Šifra zasnovana na generatoru pseudoslučajnih brojeva

Luka Milinković

***Sadržaj* — U radu je prikazana ideja simetričnog kriptosistema za šifrovanje podataka, zasnovanog na generatoru pseudoslučajnih brojeva. To je neblokovska šifra, koja uz korišćenje istog ključa omogućava različito šifrovanje različitih poruka. Dužina ključa može da bude preko hiljadu bita, a bezbednost veća nego kod trenutno, komercijalno dostupnih algoritama.**

***Ključne reči* — Ključ, kriptogram, pseudoslučajan generator, šifra, tajnost, zaštita.**

I UVOD

Prve metode šifrovanja, koje su se zasnivale na permutaciji slova u tekstu, koristili su stari Grci, još u 5. veku pre nove ere [1]. Od tada pa do danas šifrovanje je drastično napredovalo zahvaljujući kriptoanalitičarima koji se trude da svaku novu šifru što pre „razbiju“. Tako je do pre nešto više od 10 godina najsigurniji algoritam za šifrovanje bio DES (*Data Encryption Standard*) [2], koji je u upotrebu uveden još 1977. godine. To je prvi savremeni algoritam za komercijalnu upotrebu sa potpuno objavljenom specifikacijom. Realizovan je kao blok šifra sa simetričnim ključem, kod koga se tajnost podataka zasniva na ključu dužine 56 bita. Povećavanje broja operacija u sekundi koje jedan računar može da izvrši, čime se ubrzava „razbijanje“ kriptograma, doprinelo je da ovaj algoritam bude skoro neupotrebljiv za današnje šifrovanje podataka. Zamena DES-u je danas aktuelni standard, američke vlade, AES (*Advanced Encryption Standard*) [3]. Ovo je, takođe, blok šifra sa simetričnim ključem i njegov algoritam je detaljno objašnjen i javno dostupan, pa se i ovde bezbednost šifrovanih podataka zasniva na dužini ključa.

Kod AES-a se za šifrovanje koriste 3 dužine ključa i to od: 128, 192 i 256 bita, a moguće je promenom algoritma i povećavanjem njegove složenosti koristiti i duže ključeve. Program za šifrovanje opisan u ovom radu, čija se bezbednost, takođe, zasniva na dužini ključa, jer se podrazumeva da algoritam bude javno dostupan, je simetričan kriptosistem kod koga se za istu složenost i isti algoritam koriste sve dužine ključa od 156 do 1800 bita. Korak između ključeva kod ovog algoritma je samo 1 bit, a kod AES-a je 64 bita. Ova osobina AES-a drastično smanjuje ukupan broj različitih ključeva koji mogu da se koriste, u čemu je i prednost priloženog algoritma.

Luka B. Milinković, dipl. inž., Elektrotehnički fakultet u Beogradu, Srbija; telefon: +381(0)64/242-7645; e-mail: lukaui@yahoo.com

Šifra zasnovana na generatoru pseudoslučajnih brojeva nije blokovska kao kod AES-a, pa na šifrovanje utiče i ključ i ukupna dužina poruke koju treba šifrovati. Ovo omogućava realizaciju algoritma koji će različito šifrovati poruke različitih dužina koristeći isti ključ što se ne može postići kod blokovskih šifri. Ako je potrebno da šifra ipak bude blokovska, na primer, zbog dodavanja koda za ispravljanje grešaka kao što je CRC32 (Cyclic Redundancy Check) [3], može se algoritam realizovati tako da šifruje blokove, koji ne moraju da budu kraći od, na primer, 1000 bita, što je opet duže nego blokovi kod AES-a. Takođe, blokovi ne moraju da budu ni uvek iste dužine, već se mogu menjati u zavisnosti od dužine poruke.

II OPIS ALGORITMA

A. Uvod

Šifra zasnovana na generatoru pseudoslučajnih brojeva, prvenstveno je namenjena za šifrovanje poruka, koje se koriste za komunikaciju između dve osobe kroz nesigurne kanale [3]. Pored toga može se koristiti i za šifrovanje bilo kojih drugih podataka koji se čuvaju na disku.

Algoritam se može realizovati kao zaseban program ili kao dodatak programima koji se već koriste za komunikaciju, kao što su Outlook Express, Mozilla Tunderbird, Yahoo mail i drugi programi za slanje elektronske pošte.

B. Karakteristike generatora pseudoslučajnih brojeva

Algoritam za šifrovanje se zasniva na generatoru pseudoslučajnih brojeva, koji se koristi u Matlab-u [4]. Ovaj generator može da se inicijalizuje, a to znači da će za istu zadatu vrednost generisati uvek iste brojeve, čije će vrednosti biti između 0 i 1. Svi elementi generisani pomoću jednog inicijalizacionog broja su različiti. Ako se inicijalizacioni broj promeni generisaće se elementi sa novim vrednostima. Inicijalizaciona vrednost može da bude bilo koji broj u intervalu između $-(2^{32}-1)$ i $(2^{32}-1)$, što je 2^{33} različitih brojeva. Niz koji se generiše može da sadrži najviše 10^8 elemenata.

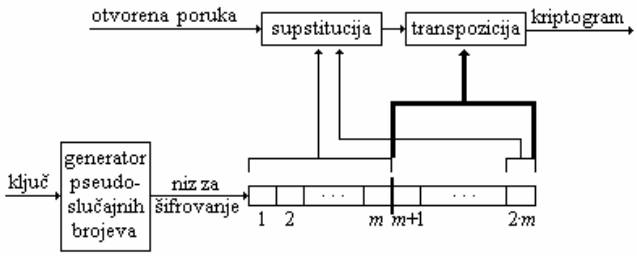
C. Metode šifrovanja

Šifrovanje koje se primenjuje u ovom algoritmu se obavlja nad bitima, pa je potrebno karaktere iz otvorene poruke [2] prevesti u bite. Ovo se obavlja tako što svaki karakter dobija odgovarajuću osmobitnu reč iz ASCII (*American Standard Code for Information Interchange*) tabele [5]. Na ovaj način se formira niz bita otvorene poruke, čija je dužina *m*.

Nad ovim bitima se prvo obavlja supstitucija, a zatim transpozicija – permutacije. Supstitucija se primenjuje tako što se nad bitima otvorene poruke i bitima generisanim pomoću pseudoslučajnog generatora izvršava operacija XOR – „ekskluzivno ili“ [2], a transpozicija tako što se novodobijeni biti ispremeštaju po nekom novom redosledu, koji je određen pseudoslučajnim generatorom.

I jedna i druga metoda za šifrovanje imaju svoje prednosti i mane. Supstitucija je dobra, jer menja vrednosti niza bita otvorene poruke i formira novi niz sa novim brojem nula i jedinica. Ovo sprečava bilo kakvu statističku analizu kriptograma da bi se došlo do otvorene poruke. Problem je to što ako kriptoanalitičar ima otvorenu poruku i šifrovani tekst znaće koji je niz bita generisan. Dobra strana permutacija je ta što kada kripoanalitičar ima otvorenu poruku i njen kriptogram, ne može da zna kako su ispremeštani biti i to neće moći da odredi bez ključa. Nedostatak ove metode je to što broj nula i jedinica u kriptogramu ostaje nepromenjen u odnosu na otvorenu poruku. Kombinacija ove dve metode je odlična za šifrovanje, jer jedna ispravlja mane druge i dobija se algoritam koji potpuno narušava statističku zavisnost između bita otvorene poruke i raspoređuje ih na potpuno slučajan način. Naravno, slučajan toliko da se pomoću pravog ključa može odrediti.

Na sl. 1 je prikazan dijagram sa kojeg se može videti redosled šifrovanja u opisanom algoritmu, a metode supstitucije i transpozicije su detaljno opisane u nastavku i na primeru u trećem poglavlju.



Sl. 1. Dijagram šifrovanja

Pomoću generatora pseudoslučajnih brojeva generiše se niz za šifrovanje, koji je duplo duži od niza bita otvorene poruke. Na osnovu poslednjeg elementa niza za šifrovanje i prve polovine tog niza formiraju se biti za supstituciju. Poslednji element određuje verovatnoću (od 0 do 1) pojavljivanja nula, pa sada najmanji elementi prve polovine niza za šifrovanje dobijaju vrednost nula dok se ne zadovolji ova verovatnoća. Preostali elementi prve polovine niza dobijaju vrednost jedan. Ovako je formiran niz za supstituciju sa slučajno raspoređenim nulama i jedinicama.

Da bi se formirao niz za permutacije potrebna su dva niza. Prvi niz je, ustvari, druga polovina niza za šifrovanje, koji se sastoji od slučajno raspoređenih brojeva između 0 i 1. One se sortiraju od manje ka većoj i formiraju drugi niz. Sada se formira niz za permutacije tako što se na pozicije elemenata iz prvog niza upisuju njihove pozicije iz drugog niza. Dobijeni niz za permutacije se sastoji od slučajno raspoređenih celih, međusobno različitih brojeva od 1 do m, tako da se svaki broj pojavljuje tačno jednom. Biti dobijeni posle supstitucije se numerišu, redom od 1 do m i to su redni brojevi bita u posmatranom nizu. Sada se ovi biti raspoređuju na nove pozicije tako što se svaki od njih postavlja na poziciju svog rednog broja u nizu za permutacije.

D. Ključevi i podključevi

Ključ se sastoji od ASCII karaktera, a ukupan broj različitih karaktera koji se koriste i koji se mogu uneti preko tastature je 96 – mala i velika slova engleske abecede, cifre i interpunkcijski znaci zajedno sa razmakom i enterom [5]. Svaki karakter se predstavlja sa 6 bita i na osnovu toga se formira binarna vrednost ključa. Neki karakteri imaju iste binarne vrednosti pa se broj različitih ključeva određuje na osnovu broja različitih kombinacija bita, a ne na osnovu broja različitih kombinacija karaktera u ključu.

Minimalna dužina ključa je 156 bita, odnosno 26 karaktera, a maksimalan broj bita koji se koristi za inicijalizacionu vrednost u pseudoslučajnom generatoru je 33 bita. Zbog toga se od svakog ključa formiraju podključevi, a svaki od njih generiše jedan deo elemenata potrebnih za šifrovanje. Svi ti elementi se kombinuju, posebno za supstituciju, a posebno za permutacije i formiraju se dva niza tako da je svaki dugačak koliko i niz bita otvorene poruke.

Posmatrajmo da se podključ sastoji samo od inicijalizacionog broja i da je njegova najveća dužina 33 bita. Na primer, ako treba da se generiše 400 elemenata broj različitih kombinacija i vrednosti elemenata ako menjamo podključeve je 2^{33}. Uzeto je 400 elemenata, jer se pomoću njih šifruje 200 bita otvorene poruke, a to je minimalan broj bita koje će da šifruje jedan podključ. Broj različitih kombinacija bita na 400 pozicija je 2^{400}, što je daleko veće od 2^{33}, a ovo je bitno za generisanje niza bita za šifrovanje supstitucijom. Ako posmatramo 400 različitih celih brojeva slučajno raspoređenih na 400 pozicija broj kombinacija je $400!\approx 2^{2886}$, što je mnogo veće od 2^{33}, a ovo je bitno za generisanje niza celih brojeva koji se koristi za šifrovanje transpozicijom. Iz ovoga se vidi da postoji daleko više kombinacija bita, za šifrovanje supstitucijom, i mnogo više kombinacija celih brojeva, za šifrovanje permutacijama nego što se može postići pomoću inicijalizacione vrednosti. Zbog toga se podključ ne sastoji samo od inicijalizacionog broja već i od težinskog faktora i obaveznih bita, jer oni povećavaju broj kombinacija koje se mogu ostvariti podključem.

Svaki podključ mora da sadrži 8 bita težinskog faktora, 4 obavezna bita i najmanje 3, a najviše 33 bita za inicijalizacionu vrednost. Prvi bit koji određuje inicijalizacionu vrednost određuje znak, – ili +, a preostali broj u dekadnom zapisu. Težinski faktor može da ima 256 različitih vrednosti i to vrednosti između 256 i 511. On određuje koliko će koji podključ da generiše elemenata

$$p_i = \frac{t_i}{\sum_{k=1}^{n} t_k} \cdot 100\%, \qquad (1)$$

gde je p_i – procenat elemenata koje generiše posmatrani podključ u odnosu na ukupan broj potrebnih elemenata, t_i

– težinski faktor posmatranog podključa, a *n* – ukupan broj podključeva. Iz jednačine (1) se vidi da ne bi valjalo da težinski faktori imaju vrednosti između 1 i 256 da se ne bi desilo da neki podključevi generišu drastično manje elemenata od drugih i da na taj način deo bita iz ključa skoro uopšte ne utiče na šifrovanje.

Obavezni biti utiču na niz koji je generisan pomoću inicijalizacionog broja iz posmatranog podključa. Prvi, da se odredi da li se prva polovina niza koristi za permutacije, a druga za supstituciju (ako je bit 0) ili obrnuto (ako je bit 1). Drugi, da se odredi da li se izabrani deo niza za permutacije posmatra u tom redosledu (0) ili u suprotnom (1). Ako se posmatra u suprotnom onda poslednji član niza postaje prvi, pretposlednji postaje drugi i tako redom. Treći, da se odredi da li da se niz za permutacije ne menja (0) ili da elementi zamene mesta (1). Zamena mesta se vrši tako što najveći i najmanji element menjaju mesta, drugi najveći i drugi najmanji menjaju mesta i tako redom. Ovo je isto kao da se posmatra samo jedan niz i da se u njemu vrši sortiranje od manjeg ka većem (0) ili od većeg ka manjem (1). Ovde se ne može ovako posmatrati, jer se ne sortira niz od svakog podključa pojedinačno već se svi nizovi iskombinuju i novodobijeni niz se sortira. Četvrti, da se odredi da li se niz za supstituciju koristi sa tako formiranim bitima (0) ili sa invertovanim, nule postaju jedinice, a jedinice postaju nule, (1).

Od svakog ključa se formira najmanje 10 podključeva, a najviše 40, ali tako da je minimalan broj bita po podključu 15 (8 za težinski faktor, 4 obavezna i 3 za inicijalizacioni broj), a maksimalan 45 (kada se za inicijalizacioni broj koriste 33 bita). Na osnovu ovih ograničenja i dužine ključa određuje se broj podključeva koje može da oformi posmatrani ključ, ali tako da broj podključeva bude minimalan. To znači da se sa povećanjem broja bita po ključu prvo povećava broj bita po podključu, a tek kada broj bita po podključu dostigne 45 bita povećava se broj podključeva. Svi podključevi treba da imaju ili isti broj bita ili da se međusobno razlikuju za najviše 1 bit. Posmatrano na ovaj način najveća dužina ključa je 1800 bita. Dužina ključa može da bude i veća ako se poveća broj podključeva koje može da formira jedan ključ, što nije problematično sa stanovišta algoritma.

Pošto je korak između ključeva najmanje 6 bita onda se 3 bita od ključa ne dodeljuju podključevima, već se koriste da se odredi koliko će se bita dodati ključu. Na ovaj način ključ gubi 3 bita, ali u zavisnosti od vrednosti ta 3 bita može da dobije od 0 do 7 bita. Ovim postupkom se postiže da korak između ključeva bude 1 bit.

E. Dodavanje slučajnih bita

Otvorena poruka pri komunikaciji može da sadrži nekoliko stotina bita, a to je mali broj bita za pouzdano šifrovanje, jer se može desiti da ključ bude duži od poruke. Da bi šifrovanje bilo pouzdano na kraj niza bita otvorene poruke se dodaju slučajni biti. Između niza bita otvorene poruke i slučajnih bita se stavljaju karakteristični biti koji služe pri dešifrovanju da bi se odredili korisni biti, a odbacili slučajni.

Još jedna prednost dodavanja bita je što se pored šifrovanja korisnih bita šifruju i nekorisni, a koliko će njih biti može se proizvoljno odrediti za svaku novu otvorenu poruku koja se šifruje, jer je za dešifrovanje dovoljno da se znaju karakteristični biti. Ovo dodatno narušava odnos između broja nula i jedinica u kriptogramu i otežava kriptoanalitičarima, jer ne znaju koliko je bita dodato niti koji su to biti. Njima je onemogućeno „razbijanje“ šifre, odnosno ključa, čak i kada znaju i otvorenu poruku i njen kriptogram, jer ne znaju koji biti su korisni, a koji višak, odnosno ne mogu da otkriju gde su biti iz otvorene poruke u kriptogramu.

Rešenje koje je primenjeno kod ovog algoritma je da se svaka poruka kraća od granice, koja iznosi 6000 bita, dopuni do nje. Ovi biti se dodaju da bi se obezbedilo pouzdano šifrovanje. Sada se na otvorenu poruku, koja ima najmanje 6000 bita, dodaje još slučajnih bita da bi se maskirao koristan sadržaj. Ovi biti nemaju veze sa bitima koji se dodaju kao dopuna do granice. Novoformirana otvorena poruka može da sadrži između 30% i 130% više bita nego pre dodavanja. Koliko će se bita dodati određuje se pomoću generatora pseudoslučajnih brojeva bez inicijalizacione vrednosti. Određivanje ovog broja je potpuno slučajno i promenljivo za svaku otvorenu poruku, a razlikovaće se, čak i kod otvorenih poruka iste dužine.

Pri komunikaciji kroz nesigurne kanale slučajni biti doprinose boljoj zaštiti otvorene poruke. Dodavanje slučajnih bita će smanjiti protok korisnog sadržaja, ali ako bi najveća otvorena poruka imala 50000 bita, odnosno 6250 karaktera, što je veoma velika poruka, po najslabijoj liniji, protoka 56kb/s prenosila bi se za vreme od 2s umesto za 1s i to samo onda kada je dodato maksimalno slučajnih bita, a to se retko dešava. I pored dvostrukog smanjenja korisnog saobraćaja, poruka se brzo prenosi.

Pri šifrovanju podataka koji se skladište na disk ne moraju se dodavati slučajni biti da se ne bi zauzimao dodatni prostor na disku, mada bi to sa stanovišta šifrovanja bilo korisno.

III PRIMER: FORMIRANJE KRIPTOGRAMA

Da bi se šifrovala otvorena poruka prvo je potrebno odrediti niz za supstituciju i niz za transpoziciju. Kako to radi algoritam biće objašnjeno na otvorenoj poruci „Fiat“ od 4 karaktera, odnosno 32 bita.

Niz bita otvorene poruke:
01000110 01101001 01100001 01110100.

Ovaj primer služi da na što jednostavniji način objasni metod šifrovanja, a ne i da dokaže kvalitet šifre, pa se zato koristi i jedan podključ koji se sastoji samo od inicijalizacionog broja, čija je vrednost 5 i sam generiše sve potrebne elemente, njih 64.

Brojevi generisani za inicijalnu vrednost 5:

0,864; 0,906; 0,200; 0,859; 0,796; 0,338; 0,729; 0,543;
0,430; 0,396; 0,214; 0,134; 0,883; 0,815; 0,745; 0,976;
0,124; 0,631; 0,931; 0,866; 0,501; 0,505; 0,984; 0,842;
0,451; 0,320; 0,234; 0,768; 0,321; 0,553; 0,917; 0,265;
0,479; 0,512; 0,949; 0,326; 0,059; 0,131; 0,594; 0,880;
0,712; 0,784; 0,387; 0,464; 0,922; 0,504; 0,053; 0,772;
0,844; 0,705; 0,208; 0,268; 0,721; 0,742; 0,857; 0,536;
0,785; 0,679; 0,817; 0,391; 0,861; 0,917; 0,666; 0,397.

Prva polovina niza, 32 elementa, i poslednji element celog niza, 0,397, određuju niz za supstituciju. Broj nula u nizu za supstituciju, N_0, je

$$N_0 = 32 \cdot 0,397 = 12,704 \approx 13 . \qquad (2)$$

Najmanjih 13 brojeva u nizu za supstituciju (markirani su crnim) dobijaju vrednost 0, a preostali vrednost 1. Na osnovu ovoga dobija se niz za supstituciju:

11011011 00001111 01110111 00010110.

Da bi se formirao niz za transpoziciju potrebna su dva niza, koja sadrže po 32 elementa. Prvi niz sadrži drugu polovinu generisanog niza, a drugi niz sadrži iste te elemente samo sortirane od manjeg ka većem.

Prvi niz:

0,479; 0,512; 0,949; 0,326; 0,059; 0,131; 0,594; 0,880; 0,712; 0,784; 0,387; 0,464; 0,922; 0,504; 0,053; 0,772; 0,844; 0,705; 0,208; 0,268; 0,721; 0,742; 0,857; 0,536; 0,785; 0,679; 0,817; 0,391; 0,861; 0,917; 0,666; 0,397.

Drugi niz:

0,053; 0,059; 0,131; 0,208; 0,268; 0,326; 0,387; 0,391; 0,397; 0,464; 0,479; 0,504; 0,512; 0,536; 0,594; 0,666; 0,679; 0,705; 0,712; 0,721; 0,742; 0,772; 0,784; 0,785; 0,817; 0,844; 0,857; 0,861; 0,880; 0,917; 0,922; 0,949.

Sada se formira niz za transpoziciju tako što se na prvu poziciju upisuje redna vrednost broja 0,479 u drugom nizu. U drugom nizu ovaj broj se nalazi na 11. poziciji (markiran crnim), pa je to vrednost koju treba upisati na prvu poziciju niza za permutacije. Na drugu poziciju se upisuje redna vrednost broja 0,512 u drugom nizu (markiran crnim). Kada se ovo sprovede za sve brojeve dobija se niz svih celih brojeva od 1 do 32 ispremeštanih na pseudoslučajan način. On čini niz za transpoziciju:

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

Nad bitima iz otvorene poruke i iz niza za supstituciju se obavlja operacija XOR i formira novi niz bita. Kriptogram je formiran kada se novi biti rasporede na pozicije svojih rednih brojeva u nizu za permutacije. Prvi bit (1) ide na 15. poziciju u kriptogramu, drugi bit (0) na 5. (markirani su crnim) i tako redom.

Šifrovanje
Supstitucija:

```
                01000110 01101001 01100001 01110100
XOR             11011011 00001111 01110111 00010110
                10011101 01100110 00010110 01100010
```

Transpozicija:

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

Kriptogram: 10010010 01011011 10111011 00010000
redni broj bita: 5 15

IV ZAKLJUČAK

Algoritam za šifrovanje opisan u ovom radu ispunjava dva važna uslova: tajnost i autentičnost podataka [3]. Tajnost podrazumeva da kriptoanalitičaru za „razbijanje“ šifre nije dovoljno da zna kriptogram i metod njegovog formiranja, već mora da zna i ključ, a autentičnost podrazumeva da ne može da izmeni šifrovanu poruku, a da se to ne primeti na prijemu.

Pošto na šifrovanje utiču ključ i dužina otvorene poruke omogućeno je pomoću istog ključa različito šifrovanje otvorenih poruka različitih dužina. Još jedna prednost ovakvog šifrovanja i korišćenja više podključeva je ta što svaki niz za supstituciju predstavlja, ustvari, jednokratnu beležnicu, savršenu šemu za šifrovanje, gde se za svako novo šifrovanje koristi novi niz bita, [2]. Algoritmi kod kojih se koristi jednokratna beležnica se ne mogu dekriptovati. Isto to važi i za niz za permutacije. Kod njega je za svako novo šifrovanje raspored celih brojeva jedinstven.

Pseudoslučajni generatori imaju jednu veliku manu, a to je da se na osnovu nekih izgenerisanih vrednosti mogu predvideti naredne [2]. U ovoj šifri to nije mana, jer generator generiše nizove za supstituciju i permutacije iz više delova. Da bi se saznala bilo koja izgenerisana vrednost mora da se zna podključ, jer je nju ne moguće odrediti kriptoanalizom kriptograma. Ako kriptoanalitičar zna podključ, onda zna samo deo niza, koji bez preostalog dela ne znači ništa ni kod supstitucije, jer se ne zna broj nula i jedinica ni njihov raspored, ni kod permutacije, jer se ne zna sortirani niz. Naravno, ni podključ ne može da se sazna bez ključa. Zbog toga su vrednosti generisane pomoću pseudoslučajnog generatora sigurne koliko i sam ključ.

LITERATURA

[1] Internet adresa: http://e.math.hr/vigenere/index.html, Hrvatski matematički elektronski časopis „math.e“

[2] Bruce Schneier, ”Applied Cryptography“, Jonh Wiles & Sons, Second Edition, 1996.

[3] Dušan B. Drajić, „Uvod u teoriju informacija i kodovanje“, Akademska Misao, Drugo izdanje, 2004.

[4] Internet adresa: http://www.mathworks.com, zvanični sajt za softverski paket Matlab

[5] Internet adresa: http://www.wikipedia.org/wiki/ASCII, besplatna enciklopedija opšteg sadržaja dostupna samo na Internetu

ABSTRACT

In this paper the mean theme is one idea about symmetric algorithm for data encryption, which is based on pseudo-random numbers generator and presents unblock cipher. It provides different encryption of the dissimilar plaintext keeping the same key. This unblock cipher has more then one thousand bits long key. Therefore the safety of the algorithm is better then safety of others algorithms which are used commercially in these days.

CIPHER BASED ON PSEUDO-RANDOM NUMBERS GENERATOR

Luka Milinković