UDK: 004.056.55

PRAKTIČNA PRIMENA SOFTVERSKIH ALATA OTVORENOG KODA U KRIPTOGRAFIJI

Dragan Ranđelović *
Kriminalističko-policijska akademija, Beograd
Miloš Ranđelović **
Help, Niš
Željko Kuzmanović ***
Skupština grada Banja Luka, BiH

Sažetak: Sigurnost računarskih sistema oduvek je bitna, a danas postaje još važnija, jer sve više korisnika na sve više načina koristi sve više informacija u sadašnjem informatičkom svetu.

Sa razvojem informacionih tehnologija i telekomunikacionih sistema i sve većom rasprostranjenošću ljudskog društva u geografskom smislu raste i mogućnost zloupotreba podataka koji se prenose otvorenim komunikacijskim putevima, što zahteva efikasniju zaštitu. U sistemu prenosa podataka napadači mogu lako da unište podatke, da ih modifikuju ili da informacije dođu u posed neovlašćenim osobama ili organizacijama, što može imati vrlo teške posledice. Problem je posebno izražen kod nekih organizacija kao što su državne administracije, pravosudne, vojne, medicinske, bankarske ali i druge. U ovom radu su objašnjeni osnovni pojmovi vezani za kriptovanje i algoritmi koji su se koristili i koji se koriste, kao i metode zaštite podataka u računarskim mrežama bazirane na kriptografiji kako bi se zaštitila takozvano sveto trojstvo sigurnosti CIA (od početnih slova engleskih reči: poverljivost – *confidentiality*, integritet – *integrity*, dostupnost – *availability*) unutar mreže računara. U radu su objašnjeni simetrični i asimetrični kriptosistemi i tehnika digitalnog potpisa. Takođe, predmet rada je predstavljanje programa *CrypTool i CryptoWorkflow*, koji spadaju u najpoznatije slobodne, tj. *shareware* softvere u oblasti kriptografije, sa posebnim naglaskom na mogućnosti primene različitih algoritama,

* Redovni profesor, dragan.randjelovic@kpa.edu.rs
** Spec. kriminalista, miciii84@gmail.com
*** Spec. kriminalista, zeljkokuzmanovic@blic.net

kao i komparativna analiza njihovog rada, a dati su i konkretni primeri njihovog korišćenja.

Ključne reči: kriptografija, simetrični i asimetrični sistemi, digitalni potpis, *CrypTool*, *CryptoWork flow*.

Uvod

Kriptografija je nauka „tajnog ključa", tj. nauka čuvanja informacija u onoj formi koja će biti čitljiva samo onima kojima je namenjena, dok će za ostale biti neupotrebljiva. Uporedo sa razvojem kriptografije razvila se i nauka kojoj je cilj da analizom kriptovanih poruka odgonetne njen sadržaj. Ta nauka o otkrivanju, odnosno „razbijanju" kriptovanih poruka naziva se kriptoanaliza, što je nastalo od grčke reči *kryptos*, što znači skriveno, i reči *analise*, što znači razmrsiti, i predstavlja proučavanje metoda za otkrivanje šifrovanih informacija, bez posedovanja tajnih podataka koji su uobičajeno potrebni da bi se pristupilo tim informacijama i obično podrazumeva pronalaženje tajnog ključa.

Netehničkim izrazima rečeno, kriptoanaliza je praksa „razbijanja šifara", mada ovaj izraz ima specijalizovano tehničko značenje.[4]

Objedinjene kriptografija i kriptoanaliza se nazivaju kriptologija (grč. *κρυπτός, kryptós – skriven + λόγος, logos – znanje, nauka*). Objekti izučavanja kriptologije su pisane (kriptografija), govorne (kriptofonija), vizuelne (slike, karte, šeme) i druge poruke.

Treba napomenuti jednu bitnu razliku između termina „kriptografija" i termina „kriptologija". *Kriptografija* je nauka koja se bavi svim aspektima sigurnosnog transporta podataka, kao što su autentifikacija (*web*, lokalne mreže i sl.), digitalni potpisi, razmena elektronskog novca, odnosno nauka koja proučava kriptovanja i dekriptovanja podataka. *Kriptologija* je, za razliku od nje, grana matematike koja se bavi matematičkim načelima, te matematičkom implementacijom kriptografskih metoda, a njome se uz matematiku danas bave i druge discipline, npr. digitalna forenzika.[5]

Originalna poruka koju pošiljalac šalje, u daljem tekstu rada će se zvati čisti tekst ili original (*plaintext* ili *cleartext*). Kodiranje poruke, tj. postupak pretvaranja originala (čistog teksta) u nečitljiv oblik nazivaćemo kriptovanjem ili enkripcijom. Tako šifrovan ili kriptovan tekst (*ciphertext* ili *chipher*) mi ćemo jednostavno nazivati kriptovanom porukom ili enkripcijom (*encryption*). Dešifrovanje poruke, tj. vraćanje poruke iz njenog kriptovanog oblika u originalni (*čisti tekst*) nazivaćemo dekriptovanje (*decryption*). Vrlo važan termin u kriptografiji je „ključ"; on ima veliku ulogu u enkripciji i dekriptovanju poruke. Skup pravila i konvencija koji definiše komunikacioni okvir između dva ili više učesnika u komunikaciji predstavlja *protokol*, u koji spadaju: uspostavljanje

4 H. Van Tilborg, *Encyclopedia of Cryptography and Security*, University of Technology Eindhoven, New York, 2005.

5 D. Randjelović; D. Delija; B. Popović, EnCase forenzički alat, *Bezbednost*, 1-2/2009, Beograd, pp. 286–312.

veze, održavanje veze, raskid veze i obnavljanje veze u slučaju prekida. Oni se upotrebljavaju za uspostavljanje sigurne komunikacije preko nepouzdanih globalnih mreža i distribuiranih sistema, a oslanjaju se na kriptografske metode zaštite kako bi korisnicima obezbedili osnovne sigurnosne usluge CIA trojstva (od početnih slova engleskih reči: poverljivost – *confidentiality*, integritet – *integrity*, dostupnost – *availability*).[6]

1. Vrste kriptografije

U prvom poglavlju rada razmotrićemo osnovne pojmove o vrstama kriptografskih tehnika.

1.1. Simetrična kriptografija

Simetrična kriptografija je najstariji oblik kriptografije, star gotovo koliko i ljudska komunikacija (naziva se i kriptografijom tajnog ključa jer se podatak kriptuje i dekriptuje istim ključem). Za proces kriptovanja u simetričnoj kriptografiji potrebno je znati algoritam kriptovanja i tajni ključ. Nekad su se algoritmi držali u tajnosti, ali se pokazalo da skrivanje algoritma ne doprinosi sigurnosti. Svi savremeni simetrični algoritmi javno su objavljeni. Zbog toga ih je u potpunosti moguće testirati i proveriti njihovu otpornost na napade, odnosno moguće ih je analizirati (kriptoanaliza). Sigurnost simetričnih algoritama zavisi od sigurnosti samog algoritma i dužine ključa. Najpoznatiji simetrični algoritam je DES (*Data Encryption Standard*), koji je razvio IBM 1977. Bio je standard za simetrične algoritme sve do 2000. godine kad ga je zamenio AES (*Advanced Encryption Standard*), koji rukuje ključevima dužine 128, 192 i 256 bita. Glavni razlog zbog kojeg je DES zamenjen AES-om jeste taj što DES ima dužinu ključa od 56 bita. Osnovna osobina simetričnih kriptosistema, tj. kriptografije tajnim ključem, jeste da je to postupak kojim se koristi jednak ključ za enkripciju i dekripciju podataka. Simetričnu kriptografiju možemo matematički prikazati izrazima: enkripcija – $C = E_k (M)$ i dekripcija – $M = D_k (C)$, gde E predstavlja enkripcijsku funkciju, D dekripcijsku funkciju, K je tajni ključ jedinstven za obe strane, M je originalna (*plaintext*) poruka, a C pripadajuća enkriptovana poruka (*ciphertext*).[7]

1.2. Asimetrična kriptografija

Kriptografija javnog ključa, poznata i kao asimetrična kriptografija, novijeg je datuma od simetrične. Kod nje se ključ za dešifrovanje razlikuje od ključa za šifrovanje. Zajedno, dva ključa se zovu ključni par; on se praktično sastoji od javnog ključa, koji može biti javno dostupan i privatnog ključa, koji mora ostati

6 A. Ruth, K. Hudson; *Sertificat Security+*, CET, Beograd, 2004.
7 V. Kovačević, *Zaštita podataka primenom kriptografskih metoda*, Seminarski rad Elektronski fakultet Niš, 2010.

tajna. U većini slučajeva javni ključ se koristi kao ključ za šifrovanje, a privatni kao ključ za dešifrovanje.

Asimetričnu kriptografiju su utemeljili Vitfild Difi (*Whitefield Diffie*) i Martin Helman (Martin Hellman) kada su 1976. godine opisali ideju kriptografije utemeljenu na dva ključa, privatnom (često zvan i tajni) i javnom ključu. Razlika između simetričnih i asimetričnih algoritama jeste u tome što simetrični algoritmi koriste isti ključ za kriptovanje i dekriptovanje, dok asimetrični algoritmi koriste različite ključeve. Informacije kriptovane javni ključem mogu se dekriptovati samo privatnim ključem, dakle to može uraditi samo osoba koja je vlasnik tajnog asimetričnog ključa. U literaturi se pojam asimetričnog kriptovanja poistovećuje sa terminom *asymmetric-key* ili *public-key* kriptovanje. Osim toga, kriptovanje javnim a dekriptovanje tajnim pokazalo se takođe kao odlično svojstvo i omogućava digitalno potpisivanje informacija tamo gde potpis može biti proveren javnim ključem od bilo koga. Ključevi treba da budu povezani jednosmernom funkcijom, odnosno ne sme da bude moguće izračunavanje privatnog ključa iz javnog ključa, bar ne u kratkom periodu. Najpoznatiji asimetrični algoritmi danas su *RSA* (*Rivest-Shamir-Adleman*), *Diffie-Hellman*, *ElGamal* itd.[8]

Asimetrični kriptosistemi, odnosno njihovi algoritmi, zasnivaju se na određenim svojstvima brojeva koji se proučavaju u teoriji brojeva. Pri kriptovanju se izvorni tekst kodira kao niz prirodnih brojeva koji se odabranom funkcijom kriptovanja i ključem kriptovanja *Ke* preračunavaju u niz brojeva kriptovanog teksta. Funkcija kriptovanja mora biti takva da iz niza brojeva kriptovanog teksta napadač samo sa izuzetno, velikim naporima, može odrediti izvorni niz brojeva, a da poznavanje ključa dekriptovanja *Kd* omogućuje lako izračunavanje izvornog niza brojeva. Asimetrično kriptovanje predstavlja složeniji vid zaštite podataka. Za njegovu realizaciju potrebna su nam dva ključa kod svakog od onih koji komuniciraju. Jedan ključ je dostupan svima preko javnih kataloga ili imenika, te se zato i naziva javnim ključem. Drugi ključ poznat je samo vlasniku i naziva se tajnim. Međutim, iako različiti, ključevi su međusobno povezani određenim transformacijama.[9]

1.3. Tehnika digitalnog potpisa

Tehnika digitalnog potpisa koristi tehniku asimetričnog kriptovanja. Pošiljalac i primalac imaju par ključeva od kojih je jedan tajni, a drugi svima dostupan javni ključ. Ključevi predstavljaju matematičke algoritme koje je izdalo sertifikaciono telo. Digitalni potpisi se koriste za identifikaciju izvora informacije, što može biti neka osoba, organizacija ili računar. Sama ideja digitalnog potpisa slična je klasičnom potpisivanju dokumenata jer, ukoliko se neki dokument želi poslati elektronskim putem, on se mora i potpisati, pri čemu je, za razliku od klasičnog potpisa, digitalni potpis gotovo nemoguće falsifikovati. Na osnovu iznetog se zaključuje da je za funkcionalnost digitalnog potpisa potrebno izvršiti dva procesa, od kojih jedan sprovodi potpisnik, a drugi primalac. Uspešnom proverom digitalnog potpisa garantuju se:

– autentičnost – pouzdanost identiteta pošiljaoca posledica je činjenice da je otisak poruke koji je šifrovan tajnim ključem moguće uspešno dešifrovati

8 *Ibidem*.
9 F. Piper; S. Murphy, *Cryptography: A very short Introduction*, Oxford, 2002.

samo primenom odgovarajućeg javnog ključa;
– integritet – upoređivanjem izračunatog i dešifrovanog otiska poruke utvrđuje se da poruka nije modifikovana;
– neporicivost – pošiljalac ne može da porekne slanje poruke pošto je potpisana njegovim tajnim ključem.

Važno je napomenuti da elektronski potpis uopšte, pa tako ni digitalni potpis, ne pruža zaštitu tajnosti podataka od neovlašćenog čitanja jer se svi podaci šalju u svom originalnom (nepromenjenom) obliku.

1.3.1. Digitalni sertifikat

Kreiranje digitalnog potpisa i njegova verifikacija vrše se, kako je već pomenuto, asimetričnim kriptografskim sistemima prilikom čega se koriste: tajni (privatni) ključ poznat jedino korisniku i javni ključ poznat širem krugu ljudi, a ne samo primaocu.

Postavlja se pitanje kako možemo biti sigurni da je to zaista javni ključ potpisnika. Rešenje ovog problema postiže se upotrebom digitalnog sertifikata. Digitalni sertifikat je digitalno potpisani dokument koji povezuje javni ključ s osobom kojoj pripada i možemo ga nazvati i digitalnom ličnom kartom jer on to i zaista jeste, tj. digitalna lična karta u „cyber prostoru", sredstvo kojim dokazujemo identitet na Internetu.

Digitalni sertifikat (*digital certificate*) predstavlja element kojim se utvrđuje veza između identiteta subjekta i njegovog javnog ključa primenom asimetričnog algoritma.

Elementi koji čine strukturu digitalnog sertifikata su: verzija formata sertifikata, serijski broj sertifikata, identifikator algoritma, naziv sertifikacionog tela, rok važnosti sertifikata, vlasnik sertifikata, polje dodatnih atributa, informacija o javnom ključu vlasnika i digitalni potpis sertifikata od strane ustanove koja je izdala sertifikat (SA od početnih engleskih reči *certificate authority*).

Prema dosadašnjim iskustvima ovakva struktura sertifikata ispunjava zahteve savremenih kriptografskih sistema zaštite. Većina savremenih sistema zaštite, koji uključuju infrastrukturu sa javnim ključevima (PKI od početnih engleskih reči *public key infrastructure*) bazira se na primeni digitalnog sertifikata.

1.4. Potpisivanje sažetka poruke

Ne štite sve vrste opisanih algoritama šifrovanja integritet, odnosno verodostojnost poruke koja je šifrovana. Funkcija za sažimanje (skraćivanje, kompresiju...) ili heš (*hash*) funkcija jeste tehnika koja obezbeđuje proveru integriteta poruke, što je važno jer je moguće da je ključ otkriven i da nam napadač šalje lažne poruke, ali i da je došlo i do greške prilikom šifrovanja, tako da primljena poruka nije identična originalnom dokumentu. Iz tog razloga kreirane su pomenute heš ili funkcije za sažimanje, koje se mogu sresti i pod imenima *one-way*, *hash function*, *message digest*, *fingerprint* algoritmi. Najpoznatiji i najkorišćeniji heš algoritmi su: *SHA-224, SHA-256, SHA-384, SHA-512, MDC-2, RIPEMD-160*, kao i stariji *SHA-1* (*Secure Hash Algorithm 1*) sa 160-bitnim sadržajem, *MD5* (*Message Digest 5*) sa 128-bitnim

sadržajem koji su trenutno u široj upotrebi, ali kod oba algoritma uočene su greške i trebalo bi da budu povučeni iz upotrebe. Heš algoritmi se svrstavaju u kriptografske algoritme bez ključa.

Ovi heš algoritmi prosto sažmu (u bukvalnom prevodu samelju) svaku poruku ili fajl bez obzira na veličinu i na izlazu dobijamo poruku konstantne dužine, u zavisnosti od algoritma. Iz dobijenog izlaza nemoguće je rekonstruisati ulaznu poruku, a bitno je da je, isto tako, gotovo nemoguće kreirati dve smislene poruke koje će imati iste vrednosti heš funkcije, te tako mi u svakom trenutku možemo da proverimo integritet poruka, odnosno da primetimo razliku u tekstu primljene poruke, prostim ponovnim proračunavanjem heš funkcije i upoređivanjem dobijenih rezultata.

Verovatnoća da u poruci neko izmeni neku stavku, tako da novi dobijeni tekst ima istu heš vrednost kao i originalni tekst, u slučaju 160 bitnih algoritama jeste zanemarljiva, zato se negde heš funkcije nazivaju i otisci prstiju poruka.

Ukoliko su poruke duge, korišćenje kriptovanja sa javnim ključem za potpisivanje cele poruke veoma je nepraktično zbog velikih dužina poruka, što iziskuje dosta resursa i troši mnogo vremena za kriptovanje. Zato se kao logično rešenje ovog problema javlja mogućnost, potpisivanja samo sažetka umesto potpisivanja cele poruke. Osoba koja šalje poruku kreira skraćenu verziju poruke tj. njen sažetak. Tako formiran sažetak potpisuje i šalje komunikacionim kanalom a osoba koja primi tako skraćenu poruku proverava njen potpis. Svaka promena izvorne poruke izaziva promenu u sadržaju, što se odražava na promenu potpisa, čime se minimizuje mogućnost zloupotrebe.[10]

Tehnika potpisivanja sadržaja poruke uglavnom koristi neku od dve heš funkcije: *MD5* (*Message Digest 5*) sa 128-bitnim sadržajem i *SHA-1* (*Secure Hash Algorithm 1*) sa 160-bitnim sadržajem. Za garantovanje sigurnosti poruke heš funkcija mora zadovoljiti dve stvari:[11]

1) funkcija sažimanja se obavlja u jednom smeru. Sadržaj se jedino može formirati na osnovu originalne poruke, ali ne i obrnuto a formiranje sadržaja treba da bude brzo i jednostavno;
2) heš funkcija je jednoznačna, tj. primena iste heš funkcije na istoj poruci daje isti sažetak.

Nakon kreiranja sadržaja poruke, vrši se kriptovanje (potpisivanje) istog korišćenjem tajnog ključa osobe koja šalje poruku (osoba *A*). Obično se za kriptovanje koristi *RSA* algoritam. Kriptovani sadržaj se upakovan zajedno sa originalnom porukom šalje osobi *B*. Osoba *A* primenom heš funkcije formira sažetak koji se potpisuje i upakovan sa porukom šalje osobi *B*. Osoba *B* prima originalnu poruku i kriptovani sadržaj zajedno sa potpisom, pa nakon prijema vrši razdvajanje. Primenom heš funkcije na originalnu poruku osoba *B* kreira drugi sadržaj. Takođe, dekriptuje kriptovani sadržaj koji je primila od osobe *A* uz pomoć javnog ključa osobe *A*. Konačno, vrši poređenje predhodno kreiranih sadržaja i ukoliko su isti verifikacija poruke je uspešno obavljena.

10 S. Sinkovski; B. Lučić, *Informaciona bezbednost i kriptografija*, ZITEH, Beograd, 2006.
11 *Ibidem*.

2. Softver *CrypTool*

CrypTool je besplatna aplikacija za učenje za *Windows*. To je programski paket namenjen obrazovanju korisnika o mogućnostima i načinu rada različitih klasičnih kao i modernih simetričnih, asimetričnih i heš kriptografskih algoritama. Može se koristiti za primenu i analizu kriptografskih algoritama koji se koristi širom sveta. Takođe je moguće digitalno potpisivanje dokumenata i njihova verifikacija, analiza načina rada *Diffie-Hellmen* protokola, komprimovanje dokumenata kao i brojne napredne mogućnosti analize šifrovanih poruka (entropija, histogrami, periodičnost uzoraka itd.).

CrypTool alat dostupan je za *Windows* operativne sisteme, a karakteriše ga vrlo pregledan i intuitivan grafički interfejs kao i brojne opcije koje korisniku olakšavaju proces savladavanja načina rada pojedinih kriptografskih algoritama. Ova aplikacija podržava i savremene nastavne metode u školama i univerzitetima, kao i svest za obuku državnih službenika i zaposlenih.

Razvoj *CrypTool*-a započeo je 1998. godine. Od svoje prvobitne upotrebe u oblasti sigurnosti informacija za firme *CrypTool* se razvio u izvanredan projekat otvorenog koda za teme vezane uz kriptologiju.

Počev od proleća 2008. godine, *CrypTool* projekat radi sa kriptoportalom za nastavnike. Portal je za sada dostupan samo na engleskom, nemačkom, španskom, poljskom i srpskom, kao platforma za nastavnike kojim deli svoje materijale za nastavu o kriptologiji.

Od proleća 2009. godine, *CrypTool* projekat poseduje i *CrypTool-Online*. Ovaj portal daje ljudima zainteresovanim za kriptologiju mogućnost da probaju razne šifre i metode šifrovanja u svom pretraživaču bez preuzimanja ili instaliranja dodatnog softvera.

CrypTool tim radi na dva projekta koji bi trebalo da budu naslednici *CrypTool 1.4.x* koji je napisan u *C++* i radi samo na operativnom sistemu Microsoft Windows. Eventualno, može se pokrenuti bez problema pod Juniksovim operativnim sistemima (UNIX operating systems) uz pomoć softvera *Wine*.[12] Verzija (port) za prebacivanje *CrypTool 1.H* na *Linuks* (LINUX operating system) sa *Qt4* je počela, ali veoma sporo napreduje. Nasuprot tome, dva potprojekta koji se razvijaju od 2007. (novo dizajnirani naslednici) koristeći arhitekturu dodataka (*Plug-in*) napravili su dobar napredak.[13]

CrypTool 2.0 razvijen u *C#* sa *Visual Studio 2008 (Express Edition)* i *WPF*. U julu 2008. objavljena je prva beta verzija (za programere i krajnje korisnike). Ona se ažurira u kontinuitetu. *CrypTool 2.0* pruža potpuno razvijenu arhitekturu i korisniku kriptografsku funkcionalnost uz neverovatni prevuci i spusti *GUI*. *CrypTool 2.0* zahteva najmanje *Microsoft Windows XP* i *NET framework* verzije *3.5SP1*. Beta verzija *CrypTool* 2.0 se stalno ažurira i najnovija stabilna verzija se može preuzeti kao instalaciona datoteka. Pored toga, izvornom kodu se može pristupiti preko *SVN* servera. Ova verzija *CrypTool*-a 2.0 još uvek ne može da zameni *CrypTool 1.x*. Beta verzija ja stabilan pregled tehnologije koju mogu da koriste programeri ili korisnici.[14]

12 Dostupno na: http://en.wikipedia.org/wiki/Code (16.1.2013.)
13 Dostupno na: http://en.wikipedia.org/wiki/CrypTool (16.1.2013.)
14 B. Esslinger, CrypTool Development Team, *The CrypTool Script*, Frankfurt am Main, 2010.

JCrypTool 1.0 je razvijen u *Javi* i zasnovan je na *Eclipse RCP*. Trenutna beta verzija (nazvana objavljenim kandidatom RC1a, namenjena programerima i korisnicima) objavljena u januaru 2010. *JCrypTool 1.0* platformski je nazavisan (*Windows*, *Linix*, *Mac*) i koristi *FlexiProvider* (moćan skup alata razvijen od strane *TU Darmstadt*) i *BouncyCastle* za *Java Criptogaphy Architecture JCA*. Izvorni kod se može preuzeti preko javnog *SVN* servera na *SourceForge* sa *anonimous* pristupom čitanja. Ova beta verzija pruža stabilnu platformu za razvoj programerima koji razvijaju datoteke i žele da doprinesu *JCrypTool* projektu.

Trenutna verzija *CrypTool 1.4.30* objavljena je 4. avgusta 2010. godine. Ova verzija zahteva *Win32* okruženje. Program sadrži neke funkcije koje zovu *Java* aplikacije. Da bi pokrenuli ove funkcije Java runtime okruženje pod *Win32* (najmanje JRE 1.6) mora biti inastalirano. Izvorni kodovi trenutne verzije (sa oznakom *CrypTool 1.4.30*) i izvori najnovijih promena su dostupni iz subverzionog repozitorijuma. Svako ima pravo čitanja iz repozitorijuma (korisničko ime *anonymous* i prazno polje za lozinku). *CrypTool* nudi sledeće funkcije: brojne klasnične i moderne kriptografske algoritme (šifrovanje i dešifrovanje, generisanje ključa, sigurne lozinke, autentikaciju, sigurnosne protokole); vizualizaciju sa nekoliko metoda (npr. *Caesar*, *Enigma*, *RSA*, *Diffie-Hellman*, digitalni potpis, *AES*); kriptoanalizu određenih algoritama (npr. *Vigenere*, *RSA*, *AES*); kriptoanalitičke metode merenja (npr. entropiju, engrame, autokorelaciju); pomoćne metode (npr. testovi za proste brojeve, rastavljanje na proste činioce, *base64* kodiranje); tutorijal o prostim brojevima; sveobuhvatnu online pomoć; skriptu sa podrškom za dalje informacije o kriptologiji.

Paket *CrypTool* je dobio razna međunarodna priznanja kao obrazovni softver (*TeleTrusT Special Award 2004, EISA 2004, IT Security Award NRW 2004, Selected Landmark in the Land of Ideas 2008*). Preuzme se oko 3.000 puta mesečno (od toga je oko trećine engleska verzija). Koristi se u školama, na univerzitetima, u preduzećima, kao i u specijalizovanim ustanovama.

2.1. *CrypTool* simetrična enkripcija metodom *Triple DES*(*3- DES*)

DES (*Data Encryption Standard*) jeste najpoznatiji algoritam za simetrično šifrovanje podataka, prvi koji je službeno prihvaćen kao standard. Razvijen je sredinom sedamdesetih godina XX veka u IBM-u, a nakon toga je prihvaćen kao američki standard (od strane NIST-a i nešto kasnije ANSI-a) za šifrovanje podataka.[15]

DES koristi blokove od 64 bita, podeljene na dva 32 bitna bloka (*L* i *D*), uz efektivnu dužinu ključa od 56 bita. Broj koraka je 16, a svaki korak koristi 48-bitni ključ (*Ki*) generisan iz izvornog ključa. U svakom koraku nad 32 bitnim podatkom.

TripleDES (poznat i kao 3-DES i TDEA) jeste simetrični blok-algoritam za šifrovanje koji takođe predstavlja proširenje DES algoritma. Nastao je kad je postalo očito da DES ne pruža dobru zaštitu, a u tom trenutku nije bilo dostupnih boljih šifarskih algoritama. Sam *TripleDES* se smatra potpuno sigurnim od *brute-force* napada, ali je vrlo spor (tri puta sporiji od DES-a). Algoritam su predložili

15 Dostupno na: http://en.wikipedia.org/wiki/Triple_DES (2013)

V. Difi (W. Diffie), M. Helman (M. Hellman) i V. Tačman (W. Tuchmann), a prihvaćen je kao standard. *TripleDES* koristi 192-bitni ključ (koji se deli u tri 64-bitna dela), a blok podataka šifruje i dešifruje koristeći standardni DES algoritam, tri puta zaredom. Ako se otvoreni tekst označi sa *P*, šifrovani tekst sa *S*, a ključ sa K1K2K3, algoritam glasi:

S = DES_Šifrovanje (DES_Dešifrovanje (DES_Šifrovanje (*P*, *K1*), *K2*), *K3*);

P = DES_ Dešifrovanje (DES_ Šifrovanje (DES_ Dešifrovanje (*C*, *K3*), *K2*), *K1*).

Kako jedna *TripleDES* operacija više puta koristi DES algoritam, posebno su definisani načini rada *TripleDES* algoritma, a način kriptovanja/dekriptovanja podataka *P* pomoću tri ključa (*K1,K2* i *K3*) dat je sledećom procedurom.

Prvi ključ *K1* se koristi za kriptovanje bloka podataka izvorne poruke *P* pomoću standardnog *DES* algoritma. Tako kriptovana poruka se dekriptuje drugim ključem *K2*. Dekriptovanjem poruke ovim ključem dobija se nova šifrovana poruka. Na kraju se rezultat dekriptovanja opet kriptuje, ovaj put ili trećim ključem *K3* ili opet prvim *K1*. Tako je konačno formirana kriptovana poruka *S*.

Naizmeničnim korišćenjem različitih ključeva povećava se efektivna dužina ključa na ukupno 168 bita, a tako i broj kombinacija koje bi eventualni napadač morao probati da bi došao do izvorne poruke. Broj kombinacija za dva različita ključa je 2112, dok za tri različita ključa ima čak 2168 kombinacija. *3-DES* algoritam, kako ga još nazivaju, rešava problem dužine ključa običnog *DES*-a, a nedostatak mu je to što je mnogo sporiji od standardnog *DES*-a.[16]

Dekriptovanje poruke koja je kriptovana pomoću *3-DES* algoritma se, kao i kod klasičnog *DES*-a, obavlja inverznim funkcijama u odnosu na kodiranje. Prvo se obavlja dekriptovanje pomoću trećeg ključa, sledi kriptovanje pomoću ključa *K2* i na kraju se izvorna poruka *P* dobija postupkom dekriptovanja pomoću ključa *K1*.

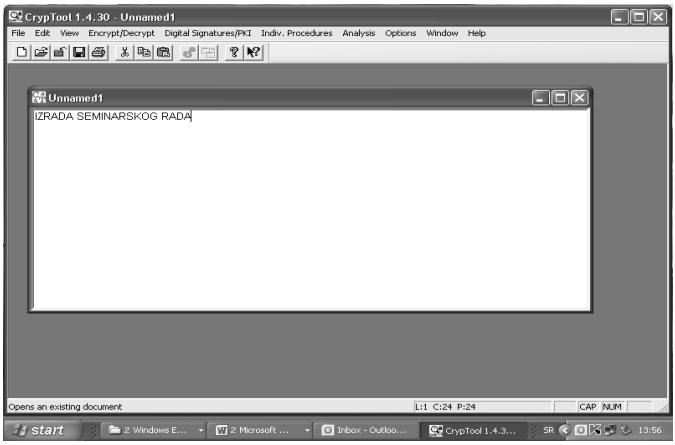
PRIMER 1

Prvo startujemo program *CrypTool*. U glavnom meniju idemo na opciju *File* i izaberemo *New*. Otvarimo novi prozor, *Unnamed1*, u koji upisujemo poruku ili tekst koji želimo da zaštitimo (kriptujemo). Na primer naša poruka neka glasi:

IZRADA SEMINARSKOG RADA.

Kada upišemo poruku u prozor *Unnamed1*, na monitoru imamo situaciju kao na slici 1:

16 Dostupno na: http://en.wikipedia.org/wiki/3-DES (2013).

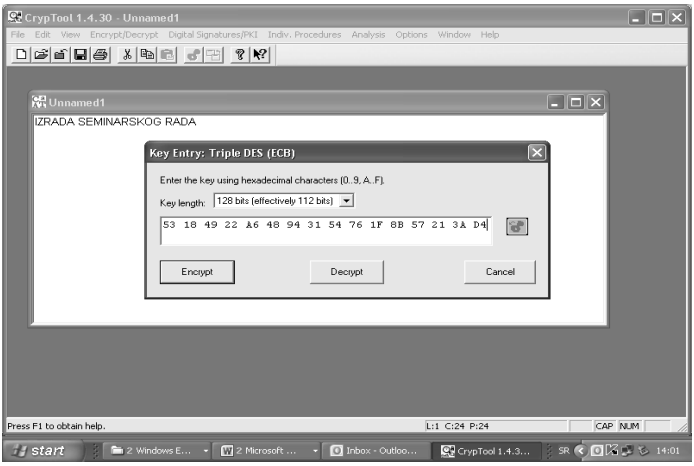


Slika 1 -*Izgled prozora Unnamed1 sa ispisanom porukom za primer 1*

Posle ovoga u glavnom meniju idemo na opciju *Encrypt/Decrypt*, a potom na opciju *Symmetric* (*modern*), te izaberemo opciju *Triple DES* (ECB);

U narednom koraku upišemo ključ pomoću koga će biti izvršena enkripcija teksta (naše poruke). Izbor ključa je slobodan i neka bude: *53 18 49 22 A6 48 94 31 54 76 1F 8B 57 21 3A D4*

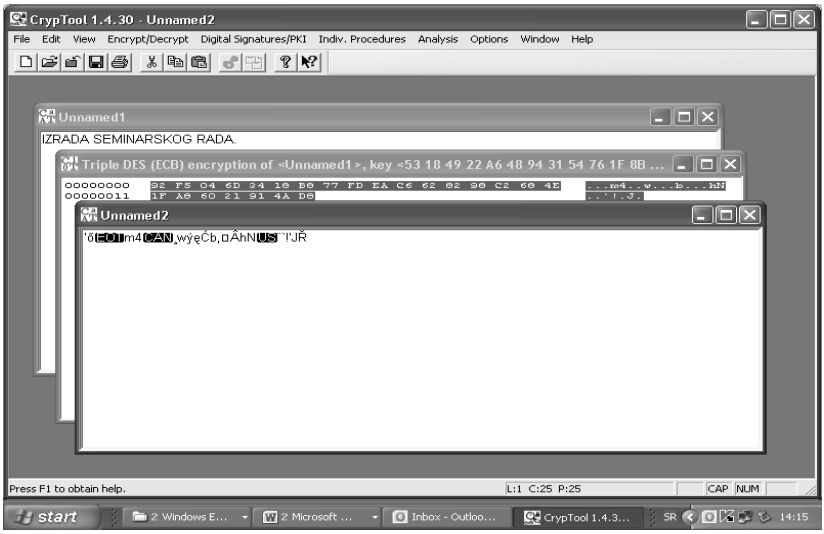
Na monitoru imamo sada slijedeću situaciju (slika 2):



Slika 2: *Upisivanje ključa*

Pritiskom na *Encrypt* naša kriptovana poruka je poprimila oblik: *ŏm4124¸ wýęĆb, ÂhN¨`!'JŘ*

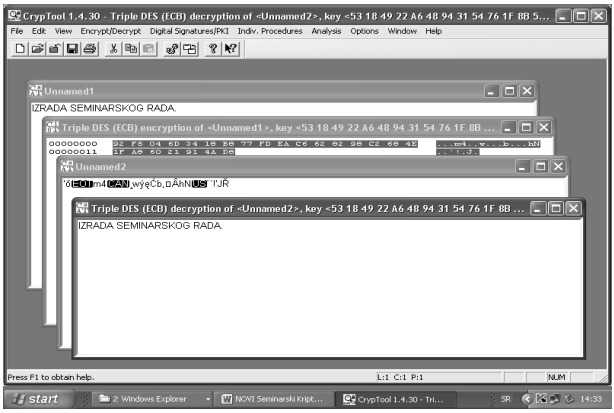
Kada želimo da dekriptujemo istu ovu poruku prvo selektujemo tekst pomoću opcije *Encoded Copy/Cut* gde mora biti selektovano *no encoding*, zatim idemo na glavni meni na opciju *File* i izaberemo opciju *New*. Otvara se novi prozor *Unnamed2*, u koji unesemo našu kriptovanu poruku. Posle toga na monitoru dobijemo sledeću situaciju (slika 3):



Slika 3 -*Izgled prozora Unnamed2 sa kriptovanom porukom*

Posle ovoga opet idemo na *Encrypt/Decrypt*, a zatim na opciju *Symmetric* (*modern*), izaberemo opciju *Triple DES (ECB)* gde unesemo ključ kojim smo kriptovali prvobitnu poruku.

Pritiskom na taster *Decrypt* dobijamo našu prvobitnu poruku prikazanu kao na slici 4:



Slika 4: *Izgled prozora sa dekriptovanom porukom*

2.2. CrypTool asimetrična enkripcija metodom *RSA*

RSA algoritam je jedan od najkorišćenijih asimetričnih algoritama danas. RSA je skraćenica koja je nastala od prezimena njegovih tvoraca: Rona Rivesta, Adija Šamira i Leonarda Adelmana (Rivest, Shamir, Adleman). Svetlost dana ugledao je davne 1977. godine. U RSA algoritmu ključnu ulogu imaju veliki prosti brojevi. To su, kao što znamo, brojevi koji su deljivi samo samim sobom i jedinicom. Prosti

brojevi (P i Q) u ovom algoritmu služe za generisanje javnog i tajnog ključa i to preko sledećih jednostavnih formula:

$K_{javni} = P \cdot Q$; $K_{tajni} = (2 \cdot (P - 1) \cdot (Q - 1) + 1) / 3$

Algoritam kodiranja i dekodiranja sastoji se iz dve formule.

Kodiranje: $M_{kodirano} = (M_{izvorno}{}^{3}) \bmod K_{javni}$

Dekodiranje: $M_{izvorno} = (M_{kodirano}{}^{Ktajni}) \bmod K_{javni}$

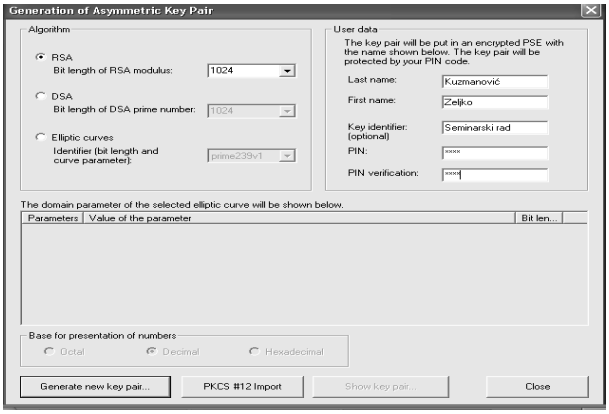
Ono što je pohvalno za ovaj algoritam, jeste njegova jednostavnost, ali i sigurnost. Na primer, kod *Pentium I*, u odnosu na dužinu ključa, vreme koje je potrebno da kompjuter brzine 1 MIPS iz javnog izračuna tajni ključ ima oko 150 MIPS-a.[17]

PRIMER 2

Startujemo program *CrypTool*. U glavnom meniju idemo na opciju *File*, i izaberemo *New* i otvorimo novi prozor, *Unnamed1*, u koji upisujemo neku poruku ili tekst koji želimo da zaštitimo (kriptujemo). Naša poruka naka sada glasi:

IZRADA SEMINARSKOG RADA JE ZAVRSENA.

Pre nego što enkriptujemo (zaštitimo) našu poruku, prvo moramo napraviti (generisati) asimetrični par ključeva. Idemo na opciju Digital Signatures/PKI u glavnom meniju, potom na opciju PKI i izaberemo opciju Generate/Import Keys; Na otvorenom prozoru biramo sledeće parametre: vrsta algoritma koji želimo da koristimo i podaci o korisniku tajnog ključa i unosimo *PIN kod* (naš izabrani PIN kod je 111) koji služi kao zaštita od neautorizovanog korišćenja koje je prikazano na slici 5:



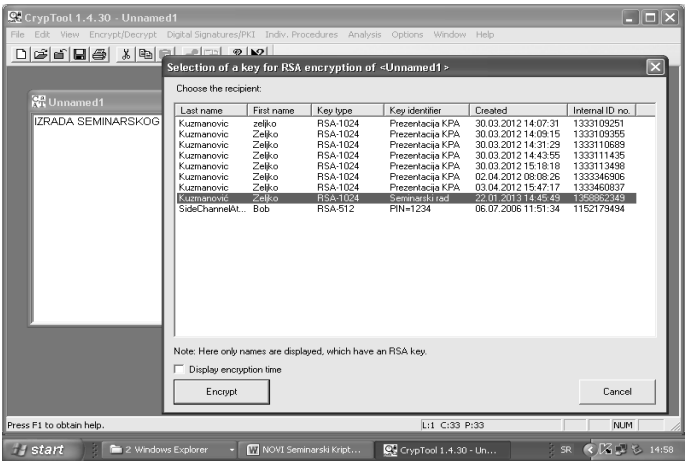
Slika 5 -*Izgled prozora Generate/Import Keys sa unešenim parametrima*

Posle izbora PIN koda pritiskom na taster *Generate new key pair…*, pojavljuje se sledeći „prozor" koji nas obaveštava da je ključ uspešno generisan. Pored toga daju se i osnovni podaci o licu koje je vasnik tajnog ključa, kao i vreme koje je bilo potrebno da se ključevi naprave.

Pritiskom na taster OK, zatvaramo prozor na *Close*, i ostajemo na početnoj strani (prozor *Unnamed1*). Idemo sada na opciju *Encrypt/Decrypt*, i opcijom

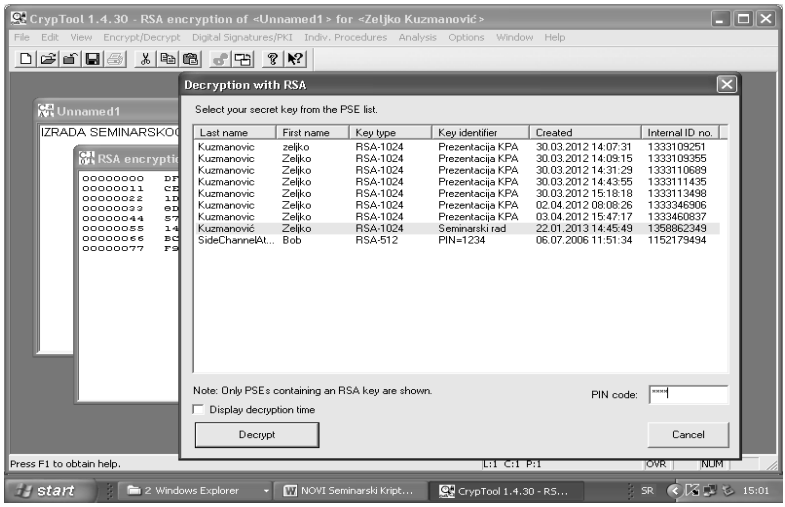
17 Dostupno na: http://en.wikipedia.org/wiki/RSA (2013)

Asymmetric, izaberemo opciju *RSA Encryption*. Posle ovog izbora pojavljuje se novi prozor gde možemo izabrati ključ za enkripciju naše poruke. Izabraćemo ključ koji smo generisali (slika 6):



Slika 6 -*Izgled prozora sa izborom ključa za enkripciju*

Pritiskom na taster *Encrypt*, pojaviće se novi prozor sa našom porukom u kriptovanom obliku. Posle ovoga idemo na opciju *Encrypt/Decrypt*, potom na opciju *Asymmetric*, i izaberemo opciju *RSA Decryption* kada se otvara novi prozor, u kome se od nas traži da izaberemo ključ, sa liste prethodno generisanih ključeva. Selektujemo naš ključ i vršimo upisivanje PIN koda (1111), pa imamo sledeću situaciju (slika 7):



Slika 7 -*Izgled prozora RSA Decryption sa upisanim ključem*

Pritiskom na taster *Decrypt*, otvara se novi prozor u kome se vidi naša izvorna poruka.

2.3. Tehnika digitalnog potpisa, *CrypTool*, heš funkcija MD5

Svrha digitalnog potpisa je da potvrdi autentičnost sadržaja poruke (dokaz da poruka nije promenjena na putu od pošiljaoca do primaoca), kao i da obezbedi garantovanje identiteta pošiljaoca poruke. Pomoću svog potpisa korisnik ovlašćuje neku radnju i preuzima odgovornost za nju.

MD5 (*Message Digest Algorithm 5*) jeste heš funkcija koja se primenjuje u aplikacijama za digitalno potpisivanje dokumenata. Dužina sadržaja koji se formira na osnovu *MD5* funkcije je kratka (128 bita) što ga čini pogodnim za brzu proveru identiteta osoba koje šalju obimne dokumente. Algoritam *MD5* funkcije je razvio Ron Rivest (Ron Rivest) 1991. godine, kao zamenu za *MD4* algoritam. Nakon pet godina otkriveni su mali nedostaci u algoritmu, te su kriptografi preporučivali upotrebu drugih heš funkcija. Nekoliko narednih godina su otkriveni dodatni nedostaci pa je upotreba ovog algoritma dovedena u pitanje. Tokom 2005. godine grupa istraživača je uspela da formira isti sadržaj primenjujući *MD5* na dva različita dokumenta. Zbog pronađenih nedostataka, danas se ovaj algoritam sve ređe koristi za digitalno potpisivanje, ali je našao primenu u proveri integriteta fajlova, gde se koristi za izračunavanje kontrolnih suma, kod kojih sigurnost nije prioritetna.

MD5 algoritam kao ulaznu informaciju koristi *w*-bitni broj. Izvorni tekst se može prikazati kao niz brojeva:[18]

$$m_0, m_1, m_2,\ldots,m_{w-2}, m_{w-1}$$

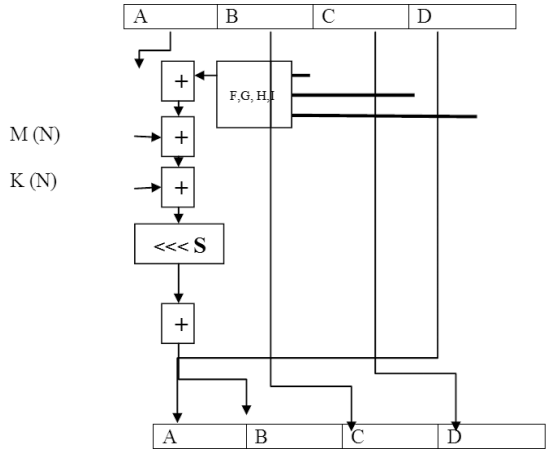
gde je broj *w*, vrednost iz proširenog skupa prirodnih brojeva. Na početku je potrebno izvršiti dopunu ulazne informacije do vrednosti koja se dobija od broja koji je celobrojni umnožak od 512 bita umanjenog za 64 bita. Na primer, ukoliko se izvorna poruka sastoji od 128 bitova ($w = 127$) potrebno je dopuniti je do 448 bitova, tj. 512 – 64 = 448. Dopuna se započinje sa početnim bitom „1“, a svi ostali bitovi za popunjavanje imaju vrednost „0“. Nakon dopune poruke, izvornoj poruci je potrebno dodati 64-bitnu reprezentaciju broja *w*. Ukoliko je dužina poruke veća i ne može da se predstavi pomoću 64 bita, poruci se dodaje samo nižih 64 bita. Dodavanjem ovih 64 bita dužina cele poruke postaje deljiva sa 512, odnosno deljiva sa 16 reči od 32- bita.

Sada se poruka može prikazati kao:

$$M[1,2,..., N]$$

Gdje je *N* broj deljiv sa 16. Ovako pripremljenu poruku algoritam kasnije koristi prilikom formiranja sadržaja. Na slici 8 je prikazan izgled jednog od moguća 64 koraka izvršenja *MD5* algoritma.

18 Dostupno na: http://en.wikipedia.org/wiki/ MD5 (2013)



Slika 8 -*Korak MD5 algoritma*[19]

Nakon predhodne pripreme poruke, potrebno je inicijalizovati 128-bitni bafer koji se sastoji od četiri 32-bitna registra A, B, C i D. Kao inicijalne vrednosti koje se upisuju u ove registre koriste se proizvoljne 32- bitne konstante.

Kada se završi inicijalizacija, pokreće se prvi korak MD5 algoritma. Ukupnih 64 koraka se deli u četiri ciklusa od po 16 koraka. Algoritam je formiran za izvršenje 512 bita poruke, što znači da ukoliko je poruka duža od 512 bita izvršenje algoritma se mora ponoviti. Algoritam se sastoji od četiri ciklusa koji imaju isti tok s tim što se prilikom izračunavanja u svakom od ciklusa koristi različita logička funkcija F, G, H i I. Funkcije se računaju po formulama:

F(X,Y,Z)=(X¬ AND Y) OR (NOT X AND Z)

G(X,Y,Z)=(X AND Z) OR (Y AND NOT Z)

H(X,Y,Z)=X XOR Y XOR Z

I(X,Y,Z)=Y XOR (X OR NOT Z)

Gde su AND, OR, NOT i XOR matematičke logičke operacije.

Tokom ciklusa se koriste operacije aritmetičkog sabiranja po modulu 2^{32} i operacija pomeranja ulevo za S pozicija, gdje je S vrednost različita za svaki ciklus. M[N] predstavlja 32-bitnu ulaznu poruku, a K[N] je konstanta koja je drugačija za svaki ciklus. Ukupno 16 M[N]-ova se koristi tokom 16 koraka u okviru svakog od 4 ciklusa. Rezultat jednog koraka se koristi kao početna vrijednost (*A*, *B*, *C* i *D*) narednog koraka. Na kraju se konačna vrednost formiranog sadržaja upisuje u registre *A*, *B*, *C* i *D*.

PRIMER 3:

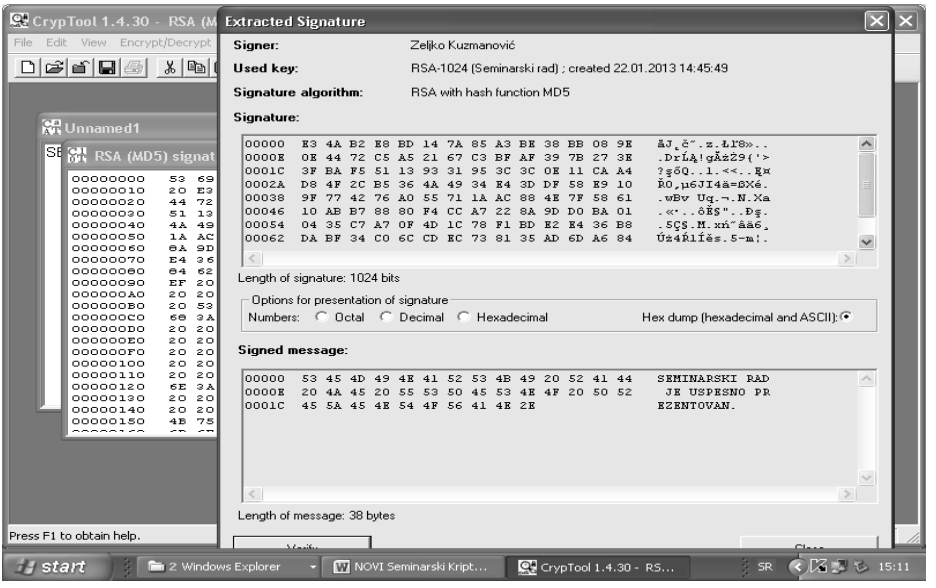
Startujemo program *CrypTool*. U glavnom meniju idemo na opciju *File*, i izaberemo *New* i otvorimo novi prozor, *Unnamed1*, u koji upisujemo neku našu poruku ili tekst koji želimo da zaštitimo (kriptujemo). Naša poruka neka glasi:

SEMINARSKI RAD JE USPEŠNO PREZENTOVAN.

19 V. Kovačević, *Zaštita podataka primenom kriptografskih metoda*, Seminarski rad Elektronski fakultet Niš, 2010.

Posle upisivanja poruke idemo na opciju *Digital Signatures/PKI*, pa zatim izaberemo opciju *Sign Document*. Otvara se novi prozor u kome moramo izabrati nekoliko stvari kao što su: heš funkcija sa kojom želimo da radimo, algoritam koji želimo da koristimo za digitalni potpis, ključ koji želimo da koristimo za potpis, kao i PIN kod za izabrani ključ. Što se tiče ključa, možemo iskoristiti prethodno generisani ključ koji je bio upotrebljen u primeru asimetrične enkripcije metodom RSA (PIN kod je bio 1111). Nakon što smo selektovali sve tražene opcije i upišemo i PIN kod, te pritiskom na taster *Sign*, otvara se novi prozor u kom se nalazi potpisani sažetak naše poruke, kao i vreme koje je bilo potrebno da se kreira taj sažetak.

Posle ovoga vršimo verifikaciju potpisanog sažetka poruke (potpisa) koji se inače obavlja na prijemnoj strani, tj. vrši se od strane osobe kojoj šaljemo poruku. U glavnom meniju idemo na opciju *Digital Signatures/PKI*, pa izaberemo opciju *Extract Signature*. Otvara se novi prozor u kome imamo osnovne podatke o osobi koja je potpisala poruku, ključu, korišćenom algoritmu, potpisanoj poruci kao i potpisu (slika 9):



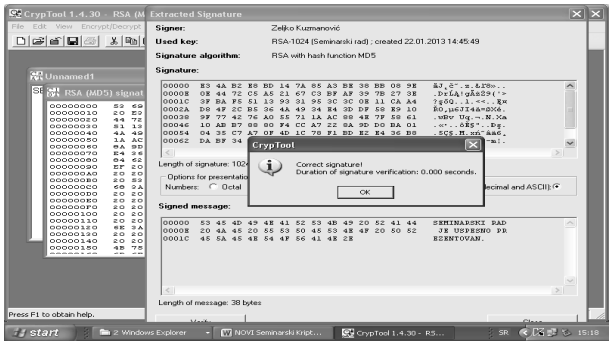
Slika 9 -*Izgled prozora potpisanog sažetka poruke*

Pritiskom na taster *Verify*, otvara se novi prozor *Signature Verification* u kome se traži da izaberemo potpis osobe koja je poslala prvobitnu poruku. Ta osoba je:

[Kuzmanovic][Zeljko][RSA-1024][1358862349][Seminarski rad]

Na prozoru pored ovoga imamo i određene informacije koje se odnose na postupak verifikacije digitalnog potpisa koji je u toku.

Pritiskom na taster *Verify signature*, izvršena je verifikacija potpisa, kao i osobe koja je potpisala sadržaj poruke koja je prikazana u slici 10:



Slika 10 -*Izgled prozora verifikacije digitalnog potpisa*

3. Softver CryptoWorkflow

CryptoWorkflow je jedan od alata za učenje složenosti i zamršenosti kriptografskih algoritama i pokriva rad samo nekoliko osnovnih algoritama (AES, DES, Viginere i XOR).

Može da radi na više operativnih sistema Windows, Linux/*BSD, OSX.

CryptoWorkflow[20] jeste aplikacija koja je dizajnirana da pomogne u učenju kriptografskih algoritama uvođenjem korisnika u interakciju. Korisnici *CryptoWorkflow* su u mogućnosti da dopune opis fajla kriptografskog algoritama, mogu da unesu ulazne vrednosti i posmatraju proces izvršavanja samog algoritma korak po korak.

Ovaj alat omogućava da prođemo korak po korak kroz sastavne delove u algoritmu, da vidimo tačno šta se dešava sa podacima u svakom trenutku. *CryptoWorkflow*, takođe omogućava da unosimo svoje prognoze za rezultate na svakom koraku u algoritmu i proverimo ispravnost našeg rada. Prilikom pokretanja procesa rada ili kretanjem kroz jednu operaciju možemo proveriti svoje odgovore i videti rad algoritma.

3.1. Instalacija i korišćenje *CryptoWorkflow*-a

Da bi se instalirao *CryptoWorkflow*, otvorite zip-fajl na lokaciji po vašem izboru.

Da biste pokrenuli program, postupite na jedan od sledećih načina:

Windows: Dvaput kliknite na izvršnu datoteku *cryptoworkflow.exe* koja se nalazi u glavnom direktorijumu programa.

Linux * BSD / OSX: Pokreni *Pyton cryptoworkflow.py* u glavnom direktorijumu, ili dvaput kliknite cryptoworkflow.py (u zavisnosti od konfiguracije vašeg sistema).

3.1.1. Korišćenje Cryptoworkflow

Cryptoworkflow je program koji vam omogućava da prođete kroz same operacije kriptografskih algoritama. Možete otvoriti *cwf* fajlove koji sadrže opise toka posla koji želite da koristite, i, opciono, popunite vaše prognoze za ishoda

20 Dostupno na: https://code.google.com/p/cryptoworkflow/ (2013)

svakog koraka u algoritmu. Tada ili pokrenete čitav tok posla, ili korak po korak, kroz jednu operaciju u jednom trenutku, da proverite svoje odgovore ili vidite rad algoritma.

3.1.2. Koncepcije rada

Postoje dve glavne koncepcije prisutne u drugim aplikacijama koje imaju uticajnu ideju za ovu aplikaciju.

Prvi je koncept vizuelnog rada – automatski rad aplikacije *Apple Computer* omogućava korisnicima da vizualizuju sekvencijalne operacije, obično izražene u programskom jeziku, kao serija podeljenih modula. Ovi moduli generalizuju određeni skup operacija dizajniranih da se ostvari zadatak.

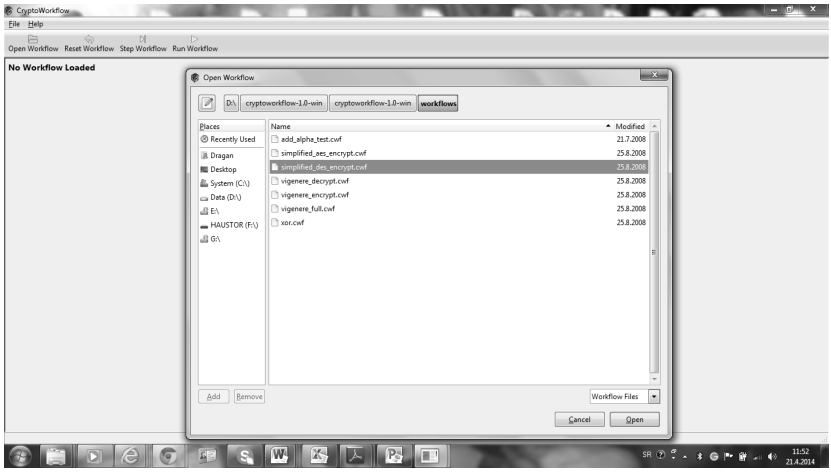
Drugi koncept je tačka prekida – tačka prekida je „mesto u kompjuterskom programu gde je prekinut niz uputstava. Kod nekih integrisanih razvojnih okruženja (IDEs), tačke prekida se koriste da se zaustavi i pregleda rad računarskog programa u konkretnom delu koda koji se izvršava.

3.2. Primer korišćenja *CryptoWorkflow*

Korišćenje i mogućnosti rada *CryptoWorkflow* su dati na primeru kriptovanja AES algoritmom.

Primer 4: Kriptovati DES algoritmom poruku 11111111 koristeći ključ 1010101010.

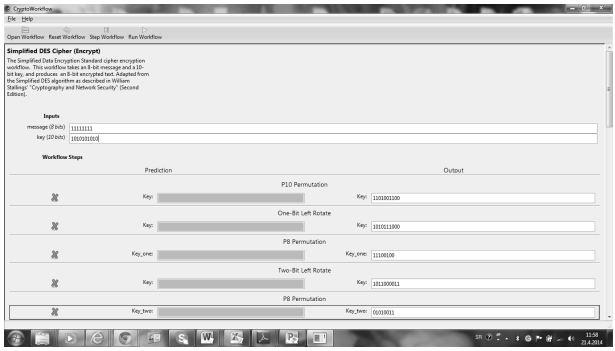
U prvom kpraku nakon otvaranja *CryptoWorkflow* i prozora izbora dijagrama toka, prema slici 11. biramo algoritam kojim ćemo izvršiti kriptovanje poruke – DES.



Slika 11: *Izgled prozora izbora aplikacije CryptoWorkflow*

Zatim unosimo željenu poruku i izabrani ključ u prozorima prema slici 12.

Nakon toga, možemo izabrati izvršenje algoritma u jednom koraku ili korak po korak radi boljeg praćenja, a po našoj želji i kontrole izvršenja algoritma, što ćemo mi izabrati, prema slici 13.



Slika 12: *Izbor DES algoritma poruke i ključa u aplikaciji CryptoWorkflow*

Slika 13: *Praćenje izvršenja DES algoritma u aplikaciji CryptoWorkflow*

Zaključak

U ovom radu je dat opis i primena kriptografskih algoritama koji se koriste u kriptografiji. U radu su opisane simetrična i asimetrična kriptografija i kriptografija heš algoritmom. Takođe, u radu su predstavljeni i programi *CrypTool i CryptoWorkflow*, čijom primenom su urađeni primeri koji ilustrativno prikazuju upotrebu osnovnih algoritama za kriptovanje.

Poređenjem ova dva softvera može se zaključiti da ovaj drugi daje dijagrame toka ali samo za pet u njemu datih algoritama i ima očigledno slabiji grafički korisnički interfejs.

Generalni zaključak bi mogao biti da je kriptografija veoma dinamična nauka koju odlikuje uska povezanost između teorije i prakse. Na ovo ukazuje činjenica da se napredak u teoriji brzo implementira u praksi, gde se još brže testira. Otkrivanjem nedostataka odmah se unapređuje teorijski rad i stečena iskustva se upotrebljavaju za izradu nove i bolje metode. Na složenost kriptografije dodatno ukazuju i činjenice da se kriptosistemi mogu realizovati hardverski, softverski ili hardversko-softverski, kao i to da se prilikom realizacije moraju zadovoljiti osnovni sigurnosni servisi. Zadovoljavajući rezultati su postignuti upotrebom kriptografskih algoritma koji su danas u upotrebi koji koriste složene

matematičke izraze kao i znanja iz elektronike i programiranja kojima se podaci maskiraju i tako bivaju osigurani za prenos. Uprkos tome, napredak u postojećim kriptografskim algoritmima nastavlja se rastućim tempom, da bi zadovoljio potrebe širenja informatičko-tehnološkog društva u kome živimo.

Literatura

1. Esslinger, B; CrypTool Development Team, *The CrypTool Script*, Frankfurt am Main, 2010.
2. Kovačević, V; *Zaštita podataka primenom kriptografskih metoda*, Seminarski rad Elektronski fakultet Niš, 2010.
3. Piper, F., Murphy, S; *Cryptography: A very short Introduction*, Oxford University press, Oxford, 2002.
4. Randjelović, D; Delija, D; Popović, B; EnCase forenzički alat, *Bezbednost*, 1-2, pp. 286–312, Beograd, 2009.
5. Ranđelović, D; *Visokotehnološki kriminal*, Kriminalističko-policijska akademija, Beograd, 2013.
6. Ruth, A; Hudson, K; *Sertificat Security+*, CET, Beograd, 2004.
7. Sinkovski, S; Lučić, B; *Informaciona bezbednost i kriptografija*, ZITEH, Beograd, 2006.
8. Stallings, W; *Cryptography and Network security*, Prentice Hall, 2003.
9. Van Tilborg, H; *Encyclopedia of Cryptography and Security*, University of Technology Eindhoven, New York, 2005.
10. http://en.wikipedia.org/wiki/Code. dostupan (16.1.2013)
11. http://en.wikipedia.org/wiki/CrypTool. dostupan (16.1.2013)
12. http://en.wikipedia.org/wiki/Triple_DES. dostupan (16.1.2013)
13. http://en.wikipedia.org/wiki/RSA. dostupan (16.1.2013)
14. http://en.wikipedia.org/wiki/MD5. dostupan (16.1.2013)
15. https://code.google.com/p/cryptoworkflow/ dostupan (16.1.2013)
16. https://code.google.com/p/cryptoworkflow/downloads/listdostupan (16.1.2013)

PRACTICAL APPLICATION OF OPEN SOURCE SOFTWARE TOOLS IN CRYPTOGRAPHY

Dragan Randjelovic
The Academy for Criminalistic and Police Studies, Belgrade
Milos Randjelovic
Help, Nis
Zeljko Kuzmanovic
City of Banja Luka, City Administration

Summary: Security of computer systems is becoming more important, because more users in more ways use more information in the computer world. With the development of information technologies and telecommunication systems and a greater diffusion in the geographical sense growing is also the possibility of misuse of data which are transferred through open and insecure communication paths which requires a more efficient protection. In the system of data transfer attackers can easily destroy data, modify them or the information can come into possession of unauthorized persons or organizations which can often have very heavy consequences. The problem is especially expressed in some organizations such as state administrations, banking, judicial, military, medical and other administrations. In this paper explained are basic terms related to crypts and algorithms which were used and which are used, as well as methods of data protection in computer networks based on cryptography in order to protect privacy inside a network of computers. Explained are symmetrical and asymmetrical cryptosystems and the technique of digital signature. Also, the subject of this paper is the presentation of the program CrypTool and CryptoWorkflow, which belong to the most famous free ie. shareware software in the field of cryptography, with special emphasis on the possibility of applying different algorithms and comparative analysis of their work and are given concrete examples of their use.

Keywords: cryptography, symmetric and asymmetric systems, digital signature, *CrypTool*, *CryptoWork* flow.