Sinteza
Međunarodna naučna konferencija Univerziteta Singidunum

Impact of Internet on Business Activities in Serbia and Worldwide
Uticaj Interneta na poslovanje u Srbiji i svetu

DOI: 10.15308/SINTEZA-2014-990-994

RAZVOJ SOPSTVENOG REŠENJA ZA KRIPTOGRAFSKU ZAŠTITU SA IMPLEMENTIRANIM MODULOM ZA GENERISANJE SIMETRIČNOG KRIPTOLOŠKOG KLJUČA

Marija Vujošević
ITLab

Abstract:
U ovom radu fokus je stavljen na razvoj sopstvenog rešenja za kriptografsku zaštitu fajlova upotrebom simetričnih šifarskih algoritama (DES, AES). U rešenju je implementiran sopstveni modul za generisanje kriptoloških ključeva, zasnovan na vrednostima dobijenim preko pokreta miša. Razvoj modula za generisanje i upravljanje ključevima podvrgnut je teorijsko-informacionoj analizi kakva se očekuje za primene ključeva u ovakve svrhe. Dobijeni rezultati predstavljeni su u uporednom prikazu sa uzorkom preuzetim sa Web stranice Random.org kojim se potvrđuje kvalitet dobijenih ključeva i neizostavne osobine TRNG-a. Deo algoritma koji se odnosi na upravljanje ključevima obezbeđuje siguran način skladištenja. Za mesta skladištenja ključeva predviđeni su hardverski tokeni sa kontrolom pristupa (USB token). Pored razvoja i implementacije navedenih kripto komponenti, stavljen je fokus i na programsku ergonomiju koja obezbeđuje pravilnu upotrebu programa u cilju smanjenja grešaka koje mogu da izazovu određene bezbednosne probleme na strani korisnika. Razvijeno rešenje podjednako se može koristiti kako u privatne tako i u poslovne svrhe.

Key words:
simetrični šifarski sistemi, generisanje simetričnih kriptoloških ključeva, pokreti miša.

UVOD

Informacije predstavljaju važan resurs u savremenom poslovanju. Bez obzira u kom se obliku čuvaju, moraju biti adekvatno zaštićene. Iz tog razloga zaštita informacija, očuvanje njene poverljivosti, integriteta i celovitosti je od presudne važnosti.

Istraživanjem dosadašnjeg stanja u ovoj oblasti i predstavljenih rešenja, došlo se do zaključka da je potrebno razviti rešenje za kriptografsku zaštitu fajlova i generisanje ključeva.

Osmišljeno rešenje je razvijeno u svrhu šifrovanja svih vrsta fajlova simetričnim šifarskim algoritmima. Korisnik ima mogućnost odabira nivoa sigurnosti koji želi da koristi i u skladu sa tim se primenjuje odgovarajući algoritam (DES, AES 128 ili 192), a samim tim i odgovarajuća dužina ključa. Implementiran je sopstveni modul za generisanje kriptoloških ključeva, zasnovan na vrednostima dobijenim preko pokreta miša. Proces generisanja ključa vrši se sakupljanjem koordinata na kojima je korisnik kliknuo mišem u okviru definisanog panela. Vreme potrebno za njegovo generisanje zavisi od odabranog nivoa sigurnosti. Nakon što je ključ generisan, od korisnika se zahteva unos lozinke kojom on dodatno štiti. Na ovaj način je implementirana dvofaktorska autentifikacija kombinovanjem nečega što korisnik ima i nečega što zna. Generisan ključ čuva se isključivo na hardverskom *tokenu* (USB *token*). Fajl koji se šifruje i izabrani ključ moraju se nalaziti na različitim lokacijama. Rešenje je razvijeno na Microsoft .NET platformi, u programskom jeziku C#.

Pažnja je stavljena i na programsku ergonomiju radi obezbeđivanja pravilne upotrebe programa kako bi se sprečile eventualne greške od strane korisnika i na taj način onemogućili potencijalni bezbednosni problemi.

U poslednjem delu izvršena je teorijsko-informaciona analiza u Matlab programskom paketu. Dobijeni rezultati predstavljeni su u uporednim testovima sa reprezentativnim uzorkom preuzetim sa Web stranice Random.org. Na osnovu ovih analiza, izveden je zaključak o dobijenim rezultatima.

PREGLED U OBLASTI ISTRAŽIVANJA

Tokom istraživanja ove oblasti, izdvojeno je nekoliko najkvalitetnijih rešenja na tržištu koja se bave ovom tematikom ([9], [10], [11], [12], [13] i [14]). Većina rešenja ([10], [11], [12] i [13]) za zaštitu fajlova koji se šifruju zahtevaju od korisnika unos lozinke za koje postoje preporuke od njenoj dužini, ali ne i sistemska ograničenja. Takođe [10], [11] i [13] koriste samo jedan šifarski algoritam, dok [12] i [14] nude korisniku opciju izbora algoritma koji žele da zaštite svoje podatke.

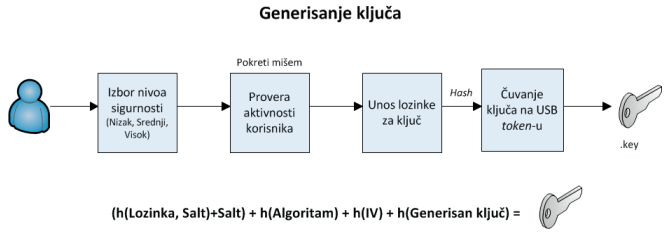
Od svih predstavljenih rešenja, [14] predstavlja najkompleksnije rešenje. Kao što je već napomenuto, omogućava odabir željenog algoritma. Za zaštitu fajlova koristi se ključ za čiji materijal se uzimaju pokreti miša, pritisci tastera na tastaturi i dr.

Sagledavajući sva ova rešenja, došlo se do zaključka da je potrebno ponuditi jedno sveobuhvatno rešenje koje će sa jedne strane omogućiti visok stepen zaštite, kao i lako i intuitivno korišćenje sa druge strane.

GENERIČKA ŠEMA PREDLOŽENOG REŠENJA

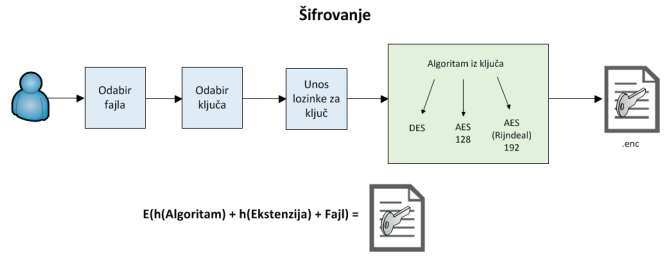
Razvijeno rešenje čine tri faze: generisanje ključa, šifrovanje i dešifrovanje. Parametri za generisanje ključa zavise od nivoa sigurnosti odabranog od strane korisnika. Nakon procesa generisanja ključa, korisnik unosi lozinku kojom se on štiti. Materijal za ključ čine lozinka, algoritam za šifrovanje (izabran na osnovu nivoa sigurnosti) i parametri dobijeni preko pokreta miša. Jedan deo tih parametara se koristi za inicijalni vektor, a drugi za sam ključ.

Korisnikova lozinka čuva se kao „posoljena“, pa hešovana vrednost. Zatim se vrši se konkatenacija heš vrednost algoritma, inicijalnog vektora i ključa. Kreirani ključ se čuva u fajlu sa ekstenzijom **.key** (Sl. 1) na izabranoj lokaciji od strane korisnika (na USB *tokenu*).



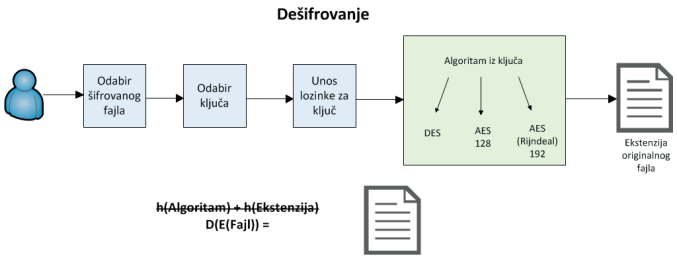
Sl. 1 - Šema aplikacije - Generisanje ključa

U procesu šifrovanja nakon uspešne validacije unete lozinke od strane korisnika, iz ključa se uzima algoritam kojim se šifruje fajl. Šifrovani fajl sadrži heš vrednosti algoritma i njegove originalne ekstenzije i naravno, sam fajl. Nakon šifrovanja, originalni fajl se briše, a šifrovani dobija ekstenziju **.enc** (Sl. 2).



Sl. 2 - Šema aplikacije - Šifrovanje

Kao i prilikom šifrovanja, i prilikom dešifrovanja od korisnika se zahteva da, nakon izbora fajla koji želi da dešifruje i odgovarajućeg ključa, unese lozinku. Nakon uspešne validacije, proverava se da li algoritam iz ključa odgovara onom iz šifrovanog fajla. Potom se iz fajla uklanjaju podaci o algoritmu i ekstenziji i vrši se dešifrovanje. Podatak o ekstenziji se koristi za vraćanje originalne ekstenzije fajla (Sl. 3).



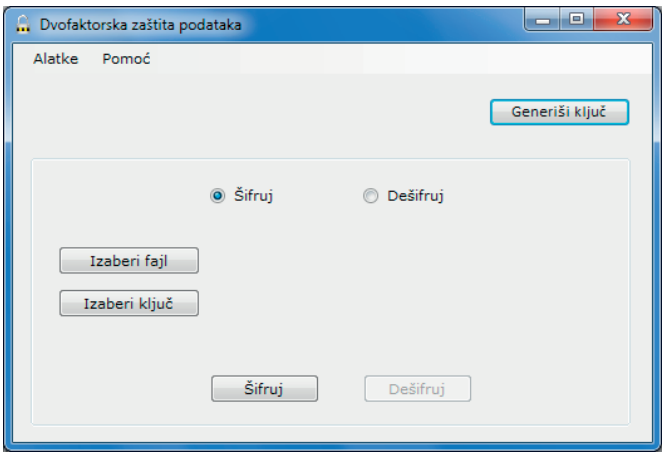
Sl. 3 - Šema aplikacije - Dešifrovanje

Postavka i objašnjenje eksperimentalnog okruženja

Microsoft .NET platforma (.NET Framework 4.0) i programski jezik C# korišćeni su za razvoj rešenja, dok je teorijsko-informaciona analiza izvršena u Matlab programskom paketu.

Svrha razvijenog rešenja predstavlja šifrovanje svih vrsta fajlova simetričnim šifarskim algoritmima korišćenjem ključa koji je generisan na slučajan način pokretima miša. Dobijeni ključ predstavlja vrstu TRNG i podvrgnut je teorijsko-informacionoj analizi.

Rešenje je osmišljeno na način da korisniku omogućava lak i intuitivan rad. Na glavnom panelu, korisnik ima mogućnost odabira režima rada: generisanje ključeva, šifrovanje ili dešifrovanje (Sl. 4).

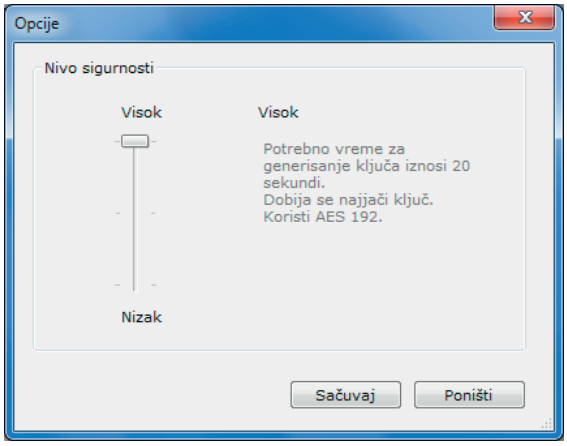


Sl. 4 - Početni panel razvijenog rešenja

Nakon instalacije i inicijalnog pokretanja aplikacije, prvenstveno je neophodno generisanje barem jednog ključa kojim se potom može vršiti šifrovanje fajlova. Pre samog generisanja istog, korisnik mora podesiti željeni nivo sigurnosti koji će se koristiti kako za generisanje ključeva (vreme potrebno za njegovo generisanje) tako i za samo šifrovanje fajlova.

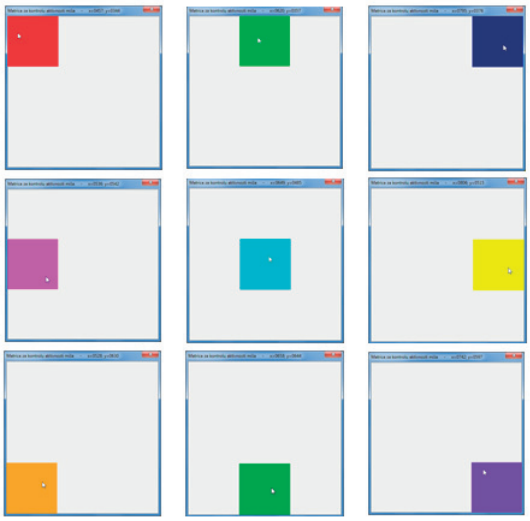
Definisana su tri nivoa sigurnosti: nizak, srednji i visok. Odabirom niskog nivoa koristiće se DES šifarski algoritam. Srednji nivo podrazumeva AES 128 (dužina bloka 128 bita, dužina ključa 128 bita). Na visokom nivou sigurnosti koristiće se AES 192 (*Rijndael*) (dužina bloka 192 bita, dužina ključa 192 bita).

Prilikom odabira određenog nivoa, korisnik dobija i kraće objašnjenje šta koji od nivoa znači kako bi mu bio olakšan odabir (Sl. 5). Heš funkcija SHA-256 koristi se za sve nivoe sigurnosti.



Sl. 5 - Odabir nivoa sigurnosti

Sledeći korak predstavlja generisanje ključa. Pre samog početka korisnik se obaveštava o akcijama koje je potrebno da preduzme tokom procesa generisanja. Nakon toga započinje proces. Od korisnika se zahteva da klikne mišem na kvadrat koji je označen bojom. Nakon svakog klika, prikazuje se sledeći slučajno izabran kvadrat. Tokom ovog perioda, skupljaju se koordinate miša prilikom klika (Sl. 6). Ukoliko korisnik nije bio aktivan barem tri sekunde, ukupno vreme potrebno za generisanje se za toliko produžava. Tokom generisanja, materijal za ključ se čuva u memoriji.

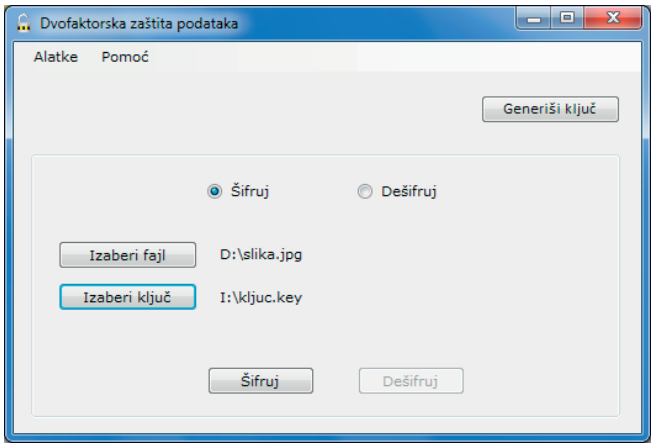


Sl. 6 - Proces generisanja ključa

Nakon završetka, korisnik se obaveštava da je ključ generisan.

U narednom koraku se od njega zahteva da unese lozinku kojom će se štititi taj ključ. Minimalna dužina mora biti 6 karaktera. Na ovaj način u procesu šifrovanja ili dešifrovanja, koristi se dvofaktorska autentifikacija upotrebom nečega što korisnik ima (ključ) i nečega što zna (lozinka). Skladištenje ključa moguće je jedino na harverskom *tokenu* (USB *token*). Čuvanje na čvrstom disku nije moguće.

Aplikacija omogućava generisanje neograničenog broja ključeva za sva tri nivoa sigurnosti. Kada poseduje ključeve, korisnik ih može koristiti za šifrovanje ili dešifrovanje odabirom željenog fajla i ključa. Fajl koji se šifruje može se nalaziti na bilo kojoj lokaciji osim one na kojoj se nalazi i ključ (Sl. 7). Pre samog šifrovanja vrši se verfikacija ključa putem lozinke koja je unesena prilikom njegovog generisanja.



Sl. 7 - Proces šifrovanja željenog fajla upotrebom generisanog ključa

Nakon šifrovanja, originalni fajl se briše, a novi šifrovani fajl dobija ekstenziju **.enc** i čuva se na istoj lokaciji na kojoj je bio i originalni fajl.

Performanse predloženog okruženja sa prikazom eksperimentalnih rezultata

Jaka informaciona analiza generisanog ključa je od velike važnosti iz razloga što postavlja teorijske okvire za utvrđivanje jačine dobijenog kriptološkog ključa. Korišćenjem Šenonove entropije dolazi se do prosečne količine informacija koje su sadržane u dobijenom ključu. Razvoj modula za generisanje i upravljanje ključevima podvrgnut je teorijsko-informacionoj analizi kakva se očekuje za primene ključeva u ovakve svrhe. Dobijeni rezultati predstavljeni su u uporednom prikazu sa uzorkom preuzetim sa Web stranice Random.org kojim se potvrđuje kvalitet dobijenih ključeva i neizostavne osobine TRNG-a.

Nad generisanim ključevima sprovedeni su testovi za procenu informacionog sadržaja. U nastavku će biti prikazani uporedni testovi slučajnih binarnih nizova iz generisanih ključeva i slučajnih binarnih nizova generisanih iz atmosferskog šuma koju su preuzeti sa Web sajta Random.org. Za ovo rešenje najvažniji su serijski test i ispitivanje entropije preklapajućih i nepreklapajućih uzoraka. Dobijeni rezultati prikazani su u tabelama III.1, III.2, III.3 i III.4.

Tabela III.1 – Serijski test - Bigrami

Tip testa	Serijski test	
	Bigrami	
	random.org	generisan ključ
00	7773	7471
01	7828	8299
10	7827	8300
11	7821	9294

Tabela III.2 – Serijski test - Trigrami

Tip testa	Serijski test	
	Trigrami	
	random.org	generisan ključ
000	3818	3537
001	3955	3935
010	3863	4172
011	3965	4127
100	3955	3936
101	3873	4364
110	3965	4127
111	3856	5167

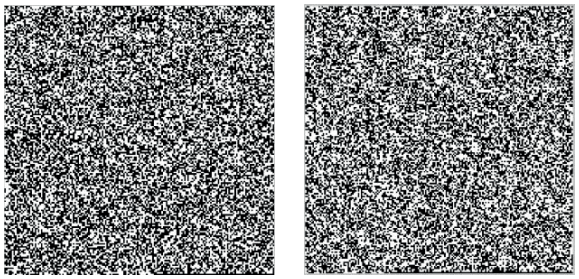
Tabela III.3 – Entropija sa preklapanjem

Tip testa	Entropija sa preklapanjem	
	random.org	generisan ključ
Mono-bit	0.9999982981270112	0.9978454731456681
Bigram	0.9999969195844207	0.9978439065269784
Trigram	0.9999512260990479	0.9974066948986483
Matrica 4x4	0.9999630196341835	0.997744280937179

Tabela III.4 – Entropija bez preklapanja

Tip testa	Entropija bez preklapanja	
	random.org	generisan ključ
Mono-bit	0.9999982981270112	0.9978454731456681
Bigram	0.9999443575006272	0.9978439065269784
Trigram	0.9999343024266536	0.9974066948986483
Matrica 4x4	0.9999630196341835	0.997744280937179

U nastavku sledi vizuelizacija oba slučajna niza čime se potvrđuje da generisani ključ zapravo predstavlja TRNG.



Sl. 8 - random.org (levo) i generisani ključ (desno)

Na slici Sl. 8, na levoj strani prikazan je šum generisan iz atmosferskog šuma, dok se na desnoj strani nalazi generisan ključ.

ZAKLJUČAK

U ovom radu predstavljeno je praktično realizovano sopstveno rešenje za kriptografsku zaštitu svih vrsta fajlova uz implementirani modul za generisanje kriptološkog ključa preko pokreta miša. Oni su uzeti kao materijal za ključ zbog potrebe da se omogući što veća slučajnost, a samim tim i entropija. Cilj je bio da se postigne pravi generator slučajnih brojeva (TRNG). Takođe je omogućeno generisanje neograničenog broja ključeva uz odabir određenog nivoa sigurnosti. Uvedena je dvofaktorksa autentifikacija prilikom šifrovanja odnosno dešifrovanja fajlova. Generisani ključevi se dodatno štite lozinkom. Dobijeni ključevi se skladište isključivo na hardverskom *tokenu* (USB *token*). Dosta pažnje je posvećeno i ergonomiji aplikacije kako bi se obezbedila laka i pravilna upotreba programa i samim tim sprečile eventualne greške od strane korisnika i onemogućili potencijalni bezbednosni problemi.

Generisani ključevi su podvrgnuti teorijsko-informacionoj analizi u eksperimentalnom okruženju kojom je potvrđeno da je postignuta željena slučajnost.

Primarna ideja u daljem radu je razvoj klijent-serverskog rešenja u kome bi se ključevi čuvali u bazi podataka na serveru, dok bi se komunikacija štitila SSL-om.

LITERATURA

[1] M. Stamp, "Information security: principles and practice", 2nd ur., New Jersey: John Wiley & Soons, 2011.

[2] B. Schneier, "Applied Cryptography: Protocols, Algorthms, and Source Code in C", 2nd ur., New Jersey: John Wiley & Sons, 1996.

[3] C. Paar, J. Pelzl, "Understanding cryptography", Verlag, Berlin, Heidelberg: Springer, 2010.

[4] M. Veinović, S. Adamović, "Kriptologija 1", Beograd: Univerzitet Singidunum, 2013.

[5] „Specification for the AES," NIST, 2001. http://www.nist.gov/CryptoToolkit

[6] A. Jagannatam, „Mersenne Twister – A Pseudo Random Number Generator," George Mason University, Department of Electrical and Computer Engineering, 2008.

[7] G. Srivastava, "Pseudorandom number generator using multiple sources of entropy", University of Victoria, 2006.

[8] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray / S. Vo, „A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications," NIST, Gaithersburg, MD, 2010.

[9] ElcomSoft, „Advantages and disadvantages of EFS and effective recovery of encrypted data," ElcomSoft Co. Ltd., Moscow, 2007.

[10] „AES Crypt," http://www.aescrypt.com

[11] „AxCrypt," http://www.axantum.com/axcrypt

[12] „CryptoForge," http://www.cryptoforge.com

[13] „Folder lock," http://www.newsoftwares.net/folderlock

[14] „TrueCrypt," http://www.truecrypt.org

[15] „Random.org," http://www.random.org

DEVELOPMENT OF CRYPTOGRAPHIC PROTECTION SOLUTION WITH IMPLEMENTED SYMMETRIC KEY GENERATION MODULE

Abstract:
In this paper the focus is placed on the development of own solution for cryptographic file protection using symmetric algorithms (DES, AES). In the application it has been implemented module for cipher keys generation, based on values obtained via mouse movement. Development of the module for generating and managing keys subjected to theoretical-information analysis which is expected for keys usages for such purposes. The results are shown in side view with the sample taken from Web page Random.org confirming the quality of the generated keys and the necessary characteristics of TRNG. A part of the algorithm relates to providing secure key management method of storage. Hardware token with access control (USB token) are used for key storage. In addition to the development and implementation of these cryptographic components, the focus is also placed on software ergonomics that ensures proper use of the application in order to reduce errors that can cause some security problems on the user side. Developed solution can be used in both private and business purposes.

Key words:
symmetric cipher systems, symmetric key generation, mouse movement.