(Applied) Cryptography

Tutorial #0

Manuel Barbosa (mbb@fc.up.pt) Rogério Reis (rvr@fc.up.pt)

MSI/MCC/MERSI – 2022/2023

Warming up to sage

1 - Use the Sage web interface to interpret the following script:

```
caesar = AffineCryptosystem(AlphabeticStrings())
a, b = (1, 3)
EncObject = caesar.encoding("AttackAtDawn")
Ciphertext = caesar.enciphering(a, b, EncObject)
Plaintext = caesar.deciphering(a, b, Ciphertext)
(EncObject, Ciphertext, Plaintext)
```

You can start the web interface from the console using:

```
$ sage --notebook
```

and then choosing Jupiter and creating a new workbook.

2 - Sage scrypts can also be run directly in the console by creating a Python file with the following contents

```
from sage.all import *
caesar = AffineCryptosystem(AlphabeticStrings())
a, b = (1, 3)
EncObject = caesar.encoding("AttackAtDawn")
print (EncObject)
Ciphertext = caesar.enciphering(a, b, EncObject)
print (Ciphertext)
Plaintext = caesar.deciphering(a, b, Ciphertext)
print (Plaintext)
```

And then running:

```
$ sage -python <file>
```

3 - We can use the python cryptography library (`pyca/cryptography`) to compute one block with AES.

```
from os import urandom
from binascii import hexlify
from cryptography.hazmat.primitives.ciphers import Cipher, algorithms, modes
key = urandom(16)
iv = urandom(16)
cipher = Cipher(algorithms.AES(key), modes.ECB())
encryptor = cipher.encryptor()
# What happens if you don't pass 16-byte input?
ct = encryptor.update(b"attack at dawn!!") + encryptor.finalize()
print(hexlify(key))
cphFile = open("ciphertext.bin", "wb")
```

```
cphFile.write(ct)
```

You may need to use `pip3` to install the latest version of `cryptography`.

Running the script (note that `sage` is not used) can be done using

```
$ python3 <file>
```

4 - We can uses openSSL in the console to invert the block.

```
openssl enc -aes-128-ecb -nopad -d -K <key_in_hex> -in ciphertext.bin
```

Can you check if inversion was correct? Explain the options used in this command.