Universidad Politécnica de Madrid:
Escuela Técnica Superior de Ingenieros Informáticos

Seguridad de las Tecnologías de la Información
Estudio del Efecto Avalancha de la función Hash MD2

Daniel Alejandro Lobo - 170500

23 de Octubre de 2017

Introducción

Se conoce como efecto Avalancha a la propiedad deseada de una función criptográfica donde pequeños cambios sobre los argumentos de entrada provocan grandes cambios en el resultado de la función. El algoritmo a estudiar en este experimento, la función hash criptográfica MD2 definida en RFC 1319, cuenta con un sólo argumento de entrada que es la cadena a la cual se le calculará el hash. El output o *digest* es una cadena de 128 bits.

Implementación:

El experimento fue realizado utilizando el lenguaje Python ya que contaba con las herramientas y librerías adecuadas para el trabajo a realizar: *"The right tool for the right job"*. Se utilizaron las librerías: Bitstring, Crypto Hash, MD2, Distance, Datetime, Matplotlib, Statistics, Random. Para garantizar el correcto funcionamiento de la primitiva criptográfica se probó la función con vectores de prueba. Los vectores de prueba se corren al iniciar el programa principal y en el código se pueden ver en el método *test_vectors()*

Estructura de una corrida:

Lo primero a hacer en una corrida del experimento es generar las cadenas de bits a cifrar. Se genera un número entero aleatorio de 128 bits (podría ser más grande también). Lo importantes es que sea de cómo mínimo de una longitud de 64 bits, según FIPS 1280-4 SHS 08/2015. Luego se repite mil veces el siguiente procedimiento:

1. Se elige al azar uno de los 128 bits para cambiar y se genera el entero con el bit cambiado. Por lo tanto, tengo la cadena original de 128 bits y también otra cadena que sólo varía en un bit con la cadena elegida originalmente. La distancia de Hamming entre ambas es, obviamente, uno.
2. Luego, antes de calcular el hash se deben convertir los enteros en su representación binaria para lo cual se utiliza la librería bitstring, más específicamente, BitArray.
3. Una vez calculados los hashes de las cadenas, se mide la Distancia de Hamming entre los dos hashes y se devuelve el valor para ser acumulado.

Este procedimiento se hace 3 veces, corriendo en total unas tres mil muestras a partir de la generación de 3 números enteros aleatorios de 128 bits. El porqué de este tamaño de muestra se debe a que, a partir de 100 muestras aproximadamente, el valor de la media, varianza y desviación estándar eran muy semejantes al de 1000 muestras, por ejemplo. Y la razón del número 3 es simplemente para tener 3 corridas diferentes entre sí y ver que, efectivamente, se obtienen resultados semejantes.

Significación del experimento y número de corridas a realizar:

Para este experimento se decidió conseguir un resultado con significancia de 92.5%. Si en alguno de los lotes el error de las medidas es mayor a 7.5% con respecto a los valores ideales, se aborta esa iteración y no suma al resultado final. Para conseguir el nivel de significancia deseado se hicieron tres lotes de 1000 corridas, como fue mencionado en el apartado anterior.

Resultados experimentales:

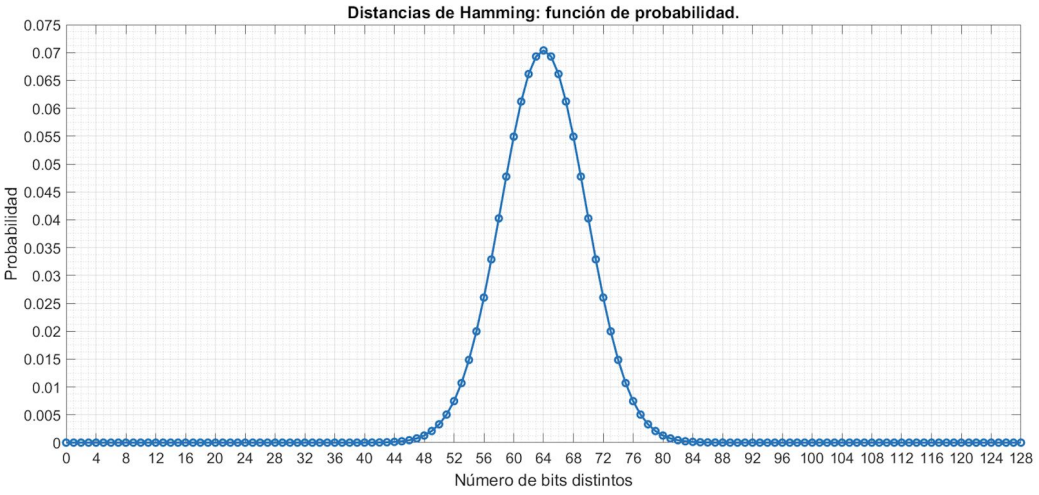
Para los resultados finales se tomaron los datos de los tres lotes y se interpretan como un solo experimento. Sobre estos datos se calculó la media, la mediana, la moda, la varianza y la desviación estándar de las distancias obtenidas. Para ello se utilizaron las funciones estadísticas incluidas en la librería *statistics.* Se puede ver en el histograma las frecuencias de aparición de cada distancia.

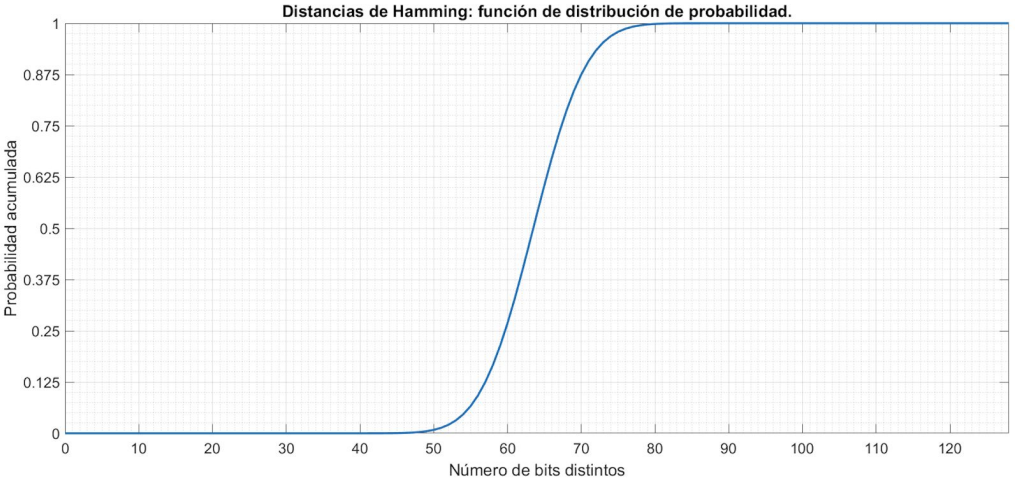
A continuación se muestran las medidas estadísticas obtenidas durante experimento:

- Media: 63.9486666667
- Moda: 64
- Mediana: 64.0
- Desviación: 2.69835602988
- Varianza: 7.28112526398

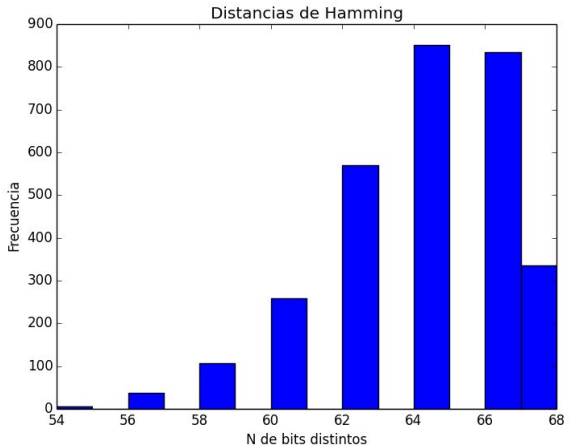
Comparación con un Oráculo Aleatorio:

Si la función criptográfica se comportara como un Oráculo aleatorio, las dos cadenas de bits resultantes serían totalmente aleatorias. De ser este el caso, la posibilidad de que un determinado bit sea distinto en las dos cadenas es de 0.5. Con esto en mente se puede ver como que la Distancia de Hamming entre dos cadenas es equivalente a realizar 128 elecciones de bits, con una probabilidad del 0.5 de que el bit sea distinto. Este comportamiento ideal es el descrito por una distribución binomial con parámetros n = 128 y p = 0.5. Por consiguiente la media de la distribución sería 64 (n * p) y la varianza 32 (n * p * (p-1)). De todas maneras, es importante recordar que según la tesis de Church–Turing, ninguna función computable por un algoritmo finito genera un verdadero Oráculo Aleatorio. Estos comportamientos "ideales" se pueden observar en los siguientes gráficos. Los cuáles son generados con el código de MatLab *binomial.m* que se puede encontrar en el material adjunto.





Sin embargo el histograma obtenido, demuestra el siguiente comportamiento.



Conclusiones

Como lo evidencian los resultados mostrados anteriormente, la distancia de Hamming entre los hashes de dos cadenas prácticamente idénticas (se diferencian por un sólo bit) sigue una distribución asimétrica en el histograma, y también lo hace la varianza obtenida con respecto a la esperada. Lo que hace que el comportamiento sea diferente al esperado de un Oráculo Aleatorio.

Por eso, se puede afirmar que los resultados de este experimento nos llevan a afirmar que la función hash criptográfica MD2 muestra un fuerte efecto avalancha pero no lo suficientemente similar al de un Oráculo Aleatorio.

Referencias:

- https://en.wikipedia.org/wiki/Avalanche_effect
- https://en.wikipedia.org/wiki/Correlation_and_dependence
- https://en.wikipedia.org/wiki/Hamming_distance
- https://en.wikipedia.org/wiki/Random_oracle
- https://en.wikipedia.org/wiki/Histogram
- https://en.wikipedia.org/wiki/Test_vector
- https://docs.python.org/3/library/hashlib.html
- https://pypi.python.org/pypi/Distance/