



Escuela de Ingeniería y Arquitectura **Universidad** Zaragoza

Práctica 1

Jorge Solán Morote	816259
Francisco Javier Pizarro	821259

27/02/2023

Lenguaje empleado

Para el desarrollo de esta práctica se ha elegido emplear Python dada su simplicidad a la hora de procesar datos como los manejados en esta práctica tales como binarios/strings así como la facilidad para implementar los test, la única desventaja que cabe destacar es que al ser un lenguaje interpretado y no fuertemente tipado se han tenido que desarrollar pruebas adicionales para comprobar que se produce la excepción adecuada en caso de emplear como parámetro un tipo de dato incorrecto, otro punto débil a tener en cuenta es el rendimiento respecto a otros lenguajes de programación como puede ser c++, no obstante en este caso se ha obtenido un rendimiento bastante aceptable dado el tamaño del problema.

Tests realizados

Para empezar hemos realizado los test de casos de uso normales siendo estos el cifrado/descifrado de tres tipos de mensajes:

- Un mensaje basado solo en letras.
- Un mensaje basado en letras y espacios.
- Un mensaje basado en cualquier carácter existente en la tabla ascii.

Otro tipo de tests realizados han sido los que verifican que el algoritmo lanza la excepción adecuada en caso de que introduzcamos valores inválidos que incumplen las propiedades requeridas por el algoritmo para funcionar:

- Introduciendo una mochila no óptima (algún valor es igual/inferior a la suma de los anteriores).
- Introduciendo una mochila muy pequeña(si el tamaño de la mochila es menor que 8 no contamos con suficientes bits para cifrar y descifrar ni siquiera la tabla ascii(255 valores).
- Introduciendo una w negativa.
- Introduciendo una $w > N$.
- Introduciendo una w tal que no sea primo respecto a N.
- Introduciendo una N tal que en la mochila exista un valor igual o mayor.

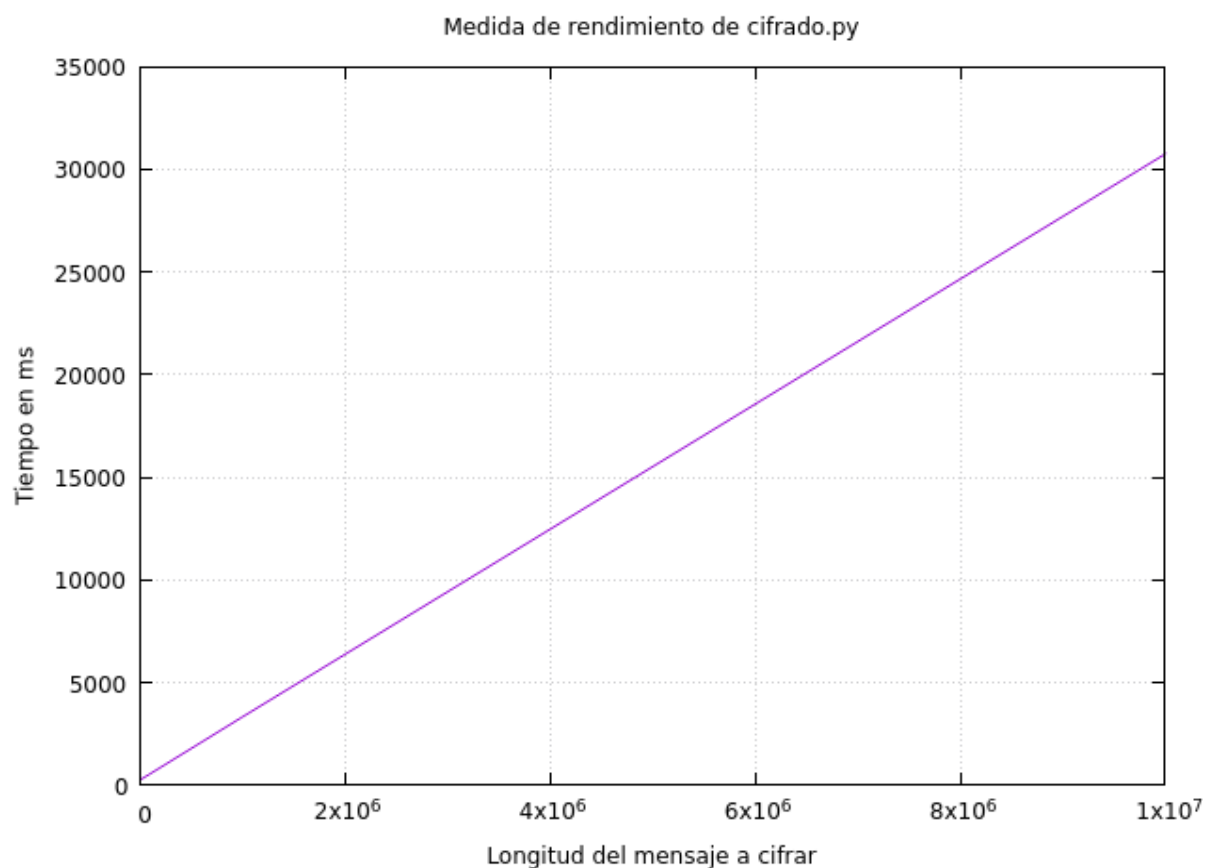
Adicionalmente dada la naturaleza del lenguaje empleado que en este caso es Python y al no ser este de forma natural estrictamente tipado y verificado en tiempo de compilación, se han tenido que realizar tests concretos para verificar que en caso de introducir en algún parámetro un tipo de dato erróneo se lanza la consecuente excepción de forma adecuada:

- Introduciendo en el parámetro mensaje un dato que no sea del tipo string.
- Introduciendo en el parámetro mochila un dato que no sea del tipo lista de enteros.
- Introduciendo en el parámetro w un dato que no sea del tipo entero.
- Introduciendo en el parámetro N un dato que no sea del tipo entero.

Test de rendimiento

Para realizar el test de rendimiento con los módulos disponibles en hendrix se ha recurrido al uso del módulo básico time para cronometrar el coste temporal del uso de la función que ofrece el módulo cifrado.py concretamente cifrarYDescifrar(), para que todas las medidas partiesen de la misma base se han empleado en todas las llamadas a esta función los mismos valores para los parámetros de la mochila, la N y la w.

Para ver cómo escalaba de forma temporal respecto al escalado de la longitud del mensaje a cifrar y descifrar se ha llamado dentro de un bucle que incrementa de forma exponencial la longitud del mensaje que se cifra y descifra, la función mencionada es cronometrada en cada iteración, para generar la gráfica se ha empleado el programa de visualización gnuplot.



El algoritmo escala de forma razonable en tiempo dado que el crecimiento de la longitud del mensaje en las pruebas realizadas ha sido exponencial mientras que el crecimiento temporal se ha mantenido lineal, concretamente en esta prueba realizada vemos que cifra y descifra un mensaje de 10.000.000 caracteres en un tiempo de 30 s. Para hacernos una idea de la magnitud del tamaño del mensaje previo hemos recurrido a compararlo con la longitud en caracteres de *El Quijote* siendo este de 2.034.611 por lo que aproximadamente se habría cifrado y descifrado en tan solo 5 segundos(empleando la misma configuración aleatoria concreta que se empleó en todas las pruebas de rendimiento).

Estas cifras sólo son representativas del crecimiento del coste temporal del algoritmo dado que estas dependen en gran medida de la mochila inicial generado de forma aleatoria en estas pruebas así como de los valores de N y w dado que dependiendo de todo ello las operaciones se realizan con mayor o menor velocidad.