PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
ESCUELA DE INGENIERIA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACION

Criptografía y Seguridad Computacional - IIC3253
AY1 - Crypto 101, One Time Pad (OTP) y Aritmetica Modular!
15 de Marzo de 2023

Preguntas

1. Trivia Modular (V o F).
   a) $10 \mod 3 = 1$
   b) $28 \mod 8 = 4$
   c) $13 \equiv 23 \pmod 5$
   d) $50 \equiv 1 \pmod 7$
   e) $1 \not\equiv 132 \pmod{132}$
   f) $1 \oplus 0 = 1$
   f) $1 \oplus 1 = 0$

2. Si $a \equiv b \pmod n$ y $c \equiv d \pmod n$, $k \in \mathbb{Z}$, demuestre:
   1) $a + c \equiv b + d \pmod n$
   2) $ac \equiv bd \pmod n$
   3) $10^n \equiv 1 \pmod 9, n \in \mathbb{N}$
   4) $c\dot{1}0^n \equiv c \pmod 9, n \in \mathbb{N}, c \in \mathbb{N}$
   5) $a - c \equiv b - d \pmod n$
   6) $a \equiv a + kn \pmod n$
   7) $a \equiv a + kn \pmod n$
   8) $a^k \equiv b^k \pmod n, k \geq 1$

3. Trivia Crypto (V o F)
   a) Chosen-plaintext attack: Adversario puede pedir encriptar algunos plain texts m.
   b) Chosen-ciphertext attack: Adversario puede pedir desencriptar algunos cyphertext c.
   c) Known plaintext attack: Adversario tiene acceso parcial al plain text m, y el cyphertext c.
   d) Ciphertext-only attack: Adversario conoce ciphertext c y quiere conseguir el plain text m.

4. Definamos un sistema de One Time Pad (OTP) que acepte unicamente caracteres alfabeticos en mayusculas (A-Z) y mensajes de largo menor o igual al pad entregado.

```
def encode(char):
  return ord(char) - 65

def decode(char):
  return chr(char + 65)
```

```
def encrypt(pad, msg):
  # Output string and extension of the pad
  out = ""

  for i in range(len(pad)):
    # Get the encoded values for both characters
    padchar, msgchar = encode(pad[i]), encode(msg[i])

    # Get the mod 26 of the net value
    net = (msgchar + padchar) % 26

    # Add the char value to the output
    out += decode(net)

  return out
```

```
def decrypt(pad, msg):
  # Output string and extension of the pad
  out = ""

  for i in range(len(pad)):
    # Get the encoded values for both characters
    padchar, msgchar = encode(pad[i]), encode(msg[i])

    # Get the mod 26 of the net value
    net = (msgchar - padchar) % 26

    # Add the char value to the output
    out += decode(net)

  return out
```

```
encrypt("B", "D")
# padchar = 1
# msgchar = 3
# net = 1 + 3 (mod 26) = 4
# out = decode(4) = E
>>> E
```

```
encrypt("HMAXCTKQICNLG", "HELLOWORLDOTP")
>>> OQLIQPYHTFBEV
decrypt("HMAXCTKQICNLG", "OQLIQPYHTFBEV")
>>> HELLOWORLDOTP
```

5. Considerando el esquema OTP definido en la pregunta anterior, implementa los siguientes ataques. Puedes utilizar las funciones definidas anteriormente o escribir una respuesta teorica.

   a. Chosen-plaintext attack

```
encrypt("?????????????", input())
>>> OQLIQPYHTFBEV
```

```
|
|
|
```

   b. Chosen-ciphertext attack

```
decrypt("?????????????", input())
>>> HELLOWORLDOTP
```

```
|
|
|
```

   c. Known plaintext attack

```
encrypt("?????????????", "HELLOWORLDOTP")
>>> OQLIQPYHTFBEV
```

```
|
|
|
```

6. Considerando el esquema OTP definido en la pregunta anterior, responde las siguientes poreguntas teoricas respecto a un Ciphertext-only attack.

```
>>> OQLIQPYHTFBEV
```

   a. Que condiciones se deben cumplir para que nuestro modelo pueda defenderse de este ataque?

```
|
|
|
```

   b. Que vulnerabilidades pueden surgir al no respetar alguna de estas condiciones?

```
|
|
|
```