



UNIVERZITET SINGIDUNUM
FAKULTET ZA INFORMATIKU I RAČUNARSTVO

Kriptografska zaštita baza podataka na deljenom Veb hosting-u

- Diplomski rad -

Mentor:

Student:

Dušan Ilić

Br. indeksa: 2009/201168

Beograd, 2013.



FAKULTET ZA INFORMATIKU I RAČUNARSTVO

UNIVERZITET SINGIDUNUM

FAKULTET ZA INFORMATIKU I RAČUNARSTVO

Beograd, Danijelova 32

Kandidat: Dušan Ilić

Broj indeksa: 2009/201168

Smer: Informatika i računarstvo

Tema: Kriptografska zaštita baza podataka na deljenom Veb hosting-u

Zadatak:.

Datum odobrenja rada:

Beograd, _____._____._____.

MENTOR

Doc. dr Aleksandar Jevremović

DEKAN

Prof. dr Miroslav Lutovac

Kriptografska zaštita baza podataka na deljenom Veb hosting-u

Sažetak: U ovom radu bavićemo se sve više prisutnom problematikom zaštite baza podataka na deljenom veb hosting-u. Naš zadatak je da obezbedimo zaštitu od administratora hosting servisa naših klijenata. Na ovaj način podaci će se nalaziti u šifrovanoj formi u toku transmisije i mirovanja. U ovom slučaju legitiman pristup će biti ostvaren samo uz odgovarajući simetrični ključ. Šifrovanje će se obavljati u Java aplet aplikaciji na strani korisnika sa AES algoritmom koji koristi simetrične ključeve. Generisanje i razmena sesijskih ključeva će se obavljati putem *Diffie-hellman* algoritma kroz javni komunikacioni kanal između korisnika i našeg Veb servera, gde prosleđujemo ključeve na anonimni Veb server na čuvanje. Šifrat se prosleđuje od korisnika do Veb servera klijenta gde se trajno čuva. Arhitektura strane klijentskog i našeg Veb servera će biti podržana sledećim nosećim tehnologijama: PHP i MySQL. Baza podataka na klijentskom MySQL serveru sadržaće šifrate koji su indeksirani na takav način koji kasnije obezbeđuju izvršavanje upita „SELECT“ nad šifrovanim podacima. Kompletno rešenje predviđa Java aplet aplikaciju koja podržava šifrovanje AES algoritmom, PHP servis za skladištenje šifrata na strani klijenta i PHP servis za razmenu i skladištenje simetričnog ključa na našem Veb serveru. Na ovaj način postigli bismo gore navedeni cilj, a to je kriptografska zaštita baze podataka na deljenom Veb hosting-u u cilju zaštite od administratora hosting servisa i drugih korisnika na istom deljenom Veb serveru. Razvijeno rešenje će potom biti testirano nasuprot Java aplet aplikaciji koja će sadržati parametre za šifrovanje sa RSA algoritmom koji koristi asimetričnu kriptografiju sa javnim ključevima, gde će u Java aplet biti upisan javni ključ vlasnika podataka, a privatni ključ će se čuvati na fizičkom uređaju vlasnika baze podataka. Java aplet prosleđuje podatke do Veb servera klijenta gde se nalazi PHP servis za upisivanje podataka u MySQL bazu podataka na trajno čuvanje.

Ključne reči: bezbednost, AES, *Diffie-Hellman*, Deljeni Veb serveri.

Cryptographic protection of databases on the shared Web hosting

Abstract: This paper will deal with more present problems of data protection on the shared Web hosting services. Our goal is to provide protection from administrators of hosting services of our clients. In this way the data will be in encrypted form during the transmission and when in storage. Legitimate approach will only be realized with the appropriate symmetric key. Encryption will be done in Java applet application on the user's side with the AES, algorithm that uses symmetric keys. Generating and sharing session keys will be performed through Diffie-Hellman algorithm through a public communication channel between the user and our Web server, which forwards the keys to the anonymous Web server for safekeeping. Cipher text is passed from the user to the Web server where the client is kept permanently. The architecture of our client and the Web server will be supported by supporting the following technologies: PHP and MySQL. The database on the client MySQL server contain cipher text indexed in such a way that later provided the query "SELECT" over the encrypted data. The complete solution provides Java applet application that supports the AES encryption algorithm, PHP cipher text storage service on the client side and PHP service for sharing and storing symmetric key to our Web server. In this way, we could achieve the above objective, and that the cryptographic protection of databases on the shared Web hosting in order to protect from the administrator hosting services and other users on the same shared Web server. Developed solution will then be tested against Java applet application that will contain the parameters for the RSA encryption algorithm that uses asymmetric public key cryptography, where the public key will be hard coded into Java applet application, and the private key will be stored on the physical device of the data owner. Java applet forwards the data to the shared Web server where the client PHP service inserts data into a MySQL database for permanent storage.

Keywords: security, AES, Diffie-Hellman, Shared Web servers

SADRŽAJ

1	Uvod.....	- 4 -
1.1	Predmet istraživanja	- 4 -
1.2	Doprinosi rada	- 5 -
1.3	Struktura rada	- 5 -
2	Pregled u oblasti istraživanja	- 7 -
2.1	Prednosti i nedostaci kriptografske zaštite na deljenom Veb hosting servisu	- 8 -
2.1.1	Prednosti upotrebe kriptografske zaštite na deljenom Veb hosting servisu	- 8 -
2.1.2	Mane upotrebe kriptografske zaštite na deljenom Veb hosting servisu	- 9 -
2.2	Razvoj oblasti kriptografske zaštite na deljenim Veb serverima	- 9 -
2.3	Primena tehnologije zasnovane na predloženim rešenjima zaštite podataka	- 10 -
3	Teorijske osnove istraživanja	- 11 -
3.1	Algoritmi sa simetričnim ključevima	- 11 -
3.1.1	Blokovske šifre	- 11 -
3.1.2	AES.....	- 14 -
3.2	Generisanje i razmena sesijskog ključa	- 17 -
3.2.1	<i>Diffie-Hellman</i>	- 17 -
3.3	Heš funkcije.....	- 18 -
3.3.1	MD5.....	- 19 -
3.4	Kriptografski sistemi sa javnim ključevima	- 21 -
3.4.1	RSA	- 21 -
3.5	<i>Java applet</i>	- 22 -
3.6	PHP.....	- 23 -
4	Pregled predloženog rešenja.....	- 25 -
4.1	Generička šema prvog predloženog rešenja	- 25 -

4.1.1	Opis šeme	25 -
4.1.2	Prednosti	26 -
4.1.3	Mane	26 -
4.1.4	Pretpostavke	26 -
4.1.5	Pregled razvijenog rešenja.....	27 -
4.2	Generička šema drugog predloženog rešenja	31 -
4.2.1	Opis šeme	32 -
4.2.2	Prednosti	32 -
4.2.3	Mane	32 -
4.2.4	Pretpostavke	32 -
4.2.5	Pregled razvijenog rešenja.....	33 -
4.3	Postavka i objašnjenje eksperimentalnog okruženja	34 -
4.4	Evaluacija ili performanse predloženog okruženja sa prikazom eksperimentalnih rezultata.....	36 -
4.5	Oblast primene predloženog rešenja.....	37 -
5	Zaključak.....	38 -
5.1	Sumarni ciljevi rada.....	38 -
5.2	Ostvareni rezultati.....	38 -
5.3	Predlog za budući rad	38 -
	Literatura	40 -

Spisak slika

Ilustracija 1	Primer šifrovanja Algoritma sa simetričnim ključem	11 -
Ilustracija 2	Model ulančavanja blokova kod šifrovanja	12 -
Ilustracija 3	Model ulančavanja blokova kod dešifrovanja	13 -
Ilustracija 4	Primer <i>ShiftRow</i> operacije.....	15 -
Ilustracija 5	Ulaz u funkciju za mešanje	16 -
Ilustracija 6	Računanje izlaza za prvu kolonu ulaza	16 -
Ilustracija 7	Rezultat funkcije mešanja	16 -
Ilustracija 8	Operacija dodavanja ključa.....	16 -
Ilustracija 9	preslikavanje u heš funkcijama	19 -
Ilustracija 10	Primer bafera MD5 heš funkcije.....	20 -

Ilustracija 11 Generička šema rešenja sa simetričnim ključevima	- 25 -
Ilustracija 12 Grafički interfejs rešenja 1	- 27 -
Ilustracija 13 Generisanje <i>Diffie-Hellman</i> parametara	- 28 -
Ilustracija 14 Funkcija prosleđivanja <i>Diffie-Hellman</i> parametara	- 28 -
Ilustracija 15 Funkcija prihvatanja Y vrednosti	- 28 -
Ilustracija 16 Derivacija ključa na osnovu deljene tajne	- 29 -
Ilustracija 17 Funkcija šifrovanja podataka	- 29 -
Ilustracija 18 Funkcija računanja deljene tajne	- 30 -
Ilustracija 19 Derivacija i upis sesijskog ključa u bazu podataka	- 30 -
Ilustracija 20 generička šema rešenja sa RSA algoritmom	- 31 -
Ilustracija 21 Grafički interfejs rešenja 2	- 33 -
Ilustracija 22 Funkcija za šifrovanje RSA algoritmom	- 33 -
Ilustracija 23 PHP funkcija za upisivanje podataka u bazu podataka	- 34 -
Ilustracija 24 <i>NetBeans IDE 7.3</i>	- 35 -
Ilustracija 25 Adobe Dreamweaver CS6	- 35 -
Ilustracija 26 MySQL baza podataka	- 36 -
Ilustracija 27 <i>XAMPP Control Panel</i>	- 36 -

Spisak Tabela

Tabela 1 Lookup tabela S-kutije kod AES-a	- 15 -
Tabela 2 Primer protokola računanja tajne vrednosti	- 18 -
Tabela 3 Primer tabele ključ	- 31 -
Tabela 4 Primer tabele šifrat	- 31 -
Tabela 5 primer tabele rsa šifrata	- 34 -

1 Uvod

1.1 Predmet istraživanja

Deljeni Veb hosting je ulivao nepoverenje korisnicima od nastanka zbog saznanja da administratori Veb hosting provajdera imaju pristup vašoj bazi i aplikacijama koje se nalaze na Veb serveru kao. Takođe se javlja i opasnost od kompromitovanja vaše baze od strane korisnika koji se nalaze na istom deljenom Veb serveru. Zbog navedenih razloga opšta praksa je da se poverljivi podaci ne čuvaju na deljenim Veb serverima. Deljeni Veb serveri su trenutno u najvećoj upotrebi gde je taj koncept usvojen i razvijen zarad ponude jeftinijih hosting usluga time što je omogućen nezavistan rad više korisnika na jednom Veb serveru. Nasuprot deljenim Veb serverima imamo *dedicated* servere koji dozvoljavaju jednog korisnika po Veb serveru, ali po mnogo većoj ceni. Korišćenjem opreme i hosting usluga kompanije u koje nemamo poverenja kao i deljenje Veb servera sa korisnicima koji nam nisu poznati, samim tim i nepoverljivi postavlja se problem zaštite privatnosti vlasnika podataka kao i privatnosti njegovih klijenata. Istraživanjem ove problematike fokusiramo se na oblasti koje se tiču kriptografskih algoritama za zaštitu i bezbedan prenos podataka do odredišta, takođe se istražuju kriptografske heš funkcije koje će se koristiti da derivaciju ključa na osnovu zajedničke tajne kod kriptografskih algoritama sa simetričnim ključevima. Tokom istraživanja da bih doneli najbolje rešenje došli smo do dva najbolja i najjednostavnija rešenja koja koriste različite kriptografske metode i testirali ih međusobno kako bih došli do najboljeg načina zaštite podataka u ovoj oblasti.

Prvo rešenje je fokusirano na razvoj servisa koji će zaštitu pružati pomoću kriptografskih algoritama sa simetričnim ključevima, kao najbolji algoritam odabran je AES kriptografski algoritam sa dužinom simetričnog ključa od 128 bita, za generisanje i razmenu zajedničke informacije koristimo *Diffie-Hellman* protokol koji generiše 160 bita zajedničke informacije na obe strane, za dobijanje ključa od 128 bita koristimo heš funkciju MD5 koju primenjujemo nad zajedničkom informacijom za dobijanje ključa odgovarajuće veličine. Razmena ključa će se obavljati između korisnika i servisa na Veb servisu pružaoca usluga, a šifrovani sadržaj se prosleđivati na Veb server klijenta.

Drugo rešenje se fokusira na razvoj servisa koji koristi kriptografski algoritam sa javnim ključevima za šifrovanje i bezbedan prenos podataka. Kao trenutno najpopularniji, RSA algoritam je odabran zbog najšire upotrebe a i najrasprostranjenije podrške na Veb platformama. Algoritam će koristiti ključ od 1024 bita i eksponent od 65537. Javni i privatni ključ kao i ostali parametri potrebni za RSA algoritam su dobijeni putem internet stranice „pepta.net“. Razmena Java apleta se obavlja između Veb servera klijenta i korisnika gde Java aplet sadrži javni ključ i modul vlasnika aplikacije koji se nakon šifrovanja prosleđuju na PHP Veb servis i upisani u bazu podataka na trajno čuvanje. Privatni ključ se nalazi na fizičkom uređaju kod vlasnika podataka i koristi se a kasnije dešifrovanje podataka.

Noseće tehnologije upotrebljene za razvoj servisa su *Java applet v7* i *PHP v5* programski jezik, alati korišćeni za pisanje programskog koda su *NetBeans IDE 7.3* za *Java applet* i *Adobe Dreamweaver* za pisanje PHP koda.

U nastavku rada predstavimo oba razvijena rešenja i upoređivanjem predstaviti prednosti i mane oba predložena rešenja.

1.2 Doprinosi rada

U daljem radu biće objašnjen doprinos istraživanja i na osnovu toga doprinos razvijenog sistema upotrebljivog u realnim uslovima. Sledeći doprinosi su neki od glavnih doprinosa koje donose rezultati istraživanja:

- Osmišljeno rešenje doprinosi bezbednosti na deljenom Veb serveru unošenjem bezbednog toka podataka od klijenata do i nakon njihovog trajnog smeštanja u bazu podataka.
- Korišćenjem *Diffie-Hellman* algoritma sesijski ključ se dogovara na bezbedno, kroz javni komunikacioni kanal na transparentan način.
- Korišćenjem simetričnog algoritma poboljšavaju se performanse i efikasnost u odnosu na sisteme sa asimetričnim algoritmima i smanjuju troškovi generisanja ključeva.
- Doprinos poverenju u smeštanje poverljivih podataka na deljene Veb servere.
- Doprinos smanjenju troškova zakupa servera upotrebom deljenih ili posvećenih servera koji su jeftiniji u odnosu na kupovanje i održavanje svojih Veb servera.
- Veliki značaj u oblasti zaštite Veb poslovanja gde se štite poverljivi podaci u ne poverljivoj okolini od svih učesnika u njoj.

Uvođenjem novog metoda zaštite i njegovim upoređivanjem sa nekim od ranije predloženih rešenja dobijaju se rezultati koji se njihovim upoređivanjem odnose direktno na doprinose prethodnog rešenja. Dobijeni rezultati kao i njihova poređenja su obrađeni u daljem tekstu rada.

1.3 Struktura rada

U prvoj tački rada se opisuju predmeti istraživanja rada, tj. Oblasti koje smo pokrili istraživanjem kao i uvod u celokupan dalji rad. Fokusi tokom istraživanja, u prvoj tački nam opisuju usmerenja u istraženim oblastima i doprinose rada u realnim uslovima. Nakon uvođenja u problematiku kojom se bavimo navedena je i struktura rada i njegov dalji tok.

Oblasti kojima se bavi ovaj rad predstavljene su u drugoj tački rada one se tiču svim oblastima i predmetima istraživanja kao i dosadašnjem razvoju oblasti i njenoj celokupnoj primeni pri čemu se objašnjavaju primene u realnim uslovima.

Teorijske osnove istraživanja i razvoja predloženih rešenja na kojima se bazira naš rad su objašnjene u trećoj tački rada. Prikazujemo teorijsku osnovu korišćenih kriptografskih mehanizama i tehnika, korišćene tehnologije u razvoju, kao i opis programskih jezika kojim su pisana eksperimentalna rešenja.

U četvrtom delu rada bavimo se pregledom predloženih, izlažu se generičke šeme sa njihovim definicijama i specifikacijama i praktično razvijene komponente sistema i njegove specifikacije, izvorni kod razvijenih komponenti uključujući i razvojna okruženja koja su korišćena

Poslednja tačka rada zaključuje celokupnu obrađenu materiju, predložene generičke šeme i specifikacije korišćene za razvoj sistema. Izvode se upoređivanja predloženih rešenja i predstavljaju dobijeni rezultati. Rad se završava sa

predlaganjem tema za budući tok razvoja ove materije i navodi nekoliko primera pravca u kome bi se kretala dalja istraživanja.

2 Pregled u oblasti istraživanja

Tokom istraživanja ove oblasti izdvojeno je nekoliko naučnih radova koji se odnose na oblast bezbednosti na deljenim Veb serverima kao i o pravilnoj upotrebi kriptografskih algoritama.

Sledeći citat je poslužio i direktno uticao na istraživanje ove oblasti i unapređenje postojećih bezbednosnih sistema na deljenim Veb serverima i predstavlja jedan od razloga razvoja unapređenog sistema kao i pisanja ovog rada.

"History has taught us: never underestimate the amount of money, time, and effort someone will expend to thwart a security system. It's always better to assume the worst. Assume your adversaries are better than they are. Assume science and technology will soon be able to do things they cannot yet. Give yourself a margin for error. Give yourself more security than you need today. When the unexpected happens, you'll be glad you did." ^[11]

Naveli smo i da je od suštinskog značaja i sama primena kriptografskih algoritama u sistemu jer često od toga zavisi kompletna bezbednost sistema. Navedeno je da:

„I najbolji algoritam pogrešno implementiran može dovesti do kompromitovanja celog sistema“. ^[2]

Ovaj kontekst se odnosi na to da nije dovoljna upotreba jakih algoritama i protokola već treba obratiti pažnju i na pravilnu upotrebu i održavanje sistema.

Kod AES algoritma koristili smo CBC režim ulančavanja blokova, ali nepravilno upotrebljeni parametri slabe sistem tako je i naglašeno da:

„Generisani inicijalizacioni vektor ne sme biti predvidljiv napadaču, jer pogađanjem inicijalizacionog vektora u CBC režimu moguće je pretpostaviti otvoreni tekst i može biti upotrebljen samo jednom“. ^[10]

Iako je CBC režim ulančavanja blokova doneo napredak u bezbednosti, korišćenjem ne bezbednih parametara može kompromitovati sistem.

Kod simetričnih algoritama postoje problemi vezani za čuvanje i distribuciju ključeva te je po literaturi navedeno:

„Tajni ključevi se ne smeju prenositi javnim komunikacionim kanalom u otvorenom obliku niti čuvati u otvorenom obliku“. ^[2]

Praksa u svetu je trenutno da se na primer lozinke čuvaju u obliku kao „posoljena“ heš vrednost time sprečavajući napadača da čak i putem kolizija u korišćenoj heš funkciji dođe do lozinke. Javni komunikacioni kanal se smatra ne bezbednim okruženjem pa se samim tim i tajni ključ ne sme distribuirati u otvorenom obliku, to je moguće rešiti kroz asimetričnu kriptografiju sa javnim ključevima.

Korišćenje RSA algoritma iako deluje jednostavnije postoji činjenica o RSA koja glasi:

„RSA se zasniva na matematički složenom postupku faktORIZACIJE (rastavljanju) proizvoda dva velika prosta broja na njegove faktore (činioce). Problem kod ovog pristupa je u tome što ne postoji čvrsti matematički dokaz da ne postoji kraći put za faktORIZACIJU.“ ^[1]

Pored ovog problema postoji za RSA algoritam takođe važi pravilo da ako se loše implementira ili koriste ne bezbedni parametri ceo sistem može biti kompromitovan.

2.1 Prednosti i nedostaci kriptografske zaštite na deljenom Veb hosting servisu

Razvoj oblasti zaštite u Veb okruženju je od velike važnosti u suprotnom može doći do povrede privatnosti mnogih pojedinaca, ali i kompanija koje čuvaju poverljive podatke na internetu.

2.1.1 Prednosti upotrebe kriptografske zaštite na deljenom Veb hosting servisu

Jedna od najočiglednijih prednosti je zaštita privatnosti i podataka korisnika na internetu, kod privatnosti korisnika nema alternative pa je u celom svetu donet zakon o zaštiti privatnosti korisničkih podataka, uvođenjem zaštite rešava se problem krađe i zloupotrebe korisničkih podataka pa samim tim i izbegavaju narušavanja privatnosti korisnika.

Takozvani *cloud* servisi su sve popularniji i sve više korisnika i kompanija čuva podatke na internetu kako bi im bili u svakom trenutku i sa svake lokacije dostupni ali zbog svoje ne bezbedne prirode podaci koji se čuvaju na deljenim Veb serverima su podložni kompromitovanju, uvođenjem kriptografske zaštite, tj. Šifrovanja podataka obezbeđujemo da su podaci dostupni samo korisnicima koji poseduju odgovarajući ključ, time je rešen problem kompromitovanja osetljivih podataka na deljenim Veb serverima.

Uvođenjem kriptografske zaštite omogućavamo i bezbedan prenos podataka u smislu da se štitimo od potencijalnih napadača i kroz javni komunikacioni kanal što znači da prenosimo šifrovanje podataka na stranu korisnika, tako su podaci zaštićeni i pre stupanja na javnu mrežu.

Kriptografska zaštita onemogućava napad tipa *SQL injection* tako što napadač dobija samo šifrovan sadržaj i bez odgovarajućeg ključa ne može doći do otvorenog teksta.

2.1.2 Mane upotrebe kriptografske zaštite na deljenom Veb hosting servisu

Uvođenje bilo kakve kriptografije znači pad performansi sistema, veličina pada zavisi od veličine podataka koji se šifruju odnosno dešifruju, kao i od performansi samog algoritma, kao primer navodi se to što je AES nekoliko puta brži za šifrovanje odnosno dešifrovanje od RSA algoritma. U sistemima gde je efikasnost i brzina važna stvar mora se pažljivo odabrati metod zaštite kako bi se minimalizovao pad performansi.

Uvođenje zaštite šifrovanjem ne mora značiti samo po sebi da je sistem totalno obezbeđen postoje opasnosti od raznih napada kao što je „čovek u sredini“ ili scenario u kom je kompromitovan tajni ključ kod simetričnih ili privatni ključ kod asimetričnih algoritama.

Implementacija kriptografske zaštite zahteva i komplikovanije održavanje servisa i samog Veb sajta što se direktno odražava na cenu održavanja.

Kod uvođenja sistema sa javnim ključevima zavisimo od pretpostavke da je sistem bezbedan kao što smo naveli gore, ne postoji čvrst matematički dokaz o sigurnoj bezbednosti.

Kod uvođenja sistema sa simetričnim ključevima čest problem koji se javlja je distribucija i skladištenje ključeva, jer ako u bilo kom od ta dva ključ bude kompromitovan svaka njegova dalja upotreba je nesigurna.

2.2 Razvoj oblasti kriptografske zaštite na deljenim Veb serverima

Oblast kriptografije je relativno mlada oblast pogotovo u Veb okruženjima dok se u poslednjih pet godina Veb poslovanje se višestruko uvećalo u celom svetu internet je dobijao sve više korisnika, a samim tim i sve više korisnika koji su pokušavali to da zloupotrebe.

U pogledu kriptografske zaštite na deljenim Veb serverima godinama nije postojalo praktično rešenje za koje se moglo reći da je relativno bezbedno, razvojem alata za dekompileiranje izvršnih fajlova nije bilo više moguće oslanjati se na softvere za zaštitu otvorenog koda. Tek 2010 godine objavljen je prvi naučni rad sa praktičnim rešenjem upotrebljivim u realnom okruženju koji je koristio algoritam sa javnim ključevima u kombinaciji sa drugim metodama za postizanje šifrovanja podataka, tako što je šifrovanje preneto na stranu korisnika, a uklonjena potreba za čuvanjem tajnog ili privatnog ključa na deljenom Veb serveru.

Ovaj vid zaštite se sporo razvijao zbog opšte prakse da se ne trebaju čuvati osetljivi podaci na deljenim veb serverima, ali kako su deljeni Veb serveri postajali sve popularniji, potreba za privatnošću na deljenim serverima je postala prioritet, ali je tek skoro problem praktično rešen na prihvatljiv način.

Rešenje koje je predloženo i dalje je imalo nedostataka, kao što je nedostatak autentifikacije i sprečavanje napada „Čovek u sredini“, taj problem smo rešili u ovom radu i u nastavku izložićemo predloženo rešenje.

2.3 Primena tehnologije zasnovane na predloženim rešenjima zaštite podataka

Prepoznavanjem potrebe za uvođenje zaštite između korisnika, deljenog Veb servera i vlasnika baze podataka istraživanjem se dolazi do samo dva prihvatljiva i praktično upotrebljiva rešenja, u našem slučaju primenili smo rešenja za pravljenje i čuvanje mejling lista. Dobro sastavljena mejling lista može vredeti mnogo u finansijskom smislu jer je ciljni marketing trenutno najaktuelnija vrsta marketinga.

Mejling liste se danas tretiraju kao osetljivi podaci, jer jednom ukradena mejling lista može dovesti do kompromitovanja integriteta kompanije ili Veb sajta, ali i omogućiti konkurenciji prednost na tržištu. Primenom nekog od dva predložena rešenja rešava se problem zaštite podataka od korisnika, kroz javni komunikacioni kanal do Veb servera klijenta.

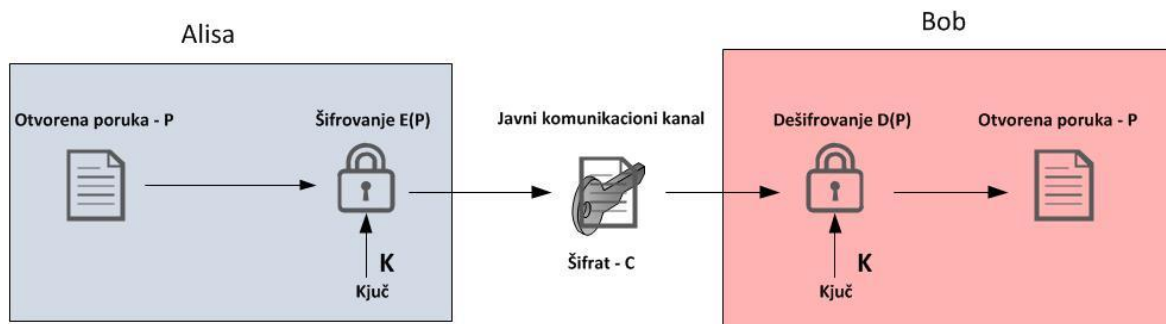
Treba napomenuti da oba rešenja rade na bazi Java aplet i PHP platformi i da je potrebno imati instaliranu najnoviju platformu za izvršavanje Java aplikacija i da je za svaki unos potrebno dozvoliti izvršavanje aplikacije na korisničkom računaru što može biti odbojno za neke korisnike.

3 Teorijske osnove istraživanja

Istraživanje o oblasti kriptografske zaštite podataka su pronađeni u nekoliko naučno istraživah radova, standarda i knjiga koje se tiču kriptografije i nosećih tehnologija primenjenih u rešenjima u daljem tekstu biće objašnjene teorijske osnove na kojima su zasnovana rešenja kao i teorijske osnove biblioteka Java apleta i PHP programskog jezika koje su korišćene za njihovu realizaciju.

3.1 Algoritmi sa simetričnim ključevima

Algoritmi sa simetričnim ključevima su vrsta kriptografskih algoritama koji koriste isti kriptografski ključ za šifrovanje otvorenog teksta i za dešifrovanje šifrata. Ključ u praksi predstavlja zajedničku tajnu između dve ili više strana, koji se može koristiti za bezbednu komunikaciju. Zahtev da obe strane imaju pristup tajnom ključu je jedan od glavnih nedostataka ovakvih tipova algoritma u odnosu na sisteme sa javnim ključevima zbog nedostatka autentifikacije.



Ilustracija 1 Primer šifrovanja Algoritma sa simetričnim ključem

Algoritmi sa simetričnim ključevima se dele na:

- Klasične
- Sekvencijalne
- Blokove

Nakon istraživanja odabrana je upotreba blokovskih šifara koje ćemo objasniti u daljem tekstu.

3.1.1 Blokove šifre

Blokovske šifre su deterministički algoritmi koji rade na grupama bitova fiksnih dužina koji se nazivaju blokovi. Blokove šifre su važan deo mnogih kriptografskih protokola i trenutno u najvećoj primeni za šifrovanje velikih podataka. Blokove šifre rade tako što se podatak promenljive veličine deli u odvojene blokove koji odgovaraju dužini blokova korišćene šifre, a zatim se vrše transformacije nad svakim blokom. Transformacije nad svakim blokom su uvek iste i određene su simetričnim ključem. Kod blokovskih šifara, za šifrovanje je potrebno uvesti i metod ulančavanja blokova, neki od metoda ulančavanja su:

- ECB (*Electronic codebook*) režim
- CBC (*Cipher block chaining*) režim
- CFB (*Cipher feedback*) režim
- OFB (*Output feedback*) režim
- CTR (*Counter*) režim

Iz pogleda bezbednosti metodi ulančavanja blokova moraju da omoguće semantičku bezbednost, što znači da ako se uzme jedan šifrat sa ne poznatim ključem, ne moguće je izvući bilo kakvu informaciju iz šifrata osim dužine poruke. Nakon istraživanja odabran je CBC režim ulančavanja blokova koji će u daljem tekstu biti objašnjen.

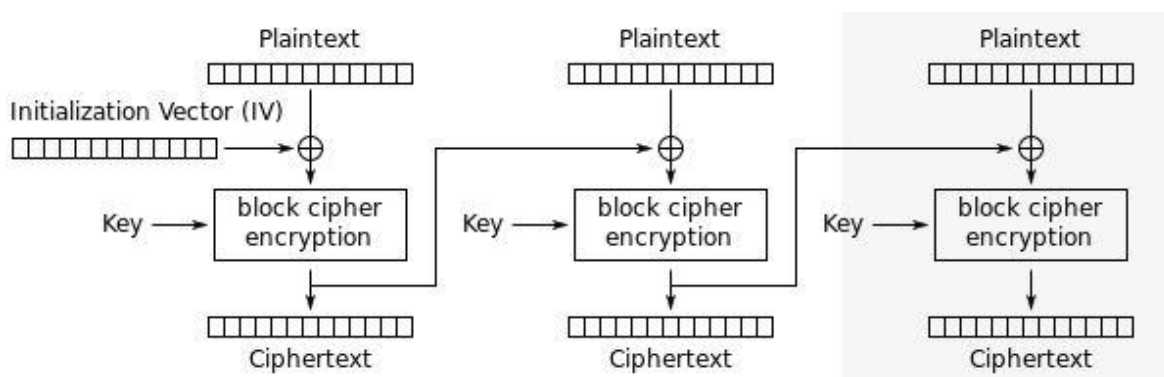
Neke od najpoznatijih blokovskih šifri su:

- DES (Data Encryption Standard)
- AES (Advanced Encryption Standard)
- IDEA (International Encryption Standard)
- RC5 (Rivest cipher 5)

Nakon istraživanja odabran je AES algoritam kao šifra predložena za korišćenje u prvom rešenju koja će koristiti blokove od 128 bita, i simetrični ključ dužine takođe 128 bita. Sam algoritam je objašnjen u nastavku teksta.

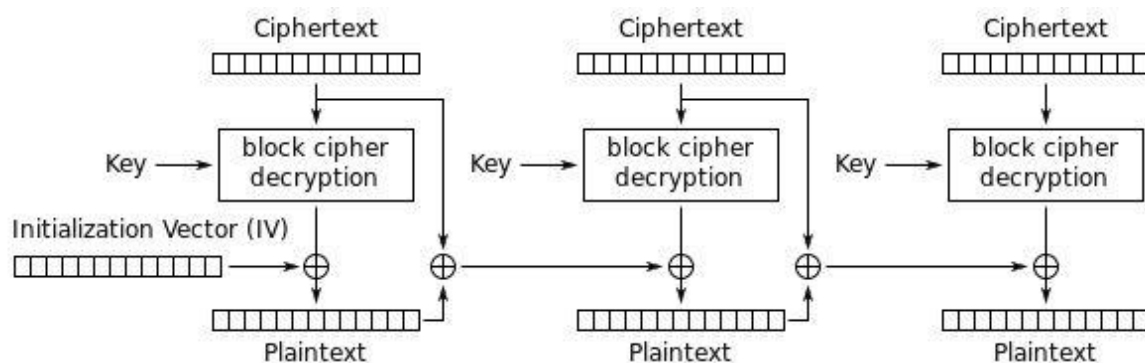
3.1.1.1 CBC režim ulančavanja blokova

Sigurnost CBC leži u slučajno ili pseudo slučajno generisanom inicijalizacionom vektoru koji se prosleđuje algoritmu zajedno sa otvorenom porukom. Inicijalizacioni vektor se ukršta sa prvim blokom otvorenog teksta upotrebom „ekskluzivno ili“ operacije („XOR“) pre šifrovanja, rezultujući blok se zatim koristi kao inicijalizacioni blok sledećem bloku otvorenog teksta. Ovim se postiže cilj da svaki sledeći blok šifrata zavisi od prethodnog.



Ilustracija 2 Model ulančavanja blokova kod šifrovanja

Korišćenjem ne odgovarajućeg inicijalizacionog vektora pri dešifrovanju izaziva pogrešno dobijen blok prve otvorene poruke, samim tim se dobija pogrešan inicijalizacioni vektor sledećeg bloka koji se propagira beskonačno kroz šifrat.



Ilustracija 3 Model ulančavanja blokova kod dešifrovanja

Kod CBC režima kada se podatak podeli na blokove određenih dužina zadnji blok obično je manji od odgovarajuće dužine pa se samim tim i ne može vršiti transformacija nad njim, da bih se problem rešio uvodi se pojam dopunjavanja *padding* koji dopunjuje zadnji blok do odgovarajuće dužine bloka. U početku se razmišljalo o korišćenju određenih dopuna kao na primer dodavanja nula, ali primećeno je da je to ne bezbedno jer se time slabi bezbednost poslednjeg bloka, a upotrebom nekih od poznatih napada ugrožava i ceo šifrat. Kako bi se rešio problem uvedene su određene šeme dopunjavanja koje onemogućavaju bilo kakve tipove napada. Neke od vrsta dopunjavanja su:

- Dopunjavanje bita *Bit padding*, koristi metod koji kaže da se na kraju otvorenog teksta stavi bit 1, a nakon njega blok dopuniti sa bitovima 0, to je jedan od prvih koraka u metodama dopunjavanja sa dva koraka i nazvan je Metod dopunjavanja 2 *Padding method 2*. ovaj metod se može koristiti nad porukama bilo kojih dužina
- Dopunjavanje bajta *Byte padding*, se koristi nad porukama koje mogu biti kodovane integralnim brojem bajtova. Ovaj metod se koristi u većini modernih blokovskih šifri i postoji više verzija :

ANSI.923

ISO 10126

PKCS7

ISO/IEC 7816-4

Zbog ograničenosti upotrebe metoda dopunjavanja u korišćenim nosećim tehnologijama, odabran je PKCS metod verzije 5 (PKCS#5).

PKCS5 dopunjavanje se vrši u celim bajtovima. Vrednost svakog dodatog bajta je broj bajtova koji su dodati. Broj dodatih bajtova će zavisi od granice bloka do koje poruka mora biti dopunjena. Ovaj metod je dizajniran da se koristi u algoritmima sa blokovima dužine 64 bitova.

3.1.2 AES

AES šifra je jedna od najrasprostranjenijih šifara u upotrebi danas, primenjuje se od 2002 godine i prvobitno ime joj je *Rijndael*, AES šifra je visoko rangirana u svetu kriptografije čak je i jedan od najboljih eksperta u toj oblasti Šamir rekao:

„AES šifra sa ključem dužine 256 bita biti zauvek bezbedna, be obzira na predstojeće napretke u računarskoj tehnologiji.“[A.Shamir]

Specifikacije AES Šifre su sledeće:

- Koristi promenljive dužine bloka otvorenog teksta : 128, 192 i 256 bitova
- Dužine ključeva se takođe može menjati : 128, 192 i 256 bitova
- Sadrži ukupno 10 do 16 rundi, zavisno od dužine ključa

U svakoj funkciji se koriste 4 funkcije:

- Nelinearni sloj (*ByteSub*)
- Sloj linearnog mešanja (*ShiftRow*)
- Nelinearni sloj (*MixColumns*)
- Dodatni sloj ključa (*AddRoundKey*)

AES dozvoljava šifrovanje blokova sa ključem različite dužine što znači da se blok otvorenog teksta dužine 128 bitova može šifrovati ključem dužine 256 bitova.

Svi ulazni podaci se tretiraju kao matrica dimenzije 4x4. Sve operacije AES-a se obavljaju nad dvodimenzionalnom nizu bitova, odnosno matrici stanja. Šifrovanje ili dešifrovanje se obavlja tako što se ulazni blok podataka kopira u matricu stanja nad kojom se sprovode operacije završno stanje matrice se kopira u izlazni šifarski blok.

- Nelinearni sloj *SubBytes* predstavlja AES-ov ekvivalent DES-ove S-kutije i može se posmatrati kao nelinearna, ali invertibilna kompozicija dve matematičke funkcije.

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Tabela 1 Lookup tabela S-kutije kod AES-a

- Linearni sloj, pomeranje redova (*ShiftRow*), predstavlja operaciju pomeranja redova, tj. Jednostavno ciklično pomeranje bajtova u poslednja tri reda matrice 4x4, gde se prvi red ne pomera, drugi red se pomera jedan bajt ulevo, treći red se pomera za dva bajta ulevo i četvrti red se pomera 3 bajta ulevo. *ShiftRow* spada u linearne funkcije zbog korišćenja prostog invertovanja u suprotnu stranu.

Ulaz						Izlaz			
d4	e0	b8	1e			d4	e0	b8	1e
27	bf	b4	41	→	1	bf	b4	41	27
11	98	5d	52	→→	2	5d	52	11	98
ae	f1	e5	30	→→→	3	30	ae	f1	e5

Ilustracija 4 Primer *ShiftRow* operacije

- Nelinearni sloj mešanja kolona (*MixColumns*), predstavlja operaciju mešanja kolone primenjenu nad svakom kolonom matrice dimenzije 4x4.

Mešanje kolona se sastoji od pomeranja i XOR operacije i najbolje je implementirana kao *Lookup* tabela. Sve obuhvatna operacija je ne linearna, ali invertibilna, kao i *SubBytes* ima ulogu ka S-kutija kod DES-a. Mešanje kolona sa pomeranjem redova u AES-u predstavlja osnovni izvor difuzije.

Ulaz			
d4	e0	b8	1e
bf	b4	41	27
5d	52	11	98
30	ae	f1	e5

Ilustracija 5 Ulaz u funkciju za mešanje

Ulaz				K1		K1'	
02	03	01	01	d4		04	
01	02	03	01	X	bf	→	66
01	01	02	03	5d		81	
03	01	01	02	30		e5	

Ilustracija 6 Računanje izlaza za prvu kolonu ulaza

Izlaz			
04	e0	48	28
66	cb	f8	06
81	19	d3	26
e5	9a	7a	43

Ilustracija 7 Rezultat funkcije mešanja

Dodavanje sloja ključa (*AddRoundKey*) je jednostavna operacija. Kao kod DES-a, algoritam rasporeda ključa je korišćen da bi generisao pod ključ za svaku rundu.

Ulaz				K-runda 1				Izlaz					
04	e0	48	28	XOR	a0	88	23	2a	=	a4	68	6b	02
66	cb	f8	06		fa	54	a3	6c		9c	9f	5b	6a
81	19	d3	26		fe	2c	39	76		7f	35	ea	50
e5	9a	7a	43		17	b1	39	05		f2	2b	43	49

Ilustracija 8 Operacija dodavanja ključa

Raspored ključa je značajan deo sigurnosti kod svih blokovskih šifri, na kraju svaka od četiri navedene funkcije je invertibilna, rezultat je da je da je čitav algoritam invertibilan i na osnovu toga se AES može nakon šifrovanja, dešifrovati.

3.2 Generisanje i razmena sesijskog ključa

Sesijski ključ je ključ koji se upotrebljava kod algoritama sa simetričnim ključevima za šifrovanje poruke po jednoj sesiji, to znači da se sesijski ključ treba upotrebiti samo jednom za jednu sesiju i odbaciti ga, a za svaku sledeću generisati nov sesijski ključ.

Uvođenjem sesijskih ključeva komplikuje se sistem, ali njegove prednosti rešavaju neke od važnijih problema u simetričnoj kriptografiji kao što su:

- Nekoliko krypto-analitičkih napada postaju lakši kada se sve veća količina materijala šifrue istim ključem, korišćenjem jednog ključa po sesiji ti napadi se znatno umanjuju.
- Asimetrična kriptografija je previše spora u mnogome, iz tog razloga koriste algoritmi sa simetričnim ključevima, ali oni zahtevaju bezbednu distribuciju ključeva. Korišćenjem asimetrične kriptografije za šifrovanje samo tajnog ključa i njegovom distribucijom za korišćenje drugih simetričnih algoritama performanse se znatno povećavaju.

Generisanje sesijskog ključa mora biti slučajno, tako da potencijalni napadač ne može predvideti naredni ključ, jer ne pravilnim generisanjem ključa slabi se snaga celog sistema.

Sesijski ključ je moguće generisati i razmeniti na dva načina:

- Generisati slučajan sesijski ključ na računaru, zatim korišćenjem asimetrične kriptografije šifrovati sesijski ključ i proslediti ga kroz javni komunikacioni kanal. Algoritam koji smo koristili u drugom ponuđenom rešenju je RSA algoritam i objašnjen je u daljem radu.
- Upotrebom *Diffie-Hellman* protokola razmeniti i uspostaviti zajedničku tajnu na osnovu koje se vrši derivacija ključa, u našem slučaju putem heš funkcije.

U našem slučaju koristili smo *Diffie-Hellman* protokol za generisanje i razmenu tajnog sesijskog ključa, zbog jednostavnije upotrebe.

3.2.1 Diffie-Hellman

Diffie-Hellman se koristi za uspostavljanje tajnog simetričnog ključa preko javnog komunikacionog kanala. On se sastoji od sledećih parametara:

- Modul p , je veliki prost broj i ovaj parametar je javan
- g , koji je primitivni koren od p i ovaj parametar je takođe javan
- a , parametar koji generiše jedna strana, ovaj parametar je tajan
- b , parametar koji generiše druga strana, ovaj parametar je tajan

primer protokola kako se računaju tajne vrednosti je dat sledećom tabelom:

Alisa				Bob		
Tajno	Javno	Računa	prosleđuje	Računa	Javno	Tajno
a	p, g		$p, g \rightarrow$			b
a	p, g, A	$g^a \bmod p = A$	$A \rightarrow$		p, g	b
a	p, g, A		$\leftarrow B$	$g^b \bmod p = B$	p, g, A, B	b
a, s	p, g, A, B	$B^a \bmod p = s$		$A^b \bmod p = s$	p, g, A, B	b, s

Tabela 2 Primer protokola računanja tajne vrednosti

Simbol s predstavlja uspostavljenu zajedničku tajnu koja se koristi kao ključ ili se na osnovu nje generiše ključ.

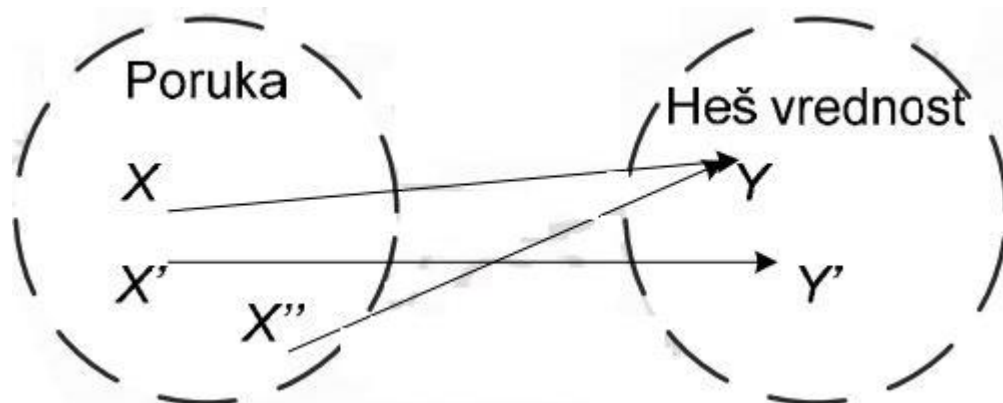
Da bi se došlo do tajni s, a i b napadač bi morao da reši problem diskretnog logaritma, koji se trenutno smatra teškim, ali ako bih se izmislio efikasan algoritam za rešavanje problema diskretnog logaritma, postalo bi lako da se dođe to tajni a i b , a time i učiniti ovaj i mnoge druge kriptografske sisteme sa javnim ključevima nesigurnim.

„Diffie-helman“ protokol štiti od prisluškivanja, ali je ranjiv na „Čovek u sredini“ napade. uvođenjem dodatnih mera bezbednosti i taj problem se može rešiti.

3.3 Heš funkcije

Heš funkcije su jednosmerne funkcije koje od podataka različite dužine prave podatak fiksne dužine koje se nazivaju heš vrednosti.

Heš funkcije se razlikuju od šifarskih algoritama po tome što ne postoji inverzna funkcija i njihova suština je u jednosmernosti koja je posledica kompresije sa gubicima. Funkcionalnosti heš funkcija se odnose na integritet prenesenih podataka, autentifikaciju učesnika u komunikaciji, neporecivost transakcija i druge.



Ilustracija 9 preslikavanje u heš funkcijama

Heš funkcija mora da poseduje sledeća svojstva:

- Kompresija, za proizvoljan broj bitova ulazne poruke x , dužina dobijene heš vrednosti $y = f(x)$ je konačna i najčešće manja od same ulazne poruke.
- Efikasnost, heš vrednost $h(x)$ treba da se jednostavno izračuna za bilo koji ulaz x . Kompleksnost izračunavanja, naravno zavisi od dužine ulazne poruke, ali ne značajno
- Jednosmernost, heš funkcije su jednosmerne funkcije. To znači da je jednostavno izračunati heš vrednost za datu poruku ali je veoma teško rekonstruisati poruku iz date heš vrednosti, tj. Praktično nemoguće.
- Otpornost na kolizije, kod dobre heš funkcije teško je naći sudar (koliziju) što znači da ako je data poruka x_1 čija je heš vrednost $h(x_1)$ teško je naći drugu poruku x_2 tako da njihove sume budu jednake, tj. da je $h(x_1) = h(x_2)$.
- Lavinski efekat, je da promena samo jednog bita na ulazu treba da rezultuje u promeni bar polovine bita dobijene heš vrednosti.

Neke od poznatijih heš funkcija su:

- MD5
- SHA1
- SHA256
- TIGER

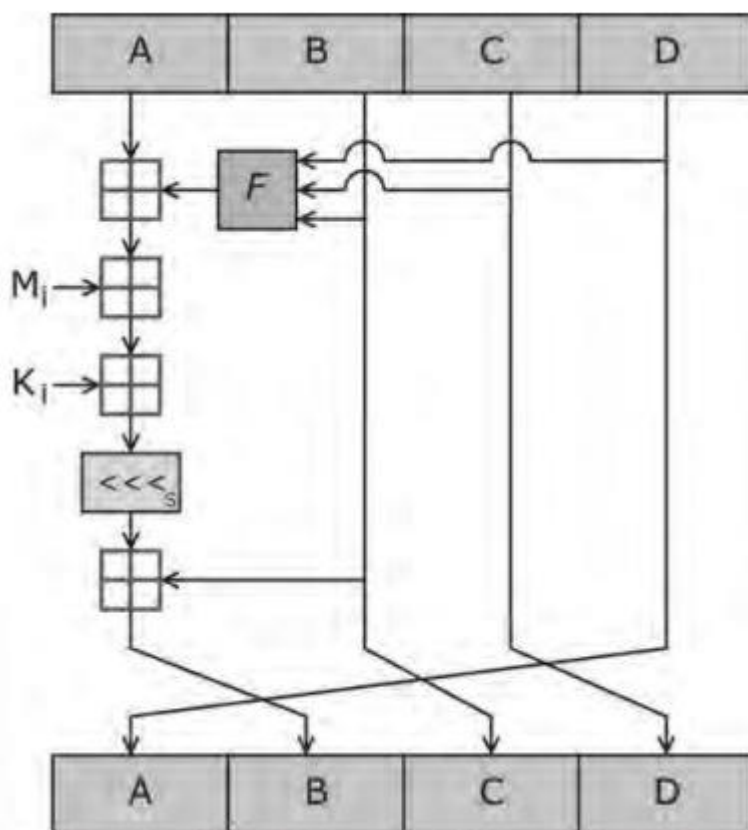
Upotrebljena funkcija u našem slučaju je MD5 heš funkcija radi derivacije ključa od 128 bita koji će se iskoristiti kao sesijski ključ.

3.3.1 MD5

MD5 heš funkcija (*message-Digest algorithm 5*) je funkcija koja se najčešće koristila. Danas ova funkcija ima slabosti i neotporna je na neke kriptografske napade, tako da se sve ređe primenjuje u kriptografiji. Još uvek ima svoju namenu u

proveri integriteta većih fajlova, a i pre svega zbog svoje brzine rada. Dužina heš vrednosti je 128 bitai tipično se predstavlja sa 32 heksadecimalne cifre.

MD5 algoritam kao ulaz koristi n -bitnu poruku $m_0, m_1, m_2, \dots, m_{n-2}, m_{n-1}$. Poruka se mora dopuniti bitovima kako bi njena ukupna dužina odgovarala broju 448, poruka se proširuje tako da joj nedostaju 64 bita, da bi njena ukupna dužina u bitovima bila deljiva sa 512. Najčešće se dopuna vrši tako što se prvo doda jedan bit vrednosti 1 a zatim bitovi sa vrednošću 0. Tako će se poruci dodati ili samo jedan bit ili u najgorem slučaju 512 bitova. Nakon proširenja poruke, poruci je potrebno dodati 64-bitnu reprezentaciju broja b , gde je b dužina izvorne poruke pre njenog proširenja. U slučaju da se dužina poruke ne može prikazati pomoću 64 bita, poruci se dodaju samo nižih 64 bita. Bitovi reprezentacije broja b se dodaju poruci kao dve 32-bitne reči, pri čemu se reč manje težine prva pridodaje.



Ilustracija 10 Primer bafera MD5 heš funkcije

Nakon što je poruka pripremljena ta MD5 algoritam, potrebno je inicijalizovati 128-bitni MD bafer. MD bafer se sastoji od četiri 32-bitnih reči A, B, C, D. Kao inicijalne vrednosti reči u heksadecimalnom sistemu se koriste: A=67452301, B=EFCDAB89, C=98BADCFE i D=10325467. Posle inicijalizovanja pokreće se MD5 algoritam koji se ponovo izvodi za svakih sledećih 512 bitova poruke. Samo jezgro algoritma predstavlja funkciju ta kompresiju koja se sastoji od četiri ciklusa. Svaki od četiri ciklusa ima sličnu strukturu, ali svaki koristi drugačiju primitivnu logičku funkciju F, F, H ili I. Konačan rezultat predstavljaju vrednosti u registrima A, B, C i D koje sabiraju sa njihovim inicijalnim vrednostima. Svaka od tih četiri registara predstavlja jednu četvrtinu dobijene heš vrednosti ulazne poruke.

3.4 Kriptografski sistemi sa javnim ključevima

Asimetrični kriptografski sistemi ili drugim imenom, sistemi sa javnim ključevima se odnose na kriptografske sisteme koji koriste dva ključa od kojih je jedan tajan, a drugi javan. Iako se razlikuju oba ključa su matematički povezana, gde se jedan ključ koristi za šifrovanje, a drugi za dešifrovanje, važno je napomenuti da ni jedan ključ sam po sebi nema mogućnost da obavlja obe funkcije. Javni ključ se može objaviti bez ugrožavanja bezbednosti, dok se privatni ključ mora čuvati od ne autorizovanih osoba.

Sistemi sa javnim ključevima koriste algoritme sa asimetričnim ključevima i nazvani su drugačije kriptografija sa asimetričnim ključevima. Takvi sistemi se zasnivaju na matematičkim vezama kao što su faktORIZACIJA celih brojeva i problem diskretnog logaritma. Iako se računanje javnog ključa primaoca kojim se šifrjuje i tajnog ključa primaoca kojim se dešifrjuje poruka smatra relativno lakim, takođe se smatra ekstremno teškim da se na osnovu poznavanja samo javnog ključa dođe do privatnog ključa.

Ovakvi sistemi premošćuju neke od problema koji se javljaju u algoritmima sa simetričnim ključevima kao što su:

- Distribucija ključa, asimetrični algoritmi ne zahtevaju razmenu ključeva pre početka komunikacije
- Digitalno potpisivanje, ono se odnosi na autentifikaciju pošiljaoca gde se poruka šifrjuje privatnim ključem, a primalac poruke može dešifrovati poruku sa javnim ključem i time utvrditi identitet pošiljaoca.

Ovakvi sistemi se takođe koriste i za razmenu tajnog ključa koji se koristi za algoritme sa simetričnim ključevima tako što se tajni ključ šifrjuje javnim ključem primaoca i prosleđuje u šifrovanom obliku kroz javni komunikacioni kanal.

Sistemi sa javnim ključevima zvuče bolje od sistema koji koriste simetrične ključeve, ali u pogledu upoređuju brzine šifrovanja, sistemi sa simetričnim ključevima su višestruko brži, te se sistemi sa javnim ključevima ne retko koriste za prenošenje tajnog simetričnog ključa umesto šifrovanja celokupnih poruka i time se dostižu odlične performanse sistema.

3.4.1 RSA

RSA (*Rivest, Shamir and Adleman*) je najpoznatiji algoritam iz asimetrične kriptografije, zasniva se na matematički složenom postupku faktORIZACIJE (rastavljanja) proizvoda dva velika prosta broja na njegove faktore (Činioce). Problem kod ovog pristupa je u tome što ne postoje čvrsti matematički dokazi da ne postoji kraći put za faktORIZACIJU.

Za šifrovanje kod RSA algoritma se koristi jednosmerna funkcija sa zamkom. Suština je u pronalaženju funkcije E koja menja otvorenu poruku M u šifrat C , tj. $C = E(M, K_e)$. Parametar K_e se naziva javni ključ i služi isključivo za šifrovanje. Ne može se upotrebiti za dešifrovanje, tj. $M \neq (C, K_e)$ zbog toga što je E jednosmerna funkcija. Ukoliko je poruka namenjena za Boba, svako može šifrovati poruku za njega pomoću njegovog javnog ključa K_e . Da bi se poruka dešifrovala Bob mora da upotrebi tajnu vrednost K_d (privatni ključ) i da primeni inverznu funkciju $M = D(C, K_d)$

kako bi se iz šifrata C dobila otvorena poruka M . Uslov za uspešno šifrovanje i dešifrovanje da je $M = D(E(M, K_e), K_d)$. Jasno je da je neophodna pogodna matematička funkcija sa zamkom. U navedenom primeru privatni ključ K_d je zamka. Poznavanjem zamke proces dešifrovanja je jednostavan. Bez zamke, dešifrovanje je praktično nemoguće. Funkcija koja zadovoljava navedene pretpostavke je oblika:

$$f(x) = X^e \bmod N$$

Uz odgovarajući izbor vrednosti e i N , ova funkcija je jednosmerna. Uz poznavanje tajne vrednosti (zamke) može da se nađe njena inverzna funkcija. Bez poznavanja zamke, ona je praktično nerešiva.

Da bi se generisao Javni i tajni par ključeva za RSA, biraju se dva velika prosta broja p i q i formira se njihov proizvod $N = p * q$, zatim se izabere broj e , uzajamno prost broj sa proizvodom $\varphi(N) = (p-1)(q-1)$ i uz uslov $e < \varphi(N)$. Konačno, potrebno je odrediti d na osnovu sledećeg $d + e^{-1}(\bmod \varphi(N))$, tj. zadovoljeno je sledeće $ed=1(\bmod \varphi(N))$. Nadalje, faktori p i q se razmatraju.

Broj N je modul, broj e je eksponent za šifrovanje, a d je eksponent za dešifrovanje.

RSA par ključeva se sastoji iz delova:

- Javni ključ (N, e)
- Privatni ključ (d)

Za operacije šifrovanja i dešifrovanja primenjuje se funkcija modularnog eksponenta na sledeći način:

- Šifrovanje: $C = M^e \bmod N$
- Dešifrovanje: $M = C^d \bmod N$

Tajnost asimetričnog šifarskog sistema sa RSA algoritmom se zasniva na složenosti faktorizacije broja N . Nakon što bi se izvršila faktorizacija broja N na činioce p i q , bilo bi lako, korišćenjem javno dostupne vrednosti e doći do privatne vrednosti d i iz uslova $ed=1(\bmod \varphi(N))$. Za dovoljno velike brojeve p i q , a samim tim i N , proces faktorizacije je dugotrajan posao. Uz dostupne računarske resurse može se uporediti sa procesom totalne pretrage po prostoru svih ključeva kod simetričnih šifarskih sistema. Potrebno je ponoviti da je problem kod asimetričnih šifarskih sistema, ne postojanje čvrstog matematičkog dokaza da je rastavljanje na faktore jedini način za razbijanje RSA.

RSA algoritam je odabran kao metod šifrovanja podataka u drugom predloženom rešenju i odabrane specifikacije su:

- Dužina ključa je 1024 bita
- Eksponent je 65537

3.5 Java applet

Za realizaciju korisničke aplikacije odabran je Java programski jezik u vidu Java applet-a. Java applet je aplikacija je obično mala aplikacija pisana u Java programskom jeziku i dostavlja se korisniku kroz internet pretraživač u vidu bajt koda. Pokreće se uz pomoć Java virtualne mašine i izvršava na korisnikovom računaru.

Korišćene kriptografske biblioteke za realizovanje predloženih rešenja su:

- *„com.sun.org.apache.xml.internal.impl.div.util.base64“*:

Ova biblioteka se koristi za kodovanje i dekodovanje ključa prilikom šifrovanja odnosno dešifrovanja kako bi ključ bio u tačno određenom formatu pre ulaska u algoritam.

- *java.math.BigInteger*:

Ova biblioteka se koristi za funkcije koje se odnose na modularne operacije kao i stepenovanje velikih brojeva.

- *Java.security.MessageDigest*:

Ova biblioteka se koristi za pozivanje MD5 heš funkcije

- *Javax.crypto.cipher*:

Ova biblioteka se koristi za pozivanje odabranog šifarskog algoritma

- *Javax.crypto.spec.IvParameterSpec*:

Ova biblioteka se koristi za pozivanje funkcije potrebne za generisanje inicijalizacionog vektora za algoritme sa simetričnim ključevima

- *Javax.crypto.spec.SecretKeySpec*:

Ova biblioteka se koristi za pozivanje funkcije gde se tajni ključ koduje odnosno dekoduje sa „base64“ funkcijom.

3.6 PHP

PHP je interpreterski jezik namenjen za razvoj serverskih Veb aplikacija, ali se takođe upotrebljava kao i drugi programski jezici. PHP kod se interpretira pomoću veb servera sa PHP procesorskim modulom koji generiše rezultujuću internet stranicu. Za predložena rešenja upotrebljen je PHP jezik za izradu servisa na serverskoj strani zbog odlične podrške mnogih serverskih platformi, kao i zbog jednostavnosti same sintakse.

U predloženim rešenjima PHP kod je poslužio za generisanja jednog dela sesijskog ključa putem „Diffie-Hellman“ protokola gde se u servis smeštaju funkcije potrebne za uspostavljanje sesijskog ključa, a sa druge strane upotrebljen je kao servis koji upisuje i obrađuje korisničke podatke. Iako su kriptografske funkcije slabije implementirane u PHP jeziku, modularne operacije potrebne za sisteme sa javnim ključevima su uključene i time omogućavaju njegovo korišćenje u svrhe zaštite.

Zbog dizajna predloženih rešenja korišćene su samo dve kriptografske funkcije a to su:

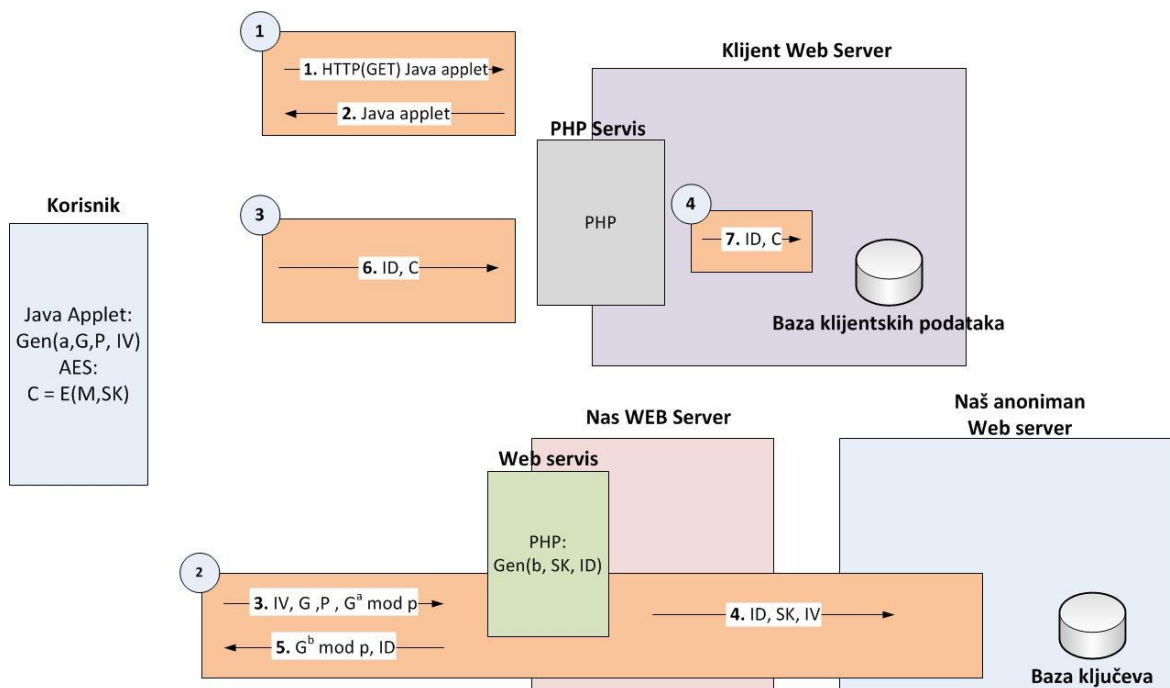
- *bcpowmod()* funkcija, se koristi prilikom računanja sesijskog ključa kod „Diffie-Hellman“ protokola, gde serverska strana računa svoj deo sesijskog ključa i nakon primanja svih parametara računa informacija na osnovu koje se računa sesijski ključ.

- *md5()* funkcija, se koristi za derivaciju sesijskog ključa na osnovu zajedničke tajne uspostavljene „Diffie-Hellman“ protokolom pomoću MD5 heš funkcije.

4 Pregled predloženog rešenja

4.1 Generička šema prvog predloženog rešenja

Nakon istraživanja i testiranja različitih pristupa predložena je sledeća šema upotrebom kriptografije sa simetričnim ključevima:



Ilustracija 11 Generička šema rešenja sa simetričnim ključevima

4.1.1 Opis šeme

- Korak 1:

Korisnik šalje zahtev za Java aplet-om Klijent Veb serveru, potom popunjava formu za unos podataka.

- Korak 2:

Korisnik pritiskom na dugme za slanje podataka pokreće „Diffie-Hellman“ protokol za generisanje sesijskog ključa između korisnika i našeg Veb servera. Kada je ključ razmenjen nas Veb servis prosleđuje identifikacioni broj, sesijski ključ i inicijalizacioni vektor na anonimni Veb server gde će biti trajno smešteni.

- Korak 3:

Nakon razmene ključa pokreće se šifrovanje podataka AES algoritmom i prosleđuje šifrat sa identifikacionim brojem na Klijent Veb server.

- Korak 4:

Identifikacioni broj i šifrat se trajno smeštaju u bazu klijentskih podataka.

4.1.2 Prednosti

Servis je projektovan tako da podržava rad sa više klijenata odnosno više klijenata se može prijaviti na naš servis i koristiti istu bazu podataka za smeštanje korisničkih podataka bez uticanja ili kompromitovanja podataka drugih klijenata.

Da bi se došlo do zaštićenih podataka potrebna je saradnja Veb servera klijenta, našeg Veb servera i anonimnog Veb servera. Zbog postojanja anonimnog Veb servera potencijalni napadač kompromitovanjem baze šifrata morao bi da kompromituje i Veb server u našem posedu da bi došao do anonimnog veb servera i baze sa ključevima, čak i tada morao bi ispitati sve ključeve svih klijenata naspram svih šifrata jednog klijenata.

Deljenjem tajne se postiže cilj da je onemogućena zloupotreba podataka bez saradnje podataka sa oba servera.

Kompromitovanjem jednog ključa ostatak klijentove baze podataka je bezbedan zbog korišćenja sesijskog ključa.

Brza i transparentna obrada podataka za šifrovanje, kao i generisanje i razmena sesijskog ključa.

4.1.3 Mane

Nedostatak autentifikacije predstavlja problem jer je šema ranjiva na napade tipa „čovek u sredini“.

Saradnja više Veb servera podiže nivo troškova, pa se rešenje predstavlja kao uslužni servis.

Korišćenje Java aplet aplikacije se mora odobriti pre svakog izvršavanja na korisničkom računaru što može uticati na saradnje.

4.1.4 Pretpostavke

Veb server sa PHP servisom za generisanje i razmenu sesijskog ključa se nalazi fizički u našem posedu.

Anonimni Veb server se nalazi kod hosting kompanije poznate samo nama i ne sadrži podatke o tome šta se čuva na njemu niti za šta služe ti podaci.

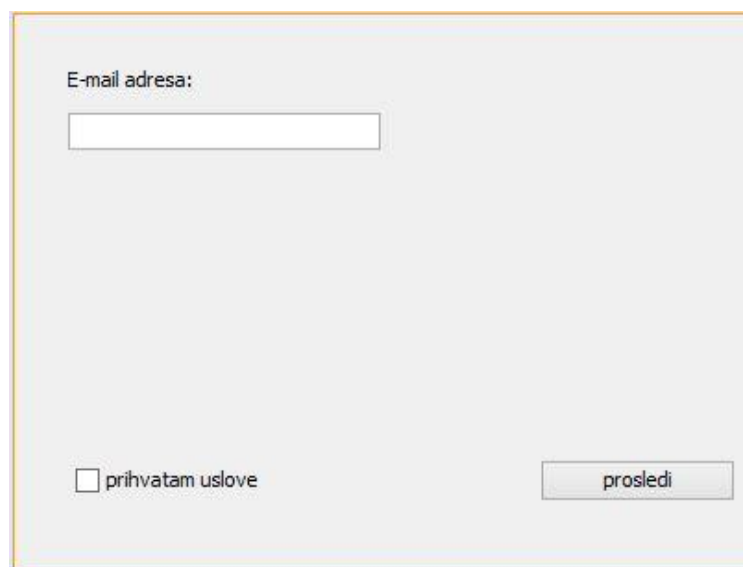
Kanal za prenošenje sesijskih ključeva od Veb servera do klijenta radi upotrebe se smatra prethodno zaštićenim.

4.1.5 Pregled razvijenog rešenja

Prvo predloženo rešenje je razvijeno u skladu sa prethodno objašnjenim karakteristikama u daljem tekstu će biti prikazane ilustracije grafičkog interfejsa i programskog koda aplikacija.

4.1.5.1 Java applet aplikacija

Grafički interfejs je odrađen na relativno jednostavan način kako bi se smanjila bespotrebna dodatna interakcija korisnika, nakon unošenja podataka pritiskom na dugme prosledi pokreću se kriptografske funkcije i funkcije za slanje podataka.

The image shows a simple Java applet window with a light gray background. At the top, there is a label 'E-mail adresa:' followed by a rectangular text input field. Below the input field, on the left, is a checkbox with the text 'prihvatam uslove' next to it. On the right side of the applet, there is a button with the text 'prosledi'.

Ilustracija 12 Grafički interfejs rešenja 1

Prvo funkcija koja se poziva jeste funkcija generisanja parametara za *Diffie-Hellman* protokol koja je realizovana sledećim kodom:

```
public class RandomPrime {  
  
    public BigInteger stepenp() {  
  
        BigInteger bip = new BigInteger(65,20, new Random(System.currentTimeMillis()))  
  
        return bip;  
    }  
    public BigInteger stepena() {  
  
        BigInteger bia = new BigInteger(65,20, new Random(System.currentTimeMillis()))  
  
        return bia;  
    }  
    public BigInteger stepeng() {  
  
        BigInteger big = new BigInteger(65,20, new Random(System.currentTimeMillis()))  
  
        return big;  
    }  
}
```

Ilustracija 13 Generisanje Diffie-Hellman parametara

Zatim se pokreće računanje vrednosti X i prosleđuje našem Veb serveru po pravilima koje smo definisali ranije :

```
dhkey kljuc = new dhkey(a, g, p);

BigInteger x = kljuc.calculateX();
System.out.println("ovo je x:" + x);

g2 = g.toString();
p2 = p.toString();

posalji send = new posalji(g2, p2, x.toString(), iv2 );
```

Ilustracija 14 Funkcija prosleđivanja Diffie-Hellman parametara

Nakon toga preuzima se odgovor, tj. vrednost Y i može se pristupiti računanju tajne i derivaciji sesijskog ključa :

```
BigInteger rez = send.sendPostRequest();
String stry = rez.toString().substring(0,20);

BigInteger yy = new BigInteger(stry);
System.out.println("ovo je y:" + yy);

int idp,idp2;
idp = rez.toString().length() - 3;
idp2 = rez.toString().length();

String id = rez.toString().substring(idp,idp2);
BigInteger idd = new BigInteger(id);
System.out.println("id je: " + idd);
kljuc.setY(rez);
```

Ilustracija 15 Funkcija prihvatanja Y vrednosti

```

BigInteger sk = kljuc.calculateKey();

System.out.println("ovo je kljuc" + sk);

try {
    MessageDigest md = MessageDigest.getInstance("MD5");
    String sks = sk.toString();
    md.update(sks.getBytes());
    byte byteData[] = md.digest();
    StringBuffer sb = new StringBuffer();

    for (int i = 0; i < byteData.length; i++) {
        sb.append(Integer.toString((byteData[i] & 0xff) + 0x100, 16).substring(1));
    }

    System.out.println("md5 hash kljuc: " + sb.toString());
    mdkljuc = sb.toString();
} catch (NoSuchAlgorithmException ex) {
    Logger.getLogger(GlavniProzor.class.getName()).log(Level.SEVERE, null, ex);
}

String manjijkljuc = mdkljuc.substring(0, 16);

IvParameterSpec ivspec;

```

Ilustracija 16 Derivacija ključa na osnovu deljene tajne

Nakon uspostave ključa sledeći deo koda je zadužen za šifrovanje podataka:

```

public static String crypt(String input, String key, IvParameterSpec ivspec) {

    byte[] crypted = null;
    try {
        SecretKeySpec skey = new SecretKeySpec(Base64.decode(key), "AES");
        Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
        cipher.init(Cipher.ENCRYPT_MODE, skey, ivspec);
        crypted = cipher.doFinal(input.getBytes());
    } catch (Exception e) {
    }
    return Base64.encode(crypted);
}

```

Ilustracija 17 Funkcija šifrovanja podataka

Zatim se šifrat prosleđuje veb serveru na upis u bazu podataka.

4.1.5.2 PHP servisi

U trenutku primanja javnih parametara poslatih od korisničke strane, PHP servis računa svoju tajnu b i prihvata javne parametre i pristupa računanju parametra Y i zajedničke tajne. Kada izračuna zajedničku tajnu naš Veb servis vraća vrednost Y i id Java aplet aplikaciji radi uspostavljanja istih parametara na obe strane:

```
$g = $_POST['g'];
$p = $_POST['p'];
$x = $_POST['x'];
$iv = $_POST['iv'];
$broj = "";

$fp = fopen("id.txt", "r");
$id = fread($fp, 4096);

for ($i=1;$i<11;$i++) {
    $rnd = rand(1,9);

    $broj = $broj.$rnd;
}
$b = $broj;

$y = bcpowmod($g,$b,$p);

$y = str_replace(".", "", $y);

$sk = bcpowmod($x,$b,$p);

$sk = str_replace(".", "", $sk);

echo $y.$id;
```

Ilustracija 18 Funkcija računanja deljene tajne

Sledeći deo koda je zadužen za derivaciju ključa na osnovu md5 heš vrednosti kao i za uspostavljanje veze sa bazom podataka na anonimnom Veb serveru i njihov upis:

```
$mdsk = md5($sk);

$id2 = $id + 1;
$fp2 = fopen("id.txt", "w");
fwrite($fp2, $id2);

require "baza.php";

mysql_query("INSERT INTO kljuc (id, sk, iv) VALUES ('$id', '$mdsk', 'iv')");
```

Ilustracija 19 Derivacija i upis sesijskog ključa u bazu podataka

PHP servis na klijentskom Veb serveru je prostog tipa i sadrži funkciju za upisivanje podataka u bazu podataka.

Izgledi MySQL tabela gde se nalaze šifrat i ključevi su prikazane sledećim primerima:

id	sk	iv
101	735bbbc22f7d4007a9905e87d8d46b4b	iv

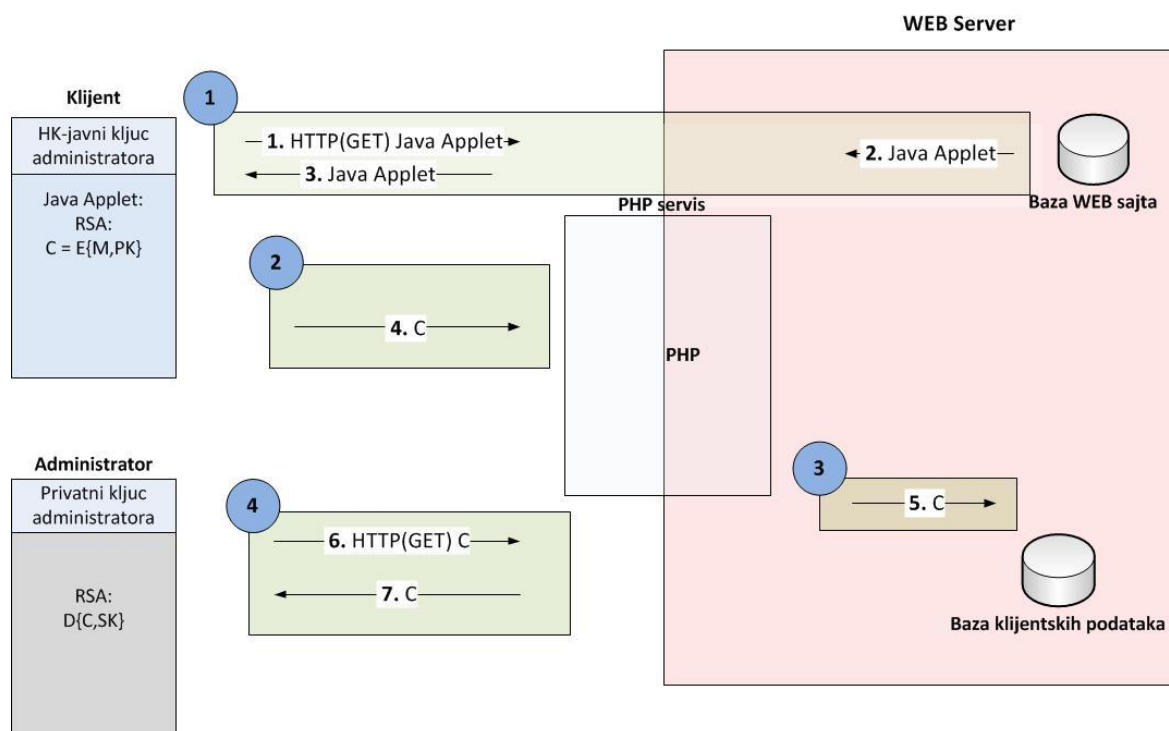
Tabela 3 Primer tabele ključ

id	email
101	FJWHZmnY+Tk6dyCLD2hqsg==

Tabela 4 Primer tabele šifrat

4.2 Generička šema drugog predloženog rešenja

Drugo predloženo rešenje koristi kriptografiju sa asimetričnim ključevima kako bi se rešio problem bezbedne komunikacije i bezbednog smeštanja podataka u bazu podataka. Drugo predloženo rešenje je prikazano sledećom šemom:



Ilustracija 20 generička šema rešenja sa RSA algoritmom

4.2.1 Opis šeme

- Korak 1:

Korisnik šalje zahtev za pokretanje java aplet aplikacije za unos podataka i popunjava formu.

- Korak 2:

Korisnik pritiskom na dugme za prosleđivanje forme pokreće funkciju za šifrovanje RSA algoritmom, gde se podaci šifruju javnim ključem koji je upisan u Java aplet aplikaciji i šifrat se prosleđuje Veb serveru.

- Korak 3:

PHP servis preuzima korisničke podatke u šifrovanom obliku i kao takve ih upisuje u bazu podataka.

- Korak 4:

Ovaj korak se sastoji od primera preuzimanja podataka, dakle vlasnik podataka preuzima šifrate sa Veb servera na bezbedan računar, gde koristi svoj privatni ključ za dešifrovanje podataka kako bi došao do otvorenog teksta.

4.2.2 Prednosti

- Jednostavnija generička šema od generičke šeme koja se koristi simetričnom kriptografijom, upotrebom jednog Veb servera uprošćava se sistem.
- Lakše održavanje ključeva pa i celog sistema.
- Nema potrebe za distribucijom ključa, javni ključ sme da putuje javnim komunikacionim kanalom bez ugrožavanja bezbednosti sistema.
- Zaštita podataka od korisnika, kroz javni komunikacioni kanal, do i nakon smeštanja u bazu podataka.
- Mogućnost rada više sa klijenata na jednom Veb serveru.

4.2.3 Mane

- Smanjene performanse sistema, zbog korišćenja RSA algoritma sa asimetričnim ključevima.
- Korišćenje algoritma za koji nema čvrstog matematičkog dokaza da je potpuno bezbedan.
- Nedostatak autentifikacije.
- Kompromitovanjem privatnog ključa celokupna baza šifrata je kompromitovana.

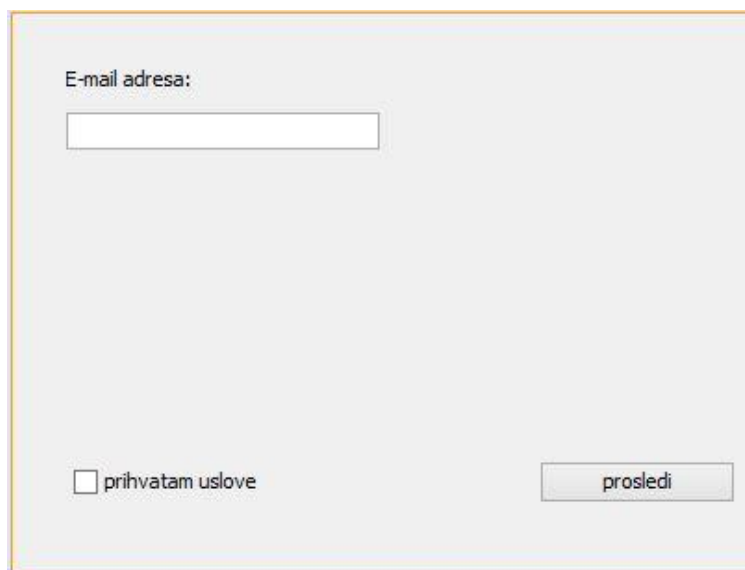
4.2.4 Pretpostavke

- Javni i privatni parametri za RSA algoritam su generisani na bezbedan način.

- Računar klijenta na kojem se vrši dešifrovanje podataka se smatra bezbednim i od poverenja.

4.2.5 Pregled razvijenog rešenja

Grafički interfejs predloženog rešenja koji koristi kriptografiju sa javnim ključevima se ne razlikuje rešenja koje koristi kriptografiju sa simetričnim ključevima, ali pritiskom na dugme za prosleđivanje podataka pozivaju se drugačije kriptografske funkcije za šifrovanje i prosleđivanje podataka:



Ilustracija 21 Grafički interfejs rešenja 2

```
public static byte[] sifruj(String text, PublicKey key) {  
    byte[] cipherText = null;  
    try {  
  
        final Cipher cipher = Cipher.getInstance("RSA");  
  
        cipher.init(Cipher.ENCRYPT_MODE, key);  
        cipherText = cipher.doFinal(text.getBytes());  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
    return cipherText;  
}
```

Ilustracija 22 Funkcija za šifrovanje RSA algoritmom

PHP servis na strani Veb servera podržava funkcije upisivanja podataka u bazu podataka.

```

<?php

$email = trim($_POST['sifrat']);

$id = $_POST['id'];

$email = str_replace(" ", "+", $email);

require "baza.php";

mysql_query("INSERT INTO mejling_lista (id, email) VALUES ('$id', '$email')");

?>

```

Ilustracija 23 PHP funkcija za upisivanje podataka u bazu podataka

Izgled MySQL tabele je skoro identičan kao i kod upisa šifrata za prvo predloženo rešenje i izgleda:

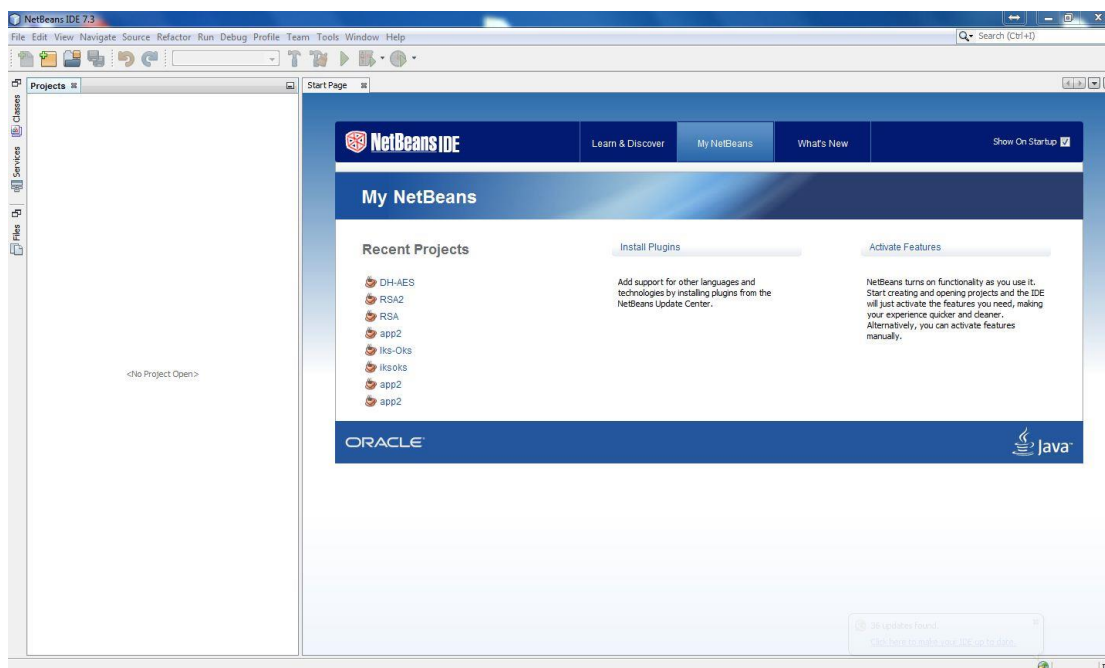
id	email
1	hda73fha89uvhQ&*@Y
2	Jls2dbf@^\$@*.scas45fd

Tabela 5 primer tabele rsa šifrata

4.3 Postavka i objašnjenje eksperimentalnog okruženja

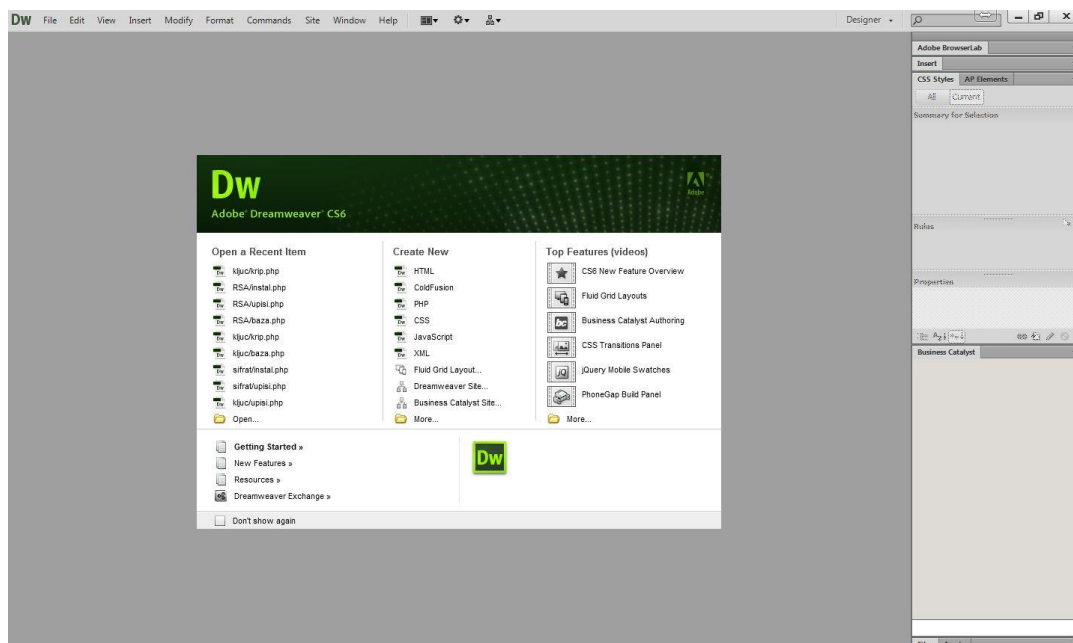
Eksperimentalna okruženja koja su korišćena za razvoj predloženih rešenja su: *NetBeans IDE 7.3*, *Dreamweaver CS6*, *MySQL* i *XAMPP*.

NetBeans okruženje je odabrano za izradu Java aplet aplikacije, korišćen je zbog jednostavnosti upotrebe kao i mogućnosti testiranja Java apleta direktno u razvojnom okruženju kroz html kod. Verzija jave koja za koju je namenjena aplikacija je 7 i novije, ova verzija je odabrana zbog odlične podrške kriptografskim funkcijama.



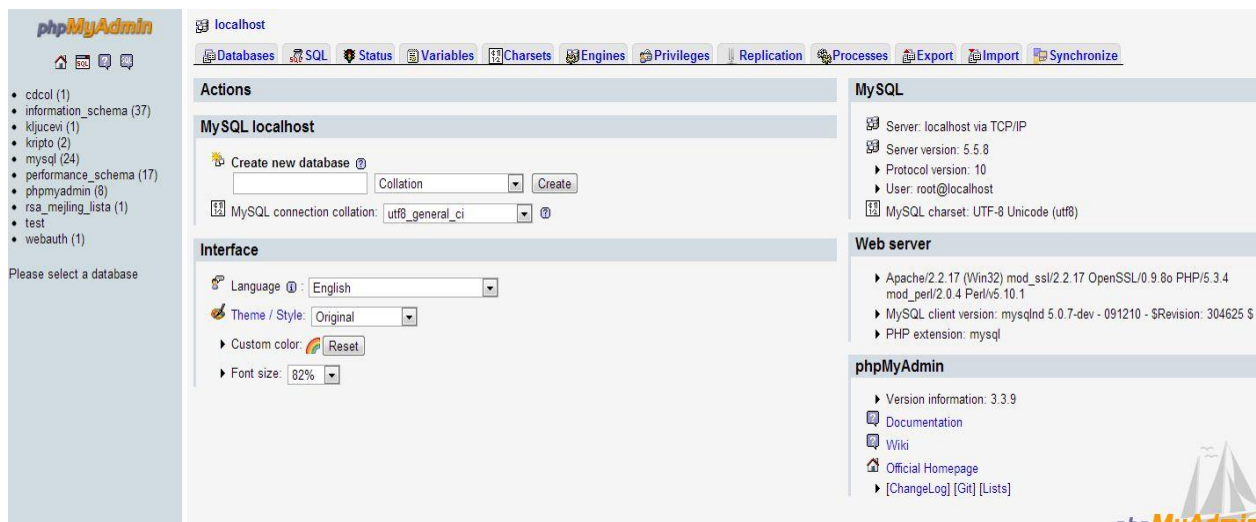
Ilustracija 24 NetBeans IDE 7.3

Za izradu PHP servisa korišćen je *Adobe Dreamweaver CS6* razvojno okruženje. Ovaj alat je odabran kao najbolja praksa izrade PHP aplikacija.



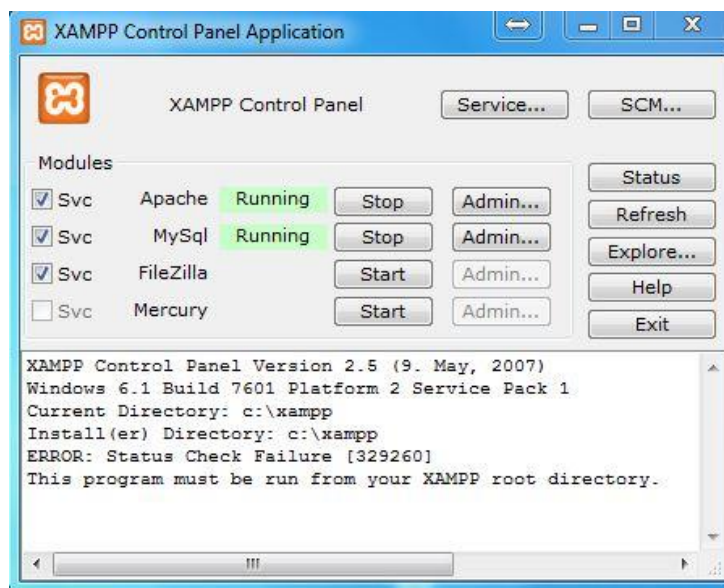
Ilustracija 25 Adobe Dreamweaver CS6

Baza podataka je pisana u SQL jeziku i za njen razvoj korišćena je MySQL baza podataka. MySQL baza podataka je odabrana kao najbolja praksa među besplatnim alatima za baze podataka.



Ilustracija 26 MySQL baza podataka

Za podizanje MySQL i Apache servera korišćen je alat XAMPP, kao alat sa odličnim ranim okruženjem i dodatnim alatima koji olakšavaju rad sa lokalnim Veb serverima odabran je i kao besplatno rešenje za vršenje eksperimenata.



Ilustracija 27 XAMPP Control Panel

4.4 Evaluacija ili performanse predloženog okruženja sa prikazom eksperimentalnih rezultata

Sledeći testovi su obavljeni na sledećim hardverskim specifikacijama:

Veb servisi su testirani upotrebom veb servera sa lokacijom u Nemačkoj, hosting kompanije „mtwebhosting“. Specifikacija procesora servera je *Intel xeon x3220 2,4 GHz Quad-core*. Brzina veze Veb servera je sinhrono 100Mb/s.

Korisnički računar kojim su testirana rešenja je desktop računar sa specifikacijom procesora *Intel core i3 3GHz Dual-core*. Brzina veze je 10/1Mb/s

Performanse prvog predloženog rešenja:

- Pokretanje Java aplet aplikacije od trenutka slanja zahteva za Java aplet: ~6.5s
- Vreme trajanja od trenutka pritiska na dugme za prosleđivanje do trenutka slanja podataka: ~3s
- Vreme trajanja šifrovanja ~0.7s

Performanse drugog predloženog rešenja:

- Pokretanje Java aplet aplikacije od trenutka slanja zahteva za Java aplet: ~6.1s
- Vreme trajanja od trenutka pritiska na dugme za prosleđivanje do trenutka slanja podataka: ~2s
- Vreme trajanja šifrovanja ~1s

4.5 Oblast primene predloženog rešenja

Dato rešenje se može primeniti u gotovo svim Veb okruženjima zarad povećanja privatnosti i bezbednosti. Najefikasnija primena je kod deljenih Veb servera pa čak i posvećenih servera gde imamo potrebu za uvođenjem zaštite od raznih napada i zaštitom osetljivih podataka korisnika, kao što su na primer:

- Zaštite mejling lista na deljenim Veb serverima
- Zaštite osetljivih korisničkih podataka na deljenim Veb serverima
- Zaštite fajlova na deljenim Veb serverima
- Zaštite od *SQL injection* napada
- Zaštite od krađe baze podataka od lica koja se nalaze na istom deljenom Veb serveru
- Zaštita privatnosti od zaposlenih lica u hosting kompaniji koja nam pruža usluge

5 Zaključak

5.1 Sumarni ciljevi rada

Predložili smo rešenje koje služi za unapređenje bezbednosti i privatnosti na deljenim Veb serverima. Ciljevi su postignuti sa oba predložena rešenja koja kombinuju različite tehnologije i kriptografske tehnike. Takođe tema je obrađena i sa aspekta podizanja svesti o svim bezbednosnim problemima i mogućim scenarijima zloupotrebe.

5.2 Ostvareni rezultati

Upoređujući predložena rešenja sa rešenjima iz 2010. godine, za rešavanje problema bezbednosti na deljenim Veb severima uglavnom se koristi kriptografija sa javnim ključevima, gde generički šema odgovara šemi koja je zasnovana na asimetričnoj kriptografiji, opisano u ovom radu. Uveli smo novi metod za zaštitu sa boljim rezultatima primenom kriptografskih mehanizama sa simetričnim ključevima. Glavno unapređenje ogleda se u postignutim boljim vremenskim performansama, gde je potrebno vreme smanjeno do 30% u režimima šifrovanja odnosno dešifrovanja, u korist kriptografskog sistema sa simetričnim ključevima.

Generička šema zasnovana na simetričnim kriptografskim mehanizmima značajno je kompleksnija i sadrži više funkcionalnih komponenti koje su neizostavne za kompletno funkcionisanje sistema. Iz tog razloga dobili smo znatno lošije rezultate u sa aspekta jednostavnosti, a to znači da sistem koji je zasnovan na simetričnom krypto sistemima je zahtevniji za implementaciju i održavanje. Ovakva osobina sistema značajno utiče na krajnju operabilnost, koja u značajnoj meri dalje utiče na ergonomiju rešenja i konačno društveno prihvatanje predloženog rešenja za zaštitu.

U predloženim rešenjima nisu rešeni problem autentifikacije i napad tipa „Čovek u sredini“. Na kraju, zaključujemo da konačno rešenje zahteva kombinovanje obe generičke šeme, jer na taj način rešavamo sve probleme koji nastaju usled nedostatka važnih kriptografskih servisa preko kojih se rešava problem iz domena autentifikacije i samim tim odbacuje mogućnost za napad „Čovek u sredini“.

5.3 Predlog za budući rad

Za budući rad predlažemo nastavak razvoja do sada predloženih rešenja u smeru implementacije servisa autentifikacije, radi sprečavanja od mogućih zloupotreba, gde treba implementirati servis autentifikacije između klijenta i servera. Predložili smo kombinaciju simetričnim i asimetrične kriptografije kao jedno od mogućih rešenja.

Takođe, predlaže se istraživanje zaštite otvorenog PHP koda, radi povećavanja bezbednosti rada PHP servisa u nebezbednim okruženjima. Trenutna rešenja nude slabu zaštitu PHP otvorenog koda, iz tog razloga je onemogućeno potencijalno umetanje tajnog simetričnog ključa u sam PHP servis na deljenom Veb serveru ili maskiranje rada PHP servisa. Na ovaj način bismo sprečili uvid u potencijalne slabosti sistema. Isti problem se javlja i u Java programskom jeziku, gde se kompajliranje koda ne smatra sigurnim pa se time mora izbegavati upisivanje tajnog simetričnog ključa u samu aplikaciju.

Na osnovu dosadašnjeg uvida u problem istraživanja, sama po sebi nameće se još jedna ideja, a to je razvoj softvera koji će omogućiti neometan rad aplikacije sa jedne strane, a sa druge bezbednost izvornog koda, odnosno na ovaj način ćemo onemogućiti potencijalnu zloupotrebu sa strane izvornog koda.

Literatura

- [1] Kriptologija I, Mladen Veinović, Saša Adamović, Univerzitet Singidunum
- [2] M. Stamp, Information security: principles and practice, 2nd ur., New Jersey: John Wiley & Soons, 2011.
- [3] A Practical Customer Privacy Protection on Shared Servers, Pramote Kuacharoen, School of Applied Statistics, National Institute of Development Administration
- [4] fips-197 Announcing the Advanced Encryption Standard (AES), National Institute of Standards and Technology (NIST), 2001
- [5] A database encryption solution that is protecting against external and internal threats, and meeting regulatory requirements, 2009
- [6] The Role of Cryptography in Database Security, Ueli Maurer, Department of Computer Science ETH Zurich
- [7] Public-Key Cryptography Standards- RSA Cryptography Specification, RFC 3447, ietf, 2003
- [8] Diffie-Hellman Key Agreement Method, RFC 2631, ietf, 1999
- [9] Data protection in databases at the local and server level, Saša Adamović, Marko Šarac, Dalibor Radovanović, Miloš Milenković, Singidunum University, International scientific conference, 2009
- [10] Bruce Schneier, Applied cryptography second edition
- [11] Bruce Schneier, Why Cryptography Is Harder Than It Looks, 1997