Homework Assignment 3
Information Security - Theory Vs. Reality
0368-4474
Submit by: 14th of July at 23:59

1 Submission Instructions

The HW assignments will include writing code and answering written questions. The submission will include the requested code file and a PDF file with the written answers. The PDF can be a scan of a *clearly handwritten* page, but typing the answers is *strongly* encouraged.

If not written otherwise, all the files should be at the root of the zip (meaning that they are not inside an inner folder).

The zip file should be of type .zip and not .rar.

Unless stated otherwise, all assignments must be written in Python 3.8.

We set up a docker environment with python3 all required packages to allow you to run and test your code. You can use other developing environments for writing your code, but you need to make sure it runs on the python version installed in the docker before submitting it.

Code that fails to run inside the docker environment will not be graded!

Instruction for running the python3 environment:

1. Login to nova or any cs server of your choice.
2. Run the following 2 commands to start the docker:
   `export UDOCKER_DIR="/specific/netapp5_2/eyalron1/SecCourseDocker"`
   `udocker run --bindhome SecDock`
3. If all works well, you should now be running inside the docker, with the docker's home directory mapped to your own home directory. Note that

you can only save files inside your home directory.

4. You can run python with the command:
   `python3`

2 Coding Assignment 3

For your third coding assignment, you are requested to implement two chosen-plaintext attacks on PKCS #1.

2.1 Bleichenbacher Attack on PKCS #1 v1.5

The first is an attack on PKCS #1 v1.5. For the purposes of this attack, you are provided with an oracle that given a ciphertext c tells whether the decryption of c conforms with the encryption standard. The attack accepts as input a bytestring c of size k, where c containts an integer that is smaller than N and k is the size of N in bytes. It returns a bytestring m of size k such that:

$$m^e \equiv c \, mod \, N$$

For this attack, please implement the missing parts (denoted by "?") in the file `bleichenbacher.py`. Keep in mind that you are expected to support inputs c that do not necessarily conform with the standard.

For a detailed explanation of the attack please see [Bleichenbacher(1998)].

2.2 Manger Attack on PKCS #1 OAEP

The second attack is an attack on PKCS #1 OAEP. Here you are provided with an oracle that given a ciphertext c returns `True` if

$$c^d < B \, mod \, N$$

and `False` otherwise. B is the (exclusive) upper bound of a conforming plaintext - in the case of OAEP $B = 2^{8(k-1)}$. This attack also accepts c and returns m such that:

$$m^e \equiv c \, mod \, N$$

For this attack, please implement the missing parts (denoted by "?") in the file `manger.py`. Here you may assume that c is a conforming ciphertext, meaning $c^d < B \, mod \, N$.

For a detailed explanation of the attack please see [Manger(2001)].

2.3 Implementing the attacks

Both attacks include a check to test if they have recovered the correct value. They return said value in the case of success and `None` in the case of failure. When properly implemented, they should never return `None`.

In addition, it is recommended to use the provided functions `divceil` and `divfloor` when dividing large numbers in order to avoid floating-point errors.

References

[Bleichenbacher(1998)] Daniel Bleichenbacher. Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. In *CRYPTO*, 1998.

[Manger(2001)] James Manger. A chosen ciphertext attack on RSA optimal asymmetric encryption padding (OAEP) as standardized in PKCS #1 v2.0. In *CRYPTO*, 2001.