OpenPGP implementacija

Table of Contents

1. Uvod

Cilj projekta je implementacija programa koji je kompatabilan sa OpenPGP standardom definisanom po dokumentu rfc4880. Program treba da pruži mogućnost generisanje para RSA ključeva, enkripcije podataka koristeći triple DES ili AES128, kompresiju podataka, potpisivanje podataka RSA ključem i SHA-1, konverzija podatka u radix64 formatu i mogućnost importovanja i eksportovanja RSA ključa. Za Implementaciju se koristio bouncy castle.

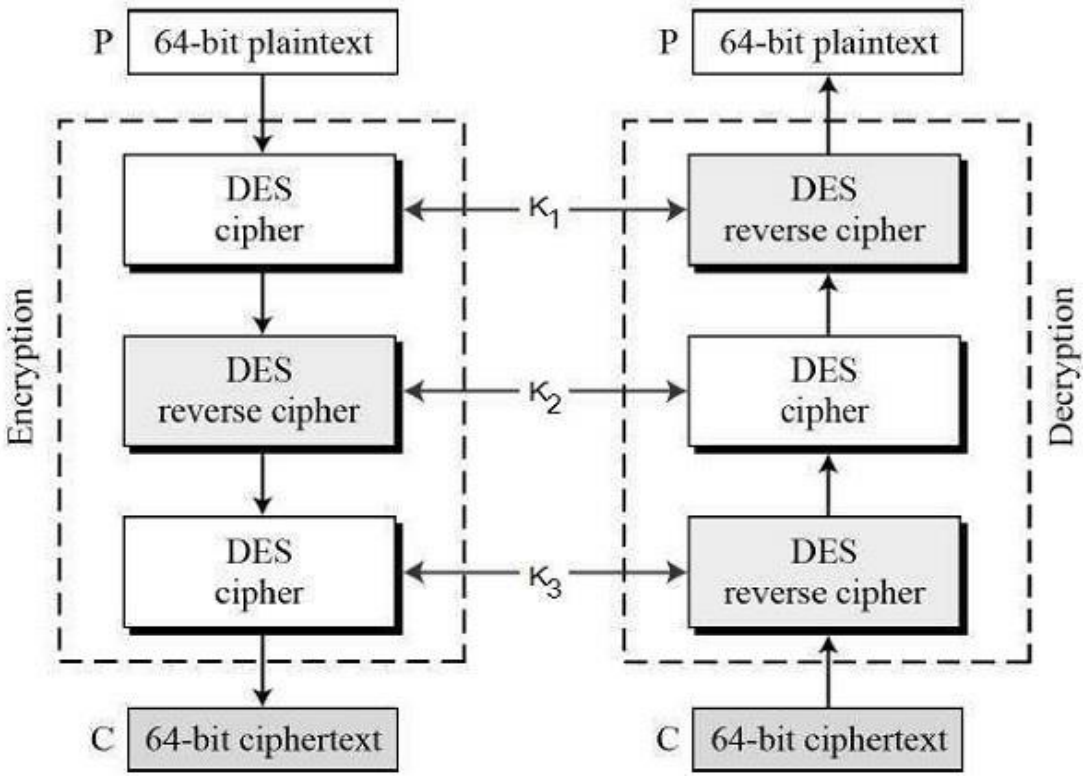
2. Korisćeni algoritmi

2.1. Simetrični algoritmi

Simetrične algoritme koristimo za enkripciju samih podataka.

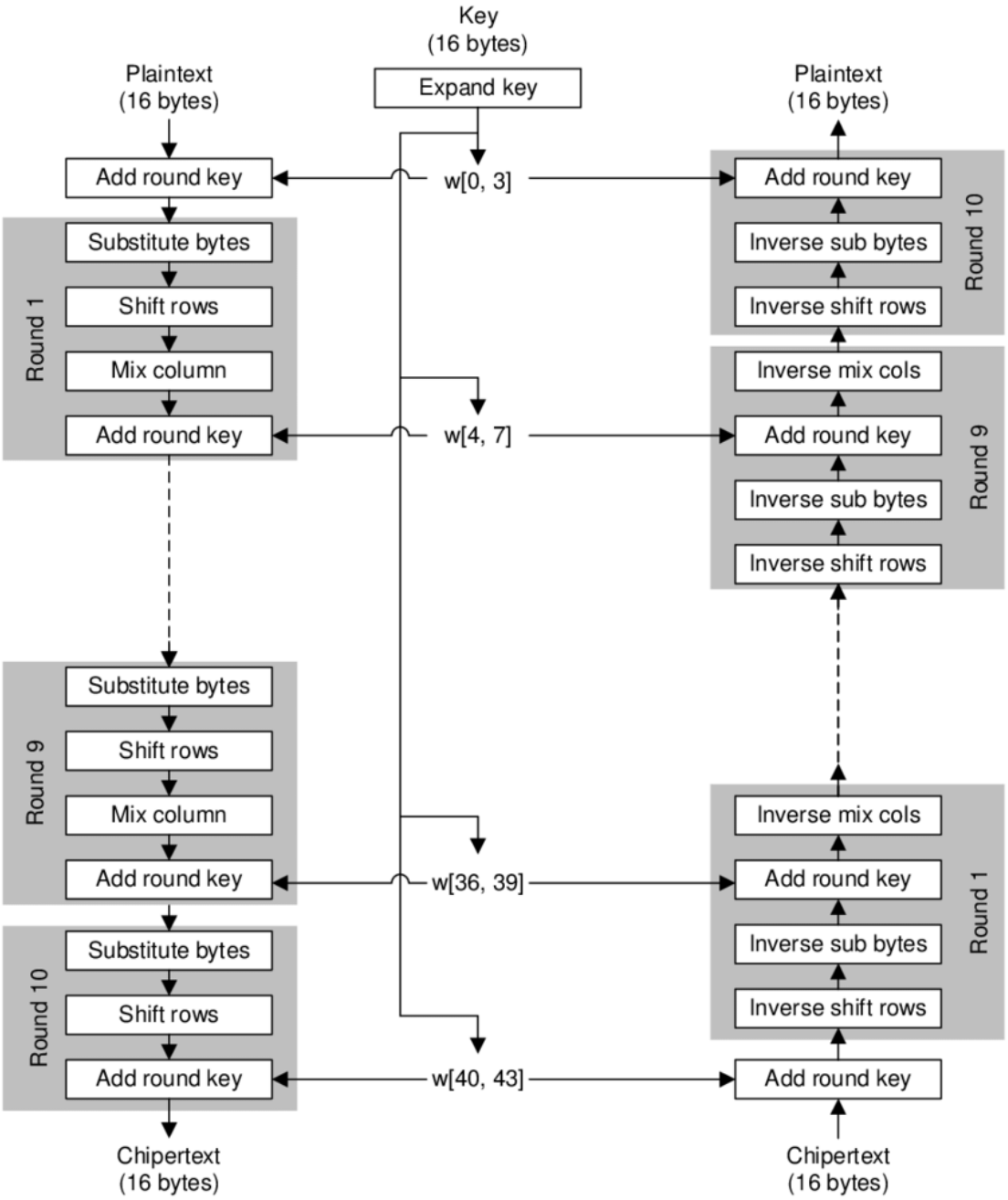
2.1.1. Triple DES

Triple DES je blokovski algoritam za enkripciju podataka, predstavlja korišćenje DES algoritma 3 puta uzastopno, veličina bloka koji se enkriptuje je 64 bita u programu je implementiran u EDE (Enkripcija, Dekripcija, Enkripcija) režimu sa 3 ključa (192 bita od čega se koriste 168 bita samog ključa).



2.1.2. AES

AES je blokovski algoritam za enkripciju podataka čiji je ključ tipično veličine 128, 196 i 256 bita, a veličina bloka koji se enkriptuje je 128 bita. Konkretna implementacija algoritma koja se koristi u programu je AES128 čiji je ključ veličine 128 bita.



2.2. Asimetrični algoritmi

Asimetrične algorime koristimo za enkripciju hesha i kljuceva simetricnih algoritma.

2.2.1. RSA

RSA je asimetrični algoritam koji funkcioniše koristeći principe “public-key” kriptografije. RSA koristi dva ključa gde se jedan smatra da je javni, a drugi privatni ključ. U zavisnosti od toga koji ključ se koristi može se obezbediti ili tajnost podataka (korišćen je javni ključ za enkripciju) ili autentikacija (korišćen je privatni ključ za enkripciju). U programu je RSA implementiran sa veličinom ključeva 1024, 2048 ili 4096 bita.

3. Klase

3.1. MainWIndow

Main klasa programa služi inicijalizaciji i stvaranje GUI-a.

public static void main(String[] args) – main method koja služi za pokretanje GUI-a.

public MainWindow() – konstruktor klase koristi se za inicijalizaciju GUI-a.

private String getPasswordDialog(String s) – otvara dialog za unos šifre.

public void setData() – popunjava sve tabele i liste potrebnim podacima.

3.2. Keys

Klasa koja se koristi za rad sa ključevima. Sadrži inner klasu KeyData, koja se samo koristi kao struktura za čuvanje bitnih podataka vezanih za ključ, i enum RSA_Key_Size, samo označava veličinu ključa.

public static Keys getInstance() – vraća instancu Keys klase.

public List<KeyData> getAllKeys() – vraća listu svih ključeva.

public KeyData GenerateNewKeyPair(String userId, String password, RSA_Key_Size keySize) – generiše nov par ključeva.

private PGPSecretKey GetSecretKey(KeyData keyData) – vraća bouncy castle objekat PGPSecretKey za odredjeni ključ.

public void ExportPrivateKey(KeyData keyData, String FileName) – eksportuje PGPSecretKey za dati ključ.

public void ExportPublicKey(KeyData keyData, String FileName) – eksportuje PGPPublicKey za dati ključ.

public boolean ImportKey(String FileName) – importuje ključ iz datog fajla, vraća true ako je bio privatni ključ.

public void ImportPrivateKey(PGPSecretKeyRing keyRing, String keyPassword) – importuje privatni ključ.

public void ImportPublicKey(PGPPublicKeyRing keyRing) – importuje javni ključ.

3.3. PGP

Klasa predstavlja implementaciju OpenPGP-a.

private static byte[] encrypt(byte[] data, List<Keys.KeyData> receivers, int encryptionType, boolean isFirst, String fileName) – enkriptuje dati niz bajtova.

private static byte[] compress(byte[] data, boolean isFirst, String fileName) – kompresuje dati niz bajtova koristeći ZIP.

private static byte[] signe(byte[] data, Keys.KeyData sender, String fileName) – potpisuje dati niz bajtova.

public static void PGPEncryptImplementation(String fileName, String saveAsFileName, Keys.KeyData sender, List<Keys.KeyData> receivers, boolean isSigned, boolean isAesUsed, boolean isTDesUsed, boolean isCompressed, boolean isRadix64) – implementira enkripciju podataka u OpenPGP koristeći iznad spomenute metode.

public static void exportData(String fileName) – eksportuje dekriptovane podatke.

public static void PGPDecryptionImplementation(String fileName) – implementira dekripciju podataka u OpenPGP.