

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 \tilde{n} 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 caracteres 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 están 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 alfabeto utilizado en E.E.U.U.A (26 caracteres), así digamos para el mensaje 'a', tenemos los siguientes posibles valores comenzando con la llave 'a' y en orden cronológico 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 carácter '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 $\{1, \dots, 26\}$.

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

Y aún más importante la longitud del conjunto de criptogramas ($\{1, \dots, 27\}$) 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. $c = 0x23aa7f$, $k = (255, 7)$.

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 XQNOUVK SZR FRNBR QMBQZKNHZ HOQRZKOR OSPRN RWSRN BQZKRZ SZ OHJHV
BRZ LRVQOKXH R JR BKZBR ASQ GQ VQOHVXRHVZ JR XQNOVKLOKHZ ASQ YROQ QJ WQHWRTH ZSUKH XQJ
GRVQ BQZQUVRVSG ZKZWSZR KGRWKZROKHZ YSGRZR LHXVKR OHZOQUKV LRZHVGR GRN JRGQZBRUJQGQZBQ
XQNHJRXH R XQVQOYR Q KCASKQVXR P YRNBR XHZXQ LHXR RJORZCRV JR GKVRXR NQ BQZXXRZ OHGH
GSVRJJRN XQJ GSZXH ORXQZRN XQ RORZBKJRXHN YHVVKUJQGQZBQ ZQWVHN P OHJWRZBQN OSPH JSWSUVQ
RNLQOBH FQKRNR VQTHVCRXH LHV JR VQNROR ASQ VHGLKR OHZBVR QJJHN NS UJRZOR P JKFKXR OVQNRB
RSJJRZXH P VSWKQZXH QBQVZRGQZBQ HLSQNR RJ LVHGHZBHVKH NHUVQ OSPR OKGR ZHN YRJRRURGHN
P R SZRN OKZOH H NQKN GKJJRN XQZBVH XQJ GRV RXFQVBKRNQ SZR LQASQNR KNJR XQ RNLQOBH XQNVBKOH
ASKCR NQR GRN RXQSRXH XQKV ASQ NS LHNKOKHZ NQ RXXFKZRUR WVROKRN R JRN NRJFRDN VHGLKQZBQN
ASQ JR QZFHFJKRZ SZRN XHN GKJJRN GRN OQVOR RJCRURNQ HBVR KNJR GRN LQASQNR YHVVKUJQGQZBQ
QNORVLRXR P QNBQVKJ VHXQRX QZ FRVKRN LRVBQN LHV RGHZBHZRGKQZBHN XQ HNOSVRN VHORN

Con frecuencias:

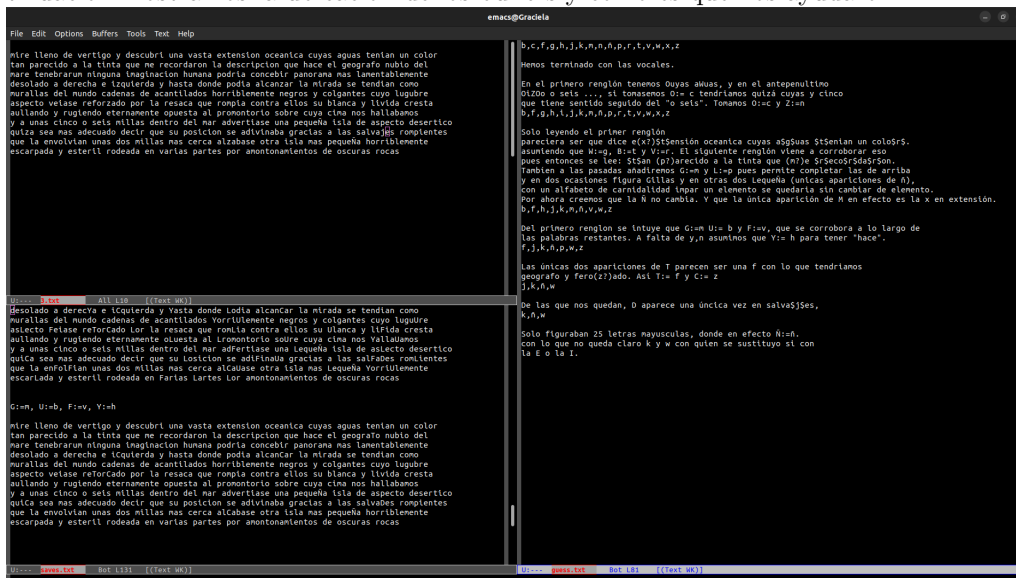
R: 125, Q: 95, H: 66, N: 58, Z: 57, V: 56, K: 52, J: 43, O: 40, X: 39, S: 34, B: 33,
G: 30, L: 18, U: 13, W: 11, F: 10, P: 10, A: 9, Y: 7, C: 5, T: 2, Ñ: 2, M: 1, D: 1

^aGanzúa

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 caracteres (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 su- posición 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, sobre todo guess.txt, se encuentran [aquí](#).

Pregunta 4**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 $E(k, m) := k_{|m|} \oplus m$.

- Muestra que este mecanismo no es perfectamente seguro.
- Construye un mecanismo perfectamente seguro para \mathcal{M} . Justifica la seguridad de tu respuesta.

Solución:

- El truco está 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).

- 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 haciéndolo 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 $y = k \gg n$, $n = r - |k|$. En caso que $|m| \leq |k|$ es idéntico al descrito en el problema.

Pregunta 6**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 simétrico 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 escogiera el adversario tenían que tener la misma longitud no había mucho que hacer.

Entonces utilizaremos Cesar con la restricción que tenía 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 escoge 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 reemplaza 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 por ciento de probabilidad de ganar.

Pregunta 8**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 $\{0, 1, \dots, 255\}$ 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 código fue escrito en python y se encuentra en el siguiente [enlace](#).

Pregunta 10