IDATT2503 - Cryptography 2023 - Extra assignment/extra training

# Exercise 1

**What does Kerckhoffs’s principle (the one we have covered) say? Why is it an important principle?**

Kerkhoff’s principle says that a cryptographic system is secure even if everything about the system, except the key, is publicly available. The reason this is important is because this indirectly implies that acquiring knowledge of the algorithm used should not jeopardize the security of the encrypted messages. Identifying algorithms used for encryption is possible with careful data analysis, and if knowing the algorithm was a security risk, the first thing one must do to break the encryption is to identify the algorithm used (which would make cracking the message considerably easier). Another risk of keeping algorithms a secret is that it won’t undergo public testing and scrutiny, meaning that flaws in the algorithm or implementation stay unfixed. Perhaps the greatest reason is that changing key is much easier than algorithm if a part of the system is compromised.

Generally, sticking to Kerckhoffs’s principle is advised. Avoiding total secrecy, up to this day, seems to be the best way of ensuring that a cryptographic system is secure.

**What in general is a good approach to develop a secure cryptographic algorithm? Give an example of ”good” and ”less good” examples in this respect.**

As stated in the previous question, by minimizing secrecy (except for the key) and by rolling out improvements to rid the algorithm of flaws. It should also be noted that the algorithm must be based on a well understood mathematical foundation. This implies something that would be considered hard to crack without the key, relative to modern and possibly future computational power.

Good examples can be: AES (It was the winner of an open competition that lasted 5 years), SHA-256, ECC (Elliptic curve cryptography).

Less good examples: **DES** (Data Encryption Standard) which was once a widely used symmetric-key algorithm which later was found to be unsecure because 56-bit key size could not withstand modern brute-force attacks. Another one is **MD5** (Message-Digest Algorithm 5), which was a widely used hash function that produced a 128-bit hash value. Even by the mid-1990s weaknesses were being reported in the compression functions which could lead to collisions. This means that it does not guarantee integrity because different data could generate the same hash. WEP (Wired Equivalent Privacy) is also an example because it uses static encryption key (the same key for all devices), which means once a key is cracked all network traffic can be decrypted until the key is changed. It also has other flaws in its implementation such as a weak encryption algorithm (RC4 and its initialization) and no security mechanism for replay attacks.

# Exercise 2

***Explain in general what ”mode of operations” are for block ciphers.***

Modes of operations are used to increase the security of a block cipher, by extending diffusion between blocks.

In cryptography, a block cipher mode of operation is an algorithm that uses a block cipher to provide information security such as confidentiality or authenticity.[1] A block cipher by itself is only suitable for the secure cryptographic transformation (encryption or decryption) of one fixed-length group of bits called a block.[2] A mode of operation describes how to repeatedly apply a cipher's single-block operation to securely transform amounts of data larger than a block.[3][4][5]**ECB**: Electronic code book (ECB) mode encrypts each block independently. It’s simple but vulnerable to pattern attacks because the same plaintext blocks result in the same ciphertext blocks. This means information can be leaked even if some parts are diffused.

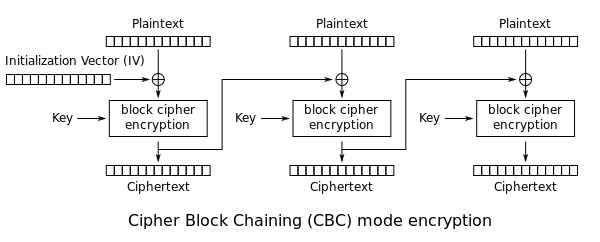
A diagram of a computer code book

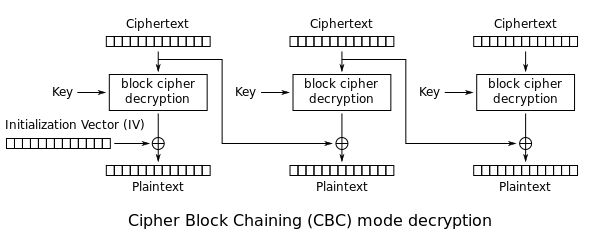
Description automatically generated

A diagram of a blockchain codebook

Description automatically generated

**CBC**: Here, each block of plaintext is XOR-ed with previous ciphertext block before encryption. This mode uses an IV (initial vector) for the first block to avoid repetitions (same message being encrypted to the same cypher block). IV does not need to be kept a secret, since its main purpose is to hide repetition among encryptions that would appear just by looking at the first block. If Alice forgets the value for IV but has encrypted message C and key K, the only message M she can recover is every except .





**CTR (counter):** The core concept and the goal are the same as CBC. This mode introduces a nonce and a counter, where the nonce serves the same role as an IV and should be an unpredictable one time value. As the IV the nonce does not need to be kept secret as its purpose is to avoid the same plaintext being encrypted to the same cypher text. The nonce introduces randomness, and the counter serves to distinguish the blocks.

A diagram of a computer code

Description automatically generated

A diagram of a computer

Description automatically generated

***What is the basic difference between CBC and CTR modes?***

CTR mode encrypts the nonce with a counter while CBC mode encrypts the text. The xor-ing happens last in CTR modes when encrypting and decrypting while in CBC the xor-ing happens mfirst when encrypting and last when decrypting.

# Exercise 3

***The security of cryptographic hash functions can be described as the hardness of three ”problems”. What are these?***

Pre-image resistance: This is the hardness of finding a message that corresponds to a given hash value. In more technical terms, given a hash output h, it should be computationally infeasible to find any input x such that hash(x)=h. This ensures that an attacker cannot deduce the original input from the hash value.

Second pre-image resistance: This property means that given x with hash y it should be computationally infeasible to find x’ with same hash y. This helps prevent creating a hash with the same data hash as the original. (attacker sticks with a given input and tries to find another input)

Collision resistance: Given x and x’ it should be difficult to find x and x’ that produce the same hash. (attacker is free to choose any pair of inputs)

***You are given the following attempt at a hash function:***

***Split the cipher into equal length blocks, bitwise XOR all the blocks. Pad the message with the pattern 010101... so that the message is equal to a whole multiple of the block length. Assess the security of this hash function considering the criteria in part a)***

For first image resistance: Having the same message length as block length means that there are no other blocks to xor with meaning that input is equal to output for the hash which means first pre image resistance problem is solved. 1010 would hash to 1010 for example

For second pre image resistance

**A screenshot of a computer program

Description automatically generated**

For collision resistance: if you solve secondary preimage resistance you can solve collision resistance the same way.

# Calculating cbc-mac

A white square with black text

Description automatically generated