INF143A 22V Applied Cryptography
Third mandatory assignment

Nikolay Kaleyski

April 11, 2022

1 General information

- **Deadline:** The assignment is due on May 3, 2022.
- **Submission:** A copy of the solution should be uploaded to mitt, under the "Assignments" tab.
- **Score:** The mandatory assignment accounts for 15% of the final grade.
- **Collaboration:** You can freely discuss the assignment with each other, but the solutions must be prepared individually; plagiarism will result in all involved parties failing the assignment.

NB: In some of the problems, you are asked to encrypt or hash a file. This means that you have to consider the file in **binary**, i.e. as a sequence of 0's and 1's, and apply the operations to this stream of 0's and 1's. If padding is necessary to reach a certain block size, you should **pad with zeros on the right**.

2 Problems

Problem 1. ***35 %*** *Consider a very simple block cipher implemented using the Gold function x^3 on 32 bits: if the 32-bit plaintext is P and the 32-bit key is K, then the ciphertext is $K^3 \oplus P$, where K^3 is computed in the finite field with primitive polynomial $x^{32} + x^{15} + x^9 + x^7 + x^4 + x^3 + 1$, and $\oplus$ denotes the XOR operation. An implementation of this cipher is available in gold.py.*

Write an extension of this block cipher that can be used on an input of arbitrary length, i.e. not only 32 bits. If the input is not a multiple of 32, pad it with 0's on the right. Implement the following modes of operation:

- *ECB;*
- *CBC;*
- *OFB.*

Encrypt the plaintext given in gold_plaintext.in using each of the three modes of operation. Use $K = 0101\ldots01$ (32 alternating zeros and ones) as the key.

Problem 2. ***25 %*** *Use the same block cipher described above with the Matyas-Meyer-Oseas construction to construct a hash function; you can use a vector of all 1's for the first round key h_0. What is the output size of this hash function? How many hashes would an attacker have to compute in order to find a pair of inputs with the same hash (in other words, what would the complexity of the so-called birthday attack be)? Hash the plaintext in gold_plaintext.in using the resulting hash function.*

Problem 3. ***25 %*** *Recall that a cryptographically secure hash function h with output length l can be used to construct a block cipher as follows:*

- *split the message (file) to be encoded into blocks $M_1, M_2, \ldots, M_K$ of l bits;*
- *pad M_K if necessary;*
- *choose an initialization vector IV and set $M_0 = IV$;*
- *to encrypt block M_i, compute $h(K||M_{i-1})$ and XOR it with M_i (where K is the secret key).*

In this exercise, you will use the SHA-256 hash function to create an encryption algorithm using the above approach. Implementations of SHA-256 should be easily accessible on all operating systems, e.g. the sha256sum command should be available on most Unix distributions. Note that the output of SHA-256 is always 256 bits long.

Write a program which accepts a 256-bit initialization vector and a secret key of arbitrary length, and encrypts or decrypts a given file using the algorithm described above. Taking $IV = 111\ldots1$ and $K = 0101\ldots01$, use the program to encrypt the file gold_plaintext.in.

Problem 4. ***15 %*** *If you visit www.uib.no, you will observe that the connection is using the HTTPS protocol. As we have discussed on the lecture, this means that UiB identifies itself using a digital signature which can be verified using UiB's public key; and the fact that what is shown as UiB's public key really is UiB's public key is guaranteed by the signature of a certification authority, whose public key is guaranteed by an even higher level authority, etc.*

For this problem, use the tools available in your browser to investigate the chain of trust guaranteeing the security of the HTTPS connection to UiB. More precisely, list all all entities involved in the chain of trust (starting from UiB), and for each of them, list:

- *what digital signature algorithm was used to authenticate them;*
- *what their public key is;*
- *what are the public parameters used in the algorithm (e.g. the prime number p for RSA).*