Códigos y Criptografía. Grado en Ingeniería Informática (2021)-Universidad de Valladolid
Primera entrega de ejercios (10 % nota final)

Fecha de entrega: 30 noviembre 2021, hora: 23:59.

Se espera la resolución de los ejercicios en SageMath. Se deben entregar por email a la dirección diego.ruano@uva.es tanto la sesión de SageMath (ejecutable) como una impresión en pdf de dicha sesión. **La resolución se debe acompañar de una breve explicación**.

Ejercicio 1

- Explica brevemente como definir un cuerpo finito, como se representan sus elementos y como se opera con ellos en SageMath. Considera tanto uno con un número primo de elementos como uno con una potencia de un número primo de elementos.

Ejercicio 2

- Muestra con un ejemplo comentado como usar la clase de códigos lineales en SageMath. En particular como definir un código lineal y como trabajar/obtener su matriz generadora, control, parámetros, descodificación, polinomio de pesos, código dual, ... (los conceptos vistos en clase).

Ejercicio 3

- Implementa el algoritmo de decodificación de códigos Reed-Solomon (decodificación única) visto en clase (sección 4.2 del libro de Justesen-Høholdt) y calcula un ejemplo donde se descodifican varias palabras, mostrando los diferentes situaciones en la decodificación que pueden suceder.

- Implementa el algoritmo de decodificación en lista de códigos Reed-Solomon (sección 4.3 del libro de Justesen-Høholdt) visto en clase y calcula un ejemplo donde se descodifican varias palabras, mostrando los diferentes situaciones en la decodificación que pueden suceder.

- Dado un código Reed-Solomon concreto de vuestra elección y diferentes valores de τ (número de errores admitidos), calcula de forma experimental la probabilidad de que el algoritmo de decodificación en lista devuelva una lista con más de un elemento.

- Nota: Como primera aproximación, se recomienda programarlo para un código Reed-Solomon concreto, por ejemplo el de los ejemplos 4.2.1 y 4.3.1 del libro de Justesen-Høholdt. De hecho, es suficiente, para obtener la máxima puntuación, con que vuestro programa funcione simplemente para un código Reed-Solomon concreto.

Ejercicio 4

- En clase hemos visto que la probabilidad de tener un error no detectable puede calcularse a partir del polinomio de pesos (es decir, se envía una palabra el código y se recibe otra palabra del código diferente). Para un código C y para una probabilidad de error en cada bit, p, de tu elección, calcula la probabilidad de tener un error no detectable (simplemente calcula el polinomio de pesos con Sage y luego usa la fórmula vista en clase).
- Implementa una función que introduzca con probabilidad p un error en cada bit de una palabra.
- Utiliza la función anteriormente programada para calcular de forma experimental la probabilidad de tener un error no detectable en una transmisión que usa el código C y la probabilidad de error p.

Diego Ruano