SEMINARSKI RAD

PANEVROPSKI UNIVERZITET APEIRON
FAKULTET POSLOVNE INFORMATIKE

Vanredne studije
Smjer »Poslovna informatika«

Predmet: PRINCIPI PROGRAMIRANJA

Tema:

»Kriptografija«

»simetrični i asimetrični algoritmi«

Predmetni nastavnik
Prof. dr Zoran Ž. Avramović, dipl.inž.elek.

Student
Dražen Petrović,vandredni student 2. godine
Index br:0097/06

Banja Luka, maj, 2008

KRATKI SADRŽAJ

1. Uvod

Sigurnost računarskih sistema postaje sve važnija, jer sve više korisnika na sve više načina koristi sve više informacija u računarskom svijetu. U takvom sistemu postoji i sve veća opasnost od neovlaštene upotrebe informacija, podmetanja krivih informacija ili uništavanja informacija. U računarskim sistemima informacije se prenose raznovrsnim otvorenim i nesigurnim komunikacijskim putevima. Pristup do tih puteva ne može se fizički zaštititi pa svaki neprijateljski nastrojen napadač može narušiti sigurnost sistema. Zbog toga zaštitni komunikacijski mehanizmi nad nesigurnim komunikacijskim kanalom postaju najvažniji oblik ostvarenja sigurnosti. Pokazuje se da je najdjelotvornija zaštita poruka njihovo kriptiranje.

U ovom radu ću pobliže objasniti osnovne pojmove vezane za kriptovanje i algoritme koji su se koristili i koji se koriste kako bi se zaštitila privatnost unutar mreže računara.

2. Osnovni termini

Kriptografija je nauka "tajnog pisanja", tj. nauka čuvanja informacija u onoj formi koja će biti čitljiva samo onima kojima je informacija namijenjena dok će za ostale biti neupotrebljiva. Usporedo sa razvojem kriptografije razvila se i nauka kojoj je cilj analizom kriptirane poruke odgonetnuti njen sadržaj. Ta nauka se naziva **kriptoanaliza**.

Pored gore navedenog, valja spomenuti jednu bitnu razliku između termina **kriptografija** i termina **kriptologija**. *Kriptografija* je nauka koja se bavi svim aspektima sigurnosnog transporta podataka kao što su na primjer autentifikacija (web, lokalne mreže i sl.), digitalni potpisi, razmjena elektroničkog novca. *Kriptologija*, je za razliku grana matematike koja se bavi matematičkim načelima, te matematičkom implementacijom kriptografskih metoda.

Originalna poruka koju je pošiljaoc će slati u daljnjem razmatranju će se zvati čisti tekst ili original. Zatim, kodiranje poruke tj. postupak pretvaranja originala (čistog teksta) u nečitljiv oblik ćemo nazvati enkripcija. Tako enkriptiran tekst ima engleski termin ciphertext, a mi ćemo je jednostavno nazvati kodiranom porukom. Nadalje, postupak dekodiranja poruke, tj. vraćanja poruke iz njenog enkriptiranog oblika u originalni (*čisti tekst*) oblik naziva se dekripcija. Vrlo važan termin u kriptografiji je **ključ**. Ključ ima veliku ulogu u enkripciji i dekripciji poruke.

3. Osnovni kriptografski algoritmi

Nekada, prije nego što su računari ušli u široku upotrebu, tj. prije nego su se dovoljno razvili, većina kriptografskih metoda šifriranja se bazirala na tajnosti **šifre**. No, tako bazirani algortimi su se pokazali dosta nepouzdani, te su se morale pronaći neke druge metode šifriranja. Današnje metode šifriranja zasnivaju se na upotrebi **ključa**. Ključ je najvažniji dio u pravilnom enkriptiranju i dekriptiranju poruka.

Upravo ovisno o načinu korištenja ključa, razvile su se dvije klase algoritama. Jedna je **simetrična**, a druga **asimetrična** klasa. Drugim riječima, postoje simetrični algoritmi kriptiranja i asimetrični algoritmi kriptiranja. Osnovna razlika je u tome da simetrični algoritmi koriste isti ključ za enkripciju i dekripciju neke poruke (ili se ključ za dekripciju može lako proizvesti iz originalnog ključa za enkripciju), dok asimetrični algoritmi koriste različite ključeve za enkripciju i dekripciju iste.

- **Simetrični algoritmi:**

Ove algoritme dijelimo u dvije grupe: **stream šifriranje** i **blok šifriranje** Stream šifriranje radi tako da se enkripcija poruke (originala) vrši bit po bit, dok se kod blok

šifriranja enkripcija vrši po blokovima podataka, tj. uzimaju se blokovi od više bitova (64, 128, 196, 256 ...), te se enkriptiraju kao cjelina. Dekripcija se najčešće vrši *inverznim enkriptiranjem*, tj. algoritam je isti, ali se podključevi enkripcije koriste obrnutim redoslijedom.

- **Asimetrični algoritmi:**

Ove algoritme nazivamo još i **public-key algorithms,** tj. algoritmi s **javnim ključem**. Razlog ovakvom nazivu je taj što je dozvoljeno da se jedan od ključeva potreban za enkripciju/dekripciju objavi javno (npr. Internet, novine). Ovdje treba obratiti pažnju na riječi "jedan od ključeva". Ono što je specifično za ovaj tip algoritma je to da se koriste **dva** ključa za enkripciju/dekripciju poruke (originala). Ideja je sljedeća: osoba **A** objavi svoj **javni ključ** preko nekog medija (npr. Internet). Osoba **B**, koja osobi **A** želi poslati tajnu poruku enkriptira tu svoju poruku s ključem koju je osoba **A** javno objavila te joj takvu poruku pošalje (recimo preko e-mail servisa). Jedino osoba **A** sa svojim **privatnim (tajnim)** ključem može dekriptirati poruku poslanu od osobe **B** i nitko drugi.

Uglavnom, simetrični algoritmi su po svojoj prirodi brži, tj. implementacija na računaru se brže odvija od implementacije asimetričnih algoritama. No, zbog nekih prednosti asimetričnih algoritama u praksi se obje vrste algoritama isprepleću u cilju bolje zaštite poruka. Obično se asimetrični algoritmi koriste za enkripciju slučajno generisanog broja koji služi kao ključ za enkripciju originalne poruke metodama simetričnih algoritama. Ovo se naziva **hibridna enkripcija**.

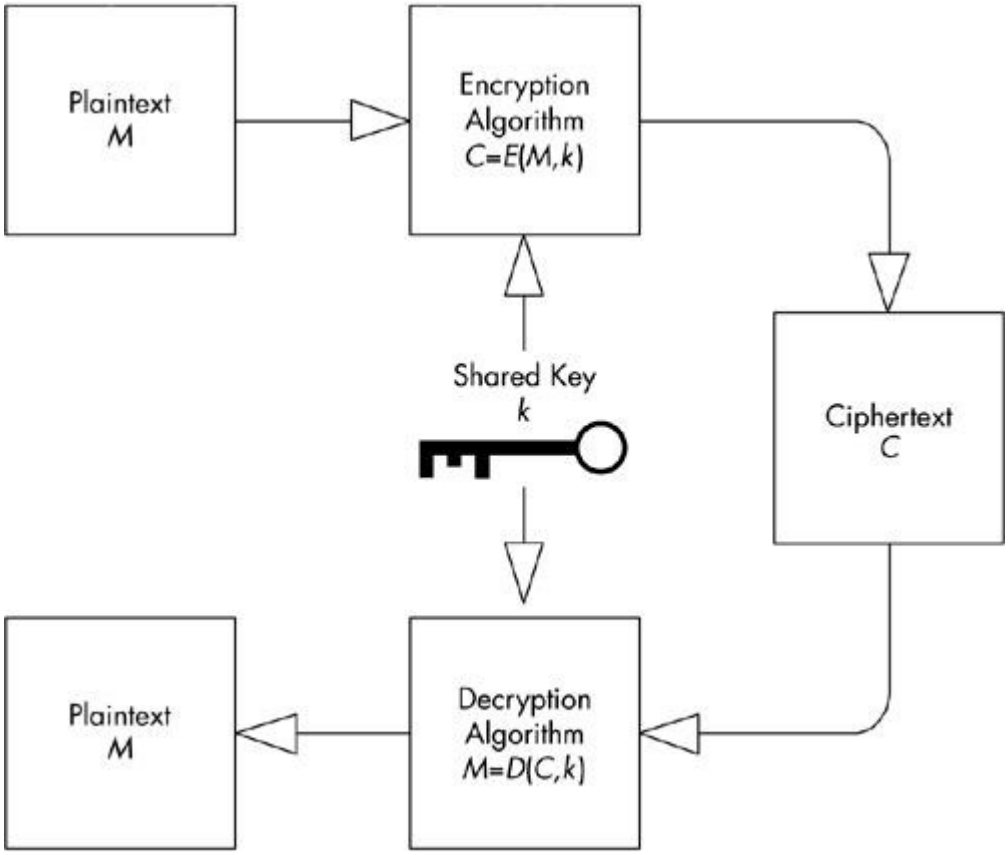
4. Simetrična kriptografija

Simetrična kriptografija je najstariji oblika kriptografije, stara gotovo koliko iljudska komunikacija (naziva i kriptografijom tajnog ključa jer se podatak kriptira idekriptira istim ključem). Za proces kriptiranja u simetričnoj kriptografiji potrebno je znati algoritam kriptiranja i tajni ključ. Nekad su se algoritmi držali u tajnosti, ali se pokazalo da skrivanje algoritmane ne doprinosi sigurnosti. Svi savremeni simetrični algoritmi javno su objavljeni. Zbog toga ih je u potpunosti moguće testirati i provjeriti njihovu otpornost na napade, odnosno moguće ih je analizirati (kriptoanaliza). Sigurnost simetričnih algoritama ovisi o sigurnosti samog algoritma i dužini ključa. Najpoznatiji simetrični algoritam je DES (Data Encryption Standard), koji je razvio IBM-a 1977. godine. Bio je standardni simetrični algoritam sve do 2000. godine kad ga je zamijenio AES (Advanced Encryption Standard), koji rukuje ključevima dužine 128, 192 i 256 bita. Glavni razlog zbog kojeg je DES zamijenjen AES-om je taj što DES ima dužinu ključa od 56 bita. Već smo rekli da je simetrična kriptografija tajnim ključem postupak kojim se koristi jednak ključ za enkripciju i dekripciju podataka. Simetričnu kriptografiju možemo matematički prikazati izrazima:

$$\text{Enkripcija: } C = E_k(M)$$

Dekripcija: $M = D_k (C)$

gdje E predstavlja enkripcijsku funkciju, D dekripcijsku funkciju, k je tajni ključ jedinstven za obje strane, M je originalna (plaintext) poruka, a C je pripadajuća enkriptirana poruka (ciphertext).



Slika 1. Simetrična kriptografija

Način korištenja simetrične enkripcije najlakše je pokazati slijedećim primjerom. Pošiljaoc i primatelj oboje posjeduju zajednički tajni ključ, koji samo oni znaju te su prethodno dogovorili zajednički kriptografski algoritam koji će koristiti. Kada Pošiljaoc želi poslati poruku primatelj - u, on enkriptira originalnu poruku (plaintext) korištenjem tajnog ključa i prethodno dogovorenog algoritma. Time dobija enkriptiranu poruku (ciphertext) koju dalje šalje primatelj – u . Primatelj prima enkriptiranu poruku (ciphertext) od pošiljaoc – a i dekriptira ju svojim privatnim ključem kako bi opet dobio originalnu poruku (plaintext). Ukoliko netko prisluškuje njihovu komunikaciju, prima samo enkriptiranu poruku, jer je jedino ona slana preko otvoreno kanala tako da je tajnost komunikacije očuvana. Mana simetrične enkripcije je što se podrazumjeva da su se dvije strane pošiljaoc i primaoc unaprijed dogovorili o vrijednosti enkripcijsko/dekripcijskog ključa koji mora ostati u tajnosti od neautoriziranih korisnika. Kod takvog prijenosa gdje se koristi jedinstveni ključ je također moguć i tzv.

napad sirovom silom ili brute-force attack. koji podrazumjeva isprobavanje svih mogćih kombinacija tajnog ključa sve dok se ne pronađe korištena kombinacija.

4.1 Simetrični algoritmi

4.1.1 Lucifer

Lucifer je prvi simetrični algoritam za kriptiranje kojeg je osmislio Horst Fiestel, razvijen od strane IBM – a u ranim sedamdesetima. Prethodnik je DES – a i mnogo je jednostavniji od njega.

Činjenice:

- prvi simetrični algoritam s blok šifriranjem
- prethodnik DES-a
- enkriptira blok veličine 128 bita
- koristi ključ veličine 128 bita
- 16 podključeva dužine 72 bita
- koristi 16 'Feistel runda' (iteracije) kod enkriptiranja
- dekripcija se vrši inverznom enkripcijom

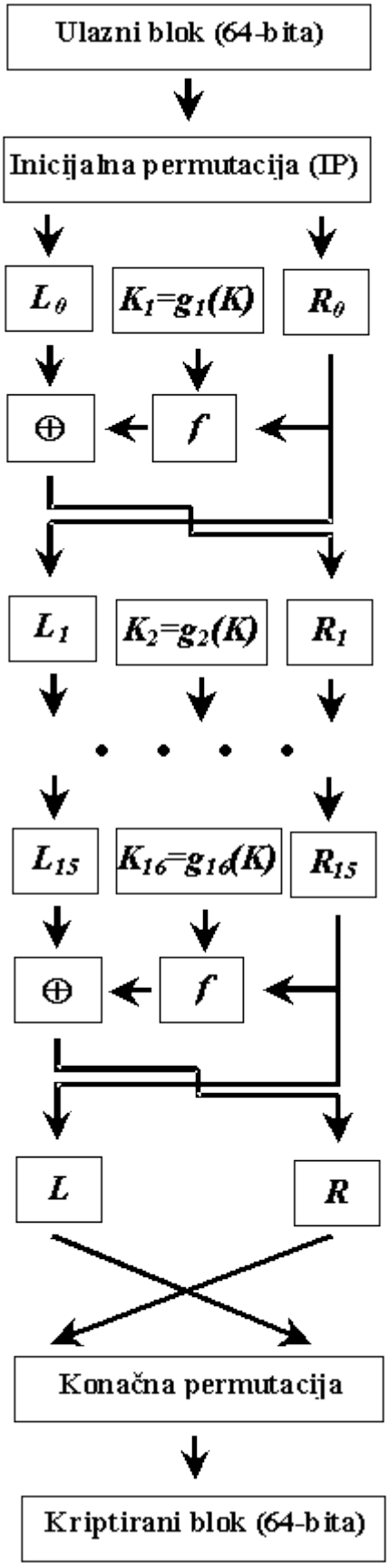
Slabosti:

- slabosti u korištenju ključa (key scheduling)
- slab je na napade diferencijalne kriptoanalize

Danas se smatra nesigurnim, no zbog dužine ključa, te brzine enkriptiranja može se koristiti za enkriptiranje u kombinaciji s nekim dobrim simetričnim algoritmom kao što je DES.

4.1.2 DES (Data Encryption Standard)

DES (Data Encryption Standard) je simetrični enkripcijski algoritam razvijen sredinom 70-tih u IBM-u, a prihvaćen kao federalni standard u SAD-u u kasnim sedamdesetima te početkom osamdesetih. 1981. ANSI je potvrdio DES kao ANSI standard (ANSI X3.92 Data Encription Standard).DES predstavlja kriptiranje koje transofmiše 64 bitne blokove podataka u 64 bitne kriptirane blokove podataka. Dužina ključa kriptiranja je 64 bita, od kojih 8 otpada na proveru pariteta, tako da je efektivna dužina ključa 56 bita.DES kriptiranje/dekriptiranje se sprovodi u nekoliko koraka . Prvo se bitovi ulaznog bloka dužine 64 bita permutiraju nekom permutacijom IP. Tada se ulazni blok podieli na dva dela po 32 bita, levi L_0 i desni deo R_0. Nad desnim blokom se obavlja funkcija $F(R_i,K_i)$, odnosno $F(R_i,K_{16-i+1})$ kod dekriptiranja, gde je R_i desnih 32 bita, a K_i je 48 bitni ključ koji se generira iz zadanog tajnog ključa kriptiranja. Vrednost dobijena operacijom XOR između vrednosti funkcije F i levih 32 bita podataka, postaje R_{i+1}, tj. desnih 32 bita za sledeći korak iteracije. L_{i+1} za slijedeći korak je R_i. Nakon 16 takvik koraka blokovi se zamenjuju te se spajaju i obavlja se konačna permutacija koja je inverzna početnoj, tj. IP^{-1}. Dobijenih 64 bita su kriptirani blokovi. Budući da se nakon dve uzastopne operacije XOR sa istim brojem dobija početna vrednost, tj. a = (aĹ b)Ĺ b , postupak dekriptiranja može se sprovesti tako da se operacije obavljaju obrnutim redosedom. Zbog simetričnosti algoritma to se postiže tako da se kriptitrani blok pusti kroz isti algoritam sa tom razlikom da se umesto ključa K_i u i-tom koraku upotrijebi ključ K_{16-i+1}. Postupak generisanja šestnaest 48 bitnih ključeva od zadanog, tajnog ključa sprovodi se u nekoliko koraka. Prvo se pomoću zadane tablice permutacije iz ključa generišu dva bloka po 28 bita. Zatim sledi 16 sledećih koraka: svaki se blok rotira u levo za određeni broj bita (u zavisnosti o kojem je koraku reč) te se iz nastalih blokova (2x28) pomoću tablicom zadate permutacije generše ključ K_i, gde je i broj koraka. Funkcija enkripcije F jeste zapravo najkritičniji deo algoritma, tj. upravo zbog njene kompleksnosti ne postoji (barem koliko je za sada poznato) način provaljivanja DES-a (osim grubom računarskom silom). Vrednost funkcije dobija se u nekoliko koraka. Najpre se od ulaznih 32 bita (R_i) proširenjem zadanom tablicom dobija 48 bita. Ta se vrednost zbraja logičkom operacijom XOR sa ključem K_i paralelno nad svakim bitom. Dobijena se 48 bitna vrednost deli na osam delova od po šest bita. Prvi i zadnji bit svakog dela predstavlja adresu reda, a srednja četiri adresu kolone u tablici selekcije, odnosno, pomoću šest određena su četiri bita. Istim postupkom nad svakom šestorkom od ulaznih 48 bita selekcijom dobijamo 32 bita. Tih se 32 bita još permutira zadatom tablicom te se dobija konačna vrednost funkcije F.



Slika 2.DES enkripcija

- nastao od LUCIFER-a, (NBS,IBM,NSA)
- enkriptira blok veličine 64 bita
- koristi ključ dužine 64 bita (56 efektivno)
- broj rundi varijabilan (ovisi o dužini ključa i dužini bloka)
- koristi 16 podključa dužine 48 bita
- koriste se Feistel runde

Zanimljivost vezana uz DES je na žalost i njegova slabost. Naime, zbog načina na koji DES kreira podključeve, postoje 4 ključa za koje je dekripcija jednaka enkricpiji. To znači da ako s tim ključem želimo enkriptirati poruku dvaput, dobili bi smo kao rezultat originalnu poruku. No, vjerovatnost enkriptiranja baš tim ključevima je jako mala pa ne utječe značajno na sigurnost.

4.1.2.1 Probijanje DES-a

DES je nastao početkom 70-ih godina, a odobren je 1977. Može se reći da je kao enkripcijski standard zadovoljio ciljeve (sigurnost) i predviđen vijek trajanja (20-25 godina), no krajem 90-ih (1997), RSA Laboratories obznanjuje **RSA Secret Key Challenge**. Cilj izazova bio je probijanje nekih od najkorištenijih algoritama enkripcije u to doba. Također, pored samog dokaza o ranjivosti današnjih algoritama (DES, RC5), očekivala su se i neka dodatna saznanja koja bi se stekla kroz izazov. Izazov se u početku sastojao od 13 zadataka. Dvanaest od njih su se sastojala od probijanja RC5 algoritma i to različitih duljina ključeva (od 40-128 bitova), dok je jedan zadatak bio probijanje DES-a. Niže je kronološki slijed probijanja algoritama:

- januar, 1997. - RSA izdaje **RSA Secret Key Challenge** ($10,000)
- septembar, 1997. - razbijen 56-bitni **RC5**

nakon 250 dana *brutte-force (exhaustive key search)* napada sa10,000 računara.Projekt se zvao **Bovine RC5 Effort**, grupa koja je vodila projekt zvala se *Distributed.net group*, a korištena metoda povezivanja računara zove se distribuirano mrežno računarstvo. Dosta važan podatak vezan za ovaj način obrade podataka je to da je korišteno samo *idle* vrijeme procesora, tj. koristilo se ono vrijeme dok je procesor bio nezaposlen. Kada bi se posvetilo potpuno vrijeme svih korištenih računara samo ovm zadatku, vrijeme probijanja ključa bilo bi puno kraće.

• 1997. - razbijen 56-bitni **DES**

  - za razbijanje *brutte-force* metodom, bilo je potrebno **96 dana**. Grupa se zvala **Deschall** i korišteno također je distribuirano mrežno računarstvo s 15,000-20,000 računara.

• januar, 1998. - RSA izdaje **DES challenge II** izazov

cilj RSA je bio da dvaput na godinu izda novi izazov za razbijanje DES-a. Po njihovim procjenama, svakom novom uspjelom pokušaju trebalo bi znatno manje vremena za razbijanje.

• februar, 1998. - razbijen 56-bitni **DES**

grupa *Distributed.net* u puno kraćem roku probija DES (**41 dan**). I ovaj put se koristilo *distribuirano mrežno računarstvo* uz ukupno 50,000 procesora. Projekt je nazvan **Monarch** i pretraženo je ukupno 85% 56-bitnog prostora ključa.

• jul, 1998. - razbijen 56-bitni **DES**

drugi u nizu izazova te godine (**DES challenge II-2**) je dobijen od Electronic Frontier Foundation (EFF) organizacije. EFF je kreirala posebno projektirano računalo nazvano **DES Cracker** koje je koštalo $220,000 i koje je probilo DES za **56 sati**. Brzina pretraživanja ovog *custom-made* računara bila je 90 biliona ključeva/sekundi.

• januar, 1999. - razbijen 56-bitni **DES**

na izazov **DES challenge III** odazvali su se opet EFF i Distributed.net grupa, samo ovaj put su ujednili snage. **DES Cracker**, sada uz pomoć distribuiranog mrežnog računarstva koje je objedinjavalo 100,000 PC računara na Internetu, probilo je poruku kodiranu 56-bitnim DES ključem za **22 sata i 15 minuta**. To je bio ujedno i novi rekord u probijanju DES šifre. Brzina pretraživanja DES prostora je bila 245 biliona ključeva/sekundi. Važno je napomenuti da osim brutte-force napada, postoje još neke slabosti u DES-u za koje se sumnja da su namjerno uvedene.

4.1.2.2 Triple DES i 2-Key 3DES

"Triple data" enkripcijski standard koji pojačava standardnu DES enkripciju. To je DES bazirani algoritam, ali koristi 2 ili 3 različita DES ključa. Prvi ključ se koristi za enkriptiranje bloka podataka izvorne poruke. Tako enkriptirana poruka se dekriptira drugim ključem. Normalno je da se dekripcijom sa ovim ključem neće dobiti originalna poruka, već nova šifrirana poruka. Na kraju se razultat dekripcije opet enkriptira, ovaj put ili trećim ključem ili opet prvim. Time se povećao broj kombinacija koje bi eventualni napadač morao probati da bi pronašao ključ. Broj kombinacija se penje (za 2 različita ključa) na 2112, dok za 3 različita ključa čak na 2168 kombinacija. 3-DES (kako ga još nazivaju) rješava problem dužine ključa običnog DES-a, no sa sobom unosi novi problem. Puno je sporiji od običnog DES-a (barem dvaput). To je i jedan od razloga zašto je raspisan natječaj za AES. Preporučen od RSA Security-a.

4.1.3 **AES** (Advanced Encryption Standard)

AES je novi algoritam enkripcije koji je zamjenio DES kao standardni algoritam enkripcije.Zašto AES? Razlog je jednostavan. Naglim razvojem informacijske tehnologije algoritmi koji su nastali prije deset, dvadeset i više godina su zastarjeli u smislu da više ne pružaju dovoljnu sigurnost. Naime, zadnjih dvadeset godina kriptoanaliza (kao i kriptografija) je također profitirala od razvoja računarske moći. Algoritmi kao DES za koje se nekad smatralo da su neprobojni, danas je moguće kompromitirati. Tijek natječaja za AES

Kako je DES prestao udovoljavati sigurnosnim zahtjevima bilo je nužno uvesti novi standard. Početnu ideju za rad na novom kriptografskom standardu nazvanom **AES** (engl. ***Advanced Encryption Standard***) NIST (engl. *The National Institute of Standards and Technology*) objavljuje 2. januara 1997. godine, da bi 12. septembara iste godine i službeno otvorio javni konkurs. 3DES (engl. *Triple DES*) je označen kao privremeni standard do kraja konkursa. Na konkurs se mogu prijaviti samo algoritmi sa sljedećim svojstvima:

- simetrični blokovski algoritmi sa javnim kodom,
- podržavanje veličine bloka od minimalno 128 bita i
- podržavanje veličine ključa od 128, 192 i 256 bita.

Na prvoj AES konferenciji (nazvanoj *AES1*) 20. oktobra 1998. NIST objavljuje prihvaćanje u natječaj 15 kandidata: CAST-256, CRYPTON, DEAL, DFC, E2, FROG, HPC, LOKI97, MAGENTA, MARS, RC6™, Rijndael, SAFER+, Serpent te Twofish.

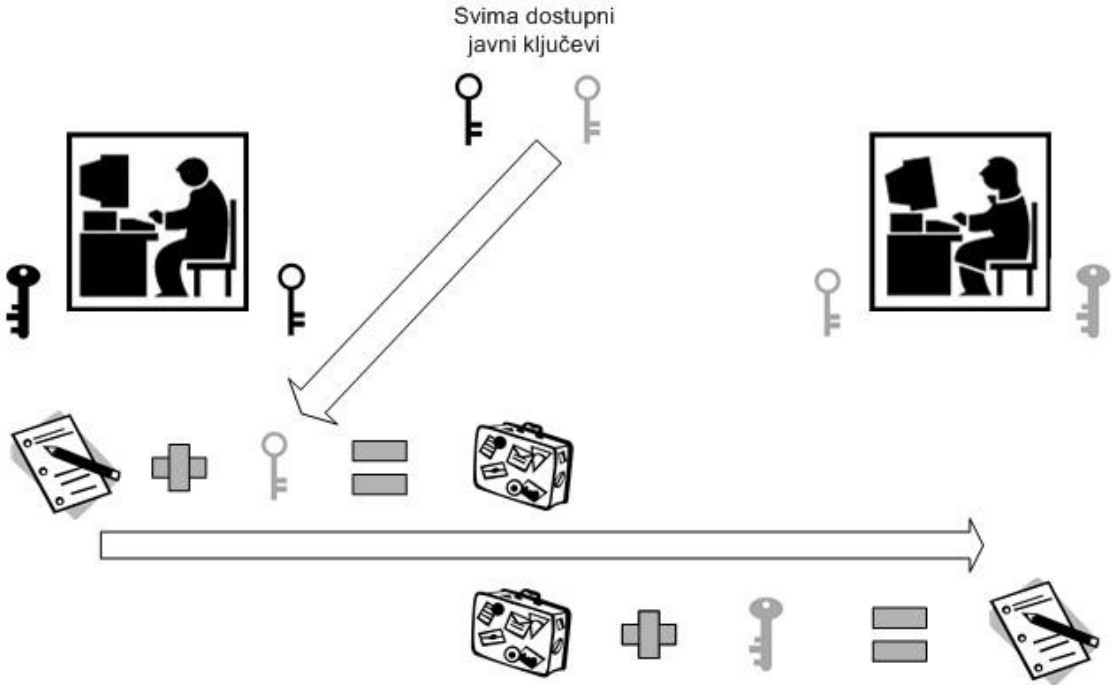
Na istoj konferenciji NIST traži javne komentare na pristigle algoritme i u tu svrhu otvara i službene stranice te forum gdje ljudi iz cijeloga svijeta mogu vidjeti kodove algoritama i sudjelovati u javnim raspravama i analizama algoritama. Svi pristigli komentari su diskutirani i analizirani na drugoj konferenciji (*AES2*) održanoj u martu 1999. Na temelju komentara, kritika i analiza 20. kolovoza 1999. odabrano je pet finalista: **MARS**, **RC6™**, **Rijndael**, **Serpent** te **Twofish**. Na trećoj AES konferenciji (*AES3*) održanoj u aprilu 2000. nastavlja se sa javnom analizom finalista sve do 15. marta 2000. godine, kada se za novi standard odabire AES.

5.Asimetrična kriptografija

Utemeljitelji asimetrične kriptografije su W. Diffie i E. Hellman koji su 1976. godine opisali ideju kriptografije koja se temelji na dva ključa, privatnom i javnom ključu. Razlik asimetričnih i asimetričnih algoritama je u tome što simetrični algoritmi koriste isti ključ za kriptiranje i dekriptiranje dok asimetrični algoritmi koriste različite ključeve za kriptiranje I dekriptiranje. Informacije kriptirane javnim ključem mogu se dekriptirati samo privatnim ključem odnosno to može samo osoba koja je vlasnik tajnog asimetričnog ključa. Osim toga kriptiranje javnim a dekriptiranje tajnim ključem pokazalo se također kao odlično svojstvo i omogućava digitalno potpisivanje informacija gdje potpis može biti provjeren javnim ključem od bilo koga.Ključevi trebaju biti povezani jednosmjernom funkcijom. Odnosno ne smije se moći izračunati privatni ključ iz javnog ključa ili se barem ne smije moći izračunati u razumnom vremenu.Asimetrični kriptosistemi zasnivaju se na određenim svojstvima brojeva koji spadaju u teoriju brojeva.Pri kriptiranju se razgovjetni tekst kodira kao niz prirodnih brojeva koji se odabranom funkcijom kriptiranja i ključem kriptiranja *Ke* preračunavaju u niz brojeva kriptiranog teksta.Funkcija kriptiranja mora biti takva da iz niza brojeva kriptiranog teksta napadač samo s velikim naporima može odrediti izvorni niz brojeva.Međutim,poznavajući ključ dekriptiranja *Kd* omogućuje lako izračunavanje izvornog niza brojeva. Asimetrično kriptiranje,slika 3.2,predstavlja složeniji vid zaštite podataka. Za njegovu realizaciju potrebna su nam dva ključa kod svakog od sugovornika. Jedan ključ je dostupan svima preko javnih kataloga ili imenika, te se zbog te osobine i naziva *javni ključ*. Drugi ključ poznat je samo vlasniku i naziva se *tajnim*.Iako su različiti, ključevi su međusobno povezani određenim transformacijama. Ako ponovo pogledamo prethodni primjer, sada je situacija bitno drukčija:Pero šifrira poruku Ani upotrebom njenog javnog ključa koji je svima dostupan. Mogao ga je dobiti putem email-a, preuzeti sa njenog Web sajta i sl. Bilo tko tko presretne ovu komunikaciju i pored toga što poznaje Anin javni ključ nemože otkriti sadržaj poruke. Poruku može dešifrirati samo Ana korištenjem svog tajnog ključa.Na ovaj način poruka je zaštićena od trećeg lica koji je prilikom presretanja šifrirane poruke onemogućen u

njenom dešifriranju jer mu je za to poreban ključ kojeg strogo u tajnosti čuva ciljni sugovornik. Glavne mane ovog kriptiranja su njegova sporost I neprikladnost za šifriranje velikih količina podataka.

Slika 3 Asimetrično kriptiranje



Često korišteni asimetrični algoritmi: RSA (Rivest-Shamir-Adleman), Diffie-Hellman, te ostali: ElGamal, Rabin, Eliptic Curves.Ovaj sistem predstavlja rješenje za prva dva uslova koja smo na početku ovog teksta postavili - zaštitu tajnosti informacija i očuvanje njihovog integriteta.Ostaje otvoreno pitanje kako da Ana bude sigurna da je poruku koju je primila uistinu poslao Pero.Osiguravanje autentičnosti informacija tj. definiranje i provjera identiteta pošiljaoca postiže se upotrebom digitalnih potpisa i digitalnih certifikata.

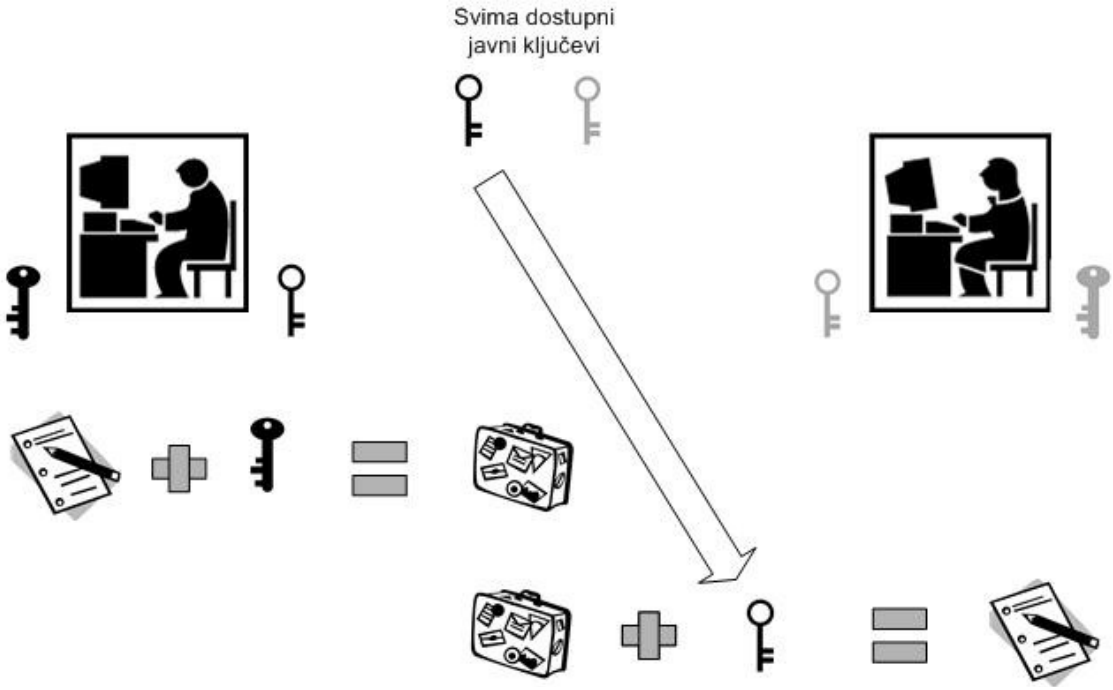
5.1 Digitalni potpis

Tehnologija digitalnog potpisa također koristi tehniku asimetričnog kriptiranja.
Dakle, pošiljatelj i primatelj imaju par ključeva od kojih je jedan tajni, a drugi svima dostupan javni ključ. Klučevi predstavljaju matematičke algoritme koje je izdalo certifikacijsko tijelo

.

Slika 4 Digitalni potpis

.



Svrha digitalnog potpisa je da potvrdi autentičnost sadržaja poruke ili integritet podataka (dokaz da poruka nije promjenjena na putu od pošiljatelja do primatelja), kao i da osigura garantiranje identiteta pošiljatelja poruke. Osnovu digitalnog potpisa čini sadržaj same poruke. Pošiljatelj primjenom određenih kriptografskih algoritama prvo od svoje poruke koja je proizvoljne dužine stvara zapis fiksne dužine (pr. 512 ili 1024 bita) koji u potpunosti oslikava sadržaj poruke.To prakticno znači da svaka promjena u sadržaju poruke dovodi do promjene potpisa.Ovako dobiven zapis on dalje šifrira svojim tajnim ključem i tako formira digitalni potpis koje se šalje zajedno porukom. Da vidimo kako to funkcionira na našem primjeru.Pero kreira digitalni potpis na osnovu poruke koju želi da pošalje Ani. Šifrira ga svojim tajnim ključem i šalje zajedno sa porukom.Ana po prijemu poruke dešifrira Perin potpis njegovim javnim ključem. Zatim primjenom istog postupka kao i Pero i ona kreira potpis na osnovu poruke koju je primila i uspoređuje ga sa primljenim potpisom. Ako su potpisi identični, može biti sigurna da je poruku uistinu poslao Pero (jer je njegovim javnim ključem uspješno dešifrirala potpis) i da je ona stigla do nje nepromjenjena (jer je utvrdila da su potpisi identični).I pored velike sigurnosti koje pruža ova metoda zaštite, i dalje postoji mogućnost prevare. Neko je mogao poslati Ani svoj javni ključ tvrdeći da je Perin, a zatim joj slati poruke za koje bi ona mislila da ih šalje Pero.Rješenje ovog problema pruža upotreba digitalnih certifikata.

5.2 Digitalni certifikati

Ako koristite sistem šifriranja javnim ključem i želite da nekom pošaljete poruku,morate prvo dobiti njegov javni ključ.Međutim, kako možete biti sigurni da je to uistinu njegov ključ? Rješenje ovog problema postiže se upotrebom Digitalnih certifikata. Možemo ih nazvati i digitalnom ličnom kartom, jer oni stvarno to i jesu - digitalna lična karta u syber prostoru, sredstvo kojim ćete vi ili osoba sa kojom komunicirate dokazati identitet na Internetu.Pošto na Internetu nema policije koja bi provjerila vaše podatke i izdala vam ličnu kartu, pojavile su se kompanije koje imaju ulogu 'treće strane', - CA(Certificate Authority) čija je uloga da provjere I utvrde nečiji identitet i nakon toga mu izdaju digitalni certifikat.Kako to funkcionira u praksi,npr. Pero podnosi zahtjev za izdavanje certifikata CA kompaniji. CA provjerava njegov identitet na osnovu ličnih dokumenata koje im je prikazao pri podnošenju zahtjeva. Ako je sve u redu Pero im prosleđuje svoj javni ključ za koji CA kreira digitalni potpis i nakon toga izdaje certifikat kojim se potvrđuje da taj javni ključ uistinu pripada Peri.Ako Pero kasnije želi da komunicira sa nekim, pri prvom kontaktu mu šalje digitalni certifikat i svoj javni ključ. S obzirom da svi poznatiji komunikaciski programi u sebi već imaju uključene javne ključeve CA kompanija kojima se vjeruje, primaoc po prijemu ove poruke lako utvrđuje validnost Perinog certifikata.Ovdje je opisan samo jedan dio primjene digitalnih certifikata. Ako želite da na vašoj Web prodavaonici omogućite kupcima plaćanje kreditnim karticama ili prodaju i pružanje povjerljivih informacija, vaš Web server (server na kome se nalazi vaša prezentacija) mora imati mogućnosti da radi kao Secure Web server.Neophodan uslov za sve ovo je da zatražite i dobijete digitalni certifikat za vaš server od nekog CA.Digitalni certifikat vašeg servera izdat od strane CA mora da sadrži sljedeće:

- Naziv vaše organizacije
- Dodatne podatke za identifikaciju
- Vaš javni ključ
- Datum do koga važi vaš javni ključ
- Ime CA koji je izdao digitalni certifikat
- Jedinstveni serijski broj

Svi ovi podaci formiraju certifikat koji se na kraju šifrira koristeći tajni ključ CA. Ako korisnik ima povjerenja u CA i ima CA javni ključ, može biti siguran u ispravnost certifikata.Velika je vjerojatnost da Web browser koji korisnik posjeduje I već sadrži javni ključ CA jer su Netscape i Microsoft procjenili kojim se CA može najviše vjerovati, pa su njihove javne ključeve uključili u svoje browsere. Najčešće korišteni standard za digitalne certifikate je X.509.

6. Asimetrični algoritmi

6.1 RSA algoritam

Pod pojomom **algoritma** podrazumevamo precizno opisan postupak za resavanje nekog problema. Obicno je to spisak uputstava ili skup pravila kojima je, korak po korak, opisan postupak za resavanje zadatog problema. Svaki korak algoritma odnosno svako upustvo iz spiska mora da bude definisana operacija. Algoritmi moraju da budu nedvosmisleni i da se zavrsavaju u konacnom broju koraka.

RSA algoritam **jedan od najkoriscenijih asimetricnih algoritama danas**.

RSA je skracenica koja je nastala od prezimena njegovih tvoraca: Rona **R**ivesta, Adi **S**hamira i Leonarda **A**dlemana. Svetlost dana ugledao je davne 1977. godine.

U RSA algoritmu kljucnu ulogu imaju veliki prosti brojevi. To su, kao sto znamo, brojevi koji su deljivi samo samim sobom i jedinicom. Prosti brojevi (P i Q) u ovom algoritmu sluze za generisanje javnog i tajnog kljuca i to preko sledecih jednostavnih formula:

Kjavni = P * Q
Ktajni = (2 * (P - 1) * (Q - 1) + 1) / 3

Algoritam kodiranja i dekodiranja sastoji se iz dve formule.

Kodiranje:
Mkodirano = (Mizvorno ^ 3) mod Kjavni

Dekodiranje:
Mizvorno = (Mkodirano ^ Ktajni) mod Kjavni
Na primer, hocemo da kodiramo rec "MAJA".
Ona u ASCII formi glasi: 77 65 74 65 (M = 77; A = 65; J = 74; A = 65).
Kao dva prosta broja mozemo uzeti, recimo P = 9839 i Q = 22391. U tom slucaju kljucevi koji ce se koristiti bice:
Kjavni = 220305049 i
Ktajni = 146848547.
Sada primenimo formule za kodiranje (koristeci samo javni kljuc):

(77657465 ^ 3) mod 220305049 = 162621874

Primalac ce primeniti formulu za dekodiranje (koristeci i javni i tajni kljuc):

(162621874 ^ 146848547) mod 220305049 = 77657465

Ono sto je pohvalno za ovaj algoritam je njegova jednostavnost, ali i sigurnost. U sledecoj tabeli dato je vreme u odnosu na duzinu kljuca potrebno da kompjuter brzine 1

MIPS iz javnog kljuca izracuna tajni kljuc (na primer, Pentium I kompjuter ima oko 150 MIPS-a).

Za enkripciju fajlova koriste se kljucevi velicine 1024, 2048 ili 4096 bita.

Duzina kljuca u bitovima i potrebno vreme:
50 - 3.9 h
100 - 74 god
150 - 10^6 god
200 - 3,8 * 10^9 god

6.2 PGP (Pretty Good Privacy)

PGP je hibridni sistem za enkripciju, jer kombinuje i simetricnu i asimetricnu enkripciju. Podaci se pre sifrovanja pakuju, ako je moguce. Ovo je korisno iz dva razloga. Prvi je manja kolicina podataka za prenos. Drugi je dodatna sigurnost, jer se pakovanjem eliminise pojavljivanje slicnih delova u izvornoj datoteci. Mnoge tehnike kriptoanalize iskoriscavaju bas te slicne delove da bi probile zastitu. Naravno, fajlovi koji su ili prekratki za pakovanje ili se ne mogu spakovati dovoljno, ostavljaju se u izvornom obliku. Posle pakovanja, PGP pravi privremeni kljuc, odnosno slucajan broj koji se generise korisnikovim pokretima misa i pritiskanjem tastera, jer su i oni takodje slucajni. Ovaj kljuc ima jednokratnu upotrebu, jer se koristi da bi se podaci sifrovali simetricnom enkripcijom. PGP zatim sifruje **samo** privremeni kljuc asimetricnom enkripcijom i pridruzuje sifrovanim podacima.

Desifrovanje se vrsi suprotnim procesom. Prvo PGP pomocu tajnog kljuca desifruje privremeni kljuc, a njim se onda dalje desifruju podaci.

6.2.1 Zasto PGP koristi hibridnu enkripciju?

Razlog je jednostavan: simetricna enkripcija je oko hiljadu puta brza od asimetricne, ali kod simetricne enkripcije postoji problem prenosa kljuca (ako se presretne kljuc, podaci se mogu desifrovati). Kada se ukombinuju ova dva nacina enkripcije, dobija se zeljeni efekat: brza enkripcija sa sigurnim prenosom kljuca. Kljuc se, dakle, prenosi, ali sifrovan tako da ga samo osoba koja ima tajni kljuc moze desifrovati.

Posto PGP koristi asimetricnu enkripciju, to znaci da ima mogucnost digitalnog potpisivanja dokumenata, uz jednu razliku. Umesto da se ceo dokument sifruje tajnim kljucem i od njega generise potpis, to se radi samo na kontrolnom kodu dokumenta (veoma slicno CRC-u). Bilo kakva promena na dokumentu rezultuje promenom u kontrolnom kodu, samim tim potpis vise nije vazeci, a vi znate da je u pitanju falsifikat. Time se izbegava dupliranje duzine dokumenta, jer se potpis ne generise od celog dokumenta.

7. Kriptoanaliza

Kriptoanaliza upravo je suprotno od kriptografije. To je nauka koja se bavi razbijanjem sifri, dekodiranjem, zaobilazenjem sistema autentifikacije, uopste provaljivanjem

kriptografskih protokola.Znači kriptoanaliza je naučna disciplina koja proučava postupke otkrivanja otvorenog teksta bez poznavanja ključa, te postupke otkrivanja ključa uz poznavanje otvorenih i/ili kriptiranih tekstova, ili uz poznavanje nekih informacijao otvorenim i/ili kriptiranim tekstovima.. Razlicite tehnike kriptoanalize nazivaju se napadi. Napadi na sigurnost se mogu razdvojiti u pasivne i aktivne napade. Pasivnimnapadom se samo prisluškuje poslana poruka. Pasivni napad je puno težedetektirati nego aktivni napad. Aktivni napad uključuje mijenjanje poruke,maskiranje, ponovno slanje i DoS ('Denial of Service') napade.

Vrste napada:

- Napad poznatim šifriraniim tekstom: ovaj napad je najteži. Kriptoanalitičarpoznaje samo algoritam kriptiranja i kriptirane tekstove.
- Napad poznatim otvorenim tekstom: za dani kriptirani tekst kriptoanalitičarpoznaje odgovarajući otvoreni tekst ili njegov dio. Kriptoanalitičar zna algoritam i 1 do N otvorenih i kriptiranih tekstova, ali ne zna ključ.
- Napad odabranim otvorenim tekstom: kriptoanalitičar odabire otvorenitekst i kriptira ga. Ovaj napad omogućuje pronalaženje slabosti ualgoritmu. Kriptoanalitičar zna algoritam, kriptirani tekst i 1 do N odabranihparova otvorenih i kriptiranih tekstova, ali ne zna ključ.
- Napad odabranim kriptiranim tekstom: kriptoanalitičar može odabratikriptirani tekst i na neki način ga dekriptirati i dobiti otvoreni tekst. Takođermože odabrati neki otvoreni tekst po svojoj želji. Kriptoanalitičar zna algoritam, kriptirani tekst i 1 do N navodnih kriptiranih tekstova sa otvorenim tekstovima, ali ne zna ključ.
- Adaptivan napad odabranim otvorenim tekstom: kriptoanalitičar koristinapad odabranim otvorenim tekstom. Rezultati napada se koriste za odabir nekog drugog otvorenog teksta. Ovim načinom moguće je unaprijediti napad. Ovaj napad poznat je pod nazivom "diferencijalna kriptoanaliza". Kriptoanalitičaru je poznat algoritam, kriptirani tekst,dabrani otvoreni tekst sa kriptirani tekstom, te odabrani kriptirani tekst sa otvorenim tekstom.
- Birthday attack: ovaj napad je dobio ime po paradoksu rođendana
- Meet in the Middle: napad je sličan birthday napadu, osim štokriptoanalitičar može provjeriti sjecište između dvaju skupova, a ne moračekati pojavljivanje vrijednosti dvaput u jednom skupu.
- Napad korištenjem srodnih ključeva: u ovom napadu pretpostavlja seznanje o odnosu između ključeva u dva različita kriptiranja. Napad možeotkriti slabosti u postupku generiranja podključeva.
- Djelomično znanje o ključu: napadač posjeduje djelomično znanje o tajnom kljluču (npr. zbog “rupe” u postupku generiranja podključeva). U dobrim kriptosustavima, djelomično znanje o ključu ne bi trebalo olakšati pronalaženje ostatka ključa. Ako nije tako, lakše je izvesti iscrpno pretraživanje.

Prema količini i kvaliteti otkrivenih tajnih informacija može se klasificirati uspjeh kriptoanalize

1. Potpuno probijanje ("Total break") - napadač otkriva ključ.

2. Globalna dedukcija ("Global deduction") - napadač otkriva funkcijski

ekvivalent algoritma za kriptiranje i dekriptiranje, ali ne nalazi ključ.

3. Lokalna dedukcija ("Instance (local) deduction") - napadač otkriva dodatne otvorene tekstove (ili kriptirane tekstove), nepoznate od prije.

4. Informacijska dedukcija ("Information deduction") - napadač dobiva Shannon-ove informacije o otvorenim tekstovima (ili kriptiranetekstovima), nepoznatih od prije.

5. Algoritam koji omogućuje razlikovanje ("Distinguishing algorithm") -napadač može razlikovati kriptirane tekst od slučajne permutacije.

7.1 Osnovna pravila zastite

Skoro svi programi za enkripciju umesto brojeva kao kljuca, koriste niz slova i brojeva, tj. lozinku. Svi ovi algoritmi su u velikom stepenu sigurni, bilo da se radi o simetricnim ili asimetricnim, ali postoji sansa da se lozinka otkrije ako se ne pridrzavamo nekih pravila pri njenoj izradi.

Idealna kombinacija zastite je kombinacija hardver-softver i pridrzavanje dole nevedenog pravila o vise niza slucajnih slova i brojeva prilikom smisljanja lozinke.

Evo nekoliko pravila kojih bi trebalo da se pridrzavamo prilikom smisljanja lozinke:

- **Idealno je da se za lozinku uzme niz slucajnih slova i brojeva**. Pri tom bi trebalo da se koristi vise od jednog niza slucajnih slova i brojeva. Oakve lozinke mnogi izbegavaju jer ih smatraju teskim za pamcenje i upotrebljavaju reci iz svakodnevnog govora. tom slucaju potrebno je pridrzavati se sledecih pravila:
- **Ne koristiti reci koje se lako mogu pogoditi**: na primer, godinu rodjenja, devojacko prezime supruge (ili svoje, ako je u pitanju korisnik zenskog pola) , ime deteta, supruznika, roditelja, bliskog rodjaka, psa/macke/kanarinca ili nekog drugog kucnog ljubimca itd.
- **Trebalo bi koristiti vise od jedne reci**. Izbegavajte upotrebu reci iz recnika u neizmenjenom obliku. Korisno je upotrebiti barem kombinaciju jedne reci sa nekim brojem.
- **Budite inventivni**. Najbolja lozinka moze biti deo citata iz neke knjige ili neka besmislena recenica.

8. Zaključak

Kada su se pojavile računarske mreže, nije se mnogo vodilo računa o njihovoj sigurnosti, jer su se tada uglavnom koristile za razmjenu elektronske pošte između istraživača sa raznih univerziteta ili za štampanje dokumenata u kompanijama koje su posjedovale više računara koji su preko mreže bili povezani na jedan zajednički štampač.

Danas, kada su računarske mreže dostupne svakome, njihovoj sigurnosti se posvećuje velika pažnja. Da nije toga, u današnje vrijeme se ne bi moglo kupovati preko mreže niti obavljati razne bankarske transakcije. Ne mali je broj ljudi koji na razne načine pokušavaju da naruše sigurnost mreža, želeći time da naude drugim ljudima i da prvenstveno steknu neku korist od toga. Oni pokušavaju da čitaju ili da mijenjaju sadržaj poruka dostupnih preko mreže, iako nisu ovlašćeni za to. Tako, na primjer:

- učenik ili student će, čitajući tuđe e-mail poruke, sebi prekratiti vrijeme,
- haker će testirati nečiji sigurnosni sistem pokušavajući da mu ukrade podatke,
- poslovni ljudi će pokušati da otkriju planove konkurentne kompanije,
- otpušteni radnik će probati da se poslodavcu osveti zbog otpuštanja,
- računovođa će pronevjeriti novac od kompanije,
- lopov će pokušati da ukrade broj tuđe kreditne kartice da bi pomoću nje kupovao,
- špijun će pokušati da sazna sa kakvom vojnom silom raspolaže neprijatelj

Svi ovi primjeri ukazuju na to da je neophodno stvoriti inteligentne mehanizme koji će računarsku mrežu učiniti sigurnom. Zadovoljavajući rezultati su postignuti upotrebom raznih algoritama šifrovanja, kojima se podaci maskiraju i tako bivaju osigurani za prenos. Korišćenjem protokola za autentifikaciju, osigurava se sigurna komunikacija između dvije strane. Osim toga, pojavom digitalnih potpisa i SSL protokola omogućeno je da se razne bankarske transakcije i razmjena dokumenata od velike važnosti obavljaju na siguran način

9. LITERATURA

1.Internet

2.William Stalings: *"Cryptography and Network Security: Principles and Practice"*

3 Internet stranice Zavoda za telekomunikacije Fakulteta elektrotehnike i računarstva:

4. Internet stranice o računalnoj sigurnosti Zavoda za elektroniku, mikroelektroniku, računalne i inteligentne sustave Fakulteta elektrotehnike i računarstva:

5. AES home page: http://www.nist.gov/aes

6. Internet stranice Fakulteta Organizacije i Informatike: