Keamanan Blok *Cipher* dengan Algoritma COBRA-F64

Amorita Kurnia Dewi, Nia Kaniawati, dan Rizki Hustiniasari

Departemen Teknik Informatika
Institut Teknologi Bandung
Jalan Ganesha 10 Bandung 40132

E-mail : if10012@students.if.itb.ac.id, if10059@students.if.itb.ac.id, if11068@students.if.itb.ac.id

Abstrak

Algoritma yang aman dari serangan *cryptanalysis* menjadi sebuah kebutuhan yang dicoba untuk dijawab. Makalah ini mempelajari tentang metode *cryptanalysis* terhadap blok *cipher* hasil enkripsi, dan mengukur tingkat keamanan dari algoritma yang digunakan untuk mengenkripsi. Algoritma yang dikaji di sini adalah algoritma COBRA-F64 yang ingin diukur tingkat keamanannya terhadap metode *cryptanalysis* yang disebut: *Differential cryptanalysis*.

Kata kunci : kriptografi, *cryptanalysis*, keamanan, COBRA-F64, blok *cipher*

1. Pendahuluan

Saat ini, di mana teknologi informasi dan komputer semakin berkembang luas, kriptografi memegang peranan yang sangat penting dalam menjaga kerahasiaan informasi. Persoalan di bidang kriptografi modern meliputi pengembangan sistem tanda tangan digital, sistem voting komputerisasi, pengendalian protokol otentikasi user, pencegahan terhadap kesalahan pesan, dan sebagainya.

Banyak permasalahan aktual di bidang teknologi perangkat lunak yang efektif diselesaikan menggunakan metode kriptografi. Di dalam kriptografi, kita harus menganggap adanya pihak-pihak jahat (seperti musuh, kriptanalis, penyadap, pengguna yang tidak berhak) yang mengetahui algoritma kriptografi, protokol, dan metode yang digunakan, serta mencoba untuk membahayakannya. Perilaku yang membahayakan *cryptosystem* dapat berupa pembacaan data oleh pihak yang tidak berhak, pemalsuan tanda tangan seseorang, pengubahan hasil voting, pelanggaran kerahasiaan voting, atau pengubahan data yang tidak akan dideteksi oleh penerima. Perilaku musuh ini secara umum disebut sebagai serangan kriptografi (atau serangan). Salah satu ciri kriptografi adalah kriptografi mengarahkan pada pengembangan metode yang melindungi kita melawan serangan musuh. Tetapi saat kita merancang *cryptosystem*, adalah tidak mungkin untuk meramalkan tipe-tipe serangan yang akan bermunculan di masa yang akan datang, berdasarkan kemajuan teori dan teknologi. Permasalahannya adalah seberapa dapat diandalkan solusi pada masalah kriptografi tertentu? Jawaban dari pertanyaan ini berhubungan langsung dengan perkiraan usaha yang diperlukan untuk menyerang *cryptosystem*. Penyelesaian masalah ini cukup kompleks, disebut *cryptanalysis*. Kriptografi dan *cryptanalysis* dipelajari dalam ilmu yang bernama *cryptology*.

Penggunaan teknologi komputer di dalam pemrosesan data dan sistem pengendalian telah memperburuk masalah dalam perlindungan informasi dari akses pihak yang tidak berhak. Perlindungan informasi pada sistem komputer memiliki fakta bahwa informasi tidak terbatas kaku pada medium – informasi dapat dengan mudah disalin dan ditransmisikan melalui saluran komunikasi. Terdapat banyak ancaman informasi yang terkenal, dan ancaman tersebut dapat diimplementasikan baik oleh musuh dari dalam maupun dari luar.

Solusi dasar untuk masalah perlindungan informasi yang mengalir pada sistem komputer berkinerja tinggi dapat diperoleh dengan metode

kriptografi. Dalam kasus ini, adalah penting untuk menggunakan algoritma penyandian yang cepat di mana tidak akan mengurangi kinerja komputer atau sistem komunikasi. Transformasi data dalam kriptografi bersifat fleksibel dan efektif untuk memberikan privasi data, keutuhan data, dan otentikasi data. Penggunaan metode kriptografi yang dikombinasikan dengan metode teknologi dan organisasi dapat memberi perlindungan terhadap serangan yang kemungkinan besar terjadi dan dalam lingkup yang luas.

Salah satu hal yang penting, penggunaan teknologi informasi ini merupakan penyelesaian efektif dari masalah perlindungan sumber daya informasi nasional yang memerlukan perlindungan data terdistribusi untuk semua pengguna. Prinsip ini pantas dan efektif, yaitu melindungi kepentingan organisasi individu merupakan dasar dari melindungi kepentingan nasional.

2. Masalah-masalah di dalam Kriptografi

Keamanan sistem kriptografi modern tidak berdasarkan pada kerahasiaan algoritma, tetapi berdasarkan kerahasiaan kunci. Kunci ini digunakan untuk mengendalikan proses transformasi kriptografi (*ciphering*), dan kunci ini mudah diubah dalam *cryptosystem*. Pengguna dapat mengubah kunci kapan saja.

Criptanalysis modern mengklasifikasikan serangan pada sistem enkripsi berdasarkan pada data-data yang telah diketahui berikut :

- *Ciphertext*
- *Plaintext* dan *corresponding ciphertext*
- *Chosen plaintext*
- *Chosen ciphertext*
- *Adapted plaintext*
- *Adapted ciphertext*
- *Hardware faults*
- *Power consumption measurements*
- *Calculation time measurements*

Dalam mengembangkan suatu sistem keamanan komputer yang bersifat umum, aspek perlindungan informasi secara teknik menjadi hal yang sangat penting. Salah satu kerumitan dari masalah ini adalah perlunya membangun sistem keamanan yang memungkinkan pengguna untuk melakukan konfigurasi terhadap sistem tersebut sesuai dengan kondisi operasional tertentu.

Dilihat dari sudut pandang efisiensi ekonomi, suatu sistem keamanan sebaiknya memenuhi spesifikasi berikut ini :

- *Cost* yang rendah
- Kemudahan dalam pemeliharaan dan pengoperasian sistem sehingga jumlah staf teknik yang dipekerjakan dapat dikurangi
- Fungsionalitas berskala penuh yang memungkinkan pengurangan jumlah alat-alat keamanan yang harus digunakan untuk memenuhi aspek keamanan secara keseluruhan
- Memungkinkan pengaktifan alat-alat keamanan tanpa menghentikan pemrosesan data yang sedang berlangsung.
- Beroperasi dengan mode *real-time*

Untuk mengembangkan sistem keamanan yang memenuhi spesifikasi di atas, diperlukan peninjauan terhadap prinsip-prinsip perancangan sebagai berikut :

- *Total disk encryption*
- *Multilevel encryption (disk encryption, transparent file encryption, encryption on demand)*
- Menjaga agar sistem operasi dasar tidak dilibatkan
- Pengontrolan terhadap integritas informasi dengan mode *real-time*

3. Perancangan *Fast Ciphers* Berdasarkan pada *Controlled Operations*

Ada beberapa *cipher* blok yang dapat digunakan, antara lain :

- SPECTR-H64
- SPECTR-128
- CIKS-128
- COBRA-F64a dan COBRA-F64b
- DDP-S64 dan DDP-S128
- COBRA-F64a dan COBRA-F64b

Berikut ini penjelasan dua buah *cipher* blok iteratif, COBRA-F64a dan COBRA-F64b, dengan masukan 64 bit dan 128 bit kunci rahasia, keduanya ditujukan pada implementasi

firmware. Algoritma-algoritma berikut dijelaskan berdasarkan penandaan pada cipher blok.

3.1 Skema Enkripsi Umum

Dalam *cipher* blok COBRA-F64a dan COBRA-F64b digunakan sebuah skema enkripsi yang tidak bersifat universal. Perlu diketahui bahwa sebuah skema bersifal universal jika perubahan pada mode enkripsi (enkripsi and dekripsi) hanya membutuhkan perubahan urutan pada putaran kunci. Namun, skema enkripsi pada algoritma COBRA-F64a dan COBRA-F64b dapat disebut semi-universal, karena satu-satunya perbedaan adalah penggunaan perintah baru - P(U, e)32/32, yang bergantung pada parameter e. Dengan menggunakan superscript "(e)", skema enkripsi umum dapat dituliskan dengan rumus berikut :

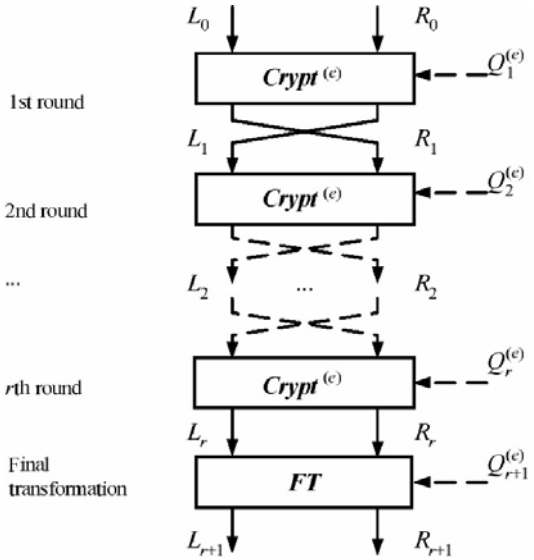
$$Y = \mathrm{F}^{(e)}(X, Q^{(e)}) \qquad (3.1)$$

di mana Q(e) = H(K, e) adalah sebuah *extended key*, dan sebuah fungsi dari 128 bit kunci K dan mode enkripsi e (F(0) - enkripsi, F(1) - dekripsi), di mana dalam mode enkripsi, X (X = GF(2)32) adalah blok masukan yang berupa data biner (*plaintext*), dan dalam mode dekripsi X adalah blok transformasi yang juga berupa data biner (*ciphertext*). Dalam mode enkripsi, nilai hasil Y (Y = GF(2)32) adalah sebuah *ciphertext*, dan dalam mode dekripsi adalah *plaintext*.

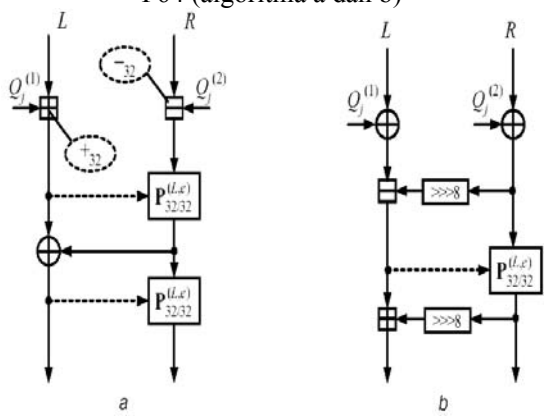
Sebuah *extended key*, Q(e) = H(K, e), harus dibentuk menggunakan suatu prosedur pembangkitan *extended key* yang aman secara kriptografis. Sebenarnya, cukup mudah untuk menciptakan sebuah barisan *pseudo-random* S = Q(0) = H(K, 0) yang panjangnya bergantung pada jumlah putaran dan panjang total dari satu *round key*.

Extended key Q(1) dihasilkan dari permutasi Q(0), permutasi ditentukan oleh fitur-fitur yang membentuk skema, termasuk transformasi awal dan akhir.

Skema umum dari transformasi ditunjukkan pada Gambar 1, dan skema satu putaran dari algoritma COBRA-F64a dan COBRA-F64b ditunjukkan pada Gambar 2.



Gambar 1 Skema enkripsi umum pada COBRA-F64 (algoritma a dan b)



Gambar 2 Prosedur Crypt(e) pada : a - COBRA-F64a dan b - COBRA-F64b

Tidak adanya transformasi awal adalah fitur dari skema umum enkripsi. Enkripsi dan dekripsi dilakukan dengan menggunakan algoritma yang sama. Hal ini disebabkan karena struktur dari setiap putaran mengandung transformasi akhir sebagai salah satu komponennya.

3.2 Aturan Penjadwalan dari Penggunaan *Round Keys*

Pada algoritma COBRA-F64a dan COBRA-F64b, dua buah 32-bit subkeys digunakan pada satu putaran dan transformasi akhir,

$$\mathbf{Q}_j^{(e)} = (Q_j^{(1,e)}, Q_j^{(2,e)}) \qquad (3.2)$$

dimana $1 \le j \le r + 1$ dan r adalah jumlah putaran pada proses enkripsi.

Pada kenyataannya penggunaan prosedur *extended key generation* yang aman sulit diterapkan. Cara yang lebih efisien adalah dengan menggunakan penjadwalan kunci.

Sebagai contoh, misalkan kunci privat K direpresentasikan sebagai kombinasi dari 4 word 32-bit, K = K0 □ K1 □ K2 □ K3, dimana K0, K1, K2, K3 ε GF(2)32. Sedangkan *extended key* Q(0) = S direpresentasikan dalam Tabel 1.

Tabel 1 Penjadwalan dalam Penggunaan *Round Keys*.

j =	1	2	3	4	5	6	7	8
S(1)i =	K1	K2	K3	K4	K2	K1	K4	K3
S(2)j =	K4	K3	K1	K2	K3	K2	K1	K4

j =	9	10	11	12	13	14	15	16
S(1)i =	K1	K2	K4	K3	K1	K4	K2	K3
S(2)j =	K2	K3	K1	K2	K3	K1	K3	K4

j =	17	18	19	20	21
S(1)i =	K1	K4	K3	K1	K2
S(2)j =	K2	K1	K4	K2	K3

Nilai dari *round key* Q(e)j = (Q(1, e)j, Q(2, e)j), untuk proses enkripsi dan dekripsi ditentukan oleh Tabel 1 dan formula berikut :

$$(Q_j^{(i,e)}, Q_{r\perp j}^{(i,e)}) = \mathrm{P}_{2..32/1}^{(c)}(S_j^{(1)}, S_{r\perp j}^{(2)}) \qquad (3.3)$$

dimana i = 1, 2 dan j = 1, 2, …, r.

Dengan mempertimbangkan properti kriptografis dari operasi-operasi yang dilakukan dan fitur-fitur yang membangun satu putaran, 16 putaran dalam proses enkripsi adalah jumlah yang optimal untuk algoritma COBRA-F64a, sedangkan untuk COBRA-F64b adalah 20. Jumlah putaran enkripsi tersebut dapat ditambah jika level keamanan yang dibutuhkan lebih rendah.

3.3 Algoritma Enkripsi

Fungsi F diimplementasikan sebagai rangkaian dari prosedur-prosedur berikut:

- r putaran dari transformasi yang menggunakan prsedur Crypt(e)
- Transformasi akhir - FT

Pada awalnya, blok X dibagi menjadi dua subblok dengan panjang yang sama - L0 dan R0, (L0, R0) = X, dimana L0, R0 □ GF(2)32. Kemudian dilakukan r putaran transformasi menggunakan prosedur Crypt(e) sesuai dengan formula berikut ini :

$$L_j = Crypt(R_{j..1}, L_{j..1}, Q_j^{(e)});$$
$$R_j = L_{j-1}, \quad (j = 1,2,...,r). \qquad (3.4)$$

Setelah putaran ke-r, transformasi akhir FT dilakukan terhadap blok X = (Rr, Lr) menurut formula sebagai berikut :

$$Y = FT(X, Q_{FT}^{(c)}). \qquad (3.5)$$

Transformasi akhir FT untuk algoritma COBRA-F64a :

$$Y = (L_{r+1}, R_{r+1}) = (R_r -_{32} Q_{r+1}^{(1,e)}, L_r -_{32} Q_{r+1}^{(2,e)}). \qquad (3.6)$$

Dan untuk algoritma COBRA-F64b, transformasi akhir FT adalah sebagai berikut:

$$Y = (L_{r+1}, R_{r+1}) = (R_r \oplus_{32} Q_{r+1}^{(1,e)}, L_r \oplus_{32} Q_{r+1}^{(2,e)}). \qquad (3.7)$$

3.4 Parameter Kecepatan dan Keamanan Kriptografis dari Ciphers

Keuntungan dari algoritma COBRA-F64a and COBRA-F64b adalah kode biner yang pendek saat diimplementasikan dalam bentuk perangkat lunak, sangat penting dalam merancang sistem kriptografi yang akan diintegrasikan ke dalam prosedur inisialisasi yang aman paa komputer.

Implementasi dalam bentuk *microprogram* dari algoritma ini menggunakan *microcontrollers* yang bekerja dengan *clock speed* 33 MHz , efisiensinya melebihi 20 Mbit/s.

Dengan implementasi software dari algoritma COBRA-F64a dan COBRA-F64b (untuk processor tipe Celeron 500 MHz, diasumsikan bahwa perintah DDP32 diimplementasikan di dalamnya sebagai instruksi khusus), perkiraan kecepatan enkripsi adalah 400 Mbit/s untuk algoritma a dan 300 Mbit/s untuk algoritma b.

Hal tersebut membuktikan bahwa perangkat lunak kriptografi yang menggunakan permutasi terkontrol yang dieksekusi berdasarkan data, jika dibandingkan dengan algoritma-algoritma kriptografi yang telah diketahui secara luas, berpotensi meningkatkan kecepatan beberapa kali.

Mengestimasi keamanan *cipher* terhadap berbagai metode *cryptanalysis* yang diketahui adalah kegiatan yang kompleks dan mahal. Karenanya, investigasi dilakukan untuk menemukan metode *cryptanalysis* yang paling efisien, metode *differential cryptanalysis* salah satunya.

Pada metode *differential cryptanalysis* perbedaan dari pasangan blok data X and X′ yang ditransformasikan, - sebutlah Δ(X, X′) - dipelajari, juga perbedaan blok δ(Y, Y′) yang berkorespondensi. Biasanya, sebuah case dipilih ketika Δ(X, X′) = (X □ X′) = Δ = konstan. Nilai Δ disebut input *difference*. Bit-bit *non-zero* dari Δ disebut *active bits*. Aktivitas dari *differential cryptanalysis* adalah untuk menemukan nilai Δ dan δ yang memenuhi p(δ, Δ) > 2−n , dimana n adalah panjang dari blok input dalam rangkaian bit, dan p(δ, Δ) = Pr (δ, Δ) adalah probabilitas kondisional. Biasanya, "*significant probabilities*" p(δ, Δ) – misal yang diturunkan dari nilai rata-rata probabilitas- adalah selisih dengan bilangan-bilangan kecil dari bit-bit aktif pada *differential cryptanalysis* yang dimaksud.

Hasil dari investigasi keamanan kriptografis dari cipher COBRA-F64a dan COBRA-F64b ditambahkan kepada metode *differential cryptanalysis* yang memiliki probabilitas paling signifikan yang berkoresponden dengan karakteristik differential yang hanya memiliki satu "bit aktif". Ingat bahwa istilah "bit aktif" digunakan dengan asumsi bahwa bobot Hamming dari selisih dua vector adalah satu. Jadi, untuk algoritma COBRA-F64a, karakteristik terbaik dari yang ditemukan adalah karakteristik *three-round*, dimana sebuah output satu bit berselisih sebesar δ1 dengan probabilitas pa(3) = 2−21 berkorespondensi dengan sebuah input satu bit berselisih Δ1. Karena itu, untuk algoritma COBRA-F64b, karakteiristik terbaik yang ditemukan adalah karakteristik *two-round*, dimana, setelah dua putaran transformasi, output satu bit berselisih δ1 dengan sebuah probabilitas pb(2) = 2−12 yang berkorespondensi dengan input berselisih Δ1 dengan satu bit aktif di sebelah kanan (atau kiri)blok data. Bagaimanapun, aplikasi iteratif bagi karakteristik ini untuk 15 putaran pada kasus pertama dan 20 pada kasus kedua, tidak mengijinkan kita untuk membuat karakteristik 15 dan 20 putaran yang efisien, karena nilai probabilitas pa(15) = 2−105 dan pb(20) = 2−120 tidak melebihi nilai probabilitas yang berkorespondensi pada blok *cipher* 64-bit yang berubah-ubah. Pembangunan karakteristik *diferensial* dengan bilangan yang besar dari bit aktif tidak memberikan hasil yang diinginkan terhadap karakteristik yang efisien, yang membuktikan tingkat keamanan cipher ini terhadap *differential cryptanalysis*.

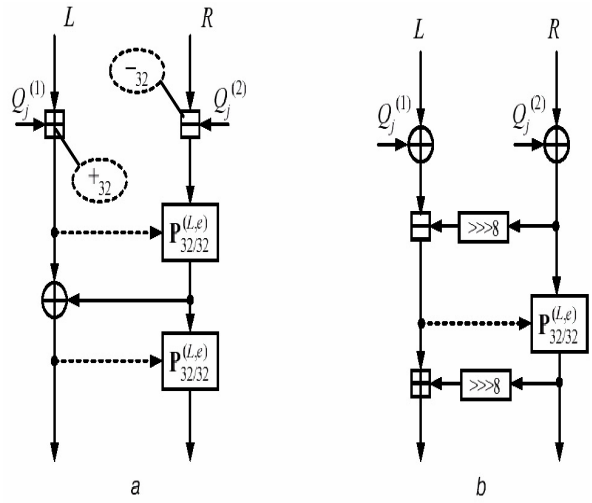
Indeks yang tinggi dari kecepatan dan keamanan *cipher*, dimana *command* DDP32 digunakan, memungkinkan kita berharap bahwa pengembang *microprocessor* akan mengikut sertakan *command* ini ke dalam *standard operations of multipurpose processors*.

4. Mengestimasi Keamanan dari Cipher COBRA-F64a dan COBRA-F64b

Seperti pada semua *cipher* yang telah dikenal sebelumnya, COBRA-F64a dan COBRA-F64b, bit-bit aktif memberikan kontribusi terbesar dalam efek yang muncul pada control input dari box permutasi terkontrol. Hal ini membawa fenomena karakteristik differential dengan bilangan kecil memiliki probabilitas yang lebih besar. Adalah sangat penting untuk menemukan karakteristik dimana *intermediary differences* juga memiliki jumlah bilangan dari bit aktif pada skema formasinya. Sesungguhnya,

keberadaan satu bit aktif pada input dari box permutasi terkontrol memberikan *multiplier* 2−3 pada ekspresi probabilitas, dimana ketersediaan bit aktif pada input dari box permutasi memberikan *multiplier* 2−5 dengan event yang berkaitan dengan transisi digit yang telah diberikan sebelumnya (Perlu juga untuk menjadikan pasangan bit aktif menjadi nol, jika tidak, maka bilangan bit aktif akan terus bertambah). Anda bisa membangun algoritma COBRA-F64a dan COBRA-F64b *intermediary differences* hanya memiliki bit aktif di sebelah kiri dan kanan subblock. Karakteristik seperti di atas memiliki *probability* maksimal.

Skema pembentukan untuk karasteristik yang berbeda dari algoritma COBRA-F64a dan COBRA-F64b diperlihatkan pada Gambar 3. Pada kedua algoritma, karakteristik yang paling efisien berhubungan dengan *difference* (ΔL0, ΔR1) yang melewati dua atau tiga putaran. Pada satu putaran algoritma COBRA-F64a, bit aktif dari *difference* dipindahkan sekali dari cabang kanan ke cabang kiri ketika operasi XOR dilakukan. Karena itu, untuk kembali ke *initial difference* pada skema kriptografi COBRA-F64a, harus dijalankan tiga putaran, dimana putaran kedua berkaitan dengan peristiwa subblok kanan dan kiri dari bit aktif memasuki digit yang sama.



Gambar 3 Skema pembentukan dengan karakteristik 3 putaran pada algoritma COBRA-F64a (a) dan karakteristik 2 putaran pada algoritma COBRA-F64b (b)

Fitur yang membedakan algoritma COBRA-F64b dari COBRA-F64a adalah pada satu putaran, aktif bit dari *difference* ΔR1 dibawa dari cabang kanan ke cabang kiri sebanyak dua kali, yang menyediakan penyimpanan *difference* (ΔL0, ΔR1) pada keluaran dari putaran pertama. Permutasi dilakukan setelah putaran pertama, menjalankan transformasi *difference* (ΔL0, ΔR1) menjadi *difference* (ΔL1, ΔR0), dan juga putaran kedua tercakup dalam skema pembentukan dengan karakteristik yang berbeda-beda, dimana dengan probabilitas 2−3, difference ΔL1 melewati bagian kiri tanpa membangkitkan bit aktif tambahan pada subblok kanan dan kiri. Permutasi subblok setelah putaran kedua mengakibatkan pembentukan *initial difference* (ΔL0, ΔR1).

Dengan mempertimbangkan probabilitas peristiwa yang berkaitan dengan formasi COBRA-F64a dengan karakteristik tiga putaran dan masukan *difference* (ΔL0, ΔR1), dapat disimpulkan dengan mudah bahwa probabilitas dari pembentukan difference (ΔL1, ΔR1) pada keluaran putaran kedua adalah sebagai berikut :

$$P' = P\{(\Delta_0^L, \Delta_1^R) \to (\Delta_1^L, \Delta_1^R)\} = 2^{-4}. \quad (4.1)$$

Probabilitas ini ditentukan oleh peristiwa bit aktif melewati operasi pengurangan pada cabang kanan, pembangkitan sebuah bit aktif pada subblok kiri setelah operasi XOR, dan performansi dari sebuah operasi permutasi terkontrol dengan bit aktif tersedia pada masukan kontrol. *Difference* (ΔL1, ΔR1) setelah putaran kedua (dan operasi permutasi) dibawa ke *difference* (ΔL1, ΔR0), dengan probabilitas

$$P' = P\{(\Delta_1^L, \Delta_1^R) \to (\Delta_1^L, \Delta_0^R)\} = 2^{-1} \cdot 2^{-1} \cdot 2^{-3} \cdot 2^{-5} = 2^{-10}. \quad (4.2)$$

Setelah putaran ketiga dijalankan, termasuk penjumlahan modulo 232 dan dua operasi permutasi terkontrol, *difference* (ΔL1, ΔR0) ditransformasi menjadi *difference* (ΔL0, ΔR1), dengan probabilitas

$$P'' = P\{(\Delta_1^L, \Delta_0^R) \to (\Delta_0^L, \Delta_1^R)\} = 2^{-1} \cdot 2^{-3} \cdot 2^{-3} = 2^{-7}. \quad (4.3)$$

Probabilitas untuk karakteristik tiga putaran adalah P(3) =2−21. Dengan menggunakan nilai ini, dimungkinkan untuk mewujudkan, dengan

jumlah putaran r ≥ 10, algoritma COBRA-F64a dapat dibedakan dari transformasi acak dengan kriptanalisis berbeda yang menggunakan karakteristik di atas. Jadi, algoritma COBRA-F64a dengan 16 putaran menyediakan batas keamanan yang cukup terhadap analisis yang berbeda-beda. Perlu diingat bahwa karakteristik tiga putaran untuk sistem kriptografi ini juga dapat menggunakan (ΔL1, ΔR1) dan (ΔL1, ΔR0) sebagai *input differences*. Variasi-variasi dari karakteristik tiga putaran memiliki probabilitas yang sama dengan karakteristik yang dijelaskan di atas. Tetapi penggunaan dari ketiga variasi tersebut tidak sepenuhnya sama. Sebagai contoh, dengan jumlah putaran yang tidak habis dibagi tiga, *difference* melewati putaran terakhir atau putaran terakhir dan selanjutnya berhubungan dengan peristiwa *difference* melewati sebuah skema yang tidak komplit dari karakteristik pembentukan yang berbeda-beda. Bergantung pada variasi *input difference*, dari yang belakangan ditiadakan satu atau dua putaran terakhir, hal ini menghasilkan nilai probabilitas yang berbeda-beda. Jika satu putaran ditiadakan, akan menghasilkan probabilitas 2−7, 2−10, atau 2−4. Jia dua putaran ditiadakan maka pasangan-pasangan probabilitas p1 = 2−7·2−10, p2 = 2−10·2−4 atau p3 = 2−4·2−7 dapat ditiadakan.

Contoh-contoh di atas menunjukkan bahwa untuk variasi yang berbeda, kemungkinan nilai perbandingan dari nilai probabilitas yang dihasilkan untuk seluruh algoritma adalah 26. Keadaan ini tidak berarti apa-apa jika jumlah putaran yang dipilih adalah jumlah yang minimal dengan tujuan untuk mempercepat prosedur enkripsi.

Dari skema pembentukan COBRA-F64b dengan karakteristik dua putaran dimana *input difference* (ΔL0, ΔR1) digunakan, dapat dilihat dengan mudah bahwa probabilitas dari pembentukan *difference* (ΔL1, ΔR0) pada masukan putaran kedua adalah :

$$P' = P\{(\Delta_0^L, \Delta_1^R) \rightarrow (\Delta_1^L, \Delta_0^R)\} = 2^{-7}. \quad (4.4)$$

Probabilitas di atas ditentukan oleh peristiwa bit aktif melewati operasi pengurangan pada putaran pertama dan pembangkitan sebuah bit aktif pada cabang kiri (*co-factor* ≈ 2−1), pertemuan bit aktif pada subblok kanan dan kiri (*co-factor* 2−5), dan peristiwa bit aktif melewati operasi penambahan pada cabang kiri yang menjadikan 0 aktif bit pada subblok kiri (co-factor 2−1).

Pada saat *difference* (ΔL1, ΔR0) melewati putaran kedua (termasuk subblok data yang ditukar), *difference* tersebut mengalami transformasi menjadi *difference* (ΔL0, ΔR1), dengan probabilitas P″ = P{(ΔL1, ΔR0) → (ΔL0, ΔR1)} = 2-5. Probabilitas dari karakteristik dua putaran adalah sebagai berikut :

$$P(2) = P'P'' = 2^{-12}. \quad (4.5)$$

Variasi lain dari karakteristik dua putaran berhubungan dengan peristiwa *difference* (ΔL1, ΔR0) melewati kedua putaran tersebut. Pada kenyataannya, kedua variasi tersebut menjalankan mekanisme yang sama di antara putaran yang berbeda. Beberapa ketidak sesuaian dalam pemilihan input menjadi persoalan untuk kasus encripsi dengan jumlah putaran ganjil. Rasio nilai probabilitas yang dihasilkan dari putaran berjumlah enkripsi ganjil, bergantung pada pemilihan salah satu dari dua varian *difference*, yaitu 22. Ungkapan yang serupa diberikan kepada cipher SPECTR-H64 dan SPECTR-128.

Untuk beberapa *cipher*, probabilitas menemukan *difference* yang diinginkan pada *output* dari algoritma enkripsi dapat ditingkatkan dengan memilih *difference input* yang berbeda dengan yang berkorespondensi ke karakteristik paling efisien. Meningkatkan probabilitas dilakukan dengan menekankan spesifikasi dari *difference* yang dibutuhkan dengan bit aktif dalam digit yang telah dispesifikasikan saat putaran pertama dijalankan. Sebagai contoh, hal ini dapat diimplementasikan pada kasus *cipher* COBRA-F64b, dimana *difference input* (ΔL1|32, ΔR1|32) pada output putaran pertama ditransformasikan ke dalam *difference* (ΔL1, ΔR0), dengan probabilitas sama dengan 1. Putaran berikutnya sesuai dengan karakteristik *two-round*.

Cukup mudah untuk menurunkan rumus berikut untuk probabilitas kemunculan *difference* (ΔL1, ΔR0) pada *output* dari algoritma:

P (r) = P{(ΔL1|32, ΔR1|32) → (ΔL1, ΔR0)} = 2-6r+5 untuk r genap
P (r) = P{(ΔL1|32, ΔR1|32) → (ΔL1, ΔR0)} = 2-6(r-1) untuk r ganjil

Dari kondisi P(r) ≥ 2−64, bilangan minimal yang direkomendasikan untuk algoritma COBRA-F64b adalah: rmin = 12. If r ≥ rmin, transformasi yang dispesifikasikan dengan algoritma ini tidak dapat dibedakan dengan metode random.

Di bawah ini adalah tabel ringkasan dari properti *differential* Tabel 2.

Tabel 2 Perbandingan data bagi *Differential Characteristics* pada *Cipher-cipher* Blok berdasarkan Operasi Terkontrol

Cipher	r	Characteristic		P(r)[**]
		Difference	Probability	
SPECTR-H64	12	(ΔL0, ΔR1)	P(2) = 1.5.2−15	2−87
SPECTR-H64	12	(ΔL0, ΔR1)	P(3) = 2−28	2−112
SPECTR-128	12	(ΔL0, ΔR1)	P(2) = 1.1.2−19	2−113
SPECTR-128[*]	12	(ΔL0, ΔR1)	P(3) = 2−32	2−128
SPECTR-128[*]	12	(ΔL0, ΔR1)	P(2) = 2−27	2−162
COBRA-F64a	16	(ΔL1, ΔR1)	p(3) = 2−21	< 2−105
COBRA-F64b	20	(ΔL0, ΔR1)	P(2) = 2−12	2−120
DDP-S64	10	(ΔA1, ΔB0)	P(2) = 2−23	2−115
DDP-S128	12	(ΔA1, ΔB0, ΔC0, ΔD1)	P(2) = 2−32	2−192

[**]Kontribusi dari karakteristik probabilitas dari difference passing dengan r putaran.
[*]Varian yang dimodifikasi.

Differential cryptanalysis menjalankan sebagian dari investigasi yang kompleks yang didapat selama merancang *cipher* dengan tujuan mengoptimalisasi beberapa primitif, sebagai inti dari keamanan kriptografi.

Secara khusus, telah didemonstrasikan bagaimana tabel distribusi E *control bits critical*. Hal ini dikompilasi dalam beberapa langkah:

1. Pertama, kriteria umum dari kompilasi tabel dirumuskan
2. Karakteristik *differential* dihitung, dan sebagai hasil, jumlah bit yang paling berkontribusi ditentukan.
3. Tabel dimodifikasi dengan mengubah posisi dari bit-bit yang muncul
4. *Differential cryptanalysis* diulangi.
5. Eksperiment dijalankan untuk menentukan probabilitas dari karakteristik *difference* yang berkorespondensi dengan satu atau lebih putaran enkripsi
6. Hasil teoritis dibandingkan dengan data eksperimen.
7. Jika teori dan eksperimen sesuai, maka kesimpulan ditarik bahwa mekanisme utama dari pembentukan difference diperhitungkan dalam model teritis, dan hasil estimasi dari *cryptanalysis* dapat dipercaya.

Jadi, pada analisis *differential* yang lengkap dari *cipher*, tiga hal di atas harus dijalankan. Ada kemungkinan bahwa nilai probabilitas berdasarkan eksperimen akan melebihi nilai teorits. Ini memiliki arti bahwa beberapa mekanisme yang memberikan kontribusi tidak diperhitungkan pada model pembentukan karakteristik.

5. Kesimpulan

Masih terbuka lapangan pencarian dan penelitian untuk menjawab kebutuhan akan algoritma yang aman secara kriptogafis. Ide-ide atas algoritma yang aman ini juga diikuti oleh ide metode kriptanalisis yang lebih cemerlang pula. Karenanya, bidang kriptografi adalah bidang yang masih terus dikembangkan.

[1] Goots, Nick, et al. *Modern Cryptography-Protect Your Data with Fast Block Ciphers*. A-LIST, LLC. 2003