



UNIVERZITET SINGIDUNUM
DEPARTMAN ZA POSLEDIPLOMSKE STUDIJE I
MEĐUNARODNU SARADNJU

MASTER RAD

**JEDNA IMPLEMENTACIJA PROTOKOLA ZA
DISTRIBUCIJU KLJUČEVA PREKO JAVNOG
KANALA NA OSNOVU SOPSTVENOG IZVORA
SLUČAJNOSTI**

Mentor:
doc.dr Saša Adamović

Student:
Milomir Tatović
410298/2013



Beograd, 2015.

JEDNA IMPLEMENTACIJA PROTOKOLA ZA DISTRIBUCIJU KLJUČEVA PREKO JAVNOG KANALA NA OSNOVU SOPSTVENOG IZVORA SLUČAJNOSTI

- MASTER RAD -

Sažetak: U ovom radu bavili smo se implementacijom jednog perfektnog protokola za distribuciju kriptoloških ključeva preko javnih kanala. Protokol je dizajniran po osnovnim principima protokola „Satelitski scenario“, koji omogućava razmenu ključa, bez prethodno razmenjene tajne vrednosti između dve strane preko javnog kanala. Naša ekvivalentna šema u odnosu na „Satelitski scenario“, koristi avio saobraćaju (koordinate aviona), gde na osnovu prikupljenih podataka direktno od aviona dobijamo korelisane binarne nizove između svih učesnika komunikacije. Korelisani nizovi predstavljaju osnovu ovog protokola za distribuciju ključeva. Detaljno je prikazana implementacija protokola, evaluacija bezbednosti protokola sprovedena je preko rigorozne teorijsko informacione analize, a sa aspekta performansi, brzina ključa je na zadovoljavajućem nivou za razmenu simetričnih kriptoloških ključeva.

Ključne reči: razmena ključeva, savršeni protokol, satelitski scenario

ONE IMPLEMENTATION OF PROTOCOL FOR KEYS DISTRIBUTION OVER PUBLIC CHANNELS BASED ON OWN SOURCE OF RANDOMNESS

- MASTER THESIS -

Abstract: In this paper, we have dealt with the implementation of a perfect protocol used for the distribution of cryptology keys via public channels. The protocol is designed according to the basic principles of the "Satellite" scenario protocol, which allows key exchange without pre-exchanged secret values between the parties across the public channel. Our equivalent scheme for Satellite scenario uses air traffic data (plane coordinates), in order to get information used for creating correlated binary sequences amongst all participants of communication. Correlated sequences are the foundation of this key distribution protocol. Protocol implementation has been shown in detail. Security evaluation of protocol was made in terms of rigorous theoretical information analysis. In terms of performance we have found that the key velocity is at a satisfactory level for cryptology key exchange.

Keywords: key exchange, perfect protocol, satellite scenario



SADRŽAJ

1. UVOD	1
1.1 CILJ ISTRAŽIVANJA	1
1.2 PREDMET ISTRAŽIVANJA.....	1
1.3 STRUKTURA RADA	3
2. PREGLED U OBLASTI ISTRAŽIVANJA.....	5
3. TEORIJSKE OSNOVE.....	9
3.1 ALGORITMI ZA ZAŠTITU PODATAKA.....	9
3.2 IDEALNI KOMUNIKACIONI KANAL	10
3.3 ALGORITAM SAVRŠENE TAJNOSTI – <i>ONE-TIME PAD</i>	10
3.4 SLUČAJNOST	12
3.5 INFORMACIONE MERE – ENTROPIJA	13
3.6 GENERATORI SLUČAJNIH NIZOVA	13
3.7 STATISTIČKA ANALIZA SLUČAJNIH NIZOVA	18
3.8 KOLMOGOROVA KOMPLEKSNOŠĆ	29
3.9 SISTEMI ZA PRAĆENJE I NAVIGACIJU	29
4. PREGLED PREDLOŽENOG REŠENJA.....	37
4.1 KOMPLETNA ŠEMA PREDLOŽENOG REŠENJA	37
4.2 DIZAJN PROTOKOLA.....	38
4.3 EKSPERIMENTALNO OKRUŽENJE	46
4.4 EVALUACIJA PERFORMANSI	49
4.5 OBLAST PRIMENE.....	52
5. ZAKLJUČAK.....	53
5.1 OSTVARENI DOPRINOSI	53
5.2 PREDLOG ZA BUDUĆI RAD.....	54
LITERATURA	55
SPISAK SLIKA.....	56
SPISAK TABELA.....	56



1. UVOD

1.1 CILJ ISTRAŽIVANJA

Osnovni zadatak u ovom radu je pronalazak ekvivalentne šeme za praktičnu implementaciju protokola „Satelitski scenario“, koji je do sada predstavljen samo na teorijskoj osnovi od autora Ueli Maurera. Ujedno to i predstavlja glavnu hipotezu ovog rada, koju je moguće operacionalizovati preko više manjih hipoteza:

- moguće je na osnovu koordinata aviona, izgraditi sopstveni izvor slučajnosti
- moguće je preko teorije skupova i podskupova postići zahtevani stepen korelisanosti između binarnih sekvenci svih učesnika u protokolu,
- moguće je na ovaj način obaviti bezbednu distribuciju ključeva preko javnog kanala,
- moguće je postići dobre performanse, koje se ogledaju u brzini ključa.

1.2 PREDMET ISTRAŽIVANJA

U današnjem svetu, svetu sa visokim stepenom konektivnosti, neizostavna je potreba za servisima koji će obezbediti tajnost razmenjenih poruka. Ovakvi servisi imaju svoje zahteve, a to je razmena kriptološkog ključa koji će se koristiti u simetričnom algoritmu za šifrovanje podataka. Iz generacije u generaciju ljudi pronalaze sve komplikovanije, a i sigurnije vidove zaštite. Alan Tjuring (engl. *Alan Turing*) uvideo je da se i najbezbedniji sistemi mogu razotkriti uz pomoć kompleksnih mašina, pa tako, pojavom računara bezbednost sistema se meri vremenom potrebnim da ih savremeni računari razotkriju. Ipak, naučnici se trude da i sa neprestanim rastom procesorske moći obezbede kvalitetne šifarske algoritme praćene i kvalitetnim kriptološkim ključevima. Kao deterministička mašina računari lako na kraj izlaze sa predvidljivim podacima, te je to razlog okretanja ka prirodi pri kreiranju kvalitetnih ključeva. Slučajnost koja se nalazi u prirodi predstavlja nedeterministički podatak koji se ne može predvideti i kao takav, idealan je za izvor pri generisanju kriptoloških ključeva. Na taj način obezbeđuje se da i računari sa velikom procesorskom moći zahtevaju neracionalno dugačak vremenski period potreban za razbijanje takvih ključeva. Naravno za kvalitet samih ključeva osim potrebne slučajnosti, veliku ulogu igra i dužina samog ključa, zasnovanim na slučajnosti koji će obezbediti veoma dug vremenski period za njihovo razbijanje. Šifarski sistemi osim kvalitetnih kriptoloških ključeva sastoje se i od matematičkih algoritama. Prema dizajnu algoritama, algoritmi se dele u dve velike grupe, na one koji koriste simetrične i one sa asimetričnim ključevima. Pored ovih postoje i algoritmi koji poseduju perfektnu teorijski dokazanu tajnost. Uslovi korišćenja ovakvih algoritama, kao što su, da ključevi treba da budu generisani na potpuno slučajan način, gde dužina ključa mora da bude jednaka dužini poruke koja se šifrue i da ključ nikada ne sme da se koristi dva puta, čine ovakve algoritme dosta kompleksnim za korišćenje. *One-Time Pad* predstavlja jedan od algoritama iz grupe perfektnih.

Ukoliko bi omogućili jednostavniju primenu *OTP-a* obezbedili bi teorijski sigurnu komunikaciju. Za takav cilj potrebno je realizovati generatore slučajnih nizova takve konfiguracije da nam obezbede da u svim tačkama komunikacije imamo slučajne nizove na



osnovu kojih ćemo dobiti kriptološke ključeve koji se mogu nesmetano koristiti za OTP šifrovanje.

Prvi zadatak je generisanje slučajnih nizova. Slučajne nizove je moguće generisati pomoću fizičkih generatora koji na osnovu prirodnih pojava na ulazu, na izlazu daju stvarno slučajne nizove. Osim ovih postoje i generatori koji za svoj ulaz koriste stvarno slučajne nizove, a na izlazu daju ekstraktvane slučajne nizove uz pomoć matematičkih funkcija. Takvi nizovi se nazivaju pseudo-slučajni, a generatori pseudo-slučajni generatori (PRNG), dok stvarno slučajne nizove generišu stvarno slučajni generatori (TRNG). I jedan i drugi generator je moguće upotrebiti za projektovanje šifarskih sistema, ali u oba slučaja je neophodno generisati čisto slučajne nizove čija je jedina uloga u domenu generisanja kriptoloških ključeva. Ipak treba obratiti pažnju na ispravno korišćenje *PRNG-a* iz razloga jer predstavljaju determinističke generatore čiji se izlaz može ponoviti.

Ukoliko obratimo pažnju na algoritme sa simetričnim ključem, kao drugi zadatak nameće se potreba za identičnim slučajnim nizovima u svim tačkama komunikacije, imajući u vidu da i perfektni šifarski sistemi koriste dizajn simetričnih šifarskih sistema. Ukoliko je potrebno realizovati komunikaciju između fizički udaljenih tačaka, distribucija ključa javnim kanalom nije moguća. U praksi ne postoji siguran komunikacioni kanal, osim “kurirske službe”, čiji je zadatak dostava ključeva, ali postavlja se pitanje šta u slučajevima u kojima ovaj scenario nije moguće realizovati.

U ovom radu pokušaćemo da obezbedimo sistem koji će omogućiti generisanje i distribuciju kriptoloških ključeva u bilo kojoj tački i time pojednostaviti upotrebu simetričnih algoritama, a pre svega algoritama perfektne tajnosti.

Za prevazilaženje problema poslužićemo se javno dostupnim podacima civilnog vazduhoplovstva koje ćemo prikupljati u realnom vremenu. Jasno je da na ovaj način omogućavamo prikupljanje istih podataka na različitim geografskim lokacijama. Prikupljene podatke potrebno kategorizovati na upotrebljive i neupotrebljive, tj. na one koji su promenljivi i nepromenljivi u vremenu. Podatke promenljive u vremenu potrebno je analizirati i proveriti da li poseduju i ako da, koliki nivo slučajnosti, te da li ih je moguće upotrebljavati kao izvor slučajnosti. Za odabrane kvalitetne podatke potrebno je razviti algoritam za binarizaciju koji će podatke o letu nekog aviona predstaviti nizom bita. Dobijene binarne podatke korišćićemo u protokolu za generisanje kriptoloških ključeva na udaljenim lokacijama, Satelitski scenario. Ključeve dobijene na bazi binarnih nizova podvrgnućemo rigoroznoj informacionoj analizi kako bismo ustanovili da li nizovi odnosno ključevi poseduju visok nivo slučajnosti i da li se, kao takvi, mogu koristiti u perfektnim šifarskim sistemima.

Razvojem ovog protokola postigli bismo sledeće doprinose:

1. Pronalaženje sopstvenog izvora slučajnosti:
Izvor slučajnosti u prirodi je mnogo. Neki se veoma lako mogu iskoristiti za generisanje bitskih sekvenci dok je za neke potrebna zahtevna infrastruktura. Odabir idealnog izvora zavisi i od dalje implementacije. Iz tog razloga korišćenje javno dostupnih podataka koji u sebi sadrže određeni nivo slučajnosti predstavlja povoljan izvor za dalju upotrebu.
2. Realizacija sopstvenog generatora slučajnih sekvenci:
Algoritmi šifarskih sistema kao što je rečeno predstavljaju matematičke funkcije. Funkcije mogu biti tajne ali i javne. Algoritmi koji su provereno sigurni, među kojima i perfektni algoritmi, su javni algoritmi. Što nam govori da tajnost nekog



šifarskog sistema zavisi od ključa. Ukoliko bi koristili javni generator čisto slučajnih nizova kao podlogu za naše kriptološke ključeve, morali bi da se pozabavimo samom bezbednosti tih generatora. Kod takvih generatora izlaz jeste čisto slučajan ali može biti kompromitovan ili dirigovan. Ne smemo se osloniti na generatore čiji izvor može biti kontrolisan i samim tim i bitovi na izlazu generatora. Iz tog razloga potrebno je razviti sopstveni generator koji nije izložen spoljnim uticajima.

3. Generisanje visoko kvalitetnih binarnih sekvenci čiji kvalitet će potvrditi baterija NIST testova:

Kao dokaz slučajnosti i sigurnosti, odnosno kao potvrdu kvaliteta generatora moraju se koristiti već provereni testovi za dokazivanje slučajnosti. Podvrgavanjem rezultata generatora ovim testovima i dobijanjem kvalitetnih rezultata obezbeđuje se uslov potreban za korišćenje ovakvih generatora u šifarskim sistemima za kriptološke ključeve.

4. Bezbedna razmena tajnog ključa između dve strane na određenim geolokacijama:

Ovakav način generisanja nizova omogućava nam da na udeljenim lokacijama dobijamo kvalitetne nizove koji međusobno poseduju određenu korelisanost. Korišćenjem takvih nizova uz već postojuće protokole za razmenu i generisanje kriptoloških ključeva moguće je na dve strane generisati identične kriptološke ključeve potrebne za perfektne šifarske sisteme.

5. Dovoljna količina bitova za upotrebu *One Time pad*:

Kao najveći nedostatak *One-Time Pad* algoritma navodi se potreba da se na obe strane komunikacije nalaze dugački kriptološki ključevi koji se koriste samo prilikom jednog šifrovanja. Na ovaj način takav nedostatak se može prevazići jer je moguće onezbediti nesmetano generisanje novih ključeva u svim tačkama komunikacije.

6. Mogućnosti za implementaciju u već postojeće kriptografske sistema za zaštitu podatak.

Kriptografski sistemi danas su retko zasnovani na perfektnim šifarskim algoritmima usled komplikovane realizacije. Primenom ovakvog sistema omogućava se jednostavna upotreba ovakvih algoritama. Osim toga, moguća je implementacija na već postojeće sisteme koji koriste druge algoritme bazirane na simetričnim kriptografskim ključevima.

1.3 STRUKTURA RADA

Na početku ovog rada bavićemo se dosadašnjim dostignućima u oblasni istraživanja. Osvrnućemo se posebno na protokole za distribuciju kriptoloških ključeva, pogotovu na Kvantni kanal i Satelitski scenario. Navešćemo osnovne prednosti mane trenutnih rešenja. Zatim ćemo u trećem poglavlju ovog rada govoriti o teorijskim osnovama potrebnim za realizaciju jednog ovakvog sistema. Navodićemo primere do sadašnjih generatora slučajnih brojeva ali bavićemo se i njihovom klasifikacijom na bazi rezultata statističkih testova koji se primenjuju na generatore. U okviru ovog poglavlja govorićemo još i o sistemima navigacije čije ćemo podatke koristiti za realizaciju našeg sistema. U četvrtom poglavlju precizno ćemo predstaviti realizaciju samog rešenja, od prikupljanja podataka, odabira



upotrebljivih do generisanja binarnih sekvenci. Navešćemo primer generisanja binarnih sekvenci na dve udaljene lokacija i primenu satelitskog scenarija na osnovu tih dobijenih sekvenci. Na taj način pokazaćemo rezultat i mogućnost primene jednog ovakvog sistema. U okviru ovog poglavlja sprovedemo i statističke testove nad dobijenim sekvencama kako bi potvrdili da se ovakav generator može smatrati slučajnim, a dobijene zajedničke sekvence iskoristićemo da proračunamo njihovu kompleksnost i na taj način ćemo pokazati da se mogu koristiti kao baza za kriptološke ključeve. Na kraju u okviru zaključka analiziraćemo rezultate dobijene u okviru četvrtog poglavlja kao i postignute ciljeve postavljene u okviru uvodnog poglavlja. Na samom kraju, podstaknuti dobijenim rezultatima, daćemo predlog za budući rad.



2. PREGLED U OBLASTI ISTRAŽIVANJA

Imajući u vidu da su najčešće korišćeni algoritmi za zaštitu podataka javni, iz tog razloga bezbednost takvog sistema počiva na ključu, te se šifrovanje svodi na upotrebu visoko kvalitetnih kriptoloških ključeva. Problem generisanja visoko kvalitetnih ključeva i njihove dalja distribucije predstavljaju kompleksna rešenja, tako se nameće potreba za daljim idejama, istraživanjima i realizacijama.

Sam problem ključeva, možemo podeliti na dva dela. Prvi je generisanje. Razvoj tehnike i računara omogućio veliki broj operacija u jedinici vremena. Kao rezultat toga jasno se nameće da ključevi moraju biti tako dobro generisani, da ni posle stotine i hiljade pokušaja računar ne može da ga otkrije. Ta činjenica uslovlja je generisanje ključeva isključivo baziranih na slučajnim sekvencama. Međutim, potrebno je dobiti takve sekvence i za njih dokazano tvrditi da su slučajne. Kao izvori takvih sekvenci navode se slučajni generatori koji su podeljeni u dve osnovne grupe. Stvarno slučajni (TRNG) i pseudo-slučajni (PRNG) generatori obezbeđuju slučajne sekvence na svom izlazu. U zavisnosti od kvaliteta sekvenci neke se mogu iskoristiti za generisanje kriptoloških ključeva.

Tako generisane ključeve potrebno je obezbediti na svim tačkama koje učestvuju u komunikaciji. Idealni sistem predstavljao bi zasebni generatori u svakoj tački komunikacije koji bi generisali iste slučajne ključeve. Kristalni oscilatori su iskorišćeni na takav način ali potreba za sinhronizacijom otežava njihovu upotrebu. Iz tog razloga najčešća je realizacija koja podrazumeva generator ključeva na jednom mestu i potrebu za njihovom daljom distribucijom. Kako idealni, totalno sigurni, kanal ne postoji drugi deo problema predstavlja distribucija ključeva.

„Golub pismoša“ ili kurirska služba svakako nisu dobri vidovi za distribuciju kodne knjige. Međutim sistem šifrovana ključeva pre distribucije nije rešenje, jer tim problem nije rešen već za korak zakomplikovan. Protokoli za distribuciju ključeva rešavaju problem distribucije simetričnih ključeva. Broj različitih protokola se iz dana u dan je sve veći. Najpoznatiji protokol iz ove vrste je svakako *Diffie-Hellman* ali se za razmenu ključeva može koristiti i asimetrični šifarski sistem - *RSA*. *DH* protokol se sastoji od para velikih prostih brojeva i jednosmerne funkcije, diskretni eksponent. Inverzna funkcija ovoj funkciji je diskretni logaritam i sigurnost ovog protokola zasnovana je na pretpostavci da je funkciju diskretnog logaritma teško izračunati. Međutim, kako je procesorska moć računara iz dan u dan sve veća, rešavanje diskretnih algoritama više nije problem, pa tako poznavanjem javnog ključa lako se može doći i do privatnog ključa. Iz tog razloga ključ koji može biti korišćen u perfektnim šifarskim sistemima moramo tražiti u kriptografskim sistemima nastalim na bazi fizičke neodređenosti.

2.1.1 Kvantna kriptografija – Kvantni kanal

Kvantni kanal predstavlja komunikacioni kanal zasnovan na bazi kvantne kriptografije koja koristi prirodnu neodređenost kvantnog sveta. Ovakav komunikacioni kanal nije moguće prisluškivati bez ometanja prenosa. C.H. Benet (engl. *C.H. Bennett*) i G. Brassard (engl. *G. Brassard*) su 1984. napravili model kvantnog kanala za generisanje i distribuciju ključa, poznat pod imenom *BB84* [1].

Činjenicu da se čestice ne nalaze na jednom mestu već na više mesta odjednom i da nije moguće izmeriti obe veličine koje opisuju svaku česticu (brzina i pozicija) jasno govori da



je kvantni svet neodređen. Ova činjenica je iskorišćenja u kriptografiji za generisanje i distribuciju ključa.

Za ovaj protokol neophodna su dva kanala. Jedan jednosmerni optički kanal kojim će putovati fotoni od pošiljaoca (Alise) ka primaocu (Bob), a drugi dvosmerni javni kanal čija sigurnost nije bitna. Fotoni mogu biti polarizovani horizontalno ili vertikalno. Horizontalno polarizovani fotoni mogu proći samo kroz horizontalno polarizovani filter, dok ostali fotoni ostaju blokirani. Međutim zaokretanjem filtra za ugao od 45° verovatnoća prolaska fotona je jednaka verovatnoći da foton ne prođe, odnosno 50%.

Polarizacija može biti predstavljanja u bilo kom koordinatnom sistemu gde se dve prave seku pod pravim uglom. Pri distribuciji fotona kvantnim kanalom koriste se dve različite polarizacije, pravougaona (horizontalno i vertikalno) i dijagonalna (leva i desna dijagonala). Alisa šalje niz različito polarizovanih fotona u sva četiri pravca (prvi red Slika 1), dok Bob proizvoljno postavlja filter za određenu polarizaciju (drugi red Slika 1). Jasno je da će Bob primiti svaki foton, s'tim što tamo gde je pogrešno postavio filter dobija slučajnu vrednost. Iz tog razloga Bob javnim kanalom Alisi javlja konfiguraciju filtra, a Alisa Bobu dogovara na kojim pozicijama je ispravno filter podešen. S' obzirom na to da za ispravno podešen filter foton koji prođe može biti u polarizovan na dva načina (kod pravougaone polarizacije: ili horizontalno ili vertikalno), moguće je predstaviti ga binarno sa 0 ili 1 (treći red Slika 1). Na taj način Alisa je Bobu distribuirala ključ.

•	0	•	•	•	0	1	•	1	•

Slika 1 - Distribucija bita kvantnim kanalom

Pod pretpostavkom da je Alisa za odabir polarizacije fotona koristila stvarno slučajni generator na ovaj način dobijamo visoko kvalitetan ključ gde dužina ključa nije limitirana i što je veoma bitno bez ikakvih unapred poznatih parametara. Ovakvi simetrični ključevi se mogu koristiti u perfektnim šifarskim sistemima kakav je *One-Time Pad*. Međutim kao osnovni nedostatak ovakvog vida distribucije predstavlja potreba za optičkim kanalom između strana koje komuniciraju.

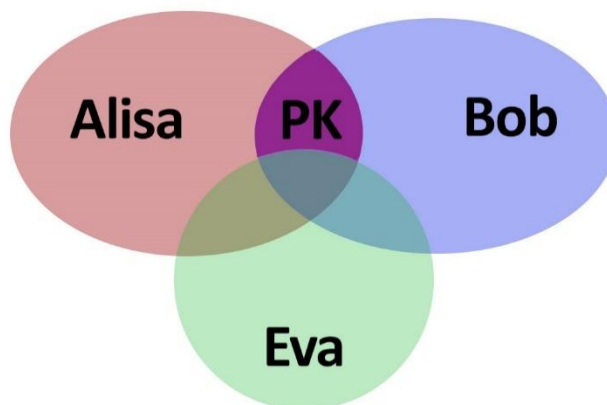
2.1.2 Satelitski scenario

Na osnovu *BB84* protokola Uli Maurer (engl. *Ueli Maurer*) je 1993. godine objavio rad [2] u kome je optički kanal zamenio satelitom koji bi konstanto emitovao radio talase slabe snage. Radio talasi bi na taj način dolazili do zemlje uz određene greške i time bi učesnici primali različit signal na zemlji. Ta činjenica predstavlja osnov Satelitskog scenarija.

Radio talasi bi predstavljali binarne stvarno slučajne sekvence, nad kojim bi nakon primanja od strane učesnika bile izvršene tri faze usaglašavanja korelisanih nizova. Prva faza je destilacija prednosti (engl. *Advantage Distillation*), druga usaglašavanje informacija (engl.

Information Reconciliation), a treća faza je pojačavanje privatnosti (engl. *Privacy Amplification*).

Sama ideja Satelitskog scenarija najlakše se predstavlja pomoću skupova, odnosno preseka skupova.



Slika 2 - Presek Skupova

Učesnici Alisa, Bob i Eva primaju sekvence i dobijaju različite nizove koji su predstavljeni skupom. Skupovi međusobno imaju zajedničke ali i individualne elemente. Alisa i Bob predstavljaju legitimne učesnike a Eva zlonamernog učesnika koji prisluškuje komunikaciju, poenta ovog protokola je da odredi elemente preseka PK (Slika 2), jer upravo taj presek predstavlja zajedničku informaciju koju poseduju i Alisa i Bob, a koju ne poseduje Eva.

1. Destilacija prednosti

Presek skupova predstavljen grafički potrebno je primeniti i na primljenoj slučajnoj sekvenci. U okviru prve faze potrebno je što više umanjiti zajedničku informaciju između Eve i legitimnih učesnika, naravno sve pod pretpostavkom da Eva nema sekvencu identičnu poslatoj jer u tom slučaju kao što je već rečeno ovaj protokol gubi smisao. Destilacija prednosti se obavlja na sledeći način:

- Alisa i Bob dele primljenu sekvencu na parove od po dva bita (bigrame)
- Nezavisno računaju *parity check* bit bitova svakog bigrama.
- Alisa šalje Bobu javnim kanalom svoje *parity check* bitove
- Ukoliko su *parity check* biti isti zadržava se prvi (levi) član bigrama, a ukoliko su različiti ceo bigram se odbacuje
- Na novo-dobijenoj sekvenci računa se ponovo *parity check* bit i ponavljaju se ostali koraci. Broj ponavljanja zavisi od pretpostavke greške između sekvenci Alise i Boba.

2. Usaglašavanje informacija

Ova faza predstavlja proces u kome se dva niza izjednačavaju. Princip rada je zasnovan na ispravljanju grešaka pomoći *parity check* funkcije. Kao i u predhodnoj fazi sekvenca se deli na blokove, s' tim što sada broj bita u okviru bloka može biti i veći, ali svakako kod oba učesnika isti. Računanjem *parity check* bita i upoređivanjem mogu se otkriti blokovi koji imaju neparni broj grešaka. Ukoliko se *parity check* bit razlikuje blok se odbacuje u suprotnom blok se zadržava. Parni broj grešaka se kod ovog sistema neće otkriti i to predstavlja anomaliju ovog sistema međutim i u ovoj fazi postoji više rundi ponavljanja,

te se takve greške mogu otkriti u narednim rundama. Broj rundi zavisi od pretpostavka o broju grešaka koje se očekuje.

3. Pojačavanje privatnosti

U ovoj fazi eliminiše se zajednička informacija Eve kao napadača i legitimnih učesnika Alise i Boba. Alisa bira funkciju koju će primeniti na preostalu sekvencu, a funkcija mora biti jedna od univerzalnih *hash* funkcija. Ukoliko i Bob nakon primene *hash* funkcije nad sekvencom dobije istu vrednost može se zaključiti da imaju identičan niz stvarno slučajnih bita koji sadrže visko stepen slučajnosti.

Ovakav protokol kao i protokol *BB84* nema definisanu maksimalnu dužinu sekvence, te se može koristiti za perfektne šifarske sisteme. Bitno je napomenuti da ni kod ovog protokola nije potrebno prethodno razmeniti neke tajne parametre. Svi dogovori koji su Alisi i Bobu neophodni u fazama usaglašavanja nizova mogu da se obavljaju javnim ne zaštićenim kanalom. Jasno je da za oba protokola je neophodno obezbediti kvalitetne slučajne sekvence.

3. TEORIJSKE OSNOVE

Mimika, crteži u pećinama, za istoriju predstavljaju prvi vid komunikacije praistorijskog čoveka. Istorija nam kaže da ti ljudi nisu znali šta su slova, pa su crtali, nisu artikulirano pričali pa su mimikom objašnjavali. Nepoznate stvari se ne mogu nacrtati, i tada, postajao je dogovor, za dva čoveka određena mimika, ili pak slika, značila je „nešto“. Ono što nam istorija ne govori, ono što sami zaključujemo je da ta slika za neka druga dva čoveka nije predstavljala to isto „nešto“. Nesvesno, prvom slikom pračovek je rekao prvu reč, ali istovremeno i prvu šifrovanu poruku - šifrat.



Slika 3 - Egipatski simboli (levo) i Skital (desno)

Prema istoriji prva zaštita podataka pojavljuje se u Egiptu. Faraoni i plemstvo svesno i namerno menjali su simbole drugim simbolima kako bi onemogućili da neko drugi razume šta je prikazano. Skital u Sprati ili kasnije Cezarova šifra u Rimu govore nam da su ljudi i pre nekoliko hiljada godina ozbiljno pristupali zaštiti nad podacima.

3.1 ALGORITMI ZA ZAŠTITU PODATAKA

Kako godine prolaze način zaštite tj. algoritam postaje sve komplikovaniji i sigurniji. Razvoj računara još više je doprineo sve bržem razvoju novih i unapređenju postojećih algoritama za zaštitu nad podacima.

Veliki broj algoritama deli se u dve grupe. Podela na algoritme sa simetričnim i asimetričnim ključem govori nam da osim kompleksnosti same matematičke funkcije važan uslov za zaštitu predstavlja i ključ.

Algoritmi sa simetričnim ključem mogu se poistovetiti sa katanacem koji ima jedan, jedinstven ključ koji otvara samo taj katanac. Ključ može biti korišćen više puta ali i pri šifrovanju i pri dešifrovanju uvek mora biti isti. Sama prednost ovakvog algoritma predstavlja i manu, jer ključ treba distribuirati nezavisno od šifrovane poruke. Ipak i pored toga ovi algoritmi su u upotrebi i danas, *Blowfish*, *Twofish*, *DES* ali i *AES* kao jedan od najsigurnijih algoritama danas.

Problem razmene ključeva 1976. godine V. Difi (engl. *W. Diffie*) i M. Helman (engl. *M. Hellman*) pokušavaju da reše protokolom o razmeni ključeva. Ova ideja je našla primenu u **algoritmima sa asimetričnim ključem**. Ovi algoritmi se ne mogu predstaviti katanacem, jer kod ovih algoritama jedan ključ „zaključava“, a drugi „otključava“ podatak. Par ključeva (javni i tajni) rešava problem algoritama sa simetričnim ključem jer sada više nema potrebe da na obe strane imamo identičan ključ. Međutim postojanje javnog ključa predstavlja opasnost od zloupotrebe. Najpoznatiji ovakav algoritam koji je danas u upotrebi je svakako *RSA*.

3.2 IDEALNI KOMUNIKACIONI KANAL

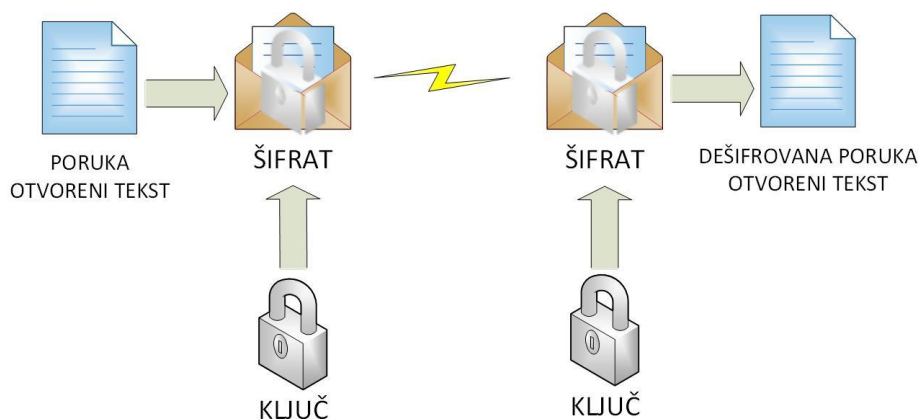
Podatke možemo da podelimo na one koji služe u komunikaciji za razmenu informacija ali i one uskladištene na nekim memorijskim uređajima na određenim geografskim lokacijama. Podaci na takvim lokacijama mogu biti i fizički odvojeni od potencijalnih napadača i njih je lakše štititi. Međutim podaci koji se „kreću“ su najizloženiji i njih je najteže štititi.

Zahtevi koji se postavljaju kada se govori o podacima korišćenim za komunikaciju su obezbeđivanje:

- **Poverljivosti** – Podatak može i mora primiti samo prijemna strana kojoj je podatak i namenjen,
- **Verodostojnost** - Prijemna strana sa sigurnošću zna da je podatak poslala tačno određena predajna strana,
- **Integritet** – Prijemna strana sa sigurnošću zna da u komunikacionom kanalu podatak nije izmenjen,
- **Neporecivost** – Predajna strana mora stajati iza poslatog podatka i ne može naknadno tvrditi da taj podatak nije poslat od strane nje.

Aspekt sigurnosti možemo na trenutak da stavimo sa strane, i pogledamo samo kvalitet prenetog podatka, idealni komunikacioni kanal trebao bi da obezbedi prenos podataka koji ni jednom delu nikada neće biti oštećen, već taj uslov nam govori da je realni komunikacioni kanal daleko od idealnog. Sa aspekta sigurnosti idealni komunikacioni kanal predstavlja siguran kanal koji nije moguće prisluškivati ili ometati, koji obezbeđuje apsolutnu zaštitu podatka koji se distribuira i koji garantuje da samo dve legitimne strane koje komuniciraju imaju kontakt sa podatkom. Približavanje realnog kanala idealnom moguće je ali uz obezbeđivanje gore pomenutih zahteva.

3.3 ALGORITAM SAVRŠENE TAJNOSTI – *ONE-TIME PAD*



Slika 4 - Šifrovanje i dešifrovanje poruke

1882. godine bankar Frank Miler (engl. *Frank Miller*) objavio je rad „*Telegraphic Code to Insure Privacy and Secrecy in the Transmission of Telegrams*“ [3] u kome on opisuje prvi *One-Time Pad* sistem. Međutim, za stvarne tvorce *One-Time Pad*-a smatraju se G. Vernam (engl. *G. Vernam*) i J. Mauborgne (engl. *J. Mauborgne*) koji su ga 1917. izmislili,

a 1919. i patentirali. Kasnije, 1949. godine Klod Šenon (engl. *Claude Shannon*) definiše „perfektnu šifru“.

Prema Šenonu šifrat perfektnih šifara ne daje nikakvu informaciju o podatku koji je šifrovan, niti o ključu koji je korišćen. Šenon ovu tvrdnju dokazuje pomoću entropije. Entropiju, kao pojam u teoriji informacija, definisao je Šenon 1948. godine u svom radu "Matematička teorija komunikacija". Šenon je predstavio entropiju kao meru za količinu neodređenosti nekog informacionog izvora, ali o tome nešto više u nastavku.

Definisanjem „perfektnu šifru“ omogućilo je da se *One-Time Pad* okarakterise kao savršeni kriptografski mehanizam. Upravo zbog toga *One-Time Pad* se smatra jedinim teoretski dokazan sigurnim algoritmom. Ovakva sigurnost naziva se informaciono-teorijska sigurnost i podrazumeva bezuslovnu ili teorijsku sigurnost u odnosu na protivnika koji raspolaže neograničenim vremenom i računarskim resursima. [4]

Princip rada *One-Time Pad* prvenstveno zasniva se na radu direktno nad binarnim podacima. Šifrovanje, odnosno dobijanje šifrata iz podatka koji se šifruje (otvoreni tekst) se obavlja pomoću binarne operacije „ekskluzivno ili“ (XOR) koja se primenjuje nad otvorenim tekstem i ključem, gde su i otvoreni tekst i ključ predstavljeni u binarnom formatu, te se operacija XOR primenjuje bit po bit otvorenog teksta i ključa. Postupak dobijanja otvorenog teksta iz šifrata, dešifrovanje, je identičan postupku šifrovanja s tim što se operacija XOR primenjuje nad šifratom i identičnim ključem koji je korišćen prilikom šifrovanja.

One-Time Pad algoritam predstavljen formulom (*Formula 1*) glasi:

$$C = P \oplus K$$

Formula 1

gde P predstavlja otvoreni tekst, K ključ, a C šifrat dobijen operacijom XOR. Prilikom dešifrovanja formula izgleda identično osim što šifrat C i otvoreni tekst P zamene mesta. U tabeli (Tabela 1) su date moguće vrednosti operacije XOR.

P	0	0	1	1
K	0	1	0	1
C	0	1	1	0

Tabela 1 – XOR vrednosti

Da bi rezultat primene *One-Time Pad* algoritama bio ispravan, a samim tim i siguran moraju se ispuniti određeni uslovi vezani za ključ koji se primenjuje u šifrovanju odnosno dešifrovanju. Prvi i osnovni uslov koji se postavlja je uslov dužine ključa. S' obzirom na to da je XOR operacija bitski bazirana, ključ i otvoreni tekst moraju biti iste bitske dužine. Iz same matematičke funkcije može se zaključiti da je za šifrovanje i dešifrovanje potreban identičan ključ, te je to drugi potreban uslov. Iz naredne formule (*Formula 2*) vidi se da dva različita otvorena teksta nakon XOR operacije sa istim ključem daju dva različita šifrata. Međutim, XOR nad tim šifratima daje ključ, čime dolazimo do trećeg potrebnog uslova, ključ može biti korišćen isključivo samo jedanputa.

$$\begin{aligned} C1 &= P1 \oplus K \\ C2 &= P2 \oplus K \\ K &= C1 \oplus C2 \end{aligned}$$

Formula 2



Četvrti uslov predstavlja veoma bitan uslov koji nastaje iz definicije perfektne šifre. Verovatnoća ključa tj. količina slučajnost koju on nosi je direktan uslov za kvalitet šifrata koji se dobija. Zato je potrebno da ključ bude stvarno slučajan odnosno da njegova entropija bude 1 (ili bar približno 1), čime bi taj ključ u sebi posedovao visok stepen neodređenosti. I poslednji uslov, podrazumevani uslov, potrebno je da samo strane koje komuniciraju poseduju ključ, tj. ključ ne sme doći kod potencijalnog napadača. Ovi uslovi vezani za ključ predstavljaju i glavni nedostatak ovog algoritma, te je *One-Time Pad* dosta nepraktičan, ali ipak ima svoju primenu. Osim ovog postoji i još jedan nedostatak, mehanizam integriteta ne postoji.

Analiza svih uslova koje je potrebno da ispuni ključ ovog algoritma, ali i većine ostalih kriptoloških algoritama, govori nam da je neophodno osim kvalitetnih algoritama, bilo da su tajni ili javni, posedovati i kvalitetne visoko slučajne ključeve, odnosno kvalitetne generatore tih ključeva. Prednost koju jedan perfektan algoritam kakav je *One-time Pad* ima u odnosu na druge sigurne algoritme je u tome što krypto-analitičar nema mogućnost da i ukoliko bi došao do korišćenog ključa presretne i dešifruje buduću komunikaciju, što je izuzetna prednost, ali opet na drugoj strani stalna potreba za veoma dugačkim slučajnim nizovima predstavlja nedostatak ovog šifarskog algoritma.

3.4 SLUČAJNOST

Da li slučajnost postoji? Da li možemo predvideti ishod nekog događaja? Pitanja su koja se provlače kroz vekove. "Susret na trgu" za Aristotela je bio slučajan događaj. Aristotel je postavio definiciju da je slučajan događaj svaki događaj kome je uzrok nepoznat. Nekoliko vekova kasnije matematičar i astronom Laplas (fr. *Pierre-Simon Marquis de Laplace*) bavio se temom slučajnosti, ali na drugačiji način. Tvrdio je da je sve determinističko i kada bi poznavao sve parametre koji definišu stanje kosmosa, mogao bi da predvidi svaki od slučajnih događaja.

Ako pogledamo novčić, tj. bacanje novčića možemo zaključiti da Laplasova teorija da je ishod poznat može tumačiti kao ispravna ali samo u slučaju da znamo početno stanje novčića, da možemo da izmerimo jačinu kojom ga bacamo tj. broj okreta i da znamo tačno kako ostale strane sile deluju na novčić. U realnom svetu ovo nije moguće, pa se kao najbolji primer za odabir jednog od dva moguća ishoda navodi upravo bacanje novčića.

Ulazak u kazino, odluka je da se prepustimo slučajnosti, igranje igara na sreću iziskuje veliki broj slučajnih ishoda, ako pretpostavimo da se iza svega krije stvarna slučajnost onda taj veliki broj ishoda na neki način mora biti dobijen, generisan, ali tako da ne postoji način spoljašnjeg zlonamernog uticaja.

Kako je već rečeno, idealni komunikacioni kanal ne postoji. Greška u prenosu, iako u nekim slučajevima mala, postoji. Šenon definiše način na koji se greška može još više umanjiti. Stvarnom podatku dodaju se slučajne sekvence i zatim se takva poruka pušta u prenos. Međutim, postoji mogućnost zloupotrebe takvog podatka, te ga je potrebno i šifrovati, a kako je već rečeno, za kvalitetno šifrovanje potreban je kvalitetan algoritam i kvalitetan ključ, nastao na bazi nekog slučajnog ishoda.



3.5 INFORMACIONE MERE – ENTROPIJA

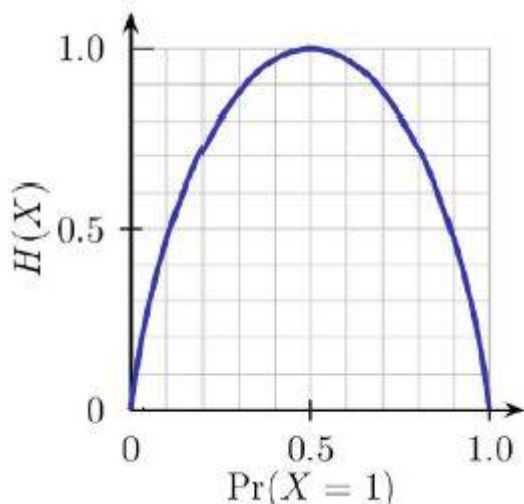
Kako bi bolje razumeli razliku između određenosti i neodređenosti, odnosno slučajnosti i ne slučajnosti moramo se prvo pozabaviti merom koja će nam sa naučne strane tvrditi da li je nešto slučajno ili ne. Entropija predstavlja meru za količinu neodređenosti ili prosečna količina informacije koju sadrže generisane poruke nekog informacionog izvora. Uz pomoć entropije koju je kao pojam definisao Klod Šenon 1948. godine moći ćemo da predstavimo tačnu vrednost neodređenosti bilo kog izvora.

Ukoliko za informacioni izvor uzmemo neki impulsni generator, i sa 1 predstavimo stanje kada impuls postoji, a sa 0 kada je izostao, tada imamo verovatnoću $P(0)$ i $P(1)$ kao verovatnoću pojavljivanja nule ili jedinice na izlazu ovakvog binarnog izvora. Verovatnoću $P(0)$ možemo predstaviti sa p , a verovatnoću $P(1)$ sa $1-p$ te tako entropija binarnog izvora glasi:

$$H = -p \log p - (1 - p) \log(1 - p)$$

Formula 3

Ukoliko je verovatnoća pojavljivanja nule odnosno jedinice jednaka, tj. $P(0) = P(1) = 0,5$ onda je entropija takvog izvora $H=1$. Tako možemo zaključiti da za neki generator možemo tvrditi da je generator slučajnih nizova ukoliko je vrednost entropije njegovog izlaza $H \approx 1$.

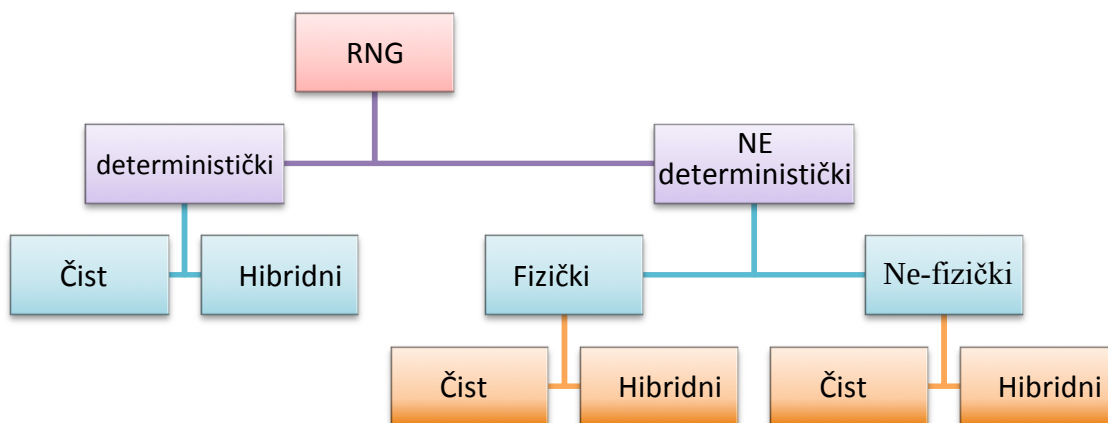


Slika 5 - Entropija slučajne varijable sa dva ishoda [5]

3.6 GENERATORI SLUČAJNIH NIZOVA

Kao što je rečeno, sveobuhvatna životna potreba za slučajnim stanjima ili brojevima, a pogotovu usled ekspanzije računarskih sistema tera nas da razmišljamo kako i na koji način dobiti slučajne podatke. Prva pomisao vuče ka računarima. Ali se onda od filozofskog pitanja da li slučajnost postoji dolazi do zdravorazumskog odgovora da su računarski čipovi deterministički, sa tačno definisanim skupovima, ma koliko velikim, konačnih stanja. Iako matematičke metode ne mogu dati slučajne brojeve, mogu dati odgovor da li su neke sekvence slučajne ili ne i koju vrednost entropije poseduju. Na osnovu toga za generatore slučajnih podataka možemo tvrditi da su slučajni. Međutim, i među slučajnim generatorima postoje oni koji su više slučajni od drugih, ali i oni kod kojih je broj slučajnih sekvenci na izlazu ograničen.





Slika 6 - Shema klasifikacije RNG ([6])

Slučajni generatori (RNG) se mogu klasifikovati prema shemi (Slika 6). Prvu grupu čine deterministički generatori, poznatiji pod imenom pseudo-slučajni generatori (engl. *Pseudo random number generator - PRNG*). Nedeterministički generatori čine drugu grupu i poznati su pod imenom Stvarno slučajni generatori (engl. *True random number generator - TRNG*). Oni se dele u dve grupe, fizički i ne-fizički. Fizički predstavljaju one koji koriste nedeterminističke efekte elektronskog kola (šum proboja Zener diode, termički šum poluprovodnika) ili fizički eksperiment (radioaktivni raspad). Ne-fizički iskorišćavaju nedeterminističke događaje kao što su sistemsko vreme, vreme pristupa Hard-diska ili korisničkih interakcija (npr. pokreti miša). Hibridni generatori u stvari predstavljaju kombinaciju stvarno slučajnih i pseudo slučajnih generatora.

3.6.1 TRNG

Imajući u vidu da postoji veliki broj različitih *TRN* generatora, najpopularniji i oni koji se najviše koriste, izgrađeni su tako da moraju da sadrže naredne tri komponente [7]:

- Entropija izvora (*Entropy source*): Kao izvor velikog broja *TRN* generatora koriste se fizički procesi kao što su termalni šum u elektronskim kolima, *jitter* (neželjena devijacija signala), Braunovo kretanje, atmosferski šum ili čak i nuklearno raspadanje atoma. Entropija izvora možda predstavlja i najvažniju komponentu, jer upravo ona određuje i najveću moguću entropiju generatora. Ipak, nisu baš svi izvori upotrebljivi u svakom trenutku. Atmosferski šum ili nuklearni raspad zahtevaju posebne uslove, dok sa druge strane dostupnost dioda i jednostavna implementacija su rezultirale velikim brojem generatora baziranih na generisanju nizova nastalih usled šuma.
- Tehnika prikupljanja (*Harvesting technique*): najbolje iskorišćenje entropije izvora predstavljeno je tehnikom prikupljanja. Potvrda kvaliteta odabrane tehnike svodi se isključivo na analizu crne kutije *TRNG*-a pomoću statističkih testova i testova stvarne slučajnosti.
- Dorada (*Post-processing*) iako nije neophodna komponenta, u kvalitetnim rešenjima se koristi. Cilj je napraviti što snažniji *TRNG* doradom izlaznih bitova. Sekundarni cilj, koji je postao važan zbog nedostataka ali i napada, ima ulogu da obezbedi otpornost na promene u okruženju i neprijateljska ometanja.

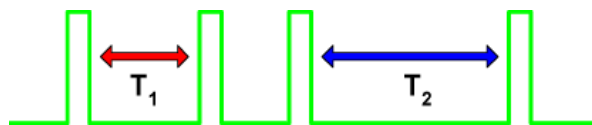
Kao što je rečeno, izvori informacije potrebne za stvarne slučajne generatore (TRNG) su svuda oko nas. Dok bacamo novčić, mi u stvari iz skupa od dva moguća ishoda dobijamo niz različitih koji nisu u korelaciji i naredni ishod ne možemo predvideti. Baš to je jedan od idealnih načina da, pomoću nekog fizičkog uređaja, naše ishode predstavimo nizom bita. Zrno u peščanom satu ili cvetanje pupoljka još su neki od primera koje možemo primetiti u svakodnevnom životu, a koriste se za generisanje slučajnih nizova.

3.6.1.1 Nuklearno raspadanje atoma na Beta način – *HotBits*

Kod TRNG problem predstavlja veličina izlazne sekvence. Kod bacanje novčića, ako rezultat svake interakcije (bačen novčić) predstavimo binarno imamo samo jedan bit. Zato je takav izvor neupotrebljiv. Pri nuklearnom raspadanju osim što se oslobađa velika energija, zbog svoje specifičnosti pravilnim posmatranjem može se iskoristiti i kao izvor stvarno slučajnog generatora sa visokim nivoom slučajnosti.

Neutron ne može samovoljno da postane proton, jer bi to narušilo fizički zakon o održanju energije, međutim pod određenom pobudom neutron može da postane proton sa tim što simultano jezgro ispaljuje (oslobađa) slobodan elektron i na taj način naelektrisanje jezgra ostaje nepromenjeno. Ukoliko uzmemo za primer atom Cezijuma, pobudom nastaje Barijum koji emituje beta česticu (oslobođen elektron), daljnom pobudom Barijuma nastaje gama zračenje. Fizičari su dokazali da pobuda ma koliko jaka bila ipak ima samo 50% šansi da atom Cezijuma pretvori u atom Barijuma u roku od 30.17 godina, što „atomske životu“ predstavlja večnost. Taj dokaz predstavlja osnov za iskorišćenje beta raspada kao generatora slučajnosti.

Za generisanje slučajnih bita merićemo intervale između beta raspada, tj. prelazak atoma Cezijuma u atom Barijuma. Kao što je prikazano na slici (Slika 7), nakon prvog raspada meri se vreme do sledećeg raspada, taj interval obeležava sa T_1 , zatim se čeka naredni (treći) raspad i nakon njega meri vreme do poslednjeg (četvrtog) raspada atoma. Taj interval se obeležava sa T_2 . Upoređivanjem ova dva intervala se dobijaju nula ili jedinica. Ukoliko su intervali jednaki merenje se odbacuje i meri se ponovo, ukoliko je interval T_1 manji od T_2 upisuje se 0, a ukoliko je T_1 veći upisuje se 1. Imajući u vidu da je vreme između dva raspadanja različito jer je šansa da će se atom raspasti 50%, verovatnoća da li će se upisati 0 ili 1 je takođe 50%. Na ovaj način dobijaju se sekvence bitova velike dužine i maksimalne entropije ($H=1$).



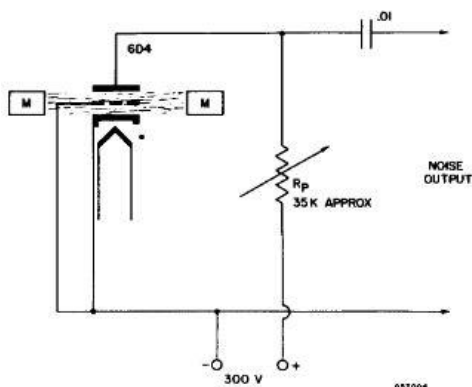
Slika 7 - Intervali [8]

Loša strana ovog generatora je što iziskuje posebne uslove za rad. Ipak, biti generisani na ovakav način dostupni širokoj narodnoj masi putem interneta u okviru projekta *HotBits*. Međutim, sa aspekta bezbednosti, upotreba bitova dobijenih od treće strane, ma koliko oni bili slučajni, za generisanje visoko kvalitetnih ključeva veoma je rizična jer se na taj način trećoj strani daje mogućnost zloupotrebe i kompromitovanja šifrovanog podatka. Iz tog razloga ovakvi biti mogu se koristiti u nekim drugim sferama (igre na sreću, smanjivanje greške prilikom prenosa), ali ne i u kriptografiji.

3.6.1.2 Elektronski šum

Rasprostranjenost elektronskih komponentata i veoma laka implementacija omogućila je da veliki broj TRN generatora koristi elektronski šum kao izvor slučajnosti. Neki od projekata ovakvih generatora su javni ili su bili predmet naučnih radova, dok su neki strogo čuvana tajna autora. U nastavku ćemo pomenuti neke.

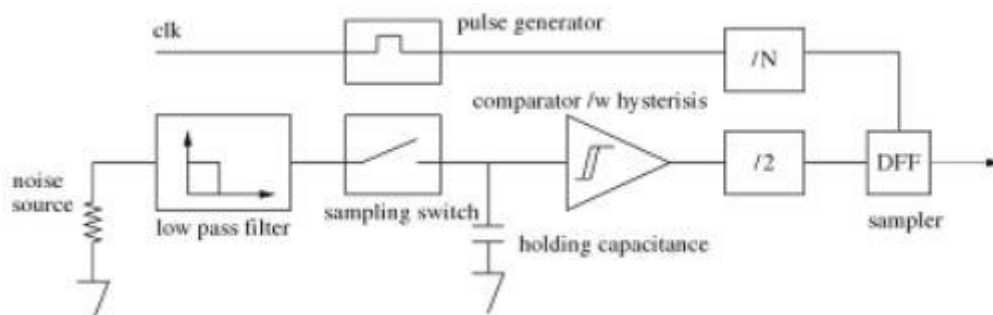
3.6.1.2.1 Realizacija TRNG-a pomoću diode 6D4



Slika 8 - Realizacija TRNG-a pomoću diode 6D4 [9]

Kao najčešći izvor za TRNG koristi se elektronski element dioda. Dioda predstavlja spoj jedne pozitivno (P) i jedne negativno (N) naelektrisane oblasti poluprovodničkog materijala germanijuma ili silicijuma. Zavisno od polarizacije diode, može se postići da se elektroni u PN spoju pokrenu i da kretanjem indukuju struju. To predstavlja šum.

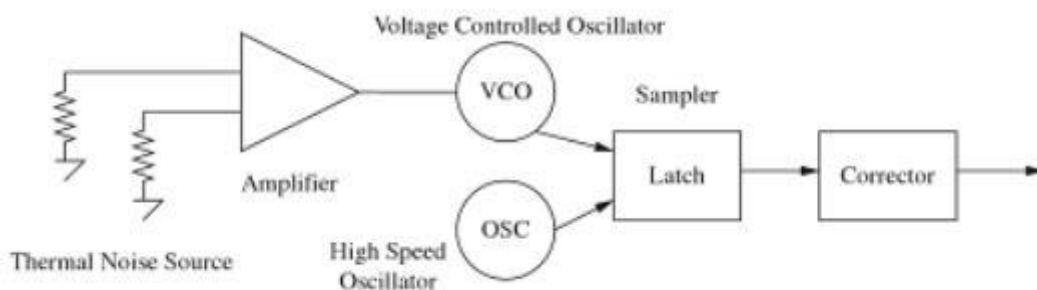
3.6.1.2.2 Baggini and Bucci



Slika 9 - Baggini and Bucci projekat [7]

Shema prikazana na slici (Slika 9) predstavlja projekat autora *Baggini* i *Bucci*. Zasnovan je na belom šumu (kombinacija istovremenih različitih zvukova gde svaki zvuk ima svoju frekvenciju). Komponente ovog projekta su analogne i digitalne.

3.6.1.2.3 Intel TRNG

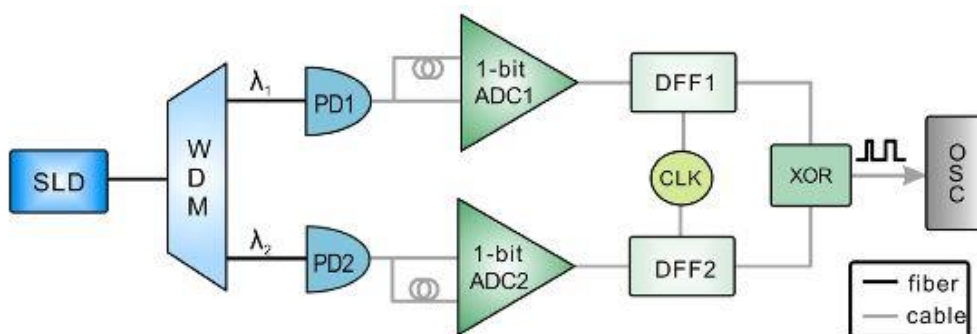


Slika 10 - The Intel TRNG [7]

Ovo je tip stvarno slučajnog generatora zasnovanog na termalnom šumu dobijenom od dva različita otpornika koji se nakon pojačanja koristi za pokretanje naponski kontrolisanog oscilatora (VCO).

3.6.1.2.4 SLD TRNG

Napretkom tehnologije, pogotovu u telekomunikacijama i prelaskom na optičke vodove broj slučajnih bita u jedinici vremena neprestano raste. Do sada pomenuti generatori mogli su da generišu sekvence reda *Mbit* u sekundi. Međutim pojavom lasera i *super-luminescent diode* (SLD) sekvence postaju znatno duže i sada su reda nekoliko *Gbit* po sekundi. *SL* diode predstavljaju *RGB* foto diode i osim svetlosti različitih boja emituju i *amplified spontaneous emission* (ASE) šum. Generator čija je shema prikazana na slici (Slika 11) koristi ASE šum i u stanju je da u realnom vremenu generiše 3 *Gbit* u sekundi i da pri tome bitovi sadrže visok stepen slučajnosti.



Slika 11 - SLD TRNG [10]

3.6.2 PRNG

Kvalitetni stvarno slučajni generatori ne mogu nekad da proizvedu dovoljno dugačke sekvence slučajnih brojeva, iz tog razloga potrebno je ekstrahovati te sekvence u znatno duže, s tim da se kvalitet samih sekvenci sačuva. Ovo je moguće pomoću matematičkih algoritama koji predstavljaju osnovu Pseudo-služajnih generatora (PRNG).

Ulaz *PRNG* predstavlja seme (engl. *seed*) tj. slučajna sekvenca koje nastaje kao izlaz *TRNG*. Izlaz *PRNG* čini sekvenca koja je uvek ista za isti ulaz. Zato za ove generatore

kažemo da su deterministički, jer možemo predvideti izlaz. Osim što je predvidljiva, sekvenca na izlazu na osnovu funkcije koja se koristi ima i maksimalnu dužinu, ali je ta dužina značajno veća od ulazne. Entropija izlazne sekvence zavisi direktno od entropije ulazne i ona nikada ne može biti veća od entropije ulazne sekvence. Ako se nad ovakvim generatorima sprovedu testovi slučajnosti dobiće se rezultati da su sekvence slučajne, međutim upotreba ovakvih generatora je ograničena, i oni se ne mogu koristiti u kriptografiji. Sa pojavom novih *TRNG* poput *SLD TRNG* koji daju veoma dugačke stvarno slučajne sekvence u maloj jedinici vremena, potreba za *PRNG* polako gubi smisao.

3.7 STATISTIČKA ANALIZA SLUČAJNIH NIZOVA

Sekvenca dobijena pomoću neke od realizacija generatora slučajnih sekvenci mora biti podvrgnuta određenom testu nakon kojeg je moguće tvrditi da je ta sekvenca zaista slučajna i da ta realizacija generatora stvarno na svom izlazu daje slučajne sekvence. Broj takvih testova veliki. Svi testovi se bave određenom anomalijom nego izvora ali i samog generatora i na taj način uz pomoć statističkih testova moguće je definisati kvalitet samih izvora slučajnosti, a samim tim i dobijenih sekvenci. Neki raniji testovi za koje se smatralo da su de facto standardni, danas se ne koriste jer ih i generatori za koje važi da su „loši“ sa lakoćom prolaze. Dizajn testova, odnosno odabir na koje će se anomalije obratiti pažnja, deli statističke testove prema oblasti primene generatora. Skup statistički testovi koji je definisan od strane *Donald Knuth* [11], koristi se nad sekvencama koje se primenjuju za potrebe različitih simulacija. Ukoliko su nam potrebni visoko kvalitetni nizovi koji se mogu koristiti u kriptološke svrhe, potrebno je nad nizom nula i jedinica, koji bi predstavljali binarne sekvence, ispitati stepen slučajnosti, odnosno međusobni odnos i zavisnost nula i jedinica. Shodno tome, testovi za testiranje slučajnosti binarne sekvence su fokusirani na različite vrste neslučajnosti koje bi mogle da postoje u nizu. Američki Nacionalni institut za standarde i tehnologiju (NIST) objavio je dokument *A Statistical Test Suite for the Validation of Random Number Generators and Pseudo Random Number Generators for Cryptographic Applications* u kome je definisan paket od petnaest testova kojima treba podvrgnuti nizove dobijene od strane stvarno ili pseudo-slučajnih generatora, kako bi saznali kvalitet nizova. Neki testovi se sastoje od nekoliko pod testova.

3.7.1 Ispitivanje učestanosti u nizu (engl. *Frequency test*)

Kako i ime testa kaže, odnosi se na učestanost pojavljivanja nula odnosno jedinica u nizu. Da bi test bio uspešan, potreban je približan broj nula i jedinica u sekvenci.

Za test su nam potrebni ulazni parametri: n koji predstavlja dužinu binarnog niza i ε koji predstavlja niz bitova nad kojim se treba izvršiti testiranje.

Primer [12]:

- *Ulazni parametri:*

$\varepsilon = 11001001000011111101101010100010001000010110100011$
 $00001000110100110001001100011001100010100010111000$
 $n = 100$



- *Obrada:*
 $S_{100} = -16$
 $s_{abs} = 1.6$
- *Izlazni parametar:*
 $P = 0.109599$
- *Zaključak:*
Na osnovu uslova $P \geq 0.01$, smatra se da je sekvenca slučajna.

3.7.2 Ispitivanje učestanosti u bloku (engl. *Block Frequency test*)

U odnosu na test učestanosti, u kome se posmatra učestanost nula i jedinica u celoj sekvenci, u ovom testu se učestanost pojavljivanja posmatra samo unutar n -bitnih blokova na koje se može izdeliti binarna sekvenca. Tako da je cilj ovog testa uočavanje jednakosti nula i jedinica unutar nekog n -bitnog bloka.

Kao parametre ovog testa, osim parametara iz prethodnog testa (n kao broj bitova u sekvenci i ε što predstavlja sekvencu), sada imamo i parametar M , koji predstavlja dužinu bloka.

Primer [12]:

- *Ulazni parametri:*
 $\varepsilon = 11001001000011111101101010100010001000010110100011$
 $00001000110100110001001100011001100010100010111000$
 $n = 100$
 $M = 10$
- *Obrada:*
 $N = 10$
 $\chi^2 = 7.2$
- *Izlazni parametar*
 $P = 0.109599$
- *Zaključak:*
Na osnovu uslova $P \geq 0.01$, smatra se da je sekvenca slučajna.

3.7.3 Ispitivanje uzastopnih ponavljanja istih bitova u nizu (eng. *Runs Test*)

Ovaj test treba da pokaže da li je oscilovanje između nula i jedinica brzo ili sporo, na taj način što se prati uzastopno ponavljanje nula ili jedinica u nizu.



Primer [12]:

- *Ulazni parametri:*
 $\varepsilon = 11001001000011111101101010100010001000010110100011$
 $00001000110100110001001100011001100010100010111000$
 $n = 100$
 $\tau = \frac{2}{\sqrt{n}} = 0.02$
- *Obrada:*
 $\pi = 0.42$
 $Vn(obs) = 52$
- *Izlazni parametar*
 $P = 0.500798$
- *Zaključak:*
 Na osnovu uslova $P \geq 0.01$, smatra se da je sekvenca slučajna.

3.7.4 Ispitivanje najdužeg uzastopnog ponavljanja jedinica u bloku (engl. *Longest Run Of Ones*)

Fokus ovog testa je na najdužem uzastopnom ponavljanju jedinica u n-bitnom nizu. Kao svrhu testa imamo određivanje toga da li se dužina najdužeg ponavljanja poklapa sa dužinom očekivanog ponavljanja u slučajnoj sekvenci. Osim parametara n i ε imamo i parametar M koji predstavlja dužinu bloka u zavisnosti od dužine bitnog uzorka.

Minimum n	M
128	8
6272	128
750 000	10^4

Primer [12]:

- *Ulazni parametri:*
 $\varepsilon = 11001001000011111101101010100010001000010110100011$
 $00001000110100110001001100011001100010100010111000110$
 $0111001101101100010110010$
 $n = 128$
 $M = 8$
- *Obrada:*



Maksimalno ponavljanje podbloka	Maksimalno ponavljanje podbloka
11001100 (2)	00010101 (1)
01101100 (2)	01001100 (2)
11100000 (3)	00010101 (1)
01001101 (2)	01001100 (2)
00010011 (2)	00000010 (1)
10000000 (1)	11010111 (3)
11001100 (2)	11100110 (3)
11011000 (2)	10110010 (2)

$$v_0 = 4; v_1 = 9; v_2 = 0; v_4 = 4; \chi^2 = 4.882457$$

- Izlazni parametar
 $P = 0.180609$

- Zaključak:
Na osnovu uslova $P \geq 0.01$, smatra se da je sekvenca slučajna.

3.7.5 Ispitivanje stanja binarne matrice (engl. *Binary Matrix Rank Test*)

Ovim testom proveravamo linearnu zavisnost između podnizova fiksne dužine iz originalnog niza.

Primer [12]:

- Ulazni parametri:
 $\varepsilon = 11001001000011111101101010100010001000010110100011$
 $00001000110100110001001100011001100010100010111000$
 $n = 100$
- Obrada:
 $N_1 = 46$
 $N_0 = 47.5$
 $d = -1.376494$
- Izlazni parametar:
 $P = 0.168669$
- Zaključak:
Na osnovu uslova $P \geq 0.01$, smatra se da je sekvenca slučajna.



3.7.6 Ispitivanje diskretne Furijeove transformacije (engl. *Discrete Fourier Transform (Spectral) Test*)

Fokus ovog testa predstavlja najviše vrednosti diskretne Furijeove transformacije niza. Svrha ovog testa je da otkrije periodične funkcije u testiranom nizu koje bi ukazale na odstupanje od pretpostavke o slučajnosti. Namera je da se otkrije da li je broj najviših vrednosti koje prelaze 95% značajno različit od preostalih 5%.

Osim već korišćenih parametara za dužinu binarnog uzorka i samog uzorka, koriste se i dodatni parametri, koji definišu izgled matrice, parametar M koji označava broj redova u matrici i parametar Q koji označava broj kolona.

Primer [12]:

- *Ulazni parametri:*
 $\varepsilon = \text{prvih } 1000000 \text{ binarnih vrednosti proširenje}$
 $n = 100000$
 $M = Q = 32$
- *Obrada:*
 $N = 97$
 $F_M = 23; F_{M-1} = 60; N - F_M - F_{M-1} = 14$
 $\chi^2 = 1.2619656$
- *Izlazni parametar:*
 $P = 0.532069$
- *Zaključak:*
Na osnovu uslova $P \geq 0.01$, smatra se da je sekvenca slučajna.

3.7.7 Ispitivanje nepreklapajućih uzoraka (engl. *Non-overlapping Template Matching Test*)

U fokusu ovog testa je učestanost pojave svih n -binarnih uzoraka gde ne dolazi do preklapanja u celom ispitivanom nizu. Test treba da pokaže da li je broj pojava uzoraka koji se ne poklapaju približno jednak očekivanom broju za slučajnu sekvencu.

Osim parametara ε i n , za testiranje se koristi i parametar m koji definiše dužinu uzorka B .

Primer [12]:

- *Ulazni parametri:*
 $\varepsilon = 2^{20}$
 $n = 2^{20}$
 $m = 9$
 $B = 000000001$



- **Obrada:**
 $\mu = 255.984375$
 $\sigma^2 = 247.499999$
 $W_1 = 259; W_2 = 229; W_3 = 271; W_4 = 245; W_5 = 272; W_6 = 262; W_7 = 259; W_8 = 246;$
 $\chi^2(obs) = 5.999377$
- **Izlazni parametar:**
 $P = 0.110434$
- **Zaključak:**
 Na osnovu uslova $P \geq 0.01$, smatra se da je sekvenca slučajna.

3.7.8 Ispitivanje preklapajućih uzoraka (engl. *Overlapping Template Matching Test*)

Karakteristika koja je u fokusu ovog testa je učestanost pojave unapred definisanog niza. Za razliku od prethodnog testa, parametar m , osim što označava dužinu uzorka, označava i broj jedinica u uzorku.

Primer [12]:

- **Ulazni parametri:**
 $\varepsilon = \text{binarnoproširenje}$ do 1000000 bita
 $n = 1000000$
 $m = 9$
 $B = 111111111$
- **Obrada:**
 $v_0 = 329; v_1 = 164; v_2 = 150; v_3 = 111; v_4 = 78; v_5 = 136$
 $\chi^2(obs) = 8.965859$
- **Izlazni parametar**
 $P = 0.110434$
- **Zaključak:**
 Na osnovu uslova $P \geq 0.01$, smatra se da je sekvenca slučajna.

3.7.9 Maurerov univerzalni statistički test (engl. *Maurer's "Universal Statistical" Test*)

Testom se pokušava ispitati mogućnost kompresije niza bez gubitka informacija. Imajući u vidu da se niz koji poseduje periodična svojstva može efikasno kompresovati, očekuje se da se niz slučajnih bitova ne može kompresovati.



Parametri ovog testa su:

- L - dužina svakog bloka;
- Q - broj blokova u inicijalnoj sekvenci;
- n - dužina bitnog uzorka;
- ε - uzorak.

Primer [12]:

- *Ulazni parametri:*

$\varepsilon = \text{binarni niz generisan pomoću } G - \text{SHA} - 1$

$n = 1048576$

$L = 7$

$Q = 1280$

- *Obrada:*

$c = 0.591311; \sigma = 0.002703; K = 148516; \text{sum} = 919924.038020$

$f_n = 6.194107; \text{expectedValue} = 6.196251; \sigma = 3.125$

- *Izlazni parametar*

$P = 0.427733$

- *Zaključak:*

Na osnovu uslova $P \geq 0.01$, smatra se da je sekvenca slučajna.

3.7.10 Ispitivanje linearne kompleksnosti (engl. *Linear Complexity Test*)

Ovaj test se koristi za dobijanje dužine linearnog pomeračkog registra (LFSR). Svrha ovog testa je utvrđivanje da li je sekvenca dovoljno kompleksna da se može smatrati slučajnom.

Parametri ovog testa su, kao i u većini testova do sada: ε , n , M i novi parametar K , koji označava broj stepeni slobode.

Primer [12]:

- *Ulazni parametri:*

$\varepsilon = \text{prvih } 1000000 \text{ binarnih vrednostiproširenja } \varepsilon$

$n = 1000000$

$M = 1000$

- *Obrada:*

$v_0 = 11; v_1 = 31; v_2 = 116; v_3 = 501; v_4 = 258; v_5 = 57; v_6 = 26$

$\chi^2(\text{obs}) = 2.700348$



- *Izlazni parametar*
 $P = 0.845406$
- *Zaključak:*
Na osnovu uslova $P \geq 0.01$, smatra se da je sekvenca slučajna.

3.7.11 Serijski test (engl. *Serial Test*)

Pomoću ovog testa utvrđuje se učestanost svih mogućih preklapanja n -bitnog niza u čitavoj sekvenci.

Osim parametara ε i n , imamo i parametar m , koji predstavlja dužinu svakog bloka.

Primer [12]:

- *Ulazni parametri:*
 $\varepsilon = \text{prvih } 1000000 \text{ binarnih vrednosti proširenja}$
 $n = 1000000$
 $m = 2$
- *Obrada:*
 $\#0s = 499971$
 $\#1s = 500029$
 $\#00s = 250116$
 $\#01s = \#10s = 249855$
 $\#11s = 250174$

$$\psi_2^2 = 0.343128; \psi_1^2 = 0.003364; \psi_0^2 = 0.000000;$$

$$\nabla\psi_2^2 = 0.339764; \nabla^2\psi_2^2 = 0.336400;$$

- *Izlazni parametar*
 $P_1 = 0.843764$
 $P_2 = 0.561915$
- *Zaključak:*
Na osnovu uslova $P \geq 0.01$, smatra se da je sekvenca slučajna za oba testa.

3.7.12 Ispitivanje približne entropije (engl. *Approximate Entropy Test*)

Fokus ovog testa je identičan fokusu prethodnog testa - posmatra se učestanost svih mogućih preklapajućih n -bitnih uzoraka u nizu.

Parametri ovog testa su takođe isti kao i u prethodnom.

Primer [12]:



- *Ulazni parametri:*
 $\varepsilon = 11001001000011111101101010100010001000010110100011$
 $00001000110100110001001100011001100010100010111000$
 $n = 100$
 $m = 2$
- *Obrada:*
 $ApEn(m) = 0.665393$
 $\chi^2(obs) = 5.550792$
- *Izlazni parametar*
 $P = 0.235301$
- *Zaključak:*
Na osnovu uslova $P \geq 0.01$, smatra se da je sekvenca slučajna.

3.7.13 Ispitivanje slučajnih zbirova (engl. *Cumulative Sums Test*)

Svrha ovog testa je da se utvrdi da li je kumulativni zbir parcijalnih sekvenci koje se javljaju u nizu suviše veliki ili mali u odnosu na očekivano ponašanje kumulativnog zbira slučajnih sekvenci.

Osim parametara ε i n , za izvršenje testa, potreban je i parametar *mode* koji može imati vrednost 0 ili 1.

Primer [12]:

- *Ulazni parametri:*
 $\varepsilon = 11001001000011111101101010100010001000010110100011$
 $00001000110100110001001100011001100010100010111000$
 $n = 100$
 $mode = 0 \parallel mode = 1$
- *Obrada:*
 $z = 1.6 \parallel z = 1.9$
- *Izlazni parametar*
 $P = 0.219194 \parallel P = 0.114866$
- *Zaključak:*
Na osnovu uslova $P \geq 0.01$, smatra se da je uslov ispunjen za oba testa i da je sekvenca slučajna.



3.7.14 Ispitivanje slučajne digresije (engl. *Random Excursions Test*)

Svrha ovog testa je da se utvrdi da li broj određenih stanja u okviru ciklusa odstupa od očekivanog. Ovaj test čini niz od osam testova, jedan test i zaključak za svako stanje: -4, -3, -2, -1, +1, +2, +3, +4.

Primer [12]:

- *Ulazni parametri:*
 $\varepsilon = \text{binarnoproširenje}$ do 1000000 bita
 $n = 1000000$
- *Obrada:*
 $J = 1490$

Stanje = x	x^2	P	Zaključak
-4	3.835698	0.573306	slučajan
-3	7.318707	0.197996	slučajan
-2	7.861927	0.164011	slučajan
-1	15.692617	0.007779	nije slučajan
1	2.485906	0.778616	slučajan
2	5.429381	0.365752	slučajan
3	2.404171	0.790853	slučajan
4	2.393928	0.7923378	slučajan

- *Zaključak:*
 U sedam od osam stanja uslov $P \geq 0.01$ je ispunjen, dakle možemo zaključiti da se radi o slučajnoj sekvenci.

3.7.15 Ispitivanje slučajne promenljive digresije (eng. *Random Excursions Variant Test*)

Kao i prethodni, i ovaj test je niz testova, samo što ih sada ima osamnaest. Svrha ovog testa je da se utvrdi ukupan broj stanja u kojima niz premašuje očekivane vrednosti slučajne sekvence.

Primer [12]:

- *Ulazni parametri:*
 $\varepsilon = \text{binarnoproširenje}$ do 1000000 bita
 $n = 1000000$



- *Obrada:*

Stanje x	P	Zaključak
-9	0.858946	slučajan
-8	0.794755	slučajan
-7	0.576249	slučajan
-6	0.493417	slučajan
-5	0.633873	slučajan
-4	0.917283	slučajan
-3	0.934708	slučajan
-2	0.816012	slučajan
-1	0.826009	slučajan
+1	0.137861	slučajan
+2	0.200642	slučajan
+3	0.441254	slučajan
+4	0.939291	slučajan
+5	0.505683	slučajan
+6	0.445935	slučajan
+7	0.512207	slučajan
+8	0.538635	slučajan
+9	0.593930	slučajan

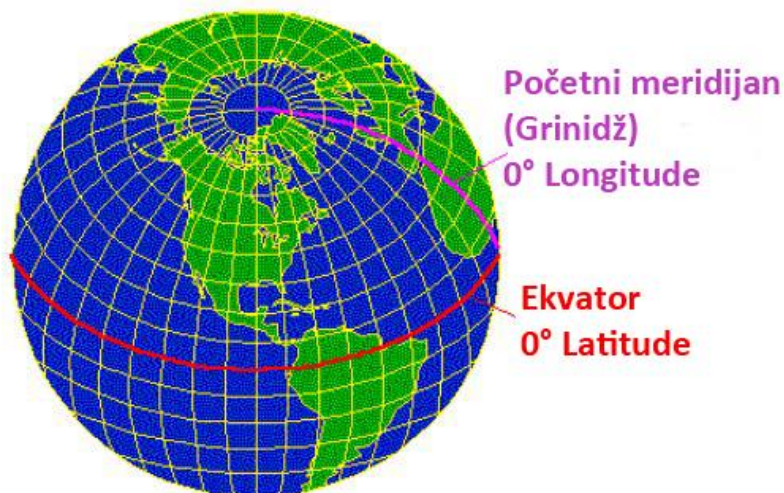
- *Zaključak:*

U svih 18 stanja uslov $P \geq 0.01$ je ispunjen, dakle možemo zaključiti da se radi o slučajnoj sekvenci.



3.9.1.1 Koordinate

Izgled Zemlje u prvoj aproksimaciji se može smatrati loptom. To je pomoglo da bi se postavio koordinatni sistem pomoću kog se mogu odrediti vrednosti svake tačke na Zemlji, pod nazivom geografski ekvatorski koordinatni sistem. Polarna osa u geografskom koordinatnom sistemu se poklapa sa osom rotacije Zemlje i na njenim krajevima se nalaze Severni i Južni pol. Osnovna kružnica geografskih polova (ekvator) deli Zemljinu sferu na južnu i severnu. Manje kružnice paralelne sa ekvatorom nazivaju se paralele, dok se sekundarne kružnice ekvatora nazivaju meridijani. Za početni meridijan, "nulti", izabran je onaj koji prolazi kroz opservatoriju u mestu Grinidž u blizini Londona.



Slika 12 - Ekvator i Grinidž

Osnovne koordinate geografskog ekvatorskog koordinatnog sistema su: [14]

- *geografska širina* (ϕ) je ugao u ravni meridijana meren od presečnog pravca sa ekvatora do pravca ka datoj tački; uzima vrednosti iz intervala $(-90^\circ, +90^\circ)$;
- *geografska dužina* (λ) je ugao diedar između meridijanskih ravni početnog(inicijalnog) meridijana i meridijana date tačke; uzima vrednosti iz intervala $(-180^\circ, +180^\circ)$. Međunarodnim dogovorom definisano je da pozitivan predznak bude za dužine ka istoku, a negativan ka zapadu u odnosu na početni meridijan;
- *nadmorska visina* (H) je visina u metrima date tačke u odnosu na posebno definisanu nivosku površinu(površinu mora).

Dogovoreno je da se umesto negativnih i pozitivnih vrednosti geografskih širina i dužina koriste uvek pozitivne vrednosti uz međunarodne oznake za pol. Umesto geografske širine ϕ i geografske dužine λ , često se koriste latituda ϕ i longituda λ . Takođe treba napomenuti da se u praksi, umesto ugaonog zapisa (stepen, minuti, sekundi), koristi zapis decimalnih stepena.

Na primer geografska širina i dužina Avalskog tornja u blizini Beograda:

- Ugaoni zapis: $44^\circ 41' 45.5'' N$, $20^\circ 30' 52'' E$;
- Decimalni stepeni 44.695972, 20.514444.

dok su geografska širina i dužina statue Isusa Hrista u blizini Rio de Žaneira:

- Ugaoni zapis: $22^\circ 57' 7'' S$, $43^\circ 12' 38'' W$;
- Decimalni stepeni -22.951944, -43.210556.



3.9.2 Navigacioni sistemi – GPS

Dolaženje do tačne vrednosti geografske širine i dužine neke tačke na planeti merenjem ugaonih koordinata i proračunom je dosta komplikovan i dug posao. Kada nam je bilo potrebno da odredimo našu tačnu lokaciju, do pojave navigacionih sistema, morali smo da računamo razna rastojanja od tačno već definisanih mesta. To je iziskivalo vreme. Za vreme Drugog svetskog rata pojavljuju se prvi navigacioni sistemi Loran i Čajka.

3.9.2.1 Nastanak

Nakon što je sovjetsko Ministarstvo za radio-tehničku industriju lansiralo *Sputnik* [15], prvi satelit, u orbitu, 1957. godine, Američki naučnici Viliam Guier i Georg Vinfenbah na univerzitetu Džon Hopkins počeli su da nadgledaju slanje radio talasa koji su dolazili sa satelita Sputnik. Nedugo zatim Ministarstvo odbrane SAD-a osniva agenciju *ARPA* (eng. *Advanced Research Projects Agency*) zaduženu za razvoj novih tehnologija za vojsku SAD-a. 1960. godine Ministarstvo odbrane SAD-a uspešno lansirao svoj prvi satelit *Transit 1B* [16], a agencija *ARPA* aktivira prvi navigacioni sistem baziran na satelitima koji je i dobio ime po satelitu, *Transit*. U to vreme Američko ratno vazduhoplovstvo počinje da razvija svoj sistem *MOSAIC*, kasnije nazvan *621B*. Osim ova dva 1964. godine vojska SAD-a počinje da razvija još jedan sistem navigacije - *SECOR*. *SECOR* je posedovao i predajnike sa zemlje ka satelitima u orbiti. 1973. godine vojska SAD-a objedinjuje sve sisteme navigacije u jedan - *DNSS*, a godinu dana kasnije, nad sistemima navigacije kontrolu preuzima agencija *DARPA* (preimenovana ranije poznata agencija *ARPA*) i ujedinjeni sistem navigacije dobija ime *NAVSTAR GPS*. Desetak godina kasnije, posle pada korejskog aviona i pogibije 269 putnika, američki predsednik Ronald Regan 1983. godine donosi odluku da *NAVSTAR GPS* može preći i u civilnu upotrebu, a kao ime sistema koristiće se ubuduće samo *GPS* [17]. Do danas se sistem nije značajno menjao, jedino se povećao broj satelita.

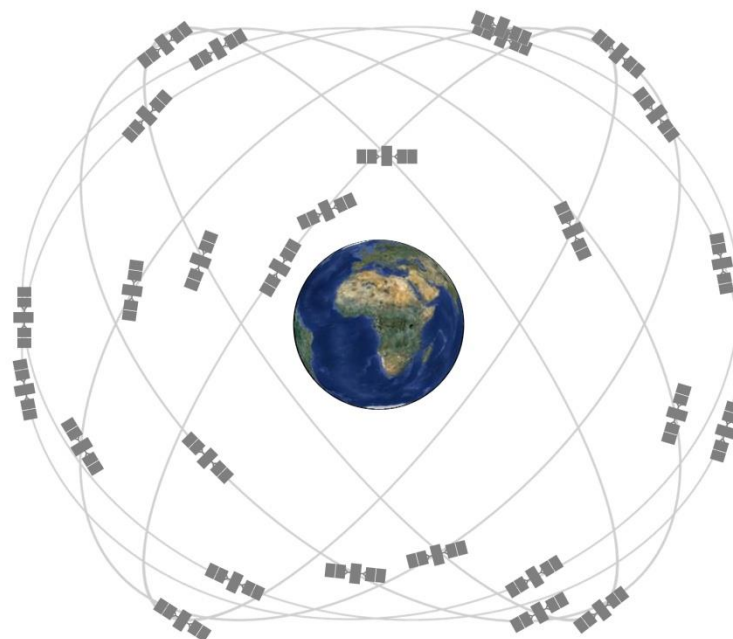
3.9.2.2 Struktura i opis rada sistema

GPS sistem navigacije se sastoji od tri komponente: komponente u vasioni - satelit, kontrolne komponente i korisničke komponente - *GPS* prijemnik.

3.9.2.2.1 Komponenta u vasioni

U Svemiru se vremenom povećavao broj satelita korišćenih za *GPS* sistem navigacije, ali su i sateliti usavršavani pa tako Blok I sateliti, koji su poslani u svemir pre nego što je *GPS* prešao u civilnu upotrebu, zbog zastarelosti su izbačeni iz upotrebe, tako da se sada u orbiti nalaze sateliti Blok II i najnoviji Blok III. Broj satelita je definisan na 24, međutim, zbog redovnih popravki i iznenadnih kvarova, broj satelita je uvek veći pa sada iznosi 32 [18]. Svi se nalaze u orbiti po tačnom rasporedu na oko 20,200 km nadmorske visine i svaki dan dva puta obišu oko Zemlje. Tačna putanja satelita omogućava da se iznad bilo koje tačke na Zemlji nalaze bar četiri satelita.

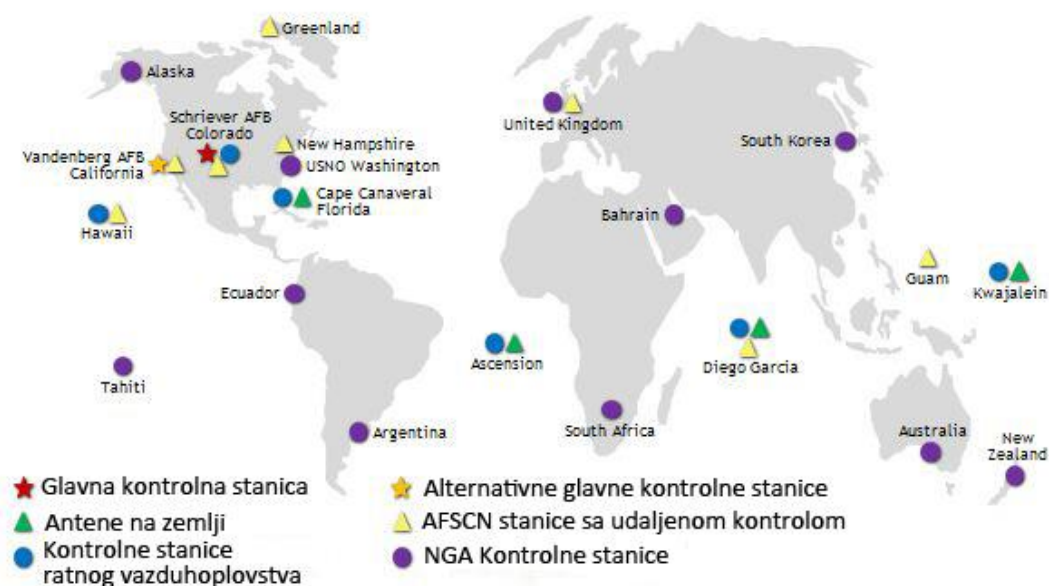




Slika 13 - Komponenta u vasioni - Satelit

3.9.2.2 Kontrolna komponenta

Kontrolu rada i sinhronizaciju satelita obavlja kontrolna komponenta. Nju čine stanice za praćenje satelita, kontrolne stanice i zemljišne antene. Kao što se vidi sa slike (**Error! Reference source not found.**) stanica za praćenje satelita ima širom sveta i sve one alju podatke kontrolnoj stanici u Kolorado Springsu (SAD). Nakon obrade podataka, podaci se pomoću zemljišnih antena šalju satelitima. Svakodnevno se dva puta satelitima šalju podaci o njihovoj tačnoj poziciji i vremenu i na taj način se vrše precizna podešavanja sistema.



Slika 14 - Kontrolne komponente na zemlji



3.9.2.2.3 Korisnička komponenta - GPS prijemnik

Do 2000. godine broj prijemnika na Zemlji bio je oko 1,5 miliona. Sada se više taj broj ne može odrediti. GPS tehnologija se danas koristi u svim životnim sferama. GPS prijemnika ima u skoro svim sredstvima za transport, GPS prijemnici se koriste i u geodeziji za lakša merenja na zemlji, ima ih i u sportu, ali najviše u IT industriji. Danas skoro svaki prenosni uređaj (mobilni telefoni, PDA, laptopovi itd.) poseduje integrisani GPS prijemnik, dok uređaju na bazi veštačke inteligencije mogu donositi odluke pomoću GPS koordinata.

Sam GPS prijemnik se sastoji od antene, mikroprocesora koji obrađuje signale i kristalnog oscilatora koji služi kao časovnik. Antena prikuplja radio-talase koje emituje satelit. Znajući da je poznata brzina putovanja radio-talasa, a časovnici su sinhronizovani, mikroprocesor može da izračuna tačnu udaljenost od određenog satelita. Da bi GPS prijemnik mogao da donese zaključak o tačnoj lokaciji na Zemlji, potrebno je da primi radio signal od bar tri satelita istovremeno, naravno za bolju preciznost potreban je što veći broj satelita. Stariji GPS prijemnici imali su šest kanala i mogli su da primaju istovremeno radio signal sa najviše šest satelita (jedan kanal, jedan satelit), danas GPS prijemnici imaju i do dvadeset kanala. GPS prijemnici, osim tačne geografske širine i dužine, kao i nadmorske visine, mogu, ukoliko se pomeraju, da proračunaju i kojom se brzinom kreću.

Nekoliko različitih tipova GPS prijemnika je prikazano na slici.



Slika 15 - GPS prijemnici

3.9.2.3 GNSS

Nakon 2000. godine, Rusija počinje intenzivno da unapređuje tada već zastareli i zaboravljeni sistem GLONASS i omogućava civilnu upotrebu. Nagli razvoj ovog sistema učinio ga je konkurentnim sistemu GPS, pa su i veliki svetski proizvođači prenosnih uređaja (Apple, Samsung itd.) počeli, krajem dvehiljaditih, da omogućavaju navigaciju koja će



paralelno moći da prikuplja podatke i sa *GPS* i sa *GLONASS* satelita. Ovim korakom loše karakteristike *GPS* sistema u zatvorenom prostoru nadoknađene su pomoću *GLONASS* realizacije koja uspešno prevazilazi ovu prepreku.

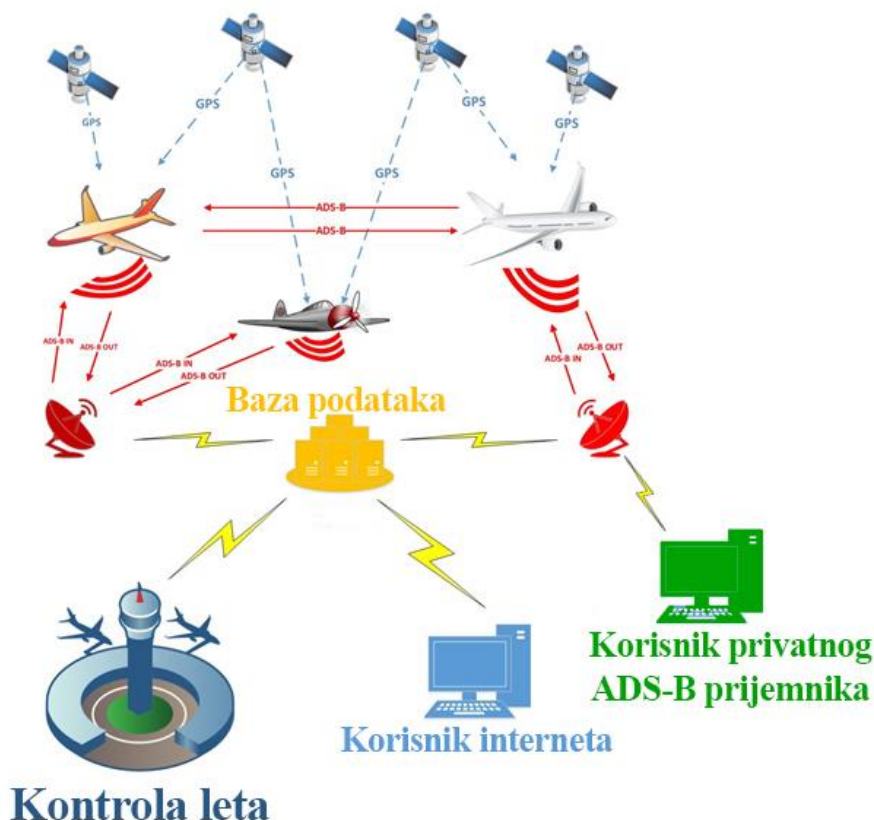
U aprilu 2013, dogovorom ruskih i američkih agencija, stvara se novi sistem *GNSS* (Globalni navigacioni satelitski sistem; engl. *Global Navigation Satellite System*) koji objedinjuje postojeće *GPS* i *GLONASS*, a od 2020. godine ovom sistemu će se priključiti još dva sistema koji su trenutno u razvoju, kineski *BeiDou* i evropski *Galileo*. Sve ovo neophodno je zbog daljeg unapređivanja i sve veće preciznosti sistema.

3.9.3 Sistem za praćenje civilnog avio saobraćaja – ADS-B

Kontrola leta počinje još 1921. godine. Kontrolni tornjevi se nakon toga naglo šire i danas ih poseduje veliki broj aerodroma. Nadgledanje aviona svodilo se na radarska osmatranja i na taj način svaki toranj je imao kontrolu nad nekom teritorijom. Međutim, kako je broj letova naglo rastao, naučnici su se trudili da omoguće kontrolu nad što većom teritorijom. S obzirom na to da je u vazduhoplovstvu veoma bitno održavati razdaljinu između aviona i na pravi način proračunati putanje, bilo je potrebno imati širu sliku o samim letelicama na nebu kao i da promene pozicija aviona brže dospeju do kontrolnih tornjeva, imajući u vidu da je za radarski sistem neophodno čitavih dvanaest sekundi da razmeni podatke sa avionom.

Razvojem veštačke inteligencije i *GPS* sistema i njihovom integracijom u vazduhoplovstvo, javlja se ideja da bi avioni mogli sami kontinualno da šalju signale na zemlju, sa podacima o njihovoj trenutnoj poziciji. Otišlo se i korak dalje, te su tako na avione, osim predajnika, smeštani i prijemnici, koji treba da prikupljaju podatke o avionima u njihovoj blizini. Tako nastaje *ADS-B* (eng. *Automatic dependent surveillance-broadcast*).





Slika 16 - ADS-B sistem

Kao što je prikazano na slici (**Error! Reference source not found.**), avioni prikupljaju podatke sa satelita GPS (sada sa GNSS) sistema, međutim, osim da samo prikažu pilotima obrađene podatke o poziciji aviona, oni ih priključuju podacima o broju leta, odredištu i polazištu i pomoću ADS-B predajnika ih šalju. Te podatke prikupljaju prijemnici na avionima u blizini i zajedno sa podacima o njihovoj poziciji dostavljaju pilotima. Osim aviona, signal sa ADS-B predajnika se prikuplja i na zemlji. Međunarodnim dogovorom podaci civilnog vazduhoplovstva su slobodni, što je pomoglo širenju ADS-B sistema. Kako bi se pokrila čitava teritorija planete Zemlje, osim antena postavljenih u okviru kontrola leta, postoji i veliki broj privatnih amaterskih antena instaliranih i prijavljenih po ADS-B kriterijumu i povezanih putem Interneta sa globalnim bazama podataka. Ovakav pristup omogućio je skoro potpunu pokrivenost Evrope, Azije, Severne Amerike i Australije, kao i dobru pokrivenost iznad okeana, dok su delovi ostalih kontinenata delimično pokriveni. Uvidevši rasprostranjenost i efikasnost sistema, međunarodne agencije za kontrolu leta su počele sve više da implementiraju ovakav vid sistema, kako bi zamenio dosada korišćene radarske sisteme.

ADS-B sistem se sastoji iz dva dela, "In" i "Out". ADS-B "out" sistem, najčešće samo ADS-B, kao što je već opisano, emituje radio signal sa aviona u okviru kog se nalaze podaci o trenutnoj poziciji aviona, brzini, kao i podaci relevantni za taj let, broj leta, tip aviona, polazište i odredište, ali i podatak o vremenskom trenutku kada je poslednji put izračunata pozicija aviona. Dok ADS-B "in" sistem predstavlja sistem pomoću koga se avionima dostavljaju podaci od kontrole leta sa zemlje. Osim podataka o ostalim avionima u blizini u

okviru tog radio signala, nalaze se i podaci o vremenskim uslovima. Na taj način olakšava se kontrola letenja i poboljšava se sigurnost.

ADS-B sistemi rade na radio frekvenciji 1090 MHz, čime je omogućeno da se vrlo lako može doći do podataka. To je i razlog što veliki broj privatnih internet sajtova poseduju svoje baze podataka i na taj način omogućavaju da korisnici širom interneta prate letove aviona u realnom vremenu. Međutim, usled pravila američke Agencije za kontrolu leta (FAA), svi letovi do 2017. godine za amatersko praćenje prikazuju se sa kašnjenjem od 5 minuta. Osim običnih korisnika, i manji aerodromi, koji do sada nisu imali svoju kontrolu leta, uz pomoć *ADS-B* sistema sada mogu pratiti letove.

Generalno gledano, prednosti *ADS-B* sistema su [19]:

- Veća sigurnost - podaci su u realnom vremenu dostupni i kontroli leta i pilotima, bez obzira na vremenske uslove ili geografsku poziciju, što omogućava sigurniji let kao i poletanje i sletanje;
- Poboljšana efikasnost - usled poznavanja trenutnog zagušenja određenih ruta, moguće je prilagođavati letove i odabir najboljih ruta samim tim štedeti vreme i novac;
- Infrastruktura - umesto velikog broja radara na raznim lokacijama koji nisu mogli da pokriju veliku teritoriju, jeftine antene prikupljaju podatke i distribuiraju ih ka kontrolama leta.

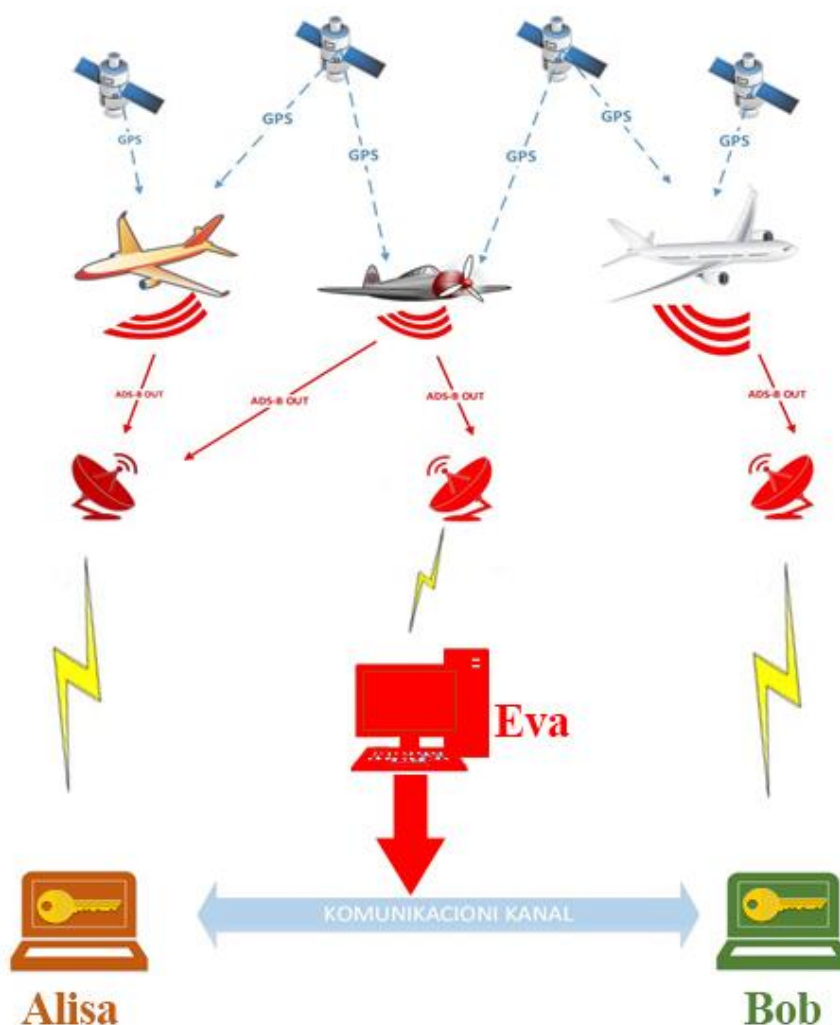
Ipak i pored ovih prednosti, broj aviona koji poseduje *ADS-B* predajnike je oko 60%. Evropski Er Bas (engl. *AirBus*) je jedina kompanija koja na svim svojim avionima ima instalirane ove predajnike. Međutim, broj aviona svakodnevno raste i može se reći da su avioni koji danas ne poseduju *ADS-B* predajnike uglavnom starije proizvodnje. Prema pravilniku FAA do 2020. svi avioni iznad SAD-a moraće da poseduju *ADS-B* predajnike.

Detaljna obrada podataka koji se nalaze u signalu koji avion šalje biće navedena i obrađena u nastavku rada.

4. PREGLED PREDLOŽENOG REŠENJA

Nakon teorijske analize svih segmenata potrebnih za jednu bezbednu komunikaciju dolazimo i do predstavljanja praktičnog rešenja. U ovom poglavlju detaljno ćemo obrazložiti ponuđeno rešenje, prikazati primer realizacije ali i predstaviti rezultate sprovedenih testova nad binarnim nizovima dobijenim pomoću ovakvog rešenja.

4.1 KOMPLETNA ŠEMA PREDLOŽENOG REŠENJA



Slika 17 - Kompletna šema predloženog rešenja

Pogledom na šemu kompletnog rešenja (Slika 17) jasno se nameće da je realizacija samog rešenja podeljena u nekoliko faza. Podatke je svakako neophodno prvo prikupiti, obraditi ih, kodovati, odrediti da li ih obe legitimne strane komunikacije poseduju i zatim ih predstaviti kao ključ potreban za šifrovanje komunikacije.

4.2 DIZAJN PROTOKOLA

4.2.1 Prva faza

Prva faza predstavlja individualno prikupljanje radio talasa svakog aktera komunikacije.

Kao što je već objašnjeno, GPS sateliti emituju radio signal koji, među ostalima, prikupljaju i vazduhoplovi. U ovom radu orijentisaćemo se samo na avione civilnog vazduhoplovstva. Avioni prikupljaju signal sa satelita i uz pomoć *GPS* instrumenata određuju svoju trenutnu poziciju. Osim što se na instrumentima pilotima prikazuje trenutna pozicija i što ih uređaji poput auto-pilota koriste, svoju trenutnu poziciju avioni dalje distribuiraju uz pomoć *ADS-B* sistema. Osim *GPS* sistema, u prethodnom poglavlju smo detaljno opisali i rad *ADS-B* sistema koji predstavlja glavni izvor informacija ovog rešenja.

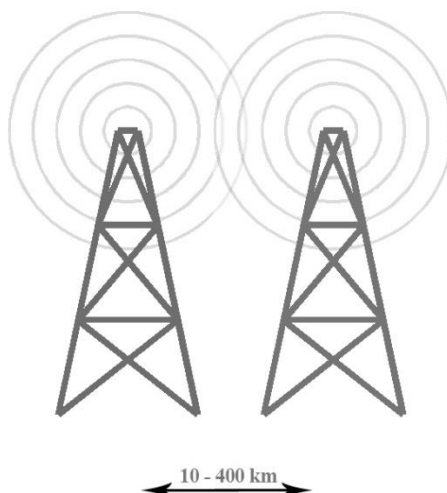
Prikupljanje podataka iz javnih ili privatnih baza u koje se smeštaju trenutni podaci *ADS-B* sistema osim što može biti protivzakonito, u neku ruku može biti i veoma opasno sa stanovišta bezbednosti. Napadač može imati kontrolu nad bazom i dirigovati koje rezultate ćemo prikupljati. Iz tog razloga, realizacija ovakvog sistema ne bi smela da se bazira na prikupljanju podataka sa nebezbednih lokacija već prikupljanjem podataka direktno od emitera (aviona). Mana ovakvog prikupljanja podataka svakako je mnogo manji broj aviona koji se mogu pratiti u jedinici vremena, jer sada više ne prikupljamo podatke iz baze u koju podaci dolaze sa hiljade prijemnih antena širom sveta već samo sa jedne. Osim što se smanjuje broj aviona, smanjuje se i razdaljina na kojoj se akteri komunikacije mogu nalaziti, ona je sada direktno zavisna od kvaliteta *ADS-B* prijemnika, pozicije prijemnika i varira od nekih desetak do četiri stotine kilometara vazdušne linije. Međutim, osim mana ovakav sistem prikupljanja ima višestruku prednost. Kao što je rečeno, podaci se prikupljaju direktno od emitera, napadač mora da poseduje opremu za prikupljanje i mora da bude na lokaciji u blizini aktera komunikacije.





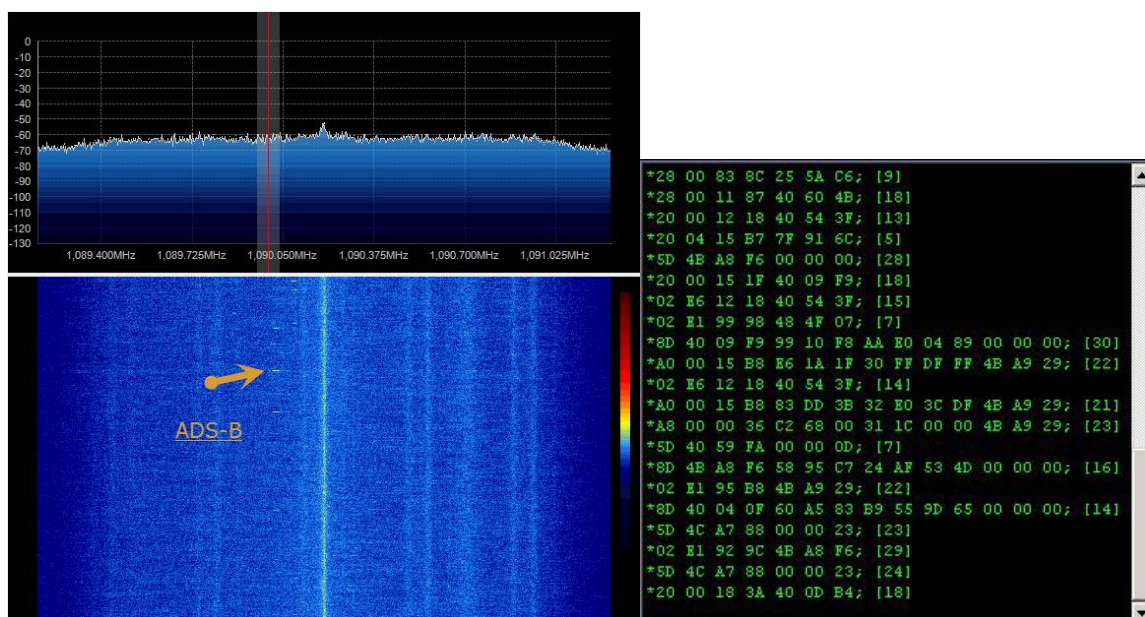
Slika 18 - ADS-B prijemnici

Prikupljanje podataka uz pomoć ADS-B sistema izvodi se radio-talasnim antenama projektovanim za kružno prikupljanje radio talasa (OMNI antene) i dekomerima radio signala. Vizuelni prikaz ovakvog prikupljanja podataka bio bi krug sa centrom, gde centar kruga predstavlja antenu a kružnica aproksimaciju granice maksimalnog dometa antene. Na slici (Slika 19) su predstavljene dve antene, kružnice oko antena predstavlja površinu sa koje se prikupljaju podaci. Presek kružnica predstavlja zajedničku površinu. Te dve antene sa kružnicama u stvari predstavljaju aktere naše komunikacije, dva legitimna učesnika (Alisa i Bob), a zajednička površina su avioni koji su u dometu i Alise i Boba. Zlonamerni akter – napadač (Eva) bi trebalo da se umetne tako da prikuplja podatke iz površine preseka.



Slika 19 - ADS-B antene (presek signala)

Na slici (Slika 20) je prikazan uhvaćeni radio signal koji nakon dekodovanja daje poruke vezane za letove čiji je signal primljen. Avioni emituju radio signal na svakih 15 ms, te je to vreme potrebno za prikupljanje novih podataka o letu.



Slika 20 - Radio talas i kodne poruke

4.2.2 Druga faza

Druga faza predstavlja kodovanje podataka o letovima.

Za ovu fazu se može reći da joj predstoji jedna pred faza. Pre kodovanja, odnosno predstavljanja podataka leta binarno potrebno je odrediti koji podaci se mogu koristiti. Tekstualne poruke dobijene nakon dekodovanja radio signala mogu se razvrstati na osam različitih poruka u zavisnosti koji podatak sa sobom nose. Svih osam poruka u sebi poseduju 24 bitnu vrednost predstavljenu heksadecimalno koja označava registraciju emitera na avionu, a osim te vrednosti u svim porukama nalazi se i vreme emitovanja same poruke. Prva poruka je poruka koja u sebi nosi jedinstveni kod leta, druga poruka sadrži geografsku

poziciju aviona dok se nalazi na zemlji (neposredno pre poletanja odnosno posle sletanja), dok treća poruka u sebi sadrži geografsku poziciju aviona koji je u fazi letenja, poletanja ili sletanja. Četvrta poruka vezana je za horizontalnu i vertikalnu brzinu kretanja, dok su peta, šesta, sedma i osma poruka kontrole ili razmene podataka među avionima. Jasno je da jedini vremenski promenljiv podatak se nalazi u trećoj poruci, odnosno jedino što je promenljivo pri letu aviona jesu podaci njegove geografske širine i dužine, ali ne i nadmorske visine. Upravo ta dva podatka će biti korišćena u za realizaciju ovog rešenja.

Podatak geografske širine i dužine predstavljen je u decimalnom zapisu od pet do osam decimala, zavisno od kvaliteta dekodera. Ceo broj pre zareza označava veliku površinu na zemlji i on se odbacuje, tako da se za kodovanje koristi isključivo decimalni zapis. Radi usaglašavanja sa svim dekoderima za postupak kodovanja koriste se samo prve četiri decimale.

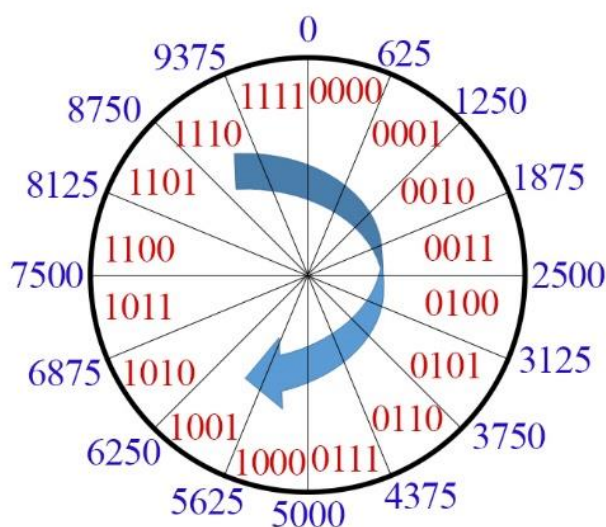
Binarnizacija, odnosno kodovanje decimala geografske širine i dužine radi se odvojeno. Prvo se sa četiri bita predstavi decimalna vrednost geografske dužine, a zatim sa nova četiri bita vrednost geografske širine.

Let 1: JU123; Beograd - Sofija
MSG,3,333,44,4C01E2,144,2014/10/24,12:03:08.913,2014/10/24,12:03:08.913,,20750,,,44. 2363 1,20. 1245 3,,,0,0,0,0
Let 2: DLH9XV, Atina - Minhen
MSG,3,333,115,3C6602,215,2015/01/25,13:23:01.075, 2015/01/25,13:23:01.075,,6575,,,48. 383 12,12. 311 56,,,0,0,0,0

Tabela 2 - Dekodovane III poruke dva leta

Ukoliko izdvojimo dve treće poruke dva različita leta (Tabela 2) vrednosti koje se koduju su prikazane u tabeli (Tabela 2) crvenom bojom.

Za kodovanje se koristi fazor koji prema dizajnu odgovara jedno-dimenzionalnom Gaborovom filtru. Faza kodovanja sprovedena je nad normalizovanim podacima sa 1D Log-Gabor filterom. Sa svakim filtrom se dobija četiri bita podatka iz jednog dela fazora. Izlaz iz svakog fazora je podešen tako da kad se ide iz jednog u drugi, menja se samo 1 bit. Ovakva kvantifikacija, obezbediće konzistentniju informaciju, jer smanjuje korelisanost između ulaznih podataka.



Slika 21 - Fazor



Kodovanjem podatka iz tabele (Tabela 2) prema fazoru sa slike (Slika 21), za let JU123 dobijamo bitsku predstavu **00110001**, a za let DLH9XV dobijamo **01100101**.

4.2.3 Treća faza

Treća faza predstavlja kreiranje i popunjavanje matrica, kao i konverzija matrice u niz.

Nakon prikupljanja i kodovanja podataka usled različite površine obuhvatanja nebeskog prostora, odnosno usled broja letova od kojih su radio signali primljeni, dužina binarnog niza nastalog spajanjem binarnih predstava letova je različita. Upoređivanje nizova različitih dužina i pronalaženja zajedničkih delova je izuzetno komplikovana operacija. Rešenje stoji u predstavljanju geografske površine na kojoj se nalaze avioni čiji je koordinate moguće prikupiti pomoću matrica, tj. vrednosti geografske dužine bi predstavljale kolone, a širine redove te matrice. U ovom rešenju strane koje komuniciraju, kreiraju lokalne kvadratne matrice (100x100) u čija se polja upisuju binarne vrednosti dobijene kodovanjem. Kolonu i red u koji upisujemo binarni podatak dobijen kodovanjem određenog leta govori nam vrednost geografske dužine i širine tog leta u trenutku kodovanja, i to, prve dve decimale geografske dužine se odnose na broj kolone, a prve dve decimale geografske širine na broj reda. Za već pomenuti JU123 binarna predstava biće upisana na poziciju u matrica M, M[23,12], a let DLH9XV na poziciju M[38,31]. Ovakvim načinom popunjavanja, možemo reći da očekujemo da matrice neće biti popunjene na svim pozicijama, te svaki od aktera komunikacije prazne pozicije popunjava lokalno generisanim slučajnim nizom takođe od osam bitova.

Na ovaj način unošenjem lokalne slučajnosti onemogućujemo zlonamernog učesnika komunikacije da pretpostavkom o svim mogućim lokacijama generiše matrice i da na taj način pokuša da sazna koje avione prate legitimni učesnici.

Sada na obe strane komunikacije imamo matrice sa po 10000 elemenata gde svaki element ima osam bitova. Ukoliko matrice konvertujemo u jednodimenzionalni niz na obe strane komunikacije dobićemo niz od po 80000 bitova čime smo rešili problem nizova različitih dužina.

Matricu ćemo predstaviti kao niz tako što ćemo na kraj svakog reda dodati naredni red, tj. iz matrice M element M[1,100] će u nizu N imati vrednost N[100], a naredni element niza N[101] će biti element matrice M[2,1] i sve tako do elementa M[100,100].

Za potrebe ove faze napisana je JAVA aplikacija čiji kod prikazan na slikama: Slika 22, Slika 23 i Slika 24.

```

13 *
14 * @author Milomir Tatovic
15 */
16 public class CreateSbitMatrix {
17
18     public static void main(String[] args) throws FileNotFoundException, IOException {
19         FileInputStream fstream = new FileInputStream("ADB-S_Data");
20         DataInputStream in = new DataInputStream(fstream);
21         BufferedReader br = new BufferedReader(new InputStreamReader(in));
22         PrintWriter writer = new PrintWriter(new FileWriter("Matrix"));
23         String strLine = null;
24         String[] tokens;
25         String[] tokens2;
26         String[] tokens3;
27         String delims = "[.]";
28         String delims2 = "[.]";
29         double Duz, Sir, M, N, FDuz, FSir = 0;
30         int VelicinaMatrice = 100;
31         String [][] PrimarMatrix = new String [VelicinaMatrice][VelicinaMatrice];
32
33         while ((strLine = br.readLine()) != null) {
34             int PM_i = 0, PM_k = 0;
35             String Temp = "";
36             tokens = strLine.split(delims);
37             tokens2 = tokens[14].split(delims2);
38             tokens3 = tokens[15].split(delims2);
39             Duz = Double.parseDouble(tokens2[1].substring(0, 4));
40             N = Double.parseDouble(tokens2[1].substring(2, 4));
41             Sir = Double.parseDouble(tokens3[1].substring(0, 4));
42             M = Double.parseDouble(tokens3[1].substring(2, 4));
43             FDuz = Double.parseDouble(tokens2[0]);
44             FSir = Double.parseDouble(tokens3[0]);
45             System.out.print(FDuz); System.out.print(" "); System.out.println(Duz);
46             System.out.print(FSir); System.out.print(" "); System.out.println(Sir);
47
48             FDuz = FDuz%2;
49             FSir = FSir%2;
50             Temp = "";

```

Slika 22 - Java kod I deo



```

51
52     if ((0<=Duz) && (Duz<625))      {Temp = Temp + "0000";}
53     else if ((625<=Duz) && (Duz<1250))  {Temp = Temp + "0001";}
54     else if ((1250<=Duz) && (Duz<1875))  {Temp = Temp + "0010";}
55     else if ((1875<=Duz) && (Duz<2500))  {Temp = Temp + "0011";}
56     else if ((2500<=Duz) && (Duz<3125))  {Temp = Temp + "0100";}
57     else if ((3125<=Duz) && (Duz<3750))  {Temp = Temp + "0101";}
58     else if ((3750<=Duz) && (Duz<4375))  {Temp = Temp + "0110";}
59     else if ((4375<=Duz) && (Duz<5000))  {Temp = Temp + "0111";}
60     else if ((5000<=Duz) && (Duz<5625))  {Temp = Temp + "1000";}
61     else if ((5625<=Duz) && (Duz<6250))  {Temp = Temp + "1001";}
62     else if ((6250<=Duz) && (Duz<6875))  {Temp = Temp + "1010";}
63     else if ((6875<=Duz) && (Duz<7500))  {Temp = Temp + "1011";}
64     else if ((7500<=Duz) && (Duz<8125))  {Temp = Temp + "1100";}
65     else if ((8125<=Duz) && (Duz<8750))  {Temp = Temp + "1101";}
66     else if ((8750<=Duz) && (Duz<9375))  {Temp = Temp + "1110";}
67     else if ((9375<=Duz) && (Duz<10000)) {Temp = Temp + "1111";}
68
69     if ((0<=Sir) && (Sir<625))      {Temp = Temp + "0000";}
70     else if ((625<=Sir) && (Sir<1250))  {Temp = Temp + "0001";}
71     else if ((1250<=Sir) && (Sir<1875))  {Temp = Temp + "0010";}
72     else if ((1875<=Sir) && (Sir<2500))  {Temp = Temp + "0011";}
73     else if ((2500<=Sir) && (Sir<3125))  {Temp = Temp + "0100";}
74     else if ((3125<=Sir) && (Sir<3750))  {Temp = Temp + "0101";}
75     else if ((3750<=Sir) && (Sir<4375))  {Temp = Temp + "0110";}
76     else if ((4375<=Sir) && (Sir<5000))  {Temp = Temp + "0111";}
77     else if ((5000<=Sir) && (Sir<5625))  {Temp = Temp + "1000";}
78     else if ((5625<=Sir) && (Sir<6250))  {Temp = Temp + "1001";}
79     else if ((6250<=Sir) && (Sir<6875))  {Temp = Temp + "1010";}
80     else if ((6875<=Sir) && (Sir<7500))  {Temp = Temp + "1011";}
81     else if ((7500<=Sir) && (Sir<8125))  {Temp = Temp + "1100";}
82     else if ((8125<=Sir) && (Sir<8750))  {Temp = Temp + "1101";}
83     else if ((8750<=Sir) && (Sir<9375))  {Temp = Temp + "1110";}
84     else if ((9375<=Sir) && (Sir<10000)) {Temp = Temp + "1111";}
85
86     int D = (int)(Duz/100);
87     int S = (int)(Sir/100);
88     PrimarMatrix [D][S]= Temp;
89 }

```

Slika 23 - Java kod II deo



```

90
91     for (int temp1 = 0; temp1 < VelicinaMatrice; temp1++){
92     for (int temp2 = 0; temp2 < VelicinaMatrice; temp2++){
93         if (PrimarMatrix [temp1][temp2]==null){
94
95             Random generator = new Random();
96             int i0 = generator.nextInt(2);
97             int i1 = generator.nextInt(2);
98             int i2 = generator.nextInt(2);
99             int i3 = generator.nextInt(2);
100            int i4 = generator.nextInt(2);
101            int i5 = generator.nextInt(2);
102            int i6 = generator.nextInt(2);
103            int i7 = generator.nextInt(2);
104
105            String a0 = Integer.toString(i0);
106            String a1 = Integer.toString(i1);
107            String a2 = Integer.toString(i2);
108            String a3 = Integer.toString(i3);
109            String a4 = Integer.toString(i4);
110            String a5 = Integer.toString(i5);
111            String a6 = Integer.toString(i6);
112            String a7 = Integer.toString(i7);
113
114            PrimarMatrix [temp1][temp2]= " "+a0+" "+a1+" "+a2+" "+a3+" "+a4+" "+a5+" "+a6+" "+a7;
115        }
116        writer.print(PrimarMatrix [temp1][temp2]+" ");
117    }
118    }
119    writer.println("");
120    System.out.println();
121    }
122    writer.close();
123
124 }
125
126 }

```

Slika 24 - Java kod III deo

4.2.4 Četvrta faza

Četvrta faza predstavlja pronalaženje zajedničkih elemenata – Kriptografski ključ

Ukoliko se prisetimo slike iz dela o Satelitskom scenariju (Slika 2 - Presek Skupova) i uporedimo je sa slikom (Slika 19 - ADS-B antene (presek signala)) primetićemo sličnost između slika. Samim tim rešenje u pronalaženju zajedničke informacije na osnovu prikupljenih podataka nameće se samo.

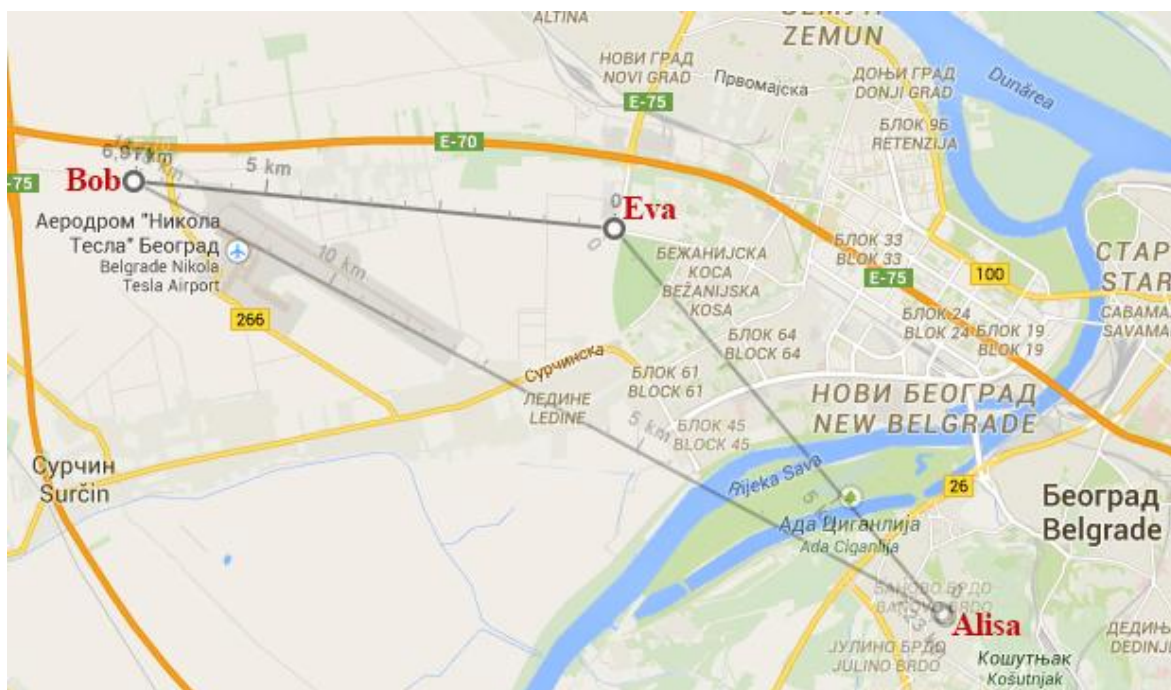
Sve što je potrebno za primenu Satelitskog scenarija naši nizovi imaju. Nastaju od istog izvora, binarni su, sadrže visok nivo slučajnosti, dok potrebnu grešku iz definicije Satelitskog scenarija u našem slučaju zamenjuje različita pokrivenost antena ali i uneta lokalna slučajnost.

U našem rešenju koristićemo samo prvu fazu Satelitskog scenarija, odnosno fazu destilaciju prednosti. Po definiciji sama faza destilacije sadrži nekoliko rundi ponavljanja u zavisnosti od pretpostavke greške, a na osnovu naših već izvedenih testova u ovom rešenju potrebno je izvršiti šest rundi ponavljanja. Nakon rundi ponavljanja legitimni akteri komunikacije dobijaju zajednički binarni niz – kriptološki ključ.



4.3 EKSPERIMENTALNO OKRUŽENJE

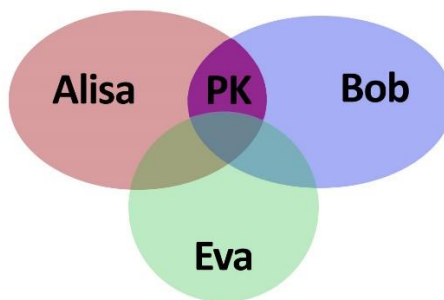
Nakon teorijskih analiza i definicija vreme je da se realizacija rešenja sprovede u realnim uslovima. U svim testiranjima bila su tri aktera komunikacije, dva legitimna (Alisa i Bob) i jedan zlonamerni akter (Eva), koja su se nalazila na tri različite geografske lokacije. Navešćemo rezultate dva primera testiranja rešenja. Odabrane lokacije imaju drastično različitu gustinu saobraćaja. Na slici (Slika 25) i slici (Slika 26) su date lokacije i udaljenost između aktera. Kao što se vidi na slikama, i razdaljina između aktera je različita. U oba testiranja na sve tri lokacije instalirana je ADS-B oprema za praćenje avio saobraćaja, a vreme praćenja je vreme prosečnog intenziteta saobraćaja te geografske zone. Osim opreme za praćenje na svim lokacijama se nalaze i dekoderi radio talasa, kao i mikroracunar za binarizaciju podataka i kreiranje usaglašenih binarnih nizova.



Slika 25 - Pozicije aktera na prvom testiranju



Slika 26 - Pozicije aktera na drugom testiranju



Slika 27 - Presek skupova

Snimanje avio saobraćaja počinje u različitom vremenskom trenutku i traje tri približno ista vremenska intervala gde je potrebno da dođe do preklapanja intervala bar 5 do 10 minuta, čime se obezbeđuje da su sva tri aktera u vreme preklapanja prikupila iste koordinate aviona u preklapljenim površinama. Na slici (Slika 27) deo preklapljene površine između Alise i Boba (obeležen sa PK) predstavlja zajedničku informaciju Alise i Boba, odnosno predstavlja koordinate aviona koje Eva ne može da prihvati.

Nakon što je mikoračunar generisao binarni niz počinje prva runda postupaka Destilacije prednosti Satelitskog scenarija. Alisa i Bob dele niz na bigrame i rade *parity check*, nakon toga Alisa šalje Bobu putem javnog kanala (Internet) rezultate, dobijaju novi niz i nad njim vrše drugu rundu, zatim treću i sve tako do šeste kada se očekuje da više neće biti razlike između nizova Alise i Boba. Pretpostavku da je potrebno i dovoljno šest rundi destilacije prednosti može se proveriti *hash* funkcijama nad binarnim nizom nakon svake runde. Ovakav binarni niz, zbog karakteristika koje nosi možemo nazvati kriptološkim ključem.

Iz razloga testiranja isti postupak obaviće se i između Alise i Eve, kao Eve i Boba. Postoji mogućnost da i u ovakvim kombinacijama aktera se dobije zajednički binarni niz ali ukoliko se nizovi uporede binarni nizovi međusobno dobija se entropija od 0,5 što nam daje za pravo da zaključimo da su dobijeni nizovi totalno različiti i da Eva kao zlonamerni akter će ako prisluškuje Alisu dok šalje *parity check* bite Bobu dobiti svoj ključ koji će biti totalno različit od ključa koji poseduju Alisa i Bob.

U tabeli je dat uporedni prikaz rezultata dobijenih u ova dva primera realizacije. Napominje se da su u prvom primeru udaljenosti između aktera manje, ali je manji i intenzitet avio saobraćaja, što nam govori da brzina ključa direktno zavisi od intenziteta avio saobraćaja na geografskom području na kome se nalaze akteri. Na broj zajedničkih bitova umnogome utiče i vreme početka i kraja prikupljanja. Pa tako Eva kao zlonamerni akter mora da duže prikuplja podatke kako bi bila sigurna da je prikupila podatke aviona koje posmatraju i Alisa i Bob i na taj način smanjuje zajedničku informaciju koju ima sa Alisom i Bobom. Osim ovog faktora, kvalitet opreme koju akteri koriste takođe utiče na broj zajedničkih bitova na kraju.

Broj zajedničkih bitova između aktera				
	Područje sa slike 24		Područje sa slike 25	
	Bob	Eva	Bob	Eva
Alisa	348	319	1452	1256
Bob	////////////////	129	////////////////	985

Tabela 3 - Broj zajedničkih bitova



4.4 EVALUACIJA PERFORMANSI

Nakon dobijanja rezultata prikazanih u tabeli (Tabela 3) potrebno je podvrgnuti rezultate analizi kako bi bili sigurni da kriptološki ključevi pre svega uopšte sadrže slučajnost, kolika je ta slučajnost i kao najbitnije da li su upotrebljivi u kriptološkim sistemima.

4.4.1 Statistički testovi

Broj različitih testova koji se mogu primeniti na neku binarnu sekvencu je veliki, te je i pouzdanost tih testova različita. Definisanjem tačno određenih testova kao i očekivanim rezultatima *NIST* je omogućio da polazne pretpostavke lako dokažu. Jedan od uslova kod većine *NIST* testova zahtevaju binarne sekvence sa bar 10000 bitova. Ako se ponovo vratimo na tabelu (Tabela 3) vidimo da tokom jednog testiranja broj prikupljenih bitova je drastično manji od potrebnog. Iz tog razloga, prikupljanje je ponovljeno više puta kako bi se dobio veći broj zajedničkih bitova. Tako su statistički testovi izvršeni nad 10000 zajedničkih bitova legitimnih učesnika komunikacije (Alisa i Bob). Kako bi se još bolje testirali nizovi dobijeni uz pomoć ovog informacionog izvora, izvršeno je i poređenje rezultata *NIST* testova nad ovim nizom kao i nad nizom dobijenim korišćenjem nuklearnog raspada atoma kao izvora slučajnosti (HotBits).

4.4.1.1 Ispitivanje učestanosti u nizu

Cilj ovog testa je ispitivanje odnosa jedinica i nula u nizu bitova. Potreban je približan broj jedinica i nula u sekvenci.

	HotBits	ADS-B
P	0.808976294576234	0.639049962629422

Tabela 4 - Ispitivanje učestanosti u nizu

Nakon izvršenog testa možemo zaključiti da su oba niza zadovoljila uslov postavljen od strane *NIST*-a:

$$P \geq 0.01$$

Formula 5

i da se mogu smatrati slučajnim. Ako poredimo rezultate, možemo zaključiti da je odnos nula i jedinica u oba niza bio približno isti, ali da je bolji rezultat pokazao *HotBits* niz.

4.4.1.2 Serijski test

Test se koristi za utvrđivanje učestanosti svih mogućih preklapanja n -bitnog niza u čitavoj sekvenci.



	HotBits	ADS-B
Trigrami		
000	1215	1553
001	1250	1204
010	1280	1361
011	1195	1099
100	1251	1207
101	1225	1099
110	1195	1256
111	1245	1221
P ₁	0.3174364812591314	0.311953164865971
P ₂	0.1088774362443912	0.17069305982499872

Tabela 5 - Serijski test

Rezultati dobijeni u ovom testu su ispunili *NIST*-ov zahtev definisan u formuli ($P \geq 0.01$ Formula 5). Ukoliko se uporede rezultati može se zaključiti da su približno isti.

4.4.1.3 Ispitivanje uzastopnih ponavljanja

Posmatrana karakteristika u ovom testu je ukupan broj uzastopnih ponavljanja jedinice ili nule u nizu.

	HotBits	ADS-B
P- Value	0.6426910024891495	0.8046503202403916

Tabela 6 - Ispitivanje uzastopnih ponavljanja

Nakon sprovedenog testa rezultati su zadovoljili zahtev definisan u formuli ($P \geq 0.01$ Formula 5). U poređenju ova dva niza nakon testa možemo zaključiti da nastao kao rezultat predloženog rešenja ima više promena iz 0 u 1 ili obrnuto.

4.4.1.4 Entropija

U ovom testu posmatra se učestalost pojave svih mogućih preklapajućih n -bitnih uzoraka u nizu.

	HotBits	ADS-B
Monobit	0.9999957227359038	0.9996264224455302
Bigram	0.9999878164470851	0.999675613429314
Trigram	0.9998791930081991	0.9994603538795782
4x4 Matrices	0.9998558732137256	0.9999749797512283

Tabela 7 - Entropija

Vrednosti i ovog testa zadovoljavaju zahteve *NIST*-a definisane u formuli



($P \geq 0.01$)

Formula 5). Možemo tvrditi da je niz dobijen iz pomoć podataka civilnog avio saobraćaja slučajan i da je u poređenju sa nizom dobijenim nuklearnim raspadom atoma i nakon ovog testa pokazao približno iste rezultate, shodno tome, možemo tvrditi da poseduje visok nivo slučajnosti i da može biti korišćen u kriptografiji za generisanje kvalitetnih kriptoloških ključeva.

4.4.2 Kvalitet ključa

Nakon sprovedenih *NIST* testova na osnovu rezultata možemo zaključiti da su binarne sekvence slučajne. Ipak za kvalitetne kriptološke ključeve osim slučajnih nizova potrebno je da same sekvence sadrže veliku kompleksnost. Primenom proračuna Kolmogorove kompleksnosti nad sekvencom dobijen je veliki broj pod nizova što nam govori da je kompleksnost tih sekvenci velika. U poređenju sa *HotBits* binarnom sekvencom, koja nije distribuirana već je samo generisana u jednoj tački komunikacije, rezultati testova nad zajedničkoj sekvenci bitova pokazuju da je naša polazna pretpostavka ispunjena i da je na ovaj način moguće generisati kvalitetne kriptološke ključeve.

Kvalitet ovakvog ključa potvrđen *NIST* testovima je nesporan. Mogućnost zlonamernog uticanja na izvorne podatke i korekcije samih podataka je takođe smanjeno jer se koriste podaci dobijeni direktno od aviona, a razdaljina prikupljanja je dosta velika te uticanje na sistem ometanjem radio talasa u realnim uslovima nije očekivan.

4.4.3 Brzina ključa

Binarna sekvenca generisana uz pomoć *SLD TRNG* daje neverovatnih 2 *Gbit*-a u samo jednoj sekundi što za nekih 10 minuta predstavlja niz od oko 1 200 000 000 000 bitova (1,2 *Tbit*). Međutim ovakva sekvenca se nalazi na samo jednoj strani, nju je potrebno distribuirati. Primenom našeg sistema za generisanje 80 000 bitova potrebno do jedan sekund, međutim za kvalitetnih 80 000 bitova koje poseduju korelisanost sa bitovima generisanim na drugoj strani potrebno je oko 10 minuta. Nakon primene protokola ostaje nam oko 1000 zajedničkih bitova te tako možemo reci od početka snimanja do dobijanja kvalitetnog ključa je potrebno do 10 minuta. Međutim treba napomenuti, da brzina ključa direktno zavisi od gustine saobraćaja, a delom i od udaljenosti između aktera komunikacije. Takođe treba napomenuti da snimanjem znatno dužim od prosečnih 10 minuta ne dobijamo veću dužinu ključa, jer se korelisanost između matrica smanjuje zbog kretanja aviona. To možemo istaći i kao još jedan vid zaštite od zlonamernog aktera.

Brzina dogovora za oko 1000 zajedničkih bitova direktno je povezana sa brzinom Maurerovaog Satelitskog scenarija. Ukoliko snimanje saobraćaja konstanto traje, konstantnim kreiranjem novih matrica, onda brzina ključa predstavlja samo vreme potrebno za usaglašavanje i pronalaženje zajedničkih bitova, što je mnogo manje od ranije potrebnih 10 minuta. Na taj način brzina ključa se uvećava i ona sada zavisi samo od brzine primene faze Destilacije Satelitskog scenarija, što nije duže od sekund, dva.

Kako je 1000 bitova malo za neku komunikaciju šifrovanu *One-Time Pad* algoritmom, jasno je da prikupljanje podataka i razmenu treba obaviti nekoliko puta u intervalima od po 10 minuta i na taj način obezbediti bazu dovoljnu da se započne komunikacija. Ukoliko bi se koristio *AES*, jedno prikupljanje bi bilo sasvim dovoljno.



4.5 OBLAST PRIMENE

Iz razloga da su binarne sekvence dobijene pomoću ovakvog izvora slučajnosti dokazano slučajne i da poseduju skoro maksimalnu kompleksnost mogu se osim u kriptografiji koristiti i u drugim oblastima. Veoma jednostavna realizacija može obezbediti sigurnost i zameniti komplikovan i skupe fizičke komponente.

Dobijanjem kriptoloških ključeva uz pomoć ovog sistema obezbeđujemo realizaciju „kurirske službe“ javnim kanalom i što je veoma bitno napomenuti bez unapred razmenjene bilo kakve „tajne“. Ovako dobijeni ključ kao što je već pokazano, ispunjava sve uslove da bude ključ perfektnih šifarskih sistema. Korišćenjem ključa za *OTP* algoritam obezbeđujemo teorijski dokazanu sigurnost, te možemo reći da realni kanal se može približiti idealnom primenom ovog sistema. Osim kod *OTP*, ovakav sistem generisanja kriptografskih ključeva omogućava nam korišćenje ovih ključeva kod svih simetričnih algoritama. Pa tako, ukoliko bi primenili *SHA-512 hash* funkciju nad zajedničkim bitovima dobili bi ove ključeve koje možemo koristiti i za *AES-512* [20] i na taj način možemo obezbediti veliku sigurnost nad šifrovanim podacima koji sada ne moraju biti samo tekstualni kao kod *OTP* jer je dužina ključa ne zavisi od bitske veličine podatka koji treba preneti.

Ukoliko bismo komunikaciju obezbedili *OTP* šifarskim sistemom, ta komunikacija bi morala biti samo tekstualna te bi onda tu komunikaciju mogli obavljati na računaru putem *chat* aplikacije u kojoj bismo integrisali sam sistem razmene prikupljenih bitova radi kreiranja ključa i šifrovanja teksta tim ključem. Za prikupljanje radio signala emitovanog od aviona morao bi da se koriste veoma pristupačne fizičke komponente prikazane na slici (Slika 18 - ADS-B prijemnici). Ako bi komunikaciju obavljali putem mobilne telefonije, razmena tekstualnih poruka mogla bi da bude realizovana pomoću *Short Message Service*-a (SMS), dok bi prikupljanje radio talasa mogao da obavlja i sam mobilni telefon uz određene modifikacije.

Korišćenjem *AES-512* za šifrovanje komunikacija više ne bi morala da bude samo tekstualna, već audio, audio-video ili samo prenos podataka. Za komunikaciju uz pomoć računara trebalo bi obezbediti aplikaciju koja bi preuzimala podatke sa fizičkih komponenti za prikupljanje radio signala, zatim javnim kanalom (npr. putem Interneta) sa aplikacijom na drugom udaljenom računaru razmenila bitove i generisala jedinstveni ključ, koji bi zatim bio upotrebljen za šifrovanje. Prilikom narednog šifrovanja ceo proces bi se ponovio i na taj način bi rešili nedostatak *AES* algoritma da uvek koristi isti ključ zbog nemogućnosti da se ključ svaki put ponovo distribuira. Ovakav scenario bi mogao da se realizuje i između mobilnih uređaja, ali i između mobilnih uređaja i računara.

5. Zaključak

Naš osnovni zadatak u ovom radu bio je pronalazak ekvivalentne šeme za praktičnu implementaciju protokola „Satelitski scenario“, koji je do sada predstavljen samo na teorijskoj osnovi od autora Ueli Maurera. Ujedno, to je i predstavljalo glavnu hipotezu ovog rada, koja je teorijski i eksperimentalno potvrđena.

Uspešno je dizajniran sopstveni izvor slučajnosti, na osnovu binarnih nizova dobijeni nakon kodovanja koordinata prikupljenih pomoću *ADS-B* sistema, potvrdili smo da podaci promenljivi u vremenu, poseduju visok nivo neodređenosti, dovoljno visok da se takvi binarni nizovi mogu koristiti u kriptološke svrhe. Pored izvora slučajnosti koji dolazi od koordinata aviona, uvedeni su lokalni izvori slučajnosti čija je uloga da dodatno uvedu šumnost u kanal. Na ovaj način prevaziđen je uslov Satelitskog scenarija u kome napadač ne sme da ima direktan pristup satelitu, odnosno da između satelita i napadače bude idealan kanal bez gubitaka. U slučaju da svi akteri poseduju istu opremu za prikupljanje podataka, njihove geografske pozicije uticaće na prikupljene podatke a ako se sva tri aktera nalaze na istoj geolokaciji, lokalna slučajnost nam obezbeđuje potrebnu razliku u matricama, odnosno i u ovom slučaju lokalnu slučajnost smatramo obaveznim šumom u prenosu kod Satelitskog scenarija. Kada govorimo o brzini ključa, treba napomenuti da vrednosti koje su teorijski dobijene u protokolu „Satelitski scenario“ nekoliko puta su manje od vrednosti koje se dobijaju u našem protokolu. Od inicijalnih korelisanih nizova dužine 40000 bitova u protokolu „Satelitski scenario“ na teorijskoj osnovi, na kraju protokola dva učesnika imaju približno 100 zajedničkih bitova, dok kod našeg protokola sa inicijalnom matricom od 80000 bitova, na kraju protokola ostaje 1000 zajedničkih bitova, koji predstavljaju materijal za simetrične kriptološke ključeve.

5.1 OSTVARENI DOPRINOSI

Navedeni postignuti rezultati su ujedno i pojedinačni doprinosi ovog rada:

- Iskorišćene su koordinate aviona, međutim za ovakvu upotrebu moguće je koristiti i neke druge promenljive (npr. praćenje brodova, rezultate sportske prognoze, promene vrednosti akcija na berzama, broj rođenih i umrlih...) Svakako pri izboru treba voditi računa da treća strana (zlonamerna) ne može posredno ili neposredno uticati na te podatke ili njihovu promenu. Iz tog razloga je su izabrane koordinate aviona koje su uz pomoć *ADS-B* tehnologije prikupljane direktno od aviona.
- Dizajniran je sopstveni izvor slučajnosti na osnovu avio saobraćaja. Kvalitet dobijenih binarnih nizova potvrđen je preko *NIST*-ove baterije testova i Kolmogorove kompleksnosti. Na ovaj način pokazano je da nizovi imaju visoke entropijske vrednosti koje odgovaraju kriptološkim zahtevima.
- Pronađena je ekvivalentna šema protokola „Satelitski scenario“ preko *ADS-B* sistema, koji omogućava postojanje korelisanih binarnih nizova između dve strane koje razmenjuju ključeve javnim kanalima.
- Na kraju, uspešno je implementiran i testiran protokol za razmenu kriptoloških ključeva preko javnih kanala. Ključevi mogu da se koriste za šifrovanje sa simetričnim šiframa (AES, 3DES). Takođe, postoji mogućnost i za upotrebu perfektnih šifarskih sistema – *OneTime pad*, međutim, tada je neophodno napraviti



kompromis između dužine šifrovanih poruka i brzine ključa, koju je moguće postići sa ovim protokolom.

5.2 PREDLOG ZA BUDUĆI RAD

Za budući rad na ovu temu, predviđeno je poboljšanje performansi protokola, koje se ogledaju u brzini ključa, ali bez narušavanja njegovih bezbednosnih osobina. Takođe, pored primene dizajna protokola „Satelitski scenario“ u našem praktičnom okruženju, predviđena je realizacija nove šeme protokola u postojećem praktičnom okruženju, a na osnovu šeme protokola koji potiče iz sve više danas popularne oblasti koja se odnosi na bezbednost na fizičkom sloju, a koja pretpostavlja upotrebu *LDPC* klase kodova za ispravljanje grešaka. Ovaj protokol bi imao neuporedivo bolje performanse sa svih funkcionalnih aspekata i sa aspekta praktične primenljivost, jer bi primena bila u bežičnim računarskim mrežama koje predstavljaju radno okruženje *LDPC* kodova.



Literatura

- [1] C. H. Bennett, F. Bessette, G. Brassard, L. Salvail i J. Smolin, „Experimental quantum cryptography,“ *Journal of cryptology*, t. 5, pp. 3-28, 1992.
- [2] U. M. Maurer, „Secret Key Agreement by Public Discussion,“ *IEEE TRANSACTIONS ON INFORMATION THEORY*, t. 39, 1993.
- [3] S. M. Bellare, Frank Miller: Inventor of the One-Time Pad, Taylor & Francis Group, 2011.
- [4] M. Veinović i S. Adamović, Kriptologija 1, Beograd: Univerzitet Singidunum, 2013.
- [5] M. Milan i A. Saša, Osnovi teorije informacije i kodovanja, Beograd: Univerzitet Singidunum, 2013.
- [6] W. SCHINDLER, „Random Number generators for cryptographic applications,“ u *Cryptographic Engineering*, Springer Verlag, 2009, pp. 5-23.
- [7] B. Sunar, „True random number generators for cryptography,“ u *Cryptographic Engineering*, Springer-Verlag, 2009, pp. 55-73.
- [8] „HotBits,“ [Na mreži]. Available: <https://www.fourmilab.ch/hotbits/>.
- [9] Sylvania, „Sylvania 6D4“.
- [10] M. Huang, A. Wang, P. Li, H. Xu i Y. Wang, „Real-time 3 Gbit/s true random bit generator based,“ *Optics Communications*, t. 325, pp. 165-169, 2014.
- [11] D. Knuth, Art of Computer Programming - Seminumerical Algorithm. Vol 2, 1997.
- [12] A. Rukhin, J. Soto i J. Nechvatal, „A Statistical Test Suite for the Validation of Random Number Generators and Pseudo Random Number Generators for Cryptographic Applications,“ 2010.
- [13] C. Floudas i P. Pardalos, Encyclopedia of Optimization, Springer Science & Business Media, 2001.
- [14] S. Šegan i N. Pejović, Osnovi astronomije, Beograd: Vesta Co., 2006.
- [15] NASA. [Na mreži]. Available: <http://nssdc.gsfc.nasa.gov/nmc/spacecraftDisplay.do?id=1957-001B>.
- [16] NASA. [Na mreži]. Available: <http://nssdc.gsfc.nasa.gov/nmc/spacecraftDisplay.do?id=1957-001B>.
- [17] M. Russell Rip i J. M. Hasik, The Precision Revolution: GPS and the Future of Aerial Warfare, US Naval Institute Press, 2002.
- [18] U. N. Observatory, „CURRENT GPS CONSTELLATION,“ [Na mreži]. Available: <http://tycho.usno.navy.mil/gpscurr.html>. [Poslednji pristup 15 10 2013].
- [19] F. Kunzi i J. Hansman, „ADS-B Benefits to general aviation and barriers to implementation,“ MIT, 2011.
- [20] A. Moh'd, Y. Jararweh i L. Tawalbeh, „AES-512: 512-Bit Advanced Encryption Standard,“ u *7th International Conference on Information Assurance and Security*, 2011.

Spisak slika

Slika 1 - Distribucija bita kvantnim kanalom	6
Slika 2 - Presek Skupova	7
Slika 3 - Egipatski simboli (levo) i Skital (desno).....	9
Slika 4 - Šifrovanje i dešifrovanje poruke	10
Slika 5 - Entropija slučajne varijable sa dva ishoda [5]	13
Slika 6 - Shema klasifikacije RNG ([6])	14
Slika 7 - Intervali [8]	15
Slika 8 - Realizacija TRNG-a pomoću diode 6D4 [9]	16
Slika 9 - Baggini and Bucci projekat [7]	16
Slika 10 - The Intel TRNG [7]	17
Slika 11 - SLD TRNG [10].....	17
Slika 12 - Ekvator i Grinidž.....	30
Slika 13 - Komponenta u vasioni - Satelit.....	32
Slika 14 - Kontrolne komponente na zemlji.....	32
Slika 15 - GPS prijemnici.....	33
Slika 16 - ADS-B sistem	35
Slika 17 - Kompletna šema predloženog rešenja	37
Slika 18 - ADS-B prijemnici	39
Slika 19 - ADS-B antene (presek signala).....	40
Slika 20 - Radio talas i kodne poruke.....	40
Slika 21 - Fazor	41
Slika 22 - Java kod I deo	43
Slika 23 - Java kod II deo	44
Slika 24 - Java kod III deo.....	45
Slika 25 - Pozicije aktera na prvom testiranju	46
Slika 26 - Pozicije aktera na drugom testiranju	47
Slika 27 - Presek skupova.....	48

Spisak Tabela

Tabela 1 – XOR vrednosti	11
Tabela 2 - Dekodovane III poruke dva leta	41
Tabela 3 - Broj zajedničkih bitova	48
Tabela 4 - Ispitivanje učestanosti u nizu	49
Tabela 5 - Serijski test	50
Tabela 6 - Ispitivanje uzastopnih ponavljanja	50
Tabela 7 - Entropija	50

