Assignment 4

Konstantinos Filippou
ics23044

Exercise 1

Let p be a large prime such that $2^{159} < p < 2^{160}$ and

$$K = M = C = \{1, 2, \dots, p-1\}.$$

Alice and Bob choose a random key $k \in K$ and define:

$$e_k(m) = km \pmod p$$
$$d_k(c) = k^{-1}c \pmod p$$

where k^{-1} is the inverse of k modulo p.

Properties

The algorithm satisfies:

- Efficient encryption: multiplication modulo p is efficient.
- Efficient decryption: the inverse k^{-1} can be computed using the Extended Euclidean Algorithm.

Security Against COA

The scheme is **not secure** under a Ciphertext-Only Attack (COA).

Given two ciphertexts:

$$c_1 = km_1 \pmod p, \quad c_2 = km_2 \pmod p,$$

we have:

$$\frac{c_2}{c_1} \equiv \frac{m_2}{m_1} \pmod p.$$

This reveals information about plaintext ratios.

Security Against KPA

The scheme is **not secure** under a Known Plaintext Attack (KPA).

Given one pair (m, c):

$$c \equiv km \pmod{p}$$

$$k \equiv cm^{-1} \pmod{p}$$

Thus the key is recovered immediately.

Without Modulo Reduction

If encryption is defined as:

$$e_k(m) = km$$

without reduction modulo p, then:

$$k = \frac{c}{m}$$

The scheme becomes even more insecure.

Exercise 2

Using Pohlig–Hellman with:

$$p = 2633, \quad e = 9$$

encrypt and decrypt the message:

"The gold is hidden in the garden!"

Solution

Since encryption is:

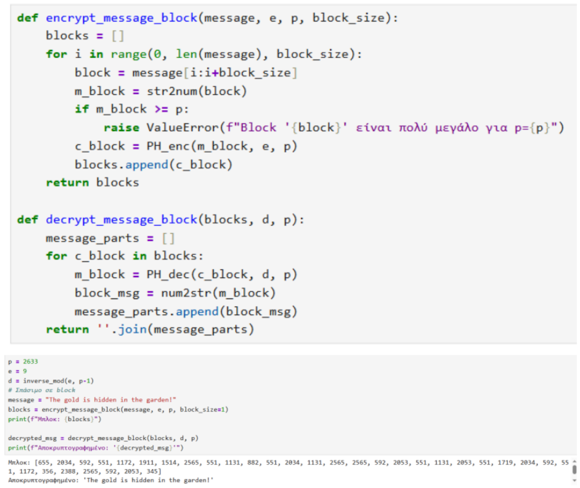
$$c = m^e \pmod{p},$$

and p is small relative to the message, block encoding is required.

Because $p = 2633$, only single-character blocks fit:

$$\text{block size} = 1.$$

Thus encryption and decryption are applied character-by-character.



```python
def encrypt_message_block(message, e, p, block_size):
    blocks = []
    for i in range(0, len(message), block_size):
        block = message[i:i+block_size]
        m_block = str2num(block)
        if m_block >= p:
            raise ValueError(f"Block '{block}' είναι πολύ μεγάλο για p={p}")
        c_block = PH_enc(m_block, e, p)
        blocks.append(c_block)
    return blocks

def decrypt_message_block(blocks, d, p):
    message_parts = []
    for c_block in blocks:
        m_block = PH_dec(c_block, d, p)
        block_msg = num2str(m_block)
        message_parts.append(block_msg)
    return ''.join(message_parts)
```

```python
p = 2633
e = 9
d = inverse_mod(e, p-1)
# Σπάσιμο σε block
message = "The gold is hidden in the garden!"
blocks = encrypt_message_block(message, e, p, block_size=1)
print(f"Μπλοκ: {blocks}")

decrypted_msg = decrypt_message_block(blocks, d, p)
print(f"Αποκρυπτογραφημένο: '{decrypted_msg}'")
```

```
Μπλοκ: [655, 2034, 592, 551, 1172, 1911, 1514, 2565, 551, 1131, 882, 551, 2034, 1131, 2565, 2565, 592, 2053, 551, 1131, 2053, 551, 1719, 2034, 592, 55
1, 1172, 356, 2388, 2565, 592, 2053, 345]
Αποκρυπτογραφημένο: 'The gold is hidden in the garden!'
```

Figure 1: Enter Caption

Exercise 3

Given:

$$p = 29$$

Ciphertext:

$$04\ 19\ 19\ 11\ 04\ 24\ 09\ 15\ 15$$

and known that:

$$24 \leftrightarrow U \quad (20)$$

Encoding:

$$A = 00, \ldots, Z = 25$$

Finding e

We solve:

$$20^e \equiv 24 \pmod{29}$$

Compute powers:

$$20^1 \equiv 20$$
$$20^2 \equiv 23$$
$$20^3 \equiv 25$$
$$20^4 \equiv 7$$
$$20^5 \equiv 24$$

Thus:

$$e = 5$$

Finding d

$$d = e^{-1} \pmod{28}$$

$$5^{-1} \equiv 17 \pmod{28}$$

Decrypting

$$m = c^{17} \pmod{29}$$

Results:

$$04 \to 6 \to G$$
$$19 \to 14 \to O$$
$$19 \to 14 \to O$$
$$11 \to 3 \to D$$
$$04 \to 6 \to G$$
$$24 \to 20 \to U$$
$$09 \to 4 \to E$$
$$15 \to 18 \to S$$
$$15 \to 18 \to S$$

Plaintext:

$$\boxed{\text{GOOD GUESS}}$$

Exercise 4

A weakness of Pohlig–Hellman is that the plaintext m may not be a generator modulo p, and may have small order.

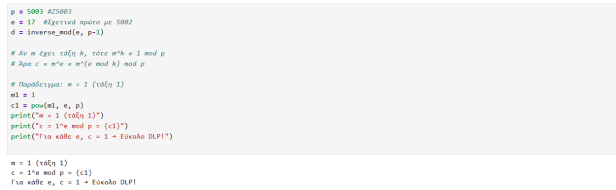
For example, in $\mathbb{Z}_{5003}$:

$$m = 1$$

Then:

$$c = 1^e \equiv 1$$

The discrete logarithm problem becomes trivial.



```
p = 5003 #Z5003
e = 17  #Σχετικά πρώτο με 5002
d = inverse_mod(e, p-1)

# Αν m έχει τάξη k, τότε m^k ≡ 1 mod p
# Άρα c ≡ m^e ≡ m^(e mod k) mod p

# Παράδειγμα: m = 1 (τάξη 1)
m1 = 1
c1 = pow(m1, e, p)
print("m = 1 (τάξη 1)")
print("c = 1^e mod p = {c1}")
print("Για κάθε e, c = 1 → Εύκολο DLP!")
```

```
m = 1 (τάξη 1)
c = 1^e mod p = {c1}
Για κάθε e, c = 1 → Εύκολο DLP!
```

Figure 2: Enter Caption

Exercise 5

We define:

- PH(bits)
- PH_enc(m,e,p)
- PH_dec(c,d,p)
- str2num(s)
- num2str(n)

Example plaintext:

"Hello there!"

Converted to integer:

$$5107117560890101249815240$$

Encryption and decryption return:

Hello there!

Thus correctness is verified.