Parte 1

Comando ENT

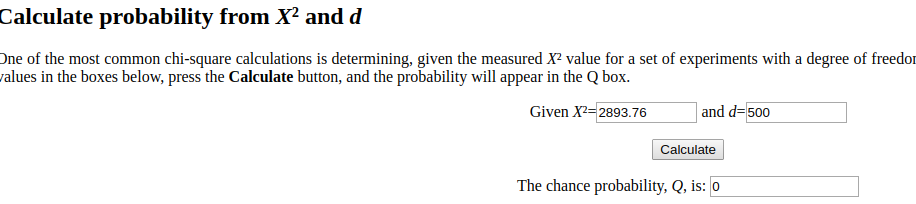
./ent -t /home/axxa/Documents/DEV/git/uc3m-Blockchain/bulk\_direcciones\_bitcoin.xlsx

Salida ENT

Caso 1:



* Entropia de 7.9 que segun la documentacion indica que el fichero cuenta con 7.9 bits por caracter por lo cual tiene una alta densidad en informacion y se podria clasificar como una generacion de direccion-llave con una aletoriedad importante
* Chi-cuadrado de 2893.76 indicando que con una muestra de 500 hay una probabilidad de 0% de generar un valor igual, lo cual podria demostrar su capacidad de generar valores aleatorios.



* La media aritmetica resulta de sumar todos los bytes y dividirlos por el tamanio del fichero. Para este caso la media tiene un valor de 130 lo cual determina que la data esta cercana a ser aleatoria.
* 3.06 para la medida de monte carlo pi que se acerca a PI con una diferencia de 0.08 que establece una aleatoriedad suficiente.
* Para la correlacion serial obtenemos 0.02 el cual es un valor cercano a 0 y con el cual podriamos concluir que la generacion aleatoria de tuplas direccion-llave cuenta con una aletoriedad notable dentro de todas las medidas aca tomadas.

Ejecutando ent con una billetera generada con encriptacion tipo BIP38 tenemos los siguientes valores:

Caso 1.1:



Exceptuando la serial correlation podemos decir que la aleatoriedad de estas tuplas se comporta mejor que generar la billetera sin encriptacion BIP38 debido a que la entropia es ligeramente mayor, chi-cuadrado se mantiene en 0%, la media aritmetica tiene menor distancia respecto a 127.5, y monte carlo pi esta mas cercano al valor de PI.

Parte 2

Se genera un fichero con una secuencia de numeros pseudo aleatoria con openssl

El fichero contiene una muestra con 312500 numeros

Caso 2:



En este caso entropia nos indica una densidad mayor al anterior ejemplo. El chi-cuadrado seniala una probabilidad del 0.99% de generar una discrepancia, pero en caso contrario la media aritmetica tiene un valor muy cercano a 127,5 el cual considera una aleatoriedad mejor al caso 1 y junto con monte carlo pi y la correlacion serial que dio mejores resultados.

Por tanto y si dejaramos de lado la metrica de chi-square podriamos decir que el caso 2 da una mayor aleatoriedad que el caso 1

Utilizando el generador de hotbits:

El fichero contiene una muestra con 128 numeros

Caso 2.1:



Este caso cuenta con la menor entropia de los demas, lo que seniala una menor aleatoriedad, pero le fue mejor con chi-cuadrado con una probabilidad del 0% de generar discrepancia. Para la media aritmetica aunque mejor que los casos 1, esta por debajo del caso 2 y cuenta con la metrica de monte carlo mas baja.

Caso 2.2:



Conclusiones



Considerando que todas las metricas tienen el mismo peso de importancia para evaluar la aleatoriedad de un algoritmo, podriamos concluir que la solucion que mas representa aletoriedad dentro de las demas es la de openssl debido a que establece el mayor numero de metricas con mayor valor. Por otro lado el algoritmo de generacion de billeteras bitcoin (Caso 1) representa el mas deficiente dentro de los 5 casos.