



Promotion 2023

Année A4

Souhail AIT LAHCEN

**Report Crypto Project
Année 4**



12/2021 - Document confidentiel

Table des matières

I	Introduction	1
II	Tink	1
III	Hash Password	1
IV	Encryption	2
V	Register and Login	2
VI	Save To Database	3
VII	Check Password	3
VIII	Project Crypto	4

Report Crypto Project 2021-2022

Souhail AIT LAHCEN

31 Décembre 2021

I Introduction

For this project, I need to create a script to allows the user to login or register with a password, username etc.. by using **Tink** and **AES-SIV**.

We can find my script, report etc.. in my repository GitHub :

<https://github.com/Souhail99/Project-Cryptography>.

Also, **as you can see in my README.md on my repository GitHub**, you don't need (in your terminal) to write **/usr/bin/python** before the name of my code but you need to put the all path , because I use this :

```
1  #!/usr/bin/env python
```

II Tink

First, we need to create or read the file that generate to have the all information about the secret key :

```
14
15 #region tink and how to use it
16 daead.register()
17 database = 'database.txt'
18
19 #Reader of the .json, necessary to read and write (if we don't this file) the secret_key with tink
20 keysetFilename = "my_keyset.json"
21
22 #if the file exist we read the file
23 if os.path.isfile(keysetFilename):
24     lecture = open(keysetFilename, "r")
25     secret_key = cleartext_keyset_handle.read(tink.JsonKeysetReader(lecture.read()))
26     lecture.close()
27
28 #else we create the file
29 else:
30     ecriture = open(keysetFilename, "w")
31     secret_key = tink.new_keyset_handle(daead.deterministic_aead_key_templates.AES256_SIV)
32     cleartext_keyset_handle.write(tink.JsonKeysetWriter(ecriture), secret_key)
33     ecriture.close()
34     lecture = open(keysetFilename, "r")
35     secret_key = cleartext_keyset_handle.read(tink.JsonKeysetReader(lecture.read()))
36     lecture.close()
37
38 daead_primitive = secret_key.primitive(daead.DeterministicAead)
39 #endregion
40
41 ## Associated
42 associated_data = b"Souhail"
43 ##
44
```

And with this, we can use the encryption correctly.

III Hash Password

So, we need also to hash the password, for this we need to choose a properly and useful mode of hashing, so for this project, I have chosen **bcrypt**.

```

44
45 #region Hash Password
46 def hash_password(pwd:str,salt=None):
47     if salt==None:
48         salt=bcrypt.gensalt()
49     else:
50         salt=salt
51     hash=bcrypt.hashpw(pwd.encode(),salt)
52     return hash,salt
53 #endregion
54
55

```

IV Encryption

For the encryption, we use **AES-SIV**, thanks to **Tink**, we can use this with a single line of code :

```

54
55
56 #region encryption
57 def encryption_machine(msg:bytes):
58     # encrypt using AES-SIV
59     ciphertext=daead_primitive.encrypt_deterministically(msg, associated_data)
60     return ciphertext
61 #endregion
62

```

V Register and Login

The user must be able to register or login. So, we need to create two functions. The first is register (**inscription in my script**), so this function ask the user to put a user-name and a password and then save them in the database. The first is login (**connexion in my script**), so this function will verify if in our database we have this user.

```

97
98 #region register
99 def inscription():
100     user=input("Your username please > ")
101     pwd=input("Your password please > ")
102     print("\n")
103     save_to_database(user,pwd)
104     print("\n")
105     print("Register finish")
106 #endregion
107
108 #region login
109 def connexion():
110     user=input("Your username please > ")
111     pwd=input("Your password please > ")
112     test,error=check_password(user, pwd)
113     if test==True and error==0:
114         print("Good password and good user, welcome :",user)
115     if test==False and error==1:
116         print("Error, this isn't the good user or pwd")
117 #endregion
118

```

VI Save To Database

After a user register its usersame and password, we need to be able to write this information in our database (**here the file is 'database.txt'**) (or create the file if we don't have this database the firstime) :

```

63
64 #region save to database
65 def save_to_database(user, pwd):
66     # use a file as a database
67     # format: user, hashed_password
68     # for example: file.write(user, hash_password(pwd))
69     hash,salt=hash_password(pwd)
70     pwdisencrypted=encryption_machine(hash)
71     pswrd = open(database, "a")
72     pswrd.write(f'{user},{pwdisencrypted.hex()}',{salt.hex()}\n')
73     pswrd.close()
74
75 #endregion
76

```

VII Check Password

I think this is the most important part of my script.

We need to be able to check in our database, after a user want to be login, if this user is in our database.

We check in every line in our database if the username exist and also if for the same line the password is the same.

After this we return the answer :

```

76
77 #region check
78 def check_password(user, pwd):
79     # read from database
80     with open(database, 'r') as f:
81         error=1
82         test=False
83         print("Welcome user :",user," we will verify if you are in our database...")
84         for line in f.readlines():
85             userdatabase,encrypteddatabase,salt= line.split(',')
86             hash,salt2=hash_password(pwd,bytes.fromhex(salt))
87             password_of_the_current_user=encryption_machine(hash)
88             # and check for authentication
89             encrypteddatabase=bytes.fromhex(encrypteddatabase)
90             if (user == userdatabase) and (encrypteddatabase == password_of_the_current_user):
91                 test=True
92                 error=0
93                 return test,error
94         return test,error
95 #endregion
96

```

VIII Project Crypto

Finally, I create a menu to be able of register, login and exit.
 If the file database doesn't exist, I ask the user to register in firstime, to create this data-
 base.
 Then, I create a region **main**, to launch the function.

```

117 #endregion
118
119 #region Project
120 def Projct Crypto():
121     c=True
122     print("Welcome to my first Crypto-Project ! ")
123     while c:
124         print("\n")
125         print("-----start-----")
126         print("\n")
127         case=input("Do you want to register (Press 1) or login (Press 2) or exit (Press 3) > ")
128         print("\n")
129         if case == "1":
130             inscription()
131         elif case == "2":
132             if os.path.isfile(database):
133                 connexion()
134             else:
135                 print("Database isn't yet created, you need to register to initialize the database for the firstime")
136         elif case == "3":
137             c=False
138             print("Au revoir !")
139         else:
140             print("Something, you don't press an available numbers, please restart")
141         print("-----end-----")
142     #endregion
143
144 #region main
145 if __name__ == '__main__':
146     Projct_Crypto()
147
148

```