APLIKASI ALGORITMA *BRUTE FORCE* DALAM PROSES *CRYPTANALYSIS*

Kaisar Siregar

Prodi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung
Jl. Dipati Ukur No. 40, Bandung, Jawa Barat
e-mail: one_winged_angel_108@yahoo.com

ABSTRAK

Algoritma *Brute-Force* atau algoritma GT (*Generate and Test*) adalah algoritma trivial namun merupakan teknik penyelesaian masalah yang amat umum digunakan. Algoritma ini pada dasarnya mengenumerasi semua kandidat solusi yang ada dan memeriksanya apakah kandidat tersebut merupakan solusi dari permasalahan tersebut. Di lain pihak *Cryptanalysis* adalah metode untuk mendapatkan isi dari informasi yang telah terenkripsi tanpa memiliki akses ke suatu informasi rahasia yang diperlukan untuk mendekripsi informasi tersebut. Penggunaan algoritma *Brute*-Force dalam *Cryptanalysis* dikenal sebagai metode *Brute-Force Attack.* Adapaun penggunaan metode ini adalah sebagai patokan untuk memecahkan suatu teknik enkripsi. Apabila telah ditemukan metode lain yang mampu bekerja lebih cepat dibandingkan dengan metode *Brute*-Force, maka teknik enkripsi tersebut dapat dikatakan telah terpecahkan. Algoritma *Brute-Force* telah diimplementasikan ke berbagai mesin *Cryptanalysis*, salah satu contohnya adalah mesin *Deei Crack* dan *COPACOBANA*. Pada masa sekarang, metode *Brute-Force* Attack sudah hampir tidak dapat diaplikasikan, karena lama waktu komputasi yang sangat besar seiring semakin berkembangnya teknik enkripsi yang ada. Namun, bila faktor waktu dikesampingkan, algoritma *Brute-Force* akan selalu dapat menghasilkan hasil yang pasti.

Kata kunci: *Brute-Force Attack,* DES, ECC, RC4, RC5

1. PENDAHULUAN

Cryptanalysis adalah metode untuk mendapatkan isi dari informasi yang telah terenkripsi tanpa memiliki akses ke suatu informasi rahasia yang diperlukan untuk mendekripsi informasi tersebut. *Cryptanalysis* berkembang secara pararel dengan perkembangan kriptografi. Metode-metode dan teknik *Cryptanalysis* yang ada bertambah banyak dari waktu ke waktu guna mengadaptasi kerumitan dari teknik kriptografi yang ada, mulai dari metode pena dan kertas, mesin kriptografi sepert *Enigma* yang digunakan pada Perang Dnia ke-2 sampa skema kriptografi yang berbasiskan computer seperti yang digunakan pada masa sekarang.

Cryptanalysis dapat dilakukan dalam berbagai macam asumsi keadaan yang berkaitan dengan seberapa banyak informasi yang dapat ditemukan dan diamati dari suatu *Cryptosystem* yang sedang diserang. Sebagai standar untuk analisis, biasanya algoritma dasar dari *Cryptosystem* tersebut diasumsikan telah diketahui.

Adapun asumsi keadaan lain yang juga sering diimplementasikan dalam metode *Cryptanalysis* adalah:

- ***Ciphertext-only**:* Keadaan di mana sang *cryptanalyst* memiliki akses terhadap koleksi dari *ciphertext* atau *codetext* yang ada.
- ***Known-Plaintext***: Keadaan di mana sang *cryptanalyst* memiliki akses terhadap koleksi dari *ciphertext* dan *plaintext* pasangannya
- ***Chosen-Plaintext (Chosen-Ciphertext)***: Keadaan di mana sang *cryptanalyst* memiliki akses terhadap koleksi dari *ciphertext* dan *plaintext* pasangannya sesuai dengan pilihannya.
- ***Adaptive Chosen-Plaintext:*** Seperti *Chosen-Plaintext,* namun sang *cryptanalyst* dapat memilih beberapa *plaintext* berdasarkan informasi yang telah dipelajari dari beberapa enkripsi sebelumnnya

Hasil dari *Cryptanalysis* dapat dibedakan berdasarkan kegunaannya. Sebagai contoh, kriptografer Lars Knudsen (1998) mengklasifikasikan berbagai macam hasil tersebut sebagai beriku:

- ***Total Break:*** Sang *Cryptanalyst* dapat memecahkan kunci rahasia suatu *Cryptosystem* secara menyeluruh

- ***Global Deuction:*** Sang *Cryptanalyst* menemukan algoritma enkripsi dan dekripsi yang serupa secara fungsonalitas, namun tanpa mempelajari kunci rahasianya.

- ***Instance (local) Deduction:*** *Sang Cryptanalyst* menemukan informasi tambahan mengenai *plaintext* (atau *ciphertext*) yang sebelumnya tidak diketahui.
- ***Information Deduction:*** Sang *Cryptanalyst* mendapakan sejumlah *Shannon Information* (Informasi mengenai suatu ketidakpastian yang berkaitan dengan sebuah variabel acak)
- ***Distinguishing Algorithm:*** Sang *Cryptanalyst* dapat membedakan suatu *cipher* dari sebuahpermutasi acak.

Sebuah proses *Cryptanalysis* juga dapat dibedakan berdasarkan kekompleksan dari proses tersebut. Adapun kekompleksan yang dimaksud dapat berupa hal-hal sebagai berikut:

- Waktu : Banyaknya operasi-operasi primitif yang harus dilakukan dalam proses *Cryptanalysis*. Operasi primitive dapat berupa instruksi computer standar seperti penjumlahan, AND, XOR, dan sebagainya.
- Memori: Besarnya kapasitas memori yang diperlukan dalam proses *Cryptanalysis.*
- Dara: Kuantitas dari *plaintext* dan *ciphertext* yang diperlukan dalam proses *Cryptanalysis.*

Dalam konteks akademis, keadaan di mana *Cryptosystem* dapat dikatakan memiliki kelemahan atau telah berhasil dipecahkan telah didefinisikan secara jelas seperti yang telah dipaparkan oleh Bruce Scheiner. Adapun definisinya secara lengkap adalah sebagai berikut:

"*Breaking a cipher simply means finding a weakness in the cipher that can be exploited with a complexity less than brute force. Never mind that brute-force might require* 2^{128} *encryptions; an attack requiring* 2^{110} *encryptions would be considered a break....*" (Schneier, 2000).

Inti dari definisi tersebut adalah suatu *Cryptosystem* dapat dikatakan telah dipecahkan apabila telah ditemukan suatu metode lain yang memiliki kompleksitas lebih sedikit bila dibandingkan dengan metode *Brute-Force*. Alasan dipilihnya *Brute-Force* sebagai metode pembanding dikarenakan metode *Brute-Force* merupakan metode yang paling dasar dan pasti dalam memecahkan suatu *cipher* namum memerlukan waktu yang relatif lama.

2. ALGORITMA BRUTE FORCE

Algoritma *Brute-Force* atau algoritma GT (*Generate and Test*) adalah algoritma trivial namun merupakan teknik penyelesaian masalah yang amat umum digunakan. Algoritma ini pada dasarnya mengenumerasi semua kandidat solusi yang ada dan memeriksanya apakah kandidat tersebut merupakan solusi dari permasalahan tersebut.

Pada dasarnya, untuk mengaplikasikan algoritma *Brute-Force* pada sebuah permasalahan (P), haruslah diimplementasikan empat buah prosedur, yaitu *first, next, valid,* dan *output*.

- *First* (P): Membangkitkan sebuah kandidat solusi pertama untuk P
- *Next* (P, c): Membangkitkan kandidat solusi berikutnya setelah kandidat c dari P
- *Valid* (P, c): Memeriksa apakah kandidat solusi c merupakan solusi dari P
- *Output* (P, c): Menggunakan kandidat solusi c sebagai solusi dari masalah P sesuai dengan aplikasinya.

Prosedur *Next* harus pula dapat memberitahu apakah sudah tidak terdapat kandidat solusi lain untuk permasalahan P setelah kandidat solusi c. Metode umum yang paling digunakan adalah mengembalikan nilai *null* atau sebuah data Λ yang jauh dari kandidat-kandidat solosi lainnya. Prosedur *First* juga harus mengembalikan nilai Λ apabila tidak ada kandidat sama sekali untuk permasalahan P.

Algoritma *Brute-Force* dapat diekspresikan dengan algoritma sebagai berikut:

```
c ⟵ first(P)

while c ≠ Λ do
  if valid(P,c) then output(P, c)
  c ⟵ next(P,c)
```

Algoritma *Brute-Force* walaupun pasti menghasilkan solusi yang benar, namun memiliki kelemahan yang besar. Untuk setiap masalah, apabila jumlah kandidat solusi bertambah, maka kompleksitasnya akan meningkat secara besar. Hal ini dikenal dengan sebutan Ledakan Kombinatorial.

Sebagai contoh, bila kita ingin mencari suatu urutan susunan dari 10 buah huruf, maka kita akan memiliki kompleksitas waktu sebesar 10! = 3.628.800 kandidat solusi yang harus dibandingkan. Hal ini akan memakan waktu kurang dari sedetik untuk diselesaikan oleh komputer biasa. Namun, dengan menambahkan sebuah huruf, yang hanya merupakan 10% dari besar data

ssebelumnya, akan membesarkan kompleksitas waktunya sebesar 1000%. Untuk 20 buah huruf, maka besar kompleksitas waktunya adalah 20! Atau sekitar $2.4x10^{18}$ yang memerlukan waktu sekitar 10.000 tahun untuk diselesaikan. Fenomena inilah yang disebut sebagai Ledakan Kombinatorial

Algoritma *Brute-Force* dapat diaplikasikan ke hampi semua masalah yang ada mulai dari pengurutan, pencarian jalan terpendek, *travelling salesman problem*, dan juga pada *Cryptanalysis*. Metode *Cryptanalysis* yang menggunakan algoritma *Brute-Force* dikenal dengan sebutan *Brute-Force Attack.*

2.1 Gambaran Umum Metode *Brute-Force Attack*

Metode *Brute-Force Attack* adalah suatu metode untuk mengalahkan suatu skema kriptografik dengan cara mencoba kandidat-kandidat solusi yang berjumlah amat banyak. Sebagai contoh, secara menyeluruh memeriksa semua kemungkinan kunci yang ada untu mendekripsi suau informasi. Secara teori semua skema kriptografi dapat dipecahkan oleh metode *Brute-Force Attack* namun pada dasarnya sebuah skema kriptografi dirancang dengan sedemikian rupa sehinga teori tersebut tidak dapat dilaksanakan secara komputasi .

Penggunaan utama dari metode *Brute-Force Attack* dalam *Cryptanalysis* adalah sebagai patokan untuk memecahkan suatu teknik enkripsi. Apabila telah ditemukan metode lain yang mampu bekerja lebih cepat dibandingkan dengan metode *Brute*-Force, maka teknik enkripsi tersebut dapat dikatakan telah terpecahkan.

Metode *Brute-Force Attack* mempunyai kasus normal apabila kunci rahasia ditemukan pada pencarian ke $0.5n$. Di mana *n* adalah besarnya *key space* yang ada. Misal ada 2^{64} kemungkinan kunci, sebuah metode *Brute-Force Attack* diharapkan akan menemukan jawabannya setelah 2^{63} kali percobaan. Sehingga dapat dihitung nilai O besar dari metode *Brute-Force Attack* adalah sebesar $O(2^n)$.

Untuk setiap percobaan kunci yang dilakukan, sang *Cryptanalyst* harus mamupu mengenali apakah dia telah menemukan kunci yang benar. Hal ini mungkin terjadi apabila salah satu atau beberapa keadaan yang diasumsikan pada proses *Cryptanalysis* telah dipenuhi (merujuk pada asumsi yang telah dipaparkan di Bab 1).

Pada masa sekarang, suatu kunci dengan panjang sebesar 64 bit telah dapatdipecahkan dengan metode *Brute-Force Attack*. Metode enkripsi *Data Encryption Standard* (DES) yang menggunakan kunci dengan panjang 56 bit telah mampu dipecahkan pada tahun 1998 dan sebuah pesan yang di enkripsi menggunakan RC5 (64 bit kunci) juga telah dipecahkan baru-baru ini oleh distributed.net juga dengan metode *Brute-Force Attack.*

Pada umumnya, metode *Brute-Force Attack* tidak dilakukan oleh sebuah komputer biasa, mengingat banyaknya kemungkinan yang harus diperiksa, namun oleh sebuah mesin pemecah kode (*code breaker machine*) yang mengimplementasikan metode ini. Beberapa saat yang lalu, telah dibuat sebuah mesin dengan nama *COPACOBANA* (*Cost-Optimized Parallel Code Breaker*), yaitu mesin berbasiskan metode *Brute-Force Attack* yang mampu bekerja mencari kunci dari berbagai jenis teknik enkripsi, termasuk DES.

Secara praktikal metode *Brute-Force Attack* tidak dapat diimplementasikan pada kunci yang memiliki panjang lebih dari 128 bit seperti yang ada pada *Advance Encryption Standard* (AES). Hal ini dikarenalan adanya argument secara fisika yang dikenal dengan *Von Neumann-Landauer Limit* yang mengimplikasikan batas minimum energy yang diperlukan untuk melakukan sebuah komputasi, yaitu

$$E = \ln(2)kT \qquad (1)$$

Dimana E adalah Energi per bit dalam Watt, T adalah temperatur media komputasi dalam Kelvin, *k* adalah konstanta Boltzmann.

Menurut rumus tersebut, jumlah energi yang diperlukan oleh metode *Brute-Force* dalam mendapatkan kunci dengan panjang 128 bit adalah sekitar 10 gigawatt (setara dengan energy yang dihasilkan 8 reaktor nuklir). Selain itu, dari segi waktu pemecahan pencarian kunci dengan 128 bit juga tidak memungkinkan. Jumlah pencarian yang dicari adalah sebanyak 2^{128} = 340.282.366.920.938.463.463.374.607.431.768.211.456 kemungkinan. Dengan sebuah alat yang mampu mengecek 10^{18} kemungkinan per detik pun memerlukan waktu sekitar 10^{13} tahun untuk menyelesaikan pencariannya.

2.2 Penggunaan Metode *Brute-Force Attack* dalam Pencarian Kunci Berbagai Metode Enkripsi

Metode *Brute-Force Attack* telah mampu memecahkan berbagai algoritma enkripsi yang menggunakan kunci yang panjangnya kurang dari 128 bit. Beberapa algoritma yang dimaksud adalah RC4, RC5, ECC, dan DES.

2.2.1. Metode *Brute-Force Attack* dalam Pencarian Kunci RC4

RC4 (atau dikenal juga dengan ARC4 atau ARCFOUR) adalah sebuah software enkripsi yang telah umum digunakan dan digunakan untuk protokol-protokol popular seperti *Secure Sockets Layer* (SSL). RC4 sangat ppuler karena kecepatannya dan kesederhanaannya dalam mengenkripsi suatu informasi.

Metode enkripsi RC4 dengan panjang kunci 40 bit telah berhasil dipecahkan dengan metode *Brute-Force Attack* di antaranya dilakukan oleh Adam Back, dkk, Damien Doligez, dan Ian Goldberg

Pada tahun 1995, Adam Back, David Byers, and Eric Young menggunakan beberapa workstation (di mana salah satunya adalah sebuah *super computer* MasPar Mp-1) untuk memecahkan RC4. Hasil yang didapat adalah sebagai berikut

Tabel 1. Hasil Pencarian Kunci RC4 oleh Adam Back, dkk

Waktu Pencarian	8 hari
Space yang diperiksa	Lebih dari 99,6%
Kecepatan Maksimum	1,924,000 kunci/dtk ($2^{20.88}$ kpd)
Kecepatan MasParMP-1	1,500,000 kunci/detik ($2^{20.52}$kpd)

Pada waktu yang sama Damien Doligez menggunakan mesin-mesin yang terdapat di INRIA, Ecole Polytechnique, dan ENS untuk memecahkan kunci RC4 yang sama pula. Di dapat hasil hampir serupa dengan Adam Back, dkk.

Tabel 2. Hasil Pencarian Kunci RC4 oleh Damien Doligez

Waktu Pencarian	8 hari
Space yang diperiksa	sekitar 50%
Kecepatan Rata-Rata	850,000 kunci/dtk ($2^{19.70}$ kpd)
Kecepatan Maksimum	1,350,000 kunci/dtk ($2^{20.36}$ kpd)

Pada bulan Januari 1997 RSA mengajukan tantangan untuk memecahkan berbagai kunci dengan panjang yang berbeda-beda. Ian Goldberg yang menggunakan *Berkeley NOW Clusters* dan beberapa mesin lainnya mampu menemukan kunci RC4 dengan panjang 40 bit dalam waktu hanya 3,5 jam. Berikut adalah detail hasilnya:

Tabel 3. Hasil Pencarian Kunci RC4 oleh Ian Goldberg

Waktu Pencarian	3,5 jam
Kecepatan Rata-Rata	~28,000,000 kunci/dtk (~2^{25} kpd)

2.2.2. Metode *Brute-Force Attack* dalam Pencarian Kunci RC5

RC5 adalah teknik enkripsi yang merupakan pengembangan dari RC4 yang juga dikembangkan oleh Ronald Rivest pada tahun 1994. RC merupakan dari *Rivest Cipher* atau dapat juga menjadi *Rin's Code.* RC5 merupakan dasar dari *Advance Encryption Standard* (AES). 3 Jenis panjang kunci RSA yang telah berhasil ditemukan dengan metode *Brute-Force Attack* adalah RC5 48 bit, RC5 56 bit, dan RC5 64 bit.

Kunci RC5 48 bit berhasil ditemukan pada tahun 1997 oleh Caronni Group ("*The Distributed Internet Crack*") dalam waktu 13 hari. Detail penemuannya adalah sebagai berikut:

Tabel 4. Hasil Pencarian Kunci RC5 48 bit oleh Caronni Group

Waktu Pencarian	13 hari
Space yang diperiksa	57.58%
Kecepatan Rata-Rata	144,220,000 kunci/dtk($2^{27.10}$ kpd)
Kecepatan Maksimum	440,000,000 kunci/dtk ($2^{28.71}$ kpd)
Mesin yang terlibat	7000 secara keseluruhan

Sedangkan kunci RC5 56 bit juga ditemukan pada tahun yang sama. Penemuan ini dilakukan oleh Bovine Group (yang nantinya akan dikenal dengan nama distributed.net). Pencarian kunci ini memakan waktu selama 250 hari. Detail pencariannya adalah sebagai berikut:

Tabel 5. Hasil Pencarian Kunci RC5 56 bit oleh Bovine Group

Waktu Pencarian	270 hari
Space yang dicari	47%
Kecepatan Rata-Rata	1,467,000,000 kunci/dtk ($2^{30.45}$ kpd)
Kecepatan Maksimum	7,000,000,000 kunci/dtk ($2^{32.70}$ kpd)
Mesin yang terlibat	Puluhan ribu mesin

Kunci RC5 64 bit mulai dicari pada tahum 1997, namun memerlukan waktu 1470 hari untuk memeriksa 60% dari jumlah *key* space yang ada. Berikut adalah detailnya:

Tabel 6. Hasil Pencarian Kunci RC5 64 bit

Waktu Pencarian	1470 hari
Space yang diperiksa	60%
Kecepatan Rata-Rata	88,000,000,000 kunci/dtk($2^{36.36}$kpd)

2.2.3. Metode *Brute-Force Attack* dalam Pencarian Kunci ECC

Elliptic Curve Cryptography (ECC) adalah metode kriptografi kunci publik yang berdasarkan struktur aljabar dari sebuah kurva elips pada suatu area berhingga. Penggunaan kurva elips pda kriptografi pertama kali disarankan oleh Neal Koblitz dan Victor S. Miller pada tahun 1985.

Sebuah perusahaan bernamaCerticom telah megajukan sekumpulan tantangan pencarian dengan metode *Brute-Force Attack* untuk menemukan kunci ECC dengan panjang 109, 131, 163, 191, 239, dan 359 bit. Kunci dengan panjang 109 bit telah diselesaikan pada bulan April 2000.

Tabel 7. Hasil Pencarian Kunci ECC 109 bit

Waktu Pencarian	120+ hari
Koordinat yang diperiksa	$2{,}5 \times 10^{15}$ koordinat yang berbeda
Mesin yang terlibat	9500 secar keseluruhan, 5000 yang aktif secara bersamaan

2.2.4. Metode *Brute-Force Attack* dalam Pencarian Kunci DES

Data Encryption Standard (DES) dikembangkan pada dekade 1970-an oleh IBM. DES secara resmi diadopsi oleh NIST (*National Institute of Standards and Technology*) sebagai standar algoritmaenkripsi pada tahun 1976. Semenjak itu DES telah menjadi kiblat teknik kriptografi di pasaran. Namun seiring berkembangnya kemampuan komputasi dalam proses *Cyptanalysis* DES menjadi relatif mudah untuk dipecahkan, bahkan oleh metode *Brute-Force Attack* sekalipun. Pada tahun 1997 secara resmi DES tidak dipergunakan lagi oleh NIST.

Walaupun DES menggunakan kunci dengan panjang 64 bit, namun pada praktikalnya panjang efektif dari kunci DES hanyalah 56 bit. Hal ini dikarenakan karena 8 bit yag hilang tersebut dijadikan sebagai bit paritas. Pada perancangan DES, NSA menambahkan sebuah *S-Box* rahasia. Pada saat metode *Differential Cryptanalysis* telah ditemukan, ditemukan bahwa *S-Box* pada DES berfungsi sebagai resistor metode tersebut.

Sudah terdapat beberapa mesin yang bekerja berdasarkan metode *Brute-Force Attack* yang telah mampu menemukan kunci DES 56 bit yang ada.

Pada tahun 1977 Whit Diffie dan Martin Hellman mempublikasikan sebuah desain mesin seharga $20.000.000 yang mampu menemukan sebuah kunci DES tiap harinya dengan kecepatan pencarian hingga 2^{38} kunci/detik nya. Beberapa dekade kemudian, pada tahun 1993 Michael Wiener mempublikasikan sebuah mesin yang memiliki harga desain senilai $500.000 dan dengan biaya pembuatan seharga $100.000 dapat menemukan sebuah kunci DES dalam waktu 35 jam (kecepatan pencarian: $2^{38,07}$ kunci per detik). Namun apabila biaya pembuatan dinaikkan menjadi $1.000.000 mesin ini akan dapat menemukan kunci DES dalam waktu 3.5 jam (kecepatan pencarian: $2^{41.39}$ kunci per detik) dan dengan biaya pembuatan sebesar $10.000.000 mesin tersebut akan mampu menemukan kunci DES dalam waktu 21 menit. (kecepatan pencarian: $2^{44,71}$ kunci per detik).

Selain kedua konsep mesin di atas telah dibuat juga beberapa mesin berbasis *Brute-Force Attack* yang telah berhasil menemukan kunci DES dalam waktu yang relatif cepat. Di antaranya adalah *Deep Crack* dan *COPACOBANA*.

Deep Crack merupakan suatu chip kripto yang disponsori oleh EFF. Chip ini memiliki kemampuan untuk memproses 88 milyar kunci per detiknya. *Deep Crack* telah terbukti dapat menemukan kunci DES 56 bit dalam waku kurang dari 3 hari. Pencarian ini dilakukan pada bulan Juli 1998 dengan hasil sebagai berikut:

Tabel 8. Hasil Pencarian Kunci DES 56 bit oleh *Deep Crack*

Waktu Pencarian	56.05 jam
Space yang dicari	24.8%
Kecepatan Rata-Rata	88,804,000,000 kunci/dtk($2^{36.37}$kpd)

Deep Crack merupakan mesin *Cryptanalysis* pertama yang dibuat dan dijalankan. Deskripsi mesin ini didokumentasi secara penuh dalam sebuah buku setebal 268 halaman.

Berbeda dengan *Deep Crack*, *COPACOBANA* merupakan mesin *Cryptanalysis* yang komponen-komponennya dijual bebas di pasaran dan terjangkau secara ekonomis. *COPACOBANA* dikembangkan oleh sebuah tim di Universitas Bochum dan Kiel yang berdomisili di Jerman. *COPACOBANA* terdiri dari 120 FPGA XILINX Spartan3-1000 yang bekerja secara parallel. Kumpulan FPGA ini dikelompokkan menjadi 20 DIMM modul yang masing-masing terdiri dari 6 FPGA. Sebuah mesin *COPACOBANA* dapat dibuat dengan biaya hanya sekitar $10.000 saja. Biaya ini hanya merupakan 1/25 dari biaya pembuatan *Depp Crack*.

Pada tanggal 15 Maret 2007, *COPACOBANA* telah mampu menemukan sebuah kunci DES dengan metode *Brute-Force Attack* hanya dalam waktu rata-rata 6,4 hari saja. Adapun waktu skenario terburuk pemncarian sebuah kunci DES dapat ditemukan dalam waktu 12,8 hari oleh *COPACOBANA*.

3. KESIMPULAN

- Algoritma *Brute-Force* pada dasarnya mengenumerasi semua kandidat solusi yang ada dan memeriksanya apakah kandidat tersebut merupakan solusi dari permasalahan tersebut.
- Aplikasi algoritma *Brute-Force* dalam *Criyptanlysis* dikenal dengan sebutan *Brute-Force Attack*.
- Metode *Brute-Force Attack* secara teori mampu memecahkan semua metode enkripsi yang ada, namun pada praktik nya metode ini kurang efektif dari segi kompleksitas waktu.
- Keefektifan Metode *Brute-Force Attack* amat bergantung pada kecepatan mesin/media yang mengimplementasikan metode tersebut.
- Sampai saat ini Metode *Brute-Force Attack* telah dapat menemukan sejumlah kunci enkripsi dari berbagai algoritma enkripsi seperti DES, RC4, RC5, dan ECC yang memiliki panjang kunci kurang dari 128 bit.

REFERENSI

[1] Munir, Rinaldi, Diktat Kuliah IF2251 Strategi Algoritmik, Penerbit ITB 2007

[2] Brute Force Search, http://en.wikipedia.org/wiki/Brute_force_search.html. Waktu Akses: 12 Mei 2008 Pukul 19.08

[3] Brute Force Attack, http://en.wikipedia.org/wiki/Brute_force_attack.html. Waktu Akses: 12 Mei 2008 Pukul 19.10

[4] RC4, http://en.wikipedia.org/wiki/RC4.html. Waktu Akses: 12 Mei 2008 Pukul 19.11

[5] RC5, http://en.wikipedia.org/wiki/RC5.html. Waktu Akses: 12 Mei 2008 Pukul 19.13

[6] ECC, http://en.wikipedia.org/wiki/Eliptic_curve_cryptography.html. Waktu Akses: 12 Mei 2008 Pukul 19.14

[7] DES, http://en.wikipedia.org/wiki/Data_Encryption_Standard.html. Waktu Akses: 12 Mei 2008 Pukul 19.17

[8] Cryptanalysis, http://en.wikipedia.org/wiki/Cryptoanalysis.html. Waktu Akses: 12 Mei 2008 Pukul 19.21

[9] Brute Force Cracking Data, http://www.cl.cam.ac.uk/~rnc1/brute.html. Waktu Akses: 14 Mei 2008 Pukul 10.23