Mrežni protokol za siguran prijenos biometrijskih podataka

Josip Krapac

1 Zadatak

Siguran prijenos biometrijskih značajki s klijenta na autorizacijski poslužitelj. Pretpostavlja se da je na klijentskom računalu instalirana klijentska aplikacija. Zadatak klijentske aplikacije je akvizicija slika lica i dlana, izlučivanje biometrijskih značajki iz dobivenih slika te unos korisničkog pina.

2 Rješenje

2.1 Klijentska strana

- **Web preglednik poziva klijentsku aplikaciju**:
  Web preglednici ne mogu pokretati klijentske programe. Da bi oni to mogli moraju se instalirati proširenja web preglednika (eng. web browser extensions). Proširenje web preglednika instalira se automatski, prilikom pristupa stranici (na web poslužitelju) koja sadrži proširenje. Proširenja tada mogu pokretati programe na klijentskom računalu, ali samo ukoliko im korisnik to dozvoli. Da bi proširenje uopće moglo zahtijevati te dozvole, potrebno je da autor proširenja digitalno potpiše (eng. digitaly sign) proširenje, kako bi korisnik prilikom njegova instaliranja mogao utvrditi identitet autora proširenja i na osnovu toga dati mu tražene dozvole.

- **Biometrijske značajke i korisnički pin se na siguran način prenose na autorizacijski poslužitelj**:
  Za siguran prijenos biometrijskih značajki na poslužitelj korišten je HTTPS protokol, koji je zapravo HTTP (eng. hypertext transfer protocol) preko SSL-a (eng. secure socket layer).[11]

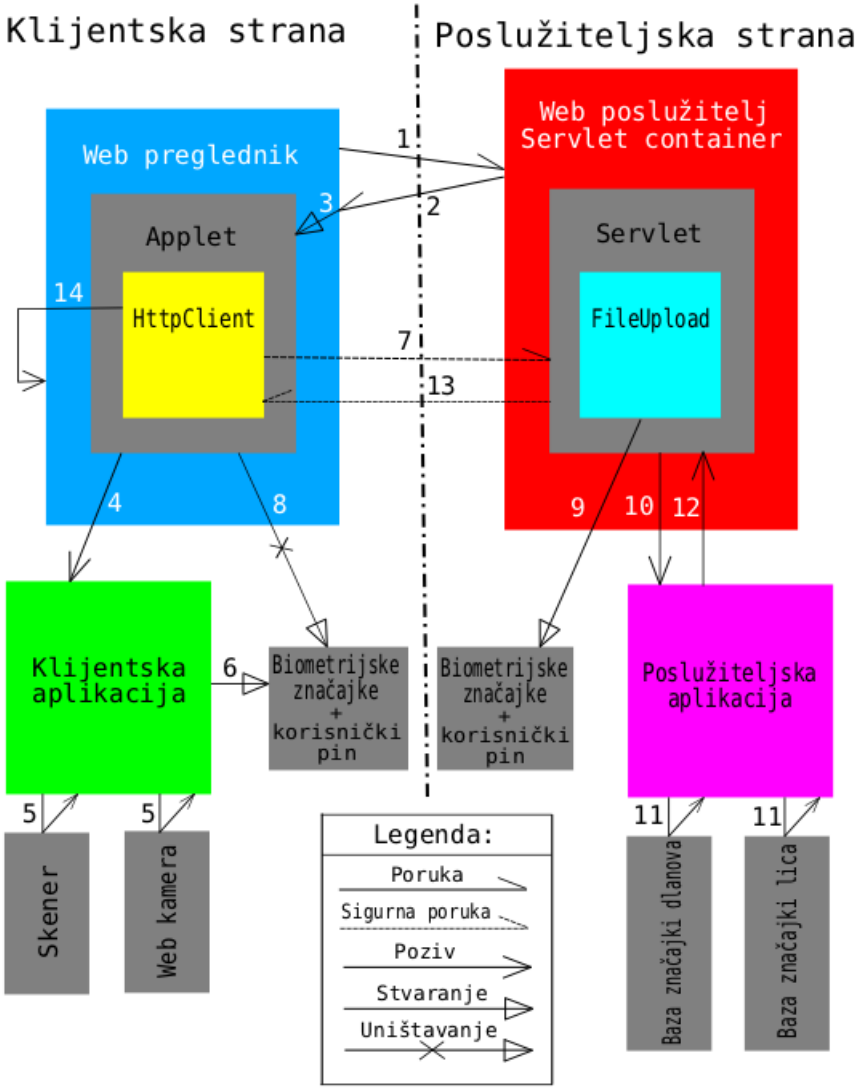
2.2 Poslužiteljska strana

- **Web poslužitelj poziva poslužiteljsku aplikaciju**:
  Web poslužitelji ne mogu pozivati lokalne aplikacije. Zato je potrebno koristiti proširenja web poslužitelja (eng. web server extensions).

- **Poslužiteljska aplikacija na temelju prenesenih biometrijskih značajki i korisničkog pina autorizira korisnika**:
  Web poslužitelj prima odluku od poslužiteljske aplikacije (uspjela/neuspjela autorizacija) te u slučaju uspješne autorizacije pokreće sigurnu sjednicu (eng. secure session), a u slučaju neuspješne autorizacije ispisuje obavijest o neuspjehu.

Sigurna sjednica je mehanizam koji web poslužitelju omogućava praćenje (autoriziranog) korisnika između HTTP/HTTPs zahtjeva, bilo na način pohranjivanja podataka koji služe za praćenje sjednice (identifikatora sjednice; eng. session id) na klijentskoj strani u vidu tzv. "kolačića" (eng. cookie), bilo u vidu zapisivanja tih podataka u URL (eng. URL rewriting), u slučaju da web preglednik ne podržava "kolačiće".

Započinjanjem sigurne sjednice postupak autorizacije smatra se završenim.



Slika 1

Dijagram toka interakcije između klijenta i poslužitelja tijekom autorizacije: 1: Zahtjev za autorizacijom; 2: Odgovor (sadrži applet); 3: Pokretanje appleta; 4: Pokretanje klijentske aplikacije (blokirajući poziv); 5: Akvizicija slika lica i dlana; 6: Pohranjivanje izlučenih značajki u datoteku; 7: Uspostava sigurnog komunikacijskog kanala i slanje značaki i korisničkog pina istim te primanje odgovora od servleta (blokirajući poziv); 8: Brisanje značajki i korisničkog pina sa klijentskog računala; 9: Izvlačenje datoteka iz poruke klijenta i njihovo spremanje na poslužiteljsko računalo; 10: Pokretanje poslužiteljske aplikacije; 11: Usporedba prenesenih značajki i korisničkog pina sa značajkama pohranjenim u bazi; 12: Javljanje donesene odluke servletu; 13: Pokretanje sigurne sjednice / Dojava o grešci; 14: Predaja dobivene sjednice web pregledniku

3 Tehnologija

3.1 Klijentska strana

Raspoložive tehologije za izvedbu proširenja web preglednika su:

1. Java Appleti[9]

2. Microsoft ActiveX Kontrole[5]

Korišten je Java Applet, jer se njime osigurava prenosivost (eng. portability). Iako je klijentska aplikacija napisana isključivo za operacijski sustav Microsoft Windows, na ovaj način su osigurane minimalne promjene u slučaju da se u budućnosti stvori potreba za klijentskom aplikacijom koja bi trebala raditi i na drugim operacijskim sustavima.

Za izgradnju Appleta korištene su Java klase koje su dio razvojnog okruženja J2SDK v.1.4.2[8].

Java Applet također koristi slijedeće Java softverske komponente:

1. Jakarta Commons HttpClient[4]
   Skup klasa koje implementiraju klijenta za HTTP/HTTPs protokol

2. Jakarta Commons FileUpload[3]
   Skup klasa koje implementiraju prijenos datoteka na web poslužitelj putem HTTP/HTTPs protokola.

Digitalni certifikat je generiran alatom *keytool* koji je dio razvojnog okruženja J2SDK v.1.4.2. U digitalnom certifikatu sadržani su podaci o vlasniku digitalnog certifikata i javni ključ vlasnika digitalnog certifikata.

Applet je digitalno potpisan alatom *jarsigner* koji standardno dolazi sa razvojim okruženjem J2SDK v.1.4.2.

3.2 Poslužiteljska strana

Raspoložive tehnologije za izvedbu proširenja web poslužitelja su:

1. CGI (Common Gateway Interface)[2]
2. Java Servleti[10] / JSP (Java Server Pages)
3. Microsoft ASP.NET (Active Server Pages)[6]
4. PHP (Personal HomePage)[7]

I na strani poslužitelja korištena je Java tehnologija, također radi prenosivosti.

Sukladno odluci da se koriste Java Servleti kao Servlet Container korišten je Apache Jakarta Tomcat[1]. Budući je on ujedno i web poslužitelj koji ima podršku za HTTPS, nije se uvidjela potreba da se u ovom stadiju razvoja koristi poseban web poslužitelj (npr. Apache web server, Microsoft IIS).

Literatura

[1] Apache Jakarta Tomcat, http://jakarta.apache.org/tomcat.

[2] Common Gateway Interface, http://www.w3.org/CGI.

[3] Jakarta Commons FileUpload, http://jakarta.apache.org/commons/fileupload.

[4] Jakarta Commons HttpClient, http://jakarta.apache.org/commons/httpclient.

[5] Microsoft ActiveX Controls, http://www.microsoft.com/com/tech/ActiveX.asp.

[6] Microsoft ASP.NET, http://www.asp.net.

[7] PHP, http://www.php.net.

[8] Sun Java 2 Platform, Standard Edition, http://java.sun.com/j2se/1.4.2.

[9] Sun Java Applets, http://java.sun.com/applets.

[10] Sun Java Servlets, http://java.sun.com/products/servlet.

[11] Eric Rescorla. *SSL and TLS: Designing and Building Secure Systems*, chapter 9. Addison-Wesley, 2003.