

# IMPACTO DE RECIENTES ATAQUES DE COLISIONES CONTRA FUNCIONES DE HASHING DE USO CORRIENTE

Hugo Daniel Scolnik<sup>1</sup>, Juan Pedro Hecht<sup>2</sup>

<sup>1</sup> Director de CERTISUR y CEO de FIRMAS DIGITALES ([scolnik@fd.com.ar](mailto:scolnik@fd.com.ar))

<sup>2</sup> CTO de FIRMAS DIGITALES ([hecht@fd.com.ar](mailto:hecht@fd.com.ar))

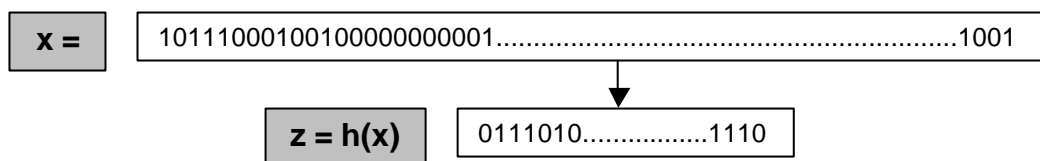
*(Versión revisada 2. - 5 de Setiembre de 2004)*

*En este trabajo se analiza el impacto generado por los recientes ataques de colisiones contra las principales funciones de hashing empleadas hoy día en innumerables aplicaciones de control de integridad y autenticación. Se intenta dilucidar hasta qué punto estos avances criptoanalíticos pueden llegar a afectar la criptografía actualmente empleada.*

## Introducción

Recientemente y en el marco de la conferencia internacional de criptología CRIPTO'04, se han presentado informes acerca de *ataques de colisión* contra una serie de *funciones de hashing* (1)(2). Estas funciones representan una familia de algoritmos destinados al control de la integridad de datos, de amplia difusión en los medios financieros y en la mecánica vinculada a la Tecnología PKI (*Public Key Infrastructure*). Dichos ataques se conocen genéricamente como *colisiones diferenciales de Chabaud-Joux* (3), y corresponden a versiones potenciadas del trabajo original presentado en 1998. Para llegar a apreciar el impacto de estos descubrimientos sobre la criptografía práctica que se usa en la actualidad, debemos analizar técnicamente el tema. Comencemos con algunas definiciones (4):

**Función de Hashing:** Es una función  $h: X \Rightarrow Z$  que transforma una cadena binaria o mensaje de longitud arbitraria ( $x$ ) en otra cadena binaria ( $z$ ) de longitud constante ( $n$ ) llamada *digesto* (generalmente de 128, 160, 256 o 512 bits). Las funciones más difundidas a la fecha son MD5, SHA, SHA1, RIPEMD128, RIPEMD160 y entre estas se cuentan las hoy en día cuestionadas. En el Cuadro N° 1 se grafica la función de hashing.



Cuadro N° 1: función de hashing transformando una cadena de megabits en un digesto de 128 bits. Queda claro que potencialmente hay 2<sup>128</sup> digestos posibles.

Debe quedar en claro que es inevitable que esta clase de funciones que transforman cadenas de longitud arbitraria (las *preimágenes*) en cadenas de longitud fija (las *imágenes*), más de una preimagen dará lugar a una misma imagen (en caso contrario existiría al menos una función biunívoca entre conjuntos de cardinalidades distintas). Esa clase de conflictos se conoce como “*colisiones*” entre valores de hashing. Para entender los procesos que se describirán a continuación resulta conveniente aclarar que una operación es *computacionalmente no factible* si no existe un algoritmo de tiempo polinomial para resolverla (complejidad Clase P), o sea esa operación es prácticamente irrealizable con los recursos computacionales disponibles actualmente (complejidad Clase NP). Se definen cinco tipos de resistencias a los ataques criptoanalíticos contra las funciones hashing:

**1. Función de hashing resistente a preimágenes:** Una *función de hashing* es resistente a preimágenes si dado el digesto  $z$  es *computacionalmente no factible* hallar algún mensaje  $x$  tal que  $h(x) = z$ . La complejidad de este ataque contra un hashing de  $n$ -bits es  $O(2^n)$ .

**2. Función de hashing resistente a segundas preimágenes:** Una *función de hashing* es resistente a segundas preimágenes si dado un mensaje  $x$  es *computacionalmente imposible* hallar un mensaje  $x' \neq x$  tal que  $h(x) = h(x')$ . La complejidad de este ataque contra un hashing de  $n$ -bits es  $O(2^n)$ .

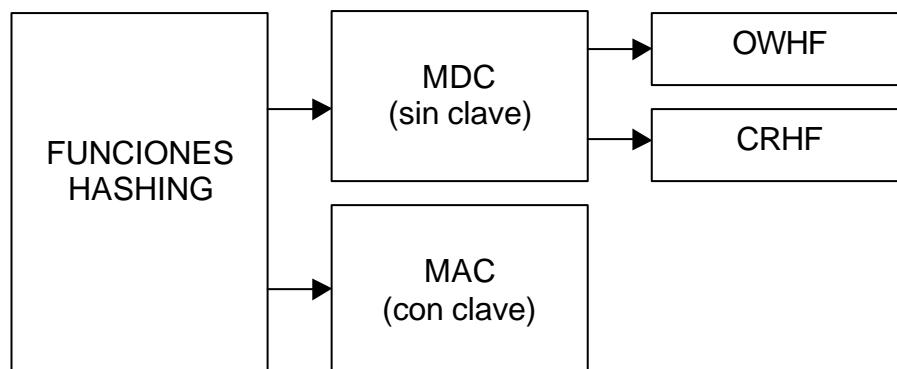
**3. Función de hashing resistente a colisiones:** Una *función de hashing* es resistente a colisiones si es *computacionalmente no factible* hallar un par de mensajes distintos  $x, x'$  tal que  $h(x) = h(x')$ . Hay dos variantes: con IV's (*initialization vectors*) fijos (la habitual) o de IV aleatorio en la cual uno de esos vectores puede variar libremente. Salvo que se indique lo contrario, nos referiremos sólo a la variante con IV's fijos. La complejidad de este ataque (IV fijo) contra un hashing de  $n$ -bits es  $O(2^{n/2})$ .

**4. Función de hashing resistente a seudocolisiones:** Una *función de hashing* es resistente a seudocolisiones si es *computacionalmente no factible* hallar un par de mensajes distintos  $x, x'$  tal que  $h(x) = h(x')$  en las cuales se puedan usar distintos IV's para ambos mensajes  $x, x'$ .

**5. Función de hashing resistente a seudopreimágenes:** Una *función de hashing* es resistente a seudopreimágenes si dado el digesto  $z$  es *computacionalmente no factible* hallar algún mensaje  $x$  y algún IV, tal que  $h(x) = z$ .

Las cinco formas de resistencia no son equivalentes. Por ejemplo, la condición 2. no garantiza 1. ni la 1. garantiza 2. Además la condición 3. no garantiza 1. Cabe destacar que los ataques de *seudocolisiones* y *seudopreimágenes* tienen relativamente poca importancia práctica (4), pero afectan la capacidad de que el algoritmo en cuestión pueda certificarse como *estándar*. Las resistencias más críticas para la seguridad son las 1. y 2.

Una *función de hashing* que cumpla 1. y 2. se denomina *función de hashing* de una vía o función débil de una vía (**OWHF**) si cumple 2. y 3. (y eventualmente 1.) recibe el nombre de resistente a colisiones o función fuerte de una vía (**CRHF**). Además ambas categorías se agrupan genéricamente como *códigos de detección de modificaciones* (**MDC's**) (usados sólo para control de integridad), otra categoría especial la forma una combinación de las funciones de hashing con claves simétricas y que se conocen como *códigos de autenticación de mensajes* (**MAC's**) (una especie primitiva de firma digital, usada para asociar identidad con datos). En el Cuadro N° 2 se presenta la clasificación de las funciones de hashing.



Cuadro N° 2: Clasificación de las funciones de hashing en códigos de detección de modificaciones (MDC) y códigos de autenticación de mensajes (MAC). A su vez los primeros se clasifican en funciones (débiles) de una vía (OWHF) y funciones resistentes a colisiones (CHRF) también llamadas funciones fuertes de una vía.

Para distintas aplicaciones se requieren distintos tipos de funciones hashing, como podremos observar en el Cuadro N° 3:

Tipo de Resistencia requerida para las aplicaciones siguientes	1. Preimagen	2. Segunda Preimagen	3. Colisiones
MDC + firma digital	REQUIERE	REQUIERE	REQUIERE
MDC + canal seguro		REQUIERE	REQUIERE
MDC + BD contraseñas	REQUIERE		
MAC + clave secreta	REQUIERE	REQUIERE	REQUIERE
MAC + clave pública		REQUIERE	

Cuadro N° 3: Clasificación de las resistencias requeridas para las funciones de hashing de acuerdo a la aplicación en la cual se las emplea.

En el momento de diseñar las funciones de hashing se busca que posean las resistencias 1. 2. 3. 4. y 5., y los atacantes buscan vulnerarlas. Si una función de hashing posee un digesto de  $n$ -bits, la probabilidad (ideal) de hallar preimágenes o segundas preimágenes a partir de una imagen dada, debe ser  $2^{-n}$ , es decir la misma de ubicarla al azar dentro de su población. De la misma manera, elegir dos preimágenes y que ambas posean la misma imagen debería tener una probabilidad ideal de  $2^{-n^2}$  (esta probabilidad, raíz cuadrada del cardinal del espacio de *hashings*, es deducible en base al algoritmo del *ataque del cumpleaños*) (4). Si por algún motivo un atacante consigue hallar preimágenes o colisiones con una probabilidad mayor que las predichas por el azar puro, ello indicaría una debilidad en las funciones de hashing. En el Cuadro N° 4 se resumen las metas del diseño y del ataque de las distintas funciones de hash.

Tipo de hashing	Metas de diseño	Fuerza Ideal	Metas de ataque
OWHF	Resistencia de Preimagen Resistencia 2° Preimagen	$2^n$ $2^n$	Producir Preimágenes Producir 2° Preimagen
CRHF	Resistencia a colisiones	$2^{n/2}$	Producir colisiones
MAC	No recuperación de clave de longitud $t$ y resistencia de cómputo de mensaje	$\min(2^t, 2^n)$	Deducir la clave o producir un nuevo mensaje legal

Cuadro N° 4: Metas en el diseño y en el ataque de las funciones de hashing. Se indican las fuerzas ideales de dichos algoritmos y que corresponden a considerar que el mejor ataque para hallar preimágenes y colisiones corresponde a la fuerza bruta o exploración secuencial de todo el espacio de búsqueda.

Pasemos ahora al mundo real. Los algoritmos MD4 (año 1990), MD5 (año 1991) (desarrollados por Ron Rivest, co-desarrollador del criptosistema de clave pública RSA) fueron los primeros de su serie. Luego surgieron como mejoras el SHA-0 (*Secure Hash Algorithm*, año 1993), SHA-1 (año 1995) y SHA-2 (año 2004) y que forman una familia de algoritmos vinculados por su estructura de *padding*, *parsing*, *message scheduling* y *register update* en una serie de *rounds* (4).

El algoritmo SHA-1 (SHA-160) y la familia informalmente conocida como SHA-2 (*estándars* SHA-224, SHA-256, SHA-384 y SHA-512) surgen para eliminar debilidades halladas en el precursor SHA-0 (5). Estas funciones de hashing han sido y son las recomendadas por la Administración Federal de EEUU (SHS: Secure Hash Standard) (5) y por reflejo en gran parte del resto del mundo. El SHA-224 es idéntico al SHA-256 excepto que varían los IV's (*initialization vectors*) y que se trunca el digesto a los 224 bits de la derecha, el SHA-384 es idéntico al SHA-512 con las mismas salvedades del caso anterior. El SHA-224/256 está basado en palabras de 32 bits y el SHA-384/512 en palabras de 64 bits.

SHA-1 es actualmente el *estándar* mundial *de facto* de las funciones de hashing. En la tecnología PKI se usan extensivamente MD5 y SHA1; por ejemplo forman parte constante del protocolo SSL v3. El MD5 y SHA-1 son usados por sistemas UNIX para almacenar las contraseñas de los usuarios en forma segura. A su vez la familia RIPEMD128 y RIPEMD160, desarrollados para el proyecto RIPE (RACE Integrity Primitives Evaluation) en 1992 y 1996, se hallan muy distribuidos en Europa. En nuestro País, el MD5 es el principal algoritmo de hashing usado por el sistema bancario, y en general, a nivel mundial, MD5 y SHA-1 son las funciones más empleadas hoy día en la firma digital de documentos para todo tipo de transacciones y workflow *off-line* y *on-line*. En el Cuadro N° 5 resumimos las principales características de las funciones de hashing más difundidas actualmente.

FUNCION	bits	Rounds	Velocidad Relativa ( <sup>1</sup> ) <i>estimado</i>	Fuerza Preimagen	Fuerza contra ataque de Colisión( <sup>2</sup> ) ( <sup>2</sup> ) <i>ataque cumpleaños</i>
MD4	128	48	1.00	$2^{128}$	(**) $2^{20}$
MD5	128	64	0.68	$2^{128}$	$2^{64}$
SHA-0	160	80	0.28	$2^{160}$	$2^{80}$
SHA-1	160	80	0.28	$2^{160}$	$2^{80}$
(*)SHA-224	224	64	( <sup>1</sup> )0.20	$2^{224}$	$2^{112}$
(*)SHA-256	256	64	( <sup>1</sup> )0.20	$2^{256}$	$2^{128}$
(*)SHA-384	384	80	( <sup>1</sup> )0.15	$2^{384}$	$2^{192}$
(*)SHA-512	512	80	( <sup>1</sup> )0.15	$2^{512}$	$2^{256}$
RIPEMD128	128	64	0.39	$2^{128}$	$2^{64}$
RIPEMD160	160	80	0.24	$2^{160}$	$2^{80}$

Cuadro N° 5: Principales funciones de hashing en uso en el ámbito mundial. Las fuerzas ideales corresponden a los valores conocidos antes de la publicación de los últimos ataques (1). (\*)Estos algoritmos emparentados se conocen genéricamente como SHA-2. (\*\*)Obsérvese que la fuerza contra las colisiones del MD4 estuvo muy reducida desde sus orígenes.

Hasta el momento de la publicación de los ataques (1)(2) existía el consenso en la comunidad criptográfica que el SHA-1 y RIPEMD160 eran prácticamente invulnerables, veremos en el punto siguiente hasta qué punto esta opinión se ha modificado.

## Ataques de colisión

En la reunión anual de criptología CRYPTO'04 surgieron a la luz una serie de vulnerabilidades de los hashings actuales empleando variantes computacionalmente eficientes del ataque de *colisiones diferenciales* (3) de Chabaud-Joux. Un equipo chino acaba de publicar un trabajo interesante (1): halló una variante computacional eficiente para generar colisiones reales (ver definición arriba, resistencia N° 3). Los algoritmos involucrados son el MD4, MD5, HAVAL128 y RIPEMD128. Además se han comprometido los algoritmos SHA-0 y RIPEMD160. De hecho se generan pares de mensajes  $x$  y  $x'$  con igual imagen  $z = h(x) = h(x')$ . Esto fue seguido por la publicación de un segundo trabajo aún más significativo por parte de un equipo australiano (2) quien halló por variantes del método Chabaud-Joux ataques de segunda preimagen contra la familia SHA-2.

En el primer trabajo (1), a pesar que aún no se conocen detalles acerca de cómo se seleccionan los mensajes  $x$  a colisionar, para el MD5, el método usa los IV originales y está basado en:

1. Uso de dos submensajes concatenados (M, N) de 512 bits cada uno (=16 DWords) *little-endian* seleccionados por un método no aclarado.

2. A partir de los submensajes M y N (organizados como un vector de 16 DWords) se obtienen los mensajes colisionantes M' y N' sumando "máscaras" que poseen DWords distintos de cero en las posiciones 4, 11 y 14:

$$\begin{aligned} M' &= M + (0, 0, 0, 0, 2^{31}, \dots, 2^{15}, \dots, 2^{31}, 0) \\ N' &= N + (0, 0, 0, 0, 2^{31}, \dots, -2^{15}, \dots, 2^{31}, 0) \end{aligned}$$

3. Resulta finalmente que

$$\text{MD5}(M, N) = \text{MD5}(M', N')$$

De hecho, los autores alegan generar M' y N' en el término de una hora en una IBM P690.

Los ataques contra HAVAL128, MD4, RIPEMD128 que han sido publicados, son prácticamente idénticos ya que todos emplean este método de "máscara" de DWords, sólo difieren en las potencias de dos empleadas en determinadas posiciones de esa máscara y las posiciones de los valores no nulos dentro de esas máscaras. En síntesis, se trata de una "familia" de ataques de *colisiones diferenciales* de Chabaud-Joux eficaces contra una "familia" de funciones.

Respecto al segundo trabajo podemos concluir que se trata de un ataque de *colisiones diferenciales* tipo Chabaud-Joux sobre la familia SHA-2, la que hasta ahora se consideró invulnerable y reemplazante potencial del SHA-1. El método está basado en la búsqueda de *patrones correctivos* sobre el registro del digesto para lograr dichas colisiones *round a round*. Previamente, otro equipo de investigadores como Gilbert y Hanshuh (6) había hallado colisiones por esta variante de *patrones correctivos* usando diferencias por función XOR, pero terminó concluyendo que la familia SHA-2 es inmune al ataque de colisiones porque su probabilidad de su ataque era inferior al límite dado por el *ataque de cumpleaños*. Sin embargo, en el presente trabajo se amplió el esquema previo usando diferencias aditivas y esto resultó ser fructífero porque permitió pasar del ataque original por colisiones a un ataque mucho más severo de *segundas preimágenes*, el que posee un límite sensiblemente superior (específicamente el cuadrado del anterior), razón por la cual los autores de este trabajo (2) extraen la conclusión que la familia SHA-2 dejó de ser inmune al ataque Chabaud-Joux, no sólo por sus propios métodos sino por lo hallado por Gilbert y Hanshuh (4) quienes usaran los límites teóricos de colisiones. A pesar que los autores (2) del trabajo señalan haber obtenido un ataque derivado de *segunda preimagen*, sólo se indican resultados de los ataques de colisiones y no se indica cómo pasar del mismo a las segundas preimágenes el que queda en términos potenciales hasta que se aporte la suficiente evidencia.

Sintetizamos en el Cuadro N° 6 los resultados de todos estos ataques, comparando la Fuerza Ideal de cada algoritmo contra la Fuerza Residual frente a este ataque, lo que ilustra el grado de debilitamiento obtenido contra colisiones

FUNCION	bits	Fuerza Ideal de Colisión	Fuerza Residual Post-Ataque de Colisión
MD4	128	$2^{20}$	1 (**)
MD5	128	$2^{64}$	$2^{32}$ (**)
SHA-0	160	$2^{80}$	$2^{40}$
SHA-1	160	$2^{80}$	$2^{39}$ (**)
<sup>(1)</sup> SHA-224	224	$2^{112}$	$2^{39}$ <sup>(a)</sup> y $2^9$ <sup>(b)</sup>
<sup>(1)</sup> SHA-256	256	$2^{128}$	$2^{39}$ <sup>(a)</sup> y $2^9$ <sup>(b)</sup>
<sup>(1)</sup> SHA-384	384	$2^{192}$	$2^{39}$ <sup>(a)</sup> y $2^9$ <sup>(b)</sup>
<sup>(1)</sup> SHA-512	512	$2^{256}$	$2^{39}$ <sup>(a)</sup> y $2^9$ <sup>(b)</sup>
RIPEMD128	128	$2^{64}$	$2^{32}$ (**)
RIPEMD160	160	$2^{80}$	$2^{40}$ (**)
HVAL128	128	$2^{64}$	$2^6$
HVAL160	160	$2^{80}$	$2^{32}$

Cuadro N° 6: Funciones de hashing atacadas ilustrando la reducción de la Fuerza Ideal frente a colisiones, es decir el grado de debilitamiento resultante en cada uno. <sup>(1)</sup> Estos algoritmos *estándar* FIPS 180-2 han sido informalmente agrupados como familia SHA-2. (\*)El MD4 se puede atacar ahora sin computadora, es decir en forma manual. (\*\*) Valores estimados. <sup>(a)</sup> Valores con estados iniciales por *round* desconocidos del registro de digesto <sup>(b)</sup> Valores con estados iniciales por *round* conocidos del registro de digesto.

## Conclusiones

Resulta innegable que los hallazgos representan una evidente señal de alerta acerca de las fallas de seguridad en las familias funcionales involucradas. La inspección del Cuadro N° 6 ilustra gráficamente la debilitación de esos algoritmos. Sin embargo, no queda claro cómo se eligen los mensajes de 1024 bits atacados y que grado de distancia (o contenido semántico) poseen esos mensajes con el espacio habitual de los mensajes en lenguajes corrientes, es decir si es o no posible generar pares de mensajes inteligibles que pudiesen colisionar. En general, lo que se ha atacado puede ser considerado como la resistencia a colisiones (ver Definición 3.) o a seudocolisiones (ver Definición 4.)

De todas formas, no se ha documentado fehacientemente la factibilidad de ataques de preimagen o ataques de segunda preimagen, es decir los dos casos más peligrosos para poder transformar estos ataques en fraudes concretos. Por otra parte, si fuese factible atacar preimágenes, los sistemas de seguridad como el control de passwords de usuarios en sistemas UNIX (que almacena los hashings de esas contraseñas) quedaría seriamente afectado por poder fraguar las contraseñas. Si fuese factible atacar segundas preimágenes, se comprometerían los controles elementales de integridad en el tráfico de datos y el uso de las firmas digitales con digesto, tan ampliamente usadas en todo el sistema financiero mundial. Además habría que replantear seriamente algunos protocolos PKI (public key infrastructure) (norma X.509 v3), por ejemplo el protocolo SSL v3 ampliamente usado para crear “túneles” virtuales en sesiones HTTPS entre servidores web y clientes. Para mencionar uno de esos aspectos, en la elaboración de los

“*Master-Secret*” que sirven de base a la seguridad en el intercambio de paquetes, intervienen tanto MD5 como SHA1.

En síntesis, alerta sí, pero alarma aún no. No hay motivos para descartar todo lo que está implementado con las funciones de hashing actuales (exceptuando al MD4 que debe ser definitivamente erradicado). Los propios autores del trabajo (2) concluyen que a pesar que los resultados comprometen severamente a la familia SHA-2, la más avanzada de la actual generación de hashings, no hay evidencia suficiente para concluir si los algoritmos SHA-2 son suficientemente seguros o no y proponen que se efectúen análisis más detallados sobre el *message scheduling*. Por otra parte pasar de la teoría a la práctica siempre implica un largo trecho, a pesar que el DES 56-bits está quebrado, por ejemplo, no podemos concretar ese quiebre en tiempos adecuados como para comprometer fraudulentamente transacciones que duren algunos minutos a lo sumo.. Eso sí, deberemos seguir atentos a lo que el futuro nos aporte sobre este tema y parece que habrá mucha actividad en la creación y ataques a los algoritmos de control de integridad.

## Bibliografía

1. WANG X., FENG D., LAI X., YU H., Collisions for hash functions MD4, MD5, HAVAL-128 and RIPEMD, (preprint - 17 Aug 2004)  
(disponible on-line en <http://eprint.iacr.org/2004/199.pdf>)
2. HAWKES P., PADDON M., ROSE G. G., On corrective patterns for the SHA-2 Family, (preprint – 23 Aug 2004)  
(disponible on-line en <http://eprint.iacr.org/2004/207.pdf> )
3. CHABAUD F., JOUX A., Differential collisions in SHA-0, Advances in Cryptology – CRYPTO’98, Lecture notes in computer science, 1462: 56-71, Springer Verlag (1998)
4. MENEZES A.J., VAN OORSCHOT P.C., VANSTONE S.A., Handbook of Applied Cryptography, CRC Press, (1997), ISBN 0-8493-8523-7  
(disponible on-line en <http://www.cacr.math.uwaterloo.ca/hac/>)
5. National Institute of Standards and Technology (NIST), Federal Information Processing Standards (FIPS), Publication 180-2, Secure Hash Standard (SHS), February (2004) – Reemplaza al SHS (SHA-1) FIPS 180-1 (1995)
6. GILBERT H., HANSCHUH H., Security analysis of SHA-256 and sisters, Selected Areas in Cryptography 2003 (SAC 2003), Ottawa, Canada (2003)