

INFLUENCIA DE LAS CONTRASEÑAS EN LA SEGURIDAD DE LOS ALGORITMOS.

Autor: José Manuel Busto López

Fecha: Marzo 2007

Copyright © 2007 José Manuel Busto López

Licencia:

Creative Commons Attribution-ShareAlike 3.0 Unported (<http://creativecommons.org/licenses/by-sa/3.0/legalcode>)

Tu eres libre de:

- copiar, distribuir, comunicar y ejecutar públicamente la obra
- hacer obras derivadas

Bajo las siguientes condiciones:

- **Atribución.** Debes reconocer y citar la obra de la forma especificada por el autor o el licenciante.
- **Licenciar Igual.** Si transforma o modifica esta obra para crear una obra derivada, sólo puede distribuir la obra resultante bajo la misma licencia, una de similar o una de compatible
- Al reutilizar o distribuir la obra, tienes que dejar bien claro los términos de la licencia de esta obra.
- Alguna de estas condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de autor
- Nada en esta licencia menoscaba o restringe los derechos morales del autor.

¿Como empezar?

Las contraseñas se han vuelto hoy en día en la parte más vulnerable de un algoritmo de cifrado. A medida que evoluciona la informática la seguridad de la contraseña disminuye debido a que se pueden procesar más contraseñas en el mismo tiempo.

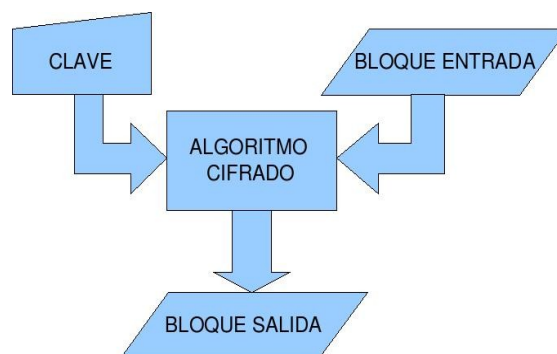
Existen diversas teorías sobre como debería ser una contraseña, pero hasta ahora ninguna ha demostrado que sea segura. Como ejemplo:

- Una contraseña con sólo letras minúsculas no es segura.
- Una contraseña de 8 o más caracteres (letras mayúsculas, minúsculas y números) es segura.

En este artículo se pretende demostrar que estos axiomas no son totalmente correctos y demostrar como se puede saber si una contraseña es o no segura desde un punto de vista matemático.

¿Qué es un algoritmo de cifrado?

Un algoritmo de cifrado es un proceso por el cual un bloque de información pasa para ser ofuscada mediante una clave secreta (figura 1).



Se distinguen 2 tipos de algoritmos de cifrado en función de las claves que usan:

- Algoritmos de cifrado simétricos: En estos algoritmos la clave de cifrado y de descifrado es la misma.
- Algoritmos de cifrado asimétricos: La clave de cifrado es distinta de la clave de descifrado.

De los primeros, los más importantes hoy en día son los algoritmos Rijndael (AES) y Serpent. En cuanto a los segundos, el algoritmo RSA es el que más se usa.

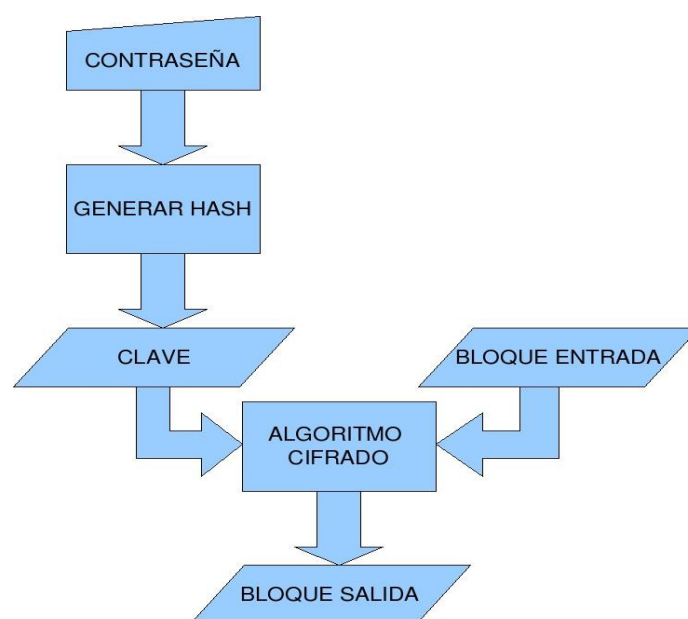
En función de como operan se pueden clasificar en:

- Algoritmos de cifrado de bloques: Estos algoritmos dividen la información en bloques de longitud determinada y luego los cifran.
- Algoritmos de cifrado de streams: Estos algoritmos cifran la información bit a bit.

¿Que es una contraseña?

Una contraseña es un conjunto de caracteres secretos de los cuales se obtiene la clave para el algoritmo de cifrado.

Todos los algoritmos de cifrado dependen de una clave para cifrar. Esta clave debe tener una longitud fija, por lo que los programas que permiten cifrar datos deben traducir la contraseña que se le introduce, la cual puede ser de cualquier longitud, a una clave de longitud determinada. Generalmente, lo que hacen estos programas es utilizar una función de derivación para sacar un resumen criptográfico o *hash* de la contraseña. En RFC 2898 se definen varios métodos para realizar la función de derivación. Estos métodos se basan en aplicar 1000 o más veces un algoritmo SHA o MD5 a la contraseña. El algoritmo SHA en su versión de 128 y 192 bits no es seguro debido a que existen colisiones de las que se pueden obtener la clave correcta sin necesidad de probar todas las claves posibles. Otros algoritmos en auge son Tiger/192, Whirlpool o SHA512. La figura 2 muestra un poco más completo un diagrama de un programa de cifrado de datos:

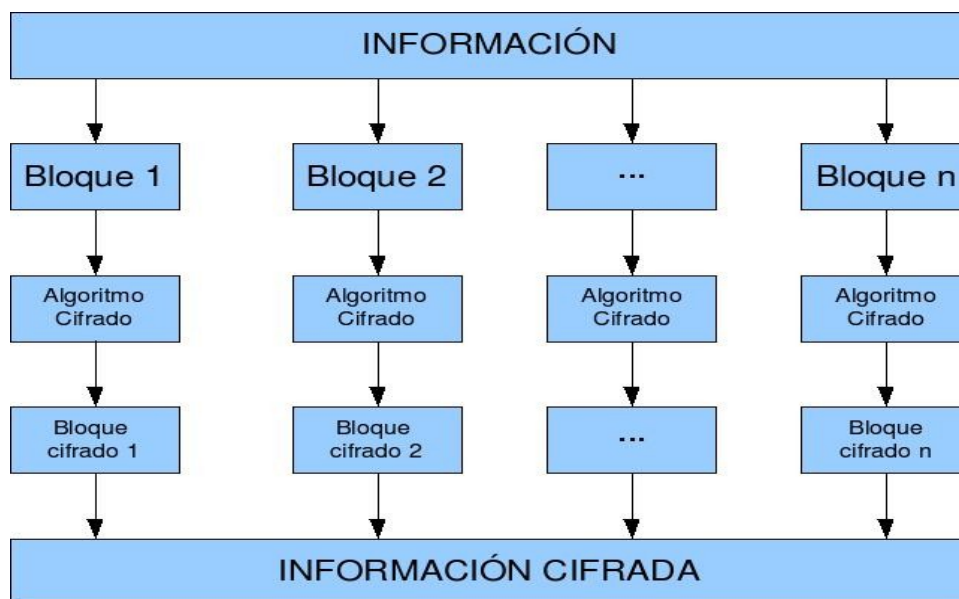


¿Como funciona un algoritmo de cifrado?

Existen variados algoritmos de cifrado . Entre los más conocidos se encuentran los siguientes:

- Rijndael / AES (Advanced Encryption Standard)
- Blowfish
- IDEA
- DES (Data Encryption Standard)
- Serpent
- RSA

Generalmente todos estos algoritmos realizan un cifrado por bloques, o sea, dividen la información a cifrar en bloques de longitud determinada, la cifran y luego forman el criptograma.



¿Que algoritmo de cifrado escojo?

Existen varias cosas a tener en cuenta a la hora de escoger el algoritmo de cifrado que protegerá nuestra información:

- Vulnerabilidad: Muchos algoritmos de cifrado son vulnerables a ataques matemáticos que reducen el número de claves que hay que probar para localizar la buena. Escoger un algoritmo que sea invulnerable a todos los ataques conocidos es siempre una buena idea. En caso de que no podamos escoger uno invulnerable, siempre deberemos seleccionar el algoritmo en funcion

del número real de claves que hay que probar hasta localizar la buena. No debemos cegarnos porque tenga una clave más larga, ya que quizás es más fácil de recuperar que en otro algoritmo con clave más corta, lo cual nos lleva al siguiente punto.

- Longitud de la clave: La longitud de la clave es el número de bits que la componen. Según esta longitud tendremos más o menos claves distintas:
 - 64 bits: $1.84467440737 \times 10^{19}$ claves distintas
 - 128 bits: $3,40282366920 \times 10^{38}$ claves distintas
 - 256 bits: $1,15792089237 \times 10^{77}$ claves distintas
 - 1024 bits: $1,7976931348 \times 10^{308}$ claves distintas

Estas claves distintas son todas las que serían necesarias probar en caso de que alguien intente descifrar el contenido mediante un ataque por fuerza bruta exhaustivo. Generalmente, debemos tender a considerar que el número real de claves que debe probar es la mitad de las claves distintas, debido a que no sabemos si empezará a probar claves por el principio, el medio o el final.

- Velocidad de cifrado/descifrado: Este aspecto no suele influir demasiado en la seguridad del algoritmo, pero es interesante tenerlo en cuenta a la hora de seleccionarlo. A más velocidad de cifrado, mayor será la cantidad de datos que podemos enviar en el mismo tiempo. También deberemos tener en cuenta la velocidad de descifrado. En algunos algoritmos, RSA por ejemplo, las velocidades de cifrado y descifrado pueden llegar a ser muy distintas y encontrarnos con que tardamos 10 minutos en cifrar un archivo grande y 30 minutos en descifrarlo (o viceversa). Lo ideal es que se tarde aproximadamente el mismo tiempo en cifrar que en descifrar.

¿Y que contraseña pongo?

Ahora es cuando nos metemos en el contenido concreto de este artículo. La contraseña suele ser la parte más vulnerable de un archivo cifrado. Existen diferentes ataques aplicables a una contraseña, entre ellos los siguientes:

- Ataque por diccionario: Se denomina así a los ataques que se realizan a partir de cadenas de caracteres en ficheros (diccionarios). Estos diccionarios pueden llegar a contener más de 1000000 palabras usadas normalmente en cada idioma.

- Ataque por diccionario con polimorfismos: Se utiliza un diccionario para seleccionar las contraseñas a usar y se aplican posibles modificaciones que se le pudieron hacer, por ejemplo, para el español estas pueden ser poner solo la primera letra en mayúscula, hacer plurales o poner otras letras que suenen bien en la palabra seleccionada (carton => caston => ca5t0n=>canton).
- Ataque por máscara: Las máscaras indican las partes de las contraseñas que pueden llevar determinados caracteres, por ejemplo, empezar por un número, tener un espacio o finalizar con el carácter “n”. Para realizar estos ataques es necesario conocer el formato de la contraseña utilizada. Si no se conoce es casi imposible que este ataque consiga localizarla.
- Ataque exhaustivo por fuerza bruta: Este ataque consiste en probar todas las contraseñas posibles en función del juego de caracteres que se crea que usa la contraseña. Este ataque es el que más tiempo lleva realizarlo, pero, en el caso de que se usen todos los caracteres posibles para la búsqueda, siempre recupera la contraseña.
- Ataque por fuerza bruta inteligente: Este ataque consiste en probar “casi” todas las contraseñas posibles, probando solo las que puedan parecer contraseñas factibles. Estos ataques reducen entre un 10 y un 20% el número de contraseñas a probar respecto a un ataque exhaustivo, consiguiendo recuperar aproximadamente un 90% de las mismas.

A la vista de esto, ya podemos deducir lo siguiente:

- Para evitar los ataques por diccionario deberemos escoger cadenas de caracteres que formen palabras inexistentes en el idioma y que no puedan formar, con simples modificaciones, palabras que si están presentes en él.
- Siempre deberemos usar contraseñas que no usen ningún tipo de máscara.

Aun así, poner contraseñas de este tipo no va a evitar que nos la puedan localizar por fuerza bruta.

¿Hay alguna forma de librarse de los ataques por fuerza bruta?

Los ataques por fuerza bruta siempre serán capaces de localizar nuestra contraseña, pero conociendo los equipos disponibles podemos hacerlo casi imposible.

Antes de demostrar como podemos crear una contraseña segura, deberemos definir este término.

- Contraseña segura: Aquella contraseña que se tarda en recuperar más de 1 año.

Generalmente se utiliza un periodo de 1 año para la recuperación de la contraseña, pues, en función de los recursos del atacante, no se suele dedicar tanto tiempo en intentar localizar una contraseña. Si queremos más seguridad, se puede incrementar el tiempo que se tarda en recuperar a 5 años.

Cuando se intenta recuperar una contraseña por fuerza bruta, la velocidad con que se recupere depende de varios factores por lo que no existe un método exacto para saber cuanto se tardará en localizar una contraseña. Habrá entonces que hacer una estimación poniendose siempre en el peor caso posible.

La fórmula general para el cálculo es la siguiente:

$$Tiempo(seg) = \frac{base^{longitud(contraseñas)}}{numero(contraseñas/seg)}$$

donde:

tiempo: N° de segundos que tardará un atacante en localizar la contraseña

base: N° de elementos del juego de caracteres utilizado

longitud: Longitud de la contraseña

numero: Contraseñas por segundo que recuperará un atacante

La informática está evolucionando constantemente. Cada poco tiempo sale un procesador más potente, por lo que deberemos tener en cuenta la evolución de la velocidad de los procesadores a la hora de poner nuestra contraseña.

Se conoce como Ley de Moore a una teoría que lanzó Gordon Moore, en el año 1965, que dice que cada año y medio se duplica el nivel de integración de los circuitos. Esto quiere decir que a cada año y medio se hacen procesadores que, ocupando lo mismo, poseen el doble de transistores. Si miramos la evolución de la potencia de los procesadores, veremos que esta ley también se cumple en su rendimiento, por lo que podemos considerar que cada año y medio se duplicarán el número de contraseñas que se pueden recuperar.

Modificando la fórmula anterior:

$$Tiempo(años) = \frac{\ln\left(\frac{3 \cdot (base^{longitud})(contraseñas)}{4 \cdot numero(contraseñas/año)}\right)}{\ln 2}$$

donde

Tiempo: Distancia en años entre lo que un atacante empieza la búsqueda y la encuentra

base: N° de elementos del juego de caracteres utilizado

longitud: Longitud de la contraseña

numero: Contraseñas por año que recuperará un atacante al inicio

Como ya se ha dicho, el tiempo en recuperar la contraseña depende de varios factores. Si aplicamos a nuestra contraseña una función de derivación PBKDF2, tal como se define en RFC 2898, para conseguir la clave de cifrado estaremos realizando un mínimo de 200000 operaciones. Si tomamos este dato como número de operaciones que tendrá que realizar un atacante para recuperar 1 contraseña, la fórmula queda como sigue:

$$Tiempo = \frac{3 \cdot (\ln(base^{longitud}) - \ln(157,68 \cdot vProc))}{\ln(4)}$$

siendo:

vProc: Operaciones por segundo que puede realizar el atacante (flops).

Si una contraseña segura se debe tardar 5 años en recuperar para evitar las posibles fluctuaciones informáticas y solo debemos considerar la mitad de las contraseñas posibles podemos simplificar la ecuación anterior y calcular un índice recuperación de contraseñas (IRC) para un ordenador cualquiera de la siguiente forma:

$$Longitud = \frac{IRC}{\ln(base)}$$

$$IRC = 8,06 + \ln(vProc)$$

El ordenador más potente del mundo en Febrero de 2006 es BlueGene/L eServer, capaz de alcanzar 280600 Gflops. (280,6 x 10¹² flops), por lo que el IRC será de 41,33.

Aplicando la fórmula a las bases más importantes:

BASE	LONGITUD
Minúsculas (inglés) => 26	13 caracteres
Minúsculas y mayúsculas => 52	11 caracteres
Minúsculas, mayúsculas y números => 62	10 caracteres
Caracteres imprimibles => 108	9 caracteres
Caracteres ASCII => 256	8 caracteres

Aplicando la misma fórmula al ordenador más potente de España, Marenostrum, capaz de alcanzar 62630 Gflops ($62,63 \times 10^{12}$ flops) obtenemos un IRC de 39,82, por lo que las bases más importantes quedan así:

BASE	LONGITUD
Minúsculas (inglés) => 26	13 caracteres
Minúsculas y mayúsculas => 52	11 caracteres
Minúsculas, mayúsculas y números => 62	10 caracteres
Caracteres imprimibles => 108	9 caracteres
Caracteres ASCII => 256	8 caracteres

Como se ve, redondeando no hay diferencia entre los 2 equipos. La diferencia de tiempo en recuperar la contraseña entre uno y otro, considerando los posibles aumentos de potencia, será inferior a 1 año.

En el caso de que utilizemos una función de derivación distinta, el tiempo de recuperación variará. Si consideramos que no se utiliza ninguna función de derivación, y, a cada ciclo de reloj se puede recuperar 1 contraseña (hoy en día cosa casi imposible), el IRC seguirá la siguiente fórmula:

$$IRC = 20,27 + \ln(vProc)$$

En la siguiente tabla se muestra cuáles son las longitudes necesarias para los equipos anteriores y para la suma de potencias de los 500 ordenadores más potentes del mundo (3521,61 Tflops):

<i>BASE</i>	<i>3521,61 Tflops IRC=56,07</i>	<i>BlueGene IRC=53,53</i>	<i>Marenostrum IRC=52,04</i>
Minúsculas (inglés) => 26	18 caracteres	17 caracteres	16 caracteres
Minúsculas y mayúsculas => 52	15 caracteres	14 caracteres	14 caracteres
Minúsculas, mayúsculas y números => 62	14 caracteres	13 caracteres	12 caracteres
Caracteres imprimibles => 108	12 caracteres	12 caracteres	12 caracteres
Caracteres ASCII => 256	11 caracteres	10 caracteres	10 caracteres

La siguiente tabla muestra lo anterior para otros equipos más “normalitos”. El tercer equipo es el número 500 del mundo por potencia de cálculo:

<i>BASE</i>	<i>1800 Mfps IRC=42,62</i>	<i>3200 Mfps IRC=42,15</i>	<i>2736 Gfps IRC=48,91</i>
Minúsculas (inglés) => 26	14 caracteres	14 caracteres	15 caracteres
Minúsculas y mayúsculas => 52	11 caracteres	11 caracteres	13 caracteres
Minúsculas, mayúsculas y números => 62	11 caracteres	11 caracteres	12 caracteres
Caracteres imprimibles => 108	10 caracteres	10 caracteres	11 caracteres
Caracteres ASCII => 256	8 caracteres	8 caracteres	9 caracteres

Queda explicar que la fórmula del IRC contiene 2 operandos. Si la velocidad del procesador es de 1 flop (o sea, que realiza 1 sola operación en punto flotante por segundo), la longitud de la contraseña necesaria para un juego de caracteres de letras minúsculas (26) sería para el primer caso de 3 caracteres y para el segundo de 6 caracteres.

¿Y pasa con las cuentas de Internet?

Todo lo que he explicado hasta ahora sirve para nuestros sistemas de ficheros, pero para las cuentas de internet (correo electrónico, blogs, etc..) no sirve la fórmula, puesto que ya no dependemos de lo que tarda el ordenador del atacante en localizar la contraseña, sino que dependemos de la velocidad de la conexión. Por ejemplo, si atacan un servidor de correo desde un equipo remoto, el atacante debe esperar a que el servidor valide la contraseña enviada y le devuelva información sobre su validez. Esto hace que atacar estas cuentas sea mucho más lento que atacar un sistema de archivos,

incluso sin contar con otros sistemas de seguridad que implementan estos servidores como CAPTCHA o bloqueos de tiempo en el acceso a la cuenta (independientemente de si última contraseña es correcta) por introducir un determinado número de contraseñas incorrectas. Si utilizamos cualquiera de las anteriores longitudes, no es probable que un atacante consiga localizar nuestra contraseña.

¿Y un pequeño resumen?

Para finalizar, resumiré un poco todo esto para el que no tenga ganas de leer.

- Contraseña segura: Aquella contraseña que se tarda en recuperar más de 1 año
- IRC= Índice de recuperación de contraseñas
- $Longitud = \frac{IRC}{\ln(base)}$
- Para recuperarla en 1 año: $IRC = 17,26 + \ln(vProc)$
- Para recuperarla en 5 años: $IRC = 20,27 + \ln(vProc)$
- vProc maxima: visitad <http://top500.org> para ver cual es el más rápido del mundo (o de vuestro país).

Bibliografía:

- RFC 2898 - PKCS #5: Password-Based Cryptography Specification Version 2.0, B. Kaliski, September 2000
- <http://top500.org>
- Fips-197 Specification of Advanced Encryption Standard (AES) , November 26, 2001