

# Tarea 2

Emilio Junoy de Juambelz

2 de abril de 2025

1. Para un alfabeto de 26 letras, ¿Cuántas matrices de  $2 \times 2$  hay que nos permitan cifrar mediante Hill? Justifica tu respuesta.

Para que una matriz de  $2 \times 2$  pueda ser utilizada para cifrar con el método de Hill, se necesita que ésta sea invertible módulo el tamaño del alfabeto, es decir 26. Para que una matriz sea invertible módulo 26 necesitamos que su determinante sea un primo relativo a 26. Podemos hacer un pequeño programa de Python que implemente el algoritmo de Euclides para calcular el máximo común divisor entre los posibles determinantes de las matrices de  $2 \times 2$  y el 26, y así determinar el número de matrices de  $2 \times 2$  invertibles módulo 26.

Realizando esto, vemos que hay un total de 163535 matrices que podemos usar para cifrar con el método de Hill, sin embargo, una de estas matrices es la identidad, así que tenemos 163535 matrices para cifrar no trivialmente con el método de Hill.

2. Investiga un caso de la vida real donde se rompió la seguridad con máximas de Kerckoffs.

Un caso muy importante en la historia es el de la máquina Enigma. Recordemos que una de las máximas de Kerckoffs es que la seguridad no debe residir en la obscuridad del método, sino en la llave utilizada. Esta máxima fue rota por los Alemanes en la segunda guerra mundial. Las claves en la máquina Enigma eran las posiciones en las que se colocaban los rotores, pero los Alemanes frecuentaban usar la misma clave o usar claves con patrones predecibles, lo que llevó a que los Aliados lograran descifrar el código Enigma. Así, las llaves que elegimos para cifrar son fundamentales para la seguridad del método, pues quebrantar esta máxima de Kerckoffs puede tener consecuencias irremediables.

3. Usando el polinomio  $x^4 + x + 1$ , da la lista de bits de salida de un LFSR asociado usando como semilla una secuencia de ceros, y cualquier otra que escojas.
4. Supón que se manda un criptotexto  $c = c_1, c_2, \dots$ , pero se pierde el bloque  $c_2$ , así que se recibe  $c = c_1, c_3, \dots$ . Al descifrar el mensaje, ¿Cuál es el efecto de un bloque no recibido al usar los modos de operación CBC, OFB y CTR?

Si usamos el modo CBC, el mensaje claro  $m_i$  depende de los textos cifrados  $c_i$  y  $c_{i-1}$  de la siguiente manera  $m_i = D_K(c_i) \oplus c_{i-1}$ . Si se pierde el bloque  $c_2$  esto afectará a los mensajes claros  $m_2$  y  $m_3$ , el mensaje  $m_2$  lo descifraríamos como  $m_2 = D_K(c_3) \oplus c_1$ , lo que no tiene sentido. El mensaje  $m_3$  lo descifraríamos como  $m_3 = D_K(c_4) \oplus c_3 = m_4$ , de esta manera, lo que pasaría es que sólo el mensaje  $m_2$  dejaría de tener sentido y el mensaje  $m_3$  no se recibe, los restantes mensajes se descifrarían de manera correcta.

Si usamos el modo OFB, el mensaje claro  $m_i$  depende de los anteriores mensajes de la siguiente manera  $m_i = c_i \oplus E_K(c_{i-1} \oplus E_K(c_{i-2} \oplus \dots))$ . Así, si se pierde el bloque  $c_2$  el único bloque de texto que podríamos descifrar correctamente es el primero.

Si usamos el modo CTR, también el único mensaje que podemos descifrar correctamente es el primero, pues en los restantes, el contador que usamos para descifrar el bloque  $c_i$  es en realidad el correspondiente al bloque  $c_{i+1}$ .