STUDI *BIRTHDAY ATTACK*

Magdalena Marlin Amanda – NIM: 13506042

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if16042@students.if.itb.ac.id

Abstrak

Makalah ini membahas tentang salah satu variasi serangan yang dilakukan terhadap algoritma kriptografi yang didasarkan pada suatu paradoks yang ditemukan dalam teori peluang di bidang matematika, *birthday paradox*. Selain membahas mengenai *birthday paradox* dan penggunaannya dalam salah satu metode serangan, makalah ini juga akan membahas kelemahan serta kelebihan dari jenis serangan tersebut apabila diterapkan pada berbagai algoritma kriptografi yang berbeda, mulai dari kriptografi paling sederhana hingga kriptografi tingkat lanjut yang cukup rumit.

***Kata kunci**:* brute force*, teori peluang,* birthday paradox*,* birthday attack, *Caesar* cipher, *Vigenere* cipher, cipher *blok*, digital signature, *fungsi* hash.

1. Pendahuluan

Seperti yang diketahui oleh banyak orang, metode paling sederhana untuk memecahkan suatu teks yang telah terenkripsi apabila algoritma enkripsi diketahui adalah dengan cara *brute force*. Meskipun cara ini terbukti dapat membongkar setiap jenis algoritma kriptografi yang ada, tapi waktu yang dibutuhkan hingga akhirnya didapat pemecahan sangat lama dan usaha yang diperlukan untuk pemrosesannya sangat besar, terutama jika kunci yang digunakan adalah sebuah kunci yang cukup panjang.

Dalam ilmu matematika, dalam teori peluang, terdapat sebuah paradoks yang diberi nama "*birthday paradox"* atau "*birthday problem*". Dengan suatu perhitungan berdasarkan teori peluang, seorang kriptanalis dapat mendapatkan suatu angka yang menunjukkan jumlah percobaan yang harus ia lakukan untuk mendapatkan kunci. Berdasarkan *birthday paradox*, percobaan yang harus dilakukan untuk mendapatkan kunci ternyata jauh lebih sedikit daripada yang selama ini dipercaya. Hal ini membuktikan bahwa serangan *brute force* hingga taraf tertentu tidak menghabiskan banyak waktu seperti yang selama ini dipercaya.

2. *Birthday Problem*

Secara logika, peluang terbesar mendapatkan dua orang dengan tanggal lahir sama adalah dengan mengambil 366 orang secara acak. Dengan asumsi tahun yang digunakan bukan tahun kabisat dan sampel yang diambil tidak ada satu pun yang kembar, sudah sewajarnya orang ke-366 memiliki tanggal lahir yang sama dengan minimal salah satu dari 365 orang lainnya.

Dengan pemikiran demikian, kebanyakan orang akan berpikir bahwa peluang sebesar lima puluh persen akan dicapai saat sampel yang diambil berjumlah setengah dari 365.

Berdasarkan teori peluang, kejadian diambilnya orang pertama, kedua, dan seterusnya merupakan kejadian "saling lepas" atau "saling bebas" yang berarti pengambilan sampel tidak akan mempengaruhi peluang pengambilan sampel selanjutnya.

Jika diilustrasikan dengan sebuah tabel, maka sampel sebanyak n orang dengan $n \leq 365$ akan menjadi:

	Tgl 1	Tgl 2	...	Tgl n
Orang	1	1	...	1

Rumus peluang berkondisi:

$$\boxed{P(B \mid A) = \frac{P(A \cap B)}{P(A)}}$$

Dua kejadian akan dianggap saling bebas apabila memenuhi syarat:

$$\boxed{P(B \mid A) = P(B)}$$

dan

$$\boxed{P(A \mid B) = P(A)}$$

Jika kejadian A adalah terpilihnya satu orang dan kejadian B adalah orang yang telah terpilih memiliki tanggal lahir x, maka didapat:

$$\boxed{P(A \mid B) = \frac{1}{1} = P(A)}$$

Terbukti bahwa kejadian pengambilan sampel berdasarkan tanggal lahir adalah kejadian saling bebas.

Untuk dua kejadian saling bebas, berlaku rumus:

$$\boxed{P(A \cap B) = P(A)P(B)}$$

Dengan demikian, dapat dihitung peluang minimal dua orang bertanggal lahir sama dengan jumlah sampel sebanyak n ($n \leq 365$). Untuk mempermudah perhitungan, akan digunakan prinsip komplemen:

$$\boxed{p(n) = 1 - \neg p(n)}$$

Menghitung peluang terjadinya tidak ada dua orang yang bertanggal lahir sama:

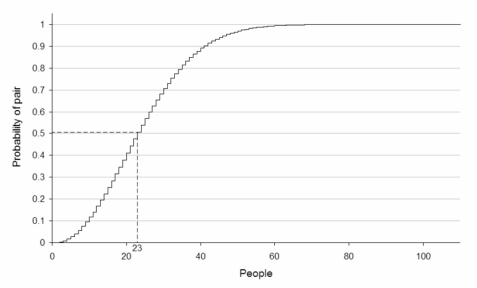
$$\neg p(n) = \frac{365}{365} \times \frac{364}{365} \cdots \frac{(365-(n-1))}{365} = \frac{365!}{365^n (365-n)!}$$

Keunikan dalam birthday problem terlihat saat dilakukan perhitungan untuk mencari tahu berapa jumlah sampel minimal yang harus diambil agar peluang minimal terdapat dua orang bertanggal lahir sama adalah lima puluh persen atau lebih.

$$\frac{1}{2} = 1 - \frac{365!}{365^n (365-n)!}$$

Didapatkan nilai n adalah 23, sangat jauh dari nilai 366.

Jika dijabarkan lebih lanjut *birthday paradox* akan menghasilkan sebuah grafik seperti di bawah ini:



Gambar 1. Grafik *Birthday Paradox*

3. *Birthday Attack*

Birthday attack adalah salah satu jenis serangan kriptografi yang diberi nama berdasarkan teori yang melatarbelakangi serangan ini, yaitu *birthday paradox*.

Biasanya serangan ini digunakan kriptanalis untuk menyerang fungsi *hash*. Ide serangan adalah mencari nilai x_1 dan x_2 jika diberikan suatu fungsi f.

$$f(x_1) = f(x_2)$$

Dalam fungsi hash, apabila ditemukan kolisi atau beberapa pesan berbeda yang memiliki pesan ringkas serupa, algoritma hash tersebut dianggap tidak aman.

Dari sebuah himpunan H, sejumlah n nilai secara acak sehingga dimungkinkan adanya pengulangan. P(n;H) adalah peluang dalam percobaan ini suatu nilai terpilih lebih dari satu kali.

$$P(n;H) \approx 1 - e^{-n(n-1)/2H} \approx 1 - e^{n^2/2H}$$

Dengan menginversi persamaan tersebut, didapatkan persamaan berikut:

$$n(p;H) \approx \sqrt{2H \ln \frac{1}{1-p}}$$

Dengan memasukkan nilai 0,5 ke dalam persamaan, didapatkan persamaan lain:

$$n(0{,}5;H) \approx 1{,}1774\sqrt{H}$$

Untuk Q(H) adalah jumlah nilai yang harus dipilih sebelum mendapatkan kolisi pertama, dapat dilakukan pembulatan menjadi:

$$Q(H) \approx \sqrt{\frac{\pi}{2}H}$$

Dengan demikian, dapat diperkirakan berapa kali percobaan yang harus dilakukan untuk mencapai nilai peluang tertentu.

4. Analisis

Pada bagian ini akan dibahas mengenai kemungkinan dipergunakannya *birthday attack* pada berbagai algoritma kriptografi, baik yang klasik maupun modern.

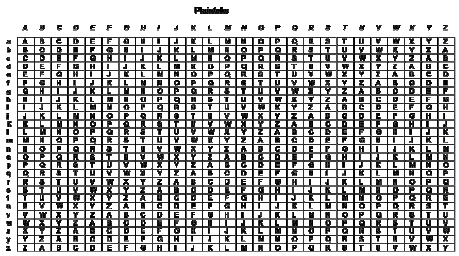
4.1 *Birthday Attack* untuk Caesar *Cipher*

Caesar *cipher* menggunakan algoritma kriptografi substitusi monoalfabetik, di mana satu abjad digantikan oleh tepat satu abjad lain. Pada kriptografi klasik, pertukaran abjad yang dilakukan hanya merupakan pergeseran abjad yang terurut, tapi modifikasi yang dilakukan selanjutnya menggunakan substitusi abjad yang sifatnya lebih acak sehingga lebih sukar dianalisis.

Caesar *cipher* tidak memiliki suatu fungsi spesifik yang menerima masukan suatu kunci untuk mengenkripsi, tapi hanya memetakan isi dari teks asli menjadi suatu cipherteks. Karena prinsip dasar dari *birthday attack* adalah mencari kecocokan antara hasil percobaan, serangan tersebut tidak bisa diterapkan pada Caesar *cipher* karena tidak adanya pembanding berupa teks asli.

4.2 *Birthday Attack* untuk Vigenere *Cipher*

Vigenere *cipher* adalah algoritma kriptografi substitusi monoalfabetik yang menggunakan kunci dan tabel enkripsi.

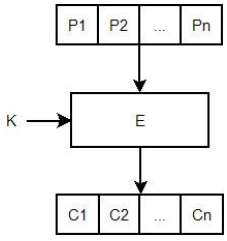


Gambar 2. Bujursangkar Vigenere

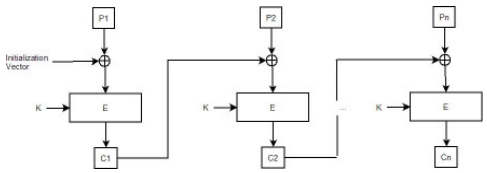
Sama halnya seperti pada Caesar *cipher*, serangan *birthday attack* tidak dapat dilakukan karena dibutuhkan suatu pembanding berupa teks asli.

4.3 *Birthday Attack* untuk *Cipher* Blok

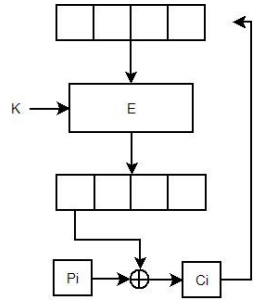
Cipher blok adalah salah satu algoritma kriptografi modern di mana berkas yang akan dienkripsi dibagi menjadi beberapa blok sama panjang yang akan diproses menjadi sebuah berkas terenkripsi. Mode operasi yang banyak digunakan adalah *Electronic Code Book*, *Cipher Block Chaining*, *Cipher Feedback*, dan *Output Feedback*.



Gambar 3. Skema *Electronic Book Cipher*



Gambar 4. Skema *Cipher Block Chaining*



Gambar 5. Skema *Cipher Feedback*

Sama seperti dua algoritma kriptografi sebelumnya, metode birthday attack tidak dapat diterapkan karena tidak adanya pembanding yang dapat digunakan.

4.4 *Birthday Attack* untuk Fungsi *Hash*

Fungsi *hash* adalah aplikasi keamanan yang menerima masukan sebuah *string* dengan panjang sembarang dan mengkonversi *string* tersebut menjadi sebuah *string* keluaran yang panjangnya akan selalu tetap.

Fungsi hash yang banyak digunakan adalah fungsi hash satu arah di mana pesan yang sudah diubah tidak dapat dikembalikan lagi menjadi pesan semula. Sifat-sifat fungsi hash satu arah adalah:

a. Fungsi H dapat diterapkan pada blok data berukuran berapa saja.
b. H menghasilkan nilai h dengan panjang tetap.
c. H(x) mudah dihitung untuk setiap nilai x yang diberikan.
d. Untuk setiap h yang diberikan, tidak mungkin menemukan x sedemikian sehingga H(x) = h.
e. Untuk setiap x yang diberikan, tidak mungkin mencari y ≠ x sehingga didapat H(y) = H(x).
f. Tidak mungkin mencari pasangan x dan y sehingga didapat H(x) = H(y).

Fungsi hash dianggap tidak aman apabila:

a. Dapat ditemukan suatu pesan yang bersesuaian dengan hasil fungsi hash-nya secara komputasi.
b. Terjadi kondisi kolisi di mana beberapa pesan yang berbeda mempunyai pesan ringkas yang serupa.

Metode *birthday attack* dapat diterapkan untuk mengetahui terjadi tidaknya kolisi dalam fungsi hash. Hingga sekarang, diketahui bahwa algoritma MD5 dianggap tidak aman lagi karena adanya kolisi yang dapat terjadi dengan serangan *birthday attack*. Cara lain untuk memperkecil kemungkinan ditemukannya kolisi adalah dengan memperbesar ukuran pesan ringkas yang merupakan hasil dari fungsi hash. Semakin besar ukuran pesan ringkas, semakin banyak variasi pesan ringkas yang dapat dibuat dan semakin banyak percobaan yang harus dilakukan untuk mencapai nilai peluang sebesar 0,5.

4.5 *Birthday Attack* untuk *Digital Signature*

Digital signature atau tanda tangan digital adalah salah satu cara untuk mencegah adanya penyangkalan atas pesan yang dikirim atau diterima. Fungsi tanda tangan digital sangat mirip dengan fungsi tanda tangan tradisional.

Birthday attack dapat digunakan untuk menipu orang agar menandatangani suatu "kontrak palsu".

Mula-mula, si pemalsu kontrak membuat dua versi kontrak, versi asli *m* dan versi palsu *m'*. Dengan memodifikasi versi asli dan versi palsu menjadi beberapa variasi, ia kemudian memasukkan variasi-variasi dari kedua versi tersebut ke dalam fungsi hash sehingga akhirnya mendapatkan hasil yang sama.

Birthday attack digunakan pada proses pencarian hasil fungsi *hash* yang sama tersebut.

Setelah didapatkan hasil fungsi hash yang sama, si pemalsu kontrak mengirimkan versi asli kepada penandatangan kontrak yang membaca dan karena menganggap kontrak tersebut sudah sesuai dengan kesepakatan, menandatanganinya kemudian mengirimkan kembali kepada si pemalsu kontrak.

Setelah kontrak *m* kembali berada di tangan si pemalsu kontrak, tanda tangan digital milik penandatangan diambil untuk kemudian dipasangkan pada kontrak palsu *m'*.

Pada kasus ini, *birthday attack* tidak digunakan untuk menyerang secara langsung tapi lebih dipergunakan untuk memproses kontrak asli dan palsu agar hasil *hash* keduanya tetap sama.

Serangan ini dapat diatasi dengan memperpanjang keluaran dari fungsi *hash* yang dipergunakan sehingga perhitungan yang dilakukan menjadi lebih banyak dan memakan waktu yang sangat lama.

5. Kesimpulan

Birthday attack sebenarnya merupakan suatu bentuk serangan *brute force*, tapi karena berdasarkan pada *birthday paradox*, komputasi yang dilakukan menjadi tidak sebanyak yang selama ini diketahui orang.

Beberapa poin yang didapat dari analisis di atas adalah:

- *Birthday attack* tidak dapat diterapkan pada algoritma kriptografi yang membutuhkan kunci. Karena tujuan utama seorang kriptanalis dalam menganalisa algoritma kriptografi dengan kunci adalah mencari kunci untuk kemudian dipergunakan mendekripsi cipherteks yang dimiliki. *Birthday attack* membutuhkan pembanding (berkas asli) yang pada kebanyakan kasus tidak dimiliki oleh kriptanalis. Selain itu, masih ada metode lain yang lebih mangkus dan sangkil untuk menyerang algoritma kriptografi yang menggunakan kunci.
- *Birthday attack* merupakan suatu serangan *brute force*, hanya saja jumlah percobaan yang dilakukan lebih dibatasi berdasarkan adanya *birthday paradox.*
- *Birthday attack* dapat digunakan untuk serangan terhadap tanda tangan digital, tapi lebih sebagai proses perantara serangan.

Berdasarkan poin-poin di atas, dapat disimpulkan bahwa *birthday attack* yang tidak berbeda jauh dengan serangan secara *brute force* tidak dapat diterapkan untuk semua jenis algoritma kriptografi dan dapat dengan mudah dicegah.

Sebagai proses antara dalam serangan terhadap tanda tangan digital, dengan asumsi fungsi *hash* tidak menghasilkan keluaran berukuran sangat besar, *birthday attack* juga cukup sukar untuk diterapkan karena kemungkinan variasi kontrak asli dan kontrak palsu sangat banyak.

Daftar Pustaka

Munir, Rinaldi. 2006. *Diktat Kuliah IF5054 Kriptografi*. Program Studi Teknik Informatika. Institut Teknologi Bandung.

Walpole, Ronald.1985. *Probability and Statistics for Engineers and Scientists*. Macmillan Publishing Company. New York.

http://mathworld.wolfram.com/BirthdayAttack.html Waktu akses: 29 Maret 2009, 10.30 WIB

http://www.chiark.greenend.org.uk/pipermail/ukcrypto/2002-October/059964.html Waktu akses: 29 Maret 2009, 10.30 WIB

http://en.wikipedia.org/wiki/Birthday_paradox Waktu akses: 29 Maret 2009, 10.30 WIB

http://en.wikipedia.org/wiki/Birthday_attack Waktu akses: 29 Maret 2009, 10.30 WIB

http://webspace.ship.edu/deensley/mathdl/stats/Birthday.html Waktu akses: 2 April 2009, 08.09 WIB

http://everything2.com/title/Birthday%2520attack Waktu akses: 2 April 2009, 09.34 WIB

http://www.ciphersbyritter.com/NEWS4/BIRTHDAY.HTM Waktu akses: 2 April 2009, 09.46 WIB

http://en.wikipedia.org/wiki/Digital_signature Waktu akses: 2 April 2009, 11.28 WIB