CENG471 – Homework 2

Deadline: 02.01.2023 23:55

Please write 2 Python scripts named **alice.py**, which represents the sender, and **bob.py** which represents the receiver. These scripts take the inputs in the form of command-line arguments (For this, simply use **sys.argv**[1] or rely on the more advanced **argparse**[2] module from the standard library). Both of these scripts can fulfill two tasks: Diffie-Hellman Key Exchange (dhke) and Data Encryption Standard (des). In dhke part, please do not depend on any package outside the standard library. In des part, please use **DES** from **Crypto.Cipher** package[3]. You can install this package using the following command: **pip install pycryptodome**. Run the algorithm in the Electronic CodeBook (**ECB**) mode.

The scripts shall be implemented in accordance with the following specification:

alice.py

- dhke mode
    - If **a**, **B** and **p** are given (They can be given in any order; for example **p**, **B** and **a**):
        i. Calculates **s**.
    - Else if **p** and **g** are given (They can be given in the reversed order):
        i. Checks whether **p** is a prime number (If not, the program is terminated here).
        ii. Checks whether **g** is a primitive root modulo **p** (If not, the program is terminated here).
        iii. Randomly generates **a**.
        iv. Calculates **A**.
    - Otherwise:
        i. Displays an error (A message such as "Please provide p and g to initialize the key exchange or provide a, B and p to calculate the key.").
- des mode
    - If **p** (plaintext; different than **p** in the dhke mode) and **k** (key; corresponds to **s** in the dhke mode) are given (They can be given in the reversed order as well):
        i. If the plaintext is not valid, adds the minimum number of trailing spaces to make it valid.
        ii. If the key is not valid, adds the minimum number of leading zeros to make it valid.
        iii. Calculates **ciphertext** (Both the bytes and the corresponding readable text. Refer to the examples below.).
    - Otherwise:
        i. Displays an error (A message such as "Please provide the plaintext and a key.").

bob.py is very similar to **alice.py**. In dhke mode, Bob produces **b** and **B** instead of **a** and **A**, and consumes **b** and **A** instead of **a** and **B**. In des mode, Bob consumes **c** (ciphertext) instead of **p** (plaintext) and produces plaintext instead of ciphertext. See the given examples below.

---

[1] See https://docs.python.org/3/library/sys.html#sys.argv.
[2] See https://docs.python.org/3/library/argparse.html.
[3] See https://pycryptodome.readthedocs.io/en/latest/src/cipher/classic.html.

A successful scenario is given below. It is okay if the format of your outputs slightly differs from these. However make sure that you display all of the calculated values (mind the red color). Note that dhke and des are "positional arguments".

Initialization of the key exchange

```
ersin@ersin-dell:~$ python3 alice.py dhke -p 23 -g 5
p = 23 OK (This is a prime number.)
g = 5 OK (This is a primitive root modulo 23.)
Alice and Bob publicly agree on the values of p and g.
However, it is advised to use any pair of p and g only once.
a = 4 (This must be kept secret.)
A = 4 (This can be sent to Bob.)

ersin@ersin-dell:~$ python3 bob.py dhke -g 5 -p 23
p = 23 OK (This is a prime number.)
g = 5 OK (This is a primitive root modulo 23.)
Alice and Bob publicly agree on the values of p and g.
However, it is advised to use any pair of p and g only once.
b = 3 (This must be kept secret.)
B = 10 (This can be sent to Alice.)
```

Finalization of the key exchange

```
ersin@ersin-dell:~$ python3 alice.py dhke -a 4 -B 10 -p 23
s = 18
(This must be kept secret. However, Bob should be able to calculate this as well.)

ersin@ersin-dell:~$ python3 bob.py dhke -b 3 -A 4 -p 23
s = 18
(This must be kept secret. However, Alice should be able to calculate this as well.)
```

Encryption and decryption of a single message

```
ersin@ersin-dell:~$ python3 alice.py des -p 'This is the secret message.' -k 18
Raw ciphertext (Normally this is sent to Bob over a network):
b"\xd55\xc3d\x9d\xf4\x11y\x10\\\x81\xd1\x88n\xdc\xc2\xbb\xed2R\x81'\x8a\xf4\x91g\x90\x18\xa49\\\xbd"
Readable ciphertext (For the sake of simplicity, we will send this to Bob. This can also be used if we use pen and paper to deliver the message.):
1TXDZJ30EXkQXIHRiG7cwrvtMlKBJ4r0kWeQGKQ5XL0=

ersin@ersin-dell:~$ python3 bob.py des -c '1TXDZJ30EXkQXIHRiG7cwrvtMlKBJ4r0kWeQGKQ5XL0='
    -k 18
Decrypted plaintext:
This is the secret message.
```

Report

In your report, please answer the questions below:

1. Imagine the following scenario for the real-world (not the digital world!): I want to send a special gift to a friend of mine who lives in another city. For this I will rely on the standard shipping service. However, I do not want any other person to see the gift. To keep it private, I put it in an opaque box and put a lock on it for which only I have the key. (Even my friend does not have the key, so she cannot unlock it!) Then I write a note on the box for her and ship the box. The note is "I am sorry but you can't open the box yet. Please put an extra lock and ship the box back to me". Several days later I get the box with 2 locks on it. What should I do now?
2. In the given above scenario, what can go wrong in terms of security? (In your explanation, please stick to our concrete scenario.)
3. How is Diffie-Hellman key exchange able to provide security despite being performed over an unsecured communication channel?
4. Is DES considered state-of-the-art or has it been largely replaced by more modern encryption algorithms? Why is this the case?

Hints

- In **alice.py**, you can apply Base64 encoding (https://docs.python.org/3/library/base64.html#base64.b64encode) in order to convert the "raw" ciphertext to the "readable" ciphertext. In **bob.py**, you need to apply the inverse function (https://docs.python.org/3/library/base64.html#base64.b64decode).
- At any point, if you see an output like **b'abc123'** (instead of **abc123**), this means that the data type is **bytes** instead of **str**. If the value is supposed to be a readable text, you simply need to decode the bytes in order to obtain an **str**. Write, for example, **print(x.decode())** instead of **print(x)**.

Grading

- If you lack the time to complete the homework, I suggest you to deliver complete solutions to some of the subtasks rather than incomplete solutions to all of the subtasks. This is also what will usually be expected from you as an engineer in the future. Test your scripts with the example inputs given above (**Copy and paste** the given commands, for example alice.py dhke -p 23 -g 5, to your terminal window in order to make sure that you did not make a typo. This interface is not optional but **mandatory**.). If the scripts cannot even handle the given examples (e.g. existence of syntax errors, runtime errors, or logical errors (i.e., incorrect outputs)), then they are useless in practice.
- A perfect solution to this homework would include a simple and short report, and clean code (which should be commented only if needed).
- This is how the subtasks will be graded:
    - alice.py dhke mode + des mode: **20 pts** + **20 pts**
    - bob.py dhke mode + des mode: **20 pts** + **20 pts**
    - Report: **5 pts** + **5 pts** + **5 pts** + **5 pts**

Rules

- You are allowed to do the homework in pairs. However, once formed and started to discuss the homework, you cannot change your team (Otherwise it is likely that I will spot partially duplicated solutions in two different submissions and will consider this cheating).
- Do not cheat. Honesty is its own reward.
- Contact me (Ersin Çine) for any questions via email (ersincine@iyte.edu.tr) or Teams (ERSIN CINE).
- Submit a single file named **ceng471_hw2_<studentno1>_<studentno2>.zip** where <studentno1> and <studentno2> are variables. If you did the homework alone, then you can submit a file named **ceng471_hw2_<studentno>.zip** where <studentno> is your student no.
- This compressed file should include 3 files only: **alice.py**, **bob.py, report.pdf**. Some of these files may be missing if you did not complete the homework. However please do not add an extra file.
- State your student number at the beginning of all files. I prefer to not see your names while grading.