Challenge 03

Flag: FLAG{HungRy@ndr01D}

Tried using Android Studio and Nox to emulate wondercrypt.apk but it's not working. Open with Android Studio to look at some basic information. Since I don't read Smali, I decided to decompile it to Java code.

Installed Apk Studio which comes with JADX to allow me to decompile the apk the Java code. The AndroidManifest.xml provides information on the location of the main files. Let's browse into "org.dontlook.wondercrypt.MainActivity"

```
plication android:allowBackup="true" android:icon="@mipmap/ic_launcher" a
 <activity android:name="org.dontlook.wondercrypt.MainActivity">
     <intent-filter>
         <action android:name="android.intent.action.MAIN"/>
         <category android:name="android.intent.category.LAUNCHER"/>
     </intent-filter>
 </activity>
 <meta-data android:name="android.support.VERSION" android:value="26.1.0"
```

Some part of the code is encoded using this algorithm.

```
public String a(String str) {
    byte[] decode = Base64.decode(str, 0);
    byte[] bArr = new byte[decode.length];
    for (int i = 0; i < decode.length; i++) {
        bArr[i] = (byte) (((byte) (decode[i] - 7)) ^ 193);
    }
    return new String(bArr);
}
```

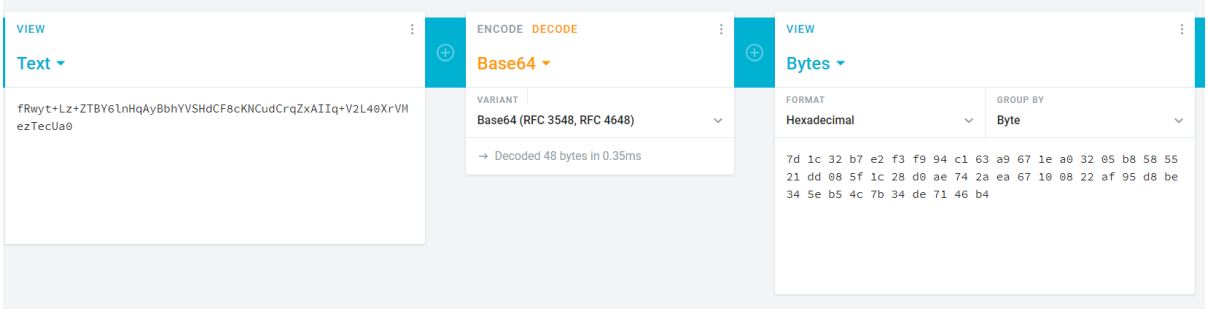
Rewriting the code in python (script in WonderCryptDecrypt.py), I managed to decode the information.

```
SHA-256
UTF-8
AES
UTF-8
AES/CBC/PKCS5Padding
WonderCrypt
Message succesfully encrypted!!
OK
>>>
```

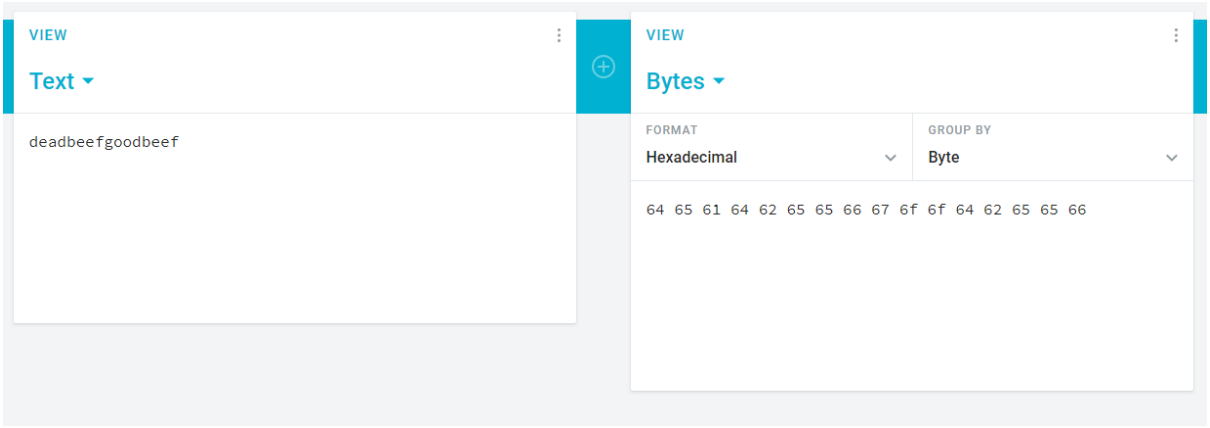
Since "setContentView((int) R.layout.activity_main)" indicates that activity_main.xml is the current view. We can find all the tag/text values in activity_main.xml. The "clean" file can be seen in WonderJava.java.

```java
import android.widget.Button;
import android.widget.TextView;
import java.security.MessageDigest;
import javax.crypto.Cipher;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;

public class MainActivity extends c {
    public String a(String str) {
        byte[] decode = Base64.decode(str, 0);
        byte[] bArr = new byte[decode.length];
        for (int i = 0; i < decode.length; i++) {
            bArr[i] = (byte) (((byte) (decode[i] - 7)) ^ 193);
        }
        return new String(bArr);
    }

    /* access modifiers changed from: protected */
    public void onCreate(Bundle bundle) {
        super.onCreate(bundle);
        setContentView((int) R.layout.activity_main);
        ((Button) findViewById(R.id.btn_encrypt)).setOnClickListener(new OnClickListener() {
            public void onClick(View view) {
                String a2 = "mylongpasswd";
                TextView textView = (TextView) MainActivity.this.findViewById(R.id.lbl_quote);
                String hexString = "deadbeef";
                String str = "deadbeefgoodbeef";
                String charSequence = "THISISTHEFLAG"; //FLAG
                try {
                    SecretKeySpec secretKeySpec = new SecretKeySpec(MessageDigest.getInstance("SHA-256").digest(a2.getBytes("UTF-8")), "AES"); //To SHA-256: "longpasswd"
                    ->9f0dc2be4178162e02a55141818a253c199ea4dfa889b3fa30f3236c78bde9e2
                    IvParameterSpec ivParameterSpec = new IvParameterSpec(str.getBytes("UTF-8")); //To Bytes(ascii to hex): "deadbeefgoodbeef" -> 64 65 61 64 62 65 65 66 67 6f 6f 64 62 65 65 66
                    Cipher instance = Cipher.getInstance("AES/CBC/PKCS5Padding");
                    instance.init(1, secretKeySpec, ivParameterSpec);
                    String encodeToString = Base64.encodeToString(instance.doFinal(charSequence.getBytes()), 0); /* fRwyt+Lz+ZTBY6lnHqAyBbhYVSHdCF8cKNCudCrqZxAIIq+V2L40XrVMezTecUa0 */
                    b b = new a(MainActivity.this).b();
                    b.setTitle("WonderCrypt");
                    b.a("Message succesfully encrypted!!");
                    b.a(-3, "OK", new DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface dialogInterface, int i) {
                            dialogInterface.dismiss();
                        }
                    });
                    b.show();
                    ((TextView) MainActivity.this.findViewById(R.id.txt_encrypted)).setText(encodeToString);
                } catch (Exception e) {
                    throw new RuntimeException(e);
                }
            }
        });
    }
```

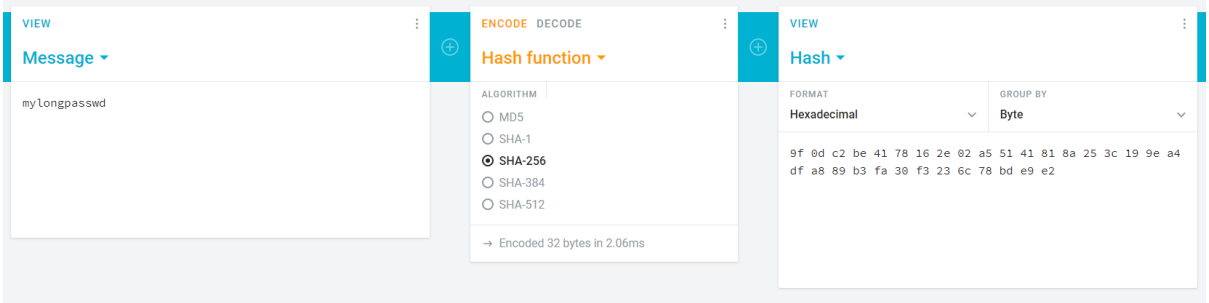
The secretKeySpec is string a2 hashed with SHA-256 and IV is string str. The encryption used is AES-CBC. Since SHA-256 is used to obtain the key, the key is 32bytes. Meaning it's an AES-256 algorithm and not AES-128. Base64 decoding the encrypted message given to hex we get:



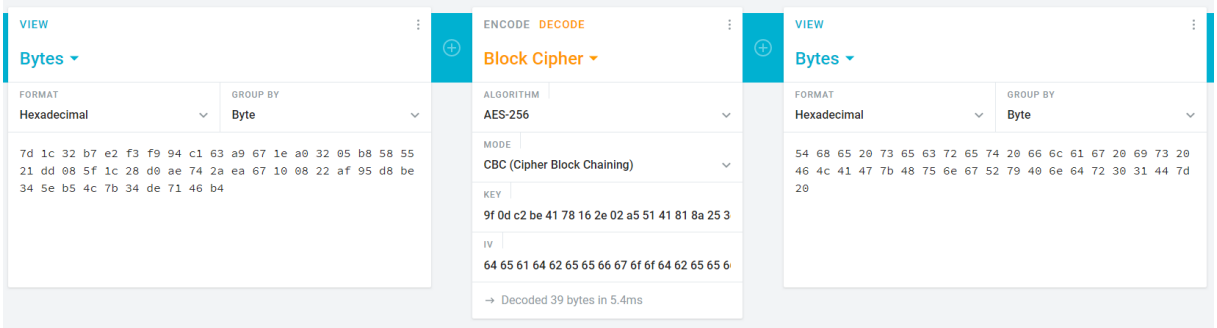
Converting the IV to hex we get:



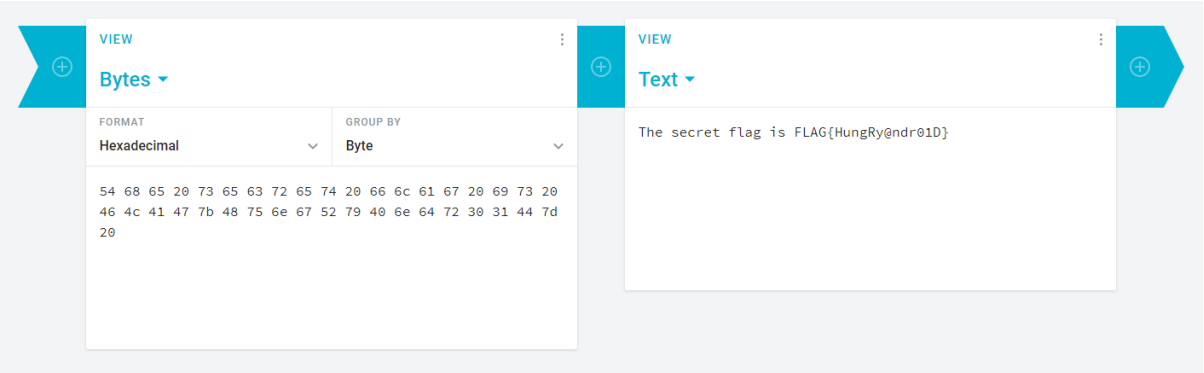
Hashing string a2 to get the key:



Decrypting the encrypted message:



Converting decrypted bytes to text.



Here we get the flag!