Projektni zadatak za predmet

Informaciona bezbednost

Secure Instant Messenger

verzija 1

24.04.2015.

Opis zadatka

Potrebno je implementirati sigurnu klijentsku aplikaciju za razmenu poruka. Sigurnost podrazumeva poverljivost, integritet i neporecivost poruka.
Serverska strana aplikacije je već napravljena i data uz zadatak u okviru Apache Tomee servera. Potrebno je implementirati klijentsku stranu.

Detalji zadatka

Klijentska aplikacija komunicira sa serverom pomoću XML poruka. Na raspolaganju su dve metode za komunikaciju sa serverom:

- Metoda `put` koja kao parametar prima objekat klase `Document`.
  - Ova metoda prosleđuje navedeni dokument serverskoj strani u vidu XML dokumenta
- Metoda `get` koja kao rezultat vraća navedenu poruku navedenog korisnika. Kao parametre prima sledeće podatke:
  - `String userId` – označava korisnika čiju poruku želimo da preuzmemo.
  - `String messageId` – označava redni broj poruke koju želimo da preuzmemo.

XML poruka koja se šalje serverskoj strani ima sledeće elemente:

- `message` – Korenski element koji sadrži sledeće podelemente
  - `recipientId` - ID korisnika kome se šalje poruka (`string`)
  - `senderId` - ID korisnika koji šalje poruku (`string`)
  - `messageText` - Sam tekst poruke

Kako bi se ispoštovali zahtevi za neporecivošću, integritetom i poverljivošću, neophodno je implementirati digitalno potpisivanje i šifrovanje XML dokumenta pre slanja.

Generisanje privatnog i javnog ključa i sertifikata

Za potrebe projekta potrebno je napraviti privatni i javni ključ, kao i sertifikat koji će sadržati javni ključ za dva korisnika A i B. **Slede opcije I i II koje su međusobno isključive i zavise od ocene za koju se radi:**

- (**Opcija I**) Studenti koji rade za ocenu **6** ili **7** prave samopotpisane sertifikate (koristi se Portecle ili keytool alat), a postupak je sledeći:

1. Korisnik A generiše svoj privatni i javni ključ, kao i svoj sertifikat i smešta ga zajedno s privatnim ključem u svoj keystore fajl (`usera.jks`).
2. Korisnik B generiše svoj privatni i javni ključ, kao i svoj sertifikat i smešta ga zajedno s privatnim ključem u svoj keystore fajl (`userb.jks`).
3. Korisnik A koristeći Portecle alat ili keytool alat eksportuje svoj sertifikat u cer fajl.
4. Korisnik B koristeći Portecle alat ili keytool alat importuje sertifikat korisnika A u svoj keystore fajl (key store fajl od korisnika B).
5. Korisnik B koristeći Portecle alat ili keytool alat eksportuje svoj sertifikat u cer fajl.
6. Korisnik A koristeći Portecle alat ili keytool alat importuje sertifikat korisnika B u svoj keystore fajl (key store fajl od korisnika A).

- (**Opcija II**) Studenti koji rade za ocenu **8, 9** i **10** treba da naprave sertifikat za sertifikaciono telo (CA) koje će potpisati sertifikate korisnika A i B, a postupak je sledeći:

1. Napisati Java program koji će generisati privatni i javni ključ za CA-a, kao i samopotpisani sertifikat za CA i ssmestiti ih u keystore fajl (`ca.jks`).
2. Napisati Java program koji generiše sertifikate za korisnike, pri čemu sertifikate potpisuje CA. Privatni ključ i sertifikat se čuvaju u keystore fajlu.
    1. Koristeći ovaj program generisati privatni i javni ključ, kao i sertifikat za korisnika A i smestiti privatni ključ i sertifikat u keystore fajl za korisnika A (`usera.jks`).
    2. Koristeći ovaj program generisati privatni i javni ključ, kao i sertifikat za korisnika B i smestiti privatni ključ i sertifikat u keystore fajl za korisnika B

(`userb.jks`).

3. Korisnik A koristeći Portecle alat ili keytool alat eksportuje svoj sertifikat u cer fajl.
4. Korisnik B koristeći Portecle alat ili keytool alat importuje sertifikat korisnika A u svoj keystore fajl (keystore fajl od korisnika B).
5. Korisnik B koristeći Portecle alat ili keytool alat eksportuje svoj sertifikat u cer fajl.
6. Korisnik A koristeći Portecle alat ili keytool alat importuje sertifikat korisnika B u svoj keystore fajl (keystore fajl od korisnika A).

Komunikacija učesnika

Komunikacija izmedju dva učesnika potrebno je realizovati kroz dva programa:

- Prvi program šalje poruku:

1. Korisnik A kreira poruku za korisnika B.
2. Korisnik A potpisuje poruku svojim privatnim ključem.
3. Korisnik A šifruje poruku simetričnim ključem i taj simetrični ključ sifruje javnim ključem korisnika B i smešta u poruku.
4. Korisnik A šalje poruku serverskoj strani.

- Drugi program prima poruku:
  1. Asinhrono, korisnik B preuzima poruku sa serverske strane.
  2. Korisnik B dešifruje poruku svojim privatnim ključem.
  3. Korisnik B verifikuje digitalni potpis poruke pomoću sertifikata korisnika A.
  4. Korisnik B čita poruku korisnika A.

Pri šifrovanju poruke, šifruje se samo `messageText` element. Pri potpisivanju, potpisuje se čitava poruka.
Šiforvanjem se postiže da iako svi mogu da vide sve šifrovane poruke, samo ciljani primalac može da je dešifruje. Potpisivanjem se postiže da integritet poruke, primalac se osigurava da poruka nije menjana i neporecivost poruke, da niko drugi do navedeni pošiljalac nije poslao poruku. U suprotnom, digitalni potpis neže odgovarati poruci i sertifikatu.

Algoritmi

Pri šifrovanju se koristi KEK (Key encryption key) metod. Kao algoritam za simetrično šiforvanje koristi se TripleDES. Asimetricni algoritam je RSA.
Pri potpisivanju se korisni enveloped stil. Potpisivanje se vrši pomoću RSA algoritma.

Zahtevi i testni slučajevi

1. Rešenje treba da omogući izdavanje potpisanih sertifikata i smeštanje istog u keystore.
2. Rešenje treba da omogući dodavanje “prijateljskih” sertifikata u keystore.
3. Pri slanju poruke, navodi se identifikator sertifikata pošiljaoca (radi potpisivanja) i sertifikata primaoca (radi šiforvanja) preko aliasa u odgovarajućem keystore fajlu. Identifikator primaoca i pošiljaoca treba da odgovara aliasu u odgovarajućem keystore-u.
4. Aplikacija ne mora da ima GUI, dovoljno je napraviti konzolnu aplikaciju s pomenutim funkcionalnostima.
5. Potrebno je demonstrirati sledeće test slučajeve:
   - slanje regularno šifrovane i potpisane poruke
   - preuzimanje i prikazivanje regularno šiforvane i potpisane poruke
   - detektovanje neregularne poruke sa izmenjenim sadržajem (invalidan potpis)
   - **za ocenu 9 i 10:** detektovanje neregularne poruke sa pogrešnim sertifikatom (senderID ne odgovara odgovarajućoj vrednosti u subject polju u sertifikatu)

Pokretanje servera

Kao što je već rečeno serverska apliakcija je već implementirana i smeštena je u okviru Apache Tomee servera (u webapps folderu nalazi se fajl `SecurIM.war`). Apache Tomee server je u stvari Tomcat server sa dodatnim bibliotekama. Pre pokretanja potrebno je podesiti JAVA_HOME i PATH variajble (opisano u narednom odeljku) i raspakovati zip fajl sa serverom (`apache-tomee-plus-1.6.0.1-sa-serverskom-app.zip`). Zip fajl sa serverom je najbolje raspakovati u folder `D:\servers` ili u `C:\servers`. Server se startuje tako što se u `bin` folderu pokrene fajl `startup.bat` (može se pokrenuti direktno dvoklikom ili se otkuca `startup.bat` u command prompt-u, ali se prvo `cd`

naredbom pozicionira u `bin` folder). Server se zaustavlja tako što se nad otvorenim command prompt-om servera pritisne `Ctrl + C` ili sa pokretanjem `shutdown.bat` u `bin` folderu. Sve primljene poruke čuvaju se u `bin/messages` folderu.

Instalacija i konfigurisanje Jave

Ukoliko nije instaliran JDK, otići na sajt jave (http://www.oracle.com/technetwork/java/javase/downloads/index.html) i tamo preuzeti najnoviju verziju JDK (**osnovna verzija mora biti 1.7**).
Za Windows OS duplim klikom pokreniti instalaciju jave i pratiti instalacioni vizard. Instalacioni folder Jave za Windows operativne sisteme zavisi od njegove verzije (x86 ili x64) i najčešće je nalik:
C:\Program Files (x86)\Java\jdk1.7.0_XY\ ili
C:\Program Files\Java\jdk1.7.0_XY\

Postavljanje *JAVA_HOME* i *PATH* promenljive u Windows 7 okruženju.

1. Desni klik miša na **My Computer** ikonu sa desktopa i selekcija stavke **Properties**.
2. Klik na **Advanced System Setings**
3. Klik na **Advanced** karticu.
4. Klik na **Environment Variables** dugme.
5. Pod stvakom **System Variables**, kliknuti dugme **New**.
6. Uneti naziv nove varijable JAVA_HOME.
7. Uneti vrednost varijable kao putanje do instalacionog foldera Java Development Kit.
   - (kao na primer C:\Program Files (x86)\Java\jdk1.7.0_XY\).
1. Dugme potvrde **OK**.
2. Izmeniti vrednost sistemske varijable **Path** tako da se na kraju doda vrednost %JAVA_HOME %\bin;
   - ; je separator vrednosti unutar varijable
1. Dugme potvrde **Apply Changes**.
2. Ponovo porenuti Windows. (poželjno)

Provera da li je JAVA_HOME postavljen. Otvoriti command promt i ukucajte *SET JAVA_HOME*.
Ukoliko vidite odgovarajuću putanju varijabla je postavljena. Ukoliko vidite naziv "Environment Variable JAVA_HOME not defined", ponovite proceduru za unos odgovarajuće varijable.
Provera da li je PATH postavljen da vidi %JAVA_HOME%\bin folder. Otvoriti command promt i ukucajte *SET PATH* i proverite da li je PATH dobar.