

# CS387 - Applied Cryptography

Ángel Sola Orbaiceta

December 2021

# 1 Concepts

Given a message  $m \in M$ , where  $M$  is the set of all possible messages, and a key  $k \in K$ , where  $K$  is the set of all possible keys, an encryption function  $E$  can be defined as:

$$E : M \times K \rightarrow C$$

where  $c \in C$  is the *ciphertext* (being  $C$  the set of all possible ciphertexts). Conversely, a decryption function  $D$  can be defined as:

$$D : C \times K \rightarrow M$$

The **correctness property** states that, for all messages and keys, decrypting the result of encrypting a message must result in the message itself. Mathematically:

$$\forall m, k : D_k(E_k(m)) = m$$

The **security property** states that the ciphertext reveals nothing about the key or original message.

## 1.1 One-Time Pad

The one-time pad is based in the XOR ( $\oplus$ ) function. The XOR function satisfies the property that any value XOR-ed with itself equals zero:  $x \oplus x = 0$ . The one-time pad uses this property so that, by using a key that's the same size as the ciphertext, we can do:

$$c = m \oplus k$$

$$m = c \oplus k$$

The one-time pad encryption and decryption functions are implemented in the *source/one\_time\_pad.py* file.

## 1.2 Perfect Cipher

Recall that given two events  $A$  and  $B$  in the same probability space, the **conditional probability** of  $B$  given that  $A$  occurred is:

$$P(B|A) = \frac{P(A \cap B)}{P(A)}$$

Now, given two messages,  $m$  and  $m^*$  drawn from the set of messages  $M$  are encrypted using a key  $k \in K$  to produce a ciphertext  $c \in C$ , a **perfect cipher** must hold that:

$$P(m = m^* | E_k(m) = c) = P(m = m^*)$$

We can now prove that the one-time pad is a perfect cipher as follows. For the one-time pad:

$$P(E_k(m) = c) = \sum_{m_i \in M} \sum_{k_i \in K} \frac{P(E_{k_i}(m_i) = c)}{|M| \times |K|} = \frac{|M|}{|M| \times |K|} = \frac{1}{|K|}$$

and, assuming a uniform distribution for the key space:

$$P(m = m^* \cap E_k(m) = c) = P(m = m^*) \times P(k = k^*) = \frac{P(m = m^*)}{|K|}$$

Therefore:

$$P(m = m^* | E_k(m) = c) = \frac{\frac{P(m=m^*)}{|K|}}{\frac{1}{|K|}} = P(m = m^*)$$

which means that the ciphertext reveals nothing about the key or original message.

**Shanon's theorem** If a cipher is perfect, it must be impractical ( $|K| \geq |M|$ ).

## 2 Application of Symmetric Ciphers

**Kolmogorov complexity** The complexity  $K$  of a sequence  $s$  ( $K(s)$ ) is the length of the shortest possible description of  $s$ .  $s$  is random if  $K(s) = |s| + C$ . The Kolmogorov complexity is uncomputable.

**A Pseudo-Random Number Generator (PRNG)** produces a long sequence of seemingly random bytes given an initial seed value. In Linux, *dev/random* can be used as a randomness pool.