

# MD5 dan SHA-1 ( Kriptografi Dengan Fungsi Hash )

**Unggul Utan Sufandi**

unggul@mail.ut.ac.id

## **Lisensi Dokumen:**

Copyright © 2003-2007 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Hash adalah suatu teknik "klasik" dalam Ilmu Komputer yang banyak digunakan dalam praktek secara mendalam. Hash merupakan suatu metode yang secara langsung mengakses record-record dalam suatu tabel dengan melakukan transformasi aritmatik pada key yang menjadi alamat dalam tabel tersebut. Key merupakan suatu input dari pemakai di mana pada umumnya berupa nilai atau string karakter. Pelacakan dengan menggunakan Hash terdiri dari dua langkah utama, yaitu:

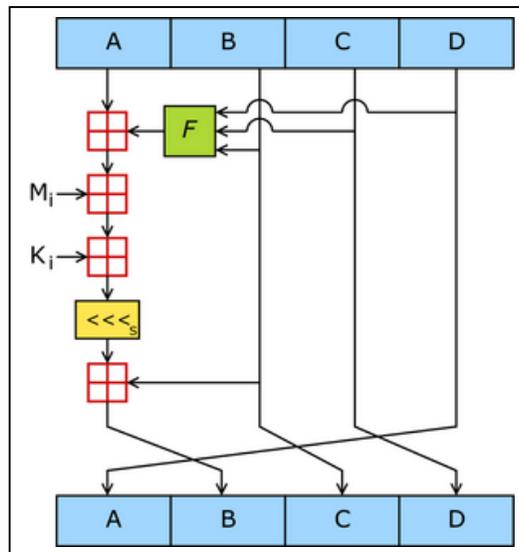
1. Menghitung Fungsi Hash. Fungsi Hash adalah suatu fungsi yang mengubah key menjadi alamat dalam tabel. Fungsi Hash memetakan sebuah key ke suatu alamat dalam tabel. Idealnya, key-key yang berbeda seharusnya dipetakan ke alamat-alamat yang berbeda juga. Pada kenyataannya, tidak ada fungsi Hash yang sempurna. Kemungkinan besar yang terjadi adalah dua atau lebih key yang berbeda dipetakan ke alamat yang sama dalam tabel. Peristiwa ini disebut dengan collision (tabrakan). Karena itulah diperlukan langkah berikutnya, yaitu collision resolution (pemecahan tabrakan).
2. Collision Resolution. Collision resolution merupakan proses untuk menangani kejadian dua atau lebih key di-hash ke alamat yang sama. Cara yang dilakukan jika terjadi collision adalah mencari lokasi yang kosong dalam tabel Hash secara terurut. Cara lainnya adalah dengan menggunakan fungsi Hash yang lain untuk mencari lokasi kosong tersebut.

## **MD5**

Fungsi hash yang paling banyak digunakan dalam keamanan jaringan komputer dan internet adalah MD5 yang dirancang oleh Ron Rivest yang juga merupakan salah satu pengembang algoritma RSA pada tahun 1991. MD5 merupakan kelanjutan dari MD4 yang dirancang dengan tujuan keamanan. Secara perhitungan matematis tidak dimungkinkan untuk mendapatkan dua pesan yang memiliki hash yang sama. Tidak ada serangan yang lebih efisien untuk membongkar/mengetahui hash suatu pesan selain brute-force.

## **CARA KERJA MD5**

MD5 mengolah blok 512 bit, dibagi kedalam 16 subblok berukuran 32 bit. Keluaran algoritma diset menjadi 4 blok yang masing-masing berukuran 32 bit yang setelah digabungkan akan membentuk nilai hash 128 bit.



Gambar 1. Algoritma MD5

Pesan diberi tambahan sedemikian sehingga panjang menjadi k-bit, dimana  $k = 512n - 64$  bit. n merupakan blok masukan. Tambahan ini diperlukan hingga pesan menjadi k bit. Kemudian 64 bit yang masing kosong, dibagian akhir, diisi panjang pesan. Inisiasi 4 variabel dengan panjang 32 bit yaitu a,b,c,d. Variabel a,b,c,d dikopikan ke variabel a,b,c,d yang kemudian diolah melalui 4 tahapan yang sangat serupa. Setiap tahapan menggunakan 16 kali operasi berbeda, menjalankan fungsi nonlinear pada tiga variabel a,b,c, atau d. Hasilnya ditambahkan ke variabel keempat, subblok pesan dan suatu konstanta. Kemudian dirotasi kekiri beberapa bit yang kemudian ditambahkan ke salah satu dari a,b,c, atau d. Kemudian nilai a,b,c, dan d menggantikan nilai a,b,c, dan d. Kemudian dikeluarkan output yang merupakan gabungan dari a,b,c, dan d. Fungsi kompresi yang digunakan oleh algoritma md5 adalah sebagai berikut :

$a \leftarrow b + ((a + g(b, c, d) + X[k] + T[i] \lll s)$ , dimana g adalah salah fungsi primitif F,G,H,I seperti dibawah ini :

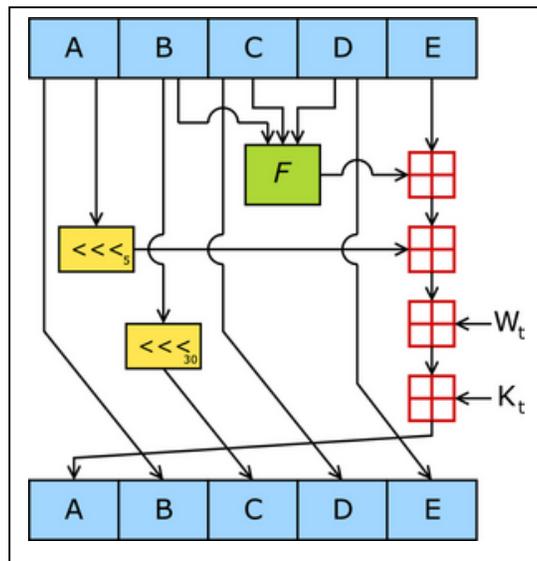
$$\begin{aligned}
 F(X, Y, Z) &= (X \wedge Y) \vee (\neg X \wedge Z) \\
 G(X, Y, Z) &= (X \wedge Z) \vee (Y \wedge \neg Z) \\
 H(X, Y, Z) &= X \oplus Y \oplus Z \\
 I(X, Y, Z) &= Y \oplus (X \vee \neg Z)
 \end{aligned}$$

dan operasi XOR, AND, OR, dan NOT adalah sebagai berikut :

$$\oplus, \wedge, \vee, \neg$$

## SHA

SHA dikembangkan oleh National Institute of Standards and Technology ( NIST ) dan National Security Agency ( NSA ) sebagai komponen Digital Signature Standart ( DSS ) . Standart hash adalah Secure Hash Standart ( SHS ) dengan SHA sebagai algoritma yang digunakan. SHS menetapkan SHA yang diperlukan untuk menjamin keamanan Digital Signature Algorithm ( DSA ).



Gambar 1. Algoritma SHA – 1

## SHA - 1

Sebuah versi revisi dari SHA sebagai FIPS 180-1 pada tahun 1995 dan secara umum dikenal sebagai SHA-1.

### CARA KERJA SHA - 1

Pesan diberi tambahan untuk membuat panjangnya menjadi kelipatan 512 bit (  $1 \times 512$  ). Jumlah bit asal adalah k bit. Tambahkan bit secukupnya sampai 64 bit kurangnya dari kelipatan 512 (  $512 - 64 = 448$  ), yang disebut juga kongruen dengan 448 ( mod 512 ). Kemudian tambahkan 64 bit yang menyatakan panjang pesan. Inisiasi 5 md variabel dengan panjang 32 bit yaitu a,b,c,d,e. Pesan dibagi menjadi blok-blok berukuran 512 bit dan setiap blok diolah. Kemudian keluaran setiap blok digabungkan dengan keluaran blok berikutnya, sehingga diperoleh output ( digest ). Fungsi kompresi yang digunakan oleh algoritma sha-1 adalah sebagai berikut :

$$A, b, c, d, e \leftarrow ( e + f(t, b, c, d) + s^5(a) + w_t + k_t ), a, s^{30}(b), c, d.$$

### PERBANDINGAN SHA-1 DAN MD5

Karena SHA-1 dan MD5 dikembangkan atau diturunkan dari MD4 maka keduanya mempunyai kemiripina satu sama lain, baik kekuatan dan karakteristiknya.

1. Keamanan terhadap serangan brute-force. Hal yang paling penting adalah bahwa SHA-1 menghasilkan diggest 32-bit lebih panjang dari MD5. Dengan brute-force maka SHA-1 lebih kuat dibanding MD5.
2. Keamanan terhadap kriptanalisis. Kelemahan MD5 ada pada design sehingga lebih mudah dilakukan kriptanalisis dibandingkan SHA-1
3. Kecepatan. Kedua algoritma bekerja pada modulo  $2^{32}$  sehingga keduanya bekerja baik pada arsitektur 32 bit. SHA-1 mempunyai langkah lebih banyak dibandingkan MD5 ( 80 dibanding MD5 64 ) dan harus memproses 160 bit buffer dibanding DM5 128 bit buffer, sehingga SHA-1 bekerja lebih lambat dibanding MD5 pada perangkat keras yang sama.
4. Simplicity. Kedua algoritma simple untuk dijelaskan dan mudah untuk diimplementasikan karena tidak membutuhkan program yang besar atau tabel substitusi yang besar pula.

5. Little-endian Versus Big-endian Arsitektur. Md5 menggunakan skema little-endian, sedangkan sha-1 menggunakan skema big-endian. Keduanya tidak memberikan keuntungan yang signifikan untuk sha-1 maupun md5.

## Referensi

<http://www.informit.com/guides/content.asp?g=java&seqNum=30&rl=1>  
<http://en.wikipedia.org/wiki/MD5>  
<http://en.wikipedia.org/wiki/SHA-1>  
[http://www.isi.edu/div7/publication\\_files/performance\\_analysis\\_md5.pdf](http://www.isi.edu/div7/publication_files/performance_analysis_md5.pdf)  
Pengantar Kriptografi. Sugi Guritman. Institut Pertanian Bogor. 2003

## Biografi Penulis



**Unggul Utan Sufandi.** Menyelesaikan S1 di Jurusan Sistem Informasi Universitas Pembangunan Nasional – Veteran, Pondok Labu, Jakarta, Pada tahun 2003 dan S2 Magister Ilmu Komputer, di Sekolah Pascasarjana Institut Pertanian Bogor, pada tahun 2007. Staff Teknis pada Pusat komputer Universitas Terbuka, Jakarta, sejak tahun 1999. Kompetensi inti pada bidang *Structured Analysis and Design, Object-oriented Analysis and Design, Object-oriented Programming.*