Втора домашна задача: Автентицирана размена на клучеви

Документација

Имплементацијата на протоколот Керберос е изведена во програмскиот јазик Java и содржи 7 класи:

KerberosImplementation : Во оваа класа е дефиниран main методот во кој што се инстанцираат објектите Alice и Bob како и KDC серверот.

KDC: Е класа во која се чуваат сите корисници заедно со нивните клучеви. Откако некој корисник ќе побара да се автентицира со серверот, корисникот му испраќа барање кое ќе ги содржи ID од корисникот, ID од корисникот со кој сакаме да воспоставиме комуникација и нонс преку методот revieveRequest. Откако корисникот ќе направи барање до серверот, серверот ги бара корисниците во сопствената база Users (HashMap) и доколку корнисниците постојат во неговата база тој ќе ги добие нивните клучеви кој ќе му послужат за енкрипција на одговорите што подоцна ќе ги испрати назад до корисникот. Серверот генерира TimeStamp во кој се чуваат времето на генерирање на TimeStamp како и рокот на истекување кој трае 8 часа. Серверот генерира и произволен уникатен сесиски клуч. Серверот како одговор на корисникот му праќа две пораки ya и yb. Во ya ги сместува: сесискиот клуч, нонс, TimeStamp и ID од вториот корисник. Потоа оваа порака ја претвора во низа од бајти и ја енкриптира со клучот од првиот корисник и со ова си дава до сигурност дека доколку корисникот кој бара автентикација е вистинскиот корисник, тој ќе го знае сопствениот клуч и ќе може да ја декриптира оваа порака и со тоа ќе го добие сесискиот клуч кој ќе се користи за понатамошна комуникација со вториот корисник, но откако ќе потврди дека TimeStamp не е истечен, односно е се уште валиден и ID е истото ID со кое корисникот испрати барање до серверот. Во yb се сместуваат: сесискиот клуч, ID на првиот корисник и TimeStamp. Потоа оваа порака се енкриптира со клучот од вториот корисник, што значи дека првиот корисник нема да може да ја прочита, но ќе мора да ја испрати до вториот корисник. Овие две пораки се стават во листа и се испраќаат назад до првичниот корисник. Следните чекори се одвиваат во следната класа User.

User: Со класата User се дефинираат својствата за еден корисник како што се ID,password, симетричен клуч Ka, nonce и референца до KDC серверот со кој што треба да се изврши автентикацијата. Во оваа класа се дефинирани методите: generateNonce() за генерирање на произволен нонс, generateSecretKey() за генерирање на клуч за корисникот, encrypt(SecretKey key, byte [] array) за енкрипција на низа од бајти со даден клуч, decrypt(SecretKey key, byte [] encryptedArray) за декрипција на енкриптирана низа од бајти

со клуч key, методи за автентикација на нонс, ID на корисникот со кој сакаме да воспоставиме врска како и метод за потврдување на валидноста на TimeStamp. Со методот sendRequest(User bob) во кој објектот bob испратен како параметар е референца до вториот корисник, се повикува методот receiveRequest (кој е објаснет погоре во класата KDC), и овој метод враќа листа во која се сместени енкриптираните пораки ya и yb. Првата порака ya се декриптира и се претвора во објект од класата ya кој потоа се верифицира со горе наведените методи. Со ова првиот корисник го добива сесискиот клуч кој го користи за да ја енкриптира пораката yab во која има сместено ID од првиот корисник и TimeStamp. Потоа пораките yab и yb се стават во листа и се испраќаат до вториот корисник (со помош на референцата испратена како параметар) преку повик на методот reveiveRequest од класата User (бидејќи има истоимен метод од KDC класата). Во таа класа вториот корисник ја добива листата со двете пораки, ја зема првата yb ја декриптира со помош на сопствениот клуч, ја верификува и го добива сесискиот клуч. Со помош на сесискиот клуч ја декриптира втората порака yab, и нејзе ја верификува и го добива ID од првиот корисник. Па бидејќи сега двете страни го имаат сесискиот клуч тие можат да си испраќаат енкриптирани пораки меѓусебно што е претставено со помош на методите sendMessage и receiveMessage кој пораките се енкриптираат и декриптираат со сесиски клуч.

Класите YA,YB,YAB служат за полесно претставување на пораките кои се спомнати во горенаведениот текст.

YA чува (сесиски клуч, нонс, TimeStamp, IDb).

YB чува (сесиски клуч, Ida, TimeStamp)

YAB чува (Ida, TimeStamp).

TimeStamp: Во оваа класа се чува времето на креирање на објектот од оваа класа, како и времето на истекување на објектот (8 часа после креирање на објектот). Објектот од оваа класа служи за верификација на валидноста на пораката, бидејќи секој сесиски клуч трае 8 часа доколку се поминати повеќе од 8 часа треба да се генерира нов сесиски клуч, односно корисниците да се автентицираат повторно.