Odabrani algoritmi kriptografije

Savanović, Ivan

Sveučilište u Rijeci – Odjel za informatiku

Jednopredmetna informatika

Ivan Savanović

ODABRANI ALGORITMI KRIPTOGRAFIJE

Završni rad

Mentor: doc. dr. sc. Marija Brkić Bakarić

Rijeka, 2019

SADRŽAJ

1. UVOD

Kriptografija je jedna od najbitnijih tehnika koje se koriste za osiguravanje podataka ili sigurnu komunikaciju. Iako je kriptografija neophodna za sigurnu komunikaciju ona samo po sebi nije dovoljna. U ovom radu je dan teorijski uvod te opći opis algoritama kriptografije. Kako bi se bolje objasnilo prikazano područje, uz teorijske osnove će biti prikazana implementacija i praktična primjenu odabranog algoritma. U sljedećem poglavlju dane su definicije i pojmovi vezani uz kriptografiju poput kriptografskih primitiva i njihove uloge u izradi novih kriptografskih sustava, vektora inicijalizacije te njihove upotrebe u kriptografskim algoritmima i ciljeva kriptografije. U okviru trećeg poglavlja prikazana je kratka povijest kriptografije, a u četvrtom poglavlju dana je podjela moderne kriptografije, tj. kriptografija sa simetričnim ključem i kriptografija javnog ključa. U petom poglavlju definiran je kriptografski ključ i mehanizmi vezani uz razmjenu ključeva. U šestom poglavlju prikazan je ilustrativan primjer i dana njegova implementacija. U posljednjem poglavlju dan je kratki zaključak.

2. KRIPTOGRAFIJA

Kriptografija je znanost koja se bavi proučavanjem matematičkih tehnika koje se odnose na aspekte informacijske sigurnosti [1]. Općenitije kriptografija se odnosi na konstruiranje i analizu protokola koji sprečavaju javnost ili neku treću stranu od čitanja ili izmjene privatne poruke. Dok je kriptografija znanost koja se bavi osiguravanjem podataka, kriptoanaliza je znanost koja se bavi proučavanjem matematičkih tehnika za probijanje kriptografskih sustava [1], odnosno metoda za čitanje tajnih poruka bez ključa. Moderna kriptografija uključuje discipline matematike, informatike, elektrotehnike, komunikologije i fizike. Neke od primjena kriptografije uključuju elektroničko trgovanje, platne kartice bazirane na čipu, digitalne valute, računalne lozinke, vojnu komunikaciju itd.

Pri izradi kriptografskih protokola koriste se kriptografski algoritmi niske razine koji se zovu kriptografski primitivi koji su objašnjeni u potpoglavlju 2.1. Kriptografski algoritmi su osmišljeni po pretpostavci računalne tvrdoće, odnosno da se takva enkripcija ne može probiti u nekom realnom vremenu. Sustav koji je osmišljen po pretpostavci računalne tvrdoće je samo teoretski moguće probiti, iako u praksi probijanje takvog sustava nije dokazano. Takvi sustavi se nazivaju računski sigurnim sustavima. Pri izvođenju algoritama se koristi fiksna ulazna veličina koja se zove vektor inicijalizacije pomoću koje se za isti ključ može dobiti drugačiji izlaz. Postoje teoretski sigurni algoritmi koje se ne mogu probiti čak s neograničenom računalnom snagom. Primjer takvog algoritma poznat je pod nazivom jednokratna bilježnica (engl. *one-time pad*), ali takvi algoritmi se teško koriste u praksi. Postizanje sigurnosti informacija u modernom dobu zahtjeva širok raspon tehničkih vještina i poznavanje zakonskih regulativa. Svi potrebni uvjeti koji se smatraju potrebnim za informacijsku sigurnost se ne mogu adekvatno ostvariti. Kriptografija nije sredstvo pružanja sigurnosti, nego samo jedna od skupa tehnika.

2.1 KRIPTOGRAFSKI PRIMITIVI

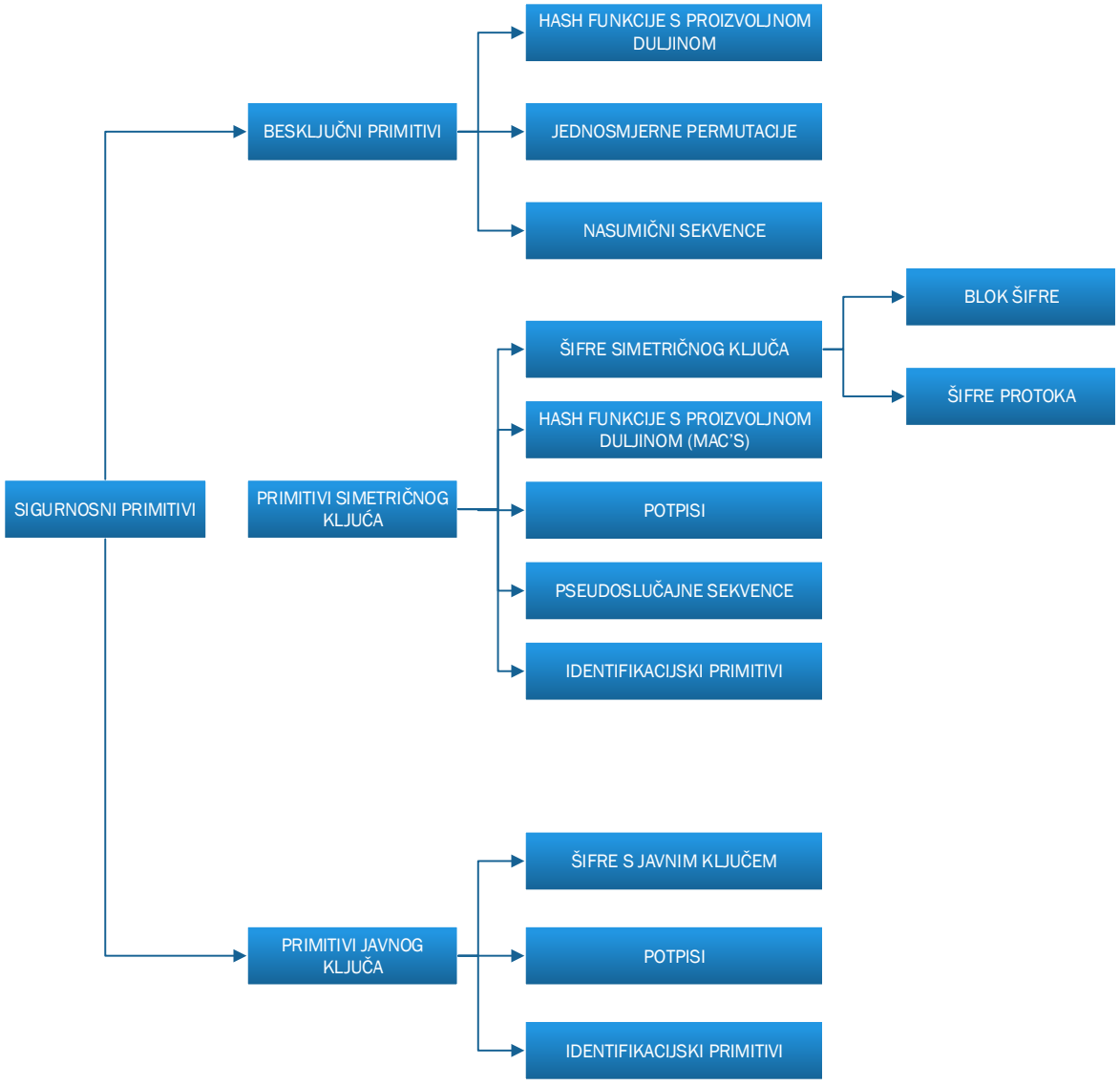
Kriptografski primitivi su utemeljeni kriptografski algoritmi niske razine koji se često koriste za izradu kriptografskih protokola za računalne sustave. Zbog toga su kriptografski primitivi osmišljeni kako bi obavili jedan zadatak na precizno definiran način.

Pošto se kriptografski primitivi koriste kao građevni blokovi oni moraju biti pouzdani odnosno obavljati svoju funkciju prema specifikacijama. Ako primitiv tvrdi da se ne može probiti sa nekim brojem računalnih operacija, a može se probiti sa manjim brojem od navedenog onda se takav kriptografski primitiv smatra neuspješnim. Gotovo svaki protokol koji koristi takav primitiv je ranjiv. Budući da je izrada novih kriptografskih rutina jako teška, sklona pogreškama i testiranje pouzdanosti traje dugo, dizajniranje novih kriptografskih primitiva u biti nikad nije razumno niti sigurno kako bi on odgovarao potrebama novog kriptografskog sustava. Također algoritmi u ovom području ne moraju samo biti dobro dizajnirani već moraju biti testirani od kriptološke zajednice. Uspješni prolazak ispitivanja daje određenu pouzdanost u sigurnost algoritma ali sigurnosni dokaz za kriptografski primitiv općenito nije dostupan. Primjer nekih kripto sustava koji koriste kriptografske primitive su TLS, SSL, SSH, VPN, PGP i drugi. Kriptografski primitivi se vrednuju s obzirom na kriterije poput što su:

- performanse, odnosno djelotvornost primitiva u određenom načinu rada, npr. broj bitova koje algoritam može šifrirati u sekundi,
- jednostavnost primjene - poteškoću primjene primitiva u praksi, a može uključivati složenost implementacije primitiva u programskom ili hardverskom okruženju,
- funkcionalnost – koji će primitivi najučinkovitije obavljati određeni cilj što je određeno osnovnim svojstvima primitiva,
- razina sigurnosti koja se često određuje s brojem operacija koje su potrebne za probijanje – obično je razina sigurnosti određena gornjom granicom utrošenog posla pri probijanju; ponekad se naziva faktor rada

- metoda rada – primitivi obično imaju različite karakteristike, primjenjuju se na različite načine i s različitim ulazima; ovisno o načinu korišenja, jedan primitiv može pružiti vrlo različitu funkcionalnost.

Neki od kriptografskih primitiva su sheme šifriranja, *hash* funkcije, sheme digitalnog potpisa, blok šifre (engl. *block cypher*) i šifre protoka (engl. *stream cypher)*. Njihovi međusobni odnosi su prikazani na slici.



Slika 1 Taksonomija kriptografskih primitiva [1]

2.2 VEKTOR INICIJALIZACIJE

Vektor inicijalizacije je ulaz fiksne veličine za kriptografski primitiv koji obično treba biti slučajan ili pseudo-slučajan. Randomizacija je ključna za sheme šifriranja kako bi postigle semantičku sigurnost tj. da ponovljena upotreba sheme s istim ključem ne dopušta napadaču da zaključi kakvi su odnosi između segmenata šifrirane poruke. Vektor inicijalizacije se koristi kako bi se izbjeglo ponavljanje tokom enkripcije i otežalo pronalaženje uzoraka. Randomizacija je također potrebna za druge primitive kao što su univerzalne *hash* funkcije i kodovi za provjeru autentičnosti poruke. Vektor inicijalizacije se ne treba čuvati u tajnosti pošto u većini slučaja to ne bi bilo praktično budući da ga primatelj treba znati kako bi dešifrirao podatke ili provjerio *hash*. Ako je vektor inicijalizacije tajan onda je on ključ.

Neki kriptografski primitivi zahtijevaju da se vektor inicijalizacije ne ponavlja, a slučajnost da se izvede interno. Takav vektor inicijalizacije se zove nonce (engl. number occuring once). Nonce u najširem smislu se odnosi na vrijednost koja se koristi samo jednom. Jedinstveni vektor inicijalizacije koji se koristi za enkripciju blok šifrom kvalificira se kao nonces. Postoje varijacije u literaturi o tome koji se izraz koristi, inicijalizacijski vektor ili nonce, za različite načine rada blok šifre. Neki autori koriste isključivo jedan ili drugi, dok neki razlikuju te izraze.

Ako je vektor inicijalizacije za shemu dobiven nasumice onda se ona zove nasumična shema, inače se zove statusna shema. Nasumične sheme uvijek zahtijevaju da vektor inicijalizacije odabran od pošiljatelja bude proslijeđen primatelju, dok statusne sheme omogućavaju da pošiljatelj i primatelj dijele zajednički vektor inicijalizacije koji se ažurira na unaprijed definiran način.

2.3 CILJEVI KRIPTOGRAFIJE

Od svih ciljeva kriptografije, iz sljedećih četiri proizlaze ostali, a to su: povjerljivost, integritet podataka, ovjera autentičnosti i neosporavanje. Temeljni cilj kriptografije je adekvatno rješavanje tih četiri područja u teoriji i praksi [1].

- Povjerljivost je usluga koja se koristi za čuvanje sadržaja informacija od svih onih koji nisu ovlašteni da ih imaju. Tajnost je jedan od sinonima koji se koristi za povjerljivost i privatnost. Postoje brojni pristupi osiguravanja povjerljivosti od fizičkih zaštita do matematičkih algoritama koji podatke čine nerazumljivim.
- Integritet podataka je usluga osiguravanja da neće doći do neovlaštene izmjene podataka. Kako bi se osigurao integritet podataka mora se omogućiti otkrivanje manipulacije podacima od neovlaštene strane. U manipulaciju podataka spadaju operacije poput umetanja novih podatka, brisanja i zamjene postojećih podataka.
- Provjera autentičnosti odnosi se na identifikaciju. Ona se primjenjuje na oba entiteta i na samu informaciju. Dvije stranke koje ulaze u komunikaciju trebaju se međusobno identificirati. Za informacije dostavljene preko kanala komunikacije treba biti provjerena autentičnost porijekla, datuma nastanka, sadržaja, vremena slanja itd. Zbog toga je ovaj aspekt kriptografije obično podijeljen u dvije glavne klase: autentifikacija entiteta i autentifikacija podrijetla podataka. Provjera podrijetla podataka implicitno osigurava integritet podataka jer ako je poruka izmijenjena izvor je također promijenjen.
- Neosporavanje je usluga sprječavanja entiteta da poriče prethodne obaveze ili postupke. Ako dođe do spora zbog toga što entitet poriče određene radnje potrebno je riješiti situaciju te se koristi procedura koja uključuje povjerljivi treći entitet za rješavanje spora.

Ostali ciljevi kriptografije su potpis, autorizacija, provjera valjanosti, kontrola pristupa, certifikacija, označavanje vremena, svjedočenje, potvrda o primitku, potvrda, vlasništvo, anonimnost i opoziv.

3. POVIJEST KRIPTOGRAFIJE

Prije moderne ere kriptografija se temeljila na povjerljivosti poruka odnosno enkripciji. Glavna klasična vrsta šifre je transpozicija koja preuređuje redoslijed slova u poruci te supstitucijske šifre koje sustavno zamjenjuju slova ili grupe slova s drugim slovima ili skupovima slova. Jedna od ranih supstitucijskih šifri bila je Cezarova šifra. Najstarija poznata upotreba kriptografije je isklesana šifrirana slika na kamenu u Egiptu oko 1900. pr. Kr. Stenografija je također prvi put razvijena u antičkom dobu. Rani primjer je od Heradota. Šifrirani tekstovi proizvedeni klasičnom šifrom otkrivaju statističke podatke o čistom tekstu, a te se informacije često mogu koristiti za razbijanje šifre.

Nakon otkrivanja frekvencijske analize u 9. stoljeću gotovo sve takve šifre se mogu razbiti od strane informiranog napadača. Frekvencija slova može pomoći pri probijanju šifri za neke povijesne tehnike šifriranja poput homofonskih šifri. Skoro sve šifre su bile podložne kriptoanalizi korištenjem tehnike frekvencijske analize do razvoja polialfabetske šifre. Vigenèrova šifra je jedna od poznatijih polialfabetskih šifri. Sredinom 19. stoljeća Charles Babbage je pokazao da je Vigenèreova šifra osjetljiva na Kasiskijev test. On omogućava kripto analitičaru da zaključi duljinu ključne riječi pomoću traženja nizova znakova koji se ponavljaju. Nizovi moraju biti dugački tri znaka ili više kako bi test bio uspješan. Kodiranje je još uvijek često bilo učinkovito u praksi iako je frekvencijska analiza bila učinkovita za probijanje mnogih šifri. Razbijanje poruke bez uporabe frekvencijske analize zahtijevalo je poznavanje korištene šifre i možda ključa.

Tek u 19. stoljeću izričito je priznato da tajnost algoritama za šifriranje nije razumna niti praktična zaštita za sigurnost poruke. Shvaćeno je da svaka adekvatna kriptografska shema uključujući i šifre treba biti sigurna čak i ako protivnik u potpunosti razumije algoritam šifre. Sigurnost ključa bi trebao biti dovoljan uvjet da se dobra šifra ne može probiti. Taj osnovni princip prvi je put bio eksplicitno naveden 1883. godine od strane Auguste Kerckhoffs i općenito se naziva Kerckhoffs princip.

Različita pomagala i uređaji su korišteni za pomoć kod šifriranja. Jedan od najstarijih je možda bio štap nazvan *skytale* u drevnoj Grčkoj.

Slika 2 Skytale [2]

U srednjem vijeku izumljena su druga pomagala kao što su rešetke za šifru koje su se također koristile za neku vrstu stenografije. Izumom polialfabetskih šifri izumljena su sofisticiranija pomagala poput Albertijevog diska s šiframa (slika 3), Johannes Trithemiusovove tablice i Thomas Jeffersonovog kotača.

Slika 3 Albertijev disk s šiframa [2]

Mnogi mehanički uređaji za šifriranje i dešifriranje izumljeni su početkom 20. stoljeća, a nekoliko je patentiranih, među njima i rotorskih strojeva među kojima je i Enigma (slika 4). Šifre koje su korištene s tim strojevima su dovele do povećanih poteškoća pri kriptoanalizi.

Slika 4 Enigma [2]

Kao što je razvoj digitalnih računala i elektronike pomogao u kriptoanalizi, tako je omogućio i izradu mnogo složenijih šifri. Računala dopuštaju šifriranje bilo koje vrste podataka koji se mogu predstaviti u binarnom formatu, za razliku od klasičnih šifri koje su šifrirale samo tekstove pisanog jezika. Korištenje računala tako je zamijenilo lingvističku kriptografiju kako za šifriranje tako i za kriptoanalizu. Mnoge računalne šifre se mogu okarakterizirati svojim djelovanjem na bitove za razliku od klasičnih šifri i mehaničkih shema koje općenito manipuliraju jezičnim znakovima. Uobičajeno je da je upotreba kvalitetne šifre vrlo učinkovita, odnosno brza, i da zahtijeva malo resursa poput memorije ili CPU-a, dok razbijanje zahtijeva puno više resursa, čineći kriptoanalizu jako neučinkovitom i nepraktičnom da je efikasno nemoguća. Proliferacija računala i komunikacijskih sustava 1960-ih dovela je do zahtjeva od strane privatnih sektora za sredstvima zaštite informacija u digitalnom obliku. DES-a (engl. *Data Encryption Standard*) je najpoznatija blok šifra napravljena 1970-ih koja se i dalje koristi kao standardno sredstvo za osiguravanje elektroničke trgovine.

Whitfield Diffie i Martin E. Hellman su 1976. godine objavili rad Novi smjerovi u kriptografiji (engl. *New Directions in Cryptography*). Ovaj rad predstavio je revolucionarni koncept javnog ključa te je pružio novu metodu za razmjenu ključeva čija se sigurnost temelji na nepostojanju općenite metode rješavanja problema diskretnog logaritma. Iako autori u to vrijeme nisu imali praktičnu realizaciju sheme šifriranja javnog ključa, ideja je bila jasna i generirala je veliki interes i aktivnost u kriptografskoj zajednici. Godine 1978. Rivest, Shamir i Adleman otkrili su prvo praktično šifriranje i potpisivanje s javnim ključevima, koje se sada naziva RSA. Bazirana na problemu faktorizacije cijelih brojeva što je dovelo do povećanog napora u traženju učinkovitijih metoda za analizu čimbenika. Osamdesetih godina prošlog stoljeća došlo je do značajnog napretka u ovom području ali ništa što bi učinilo RSA sustav nesigurnim. Još jednu od praktičnih i jakih shema javnog ključa pronašao je ElGamal 1985. godine, a također se temelji na diskretnom logaritamskom problemu.

Jedan od najznačajnijih doprinosa kriptografije s javnim ključem je digitalni potpis. Godine 1991. donesen je prvi međunarodni standard za digitalne potpise (ISO / IEC 9796). Temelji se na shemi javnog ključa RSA. Postoji vrlo malo kripto sustava za koje je dokazano da su bezuvjetno sigurni. Jednokratna bilježnica (engl. *One-time pad*) je jedan, a to je dokazao Claude Shannon. Jednokratna bilježnica koristi jednokratni ključ koji je barem dug koliko i poruka koja se šifrira. Zatim se svaki znak ili bit šifrira s ključem koristeći modularno zbrajanje. Ako je ključ dobiven nasumično, barem je dug kao poruka, nikad više nije korišten i bio je potpuno tajan tada je nemoguće probiti šifriranu poruku. Postoji nekoliko važnih algoritama koji su dokazani sigurni pod određenim pretpostavkama. Na primjer RSA se smatra sigurnim sustavom pod pretpostavkom da faktorizacija iznimno velikih brojeva nije moguća. Postoje sustavi slični RSA poput onog od Michael O. Rabin koji je dokazivo siguran pod uvjetom da je faktorizacija $n = p \times q$ nemoguća ali u praksi je skoro neiskoristiv. Neprekidna poboljšanja u računalnoj procesnoj snazi povećala su opseg napada brutalnom silom.

4. MODERNA KRIPTOGRAFIJA

Klasična kriptografija manipulira znakovima, odnosno slovima i brojevima. Tehnike koje se koriste za šifriranje su tajne, odnosno poznavanje tehnike je omogućavalo otkrivanje poruke. Također zahtijeva cjelokupni kripto sustav za povjerljivu komunikaciju. Za razliku od klasične kriptografije moderna kriptografija se izvodi na binarnim nizovima te se oslanja na poznate matematičke algoritme za šifriranje podataka. Tajnost se ostvaruje tajnim ključem koji onemogućava neovlaštenoj osobi dobivanje izvorne informacije čak iako zna izvorni algoritam korišten u šifriranju. Strane zainteresirane za sigurnu komunikaciju trebaju posjedovat samo tajni ključ, a ne cijeli kripto sustav. Moderna kriptografija se može podijeliti na: kriptografiju sa simetričnim ključem, kriptografija sa javnim ključem i *hash* funkcije.

4.1 KRIPTOGRAFIJA SA SIMETRIČNIM KLJUČEM

Kriptografija sa simetričnim ključem još se uvijek koristi u modernoj kriptografiji, a odnosi se na metodu enkripcije u kojoj pošiljatelj i primatelj dijele isti ključ. Također može biti da su njihovi ključevi različiti ali povezani na lako izračunljiv način. Najčešći problem ove metode je prijenos ključa. Šifre sa simetričnim ključem su implementirane kao blok šifre ili šifre protoka.

Blok šifra je shema šifriranja koja prekida tekstualne poruke koje se prenose u nizove fiksne duljine, koji se zovu blokovima, te šifrira redom blokove. Standard za šifriranje podataka (engl. *Data Encryption Standard* – DES) i napredni standard za šifriranje (engl. *Advanced Encryption Standard* – AES) su blok metode koje je američka vlada proglasila službenim metodama šifriranja. DES je korištena i javno dostupna od 1976. godine te se danas smatra nesigurnom. Ona je povučena nakon usvajanja AES-a. Unatoč tome ona se naširoko koristi od šifriranja bankomata do e-pošte i sigurnog udaljenog pristupa. Mnoge druge blok šifre različite kvalitete su dizajnirane, ali većina je

probijena poput brzog algoritma za šifriranje podataka (engl. *Fast data Encipherment Algorithm* – FEAL) koji je dizajniran kao alternativa DES-a.

Šifre protoka za razliku od blok šifri stvaraju proizvoljno duge elemente ključa koji je kombiniran s običnim tekstom bit po bit ili znak po znak, poput jednokratne bilježnice. U šifri protoka izlazni tok se kreira na temelju unutarnjeg stanja koje je skriveno i mijenja se kako šifra djeluje. To unutarnje stanje u početku je postavljeno pomoću elemenata tajnog ključa. Jedna od takve šifre je RC4 (Rivest Cipher 4) koja je široko korištena. Blok šifre se mogu koristiti kao šifre protoka.

Prednosti šifri sa simetričnim ključem:

- One mogu imati visoku brzinu prijenosa podataka.
- Imaju relativno kratke ključeve.
- Mogu se koristiti kao kriptografski primitivi za izradu različitih kriptografskih sustava
- Duga povijest.

Nedostaci šifri sa simetričnim ključem:

- Pri komunikaciji dviju strana ključ mora ostati tajan na oba kraja.
- U velikim mrežama postoji puno parova ključeva s kojima se upravlja.
- Dobra praksa je da se ključevi često mijenjaju.
- Dijelovi digitalnog potpisa koji koriste šifriranje sa simetričnim ključem zahtijevaju velike ključeve ili upotrebu pouzdane treće stranke (engl. *trusted third party* – TTP).

4.2 KRIPTOGRAFIJA JAVNOG KLJUČA

Kod javnog ključa (engl. *public-key*), također općenitije nazvanog asimetrični ključ, koriste se dva različita ključa koja su matematički povezana, odnosno javni i privatni ključ. Sustav javnog ključa je konstruiran na takav način da je izračun jednog ključa (privatnog ključa) računski neizvediv od drugog ključa (javnog ključa) iako su oni povezani. Oba ključa se generiraju potajno kao međusobno povezani par. U sustavima s javnim ključem javni ključ se može slobodno distribuirati, dok njegov privatni ključ mora ostati tajan pošto se javni ključ koristi za šifriranje a tajni ključ za dešifriranje. Diffie i Hellman su pokazali Diffie-Hellmanovim protokolom za razmjenu ključeva da je kriptografija javnog ključa moguća iako takav sustav tada nije postojao. To rješenje se danas široko koristi u sigurnim komunikacijama kako bi se dvije strane dogovorile oko zajedničkog ključa za šifriranje potajno. Jedan od prvih kripto sustava javnog ključa je RSA (Rivest – Shamir - Adleman) koji se koristi za sigurno prenošenje podataka. Ostali algoritmi javnog ključa su Cramer-Shoup kripto sustav, ElGamal enkripcija i kriptografija eliptičke krivulje (engl. *Elliptic-Curve Cryptography* – ECC).

Kriptografija s javnim ključem također se koristi za implementaciju digitalnog potpisa. Kod digitalnih potpisa postoje dva algoritma. Jedan za potpisivanje u kojem se tajni ključ koristi za obradu poruke ili *hash* vrijednosti poruke ili oboje, a drugi za provjeru gdje se koristi odgovarajući javni ključ. RSA i DSA (engl. *Digital Signature Algorithm*) su dvije najpopularnije sheme digitalnog potpisa. Algoritmi javnog ključa najčešće se temelje na računskoj složenosti „tvrdih“ problema, često iz teorije brojeva. Zbog poteškoća s osnovnim problemima većina algoritama javnog ključa koriste operacije poput modularnog množenja i potenciranja koje su računski mnogo više zahtjevne nego većina tehnika korištenih u blok šiframa. Zbog toga su kripto sustavi javnog ključa obično hibridni kripto sustavi u kojima se za samu poruku koristi brzi algoritam šifriranja visoke kvalitete, dok se odgovarajući simetrični ključ koji je šifriran pomoću algoritma javnog ključa šalje porukom.

Prednosti kriptografije javnog ključa:

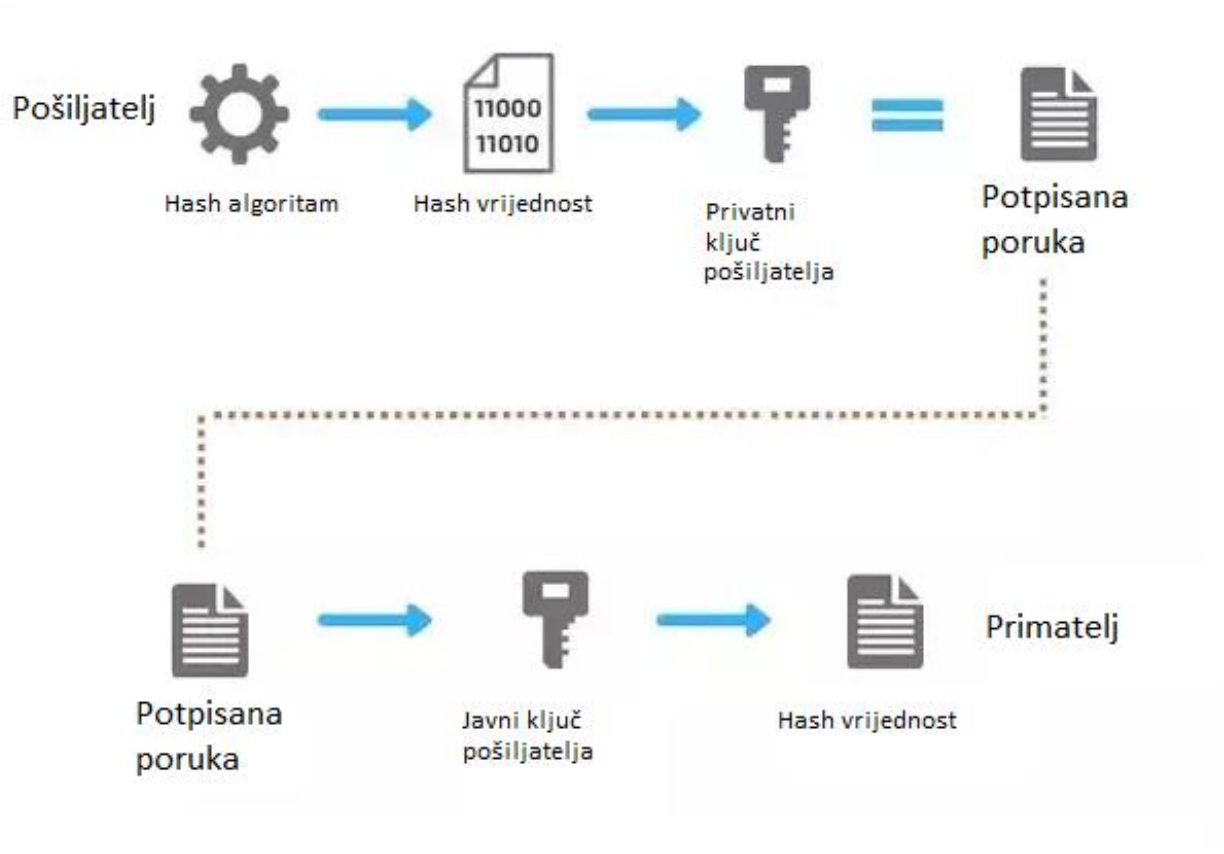
- Samo privatni ključ mora biti tajan.
- Administracija ključeva na mreži zahtjeva prisutnost donekle pouzdanog TTP-a.
- Ovisno o načinu korištenja, par javnog ključa i privatnog ključa mogu ostati nepromijenjeni tijekom dugog vremenskog razdoblja.
- Daje manji ključ od enkripcije sa simetričnim ključem koja se koristi u digitalnim potpisima.
- U velikim mrežama broj potrebnih ključeva je znatno manji nego pri upotrebi simetričnog ključa.

Nedostaci kriptografije javnog ključa:

- Propusnost je puno manja od šifri sa simetričnim ključem.
- Ključevi su puno veći od ključeva kod šifri sa simetričnim ključem.
- Nijedna shema javnog ključa nije dokazana kao sigurna. Sigurnost se temelji na pretpostavci teškoće malog broja teoretskih problema.
- Kratka povijest.

U modernoj kriptografiji jedan od temeljnih primitiva čine *hash* funkcije. *Hash* funkcija je računski učinkovita funkcija mapiranja binarnih nizova proizvoljne duljine u binarne nizove neke fiksne duljine koje se zovu *hash* vrijednost. Za *hash* funkciju koja daje n-bitne *hash* vrijednosti i ima poželjna svojstva vjerojatnost da se nasumično izabrani niz mapira na određenu n-bitnu *hash* vrijednost je 2^{-n}. Ideja je da *hash* vrijednost služi kao kompaktni predstavnik ulaznog niza. *Hash* funkcija koja se koristi za šifriranje je tipično izabrana takva da je računski neizvediv pronalazak dva različita ulaza koji su *hash*irani u zajedničku vrijednost i da je računski neizvedivo pronaći ulaz na temelju *hash* vrijednosti. Najčešća kriptografska upotreba *hash* funkcija je kod digitalnih potpisa i za integritet podataka. Kod digitalnih potpisa (slika 5) obično se duga poruka hashira koristeći javno dostupne *hash* funkcije i potpisuje se samo *hash* vrijednost. Primatelj

poruke zatim hashira primljenu poruku i provjerava dali je primljeni potpis ispravan za primljenu *hash* vrijednost.



Slika 5 Digitalni potpis [9]

Time se štedi prostor i vrijeme u usporedbi s potpisivanjem poruke izravno. *Hash* funkcije mogu se koristiti za integritet podataka. *Hash* vrijednost koja odgovara određenom ulazu izračunava se u nekom trenutku. Integritet ove *hash* vrijednosti je na neki način zaštićen. U sljedećem trenutku kako bi se potvrdio integritet podataka *hash* vrijednost se ponovno izračunava pomoću unosa i uspoređuje s izvornom *hash* vrijednosti. Neke od dugo korištenih *hash* funkcija su MD4, koja je probijena, te poboljšana varijanta MD5 koja također ima svojih slabosti.

5. UPRAVLJANJE KLJUČEVIMA

Ključ u kriptografiji je parametar ili informacija koja određuje funkcionalni izlaz kriptografskog algoritma.

Upravljane ključevima (engl. *key management*) je skup tehnika i postupaka koje se odnose na stvaranje, razmjenu, pohranjivanje, korištenje, brisanje i razmjenu ključeva. Upravljanje ključevima igra ključnu ulogu u kriptografiji. Koristi se za osiguravanje povjerljivosti, autentičnosti entiteta, izvornosti podataka, integriteta podataka i digitalne potpise. Cilj dobrog kriptografskog dizajna je smanjiti složenost problema za pravilno upravljanje ključevima kako bi bilo potrebno čuvati mali broj kriptografskih ključeva koji su na kraju osigurani s hardverom, softverom ili proceduralnim kontrolama. Povezivanje u komunikacijskom okruženju uključuje najmanje dvije strane, pošiljatelja i primatelja, u stvarnom vremenu, dok u okruženju za pohranu podataka može postojati samo jedna strana koja pohranjuje i dohvaća podatke u različitim vremenskim trenutcima.

Upravljanje ključevima je obično u kontekstu određene sigurnosne prakse (engl. *security policy*) koja implicitno ili eksplicitno definira prijetnje za koje je sustav namijenjen. Ona može utjecati na strogost kriptografskih zahtjeva ovisno o podložnosti okoliša na različite vrste napada. One tipično specificiraju:

- Prakse i postupke koji se moraju slijediti u provođenju tehničkih i administrativnih aspekata upravljanja ključevima, bilo ručno ili automatsko.
- Obveze i odgovornosti svake strane.
- Informacije o reviziji koje treba čuvati za naknadne recenzije ili izvješća sigurnosnih događaja.

Razmjena ključeva ili uspostava ključeva je bilo koja metoda u kriptografiji kojom se kriptografski ključevi razmjenjuju između dvije strane, odnosno koja omogućuje dvjema ili više stranaka da uspostave zajednički ključ za šifriranje koji mogu koristiti za šifriranje ili za potpisivanje podataka koje planiraju razmjenjivati. Da bi dvije strane mogle

komunicirati povjerljivo najprije moraju međusobno razmijeniti tajni ključ koji će se koristiti za šifriranje i dešifriranje poruka. Protokoli razmjene ključeva su smišljeni kako bi se riješio problem povjerljivosti prilikom uspostave tajnog ključa između dvije ili više stranaka, a da se pri tome ne dopusti neovlaštenoj strani presretanje, zaključivanje ili na neki drugi način dobivanje ključa.

Uspostava ključeva može se općenito podijeliti na protokol transporta ključa (engl. *key transport*) i protokol dogovora o ključu (engl. *key agreement*), s tim da postoje varijacije na te protokole. Protokol transporta ključa je tehnika uspostave ključa gdje jedna strana pravi ili na drugi način dobiva tajnu vrijednost i sigurno je prenosi drugoj strani ili stranama. Protokol dogovora o ključu je tehnika uspostave ključa gdje se dvije ili više strana mogu dogovoriti o ključu na takav način da obje utječu na ishod, idealno tako da nijedna strana ne može unaprijed odrediti dobivenu vrijednost. Većina protokola imaju za cilj stvaranje različitih ključeva pri svakom izvršenju protokola. U nekim slučajevima početni materijal za ključ unaprijed definira fiksni ključ koji će biti dobiven pri svakom izvršenju protokola od određenog para ili grupe korisnika. Općenito je poželjno da svaka strana u uspostavi ključa može utvrditi identitet druge strane koja bi mogla dobiti rezultirajući ključ.

6. IMPLEMENTACIJA KRIPTOGRAFSKOG ALGORITMA

Program je napisan u jeziku C#. Koristi naivnu enkripciju javnog ključa, odnosno na jednostavan i očit način je šifrirana. Kao ulaz program prima tekst iz datoteke koju potom šifrira te dešifrira. Primjer rada programa dan je na slici 6, slici 7 i slici 8. Program ima jednu klasu koja sadrži sljedeće atribute:

```
private string fileName;
private static int privateKey = 3;
private static int number = 2;
private static int publicKey = privateKey * number;
private List<byte[]> EncryptedLines;
```

- fileName – ime tekstualne datoteke
- privateKey – konstanta koja predstavlja privatni ključ
- number – konstanta pomoću koje se dobije javni ključ
- publicKey – konstanta koja predstavlja javni ključ koji je rezultat množenja nekog broja i privatnog ključa
- EncryptedLines – lista bajtova koja se ispuni sa šifriranim linijama teksta

ToByte je privatna metoda koja pretvara *string* u polje bajtova.

```
private byte[] ToByte (string s)
    {
        UnicodeEncoding ue = new UnicodeEncoding();
        byte[] messageBytes = ue.GetBytes(s);
        return messageBytes;
    }
```

Encrypt je privatna metoda koja prima polje bajtova te ga šifrira pomoću privatnog i javnog ključa

```
        private byte[] Encrypt(byte[] b)
        {
            for (int i = 0; i < b.Length; i++)
            {
                b[i] = (byte)(b[i] + (privateKey * publicKey));
            }

            return b;
        }
```

Decrypt je privatna metoda koja prima polje bajtova te ga dešifrira pomoću privatnog i javnog ključa

```
private byte[] Decrypt(byte[] b)
{
    for (int i = 0; i < b.Length; i++)
    {
        b[i] = (byte)(b[i] - (privateKey * publicKey));
    }

    return b;
}
```

Pri inicijalizaciji programa izvršava se forma *Form1_Load*. Ona ispisuje javni ključ.

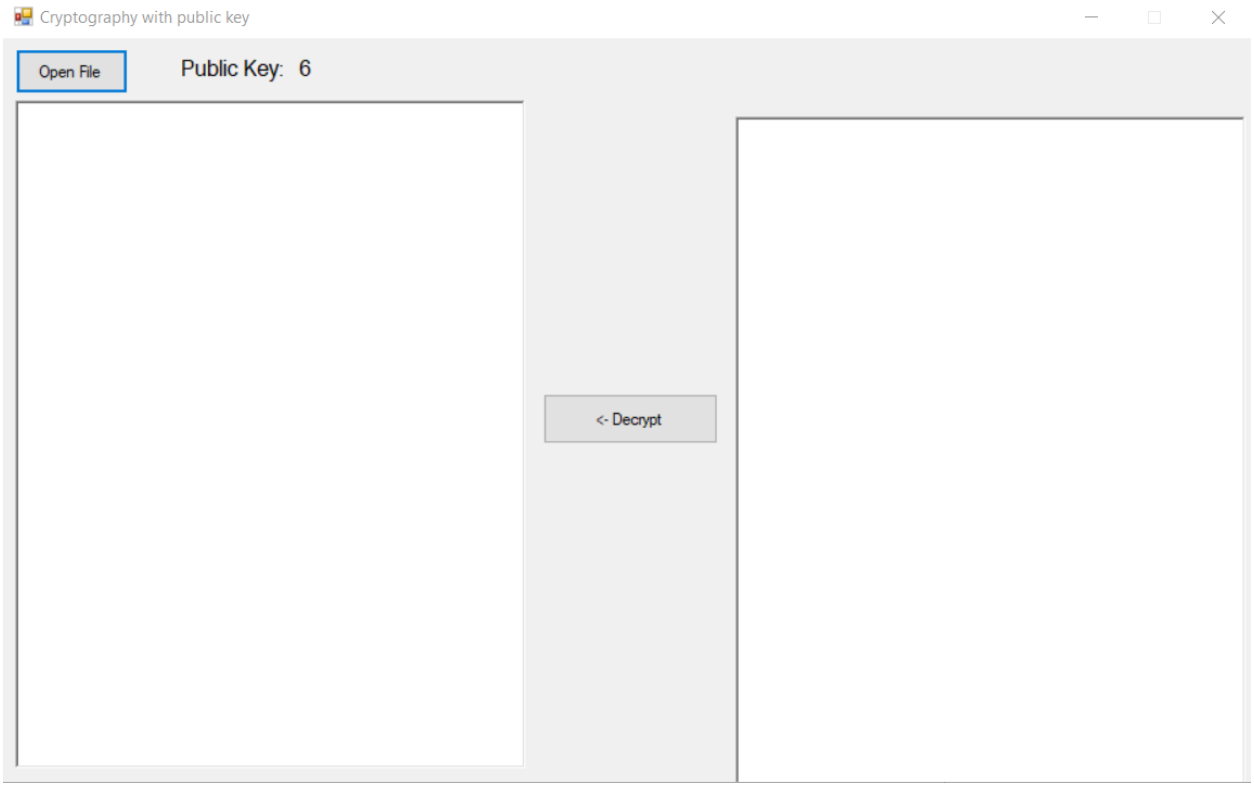
```
private void Form1_Load(object sender, EventArgs e)
{
    label2.Text = publicKey.ToString();
}
```

Kada se pritisne dugme *Open File* pomoću forme *button2_Click* otvara se prozor za odabir datoteke. Nakon odabira datoteke, ispiše se tekst i šifriran tekst.

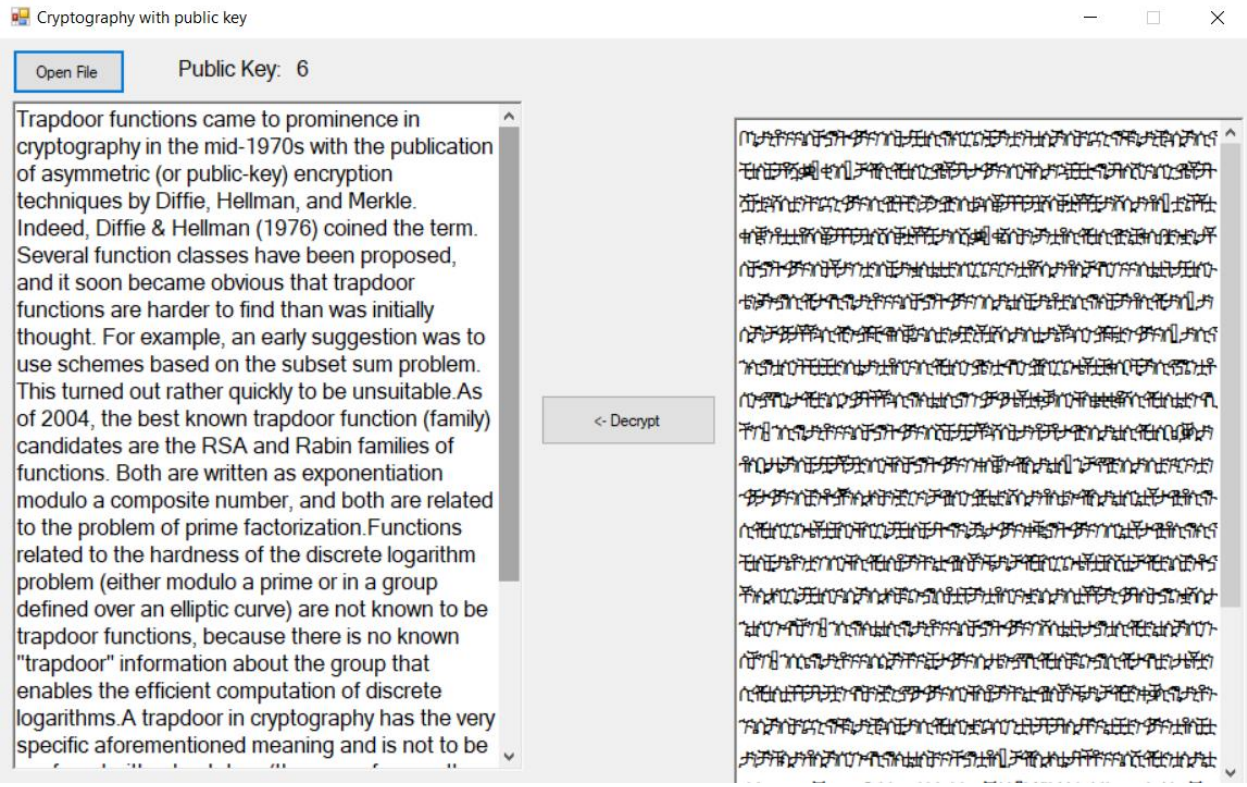
```
private void button2_Click(object sender, EventArgs e)
{
    if(openFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)
    {
        fileName = openFileDialog1.FileName;
        string[] lines = System.IO.File.ReadAllLines(fileName);
        EncryptedLines = new List<byte[]>();
        foreach (var line in lines)
        {
            richTextBox1.AppendText(line);
            var byteArray = ToByte(line);
            var tmp = Encrypt(byteArray);
            EncryptedLines.Add(tmp);
         richTextBox2.AppendText(Encoding.Unicode.GetString(tmp, 0, tmp.Length));

        }

    }
}
```

Kade se pritisne dugme <- *Decrypt* pomoću forme *button3_Click* šifrirani tekst se dešifrira i prikaže.

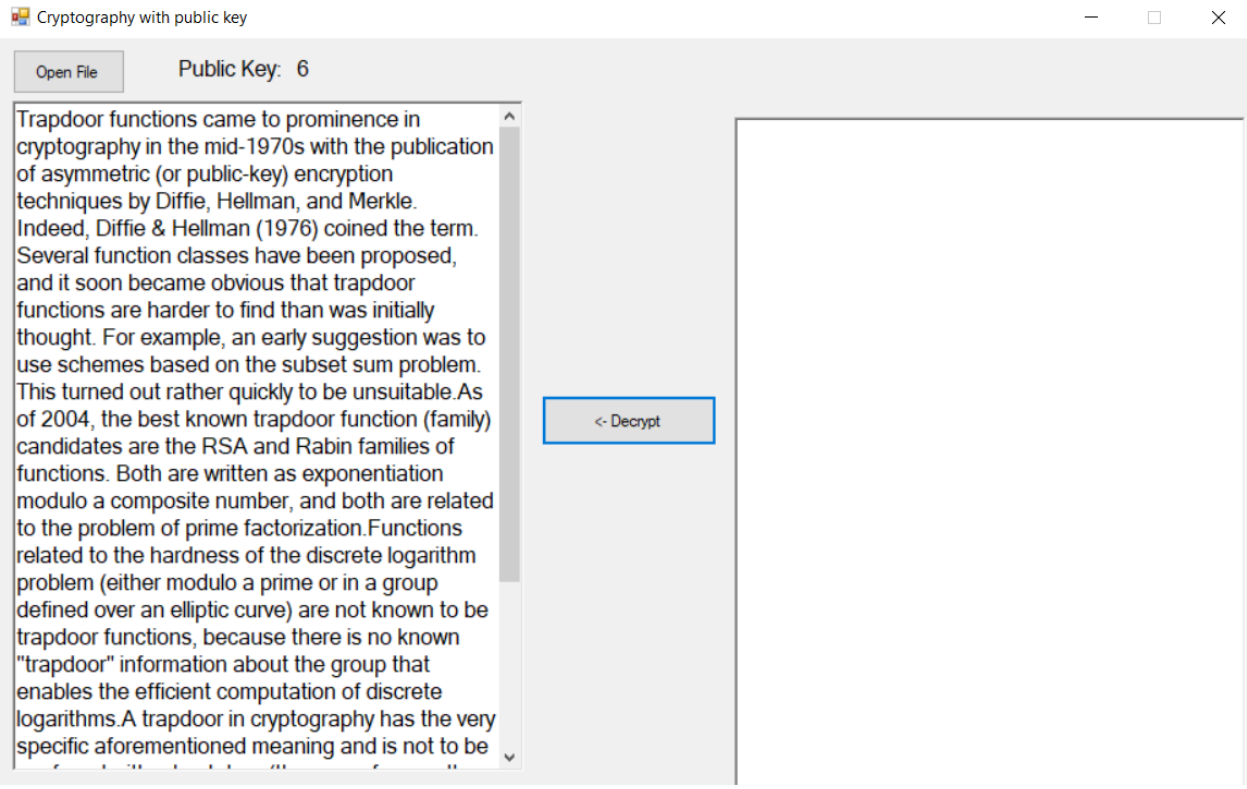
```
private void button3_Click(object sender, EventArgs e)
{
    if(richTextBox2.Text.Length > 0 && EncryptedLines.Count > 0)
    {
        richTextBox1.Text = "";
        System.Threading.Thread.Sleep(1000);
        foreach (byte[] line in EncryptedLines)
        {
            var tmp = Decrypt(line);
         richTextBox1.AppendText(Encoding.Unicode.GetString(tmp, 0, tmp.Length));
        }
        richTextBox2.Text = "";
    }
}
```



Slika 6 Program nakon pokretanja



Slika 7 Originalni tekst i šifrirani tekst



Slika 8 Program nakon dešifriranja

7. ZAKLJUČAK

U ovom radu je dan uvod u kriptografiju. Prikazan je razvoj kriptografije od prve šifre koja se koristila 1900. pr. Kr do razvoja prvih strojeva za šifriranje i moderne kriptografije. Moderna kriptografija dijeli se na sustave sa tajnim ključem (simetričnim ključem) i javnim ključem (asimetričnim ključem). Prikazane su prednosti i mane kriptografije sa simetričnim ključem i kriptografije javnog ključa.

U praktičnom dijelu dana je implementacija programa koji demonstrira osnovni princip enkripcije s javnim ključem. Program koristi naivnu enkripciju javnog ključa na tekstu učitanom iz datoteke. Kako bi se šifrirao tekst iz datoteke, on se najprije pretvara u bajtove nad kojima se vrši enkripcija.

8. LITERATURA

[1] Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone. (2001). *Handbook of Applied Cryptography*

[2] *Cryptography*. (2019). Dohvaćeno iz wikipedia: https://en.wikipedia.org/wiki/Cryptography#History_of_cryptography_and_cryptanalysis¸

[3] *Initialization Vector*. (2019). Dohvaćeno iz ldapwiki: https://ldapwiki.com/wiki/Initialization%20Vector

[4] *Cryptographic primitives*. (2019). Dohvaćeno iz packtpub: https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781787125445/3/ch03lvl1sec27/cryptographic-primitives

[5] *Initialization vector*. (2019). Dohvaćeno iz techopedia: https://www.techopedia.com/definition/26858/initialization-vector

[6] *Key agreement protocol*. (2019). Dohvaćeno iz doubleoctopus: https://doubleoctopus.com/security-wiki/protocol/key-agreement-protocol-2/

[7] *My crypto - Kriptologija skripta.* (2019). Dohvaćeno iz studocu: https://www.studocu.com/en/document/sveuciliste-u-zagrebu/kriptologija/practical/my-crypto-kriptologija-skripta/2343206/view

[8] *Legal Restrictions on Cryptography*. (2019). Dohvaćeno iz oreilly: https://www.oreilly.com/library/view/web-security-privacy/0596000456/ch04s04.html

[9] *Computer Science: How do digital signatures work?* (2019). Dohvaćeno iz quora: https://www.quora.com/Computer-Science-How-do-digital-signatures-work

[10] *Key-agreement protocol*. (2019). Dohvaćeno iz wikipedia: https://en.wikipedia.org/wiki/Key-agreement_protocol

[11] *Public Key Infrastructure*. (2019). Dohvaćeno iz tutorialspoint: https://www.tutorialspoint.com/cryptography/public_key_infrastructure.htm

[12] *Modern Cryptography*. (2019). Dohvaćeno iz tutorialspoint: https://www.tutorialspoint.com/cryptography/modern_cryptography.htm

[13] Stallings, W. (2005). *Cryptography and Network Security Principles and Practices, Fourth Edition* Prentice Hall

[14] *Introduction to Cryptography, version 8.0.* (2002). Dohvaćeno iz https://www.cs.unibo.it/babaoglu/courses/security/resources/documents/intro-to-crypto.pdf

[15] Mohamed Barakat, C. E. (2018). *An Introduction to Cryptography.* Dohvaćeno iz https://www.cs.unibo.it/babaoglu/courses/security/resources/documents/intro-to-crypto.pdf

[16] Nigel Smart, (2004). *Cryptography: An Introduction 3rd Edition.* https://www.cs.umd.edu/~waa/414-F11/IntroToCrypto.pdf.

[17] Laurens Van Houtven, (2013). *Crypto 101.*