Белорусский Государственный Технологический Университет

Факультет информационных технологий

Отчет к лабораторной работе №2

«ANDROID AES CRYPTING»

Выполнил: студент 3 курса 7 группы Горбачев Д.А.

Проверила: Герман Юлия Олеговна

Минск 2018

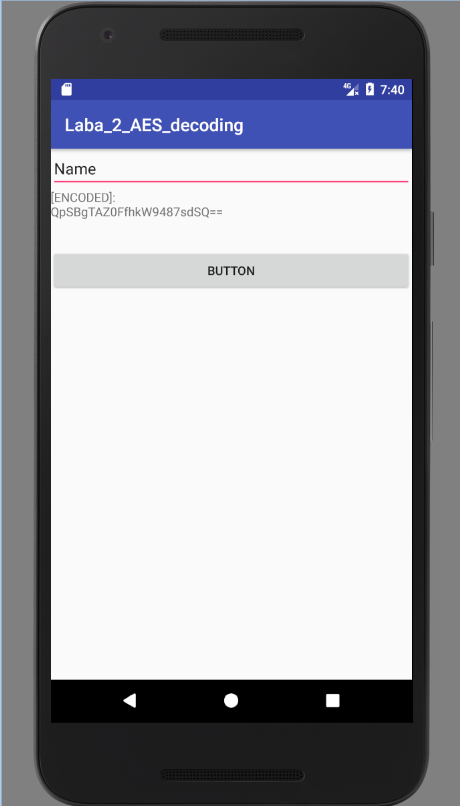
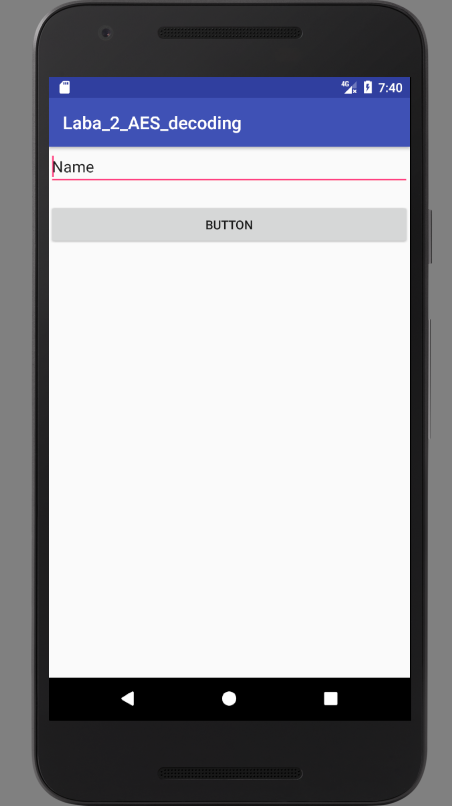
**Цель.** Научиться простому кодированию/декодированию с помощью алгоритма AES.

Ход выполнения

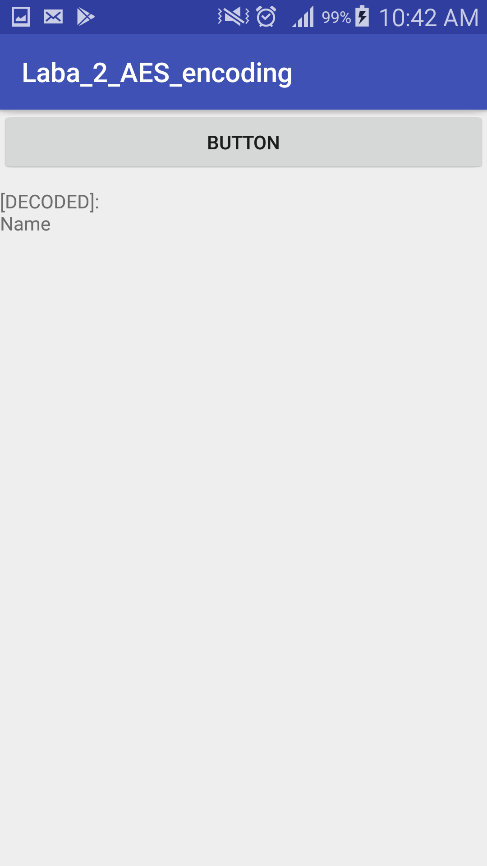
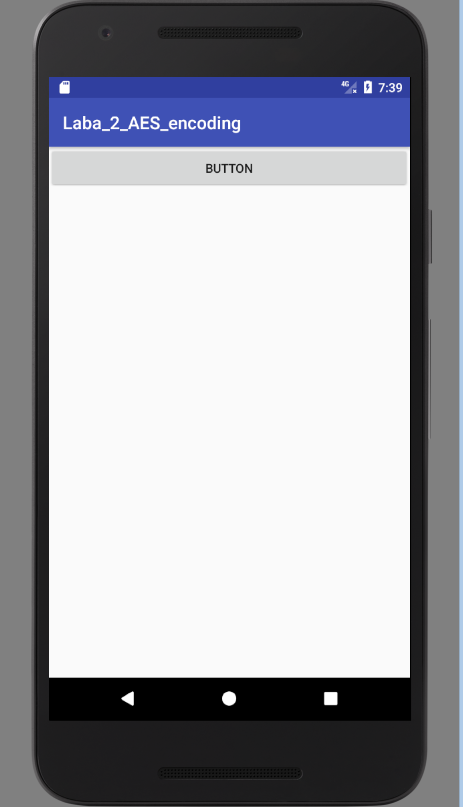
1. Написать два раздельных приложения: одно шифрует, а другое дешифрует текстовую строку. Для этого надо генерируемый ключ сохранить в файле.

**key** = (keyGenerator.generateKey().getEncoded());

**/////////////////////////////////////////////////////////////////////////////**



**public class** MainActivity **extends** AppCompatActivity {  
  
 **final** String **DIR\_SD** = **"MyFiles"**;  
 **final** String **FILENAME\_SD** = **"Code.txt"**;  
  
 **private** SecretKeySpec **sks** = **null**;  
 **private** String **line** = **null**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 }  
  
 **public void** onClick(View v){  
 **line** = ((EditText)findViewById(R.id.***LineForEncoding***)).getText().toString();  
 **if**(!**line**.isEmpty()) {  
 newKey();  
 encoding();  
 }**else**{  
 Toast.*makeText*(getApplicationContext(), **"Line is empty!!!"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
  
 **private void** encoding(){  
 **byte**[] encodedBytes = **null**;  
 **try** {  
 Cipher c = Cipher.*getInstance*(**"AES"**);  
 c.init(Cipher.***ENCRYPT\_MODE***, **sks**);  
 encodedBytes = c.doFinal(**line**.getBytes());  
 } **catch** (Exception e) {  
 Log.*e*(**"Crypto"**, **"AES encryption error"**);  
 }  
 TextView encodedTextView = (TextView)findViewById(R.id.***textViewEncoded***);  
 encodedTextView.setText(**"[ENCODED]:\n"** +  
 Base64.*encodeToString*(encodedBytes, Base64.***DEFAULT***) + **"\n"**);  
  
 writeToFile(encodedBytes);  
 }  
  
 **private void** newKey(){  
 **try** {  
 SecureRandom sr = SecureRandom.*getInstance*(**"SHA1PRNG"**);  
 sr.setSeed(**"any data used as random seed"**.getBytes());  
 KeyGenerator kg = KeyGenerator.*getInstance*(**"AES"**);  
 kg.init(256, sr);  
 **sks** = **new** SecretKeySpec((kg.generateKey()).getEncoded(), **"AES"**);  
 } **catch** (Exception e) {  
 Log.*e*(**"Crypto"**, **"AES secret key spec error"**);  
 }  
 }  
  
 **private void** writeToFile(**byte**[] s){  
 Gson gson = **new** Gson();  
 String jsonString = gson.toJson(**new** Code(**sks**, s));  
 Log.*d*(**"Hello"**, jsonString);  
  
 **if** (!Environment.*getExternalStorageState*().equals(Environment.***MEDIA\_MOUNTED***)) {  
 Log.*d*(**"Hello"**, **"SD-карта не доступна: "** + Environment.*getExternalStorageState*());  
 **return**;  
 }  
 File sdPath = Environment.*getExternalStorageDirectory*();  
 sdPath = **new** File(sdPath.getAbsolutePath() + **"/"** + **DIR\_SD**);  
 sdPath.mkdirs();  
 File sdFile = **new** File(sdPath, **FILENAME\_SD**);  
 **try** {  
 BufferedWriter bw = **new** BufferedWriter(**new** FileWriter(sdFile));  
 bw.write(jsonString);  
 bw.close();  
 Toast.*makeText*(getApplicationContext(), **"File write!!!"**, Toast.***LENGTH\_SHORT***).show();  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 }  
}



**public class** MainActivity **extends** AppCompatActivity {  
  
 **final** String **DIR\_SD** = **"MyFiles"**;  
 **final** String **FILENAME\_SD** = **"Code.txt"**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 }  
  
 **public void** onClick(View v){  
 read();  
 }  
  
 **private void** encoding(SecretKeySpec sks, **byte**[] line){  
 **if**(line.**length** > 0){  
 **byte**[] decodedBytes = **null**;  
 **try** {  
 Cipher c = Cipher.*getInstance*(**"AES"**);  
 c.init(Cipher.***DECRYPT\_MODE***, sks);  
 decodedBytes = c.doFinal(line);  
 } **catch** (Exception e) {  
 Log.*e*(**"Crypto"**, **"AES decryption error"**);  
 }  
  
 TextView decodedTextView = (TextView)findViewById(R.id.***textViewDecoded***);  
 decodedTextView.setText(**"[DECODED]:\n"** + **new** String(decodedBytes));  
 }  
 }  
  
 **private void** read(){  
 Gson gson = **new** Gson();  
  
 **if** (!Environment.*getExternalStorageState*().equals(Environment.***MEDIA\_MOUNTED***)) {  
 Log.*d*(**"Hello"**, **"SD-карта не доступна: "** + Environment.*getExternalStorageState*());  
 **return**;  
 }  
  
 **if** (!Environment.*getExternalStorageState*().equals(  
 Environment.***MEDIA\_MOUNTED***)) {  
 **return**;  
 }  
 File sdPath = Environment.*getExternalStorageDirectory*();  
 sdPath = **new** File(sdPath.getAbsolutePath() + **"/"** + **DIR\_SD**);  
 File sdFile = **new** File(sdPath, **FILENAME\_SD**);  
 **try** {  
 BufferedReader br = **new** BufferedReader(**new** FileReader(sdFile));  
 String str = **""**;  
 **while** ((str = br.readLine()) != **null**) {  
 Code dataItems = gson.fromJson(str, Code.**class**);  
 encoding(dataItems.getKey(), dataItems.getCode());  
 }  
 } **catch** (FileNotFoundException e) {  
 e.printStackTrace();  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 }  
}

Для записи во внешнее хранилище нужно еще указать в манифест файле разрешение:

<**uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"**/>  
<**uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"**/>

Вывод: изучили простое кодирование/декодирование информации с помощью алгоритма AES.