Assignment Series #A3 Modern ciphers

1. Introduction

> This challenge will provide an intro into modern cryptography, basic concepts and differences between stream and block ciphers. Follow the questions to direct your self-study. Check the theory module or consult Wikipedia if you happen to struggle answering a question.

1. Alice and Bob living in 2020 make use of modern crypto systems. You are in the role of Mallory and got hold of some exchanged ciphertexts.
   Can you run a brute-force attack (cryptographers call this exhaustive key search) against the ciphertexts if
   - you know the plaintext is random?
   - you know part of the plaintext but not the algorithm used?
   - you know part of the plaintext and the algorithm used?
2. You have learned that we must assume an attacker to know the algorithm in the very detail. Thus, what will the secrecy of a cipher text ultimately rely on? Who described this principle first?
3. Symmetric ciphers can be categorized in either stream or block ciphers. Could you
   - briefely explain the difference of the two approaches
   - name algortihms as an example for both
   - discuss speed of stream and block ciphers
   - describe the differences in error propagation
   - explain the term "message dependency"

2. Answers

1. My sponteous answer to all three questions would be `no` if there are no `failure implementations` of the algorithm itself are known or any `side-channel attacks` possible. If you take forexample `AES` there is no way to break it with normal CPU power.

Looking up for the term `exhaustive key search` we find the following approaches:

- Ciphertext-only:
  The cryptanalyst has access only to a collection of ciphertexts or codetexts.
- Known-plaintext: The attacker has a set of ciphertexts to which he knows the corresponding plaintext.
- Chosen-plaintext (chosen-ciphertext):
  The attacker can obtain the ciphertexts (plaintexts) corresponding to an arbitrary set of plaintexts (ciphertexts) of his own choosing.
- Adaptive chosen-plaintext:
  Like a chosen-plaintext attack, except the attacker can choose subsequent plaintexts based on information learned from previous encryptions. Similarly Adaptive chosen ciphertext attack.
- Related-key attack: Like a chosen-plaintext attack, except the attacker can obtain ciphertexts encrypted under two different keys. The keys are unknown, but the relationship between them is known; for example, two keys that differ in the one bit.

2. It's the Kerckhoff's prinicple
   https://en.wikipedia.org/wiki/Kerckhoffs%27s_principle

3. The main difference between Block cipher and Stream cipher is that block cipher converts Converts the plain text into cipher text by taking plain text's block at a time. While stream cipher Converts the plain text into cipher text by taking 1 byte of plain text at a time.

   `RC4` is an example of a modern symmetric-`key stream cipher`.
   `RC2`, `RC5`, and `RC6` are symmetric-`key block ciphers`.

   The main advantages of pure stream ciphers over block ciphers are usually 1) raw speed, and 2) code / circuit size. In some cases block ciphers can beat stream ciphers in speed. Please follow the discussion here:
   https://crypto.stackexchange.com/questions/62095/stream-cipher-and-block-one-which-one-is-faster-to-encrypt-decrypt

   `Electronic Codebook Mode (ECB)`

   > This is the simplest mode of operation. Each plaintext block is encrypted/decrypted individually. Resulting in a cipher text that is the same for each given plaintext and key. In this mode, no error propagation occurs, as all blocks are independant. In this mode an opponent can change the order of the ciphertext blocks or even remove them, replay them without causing failure of decryption.

   This mode is to be used with extreme caution, as it is very vulnerable to a whole host of attacks.

   `Cipher Feedback Mode (CFB)`

   > This mode allows transmission of messages of less bits than the blocksize (for example used for interactive terminal sessions). In other words, it allows the DES block cipher to be used as a stream cipher. It has error propagation properties similar to CBC.

   > CFB works by encrypting an initialization vector to make an output block, then exclusive ORs consecutive bits of the output block with consecutive bits of plaintext to make the ciphertext. Once one block's worth of ciphertext is produced, it becomes the input to the block cipher to make another output vector, and the process repeats itself.

   `Message dependancy` means how adjacent plaintext blocks affect encryption of a plaintext block.