CAA 24-25

Exercise Sheet on Symmetric Encryption #2

1 CTR

In this exercise, we are going to look more carefully at the CTR mode of operation.

1. If AES-256 is the underlying block cipher and if the nonce is 118-bit long, what is the maximal size a plaintext can have? Justify.
2. We decide to use a nonce of 8 bits (still with AES-256). Which new constraint do we have in our system? Be as precise as possible. In particular, this constraint might be different based on the way we choose the nonce.
3. A bank comes to you and wants some help with CTR. They are using 3 key 3-DES (key size: 168 bits, block size 64 bits). Here is their usage scenario:
   - Changing 3-DES is **not** an option.
   - All transaction are sent using the same key.
   - They send 2^{30} transactions per year.
   - A transaction is at most 2^{26} bits long.
   - The symmetric key is changed every year.

   They are wondering what nonce size they should use. What do you tell them? Justify.

2 IoT Scenarios

You are going to analyse **two different** IoT scenarios in which you will have to make algorithmic choices. To simplify, we will only consider the following five algorithms:

- AES-ECB
- AES-CBC
- AES-CTR
- AES-GCM
- Chacha20-Poly1305

1. The IoT device sends 256-bit messages to a recipient device. The ciphertext (including tags, IV, ...) should have **the same size** as the plaintext due to physical constraints. We are considering **only passive adversaries**. Devices do not have any memory except for storing the symmetric key. For each of the five algorithms, justify if it can be used in this scenario or not. **You can also slightly modify them.** What is your final proposition? Be precise and analyse the security of your answer.

2. The IoT device has to send **every hour during a year a 8-bit message** that gives instructions to another device. Each bit of the message is mapped to a particular behavior of the recipient (ex: turn the green light on). The emitting and receiving devices both have a 16-bit memory that you can use as you wish. They both also have enough additional space to store a symmetric key. Physical constraints do not allow you to send more than 64 bits per hour. We want to protect against an **active** adversary. The messages also have to stay confidential. For each of the five algorithms, justify if it can be used in this scenario or not. **You can also slightly modify them.** What is your final proposition? Be precise, justify your answer and analyse its security.

3 Authenticated Encryption with Libsodium

Read about the libsodium library: `https://libsodium.gitbook.io/doc/`. The goal of this exercise will be to use libsodium to encrypt and authenticate a message.

1. What are the advantages of libsodium? What are the drawbacks?

2. Which algorithms are used in libsodium to do authenticated encryption? Check the AEAD section and the Authenticated Encryption section.

3. What is the difference between combined and detached mode?

4. Implement a program that takes the command line argument, encrypts it with a random key and a random nonce, decrypts and prints the result. Once this is done, flip a bit of the ciphertext and try to decrypt it again. You can use the default authenticated encryption algorithm.

5. Do the same with AES-GCM and flip a bit in the authenticated data. For this, you can allow your program to take two command line arguments: one for the plaintext and one for the authenticated data.