CS178 F21 Homework 1

Michael Glushchenko, 9403890

October 7, 2021

1 Decoding a String

I used the following Python code to decrypt the two strings in problem one:

```
import numpy as np
import string
```

```
test = "pynnouolnx" #should be "helloworld" when decoded
cipher1 = "gugnnmcssyyx" #question 1 part 1
cipher2 = "bauoligjqpwlx" #question 2 part 2
```

```
# encryption function
def f(x):
    return (pow(x,5) - 3*pow(x,3) + 7*x - 4) % 26
```

```
alphabet = np.array(list(string.ascii_lowercase))

temp = np.arange(26)
for i in temp:
    temp[i] = f(i)
key = alphabet[temp]
```

```
# decryption function
def decrypt(word):
    result = ""
    for letter in list(word):
        result += alphabet[list(key).index(str(letter))]
    return result
```

```
print("Results: ", decrypt(test), decrypt(cipher1), decrypt(cipher2))
```

```
Results:  helloworld iwillsucceed byworkinghard
```

2 A Deterministic Algorithm is Probabilistic.

A **probabilistic algorithm** is a Turing machine equipped with a random tape. The algorithm's output is a probability distribution over some set of elements (the algorithm spits out an output from the set according to the distribution). These algorithms may produce different outputs upon the same input, the chances given according to the probability distribution.

A **deterministic algorithm** is simply an algorithm where the probability distribution describing the algorithm's output is concentrated on a single element, meaning that, given the same input, the algorithm will always give back the same output. We can see that a deterministic algorithm $\mathcal{A}$ is a probabilistic algorithm with some specific "parameters" applied to the probabilistic algorithm definition (size of the output set for each given input is set to 1, probability distribution is set to be concentrated on one element).

3 Calculating Probabilities.

3.0.1

$$\mathcal{P}[x = x_0;\ x \leftarrow \mathcal{D}_1]$$

$\mathcal{D}_1 \sim Uniform(\{0,1\})$, so it's probability density function is $f(x) = 1/2$. So,

$$\mathcal{P}[x = x_0;\ x \leftarrow \mathcal{D}_1] = \begin{cases} 1/2, & x_0 = 0 \\ 1/2, & x_0 = 1 \end{cases}$$

3.0.2

$$\mathcal{P}[x = x_0;\ x \leftarrow \mathcal{D}_2]$$

$\mathcal{D}_2 \sim Bernoulli(p)$, so it's probability density function is $f(x) = \begin{cases} p, & x = 1. \\ 1-p, & x = 0. \end{cases}$.

Thus,

$$\mathcal{P}[x = x_0;\ x \leftarrow \mathcal{D}_2] = \begin{cases} p, & x_0 = 1. \\ 1-p, & x_0 = 0. \end{cases}$$

3.0.3

$$\mathcal{P}[x_0 \leftarrow \mathcal{A}(x);\ x \leftarrow \mathcal{D}_1]$$

$\mathcal{D}_1 \sim Uniform(\{0,1\})$ has the density function $f(x) = \frac{1}{2}$. The probability of obtaining a 0 as the output of $\mathcal{D}_1$ is $\frac{1}{2}$, so that is also the probability of getting a 0 as the input of $\mathcal{A}$, and thus also the probability of getting a 1 as the output of $\mathcal{A}$. The probability of $\mathcal{D}_1$ outputting a 1 is $\frac{1}{2}$, so the probability of getting a 1 as the input of $\mathcal{A}$ is $\frac{1}{2}$, and thus the probability of $\mathcal{A}$ outputting a 0 is $\frac{1}{2}$. The probability of each possible outcome is $\frac{1}{2}$, so we can conclude that

$$\mathcal{P}[x_0 \leftarrow \mathcal{A}(x);\ x \leftarrow \mathcal{D}_1] = \begin{cases} 1/2, & x_0 = 0 \\ 1/2, & x_0 = 1 \end{cases}$$

3.0.4

$$\mathcal{P}[x_0 \leftarrow \mathcal{A}(x);\ x \leftarrow \mathcal{D}_2]$$

$\mathcal{D}_2 \sim Bernoulli(p)$. The output of this distribution is used as the input for $\mathcal{A}(x)$. Since $\mathcal{D}_2$ outputs 1 with a probability p, $\mathcal{A}(x)$ will output 0 with probability p, and 1 with probability $1-p$. So,

$$\mathcal{P}[x_0 \leftarrow \mathcal{A}(x);\ x \leftarrow \mathcal{D}_2] = \begin{cases} p, & x_0 = 0. \\ 1-p, & x_0 = 1. \end{cases}$$

4 One-time Perfect Security

4.1 Given.

(Gen, Enc, Dec) defined as follows:

1. $Gen(1^n)$ samples a key $k \in \mathcal{K}$.
2. $Enc(k, m)$ takes as input an n-bit m and a key k, outputs an n-bit c.
3. $Dec(k, c)$ takes as input an n-bit c and a key k, outputs an n-bit m.

(Gen', Enc', Dec') defined as follows:

1. $Gen'(1^n)$ outputs a key $(k_1, k_2) \in \mathcal{K} \times \mathcal{K}$, $k_1 \perp\!\!\!\perp k_2$.
2. $Enc'((k_1, k_2), m) = Enc(k_2, Enc(k_1, m))\ \forall m \in \{0,1\}^n, (k_1, k_2) \in \mathcal{K} \times \mathcal{K}$.
3. $Dec'((k_1, k_2), c) = Dec(k_1, Dec(k_2, c))\ \forall c \in \{0,1\}^n, (k_1, k_2) \in \mathcal{K} \times \mathcal{K}$.

4.2 Claim.

We claim that if (Gen, Enc, Dec) satisfies one-time perfect security, then (Gen', Enc', Dec') satisfies one-time perfect security.

4.3 Proof.

Assume the following distributions are identical $\forall m_1, m_2 \in \{0,1\}^n$, $|m_1| = |m_2|$:

$$\mathcal{D}_1 := \{c := Enc(k,\ m_1);\ k \leftarrow Gen(1^n),\ m_1 \in \{0,1\}^n\}$$

$$\mathcal{D}_2 := \{c := Enc(k,\ m_2);\ k \leftarrow Gen(1^n),\ m_2 \in \{0,1\}^n\}.$$

We now have to prove that the following two distributions are equivalent:

$$\mathcal{D'}_1 := \{c := Enc(k_2, Enc(k_1, m_1));\ k_1 \leftarrow Gen(1^n),\ k_2 \leftarrow Gen(1^n),\ m_1 \in \{0,1\}^n,\ k_1 \perp\!\!\!\perp k_2\}$$

$$\mathcal{D'}_2 := \{c := Enc(k_2, Enc(k_1, m_2));\ k_1 \leftarrow Gen(1^n),\ k_2 \leftarrow Gen(1^n),\ m_2 \in \{0,1\}^n,\ k_1 \perp\!\!\!\perp k_2\}.$$

Define $m'_1, m'_2 \in \{0,1\}^n$ as $m'_1 = Enc(k_1, m_1)$ and $m'_2 = Enc(k_1, m_2)$. Then,

$$Enc(k_2, Enc(k_1, m_1)) = Enc(k_2, m'_1)$$

$$Enc(k_2, Enc(k_1, m_2)) = Enc(k_2, m'_2),$$

from which it follows that $\mathcal{D'}_1$ and $\mathcal{D'}_2$ can be written as:

$$\mathcal{D'}_1 := \{c := Enc(k, m'_1);\ k \leftarrow Gen(1^n),\ m'_1 \in \{0,1\}^n\}$$

$$\mathcal{D'}_2 := \{c := Enc(k, m'_2);\ k \leftarrow Gen(1^n),\ m'_2 \in \{0,1\}^n\}$$

We have $|m_1| = |m_2| \Rightarrow |m'_1| = |m'_2|$, and we know $m'_1, m'_2 \in \{0,1\}^n$. Thus we can conclude that $\mathcal{D'}_1 \equiv \mathcal{D}_1 \equiv \mathcal{D}_2 \equiv \mathcal{D'}_2$, which implies $\mathcal{D'}_1 \equiv \mathcal{D'}_2$. Thus the encryption scheme (Gen', Enc', Dec') satisfies one-time perfect security. $\square$