Sigurne sheme enkripcija s tajnim ključem

Repić, Mirna

Sveučilište Josipa Jurja Strossmayera u Osijeku
Odjel za matematiku
Sveučilišni diplomski studij matematike, smjer Matematika i računarstvo

Mirna Repić

Sigurne sheme enkripcija s tajnim ključem

Diplomski rad

Osijek, 2022.

Sveučilište Josipa Jurja Strossmayera u Osijeku
Odjel za matematiku
Sveučilišni diplomski studij matematike, smjer Matematika i računarstvo

Mirna Repić

Sigurne sheme enkripcija s tajnim ključem

Diplomski rad

Mentor: prof. dr. sc. Ivan Matić

Osijek, 2022.

Sadržaj

Uvod

Dolaskom digitalnog doba i kriptografija ulazi u novo razdoblje, razdoblje moderne kriptografije. Ipak, glavna ideja ostaje ista - pronaći algoritme koji će sigurno, ali i efikasno, omogućiti dvjema stranama komunikaciju koja je razumljiva samo njima. Jedan od najvažnijih pojmova moderne kriptografije je pojam računalne sigurnosti. U ovom radu proučavat ćemo (računalno) sigurne sheme enkripcija s tajnim ključem.

U prvom poglavlju definirat ćemo kriptografiju i upoznati se s osnovnim pojmovima klasične kriptografije. Zatim ćemo u drugom poglavlju raspravljati o računalnoj sigurnosti koristeći dva principa: konkretni i asimptotski. U trećem poglavlju promatrat ćemo osnovni pojam sigurnosti za sheme enkripcije s tajnim ključem. Zanimat će nas sigurnost od napada na šifrirani tekst gdje protivnik promatra samo jedan šifrirani tekst, odnosno, sigurnost kada se zadani ključ koristi za enkripciju samo jedne poruke. Također, ponudit ćemo definiciju računalno sigurne enkripcije uvodeći pojam nerazlučivosti. U četvrtom poglavlju konstruirat ćemo sigurne sheme enkripcije uvodeći pojmove pseudoslučajnih generatora i protočnih šifri. Za kraj, predstavit ćemo jače sigurnosne ideje poput sigurnosti za višestruke enkripcije, napada na odabrani otvoreni tekst te CPA- sigurnosti.

1 Osnovni kriptografski pojmovi

Kako bismo mogli dublje zaroniti u temu ovog rada, najprije moramo definirati neke osnovne pojmove s kojima ćemo se u radu susretati.

Kriptografija je znanstvena disciplina koja se bavi proučavanjem metoda za slanje poruka u formi koja je razumljiva samo onima kojima je informacija namijenjena, dok će svima ostalima ta poruka biti neupotrebljiva. U daljnjem razmatranju, dvije strane koje komuniciraju putem komunikacijskog kanala nazivamo *pošiljatelj* i *primatelj*. Često ćemo ih nazivati i *poštene strane*. Poruku koju pošiljatelj želi poslati primatelju zovemo *otvoreni tekst*. Pošiljatelj transformira otvoreni tekst koristeći unaprijed dogovoreni *ključ* postupkom koji zovemo *šifriranje* ili *enkripcija* te tako dobiva šifriranu poruku (odnosno, *šifrat*) koju šalje putem komunikacijskog kanala. Primatelj zna ključ te pomoću njega može odrediti otvoreni tekst. Taj postupak vraćanja poruke iz njenog enkriptiranog oblika u originalni oblik nazivamo *dešifriranje* ili *dekripcija*. Osobu koja prisluškuje komunikacijski kanal nazivamo *protivnik*, *prisluškivač* ili *napadač*. *Napadi* su zlonamjerne akcije koje izvode neautorizirani članovi komunikacijskog modela. Uspješan napad na kriptografski sustav podrazumijeva pronalaženje praktičnog načina da protivnik od šifriranog teksta dobije otvoreni tekst. Neformalno, to ćemo često nazivati i razbijanjem šifre. *Kriptografski algoritam* ili *šifra* funkcija je koja se koristi za enkripciju i dekripciju. Ovdje se ustvari radi o dvjema funkcijama koje preslikavaju osnovne elemente otvorenog teksta (najčešće su to slova, grupe slova ili bitova) u osnovne elemente šifrata, i obratno. Navedene se funkcije biraju iz familije funkcija, ovisno o korištenom ključu. Skup svih mogućih vrijednosti ključeva nazivamo *prostor ključeva*. *Kriptosustav* se sastoji od kriptografskog algoritma, te svih mogućih otvorenih tekstova, šifrata i ključeva.

Ako otvoreni tekst označimo s m, a šifrat s c, tada funkcija enkripcije Enc djeluje na m i vraća c, odnosno

$$\mathsf{Enc}(m) = c.$$

Funkcija dekripcije Dec djeluje na c i vraća m, odnosno

$$\mathsf{Dec}(c) = m.$$

Kako je cijeli smisao enkripcije, a zatim dekripcije poruke ponovno dobiti otvoreni tekst, mora biti zadovoljen sljedeći izraz:

$$\mathsf{Dec}(\mathsf{Enc}(m)) = m.$$

Sve suvremenije metode enkripcije temelje se na uporabi ključa. Ovisno o načinu korištenja ključa, odnosno obzirom na njegovu tajnost, razvijene su dvije klase algori-

tama kriptiranja, a time i dvije vrste kriptosustava: simetrični i asimetrični. *Simetrični kriptosustavi* (ili *kriptosustavi s tajnim ključem*) koriste isti ključ i za enkripciju i za dekripciju, pa sigurnost ovih kriptosustava leži u tajnosti ključa. *Asimetrični kriptosustavi* ili *kriptosustavi s javnim ključem* koriste različite ključeve prilikom enkripcije i dekripcije, a ključ za enkripciju je javan i bilo tko može šifrirati poruku pomoću njega, no samo osoba koja ima odgovarajući dekripcijski ključ može doći do otvorenog teksta iz šifrata.

U modernoj kriptografiji, kodove koji omogućuju dvjema stranama tajnu komunikaciju u prisutnosti prisluškivača koji može nadzirati svu komunikaciju između njih nazivamo *sheme enkripcije*. Mi ćemo u nastavku promatrati simetrične sheme enkripcije, odnosno sheme enkripcije s tajnim ključem.

2 Računalna sigurnost

Jedan od temeljnih pojmova moderne kriptografije je pojam računalne sigurnosti. To je relaksirani pojam savršene tajnosti informacijsko-teorijske prirode kojoj je temeljni zahtjev da nikakva informacija o enkriptiranoj poruci ne procuri, čak ni prisluškivaču s neograničenom računskom snagom. U kontekstu računalne sigurnosti i praktičnim svrhama, shema enkripcije bi se i dalje smatrala sigurnom ako bi prisluškivačima s ograničenom računskom moći propuštala i malu količinu informacija. Primjerice, shema koja propušta informacije s vjerojatnošću od najviše 2^{-60} do prisluškivača koji ulaže do 200 godina računskog napora na najbržem dostupnom superračunalu prikladna je za svaku primjenu u stvarnom svijetu.

> *Sigurnosne definicije koje uzimaju u obzir računalna ograničenja protivnika i pri tom dopuštaju malu vjerojatnost neuspjeha, nazivaju se računalnim.*

Računalna sigurnost uključuje dvije relaksacije u odnosu na informacijsko-teorijske pojmove sigurnosti, a koje su neophodne u slučaju enkripcije kako bi se nadišla ograničenja savršene tajnosti.

1. *Sigurnost je zajamčena samo protiv učinkovitih protivnika koji rade određeno vrijeme.*

2. *Protivnici mogu potencijalno uspjeti, odnosno sigurnost može potencijalno biti narušena, s vrlo malom vjerojatnošću.*

Prva relaksacija govori da s obzirom na dovoljno vremena ili dovoljno računalnih resursa protivnik može narušiti sigurnost. No, ukoliko možemo učiniti resurse potrebne za razbijanje sheme većim od onih dostupnih bilo kojem realnom protivniku, tada je shema neprobojna za sve praktične svrhe. Druga relaksacija kaže da ako vjerojatnost potencijalnog uspjeha protivnika možemo učiniti dovoljno malom, nemamo razloga za brigu.
Da bismo dobili smislenu teoriju koju ćemo moći koristiti, moramo precizno definirati gornje relaksacije. To možemo učiniti koristeći dva opća pristupa - konkretni i asimptotski pristup.

2.1 Konkretni pristup

Konkretnim pristupom definicija sigurnosti ima sljedeći oblik

> *Kažemo da je shema (t, ϵ)-sigurna ako bilo koji protivnik u najviše t vremena rada uspije razbiti shemu s vjerojatnošću najviše ϵ.*

Konkretan pristup računskoj sigurnosti kvantificira sigurnost kriptografske sheme eksplicitno ograničavajući maksimalnu vjerojatnost uspjeha bilo kojeg randomiziranog protivnika koji radi određeno vrijeme, odnosno ulaže određenu količinu računskog napora. Općenito se pretpostavlja da moderne sheme enkripcije s tajnim ključem daju gotovo optimalnu sigurnost. Promotrimo sljedeći primjer kako bismo objasnili što pod tim podrazumijevamo.

Primjer 2.1. *([2]) Pretpostavimo da govorimo o pretraživanju prostora ključeva grubom silom i bez prethodne obrade. Ako pretpostavimo da ključ ima duljinu n, prostor ključa ima veličinu 2^n. Protivnik koji radi t vremena uspješno bi razbio shemu s vjerojatnošću najviše $ct/2^n$, gdje je c neka fiksna konstanta. Kada bismo radi jednostavnosti uzeli da je $c = 1$, ključ duljine $n = 60$ pružio bi odgovarajuću sigurnost protiv protivnika koji koristi stolno računalo. Naime, procesor od $4GHz$ kakav je karakterističan za stolna računala izvodi $4 \cdot 10^9$ ciklusa u sekundi. Isti procesor za 2^{60} ciklusa zahtijeva $2^{60}/(4 \cdot 10^9)$ sekundi, što je oko 9 godina. Ipak, najbrže superračunalo može izvesti otprilike $2 \cdot 10^{16}$ operacija s pomičnim zarezom u sekundi, pa 2^{60} takvih operacija zahtijeva tek minutu na takvom računalu. Kada bismo duljinu ključa povećali i za n uzeli $n = 80$, vrijeme bi se drastično povećalo pa bi čak i superračunalu trebalo oko 2 godine da izvrši 2^{80} operacija. Ipak, ovdje smo radi razumijevanja pojednostavili situaciju tako da smo uzeli $c = 1$, a u praksi će c biti veći od 1. Također, izostavili smo uzeti u obzir vrijeme potrebno za pristup memoriji i mogućnost paralelnog računanja na mreži računala što znatno utječe na izvedbu grubom silom. Ako je vjerojatnost da protivnik uspješno povrati enkriptiranu poruku u jednoj godini najviše 2^{60}, onda je mnogo vjerojatnije da će i pošiljatelja i primatelja pogoditi grom u tom istom vremenskom razdoblju. Događaj koji se događa jednom u stotinu godina može se grubo procijeniti da se dogodi s vjerojatnošću 2^{30} u bilo kojoj sekundi. Nešto što se dogodi s vjerojatnošću 2^{60} u bilo kojoj sekundi je 2^{30} puta manje vjerojatno i moglo bi se očekivati da će se dogoditi otprilike jednom svakih 100 milijardi godina.*

Važno je naglasiti da se ovdje susrećemo s određenim tehničkim i teorijskim poteškoćama. Jasno je da je konkretan pristup od velike važnosti za korisnika kriptografske sheme budući da su konkretna jamstva ono što ga u konačnici i zanima. Ipak, vrlo je teško dati precizna konkretna jamstva, kao i tumačiti sigurnosne tvrdnje. Iz jamstva da niti jedan protivnik koji radi x godina ne može razbiti zadanu shemu s vjerojatnošću boljom od ϵ ne saznajemo ništa o tom koju vrstu računalne snage pretpostavljamo, uzima li se u obzir napredak računalne snage u budućnosti, pretpostavlja li procjena korištenje gotovih algoritama ili namjenskih softvera optimiziranih za napad, kao ni niz drugih čimbenika. Još jedan veliki nedostatak konkretnog pristupa je da nam takvo

jamstvo vrlo malo govori o vjerojatnosti uspjeha za period kraći od x, odnosno govori nam samo da može biti najviše ϵ, dok istovremeno ne govori ništa za period duži od x godina.

2.2 Asimptotski pristup

Asimptotski pristup sigurnosti može pomoći u zaobilaženju problema koji prate konkretni pristup. Ipak, njega možemo koristiti samo u slučaju kada konkretna sigurnost nije neposredna briga. Suština asimptotskog pristupa jest da uvođenjem sigurnosnog parametra cjelobrojne vrijednosti, kojeg ćemo označavati s n, parametriziramo obje kriptografske sheme, kao i sve uključene strane - poštene strane i protivnika. Prilikom inicijalizacije sheme, poštene strane generiraju ključeve tako da odabiru neku vrijednost n za sigurnosni parametar kojeg, radi praktičnosti, možemo smatrati duljinom ključa. Pretpostavlja se da je vrijednost tog parametra poznata i napadaču. Sada ćemo vrijeme rada protivnika i njegovu vjerojatnost uspjeha promatrati kao funkcije sigurnosnog parametra n. U ovom pristupu izjednačavamo učinkovite protivnike s randomiziranim, odnosno vjerojatnosnim algoritmima koji rade u polinomijalnom vremenu n. To implicira da postoji polinom p takav da protivnik radi u vremenu najviše $p(n)$ kada je sigurnosni parametar baš n. Osim toga, poštene strane će isto raditi u polinomijalnom vremenu, kako bi ovo bilo učinkovito u primjeni. Važno je naglasiti da protivnik može biti mnogo moćniji i raditi puno dulje od poštenih strana. Dodatno, poistovjećujemo pojam "male vjerojatnosti uspjeha" s vjerojatnostima manjim od bilo kojeg inverznog polinoma od n. Takve se vjerojatnosti nazivaju zanemarivim vjerojatnostima, što ćemo kasnije i formalno definirati. Definicija asimptotske sigurnosti je sljedećeg oblika:

> *Kažemo da je shema sigurna ako bilo koji protivnik koji radi u polinomijalnom vjerojatnosnom vremenu* (PPT) *uspijeva razbiti shemu s najviše zanemarivom vjerojatnošću.*

Asimptotski pristup omogućava poštenim stranama da same odrede sigurnost sheme odabirući odgovarajuću vrijednost sigurnosnog parametra n. Povećanjem sigurnosnog parametra povećava se i vrijeme potrebno za izvođenje sheme, tako da će poštene strane htjeti postaviti vrijednost sigurnosnog parametra na što je manju moguću uzimajući u obzir prirodu protivnika. Ukoliko sigurnosni parametar poistovjetimo s duljinom ključa, tada bi to odgovaralo činjenici da vrijeme potrebno za napad koji uključuje iscrpno pretraživanje eksponencijalno raste u duljini ključa. Pogledajmo primjer koji ilustrira navedeno.

Primjer 2.2. *([2]) Uzmimo za primjer shemu koja je asimptotski sigurna. Tada protivnik koji radi n^3 minuta može razbiti shemu s vjerojatnošću od $2^{40} \cdot 2^{-n}$ što je zanemariva funkcija od n. Kada je $n \leq 40$ tada protivnik koji radi 40^3 minuta, što je oko mjesec i pol, može razbiti shemu s vjerojatnošću 1. Dakle, takvi odabiri za n nisu korisni. Čak i za odabir $n = 50$ protivnik koji radi 50^3, odnosno oko tri mjeseca, može razbiti shemu s vjerojatnošću od otprilike 0.001, što možda u nekim slučajevima nije prihvatljivo. Kada bismo za n uzeli mnogo veći broj, primjerice $n = 500$, protivnik koji radi 200 godina razbio bi shemu s vjerojatnošću samo od približno 2^{-500}.*

Mogućnost povećanja sigurnosti povećanjem sigurnosnog parametra ima važne praktične posljedice, budući da poštenim stranama omogućuje obranu od povećanja računalne snage. Pogledajmo primjer koji ilustrira kakav učinak dostupnost bržih računala može imati na sigurnost u praksi.

Primjer 2.3. *([2]) Radi jednostavnosti, uzmimo kriptografsku shemu u kojoj poštene strane rade $10^6 \cdot n^2$ ciklusa, a koju protivnik koji radi $10^8 \cdot n^4$ ciklusa uspjeva razbiti s vjerojatnošću najviše $2^{-n/2}$. Također, pretpostavimo da sve strane koriste računala snage procesora $2GHz$ i da poštene strane postave sigurnosni parametar $n = 80$. Tada poštene strane rade $10^6 \cdot 6400$ ciklusa, odnosno 3.2 sekundi, a protivnik koji radi $10^8 \cdot 80^4$, odnosno oko 3 tjedna, uspjeva razbiti shemu s vjerojatnošću najviše 2^{-40}. Zamislimo da računala od $8GHz$ postanu dostupna te svi prijeđu na njih. Poštene strane mogu povećati sigurnosni parametar na $n = 160$ i održavati vrijeme rada od 3.2 sekunde ($10^6 \cdot 160^2$ ciklusa pri brzini od $8 \cdot 10^9$ ciklusa u sekundi). Povećavanje sigurnosnog parametra zahtijevat će i generiranje novog ključa. Protivnik pak mora raditi više od 8 milijuna sekundi, odnosno više od 13 tjedana kako bi postigao vjerojatnost uspjeha od 2^{-80}. Prelazak na brže računalo rezultiralo je težim poslom za protivnika.*

Važno je naglasiti da će čak i kada se koristi asimptotski pristup biti potrebno konkretno jamstvo sigurnosti prilikom implementacije kriptosustava u praksi. Ipak, općenito je slučaj da se asimptotski sigurnosni zahtjev može prevesti u konkretnu sigurnosnu granicu za bilo koju željenu vrijednost n.

2.2.1 Formalnija diskusija o asimptotskom pristupu

Učinkoviti algoritmi, u ovom kontekstu, su algoritmi koji rade u polinomijalnom vremenu.

Definicija 2.1. *([2]) Algoritam $\mathcal{A}$ radi u polinomijalnom vremenu ako postoji polinom p takav da se, za svaki ulaz $x \in \{0,1\}^*$, izračunavanje $\mathcal{A}(x)$ završava unutar najviše*

*$p(|x|)$ koraka, pri čemu $|x|$ označava duljinu od x, a operator * predstavlja Kleenijevu zvijezdu, odnosno* $\{0,1\}^* = \bigcup_{k=0}^{\infty} \{0,1\}^k$.

Pri asimptotskom pristupu, promatramo protivnike čije je vrijeme rada polinomijalno u sigurnosnom parametru n. Budući da vrijeme rada algoritma mjerimo u smislu duljine njegovog unosa, ponekad kao ulaz dajemo algoritme sa sigurnosnim parametrom u unarnom zapisu, odnosno kao 1^n - niz od n jedinica. Algoritmi koje poštene strane i protivnik pokreću mogu uzeti i druge ulaze osim sigurnosnog parametra, primjerice, poruku koja treba biti enkriptirana. Dopuštamo da njihovo vrijeme rada bude polinomijalno u ukupnoj duljini njihovih ulaza.

Također, dopuštamo da svi algoritmi budu randomizirani. Svaki takav algoritam može "baciti novčić" u svakom koraku svog izvršenja, odnosno algoritam može pristupiti nepristranom slučajnom bitu u svakom koraku. Ekvivalentno, možemo promatrati randomizirani algoritam kao onaj kojemu je, osim svog ulaza, dana uniformno raspoređena nasumična traka dovoljne duljine čije bitove može koristiti prema potrebi tijekom svog izvođenja. Uzimamo u obzir randomizirane algoritme najprije zato što je slučajnost bitna za kriptografiju i stoga poštene strane moraju biti vjerojatnostne pa je prirodno dopustiti i protivnicima da budu vjerojatnosni. Zatim i jer je randomizacija praktična i protivnicima daje dodatnu moć. Budući da je naš cilj modelirati sve realistične napade, preferiramo liberalniju definiciju učinkovitog računanja - želimo obuhvatiti i najgori scenarij.

Da bismo formalnije raspravili o zanemarivoj vjerojatnosti uspjeha, najprije moramo definirati zanemarivu funkciju.

Definicija 2.2. *([2]) Za funkciju f kažemo da je zanemariva ako za svaki polinom p postoji $N \in \mathbb{N}$ takav da za svaki $n > N$ vrijedi da je $f(n) < \frac{1}{p(n)}$.*

Dakle, zanemariva funkcija ona je koja je asimptotski manja od bilo koje inverzne polinomijalne funkcije. Gornju definiciju možemo i preformulirati: za svaki polinom p i sve dovoljno velike vrijednosti n vrijedi da je $f(n) < \frac{1}{p(n)}$. Ekvivalentna formulacija gore navedenog zahtijeva da za sve konstante c postoji N takav da za sve $n > N$ vrijedi da $f(n) < n^c$. Proizvoljnu zanemarivu funkciju obično označavamo s negl.
Tehnička prednost rada sa zanemarivim vjerojatnostima uspjeha je ta što podliježu određenim svojstvima zatvaranja što nam pokazuje iduća propozicija.

Propozicija 2.1. *Neka su negl_1 i negl_2 zanemarive funkcije. Tada vrijedi sljedeće:*

1. *Funkcija negl_3 definirana s $\mathsf{negl}_3(n) = \mathsf{negl}_1(n) + \mathsf{negl}_2(n)$ također je zanemariva.*

2. *Za bilo koji polinom p takav da je $p(n) > 0$ za sve n funkcija negl_4 definirana s $\mathsf{negl}_4(n) = p(n) \cdot \mathsf{negl}_1$ je zanemariva.*

Iz druge tvrdnje slijedi da ako se određeni događaj dogodi sa samo zanemarivom vjerojatnošću u određenom eksperimentu, onda se on događa s zanemarivom vjerojatnošću čak i ako se eksperiment ponovi polinomijalno mnogo puta. Posljedica toga je da ako funkcija g nije zanemariva, tada nije ni funkcija

$$f(n) \stackrel{\text{def}}{=} g(n)/p(n)$$

za bilo koji pozitivni polinom p.

Možemo zaključiti da se svaka sigurnosna definicija sastoji od definicije onoga što se smatra razbijanjem sheme i specifikacije moći protivnika. Kada je riječ o računskoj moći protivnika, od sada ćemo protivnika modelirati kao učinkovitog i stoga razmatrati samo suparničke strategije koje se mogu implementirati u vjerojatnostnom polinomijalnom vremenu. Osim toga, definicije će biti formulirane tako da se razbijanje sheme koje se dogodi sa zanemarivom vjerojatnošću ne smatra značajnim. Sada možemo postaviti opći oblik svake sigurnosne definicije.

> *Kažemo da je shema sigurna ako je za svakog vjerojatnostnog protivnika s polinomijalnim vremenom A koji izvodi napad nekog formalno specificiranog tipa, vjerojatnost da $\mathcal{A}$ uspije u napadu, gdje je uspjeh također formalno određen, zanemariva.*

Takva je definicija asimptotska jer je moguće da za male vrijednosti n protivnik može uspjeti s velikom vjerojatnošću, pa bi bilo točnije formulirati gornju tvrdnju na sljedeći način:

> *Kažemo da je shema sigurna ako za svakog PPT protivnika $\mathcal{A}$ koji izvodi napad nekog formalno specificiranog tipa, i za svaki pozitivni polinom p, postoji cijeli broj N takav da kada $n > N$ vjerojatnost da $\mathcal{A}$ uspije u napadu je manja od $1/p(n)$.*

Važno je naglasiti da ništa ne jamčimo za vrijednosti $n \leq N$.

U definiranju općeg pojma asimptotske sigurnosti napravili smo dva izbora - identificirali smo učinkovite suparničke strategije s klasom vjerojatnosnih algoritama s polinomijalnim vremenom i izjednačili malu vjerojatnost uspjeha sa zanemarivim vjerojatnostima. Iako su oba izbora do neke mjere proizvoljna, postoji više razloga zašto smo tako odlučili graditi teoriju, a ne primjerice definiranjem učinkovitih strategija kao

onih koje rade u kvadratnom vremenu ili malih vjerojatnosti uspjeha kao onih koje su ograničene s 2^n. Jedna od prednosti korištenja vjerojatnosnog polinomijalnog vremena kao mjere učinkovitosti je da ne zahtijeva da precizno specificiramo naš model računanja. To slijedi iz proširene Church-Turingove teze koja tvrdi da su svi "razumni" modeli izračunavanja polinomijalno ekvivalentni. Stoga, možemo predstaviti algoritme u pseudokodu visoke razine i biti uvjereni da ako naša analiza pokaže da se ovi algoritmi izvode u polinomijalnom vremenu, onda će se i svaka smislena implementacija izvoditi u polinomijalnom vremenu. Još jedna prednost vjerojatnosnih algoritama s polinomijalnim vremenom je činjenica da oni zadovoljavaju poželjna svojstva zatvaranja. Algoritam koji polinomijalno upućuje mnogo poziva potprogramu polinomijalnog vremena, i uz to radi samo polinomijalno izračunavanje, i sam će se izvoditi u polinomijalnom vremenu. Najvažnija značajka zanemarivih vjerojatnosti je ranije spomenuto svojstvo zatvaranja da bilo koji polinom množen zanemarivom funkcijom je i opet zanemariv. To posebno znači da ako algoritam izvrši polinomijalno mnogo poziva nekom potprogramu koji će biti neuspješan sa zanemarivom vjerojatnošću svaki put kada se pozove, tada je vjerojatnost da bilo koji od poziva tog potprograma ne uspije i dalje zanemariva.

2.2.2 Nužnost relaksacija

Kako smo već na početku ovog poglavlja naveli, računalna tajnost uvodi dvije relaksacije u odnosu na savršenu tajnost. Prva relaksacija bila je da je sigurnost zajamčena samo protiv učinkovitih protivnika, dok se druga odnosila na dopuštanje male vjerojatnosti uspjeha. Važnost relaksacija leži u postizanju praktičnih shema enkripcije, a posebice u zaobilaženju negativnih rezultata savršeno tajne enkripcije. Pretpostavimo da imamo shemu enkripcije u kojoj je veličina ključnog prostora K mnogo manja od veličine prostora poruka M što će implicirati da shema ne može biti savršeno tajna. Bez obzira na to kako je shema enkripcije konstruirana, primjenjuju se dva napada:

1. S obzirom na šifrirani tekst c, protivnik može dekriptirati c koristeći sve ključeve $k \in K$. Time se dobiva lista svih poruka čiji je c mogući šifrat. Kako ovaj popis ne može sadržavati sve iz M zbog uvjeta da je $|K| < |M|$, ovaj napad propušta neke informacije o poruci koja je enkriptirana. Zamislimo scenarij kada bi protivnik izvodio napad na poznati otvoreni tekst i vidio da šifrirani tekstovi $c_1, \ldots, c_l$ odgovaraju redom porukama $m_1, \ldots, m_l$. Tada bi mogao pokušati dekriptirati svaki od šifriranih tekstova koristeći sve ključeve dok ne pronađe ključ k za koji je $\mathsf{Dec}_k(c_i) = m_i$ za sve i. S obzirom na šifrirani tekst c koji je šifrirani oblik nepoznate poruke m, gotovo je sigurno slučaj da je $\mathsf{Dec}_k(c) = m$. Napadi iscrpnog pretraživanja poput prethodnog omogućuju protivniku

da uspije s vjerojatnošću jednakoj 1 u vremenu linearnom u $|K|$.

2. Kada protivnik sazna da šifrirani tekstovi $c_1, \ldots, c_l$ odgovaraju redom porukama $m_1, \ldots, m_l$, on može pogoditi ujednačeni ključ $k \in K$ i provjeriti je li $\mathsf{Dec}_k(c_i) = m_i$ za sve i. Tada protivnik može koristiti k za dekripciju svega što su poštene strane naknadno enkriptirale. Ovdje protivnik radi u konstantnom vremenu i uspijeva s nenultom, iako vrlo malom, vjerojatnošću $1/|K|$.

Ukoliko želimo šifrirati mnogo poruka pomoću jednog kratkog ključa, sigurnost se može postići samo ako ograničimo vrijeme rada protivnika tako da protivnik nema dovoljno vremena da izvrši pretragu "grubom snagom", te ako smo spremni dopustiti vrlo malu vjerojatnost uspjeha, što će značiti da je drugi napad isključen.

3 Definicija računalno sigurne enkripcije

Kako bismo mogli dati definiciju računalno sigurne enkripcije, najprije moramo poznavati pojmove determinističkog i randomiziranog algoritma. Deterministički algoritam je algoritam koji će pri svakom izvršavanju za isti unos dati isti izlaz slijedeći svaki put isti niz naredbi, bez obzira u kojim se uvjetima izvršava. S druge strane, postoje stohastički algoritmi koji barem u jednom dijelu izvršavanja donose odluku o daljnjem tijeku izvršavanja slučajnim odabirom. To bi značilo da za iste ulaze, pod istim uvjetima, isti stohastički algoritmi mogu dati različite izlaze. Dakle, algoritme s obzirom na način donošenja odluka možemo podijeliti na determinističke i stohastičke. Algoritmi koje ćemo u nastavku često spominjati su vjerojatnosni, odnosno randomizirani algoritmi koji su posebna vrsta stohastičkih algoritama.

Definicija 3.1. *([3]) Za algoritam kažemo da je randomiziran ukoliko donosi slučajne (ili pseudoslučajne) odluke.*

Također, dajemo definiciju i sheme enkripcije s tajnim ključem.

Definicija 3.2. *([2]) Shema enkripcije s tajnim ključem je trojka randomiziranih algoritama polinomijalnog vremena* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$*:*
1. Algoritam za generiranje ključa Gen *za ulaz prima* 1^n *(tj. sigurnosni parametar zapisan unarno) te kao izlaz daje ključ* k*. Pišemo* $k \leftarrow \mathsf{Gen}(1^n)$*, naglašavajući da je* Gen *randomizirani algoritam. Bez smanjenja općenitosti, pretpostavljat ćemo da bilo koji ključ* k *dobiven* Gen *algoritmom zadovoljava nejednakost* $|k| \geq n$*.*
2. Algoritam enkripcije Enc *za ulaz prima ključ* k *i poruku otvorenog teksta* $m \in \{0,1\}^*$*, te kao izlaz daje šifrirani tekst* c*. Budući da* Enc *može biti randomiziran, ovo ćemo pisati kao* $c \leftarrow \mathsf{Enc}_k(m)$*.*
3. Algoritam dekripcije Dec *za ulaz prima ključ* k *i šifrirani tekst* c *te na izlazu daje ili poruku* m *ili grešku. Pretpostavljamo da je* Dec *deterministički algoritam, pa pišemo* $m := \mathsf{Dec}_k(c)$ *(u slučaju da se nije dogodila greška). Generičku grešku označavamo simbolom* $\perp$*.*
Za svaki n*, za svaki ključ* k *koji da algoritam* $\mathsf{Gen}(1^n)$ *i za svaki* $m \in \{0,1\}^*$ *mora vrijediti* $\mathsf{Dec}_k(\mathsf{Enc}_k(m)) = m$*.*
Ako je $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *takav da je za* k *dobiven algoritmom* $\mathsf{Gen}(1^n)$ *algoritam* Enc_k *definiran samo za poruke* $m \in \{0,1\}^{l(n)}$*, onda kažemo da je* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *shema enkripcije s tajnim ključem fiksne duljine za poruke duljine* $l(n)$*.*

Gotovo uvijek će biti slučaj da $\mathsf{Gen}(1^n)$ ispisuje ujednačeni n-bitni niz kao ključ te ćemo tada izostaviti Gen i shemu enkripcije tajnog ključa definirati pomoću para algoritama $(\mathsf{Enc}, \mathsf{Dec})$. Također, otvoreni tekst m promatramo kao element skupa $\{0,1\}^*$.

3.1 Osnovna definicija sigurnosti

Za početak ćemo promatrati osnovni pojam sigurnosti za enkripciju s tajnim ključem. Zanimat će nas sigurnost od napada na šifrirani tekst gdje protivnik promatra samo jedan šifrirani tekst ili, ekvivalentno, sigurnost kada se zadani ključ koristi za enkripciju samo jedne poruke.

3.1.1 Motivacija

Definicija sigurnosti sastoji se od dvije komponente:

- **modela prijetnje** - odnosno specifikacije pretpostavljene moći protivnika,
- **sigurnosnog cilja** - obično specificiranog opisom onoga što čini "razbijanje" sheme.

Kao što smo ranije spomenuli, u nastavku ćemo promatrati najjednostavni model prijetnje - onaj gdje imamo protivnika koji promatra enkripciju jedne poruke. Pri tome ćemo imati dvije pretpostavke o protivniku - prva će biti da samo prisluškuje, a druga da radi u polinomijalnom vremenu. Obratimo pozornost da pri tome nemamo nikakve pretpostavke o protivničkoj strategiji prilikom dekripcije enkriptiranog teksta koji promatra, što je ključno za dobivanje smislenih pojmova sigurnosti jer će definicija osiguravati zaštitu od bilo kojeg računalno ograničenog protivnika, bez obzira na algoritam koji koristi.

Ideja koja leži iza ovakve definicije je da protivnik ne bi trebao biti u mogućnosti dobiti bilo kakve djelomične informacije o otvorenom tekstu iz enkriptiranog teksta. Definicija semantičke sigurnosti točno formalizira ovaj pojam i bila je prva predložena definicija računalno sigurne enkripcije. Ipak, semantička sigurnost je složena i vrlo teška za rad, te postoji ekvivalentna definicija koja se zove definicija nerazlučivosti koja je mnogo jednostavnija i s kojom ćemo se baviti u nastavku.

Definicija nerazlučivosti oblikovana je prema alternativnoj definiciji savršene tajnosti. Ta definicija savršene tajnosti razmatra eksperiment $\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}$ u kojem protivnik $\mathcal{A}$ šalje dvije poruke m_0 i m_1, a zatim dobiva enkripciju jedne od tih poruka pomoću jedinstvenog ključa. Definicija kaže da je shema Π sigurna ako nijedan protivnik $\mathcal{A}$ ne može odrediti koja je od poruka m_0, m_1 enkriptirana i njemu vraćena s vjerojatnošću različitom od $1/2$, što je vjerojatnost da je $\mathcal{A}$ točan akore pogađa.
I dalje ćemo promatrati isti eksperiment, ali uvodimo dvije relaksacije koje čine temeljne elemente računalne sigurnosti:

1. Razmatrat ćemo samo protivnike koji rade u polinomijalnom vremenu, dok je alternativna definicija savršene tajnosti razmatrala i protivnike s neograničenim vremenom rada.

2. Dopuštat ćemo da protivnik može odrediti enkriptiranu poruku s vjerojatnošću zanemarivo boljom od 1/2.

Ipak, vjerojatno najveća razlika je da ćemo sada eksperimentirati s parametrom sigurnosti n. Mjerit ćemo vrijeme rada protivnika $\mathcal{A}$, kao i njegovu vjerojatnost uspjeha kao funkciju od n. Formalno, $\mathsf{Priv}K^{\mathsf{eav}}_{A,\Pi}(n)$ bit će oznaka eksperimenta koji se izvodi sa sigurnosnim parametrom n te ćemo s

$$P(\mathsf{Priv}K^{\mathsf{eav}}_{A,\Pi}(n) = 1) \tag{1}$$

označavati vjerojatnost da će rezultat danog eksperimenta biti 1. Osim toga, izričito ćemo zahtijevati od protivnika da pošalje dvije poruke m_0, m_1 jednake duljine.

3.1.2 Nerazlučivost u prisutnosti prisluškivača

Eksperiment $\mathsf{Priv}K^{\mathsf{eav}}_{A,\Pi}(n)$ definiran je za bilo koju shemu enkripcije s tajnim ključem $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, bilo kojeg protivnika $\mathcal{A}$ i bilo koju vrijednost sigurnosnog parametra n na sljedeći način:

Definicija 3.3. *([2]) **Eksperiment kontradiktorne nerazlučivosti** $\mathsf{Priv}K^{\mathsf{eav}}_{A,\Pi}(n)$:*

1. *Protivnik $\mathcal{A}$ dobiva 1^n kao ulaz, i za izlaz daje par poruka m_0, m_1 pri čemu vrijedi $|m_0| = |m_1|$.*

2. *Ključ k generira se pomoću algoritma $\mathsf{Gen}(1^n)$ i odabire se uniformni bit $b \in \{0,1\}$. Šifrirani tekst $c \leftarrow \mathsf{Enc}_k(m_b)$ se izračunava i daje protivniku $\mathcal{A}$. Šifrirani tekst c nazivamo izazovnim šifriranim tekstom.*

3. *$\mathcal{A}$ vraća bit b'.*

4. *Rezultat eksperimenta će biti 1 ako vrijedi da je $b' = b$, a 0 inače. Ako je $\mathsf{Priv}K^{\mathsf{eav}}_{A,\Pi}(n) = 1$, kažemo da je protivnik $\mathcal{A}$ uspio.*

Ne postavljamo ograničenja na duljinu m_0 i m_1, osim što zahtijevamo da su iste duljine. S obzirom da $\mathcal{A}$ radi u polinomijalnom vremenu, tada će i m_0 i m_1 imati polinomijalnu duljinu u n. Kada je Π shema fiksne duljine za poruke duljine $l(n)$, eksperiment modificiramo tako da zahtijevamo $m_0, m_1 \in \{0,1\}^{l(n)}$.
Prisjetimo se, zahtijevali smo da protivnik može samo prisluškivati. To implicira da

je njegov unos ograničen na jedan šifrirani tekst, te da nema daljnju interakciju s pošiljateljem ili primateljem. Definicija nerazlučivosti navodi da je shema enkripcije sigurna ako nijedan PPT protivnik $\mathcal{A}$ ne uspije pogoditi koja je poruka enkriptirana u $\mathsf{Priv}K^{\mathsf{eav}}_{A,\Pi}(n)$ eksperimentu s vjerojatnošću znatno boljom od slučajnog pogađanja, odnosno većom od 1/2.

Definicija 3.4. *([2]) Kažemo da shema enkripcije s tajnim ključem* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *ima nerazlučivu enkripciju u prisutnosti prisluškivača, ili kažemo da je* eav*-sigurna, ako za sve randomizirane protivnike s polinomijalnim vremenom* $\mathcal{A}$ *postoji zanemariva funkcija negl takva da za svaki n vrijedi*

$$P(\mathit{PrivK}^{\mathsf{eav}}_{A,\Pi}(n) = 1) \leq \frac{1}{2} + \mathit{negl}(n).$$

Svaka savršeno tajna shema enkripcije ima nerazlučive enkripcije u prisutnosti prisluškivača, pa je jasno da je prethodna definicija slabija od definicije savršene tajnosti. Ono što mi želimo pokazati je da postoje sheme enkripcije koje zadovoljavaju gore navedeno u kojima je ključ kraći od poruke. Odnosno, pokazat ćemo da postoje sheme koje zadovoljavaju definiciju nerazlučivosti, ali ne i definiciju savršene tajnosti.

Definicija nerazlučivosti zahtijeva da niti jedan PPT protivnik ne može odrediti koja je od dvije poruke enkriptirana s vjerojatnošću znatno boljom od 1/2. Evivalentna formulacija bila bi da se svaki PPT protivnik ponaša isto, bez obzira vidi li enkripciju od m_0 ili m_1. Budući da $\mathcal{A}$ daje jedan bit, tvrdnja da će se ponašati isto zapravo znači da u svakom slučaju vraća 1 s gotovo istom vjerojatnošću. Formalni zapis mogao bi ići na isti način, ali bismo koristili fiksni bit b umjesto da ga nasumično biramo.

Neka $\mathsf{out}_{\mathcal{A}}(\mathsf{Priv}K^{\mathsf{eav}}_{\mathcal{A},\Pi}(n,b))$ označava izlazni bit b' od $\mathcal{A}$ u eksperimentu. Sljedeća tvrdnja kaže da nijedan protivnik $\mathcal{A}$ ne može odrediti radi li se o eksperimentu $\mathsf{Priv}K^{\mathsf{eav}}_{\mathcal{A},\Pi}(n,0)$ ili $\mathsf{Priv}K^{\mathsf{eav}}_{\mathcal{A},\Pi}(n,1)$.

Definicija 3.5. *([2]) Kažemo da shema enkripcije s tajnim ključem* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *ima nerazlučive enkripcije u prisutnosti prisluškivača ako za sve PPT protivnike* $\mathcal{A}$ *postoji zanemariva funkcija negl takva da vrijedi:*

$$P(\mathit{out}_{\mathcal{A}}(\mathit{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n,0)) = 1) - P(\mathit{out}_{\mathcal{A}}(\mathit{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n,1)) = 1) \leq \mathit{negl}(n).$$

3.1.3 Enkripcija i duljina otvorenog teksta

Općenito pojam sigurne enkripcije ne zahtijeva da shema enkripcije sakrije duljinu otvorenog teksta. Čak štoviše, sve najčešće korištene sheme enkripcije otkrivaju duljinu otvorenog teksta ili barem njezinu blisku aproksimaciju. Motivacija koja leži iza

toga je što je nemoguće podržati poruke proizvoljne duljine, a istovremeno skrivati sve informacije o duljini otvorenog teksta. Isto tako, često je duljina otvorenog teksta već javna ili nije povjerljiva. Međutim, nekada propuštanje duljine otvorenog teksta može stvoriti problem. Zato je vrlo važno na početku utvrditi je li propuštanje informacije o duljini otovorenog teksta problem i kada jest - poduzeti korake za ublažavanje ili sprječavanje takvog scenarija manipuliranjem duljinom poruka prije same enkripcije.

Navedimo neke primjere kada nije poželjno propuštanje duljine otvorenog teksta.

Jednostavni numerički/tekstualni podaci Promotrimo primjer sheme enkripcije koja točno otkriva duljinu otvorenog teksta. Tada bi takva enkripcija riječi "yes" i "no" odala koji odgovor je netko ponudio.

Automatski prijedlozi Mnoga web-mjesta nude opciju “automatskog dovršavanja” ili “auto-sugestije” pomoću koje web-poslužitelj predlaže popis mogućih riječi ili izraza na temelju djelomičnih informacija koje je korisnik već unio. Broj prijedloga može otkriti informacije o slovima koje je korisnik do sada utipkao.

Pretraživanje baze podataka Zamislimo da korisnik postavlja upit na bazu za sve zapise koji odgovaraju nekom pretraživanom pojmu. Slično kao u prethodnom slučaju, broj vraćenih zapisa može otkriti mnogo informacija o tome što je korisnik tražio.

Komprimirani podaci Komprimiranje podataka je proces kodiranja informacija korištenjem manje bitova od izvornog prikaza. Ako je otvoreni tekst komprimiran prije enkripcije, tada bi se informacije o otvorenom tekstu mogle otkriti čak i ako su svi šifrirani podaci iste, fiksne duljine. Primjerice, kratki komprimirani otvoreni tekst bi ukazivao na to da izvorni (nekomprimirani) otvoreni tekst ima puno suvišnosti. Ako protivnik može kontrolirati dio onoga što se šifrira, može saznati i dodatne informacije o otvorenom tekstu. Napadi te vrste korišteni su za napade na šifrirani HTTP promet za otkrivanje tajnih kolačića sesije.

4 Konstrukcija sigurnih shema enkripcije

Kako bismo mogli konsturirati sigurne sheme enkripcije, potrebno je uvesti pojmove pseudoslučajnih generatora (PRG) i protočnih šifri *(eng. Stream šifre).*

4.1 Pseudoslučajni generatori

Pseudoslučajni generator G je učinkoviti deterministički algoritam za transformaciju kratkog uniformnog niza (koji ćemo zvati sjeme) u duži izlazni niz koji će samo izgledati uniformno, a koji ćemo zvati pseudoslučajni niz. Možemo reći da generator pseudoslučajnosti koristi malu količinu prave slučajnosti kako bi generirao veliku količinu pseudoslučajnosti. Pseudoslučajni generator vrlo je koristan kada je potrebno generirati velik broj bitova koji djeluju nasumično generirano, budući da je generiranje pravih nasumičnih bitova teško i sporo. U kriptografiji je određivanje dobrih pseudoslučajnih generatora od krucijalne važnosti. Ukoliko protivnik može razlikovati izlaz generatora od uniformnog niza, a mi nismo upoznati sa strategijom koju koristi, sigurnost može biti značajno narušena. Kako bismo to izbjegli, izlaz svakog pseudoslučajnog generatora trebao bi svakom učinkovitom promatraču izgledati kao uniformni niz. To bi značilo da pseudoslučajni generator treba proći sve (učinkovite) statističke testove, odnosno za bilo koji učinkoviti statistički test (ili diferencijator) D, vjerojatnost da D vrati 1 kada je zadan izlaz pseudoslučajnog generatora trebala bi biti približno jednaka vjerojatnosti da D vrati 1 kada je zadan uniformni niz iste dužine. *Važno je dodati da formalno gledano, nema smisla reći da je bilo koji fiksni niz pseudoslučajan, kao ni bilo koji fiksni niz nazvati slučajnim. Ipak, ovdje ćemo neformalno niz uzorkovan prema uniformnoj distribuciji zvati "uniformnim nizom", a izlaz niza generatora pseudoslučajnih nizova "pseudoslučajnim nizom".*

S druge strane, možemo raspravljati kakva je to pseudoslučajna distribucija. Neka je Dist distribucija na l-bitnim nizovima. To će značiti da Dist svakom nizu iz $\{0,1\}^l$ dodjeljuje neku vjerojatnost. Uzorkovanje iz Dist predstavlja odabir l-bitnog niza prema ovoj vjerojatnosnoj distribuciji. Neformalno, Dist je pseudoslučajna distribucija ako se eksperiment u kojem se niz uzorkuje iz Dist ne razlikuje od eksperimenta u kojem se uzorkuje uniformni niz duljine l. Bilo kojem algoritmu koji radi u polinomijalnom vremenu trebalo bi biti nemoguće reći, s vjerojatnošću većom od pogađanja, je li mu dan niz uzorkovan prema Dist-u ili mu je dan uniformni l-bitni niz. Drugim riječima, pseudoslučajni niz jednako je dobar kao i uniformni niz sve dok uzimamo u obzir samo promatrače koji rade u polinomijalnom vremenu.

4.1.1 Formalna definicija

Kako bismo mogli uvesti formalnu definiciju pseudoslučajnosti, najprije trebamo definirati funkciju $G : \{0,1\} \to \{0,1\}^l$ i Dist kao distribuciju na l-bitnim nizovima dobivene odabirom uniformnog $s \in \{0,1\}^n$ i izlazom $G(s)$. Tada vrijedi da je G pseudoslučajni generator ako i samo ako je distribucija Dist pseudoslučajna.

Kao što smo u prethodnom dijelu već naveli, G je pseudoslučajni generator ako nijedan učinkoviti diferencijator ne može otkriti je li mu G zadao izlaz niza ili je niz odabran slučajno iz uniformne distribucije. Kao u definiciji 3.4, ovo je formalizirano zahtjevom da svaki učinkoviti algoritam daje 1 s gotovo istom vjerojatnošću kada je zadan $G(s)$ (za uniformno sjeme s) i kada je zadan uniformni niz. Definiciju možemo dobiti i u asimptotskom okruženju ukoliko dopustimo da sigurnosni parametar n odredi duljinu sjemena. Zatim inzistiramo da G bude izračunljiv učinkovitim algoritmom, a iz praktičnih razloga zahtijevamo da G-ov izlaz bude duži od njegovog ulaza.

Definicija 4.1. *([2]) Neka je l polinom i neka je G deterministički algoritam polinomijalnog vremena takav da je za bilo koji n i bilo koji ulaz $s \in \{0,1\}^n$ rezultat $G(s)$ niz duljine $l(n)$. Kažemo da je G pseudoslučajan generator ako vrijedi sljedeće:*

1. ***Proširenje**: Za svaki n vrijedi da je $l(n) > n$.*

2. ***Pseudoslučajnost**: Za bilo koji PPT algoritam D postoji zanemariva funkcija negl takva da*

$$|P(D(G(s)) = 1) - P(D(r) = 1) \leq \mathsf{negl}(n)|,$$

pri čemu prva vjerojatnost odgovara vjerojatnosti uniformnog odabira $s \in \{0,1\}^n$ i slučajnosti od D, a druga vjerojatnost odgovara vjerojatnosti uniformnog odabira $r \in \{0,1\}^{l(n)}$ i slučajnosti od D.

Dodatno, l zovemo faktorom ekspanzije od G.

4.1.2 Rasprava

Kada bismo promotrili slučaj $l(n) = 2n$ gdje G udvostručuje duljinu unosa vidjeli bismo da je izlazna distribucija pseudoslučajnog operatora G daleko od uniformne. Naime, u slučaju uniformne distribucije na $\{0,1\}^{2n}$, svaki od 2^{2n} mogućih nizova bira se s vjerojatnošću točno 2^{-2n}. S druge strane, promotrimo distribuciju izlaza od G kada se G pokreće na uniformnom sjemenu. Kada G primi ulaz duljine n, broj različitih nizova u rasponu od G je zato najviše $2^n/2^{2n} = 2^{-n}$, a vidimo da se većina nizova duljine $2n$ ne pojavljuje kao izlaz. Ono što nam to govori jest da je trivijalno razlikovati slučajni niz

i pseudoslučajni niz ako raspolažemo s neograničeno vremena. Promatrajmo ponovno isti G i uzmimo u obzir diferencijatora D koji radi u eksponencijalnom vremenu i na sljedeći način: $D(w)$ vraća 1 ako i samo ako postoji $s \in \{0,1\}^n$ takav da je $G(s) = w$. Ovo se izračunavanje provodi u eksponencijalnom vremenu iscrpnim izračunavanjem $G(s)$ za svaki $s \in \{0,1\}^n$. Ako je w bio izlaz koji daje G, tada D vraća 1 s vjerojatnošću 1. Suprotno tomu, ako je w uniformno raspoređen u $\{0,1\}^{2n}$, tada je vjerojatnost da postoji s takav da vrijedi $G(s) = w$ najviše 2^{-n}. Odnosno, vrijedi

$$|P(D(r) = 1) - P(D(G(s)) = 1)| \geq 1 - 2^{-n}.$$

Ovo je samo još jedan primjer napada grubom silom i ne proturječi pseudoslučajnosti od G budući da napad nije učinkovit.

4.1.3 Sjeme i njegova duljina

Možemo povući paralelu između veze enkripcijske sheme i kriptografskog ključa s vezom sjemena i pseudoslučajnog generatora. Baš kao kriptografski ključ, i sjeme se mora tajiti od protivnika. Ono se bira uniformno i dovoljno dugačko tako da protivnik primjenom grube sile nije u mogućnosti doći do njega. U asimptotskom smislu, to će značiti da ćemo duljinu sjemena birati tako da bude jednaka sigurnosnom parametru, pa će iscrpna pretraga svih mogućih sjemena zahtijevati eksponencijalno vrijeme. U praksi, sjeme mora biti dovoljno dugo tako da je nemoguće isprobati sve moguće opcije unutar određenog vremenskog okvira.

4.2 Protočne šifre

Definicija pseudoslučajnih generatora koju smo ponudili u prethodnom poglavlju podrazumijeva da je faktor ekspanzije konstanta te da generator proizvodi cijeli svoj izlaz iz jednog pokretanja. U praksi ćemo za instanciranje pseudoslučajnih generatora koristiti protočne šifre koje su koncipirane na drugačiji način. Protočne šifre proizvode pseudoslučajne izlazne bitove postupno i na zahtjev, tako da aplikacija može zatražiti točno onoliko pseudoslučajnih bitova koliko je potrebno. To predstavlja značajan napredak u učinkovitosti i fleksibilnosti. Dakle, aplikacija može zahtijevati manje bitova - odnosno onoliko koliko je dovoljno, dok istovremeno ne postoji gornja granica za broj bitova koji se mogu zatražiti.

Formalno, protočne šifre shvaćat ćemo kao par deterministčkih algoritama (Init, GetBits) pri čemu:

Init na ulazu uzima sjeme s i opcionalni inicijalizacijski vektor IV te vraća početno stanje st_0.

GetBits na ulazu uzima informacije o stanju st_i te vraća bit y i stanje st_{i+1}. *U praksi, y je blok od nekoliko bitova, no ovdje ga tretiramo kao jedan bit radi općenitosti i jednostavnosti.*

S obzirom na protočnu šifru i bilo koji željeni faktor proširenja l, možemo definirati algoritam G koji preslikava ulaze duljine n u izlaze duljine $l(n)$.

Algoritam 1: Konstrukcija G_l pomoću (Init, GetBits)

Ulaz: Sjeme s i opcionalni inicijalizacijski vektor IV

Izlaz: $y_1, \ldots, y_l$

$\mathsf{st}_0 := \mathsf{Init}(s, IV)$

for $i = 1$ *to* l **do**

  $(y_i, \mathsf{st}_i) := \mathsf{GetBits}(\mathsf{st}_{i-1})$

end

return $y_1, \ldots, y_l$

Kao što vidimo, algoritam pokreće Init, a zatim opetovano izvodi GetBits ukupno l puta. Protočnu šifru smatramo sigurnom, u osnovnom smislu, ako ne zahtijeva IV i ako za bilo koji polinom l takav da $l(n) > l$ vrijedi da je funkcija G_l konstruirana na prethodno opisan način pseudoslučajni generator s faktorom ekspanzije l.

4.3 Dokaz redukcijom

Želimo li dokazati da je određena konstrukcija računalno sigurna, morat ćemo se osloniti na nedokazane pretpostavke, osim ako je shema informacijsko teoretski sigurna. Nažalost, za većinu modernih kriptografskih konstrukcija ne može se bezuvjetno dokazati da su sigurne. Takvi bi dokazi zahtijevali odgovore na još uvijek otvorena pitanja u teoriji računalne složenosti. Zbog toga se dokazi sigurnosti obično oslanjaju na pretpostavke. Moderna kriptografija zahtijeva da sve takve pretpostavke budu eksplicitne i matematički precizne - prvenstveno jer to zahtijevaju matematički dokazi sigurnosti, ali i zbog validacija pretpostavki, usporedbi shema i razumijevanja potrebnih pretpostavki. Mi ćemo najprije pretpostaviti da je neki matematički problem težak, ili da je neki kriptografski postupak niske razine siguran, a zatim dokazati da je dana konstrukcija temeljena na tom problemu ili postupku sigurna pod ovom pretpostavkom. Taj ćemo pristup koristiti vođeni idejom da pretpostavke koje je jednostavnije izreći je i lakše proučavati te potencijalno opovrgnuti. Tako je primjerice pretpostavka da je neki matematički problem teško riješiti jednostavnija za proučavanje i evaluaciju od pretpostavke da shema enkripcije zadovoljava složenu sigurnosnu definiciju. Još jedna

prednost oslanjanja na pretpostavke "niže razine", umjesto primjerice na pretpostavku da je konstrukcija sigurna, je da se te pretpostavke obično mogu koristiti u drugim konstrukcijama te mogu osigurati modularnost. Dokaz da je kriptografska konstrukcija sigurna sve dok je neki temeljni problem težak općenito se nastavlja predstavljanjem eksplicitne redukcije koja pokazuje kako transformirati bilo kojeg učinkovitog protivnika $\mathcal{A}$ koji uspijeva razbiti konstrukciju u učinkoviti algoritam $\mathcal{A}'$ koji rješava neki problem kojeg smo smatrali teškim.

4.3.1 Skica dokaza

Najprije ćemo pretpostaviti da se neki problem X ne može riješiti u nekom točno definiranom smislu bilo kojim algoritmom polinomijalnog vremena, osim eventualno sa zanemarivom vjerojatnošću. Zatim želimo dokazati da je neka kriptografska konstrukcija Π sigurna (opet, u nekom smislu koji je točno definiran). Koraci dokaza bit će sljedeći:

1. Pretpostavimo da neki učinkoviti protivnik $\mathcal{A}$ napada kriptografsku konstrukciju Π. S $\epsilon(n)$ označimo vjerojatnost uspjeha ovog protivnika.

2. Konstruiramo učinkoviti algoritam $\mathcal{A}'$, kojeg ćemo zvati redukcija, a koji pokušava riješiti problem X koristeći protivnika $\mathcal{A}$ kao potprogram. Pri tom, znamo da algoritam $\mathcal{A}'$ ne zna ništa o tome kako $\mathcal{A}$ radi - zna samo da će $\mathcal{A}$ napasti Π. S obzirom na neki ulazni primjer x problema X, naš algoritam $\mathcal{A}'$ će za $\mathcal{A}$ simulirati instancu od Π tako da:

   (a) Koliko $\mathcal{A}$ može reći, u interakciji je s Π. Odnosno, rezultat od $\mathcal{A}$ kada ga $\mathcal{A}'$ pokreće kao potprogram trebala bi biti distribuirana identično (ili barem blizu) kao rezultat od $\mathcal{A}$ kada je u interakciji sa samim Π.

   (b) Ako $\mathcal{A}$ uspije razbiti instancu Π koju simulira $\mathcal{A}'$, to bi trebalo omogućiti $\mathcal{A}'$ da riješi instancu x koja mu je dana, barem s inverznom polinomijalnom vjerojatnošću $1/p(n)$.

3. Promatrani zajedno, 2.(a) i 2.(b) impliciraju da $\mathcal{A}'$ rješava X s vjerojatnošću $\epsilon(n)/p(n)$. Ako $\epsilon(n)$ nije zanemariv, onda nije ni $\epsilon(n)/p(n)$. Štoviše, ako je $\mathcal{A}$ učinkovit, onda dobivamo učinkoviti algoritam $\mathcal{A}'$ koji rješava X s nezanemarivom vjerojatnošću, što je u suprotnosti s početnom pretpostavkom.

4. Uzimajući u obzir pretpostavku o X, zaključujemo da nijedan učinkoviti protivnik $\mathcal{A}$ ne može razbiti Π s nezanemarljivom vjerojatnošću. Odnosno, Π je računalno siguran.

4.4 Sigurna shema enkripcije fiksne duljine

Upravo skica dokaza iz prethodnog poglavlja omogućit će nam da pokažemo kako koristiti bilo koji pseudoslučajni generator za konstruiranje shema enkripcije. Pseudoslučajni generator pruža prirodan način za izradu sigurne sheme enkripcije fiksne duljine s ključem kraćim od poruke. Ovdje ćemo se susresti s pojmom šifriranja s jednokratnim ključem (blokom) *(eng. one-time pad)* u kojem se enkripcija vrši tako da se kroz XOR provode slučajni blok i poruka. Ideja je da umjesto slučajnog koristimo pseudoslučajni blok. Međutim, primatelj i pošiljatelj umjesto da dijele pseudoslučajni blok mogu dijeliti sjeme koje se koristi za generiranje tog bloka kada je potrebno. Sjeme će biti kraće od bloka, a time kraće i od poruke. Promatrano sa sigurnosne strane, ideja je da pseudoslučajni niz izgleda nasumično bilo kojem protivniku koji radi u polinomijalnom vremenu te da tako računalno ograničen prisluškivač ne može razlikovati poruku šifriranu korištenjem jednokratnog ključa ili poruku šifriranu pseudo-jednokratnom shemom šifriranja.

4.4.1 Formalna konstrukcija

Odaberimo neku fiksnu duljinu poruke l i neka G bude pseudoslučajni generator s faktorom ekspanzije l, odnosno neka vrijedi $|G(s)| = l(|s|)$. Kako je shema enkripcije definirana s tri algoritma, definirat ćemo svaki od njih. Algoritam za generiranje ključa je trivijalan: $\mathsf{Gen}(1^n)$ daje uniformni ključ k duljine n. Enkripcija se vrši primjenom G na ključ (koji služi kao sjeme) kako bi se dobio jednokratni ključ (blok) koji se zatim provodi kroz XOR s otvorenim tekstom. Dekripcija primjenjuje G na ključ i XOR-om dobiveni blok s enkriptiranim tekstom za dohvaćanje poruke. Shema enkripcije s tajnim ključem temeljena na bilo kojem pseudoslučajnom generatoru formalno je dana idućom konstrukcijom.

Konstrukcija 4.1. *([2]) Neka je G pseudoslučajni generator s faktorom ekspanzije l. Definirajmo shemu enkripcije s tajnim ključem za poruke duljine l na sljedeći način:*

Gen: ulaz je 1^n te odabiremo uniformni $k \in \{0,1\}^n$ i ispisujemo ga kao ključ,

Enc: ulaz je ključ $k \in \{0,1\}^n$ i poruka $m \in \{0,1\}^{l(n)}$, a izlaz je enkripcija

$$c := G(k) \oplus m,$$

Dec: ulaz je ključ $k \in \{0,1\}^n$ i enkriptirani tekst $c \in \{0,1\}^{l(n)}$, a izlaz je poruka

$$m := G(k) \oplus c.$$

Teorem 4.1. *([2]) Ako je G pseudoslučajni generator, tada je konstrukcija* 4.1. *shema enkripcije s tajnim ključem fiksne duljine koja ima nerazlučive enkripcije u prisutnosti prisluškivača.*

Dokaz. Neka Π označava konstrukciju 4.1. Želimo pokazati da Π zadovoljava definiciju 3.3. Naime, pokazujemo da za bilo kojeg vjerojatnosnog protivnika $\mathcal{A}$ s polinomijalnim vremenom postoji zanemariva funkcija negl takva da vrijedi

$$P(\mathsf{Priv}K^{\mathsf{eav}}_{\mathcal{A},\Pi}(n) = 1) \leq \frac{1}{2} + \mathsf{negl}(n) \tag{2}$$

Možemo naslutiti da kada bi Π koristio uniformni blok umjesto pseudoslučajnog $G(k)$, tada bi rezultirajuća shema bila identična shemi enkripcije jednokratnim ključem i $\mathcal{A}$ ne bi mogao točno pogoditi koja poruka je enkriptirana s vjerojatnošću boljom od $1/2$. Stoga ako nejednakost (2) ne vrijedi, tada $\mathcal{A}$ mora implicitno razlikovati izlaz G od slučajnog niza. Ovo ćemo pokazati pomoću redukcije. Odnosno, pokazat ćemo kako koristiti $\mathcal{A}$ za konstruiranje učinkovitog diferencijatora D, sa svojstvom da je D-ova sposobnost da razlikuje G-ov izlaz od uniformnog niza izravno povezana s $\mathcal{A}$-ovom sposobnošću da odredi koju je poruku šifrirao Π. Sigurnost od G tada implicira sigurnost od Π.
Neka je $\mathcal{A}$ proizvoljan PPT protivnik. Konstruiramo diferencijator D koji uzima niz w kao ulaz, a čiji je cilj odrediti je li w odabran uniformno (tj. w je slučajni niz) ili je w generiran odabirom uniformnog k i izračunavanjem $w := G(k)$ (tj. w je pseudoslučajni niz). Konstruiramo D tako da oponaša eksperiment prisluškivanja za $\mathcal{A}$ i promatra hoće li $\mathcal{A}$ uspjeti ili ne. Ako $\mathcal{A}$ uspije, onda D pogađa da w mora biti pseudoslučajni niz, dok ako A ne uspije onda D pogađa da je w slučajni niz. Odnosno, to bismo mogli zapisati na sljedeći način:

Diferencijator D**:** Za ulaz je dan niz $w \in \{0,1\}^{l(n)}$. Pretpostavljamo da se n može odrediti iz $l(n)$ sljedećim postupkom:

1. Pokrenuti $\mathcal{A}(1^n)$ da bismo dobili par poruka $m_0, m_1 \in \{0,1\}^{l(n)}$.
2. Odabrati uniformni bit $b \in \{0,1\}$. Postaviti $c := w \oplus m_b$.
3. Dati $\mathcal{A}$-u c i dobiti izlaz b'. Izlaz je 1 ako je $b' = b$, 0 inače.

Kako je pretpostavka da $\mathcal{A}$ radi u polinomijalnom vremenu, onda očito i D radi u polinomijalnom vremenu. Dodatno, definiramo modificiranu shemu enkripcije $\widetilde{\Pi} = (\widetilde{\mathsf{Gen}}, \widetilde{\mathsf{Enc}}, \widetilde{\mathsf{Dec}})$ koja je upravo shema enkripcije s jednokratnim ključem, no sada uključujemo i sigurnosni parametar koji određuje duljinu poruka koju treba šifrirati. Odnosno, $\widetilde{\mathsf{Gen}}(1^n)$ daje uniformni ključ k duljine $l(n)$ i enkripciju $c := k \oplus m$ poruke

$m \in 2^{l(n)}$ pomoću ključa $k \in \{0,1\}^{l(n)}$. Savršena tajnost sheme enkripcije s jednokratnim ključem implicira:

$$P(\mathsf{Priv}K^{\mathsf{eav}}_{\mathcal{A},\widetilde{\Pi}}(n) = 1) = \frac{1}{2}. \tag{3}$$

Sada primjećujemo dvije važne stvari:

1. Ako je w odabran uniformno iz $\{0,1\}^{l(n)}$, onda je rezultat od $\mathcal{A}$ kada ga D pokreće kao potprogram ista slici od $\mathcal{A}$ u eksperimentu $\mathsf{Priv}K^{\mathsf{eav}}_{\mathcal{A},\widetilde{\Pi}}(n)$. To slijedi jer kada $D(w)$ pokreće $\mathcal{A}$ kao potprogram, $\mathcal{A}$ dobiva šifrirani tekst $c = w \oplus m_b$ gdje je $w \in \{0,1\}^{l(n)}$ uniforman. Budući da D daje 1 baš kada $\mathcal{A}$ uspije u svom eksperimentu prisluškivanja, vrijedi

$$P_{w \leftarrow \{0,1\}^{l(n)}}(D(w) = 1) = P(\mathsf{Priv}K^{\mathsf{eav}}_{\mathcal{A},\widetilde{\Pi}}(n) = 1) = \frac{1}{2}. \tag{4}$$

2. Kada bi se w generirao odabirom uniformnog $k \in \{0,1\}^n$ i potom postavljanjem $w := G(k)$, rezultat od $\mathcal{A}$ pokretanjem od strane D kao potprograma distrubuirala bi se identično kao rezultat od $\mathcal{A}$ iz eksperimenta $\mathsf{Priv}K^{\mathsf{eav}}_{\mathcal{A},\widetilde{\Pi}}(n)$. To slijedi iz činjenice da $\mathcal{A}$, kada ga D pokreće kao potprogram, dobiva šifrirani tekst $c = w \oplus m_b$ gdje je $w = G(k)$ za uniformni $k \in \{0,1\}^n$. Pišemo

$$P_{k \leftarrow \{0,1\}^n}(D(G(k)) = 1) = P(\mathsf{Priv}K^{\mathsf{eav}}_{\mathcal{A},\Pi}(n) = 1). \tag{5}$$

Budući da je G pseudoslučajni generator i da D radi u polinomijalnom vremenu, znamo da postoji zanemariva funkcija negl takva da

$$|P_{w \leftarrow \{0,1\}^{l(n)}}(D(w) = 1) - P_{k \leftarrow \{0,1\}^n}(D(G(k)) = 1)| \leq \mathsf{negl}(n).$$

Iz (4) i (5) slijedi

$$\left|\frac{1}{2} - P(\mathsf{Priv}K^{\mathsf{eav}}_{\mathcal{A},\Pi}(n) = 1)\right| \leq \mathsf{negl}(n),$$

a što implicira

$$P(\mathsf{Priv}K^{\mathsf{eav}}_{\mathcal{A},\Pi}(n) = 1) \leq \frac{1}{2} + \mathsf{negl}(n).$$

Budući da je $\mathcal{A}$ bio proizvoljni PPT protivnik, dokazali smo da Π ima nerazlučive enkripcije u prisutnosti prisluškivača. □

4.4.2 Rasprava

Može se činiti da nismo dobili ništa više nego nam shema enkripcije s jednokratnim ključem nudi s obzirom da ona također enkriptira l-bitnu poruku tako što je provodi kroz XOR s l-bitnim nizom. Ipak, ova konstrukcija nam dopušta da l-bitni niz $G(k)$

može biti mnogo duži od dijeljenog ključa k. Konkretno, korištenjem gornje sheme moguće je sigurno šifrirati datoteku od 1 MB koristeći samo 128-bitni ključ. Oslanjajući se na računsku tajnost, tako smo zaobišli rezultat nemogućnosti teorema koji kaže da svaka savršeno tajna shema šifriranja mora koristiti ključ čija je duljina barem jednake duljini poruke.

Ne dokazujemo bezuvjetno da je konstrukcija 4.1 sigurna. Umjesto toga, dokazujemo da je sigurna pod pretpostavkom da je G pseudoslučajni generator. Ovaj pristup reduciranja sigurnosti konstrukcije više razine na postupak niže razine ima niz prednosti. Jedna od tih prednosti je što je općenito lakše dizajnirati postupak niže razine nego postupak više razine. Također je lakše izravno analizirati algoritam G s obzirom na definiciju niže razine nego analizirati složeniju shemu Π s obzirom na definiciju više razine. To ne znači da je konstruiranje pseudoslučajnog generatora jednostavno - znači samo da je lakše nego konstruirati shemu enkripcije od nule. *(U ovom slučaju shema enkripcije ne radi ništa osim XOR izlaza pseudoslučajnog generatora s porukom, tako da to zapravo nije točno. Međutim, kod složenijih konstrukcija mogućnost svođenja zadatka na jednostavniji je od velike važnosti.)* Još jedna od prednosti je da se jednom konstruirani odgovarajući G može koristiti kao komponenta u raznim drugim shemama.

4.4.3 Konkretna sigurnost

Teorem 4.1, kao i njegov dokaz, konstruirali smo u asimptotskom okruženju. Lako možemo prilagoditi dokaz tako da veže konkretnu sigurnost sheme enkripcije u smislu konkretne sigurnosti od G. U nastavku ćemo smatrati da nismo u asimptotskom okruženju, odnosno smatrat ćemo da radimo s konkretnim brojevima. S Π označimo konstrukciju 4.1 i fiksirajmo neku vrijednost od n. Pretpostavimo da je G za danu vrijednost n (t, ϵ)-pseudoslučajan tako da za svaki diferencijator D koji radi u vremenu najviše t vrijedi

$$|P(D(r) = 1) - P(D(G(s)) = 1)| \leq \epsilon. \tag{6}$$

Možemo uzeti da je $t \approx 2^{80}$ i $\epsilon \approx 2^{-60}$, iako nam točne vrijednosti nisu od važnosti za ovu diskusiju. Tvrdimo da je Π (t, ϵ)-sigurna za neku (malu) konstantu c, pri čemu smatramo da za sve $\mathcal{A}$ koji rade najviše u vremenu $t - c$ vrijedi

$$P(\mathsf{Priv}K^{\mathsf{eav}}_{\mathcal{A},\Pi}(n) = 1) \leq \frac{1}{2} + \epsilon. \tag{7}$$

Da bismo to pokazali, pretpostavimo da je $\mathcal{A}$ neki proizvoljni protivnik koji radi u vremenu najviše $t - c$. Diferencijator D, kako je konstruirano u dokazu teorema 4.1, ima vrlo malo dodatnih zadaća osim pokretanja $\mathcal{A}$. Odabir c na odgovarajući način osigurava da D radi u vremenu najviše t. Iz pretpostavke o konkretnoj sigurnosti od

G slijedi nejednakost (6), a postupajući točno kao u dokazu teorema 4.1, dobivamo nejednakost (7).

5 Jače sigurnosne ideje

Definicija sigurnosti koju smo promatrali u prethodnim poglavljima je relativno slaba definicija jer smo uzimali u obzir samo protivnika koji pasivno prisluškuje jedan enkriptirani tekst poslan između poštenih strana. Sada ćemo promatrati dva jača pojma sigurnosti pri čemu ćemo kod jednog modificirati sigurnosni cilj, a kod drugog ojačati model napada.

5.1 Sigurnost za višestruke enkripcije

Definiciju 3.4. bilo bi praktično proširiti na slučaj kada poštene strane u komunikaciji šalju više šifriranih tekstova pri čemu su svi generirani pomoću istog ključa. Za to nam je najprije potrebna shema enkripcije koja je sigurna za enkripciju više poruka. Započet ćemo s odgovarajućom definicijom sigurnosti za ovaj slučaj te, kao u slučaju definicije 3.4., prvo uvodimo odgovarajući eksperiment definiran za bilo koju shemu šifriranja Π, protivnika $\mathcal{A}$ i sigurnosni parametar n:

Eksperiment prisluškivanja više poruka $\mathsf{Priv}K^{\mathsf{mult}}_{\mathcal{A},\Pi}(n)$:

1. Protivnik $\mathcal{A}$ dobiva 1^n kao ulaz, i za izlaz daje par jednako dugih lista poruka $\overrightarrow{M}_0 = (m_{0,1}, ..., m_{0,t})$ i $\overrightarrow{M}_1 = (m_{1,1}, ..., m_{1,t})$ pri čemu vrijedi $|m_{0,i}| = |m_{1,i}|$ za svaki i.
2. Ključ k generira se pomoću algoritma $\mathsf{Gen}(1^n)$ i odabire se uniformni bit $b \in \{0, 1\}$. Za svaki i izračunava se šifrirani tekst $c_i \leftarrow \mathsf{Enc}_k(m_{b,i})$ i lista $\overrightarrow{C} = (c_1, ..., c_t)$ se daje protivniku $\mathcal{A}$.
3. $\mathcal{A}$ vraća bit b'.
4. Rezultat eksperimenta će biti 1 ako vrijedi da je $b' = b$, a 0 inače.

Definicija sigurnosti je ista kao i prije, osim što se sada odnosi na gornji eksperiment.

Definicija 5.1. *([2]) Kažemo da shema enkripcije s tajnim ključem* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *ima nerazlučive višestruke enkripcije u prisutnosti prisluškivača ako za sve randomizirane protivnike s polinomijalnim vremenom* $\mathcal{A}$ *postoji zanemariva funkcija negl takva da za svaki n vrijedi*

$$P(\mathsf{Priv}K^{\mathsf{mult}}_{\mathcal{A},\Pi}(n) = 1) \leq \frac{1}{2} + \mathsf{negl}(n),$$

pri čemu se vjerojatnost uzima iz slučajnosti koju koristi $\mathcal{A}$ *i slučajnosti korištene u eksperimentu.*

Svaka shema koja ima nerazlučive višestruke enkripcije u prisutnosti prisluškivača također zadovoljava definiciju 3.4, budući da eksperiment $\mathsf{PrivK}^{\mathsf{eav}}$ odgovara posebnom slučaju $\mathsf{PrivK}^{\mathsf{mult}}$ gdje protivnik daje dvije liste koji sadrže samo jednu poruku. Zapravo, nova definicija je strogo jača od definicije 3.4, kao što pokazuje iduća propozicija.

Propozicija 5.1. *Postoji shema enkripcije s privatnim ključem koja ima nerazlučive enkripcije u prisutnosti prisluškivača, ali ne i nerazlučive višestruke enkripcije u prisutnosti prisluškivača.*

Dokaz. Primjer sheme koja zadovoljava propoziciju je shema enkripcije s jednokratnim ključem (jednokratni blok). Jednokratni blok je savršeno tajan, pa tako ima i nerazlučive enkripcije u prisutnosti prisluškivača. Pokazat ćemo da nije siguran u smislu definicije 5.1.

Uzimamo u obzir sljedećeg protivnika $\mathcal{A}$ koji napada shemu (u smislu definiranom eksperimentom $\mathsf{PrivK}^{\mathsf{mult}}$): $\mathcal{A}$ daje $\overrightarrow{M}_0 = (0^l, 0^l)$ i $\overrightarrow{M}_1 = (0^l, 1^l)$. Dakle, prvi sadrži isti otvoreni tekst dvaput, dok drugi sadrži dvije različite poruke. Neka je $\overrightarrow{C} = (c_1, c_2)$ lista enkriptiranih tekstova koje $\mathcal{A}$ prima. Ako je $c_1 = c_2$, tada $\mathcal{A}$ daje $b' = 0$, inače $\mathcal{A}$ daje $b' = 1$.

Sada analiziramo vjerojatnost da je $b' = b$. Ono što je ključno jest da je jednokratni blok deterministički, pa šifriranje iste poruke dvaput (s istim ključem) daje isti šifrirani tekst. Dakle, ako je $b = 0$ onda moramo imati $c_1 = c_2$ i $\mathcal{A}$ mora dati 0. S druge strane, ako je $b = 1$ onda se svaki put šifrira druga poruka, pa vrijedi $c_1 \neq c_2$ i $\mathcal{A}$ daje 1. Zaključujemo da $\mathcal{A}$ ispravno daje $b' = b$ s vjerojatnošću 1, pa shema šifriranja nije sigurna u prema definiciji 5.1.. □

Nužnost vjerojatnosnog šifriranja. Iz prethodnog dokaza možemo pomisliti da je definiciju 5.1. nemoguće postići korištenjem bilo koje sheme šifriranja. No, to je točno samo ako je shema šifriranja deterministička kada šifriranje iste poruke više puta (koristeći isti ključ) uvijek daje isti rezultat.

Teorem 5.1. *([2]) Ako je* Π *shema šifriranja (bez stanja) u kojoj je* Enc *deterministička funkcija ključa i poruke, tada* Π *ne može imati nerazlučive višestruke enkripcije u prisutnosti prisluškivača.*

To ne treba shvatiti kao da je definicija 5.1. prejaka. Propuštanje činjenice da su dvije šifrirane poruke iste do prisluškivača može značajno narušiti sigurnost. Da bismo konstruirali shemu sigurnu za enkripciju više poruka, moramo dizajnirati shemu u kojoj je enkripcija slučajna tako da se mogu proizvesti različiti šifrirani tekstovi kada se ista poruka šifrira više puta.

5.2 Napadi na odabrani otvoreni tekst

Napadi odabranog otvorenog teksta obuhvaćaju sposobnost protivnika da vrši (djelomičnu) kontrolu nad onim što poštene strane šifriraju. Promatramo slučaj gdje dvije poštene strane dijele ključ k, a protivnik može utjecati na njih da šifriraju poruke $m_1, m_2, ...$ (koristeći k) i šalju dobivene šifrirane tekstove preko kanala koji protivnik može promatrati. Kasnije, protivnik promatra šifrirani tekst koji odgovara nekoj nepoznatoj poruci m šifriranoj pomoću istog ključa k; pretpostavimo da protivnik zna da je m jedna od dvije mogućnosti m_0, m_1. Sigurnost od napada odabranog otvorenog teksta znači da čak i u ovom slučaju napadač ne može reći koja je od ove dvije poruke šifrirana s vjerojatnošću znatno boljom od slučajnog pogađanja.

Primjer 5.1. *([2]) Tijekom Drugog svjetskog rata Britanci su postavljali mine na određena mjesta, znajući da će Nijemci - kada ih pronađu - šifrirati lokacije i poslati ih natrag u stožer. Ove šifrirane poruke koristili su kriptoanalitičari u Bletchley Parku kako bi razbili njemačku shemu šifriranja.*

Primjer 5.2. *([2]) U svibnju 1942. kriptoanalitičari američke mornarice presreli su šifriranu poruku Japanaca koju su uspjeli djelomično dekodirati. Rezultat je pokazao da Japanci planiraju napad na AF, gdje je AF bio fragment šifriranog teksta koji američki kriptoanalitičari nisu mogli dekodirati. Iz drugih razloga, vjerovali su da je cilj bio otok Midway. Nažalost, njihovi pokušaji da uvjere Washington u svoja vjerovanja bili su bezuspješni; opće je vjerovanje bilo da Midway nikako ne može biti meta. Kriptoanalitičari mornarice osmislili su sljedeći plan: naložili su američkim snagama u Midwayu da pošalju lažnu poruku da su njihove zalihe slatke vode niske. Japanci su presreli ovu poruku i odmah izvijestili nadređene da je "AF ima nisku razinu vode". To je američkim kriptoanalitičarima bio dokaz da AF odgovara Midwayu te su iz Washingtona poslali tri nosača zrakoplova na to mjesto. Rezultat je bio da je Midway spašen, a Japanci su pretrpjeli značajne gubitke. Ova bitka bila je prekretnica u ratu između SAD-a i Japana na Pacifiku. Kriptoanalitičari mornarice ovdje su izveli napad odabranog otvorenog teksta, jer su uspjeli utjecati na Japance (iako na zaobilazni način) da šifriraju riječ "Midway". Da je japanska shema enkripcije bila sigurna od napada odabranog otvorenog teksta, ova strategija američkih kriptoanalitičara ne bi uspjela (a povijest bi se možda odigrala sasvim drugačije).*

5.3 CPA-sigurnost

U formalnoj definiciji modeliramo napade odabranog otvorenog teksta dajući protivniku $\mathcal{A}$ pristup izvoru (eng. *oracle*), koji se promatra kao "crna kutija" koja šifrira

poruke koje mu $\mathcal{A}$ da, koristeći ključ k nepoznat $\mathcal{A}$-u. Odnosno, pretpostavljamo da $\mathcal{A}$ ima pristup izvoru $\mathsf{Enc}_k(\cdot)$. $\mathcal{A}$ šalje upit izvoru na način da mu daje poruku m kao ulaz, a izvor vraća šifrirani tekst $c \leftarrow \mathsf{Enc}_k(m)$ kao odgovor. (Kada je Enc randomiziran, izvor koristi novu slučajnost svaki put kada odgovori na upit.) Protivniku je dopuštena adaptivna interakcija s izvorom, koliko god puta želi.

Razmotrimo sljedeći eksperiment definiran za bilo koju shemu šifriranja $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, protivnika $\mathcal{A}$ i sigurnosni parametar n:

CPA eksperiment nerazlučivosti $\mathsf{Priv}K^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n)$:

1. Ključ k se generira pokretanjem $\mathsf{Gen}(1^n)$.
2. Protivniku $\mathcal{A}$ se daje ulaz 1^n i pristup izvoru $\mathsf{Enc}_k(\cdot)$, te izlazi par poruka m_0, m_1 iste duljine.
3. Odabire se uniformni bit $b \in \{0,1\}$, a zatim se izračunava šifrirani tekst $c \leftarrow \mathsf{Enc}_k(m_b)$ i daje $\mathcal{A}$.
4. Protivnik $\mathcal{A}$ nastavlja imati pristup izvoru $\mathsf{Enc}_k(\cdot)$, i daje bit b'.
5. Rezultat eksperimenta je definiran kao 1 ako je $b' = b$, a 0 inače. U prvom slučaju kažemo da $\mathcal{A}$ uspijeva.

Definicija 5.2. *([2]) Kažemo da shema enkripcije s tajnim ključem $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ ima nerazlučive enkripcije pri napadu odabranim otvorenim tesktom, ili da je CPA-sigurna, ako za sve slučajno odabrane protivnike koji rade u polinomijalnom vremenu $\mathcal{A}$ postoji zanemariva funkcija negl takva da za svaki n vrijedi*

$$P(\mathsf{Priv}K^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) = 1) \leq \frac{1}{2} + \mathsf{negl}(n),$$

pri čemu se vjerojatnost uzima iz slučajnosti koju koristi $\mathcal{A}$ i slučajnosti korištene u eksperimentu.

5.3.1 CPA-sigurnost za višestruke enkripcije

Definiciju 5.2 možemo proširiti na slučaj višestrukih enkripcija na sličan način kao što smo to radili u slučaju definicije 3.4., koristeći liste otvorenih tekstova. Ipak, ovdje koristimo drugačiji pristup koji je nešto jednostavniji i ima prednost modeliranja napadača koji mogu adaptivno odabrati otvorene tekstove za šifriranje, čak i nakon promatranja prethodnih šifriranih tekstova. Napadaču dajemo pristup "lijevom" ili "desnom" izvoru $\mathsf{LR}_{k,b}$ koji na ulazu para poruka jednake duljine m_0, m_1, izračunava šifrirani tekst $c \leftarrow \mathsf{Enc}_k(m_b)$ i vraća c. Odnosno, ako je $b = 0$, onda protivnik prima

šifriranje "lijevog" otvorenog teksta, a ako je $b = 1$ tada prima šifriranje "desnog" otvorenog teksta. Ovdje je b nasumično odabran bit na početku eksperimenta, a kao i u prethodnim definicijama, cilj napadača je pogoditi b. Ovo generalizira prethodnu definiciju sigurnosti enkripcije više poruka (definicija 5.1.) jer umjesto izlazne liste $(m_{0,1}, ..., m_{0,t})$ i $(m_{1,1}, ..., m_{1,t})$, od kojih će jedna biti šifrirana, napadač sada može uzastopno tražiti $\mathsf{LR}_{k,b}(m_{0,1}, m_{1,1}), ..., \mathsf{LR}_{k,b}(m_{0,t}, m_{1,t})$. Ovo također uključuje napadačev pristup izvoru, budući da napadač može jednostavno zatražiti $\mathsf{LR}_{k,b}(m, m)$ kako bi dobio $\mathsf{Enc}_k(m)$. Sada formalno definiramo ovaj eksperiment, nazvan eksperiment LR-izvor. Neka je Π shema enkripcije, $\mathcal{A}$ protivnik, a n sigurnosni parametar:

LR-izvor eksperiment $\mathsf{Priv}K^{\mathsf{LR\text{-}cpa}}_{\mathcal{A},\Pi}(n)$:

1. Ključ k se generira pokretanjem $\mathsf{Gen}(1^n)$.
2. Odabire se uniformni bit $b \in \{0, 1\}$.
3. Protivniku $\mathcal{A}$ se daje ulaz 1^n i pristup izvoru $\mathsf{LR}_{k,b}(\cdot, \cdot)$.
4. Protivnik $\mathcal{A}$ daje bit b'.
5. Rezultat eksperimenta je definiran kao 1 ako je $b' = b$, a 0 u suprotnom. U prvom slučaju kažemo da $\mathcal{A}$ uspijeva.

Definicija 5.3. *([2]) Kažemo da shema enkripcije s tajnim ključem Π ima nerazlučive višestruke enkripcije pri napadu otvorenim tesktom, ili da je CPA-sigurna za višestruke enkripcije, ako za sve randomizirane protivnike s polinomijalnim vremenom $\mathcal{A}$ postoji zanemariva funkcija negl takva da za svaki n vrijedi*

$$P(\mathit{PrivK}^{\mathit{LR\text{-}cpa}}_{\mathcal{A},\Pi}(n) = 1) \leq \frac{1}{2} + \mathit{negl}(n),$$

pri čemu se vjerojatnost uzima iz slučajnosti koju koristi $\mathcal{A}$ i slučajnosti korištene u eksperimentu.

Naša ranija rasprava pokazuje da je CPA-sigurnost za više enkripcija barem jednako jaka kao i sve naše prethodne definicije. Konkretno, ako je shema šifriranja s privatnim ključem CPA-sigurna za višestruke enkripcije, onda je očito i CPA sigurna. Važno je da vrijedi i obrnuto: CPA-sigurnost podrazumijeva CPA-sigurnost za višestruke enkripcije.

Teorem 5.2. *Svaka shema šifriranja privatnog ključa koja je CPA-sigurna također je CPA-sigurna za višestruko šifriranje.*

Ovo je značajna tehnička prednost CPA-sigurnosti. Dakle, dovoljno je dokazati da je shema CPA-sigurna (za jednu enkripciju), a zatim slijedi da je CPA-sigurna i

za višestruke enkripcije. Sigurnost od napada na odabrani otvoreni tekst danas je minimalni pojam sigurnosti koji bi shema enkripcije trebala zadovoljiti, iako se sve češće zahtijevaju i jača sigurnosna svojstva.

Literatura

[1] A. Dujella, M. Maretić, *Kriptografija*, Element, Zagreb, 2007.

[2] J. Katz, Z. Lindell, *Introduction to Modern Cryptography*, Second edition, Taylor Francis Group, Boca Raton, 2015.

[3] M. Mitzenmacher, E. Upfal, *Probability and Computing - Randomized Algorithms and Probabilistic Analysis*, Cambridge University Press, Cambridge, 2005.

Sažetak

U ovom radu započinjemo proučavanje moderne kriptografije uvođenjem pojma računalne sigurnosti. Zatim definiramo računalno sigurnu enkripciju te konstruiramo sigurne sheme enkripcije s tajnim ključem uvođenjem pojma protočnih šifri i pojma pseudoslučajnosti koji obuhvaća ideju da nešto može "izgledati" potpuno slučajno iako to nije. Ovaj moćni koncept leži u osnovi većeg dijela moderne kriptografije, a ima primjene i implikacije i u mnogim drugim područjima. Za kraj, proučavamo neke jače sigurnosne ideje koje obuhvaćaju napade odabranog otvorenog teksta i CPA-sigurnost te pokazujemo prednosti njihovog korištenja.

Ključne riječi: kriptografija, računalna sigurnost, sigurne sheme enkripcije s tajnim ključem, protočne šifre, pseudoslučajni generatori, nerazlučive enkripcije, cpa-sigurnost, višestruke enkripcije, prisluškivač

Summary

In this thesis we begin our study of modern cryptography by introducing the concept of computational security. We then define computationally secure encryption and construct secure private-key encryption schemes by introducing the notion of steam ciphers and the notion of pseudorandomness that encompasses the idea that something can "look" completely random even though it is not. This powerful concept underlies much of modern cryptography and has applications as well as implications in many other areas. Finally, we introduce some stronger security ideas that include attacks on selected plaintext and CPA-security and show the benefits of using them.

Key words: cryptography, computational security, secure private-key encryption schemes, stream ciphers, pseudorandom generators, indistinguishable encryption, cpa-security, multiple encryption, eavesdropper

Životopis

Rođena sam 22. kolovoza 1994. godine u Osijeku gdje sam i završila Osnovnu školu Ivana Filipovića. 2009. godine upisujem III. gimnaziju u Osijeku, prirodoslovno matematički smjer. Po završetku srednjoškolskog obrazovanja upisujem Sveučilišni preddiplomski studij matematike na Odjelu za matematiku u Osijeku. Tijekom preddiplomskog studija aktivno sudjelujem u radu udruge za međunarodnu razmjenu studenata IAESTE te odlazim na međunarodne konferencije. 2019. godine ostvarujem svoju prvu studentsku mobilnost odlaskom na stručnu praksu na Tehničko sveučilište u Krakovu gdje radim na proučavanju teorije igara i primjeni Pareto-Nash-Stackelberg igara u sigurnosnim sustavima. Preddiplomski studij završila sam 2020. godine s temom završnog rada *Faktorizacija u* $\mathbb{Z}[x]$ pod mentorstvom prof. dr. sc. Ivana Matića. Iste godine upisujem diplomski studij na Odjelu za matematiku, smjer Matematika i računarstvo. Na drugoj godini diplomskog studija odlazim na Erasmus+ praksu u Madrid te u kompaniji Tribalyte Technologies SL radim kao full-stack developer. Po povratku sa stručne prakse nastavljam raditi kao student frontend developer u zagrebačkoj kompaniji minus5.