Универзитет у Београду

Електротехнички факултет

Одсек СИ

Заштита података

Пројектни задатак – извештај

Студенти:

Никола Миличевић 0387/17

Александра Богићевић 0390/17

Београд, школска година 2020/2021

Историја измена

Датум	Верзија	Кратак опис	Аутор
30.05.2021.	1.0	Иницијална верзија	Александра Богићевић, Никола Миличевић

Садржај

Увод

У овом пројектном задатку, у оквиру предмета Заштита података, реализована је апликација са графичким корисничким интерфејсом у програмском језику *Java* која омогућава следеће функционалности:

- Генерисање новог и брисање постојећег пара кључева
- Увоз и извоз јавног или приватног кључа у *.asc* формату
- Приказ прстена јавних и приватних кључева са свим потребним информацијама
- Слање поруке (уз обезбеђивање енкрипције и потписивања)
- Примање поруке (уз обезбеђивање декрипције и верификације)

У наставку извештаја биће детаљније описан пројекат, приказане реализоване класе, са потписима и описима метода.

Напомена

При изради су коришћени алгоритми који су додељени групи 4, а то су:

- *RSA* за потписивање и енкрипцију са кључевима величине 1024, 2048 или 4096 бита, од алгоритама за симетричне кључеве,
- *3DES* са *EDE* конфигурацијум и три кључа и *IDEA*, од алгоритама за симетричне кључеве.

Од библиотека коришћена је *Bouncy Castle Cryptography Library*.

Све класе се налазе у оквиру *etf.openpgp.mn170387dba170390d* пакета.

Реализоване класе

У наставку су побројане реализоване класе:

- Main
- StartupController
- MyKey
- NewKeyPairController
- PasswordController
- PwKeyController
- SecOrPubController
- SignEncryptController
- DecrVerifController

Следи опис класа и њихових метода.

Main class

```
public class Main extends Application {

    public static Stage primaryStage;
    public static FXMLLoader loader;

    public static PGPPublicKeyRingCollection pkrcoll;
    public static PGPPublicKeyRingCollection pkrcollmy;
    public static PGPSecretKeyRingCollection skrcoll;

    static {
        try {
            InputStream inputPkrMy = new FileInputStream("usermy.pkr");
            InputStream inputPkr = new FileInputStream("user.pkr");
            InputStream inputSkr = new FileInputStream("user.skr");

            inputPkrMy = new ArmoredInputStream(inputPkrMy);
            inputPkr = new ArmoredInputStream(inputPkr);
            inputSkr = new ArmoredInputStream(inputSkr);

            pkrcoll = new BcPGPPublicKeyRingCollection(inputPkr);
            skrcoll = new BcPGPSecretKeyRingCollection(inputSkr);
            pkrcollmy = new BcPGPPublicKeyRingCollection(inputPkrMy);


        } catch (IOException e) {
            e.printStackTrace();
        } catch (PGPException e) {
            e.printStackTrace();
        }
    }

    @Override
    public void start(Stage primaryStage);

    public static Stage getPrimaryStage();

    public static void main(String[] args);

}
```

Класа *Main* покреће програм. Како је интерфејс рађен у *JavaFX*, класа *Main* покреће главни контролер – *StartupController* који приказује прозор са апликацијом.

StartupController class

```
1  public class StartupController implements Initializable {
2
3      @SuppressWarnings("unchecked")
4      @Override
5      public void initialize(URL arg0, ResourceBundle arg1);
6
7      @FXML
8      void decryptOrVerifyFile(ActionEvent event);
9
10     @FXML
11     void generateKeyPair(ActionEvent event);
12
13     @FXML
14     void importKey(ActionEvent event) throws IOException, PGPException;
15
16     @FXML
17     void signOrEncryptFile(ActionEvent event);
18
19     public ObservableList<MyKey> getKeys();
20
21     public void refreshz();
22
23 }
```

Ова класа представља главни контролер апликације.

У методи *initialize* се постављају компоненте на прозор апликазије и одређује се функција сваког елемента. Исписују се кључеви које корисник поседује и омогућава се да се десним кликом на један од кључева он обрише или извезе.

Методе *decryptOrVerifyFile*, *generateKeyPait*, *signOrEncryptFile* се позивају притиском на дугме са истим натписом, чиме се позивају контролери *DecrVerifController*, *NewKeyPairController* и *SignEncryptController*.

Метода *importKey* омогућава увоз кључа који је сачуван на рачунару на ком се покреће апликација, у формату *.asc*.

Метода *ObservableList* прави колекцију кључева која се потом табеларно исписује на почетном екрану апликације.

Метода *refreshz* омогућава освежавање табеле с кључевима.

MyKey

```
public class MyKey {
    private String name;
    private String email;
    private String keyID;
    private long keyIdLong;
    private boolean isPublic;

    public MyKey(String name_, String email_, String keyID_, long keyidlong, boolean flag);

    public String getName();

    public void setName(String name);

    public String getEmail();

    public void setEmail(String email);

    public String getKeyID();

    public void setKeyID(String keyID);

    public long getKeyIdLong();

    public void setKeyIdLong(long keyIdLong);

    public boolean isPublic();

    public void setPublic(boolean isPublic);

    @Override
    public String toString();

}
```

Класа *MyKey* представља структуру кључа и садржи све потребне информације које се приказју кориснику.

Методе су гетери и сетери чија имена јасно говори чему служе.

NewKeyPairController

```
public class NewKeyPairController implements Initializable {

    @Override
    public void initialize(URL arg0, ResourceBundle arg1);

    private void setKeySize(ActionEvent e);

    public static void setStage(Stage st);

    @FXML
    void submitKeyPairData(ActionEvent event);


    public static void setPwAndGen(String pw) throws PGPException, IOException;

    public static void generate(String id, String password, int keysize) throws PGPException, IOException;

    private static PGPKeyRingGenerator generateKeyRingGenerator(String id, char[] pass, int s2kcount, int keysize)
            throws PGPException, FileNotFoundException;
}

```

Класа *NewKeyPairController* служи за приказивање прозора неопходног да би корисник успешно креирао пар кључева и за креирање истих.

У методи *initialize* се поставља прозор, при чему ће корисник моћи да обележи како жели да генерише свој пар кључева.

submitKeyPairData је метода у којој се проверавају унете вредности и потом се од корисника тражи генерисање лозинке за дати кључ.

Метода *generateKeyRingGenerator* формира генератор кључева помоћу ког корисник генерипе пар кључева. Генератору се прослеђују сви неопходни параметри, попут величине кључа који се генерише, алгоритама који се користи за енкрипцију и потписивање и прослеђује лозинку под којом се кључ чува.

Метода *generate* додаје кључеве (који су генерисани помоћу генератора из претходно описане методе) у одређене *KeyRing*-ове, при чему бивају запамћени у систему и кориснику се поново приказује почетни екран. Тада, на почетном екрану, у табели може видети и новогенерисани пар кључева.

PasswordController class

```
1  public class PasswordController {
2
3      public static void setStage(Stage st) {
4          stage = st;
5      }
6
7      @FXML
8      void passwordCheckAndGen(ActionEvent event) throws PGPException, IOException;
9  }
```

Класа која служи за проверу исправности унетих лозинки при креирању исте.

PwKeyController class

```
1  public class PwKeyController {
2
3      public static void setStage(Stage st) {
4          stage = st;
5      }
6
7
8      @FXML
9      void pwDeleteKey(ActionEvent event);
10
11 }
```

Класа која поседује методу која пружа могућност брисања кључева.

SecOrPubController class

```
1  public class SecOrPubController {
2
3      public static void setStage(Stage st) {
4          stage = st;
5      }
6
7      @FXML
8      void export(ActionEvent event);
9
10     @FXML
11     void getDecision(ActionEvent event);
12 }
```

Помоћна класа која при експортовању кључева доставља додатне информације класи *StartupController*, а то је да ли се покушава експортовање јавног или тајног кључа.

SignEncryptController

```
1  public class SignEncryptController implements Initializable {
2
3      public static void setStage(Stage st);
4
5      @FXML
6      void encryptCheckboxChange(ActionEvent event);
7
8      @FXML
9      void radixCheckboxChange(ActionEvent event);
10
11     @FXML
12     void signCheckboxChange(ActionEvent event);
13
14     @FXML
15     void zipCheckboxChange(ActionEvent event);
16
17
18     @FXML
19     void submit(ActionEvent event);
20
21     @FXML
22     void selectFile(ActionEvent event);
23
24     @Override
25     public void initialize(URL arg0, ResourceBundle arg1);
26
27     private List<String> getSecretKeys();
28
29     private List<String> getPublicKeys();
30
31
32
33     public void getSecretKey(ActionEvent e);
34
35     public void getPublicKeyIDsHEX();
36
37
38     @FXML
39     void getAlgo(ActionEvent event);
40
41 }
```

Класа у којој се врши енкрипција и потписивање поруке (тзв. слање поруке).

Методе које у називу имају CheckboxChange региструју промене на корисничком интерфејсу. Наиме, корисник треба да обележи да ли и за кога жели да енкриптује поруку, да ли жели да потпише (и којим кључем), да ли жели да компресује податке и да ли жели да их претвори у radix-64. Избор вржи обележавањем одређених поља.

Метода *selectFile* отвара претрагу датотека на компјутеру и тражи да се одабере порука за слање, док метода *submit* покреће алгоритам потписивања/анкриптовања/зиповања/радикс конвертовања, у зависности од тога шта је корисник обележио. Прво се врши потписивање, потом зиповање, енкрипција и на крају радикс конверзија.

GetPublicKeys и *getSecretKeys* су помоћне методе које дохватају све кључеве које корисник поседује, како би му се исписале опције којим кључем жели да потпише и за кога жели да енкриптује.

DecrVerifController class

```
1  public class DecrVerifController {
2
3      public PGPSecretKey findSecretKey(long publicKeyId);
4
5      public PGPPublicKey findPublicKey(long id);
6
7
8      @SuppressWarnings("resource")
9      @FXML
10     void decryptOrVerifyFile(ActionEvent event);
11
12 }
```

Класа у којој се врши декрипција и верификација поруке (тзв. пријем поруке).

Метода *findSecretKey* проналази приватни кључ на основу јавног кључа.

Метода *findPulicKey* проналази јавни кључ на основу ид кључа.

Метода *dectyptOrVerifyFile* отвара прозор за селекцију фајла над којим корисник жели да ради. На основу датог фајла апликација препознаје о којим пакетима се ради и који алгоритми су коришћени за потписивање и енкрипцију, уколико су коришћени.

Порука се дешифрује и кориснику се приказују информације ко је потписао поруку, чиме је енкриптована и да ли је обезбеђен интегритет.