Exercise Sheet 10

Cryptography and Security 2025

Exercise 1 CFB-MAC

In this problem, we study a MAC scheme based on the CFB encryption mode. We consider a block cipher $\mathsf{E} : \{0,1\}^{64} \times \{0,1\}^{64} \to \{0,1\}^{64}$, where $\mathsf{E}_k(x) = \mathsf{E}(k,x)$ denotes the encryption of the plaintext x under the key k. The CFB-MAC of a given message $m \in \{0,1\}^*$ with the key k is obtained by first encrypting m with E_k using the CFB encryption mode and then combining the output blocks by XORing them together. More precisely, for a message $m = x_1 \| x_2 \| \cdots \| x_n$, $\text{CFB-MAC}_k(m) = y_1 \oplus y_2 \oplus \cdots \oplus y_n$, where $y_i = \mathsf{E}_k(y_{i-1}) \oplus x_i$ for $i = 2, \ldots, n$ and $y_1 = \mathsf{E}_k(\text{IV}) \oplus x_1$, IV being an initialization vector. For the sake of simplicity, we assume that all messages have a length that is a multiple of 64 bits. We also assume in all the questions of this problem that IV is constant and known.

1. Assume we have access to an oracle $\mathcal{O}$ that computes the CFB-MAC under a given secret key k and a fixed known IV. Show that you can recover $\mathsf{E}_k(\text{IV})$ by querying *only one* message to the oracle.

2. Assume that an adversary has access to an oracle $\mathcal{O}$ that computes the CFB-MAC under a given secret key k and a fixed known IV. The adversary would like to find a CFB-MAC collision on two different messages of 192 bits. How many messages of 192 bits does the adversary need to query to $\mathcal{O}$ in order to get a collision with probability close to $0.9996 \approx 1 - e^{-8}$?

3. Given a message m of n blocks and $h = \text{CFB-MAC}_k(m)$. Show how it is possible to generate a new message m' of n blocks and a $h' \in \{0,1\}^{64}$ such that $m' \neq m$ and $\text{CFB-MAC}_k(m') = h'$.

4. Assume we are given IV, $\mathsf{E}_k(\text{IV})$, and a $h \in \{0,1\}^{64}$. Show how it is possible to generate a message m of two blocks, such that $\text{CFB-MAC}_k(m) = h$.

5. Can we extend the attack of the previous question to messages m of more than two blocks? Explain your answer.

Exercise 2 Analysis of the Floyd Cycle Finding Algorithm

We recall here the Floyd cycle finding algorithm. Remember that the algorithm forms a graph which looks like a ρ. We call the length of the tail λ and the length of the loop τ.
Input: an initial string x_0, a function $F : \{0,1\}^* \to \{0,1\}^n$.
Output: The length of the tail λ.

1: $a \leftarrow x_0$ //(tortoise)
2: $b \leftarrow x_0$ //(hare)
3: **repeat**
4: $\quad a \leftarrow F(a)$
5: $\quad b \leftarrow F(F(b))$
6: **until** $a = b$
7: $a \leftarrow x_0$
8: $\lambda \leftarrow 0$
9: **while** $a \neq b$ **do**
10: $\quad a \leftarrow F(a)$
11: $\quad b \leftarrow F(b)$
12: $\quad \lambda = \lambda + 1$
13: **end while**
14: **return** λ

1. Explain how to adapt the algorithm so that it finds a collision in the function F. In which case is your algorithm failing?

2. Let $x_i = F(x_{i-1})$. Show that the condition $a = b$ in the first loop is verified if and only if $i \geq \lambda$ and $\tau | i$. Explain at what point of the graph the first loop stops.

3. How many iterations (in term of λ and τ) are required for the first loop to terminate?

4. How many times (in term of λ and τ) do we do the second while loop?

5. Assume that $\Pr[\lambda \leq \tau] = 1/2$ and that $\Pr[\lambda \geq 2\tau]$ is negligible, i.e., close to 0. Knowing that $E[\lambda] = E[\tau] = \sqrt{\pi N/8}$ for a random function over a set of cardinality N, what is the overall expected complexity (in terms of calls to F) of the algorithm? How much memory does it use?