Studi dan Analisis Collision pada Fungsi Hash

Tessa Ramsky - NIM : 13504124
Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10 Bandung
Email: if14124@students.if.itb.ac.id

Abstract – *Makalah ini membahas tentang studi analisis collision yang terdapat di dalam fungsi hash. Studi yang dilakukan mencangkup pemahaman mengenai fungsi hash, MD5 sebagai contoh dari fungsi hash, serta collision yang erat kaitannya dengan fungsi hash. Selain itu makalah ini juga membahas salah satu contoh serangan collision, yaitu birthday attack pada digital signature.*

Serangan terhadap fungsi hash ini juga perlu untuk dianalisis penyebab serta langkah-langkah yang perlu dilakukan untuk menghindarinya.

Kata Kunci*: collision, birhtday attack, hash, MD, Digital signature*

1. PENDAHULUAN

Kriptografi merupakan ilmu yang mempelajari bagaimana membuat suatu pesan yang dikirim pengirim dapat disampaikan kepada penerima dengan aman. Selain itu, pesan yang dikirim harus terjaga kerahasiaannya dari penyerang atau kriptanalis. Dalam kriptografi, terdapat beberapa permasalahan yang berkaitan dengan masalah keamanan. Salah satunya yaitu yang berkaitan dengan verifikasi data. Berbagai serangan dilakukan oleh para kriptanalis untuk memecahkan pesan tersebut.

1.1. Fungsi Hash

Fungsi hash adalah suatu fungsi dengan masukan dengan panjang yang bebas dan memetakannya sehingga menghasilkan output yang panjang nya tetap.

Dalam dunia kriptografi, fungsi hash bukanlah merupakan suatu barang yang baru. Fungsi hash memiliki daya tarik tersendiri dikarenakan cukup banyak aplikasi yang menggunakan fungsi ini dalam penerapannya. Fungsi hash digunakan sebagai autentikasi, integritas dan digital signature. Salah satu aplikasinya yaitu penggunaan password dalam aplikasi digital atau internet.

Ada banyak terdapat algoritma yang memanfaatkan fungsi hash. Sampai beberapa tahun ketika *brute-force* dan kriptanalis berkembang pesat, MD5 adalah algoritma fungsi hash yang paling banyak digunakan.

1.2.MD5

MD5 adalah fungsi hash satu-arah yang dibuat oleh Ronald Rivest pada tahun 1995 yang merupakan perbaikan dari MD4 setelah MD4 berhasil diserang oleh kriptanalis.

Secara umum algoritma MD5 membangkitkan pesan ringkas bekerja dengan cara :
1. Menambahkan bit-bit pengganjal;
2. Menambahkan nilai panjang pesan semula;
3. Inisialisasi penyangga (*buffer*);
4. Pengolahan pesan dalam blok berukuran 512 bit.

Input algoritma ini adalah sebuah berita dengan panjang yang bervariasi dan menghasilkan output sebuah 128-bit message digest.

2. *COLLISION*

Collision pada arti harafiahnya adalah tumbukan, atau tabrakan. Pada enkripsi MD5, *collision* ini berarti membuat dua file yang memiliki output yang sama setelah diolah dengan enkripsi MD5. Dengan kata lain jika dituliskan dalam persamaan :

$$MD5(A) = MD5(B)$$

Dimana A dan B adalah dua pesan yang berbeda.

Fungsi Hash akan bersifat satu-arah, yaitu jika diberikan sebuah Y hasil dari fungsi *h*, maka akan tidak mungkin (secara perhitungan) untuk menemukan sebuah berita X dimana $h(X) = Y$ (*preimage resistant*) dan diberikan X dan h(X) maka akan tidak mungkin (secara perhitungan) untuk menemukan sebuah berita $X' \neq X$ dimana $h(X') = h(X)$ (*second preimage resistant*).

Sebuah *Collision Resistance Hash Functions (CRHF)* adalah fungsi *h* yang memenuhi kondisi :
a. Sebuah X dapat mempunyai panjang yang bervariasi dan hasil dari *h*(x) hanya mempunyai sebuah panjang yang tetap yaitu *n* bit
b. Fungsi hash tersebut harus memenuhi *preimage resistance* dan *second preimage resistance.*
c. Fungsi hash tersebut harus bersifat *collision*

resistant, yaitu dimana tidak mungkin (secara perhitungan) untuk menemukan dua berita yang mempunyai nilai hash yang sama.

Pre-image Resistance merupakan suatu keadaan diamana suatu nilai pesan ringkas *h*, secara komputasi tidak boleh ditemukan cara untuk mendapatkan pesan *x* yang bersesuaian dengan pesan ringkasnya jika :

$$f(x) = h$$

Sedangkan pada *Second Pre-image Resistance,* tidak boleh ditemukan pesan *x'* sehingga

$$f(x) = f(x').$$

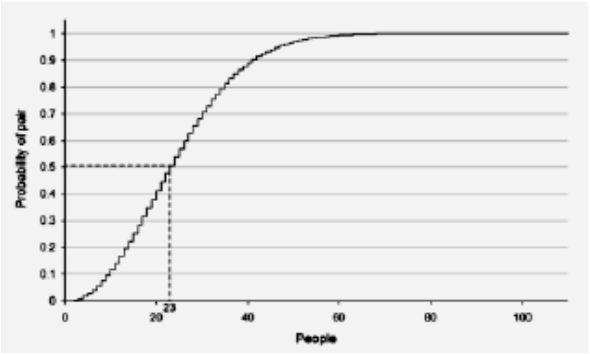
Salah satu serangan yang memanfaatkan sifat cillision daru fungsi hash ini adalah *birthday attack.*

3. *BIRTHDAY ATTACK*

2.1. *Birthday Paradox*

Collision yang terdapat di dalam teori probabilitas *birthday paradox* ini menyatakan bahwa disediakan sekelompok orang yang terdiri dari 23 atau lebih yang dipilih secara acak, kemungkinannya lebih dari 50% bahwa dua orang dari mereka akan memiliki hari yang tahun yang sama.

Grafik dibawah ini menunjukkan probabilitas sedikitnya dua orang memiliki ulang tahun yang sama diantara beberapa orang. Garis vertical menunjukkan probabilitas dan garis horizontal menunjukkan jumlah orang.



Kunci untuk memahami paradox ini ialah dengan berpikir bahwa tidak ada dua orang yang memiliki hari ulang tahun yang sama. Yang kemungkinan ada adalah kemungkinan orang pertama memiliki ulang tahun yang berbeda dengan orang kedua dan orang ketiga memiliki ulang tahun berbeda lagi dan orang keempat dan seterusnya. Jika terdapat sampel orang sebanyak n, orang pertama memiliki 365 kemungkinan ulang tahun yang dapat dipilih. Orang kedua hanya memiliki 364 kemungkinan, orang ketiga memiliki 363 kemungkinan, dan seterusnya.

$$\overline{p}(n) = 1\left(1-\frac{1}{365}\right)\left(1-\frac{2}{365}\right)\ldots\left(1-\frac{n-1}{365}\right)$$

$$p(n) = 1-\overline{p}(n)$$

Dimana n menunjukkan jumlah sample.

2.2.*Birthday Attack* pada *Digital Signature*

Digital signatures digunakan sebagai fungsi integritas suatu pesan yang dikirim yang digunakan untuk memverifikasi keaslian dari surat tersebut. Masukan untuk algoritma fungsi hash dapat berubah-ubah panjangnya (atau sangat panjang). Namun, keluarannya menghasilkan nilai hash yang panjang nya tetap (misalnya 128 bit, 256 bit). Apabila user A ingin mengirimkan sebuah pesan kepada user B, maka sebelumnya user A mencari *digital signature* pesan yang telah terenkripsi dengan menggunakan suatu algoritma fungsi hash. Dan mengenkripsi digital signature tersebut. Kemudian user B yang menerima pesan tersebut akan mendekripsi digital signature tersebut untuk mengetahui apakah pesan tersebut asli atau tidak.

Digital signature merupakan sarana yang paling rawan mendapat serangan *birthday attack*. Sebuah pesan m yang diberikan fungsi hash *h* akan memberikan nilai hash berupa *h*(m). Apabila pesan m tadi dimodifikasi sedemikian rupa menjadi pesan m' dan diberikan fungsi *h* yang sama, maka ada kemungkinan nilai hash dari *h*(m') = *h*(m). Hal ini tentunya membuat dokumen tersebut mudah mendapatkan serangan dari luar. Pengirim pesan dapat mengirimkan pesan yang palsu kepada penerima yang memiliki nilai hash yang sama, sehingga penerima tidak mengetahui bahwa pesan tersebut tidak asli.

Misalkan terdapat Aibon, Rika, dan Nono, dimana Aibon mengirimkan pesan kepada Rika, dan Nono bertindak sebagai penyerang.

Aibon menulis surat kepada Rika yang berisi:

```
Dear Rika,

I would like to recommend Yossi
for the job of regional manager
here at UFA Co. Ltd. Yossi is an
excellent employee...
```

Kemudian Aibon memberikan digital signature dengan fungsi hash pada surat tersebut, dan memberikannya pada Nono (sekretaris Aibon), untuk mengirimkan surat tersebut pada Rika.

Anggap Nono juga menginginkan posisi manager

sepreti yang ditulis pada surat tersebut, sehinggal ia tidak ingin Rika menerima surat tersebut. Maka ia dapat mengubah isi surat tersebut menjadi:

```
Dear Rika,

I think that Yossi is a poor
choice for regional manager at UFA
Co. Ltd. His work record is
terrible, he is frequently late...
```

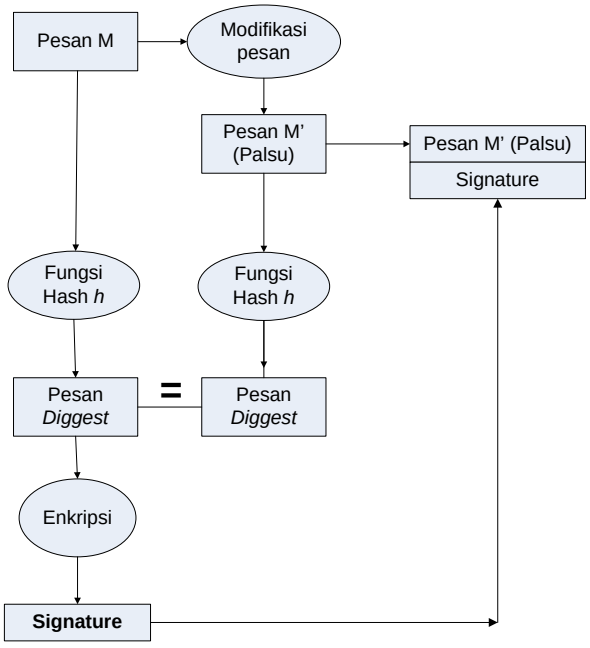
Permasalahannya adalah, pesan di atas tidak dapat diberikan *digital signature* oleh Nono karena ia tidak memiliki kunci *private* milik Aibon.

Maka, Nono menulis isi surat sebagai berikut:

```
Dear Rika,

I (feel|think) that (Yossi|Yossi
Smith) (is not|isn't) a (good|fair)
choice for the (position|job) of
regional manager. He is
(frequently|often) late...
```

Kata-kata yang ditandai dengan (a|b) merupakan kata-kata alternatif yang dapat digunakan. Baik kata a maupun b akan memberikan makna yang sama. Kemudian Nono membuat *Script* khusus untuk menghasilkan semua kemungkinan surat yang menggunakan kata-kata alternatif tersebut. Apabila terdapat n kata alternatif, maka *script* akan dijalankan sebanyak 2^n kali. Proses ini akan memakan waktu komputasi yang cukup lama. Jika ditemukan pesan dengan nilai hash yang sama, Nono dapat mensubtitusinya dengan pesan Aibon.



Yang patut menjadi perhatian utama di algoritma MD5 dalam masalah birthday attack adalah pesan dengan panjang sembarang akan selalu menghasilkan pesan ringkas dengan panjang 128, sehingga jumlah kemungkinan keluaran yang harus dihasilkan adalah 2^{128} kemungkinan pesan ringkas unik.

4. PEMBAHASAN *COLLISION*

4.1.*Collision* pada Dunia Nyata

Pesan-pesan yang diberikan fungsi hash dapat dianalogikan sebagai makhluk hidup, sebagai contoh manusia. Sebagai *digital signature* dari setiap pesan dapat dianalogikan sebagai Sidik Jari dan DNA,

a. Sidik Jari

Tidak ada satu individu manusia pun di dunia ini yang meiliki sidik jari dan DNA yang sama. Bahkan dua orang kembar pun, meskipun terlihat mirip, namun memilikik sidik jari yang berbeda. Sehingga sidik jari ini dapat digunakan sebagai salah satu cara untuk mengidentifikasi manusia. Prinsisp ini tentunya sangat cocok untuk diterapkan pada fungsi hash, dimana tidak terjadi *collision* dimana suatu sidik jari dimiliki oleh dua orang.

Gambar. Sidik jari yang indentik

b. DNA

Sama seperti sidik jari, bahwa tidak ada individu di dunia yang memiliki DNA yang sama. Dengan DNA dapat mengidentifikasi pemilik dari DNA tersebut yang sifatnya unik. Akan tetapi, DNA memiliki pola tertentu yang dapat menghubungkan antara satu individu dengan individu lainnya.

Penerapan algoritma fungsi hash yang mencontoh dua analogi di atas tentunya dapat menghindari berbagai collision terutama seperti yang terjadi pada *birthday attack*.

4.2.Hal-hal yang dapat dilakukan untuk menghindari terjadinya *Collision*.

Setelah mengetahui ide dasar dari fungsi hash dan serangan terhadapnya, maka dapat dikatakan bahwa fungsi ini masih sangat rentan untuk mendapatkan serangan. Oleh karena itu, diperlukan berbagai masukan untuk pengembangan fungsi ini lebih lanjut.

a. Untuk memastikan terhindar dari *birthday attack,* maka cara yang paling dinilai ampuh adalah dengan menjamin keamanan media transmisi pesan, yaitu dengan menggunakan SSH, serta tanpa perantara pengiriman (orang ketiga).
b. Menambah panjang output dari fungsi hash. Semakin panjang ouputnya, maka semakin susah pesan tersebut dicari nilai *collision*-nya. Namun, panjangnya nilai hash tersebut juga mempengaruhi lamanya waktu komputasi.
c. Memperpanjang pesan awal, sehingga sulit untuk dilakukan modifikasi karena banyaknya kata-kata yang harus dimodifikasi. Semakin banyak kata-kata yang dimodifikasi, maka semakin lama pula waktu komputasi yang dibutuhkan. Akan tetapi, seiring dengan perkembangan teknologi saat ini, dengan dukngan hardware dan software yang canggih, permasalahan komputasi ini dapat diatasi.
d. Menambah variasi karakter pada hasil dari fungsi hash, sehingga kemungkinan nilai hash yang dihasilkan pun menjadi semakin banyak. Hal ini dapat menghindari terjadinya collision.
   Misalnya dengan menggunakan keseluruhan karakter ASCII untuk setiap digit pada hasil hash. Jadi, seandainya hasil hash memiliki panjang m karakter, maka variasi fungsi hash yang dapat dibentuk adalah:

   255^m

   Semakin besar nilai m, maka semakin banyak pula variatif dari fungsi hash tersebut.

4.3.Menemukan Algoritma Hash yang Baru

Hingga saat ini, masih belum ada algoritma hash yang dijadikan standard, padahal aplikasi yang menggunakan fungsi hash seperti pada digital signature dan autentikasi password banyak dipakai di perusahaan-perusahaan, perbankan, dan aplikasi online lainnya. Hal ini akan menjadi masalah tersendiri bagi keamanan informasi.

Ditambah lagi dengan telah ditemukannya full-collision pada beberapa algoritma fungsi hash termasuk MD5, maka perlunya diciptakan algoritma hash yang baru yang lebih rentan terhadap kriptanalis terutama dalam hal menghindari terjadinya collision. Adanya collision ini membuat MD5 tidak lagi baik digunakan sebagai digital signature pada dokumen digital.

5. KESIMPULAN

Kesimpulan yang dapat diambil dari studi analisis collision pada fungsi hash dan birthday attack ini adalah:

a. *Birthday attack* adalah sebuah proses yang menggunakan prinsip *birthday paradox* untuk memilih masukan terhadap fungsi *hash f* yang akan diserang sampai menemukan sepasang masukan tersebut yang dapat menghasilkan sebuah kolisi
b. Collision padafungsi hash dapat digunakan untuk mencari masukan hash yang memiliki nilai hash yang sama
c. Adanya collision ini membuat MD5 tidak lagi baik digunakan sebagai Digital Signature pada dokumen digital
d. Penerapan fungsi hash sebaiknya mengikuti contoh nyata yaitu sidik jari pada manusia.

DAFTAR REFERENSI

[1] Munir, Rinaldi. (2006). Diktat Kuliah IF5054 Kriptografi. Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika , Institut Teknologi Bandung.

[2] http://en.wikipedia.org/wiki/Birthday_attack

[3] http://everything2.com/index.pl?node_id=521205

[4] http://www.win.tue.nl/hashclash/Nostradamus/