Studi dan Perbandingan Serangan Kriptografi *Birthday Attack* dan *Meet-in-the-middle Attack*

Muhamad Reza Firdaus Zen – NIM : 13504048
Sekolah Teknik Elektro dan Informatika ITB, Bandung , email: if14048@students.if.itb.ac.id

Abstrak – Serangan kriptografi *Birthday Attack* dan *Meet-in-the-middle Attack* merupakan serangan serangan yang memanfaatkan *space-time tradeoff* (situasi dimana penggunaan memori dapat direduksi dengan syarat eksekusi program yang lebih lambat, atau sebaliknya). *Birthday Attack* adalah tipe dari serangan kriptografi yang mengeksploitasi matematika dibalik paradoks ulang-tahun yang menyatakan dalam grup 23 orang yang dipilih secara acak, terdapat 50% lebih probabilitas minimal dua orang mempunyai ulang tahun yang sama. Secara spesifik, jika fungsi f(x) menghasilkan satu dari H output yang berbeda dengan probabilitas sama dan H cukup besar, maka setelah evaluasi fungsi dengan input berbeda sebanyak $1.2H^{1/2}$ diharapkan dapat output yang sama. Berbeda dengan *Birthday Attack*, *Meet-in-the-middle Attack* mencoba menemukan nilai dalam *range* dan *domain* masing masing dari komposisi dua fungsi dimana pemetaan lewat fungsi pertama adalah sama dengan invers lewat fungsi kedua. Karya tulis yang akan dibuat akan menjelaskan dan membandingkan kedua serangan tersebut dimana keduanya sama sama memanfaatkan *space-time tradeoff*.

***Kata Kunci**: space-time tradeoff, Birthday Attack, Meet-in-the-middle Attack.*

1. Pendahuluan

Kriptografi adalah ilmu yang mempelajari bagaimana membuat suatu pesan yang dikirim pengirim dapat disampaikan kepada penerima dengan aman. Selain itu, pesan yang dikirim harus terjaga kerahasiaannya (baik plainteks maupun kuncinya) dari penyadap (*eavesdropper*) atau kriptanalis. Kriptografi juga diharapkan dapat memenuhi kebutuhan bagi datadata seperti :

1. Kerahasiaan (*confidentiality*) dari pihak pihak yang tidak berwenang.
2. Keutuhan (*integrity*) atas data-data agar sampai secara utuh .
3. Jaminan atas identitas dan keabsahan (*authenticity*) dari data yang dikirim dilakukan dengan menggunakan tanda tangan atau sertifikat digital.
4. Dapat dijadikan barang bukti yang tidak bisa disangkal (*non-repudiation*) dengan memanfaatkan tanda tangan digital dan sertifikat digital.

Selain ada pihak yang ingin menjaga agar pesan tetap aman, ada juga ternyata pihak-pihak yang ingin mengetahui pesan rahasia tersebut secara tidak sah. Bahkan ada pihak-pihak yang ingin agar dapat mengubah isi pesan tersebut. Caranya yaitu dengan serangan kriptografi. Yang dimaksud dengan serangan adalah setiap usaha atau percobaan yang dilakukan oleh kriptanalis untuk menemukan kunci atau menemukan plainteks dari cipherteksnya. Berdasarkan ketersediaan data yang ada, serangan terhadap kriptografi diklasifikasi menjadi :

1. *Ciphertext only attack*, penyerang hanya mendapatkan pesan yang sudah tersandikan saja.
2. *Known plaintext attack*, dimana penyerang selain mendapatkan cipheteks, juga mendapatkan potongan pesan asli.
3. *Chosen plaintext attack*, sama dengan *known plaintext attack*, namun penyerang dapat memilih penggalan mana dari pesan asli yang akan disandikan.

Kedua serangan yang akan dibahas termasuk dalam kategori *known plaintext attack* atau *chosen plaintext attack.*

2. *Birthday Attack*

Birthday paradox

Dalam teori probabilitas, *Birthday problem / paradox* menyinggung probabilitas bahwa dalam suatu kelompok orang yang dipilih secara acak, satu atau beberapa pasangan dari mereka mempunyai *birthday /* ulang tahun yang sama. Dalam kelompok 23 atau lebih orang yang dipilih secara acak, ada kemungkinan 50% lebih satu atau beberapa pasangan diantara mereka mempunyai ulang tahun yang sama. Untuk grup 57 orang atau lebih, probabilitasnya dapat mencapai lebih dari 99%, walaupun tidak bisa benar benar 100% terkecuali terdaoat sedikitnya 366 orang dalam grup tersebut. Matematika dibalik paradox ini digunakan dalam serangan kriptografi bernama *birthday attack.*

Kunci untuk memahami paradox ini ialah dengan berpikir bahwa tidak ada dua orang yang berbagi hari ulang tahun. Yang kemungkinan ada ialah

kemungkinan orang pertama memiliki ulang tahun yang berbeda dengan orang kedua dan orang ketiga memiliki ulang tahun berbeda lagi dan orang keempat

berbeda pula 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.

Misal Alice ada di dalam ruangan dengan 23 orang dan mempunyai 22 kemungkinan untuk memiliki ulang tahun yang sama dengan orang lain (terdapat 22 pasangan dalam ruangan). Jika dia gagal mencocokkan ulang tahun, dia pergi dan orang selanjutnya misal, Bob mempunyai 21 kali kemungkinan untuk memiliki ulang tahun yang sama dengan orang lainnya. Jika dia gagal mencocokkan ulang tahun juga, dia pergi dan orang selanjutnya misal, Carol mempunyai 20 kali kemungkinan untuk mencocokkan dan seterusnya langkah ini diulang. 22 pasangan ditambah 21 pasangan ditambah 20…dan seterusnya sampai ditambah 1 sama dengan 253. Tiap pasangan memiliki 1/365 kemungkinan memiliki hari ulang tahun yang sama. Karena dalam tiap kelompok 23 orang terdapat 253 pasangan, dimana lebih dari setengah jumlah hari dalam setahun, kemungkinan salah satu pasangan memiliki ulang tahun yang sama tidak kecil. Untuk 28 orang, jumlah pasangan melebihi jumlah hari dalam setahun dan probabilitas salah satu pasangan memiliki ulang tahun yang sama lebih besar dari kelompok 23 orang.

Untuk menghitung aproksimasi probabilitas dalam ruangan yang terdiri dari n orang, dengan sedikitnya dua orang memiliki ulang tahun yang sama, variasi dalam distribusi seperti tahun kabisat diabaikan terlebih dahulu dan berasumsi bahwa 365 kemungkinan ulang tahun adalah sama. Distribusi ulang tahun dalam dunia nyata tidak seragam. Akan lebih mudah jika pertama mengkalkulasikan probabilitas $\underline{p}(n)$ dimana semua ulang tahun n berbeda. Jika $n \leq 365$, diberikan

$$\underline{p}(n) = 1 \,.\, (1 - 1/365) \,.\, (1 - 2/365)\ldots(1 - \{n-1\}/365) = \{365 \,.\, 364\ldots(365 - n + 1)\}/365^n = 365! \,/\, \{365^n (365 - n)!\}$$

Kejadian sedikitnya dua dari n orang memiliki ulang tahun yang sama adalah

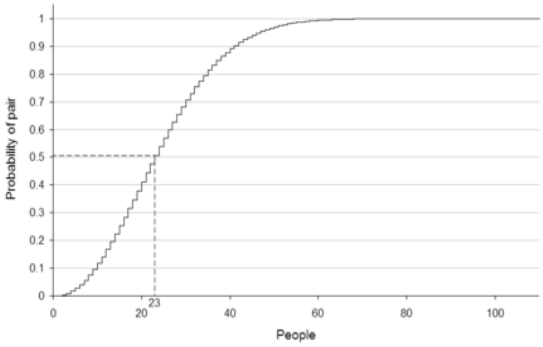
$$p'(n) = 1 - p(n)$$

Probabilitas ini melebihi 1/2 untuk n = 23 (dengan nilai sekitar 50.7 %). Tabel berikut ini menampilkan probabilitas untuk nilai n yang lain.

n	p'(n)
10	12%
20	41%
23	50.7%
30	70%
50	97%
100	99.99996%
200	99.999999999999999999999999998%
300	$(1 - 7\times10^{-73}) \times 100\%$
350	$(1 - 3\times10^{-131}) \times 100\%$
366	100%

Tabel 1: probabilitas sedikitnya dua dari n orang memiliki ulang tahun yang sama

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



Gambar 1: grafik sedikitnya dua dari n orang memiliki ulang tahun yang sama

Birthday attack

Digital Signature merupakan media yang rawan dalam Serangan *birthday attack.* Berikut contoh praktisnya. Misal Departemen ilmu computer universitas X mempunyai satu posisi untuk anggota fakultas dan dua kandidat, Tom dan Dick. Tom dua tahun lebih lama telah bekerja di tempat tersebut daripada Dick, jadi dia men dapat kesempatan terlebih dahulu. Jika dia mendapatkannya, Dick tidak akan mendapat apa apa. Tom tahu bahwa ketua departemen, Marylin, menaruh respek yang besar atas pekerjaannya, jadi dia meminta dia untuk menulis surat rekomendasi untuk dia kepada Dean, yang akan menentukan dalam kasus Tom. Setelah terkirim, semua surat menjadi *confidential.* Marylin menyuruh sekretarisnya, Ellen, untuk menulis kepada Dean surat rekomendasi untuk Tom, menguraikan apa saja yang dia inginkan disitu. Setelah selesai, Marylin akan me-*review* nya , memperhitungkan dan me-*sign* 64-bit *digest* dan mengirimnya kepada Dean. Ellen dapat mengirim suratnya kemudian melalui e-mail. Akan tetapi, Ellen yang merupakan teman baik Dick tidak ingin Tom mendapat posisi tersebut, jadi dia menulis surat dibawah ini (surat dalam bahasa inggris).

Dear Dean Smith,

This [*letter* | *message*] is to give my [*honest* | *frank*] opinion of Prof. Tom Wilson, who is [*a*

candicate | *up*] for tenure [*now* | *this year*]. I have [*known* | *worked with*] Prof. Wilson for [*about* | *almost*] six years. He is an [*outstanding* | *excellent*] researcher of great [*talent* | *ability*] known [*worldwide* | *internationally]* for his [*brilliant* | *creative*] insights into [*many* | *a wide variety of*] [*difficult* | *challenging*] problems.

He is also a [*highly* | *greatly*] [*respected* | *admired]* [*teacher* | *educator*]. His students give his [*classes* | *courses*] [*rave* | *spectacular*] reviews. He is [*our* | *the Department's*] [*most popular* | *best-loved*] [*teacher* | *instructor*].

[*In addition* | *Additionally*] Prof. Wilson is a [*gifted* | *effective*] fund raiser. His [*grants* | *contracts*] have brought a [*large* | *substantial*] amount of money into [*the* | *our*] Department. [*This money has* | *These funds have*] [*enabled* | *permitted*] us to [*pursue* | *carry out*] many [*special* | *important*] programs, [*such as* | *for example*] your State 2000 program. Without these funds we would [*be unable* | *not be able*] to continue this program, which is so [*important* | *essential*] to both of us. I strongly urge you to grant him tenure.

Segera setelah Ellen selesai mengetik surat tersebut, dia juga menulis surat kedua :

This [*letter* | *message*] is to give my [*honest* | *frank*] opinion of Prof. Tom Wilson, who is [*a candidate* | *up*] for tenure [*now* | *this year*]. I have [*known* | *worked with*] Tom for [*about* | *almost*] six years. He is a [*poor* | *weak*] researcher not well known in his [*field* | *area*]. His research [*hardly ever* | *rarely*] shows [*insight in* | *understanding of*] the [*key* | *major*] problems of [*the* | *our*] day.

Furthermore, he is not a [*respected* | *admired*] [*teacher* | *educator*]. His students give his [*classes* | *courses*] [*poor* | *bad*] reviews. He is [*our* | *the Department's*] least popular [*teacher* | *instructor*], known [*mostly* | *primarily*] within [*the* | *our*] Department for his [*tendency* | *propensity*] to [*ridicule* | *embarrass*] students [*foolish* | *imprudent*] enough to ask question in his classes.

[*In addition* | *Additionally*] Tom is a [*poor* | *marginal*] fund raiser. His [*grants* | *contracts*] have brought only a [*meager* | *insignificant*] amount of money into [*the* | *our*] Department. Unless new [*money is* | *funds are*] quickly located, we may have to cancel some essential programs, such as your State 2000 program. Unfortunately, under these [*conditions* | *circumstances*] I cannot in good [*conscience* | *faith*] recommend him to you for [*tenure* | *a permanent position*].

Setelah itu Ellen memprogram komputernya untuk mengkomputasi 2^{32} *message digest* dari tiap surat dalam satu malam. Kemungkinannya adalah , satu *digest* dari surat pertama akan bercocokkan dengan satu *digest* dari surat kedua. Jika tidak, Ellen dapat menambah beberapa opsi dan mencoba lagi padawaktu akhir pesan. Andai ia menemukan kecocokan. Misal surat "baik" disebut A dan surat "buruk" disebut B. Ellen sekarang mengirim surat e-mail kepada Marylin untuk disetujui. Surat B dia sembunyikan secara baik, dan tidak menunjukkannya kepada siapa siapa. Marylin kemudian menyetujui, dan mengkomputasi 64-bit *message digest* miliknya, me-*sign diges* nya dan mengirim melalui e-mail *digest* yang telah di-*sign* kepada Dean Smith. Secara terpisah, Ellen mengirim melalui e-mail surat B kepada Dean (bukan surat A yang memang seharusnya dia kirimkan). Setelah mendapat surat dan me-*sign message digest,* Dean menjalankan algoritma *message digest* pada surat B, yang kemudian dia melihat bahwa disetujui oleh Marylin dan kemudian memecat Tom. Dean tidak menyadari bahwa Ellen berhasil men-*generate* dua surat dengan *message digest* yang sama dan mengirim dia surat berbeda yang Marylin lihat dan setujui.

3. *Meet-In-the-Middle Attack*

Jika *chosen* atau *known* plainteks dan hasil dari enkripsi teksnya juga tersedia, serangan *Meet-In-the-Middle Attack* dapat dilakukan jika struktur dari algoritma enkripsinya memungkinkan. Dalam serangan *known plaintext,* penyerang mengetahui keseluruhan atau sebagian dari pesan yang sedang dienkripsi, mungkin sebagian *header* standar atau *trailer fields.* Dalam *chosen plaintext attack*, penyerang dapat memaksa sebagian plainteks yang dipilih untuk dienkripsi, mungkin dengan cara "membocorkan" teks yang akan dikirim oleh penyerang melalui *encrypted channel*.

Penjelasan sangat sederhana dari *Meet-In-the-Middle Attack* adalah sebagai berikut. Penyerang dapat mengenkripsi setengah *known* atau *chosen* plainteks dengan kemungkinan setengah kunci pertama, mengurutkan outputnya, dan kemudian mendekripsi setengah dari cipherteks dengan seluruh setengah kunci kedua. Jika terdapat kecocokan, kunci keseluruhan dapat dibangun dari setengah pesan yang digunakan untuk mendekripsi bagian lain dari pesan atau pesan lainnya. Kasus terbaik dalam hal serangan ini, serangan ini dapat memotong setengah dari eksponen dari proses serangan yang dilakukan oleh penyerang dan di lain pihak menambahkan usaha yang besar. *Meet-In-the-Middle Attack* menuntut penyerang memiliki pengetahuan yang mendalam tentang algoritma kriptografi. Walaupun algoritma dasar bukan merupakan subjek dari *Meet-In-the-Middle Attack* , usaha untuk menciptakan algoritma yang lebih kuat dengan menerapkan algoritma dasar dua kali (atau dua algoritma berbeda secara sekuensial) dengan kunci berbeda mungkin mendapatkan tambahan keamanan yang tidak sesuai harapan. Karena penggunaan cara ini merupakan subjek dari *Meet-In-the-Middle Attack.*

Untuk lebih jelas langkah langkah apa saja yang dilakukan dalam *Meet-In-the-Middle Attack,* kita mulai dari system yang didefinsikan sebagai berikut : dua system kriptografi ditunjukkan dalam $encrypt_\alpha$ dan $encrypt_\beta$ (dengan fungsi invers $decrypt_\alpha$ dan $decrypt_\beta$ secara berturut turut) dikombinasikan dengan mengaplikasikan yang satu kemudian yang lain untuk memberikan kriptosistem komposit. Masing masing menerima 64 bit kunci (untuk nilai dari 0 sampai 18446744073709551615) yang kita sebut key_α atau key_β.
Jadi dari plainteks yang diberikan, kita dapat mengkalkulasi kriptoteks sebagai

$$cryptotext = encrypt_\beta(key_\beta, encrypt_\alpha(key_\alpha, plaintext))$$

dan secara berkoresponden kita dapat mengkalkulasikan plainteks sebagai

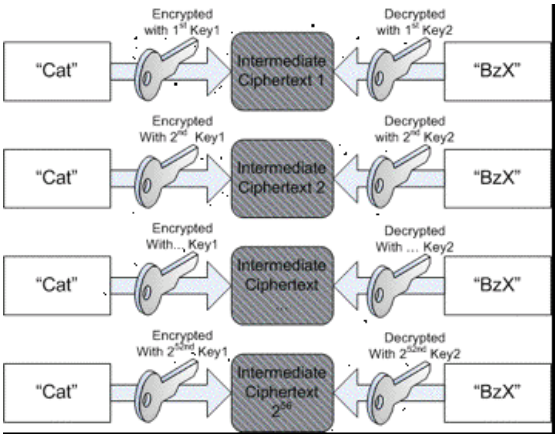
$$plaintext = decrypt_\alpha(key_\alpha, decrypt_\beta(key_\beta, cryptotext))$$

Sekarang, sesuai yang masing masing system telah diberikan sebanyak 64 bit kunci, jumlah kunci yang dibutuhkan untuk enkripsi dan dekripsi adalah 128 bit, jadi dengan analisis sederhana diasumsikan hal ini sama dengan 128 bit cipher.
Diberikan *storage* yang cukup, kita dapat mereduksi kekuatan kunci yang efektif menjadi beberapa bit lebih besar dibanding yang terbesar dari dua kunci yang digunakan sebagai berikut :

1. Diberikan pasangan plainteks dan cipherteks, gunakan $encrypt_\alpha$ plainteks dengan masing masing kunci yang mungkin, men-*generate* 2^{64} *intermediate cryptotext*..
   $cryptotext_1 \longrightarrow cryptotext_n$ dimana $n = 2^{64}$
2. Simpan tiap *n* cryptotexts dalam *hash table* dimana tiap teks dapat direferensikan dari cryptotext-nya, dan ambil kunci yang digunakan untuk men-*generate* cryptotext.
3. Gunakan $decrypt_\beta$ pada cipherteks untuk tiap kunci yang mungkin untuk kemudian membandingkan *intermediate plaintext* dengan *hash table* yang dibentuk sebelumnya. Didapatkan pasangan kunci (satu untuk tiap dua algoritma yang digunakan, α dan β)
4. Ambil dua kunci dari langkah 3, periksa masing masing dengan pasangan plainteks / cryptotext kedua. Jika terdapat kecocokan, kemungkina sangat tinggi telah didapatkan pasangan kunci yang valid untuk pesan. – bukan dalam operasi 2^{128}, tapi operasi $2x2^{64}$ (dimana secara signifikan lebih panjang dikarenakan operasi *hash table,* tapi tidak terlalu banyak)

Beberapa teknik kriptografi rawan terhadap *meet-in-the-middle attack.* Dua contoh yang cukup dikenal adalah *double*-DES *encryption* dan logaritma diskrit dengan *limited Hamming weight exponent.* Contoh lainnya adalah serangan pada skema untuk menggunakan *untrusted* server untuk melakukan sebagian besar pekerjaan dalam komputasi RSA.

Berikut contoh *meet-in-the-middle attack* yang sukses versus double DES. Untuk meningkatkan kekuatan dari 56 bit DES, double DES (dua kali enkripsi DES menggunakan dua kunci berbeda, dengan totak panjang kunci 112 bit) digunakan. Dalam contoh ini, missal plainteks nya adalah "Cat" dan hasil cipherteks double DES adalah "BzX". Penyerang ingin menemukan dua kunci (key1 dan key2) yang digunakan untuk enkripsi. Penyerang pertama melakukan *brute force attack* pada kay1 menggunakan 2^{56} kunci single-DES untuk mengenkripsi plainteks "Cat" dan menyimpan tiap kunci dan tiap hasil *intermediate* cipherteksnya pada table. Kemudian dilakukan *brute force* pada key2, mendeskripsi "BzX" sampai 2^{56} kali. Ketika *brute force attack* kedua mendeskripsi *intermediate* cipherteks pada table, serangan telah selesai dan kedua kunci telah diketahui oleh penyerang. Serangan memakan maksimum 2^{56} percobaan total. Ini jauh lebih mudah dibanding 2^{112} usaha.



Gambar 2: sketsa proses *meet-in-the-middle attack*

4. PEMBAHASAN

Dari perbandingan kedua serangan kriptografi diatas yaitu *Birthday attack* dan *meet-in-the-middle attack* ternyata kedua serangan tersebut memiliki kemiripan. Kemiripannya antara lain kedua serangan ini mengandalkan *brute force* dalam mencapai hasil. *Birthday attack* memanfaatkan probabilitas matematika dibalik paradox ulang tahun (*birthday paradox*) sementara *meet-in-the-middle attack* mengandalkan *overlapping* dari dua set nilai. Perbedaan mendasar dari dua serangan kriptografi ini cukup sederhana untuk ditunjukkan. *Birthday attack* hanya memperhatikan kemunculan duplikat dari suatu nilai dari satu set sedangkan *meet-in-the-middle attack* menunjukkan kejadian *overlapping* dari dua set.

Bila dibandingkan *meet-in-the-middle attack* merupakan serangan yang lebih fleksibel dibandingkan dengan *birthday attack* dan lebih banyak keuntungannya. Daripada menunggu suatu nilai muncul dua kali dalam satu set pada *birthday*

attack, penyerang / kriptanalis dapat mencek titik pertemuan dari dua set pada *meet-in-the-middle attack*. Sebagai contoh, misal penyerang memiliki pengetahuan yang memadai untuk jenis pesan plainteks dari seorang pengirim yang kebetulan menggunakan 64 bit kunci. Penyerang akan membangun satu dari dua set, dalam table, dengan mengkomputasi 2^{32} hasil *hash* yang unik untuk pesan plainteks yang sama, dengan kunci unik yang berasosiasi dengan tiap hasil *hash*, menggunakan fungsi *hash* yang sama sebagai pengirim. Kemudian penyerang akan "mengintip" tiap pesan untuk mencek apakah nilai *hash* yang terenkripsi ada dalam tabel nilai *hash* sebelumnya. Jika terdapat kesamaan, ada kemungkinan kunci yang berkorespondensi dalam table adalah benar. Ini berbeda dengan *birthday attack* dimana penyerang harus menunggu tanpa berbuat apa apa kemunculan duplikat sebuah nilai.

5. KESIMPULAN

Kesimpulan yang dapat ditarik dari pembahasan diatas adalah bahwa serangan kriptografi *birthday attack* dan *meet-in-the-middle attack* merupakan serangan kriptografi yang memiliki kemiripan dalam operasinya yaitu memakai *brute force* sebagai basis serangannya.
Meet-in-the-middle attack memiliki fleksibilitas yang lebih baikdibanding *birthday attack* namun sang kriptanalis dituntut untuk memiliki pengetahuan tentang pesan yang dikirim.

DAFTAR REFERENSI

[1] Munir, Rinaldi. (2006) *Diktat Kuliah IF5054 Kriptografi*. Institut Teknologi Bandung.
[2] Tanenbaum, A. S (2003) *Computer Networks*. Vrije Universiteit Amsterdam
[3] http://www.giac.org/resources/whitepaper/ cryptography/57.php
[4] www.cse.ohio-state.edu/cgibin /rfc/rfc1750.html