Univerzitet u Beogradu – Elektrotehnički fakultet
Katedra za računarsku tehniku i informatiku

Odsek za softversko inženjerstvo

Domaći zadatak iz Zaštite podataka

Boško Ćurčin 0549/2016

Nikola Dimitrijević 0597/2016

Beograd, januar 2020.

1. Realizacija aplikacije

Realizacija igre ima 18 klasa. U nastavku slede opisi istih, sa potpisima i opisima metoda i polja.

1.1. Dekriptor.java

Klasa koja sadrži metode kojima se fajl dekriptuje.

1.1.1. Metode klase

```
	/**
	 * Metoda koja dekriptuje fajl i vraca instancu klase RezultatDekripcije
	 *
	 * @param in
	 *            Input stream fajla koji se dekriptuje
	 * @param passwd
	 *            Lozinka kojom je zasticen privatni kljuc koji odogvara javnom
	 *            kljucu kojim je poruka enkriptovana
	 * @param out
	 *            Output stream dekriptovanog fajla
	 * @return Instanca klase RezultatDekripcije
	 * @throws Exception
	 */
	public static RezultatDekripcije decryptFile(InputStream in, char[] passwd,
OutputStream out) throws Exception

	/**
	 * Metoda koja obradu sadržaja enkriptovane poruke i upis u output fajl
	 *
	 * @param ld
	 *            Objekat sadržaja enkriptovane poruke
	 * @param out
	 *            OutputStream fajla za upis dekriptovanog fajla
	 * @param sig
	 *            Objekat omotačke klase PGPSignature
	 * @return Naziv dekriptovanog fajla
	 * @throws IOException
	 * @throws SignatureException
	 */
	private static String processLiteralData(PGPLiteralData ld, OutputStream
out, PGPOmotacPotpisa sig)
			throws IOException, SignatureException
```

1.2. Enkriptor.java

Klasa koja sadrži metode kojima se fajl enkriptuje uz mogućnost potpisivanja.

1.2.1. Metode klase

```
	/**
	 * Metoda koja vrsi enkripciju uz opciono potpisivanje i zipovanje/radix64
	 * konverziju
	 *
	 * @param out
	 *            OutputStream enkriptovanog fajla
	 * @param fileName
	 *            Naziv fajla koji treba da se enkriptuje
	 * @param encKeys
	 *            Kolekcija javnih kkljuceva za enkripciju
	 * @param armor
	 *            Da li je radix64 konverzija omogucena
	 * @param signWithKey
	 *            Tajni kljuc za potpis poruke
	 * @param signKeyPass
	 *            Lozinka pod kojom se tajni kljuc cuva
	 * @param zip
	 *            Da li je zip kompresija omogucena
	 * @param algoritam
	 *            Simetricni algoritam za enkriptovanje
	 * @throws Exception
	 */
	public static void encryptFile(OutputStream out, String fileName,
Collection<PGPPublicKey> encKeys, boolean armor,
                   PGPSecretKey signWithKey, char[] signKeyPass, boolean zip,
String algoritam) throws Exception
```

1.3. DodavanjeBrisanje.java

JPanel koji sadrži GUI elemente za dodavanje i brisanje ključeva.

1.3.1. Polja klase

```
    /**
     * Unos imena korisnika za kreiranje novog para kljuceva
     */
    private JTextField imeTextField;
    /**
     * Unos mejla korisnika za kreiranje novog para kljuceva
     */
    private JTextField mejlTextField;
    /**
     * Unos lozinke korisnika za kreiranje novog para kljuceva
```

```
 */
private JPasswordField passwordField;
/**
 * Unos lozinke kljuca za brisanje kljuca iz prstena
 */
private JPasswordField deleteKeyPasswordField;
/**
 * Lista javnih kljuceva koji mogu da se obrisu
 */
private JComboBox brisanjeJCh;// Brisanje javnog
/**
 * Lista javnih kljuceva koji mogu da se eksportuju
 */
private JComboBox exportJCh;// Export javnog
/**
 * Lista privatnih kljuceva koji mogu da se obrisu
 */
private JComboBox brisanjePCh;// Brisanje privatnog
/**
 * Lista privatnih kljuceva koji mogu da se e
 */
private JComboBox exportPCh;// Export privatnog
/**
 * Da li se importuje privatni ili javni kljuc
 */
private JCheckBox chckbxNewCheckBox;// Da li je kljuc privatan
/**
 * Poruka o uspehu/neuspehu
 */
private JLabel poruka;
```

1.3.2. Metode klase

```
/**
 * Konstruktor klase u kojem se inicijalizuju GUI elementi
 */
public DodavanjeBrisanje()

/**
 * Vrsi update liste javnih kljuceva nakon nekih promena
 */
public void updatePubKeysList()

/**
 * Vrsi update liste privatnih kljuceva nakon nekih promena
 */
public void updatePrivKeysList()

/**
 * @param data niz bajtova koji definisu kljuc
 * @throws Exception
 */
private void saveKey(byte[] data) throws Exception
```

1.4. JavniKljucevi.java

GUI prikaz liste javnih kljuceva

1.4.1. Polja klase

```
/**
 * Lista tajnih kljuceva
 */
private JList rsaTajnost;
/**
 * GUI element za ispis
 */
JScrollPane lista;
```

1.4.2. Metode klase

```
/**
 * Konstruktor, inicijalizuje GUI elemente
 */
public JavniKljucevi()

/**
 * Vrsi update liste javnih kljuceva nakon nekih promena
 */
public void updatePubKeysList()
```

1.5. Main.java

Main klasa koja instancira celokupan GUI.

1.5.1. Polja klase

```
/**
 * JPanel-i za funkcionalnosti aplikacije
 */
private JPanel dodavanjeBrisanje;
private JPanel prijemPoruke;
private JPanel slanjePoruke;
private JPanel javniKljucevi;
private JPanel privatniKljucevi;
/**
 * Liste javnih i privatnih kljuceva
 */
public static DefaultListModel privateKeys;
public static DefaultListModel publicKeys;
```

1.5.2. Metode klase

```
/**
 * Konsturktor, inicijalizuje polja klase
 */
public Main()

/**
 * Akcija menija, obradjue dogadjaje promene panela
 *
 */
private class MenuAction implements ActionListener

/**
 * Inicijalizacija GUI-a aplikacije
 */
private void initMenu()

/**
 * Obrada promene vidljivog panela
 * @param panel Panel za promenu
 */
private void changePanel(JPanel panel)

/**
 * Vrsi update liste javnih kljuceva nakon nekih promena
 */
public void updatePubKeysList()

/**
 * Vrsi update liste privatnih kljuceva nakon nekih promena
 */
public void updatePrivKeysList()

/**
 * Glavna nit aplikacije
 * @param args
 */
public static void main(String[] args)
```

1.6. PGPOmotacPotpisa.java

Omotačka klasa sa PGPSignature i PGPOnePassSignature.

1.6.1. Polja klase

```
/**
 * Potpis poruke, i da li je on OnePass ili nije
 */
PGPOnePassSignature sigOnePass;
PGPSignature sigOldStyle;
boolean isOnePass;
```

1.6.2. Metode klase

```
	/**
	 * Metode koje omotavaju metode klasa PGPSignature i PGPOnePassSignature
	 */

	public PGPOmotacPotpisa(PGPOnePassSignature sigOnePass)

	public PGPOmotacPotpisa(PGPSignature sigOldStyle)

	public void encode(OutputStream outStream)

	public byte[] getEncoded() throws IOException

	public int getKeyAlgorithm()

	public int getHashAlgorithm()

	public long getKeyID()

	public long getSignatureType()

	public void initVerify(PGPPublicKey pubKey, String provider)

	public void update(byte b) throws SignatureException

	public void update(byte[] bytes) throws SignatureException

	public void update(byte[] bytes, int off, int len) throws
SignatureException

	public boolean verify(PGPSignature pgpSig) throws PGPException,
SignatureException
```

1.7. Potpisivac.java

Klasa koja je roditeljska klasa za implementacije za traženje najboljih poteza različitim algoritmima.

1.7.1. Metode klase

```
	/**
	 * @param plainText Text koji treba da se potpise
	 * @param enckey Tajni kljuc za potpisivanje
	 * @param pass Lozinka pod kojom se cuva tajni kljuc
	 * @param zip Da li se koristi zip kompresija
	 * @param radix Da li se koristi radix64 konverzija
	 * @return Potpisana poruka
	 * @throws Exception
	 */
	public static String signText(String plainText, PGPSecretKey enckey, char[]
pass, boolean zip, boolean radix)
			throws Exception

	/**
	 * @param in InputStream fajla koji se potpisuje
	 * @param out OutputStream u koji ce se upisati potipisana poruka
	 * @param key Tajni kljuc za potpisivanje
	 * @param pass Lozinka pod kojom se cuva tajni kljuc
	 * @param textmode Tip poruke
	 * @throws IOException
	 * @throws NoSuchAlgorithmException
	 * @throws NoSuchProviderException
	 * @throws PGPException
	 * @throws SignatureException
	 */
	public static void signFile(InputStream in, OutputStream out, PGPSecretKey
key, char[] pass, boolean textmode)
			throws IOException, NoSuchAlgorithmException,
NoSuchProviderException, PGPException, SignatureException
```

1.8. PPGPJavniKljuc.java

Omotačka klasa koja omogućava ispis PGPPublicKey-a.

1.8.1. Polja klase

```
/**
 * PGP javni kljuc
 */
PGPPublicKey base;
```

1.8.2. Metode klase

```
/**
 * Konstruktor
 * @param iBase PGP javni kljuc
 */
public PPGPPJavniKljuc(PGPPublicKey iBase)

/**
 * Getter
 * @return PGP javni kljuc
 */
public PGPPublicKey getPublicKey()

@Override
public String toString()
```

1.9. PPGPPrstenJavnihKljuceva.java

Omotačka klasa koja omogućava ispis PGPPublicKeyRing-a.

1.9.1. Polja klase

```
/**
 * Prsten javnih kljuceva
 */
PGPPublicKeyRing base;
```

1.9.2. Metode klase

```
/**
 * Konsturktor
 * @param iBase Prsten javnih kljuceva
 */
public PPGPPrstenJavnihKljuceva(PGPPublicKeyRing iBase)

/**
 * Getter
 * @return Prsten javnih kljuceva
 */
public PGPPublicKeyRing getPublicKeyRing()

/**
 * Dohvatanje master kljuca
 * @return Master kljuceva
 */
public PGPPublicKey getMasterKey()

/**
 * Dohvatanje kljuca za enkripciju
 * @return Kljuc za enkripciju
 */
public PGPPublicKey getEncryptionKey()


@Override
public boolean equals(Object obj)

@Override
public int hashCode()

@Override
public String toString()
```

1.10. PPGPPrstenTajnihKljuceva.java

Omotačka klasa koja omogućava ispis PGPSecretKeyRing-a

1.10.1. Polja klase

```
/**
 * Prsten tajnih kljuceva
 */
PGPSecretKeyRing base;
```

1.10.2. Metode klase

```
/**
 * Konstruktor
 * @param iBase Prsten tajnih kljuceva
 */
public PPGPPrstenTajnihKljuceva(PGPSecretKeyRing iBase)

/**
 * Getter
 * @return Prsten tajnih kljuceva
 */
public PGPSecretKeyRing getSecretKeyRing()

/**
 * Dohvatanje master kljuca
 * @return Tajni kljuc
 */
public PGPSecretKey getMasterKey()

/**
 * Dohvatanje kljuca za dekripciju
 * @return Tajni kljuc
 */
public PGPSecretKey getDecryptionKey()

/**
 * Dohvatanje kljuca za potpisivanje
 * @return Tajni kljuc
 */
public PGPSecretKey getSigningKey()

@Override
public String toString()
```

1.11. PPGPTajniKljuc.java

Omotačka klasa koja omogućava ispis PGPSecretKey-a.

1.11.1. Polja klase

```
/**
 * Tajni kljuc
 */
PGPSecretKey base;
```

1.11.2. Metode klase

```
/**
 * Konstruktor
 * @param iBase Tajni kljuc
 */
public PPGPTajniKljuc(PGPSecretKey iBase)

/**
 * Getter
 * @return Tajni kljuc
 */
public PGPSecretKey getSecretKey()

@Override
public String toString()
```

1.12. PrijemPoruke.java

JPanel koji sadrži GUI elemente za dekripciju i verifikaciju poruka.

1.12.1. Polja klase

```
/**
 * Tekstualno polje za unos putanje do zeljenog fajla za dekripciju
 */
private JTextField putanja;

/**
 *     Lozinka za privatni kljuc kojim je fajl enkriptovan
 */
private JPasswordField lozinka;

/**
 * Tekstualno polje za ispis putanje odabranog fajla za proveru potpisa
 */

private JTextField putanjaPotpisa;
```

1.12.2. Metode klase

```
/**
 * Konstruktor, inicijalizacija GUI-ja
 */
public PrijemPoruke()

/**
 * Odabirac fajlova
 * @return Odabrani fajl
 */
public File chooseFile()

/**
 * Metoda koja prikazuje uspesnost dekripcije
 * @param resRezultat dekripcije
 * @return   (ne)uspesnost dekripcije
 */
boolean confirmDecryption(RezultatDekripcije res)

/**
 * Metoda koja prikazuje uspesnost verifikacije
 * @param resRezultat verifikacije
 */
void confirmVerification(String res)
```

1.13. PrivatniKljucevi.java

GUI prikaz liste privatnih kljuceva.

1.13.1. Polja klase

```
/**
 *  Lista tajnih kljuceva
 */
private JList rsaEnkripcija;

/**
 *     GUI element za ispis
 */
JScrollPane lista;
```

1.13.2. Metode klase

```
/**
 * Konstruktorm, inicijalizacija
 */
public PrivatniKljucevi()
```

```
/**
 * Apdejtovanje liste privatnih kljuceva
 */
public void updatePrivKeysList()
```

1.14. PristenKljuceva.java

Klasa u kojoj su metode za obradu prstena javnih i tajnih kljuceva.

1.14.1. Polja klase

```
/**
 * Ime fajla u kome se cuvaju privatni kljucevi
 */
private static final String PRIVATE_KEYRING_FILE = "privatniKljucevi.bpg";

/**
 * Ime fajla u kome se cuvaju javni kljucevi
 */
private static final String PUBLIC_KEYRING_FILE = "javniKljucevi.bpg";

/**
 * Default simetricni algoritam
 */
private static final int KEY_ENCRYPTION_ALGO = PGPEncryptedData.CAST5;

/**
 * Prsten javnih kljuceva
 */
private static PGPPublicKeyRingCollection pubring;

/**
 * Prsten privatnih kljuceva
 */
private static PGPSecretKeyRingCollection secring;

/**
 * Fajl u kome ce da se cuvaju javni kljucevi
 */
private static File pubringFile = new File(PUBLIC_KEYRING_FILE);
/**
 * Fajl u kome ce da se cuvaju privani kljucevi
 */
private static File secringFile = new File(PRIVATE_KEYRING_FILE);
```

1.14.2. Metode klase

```
/**
 * Ucitavanje prstenova javnih i privatnih kljuceva
 */
private static void loadKeyrings()

/**
```

```
         * Import javnog kljuca
         * @param iKeyStream InputKeyStream javnog kljuca
         * @return   Prsten javnih kljuceva
         * @throws IOException
         */
        public static PGPPublicKeyRing importPublicKey(InputStream iKeyStream)
throws IOException

        /**
         * Import privatnog kljuca
         * @param iKeyStream InputStream privatnog kljuca
         * @return Prsten privatnih kljuceva
         * @throws IOException
         * @throws PGPException
         */
        public static PGPSecretKeyRing importPrivateKey(InputStream iKeyStream)
throws IOException

        /**
         * Kreiranje novog para javnog i tajnog kljuca
         * @param iKeySize  Velicina kljuca u bitima
         * @param iUserID   ID korisnika koji pravi kljuceve
         * @param iPassphrase     Lozinka pod kojom ce se cuvati privatni kljuc
         * @throws Exception
         */
        public static void generateNewKeyPair(int iKeySize, String iUserID, char[]
iPassphrase) throws Exception

        /**
         * Dohvatanje kolekcije prstena javnih kljuceva
         * @return Kolekcija prstena javnih kljuceva
         */
        public static Collection<PPGPPrstenJavnihKljuceva> getPublicKeys()

        /**
         * Dohvatanje kolekcije prstena tajnih kljuceva
         * @return Kolekcija prstena tajnih kljuceva
         */
        public static Collection<PPGPPrstenTajnihKljuceva> getPrivateKeys()

        /**
         * Dohvatanje tajnog kljuca pomocu ID-ja korisnika
         * @param iID  ID korisnika
         * @return   Tajni kljuc
         * @throws PGPException
         */
        public static PGPSecretKey getPrivateKeyByID(long iID) throws PGPException

        /**
         * Dohvatanje javnog kljuca pomocu ID-ja korisnika
         * @param iID ID korisnika
         * @return   Javni kljuc
         * @throws PGPException
         */
        public static PGPPublicKey getPublicKeyByID(long iID) throws PGPException

        /**
         * Brisanje javnog kljuca
         * @param iKey Javni kljuc koji je korisnik odabrao
```

```
     * @throws IOException
     */
    public static void deletePublicKey(PGPPublicKeyRing iKey) throws IOException

    /**
     * Brisanje privatnog kljuca
     * @param iKey Privatni kljuc koji je korisnik odabrao
     * @throws IOException
     */
    public static void deletePrivateKey(PGPSecretKeyRing iKey) throws IOException
```

1.15. RezultatDekripcije.java

Klasa koja nam enkapsulira rezultat dekripcije.

1.15.1. Polja klase

```
    /**
     * Polja koja opisuju rezultat dekripcije
     */
    private String decryptFileName = "";
    private boolean isSigned = false;
    private PPGPPJavniKljuc signee = null;
    private boolean isSignatureValid = false;
    private Exception signatureException = null;
    private String decryptedText;
```

1.15.2. Metode klase

Geteri i seteri gorenavedenih polja.

1.16. SlanjePoruke.java

GUI klasa kojom se korisniku omogucava slanje poruke, uz odabir zeljenih algoritama, kao i zip kompresije ili radix64 konverzije.

1.16.1. Polja klase

```
/**
 * GUI polja za korisnicki unos
 */
private JTextField textField;
private JPasswordField passwordField;
JList rsaTajnost;
JScrollPane lista;
JComboBox rsaEnkripcija;
JComboBox choice;
JCheckBox checkZip;
JCheckBox checkRadix;
JCheckBox checkAutPot;
JCheckBox checkEnkTaj;
```

1.16.2. Metode klase

```
/**
 * Konstruktor, inicijalizacija
 */
public SlanjePoruke()

/**
 * Odabirac fajlova
 * @return Odabrani fajl
 */
public File chooseFile()

/**
 * Apdejtovanje liste javnih kljuceva
 */
public void updatePubKeysList()

/**
 * Apdtejtovanje liste privatnih kljuceva
 */
public void updatePrivKeysList()
```

1.17. Verifikator.java

Klasa koja nam sluzi za verifikaciju potpisanih fajlova.

1.17.1. Metode klase

```
        /**
         * Pomocna metoda kojom se radi verifikacija poruke
         * @param plainText Tekst potpisane poruke
         * @return   Javni kljuc kojim je poruka verifikovana
         * @throws Exception
         */

        public static String verifyText(String plainText) throws Exception

        /**
         * Metoda kojom se radi verifikacija poruke
         * @param in Tekst potpisane poruke
         * @param keyIn Tekst potpisa poruke
         * @return   Javni kljuc kojim je poruka verifikovana
         * @throws Exception
         */
        public static String verifyFile3(String in, InputStream keyIn) throws
Exception
```