

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

Историја измена	2
Увод.....	4
Напомена	4
Реализоване класе	5
Main class	6
StartupController class	7
MyKey	8
NewKeyPairController	9
PasswordController class	10
PwKeyController class	11
SecOrPubController class	12
SignEncryptController	13
DecrVerifController class	15

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

```
1 public class Main extends Application {
2
3     public static Stage primaryStage;
4     public static FXXMLLoader loader;
5
6     public static PGPPublicKeyRingCollection pkrcoll;
7     public static PGPPublicKeyRingCollection pkrcollmy;
8     public static PGPSecretKeyRingCollection skrcoll;
9
10    static {
11        try {
12            InputStream inputPkrMy = new FileInputStream("usermy.pkr");
13            InputStream inputPkr = new FileInputStream("user.pkr");
14            InputStream inputSkr = new FileInputStream("user.skr");
15
16            inputPkrMy = new ArmoredInputStream(inputPkrMy);
17            inputPkr = new ArmoredInputStream(inputPkr);
18            inputSkr = new ArmoredInputStream(inputSkr);
19
20            pkrcoll = new BcPGPPublicKeyRingCollection(inputPkr);
21            skrcoll = new BcPGPSecretKeyRingCollection(inputSkr);
22            pkrcollmy = new BcPGPPublicKeyRingCollection(inputPkrMy);
23
24        } catch (IOException e) {
25            e.printStackTrace();
26        } catch (PGPException e) {
27            e.printStackTrace();
28        }
29    }
30
31    @Override
32    public void start(Stage primaryStage);
33
34    public static Stage getPrimaryStage();
35
36    public static void main(String[] args);
37
38
39 }
```

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, PGPEException;
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 апликације 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 изведе.

Методе *decryptOrVerifyFile*, *generateKeyPair*, *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

```
1 public class MyKey {
2     private String name;
3     private String email;
4     private String keyID;
5     private long keyIdLong;
6     private boolean isPublic;
7
8     public MyKey(String name_, String email_, String keyID_, long keyidlong, boolean flag);
9
10    public String getName();
11
12    public void setName(String name);
13
14    public String getEmail();
15
16    public void setEmail(String email);
17
18    public String getKeyID();
19
20    public void setKeyID(String keyID);
21
22    public long getKeyIdLong();
23
24    public void setKeyIdLong(long keyIdLong);
25
26    public boolean isPublic();
27
28    public void setPublic(boolean isPublic);
29
30    @Override
31    public String toString();
32
33 }
```

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

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

NewKeyPairController

```
1 public class NewKeyPairController implements Initializable {
2
3     @Override
4     public void initialize(URL arg0, ResourceBundle arg1);
5
6     private void setKeySize(ActionEvent e);
7
8     public static void setStage(Stage st);
9
10    @FXML
11    void submitKeyPairData(ActionEvent event);
12
13
14    public static void setPwAndGen(String pw) throws PGPEException, IOException;
15
16    public static void generate(String id, String password, int keysize) throws PGPEException, IOException;
17
18    private static PGPKKeyRingGenerator generateKeyRingGenerator(String id, char[] pass, int s2kcount, int keysize)
19        throws PGPEException, FileNotFoundException;
20
21 }
22
```

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 генерише 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 PGPEException, 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 бла бла трућ

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 бла бла