UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Jure Leban

VARNI PRIJAVNI SISTEM

DIPLOMSKO DELO
VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

Mentor: doc. dr. Tomaž Dobravec

Ljubljana, 2013

Univerza *v Ljubljani*
Fakulteta *za računalništvo in informatiko*

Tržaška 25
1001 Ljubljana, Slovenija
telefon: 01 476 84 11
01 476 83 87
faks: 01 426 46 47
01 476 87 11
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00353/2012
Datum: 01.10.2012

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **JURE LEBAN**

Naslov: **VARNI PRIJAVNI SISTEM**
A SECURE WEB AUTHORIZATION AND AUTHENTICATION SYSTEM

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Varni prijavni sistem omogoča varno prijavo in uporabo spletne strani ali spletne storitve.

V diplomskem delu preglejte različne obstoječe prijavne sisteme in jih primerjajte po več kriterijih (enostavnost uporabe, varnost, hitrost, ...). Opišite postopke in protokole, ki jih ti sistemi uporabljajo za zagotavljanje čim večje varnosti.

Zasnujte lasten sistem za varno prijavo v spletno stran. Uporabite napredne pristope za zagotavljanje varnosti. Sistem implementirajte z uporabo odprto kodnih rešitev in ga dobro dokumentirajte. Sistem naj bo prilagodljiv in naj omogoča preprosto uporabo tako za administratorja kot tudi za uporabnika.

Mentor:
doc. dr. Tomaž Dobravec

Dekan:
prof. dr. Nikolaj Zimic

Fakulteta za računalništvo in informatiko
2
UNIVERZA V LJUBLJANI

I Z J A V A O A V T O R S T V U

diplomskega dela

Spodaj podpisani Jure Leban,

z vpisno številko 63080278,

sem avtor diplomskega dela z naslovom:

Varni prijavni sistem

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Tomaža Dobravca
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne _____ Podpis avtorja: _____

ZAHVALA

Zahvaljujem se vsem, ki so mi kakorkoli pomagali in svetovali pri izdelavi diplomskega dela. Izpostavil bi predvsem mentorja doc. dr. Tomaža Dobravca, ki mi je bil med izdelavo diplomskega dela ves čas na voljo, starše, ki so me tako in drugače podpirali pri študiju in svoje dekle Tejo Vončina.

KAZALO VSEBINE

SEZNAM UPORABLJENIH KRATIC IN POJMOV

LAMPP je distribucija sistema Apache.
PowerDesigner je programsko orodje za načrtovanje podatkovnih baz.
NetBeans IDE je programsko orodje, ki se ga uporablja pri razvoju aplikacij.
PHP (PHP Hypertext Preprocessor) je skriptni programski jezik za ravoj spletnih strani.
jQuery je knjižnica za skriptni programski jezik JavaScript.
CSS (Cascading Style Sheets) je slogovni jezik za oblikovanje spletnih strani.
JavaScript je skriptni programski jezik za razvoj spletnih strani.
HTML (HyperText Markup Language) je označevalni jezik za razvoj spletnih strani.
OpenSSL je knjižnica z implementiranim protokolom TLS.
MySQL je sistem za upravljanje s podatkovnimi bazami.
SSL (Secure Socket Layer) je protokol, ki se ga uporablja za varno komunikacijo v internetu.
TLS (Transport Layer Security) je naslednik protokola SSL.
CMS (Content Management System) je sistem za upravljanje vsebin spletnih strani.
TFA (Two-factor authentication) je metoda za overjanje uporabnika.
RC4 je šifrirni algoritem.
SHA1 je zgostitvena funkcija.
ECDHE_RSA je kombinacija protokola RSA z eliptičnimi krivuljami in protokolom Diffie-Hellman.
DHE_RSA je kombinacija protokola RSA in protokola Diffie-Hellman.
RSA je algoritem, ki se ga uporablja pri kriptografiji javnih ključev.
SMS (Short Message Service) je storitev za prenos besedilnih sporočil.
AES (Advanced Encryption Standard) je šifrirni algoritem.
CAMELLIA je šifrirni algoritem.
PHPMailer je razred, katerega funkcije se uporablja pri pošiljanju elektronske pošte.
DN (Distinguished name) je razločevalno ime, ki ga vsebuje vsak digitalni certifikat.
SecureID je mehanizem, ki se ga uporablja za dvostopenjsko overjanje.
PIN (Personal identification number) je skrivno število, ki ga uporabnik rabi za uspešno overjanje.
MPJU je kratica, ki označuje Ministrstvo za pravosodje in javno upravo v Sloveniji.
3DES (Data Encryption Standard) je šifrirni algoritem.
EDE (Encrypt-Decrypt-Encrypt) je način delovanja 3DES algoritma.
CBC (Cipher-block chaining) je način bločnega šifriranja.
ECC (Elliptic curve cryptography) je pristop pri kriptografiji javnih ključev s pomočjo eliptičnih krivulj.
MITM (Man-in-the-middle attack) je eden od mnogih računalniških napadov.

DH (Diffie-Hellman) je kratica, ki označuje algoritem Diffie-Hellman. Uporablja se ga pri kriptografiji javnih ključev.
Linux je operacijski sistem.
SUPB je kratica, ki označuje sistem za upravljanje s podatkovno bazo.
Apache je odprtokoden spletni strežnik.
HTTPS (Hypertext Transfer Protocol Secure) je varna različica protokola http.
MAC (Message authentication code) je koda za zagotavljanje celovitosti in pristnosti sporočila.
Cipher suite je nabor protokolov pri rokovanju SSL.
PKI (Public-key infrastructure) je kratica, ki označuje infrastrukturo javnih ključev.
CA (Certification authority) je kratica, ki označuje certifikatno agencijo.
Root CA je vrhovni overitelj v infrastrukturi javnih ključev.
X.509 je standard digitalnih certifikatov.
CSR (Certificate Signing Request) je zahteva po izdaji digitalnega certifikata.

POVZETEK

V diplomski nalogi je podrobneje predstavljenih nekaj spletnih prijavnih sistemov, ki so večini dobro znani. Podal sem svojo subjektivno kritiko, ki ne predstavlja profesionalne študije opisanih prijavnih sistemov, ampak je zgolj moje mnenje. Predstavljene so tudi glavne funkcionalnosti in orodja, ki sem jih uporabil za izdelavo lastnega prijavnega sistema. Vsem prijavnim sistemom, vključno z mojim, je skupno to, da komunikacija med strežnikom in odjemalcem poteka preko varne povezave HTTPS oz. preko protokola HTTP, ki za varnost uporablja protokol SSL. Že med samo izdelavo aplikacije se mi je porodila ideja, da bi bil prvi korak k izboljšanju varnosti aplikacije izvedba oz. implementacija dvostopenjskega overjanja, s čimer potencialnemu napadalcu močno otežiš delo. V diplomskem delu sem predstavil tudi kriptografijo javnih ključev, glavne razlike med simetrično in asimetrično kriptografijo ter uporabo in zgradbo digitalnih certifikatov.

Ključne besede: HTTPS, SSL, dvostopenjsko overjanje, digitlani certifikat, kriptografija javnih ključev

ABSTRACT

The thesis presents in detail some well-known web applications. I gave a subjective critique which doesn't represent proffesional study, it's just my opinion. The thesis also presents the main features and tools that have been used in the development of my own web applicaton system. Public-key criptography which contains detail presentation of use and structure of digital certificates and differences between symmetric and asymmetric criptography. All login systems, which are mentioned hereafter, have in comon that their communication between client and web server is taking place via secure connection called HTTPS. During the development I came up with idea that the first step to improve the safety would be the implementation of two-factor authentication (TFA), which makes a job difficult to a potential attacker.

Keywords: HTTPS, SSL, two-factor authentication, digital certificate, public-key criptography

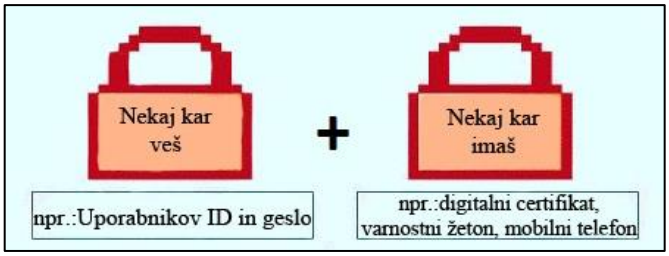
1 UVOD

Prijavni sistem nam omogoča overjanje oz. prijavo v spletno aplikacijo, ki uporabniku nudi varno uporabo storitve. Za zagotavljanje varnosti se uporablja različne protokole, mehanizme in pristope, katerih glavni namen je, da uporabnika ter njegove podatke čim bolje zavarujejo pred zlonamernimi programi oziroma ljudmi, ki zlonamerne programe pišejo. V nadaljevanju bo govora o različnih prijavnih sistemih, ki za zagotavljanje varnosti uporabljajo različne pristope in metode. Predstavljena bo tudi spletna aplikacija in razvoj le-te, s katero sem se poskušal čim bolj približati ne samo prijavnemu sistemu, ampak prijavnemu sistemu, ki je varen. Overjanje uporabnika bo tako možno na dva načina, in sicer z uporabniškim imenom ter geslom ali pa z digitalnim certifikatom. Kaj overjanje pravzaprav sploh pomeni? Overjanje je proces zanesljivega preverjanja identitete nekoga ali nečesa [1]. Pri varnosti je govora tudi o pooblastitvi, ki se ne more zgoditi brez predhodnega overjanja [2]. Pooblastitev je proces preverjanja, če lahko narediš to, kar imaš namen storiti [2].

Torej nam mora varni prijavni sistem omogočati tako overjanje kot pooblastitev uporabnika. Kot sem že prej omenil, je bilo potrebno za uspešno opravljen prvi del diplomskega dela razviti spletno aplikacijo, ki bi omogočala prijavo na prej omenjena načina. Razdeljena bi bila na dva dela – uporabniški in administratorski. Za reševanje problema pooblastitve pa sem uporabil podoben pristop, kot so ga razvijalci uporabili pri Linuxu. V omenjenem operacijskem sistemu so uporabniki organizirani v tri skupine: Uporabnik (*ang. users*), grupa (*ang. groups*), ostali (*ang. others*). Da pa bi komunikacija med odjemalcem in strežnikom potekala varno, pa bo skrbel protokol SSL (Secure Sockets Layer). SSL je kriptografski protokol, ki omogoča varno komunikacijo v internetu [3].

2 VARNI PRIJAVNI SISTEMI

V tem poglavju bom opisal različne prijavne sisteme ter jih med seboj primerjal. Nekateri nam omogočajo prijavo samo z uporabniškim imenom in geslom, drugi tudi z digitalnim certifikatom. Nekateri prijavni sistemi so šli dlje in uvedli dvostopenjsko overjanje. Filozofija dvostopenjskega overjanja (ang. Two-factor authentication - TFA) temelji na vsaj dveh od treh prijavnih faktorjev. To so: nekaj, kar uporabnik ve; nekaj, kar uporabnik ima; in nekaj, kar uporabnik je [4]. Po navadi se uporabljata prva dva faktorja, kar prikazuje slika 1. Tretji faktor (nekaj, kar uporabnik je) se uporablja pri zelo občutljivih sistemih, ki potrebujejo visoko stopnjo varnosti, in spada pod biometrično overjanje. Uporabnika se biometrično overi s skeniranjem prstnega odtisa, šarenice očesa ali pa s prepoznavanjem glasu. Prvi faktor je laični javnosti najbolj znan, saj se uporablja v veliko primerih. Ko se uporabnik prijavi v svoj e-poštni predal, je primoran posredovati svoje uporabniško ime in geslo, torej nekaj, kar ve. Tudi drugi prijavni faktor je širši množici kar dobro znan, saj se digitalni certifikati in generatorji enkratnih gesel zelo pogosto uporabljajo pri prijavi v bančne sisteme. Prijavni sistemi imajo velikokrat možnost, da se uporabnik sam odloči, na kakšen način se bo overjanje izvajalo: ali se bo uporabljalo enostopenjsko ali dvostopenjsko overjanje. Z uporabo dvostopenjskega načina overjanja bomo potencialnemu napadalcu močno otežili delo. V primeru, da se napadalec nekako dokoplje do našega uporabniškega imena in gesla, nima pa generatorja enkratnih gesel, ne bo mogel izvesti popolnega overjanja.

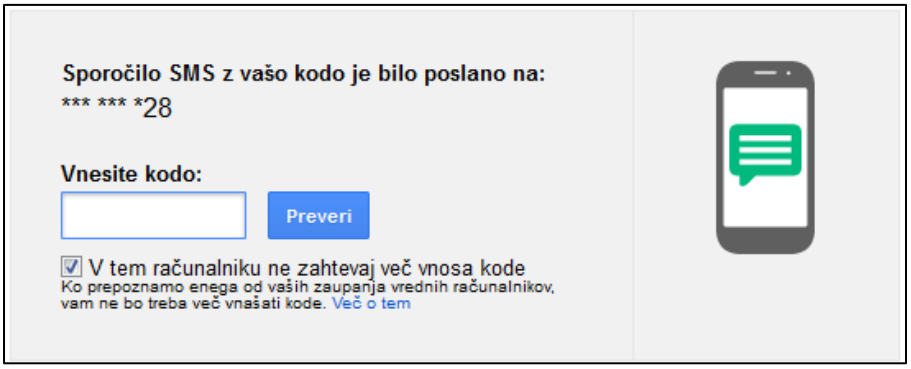


Slika 1: Dvostopenjsko overjanje [5]

2.1 Storitev Gmail

Ena od mnogih storitev, ki jih ponuja spletni velikan Google, je tudi elektronska pošta. Za dostop do poštnega predala se mora uporabnik avtenticirati tako, da vnese uporabniško ime in geslo. Uporabnik ima možnost, da v nastavitvah nastavi preverjanje uporabnika v dveh korakih. Ko se naslednjič poizkusimo prijaviti v poštni predal, storitev po uspešnem overjanju s pomočjo uporabniškega imena in gesla od nas zahteva vnos šestmestnega števila, ki smo ga pred tem v obliki SMS sporočila prejeli na telefon. Šele, ko vnesemo to število, nas program spusti naprej. Primer je prikazan na sliki 2. Preverjanje v dveh korakih je zelo dober varnostni

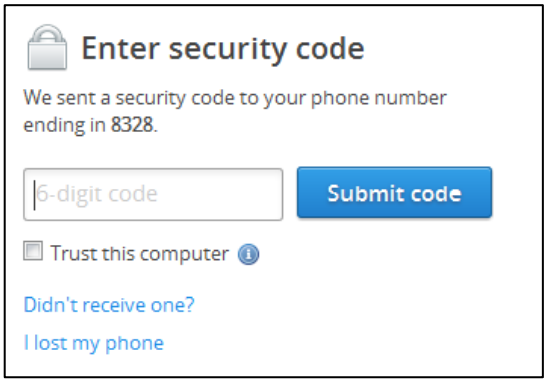
mehanizem. Predvsem je priročno to, da lahko svoj domači računalnik dodaš med zaupanja vredne računalnike, zato ti ni potrebno vedno vnašati prejetega naključnega števila. Vnos je potreben samo takrat, ko se želiš prijaviti z računalnika, ki ni na listi zaupanja vrednih računalnikov. Za varnost je poskrbljeno s protokolom TLS 1.1. Za izmenjavo simetričnega ključa, ki se ga kasneje uporabi za šifriranje povezave, se uporablja protokol ECDHE_RSA. Za šifriranje povezave pa se uporablja protokol RC4, katerega simetrični ključ je dolg 128 bitov. Za dokazovanje pristnosti sporočila se uporablja zgostitvena funkcija SHA1. Strežnik se nam predstavi z digitalnim certifikatom podpisanim s strani certifikatne agencije Thawte.



Slika 2: Preverjanje v dveh korakih

2.2 Storitev Dropbox

Storitev Dropbox nam omogoča hranjenje podatkov v oblaku, nudi pa tudi sinhronizacijo z domačim računalnikom. Seveda tudi ta storitev od nas zahteva registracijo. Po uspešno izvedeni registraciji nam na podan elektronski naslov pošljejo potrditveno pismo s podatki za prijavo. Dropbox ima možnost dvostopenjske prijave. Ko se prvič prijavimo v sistem, lahko to omogočimo v nastavitvah. Dvostopenjska prijava poteka tako, da najprej vpišeš svoje uporabniško ime in geslo. Naslednji korak je prikazan na sliki 3. Sistem od nas zahteva šestmestno naključno število, ki smo ga prejeli na mobilni telefon preko sporočila SMS. Šele ko vnesemo to naključno število, smo dokončno avtenticirani. Sistem za prenos in hrambo podatkov uporablja protokol TLS 1.1 v kombinaciji z AES-256-bitno enkripcijo. Za izmenjavo ključev sta uporabljena protokol ECDHE_RSA in zgostitvena funkcija SHA1 za preverjanje pristnosti prenesenih sporočil. Strežnik se nam predstavi z digitalnim certifikatom, ki je podpisan s strani certifikatne agencije Thawte.



Slika 3: Zahteva po naključnem številu

2.3 Storitev Bank@net

Bank@net je spletna aplikacija za spletno bančništvo, katere lastnik je banka Nova KBM d. d. Omenjena banka ima dve aplikaciji za opravljanje spletnega bančništva. Ena je t.i. Bank@net, ki je namenjena navadnim uporabnikom, druga pa je poslovni Bank@net, namenjena uporabnikom, ki imajo pri banki odprt poslovni račun. Največja razlika med njima je ta, da poslovni Bank@net omogoča tudi uporabo digitalnega potrdila za overjanje uporabnika. V nadaljevanju bom govoril o različici za navadne uporabnike. Tudi ta storitev od uporabnika zahteva vnos uporabniškega imena in gesla, ki ju pridobiš na banki. Če želiš opravljati tudi zahtevnejše operacije, kot so plačila računov, prenosi sredstev med računi, pa je uporabniku na voljo tudi overjanje z enkratnimi gesli, ki jih generira identifikacijska kartica SecureID prikazana na sliki 4.

Slika 4: Identifikacijska kartica SecureID [6]

Storitev uporablja protokol TLS 1.0 v kombinaciji z algoritmom RC4, katerega dolžina simetričnega ključa je 128 bitov. Za izmenjavo ključa, ki je v nadaljevanju uporabljen za šifriranje povezave, je uporabljen protokol RSA. Za pristnost poslanih sporočil med uporabnikom in strežnikom pa aplikacija uporablja zgostitveno funkcijo SHA1. Tudi pri tej aplikaciji je poskrbljeno, da lahko uporabnik v nastavitvah izbere način prijave. Načina prijave sta dva – navaden, kjer vtipkamo uporabniško ime in geslo, ter prijava s pomočjo kode PIN in kartice SecureID, ki spada pod dvostopenjsko overjanje. Najprej namreč vtipkamo

uporabniško ime, geslo pa je sestavljeno iz kode PIN, ki spada v prvi prijavni faktor (nekaj, kar vemo), in naključnega števila generiranega s strani kartice SecureID, ki spada v drugi prijavni faktor (nekaj, kar imamo). Strežnik banke se nam predstavi z digitalnim certifikatom podpisanim s strani certifikatne agencije VeriSign.

2.4 Storitev e-Ucilnica na FRI

Storitev temelji na odprtokodnem sistemu Moodle. Moodle je t. i. CMS (Course Management System), s pomočjo katerega lahko postavimo virtualno učilnico oz. e-učilnico. Tako virtualno učilnico uporablja tudi Fakulteta za računalništvo in informatiko v Ljubljani (v nadaljevanju: FRI). Tudi tu se je potrebno v sistem prijaviti. Prijava poteka s pomočjo uporabniškega imena in gesla. Storitev ne uporablja dvostopenjskega overjanja. Še do pred kratkim storitev ni uporabljala varnostnega protokola TLS. Sedaj za varnost skrbi protokol TLS 1.1. Za izmenjavo simetričnega ključa se uporablja protokol DHE_RSA. Za šifriranje povezave pa je poskrbljeno s simetričnim algoritmom CAMELLIA z dolžino ključa 256 bitov. Za preverjanje pristnosti sporočil pa se največkrat uporablja funkcijo SHA1 z dolžino izhodnega bloka 160 bitov. Strežnik svojo identiteto dokazuje z digitalnim certifikatom, ki je izdan s strani nizozemske certifikatne agencije TERENA.

2.5 Storitev e-Student na FRI

Storitev e-student je namenjena študentom in zaposlenim na FRI. Študenti imajo pregled nad ocenami, lahko pa naročajo tudi potrdila o vpisu ter se prijavljajo oz. odjavljajo z izpitnih rokov. Profesorji storitev uporabljajo za vpisovanje ocen, razpisovanje izpitnih rokov in podobno. Prijava poteka na dva načina. Lahko se prijaviš z uporabniškim imenom in geslom, vendar je število prijav omejeno. Glavni način prijave poteka s pomočjo digitalnega certifikata. Digitalni certifikat je možno zaprositi pri slovenski certifikatni agenciji SIGEN-CA (Slovenian General Certification Authority), katerega izdajatelj je Ministrstvo za pravosodje in javno upravo (MPJU), ali pa pri certifikatni agenciji @friCA, ki deluje interno v okviru FRI. Strežnik storitve e-student se nam predstavi z digitalnim certifikatom izdanim s strani prej omenjene certifikatne agencije, ki ni mednarodno registrirana. Zaradi tega nas brskalnik obvesti z opozorilom, da digitalno potrdilo izdajatelja ni zaupanja vredno, kar je prikazano na sliki 5. Ker pa vemo, da certifikatna agencija deluje interno, znotraj FRI, nas opozorilo ne odvrne od nadaljevanja postopka. Uporabljen je protokol TLS 1.0. Za šifriranje povezave se uporablja simetrični algoritem 3DES_EDE_CBC, katerega dolžina ključa je 112 bitov. Za izmenjavo simetričnega ključa se uporablja asimetrični algoritem DHE_RSA. Za preverjanje pristnosti poslanih sporočil je tudi tu uporabljena zgoščevalna funkcija SHA1.

Slika 5: Potrdilo ni vredno zaupanja

2.6 Povzetek

Od zgornjih storitev bi izpostavil dve, za kateri menim, da bi bila primerjava najbolj zanimiva. To sta storitvi Dropbox in Bank@net. Primerjava je prikazana v tabeli 1.

Storitev	Verzija TLS	Algoritem za izmenjavo simetričnega ključa	Algoritem za šifriranje povezave	Funkcija za preverjanje pristnosti sporočil	Dvostopenjsko overjanje
Dropbox	1.1	ECDHE_RSA	AES-256	SHA1	DA (SMS)
Bank@net	1.0	RSA	RC4-128	SHA1	DA (SecureID)

Tabela 1: Primerjava med storitvama Dropbox in Bank@net

V zgornji tabeli lahko opazimo, da storitev Dropbox za zagotavljanje varnosti uporablja naprednejše algoritme. Zanimivo je dejstvo, da je Dropbox storitev, ki nam omogoča shranjevanje datotek v oblaku, storitev Bank@net pa nam omogoča bančno poslovanje. Torej bi morala biti Bank@net vsaj tako varna kot Dropbox. Slednji uporablja protokol TLS verzije 1.1, ki mu je bilo dodanih nekaj varnostnih popravkov (sprememba inicializacijskega vektorja, zaščita proti napadom na CBC način šifriranja,…) [7]. Naslednja razlika je pri algoritmu za šifriranje povezave. Algoritem AES se smatra za boljšo izbiro od algoritma RC4. AES je novejši in uporablja bolj komplicirane metode šifriranja, medtem ko je RC4 v primerjavi z AES hitrejši, saj spada v skupino pretočnih šifrirnih algoritmov, ki šifrirajo vsak bit sporočila posebej. AES spada v skupino blokovnih šifrirnih algoritmov. Kot že samo ime pove, le-ti sporočilo razdelijo na bloke enake dolžine ter vsakega posebej šifrirajo, kar je počasneje v primerjavi s pretočnim šifriranjem. Nepravilna uporaba algoritmov, ki delujejo na

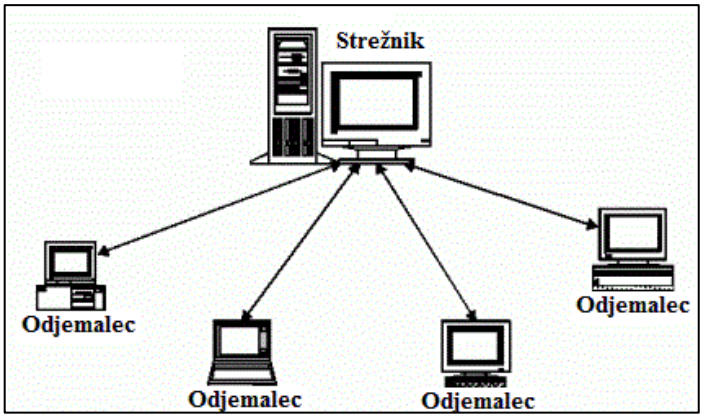
način pretočnega šifriranja, lahko privede do varnostnih problemov. Velika razlika je tudi med algoritmoma za izmenjavo simetričnega ključa. Dropbox uporablja kombinacijo večih algoritmov. Zapis ECDHE_RSA si razložimo na naslednji način: Za izmenjavo ključa se uporablja efemeren (kratkotrajen) Diffie-Hellman v kombinaciji z eliptičnimi krivuljami-ECC (*ang. Elliptic Curve Criptography*), katerih uporaba omogoča generiranje ključev krajše dolžine, ki zagotavljajo enako stopnjo varnost kot uporaba daljših ključev. Primerjava je prikazana v tabeli 2 [8]. Za overjanje je poskrbljeno s protokolom RSA. Vse, kar strežnik pošlje, se podpiše z njegovim javnim ključem. Efemerni Diffie-Hellman v kombinaciji z algoritmom RSA omogoča tajnost za naprej, medtem ko navaden RSA tega ne omogoča. Vzrok je v tem, ker se za vsako sejo uporabi drug javni ključ. V primeru, da bi napadalec zlomil javni ključ, bi lahko dekriptiral samo eno sejo. Na drugi strani pa navaden RSA za vsako sejo uporablja isti javni ključ. Obe storitvi imata možnost dvostopenjskega overjanja. Bank@net s pomočjo identifikacijske kartice SecureID, ki nam generira naključno šestmestno število, Dropbox pa nam na telefon pošlje šestmestno naključno število, ki ga po uspešno vnesenem uporabniškim imenom in geslom naknadno vpišemo. Oba načina sta ranljiva na t. i. MITM (*ang. Man In The Middle*) napade.

Simetrični	ECC	DH/DSA/RSA – asimetrični
80	163	1024
112	233	2048
128	283	3072
192	409	7680
256	571	15360

Tabela 2: Primerjava med dolžinami ključev v bitih med posameznimi algoritmi, ki zagotavljajo enako stopnjo varnosti

3 RAZVOJ APLIKACIJE

Razvili smo aplikacijo, ki omogoča varno prijavo v sistem, podobno, kot storitve, ki sem jih prej opisoval. Deluje na osnovnem principu odjemalec-strežnik, kot je prikazano na sliki 6. Podatki so shranjeni na strežniku, odjemalec preko uporabniškega vmesnika do njih dostopa in z njimi upravlja. Prioritetna platforma, na kateri bo aplikacija tekla, je platforma Linux.



Slika 6: Model odjemalec-strežnik [9]

3.1 Analiza zahtev

Prijavni sistem nam bo omogočal prijavo na dva načina:

- prijava z uporabniškim imenom in geslom,
- prijava z digitalnim certifikatom.

Uporabnik se lahko prijavi v različne skupine. Vsaka skupina ima enega ali več administratorjev, ki s skupino upravljajo in imajo naslednje privilegije:

- urejanje obstoječih uporabnikov,
- dodajanje novih uporabnikov,
- podvajanje določenega uporabnika v drugo skupino,
- spreminjanje prijavne strani skupine po svojem okusu.

Navaden uporabnik ima naslednje privilegije:

- spreminjanje prijavnega gesla,
- v primeru, da uporabnik (navaden ali administrator) pozabi geslo, lahko zaprosi za ponastavitev le-tega,

- uporabnik lahko zaprosi za uporabniško ime in geslo ter certifikat.

Obstaja tudi administrator celotnega sistema z naslednjimi privilegiji:

- počne lahko vse, kar počne administrator skupine,
- ureja trenutne skupine,
- dodaja nove skupine.

3.2 Uporabljena orodja in tehnologije za razvoj aplikacije

- LAMPP
- PowerDesigner
- NetBeans IDE
- PHP
- jQuery
- CSS
- JavaScript
- HTML
- OpenSSL
- MySQL

LAMPP (**L**inux, **A**pache, **M**ySQL, **P**HP) je skupek programske opreme, ki razvijalcu močno olajša postavitev spletnega strežnika. Paket Lampp se uporablja na operacijskem sistemu Linux. Paket vsebuje spletni strežnik Apache, MySQL SUPB (sistem za upravljanje s podatkovno bazo) in skriptni jezik PHP (PHP Hypertext Preprocessor) za razvoj dinamičnih spletnih strani.

PowerDesigner je orodje za razvoj modelov podatkovne baze. Predvsem mi je olajšalo načrtovanje konceptualnega in logičnega modela podatkovne baze. Iz logičnega modela se na koncu generira fizični podatkovni model, ki ga uvoziš v prej omenjeni SUPB.

NetBeans je integrirano razvojno okolje (*ang. IDE*), v katerem se primarno razvija s programskim jezikom Java, možna pa je tudi uporaba skriptnega jezika PHP.

PHP je, kot sem že prej omenil, skriptni jezik za razvoj dinamičnih spletnih strani. Izvaja se na strežniku.

JavaScript je ravno tako skriptni jezik, vendar se le-ta izvaja na strani odjemalca. Predvsem se dobro izkaže pri posrednem komuniciranju s podatkovno bazo, kar nam omogoča dinamičen prikaz podatkov na strani.

jQuery je knjižnica na osnovi skriptnega jezika JavaScript. Razvita je bila predvsem zato, da bi se razvijalcem olajšalo dinamično spreminjanje HTML-ja.

CSS (**C**ascading **S**tyle **S**heets) je slogovni jezik, ki se uporablja za oblikovanje predlog spletnih strani. Razvit je bil zato, da se vsebina spletne strani loči od izgleda oz. predloge.

HTML (**H**yper **T**ext **M**arkup **L**anguage) je označevalni jezik za izdelavo spletnih strani, ki predstavlja osnovno vsake spletne strani.

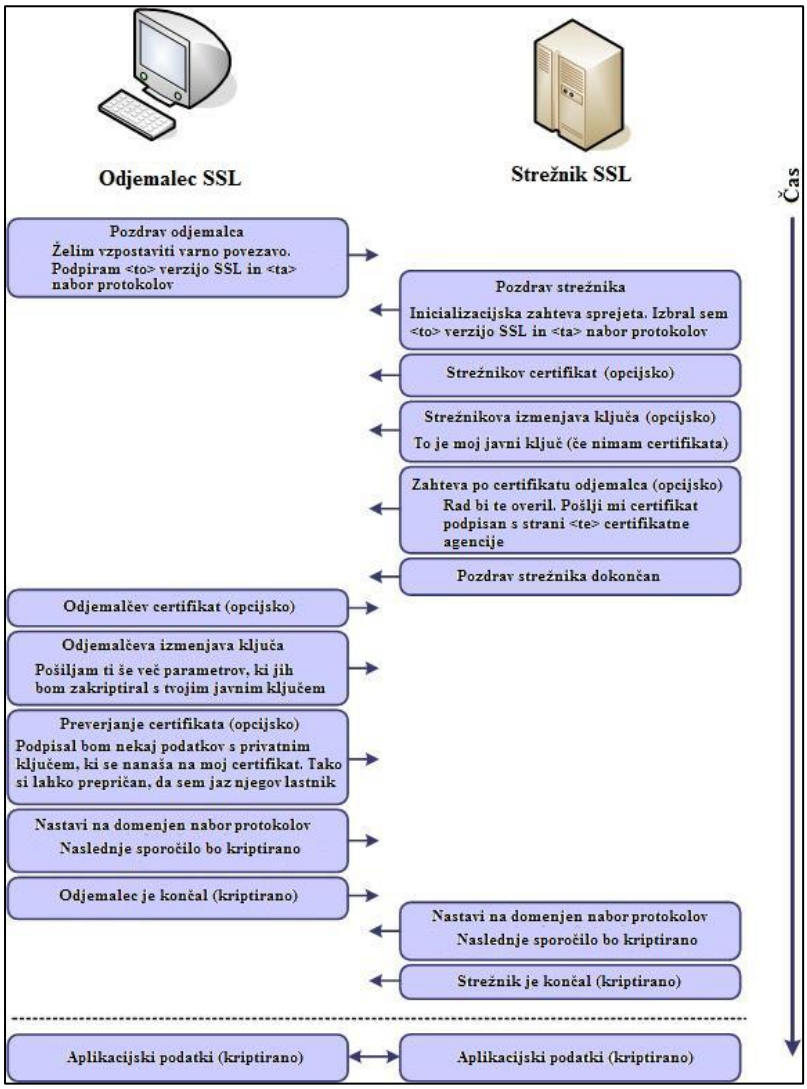
OpenSSL je odprtokodna knjižnica napisana v programskem jeziku C z implementiranim protokolom TLS. Dobiš jo skupaj z odprtokodnim paketom Lampp. Uporabil sem jo za izdelavno digitalnih certifikatov ter za izvedbo HTTPS (**H**yper **T**ransfer **P**rotocol **S**ecure) protokola. HTTPS dejansko ni protokol, ampak je protokol HTTP (**H**yper **T**ransfer **P**rotocol), ki smo mu kot podplast dodali protokol TLS (**T**ransport **L**ayer **S**ecurity), ki omogoča kriptiranje segmentov povezave na aplikacijski ravni. Le-ti so potem preko predstavitvene plasti in plasti seje poslani na transportno plast.

MySQL je, kot sem prej omenil, sistem za upravljanje s podatkovnimi bazami (SUPB), ki za upravljanje s podatki uporablja jezik SQL (**S**tructured **Q**uery **L**anguage).

Ker je bila pri razvoju varnega prijavnega sistema poglavitvena naloga vpeljava protokola SSL/TLS v kombinaciji z uporabo digitalnih certifikatov, katerih uporaba se pokaže pri kriptografiji z javnimi ključi, bom v nadaljevanju oba tudi opisal.

3.3 Delovanje SSL/TLS protokola

Kot sem že prej omenil, protokol TLS ovijemo okoli protokola HTTP. Rezultat tega je protokol HTTPS, ki nam omogoča varno komunikacijo v internetu. Razčistiti je potrebno tudi rabo imena SSL/TLS. Protokol SSL je samo predhodnik protokola TLS. V bistvu gre za en in isti protokol, le da se je ime naslednika spremenilo [10]. Cilj protokola je, da se med stranema vzpostavi varen kanal, po katerem bo potekala komunikacija. Odjemalec in strežnik se dogovorita, kakšen nabor protokolov (*ang. cipher suite*) bosta uporabila za vzpostavitev varnega kanala. Nabor protokolov je sestavljen iz protokola za izmenjavo simetričnega ključa, protokola za kriptiranje povezave in funkcije za preverjanje pristnosti sporočil (*ang. MAC-**M**essage **A**uthentication **C**ode*). Delovanje je prikazano na sliki 7. Razlaga se po korakih nahaja pod sliko.



Slika 7: Delovanje protokola SSL (SSL rokovanje) [11]

- Odjemalec strežniku pošlje sporočilo CLIENT HELLO (*sl. Pozdrav odjemalca*), ki vsebuje: najvišjo verzijo protokola TLS, ki ga odjemalec podpira, naključno število, seznam simetričnih protokolov za šifriranje povezave, seznam asimetričnih protokolov za izmenjavo simetričnega ključa, seznam zgoščevalnih funkcij za zagotavljanje integritete sporočil, seznam kompresijskih metod, ki jih odjemalec podpira, in opcijsko ID seje, ki je bila lahko vzpostavljena že prej.
- Strežnik odgovori s sporočilom SERVER HELLO (*sl. Pozdrav strežnika*), ki vsebuje: izbrano najvišjo verzijo protokola TLS, ki ga podpirata oba, naključno število, protokole za kriptiranje povezave, izmenjavo ključa, zgoščevalno funkcijo za zagotavljanje integritete in kompresijsko metodo. Vse je bilo izbrano s seznama, ki ga je poslal odjemalec. V primeru, da strežnik prejme tudi ID seje, preveri še svoj predpomnilnik (*ang. cache*), in sicer, ali že obstaja tak ID seje, ki se ujema s prejetim. Če obstaja in se strežnik odloči, da ga ponovno uporabi, se v sporočilo doda tudi vrednost ID-ja seje, sicer bo vrednost drugačna, kar pomeni, da se vzpostavi nova seja.
- V primeru, da se zahteva overjanje strežnika, le-ta pošlje sporočilo SERVER CERTIFICATE (*sl. Strežnikov certifikat*), ki vsebuje strežnikov certifikat.
- V primeru, ko sporočilo SERVER CERTIFICATE ne vsebuje dovolj podatkov, da bi odjemalec lahko poslal skupno naključno število (*ang. premaster secret*), ki se ga izračuna iz naključnega števila, ki je bilo prejeto v SERVER HELLO sporočilu, strežnik pošlje SERVER KEY EXCHANGE (*sl. Strežnikova izmenjava ključa*) sporočilo. Sporočilo vsebuje podatke, na podlagi katerih odjemalec potem lahko pošlje naključno število.
- Takoj za tem strežnik pošlje sporočilo CERTIFICATE REQUEST (*sl. Zahteva po certifikatu*). Strežnik od odjemalca zahteva certifikat, podpisan s strani določene certifikatne agencije (*ang. CA*).
- Strežnik na koncu pošlje sporočilo SERVER HELLO DONE (*sl. Strežnikov pozdrav dokončan*), kar pomeni, da je strežnik končal s pošiljanjem sporočil in čaka na potezo odjemalca.
- Če strežnik zahteva odjemalčev certifikat, potem le-ta pošlje CLIENT CERTIFICATE (*sl. Odjemalčev certifikat*) sporočilo, ki ta certifikat vsebuje.
- Takoj zatem odjemalec pošlje CLIENT KEY EXCHANGE (*sl. Odjemalčeva izmenjava ključa*) sporočilo, ki vsebuje: skupno naključno število (*ang. premaster secret*), ki ga s strežnikovim javnim ključem zakriptira in pošlje. Strežnik ga nato s svojim privatnim ključem dekriptira.
- Odjemelc pošlje CERTIFICATE VERIFY (*sl. Preverjanje certifikata*) sporočilo, ki vsebuje podpis prej poslanih sporočil. Podpis je izvršil s svojim privatnim ključem. Strežnik lahko s odjemelčevim javnim ključem preveri, ali ima odjemelc dostop do svojega privatnega ključa in ali je res lastnik tega certifikata.

- Odjemalec in strežnik nato iz naključnih števil, ki sta jih prejela, in skupnega naključnega števila, izračunata glavno naključno število (*ang. master secret*). To število je dejansko simetrični ključ, ki se ga uporabi za šifriranje povezave.
- Odjemalec nato pošlje CHANGE CIPHER SPEC (*sl. Spremeni na domenjeni nabor protokolov*) sporočilo, ki pomeni, da bo vsa nadaljnja komunikacija šifrirana z izbranim simetričnim protokolom.
- Odjemalec na koncu pošlje še CLIENT FINISHED (*sl. Odjemalec je končal*) sporočilo. To je kriptirano in vsebuje MAC (*ang. Message Authentication Code*), ki se ga izračuna na podlagi vseh do sedaj poslanih in prejetih sporočil, ter rezultat zgoščevalne funkcije (*ang. hash*), ki se ga ravno tako izračuna na podlagi vseh prejetih in poslanih sporočil. MAC ščiti tako celovitost (integriteto) kot pristnost (avtentičnost) sporočila.
- Strežnik nato sporočilo dekriptira in preveri obe vrednosti. Če se kakšna izmed vrednosti ne ujema, se seja prekine.
- Tudi strežnik na koncu pošlje CHANGE CIPHER SPEC sporočilo, ki pomeni, da bo vsa nadaljnja komunikacija šifrirana z izbranim simetričnim protokolom.
- Strežnik ravno tako pošlje SERVER FINISHED (*sl. Strežnik je končal*) sporočilo, ki vsebuje enake parametre kot sporočilo CLIENT FINISHED.
- Odjemalec ravno tako preveri integriteto in pooblastitev sporočila.
- Od tu naprej je vsa komunikacija med strežnikom in odjemelcem šifrirana [12, 13].

3.4 Delovanja algoritma RSA

Ker je algoritem RSA največkrat uporabljen algoritem za izmenjavo ključa, bom v nadaljevanju predstavil njegovo delovanje. RSA je algoritem za kriptografijo z javnimi ključi. Spada v družino asimetričnih algoritmov. Pojem je predstavljen v naslednjem poglavju. Algoritem deluje na principu faktorizacije velikih števil. Ime je dobil po treh akterjih, ki so algoritem leta 1977 prvič predstavili javnosti. To so: Ron Rivest, Adi Shamir in Leonard Adleman. Enak algoritem je leta 1973 razvil angleški matematik Clifford Christopher Cocks, vendar je bil zaupne narave do leta 1997 [14].

3.4.1 Izračun ključev

Za izračun javnega in privatnega ključa se uporablja naslednji postopek:

- Izberi dve naključni, različni praštevili `p` in `q`
- Izračunaj vrednost `n` po formuli prikazani v naslednji enačbi:

$$n = p \cdot q$$

- Izračunaj Eulerjevo funkcijo po formuli prikazani v naslednji enačbi:

$$\varphi(n) = (p-1) \cdot (q-1)$$

- Izberi celo število e, za katerega velja $1 < e < \varphi(n)$. Veljati mora tudi, da sta si števili e in $\varphi(n)$ tuji si števili. Iz tega sledi, da je njun največji skupni delitelj število 1. Izračuna se ga po formuli prikazani v naslednji enačbi:

$$\gcd(e, \varphi(n)) = 1$$

- Izračunaj d po formuli, ki je prikazana v naslednji enačbi:

$$d \cdot e \equiv 1 \pmod{\varphi(n)}$$

Javni ključ je sestavljen iz števil n in e, privatni ključ pa iz števil n in d. Pri tem je e javni eksponent, s katerim se šifrira, d pa privatni eksponent, s katerim se dešifrira. Recimo, da želi Metka poslati Janku skrivno geslo. Janko z upoštevanjem zgornjih korakov izračuna javni in privatni ključ. Javni ključ (n,e) pošlje Metki. Metka z Jankovim javnim ključem zašifrira sporočilo, ki ga predstavi kot naravno število m. Formula je prikazana v naslednji enačbi:

$$kriptogram = m^e \pmod{n}$$

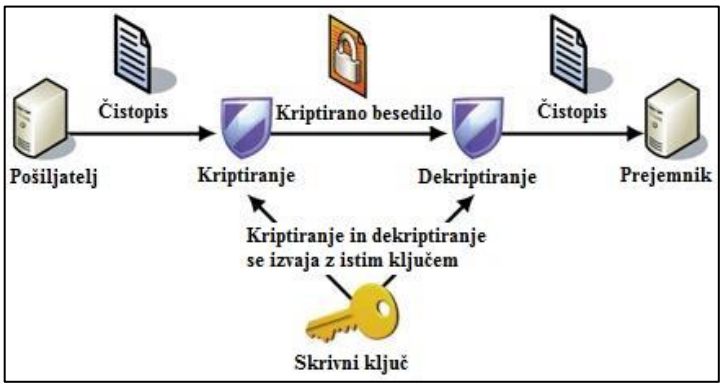
Janko s svojim privatnim ključem (n,d) dekriptira kriptogram po formuli prikazani v naslednji enačbi:

$$m = kriptogram^d \pmod{n}$$

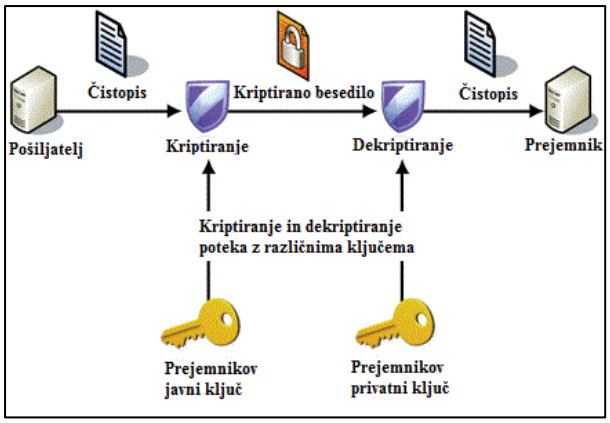
Iz števila m izvleče sporočilo, ki ga le-ta predstavlja [14, 15].

3.5 Kriptografija javnih ključev

Za izmenjavo simetričnega ključa, ki se ga uporabi za šifriranje povezave, se uporablja protokole, ki spadajo v kriptografijo z javnimi ključi (*ang. Public-Key Cryptography*). Kriptografija javnih ključev uporablja asimetrične algoritme za izmenjavo simetričnega ključa. Glavna razlika med asimetričnimi in simetričnimi algoritmi je v tem, da prvi uporabljajo par ključev (javni in privatni), drugi pa samo en ključ (simetrični), ki je skupen obema stranema. Razlika je najlepše opazna, če primerjamo sliko 8 in sliko 9.



Slika 8: Simetrična kriptografija [16]

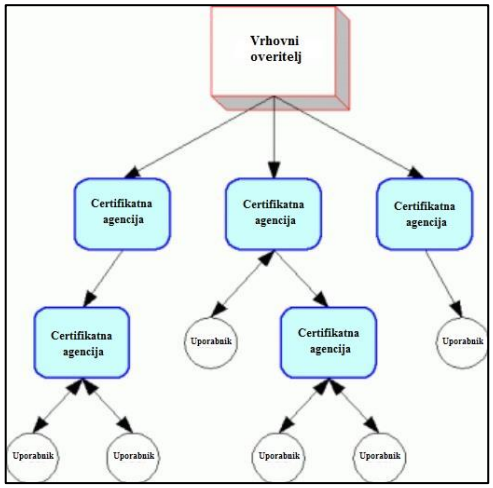


Slika 9: Asimetrična kriptografija [16]

Zelo pogosto uporabljen algoritem pri asimetrični kriptografiji je algoritem RSA, ki ga uporablja tudi moja različica varnega prijavnega sistema.

3.6 Uporaba digitalnih potrdil (certifikatov)

Digitalno potrdilo je dokument, ki potrjuje povezavo med javnim ključem in osebo ali institucijo ali strežnikom [17]. Uporablja se za overjanje oz. za dokazovanje identitete. Eden od prvih pojmov, s katerimi se srečamo, je infrastruktura javnih ključev (*ang. PKI-**P**ublic-**K**ey **I**nfrastructure*). Je skupek programske in stojne opreme ter ljudi in politike, ki skupaj določajo proces kreiranja, upravljanja, distribuiranja, shranjevanja, preklica in uporabe digitalnih potrdil [18]. Drugi pomemben pojem je certifikatna agencija (*ang. CA-**C**ertificate **A**uthority*). CA je del PKI, ki izdaja digitalna potrdila. Celotna infrastruktura javnih ključev pa temelji na modelu zaupanja (*ang. trust model*). Uporabniki si med seboj zaupajo na podlagi certifikatov, ki so bili izdani in podpisani s strani certifikatne agencije. Ta je pooblaščena za izdajo certifikatov fizičnim osebam s strani vrhovnega overitelja (*ang. root CA*). Model je prikazan na sliki 10.



Slika 10: Hierarhični model zaupanja [19]

3.7 Zgradba digitalnega potrdila

Digitalno potrdilo nosi naslednje informacije:

- Različica
- Serijska številka
- Podpisni algoritem
- Izdajatelj
- Velja od
- Velja do
- Nosilec
- Javni ključ
- Algoritem za razpoznavni podpis
- Razpoznavni odtis

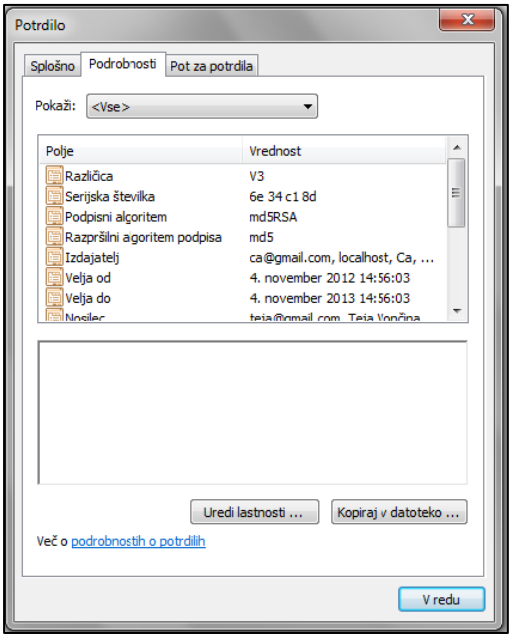
Digitalnemu potrdilu se lahko doda še druga opcijska polja. Ta, ki so našteta zgoraj, pa so obvezna. Standard, ki definira tako zgradbo certifikata, se imenuje X.509 verzija 3. Izdan je bil 3. julija, leta 1988 [13]. Končnice, ki predstavljajo certifikat X.509 v3 so:

- .pem
- .cer, .crt, .der

Razlika med .pem in cer, .crt, ter .der je predvsem v načinu kodiranja. Format .pem je kodiran v načinu Base64. Datoteke s končnico .cer, .crt ali .der pa so kodirane v binarnem DER načinu. Pogosti pa so tudi certifikati kodirani v načinu Base64.

- .p7b, .p7c (Stuktura, ki vsebuje samo podpis podatkov brez le-teh in certifikat ali listo preklicanih certifikatov (*ang. CRL*).)
- .p12 (Lahko vsebuje javni ključ certifikata in privatni ključ, ki je varovan z geslom.)
- .pfx (Predhodnik formata PKCS#12, ki ima končnico .p12.)

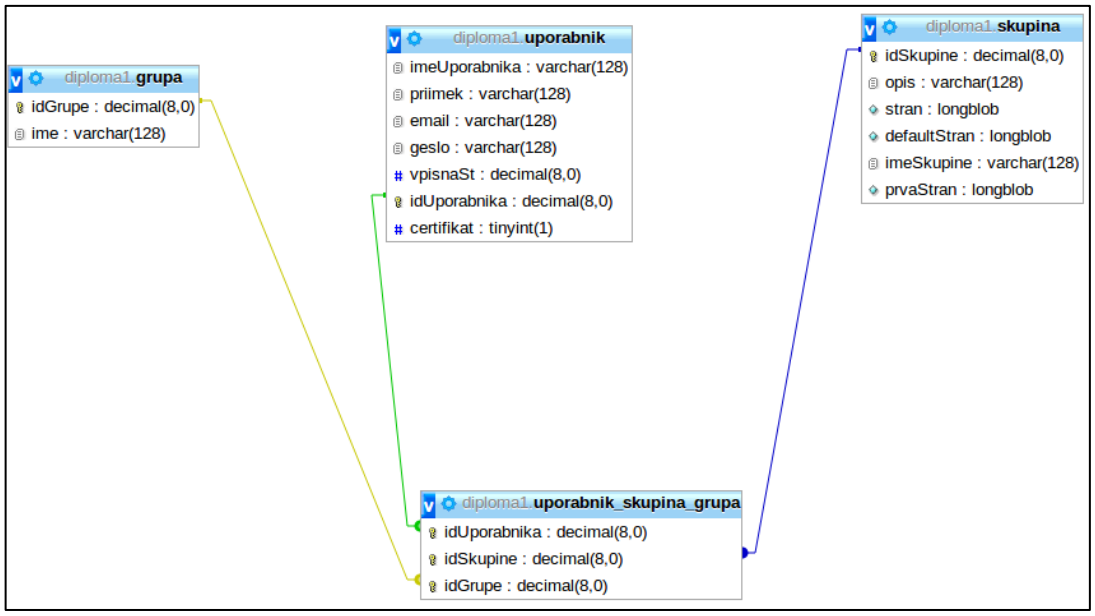
Za nas je najbolj pomembna končnica .p12. Če želimo digitalno potrdilo uvoziti v naš brskalnik, mora imeti končnico .p12. Primer digitalnega potrdila je prikazan na sliki 11.



Slika 11: Digitalno potrdilo, ki je bilo uvoženo v brskalnik Google Chrome

3.8 Model podatkovne baze

Pri načrtovanju podatkovne baze, ki jo aplikacija uporablja, sem prišel do rešitve, ki je sila enostavna. Logični podatkovni model je prikazan na sliki 12. Vsak uporabnik spada v točno določeno skupino. Isti uporabnik je lahko v več skupinah. Skupine so med seboj ločene. Aplikacija vsako skupino posebej vidi kot samostojno celoto. Znotraj pa so uporabniki skupine ločeni glede na grupo, v katero spadajo. Grupa nam pove, kakšne pravice so dodeljene uporabniku.



Slika 12: Logični podatkovni model

Zapisi v bazi podatkov s pomembnejšimi atributi so prikazani na sliki 13. Stolpec certifikat nam pove, ali uporabnik poseduje digitalni certifikat ali ne.

imeUporabnika	priimek	vpisnaSt	email	certifikat	idGrupe	ime	idSkupine	opis
sudo	su	666	super_user@gmail.com	0	0	superUser	0	super user
jure	Leban	63080278	jure.leban@gmail.com	1	1	admin	1	domace naloge
jure	Leban	63080278	jure.leban@gmail.com	1	1	admin	2	eucilnica

Slika 13: Zapisi v bazi podatkov

4 PRIKAZ REŠITEV IN GLAVNIH FUNKCIONALNOSTI

Kot sem že omenil, je ena od zahtev tudi ta, da se uporabnik lahko prijavi tudi na podlagi certifikata brez vpisovanja uporabniškega imena in gesla. Za izdajanje in podpisovanje certifikatov sem s knjižnico OpenSSL ustvaril svojo certifikatno agencijo, s katero sem izdajal in podpisoval certifikate. Ustvaril sem t. i. interno certifikatno agencijo.

4.1 Kreiranje lastne certifikatne agencije (*ang. CA*)

Najprej ustvarimo privatni ključ dolžine 1024 bitov. V ukazno vrstico vnesemo naslednji ukaz:

```
openssl genrsa –out ca.key 1024
```

Na podlagi zgoraj ustvarjenega privatnega ključa zgeneriramo samopodpisljiv certifikat, ki bo veljal dve leti. Certifikat predstavlja certifikatno agencijo, s pomočjo katere lahko podpišemo na novo izdane certifikate:

```
openssl req –new x509 –extensions v3_ca –key ca.key –out
ca.crt –days 730
```

4.2 Ustvarjanje in podpisovanje certifikata

Predpostavimo, da želimo za novega uporabnika ročno ustvariti in podpisati certifikat. To naredimo na sledeč način:

- Zgeneriramo privatni ključ:

```
openssl genrsa -out client.key 1024
```

- Zgeneriramo zahtevo za izdajo certifikata (*ang. CSR – **C**ertificate **S**igning **R**equest*):

```
openssl req -new -key client.key -out client.csr
```

- S prej ustvarjeno certifikatno agencijo podpišemo certifikat:

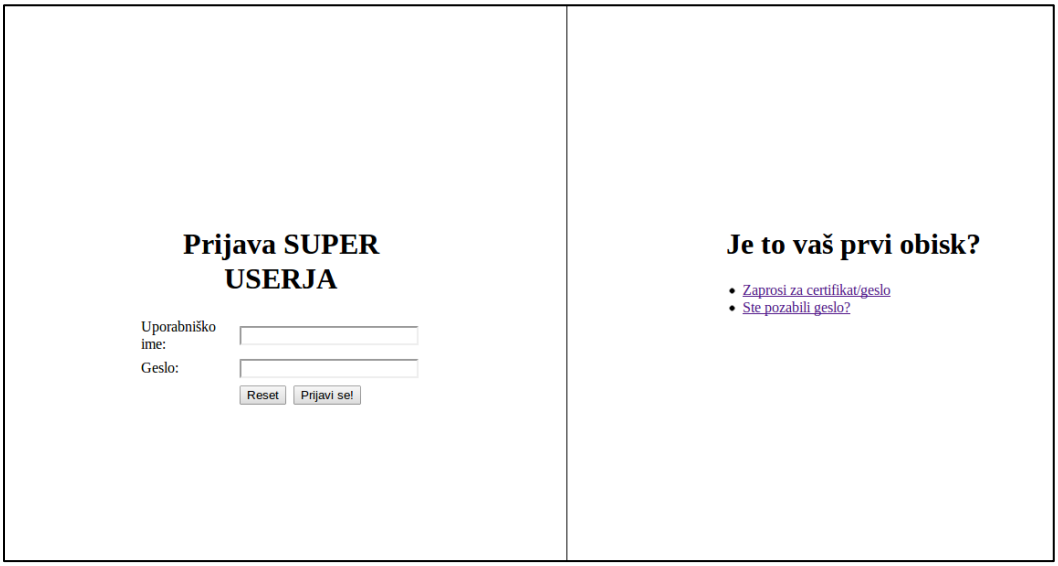
```
openssl x509 -req -days 365 -CA ca.crt -CAkey ca.key -
CAcreateserial -in client.csr -out client.crt
```

- Če želimo certifikat uvoziti v brskalnik, ga moramo iz končnice .crt pretvoriti v končnico .p12, ki predstavlja format PKCS#12:

```
openssl pkcs12 -export -clcerts -in client.crt -inkey
client.key -out client.p12
```

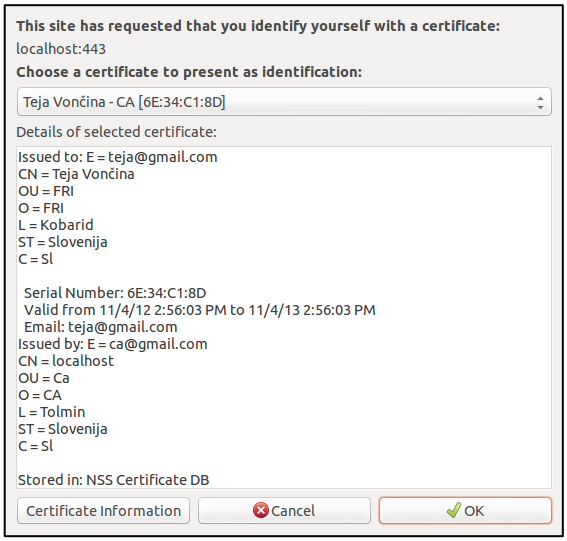
4.3 Overjanje uporabnika

Uporabnik se lahko prijavi na klasičen način, torej s preverjanjem uporabniškega imena in gesla, ali pa z uporabo certifikata. Prijavna stran za administratorja je prikazana na sliki 14. Administrator celotnega sistema ima na začetku privzeto uporabniško ime in geslo, ki ju naknadno lahko spremenimo.



Slika 14: Prijava administratorja celotnega sistema

Še preden se nam prikaže vmesnik za vpis uporabniškega imena in gesla, nas strežnik vpraša, če imamo digitalni certifikat. Če ga imamo, ga izberemo s seznama naloženih certifikatov. Primer je prikazan na sliki 15.



Slika 15: Zahteva po digitalnem certifikatu

Če certifikata nimamo, se nam prikaže prej omenjeni uporabniški vmesnik za preverjanje identitete na podlagi uporabniškega imena in gesla. Programska koda, ki vse to opravi, je prikazana na sliki 16. Ali je uporabnik že prijavljen, preveri funkcija z imenom `checkUser($_GET['skupina'])`. V primeru, da je, ga samo preusmerimo na pravo stran. V primeru, da uporabnik ni prijavljen, pa po vnesenem uporabniškem imenu in geslu s funkcijo `getLoginParams($_GET['skupina'])` preverimo, ali ta uporabnik v bazi podatkov res obstaja. Če obstaja, ga ravno tako preusmerimo na ustrezno stran, sicer uporabnika obvestimo o neujemanju vpisanih podatkov in podatkov v podatkovni bazi.

```php
<?php
    include 'Funkcije/funkcije.php';
    if(!empty($_GET["skupina"])){
            if(checkUser($_GET["skupina"])){
                $skupina = $_GET['skupina'];
                header("Location: https://".$_SERVER['HTTP_HOST']."/SSL_LINUX/prvaStranSkupine/prikazi.php?skupina=$skupina");
            }else{
                if(isset($_POST["submit"])){
                    if(getLoginParams($_GET["skupina"])){
                        $skupina = $_GET['skupina'];
                        header("Location: https://".$_SERVER['HTTP_HOST']."/SSL_LINUX/prvaStranSkupine/prikazi.php?skupina=$skupina");
                    }else{
                       echo  "<script>alert('Napačno uporabniško ime/geslo oz. uporabnik v tej skupini ne obstaja!')</script>";
                    }
                }
            }
          poveziZBazo();
           echo LoginStran($_GET["skupina"]);
    }else{
      die("Stran ne obstaja");
    }
?>
```

Slika 16: Programska koda za preverjanje identitete uporabnika

Če se uporabnik predstavi z digitalnim certifikatom, je ravno tako potrebno primerjati podatke, ki jih nosi digitalni certifikat, s podatki, ki so zapisani v podatkovni bazi. Del programske kode, ki preverja digitalni certifikat, se skriva v prej omenjeni funkciji `checkUser($_GET['skupina'])`. Ta del kode se izvaja kot funkcija in nosi ime `preglejCertifikat()`. Prikazana je na sliki 17. Skriptni jezik PHP uporablja tudi t. i.

super globalne spremenljivke, ki so dosegljive povsod. Ena takih globalnih spremenljivka je tudi spremenljivka `$_SERVER`, ki vsebuje informacije o strežniku in izvajalnem okolju. Spremenljivka `$_SERVER` je v bistvu podatkovno polje spremenljivk. Vse spremenljivke, ki pripadajo protokolu SSL, najdemo znotraj te globalne spremenljivke. V funkciji `preglejCertifikat()` tako preverjamo pet spremenljivk SSL, ki se nanašajo na digitalni certifikat odjemalca. To so:

- SSL_CLIENT_M_SERIAL (Spremenljivka, ki predstavlja serijsko številko certifikata.)
- SSL_CLIENT_V_END (Spremenljivka, ki predstavlja veljavnost certifikata merjena v dnevih.)
- SSL_CLIENT_VERIFY (Spremenljivka, ki predstavlja stanje odjemelčevega certifikata. Možne vrednosti so: NONE, SUCCESS, GENEROUS, FAILED. Seveda pričakujemo, da je vrednost spremenljivke SUCCESS, kar pomeni, da je odjemalčev certifikat veljaven.)
- SSL_CLIENT_I_DN (Spremenljivka, ki predstavlja ime izdajatelja odjemalčevega certifikata.)
- SSL_CLIENT_V_REMAIN (Spremenljivka, ki predstavlja koliko dni je certifikat še veljaven.)

```
function preglejCertifikat(){
    if(!isset($_SERVER['SSL_CLIENT_M_SERIAL'])
            || !isset($_SERVER['SSL_CLIENT_V_END'])
            || $_SERVER['SSL_CLIENT_VERIFY']!="SUCCESS"
            || !isset($_SERVER['SSL_CLIENT_I_DN'])
        ){
            return false;
        }
    if($_SERVER['SSL_CLIENT_V_REMAIN'] <= 0){
        return false;
    }
    return true;
}
```

Slika 17: Preverjanje odjemalčevega certifikata

4.4 Kreiranje in posredovanje certifikata uporabniku

Če se uporabnik želi registrirati, na vstopni strani klikne na povezavo »Zaprosi za certifikat/geslo« in izpolni obrazec prikazan na sliki 18.

Slika 18: Obrazec za pošiljanje certifikata

Na sliki 19 je prikazan izsek programske kode, ki kreira in pošlje nov certifikat na elektronski naslov, ki ga je uporabnik vpisal v zgoraj omenjeni obrazec. Ko se ta dva koraka uspešno izvedeta, se ustvari nov zapis v podatkovni bazi v obliki novega uporabnika. Nov certifikat kreira funkcija `kreirajCertifikat($ime, $priimek, $email, $kraj, $geslo)`, ki je predstavljena na sliki 20. Pošiljanje certifikata opravlja funkcija `posljiCertifikat($ime, $email, $geslo)`, ki je predstavljena na sliki 21. Za pošiljanje e-pošte sem uporabil razred z imenom PHPMailer, ki je prosto dostopen na spletu [20]. Opisani so tudi konkretni primeri uporabe, s katerimi sem si pomagal. Uporabnik na podan elektronski naslov prejme certifikat v formatu .p12. Kot sem že omenil, je to format, ki je primeren za takojšen uvoz digitalnega certifikata v brskalnik. Zraven certifikata uporabnik prejme tudi uporabniško ime in geslo, ki omogočata prijavo brez uporabe digitalnega certifikata. Shranjevanje novega uporabnika opravi funkcija `shraniNovega($ime, $priimek, $email, $geslo, $vpisna, $skupina)`, ki je predstavljena na sliki 22.

```
if($ok){
    if(kreirajCertifikat($ime, $priimek, $email, $kraj, $geslo)){
      //shrani v bazo
        if(posljiCertifikat($ime,$email,$geslo)){
              if(shraniNovega($ime, $priimek, $email, $geslo, $vpisna, $skupina)){
                  echo "<script>alert('Certifikat je poslan na vaš email, ustvarjen je nov uporabnik!')</script>";
              }
        }
        else{
              echo "Mail error!";
        }
    }
}
```

Slika 19: Kreiranje in pošiljanje certifikata ter vnos novega uporabnika v bazo podatkov

```php
function kreirajCertifikat($ime, $priimek, $email, $kraj, $geslo){
    $flag = false;
    $dn = array(
        "countryName" => "Sl",
        "stateOrProvinceName" => "Slovenija",
        "localityName" => $kraj,
        "organizationName" => "FRI",
        "organizationalUnitName" => "FRI",
        "commonName" => $ime." ".$priimek,
        "emailAddress" => $email
    );

    $privkey = openssl_pkey_new();
    $csr = openssl_csr_new($dn, $privkey);

    $privkeyCA = array(file_get_contents("../certifikati/ca.key"),"test");
    $serial = substr(number_format(time() * rand(),0,'',''),0,10);
    $usercert = openssl_csr_sign($csr, file_get_contents('../certifikati/ca.crt'),$privkeyCA,365,NULL,$serial);
    if($usercert != false){
        if(openssl_pkcs12_export($usercert, $out, $privkey, $geslo)){
            $flag = true;
        }
        file_put_contents("../certifikati/$ime.p12", $out);

    }
    return $flag;
}
```

Slika 20: Kreiranje novega certifikata

Kreiranje certifikata, ki je prikazano na sliki 20, je zelo enostavno. Najprej kreiramo razločevalno ime (*ang. Distinguished name-DN*), ki je sestavljeno iz različnih atributov, ki so potrebni za nedvoumno prepoznavanje lastnika digitalnega certifikata. Nato se kreira par ključev (privatni in javni). Naslednji korak je generiranje zahteve za izdajo certifikata (*ang. **C**ertificate **S**igning **R**equest-CSR*). Nato s privatnim ključem certifikatne agencije podpišemo zahtevo ter certifikat izvozimo v formatu .p12. Certifikat je potem poslan na uporabnikov elektronski naslov.

```php
function posljiCertifikat($ime,$email,$geslo){
    include '../conf/mail.php';
    require_once('libphp-phpmailer/class.phpmailer.php');

    $mail = new PHPMailer();
    $mail->IsSMTP(); // telling the class to use SMTP
    try {
        $mail->SMTPDebug  = 0;                     // enables SMTP debug information (for testing)
                                                   // 1 = errors and messages
                                                   // 2 = messages only
        $mail->SMTPAuth   = true;                  // enable SMTP authentication
        $mail->SMTPSecure = "tls";                 // sets the prefix to the servier
        $mail->Host       = $host;      // sets GMAIL as the SMTP server
        $mail->Port       = 587;                   // set the SMTP port for the GMAIL server
        $mail->Username   = $username;  // GMAIL username
        $mail->Password   = $password;            // GMAIL password

        $mail->SetFrom('jure.leban@gmail.com', 'Jure Leban');

        $mail->AddReplyTo("jure.leban@gmail.com","Jure Leban");

        $mail->Subject    = "Certifikat";
        //$mail->AddAddress("$email"); ---> če želiš da dela pravilno
        $mail->AddAddress("jure.leban@gmail.com");//samo za test dam svoj email

        $mail->IsHTML(false);

        $mail->AddAttachment("../certifikati/".$ime.".p12");

        $mail->Body = "Prenesite priponko. To je vaš certifikat, ki ga uvozite v vaš brskalnik!\n
                       Če se želite prijaviti brez certifikata, uporabite spodnje uporabniško ime in geslo.
                       Uporabniško ime: $email\n
                       Geslo: $geslo\n
                       Posredovano geslo uporabite tudi za uvoz certifikata v brskalnik.";
        return $mail->Send();
    }catch (phpmailerException $e) {
        echo $e->errorMessage();
    }catch (Exception $e) {
        echo $e->getMessage();
    }

}
```

Slika 21: Pošiljanje certifikata na elektronski naslov uporabnika

```php
function shraniNovega($ime, $priimek, $email, $geslo, $vpisna, $skupina,$grupa=2, $certifikat=1){
    poveziZBazo();
    $flag=true;
    $query_StVrstic = mysql_query("SELECT idUporabnika FROM uporabnik order by idUporabnika desc limit 1");
    $row_StVrstic = mysql_fetch_array($query_StVrstic);
    $stVrstic = intval($row_StVrstic['idUporabnika']);
    $idUporabnika = $stVrstic+=1;
    $geslo = sha1($geslo);
    $idSkupine = $skupina;
    if($certifikat == 0){//ker se v $skupina shrani IME in ne CIFRA
        $idSkupine = idSkupine($skupina);
    }
    //echo $idUporabnika." ".$ime." ".$priimek." ".$vpisna." ".$skupina;

    $query = "INSERT INTO diploma1.uporabnik (imeUporabnika, priimek, email, geslo, vpisnaSt, idUporabnika, certifikat)
              VALUES ('$ime','$priimek','$email','$geslo','$vpisna','$idUporabnika','$certifikat')";
    $query1 = "INSERT INTO diploma1.uporabnik_skupina_grupa (idUporabnika, idSkupine, idGrupe)
              VALUES('$idUporabnika','$idSkupine','$grupa')";
    poveziZBazo();
    $cheackIfRecordExists = mysql_query("SELECT u.imeUporabnika, u.priimek FROM uporabnik u, diploma1.uporabnik_skupina_grupa usg
                             WHERE u.idUporabnika = usg.idUporabnika
                             AND usg.idSkupine = '$idSkupine'
                             AND u.vpisnaSt='$vpisna'
                             AND u.email='$email'") or die("Napaka pri preverjanju duplikata: ".  mysql_error());

   if(mysql_num_rows($cheackIfRecordExists) == 0){
        if(!mysql_query($query)){
            $flag = false;
            die("Napaka pri vnosu0: ".mysql_error());
        }
        if(!mysql_query($query1)){
            $flag = false;
            die("Napaka pri vnosu1: ".mysql_error());
        }
   }else{
       $flag=false;
   }
    return $flag;
}
```

Slika 22: Vnos novega uporabnika v bazo podatkov

4.5 Omogočanje prijave v primeru izgube gesla

V primeru, da uporabnik pozabi oz. izgubi geslo, ki je potrebno za prijavo, lahko zaprosi za novo. Na vstopni strani klikne na povezavo »Ste pozabili geslo?«. Pojavi se polje za vnos uporabniškega imena, s katerim se je uporabnik registriral. V našem primeru je to elektronski naslov uporabnika. Po vnosu elektronskega naslova se v ozadju izvede spodnji izsek kode, ki preveri, ali uporabnik z vpisanim elektronskim naslovom v bazi obstaja. Izsek kode je prikazan na sliki 23. Če uporabnik obstaja, se zgenerira novo geslo in ga vpišemo v bazo podatkov namesto starega. Na novo zgenerirano geslo s pomočjo že prej omenjenega razreda PHPMailer pošljemo na uporabnikov elektronski naslov. Uporabnik se tako lahko z novim geslom prijavi v sistem in ga spremeni. Vse do sedaj opisane korake izvede funkcija `posljiPozabljenoGeslo($_POST['email'])`. Znotraj te funkcije se izvede tudi preverjanje, ali uporabnik z vnesenim elektronskim naslovom obstaja. Funkcija, ki to opravi, se imenuje `dobiGeslo($email)` in je predstavljena na sliki 24.

```php
<?php
    include 'Funkcije/funkcije.php';
    if(@$_POST['submit']){
        if(!posljiPozabljenoGeslo($_POST['email'])){
            echo "<script>alert('Vpisan e-naslov je neveljaven!')</script>";
        }else{
            echo "<script>alert('Geslo je ponastavljeno, poglej email!')</script>";
            //header("Refresh: 0;url=login_cookie.php");
        }
    }
?>
```

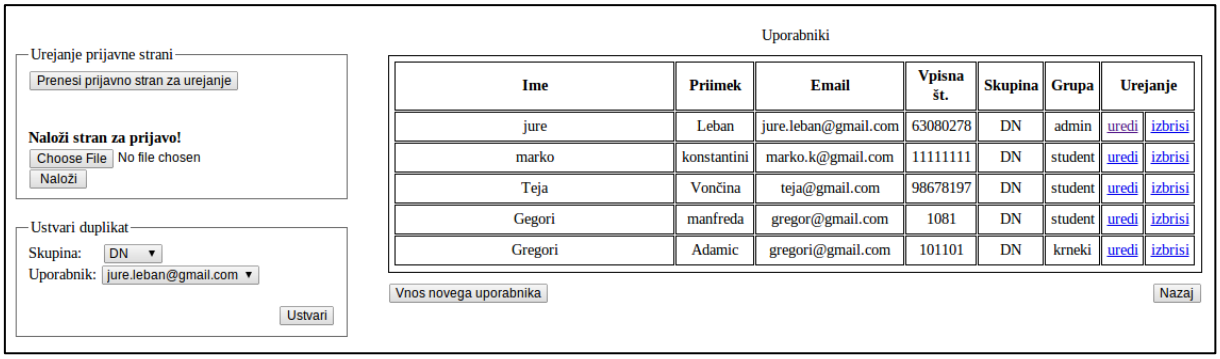
Slika 23: Pošiljanje novega gesla

```php
function dobiGeslo($email){
    poveziZBazo();
    if($query = mysql_query("SELECT idUporabnika FROM uporabnik WHERE email='$email' LIMIT 1")){
        if(mysql_num_rows($query) == 1){//uporabnik ki zaproša za geslo obstaja
           $geslo = generatePassword(); $gesloBaza = sha1($geslo);
           if(!mysql_query("UPDATE uporabnik SET geslo='$gesloBaza' WHERE email = '$email'")){
               die("Napaka:".mysql_error());
           }else{
               return $geslo;
           }
        }else{
            return false;
        }
    }
}
```

Slika 24: Preverjanje, ali uporabnik z vnesenim elektronskim naslovom obstaja

4.6 Spreminjanje prijavne strani skupine

Vsak administrator skupine ima možnost vizualnega spreminjanja prijavne strani. Ko se uporabnik prijavi, preverimo, ali gre za administratorja skupine ali za navadnega uporabnika. Če gre za administratorja, potem ima možnost dostopa do administratorske strani prikazane na sliki 25. Administrator skupine lahko izvaja vse operacije nad uporabniki določene skupine.



Slika 25: Administratorska stran

Lahko dodaja, briše, ureja in prestavlja uporabnike iz ene skupine v drugo. Ena od pravic, ki jih ima administrator skupine, je tudi spreminjanje prijavne (vstopne) strani. To lahko opravi na zgornji sliki, kjer piše »Urejanje prijavne strani«. Prijavno stran hranim v podatkovni bazi. S klikom na gumb »Prenesi prijavno stran za urejanje« se izvede programska koda, ki prijavno stran prenese iz podatkovne baze na naš računalnik. Koda je prikazana na sliki 26. Ravnokar preneseno prijavno stran uredimo po svojem okusu. S klikom na gumb »Choose file« se nam odpre okno, kjer izberemo na novo urejeno prijavno stran ter jo z gumbom »Naloži« naložimo nazaj v podatkovno bazo. Ob naslednjem obisku prijavne strani bo le-ta spremenjena.

```php
<?php
    include '../Funkcije/funkcije.php';
if(!empty($_GET['skupina'])){
    if((checkUser($_GET['skupina']) && admin($_GET['skupina'])) || $_SESSION['SU']){
        if(!skupina_exists($_GET['skupina']))
                die("Skupina ne obstaja");
       $idSkupine = idSkupine($_GET['skupina']);
        poveziZBazo();
       $result = mysql_query("SELECT * FROM skupina WHERE idSkupine=$idSkupine") or die("Napaka pri poizvedbi: ".mysql_error());
       $row = mysql_fetch_array($result);
       if(mysql_num_rows($result)==1 && !is_null($row['stran'])){
           $stran = $row['stran'];
           $opis = $row['opis'];
           header("Pragma: public");
           header("Expires: 0");
           header("Cache-Control: must-revalidate, post-check=0, pre-check=0");
           header("Content-Type: application/force-download");
           header("Content-Type: application/octet-stream");
           header("Content-Type: application/download");
           header("Content-Disposition: attachment; filename=".basename("$opis".".php").";");
           header("Content-Transfer-Encoding: binary");
           echo $stran;
       }else{
           $defaultStran = $row['defaultStran'];
           $opis = $row['opis'];
           header("Pragma: public");
           header("Expires: 0");
           header("Cache-Control: must-revalidate, post-check=0, pre-check=0");
           header("Content-Type: application/force-download");
           header("Content-Type: application/octet-stream");
           header("Content-Type: application/download");
           header("Content-Disposition: attachment; filename=".basename("$opis".".php").";");
           header("Content-Transfer-Encoding: binary");
           echo $defaultStran;
       }
    }else{
       header("Location:../login_cookie.php?skupina=".$_GET['skupina']."");
    }
}else{
    die("napaka v http parametrih!");
}

?>
```

Slika 26: Prenos prijavne strani

Del programske kode, ki se izvede ob pritisku na gumb »Naloži«, je prikazan na sliki 27.

```php
if(isset($_POST['upload']) && $_FILES['userfile']['size'] > 0 && $_FILES['userfile']['error'] == 0){
    $Dovoljene_koncnice = array("php");
    @$dejanska_koncnica = end(explode(".", $_FILES['userfile']['name']));
    if(in_array($dejanska_koncnica, $Dovoljene_koncnice)){
        $tmp_file = $_FILES['userfile']['tmp_name'];
        $fp = fopen($tmp_file, 'r');
        $data = fread($fp, filesize($tmp_file));
        $data = addslashes($data);
        fclose($fp);
        if(empty($_POST['idSkupine'])){//za primer, ko updejtamo stran trenutne skupine(admin ureja svojo skupino)
            $idSkupine = idSkupine($skupina);
            poveziZBazo();
            mysql_query("UPDATE skupina SET stran='$data' WHERE idSkupine = '$idSkupine'")
                    or die("Napaka pri uploadu: ".mysql_error());
```

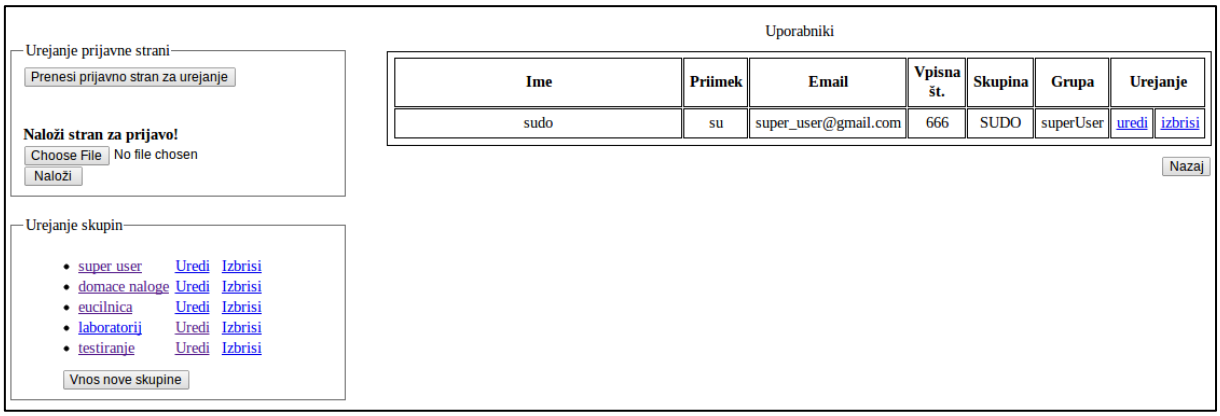
Slika 27: Nalaganje spremenjene prijavne strani

Zgornji izsek kode preverja tudi končnico datoteke, ki jo nalagamo. Dovoljene so samo datoteke, katerih ime se konča s končnico .php.

4.7 Urejanje skupine

S skupinami lahko operira samo administrator celotnega sistema. Lahko počne vse, kar počne administrator skupine. Ima pa tudi dodatne možnosti nadzora nad skupinami: dodajanje novih in urejanje ter brisanje obstoječih. Nadzorna plošča administratorja sistema je prikazana na

sliki 28. V tabeli uporabnikov je prikazan samo en uporabnik. Predstavlja pa administratorja celotnega sistema, ki je trenutno prijavljen. V spodnjem levem kotu slike ima pregled nad vsemi skupinami, ki so v sistemu. S klikom na ime skupine uporabnika (administratorja sistema) preusmeri na nadzorno ploščo le-te. S klikom na povezavo »Administratorska stran« pridemo na stran, kjer lahko upravljamo z določeno skupino in uporabniki le-te. Primer je na sliki 25.



Slika 28: Nadzorna plošča administratorja sistema

4.7.1 Brisanje skupine

Administrator sistema ima, kot sem prej omenil, tudi možnost brisanja skupine. Operacija brisanja skupine izvede brisanje nad dvema tabelama, in sicer nad tabelo z imenom `uporabnik_skupina_grupa` in nad tabelo z imenom `skupina`. Izsek kode je predstavljen na sliki 29.

```
mysql_query("DELETE FROM uporabnik_skupina_grupa
             WHERE idSkupine='".$_GET['idSkupine']."'")
          or die("Napaka pri brisanju uporabnik_skupina_grupa".  mysql_error());
mysql_query("DELETE FROM skupina
             WHERE imeSkupine='".$_GET['imeSkupine']."'") or die("Napaka pri brisanju skupine");
```

Slika 29: Brisanje skupine

Operacija brisanja nad tabelo z imenom `uporabnik_skupina_grupa` izbriše vse zapise, ki se nanašajo na id skupine, ki jo želimo izbrisati. Operacija, ki se izvaja nad tabelo `skupina`, pa se nanaša na ime skupine, nad katero trenutno izvajamo operacijo. Uporabnikov s tem ne izbrišemo, saj so v ločeni tabeli z imenom `uporabnik`, nad katero pa ne izvajamo operacij. Uporabnikov določene skupine iz tabele `uporabnik` niti ni priporočljivo brisati saj se lahko isti uporabnik nahaja v več različnih skupinah. Tabeli `uporabnik_skupina_grupa` in `skupina` sta prikazani na sliki 30.

idUporabnika	idSkupine	idGrupe
26	0	0
1	1	1
5	1	2
17	1	2
24	1	2
27	1	4
1	2	1
17	2	1
25	2	4

idSkupine	opis	stran	defaultStran	imeSkupine	prvaStran
0	super user	[BLOB - 1.9 KiB]	[BLOB - 1.9 KiB]	SUDO	[BLOB - 443 B]
1	domace naloge	[BLOB - 1.9 KiB]	[BLOB - 1.9 KiB]	DN	[BLOB - 443 B]
2	eucilnica	[BLOB - 1.9 KiB]	[BLOB - 1.9 KiB]	EU	[BLOB - 445 B]
3	laboratorij	[BLOB - 1.9 KiB]	[BLOB - 1.9 KiB]	LAB	[BLOB - 444 B]
4	testiranje	[BLOB - 1.9 KiB]	[BLOB - 1.9 KiB]	TEST	[BLOB - 443 B]

Slika 30: Zapisi v tabelah `uporabnik_skupina_grupa` (leva) in `skupina` (desna)

4.8 Preverjanje uporabnika

V sistemu se lahko pojavijo trije različni tipi uporabnikov. To so: navaden uporabnik, administrator skupine in administrator sistema. Lastnosti, ki določajo vsakega od teh treh tipov uporabnikov, so shranjene v super globalni spremenljivki `$_SESSION`. Omenjena spremenljivka je dejansko asociativno polje, ki vsebuje spremenljivke, katerih vsebina so informacije o trenutno vzpostavljeni seji. Odvisno od tega, kaj želimo nekemu uporabniku prikazati, uporabimo dve funkciji, ki preverjata, za kakšen tip uporabnika gre. Funkciji nosita ime `checkUser($_GET['skupina'])` in `admin($_GET['skupina'])`. Na podlagi prejetega parametra o skupini preverita lastnosti o trenutno prijavljenem uporabniku. Obe funkciji uporabljata super globalno spremenljivko `$_SESSION`. V ravnokar omenjenem polju hranim naslednje spremenljivke:

- avtenticiran (Spremenljivka, ki nam pove, ali je uporabnik overjen ali ne.)
- vpisnaSt (Spremenljivka, ki vsebuje vpisno številko uporabnika.)
- upIme (Spremenljivka, ki vsebuje uporabniško ime uporabnika.)
- cert (Spremenljivka, ki nam pove, ali ima uporabnik certifikat ali ne.)
- SU (Spremenljivka, ki nam pove, ali je uporabnik administrator sistema.)

Obe prej omenjeni funkciji uporabimo kmalu na začetku skripte PHP. Izsek kode, ki ponazarja primer uporabe, je prikazan na sliki 31. Če bo torej spodnji pogoj resničen, potem se bo lahko izvedel nek del kode, ki bo uporabniku nekaj prikazal. Pogoj bo resničen, ko bo resničen prvi del pogoja, ki predstavlja navadnega administratorja, ali pa, ko bo resničen drugi del pogoja, katerega spremenljivka predstavlja administratorja celotnega sistema.

```
if((checkUser($_GET['skupina']) && admin($_GET['skupina'])) || $_SESSION['SU']){
```

Slika 31: Uporaba funkcij za preverjanje uporabnika

4.9 Navodila za namestitev spletne aplikacije

1. Namestimo LAMPP paket, ki ga dobimo na spletnem naslovu http://www.apachefriends.org/en/xampp-linux.html. Za nas so pomembni naslednji dodatki, ki jih paket vsebuje: spletni strežnik Apache, sistem za upravljanje s podatkovno bazo MySQL, skriptni jezik PHP in knjižnica OpenSSL. Lampp zaženemo z ukazom `sudo /opt/lampp/lampp start`.

1.1 Spremenimo lastništvo direktorija `htdocs`. Uporabimo ukaz `chown -R jure:jure htdocs`. V direktoriju `/opt/lampp/etc` poiščemo konfiguracijsko datoteko httpd.conf in jo odpremo s poljubnim urejevalnikom besedila. V razdelku `<IfModule unixd_module>` spremenimo vrednost vrstice User na svoje uporabniško ime (v mojem primeru jure).

2. Dostop do strani https://localhost/phpmyadmin/ je onemogočen. To rešimo na sledeč način: V direktoriju `/opt/lampp/etc/extra/` odpremo konfiguracijsko datoteko httpd-xampp.conf. Poiščemo odsek `<Directory »/opt/lampp/phpmyadmin«>`. Pred `</Directory>` dodamo vrstico `Require all granted` in ponovno zaženemo Lampp z ukazom `sudo /opt/lampp/lampp restart`. Dostop do prej omenjene strani je sedaj omogočen, vendar ni zaščiten z geslom. Ravno tako ni zaščitena osnovna stran https://localhost/, s čimer se bomo ukvarjali kasneje.

2.1 Ob poskusu dostopa do spletnega naslova https://localhost/phpmyadmin/ se pojavi napaka »Existing configuration file (./config.inc.php) is not readable.« Odpravimo jo tako, da gremo v direktorij `/opt/lampp/phpmyadmin` in datoteki config.inc.php spremenimo lastništvo na svoje uporabniško ime (v mojem primeru jure). Ukaz, ki to opravi, je: *`sudo chown jure:jure config.inc.php`*.

2.2 Za dostop do spletnega strežnika Apache in orodja phpMyAdmin nastavimo varnostna gesla. Ukaz, ki sledi, je: *`sudo /opt/lampp/lampp security`*. Sledimo navodilom. Ker je ob tem prišlo do spremembe konfiguracijske datoteke config.inc.php, moramo v direktoriju `/opt/lampp/phpmyadmin` datoteki zopet spremeniti lastništvo na svoje uporabniško ime. Ukaz je enak kot pri alineji 2.1.

3. V direktorij `/opt/lampp/htdocs` skopiramo datoteko SSL_LINUX ter vso njeno vsebino. Sledi ukaz `sudo cp -r SSL_LINUX/ /opt/lampp/htdocs/`. Gremo na stran https://localhost/phpmyadmin in kliknemo na zavihek »Zbirke podatkov«.

Ustvarimo novo zbirko z imenom »diploma1«. Pravilo za razvrščanje znakov nastavimo na »utf8_slovenian_ci«. Na levi strani se nam pojavi zavihek »diploma1«, na katerega kliknemo. Gremo na zavihek »Uvozi«. Poiščemo datoteko diploma1_TriTabele.sql in jo uvozimo. Sedaj lahko uspešno dostopamo do spletnega naslova:

https://localhost/SSL_LINUX/login_cookie.php?skupina=DN

Če se hočemo prijaviti kot administrator sistema, gremo na spletni naslov:

https://localhost/SSL_LINUX/login_cookie.php?skupina=SUDO

Prijavimo se z uporabniškim imenom »super_user@gmail.com« in geslom »admin«.

4. Direktoriju `certifikati` spremenimo lastništvo na svoje uporabniško ime (da bomo lahko v njem pisali in brali). Sledi že znan ukaz: `sudo chown -R jure:jure certifikati`.

5. Sledi konfiguracija knjižnice OpenSSL. V direktoriju `/opt/lampp/etc/extra` odpremo konfiguracijsko datoteko httpd-ssl.conf in naredimo sledeče:

   - Poiščemo vrstico `#Certificate Authority (CA):` in dodamo vrstico `SSLCACertificateFile` »`/opt/lampp/etc/ssl.crt/ca.crt`«.
   - Poiščemo vrstico `#Client Authentication (Type):` in spremenimo vrednost vrstice `SSLVerifyClient` na `optional` in vrednost vrstice `SSLVerifyDepth` na `1`.
   - Poiščemo vrstico `#SSLOptions + FakeBasicAuth + ExportCertData + StrictRequire` in izbrišemo komentar-izbrišemo znak `#`.

6. V direktorij `/opt/lampp/etc` skopiramo datoteke ssl.crt, ssl.csr in ssl.key, ki vsebujejo vse potrebno za delo z digitalnimi certifikati (certifikatna agencija, digitalni certifikat strežnika). Za preizkus v brskalnik uvozimo certifikat odjemalca, s katerim se želimo prijaviti v sistem. Primer: Uporabnik teja@gmail.com se želi prijaviti v skupino EU s certifikatom. V direktoriju `certifikati` poiščemo uporabniku pripadajoči certifikat in ga uvozimo v brskalnik. Geslo za odpiranje certifikata je »teja«.

7. Ker smo popravljali konfiguracijske datoteke, je potrebno paket Lampp ponovno zagnati z že znanim ukazom `sudo /opt/lampp/lampp restart`.

8. Gremo na spletni naslov https://localhost/SSL_LINUX/login_cookie.php?skupina=EU, izberemo certifikat, ki smo ga prej uvozili in se prijavimo v sistem brez vpisovanja uporabniškega imena in gesla.

5 SKLEPNE UGOTOVITVE

Ob pregledu in raziskovanju različnih prijavnih sistemov se mi je porodilo nekaj idej, ki bi mojo aplikacijo naredile varnejšo. Skoraj vsi varni prijavni sistemi uporabljajo dvostopenjsko overjanje. Zanimiv primer je družabno omrežje Twitter, ki je pred kratkim doživelo hekerski napad, kjer je bilo odtujenih na tisoče gesel, elektronskih naslovov,… Takoj po vdoru so se pojavila nova delovna mesta za inženirje, katerih naloga je bila implementacija dvostopenjskega overjanja [21]. Tako sem tudi sam prišel na idejo, da bi bila implementacija dvostopenjskega overjanja prvi korak za izboljšanje varnosti prijavnega sistema. Dvostopenjsko overjanje bi temeljilo na principu pošiljanja naključnega števila na uporabnikov mobilni telefon. Pri razvoju sem imel največ težav pri konfiguraciji spletnega strežnika Apache, saj ga je potrebno skonfigurirati tako, da je komunikacija med strežnikom in odjemalcem potekala preko https povezave in da je od uporabnikov zahteval digitalni certifikat. Med razvojem sem med številnimi manjšimi problemi naletel še na problem, kako poslati elektronsko pošto. Problem se da res elegantno rešiti s pomočjo že omenjenega razreda PHPMailer. Skriptni jezik PHP ima za to že vgrajeno funkcijo `mail()`, ki pa je nisem uspel usposobiti. Tudi pošiljanje priponke je z omenjeno funkcijo relativno težje kot z uporabo razreda PHPMailer. Kot zanimivost naj povem še to, da ob prevzemanju občutljivih podatkov, ki jih potrebuje vsak, ki želi preko spletne aplikacije Bank@net dostopati do svojega bančnega računa odprtega pri banki Nova KBM d. d. nisem bil primoran dokazovati svoje identitete. Torej bi teoretično lahko podatke prevzel kdorkoli. Ve se, kaj bi v takem primeru sledilo.

SEZNAM SLIK

SEZNAM TABEL

LITERATURA

[1] C. Kaufman, R. Perlman in M. Speciner, "*Network Security Private Communication in a PUBLIC World*", Second edition, New Jersey 07548, Upper Saddle River: Prentice Hall PTR, A division of Pearson Education, Inc., 2002.

[2] (2010) Wikipedia - Authentication. Dostopno na: http://en.wikipedia.org/wiki/Authentication.

[3] (2012) Wikipedia - Transport Layer Security (TLS). Dostopno na: http://en.wikipedia.org/wiki/Secure_Sockets_Layer.

[4] (2013) Wikipedia - Multi-factor authentication. Dostopno na: http://en.wikipedia.org/wiki/Two-factor_authentication.

[5] (2012) Hong Kong Monetary Authority. Dostopno na: http://www.hkma.gov.hk/eng/key-functions/banking-stability/internet-banking/two-factor-authentication.shtml.

[6] (2011) Dexternights. Dostopno na: http://www.dexternights.com/2011/03/20/rsa-security-not-forthcoming-on-security-breach/.

[7] (2012) Wikipedia - TLS 1.1. Dostopno na: http://en.wikipedia.org/wiki/Transport_Layer_Security#TLS_1.1.

[8] (2006) The Internet Engineering Task Force (IETF). Dostopno na: http://www.ietf.org/rfc/rfc4492.txt.

[9] (2012) Wildbunny. Dostopno na: http://www.wildbunny.co.uk/blog/2012/10/09/how-to-make-a-multi-player-game-part-1/.

[10] (2012) Wikipedia - Client-authenticated TLS handshake. Dostopno na: http://en.wikipedia.org/wiki/Transport_Layer_Security#Client-authenticated_TLS_handshake.

[11] (2005) Beefchunk. Dostopno na: http://beefchunk.com/documentation/bin/apache/apache2_with_ssl_tls/part1.htm.

[12] (2006) IETF Tools - rfc4346. Dostopno na: https://tools.ietf.org/html/rfc4346.

[13] (2013) Wikipedia - X.509. Dostopno na: http://en.wikipedia.org/wiki/X.509.

[14] (2013) Wikipedia - RSA (algorithm). Dostopno na: http://en.wikipedia.org/wiki/RSA_(algorithm).

[15] (2011) MaFiRa-Wiki. Dostopno na: http://wiki.fmf.uni-lj.si/wiki/Naloga:_Algoritem_za_zakrivanje_z_javnim_klju%C4%8Dem.

[16] (2005) MSDN. Dostopno na: http://msdn.microsoft.com/en-us/library/ff650720.aspx.

[17] (2006) Overitelj na Ministrstvu za pravosodje in javno upravo RS. Dostopno na: http://www.si-ca.si/kripto/kr-cert.htm.

[18] (2013) Wikipedia - Public-key infrastructure. Dostopno na: http://en.wikipedia.org/wiki/Public-key_infrastructure.

[19] (2005) Galexia. Dostopno na: http://www.galexia.com/public/research/assets/pki_interoperability_models_2005/pki_interoperability_models_2005-4_1_.html.

[20] (2009) PHPMailer. Dostopno na: http://phpmailer.worxware.com/.

[21] (2013) Slo - Tech. Dostopno na: https://slo-tech.com/novice/t554814.