Introduction to Security (WS2017/18) Prof. Matteo Maffei

Project 3 – Cryptography

Due *31.01.2018* TU Wien

In this project you will get hands-on experience in different kinds of web attacks. Overall, this project has a maximum of **10 points**.

Before you start with the project, carefully read this document!

Setting up Your System for the Project

This project uses the same setup as Project 1 - System Security. The virtual machine contains a folder `/home/security/crypto` with the files relevant for this assignment.

General Submission Preparation

For each of the following attacks your task is twofold:

Part 1: Understand the vulnerability and develop an attack mentally. Collect the ideas and strategies that you have developed for the mental part in a `.tex` file (please use the template provided in TUWEL). For each attack, we expect the file to contain the following information (written in your own words):

- Why is the scheme vulnerable?
- Which strategy do you follow in order to exploit the vulnerability, that is, which high-level steps do you take in order to implement the attack?

In the `.tex` file, do not forget to insert your references if you have consulted material outside of the lecture and cite your sources if you have discussed with other students. Before you submit, compile the file, such that you send us the `.pdf` version only and not the `.tex` version. The naming conventions are detailed in the submission instructions at the end of this document.

Part 2: Implement the attack and exploit the vulnerability. We expect, the **well-commented** source code of your attack implementation and precise compilation (if necessary) and execution intstructions.

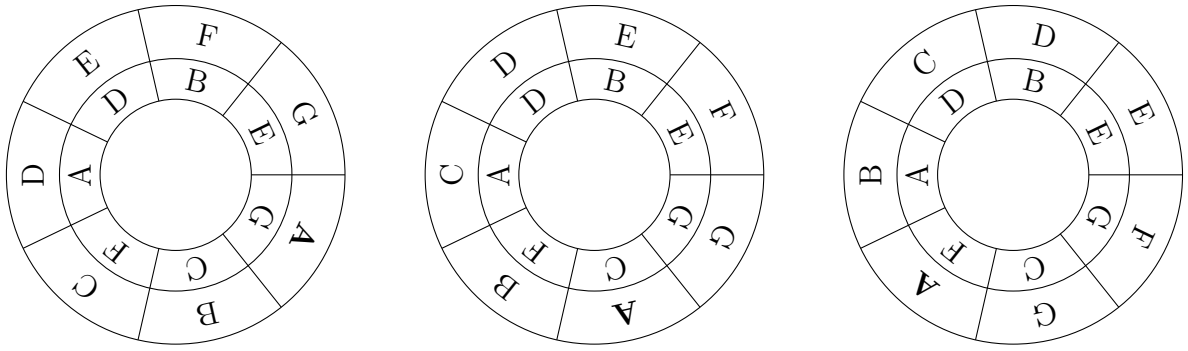
Remark Please keep in mind that the submission deadlines are strict. If you miss a deadline your submission will be graded with 0 points!

Grading

You will be graded based on your answers in the `.pdf` file and the quality of your attack. Full points will be given in case the implemented attack works and the explanation demonstrates that you have understood the vulnerability and your strategy coincides with your attack. In case the implementation does not work or you do not give any explanation, you will receive 0 points. If you give unsatisfactory explanations, we subtract partial points.

General Hints

- The grader will use a fresh virtual machine to grade your submission. If you perform any modifications to the system (e.g. install additional software, modify files, etc.) make sure to re-test your submission in a fresh virtual machine.
- Use a shared folder to export the your exact attack files from the virtual machine.
- Every piece of code that needs to be copied and pasted, has to be in a plain text file. Copying from pdfs is very likely to introduce errors due to encoding, spacings, etc, so it is in your own interest to avoid this.



a Configurartion for first character

b Configurartion for second character

c Configurartion for third character

Figure 1: Simple example of the encryption device

Cryptography

Encryption Scheme

Consider the following encryption scheme: The encrypter and decrypter each have a device with two discs. The outer disc contains the publicly known alphabet and the inner disc contains a secret permutation of this alphabet which is the same for encrypter and decrypter; this is the shared symmetric key. A character is encrypted by locating it on the outer disc and replacing it with the corresponding character on the inner disc. After every character, the outer disc is turned clockwise by one step. Decryption is performed analoguously by substituting the character on the inner disc by the corresponding character on the outer disc.

As a simple example we show the encryption of the word "BEE" using the alphabet ['A','B','C','D','E','F','G'] and the inner disk ['G','C',F','A','D','B','E']:

- The first letter "B" is substituted by "C" according to the configuration in Figure 1a. The outer disk is turned clockwise one step to reach the configuration in Figure 1b.
- The second letter "E" is substituted by "B" according to the configuration in Figure 1b. The outer disk is turned clockwise one step to reach the configuration in Figure 1c.
- The third letter "E" is substituted by "E" according to the configuration in Figure 1c.

The resulting ciphertext is hence "CBE".

To decrypt, we bring the device back into the initial position (Figure 1a) and perform the same steps as for encryption, but going from the inner disc to the outer disc in the substitution step.

Setting

In the virtual machine there is a directory `/home/security/crypto`. The directory contains a reference implementation of the presented encryption scheme `crypto` and its source code `crypto.ml`, as well as two files file `secret` and `cipher` that contain two encryptions with the same secret key.

There a two important facts for the reference implementation:

- The alphabet used is {'A';'B';'C';'D';'E';'F';'G';'H';'I';'J';'K'; 'L';'M';'N';'O';'P';'Q';'R';'S';'T'; 'U';'V';'W';'X';'Y';'Z'; '0';'1';'2';'3';'4';'5';'6';'7';'8';'9';'.';',';'!';'?';'(';')';'[';']'} (44 characters)

- Upon input of a plaintext or ciphertext, first all lower case letters are replaced by the corresponding uppercase letter, then all characters that are not part of the alphabet are removed (including whitespaces).

Your task

Your overall goal is to decrypt the file `secret`. To this aim, perform the following steps:

- Find out to which extent the scheme is vulnerable to frequency analysis.
- Implement an algorithm that derives the most probable key for a given encryption of an English text.
- Use your algorithm on the file `cipher` to derive the secret key
- Use the derived key to decrypt the file `secret`

For your submission please include:

- The explanation of the vulnerability of the crypto scheme
- The **well-commented** source code of your key extraction algorithm. You can fill in the missing function in `crypto.ml` or write your own program in Python, C, Perl, PHP, Java or Javascript.
- The derived secret key
- The decryption of the file `secret`

To grade your submission, the grader will:

- compile your programm (if necessary), following your compilation instructions
- run your program, following your execution instructions, to obtain the key
- decrypt the file `secret` with the obtained key, using the reference implementation
- check that the correct decryption result is produced

Please make sure that all these steps work and that your explanationas are sufficient to execute these steps.

Hints

- You can use
  `./encrypt.sh` *key_file plaintex_file result_file* and
  `./decrypt.sh` *key_file ciphertext_file result_file*
  to test the algorithm. There is an example key file in the directory `crypto`.
- If you found the correct key, both files `cipher` and `secret` decrypt to meaningful texts (although they are a bit unreadable due capitalization and missing whitespaces).
- After every full turn of the outer wheel, the initial configuration is reached again.
- The letter 'e' is the most commonly used letter in the English language.

Submission Instructions

Please submit your solution by ***January 31st*** through TUWEL.
Upload a single `.zip` archive containing the following files:

- `report3.pdf`: The pdf containing your attack description and execution instructions
- **well-commented** source code files