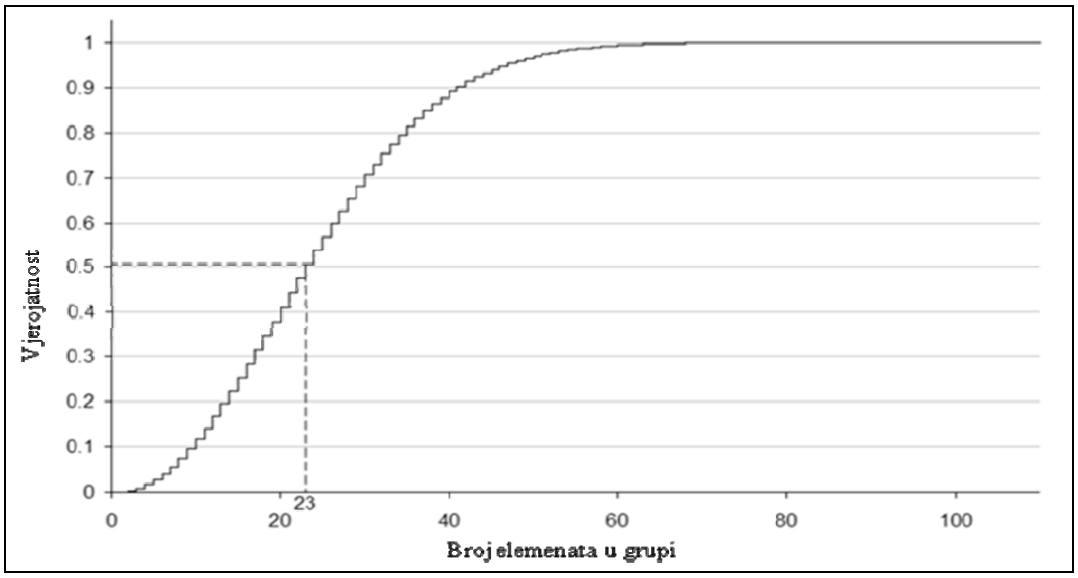
$$(1-1/n)(1-2/n)...(1-(k-1)/n) = 1^{(1-i/n)}.$$

Broj ulaza k potrebnih za generiranje kolizije sa „hash“ funkcijom može se približno odrediti putem izraza:

$$k \sim Ö(2n\ln(1/(1-e)))$$

gdje je *n* veličina rezultata „hash“ funkcije, a *e* vjerojatnost pojavljivanja kolizije. Npr. za vjerojatnost kolizije 50% dobiva se k~1.17Ö n. To znači da je za dobivanje kolizije potrebno Ö (2n)=2 n / 2 različitih ulaza.

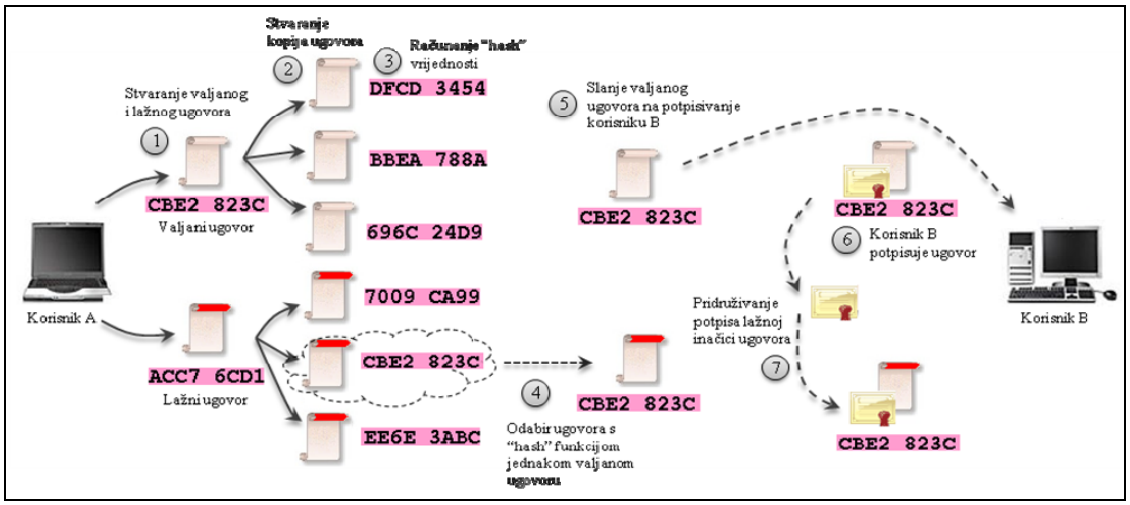
Slika 4 prikazuje odnos vjerojatnosti pronalaska kolizije i broja elemenata u grupi koja se analizira. Vidljivo je da je u grupi od 23 elementa vjerojatnost pronalaska istih elemenata jednaka 50 %. Ako se ovaj problem primjeni na ljude, može se pretpostaviti da u grupi od 23 osobe postoje dvije koje imaju rođendan na isti dan s vjerojatnošću od 50 %. Za 57 i više osoba, vjerojatnost je veća od 99%, a dostiže 100% kada se broj ljudi poveća na 366.



Slika 4 „Birthday“ problem

„Birthday“ napad traži bilo koje poklapanje para pa zahtjeva N stanja i generiranje O(N) stanja. Stoga „birthday“ napad može pronaći poklapajući par elemenata mnogo brže nego iscrpno pretraživanje koje traži poklapanje za odabrani cilj. Naivna implementacija „birthday“ napada jednostavno isprobava različite ulaze, dok god se ne pronađe poklapanje „hash“ vrijednosti. Za ovaj postupak bila bi potrebna velika količina memorije u slučaju kada su korištene složene „hash“ funkcije (npr. za probijanje 48-bitne „hash“ funkcije bilo bi potrebno $2*48*1.17\ \sqrt{(2^48)}$~1797Mb).

Digitalni potpisi mogu biti ranjivi na opisani napad jer se poruka *m* obično potpisuje s prvim izračunatim izlazom f(m), gdje je *f* kriptografska „hash“ funkcija, a zatim se koristi neki tajni ključ za potpisivanje f(m). Ako se pretpostavi da korisnik A želi navesti korisnika B na potpisivanje lažnog ugovora trebao bi pripremiti ugovor *m* i lažnu kopiju *m*'. Korisnik A tada pronalazi pozicije gdje *m* može biti izmijenjen bez promjene značenja (zarezi, praznine i sl.). Kombiniranjem tih izmjena korisnik A može kreirati veliki broj varijacija ugovora *m*. Slično tome, korisnik A također stvara veliki broj inačica lažnog ugovora *m*'. Zatim se na kopije primjeni „hash“ funkcija dok se ne pronađe inačica koja daje istu „hash“ vrijednost kao i originalni ugovor. Korisnik A šalje korisniku B na potpisivanje originalni ugovor te nakon potpisivanja uzima potpis i pridružuje ga lažnoj inačici. Opisani postupak prikazan je na Slika 5.Kako bi se izbjegao ovakav problem potrebno je odabrati veliku duljinu izlaznih vrijednosti „hash“ funkcije (dvostruko više bita nego je potrebno za sprječavanje „brute force“ napada) kako bi *m*' bilo nemoguće za izračunati.

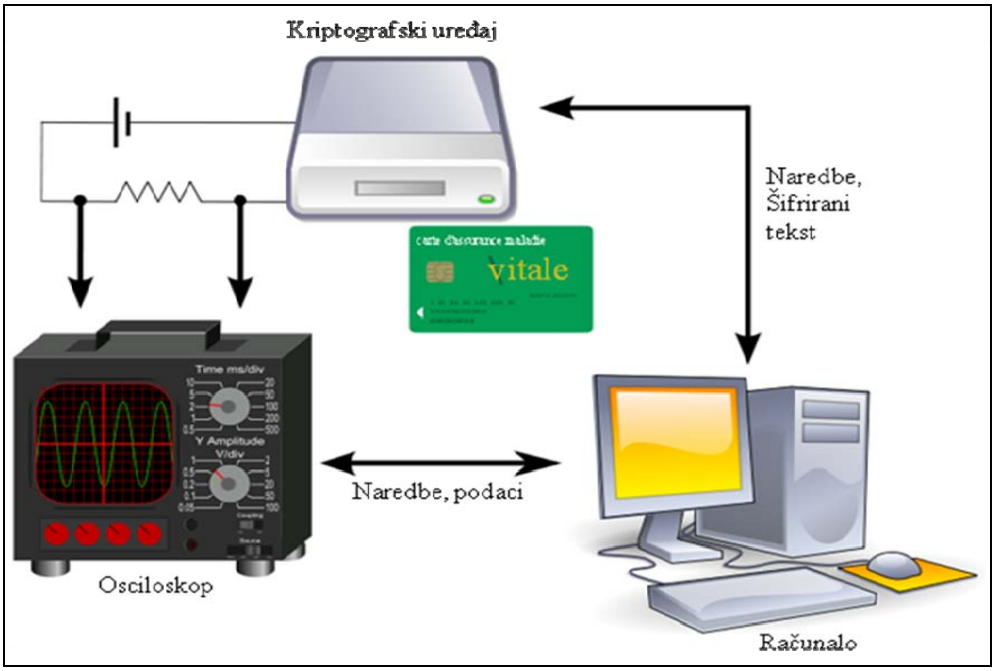


Slika 5 Birthday napad kod digitalnih certifikata

3.4. „Side channel“ napadi

U kriptografiji, „side channel“ napadi temeljeni su na informacijama prikupljenim iz fizičke implementacije kriptografskog sustava. Informacije koje se skupljaju uključuju podatke o vremenu trajanja kriptiranja, korištenju energije pa čak i zvučnim efektima i sl. Mnogi od ovih napada zahtijevaju odgovarajuće tehničko znanje o unutrašnjim operacijama sustava na kojem je implementirana kriptografski algoritam.

Jedan od oblika ovog napada je **analiza energije** (eng. power analysis). Radi se o napadu u kojem napadač proučava iskorištavanje energije uređaja za izvođenje kriptografije. Pokazalo se da na taj način on može izvući kriptografske ključeve i druge tajne informacije iz uređaja. Jednostavniji oblik ovog napada je SPA (eng. Simple power analysis) koja se odnosi na vizualni pregled grafa električne aktivnosti kroz vrijeme. Napredniji oblik tehnike je DPA (eng. Differential power analysis) koji može napadaču omogućiti izračun unutrašnjih vrijednosti statističkom analizom podataka prikupljenih sa mnogih kriptografskih operacija. Shema spomenutog postupka dana je na Slika 6. Oba su postupka predstavili 1998. godine Paul Kocher, Joshua Jaffe i Benjamin Jun.



Slika 6 DPA

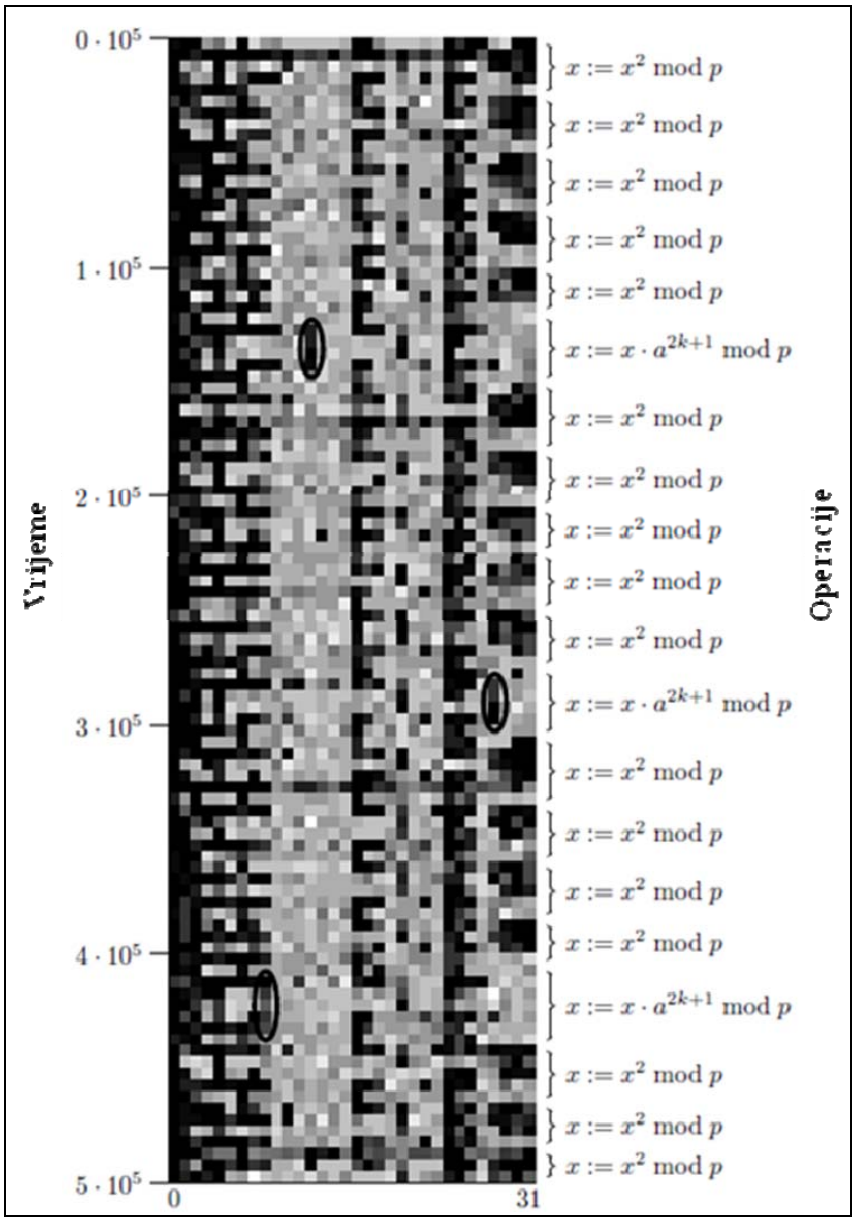
Drugi predstavnik ove skupine je **vremenski napad** (eng. timing attack). Riječ je o napadu u kojem napadač pokušava ugroziti kriptografski sustav analiziranjem vremena koje se koristi za izvođenje kriptografskog algoritma. Svaka logička operacija zahtjeva određeno vrijeme za izvođenje, a to vrijeme može se razlikovati s obzirom na ulazne vrijednosti. Preciznim mjerenjem vremena svake operacije, napadač može otkriti metodu kriptiranja.

3.4.1. Vremenski napad

Vremenski napad je primjer u kojem napadač iskorištava implementacijsku snagu algoritma, a ne sam algoritam. Sam se postupak temelji na mjerenju vremena obavljanja određenih operacija kako bi se odredio ulaz.

Slika 7. prikazuje primjer mjerenja vremena izvođenja operacija isječka programa u alatu „OpenSSL“ inačice 0.9.7c. Boja svakog bloka ukazuje na broj ciklusa koji su potrebni za pristup zapisima u pričuvnoj memoriji. Na primjer, crni blok označava trajanje od 170 ciklusa, a bijeli 120 ciklusa. Zaokružena područja otkrivaju informacije o broju a^{2k+1} koji se koristi.

Isti se algoritam može ponovno implementirati kako bi se spriječilo odavanje informacija o vremenu na način da se osigura izvođenje svake operacije u određenom vremenskom intervalu. U takvim su sustavima vremenski napadi beskorisni.



Slika 7 Mjerenje vremena izvođenja operacija

Izvor: Daemonology.net

Praktičnost napada ukazuje na nekoliko stvari:

- Neovisnost o algoritmu – teoretska sigurnost algoritma ne mora biti ugrožena kako bi se omogućilo izvođenje vremenskog napada.
- Pronalazak informacija o vremenu je rutina – mjerenje vremena odziva ne zahtjeva posebna znanja o kriptografskim sustavima.

Vremenski napad je učinkovit protiv raznih algoritama uključujući RSA, ElGamal i DES. Godine 2003. Boneh and Brumley demonstrirali su jednostavni vremenski napad na SSL web poslužitelje te uspjeli otkriti privatni ključ u samo nekoliko sati. Demonstracija je potaknula razvoj i uporabu tzv. „blinding“ tehnika u SSL implementaciji kako bi se poništila povezanost ključa i vremena kriptiranja.

Ovi napadi su jednostavniji za izvesti ako se pozna unutrašnjost implementacije i sam kriptografski sustav koji se koristi.

3.5. Mrežni napadi

Mrežni napadi su posebna skupina napada gdje se zlonamjerni korisnik raznim aktivnostima na mreži miješa u komunikaciju korisnika. Iako ove metode ne predstavljaju oblike matematičke kriptoanalize, važne su jer napadačima omogućuju presretanje informacija ključnih za provođenje kriptoanalize.

Osnovni predstavnik ove skupine napada je **MITM (eng. man-in-the-middle) napad**. Radi se o obliku aktivnog prisluškivanja u kojem napadač uspostavlja neovisnu vezu sa žrtvama i prenosi poruke među njima. Pri tome žrtve imaju dojam da pričaju izravno preko privatne veze iako cijelim razgovorom upravlja napadač. Također, napadač mora biti u mogućnosti presresti sve poruke koje žrtve razmjenjuju kako bi umetnuo nove poruke. Ovaj je napad uspješan samo kada se napadač može lažno predstaviti na oba kraja komunikacije. Većina kriptografskih protokola uključuje neki oblik krajnje autentifikacije za obranu od ovakvih i sličnih prijetnji.

Napad ponavljanjem (eng. replay attack) je oblik mrežnog napada u kojem je valjan prijenos podataka zlonamjerno ponovljen ili prekinut. Izvođenje napada nije ograničeno samo na pokretača komunikacije, nego tu mogućnost ima i suprotna strana. Jedan od mogućih načina obrane od ovakvih napada je uporaba oznaka sjednica (eng. session tokens).

3.6. Vanjski napadi

Vanjski napadi odnose se na skupinu metoda kojom napadači skupljaju osjetljive podatke. U slučaju kriptoanalize skupljaju se podaci o tajnim ključevima.

Jedan od oblika vanjskih napada je **„black-bag“ napad** kao blaži oblik izraza za skupljanje kriptografskih podataka putem krađe, a sastoji se od instalacije programa za zapis lozinki unesenih preko tipkovnice ili trojanskih konja. Iako su razvijene razne tehnike dohvaćanja informacija o kriptografskim sustavima i šiframa, ove metode se mogu odnositi i na jednostavno kopiranje/prepisivanje lozinki. Cilj ovih metoda je skupljanje osjetljivih informacija, poput kriptografskih ključeva, pa predstavlja kontrast matematičkim ili tehničkim kriptografskim napadima. Zbog toga ove metode tehnički ne spadaju u oblike kriptoanalize, ali pridaje im se velika pažnja zbog ozbiljnosti prijetnji koje nose.

Drugi oblik vanjskih napada koji se koristi u kriptoanalizi je **„rubber-hose“ napad**. Radi se o obliku otkrivanja kriptografskih tajni od osoba koje su pod prisilom. Ovaj napad također predstavlja kontrast matematičkim i tehničkim oblicima kriptoanalize.

4. Primjeri uporabe

4.1. Primjer računanja podudaranja

Kriptoanaliza računanjem podudaranja bit će prikazana na primjeru kriptiranog teksta danog u nastavku (grupiranje u skupine od 5 znakova nije povezano s pravom duljinom riječi).

```
QPWKA LVRXC QZIKG RBPFA EOMFL  JMSDZ VDHXC XJYEB IMTRQ WNMEA
IZRVK CVKVL XNEIC FZPZC ZZHKM  LVZVZ IZRRQ WDKEC HOSNY XXLSP
MYKVQ XJTDC IOMEE XDQVS RXLRL KZHOV
```

Pretpostavlja se da je tekst pisan engleskim jezikom te kriptiran Vigenère-ovim kodom s normalnim A-Z komponentama i kratkim ključem koji se ponavlja. Moguće je šifrirani tekst podijeliti u nekoliko stupaca kao što je prikazano u nastavku.

```
QPWKALV
RXCQZIK
GRBPFAE
OMFLJMS
DZVDHXC
XJYEBIM
TRQWN…
```

Ako se duljina ključa poklopi s pretpostavljenim brojem stupaca, sva slova jednog stupca mogu se dekriptirati uporabom istih ključnih slova (jednostavna Cezarova šifra primijenjena na nešifrirani tekst u engleskom jeziku).

Odgovarajuća skupina šifriranih znakova trebala bi imati razdiobu vjerojatnosti sličnu vrijednosti određenoj za engleski jezik. Računajući indeks IC za sve stupce potrebno je dobiti vrijednost oko 1.73. Tablica vrijednosti indeksa IC za sve pretpostavljene vrijednosti ključa tj. od 1 do 10 prikazana je u nastavku (Tablica 6).

Duljina	Indeks IC
1	1.12
2	1.19
3	1.05
4	1.17
5	1.82
6	0.99
7	1.00
8	1.05
9	1.16
10	2.07

Tablica 6 Vrijednosti indeksa IC s obzirom na duljinu ključa

Izračunata vrijednost indeksa IC pokazuje da je odgovarajuća duljina ključa 5 pa je tekst potrebno podijeliti u 5 stupaca.

QPWKA
LVRXC
QZIKG
RBPFA
EOMFL
JMSDZ
VDH…

Duljinu ključa moguće je matematički odrediti primjenom Cezarovog dešifriranja cijelog stupca za svaku od 26 mogućnosti (A-Z). Zatim je potrebno izabrati ključno slovo koje daju najveću korelaciju među vjerojatnostima dešifriranog stupca i relativnih vjerojatnosti slova za normalni tekst u engleskom jeziku. Spomenutu korelaciju moguće je prikazati kao:

$$\chi = \sum_{i=1}^{c} n_i f_i$$

gdje je n_i vjerojatnost promatranog slova u stupcu, a f_i relativna vjerojatnost slova u engleskom jeziku.

Nakon računanja određuje se ključna riječ kao „EVERY“, što se koristi za dekriptiranje te se dobije tekst („XX“ je oznaka kraja poruke):

MUST CHANGE MEETING LOCATION FROM BRIDGE TO UNDERPASS
SINCE ENEMY AGENTS ARE BELIEVED TO HAVE BEEN ASSIGNED
TO WATCH BRIDGE STOP MEETING TIME UNCHANGED XX

4.2. Primjer diferencijalne kriptoanalize

Diferencijalna kriptoanaliza obavlja se preko parova nekriptiranih tekstova (X' i X'') i pripadnih parova kriptiranih tekstova (Y' i Y''). Na primjer zadana su tri para nekriptiranih tekstova i ključ:

K = 7c34b6e98f523854

X_1'= 3b92ade058430402
X_1'' = b0461a5858430402

X_2' = c8adf738ab761c20
X_2'' = d302e329ab761c20

X_3' = ef928525e61f62f1
X_3'' = d63da834e61f62f1

Zadane nekriptirane tekstove potrebno je prvo kriptirati čime se dobiju sljedeći parovi:

a84d9040984d336b
5b1c15c8c66ab3c4

aac1c1fb612511d2
ee1a40982858724a

90955c3e9e35d6f6
b65a99409ea40565

Zatim je potrebno izračunati Ej i E*j nizove duge 6 bita i C'j niz dug 4 bita:

E_1 = 55025bca0201
E^*_1 = 2f68f80abe50
C'_1 = b6f78be2

E_2 = d55603e03ff7
E^*_2 = 75c0f42014f1
C'_2 = a3862df2

E_3 = 4a14aaaf81fd
E^*_3 = 5ac2f54f2a01
C'_3 = 7904fb6b

Izračunati podaci se koriste za računanje tri puta po osam ispitinih skupova. Znači, za svaku od tri grupe ulaznih podataka računaju se skupovi test1, test2,... test8 te napravi presjek skupova svih grupa podataka. Nakon računanja dobiju se podaci:

24 27 26 5 4 6 **51** 45
18 20 26 27 12 50 52 60 61 42
19 18 22 57 **56** 60
21 54
51 54 63 60 33 **17** 3 6 15 12
48 52 55 59 60 **63**
10 29 17 18 **44** 32 35 59
13 **28** 32 49

51 32 45 27 5 8
23 **18** 30 27 53 49 60 56 34 43
28 29 6 7 **56** 60 35 39
13 12 **21** 20 35 34 59 58
61 55 44 33 28 **17** 13 7
3 1 19 17 **63** 61
44 0
52 50 47 46 41 40 29 **28** 27 26

19 23 24 28 **51** 55
34 47 48 61 2 15 16 **18** 29 31
49 **56** 33 40
46 32 63 49 10 8 3 4 27 28 23 **21**
41 40 45 44 55 15 **17** 16 21 20
63 53 9 3
3 **44**
49 47 33 32 29 **28** 19 13

Dekadske vrijednosti 6-bitnih dijelova podključa K_3 su: 51, 18, 56, 21, 17, 63, 44 i 28. Da je postojao još jedan par nekriptiranih tekstova, četvrta skupina podataka bi sadržavala dane brojeve. Ovakve odsječke potrebno je spojiti u cjeloviti podključ što daje K_3 = cd2e1547fb1c. Zatim se pokreće algoritam za generiranje podključeva unatrag.

Dobije se ključ:

```
01111100 00110100 10x10x10 1x101000 10001110 0101xx10 001x10x0 x1010100
```

Bitovi koji ne sudjeluju u ključu K_3 označeni su znakom „x“, a svi paritetni bitovi (svaki osmi bit) postavljeni su na nule. Posljednji korak uključuje provjeru svih vrijednosti za bitove koji nedostaju preko parova nekriptirani/kriptirani tekst.

Konačni ključ je:

```
01111100 00110100 10110110 11101001 10001111 01010010 00111000 01010100
```

Znači, diferencijalnom kriptoanalizom za DES sa 3 runde dobije se ključ koji glasi: K = 7c34b6e98f523854, što je upravo vrijednost pretpostavljena na početku. Time je pokazano da se danim algoritmom problem pronalaženja ključa može svesti na pretraživanje prostora ključeva veličine 256.

4.3. Primjer linearne kriptoanalize

Rad linearne kriptoanalize prikazan je napadom na modificiranu inačicu algoritma S-DES.

Neka funkcija f prima ulazne vrijednosti *x* duge 8 bita i podključeve *k* duge 8 bita te pruža izlazne vrijednosti *y* duge 8 bita. Ovo je moguće zapisati kao:

as y = f(x, k) (mod 2).

Problem se javlja upravo ako je algoritam S-DES dizajniran tako da je funkciju f moguće zapisati kao linearnu kombinaciju vrijednosti *x* i *k* modulo 2. Znači dobije se sljedeći zapis (gdje su M i D matrice veličine 8x8):

y = f(x, k) = Mx + Dk (mod 2).

Funkcija f tada izgleda ovako:

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} M_{0,0} & M_{0,1} & M_{0,2} & M_{0,3} & M_{0,4} & M_{0,5} & M_{0,6} & M_{0,7} \\ M_{1,0} & M_{1,1} & M_{1,2} & M_{1,3} & M_{1,4} & M_{1,5} & M_{1,6} & M_{1,7} \\ M_{2,0} & M_{2,1} & M_{2,2} & M_{2,3} & M_{2,4} & M_{2,5} & M_{2,6} & M_{2,7} \\ M_{3,0} & M_{3,1} & M_{3,2} & M_{3,3} & M_{3,4} & M_{3,5} & M_{3,6} & M_{3,7} \\ M_{4,0} & M_{4,1} & M_{4,2} & M_{4,3} & M_{4,4} & M_{4,5} & M_{4,6} & M_{4,7} \\ M_{5,0} & M_{5,1} & M_{5,2} & M_{5,3} & M_{5,4} & M_{5,5} & M_{5,6} & M_{5,7} \\ M_{6,0} & M_{6,1} & M_{6,2} & M_{6,3} & M_{6,4} & M_{6,5} & M_{6,6} & M_{6,7} \\ M_{7,0} & M_{7,1} & M_{7,2} & M_{7,3} & M_{7,4} & M_{7,5} & M_{7,6} & M_{7,7} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} D_{0,0} & D_{0,1} & D_{0,2} & D_{0,3} & D_{0,4} & D_{0,5} & D_{0,6} & D_{0,7} \\ D_{1,0} & D_{1,1} & D_{1,2} & D_{1,3} & D_{1,4} & D_{1,5} & D_{1,6} & D_{1,7} \\ D_{2,0} & D_{2,1} & D_{2,2} & D_{2,3} & D_{2,4} & D_{2,5} & D_{2,6} & D_{2,7} \\ D_{3,0} & D_{3,1} & D_{3,2} & D_{3,3} & D_{3,4} & D_{3,5} & D_{3,6} & D_{3,7} \\ D_{4,0} & D_{4,1} & D_{4,2} & D_{4,3} & D_{4,4} & D_{4,5} & D_{4,6} & D_{4,7} \\ D_{5,0} & D_{5,1} & D_{5,2} & D_{5,3} & D_{5,4} & D_{5,5} & D_{5,6} & D_{5,7} \\ D_{6,0} & D_{6,1} & D_{6,2} & D_{6,3} & D_{6,4} & D_{6,5} & D_{6,6} & D_{6,7} \\ D_{7,0} & D_{7,1} & D_{7,2} & D_{7,3} & D_{7,4} & D_{7,5} & D_{7,6} & D_{7,7} \end{bmatrix} \begin{bmatrix} k_0 \\ k_1 \\ k_2 \\ k_3 \\ k_4 \\ k_5 \\ k_6 \\ k_7 \end{bmatrix} \pmod 2$$

Sve su permutacije i xor funkcije linearne jer ih je moguće zapisati kao:

P4 = y = h(x):

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \pmod 2$$

XOR4 = z = i(x, y):

$$\begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} \pmod 2$$

Funkcija SW je također permutacija, tj. linearna je te ju je moguće zapisati kao:

y = g(x) = Ex

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} \pmod 2$$

Da je S-tablica linearna pronalazak linearne funkcije sveo bi se na y = f(x, k) (mod 2).

Postoje dva korištenja funkcije *f* za svaki od podključeva S_1 i S_2 s funkcijom SW u sredini. Ako je *P* nekriptirani tekst, a *C* kriptirani tekst te se ignorira inicijalna i konačna permutacija, vrijedi:

C = f(g(f(P, K1)), K2)
=M'g(f(P, K1)) + D'K2
=E'M'f(P, K1) + D'K2
=E'M'(M'P + D'K1) + D'K2
=E'M^2'P + E'M'D'K1 + D'K2 (mod 2)

Definiranje nove konstantne matrice:
R= E´M^2, S= E´M´D, and T=D

Za nezavisne podključeve vrijedi:
C = R´P + S´K1 + T´K2 (mod 2)

Ako podključevi nisu generirani iz istog ključa vrijedi:
C = R´P + S´K1 + T´K2 + U´K3 + V´K4 (mod 2)

Svaki element u R, S, T, U i V je ili 0 ili 1 pa vrijedi:
C0 = P3 + P4 + P5 + $K1_0$ + $K1_2$ + $K1_3$ + $K1_5$ + ... + $K4_5$ + $K4_7$

Ovo pokazuje da je bit 0 kriptiranog teksta jednak xor funkciji bitova nekriptiranog teksta na mjestu 3, 4, 5 i ključa 1 na mjestu 1, 2, 3 itd. Slični izrazi dobiju se za bitove 1-7 šifriranog teksta.

Napad u kojem se koristi jedan šifrirani/nešifrirani par daje 8 jednadžbi s 32 nepoznanice čime ne možemo dobiti rješenje sustava. Za 4 šifrirana/nešifrirana para tekstova dobiju se 32 jednadžbe s 32 nepoznanice što je, uz uporabu Gaussove eliminacije, moguće riješiti kroz 32 izračuna. Na žalost, S-DES koristi nelinearne S-tablice što zahtjeva mnogo složenije izračune. Ipak, ovo ne znači da se linearna jednadžba funkcije *f* neće održati za neke parove ulaza i izlaza (u savršenom slučaju vjerojatnost bi bila 50 %).

4.4. Primjer „birthday“ napada

Jedan od primjera uporabe „birthday“ napada je Pollard-ov „ro algoritam“ za logaritme. Riječ je o činjenici da funkcija f : S -> S koja preslikava vrijednosti iz konačnog skupa *S* u taj isti skup mora u nekoj točki kreirati periodičnu sekvencu. Dovoljno dugo ponavljanje sekvence hi = fi(fi-1(...(f1(x)))) učinit će je periodičnom što znači da postoje brojevi i > j gdje je h_i=h_j.

Kako postupak napreduje i kreira se periodična sekvenca, dobiva se oblik grčkog slova “ro” po čemu je ova metoda i dobila ime. Kako bi se metoda koristila za traženje kolizije, potrebne su dvije stvari:

1. **algoritam za detekciju ciklusa**, za što postoji nekoliko mogućnosti (npr. Floyd-ov algoritam ili Brent-ov algoritam) i
2. **način određivanja predperioda**, što je sekvenca od x1 do x2.

Tada je moguće početi tražiti koliziju tako da se odabere slučajna početna vrijednost *x1* i počinje računati sekvenca uz izvođenje algoritma za detekciju ciklusa. Pohranjuju se samo one vrijednosti potrebne za određivanje posljednje vrijednosti koja uzrokuje koliziju. Budući da se točka kolizije može preskočiti, važno je sačuvati dovoljan broj vrijednosti da bi se spriječilo ponovno računanje sekvence. Trivijalan način za to je da se sekvenca započne ponovno dok se ne nađe točka kolizije.

Primjer računanja diskretnih logaritama moguć je preko sljedećeg programskog odsječka pisanog u programskom jeziku C++.

```
#include <stdio.h>
  const int n = 1018, N = n + 1;  /* N = 1019 - primarni broj    */
 const int alpha = 2;            /* generator                   */
 const int beta = 5;             /* 2^{10} = 1024 = 5 (N) */

 void new_xab( int& x, int& a, int& b ) {
   switch( x%3 ) {
   case 0: x = x*x     % N;  a =  a*2  % n;  b =  b*2  % n;  break;
   case 1: x = x*alpha % N;  a = (a+1) % n;                  break;
   case 2: x = x*beta  % N;                  b = (b+1) % n;  break;
   }
 }

 int main() {
   int x=1, a=0, b=0;
   int X=x, A=a, B=b;
   int i;
   for( i = 1; i < n; ++i ) {
     new_xab( x, a, b );
     new_xab( X, A, B ); new_xab( X, A, B );
     printf( "%3d  %4d %3d %3d  %4d %3d %3d\n", i, x, a, b, X, A, B );
     if( x == X ) break;
   }
   return 0;
}
```

Rezultat programa za računanje grupe generirane s modulom 2 uz primarni broj N=1019 je:

```
i     x   a   b     X   A   B
-----------------------------
 1     2   1   0    10   1   1
 2    10   1   1   100   2   2
 3    20   2   1  1000   3   3
 4   100   2   2   425   8   6
 5   200   3   2   436  16  14
 6  1000   3   3   284  17  15
 7   981   4   3   986  17  17
 8   425   8   6   194  17  19
.............................
48   224 680 376    86 299 412
49   101 680 377   860 300 413
50   505 680 378   101 300 415
51  1010 681 378  1010 301 416
```

5. Zaključak

Razvojem tehnologije i računarske znanosti došlo je do poboljšavanja kriptografskih metoda kako bi se povećala sigurnost sustava. Međutim, isti taj napredak doveo je do usavršavanja metoda otkrivanja nedostataka u kriptografskim shemama. Tijekom povijesti, metode su postajale sve sofisticiranije i složenije, a bivale su usmjerene prema novim kriptografskim algoritmima. Veliki broj algoritama probijen je ubrzo nakon predstavljanja.

Metode kriptografije pružile su sigurnost i tajnost informacijama koji su često predstavljale ključne podatke tijekom povijesti. Njihovo uvođenje i uporaba osigurali su nesmetanu razmjenu i pohranu podataka na velike udaljenosti jer je dešifriranje takvih podataka bilo moguće samo uz poznavanje tajnih ključeva. Ipak razvijeni su postupci koji su, zahvaljujući određenim nedostacima u kriptografskim shemama, omogućili otkrivanje tajnih ključeva za dekriptiranje. Spomenuti postupci čine metode kriptoanalize koja je usmjerena k 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 usmjerene prema sustavima koji još nisu probijeni, moguće je očekivati više napada na AES standard. Isto tako, moguće je da će povećanje snage i memorijskog kapaciteta računala u budućnosti pružiti dovoljne resurse za provođenje „brute force“ napada na većinu algoritama. Time bi se osigurao proboj gotovo svih šifri te dovelo do potrebe stvaranja novih koje će zahtjevati još veće memorijske i vremenske resurse prilikom provođenja takvih napada.

6. Reference

[1] Kriptoanaliza, http://en.wikipedia.org/wiki/Cryptanalysis, rujan, 2009.

[2] Analiza učestalosti, http://en.wikipedia.org/wiki/Frequency_analysis, rujan, 2009.

[3] Računanje podudaranja, http://en.wikipedia.org/wiki/Index_of_coincidence, rujan 2009.

[4] Kasiski napad, http://en.wikipedia.org/wiki/Kasiski_examination, rujan, 2009.

[5] Diferencijalna kriptoanaliza, http://en.wikipedia.org/wiki/Differential_cryptanalysis, rujan, 2009.

[6] Howard M. Heys: A Tutorial on Linear and Differential Cryptanalysis, http://www.engr.mun.ca/~howard/PAPERS/ldc_tutorial.pdf, rujan, 2009.

[7] Bumerang napad, http://en.wikipedia.org/wiki/Boomerang_attack, rujan, 2009.

[8] Napad „brute force“, http://en.wikipedia.org/wiki/Brute_force_attack, rujan, 2009.

[9] Daviesov napad, http://en.wikipedia.org/wiki/Davies%27_attack, rujan, 2009.

[10] Integralna kriptoanaliza, http://en.wikipedia.org/wiki/Integral_cryptanalysis, rujan, 2009.

[11] Linearna kriptoanaliza, http://en.wikipedia.org/wiki/Linear_cryptanalysis, rujan, 2009.

[12] Linearna kriptoanaliza, http://nsfsecurity.pr.erau.edu/crypto/lincrypt.html, rujan, 2009.

[13] Meet-in-the-midle napad, http://en.wikipedia.org/wiki/Meet-in-the-middle_attack, rujan, 2009.

[14] Mod n kriptoananaliza, http://en.wikipedia.org/wiki/Mod-n_cryptanalysis, rujan, 2009.

[15] Napad povezanim ključevima, http://en.wikipedia.org/wiki/Related-key_attack, rujan, 2009.

[16] Napad klizanjem, http://en.wikipedia.org/wiki/Slide_attack, rujan, 2009.

[17] XSL napad, http://en.wikipedia.org/wiki/XSL_attack, rujan, 2009.

[18] Birthday napad, http://en.wikipedia.org/wiki/Birthday_attack, rujan, 2009.

[19] Pollard-ov ro algoritam, http://en.wikipedia.org/wiki/Pollard%27s_rho_algorithm_for_logarithms, rujan, 2009.

[20] Analiza energije, http://en.wikipedia.org/wiki/Power_analysis, rujan, 2009.

[21] Vremenski napad, http://en.wikipedia.org/wiki/Timing_attack, rujan, 2009.

[22] Daniel J. Bernstein: Cache-timing attacks on AES, http://cr.yp.to/antiforgery/cachetiming-20050414.pdf, rujan, 2009.

[23] MITM napad, http://en.wikipedia.org/wiki/Man-in-the-middle_attack, rujan, 2009.

[24] Napadi ponavljanjem, http://en.wikipedia.org/wiki/Replay_attack, rujan 2009.

[25] Black-bag kriptoanaliza, http://en.wikipedia.org/wiki/Black-bag_cryptanalysis, rujan, 2009.

[26] Rubber-hose napad, http://en.wikipedia.org/wiki/Rubber-hose_cryptanalysis, rujan, 2009.