ELEKTROTEHNIČKI FAKULTET UNIVERZITETA U BEOGRADU

PREGLED KRIPTOGRAFSKIH NAPADA

– Diplomski rad–

Kandidat:

Ana Ivanović 2009/262

Mentor:

doc. dr Zoran Čiča

Beograd, Septembar 2015.

SADRŽAJ

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 [1].

Cilj ovog rada jeste objašnjenje šifri, sa akcentom na moderne šifre, klasifikovanje kriptografskih napada i izlaganje osnovnih termina neophodnih za razumijevanje istih.

U drugom poglavlju je data podjela šifri i detaljnija objašnjenja odabranih šifri, DES, AES i RSA.

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

U četvrtom poglavlju bavimo se trima prethodno opisanim šiframa, DES, AES i RSA, prikazani su neki od napada na ove šifre i date su procjene o njihovoj sigurnosti i budućem korišćenju u savremenim tehnologijama.

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

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[4]. Danski kriptograf *Auguste Kerckhoffs* 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. *Claude Shannon* 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 supstitucije (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 supstituciju koristi više alfabeta, i monoalfabetske, kod kojih se supstitucija 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 i asimetrične.

- 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. Dva su 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. a*symmetric key cipher*) – 1976. godine ovaj tip šifre su uveli *Whitfield Diffie*, *Martin Hellman* i *Ralph Merkle*. Šifrovanje se vrši pomoću javnog ključa, a dešifrovanje pomoću privatnog tj. tajnog ključa [2]. Asimetrično šifrovanje se tipično

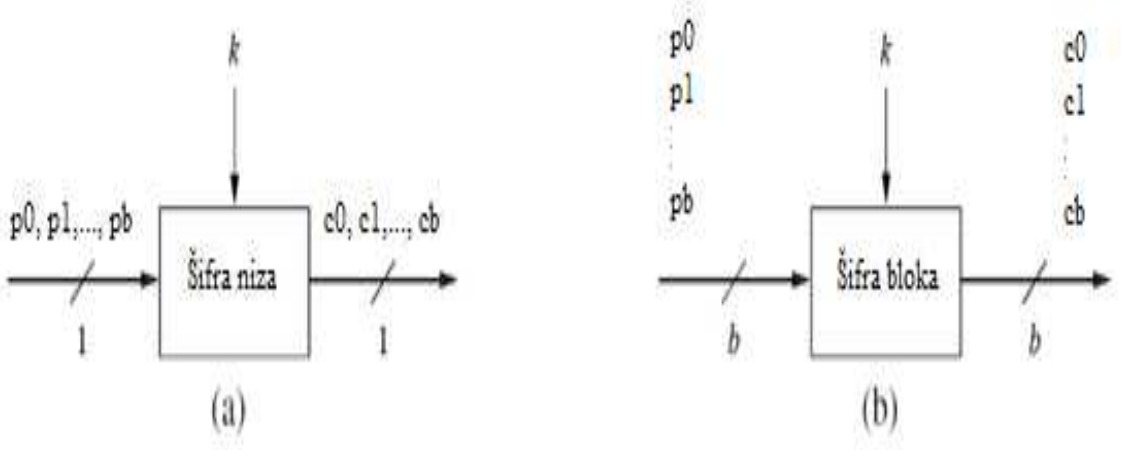
koristi kod autentifikacije i digitalnih potpisa (pružaju dokaz autentičnosti i porijekla pravnih dokumenata u elektronskoj formi), kao i za razmjenu ključeva (engl. *key establishment*) simetričnih šifri [5].

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 razmjenjuju 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 2.1.1 je predstavljena operativna razlika izmedju šifre niza (slika 2.1.1a) i šifre bloka (slika 2.1.1b), kada se vrši šifrovanje *b* bita, gdje je *b* širina blok šifre (odnosno veličina bloka).



Slika 2.1.1. 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_Ki D_K.

Šifrovanje se vrši prema formuli:

$$C_i = E_{K_I}(P_i) \equiv P_i + K_i \mathrm{mod} 2 \tag{2.1.2}$$

Formula dešifrovanja:

$$P_I = D_{K_i}(C_i) \equiv C_I + K_i \mathrm{mod} 2 \tag{2.1.3}$$

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.

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 umnožak 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 umnožak dužine bloka.

ECB je najjednostavniji režim šifrovanja. Otvoreni tekst se izdijeli na blokove čija je dužina ista kao dužina bloka šifre koju koristimo (na primjer AES ili 3DES), 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 P_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.

Princip rada šifri bloka će biti objašnjen na primjeru dvije značajne blok šifre DES i AES.

i) DES

DES (engl. *Data Encryption Standard*) je simetrična blok šifra kod koje se šifrovanje vrši na nivou 64-bitnog bloka. Pripada grupi *Feistel* šifara. *Feistel* šifre su simetrične strukture koje se koriste kao gradivne jedinice blok šifri. Karakteriše ih sličnost između operacija šifrovanja i dešifrovanja, čak su u nekim slučajevima te operacije identične s tim da se razlikuje proces generisanja ključeva. *Feistel* se može definisati i kao iterativna šifra sa internom funkcijom **F** koja se naziva funkcija runde.

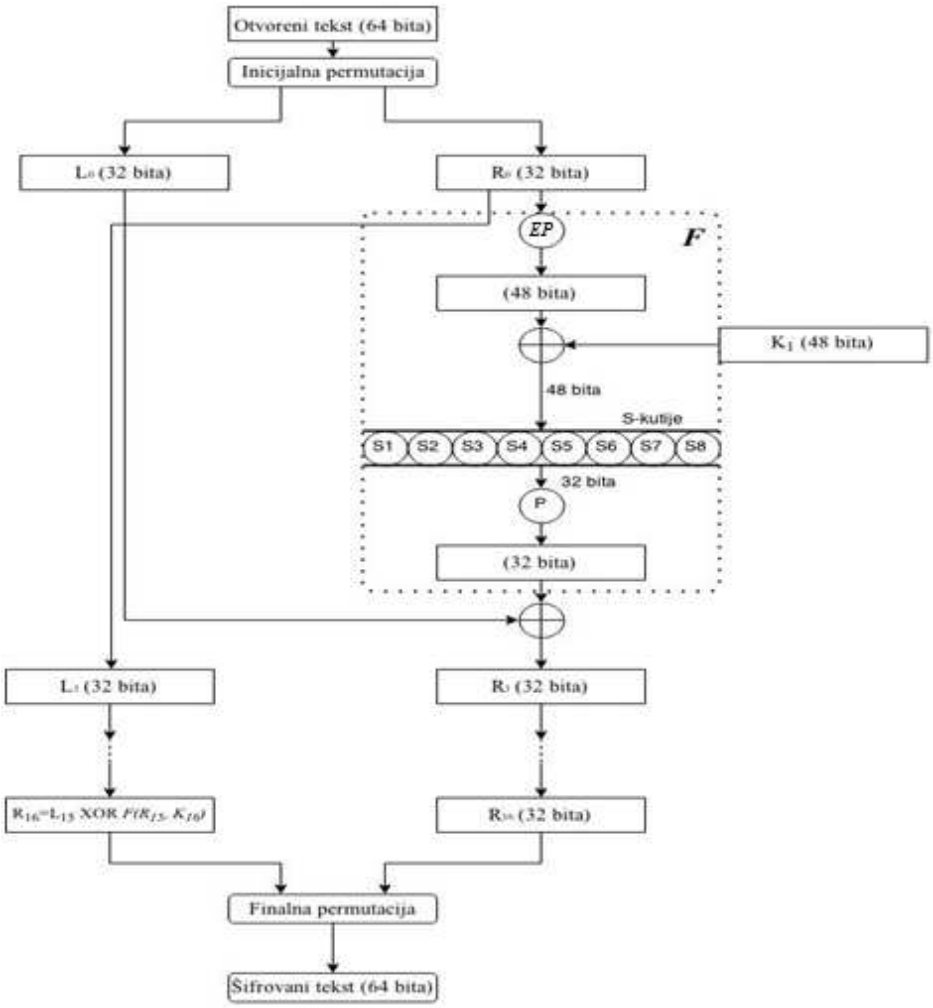
Ulaz algoritma DES je 64-bitni blok otvorenog teksta, izlaz je, takođe 64-bitni, blok šifrovanog teksta. Ključ je dužine 56 bita (uobičajeno se predstavlja sa 64 bita pri čemu je svaki osmi bit bit parnosti i zanemaruje se).

U osnovi, DES je kombinacija dvije osnovne tehnike šifrovanja: difuzije i konfuzije. DES se sastoji od 16 rundi: iste operacije se primjenjuju na jedan blok otvorenog teksta šesnaest puta. Na slici 2.1.4. je dat prikaz blok sheme jedne runde šifrovanja.

Inicijalna permutacija se obavlja prije prve runde, a njoj inverzna finalna permutacija predstavlja poslednji korak šifrovanja. Inicijalna i finalna permutacija, su permutacije na nivou bita. Predstavljene su u tabeli 2.1.5 i ne utiču na sigurnost *DES,* tj. nemaju kriptografski značaj, i ne zna se tačan razlog koji stoji iza uvođenja ove dvije operacije. Tabela inicijalne permutacije se čita sa lijeva na desno od vrha ka dnu. Iz tabele možemo vidjeti da se ulazni bit 58 mapira na izlaznu poziciju 1, ulazni bit 50 na poziciju 2 i tako dalje. Tabela finalne permutacije se čita na isti način.

Blok otvorenog teksta, nakon inicijalne permutacije, dijeli se na lijevu i desnu 32-bitnu polovinu, *L* i *R*. Desna polovina *R* proširuje se sa 32 na 48 bita tako što se vrši *expansion permutation*, koja permutuje bite i ujedno vrši udvajanje pojedinih bita kako bi se postiglo proširenje. Ova operacija, na slici 2.1.4 označena sa *EP*, ulaznih 32 bita izdijeli na osam 4-bitnih blokova i svaki od blokova proširi na 6-bitne blokove. U tabeli 2.1.6 data je tablica proširenja *EP*. Iz tabele se vidi da se 16 od 32 ulazna bita pojavljuju dva puta u 48-bitnom izlazu, s tim da se jedan ulazni bit nikad ne pojavljuje dva puta u istom izlaznom 6-bitnom bloku. *EP* povećava difuziju tako što neki ulazni biti utiču na dvije različite izlazne lokacije i čini desnu stranu *R* iste dužine kao ključ K_i kako bi u sledećem koraku bila moguća XOR operacija. Glavna kriptografska svrha ove permutacije jeste to što, dozvoljavanjem da jedan bit utiče na dvije supstitucije, zavisnost izlaznih bita od ulaznih se povećava i javlja se efekat lavine (engl. *Avalanche effect*[1]). DES je dizajniran tako da je svaki bit šifrovanog teksta funkcija svih bita otvorenog teksta i svih bita ključa.

[1] Dozvoljavanjem da jedan bit utiče na dvije substitucije zavisnost izlaznih od ulaznih bita brzo raste. Nakon pet rundi DES svaki bit šifrovanog teksta je funkcija svakog bita ključa i otvorenog teksta. Nakon osam rundi šifrovani tekst je proizvoljna funkcija svakog bita ključa i otvorenog teksta.



Slika 2.1.4. Blok shema jedne runde DES

Inicijalna permutacija								Krajnja permutacija							
58	50	42	34	26	18	10	2	40	8	48	16	56	24	64	32
60	52	44	36	28	20	12	4	39	7	47	15	55	23	63	31
62	54	46	38	30	22	14	6	38	6	46	14	54	22	62	30
64	56	48	40	32	24	16	8	37	5	45	13	53	21	61	29
57	49	41	33	25	17	9	1	36	4	44	12	52	20	60	28
59	51	43	35	27	19	11	3	35	3	43	11	51	19	59	27
61	53	45	37	29	21	13	5	34	2	42	10	50	18	58	26
63	55	47	39	31	23	15	7	33	1	41	9	49	17	57	25

Tabela 2.1.5 Inicijalna i Finalna permutacija DES

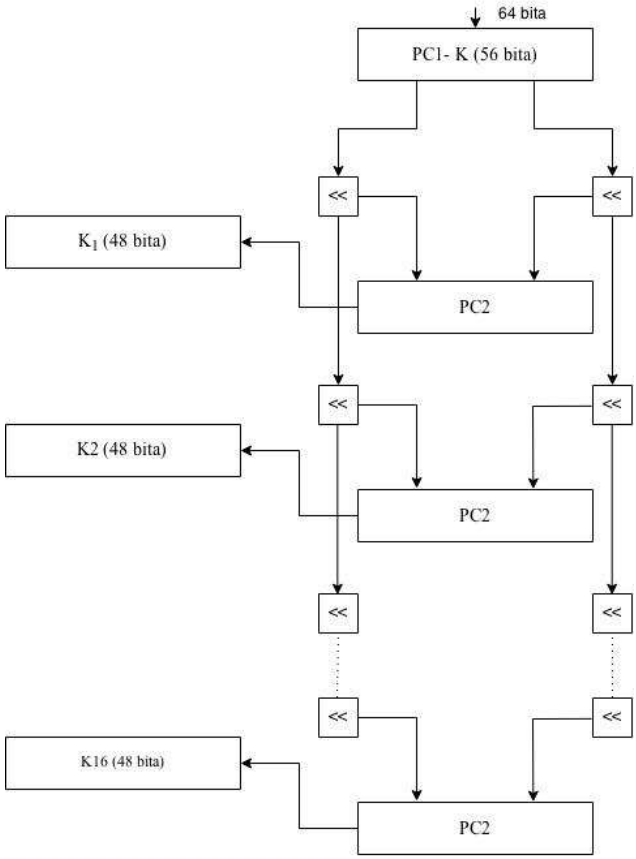
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Tabela 2.1.6 Expansion Permutation

Ključ dužine 64 bita se svodi na 56-bitni ključ *K*, tako što se odbacuju biti parnosti - svaki osmi bit se zanemaruje. Ova operacija se naziva *permuted choice 1*. Iz ključa *K* se generiše 48-bitni podključ K_i, koji će se koristiti za šifrovanje u i-toj rundi. Podključevi K_i generišu se na sljedeći način: ključ *K* se dijeli na dvije 28-bitne polovine koje se rotiraju ulijevo za *n* bita u zavisnosti od runde, što je prikazano u tabeli 2.1.7. Nakon pomjeranja, selektuje se 48 od 56 bita (iz svake polovine se uzima po 24 bita), i dobija se podključ K_i. Pošto se ovom operacijom vrši i permutacija i kompresija bita ključa *K* (56 bita) to se ona naziva komprimovana permutacija (engl. *Compression permutation*) ili *permuted choice 2*. Svaki bit ključa K se iskoristi u približno 14-16 podključeva K_i, iako nisu svi biti iskorišćeni isti broj puta. Algoritam kojim se vrši generisanje podključeva K_i naziva se *key schedule* i prikazan je na slici 2.1.8.

runda	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
n	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Tabela 2.1.7 Rotiranje bita



Slika 2.1.8 Blok shema generisanja podključeva K_i

Sledeći korak podrazumijeva kombinovanje proširene desne polovine *R* sa ključem K_i operacijom XOR (bitsko ekskluzivno ili) i rezultujući 48-bitni niz se šalje na ulaz bloka sa *S*-kutijama.

Postoji osam S-kutija (engl. *Substitution box*) od kojih svaka ima 6-bitni ulaz i 4-bitni izlaz, kao što je prikazano na slici 2.1.9. Ulaznih 48 bita se dijeli u osam 6-bitnih podblokova. Svaka S-kutija se može predstaviti pomoću tabele koja ima 4 reda i 16 kolona, ulazni biti nose informaciju pod kojim rednim brojem reda i kolone tražimo izlaz. Način čitanja tabela S-kutija je sledeći: prvi i poslednji bit svakog 6-bitnog ulaza određuju red tabele, a četiri unutrašnja bita određuju kolonu. Rezultat je osam 4-bitnih podblokova koji se kombinuju u jedan 32-bitni blok.

S_1

	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x1100x	x1101x	x1110x	x1111x
0yyyy0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0yyyy1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
1yyyy0	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
1yyyy1	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_2

	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x1100x	x1101x	x1110x	x1111x
0yyyy0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
0yyyy1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
1yyyy0	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
1yyyy1	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3

	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x1100x	x1101x	x1110x	x1111x
0yyyy0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
0yyyy1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
1yyyy0	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1yyyy1	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4

	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x1100x	x1101x	x1110x	x1111x
0yyyy0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
0yyyy1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
1yyyy0	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
1yyyy1	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S_5

	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x1100x	x1101x	x1110x	x1111x
0yyyy0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
0yyyy1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
1yyyy0	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
1yyyy1	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6

	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x1100x	x1101x	x1110x	x1111x
0yyyy0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
0yyyy1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
1yyyy0	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
1yyyy1	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S_7

	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x1100x	x1101x	x1110x	x1111x
0yyyy0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
0yyyy1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1yyyy0	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
1yyyy1	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8

	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x1100x	x1101x	x1110x	x1111x
0yyyy0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
0yyyy1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
1yyyy0	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
1yyyy1	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Slika 2.1.9 S-kutije

S-kutije su srž DES šifre u smislu kriptografske snage, jedini su nelinearni elementi u algoritmu i obezbjeđuju konfuziju. S-kutije su dizajnirane prema sledeća četiri kriterijuma:

1) Svaka S-kutija ima šest ulaznih i četiri izlazna bita.
2) Nijedan od izlaznih bita ne smije biti blizak linearnoj kombinaciji ulaznih bita.

3) Ako su prvi i poslednji ulazni bit fiksirani a četiri središnja bita se mijenjaju, onda se sve četiri moguće izlazne vrijednosti mogu desiti tačno jednom.
4) Ako se dva ulaza S-kutije razlikuju u tačno jednom bitu onda se njima odgovarajući izlazi moraju razlikovati u najmanje dva bita.
5) Ako se dva ulaza S-kutije razlikuju u dva središnja bita onda se njima odgovarajući izlazi moraju razlikovati u najmanje dva bita.
6) Ako se dva ulaza u S-kutiju razlikuju u prva dva bita a imaju identična poslednja dva bita onda njima odgovarajući izlazi moraju biti različiti.

Nakon S-kutija 32-bitni rezultat se permutuje tako što svakom bitu dodjeljujemo novu poziciju i svi biti se iskoriste samo jednom. Permutacija *P* obezbjeđuje difuziju tako što četiri izlazna bita svake S-kutije permutuje na takav način da će u sledećoj rundi prolaziti kroz druge S-kutije. Difuzija uzrokovana S-kutijama i permutacijom *P* garantuje da je svaki bit, do kraja pete runde šifrovanja, funkcija svakog bita ključa i svakog bita otvorenog teksta, odnosno javlja se efekat lavine.

Na kraju se rezultat permutacije kombinuje operacijom XOR sa L polovinom, nakon čega lijeva i nova desna polovina mijenjaju mjesta. Ovaj postupak se ponavlja 16 puta, pri čemu u posljednjoj rundi lijeva i desna polovina ne mijenjaju mjesta već se finalna permutacija, koja je inverzna inicijalnoj, primjenjuje na konkatanaciji $L_{16}R_{16}$.

Funkcija dešifrovanja je ista kao funkcija šifrovanja. Osnovna ideja je da se dešifrovanjem vrši obrnut proces šifrovanja - u prvoj rundi dešifrovanja se vrši šifrovanje 16. runde šifrovanja. DES ima osobinu komplementarnosti [7]:

$$DES_{\bar{\mathrm{K}}}(\bar{P}) = \overline{DES_{\mathrm{K}}(P)} \tag{2.1.10}$$

Varijacija DES šifre je 3DES koja se sastoji od tri uzastopna DES šifrovanja:

$$C = DES_{K_3}(DES_{K_2}(DES_{K_1}(P))) \tag{2.1.11}$$

ii) AES

AES (engl. *Advanced Encryption Standard*) je najrasprostranjenija simetrična blok šifra današnjice. Razvili su je belgijski kriptografi *Joan Daemen* i *Vincent Rijmen* i nazvali je *Rijndael*. Ime je promijenjeno u AES nakon standardizacije u Americi 2001. godine. Dio je nekoliko industrijskih standarda i koristi se u mnogim komercijalnim sistemima kao što su IPsec (*Internet security standard*), TLS (*Transport Layer Security*), Wi-Fi enkripcioni standard IEEE 802.11i, SSH (*Secure Sockets Layer*), Skype... Do danas sem *brute-force* napada, koji će biti objašnjen u nastavku rada, nije bilo uspješnih napada na AES [13].

Za razliku od DES šifre kod koje se u jednoj rundi šifruje samo pola bloka, 32 bita, u slučaju AES šifre u jednoj iteraciji se vrši šifrovanje svih 128 bita. Tri su dužine ključeva i od njih zavisi broj rundi šifrovanja, što je prikazano u tabeli 2.1.12.

Dužina ključa [b]	128	192	256
Broj rundi	10	12	14

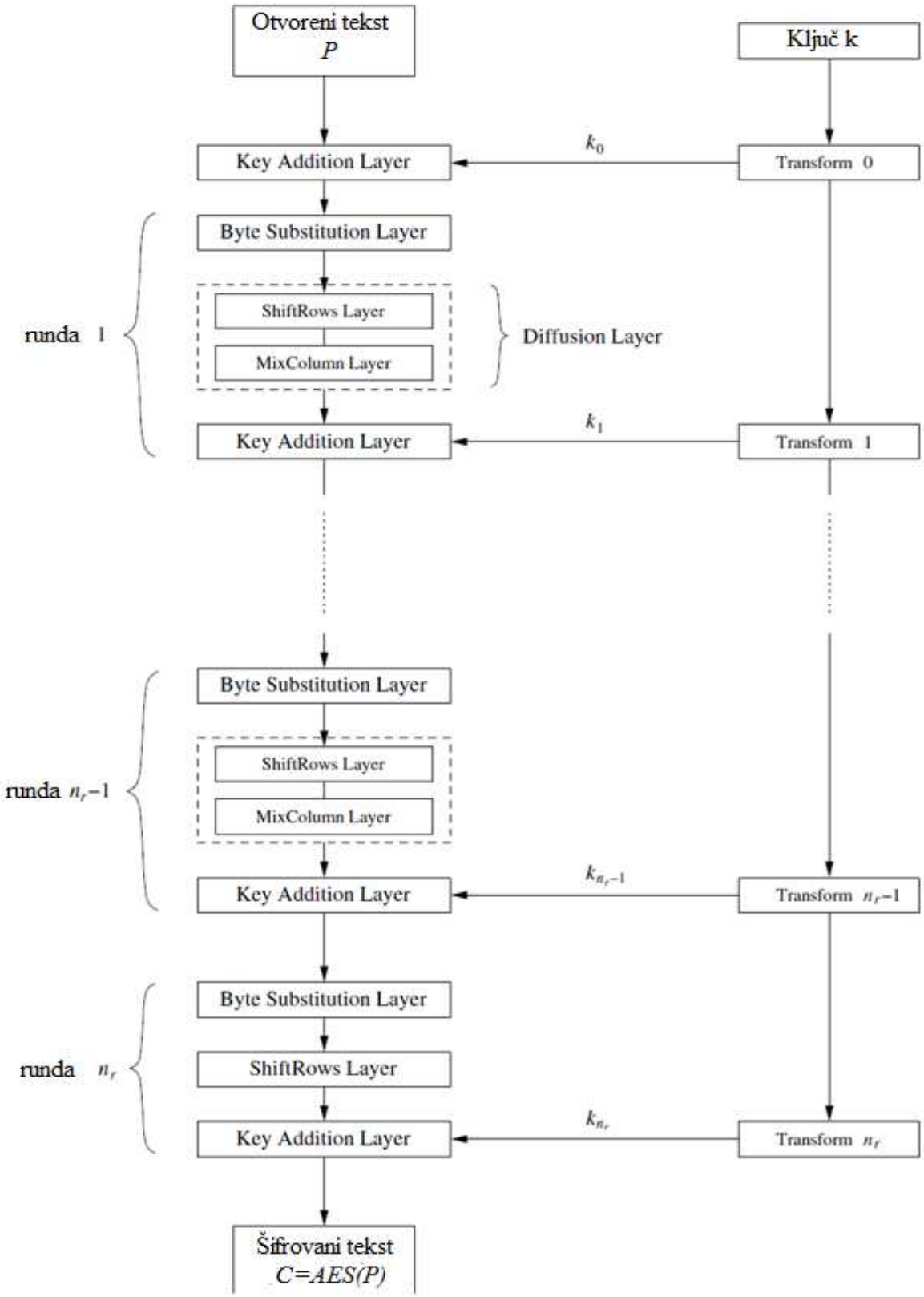
Tabela 2.1.12 Broj rundi u zavisnosti od dužine ključa

AES se sastoji od tri sloja. Svaki sloj obrađuje svih 128 bita. Svaka runda izuzev prve se sastoji od sva tri sloja, kao što se može vidjeti na slici 2.1.13. Otvoreni tekst je označen sa *P*, šifrovani sa *C*, broj rundi sa n_r, i podključevi sa k_i. Slojevi su sledeći:

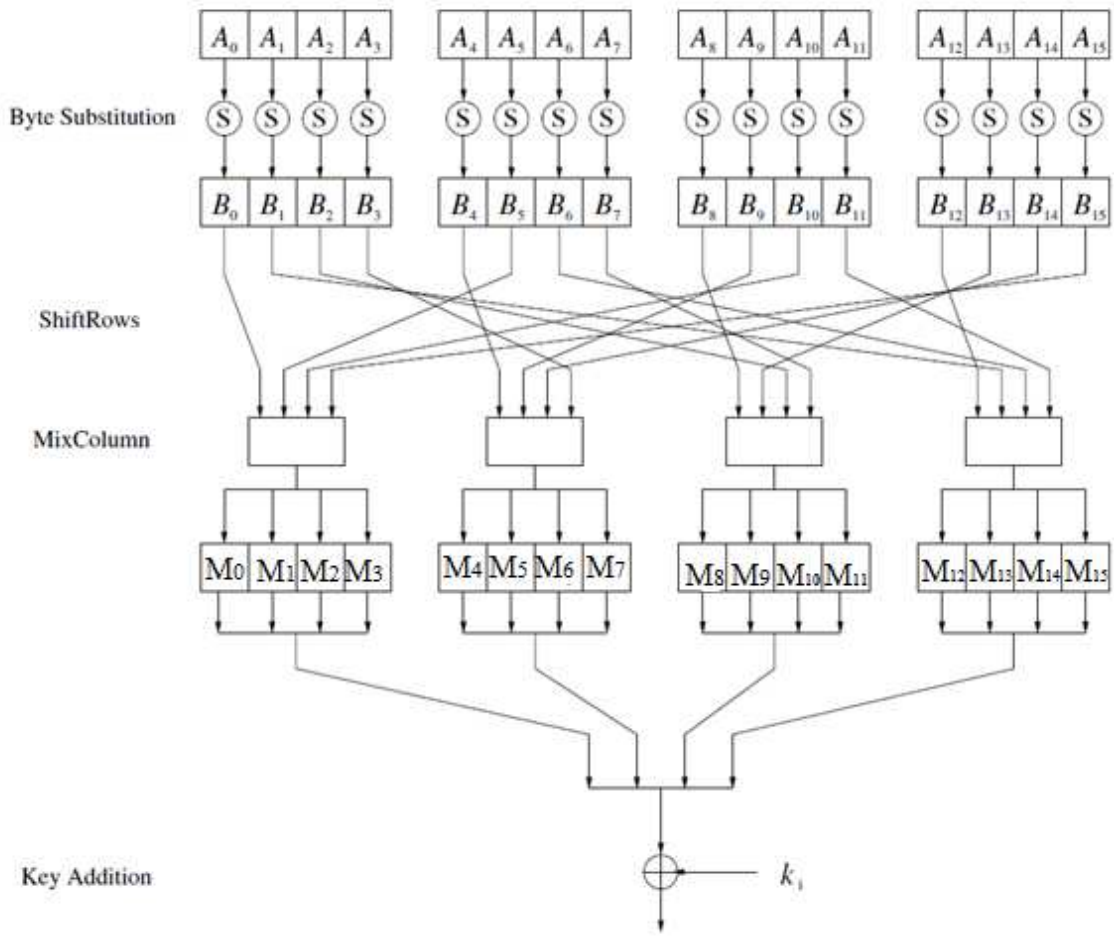
- *Key Addition* sloj: Podključ k_i se operacijom XOR dodaje na stanje. Prije prve runde se k_0 dodaje na otvoreni tekst *P*, u prvoj se k_1 dodaje na rezultat XOR operacije *P* i k_0 , itd.
- *Byte Substitution* (S-kutija) sloj: Svaki element se nelinearno transformiše korišćenjem lukap tabela sa posebnim matematičkim osobinama. Na ovaj način se postiže konfuzija.
- *Diffusion* sloj: Obezbjeđuje difuziju. Sastoji se od dva podsloja koji obavljaju linearne operacije:
  - *ShiftRows*: permutuje podatke na nivou bajtova
  - *MixColumn*: Matrica koja kombinuje (miješa) blokove od po četiri bajta.

Slično kao kod DES šifre, podključevi se generišu iz originalnog ključa.

Radi lakšeg razumijevanja toka podataka kroz AES šifru detaljni prikaz jedne runde je prikazan na slici 2.1.14.



Slika 2.1.13 Blok shema n_r rundi AES



2.1.14 Blok shema jedne runde AES

Neka je 16-bajtni ulaz i-te runde niz $A=A_0, A_1, ..., A_{15}$ koji ćemo napisati u vidu 4×4 matrice kao što je prikazano tabelom 2.1.15

A0	A4	A8	A12
A1	A5	A9	A13
A2	A6	A10	A14
A3	A7	A11	A15

Tabela 2.1.15 Matrica 16-bajtnog ulaza u jednu rundu AES

U rundi se operacije vrše nad elementima, kolonama ili redovima navedene matrice. Na sličan način ključ *K* možemo predstaviti preko matrice 4×4 ako se koristi 128-bitni ključ, 6×6 za 192-bitni ključ ili 8×8 za 256-bitni ključ [13].

Sa slike 2.1.13 se vidi da je prvi sloj svake runde *Byte Substitution* sloj. Ovaj sloj se sastoji od šesnaest paralelnih S-kutija od kojih svaka ima osam ulaznih i osam izlaznih bita. Sve S-kutije su identične, za razliku od S-kutija u DES šifri gdje je svih osam S-kutija bilo jedinstveno. Na *Byte Substitution* sloju se svaki bajt A_i mijenja nekim drugim bajtom B_i. S-kutije su jedini nelinearni elementi u AES šifri. Supstitucija koja se vrši u S-kutijama je bijektivno preslikavanje, svaki od $2^8 = 256$ mogućih ulaznih elemenata se slika u tačno određene izlazne elemente. Lukap tabela S-kutija, za ulazni element (xy) je data na slici 2.1.16

									y							
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
x 8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Slika 2.1.16 Lukap tabela S-kutije

S-kutije su projektovane tako da ne postoju ulazna vrijednost A_i takva da se prolaskom kroz S-kutiju slika u sebe samu, tj. važi $S(A_i) \neq A_i$, $\forall i$. Matricom B označavamo 16-bajtni izlaz *Byte Substitution* sloja.

Sledeći je podsloj *ShiftRows* koji pripada *Diffusion* sloju. *ShiftRows* rotira drugi red matrice B za tri bajta udesno, treći red za dva bajta udesno i četvrti red za jedan bajt udesno. Prvi red matrice B se ne mijenja. Tabelom 2.1.17 su predstavljene matrice prije i poslije prolaska kroz *ShiftRows* podsloj, B i B'.

B				B’			
B0	B4	B8	B12	B0	B4	B8	B12
B1	B5	B9	B13	B5	B9	B13	B1
B2	B6	B10	B14	B10	B14	B2	B6
B3	B7	B11	B15	B15	B3	B7	B11

Tabela 2.1.17 Ulazna B i izlazna B’ matrica ShiftRows podsloja

Na *MixColumn* podsloju se vrši linearna transformacija kojom se miješaju kolone matrice B’. Kombinacija operacija koje se vrše na *ShiftRows* i *MixColumn* podslojevima čini mogućom da nakon samo tri runde šifrovanja svaki bajt rezultujućeg šifrovanog teksta zavisi od svih bajtova otvorenog teksta. *MixColumn* transformacija, čiji je rezultat matrica *M*, se postiže tako što se svaka kolona matrice *B’* posmatra kao vektor i množi se fiksnom 4×4 matricom. Prva kolona matrice *M* se dobija po sledećoj formuli :

$$\begin{bmatrix} M_0 \\ M_1 \\ M_2 \\ M_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} B_0 \\ B_5 \\ B_{10} \\ B_{15} \end{bmatrix} \tag{2.1.18}$$

Preostale tri kolone matrice *M* se računaju analogno prvoj, množenjem odgovarajućih kolona matrice *B’* sa 4×4 matricom koja je data u jednačini 2.1.18.

Matrica *M* se XOR operacijom dodaje 16-bajtnom podključu na *Key Addition* sloju. Podključevi koji se koriste u rundama se dobijaju iz originalnog ključa *K* koji može biti dužine 128, 192 ili 256 bita. Broj podključeva je za jedan veći od broja rundi, što se vidi na slici 2.1.13. U prvoj rundi se direktno na otvoreni tekst XOR operacijom dodaje podključ k_0. Proces kada se i na početku i na kraju šifre vrši XOR dodavanje podključa naziva se *key whitening*, i koristi se u cilju povećanja sigurnosti iterativne blok šifre.

2.2. Asimetrične šifre

Diffie, *Hellman* i *Mekle* 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_{pub}, koji je dostupan svima na uvid, a dešifrovanje se vrši tajnim ključem k_{pr}, koji je poznat samo ovlašćenim korisnicima.

Asimetrična kriptografija ima brojne primjene:

- Razmjena ključeva: U protokolima za razmjenu ključeva preko nesigurnog kanala. Primjer takvih protokola su Diffie-Hellman razmjena ključeva i RSA protokol razmjene ključeva. Jedna od upotreba asimetričnih šifri jeste razmjena ključeva simetričnih šifri. Ključ *K*, neke simetrične šifre, se šifruje javnim ključem k_{pub} na predajnoj strani, a na prijemnoj strani se dešifruje tajnim ključem k_{pr}, nakon čega prijemna i predajna strana komunikaciju šifruju korišćenjem simetrične šifre i ključa *K*.

- Nemogućnost izbjegavanja odgovornosti (engl. *Nonrepudiation)*: Za pružanje dokaza integriteta i porijekla poruke se koriste digitalni potpisi. Primjeri tih algoritama su RSA, DSA ili ECDSA.
- Identifikacija: Možemo identifikovati entitete korišćenjem *challenge-and-response* protokola zajedno sa digitalnim potpisima, npr. u aplikacijama kao što su smart kartice u bankarstvu ili u mobilnim telefonima.
- Šifrovanje: Možemo šifrovati poruke korišćenjem RSA ili ElGamal šifre.

Treba uočiti da se šifrovanje i identifikacija mogu postići i korišćenjem simetričnih šifri. Velika mana asimetričnih šifri jeste to što je takvo šifrovanje podataka veoma sporo, većina šifri niza i bloka mogu vršiti šifrovanje od sto do hiljadu puta brže od asimetričnih šifri. Stoga se asimetrične šifre rijetko koriste za šifrovanje velike količine podataka. Većina protokola koji su danas u upotrebi su hibridni protokoli, koji objedinjuju simetrične i asimetrične šifre: simetrične šifre koriste za šifrovanje podataka, a asimetrične za razmjenu ključeva i garanciju integriteta i porijekla podataka (*Nonrepudiation*). Primjer hibridnog protokola je SSL/TLS koji se koristi za sigurnu Web konekciju, kao i IPsec.

Problem koji se javlja kod asimetričnih šifri jeste autentičnost javnih ključeva, odnosno da li sa sigurnošću možemo da znamo da dati javni ključ pripada datoj osobi. Taj problem se u praksi rešava upotrebom sertifikata. Sertifikat vezuje jedan javni ključ za jedan identitet. Drugi problem jeste velika dužina javnih ključeva koja uzrokuje sporo izvršavanje kodova.

Postoje tri familije asimetričnih šifri koje su od praktičnog značaja, od kojih se svaka zasniva na jednoj oblasti matematike:

- Rastavljanje cijelih brojeva na proste činioce: Nekoliko asimetričnih šifri se zasniva na činjenici da je teško velike cijele brojeve rastaviti na proste činioce. Predstavnik takvih šifri je RSA.
- Diskretni logaritam: Asimetrične šifre koje se zasnivaju na problemu diskretnog logaritma u konačnom polju su *Diffie-Hellman* razmjena ključeva, ElGamal šifra i DSA (engl. *Digital Signature Algorithm*).
- Eliptičke krive: Predstavljaju generalizaciju problema diskretnog logaritma. Primjer takvih šifri su ECDH (engl. *Elliptic Curve Diffie-Hellman key exchange*) i ECDSA (engl. *Eliptic Curve Digital Signature Algorithm*).

Prve dvije familije su uvedene sredinom ’70. godina prošlog vijeka dok su Eliptičke krive predložene sredinom ’80. godina. Ne postoje uspješni napadi na ove šifre ukoliko su parametri šifri, a naročito dužina ključa, pažljivo odabrani.

Za šifre se uvodi pojam nivoa sigurnosti. Kaže se da šifra ima nivo sigurnosti od n bita ako se za datu šifru najbolji poznati napad izvršava u 2^n koraka. U slučaju simetričnih šifri nivo sigurnosti od *n* bita ujedno znači da je i dužina ključa date simetrične šifre jednaka *n* bita, dok kod asimetričnih šifri to ne važi. U tabeli 2.2.1. su date preporučene dužine ključeva za asimetrične i simetrične šifre bezbjedonosnog nivoa 80, 128, 192 i 256 bita.

	Nivo bezbjednosti $[b]$:		80	128	192	256
Dužina ključa $[b]$	Asimetrične šifre	RSA	1024	3072	7680	15360
		DH, DSA, Elmagal	1024	3072	7680	15360
		ECDH, ECDSA	160	256	384	512
	Simetrične šifre	AES, 3DES	80	128	192	256

Tabela 2.2.1. Dužina ključa u zavisnosti od nivoa sigurnosti šifre

Neželjena posledica korišćenja ključeva velike dužine jeste povećanje vremena izvršavanja koda. U slučaju RSA šifre ako povećamo dužinu ključa sa 1024 bita na 3076 bita šifrovanje je $3^3 = 27$ puta sporije. Iz tog razloga se asimetrične šifre ne koriste za šifrovanje velike količine podataka.

Radi boljeg razumijevanja načina funkcionisanja asimetričnih šifri, naročito RSA šifre, treba navesti osnovne matematičke principe na kojima počivaju.

Tehnike iz teorije brojeva na kojima se zasniva asimetrična kriptografija su Euklidov algoritam, Ojelerova Φ funkcija, Fermaova mala teorema i Ojlerova teorema.

i) Euklidov algoritam

Velike brojeve koji se koriste u asimetričnim šiframa nije lako rastaviti na činioce pa se u tu svrhu koristi Euklidov algoritam koji se zasniva na jednakosti

$$\forall r_{0,} r_1 \in \boldsymbol{Z}^+,\ r_0 > r_1\ , \qquad NZD(r_0, r_1) = NZD(\ r_0 - r_1, r_1) \tag{2.2.2}$$

Ovim algoritmom svodimo nalaženje NZD dva cijela broja na nalaženje NZD dva manja broja. Jednačinu 2.2.2. primjenjujemo iterativno u obliku

$$NZD(r_0, r_1) = NZD(\ r_0 - m\, r_1, r_1), m \geq 1 \tag{2.2.3}$$

dok god je $(r_0 - m\, r_1) > 0$. Algoritam se optimizuje tako što se odabira najveća vrijednost koeficijenta m, tj. jednačinu 2.2.3. pišemo u obliku

$$NZD(r_0, r_1) = NZD(\ r_0 mod\ r_1, r_1) \tag{2.2.4}$$

Tako da u posljednjoj iteraciji dobijamo izraz

$$NZD(r_0, r_1) = \cdots = NZD\big(\ r_p, 0\big) = r_p \tag{2.2.5}$$

gdje je r_p ostatak dobijen u posljednjoj iteraciji Euklidovog algoritma.

Prošireni Euklidov algoritam za rezultat daje linearnu kombinaciju ulaznih parametara r_0i r_1 u obliku

$$NZD(r_0, r_1) = s \times r_0 + t \times r_1 \tag{2.2.6}$$

Gdje su koeficijenti s i t iz skupa **Z** i dobijaju se tako što se u svakoj iteraciji Euklidovog algoritma trenutni ostatak r_i predstavljamo kao linearnu kombinaciju ulaznih parametara:

$$r_i = s_i \times r_0 + t_i \times r_1 \quad (2.2.7)$$

U posljednjoj iteraciji dobijamo izraz

$$r_p = NZD(r_0, r_1) = s_p \times r_0 + t_p \times r_1 = s \times r_0 + t \times r_1 \quad (2.2.8)$$

Proširenи Euklidov algoritam koristimo kada je potrebno za cijeli broj a naći njemu inverzan broj b po modulu m. Tada pokrećemo prošireni Euklidov algoritam sa ulaznim parametrima m i a. Broj b je jednak vrijednosti parametra t iz formule 2.2.8, tj. t je broj inverzan broju a po modulu m.

ii) Ojlerova Φ funkcija

Posmatramo prsten $\boldsymbol{Z}m = \{0,1,\dots,m-1\}$. Ukupan broj elemenata skupa $\boldsymbol{Z}m$ koji su sa brojem m uzajamno prosti (zajednički djelilac im je broj 1) daje Ojlerova Φ funkcija, u oznaci $\Phi(m)$. Npr. u skupu $Z6, \Phi(6) = 2$.

Računanje Ojlerove $\Phi(m)$ funkcije se vrši ne prebrojavanjem svih elemenata skupa $\boldsymbol{Z}m$ koji su uzajamno prosti sa m već pomoću sledeće teoreme:

Neka je broj m rastavljen na proste činioce na sledeći način:

$$\mathrm{m} = \prod_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{p}_\mathrm{i}^{\mathrm{e_i}} \quad (2.2.9)$$

Gdje su p_i prosti brojevi a e_i pozitivni cijeli brojevi. Tada se Ojlerova Φ funkcija računa pomoću formule:

$$\Phi(m) = \prod_{\mathrm{i}=1}^{\mathrm{n}} (\mathrm{p}_\mathrm{i}^{\mathrm{e_i}} - \mathrm{p}_\mathrm{i}^{\mathrm{e_i}-1}) \quad (2.2.10)$$

iii) Mala Fermaova teorema

Mala Fermaova teorema pomaže pri utvrđivanju da li je neki broj prost ili ne.

Neka je *a* cijeli broj a *p* prost broj. Tada prema Fermaovoj maloj teoremi važi:

$$a^p \equiv a \ (mod\ p) \quad (2.2.11)$$

Ili u drugom obliku:

$$a^{p-1} \equiv 1 \ (mod\ p) \quad (2.2.12)$$

iv) Ojlerova teorema

Generalizacija Male Fermaove teoreme za dva uzajamno prosta cijela broja a i m je Ojlerova teorema koja glasi

$$a^{\Phi(m)} \equiv 1 \ (mod\ m) \quad (2.2.13)$$

Pošto važi za moduo m Ojlerova teorema je primjenjljiva u prstenima $\boldsymbol{Z}_m$.

2.2.2. RSA

RSA (*Rivest-Shamir-Adleman*) je asimetrična šifra koja ima najširu upotrebu. Objavljena je 1977. godine u časopisu *Scientific American*.

U praksi se RSA najčešće koristi za šifrovanje male količine podataka, u prvom redu za transport ključeva i za digitalne potpise. RSA se ne uvodi kao zamjena za simetrične šifre jer je proces šifrovanja nekoliko puta sporiji nego što je to slučaj kod simetričnih šifri. Danas se RSA koristi u velikom broju komercijalnih sistema, u veb serverima i brauzerima za siguran prenos veb saobraćaja, za garanciju privatnosti i autentifikaciju elektronske pošte, kao i za elektronsko plaćanje kreditnim karticama [14].

RSA šifrovanje i dešifrovanje se vrši u prstenu $\boldsymbol{Z}_n$. Neka je javni ključ $(n, e) = \mathrm{k_{pub}}$, otvoreni tekst P i šifrovani tekst C. Tada je funkcija šifrovanja $E_{k_{pub}}(\mathrm{P})$:

$$C = E_{k_{pub}}(P) \equiv P^e mod\ n \tag{2.2.14}$$

gdje su $P, C\ \in \boldsymbol{Z}_n$.

Funkcija dešifrovanja, za dati privatni ključ $\mathrm{k_{pr}} = \mathrm{d}$ i šifrovani tekst C, $D_{k_{pr}}(\mathrm{C})$ je data sledećom formulom :

$$P = D_{k_{pr}}(C) \equiv C^d mod\ n \tag{2.2.15}$$

U praksi su vrijednosti P, C, n *i* d brojevi duži od 1024 bita. Vrijednost e se naziva javni eksponent ili eksponent šifrovanja, a vrijednost d privatni eksponent ili eksponent dešifrovanja. Generisanje javnog i privatnog ključa za RSA šifru se vrši na sledeći način:

1) Biramo dva velika prosta broja p i q
2) Računamo n kao $n = p \times q$
3) Računamo Ojlerovu Φ funkciju kao $\Phi(n) = (\mathrm{p} - 1)(\mathrm{q} - 1)$
4) Biramo eksponent šifrovanja $e \in \{1,2, \dots, \Phi(\mathrm{n}) - 1\}$ tako da je $NZD\big(\mathrm{e}, \Phi(\mathrm{n})\big) = 1$
5) Računamo eksponent dešifrovanja d tako da važi $\mathrm{d} \times \mathrm{e} \equiv 1\ (\mathrm{mod}\ \Phi(\mathrm{n}))$

Prosti brojevi $p\ i\ q$ se biraju tako što RNG (engl. *Random Number Generator*) generiše slučajne cijele brojeve, a zatim se provjerava jesu li generisani brojevi prosti. Vjerovatnoća da je slučajno odabran cijeli broj ṕ prost je jednaka

$$P(\text{ṕ je prost broj}) = \frac{1}{\ln \text{ṕ}} \tag{2.2.16}$$

Da bi duplirali tu vjerovatnoću uzimamo u obzir samo neparne brojeve pa vjerovatnoća tada iznosi

$$P(\text{ṕ je prost broj}) = \frac{2}{\ln \text{ṕ}} \tag{2.2.17}$$

Vjerovatnoća da je cijeli broj prost opada sporo, srazmjerno dužini tog broja u bitima, što znači da je za velike parametre koje koristimo u RSA (npr. dužine 4096 bita) učestanost pojavljivanja prostih brojeva i dalje dovoljno velika.

Generisanje parametara d i e se vrši jednovremeno, korišćenjem proširenog Euklidovog algoritma. U praksi se prvo kreće sa odabirom eksponenta šifrovanja e iz opsega $(0, \Phi(n))$. Vrijednost e mora ispuniti uslov $NZD(e, \Phi(n)) = 1$ da bi bilo moguće izračunati eksponent dešifrovanja d.

Primijenimo prošireni Euklidov algoritam na ulazne parametre n i e i upospostavljamo relaciju:

$$NZD(e, \Phi(n)) = s \times \Phi(n) + t \times e \tag{2.2.18}$$

Ako je ispunjen uslov $NZD(e, \Phi(n)) = 1$ znamo da je e validan eksponent šifrovanja. Iz jednačine 2.2.8. znamo da je parametar *t*, dobijen proširenim Euklidovim algoritmom, inverzan parametru e, odnosno:

$$d \equiv t(mod\ \Phi(n)) \tag{2.2.19}$$

Prilikom šifrovanja i dešifrovanja otvoreni i šifrovani tekst stepenujemo do stepena e odnosno d. Da bi se smanjio broj koraka tj. operacija množenja RSA šifra koristi algoritam kvadriranje-i-množenje (*square-and-multiply*). Algoritam se zasniva na skeniranju bita eksponenta sa lijeva na desno. U svakoj iteraciji tj. za svaki skenirani bit eksponenta trenutni rezultat se množi. Ako i samo ako trenutno skenirani bit ima vrijednost 1, nakon kvadriranja prethodnog rezultata se vrši jedno množenje. Algoritam je lako pokazati na primjeru:

$$x^{12} = x^{1100} = (x^2 \times x)^{2^2} = (x^3)^4 = x^{12} \tag{2.2.20}$$

Umjesto dvanaest operacija množenja imamo jedno množenje i tri kvadriranja, tj. ukupno četiri operacije.

Primjer RSA šifrovanja i dešifrovanja:

Neka su dva prosta broja $p = 3, q = 11$. Tada je parametar n RSA šifre jednak

$$n = p \times q = 33 \tag{2.2.21}$$

Ojlerova Φ funkcija za vrijednost n je

$$\Phi(n) = (p-1)(q-1) = 20 \tag{2.2.22}$$

Biramo eksponent šifrovanja $e = 3$ tako da je zadovoljen uslov

$$NZD(\Phi(n), e) = 1 \tag{2.2.23}$$

Na jednakost 2.2.23. primijenimo prošireni Euklidov algoritam:

$$NZD(e, \Phi(n)) = s \times \Phi(n) + t \times e \tag{2.2.24}$$

$$NZD(\Phi(n), e) = \mathrm{NZD}(20, 3) = \mathrm{NZD}(\Phi(\mathrm{n}) - 6\mathrm{e}, \mathrm{e}) \tag{2.2.25}$$

U prvoj iteraciji dobijamo parametre $s_1 = 1, t_1 = -6$

Druga iteracija nalazi

$$\mathrm{NZD}(2, 3) = \mathrm{NZD}(\mathrm{e} - \Phi(\mathrm{n}) + 6\mathrm{e}, \Phi(\mathrm{n}) - 6\mathrm{e}) \tag{2.2.26}$$

To je ujedno i posljednja iteracija kojom dobijamo

$$\mathrm{NZD}(2,3)=\mathrm{NZD}(2,1)=1 \tag{2.2.27}$$

Proširenim Euklidovim algoritmom dobijamo koeficijent t koji je jednak eksponentu dešifrovanja d:

$$NZD(2, 3) = -\Phi(\mathrm{n}) + 7 \tag{2.2.28}$$

$$d \equiv 7\ (mod\ \Phi(n)) \tag{2.2.29}$$

Ukoliko šifrujemo otvoreni tekst $P = 4$ šifrovani tekst koji se dobija je

$$C = P^e mod\ n \tag{2.2.30}$$

$$C = 4^3 mod\ 33 = 31$$

Na prijemnoj strani se vrši dešifrovanje

$$P = C^d mod\ n \tag{2.2.31}$$

$$P = 31^7 mod\ 33 = 4 \tag{2.2.32}$$

3. KRIPTOGRAFSKI NAPADI

Uspješan napad na kriptografski sistem odnosno šifru podrazumijeva nalaženje praktičnog načina da napadač od šifrovanog teksta dobije otvoreni tekst [3]. 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 [6]:

- 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.

Side Channel napadi su posebna grupa napada koji su zasnovani na prikupljenim informacijama o fizičkoj implementaciji šifre i dijele se na:

- Analiza energije (engl. *Power analysis*)
- Vremenski napad (engl. *Timing attack*)

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 napad 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 [18].

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 [18]. 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 [18].

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 [19].

3.2. Napadi na klasične šifre

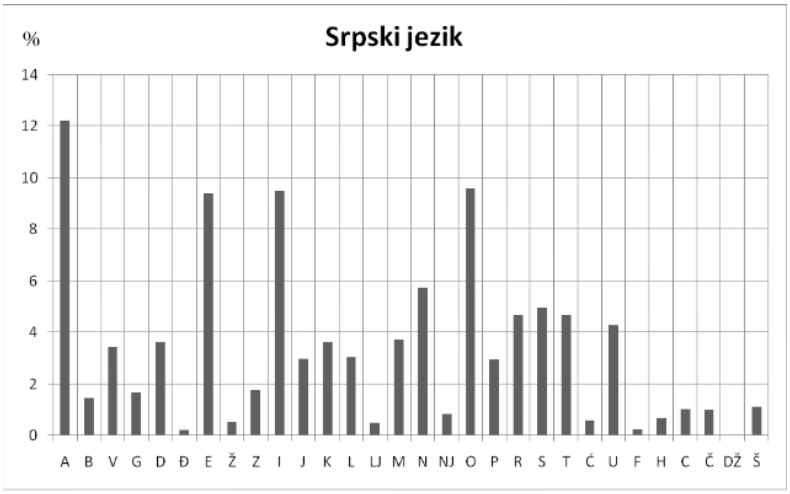
Klasična kriptoanaliza je najstariji oblik analize kriptografskih šifri i u obzir uzima pravilnosti jednog jezika.

3.2.1. Analiza učestanosti

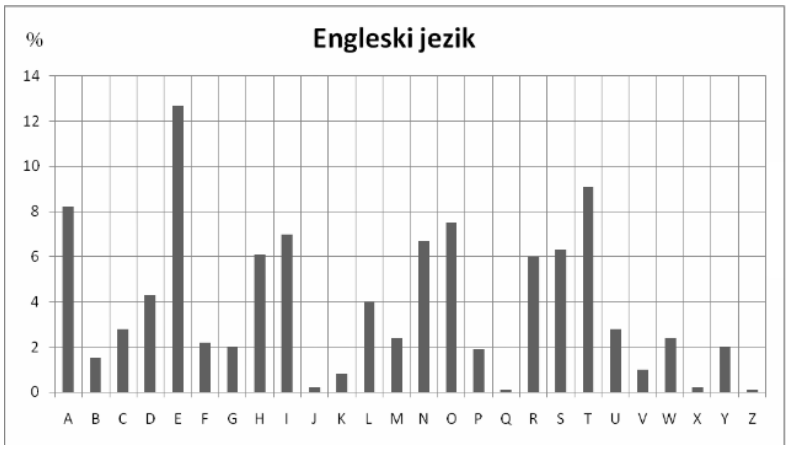
Za svaki jezik, na dovoljno velikom uzorku teksta, mogu se uočiti pravilnosti po pitanju učestanosti pojavljivanja pojedinačnih slova, kombinacije dva odnosno tri slova, itd. Tačno određena distribucija pojave slova postoji za svaki jezik. Nekada je moguće navedene karakteristike otvorenog teksta uočiti i u šifrovanom tekstu i iskoristiti ih u napadu „samo šifrovani tekst“[6].

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 3.2.1[2].

[2] Slike su preuzete sa http://tinyurl.com/ldxqbvg



Slika 2.2.1. Frekvencijska distribucija slova u tekstu napisanom na srpskom jeziku



Slika 3.2.1 Frekvencijska distribucija slova u tekstu napisanom na 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 neće 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 drugačiju frekvencijsku distribuciju od otvorenog teksta, pa napad analizom učestanosti može biti neuspješan [10].

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 [11]. 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

$$n = \sum_{1}^{N} n_i \tag{2.2.1}$$

Ukupan broj parova istih slova u tekstu U_1 je:

$$U_1 = \sum_{i=1}^{N} \frac{n_i(n_i - 1)}{2} \tag{2.2.2}$$

Ukupan broj parova slova u tekstu U je:

$$U = \frac{n(n-1)}{2} \tag{2.2.3}$$

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

$$IC = \frac{U_1}{U} = \frac{\sum_{i=1}^{N} \frac{n_i(n_i - 1)}{2}}{\frac{n(n-1)}{2}} = \frac{\sum_{1}^{N} n_i(n_i - 1)}{n(n-1)} \tag{2.2.4}$$

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 razbijena *Vigenére* šifre. Kod šifrovanja *Vigenére* 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

Diferencijalna i linearna kriptoanaliza su srodni napadi koji se primarno koriste protiv iterativnih simetričnih blok šifri (engl. *Iterative symmetric key block ciphers*). Takve blok šifre vrše više uzastopnih rundi šifrovanja korišćenjem drugačijeg podključa (engl. *subkey*) za svaku rundu. Primjeri takvih šifri su *DES* i *AES*. U oba slučaja, i kod linearne i kod diferencijalne kriptoanalize, napadač proučava razlike u šifrovanim tekstovima koji su rezultat svake runde šifrovanja. Šifrovanje je dobro ako se njom dobija šifrovani tekst koji se čini nasumičnim tako da mala promjena u otvorenom tekstu uzrokuje slučajnu promjenu u šifrovanom tekstu, tj. šifra ima osobinu difuzije. Jedan bit novog šifrovanog teksta, za odgovarajući izmijenjeni otvoreni tekst, ima jednaku vjerovatnoću da bude 1 ili 0. Oba napada traže slučajeve u kojima ovo „50% pravilo“ ne važi, i na taj način otkrivaju potencijalne ključeve. Diferencijalna i linearna kriptoanaliza pripadaju grupi napada „poznat otvoren tekst“.

Diferencijalna kriptoanaliza traži odnos između šifrovanih tekstova dva srodna otvorena teksta, i bazira se na statističkoj analizi dva ulaza i dva izlaza jedne šifre. Par srodnih otvorenih tekstova se formira tako što za jedan otvoreni tekst P i ponavljajući binarni niz X, nađe otvoreni tekst P_2 takav da

$$P_2 = P \oplus X \tag{3.3.1}$$

Gdje $\oplus$ označava operaciju XOR. Napadač šifruje P i P2 svim mogućim podključevima, traži one šifrovane tekstove za koje ne važi „50% pravilo“ i ključ koji daje takav šifrovani tekst je najvjerovatniji podključ [1].

3.3.2. Linearna kriptoanaliza

Linearna kriptoanaliza zahtijeva veliku količinu parova otvoren/šifrovan tekst koji su šifrovani ključem čiju vrijednost napadač ne poznaje [12]. Napad se bazira na statističkoj analizi jedne runde dešifrovanja na velikom uzorku šifrovanog teksta. Napadač dešifruje svaki šifrovani tekst korišćenjem svih mogućih podključeva za jednu rundu šifrovanja i proučava rezultate. Ključ koji daje rezultat koji ne zadovoljava „50% pravilo“ je najvjerovatniji podključ [1].

3.3.3. Dejvisov napad

Dejvisov napad je osmišljen za DES šifru ali se može primijeniti i na druge *Feistel* šifre[3]. 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

[3] Feistel šifre su Blowfish, Twofish, 3DES, RC5, FEAL, Lucifer…

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 i to su IDEA, GOST, SAFER, CAST, DES-X, RC2, TEA [18].

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 $\overline{K}$ za dešifrovanje. Bar jedan program za šifrovanje poruka niz uzastopnih poruka šifruje ključevima $K, K+1, K+2$[4]… 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 [9].

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[5][1]. 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[6], imamo 2^{56} 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 [8].

---

[4] Ove implementacije su zaštićene autorskim pravima, pa nema referenci na raspolaganju [9].

[5] Ako se otvoreni tekst dužine 15 B 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“[5].

[6] *Single DES* koristi ključ dužine 56 bita.

3.3.6. Meet-in-the-middle napad

Meet-in-the-middle napad su osmislili *Diffie* i *Hellman*, i spada u grupu napada „poznat otvoren tekst“. 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.

$$C = E_{K_r}\left(E_{K_l}(P_1)\right) \tag{2.3.1}$$

Napadač mora da poznaje bar dva para otvoreni/šifrovani tekst. Ako bismo pokušali *brute-force* napad na ovako šifrovan tekst trebalo bi pretražiti sve moguće kombinacije oba ključa tj. efektivna dužina ključa bi bila 2*k*, 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

$$E_{K_{l,i}}(P_1) = M_{l,1} \tag{2.3.2}$$

a indeks *i* uzima vrijednosti $\{1,2,\dots,2^k\}$. $M_{l,1}$ 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

$$M_{r,1} = M_{l,i} \tag{2.3.3}$$

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 duplim 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, npr. 3DES [13].

Primjer uspješnog meet-in-the-middle napada je napad na *Double DES* [2].

3.3.7. Standardni ASCII napad

Standardni ASCII napad pripada grupi napada „samo šifrovani 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.

3.4. *Side channel* napadi

Side channel napad podrazumijeva korišćenje dodatnih informacija zasnovanih na fizičkoj implementaciji šifre, uključujući i hardver koji se koristi za šifrovanje i dešifrovanje podataka. Prethodno navedeni napadi podrazumijevaju da napadač ima pristup otvorenom i/ili šifrovanom tekstu, i po mogućnosti i algoritmu šifrovanja. Kod *Side channel* napada informacije koje se koriste uključuju podatke o vremenu trajanja šifrovanja (dešifrovanja), potrošnji energije, itd. [1].

3.4.1. Analiza energije

Napadač proučava potrošnju energije kriptografskog sistema, i na taj način može doći do informacije o ključevima kao i druge tajne informacije. Primjer takvog napada je dat u odjeljku 4.3.

3.4.2. Vremenski napad

Napadač pokušava da ugrozi kriptografski sistem analizom vremena potrebnog za izvođenje algoritma za šifrovanje. Pošto svaka logička operacija traje tačno određeno vrijeme, preciznim mjerenjem vremena može se otkriti metoda šifrovanja.

4. NAPADI NA SAVREMENE ŠIFRE

4.1. DES

Dva glavna argumenta protiv uvođenja DES šifre su bila u vezi sa dužinom ključa i S-kutijama. Mala dužina ključa, 56 bita, ostavlja šifru podložnom *brute-force* napadima. U periodu kada je DES prvi put predložen javnosti dizajn S-kutija je bio tajan i smatralo se da dizajneri DES šifre eksploatacijom matematičkih karakteristika S-kutija mogu uspješno sprovesti analitički napad. Uprkos navedenom trenutno mogući analitički napadi na DES nisu veoma efikasni, ali pošto je moguć uspješan *brute-force* napad DES nije pogodan za većinu aplikacija.

U slučaju *brute-force* napada napadač ima pristup jednom paru otvoreni/šifrovani tekst P_1, C_1, zna da je funkcija šifrovanja takva da je $P = DES_k(C)$. Dužina ključa DES šifre je 56 bita pa napadač testira svih 2^{56} mogućih ključeva sve dok ne nađe ključ koji zadovoljava jednakost:

$$DES_{k_i}{}^{-1}(C_1) = P_1, i = \{1, \dots, \quad 2^{56} - 1\} \tag{4.1.1}$$

Vjerovatnoća nalaženja ključa je $\frac{1}{2^{56}}$. Da bi sa sigurnošću moglo tvrditi da je ključ ispravan potrebno ga je provjeriti na još jednom paru P_2, C_2. Regularni računari nisu podobni za vršenje ovakvog pretraživanja, ali postoje mašine koje su namjenski napravljene za pretragu ključeva.

Whitfield Diffie i *Martin Hellman* su 1977. godine procijenili da je moguće napraviti *brute-force* mašinu koja bi mogla razbiti DES šifru u roku od par dana i da bi ta mašina koštala približno 20,000,000 dolara. EFF (*Electonic Frontier Foundation*) je 1998. godine izgradila hardversku mašinu *Deep Crack* kojom je uspješno napadnut DES za samo 56 sati. Prosječno vrijeme pretrage *Deep Crack* mašine je bilo 15 dana, a izrada same mašine je koštala manje od 250,000 dolara. Uspjeh *Deep Crack* mašine je smatran zvaničnom demonstracijom da DES više nije sigurna šifra.

Slika 4.1.2 *Deep Crack* mašina

Ključ dužine 56 bita ne garantuje sigurnost naročito za šifrovanje povjerljivih podataka, tako da DES treba koristiti samo u aplikacijama u kojima je bitna kratkotrajna bezbjednost, reda par sati, ili kada je značaj podataka koji se šifruju mali. Varijacije DES, naročito 3DES su i dalje sigurne.

1990. godine *Eli Birham* i *Adi Shamir* su otkrili Diferencijalnu kriptoanalizu, tip napada koji se u principu može primijeniti na bilo koju blok šifru. Ispostavilo se da su S-kutije DES-a otporne na ovaj napad. Član IBM tima koji je radio na projektovanju DES je nakon otkrića Diferencijalne kriptoanalize izjavio da su u vrijeme dizajniranja šifre znali da je ovakav napad moguć i da je to razlog zašto nije odmah objavljen dizajn S-kutija. 1993. godine je *Mitsuru Matsui* objavio napad Linearna kriptoanaliza, koji se takođe oslanja na strukturu S-kutija. Pokazalo se da za uspješnu diferencijalnu kriptoanalizu napadaču treba 2^{47} parova otvoreni/šifrovani tekst. U slučaju linearne kriptoanaize napadač treba da ima 2^{43} parova otvoreni/šifrovani tekst.

Tri teoretska napada: Dejvisov napad, diferencijalna i linearna kriptoanaliza, su u teoriji manje kompleksni nego *brute-force*, ali zahtijevaju veliki broj poznatih i odabranih otvorenih tekstova do kojih napadač u praksi ne može doći, kao i velike resurse po pitanju vremena i memorije koju napadač treba da ima na raspolaganju. Iz navedenog se zaključuje da je najpraktičniji napad na DES *brute-force*.

Tri grupe napada na DES pod imenom *DES Challenges* su serije *brute-force* napada na DES koje su organizovane od strane *RSA Security* sa ciljem pokazivanja manjka bezbjednosti koju nudi DES. Prvi napad, *DES Challenge I*, je počeo 1997.godine i obavljen je u roku od 96 dana od strane *distributed.net*. *DES Challenge II-1* je obavljen 1998.godine, i trajao je 39 dana. *DES Challenge II-2* je obavljen za samo 56 sati, 1998.godine, korišćenjem Deep Crack mašine. *DES Challenge III* je bio zajednički poduhvat između *distributed.net* i *Deep Crack* mašine. Ključ je nađen za 22 sata i 15 minuta, 1999. godine.

Predloženi su napadi i na DES sa manjim brojem rundi (potpuno DES šifrovanje se vrši u 16 rundi). Takvi napadi su dali informacije u tome koliki je minimalni broj rundi koji treba da ima DES šifra da bi se smatrala bezbjednom, i koliku sigurnosnu marginu ima originalni 16-rundi DES.

Od sredine 1970. do sredine 1990. godine DES je bila dominantna simetrična šifra, a danas se koristi u obliku 3DES koji do danas nije uspješno napadnut. Mana 3DES šifre je u vezi sa brzinom šifrovanja (dešifrovanja), koja je 3 puta manja nego kod DES šifre.

Iako danas DES više nije bezbjedna šifra, i dalje ima primjenu u *legacy* aplikacijama. U informacionim tehnologijama *legacy* aplikacije su aplikacije naslijeđene iz jezika, platformi i tehnika koje pripadaju prošlim generacijama, u odnosu na sadašnju tehnologiju.

U tabeli 4.1.3 je data istorija najznačajnijih predloženih i implementiranih napada na DES.

godina	Predloženi ili implementirani napadi
1977	*Diffie* i *Hellman* dali procjenu troškova izgradnje mašine za pretragu ključeva, kojom bi se mogao uspješno pronaći DES ključ.
1990	*Birham* i *Shamir* objavljuju napad Diferencijalna kriptoanaliza koja zahtijeva 2^{47} parova otvoreni/šifrovani tekst.
1993	*M. Wiener* predlaže nacrt hardverske mašinu za pretragu ključeva koja bi pretragu obavila u prosjeku za 36 sati i čija bi izgradnja koštala 1, 000,000 dolara.
1993	*Matsui* objavljuje napad Linearna kriptoanaliza koja zahtijeva 2^{43} odabranih šifrovanih tekstova
Jun 1997.	DES *Challenge I*: Uspješno sproveden brute-force napad za koji je bilo potrebno 4.5 mjeseca.
Februar 1998	DES *Challenge II-1*: brute-force napad za koji je bilo potrebno 39 dana.
Jul 1998.	DES *Challenge II-2*: brute-force napad upotrebom *Deep Crack* mašine. Cijena napada je iznosila 250,000 dolara i napad je trajao 56 sati.
Januar 1999.	DES *Challenge III*: brute-force napad upotrebom *Deep Crack* za koji je bilo potrebno ukupno 22 sata.
April 2006.	Univerziteti Bochum i Kiel su sagradili COPACABANA mašinu za pretragu ključeva čija je izgradnja koštala 10,000 dolara. Srednje vrijeme pretrage je iznosilo 7 dana.

Tabela 4.1.3 Istorija napada na DES

U tabeli 4.1.4 je data procjena vremena potrebnog za izvođenje uspješnog *brute-force*, na simetrične šifre, u zavisnosti od dužine ključa. Vrijeme koje je potrebno za *brute-force* napad na neku šifru daje informaciju o bezbjednosti te šifre [13].

Dužina ključa [**b**]	Procjena bezbjednosti
56-64	Kratkoročna: reda nekoliko sati ili dana.
112-128	Dugoročna: nekoliko decenija ukoliko pretpostavimo da neće biti kvantnih računara.
256	Dugoročna: nekoliko decenija čak uz postojanje kvantnih kompjutera.

Tabela 4.1.4 Bezbjednost simetričnih šifri u zavisnosti od dužine ključa

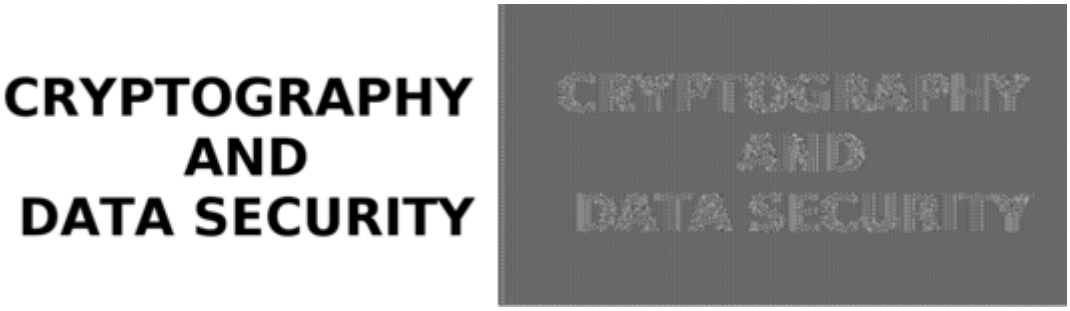
4.2. AES

Do danas nije otkriven analitički napad na AES šifru koji je manje kompleksan od *brute-force* napada. 2003. godine je američka Vlada objavila dokument kojim se odobrava korišćenje AES šifre za šifrovanje povjerljivih podataka i to tako da AES(128, 192 i 256) se može koristiti za zaštitu povjerljivih informacija nivoa SECRET, a AES(192 i 256) se može koristiti za šifrovanje podataka nivoa povjerljivosti TOP SECRET [15]. AES šifra sa smanjenim brojem rundi je podložna napadima koji su samo od akademskog značaja. Za 6 rundi AES je moguć napad korišćenjem 6×2^{32} blokova otvorenog teksta (napad odabran otvoreni tekst), a ključ se generiše u 2^{44} koraka, tj. operacija. To znači da napadač mora da šifruje i analizira približno 400 GB otvorenog teksta, ako se jedna operacija izvrši za vrijeme od jedne mikrosekunde ukupno trajanje napada bi bilo 200 dana. Napad na 7 rundi AES zahtijeva 2^{128} odabranih otvorenih tekstova i 2^{120} operacija. U slučaju da jedna operacija traje jednu nanosekundu čitav proces bi trajao 4×10^{19} godina[16]! Treba uočiti kako se sigurnost AES šifre značajno povećava dodavanjem samo jedne runde šifrovanja.

Naučni rad autora *Ferguson*, *Schroeppel* i *Whiting* koji je objavljen 2001.godine pokazuje da se AES šifrovanje može predstaviti kao suma 2^{25} verižnih razlomaka, i da ta osobina AES šifre može poslužiti kao osnova novog napada koji bi se u budućnosti mogao razviti [17].

Nicolas Courtois i *Josef Pieprzyk* su 2002. godine objavili rad kojim uvode metodu XSL (engl. *Extended Sparse Linearization*). XSL se bazira na činjenici da je AES šifru moguće predstaviti pomoću sistema od 8,000 kvadratnih jednačina sa 1,600 nepoznatih. Cijenu ovakvog napada nemoguće je predvidjeti, a kao kritika je dato i to što autori nisu dali praktičan primjer napada.

U praksi se šifre bloka koriste kao gradivne jedinice šifri niza koje šifruju velike otvorene tekstove. Jedan od načina šifrovanja velikih otvorenih tekstova (dužina otvorenog teksta prelazi dužinu bloka AES šifre) jeste ECB (engl. *Electronic Code Book*). ECB zahtijeva da dužina otvorenog teksta bude cjelobrojni umnožak dužine bloka šifre koju koristimo, u ovom slučaju AES. Ako to nije slučaj, otvoreni tekst se dopunjuje do zahtijevane veličine (*padding*). Svaki blok otvorenog teksta se šifruje AES šifrom. Ovakvo šifrovanje je determinističko: isti blokovi otvorenog teksta se slikaju u iste blokove šifrovanog teksta, i to je osnova napada na ovakvu upotrebu AES šifre. Ako se ECB koristi za šifrovanje novčanih transakcija jedan transfer će se sastojati od pet blokova: ID Banke pošiljaoca, račun pošiljaoca, ID banke primaoca, račun primaoca, novčani iznos. Napadač otvori račun u obadvije banke, i sa jednog računa uplati na drugi minimalan novčani iznos. Posmatra šifrovani tekst koji odgovara transakciji a koji se takođe sastoji iz pet blokova. Čuva blokove 1, 3, 4 koji predstavljaju identifikacione brojeve obadvije banke kao i šifrovanu verziju njegovog računa u drugoj banci. Uz uslov da banke ne mijenjaju ključeve često napadač zna da se jedan isti ključ koristi za šifrovanje više transfera. Napadač zatim presrijeće transakciju između neka druga dva računa datih banaka, i zamjenjuje blok 4 šifrovanog teksta sa svojim šifrovanim blokom 4. Banka primaoca ne može otkriti tu zamjenu. Ovakav napad nije napad na AES šifru već eksploatiše način na koji se ta šifra koristi u sistemu. Kao prevencija protiv ovakvog napada se koriste tehnike očuvanja integriteta poruka, kao što su digitalni potpisi i MACs (engl. *Message Authentication Codes*). Drugi način onemogućavanja napada jeste česta promjena ključeva. Mana ECB šifrovanja se vidi na slici 4.2.1 [13]. Bez obzira što se za šifrovanje koristila AES šifra i ključ dužine 256 bita, zbog determinističke prirode ECB šifrovana fotografija je i dalje razumljiva golim okom.



Slika 4.2.1. Fotografija prije i nakon šifrovanja AES 256 b, ECB mod

Ukoliko se umjesto ECB režima koristio CBC režim ovakav napad ne bi bio moguć jer napadač ne može da prepozna obrasce u šifrovanom tekstu i dovede ih u vezu sa otvorenim tekstom.

4.3. RSA

Ne može se koristiti privatni ključ male dužine, a da se ne kompromituje bezbjednost RSA. Ako bismo odabrali d kao u primjeru RSA šifrovanja, navedenom u odjeljku 2.2.2, onda bi napadač mogao za kratko vrijeme *brute-force* napadom otkriti vrijednost d. Čak i za velike brojeve, recimo 128 bita, postoje *key recovery* napadi. Dokazano je da privatni ključ mora imati minimalnu dužinu od $0.3t$ gdje je *t* dužina modula *n* u bitima. U praksi se parametar *e* može odabrati da bude što manje dužine dok *d* mora imati punu dužinu modula n. Bezbjednost RSA leži u nemogućnosti rastavljanja velikih brojeva na činioce. Današnja tehnologija omogućava rastavljanje na činioce brojeva dužine 1024 bita, tj. brojeva koji imaju oko 310 decimalnih cifara.

RSA šifrovanje je determinističko tj. za jedan tačno određeni ključ i otvoreni tekst dobija se tačno određeni šifrovani tekst. Napadač može izvesti statističke karakteristike otvorenog teksta iz šifrovanog teksta. Zbog te osobine napadač može da sprovede napad odabran otvoreni tekst tako što će šifrovati otvorene tekstove i tražiti podudaranja u šifrovanim tekstovima. Još jedna karakteristika RSA šifrovanja je da otvorenim tekstovima $P = 0, P = 1, P = -1$ uvijek odgovaraju šifrovani tekstovi $C = 0, C = 1, C = -1$, respektivno.

Predložen je veliki broj napada na RSA šifru, od kojih se većina zasniva na eksploataciji implementacije RSA šifre. Napadi mogu biti na protokol, matematički ili *side-channel* napadi.

Napadi na protokol eksploatišu način na koji se RSA šifra koristi.

RSA je *malleable*. Za kriptografsku šifru se kaže da je *malleable* ako napadač može da transformiše šifrovani tekst u drugi šifrovani tekst. To se postiže tako što napadač zamjenjuje šifrovani tekst C sa $s^e \times C$, gdje je s neki cijeli broj. Tada se na prijemnoj strani prilikom dešifrovanja dobija otvoreni tekst $(s^e \times C) \equiv s^{ed} C^{ed} \equiv s \times P \; mod \; n$. Takvi ciljani napadi mogu biti veoma štetni, naročito pri novčanim transakcijama kada napadač ima mogućnost da biranjem broja s manipuliše iznos koji se prenosi. Kao rješenje ovog problema se koristi *padding* koji podrazumijeva dodavanje bita u otvoreni tekst prije šifrovanja, a na prijemnoj strani, nakon dešifrovanja, dodati biti se odbacuju. Tehnika za *padding* RSA poruka je OAEP (engl. *Optimal Asymetric Encryption Padding*). Moderni bezbjedonosni standardi daju tačno uputstvo korišćenja RSA čijim se pridržavanjem onemogućavaju ovakvi napadi.

Ukoliko jedan isti moduo $n = p \times q$ RSA šifre koristi više korisnika, pri čemu svaki korisnik ima jedinstven par ključeva (e_i, d_i), na prvi pogled se čini da korisnik koji ima parametre (e_a, d_a) ne može dešifrovati poruke namijenjene korisniku sa parametrima (e_b, d_b). Ta tvrdnja je

netačna i takvo korišćenje RSA šifre nije sigurno. Korisnik a može uz pomoć svojih eksponenata (e_a, d_a) rastaviti moduo n na proste činioce p i q a zatim poznavajući eksponent šifrovanja korisnika b, e_b, izračunati eksponent dešifrovanja d_b. Time se pokazuje da se jedan moduo RSA šifre ne smije koristiti za više entiteta već jedna vrijednost modula odgovara tačno jednom korisniku [14].

Najbolji matematički napad jeste rastavljanje modula n na proste činioce. Napadač zna šifrovani tekst C, eksponent šifrovanja e i moduo n. Za cilj ima izračunavanje eksponenta dešifrovanja d za koji zna da ima osobinu $e \times d \equiv 1\ (mod\ \Phi(n))$. Kada bi znao vrijednost $\Phi(n)$, pomoću proširenog Euklidovog algoritma bi mogao doći do vrijednosi d. Najbolji način za računanje vrijednosti $\Phi(n)$ jeste rastavljanje n na proste činioce p i q. Ukoliko napadač uspije u tome napad se izvršava u tri koraka:

$$\Phi(n) = (p-1)(q-1) \tag{4.3.1}$$

$$d^{-1} \equiv \mathrm{e}\ (mod\ \Phi(n)) \tag{4.3.2}$$

$$P \equiv C^d\ mod\ n \tag{4.3.3}$$

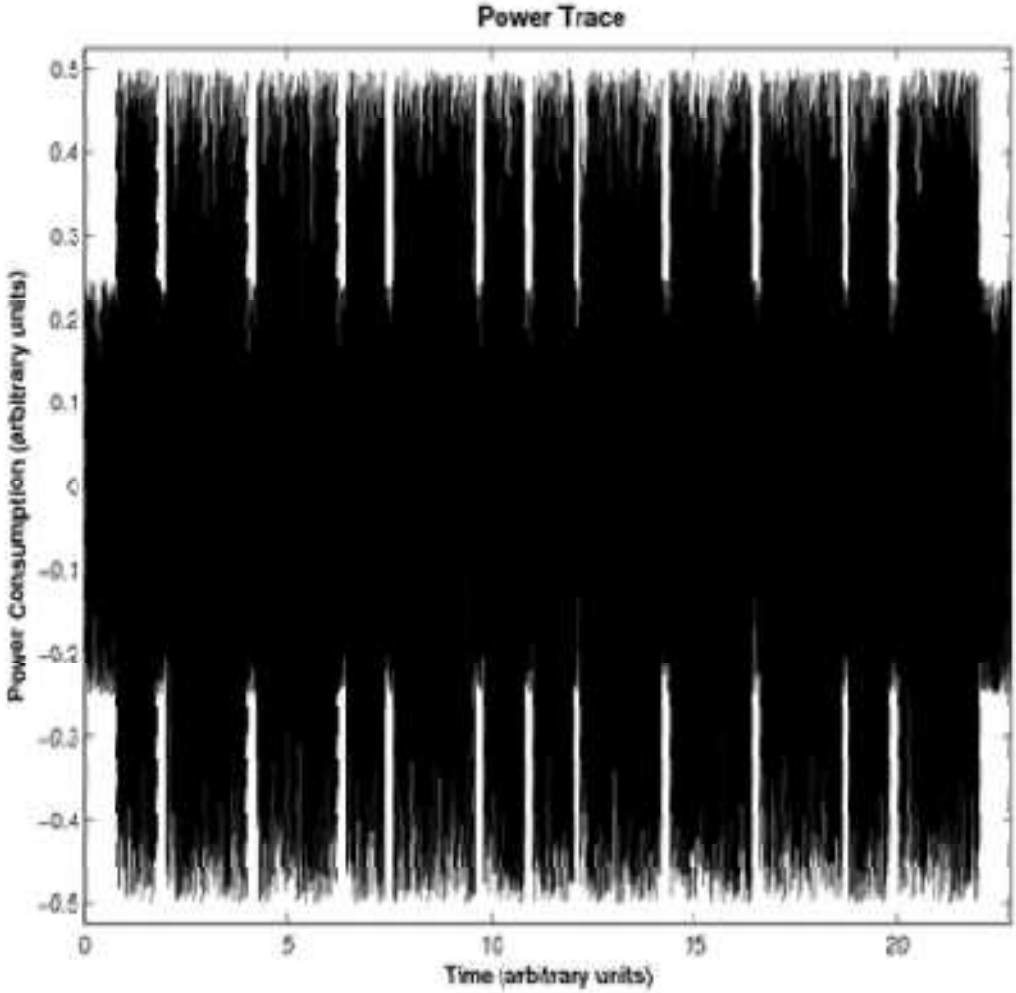
Da bi se spriječio ovakav napad, n mora biti dovoljno veliko, dužine preko 1024 bita. Do skoro su sve RSA aplikacije koristile *n* dužine 1024 bita. Danas se vjeruje da je moguće razbiti broj te dužine za vrijeme od deset do petnaest godina. Za postizanje dugotrajne bezbjednosti, reda nekoliko decenija, preporučuje se da RSA parametri budu dužine 2048 do 4096 bita. Jedan od najbržih algoritama za rastavljanje cijelih brojeva na proste činioce je *General Number Field Sieve* [14].

Napadač *Side-channel* napadom informacije o eksponentu dešifrovanja d dobija posmatranjem potrošnje snage tj. električne energije (engl. *Power consumption*) ili praćenjem vremena koje je potrebno za izvršavanje algoritma (engl. *Time behaviour*). Da bi se ostvarili ovi napadi napadač mora imati direktan pristup RSA implementaciji, npr. mobilnom telefonu ili smart kartici.

Slika 4.3.5. prikazuje grafik snage RSA implementacije na mikroprocesoru, tj. prikazuje potrošnju električne struje u vremenu u toku RSA dešifrovanja. Cilj napada je izdvajanje eksponenta dešifrovanja d. Na grafiku se vide periodi aktivnosti između kratkih perioda kada je aktivnost manja. Najveću potrošnju struje zahtijevaju operacije množenja i kvadriranja, kada se računa

$$P \equiv C^d\ mod\ n \tag{4.3.4}$$

Ako se bliže pogleda grafik potrošnje struje vidimo da su neki intervali aktivnosti dva puta duži od drugih. To se dešava zbog algoritma kvadriranje-i-množenje. Duži intervali označavaju dvije vezane operacije kvadriranja i množenja (bit eksponenta je 1) dok kraći intervali označavaju samo kvadriranje (bit eksponenta je 0). Pošto znamo vezu izmedju redosljeda operacija kvadriranja i množenja i binarnog zapisa eksponenta *d* možemo zaključiti vrijednosti bita privatnog ključa.



Slika 4.3.5 Grafik potrošnje električne energije jedne RSA implementacije u toku operacije dešifrovanja

Sa slike 4.3.5 možemo očitati dvanaest bita eksponenta dešifrovanja d:

Biti d: 0 1 1 0 1 0 0 1 1 1 0 1

Ovakav napad se može onemogućiti tako što se vrše „lažna“ množenja nakon kvadriranja koja odgovaraju 0 bitima eksponenta. Tada se sa grafika potrošnje struje ne može izvesti zaključak o bitima privatnog ključa *d.* Bitno je naglasiti da jedna kriptografska šifra može biti matematički jaka, ali i dalje ranjiva na *side-channel* napade [13].

Nekoliko alternativnih oblika RSA šifre je predloženo. Jedan predlog je da se koristi moduo n koji ima više od dva prosta činioca, npr $n = p \times q \times r$. Takođe je predloženo i da n ima oblik $n = p^2 q$. U oba slučaja je moguće povećati brzinu dešifrovanja dva do tri puta u odnosu na standardni RSA [13].

5. ZAKLJUČAK

U radu je data klasifikacija šifri kao i pregled napada na kriptografske šifre. Pojedinačno je raspravljano o mehanizmima šifrovanja simetričnih šifara DES i AES, i asimetrične šifre RSA, a posebno se obratila pažnja na bezbjednost ovih šifara. 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*. Kao primjer navodimo jednostavnu šifru supstitucije, kod koje će se bez obzira na dužinu ključa frekvencijskom analizom uspješno sprovesti napad šifrovani tekst. 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.

Budućnost kriptoanalize zavisi od tehnološkog napretka, ali i od napredovanja teorijske kriptoanalize. U radu je dat primjer diferencijalne kriptoanalize koja je bila poznata kreatorima DES šifre skoro dvije decenije prije nego što su se stekli uslovi da se taj napad sprovede u praksi, pa se stoga značaj teorijskih napada ne smije odbaciti. Za predviđanje neposredne budućnosti u obzir uzimamo Murov zakon kojim se, između ostalog, procjenjuje da se računarska moć udvostručuje svakih 18-24 mjeseca dok cijene ostaju nepromijenjene. Uticaj Murovog zakona na razvoj kriptoanalize se može odmah uočiti: ukoliko je danas za uspješan napad na jednu šifru potrebno mjesec dana i računara u vrijednosti od 1,000,000 dolara onda će za 18 mjeseci cijena istog napada biti 500,000 dolara jer će napadaču biti potrebno upola manje računara. Najveću prijetnju današnjim šiframa predstavljaju kvantni računari koji funkcionišu po principima kvantne mehanike. Umjesto standardnog predstavljanja podataka pomoću bita, kao što je to slučaj kod današnjih digitalnih računara, kvantni računari operacije vrše nad kvantnim bitima (engl. *Qubit*). Veliki broj stranih država izdvaja sredstva za finansiranje teorijskih istraživanja i praktičnog razvoja kvantnih računara što daje osnova pretpostavki da će u budućnosti takvi računari biti komercijalizovani tj. dostupni civilnom stanovništvu. Šorov algoritam, objavljen 1994. godine, je algoritam koji se pokreće na kvantnom računaru i kojim se vrši rastavljanje cijelih brojeva na proste činioce i to za značajno kraće vrijeme nego što je to slučaj kod računara današnjice. Onog trenutka kada nalaženje prostih činioca velikih cijelih brojeva postane računarski izvodljivo, asimetrične šifre koje se danas koriste, u prvom redu RSA, DSA i ECDSA, neće biti sigurne.

Kriptografija je matematička disciplina sa strogo ustanovljenim principima, globalno prihvaćenim standardima, i svakodnevno je u upotrebi u Internet komunikaciji kao i u velikom broju aplikacija. 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.

LITERATURA

[1] Eric Conrad, Types of Cryptographic Attacks, https://www.academia.edu/4739047/Types_of_Cryptographic_Attacks .

[2] Edward Schaefer, *An introduction to cryptography and cryptanalysis*, http://tinyurl.com/qhjhs65

[3] Bruce Schneier, *A self-study course in block-cipher cryptanalysis*, https://www.schneier.com/paper-self-study.html

[4] Fauzan Mirza, *Block ciphers and cryptanalysis*, http://borax.polux-hosting.com/madchat/crypto/codebreakers/fmirza-report.pdf

[5] James Price, *Cryptanalysis and Brute Force Attacks*, https://www.scribd.com/doc/218489495/Cryptanalysis-and-Brute-Force-Attacks-Paper

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

[7] Niels Ferguson, Bruce Schneier, Tadayoshi Kohno, *Cryptography Engineering- Design principles and practical applications*, Wiley Publishing, 2010.

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

[9] John Kelsey, Bruce Schneier, David Wagner, *Key-Schedule Cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES*, https://www.schneier.com/paper-key-schedule.pdf

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

[11] William F. Friedman, *The index of coincidence and its applications in cryptanalysis*, Aegan Park Press 1987.

[12] Mitsuru Matsui, *„Linear Cryptanalysis Method for DES Cipher“*, *International Association for Cryptologic Research*, Volume 767, Pages 386-397, 1993.

[13] Christof Paar, Jan Pelzl, *Understanding Cryptography*, Springer-Verlag Berlin Heidelberg, 2010.

[14] Dan Boneh, *Twenty years of attacks on the RSA cryptosystem*, crypto.stanford.edu/~dabo/papers/RSA-survey.pdf, 1990.

[15] Sabbir Mahmud, *A study on parallel implementation of AES*, Independent University, Bangladesh, 2004.

[16] Reinhard Wobst, *Cryptology unlocked*, Wiley Publishing, 2007.

[17] Niels Ferguson, Richard Schroeppel, Doug Whiting, *A simple algebraic representation of Rijndael*, 2001.

[18] Henk C. A. van Tilborg, Sushil Jajodia, *Encyclopedia of Cryptography and Security*, Springer US, 2005.

[19] https://www.linkedin.com/pulse/20141201173411-1571978-chosen-ciphertext-attack-cca