Bezbednosni mehanizam za prenos i skladištenje podataka u oblaku

Security mechanism for cloud end-to-end encryption

Marija Vujošević, AllTele AB, Mladen Veinović, Univerzitet Singidunum

***Sažetak* — U ovom radu proučava se oblast zaštite podataka na udaljenim serverima u oblaku i tokom prenosa komunikacionim kanalima. Vrši se pregled najkvalitetnijih komercijalnih rešenja iz ove oblasti. Pažnja će biti posvećena onim rešenjima koja imaju implementiranu *end-to-end* zaštitu. Fokus će biti stavljen na razvoj sopstvenog rešenja u kome će biti implementiran predstavljeni bezbednosni mehanizam uz modul za generisanje kriptoloških ključeva.**

Ključne reči – end-to-end zaštita, skladištenje podataka u oblaku, zaštita podataka, simetrični i asimetrični šifarski sistemi, generatori slučajnih brojeva, autentifikacija

Abstract – ***In this paper we study the Cloud computing data security. Some of the most quality commercial solutions that have implemented end-to-end encryption will be explined.The focus will be set on own solutions development in which this security mechanism will be implemented including a symmetric key generation module.***

Keywords – end-to-end encryption, Cloud computing, data security, symmetric and public-key cryptography, random number generator, authentication

I. 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.

U današnje vreme, kada većina korisnika brine o bezbednosti svojih podataka, potrebno je naći metode i rešenja koja će omogućiti visok stepen zaštite. Iz tog razloga predmet istraživanja u ovom radu predstavlja zaštitu podataka pre njihovog smeštanja na udaljeni server (u oblaku), ali i zaštitu komunikacionih kanala.

Metoda slanja podataka na server u otvorenom tekstu, a zatim šifrovanje i smeštanja istih, ostavlja mogućnost serveru da pročita podatke pre šifrovanja. Na ovaj način nivo bezbednosti opada. Iz tog razloga uvodi se novi princip kojim se to sprečava. Reč je o *end-to-end* zaštiti koja podrazumeva šifrovanje fajlova na računaru korisnika pre slanja na server. Ovim se postiže viši nivo sigurnosti time što server ne može doći do sadržaja fajlova koji se na njemu čuvaju.

Tokom istraživanja izvršen je pregled nekih od najkvalitetnijih komercijalnih rešenja. U eksperimentalnom okruženju ona su podvrgnuta određenim testovima kako bi se sagledale njihove karakteristika i uočile prednosti i nedostaci. Svi ti podaci uzeti su u obzir prilikom razvoja sopstvenog rešenja koje će biti predstavljeno u ovom radu.

Rad je namenjen svima koji brinu o bezbednosti svojih podataka i podjednako se odnosi na privatne i poslovne korisnike.

Naučni cilj ovog rada predstavlja istraživanje rešenja koja nude skladištenje korisničkih podataka u oblaku vršeći njihovu sinhronizaciju sa lokalnih računara. Fokus je stavljen na ona rešenja koja omogućavaju njihovo šifrovanje pre slanja putem komunikacionog kanala.

II. PREGLED U OBLASTI ISTRAŽIVANJA

Istraživanjem oblasti skladištenja podataka u oblaku, odabrana su neka od najkvalitetnijih rešenja na tržištu kako bi se sagledao njihov način funkcionisanja koji će u nastavku biti predstavljen.

Tresorit [6] je *on line* servis za skladištenje podataka u oblaku. Usklađen je sa najstrožim bezbednosnim propisima i koristi isključivo industrijski standardizovane algoritme. Za šifrovanje fajlova primenjuje AES-256 simetrični šifarski algoritam na strani klijenta dok se oni još uvek nalaze na lokalnom računaru i tek nakon toga ih šalje u oblak. Na ovaj način se onemogućava pristup podacima od strane servera. Transakcije se autentifikuju pomoću RSA-4096. Fajlovi su takođe zaštićeni HMAC autentifikacijom koja se primenjuje na SHA-512 heševe. Šifrovani fajlovi se smeštaju u oblak preko TLS zaštićenih komunikacionih kanala čime se garantuje poverljivost podataka. Sistem za upravljanje ključevima kreiran je tako da dokumentima može pristupiti samo onaj kome je taj dokument namenjen. Kada korisnik modifikuje dokument on se automatski ponovo šifruje novim ključem. Važna karakteristika je ta da se ključ kojim se šifruju podaci nalazi jedino kod korisnika aplikacije.

SpiderOak [7] predstavlja *on line* alat za skladištenje podataka, njihovo deljenje, sinhronizaciju i pristup. Korisnik svoje podatke štiti lozinkom koja se nikada ne prenosi na server u svom izvornom obliku (otvorenom tekstu). U slučaju da

DOI: 10.7251/ZRSNG1501001V

zaboravi svoju lozinku ne postoji način da se ti podaci povrate tj. dešifruju. Prilikom inicijalnog pokretanja aplikacije, na računaru se generiše serija jakih ključeva. Oni se potom šifruju korisničkom lozinkom i tako šifrovani skladište se na server. Generisanje ključa se vrši pomoću JavaScript-a. Nakon što se korisnik registruje na sajtu, JavaScript kôd pravi heš vrednost lozinke pomoću *bcript* algoritma pre slanja na server. Nakon inicijalnog pokretanja aplikacije od korisnika se traži da unese korisničko ime i lozinku. Originalna lozinka se ne čuva nigde. Ovo predstavlja neophodne korake ka kreiranju pravog *zero-knowledge* okruženja. Upotrebom *ovog* sistema privatnosti korisnici ne moraju da brinu o tome da li je kompanija koja čuva njihove podatke od poverenja. SpiderOak koristi AES-256 u CFB modu i HMAC-SHA256.

U TeamDrive [8] rešenju, fajlovi se skladište u Prostor (engl. *Space*). Svaki Prostor ima svoj 256-bitni AES ključ. Ključ, koji se nalazi lokalno na klijentskom računaru i poznat je samo njemu, koristi se da omogući pristup i šifrovanje fajlova u Prostoru pre nego što oni napuste računar krajnjeg korisnika. Svaki Prostor može imati veći broj krajnjih korisnika ili članova koji imaju pristup fajlovima koji se u njemu nalaze. Ovo rešenje kreira par ključeva (RSA-3072) koji se koriste za siguran prenos pozivnica između korisnika za deljenje zajedničkih fajlova. Javni ključ se čuva na centralnom serveru kojim upravlja TeamDrive. Fajlovi se šifruju na računaru korisnika pre nego što budu poslati na server, korišćenjem AES-256 algoritama. Kada korisnik pozove novog člana grupe, simetrični ključ foldera koji se deli šifruje se javnim ključem člana grupe kome je poziv upućen i potom mu se šalje zahtev. Član grupe kome je zahtev upućen dešifruje simetrični ključ svojim privatnim ključem i nakon toga dobija pristup deljenom folderu. Ovaj postupak osigurava da samo ovlašćeni članovi grupe imaju pristup podacima.

Jedno od najpopularnijih rešenja za skladištenje podataka u oblaku u svetu sigurno je Dropbox [10] (pored Google Drive i Microsoft OneDrive). Za razliku od do sada predstavljenih rešenja, on ne nudi zaštitu podataka na strani korisnika, ali postoji mogućnost da bude razvijena u budućnosti. Što se njegovih sigurnosnih performansi tiče, one se odnose na upotrebu AES-256 šifarskog algoritma i SSL protokola za zaštitu komunikacionih kanala. [2] Takođe, korisnicima se nudi i dodatan nivo sigurnosti upotrebom dvofaktorske autentifikacije koja može biti realizovana upotrebom aplikacije za generisanje kôda na mobilnom telefonu (engl. *Time-Based One-Time Password*) ili slanjem kôda putem tekstualnih poruka (SMS). Pre same uspostave dvofaktorske autentifikacije, korisnik dobija 16-cifarni rezervni kôd koji će moći da iskoristi ukoliko izgubi svoj mobilni telefon.

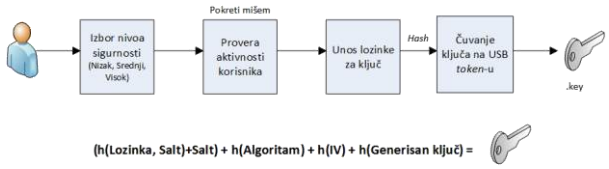
III. PREGLED PREDLOŽENOG REŠENJA

A. *Generička šema predloženog rešenja*

Osnovna ideja pri osmišljavanju ovog rešenja, a zatim i njegovoj realizaciji, bila je kreiranje rešenja koje će korisnicima omogućiti bezbedan, a s druge strane lak i intuitivan način skladištenja svojih podataka na udaljeni server (u oblak). Za razumevanje njegovog funkcionisanja neophodno je napraviti generičku šemu.

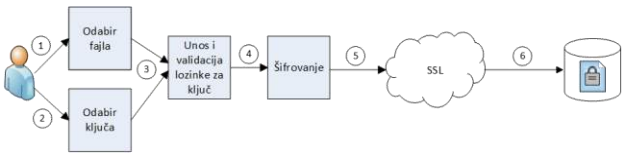
Rešenje čine tri faze: generisanje ključa, šifrovanje i dešifrovanje. Parametri za generisanje ključa zavise od nivoa sigurnosti odabranog od strane korisnika.

Proces generisanja ključa započinje odabirom nivoa sigurnosti od koga zavisi vreme koje će biti potrebno da se ključ generiše. Nakon toga, u panelu predviđenom za to, prikupljaju se parametri tj. koordinate miša koji se koriste kao materijal za ključ. Po završetku generisanja, korisnik unosi lozinku kojom se on štiti. Materijal za ključ čine lozinka, algoritam za šifrovanje (izabran na osnovu nivoa sigurnosti) i parametri dobijeni preko miša. Jedan deo tih parametara se koristi za inicijalni vektor, a drugi za sam ključ. Korisnikova lozinka se čuva kao „posoljena“, pa hešovana vrednost. Zatim se vrši konkatenacija heš vrednosti algoritma, inicijalnog vektora i ključa. Nakon toga, kreira se fajl sa ekstenzijom **.key** (Sl. 1) i čuva na izabranoj lokaciji od strane korisnika (na USB *tokenu*). [4]



Sl. 1. Šema aplikacije - Generisanje ključa

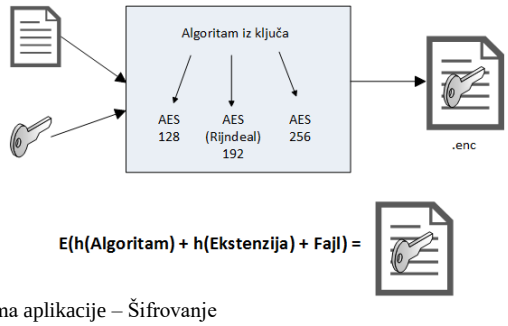
Nakon što je generisan (barem jedan) ključ, započinje proces šifrovanja (Sl. 2). Po odabiru fajla koji će se šifrovati i ključa, najpre se vrši validacija lozinke kojom je zaštićen, kako bi se onemogućila njegova upotreba od strane neovlašćenih korisnika. Nakon uspešne validacije započinje proces šifrovanja (detaljno će biti opisan u nastavku). Tako šifrovan fajl se, preko zaštićene SSL konekcije [2], šalje i smešta u bazu podataka na udaljenom serveru. Ključ se ni u jednom trenutku ne šalje na server, uvek ostaje kod korisnika, smešten na njegovom USB *tokenu*.



Sl. 2. Šema aplikacije – Proces šifrovanja i bezbednog skladištenja fajla na udaljeni server (u oblak)

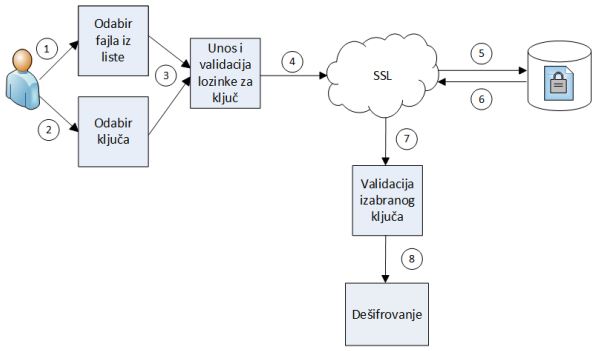
U procesu šifrovanja nakon uspešne validacije unete lozinke od strane korisnika, iz ključa se uzima algoritam kojim se fajl šifruje. Šifrovani fajl sadrži heš vrednosti algoritma i njegove originalne ekstenzije i naravno, sam fajl (Sl. 3).

DOI: 10.7251/ZRSNG1501001V



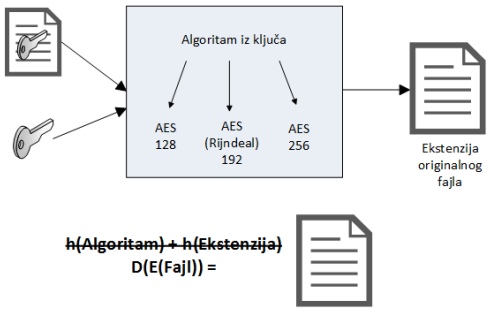
Sl. 3. Šema aplikacije – Šifrovanje

U fazi dešifrovanja potrebno je izabrati iz liste fajl sa servera koji se želi dešifrovati, kao i odgovarajući ključ. Funkcija *end-to-end* zaštite je ta da se fajlovi dešifruju na lokalnom računaru korisnika, ne na serveru. Iz tog razloga, odabrani šifrovani fajl se preko zaštićene SSL veze [2] prenosi na računar korisnika, gde se potom vrši validacija izabranog ključa i nakon toga se izabrani fajl dešifruje (Sl. 4).



Sl. 4. Šema aplikacije – Proces preuzimanja šifrovanog fajla sa udaljenog računara (iz oblaka) i njegovo dešifrovanje

Prilikom dešifrovanja, kao i prilikom šifrovanja, od korisnika se zahteva da, nakon što izabere fajl koji želi da dešifruje i odgovarajući ključ, unese lozinku. Ukoliko je ona ispravna, proverava se da li algoritam iz ključa odgovara onom iz šifrovanog fajla. Potom se iz fajla uklanjaju podaci o algoritmu i ekstenziji i on se dešifruje. Podatak o ekstenziji se koristi za vraćanje originalne ekstenzije fajla (Sl. 5).



Sl. 5. Šema aplikacije – Dešifrovanje

Microsoft SQL Server 2012 korišćen je kao server baze podataka. Bazu čine tri tabele: *User*, *EncryptData* i *ActivityLog* (Sl. 6). Tabela *User* čuva podatke o registrovanim korisnicima. Pored osnovnih podataka, čuvaju se datum i vreme registrovanja korisnika u sistem, kao i datum i vreme njegovog poslednjeg prijavljivanja. Šifrovani fajlovi skladište se u *EncryptData* tabeli u binarnom formatu. U njoj se još čuvaju i podaci o nazivu fajla, njegovoj ekstenziji kao i naziv fajla koji će biti prikazan korisniku u fazi dešifrovanja. *ActivityLog* tabela čuva podatke o aktivnosti korisnika u sistemu, odnosno o tome kada je korisnik kreirao određeni fajl i kada mu je poslednji put pristupio tj. dešifrovao ga.

Sl. 6. Dijagram baze podataka

B. *Postavka i objašnjenje eksperimentalnog okruženja*

Razvijeno rešenje čini klijent-server arhitektura. Celokupno rešenje razvijeno je na Microsoft platformi. Na serverskoj strani, za skladištenje šifrovanih podataka, koristi se SQL Server 2012 baza podataka, dok je za komunikaciju sa klijentom koristi implementiran WCF (engl. *Windows Communication Foundation*) servis. Klijenta aplikacija razvijena je u programskom jeziku C# (Microsoft .NET *Framework* 4.0). U testnom okruženju korišćena je aplikacija namenjena za te svrhe, dok je Matlab R2012b kao okruženje za grafički prikaz dobijenih rezultata.

Rešenje je razvijeno sa ciljem da korisniku omogući siguran i lak način za skladištenje svojih podataka na udaljenom serveru. To je postignuto implementacijom *end-to-end* zaštite koja podrazumeva šifrovanje fajlova na strani korisnika i njegovo skladištenje u bazi podataka na serveru. Šifrovani fajlovi se smeštaju u oblak preko SSL zaštićenog komunikacionog kanala. [2] Na ovaj način se garantuje poverljivost podataka.

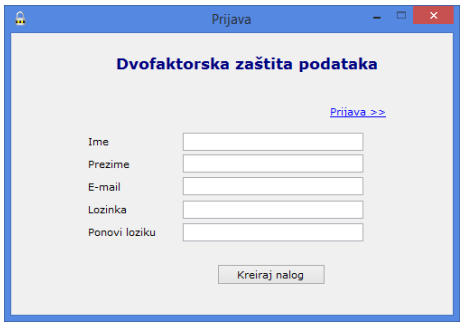
Šifrovanje svih vrsta fajlova vrši se simetričnim šifarskim algoritmima korišćenjem ključa koji je generisan na slučajan način pokretima miša. Dobijeni ključ predstavlja vrstu TRNG [1] i podvrgnut je teorijsko-informacionoj analizi.

Kako bi se ostvarila sigurna komunikacija, neophodno je kreiranje sertifikata (serverskog i klijentskog). Za potrebe

DOI: 10.7251/ZRSNG1501001V

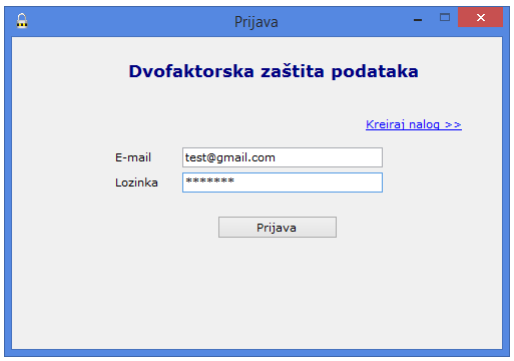
razvoja i testiranja aplikacije, generisani su *self-signed* sertifikati (nisu potpisani od strane vrhovnog CA tela).

Prvi korak nakon instalacije klijentske aplikacije jeste registrovanje korisnika na sistem (Sl. 7). Od njega se traži da unese svoje podatke i lozinku. Oni se čuvaju u bazi podataka, a lozinka se skladišti kao hešovana vrednost (SHA-256). Minimalna dužina lozinke je 6 karaktera.



Sl. 7. Kreiranje korisničkog naloga

Nakon što se korisnik jedanput registrovao, svakim narednim pokretanjem aplikacije, od njega će biti zatraženo prijavljivanje na sistem unošenjem *e-mail*-a i lozinke kako bi se izvršila autentifikacija korisnika (vrši se validacija heša lozinke) (Sl. 8).



Sl. 8. Prijavljivanje korisnika na sistem

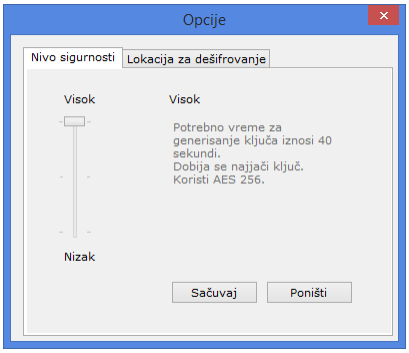
Da bi upotreba aplikacije bila moguća, neophodno je generisanje barem jednog ključa kojim se može vršiti šifrovanje fajlova. Pre samog generisanja istog, potrebno je iz menija *Alatke* odabrati opciju *Opcije* koja omogućava odabir nivoa sigurnosti koji će se koristiti kako za generisanje ključeva (vreme potrebno za njegovo generisanje) tako i za samo šifrovanje fajlova.

Definisana su tri nivoa sigurnosti:

- Nizak
- Srednji
- Visok

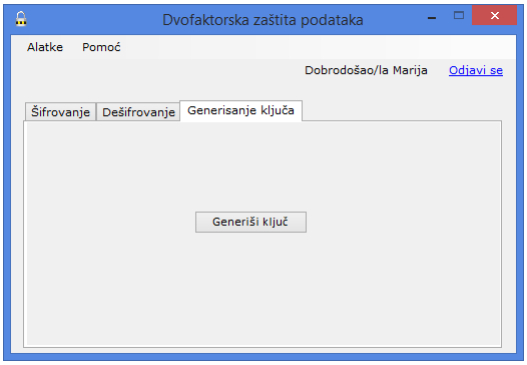
Nizak nivo koristi AES 128 šifarski algoritam, srednji AES 192 (*Rijndael*), dok visoki podrazumeva upotrebu AES 256. [4]

Prilikom odabira određenog nivoa, korisnik dobija i kraće objašnjenje šta koji od nivoa znači kako bi mu bio olakšan odabir (Sl. 9). Heš funkcija SHA-256 se koristi za sve nivoe sigurnosti. [4]



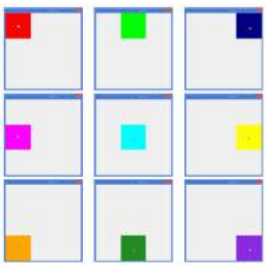
Sl. 9. Odabir nivoa sigurnosti

Glavni panel razvijenog rešenja podeljen je u tri sekcije: generisanje ključa, šifrovanje i dešifrovanje. Iz sekcije *Generisanje ključa*, odabirom dugmeta *Generiši ključ* započinje proces generisanje (Sl. 10).



Sl. 10. Faza generisanja ključa

Nakon svih neophodnih podešavanja započinje proces generisanja. Od korisnika se zahteva da klikom mišem odabere označeni kvadrat. Nakon svakog klika, prikazuje se sledeći slučajno izabran kvadrat. Tokom ovog perioda, skupljaju se koordinate miša prilikom klika (Sl. 11). Ukoliko korisnik nije bio aktivan barem tri sekunde, ukupno vreme potrebno za generisanje se za toliko uvećava. Tokom generisanja, materijal za ključ se čuva u memoriji. [4]



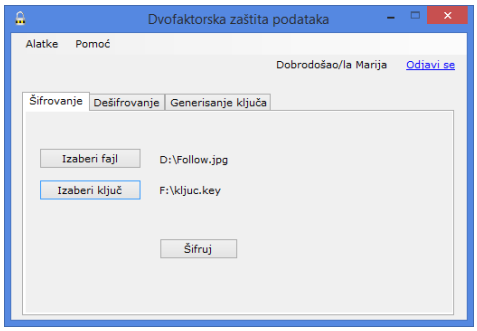
Sl. 11. Proces generisanja ključa

DOI: 10.7251/ZRSNG1501001V

U narednom koraku zahteva se unos lozinke kojom će se štititi taj ključ. Minimalna dužina mora biti 6 karaktera. Na ovaj način u procesu šifrovanja ili dešifrovanja, korisnik će morati da koristi dvofaktorsku autentifikaciju koristeći nešto što ima (ključ) i nešto što zna (lozinka). Sledeći korak je čuvanje ključa na odabranom harverskom *tokenu* (USB *token*). Čuvanje na čvrstom disku nije moguće. Ključ se nalazi isključivo kod klijenta i ni u kom slučaju se ne šalje na server. Rešenje omogućava generisanje neograničenog broja ključeva za sva tri nivoa sigurnosti. [4]

Kada korisnik poseduje (barem jedan) ključ, može koristi rešenje u svrhu šifrovanja, odnosno dešifrovanja svojih fajlova.

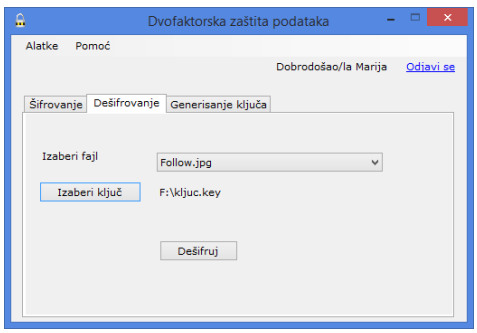
U procesu šifrovanja potrebno je izabrati fajl koji se želi šifrovati i generisani ključ kojim će se on šifrovati. Fajl koji se šifruje može se nalaziti na bilo kojoj lokaciji na klijentskom računaru osim one na kojoj se nalazi ključ (Sl. 12). Pre samog šifrovanja vrši se verifikacija ključa putem lozinke koja je unesena prilikom njegovog generisanja. [4]



Sl. 12. Faza šifrovanja

Tako šifrovani fajl šalje se na server. Pre samog snimanja, vrši se provera da li fajl od ranije postoji u bazi. Ukoliko postoji, korisnik se o tome obaveštava i skladištenje nije moguće. Ukoliko je sve u redu, sam fajl se smešta u *EncryptData* tabelu. Radi praćenja istorije fajla, u tabeli *ActivityLog*, beleži se aktivnost korisnika, odnosno vreme kada je i koji korisnik uskladištio fajl.

Nakon skladištenja u bazu, originalni fajl se briše sa klijentskog računara.



Sl. 13. Faza dešifrovanja

U sekciji Dešifrovanje (Sl. 13), korisniku se prikazuje lista fajlova koje je prethodno smestio u oblak. On bira onaj koji želi da dešifruje i učitava ključ (sa USB *tokena*) i potom unosi lozinku za validacija ključa. Ukoliko je validacija uspešna, šifrovani fajl se preuzima sa servera (u memoriju klijentskog računara), a zatim proverava da li je izabrani ključ odgovarajući (da li je taj korišćen i prilikom šifrovanja). Ukoliko jeste, fajl se dešifruje i skladišti na lokaciji koja je prethodno izabrana u podešavanjima aplikacije. U suprotnom, šifrovani fajl se briše iz memorije.

C. 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. [4]

Nad dobijenim ključevima sprovedeni su testovi za procenu informacionog sadržaja. Za ovo rešenje korišćeni su serijski test i ispitivanje entropije preklapajućih i nepreklapajućih uzoraka. [3] Dobijeni rezultati predstavljeni su u uporednom prikazu sa uzorkom preuzetim sa www.random.org. Rezultati testova prikazani su u tabelama 1, 2, 3 i 4. [4]

TABELA 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 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

DOI: 10.7251/ZRSNG1501001V

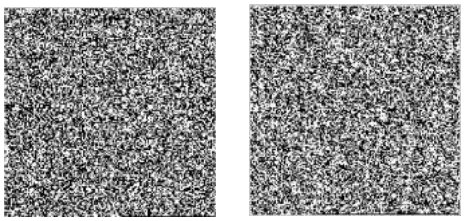
TABELA 3 ENTROPIJA SA PREKLAPANJEM

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

TABELA 4 ENTROPIJA BEZ PREKLAPANJA

Tip testa	Entropija bez preklapanja	
	random.org	***generisan ključ***
Monobit	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. [1]



Sl. 14. random.org (levo) i generisani ključ (desno)

Na sl. 14, na levoj strani prikazan je šum generisan iz atmosferskog šuma [5], dok se na desnoj strani nalazi generisan ključ.

IV. ZAKLJUČAK

U ovom radu najpre su predstavljena neka od najkvalitetnijih rešenja na tržištu iz oblasti skladištenja podataka u oblaku, sagledane su njihove prednosti i nedostaci i na osnovu toga odlučeno je šta će biti implementirano u rešenju koje je predstavljeno u ovom radu.

Nakon toga predstavljeno je predloženo i razvijeno rešenje. Ono predstavlja sistem za bezbedan prenos i skladištenje (šifrovanih) podataka u oblaku. Cilj je bio da se obezbedi *end-to-end* zaštita, odnosno prenos unapred šifrovanih fajlova na server. Autentifikacija korisnika na sistem kao i sam prenos fajlova odvija se preko SSL zaštićenog kanala [2] čime se garantuje poverljivost podataka.

Rešenje obezbeđuje kriptografsku zaštitu svih vrsta fajlova uz implementirani modul za generisanje kriptološkog ključa preko pokreta miša koji su uzeti kao materijal za ključ zbog potrebe da se omogući što veća slučajnost, a samim tim i entropija. Drugi cilj ovog rada bio je da se postigne pravi generator slučajnih brojeva (TRNG). [1] Takođe je omogućeno generisanje neograničenog broja ključeva uz odabir određenog nivoa sigurnosti. Dobijeni ključevi se skladište isključivo na hardverskom *tokenu* (USB *token*). Dosta pažnje je posvećeno i ergonomiji aplikacije kako bi korisnici sa lakoćom mogli da je koriste.

Generisani ključevi su podvrgnuti teorijsko-informacionoj analizi u eksperimentalnom okruženju kojom je potvrđeno da je postignuta željena slučajnost. Kao reprezentativni uzorak sa kojim su upoređivane dobijene vrednosti korišćene su vrednosti preuzete sa sajta random.org.

Da bi se sprečila zloupotreba ključeva od strane drugih korisnika ili u slučaju krađe istih, uvedena je dvofaktorksa autentifikacija prilikom šifrovanja odnosno dešifrovanja fajlova. Generisani ključevi se dodatno štite lozinkom.

LITERATURA

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

[2] Bhiogade, S., „Secure Socket Layer,“ u *InSITE - "Where Parallels Intersect"*, Mumbai, India, 2002.

[3] Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S., Levenson, M., Vangel, M., Banks, D., Heckert, A., Dray, J., Vo, S., „A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications,“ National Institute of Standards and Technology, Gaithersburg, 2010.

[4] Vujošević, M., „Razvoj sopstvenog rešenja za kriptografsku zaštitu sa implementiranim modulom za generisanje simetričnog kriptološkog ključa,“ na „*Prva međunarodna naučna konferencija - Sinteza 2014“,* Univerzitet Singidunum, Beograd, april.2014, 900-994, DOI: 10.15308/SINTEZA-2014-990-994

[5] „Statistical Analysis,“ Random.org, http://www.random.org/analysis

[6] Tresorit, https://tresorit.com

[7] SpiderOak, https://spideroak.com

[8] TeamDrive, http://www.teamdrive.com

[9] MEGA, https://mega.co.nz.

[10] Dropbox, https://www.dropbox.com

DOI: 10.7251/ZRSNG1501001V