Perbandingan Fungsi Hash SHA-1 dengan MD5

Dini Lestari Tresnani - 13508096
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
if18096@students.if.itb.ac.id

***Abstract*—Dengan berkembangnya dunia teknologi di era globalisasi ini, keamanan data semakin dipertanyakan. Karena itu dibutuhkanlah sebuah metode untuk mengamankan data tersebut. Salah satu metodenya adalah SHA-1 dan MD5. Namun metode mana yang sebaiknya dipilih untuk mengamankan data dapat dilihat dengan membandingkan keduanya seperti yang bisa dilihat pada makalah ini. Perbandingan kedua metode didapatkan dari uji coba langsung dan analisis dari hasil uji coba tersebut.**

***Index Terms*—SHA-1, MD5, analisis, perbandingan, fungsi hash.**

I. PENDAHULUAN

Siapa yang tidak tahu bahwa dunia sekarang sedang dilanda oleh kemajuan teknologi yang sangat pesat. Terutama di era globalisasi ini teknologi yang berkembang dan masuk ke Indonesia lebih pesat lagi. Berbagai kemudahan telah dilahirkan dari perkembangan teknologi saat ini. Jika zaman dulu untuk membeli segala sesuatunya seseorang harus pergi langsung ke toko, saat ini ia hanya tinggal membuka situs yang menjual barang yang diinginkan lalu akan ada seseorang yang mengantarkan barang tersebut ke rumah. Tidak hanya saat seseorang ingin membeli barang, saat ini untuk mengirim uang kepada orang lain tidak perlu mendatangi orang tersebut secara langsung. Cukup mendatangi ATM terdekat dan memasukkan nomor rekening orang tersebut dan uang akan segera dikirim. Bahkan dengan kemajuan terkini, pengiriman uang bisa dilakukan tanpa harus bersusah payah datang ke mesin ATM. Pengiriman uang dapat dilakukan melalui telepon genggam atau komputer saja.

Namun berbagai kemudahan ini tentu ada bayarannya. Bayarannya adalah ada data pribadi kita yang diminta oleh aplikasi yang menyediakan kemudahan ini, yang akan fatal jadinya apabila data ini jatuh ke tangan orang yang salah.

Contohnya, untuk mengirimkan uang melalui mesin ATM, seseorang harus terlebih dahulu memiliki nomor PIN dari kartu ATM yang dimiliki. Bayangkan yang terjadi apabila nomor ini jatuh ke tangan orang yang salah. Tidak jadi soal apabila saldo dari rekening pada kartu ATM hanya sedikit. Bayangkan jika saldo yang dimiliki jumlahnya jutaan, puluhan juta, ratusan juta, atau bahkan milyaran.

PIN atau lebih umum dikenal dengan password, pada setiap aplikasi, umumnya disimpan pada sebuah database yang terhubung dengan user tertentu. Apabila sebuah password disimpan begitu saja pada database tersebut, tentunya bagi "tangan-tangan jahil" yang ingin membobol password seseorang cukup membobol database tersebut dan didapatkan password seluruh pengguna. Karena itulah timbul pemikiran untuk menyimpan password ato kode-kode rahasia ini ke dalam suatu bentuk lain yang rahasia juga sehingga apabila database berhasil dibobol, password atau kode rahasia tetap menjadi sebuah rahasia yang hanya pengguna dan Tuhan yang tahu.

Bentuk dari penyimpanan password atau kode rahasia lainnya pada database adalah dengan menyimpan fungsi hash dari password atau kode rahasia tersebut.

Suatu fungsi *hash* h memetakan bit-bit string dengan panjang sembarang ke sebuah string dengan panjang tertentu misal n. Dengan domain D dan range R maka: Proses *hashing* merupakan proses pemetaan suatu input string menjadi output disebut. Output dari fungsi *hash* disebut nilai *hash* atau hasil *hash*.[1]

II. DASAR TEORI

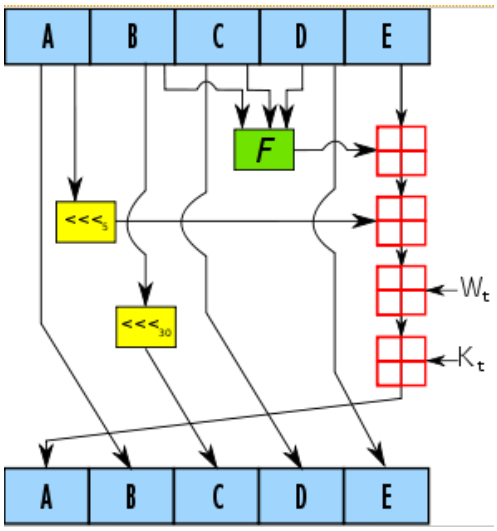
Fungsi hash memiliki berbagai macam variasi. Namun variasi dari fungsi hash yang akan dibahas pada makalah ini ada 2. Yaitu fungsi hash SHA-1 dan fungsi hash MD5.

A. SHA-1

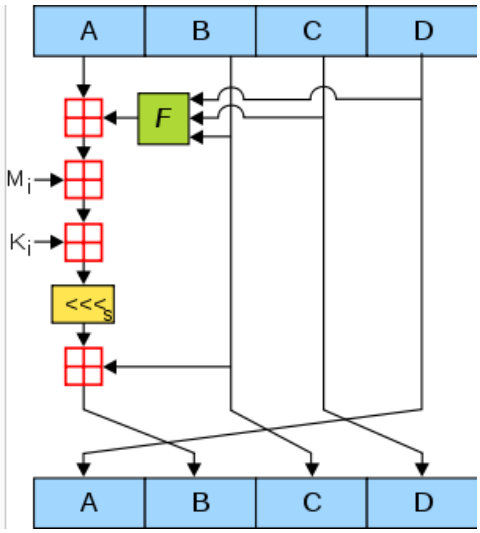
Dalam kriptografi, SHA-1 adalah fungsi hash kriptografi yang didesain oleh NSA (*National Security Agency*) dan diumumkan oleh NIST sebagai *U.S. Federal Information Processing Standard*. SHA artinya adalah *Secure Hash Algorithm*. SHA hingga saat ini ada 3 macam yaitu SHA-0, SHA-1, dan SHA-2. Struktur SHA-1 dan SHA-0 mirip satu sama lain. Hal ini dikarenakan SHA-1 merupakan perbaikan dari SHA-0.

Gambar di bawah ini adalah skema proses SHA-0 dan

SHA-1.



Gambar 1. Skema Proses SHA-0 dan SHA-1



Gambar 2. Skema Proses MD5

Cara kerja SHA-1 adalah dengan cara pesan diberi tambahan untuk membuat panjangnya menjadi kelipatan 512 bit (l x 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 (diggest).[2]

Untuk lebih lengkapnya mengenai cara kerja algoritma fungsi hash SHA-1 dapat dilihat pada artikel http://ilmukomputer.org/2007/03/27/md5-dan-sha-1-kriptografi-dengan-fungsi-hash/ atau mencari artikel lain pada situs pencarian yang ada.

B. MD5

MD5 merupakan bagian dari Message-Digest Algoritm yang merupakan bagian dari fungsi hash kriptografi yang menggunakan nilai hash 128-bit (16-byte). MD5 sudah sering diaplikasikan pada banyak aplikasi keamanan, selain itu juga sering digunakan untuk mengecek integritas dari sebuah file. Namun sayangnya, MD5 tidak dapat dipergunakan untuk aplikasi seperti SSL certificates ataupun digital signature. Sebuah fungsi hash MD5 biasanya diekspresikan sebagai 32-digit angka hexadecimal.

MD5 didesain oleh Ron Rivest pada tahun 1991 untuk menggantikan fungsi hash sebelumnya yaitu MD4. Pada tahun 1996, sebuah celah ditemukan pada desain MD5. Walaupun celah tersebut bukan merupakan sebuah kelemahan yang fatal, banyak yang mulai beralih menggunakan SHA-1. Pada tahun 2004, celah yang lebih serius telah ditemukan. Membuat MD5 dipertanyakan penggunaannya.

Cara kerja MD5 adalah dengan 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.

Pesan dimodifikasi sedemikian rupa sehingga panjang menjadi k-bit, dimana k = 512n-64 bit. [2]

Untuk lebih lengkapnya mengenai cara kerja algoritma fungsi hash MD5 dapat dilihat pada artikel http://ilmukomputer.org/2007/03/27/md5-dan-sha-1-kriptografi-dengan-fungsi-hash/ atau mencari pada situs pencarian.

III. UJI COBA

A. SHA-1 Tools

Aplikasi yang digunakan untuk mendapatkan nilai hash SHA-1 adalah aplikasi yang dibuat sendiri menggunakan kakas Microsoft Visual Studio 2010 dengan bahasa pemograman C# dengan menggunakan pseudocode berikut: (pseudocode didapatkan dari internet, namun aplikasi dibuat sendiri)

```
h0 = 0x67452301
h1 = 0xEFCDAB89
h2 = 0x98BADCFE
h3 = 0x10325476
h4 = 0xC3D2E1F0

append the bit '1' to the message
append 0 ≤ k < 512 bits '0', so that
the resulting message length (in bits)
   is congruent to 448 ≡ −64 (mod 512)
append length of message (before pre-
processing), in bits, as 64-bit big-
endian integer
```

```
break message into 512-bit chunks
for each chunk
    break chunk into sixteen 32-bit
big-endian words w[i], 0 ≤ i ≤ 15

    for i from 16 to 79
        w[i] = (w[i-3] xor w[i-8] xor
w[i-14] xor w[i-16]) leftrotate 1

    a = h0
    b = h1
    c = h2
    d = h3
    e = h4

    for i from 0 to 79
        if 0 ≤ i ≤ 19 then
            f = (b and c) or ((not b)
and d)
            k = 0x5A827999
        else if 20 ≤ i ≤ 39
            f = b xor c xor d
            k = 0x6ED9EBA1
        else if 40 ≤ i ≤ 59
            f = (b and c) or (b and d)
or (c and d)
            k = 0x8F1BBCDC
        else if 60 ≤ i ≤ 79
            f = b xor c xor d
            k = 0xCA62C1D6

        temp = (a leftrotate 5) + f +
e + k + w[i]
        e = d
        d = c
        c = b leftrotate 30
        b = a
        a = temp

    h0 = h0 + a
    h1 = h1 + b
    h2 = h2 + c
    h3 = h3 + d
    h4 = h4 + e

digest = hash = h0 append h1 append h2
append h3 append h4
```

Tampilan program untuk mendapatkan nilai hash SHA-1 adalah seperti pada gambar di bawah ini.

Gambar 3. Tampilan Menu Utama Program

Pada menu utama, dipilih SHA-1 dan tampilan berubah menjadi seperti di bawah ini.

Gambar 4. Tampilan Program Nilai Hash SHA-1

Masukan string yang digunakan untuk melakukan uji coba ada 3. String-string tersebut adalah:

String 1:

```
nama saya dini lestari tresnani dan
ini adalah makalah kripto saya.
```

String 2:

```
nama saya dini lestari tresnani dan
ini adalah makalah kripto saya
```

String 3: (string kosong)

```

```

Dari aplikasi tersebut didapatkan nilai hash SHA-1 dari string di atas seperti di bawah ini:

String 1:

```
5e333f84854b953c1ee5fd9e4552b676c9bab
116
```

String 2:

```
1fe2335e5ef0c5093c5d106d13ecc3289f6ea
753
```

String 3:

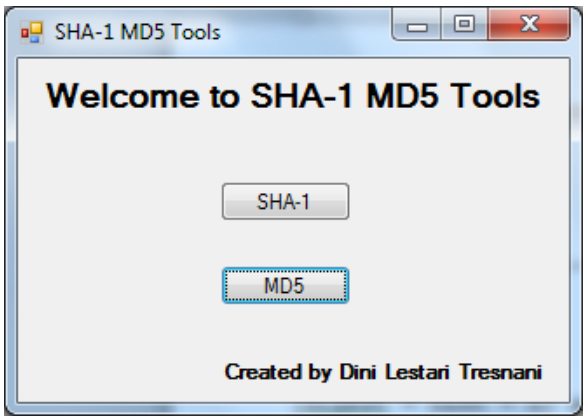
```
da39a3ee5e6b4b0d3255bfef95601890afd80
709
```

B. MD5 Tools

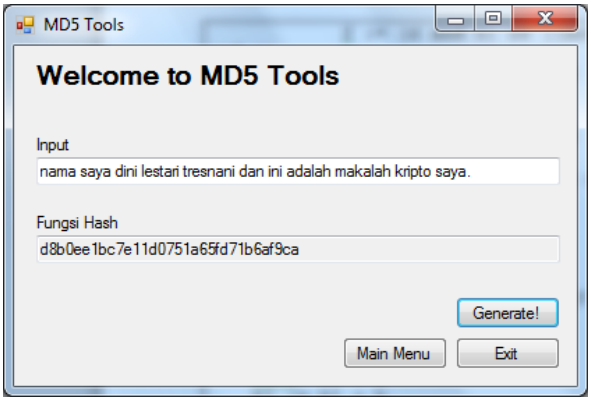
Aplikasi yang digunakan untuk mendapatkan nilai hash MD5 adalah aplikasi yang dibuat sendiri menggunakan kakas Microsoft Visual Studio 2010 dengan bahasa pemograman C# dengan menggunakan pseudocode berikut: (pseudocode didapatkan dari internet, namun aplikasi dibuat sendiri)

```
var int[64] r, k

r[ 0..15] := {7, 12, 17, 22,  7, 12,
17, 22,  7, 12, 17, 22,  7, 12, 17,
22}
r[16..31] := {5,  9, 14, 20,  5,  9,
14, 20,  5,  9, 14, 20,  5,  9, 14,
20}
r[32..47] := {4, 11, 16, 23,  4, 11,
16, 23,  4, 11, 16, 23,  4, 11, 16,
23}
r[48..63] := {6, 10, 15, 21,  6, 10,
15, 21,  6, 10, 15, 21,  6, 10, 15,
21}

for i from 0 to 63
    k[i] := floor(abs(sin(i + 1)) × (2
pow 32))

var int h0 := 0x67452301
var int h1 := 0xEFCDAB89
var int h2 := 0x98BADCFE
var int h3 := 0x10325476

append "1" bit to message
append "0" bits until message length
in bits ≡ 448 (mod 512)
append

for each 512-bit chunk of message
    break chunk into sixteen 32-bit
little-endian words w[j], 0 ≤ j ≤ 15

    var int a := h0
    var int b := h1
    var int c := h2
    var int d := h3

    for i from 0 to 63
        if 0 ≤ i ≤ 15 then
            f := (b and c) or ((not b)
and d)
            g := i
        else if 16 ≤ i ≤ 31
            f := (d and b) or ((not d)
and c)
            g := (5×i + 1) mod 16
        else if 32 ≤ i ≤ 47
            f := b xor c xor d
            g := (3×i + 5) mod 16
        else if 48 ≤ i ≤ 63
            f := c xor (b or (not d))
            g := (7×i) mod 16

        temp := d
        d := c
        c := b
        b := b + leftrotate((a + f +
k[i] + w[g]) , r[i])
        a := temp

    h0 := h0 + a
    h1 := h1 + b
    h2 := h2 + c
    h3 := h3 + d
var char digest[16] := h0 append h1
append h2 append

  leftrotate (x, c)
      return (x << c) or (x >> (32-
c));
```

Tampilan program untuk mendapatkan nilai hash MD5 adalah seperti pada gambar di bawah ini.



Gambar 5. Tampilan Menu Utama Program



Gambar 6. Tampilan Program Nilai Hash MD5

Masukan string yang digunakan untuk melakukan uji coba ada 3. String-string tersebut adalah:

String 1:

```
nama saya dini lestari tresnani dan ini adalah makalah kripto saya.
```

String 2:

```
nama saya dini lestari tresnani dan ini adalah makalah kripto saya
```

String 3: (string kosong)

```

```

Dari aplikasi tersebut didapatkan nilai hash MD5 dari string di atas seperti di bawah ini:

String 1:

```
d8b0ee1bc7e11d0751a65fd71b6af9ca
```

String 2:

```
4e05a7700381fa48483d20efd4d0e620
```

String 3:

```
d41d8cd98f00b204e9800998ecf8427e
```

IV. ANALISIS

Melihat dari hasil uji coba yang dilakukan dengan menggunakan 3 string contoh, maka di dapatkan tabel perbandingan seperti di bawah ini.

Tabel 1. Tabel Perbandingan Fungsi Hash

String	SHA-1	MD5
nama saya dini lestari tresnani dan ini adalah makalah kripto saya.	5e333f84854b953c1ee5fd9e4552b676c9bab116	d8b0ee1bc7e11d0751a65fd71b6af9ca
nama saya dini lestari tresnani dan ini adalah makalah kripto saya	1fe2335e5ef0c5093c5d106d13ecc3289f6ea753	4e05a7700381fa48483d20efd4d0e620
	da39a3ee5e6b4b0d3255bfef95601890afd80709	d41d8cd98f00b204e9800998ecf8427e

Pada tabel di atas dapat dilihat bahwa dalam masalah hasil yang berubah saat ada satu karakter saja dihilangkan, kedua fungsi hash sudah kuat. Karena saat string pertama dihilangkan karakter titiknya, nilai hash yang dihasilkan sangat jauh berbeda.

Kemudian dibandingkan hasil fungsi hash SHA-1 dengan hasil fungsi hash MD5.

Hasil fungsi hash SHA-1 adalah 160 bit. Sedangkan hasil fungsi hash MD5 adalah 128 bit. Maka fungsi SHA-1 32 bit lebih panjang dari fungsi hash MD5. Karena itu dapat diambil kesimpulan SHA-1 lebih sulit ditebak daripada MD5.

Namun, dikarenakan fungsi hash SHA-1 menghasilkan 160 bit sedangkan fungsi hash MD5 menghasilkan 128 bit, maka pada realitanya, fungsi hash SHA-1 membutuhkan waktu yang lebih lama untuk mengolah fungsinya dibandingkan fungsi hash MD5.

Kedua fungsi hash menggunakan modulo 2^{32} sehingga kedua fungsi pasti dapat berjalan dengan sangat baik pada komputer dengan arsitektur 32 bit.

Menurut beberapa sumber, itulah yang menyebabkan fungsi hash MD5 lebih rentan terhadap serangan brute-force dibandingkan fungsi hash SHA-1.

Pada penelitian berdasarkan literatur, diketahui bahwa baik SHA-1 maupun MD5 telah ditemukan kriptanalisisnya. Namun berdasarkan beberapa literature yang ditemukan fungsi hash MD5 dianggap lebih mudah dikriptanalisis dibandingkan fungsi hash SHA-1.

Pada arsitektur MD5, skema yang digunakan adalah skema little-endian. Sedangkan pada arsitektur SHA-1, skema yang digunakan adalah skema big-endian. Pemilihan skema ini tidak berdampak terlalu signifikan pada perbedaan keduanya.

Menurut beberapa pandangan, kedua algoritma ini merupakan algoritma yang simple dan mudah untuk diimplementasikan. Karena itu kedua algoritma ini terkenal dan sangat sering dipakai. Walaupun sekarang ini telah ditemukan algoritma SHA-2 yang dirasa lebih aman daripada kedua algoritma ini.

V. KESIMPULAN

Baik fungsi hash SHA-1 maupun fungsi hash MD5 memiliki keunggulan dan kelemahannya masing-masing dan dapat dikategorikan menjadi 5 kategori.

1. Serangan

   MD5 lebih rentan diserang terutama jika menggunakan algoritma Brute-Force karena jumlah bit yang hanya 128 bit. Sedangkan SHA-1 lebih sulit karena jumlah bit yang dihasilkan 160 bit.
2. Kriptanalisis

   Baik MD5 maupun SHA-1 telah ditemukan kriptanalisis yang tepat. Namun MD5 lebih mudah ditemukan daripada SHA-1.
3. Kecepatan

   Karena SHA-1 menghasilkan 160 bit sedangkan MD5 menghasilkan 128 bit, maka dalam menghasilkan fungsi hash MD5 kecepatan eksekusinya lebih cepat disbanding menghasilkan SHA-1.

4. Sederhana
   Kedua fungsi hash dianggap sama-sama sederhana dan sama-sama mudah diimplementasikan.
5. Arsitektur
   Baik yang menggunakan big-endian ataupun little-endian tidak memiliki perbedaan yang signifikan.

REFERENSI

[1] http://ilmu-kriptografi.blogspot.com/2009/05/fungsi-hash.html
[2] http://ilmukomputer.org/2007/03/27/md5-dan-sha-1-kriptografi-dengan-fungsi-hash/
[3] http://dark-holi.blogspot.com/2011/03/perbedaan-antara-md5-dan-sha.html
[4] http://rusda04.wordpress.com/2011/03/24/perbandingan-antara-md5-dan-sha/
[5] http://rizkiekasatria.wordpress.com/2011/03/24/md5-sha/
[6] Slide Kuliah IF3058 Kriptografi "Algoritma MD5"
[7] Slide Kuliah IF3058 Kriptografi "Secure Hash Algorithm (SHA)"

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 7 Mei 2011

Dini Lestari Tresnani - 13508096