Exercise 1.45

Let N be a large integer and let $K = M = C = \mathbb{Z}/N\mathbb{Z}$. For each of the functions

$$e : K \times M \to C$$

listed in (a)–(c), answer the following questions:

1 Is e an encryption function?

2. If e is an encryption function, what is its associated decryption function d?

3. If e is not an encryption function, can you make it into an encryption function by using some smaller, yet reasonably large, set of keys?

(a) $e_k(m) \equiv k - m(\mod N)$

Ans:

1. Yes, $e_k(m) \equiv k - m \pmod N$ is an encryption function, since subtraction is reversible.
2. To recover the message m from the ciphertext c, we can solve for m using $m \equiv k - c \pmod N$. Thus, the decryption function is the function $d : K \times C \to M$ defined by

$$d_k(c) \equiv k - c \pmod N$$

(b) $e_k(m) \equiv k \cdot m \pmod N$

Ans:

1. It depends on whether or not k has a modular inverse. If k has a modular inverse, then $e_k(m) \equiv k \cdot m \pmod N$ is an encryption function. Here, k has modular inverse if $(k, N) = 1$.
2. If k is invertible modulo N, define its modular inverse k^{-1}, such that $k^{-1} \cdot k \equiv 1 \pmod N$. Then the decryption function is

$$d_k(c) \equiv k^{-1} \cdot c \pmod N.$$

(c) $e_k(m) \equiv (k+m)^2 \pmod{N}$

Ans:

1. No, this function is not always invertible, because squaring loses sign information and can lead to collisions (that is, multiple plaintexts mapping to the same ciphertext).
2. If $k+m$ is restricted to a subset of values that allow unique reversibility, then we can make the function an encryption function. For example, working only within a specific modular residue class such as the subset of quadratic residues modulo N.

Exercise 1.47

Alice and Bob choose a key space K containing 256 keys. Eve builds a special-purpose computer that can check $10,000,000,000$ keys per second.

(a) How many days does it take Eve to check half of the keys in K?

Ans

The time required for Eve to check half the keys in K is computed as follows. Half the keys is $\frac{256}{2} = 128$ keys. At a rate of 10^{10} keys per second, the time taken for Alice to check half of the keys in K is $\frac{128}{10^{10}}$ seconds $= 1.28 \times 10^{-8}$ seconds which is approximately $\frac{1.28}{864 \times 10^{10}}$ days.

(b) Alice and Bob replace their key space with a larger set containing 2^B different keys. How large should Alice and Bob choose B in order to force Eve's computer to spend 100 years checking half the keys? (Use the approximation that there are 365.25 days in a year.)

Ans

Half the keys is $\frac{2^B}{2} = 2^{B-1}$, and 100 years in seconds is $100 \times 365.25 \times 24 \times 60 \times 60 = 3155760000$ seconds. Set up the equation

$$\frac{2^{B-1}}{10^{10}} \text{ sec} = 3155760000 \text{ sec},$$

to obtain

$$2^{B-1} = 3.15576 \times 10^{19}.$$

Take logarithms to get

$$\begin{aligned} B &= 1 + \log_2(3.15576 \times 10^{19}) \\ &= 1 + \frac{\log_{10}(3.15576) + \log_{10}(10^{19})}{\log_{10}(2)} \\ &= 1 + \frac{\log_{10}(3.15576) + 19}{\log_{10}(2)} \\ &= 1 + \frac{19.4991}{0.3010} \\ &= 65.7746. \end{aligned}$$

Thus, Alice and Bob should choose $B \approx 66$ to ensure 100 years of security against Eve's computer.