Pregunta 1
Se tiene el siguiente mecanismo de cifrado para mensajes de una sola letra:

$$\begin{aligned}\mathcal{M} &= \{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, w, x, y, z\}\\ K &= \{a, b, c, d, e, f, g, h, i, j, k, l, m, n, \tilde{n}, o, p, q, r, s, t, u, w, x, y, z\}\\ E(k, m) &:= k + m \bmod 26\\ D(k, c) &:= c - k \bmod 26\end{aligned}$$

a) ¿ Es perfectamente seguro este mecanismo? Justifica.

b) Si quitamos la llave ñ y lo demás se mantiene igual. ¿Cambia la seguridad del mecanismo? Justifica.

Solución:

a) No es perfectamente seguro pues no cumple la siguiente propiedad:

"Las cardinalidades de los conjuntos claves, mensajes y criptogramas son iguales"

Mostraremos por que:

Al tener una suma en la encriptación y una resta en la decriptación asumimos que:

1. Operamos los índices de los caractéres tanto del mensaje plano como el de la contraseña (en su respectivo alfabeto).

2. El mensaje cifrado es un número.

Sin aún delimitar el alfabeto del criptograma(siguiente pregunta) ya sabemos que la cardinalidad de conjuntos claves es 27 pero de mensajes es 26.

Por lo tanto no es perfectamente seguro.

b) Bajo este criterio la cardinalidad de conjuntos clave y mensaje sí son iguales, veremos que el de criptogramas no y que además no cumplen con la siguiente propiedad:

"Para toda pareja (c, m), existe **una** k tal que m se cifra como c usando k."

No importa si consideramos que los alfabetos estan en índice base cero o uno, llegaremos a la misma conclusión. Como en la operación usan módulo veintiseis consideramos pertinente usar índice base uno.

Ahora tenemos como posibles claves los elementos del alfebeto utilizado en E.E.U.U.A (26 crácteres), así digamos para el mensaje 'a', tenemos los siguientes posibles valores comenzando con la llave 'a' y en orden cronologico hasta llegar a 'z':

$$[2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27$$

Para el cáracter 'z' los siguientes:

$$[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26]$$

Observamos que es imposible para el mensaje 'a' producir el criptograma 1, pues el índice de 'a' ya es 1 y con las claves operamos en un rango $\{1, .., 26\}$.

No existe una clave k tal que 'a' se cifré como 1 usando k.

Y aún más importante la longitud del conjunto de criptogramas($\{1, .., 27\}$) no es igual a la de claves ni mensajes.

***Nota:** Para conseguir los posibles valores utilizamos código escrito en Haskell que se encuentra en el siguiente enlace

Pregunta 2
Considera los mecanismos de cifrado adaptados a cadenas de bytes y descifra los siguientes criptotextos con la llave correspondiente.

a) Sustitución monoalfabética.

C = `0xfb0762a891`,

llave de cifrado k es la permutación ($04 \mapsto 91$, $01 \mapsto 32$, $75 \mapsto 62$, `ff` $\mapsto$ `fb`)

y los demás bytes quedan fijos.

b) Vigenére. $c =$ `0x00ab23cd45`, $k =$ `0xfa03`.

c) Afín. $c =$ `0x23aa7f`, $k = (255, 7)$.

Solución:

a) `m = 0xff0775a804`

b) Como primer paso lo que hacemos es convertir las cadenas c y k a decimal: (ponemos los '—' para hacer más visible la separación)

$$c' = 0|171|35|205|69$$
$$k' = 250|03$$

Con esto en cuento podemos proceder a generar los elementos de `m`.

```
 (0 - 250) mod 256 = 6
(171 - 3) mod 256 = 168
(35 - 250) mod 256 = 41
(205 - 3) mod 256 = 202
(69 - 250) mod 256 = 75
```

$$\boxed{\therefore \texttt{m} = \texttt{0x06a829ca4b}}$$

c) Hacemos lo mismo que en el anterior, pasar a decimal:

$$c' = 35|170|127$$

Procedemos a aplicar el algoritmo:

$$c_1 = (255 * m_1 + 7) \texttt{ mod 256 = 35} \longrightarrow \texttt{m}_1 \texttt{ = 228}$$
$$c_2 = (255 * m_2 + 7) \texttt{ mod 256 = 170} \longrightarrow \texttt{m}_2 \texttt{ = 93}$$
$$c_3 = (255 * m_3 + 7) \texttt{ mod 256 = 127} \longrightarrow \texttt{m}_3 \texttt{ = 136}$$

$$\boxed{\therefore \texttt{m} = \texttt{0xe45d88}}$$

Pregunta 3

El siguiente texto fue crifrado con una sustitución monoalfabética. Encuentra el texto claro en español y describe tu procedimiento de criptoanálisis. Puedes apoyarte del programa Ganzua [a] u otro.

```
GKVQ JJQZH XQ FQVBKWH P XQNOSUVK SZR FRNBR QMBQZNKHZ HOQRZKOR OSPRN RWSRN BQZKRZ SZ OHJHV
BRZ LRVQOKXH R JR BKZBR ASQ GQ VQOHVXRVHZ JR XQNOVKLOKHZ ASQ YROQ QJ WQHWVRTH ZSUKH XQJ
GRVQ BQZQUVRVSG ZKZWSZR KGRWKZROKHZ YSGRZR LHXVKR OHZOQUKV LRZHVRGR GRN JRGQZBRUJQGQZBQ
XQNHJRXH R XQVQOYR Q KCASKQVXR P YRNBR XHZXQ LHXKR RJORZCRV JR GKVRXR NQ BQZXKRZ OHGH
GSVRJJRN XQJ GSZXH ORXQZRN XQ RORZBKJRXHN YHVVKUJQGQZBQ ZQWVHN P OHJWRZBQN OSPH JSWSUVQ
RNLQOBH FQKRNQ VQTHVCRXH LHV JR VQNROR ASQ VHGLKR OHZBVR QJJHN NS UJRZOR P JKFKXR OVQNBR
RSJJRZXH P VSWKQZXH QBQVZRGQZBQ HLSQNBR RJ LVHGHZBHVKH NHUVQ OSPR OKGR ZHN YRJJRURGHN
P R SZRN OKZOH H NQKN GKJJRN XQZBVH XQJ GRV RXFQVBKRNQ SZR LQASQÑR KNJR XQ RNLQOBH XQNQVBKOH
ASKCR NQR GRN RXQOSRXH XQOKV ASQ NS LHNKOKHZ NQ RXKFKZRUR WVROKRN R JRN NRJFRDQN VHGLKQZBQN
ASQ JR QZFHJFKRZ SZRN XHN GKJJRN GRN OQVOR RJCRURNQ HBVR KNJR GRN LQASQÑR YHVVKUJQGQZBQ
QNORVLRXR P QNBQVKJ VHXQRXR QZ FRVKRN LRVBQN LHV RGHZBHZRGKQZBHN XQ HNOSVRN VHORN
```

Con frecuencias:

```
R: 125, Q: 95, H: 66, N: 58, Z: 57, V: 56, K: 52, J: 43, O: 40, X: 39, S: 34, B: 33,
G: 30, L: 18, U: 13, W: 11, F: 10, P: 10, A: 9, Y: 7, C: 5, T: 2, Ñ: 2, M: 1, D: 1
```

[a] Ganzüa

Solución:

No nos animamos a usar ganzua por que fue un problema instalarlo, pero fuimos viendo videos e intentando inicialmente a mano y en papel, pero nos generaba mucho ruido y volver a iniciar era un desastre lleno de frustración e impotencia y el tiempo parecia perdido a pesar de la experiencia de ir viendo cuales podian ser las correctas.

Es por eso que decidimos realizarlo con ayuda de emacs, con los comandos para sustituir(M-x replace-string), para buscar cáracteres (C-s M-c) con case-sensitive y con ayuda de los buffers nos permitio (con ayuda de lo hecho en papel) a tener claras las ideas y ver por donde ir, en cuanto nos equivocabamos bastaba con hacer redo (C-x u) en los respectivos buffers y volviamos a un punto donde pareciamos ir seguro. A continuación mostramos la ubicación de los buffers y como es que nos ayudaron:

En el buffer 3.txt teniamos el texto en el que ibamos reemplazando caracteres mayusculos con nuestra suposición actual, en el buffer guess.txt es donde desarrollamos nuestro pensamiento y en que nos basamos para hacer las conjeturas, se puede ver que llevabamos para cada sustitución un alfabeto de caracteres pendientes. Y por último en saves.txt ibamos guardando el texto hasta con las sustituciones que creiamos pertinentes, estos últimos dos fueron los buffers donde más tuvimos que borrar pues regresabamos a "versiones" anteriores cuando cometiamos un error.

***NOTA:** Estos archivos, sobrea todo guess.txt, se encuentran aquí.

Pregunta 4

En este ejercicio calcularás la probabilidad de ganar el juego para ciertos adversarios. Considera el mecanismo de Vigenère para textos de longitud 3 sobre el alfabeto de 26 letras.

- Supón que las posibles llaves son solo las cadenas de 2 letras y la llave se escoge aleatoriamente. El adversario A propone los mensajes $m_0 = aaa$ y $m_1 = aab$, y cuando recibe el criptotexto c compara la primera letra con la tercera y devuelve 0 si son iguales, 1 en caso contrario. Calcula la probabilidad de que A gane el juego.

- Supón que las posibles llaves son cadenas de 1, 2 o 3 letras, para generar una llave se escoge aleatoriamente $t \in \{1, 2, 3\}$ y luego se escoge aleatoriamente una cadena de longitud t. El adversario A propone los mensajes $m_0 = aab$ y $m_1 = abb$, y cuando recibe el criptotexto c compara la primera letra con la segunda y devuelve 0 si son iguales, 1 en caso contrario. Calcula la probabilidad de que A gane el juego.

- Construye un mejor adversario que el del inciso b), es decir, un adversario con una probabilidad mayor de ganar el juego.

Solución:

a) Sea $\mathbb{P}(A\ gane)$ la probabilidad de que A gane el juego.

Entonces, si m_0 fuera encriptado, la primera y la tercera letra de c son siempre iguales, y si m_1 fuera encriptado, son siempre diferentes. Es decir:

$$c = c_0c_1c_2$$

Para m_0 siempre pasa que $c_0 = c_2$ y para m_1 tenemos que $c_0 \neq c_2$.

Como cada uno de los dos mensajes tiene la misma probabilidad de estar encriptado, la probabilidad de que A gane el juego es simplemente la probabilidad de que la primera y la tercera letra de c sean diferentes, que es $1/2$.

$$\boxed{\therefore \mathbb{P}(A\ gane) = \frac{1}{2}}$$

b) Para esto tenemos que :

$$\mathbb{P}(A\ gane) = \frac{1}{2}\mathbb{P}(A\ gane \mid m_0) + \frac{1}{2}\mathbb{P}(A\ gane \mid m_1)$$

Analicemos cada unos de los casos:

1. $\mathbb{P}(A\ gane \mid m_0)$

$$= \sum_{t=1}^{3} \frac{1}{3}\mathbb{P}(A\ gane\ |t, m_0)$$
$$= \frac{1}{3}(1) + \frac{1}{3}(\frac{1}{26}) + \frac{1}{3}(\frac{1}{26})$$
$$= \frac{14}{39}$$

2. $\mathbb{P}(A\ gane \mid m_1)$

$$= \sum_{t=1}^{3} \frac{1}{3}\mathbb{P}(A\ gane\ |t, m_1)$$
$$= \frac{1}{3}(1) + \frac{1}{3}(1 - \frac{1}{26}) + \frac{1}{3}(1 - \frac{1}{26})$$
$$= \frac{79}{234}$$

$$\therefore \mathbb{P}(A\ gane) = \frac{1}{2}(\frac{14}{39}) + \frac{1}{2}(\frac{79}{234})$$

c)
- Proponer dos mensajes m_0 y m_1 que difieran en una letra en una posición fija. Por ejemplo, $m_0 = aab$ y $m_1 = abb$.
- Si el criptotexto recibido tiene una letra en la posición fija que se sabe que es diferente entre m_0 y m_1, entonces el adversario devuelve 1. De lo contrario, devuelve 0.

Pregunta 5
Considera la siguiente modificación al sistema one-time real pad; se usan mensajes de longitud arbitraria pero acotada, es decir, $\mathcal{M} = \{0,1\}^{\leq l}$ (cadenas de bits de longitud entre 1 y l). Para encriptar se usan llaves de tamaño l y al aplicar el XOR solo se usa el número de bits necesarios de la llave, es decir, $K = \{0,1\}^l$ y $E(k,m) := k_{|m|} \oplus m$.

a) Muestra que este mecanismo no es perfectamente seguro.

b) Construye un mecanismo perfectamente seguro para $\mathcal{M}$. Justifica la seguridad de tu respuesta.

Solución:

a) El truco esta en que el mensaje puede ser de longitud $x \in [1, l]$, con l fija. Y la clave **forzosamente** es de longitud l.

Así el conjunto de posibles claves es de longitud 2^l (debido a la cardinalidad del alfabeto). Por otro lado la cardinalidad del conjunto de mensajes es de $\sum_{i=1}^{l} 2^i$, que para $l > 1$ es diferente. Por lo tanto hay más mensajes que claves (NO es seguro).

b) Primero hay que lograr que el número de llaves sea el mismo que mensajes. Como no tenemos forma de alterar a $\mathcal{M}$ solo nos queda moverle a la definición de K.

La única forma en la que sean iguales es haciendolo igual, es decir $K = \{0,1\}^{\leq l}$.

Por último debemos cambiar el modo de encriptación, este debe ser mantener la propiedad de que todas las claves sean equiprobables, esto nos lo da el xor por default. Entonces solo basta con decir que haremos con las claves que son de menor longitud que los mensajes. si $|m| = r$ y $r > |k|$ entonces se crea una nueva clave $E(k,m) := k' \oplus m$, donde $k' = k \oplus y$ con $y = k >> n$, $n = r - |k|$. En caso que $|m| \leq |k|$ es identico al descrito en el problema.

Pregunta 6
En varios esquemas de cifrado existe una llave con la que, al aplicar el método de cifrado, se deja el texto intacto, es decir, el criptotexto es igual que el texto claro. Por ejemplo, en el caso de one-time pad la llave que consiste de ceros tiene este efecto, ya que $E(0,m) = 0m = m$. Si modificamos el esquema de one-time pad para que no se utilice la cadena de ceros, ¿el esquema sigue manteniendo la seguridad perfecta? Justifica tu respuesta.

Solución:
La seguridad perfecta en el cifrado de clave única depende de que la clave utilizada sea verdaderamente aleatoria y secreta, y que se utilice sólo una vez. Si la misma clave se utiliza más de una vez, o si la clave no es completamente aleatoria o secreta, entonces la seguridad perfecta no se puede garantizar.

Es importante tener en cuenta que la seguridad perfecta del OTP se basa en la aleatoriedad e impredecibilidad de la clave utilizada. Por lo que, si se elimina la cadena de ceros y utilizamos una clave generada de manera aleatoria e impredecible de igual longitud que el mensaje, entonces el esquema seguiría siendo seguro y tendría seguridad perfecta.

Pero, si se llegara a usar una clave que no sea aleatoria o que se repita, entonces el esquema dejaría de ser seguro y la seguridad perfecta no estaría garantizada

Pregunta 7
Sea Π un mecanismo de cifrado simétrico arbitrario en el que $|K| < |\mathcal{M}|$. Considera el juego de distinguibilidad usando Π y construye un adversario que gane el juego con probabilidad mayor a $\frac{1}{2}$.

Un claro ejemplo de cifrado símetrico es `XOR`. De $|K| < |\mathcal{M}|$ sabemos que el sistema no es seguro, y en particular tratamos de tomar el mecanismo de la pregunta cinco, pero al ser binario y como ambos mensajes que escojiera el adversario tenian que tener la misma longitud no habia mucho que hacer.

Entonces utilizaremos Cesar con la restricción que tenia el ejercicio cinco:

$$M = \{a, b, c\}^{\leq 3}(\text{cadenas de longitud entre 1 y 3})$$
$$K = \{1, 2, 3\}$$

La cardinalidad de los posibles mensajes es de 39 y la de las contraseñas de 3.

El adversario escoje como mensajes $m_1 = aa$ y $m_2 = ab$. Se los pasa a la computadora encripta alguno y lo devuelve. Como es cesar cada letra se mapea a una del alfabeto y se remplaza en todas sus ocurrencias. Así hay de dos o devuelve algo de la forma xx o xy, con $x, y \in \{a, b, c\}$. El adversario sabe que si es de la forma xx entonces corresponde a la encriptación de m_1 y si no a m_2.

El adversario tiene un cien porciento de probabilidad de ganar.

Pregunta 8
Considera el one-time pad para cadenas de 8 bits y los mensajes son letras en mayúscula o minúscula y espacios, todas bajo la codificación ASCII.

- Si los criptotextos $c_1 = 10110111$ y $c_2 = 11100111$ fueron cifrados con la misma llave, ¿qué puedes deducir de los caracteres originales?
- ¿Qué puedes deducir con los criptotextos 01100110, 00110010, 00100011?

Solución:

a) Notemos que si dos criptotextos fueron cifrados con la misma llave, entonces la operación **XOR** entre ellos dará como resultado una cadena de bits que nos permitirá deducir información sobre los caracteres originales.
Y, también conociendo la llave con la que fueron cifrados podemos hacer otra operación **XOR** entre el del **XOR** anterior y la llave para obtener los caracteres originales correspondientes

b) Suponiendo que los 3 fueron cifrados con la misma llave, entonces podemos realizar una operación **XOR** entre ellos para obtener información sobre los caracteres originales.

```
01100110 XOR 00110010 XOR 00100011 = 01110101
```

`01110101` corresponde en ASCII a el caracter u.

Pregunta 9
El archivo `foto.cifrada` es una imagen encriptada con el mecanismo de César adaptado al alfabeto de bytes, es decir, el alfabeto es el conjunto $\{0, 1, ..., 255\}$ y los desplazamientos se hacen módulo 256. La imagen original está en formato PNG. Haz un programa para probar todas las llaves posibles y recupera la imagen. ¿Cuál es la llave? Anexa tu código.

Solución:
La llave es 180, el codigo fue escrito en python y se encuentra en el siguiente enlace.

Pregunta 10