# *SIMPLIFIED* KRIPTANALISIS LANJAR DAN KRIPTANALISIS DIFERENSIAL SERTA IMPELEMENTASINYA PADA SIMPI: CIPHER BLOK SEDERHANA BERBASISKAN SUBSTITUSI-PERMUTASI

Chandra Gondowasito – NIM : 13504100

*Program Studi Informatika*
*Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung*
*Jl. Ganesha 10, Bandung*
E-mail : if114100@students.if.itb.ac.id

**Abstrak**

Kriptanalisis lanjar dan kriptanalisis diferensial adalah dua teknik serangan yang *powerfull* terhadap cipher blok kunci simetri. Namun konsep dari kedua serangan ini biasanya diasosiasikan dalam konteks suatu cipher tertentu sehingga menjadi rumit dan sulit dimengerti. Dalam makalah ini, dijelaskan konsep kedua serangan secara lebih sederhana sehingga mudah dimengerti dan diikuti dengan implementasi praktisnya pada cipher SIMPI.. SIMPI didesain sebagai sebuah cipher blok yang simpel namun memiliki struktur yang realistis seperti kebanyakan cipher modern dan berbasiskan pada operasi substitusi-permutasi. Data percobaan kriptanalisis lanjar dan diferensial terhadap SIMPI disajikan sebagai konfirmasi berlakunya konsep yang dijelaskan..

***Kata kunci**: Kriptanalisis Lanjar,Kriptanalisis Diferensial, SIMPI, S-box, ekspresi lanjar, persamaan lanjar, pendekatan, karakteristik diferensial, perbedaan, bias, upa kunci, bit, enkripsi, dekripsi.*

## 1. Pendahuluan

Dalam makalah ini disajikan dua teknik kriptanalisis yang *powerful*: kriptanalisis lanjar [1] dan kriptanalisis diferensial [2] dan pengaplikasiannya pada blok cipher kunci simetri. Kriptanalisis lanjar diperkenalkan oleh Matsui pada EUROCRYPT '93 sebagai serangan teoritis pada Data Encryption Standard (DES) [3] yang kemudian sukses dipakai dalam kriptanalisis praktis terhadap DES [4]; kriptanalisis diferensial perrtama kali diperkenalkan oleh Biham dan Shamir pada CRYPTO '90 untuk menyerang DES dimana detail dari serangan tersebut dipublikasikan dalam sebuah buku [5]. Meskipun target awal dari kedua serangan tersebut adalah DES, kekepraktisan penggunaannya terhadap berbagai cipher blok lainnya telah membuat kedua teknik kriptanalisis ini sebagai pedoman pertimbangan akan keamanan semua blok cipher. Sebagai contoh, banyak kandidat yang diusulkan sebagai Advanced Encryption Standard didesain untuk kebal terhadap kedua teknik serangan ini. Hal ini sagat jelas pada cipher Rijndael[7], algoritma enkripsi yang terpilih sebagai AES.Konsep yang disajikan dalam makalah ini bisa digunakan untuk membentuk pemahaman awal yang diperlukan untuk memahami prinsip desain dan analisis keamanan dari cipher Rijndael dan cipher-cipher lainnya yang banyak diusulkan.

Makalah ini menyajikan konsep dasar dari kriptanalisis lanjar dan diferensial namun bukan merupakan sumber untuk memahami berbagai penambahan dan penigkatan kualitas pada kedua teknik serangan ini. Dalam makalah ini didesain sebuah cipher blok SIMPI yang strukturnya simpel namun cukup realistis untuk mempelajari implementasi dari konsep-konsep kedua serangan krtiptanalisis lanjar dan diferensial.

Dalam makalah ini kedua teknik yang dibahas telah di-*simplified* sehingga berlaku secara umum. Pustaka kriptografi yang konvensional [11][12][13][14] biasanya menyajikan amteri mengenai cipher blok secara sangat deskriptif dengan kurang detailnya dalam mewngilustrasikan konsep kedua serangan. Kebanyakan materi mengenai serangan ini yang dipublikasikan memiliki fokus pada riset dan memberikan intuisis yan gkecil dan penjelasan bagi non-ahli. Ketika konsep dasar dijelaskan pada literatur (seperti pada makalah asli Matsui dan Biham-Shamir), konsep ini disajikan sebagai serangan terhadap DES, sehingga membuat konsep yang dimilikinya kurang terpahami.

## 2. SIMPI: Cipher Sederhana Berbasiskan Substitusi dan Permutasi

Dalam menghadirkan konsep kriptanalisis ini didesainlah seuah cipher blok kunci simetri, SIMPI. SIMPI adalah sebuah cipher sederhana

berbasiskan substitusi dan permutasi. Cipher SIMPI, seperti yang dapat dilihat pada gambar 1 menerima masukan blok 16 bit dan mengenkripsi blok dengan mengulang operasi-operasi dasar sebanyak 4 kali. Tiap putaran terdiri atas operasi:

- Substitusi,
- Transposisi bit-bit (permutasi posisi bit), dan
- *Key Mixing* (XOR dengan upa kunci).

Struktur dasar in pertama kali diperkenalkan oleh Feistel pada 1973[5] dan operasi-operasi dasar ini mirip dengan yang terdapat pada DES dan cipher modern lainnya termasuk Rijndael. Jadi, meskipun SIMPI kelihatnnya sederhana, analisis serangan terhadap cipher ini bisa memberikan gambaran akan keamanan struktur cipher serupa yang lebih rumit.

### 2.1 Substitusi

Dalam SIMPI, masukan blok 16 bit dibagi menjadi empat upa blok berukuran empat bit. Tiap upa blok menjadi masukan untuk sebuah S-box 4x4 (substitusi dengan masukan 4 bit dan keluaran 4 bit), yang diimplementasikan sebagai sebuah *look-up tabel* berukuran 16 dengan entri empat bit. Karakteristik paling penting dari S-box adalah pemetaan nirlanjar, yaitu bit keluaran tidak bisa dimodelkan sebagai operasi lanjar dari bit masukan.

SIMPI hanya menggunakan satu tipe S-box untuk dipakai ulang dalam tiap putaran dan digunakan bersama-sama dalam satu putaran (pada DES, semua S-box dalam satu putaran berbeda, dan tiap putaran menggunakan *set* S-box yang sama). Serangan kriptanalisis lanjar dan diferensial bekerja tak peduli apakah hanya ada satu tipe S-box ataukah semua S-box berbeda. Pemetaan yang digunakan dalam SIMPI(tabel 1) diambil dari S-box DES(baris pertama S-box pertama). Dalam tabel 1, bit-bit direpresentasikan dalam heksadesimal dan Most Significant Bit (MSB) adalah bit masukan terkiri pada gambar 1.

| *masukan* | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *keluaran* | E | 4 | D | 1 | 2 | F | B | 8 | 3 | A | 6 | C | 5 | 9 | 0 | 7 |

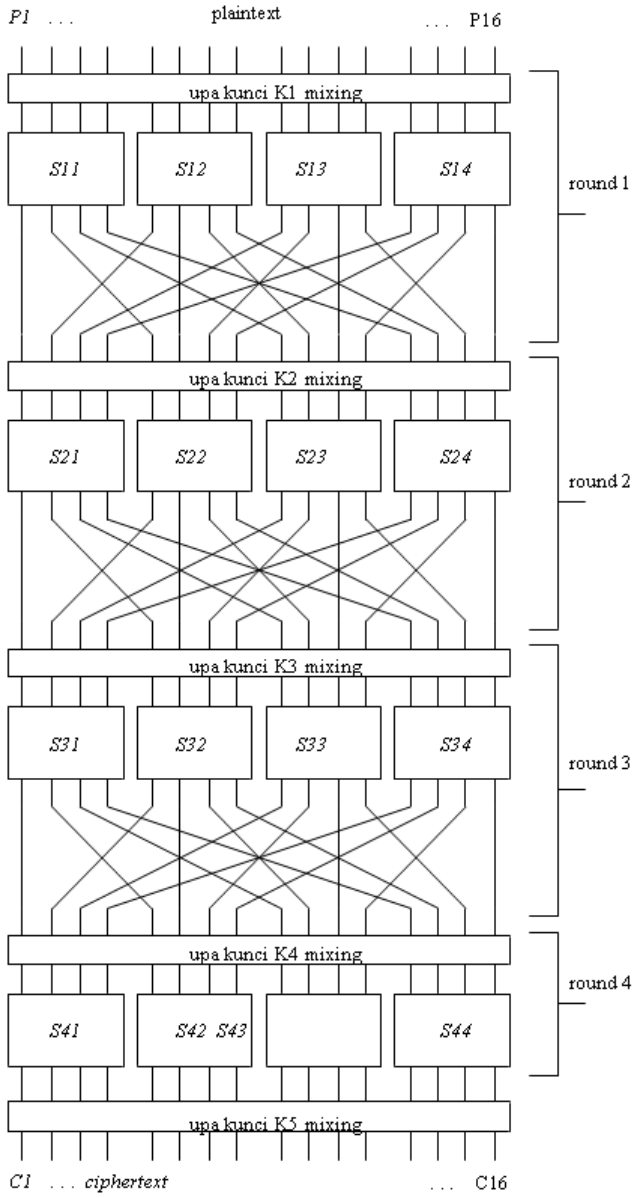
Tabel 1. Pemetaan S-box(dalam heksadesimal)

### 2.2 Permutasi

Permutasi tiap putaran SIMPI adalah transposisi bit-bit atau permutasi posisi bit. Permutasi pada gambar 1 diberikan pada tabel 2 (dengan nomor merepresentasikan posisi bit dalam blok, dengan 1adalah bit terkiri dan 16 adalah bit terkanan). Perhatikan bahwa tidak perlu dilakukan permutasi pada putaran terakhir karena tidak berpengaruh pada keamanan cipher sehingga SIMPI tidak melakukannya.

| *masukan* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *keluaran* | 1 | 5 | 9 | 13 | 2 | 6 | 10 | 14 | 3 | 7 | 11 | 15 | 4 | 8 | 12 | 16 |

Tabel 2. Permutasi pada SIMPI




Gambar 1. SIMPI: Cipher Substitusi-Permutasi

### 2.3 Key Mixing

Untuk mendapatkan efek *key mixing*, SIMPI menggunakan operasi XOR antara bit-bit upa kunci tiap putaran dengan bit-bit blok masukan pada putaran. Operasi ini juga dilakukan setelah putaran terakhir untuk memastikan bahwa operasi substitusi terakhir tidak dengan mudahnya diabaikan oleh kriptanalis dengan hanya melakukan operasi balikan dari substitusi terakhir pada cipherteks. Pada cipher umum, upa kunci tiap putaran diperoleh dari kunci utama cipher melalui proses yang dikenal sebagai *key schedulling*. Namun pada SIMPI,

tiap upa kunci adalah tidak saling berkaitan dan saling bebas.

### 2.4 Dekripsi

Untuk melakukan dekripsi SIMPI, blok data dilewatkan secara terbalik pada cipher. Jadi, dekripsi SIMPI menggunakan bentuk yang sama seperti pada gambar 1 tetapi pemetaan pada S-box adalah inversi dari pemetaan dalam enkripsi (masukan menjadi keluaran, keluaran menjadi masukan). Untuk mendekripsi SIMPI, haruslah setiap S-box besifat bijeksi (pemetaan satu ke satu, jumlah bit masukan dan bit keluaran sama). Agar dekripsi bisa dilakukan, upa kunci digunakan dengan urutan terbalik dan bit-bit upa kunci harus ditransposisi sesuai permutasi pada enkripsi agar dekripsi tetap seperti gambar 1.

Dengan tidak adanya permutasi pada putaran terakhir membuat dekripsi SIMPI memiliki struktur yang sama dengan enkripsi (bila ada permutasi setelah putaran perakhir maka proses dekripsi harus diawali permutasi sebelum memasuki putaran pertama).

## 3. Kriptanalisis Lanjar

Di bagian ini, disajikan konsep *simplified* kriptanalisis lanjar kemudian penerapannya terhadap SIMPI.

### 3.1 Gambaran Serangan Dasar

Kriptanalisis lanjar mencoba mengambil manfaat dari besarnya kemungkinan berlakunya ekspresi lanjar yang melibatkan bit-bit plainteks, cipherteks (meski sebenarnya yang digunakan adalah bit-bit keluaran dari putaran sebelum putaran terakhir), dan bit-bit upa kunci. Serangan ini termasuk dalam kategori *known-plaintext attack*, yaitu penyerang dianggap memiliki sejumlah plainteks dan cipherteks yang bersesuaian namun penyerang tidak mengetahui bagaimana memilih plainteks(dan cipherteks yang bersesuaian) dari yang tersedia. Dalam penerapannya dan dalam skenario,diasumsikan penyerang memiliki akses ke plainteks yang acak beserta cipherteks yang bersesuaian.

Ide dasar dari serangan ini adalah untuk mendekati(mengaproksimasi) operasi pada bagian cipher dengan ekspresi lanjar dimana kelanjaran yang dimaksud ini mengacu pada operasi terhadap bit modulus dua (misalnya XOR yang dilambangkan sebagai "⊕"). Ekspresi lanjar yang dimaksud adalah dalam bentuk:

$$X_{i1} \oplus X_{i2} \oplus X_{i3} \ldots \oplus X_{iu} \oplus Y_{j1} \oplus Y_{j2} \oplus \ldots \oplus Y_{jv} = 0 \quad (1)$$

dengan $X_i$ merepresentasikan bit ke-i dari masukan $X = [X_1, X_2, \ldots]$dan $Y_j$ merepresentasikan bit ke-j dari keluaran $Y = [Y_1, Y_2, \ldots]$. Persamaan ini merepresentasikan penjumlahan XOR dari u bit masukan dan v bit keluaran.

Pendekatan yang digunakan dalam kriptanalisis lanjar adalah untuk menentukan ekspresi dalam bentuk diatas yang kemungkinan kemunculannya besar atau kecil (tidak ada kelanjaran yang jelas seperti di atas seharusnya berlaku untuk semua nilai masukan dan keluaran atau cipher berarti sangat lemah). Jika cipher memiliki kecenderungan memenuhi persamaan (1) dengan kemungkinan yang besar atau tidak memenuhinya dengan kemungkinan yang besar, ini membuktikan bahwa kemampuan pengacakan cipher lemah. Misalkan kita secara acak memilih nilai untuk u+v bit dan memasukkannya dalam persamaan diatas; kemungkinan bahwa persamaan tersebut berlaku adalah tepat ½. Adalah penyimpangan/ bias dari kemungkinan ½ inilah yang dimanfaatkan dalam kriptanalisis lanjar: semakin jauh penyimpangan kemungkinan berlakunya ekspresi lanjar (1) dari ½, serangan kriptnalaisis lanjar semakin menjanjikan. Untuk selanjutya, kita akan menyebut besarnya penyimpangan dari ½ pada kemungkinan berlakunya persamaan lanjar diatas sebagai bias dari kemungkinan lanjar. Jadi bila persamaan diatas berlaku dengan kemungkinan $p_L$ untuk plainteks dan cipherteksa yang bersesuaian yang dipilih secara acak, biasnya adalah $p_L$-1/2. Semakin besar tingkatan nilai kemungkinan bias, yaitu. $|p_L-1/2|$, semakin bagus penggunaan kriptanalisis lanjar karena jumlah *known* plainteks yang diperlukan dalam penyusunan serangan lebih sedikit.

Terdapat beberapa cara untuk menyusun serangan kriptanalisis lanjar. Makalah ini berfokus pada apa yang Matsui sebut sebagai algoritma 2 [1]. Kita akan melakukan penyusunan pendekatan lanjar yang melibatkan bit-bit plainteks yang direpresentasikan sebagai X pada (1) dan bit-bit masukan untuk putaran terakhir cipher (dengan kata lain adalah keluaran dari putaran sebelum putaran terakhir cipher) sebagai Y pada (1). Bit-bit plainteks adalah acak sehingga masukan untuk putaran terakhir adalah juga acak. Persamaan (1) bisa diekspresikan dalam bentuk lain dengan bagian kanan dari kesamaan adalah penjumlahan(XOR) sejumlah dari bit upa kunci. Pada persamaan (1) dimana sebelah kanan dari kesamaan adalah "0", sebenarnya telah tersirat keterlibatan bit-bit upa kunci: bit-bit upa kunci adalah tertentu namun tidak diketahui (akan dicari dalam serangan lanjar) dan secara tersirat

telah kita nyatakans sebagai "0" pada sebelah kanan kesamaan dan sebagai kemungkinan ($p_L$) dipenuhinya ekpsresi lanjar tersebut. Jika penjumlahan dari bit-bit upa kunci yang terlibat adalah "0", bias dari (1) memiliki tanda yang sama(+ atau -) dengan bias dari ekspresi yang melibatkan penjumlahan bit-bit upa kunci dan jika penjumlahan dari bit-bit upa kunci yang terlibat adalah "1", bias dari (1) akan memiliki tanda yang berkebalikan dengan bias bila ekspresi (1) melibatkan penjumlahan bit-bit upa kunci

Perhatikan bahwa $p_L=1$ mengindikasikan ekspresi lanjar (1) sebagai ekspresi lanjar yang sempurna bagi pendekatan perilaku cipher yang berarti cipher memiliki kelemahan yang fatal. Jika $p_L=0$, (1) memberikan hubungan *affine* untuk cipher, yang juga mengindikasikan kelemahan cipher yang sangat berbahaya. Dalam sistem penjumlahan modulo 2, sebuah fungsi *affine* adalah komplemen dari fungsi lanjar. Pendekatan lanjar yang diindikasikan dengan $p_L>1/2$ dan pendekatan *affine* yang diindikasikan dengan $p_L<1/2$ adalah sama-sama sasaran empuk bagi kriptanalisis lanjar sehingga akan digunakan istilah lanjar untuk menyatakan baik hubungan lanjar maupun *affine*.

Pertanyaan yang tentu muncul adalah bagaimana menyusun ekspresi yang sangat lanjar sehingga bisa dieksploitasi. Hal ini dilakukan dengan memperhatikan karakteristik dari satu-satunya komponen cipher substitusi-permutasi yang nirlanjar yaitu S-box. Bila karakteristik nirlanjar dari S-box dicatat, menjadi mungkin untuk menyusun pendekatan lanjar antara sejumlah bit-bit masukan dan keluaran dari S-box. Selanjutnya dimungkinkan penggaabungan trehadap pendekatan lanjar dari tiap S-box besama-sama sehingga bit-bit antara (bit-bit data yang dihasilkan dalam proses cipher) bisa lagi berpengaruh dan hany tersisa suatu ekspresi lanjar dengan bias yang tinggi yang hanya melibatkan plainteks dan bit-bit masukan untuk putaran terakhir.

### 3.2 Prinsip Pilling-Up

Sebelum mencoba menyusun ekspresi lanjar terhadap SIMPI, diperlukan pengetahuan sebagai sarana pembantu dalam menyusun serangan. Misalkan terdapat dua bit acak X1 dan X2. Perhatikan hubungan sederhana berikut: X1⊕X2=0 adalah ekspresi lanjar dan sama dengan X1=X2; X1⊕X2=1 adalah ekspresi *affine* dan adalah sama dengan X1≠X2. Selanjutnya misalkan peluang distribusi persamaan lanjar adalah:

$$\Pr(X_1=i) = \begin{cases} p_1 & ,i=0 \\ 1-p_1 & ,i=1 \end{cases}$$

dan

$$\Pr(X_2=i) = \begin{cases} p_2 & ,i=0 \\ 1-p_2 & ,i=1 \end{cases}$$

Jika kedua variabel yang acak adalah saling bebas, maka

$$\Pr(X_1=i,X_2=j) = \begin{cases} p_1p_2 & ,i=0,\ j=0 \\ p_1(1-p_2) & ,i=0,\ j=1 \\ (1-p_1)p2 & ,i=1,\ j=0 \\ (1-p_1)(1-p_2) & ,i=1,\ j=1 \end{cases}$$

sehingga:

$$\begin{aligned} \Pr(X_1\oplus X_2=0) &= \Pr(X_1=X_2) \\ &= \Pr(X_1=0, X_2=0)+\Pr(X_1=1, X_2=1) \\ &= p_1p_2+(1-p_1)(1-p_2) \end{aligned}$$

dan kita misalkan bahwa $p_1=1/2+\epsilon_1$ dan $p_2=1/2+\epsilon_2$, dengan $\epsilon_1$ dan $\epsilon_2$ menyatakan kemungkinan bias dan $-1/2\le\epsilon_1,\epsilon_2\le+1/2$. Jadi

$$\Pr(X_1\oplus X_2=0)=1/2+2\epsilon_1\epsilon_2$$

dan bias $\epsilon_{1,2}$ dari $X_1\oplus X_2=0$ adalah

$$\epsilon_{1,2}=2\epsilon_1\epsilon_2$$

Hal ini bisa diperluas menjadi lebih dari dua variabel biner yang acak, $X_1$ sampai $X_n$, dengan kemungkinan $p_1=1/2+\epsilon_1$ sampai $p_n=1/2+\epsilon_n$. Kemungkinan bahwa $X_1\oplus \ldots \oplus X_n=0$ berlaku dapat ditentukan dengan memanfaatkan *Lemma Pilling-Up* yang mengasumsikan bahwa kesemua *n* variabel biner acak tersebut adalah saling bebas.

***Lemma Pilling-Up* (Matsui [1])**

Untuk n varibel biner acak yang saling bebas $X_1,X_2,\ldots,X_n$,

$$\Pr(X_1\oplus \ldots \oplus X_n=0)=1/2+2^{n-1}\prod_{i=1}^{n}\epsilon_i$$

sehingga

$$\epsilon_{1,2,\ldots,n}=2^{n-1}\prod_{i=1}^{n}\epsilon_i$$

dimana $\epsilon_{1,2,\ldots,n}$ menyatakan bias dari $X_1\oplus \ldots \oplus X_n=0$.

Perhatikan bahwa bila $p_i=0$ atau 1 untuk semua i, maka $\Pr(X_1\oplus \ldots \oplus X_n=0)=0$ atau 1. Jika hanya satu $p_i=1/2$, maka $\Pr(X_1\oplus \ldots \oplus X_n=0)=1/2$.

Dalam menyusun pendekatan lanjar terhadap suatu cipher, nilai dari $X_i$ sebenarnya akan merepresentasikan pendekatan lanjar dari S-box. Sebagai contoh, misalkan terdapat empat variabel biner acak yang saling bebas $X_1,X_2,X_3,X_4$. Misalkan pula $\Pr(X_1+X_2=0)=1/2+\epsilon_{1,2}$ dan $\Pr(X_2+X_3=0)=1/2+\epsilon_{2,3}$. Perhatikan bahwa penjumlahan $X_1 \oplus X_3$ dapat diturunkan dengan menambahkan $X_1 \oplus X_2$ dengan $X_2 \oplus X_3$, sehingga:

$$\Pr(X_1 \oplus X_3=0)=\Pr([X_1 \oplus X_2] \oplus [X_2 \oplus X_3]=0)$$

Jadi ekspresi lanjar dikombinasikan untuk membentuk eksprei lanjar yang baru. Karena kita bisa menganggap bahwa variabel acak $X_1 \oplus X_2$ dan $X_2 \oplus X_3$ adalah saling bebas, kita bisa menggunakan *Lemma Pilling-Up* untuk memperoleh

$$\Pr(X_1 \oplus X_3=0)=1/2+2\epsilon_{1,2}\epsilon_{2,3}$$

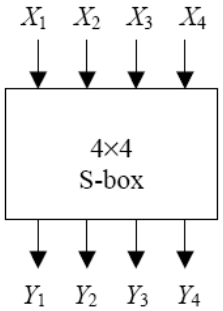
sehingga

$$\epsilon_{1,3}=2\epsilon_{1,2}\epsilon_{2,3}$$

Sesuai dengan apa yang selanjutnya dibahas, ekspresi $X_1 \oplus X_2=0$ dan $X_2 \oplus X_3=0$ akan analog dengan pendekatan lanjar terhadap S-box dan $X_1 \oplus X_3=0$ beranalogi dengan pendekatan lanjar terhadap cipher dimana bit antara, yaitu $X_2$ tidak lagi perlu dilibatkan. Tentu saja analisis yang sebenarnya akan lebih rumit karena melibatkan banyak pendekatan untuk tiap S-box.

### 3.3 Menganalisis Bagian dari Cipher

Sebelum melangkah lebih jauh pada penyerangan cipher secara keseluruhan, perlu pengetahuan tentang mudah tidaknya suatu S-box diserang. Misalkan S-Box pada gambar 2 menerima input $X=[X_1X_2X_3X_4]$ dan keluarannya adalah $Y=[Y_1Y_2Y_3Y_4]$. Semua pendekatan lanjar yang mungkin bisa diperiksa-untuk-mengetahui-kebergunaannya dengan cara menghitung nilai kemungkinan biasnya masing-masing. Jadi, kita periksa semua ekspresi dalam bentuk persamaan (1) dengan $X$ dan $Y$ masing masing merupakan masukan dan keluaran dari S-Box.




Gambar 2. Pemetaan S-box

Sebagai contoh, untuk S-box pada SIMPI, misalkan kita mengambil persamaan linear berbentuk $X_2 \oplus X_3 \oplus Y_1 \oplus Y_3 \oplus Y_4=0$ atau setara dengan:

$$X_2 \oplus X_3=Y_1 \oplus Y_3 \oplus Y_4$$

Dengan mencoba keenambelas kemungkinan nilai masukan untuk X dan mengamati keluarannya (yang bersesuaian dengan Y), diketahui bahwa terdapat 12 kasus dimana persamaan diatas berlaku. Jadi, kemungkinan biasnya adalah 12/16-1/2=1/4. hal ini dapat dilihat pada Tabel 3.

Dengan cara yang sama pula, bila persamaan yang dianalisis adalah

$$X_1 \oplus X_4=Y_2$$

dapat dilihat bahwa kemungkinan biasnya adalah 0 dan untuk persamaan

$$X_3 \oplus X_4=Y_1 \oplus Y_4$$

kemungkinan biasnya adalah 2/16-1/2=-3/8. Untuk kasus terakhir ini, pendekatan yang digunakan adalah pendekatan *affine* seperti yang diindikasikan dengan adanya tanda minus. Namun, keberhasilan suatu serangan bergantung pada nilai mutlak dari biasnya dan seperti yang akan dibahas, pendekatan *affine* bisa digunakan sama baiknya seeprti pendekatan lanjar.

| $X_1$ | $X_2$ | $X_3$ | $X_4$ | $Y_1$ | $Y_2$ | $Y_3$ | $Y_4$ | $X_2 \oplus X_3$ | $Y_1 \oplus Y_3 \oplus Y_4$ | $X_1 \oplus X_4$ | $Y_2$ | $X_3 \oplus X_4$ | $Y_1 \oplus Y_4$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |

Tabel 3. Sampel Pendekatan Lanjar Terhadap S-box

Penghitungan semua pendekatan lanjar untuk S-box dalam SIMPI diberikan dalam Tabel Pendekatan Lanjar pada tabel 4. Tiap elemen dalam tabel merepresentasikan jumlah

kesesuaian antara persamaan lanjar yang disajikan dalam bentuk heksadesimal oleh bagian ”Nilai Masukan” dengan penjumlahan bi-bit keluaran yang direpresentasikan dalam heksadesimal oleh bagian “Nilai Keluaran” yang telah dikurangi dengan delapan. Jadi, membagi nilai elemen tabel dengan 16 memberikan nilai kemungkinan bias untuk kombinasi lanjar tertentu dari bit-bit masukan dan keluaran. Nilai heksadesimal merepresentasikan penjumlahan bila dilihat sebagai nilai biner yang mengindiksaikan variabel yang terlibat dalam penjumlahan. Untuk kombinasi lanjar dari variabel masukan yang direpresentasikan sebagai $a_1 \cdot X_1 \oplus a_2 \cdot X_2 \oplus a_3 \cdot X_3 \oplus a_4 \cdot X_4$ dengan $a_i \in \{0,1\}$dan "$\cdot$" melambangkan operator biner AND, nilai heksadeimal merepresentasikan nilai biner $a_1a_2a_3a_4$., dengan $a_1$ adalah *most significant bit*. Serupa pula, untuk kombinasi lanjar dari bit-bit keluaran $b_1 \cdot Y_1 \oplus b_2 \cdot Y_2 \oplus b_3 \cdot Y_3 \oplus b_4 \cdot Y_4$ dengan $b_i \in \{0,1\}$, nilai heksadesimal merepresentasikan niali biner $b_1b_2b_3b_4$. Jadi, nilai bias dari persamaan lanjar $X_3 \oplus X_4 = Y_1 \oplus Y_4$ (masukan heksa 3 dan keluaran heksa 9) adalah -6/16 =-3/8 dan kemungkinan persamaan lanjar tersebut benar adalah ½-3/8=1/8.

Terdapat beberapa karakteristik dasar dari tabel pendekatan lanjar. Kemungkinan bahwa sembarang kombinasi penjumlahan lanjar tidak kosong dari bit-bit keluaran adalah tepat ½ karena sembarang kombinasi lanjar dari bit-bit keluaran pasti memiliki jumlah nol dan satu yang sama untuk S-box yang bersifat bijeksi (S-box nxn). Begitu pula sembarang kombinasi penjumlahan lanjar yang tidak melibatkan bit-bit keluaran sama dengan kombinasi lanjar yang tidak melibatkan bit-bit masukan sehingga menghasilkan nilai bias +1/2 dan nilai tabel +8 untuk pojok kiri atas. Karakteristik ini terlihat pada tabel dimana baris teratas tabel bernilai nol semuanya, kecuali nilai elemen yang terkiri. Dan kolom pertama semua berisi nilai nol kecuali nilai bagian teratas. Dapat diamati pula bahwa penjumlahan dari sembarang baris atau sembarang kolom adalah +8 atau -8.

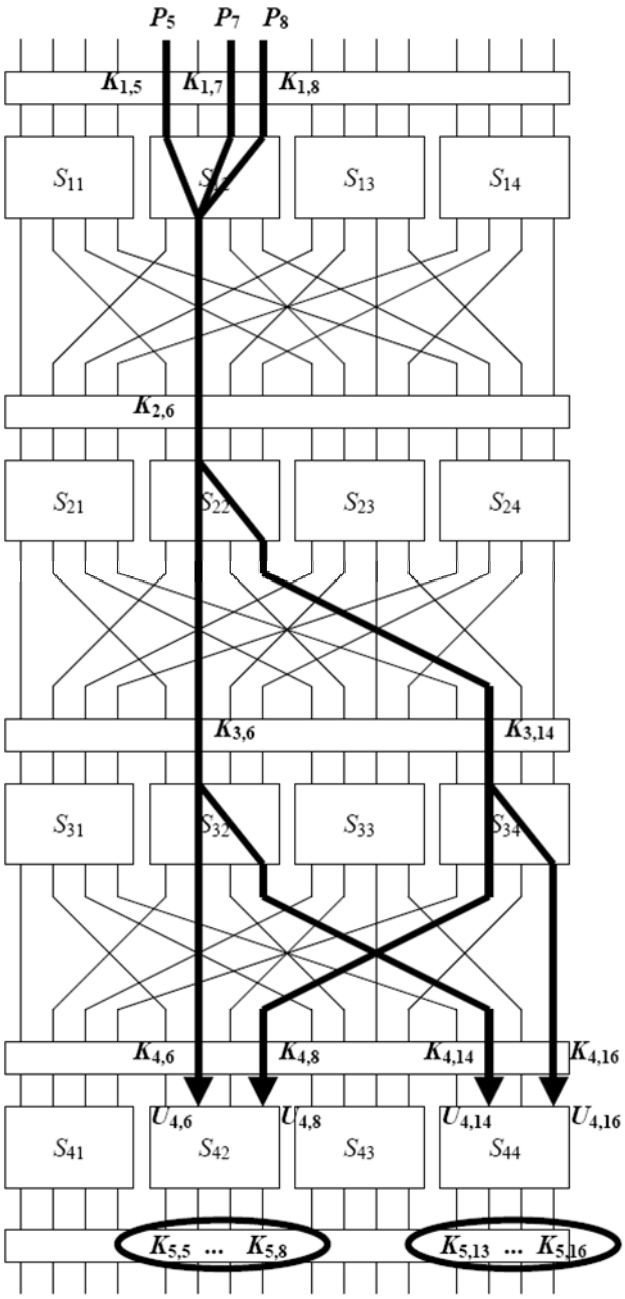
| | | Nilai Keluaran | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| Nilai Masukan | 0 | +8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 1 | 0 | 0 | –2 | –2 | 0 | 0 | –2 | +6 | +2 | +2 | 0 | 0 | +2 | +2 | 0 | 0 |
| | 2 | 0 | 0 | –2 | –2 | 0 | 0 | –2 | –2 | 0 | 0 | +2 | +2 | 0 | 0 | –6 | +2 |
| | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | +2 | –6 | –2 | –2 | +2 | +2 | –2 | –2 |
| | 4 | 0 | +2 | 0 | –2 | –2 | –4 | –2 | 0 | 0 | –2 | 0 | +2 | +2 | –4 | +2 | 0 |
| | 5 | 0 | –2 | –2 | 0 | –2 | 0 | +4 | +2 | –2 | 0 | –4 | +2 | 0 | –2 | –2 | 0 |
| | 6 | 0 | +2 | –2 | +4 | +2 | 0 | 0 | +2 | 0 | –2 | +2 | +4 | –2 | 0 | 0 | –2 |
| | 7 | 0 | –2 | 0 | +2 | +2 | –4 | +2 | 0 | –2 | 0 | +2 | 0 | +4 | +2 | 0 | +2 |
| | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | –2 | +2 | +2 | –2 | +2 | –2 | –2 | –6 |
| | 9 | 0 | 0 | –2 | –2 | 0 | 0 | –2 | –2 | –4 | 0 | –2 | +2 | 0 | +4 | +2 | –2 |
| | A | 0 | +4 | –2 | +2 | –4 | 0 | +2 | –2 | +2 | +2 | 0 | 0 | +2 | +2 | 0 | 0 |
| | B | 0 | +4 | 0 | –4 | +4 | 0 | +4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | C | 0 | –2 | +4 | –2 | –2 | 0 | +2 | 0 | +2 | 0 | +2 | +4 | 0 | +2 | 0 | –2 |
| | D | 0 | +2 | +2 | 0 | –2 | +4 | 0 | +2 | –4 | –2 | +2 | 0 | +2 | 0 | 0 | +2 |
| | E | 0 | +2 | +2 | 0 | –2 | –4 | 0 | +2 | –2 | 0 | 0 | –2 | –4 | +2 | –2 | 0 |
| | F | 0 | –2 | –4 | –2 | –2 | 0 | +2 | 0 | 0 | –2 | +4 | –2 | –2 | 0 | +2 | 0 |

Tabel 4. Tabel Pendekatan Lanjar

### 3.4 Menyusun Pendekatan Lanjar untuk Keseluruhan Cipher

Sekali informasi mengenai pendekatan lanjar tiap S-box dalam cipher Substitusi-Permutasi diperoleh, kita telah memiliki sarana untuk melanjutkan ke proses menyusun pendekatan lanjar untuk keseluruhan cipher dalam bentuk persamaan (1). Hal ini bisa diperoleh dengan menggabungkan pendekatan lanjar dari tiap S-box yang sesuai. Dengan menyusun sebuah pendekatan lanjar yang hanya melibatkan bit-bit plainteks dan bit-bit data masukan untuk putaran terakhir, dimungkinkan dilakukan serangan terhadap cipher memperoleh bagian dari bit-bit upa kunci yang dipakai pada putaran terakhir. Penggambarannya diberikan dengan contoh berikut ini.

Diamati pendekatan yang melibatkan $S_{12}$, $S_{22}$, $S_{32}$, dan $S_{34}$ sebagaimana diilustrasikan pada gambar 3. Perlu dicatat bahwa dengan melakukan ini, sebenarnya disusun sebuah ekspresi pendekatan untuk tiga putaran pertama dari cipher(bukan keempat putaran). Akan diperlihatkan tentang kebergunaan hal ini dalam memperoleh bit-bit upa kunci yang digunakan setelah putaran terakhir.




Gambar 3. Sampel Pendekatan Lanjar

Misalkan diambil pendekatan berikut terhadap S-Box:

- $S_{12}$: $X_1 \oplus X_3 \oplus X_4 = Y_2$ dengan kemungkinan 12/16 dan bias +1/4
- $S_{22}$: $X_2 = Y_2 \oplus Y_4$ dengan kemungkinan 4/16 dan bias –1/4
- $S32$: $X_2 = Y_2 \oplus Y_4$ dengan kemungkinan 4/16 dan bias –1/4
- $S34$: $X_2 = Y_2 \oplus Y_4$ dengan kemungkinan 4/16 dan bias –1/4

Dengan $U_i(V_i)$ melambangkan blok 16-bit dari bit-bit masukan(keluaran) pada S-box putaran ke-i dan $U_{i,i}(V_{i,j})$ melambangkan bit ke-j dari blok $U_i(V_i)$ (dimana bit-bit dinomori dari 1 sampai 16 dari kiri ke kanan pada SIMPI). Serupa pula, $K_i$ merepresentasikan bit-bit dari blok upa kunci yang di-XOR-kan dengan masukan pada putaran ke-i, dengan pengecualian bahwa $K_5$ adalah upa kunci yang di-XOR-kan dengan keluaran dari putaran keempat.

Jadi, $U_1 = P \oplus K_1$ dengan P merepresentasikan blok plainteks 16 bit dan "⊕" melambangkan operasi bit XOR. Menggunakan pendekatan lanjar pada putaran pertama, kita peroleh

$$V_{1,6} = U_{1,5} \oplus U_{1,7} \oplus U_{1,8} \quad (2)$$
$$= (P_5 \oplus K_{1,5}) \oplus (P_7 \oplus K_{1,7}) \oplus (P_8 \oplus K_{1,8})$$

dengan kemungkinan ¾. Untuk pendekatan pada putaran kedua, kita memiliki

$$V_{2,6} \oplus V_{2,8} = U_{2,6}$$

dengan kemungkinan ¼.

Karena $U_{2,6} = V_{1,6} \oplus K_{2,6}$, maka

$$V_{2,6} \oplus V_{2,8} = V_{1,6} \oplus K_{2,6}$$

dengan kemungkinan ¼ dan dengan mengkombinasikannya dengan (2) yang berlaku dngan kemungkinan ¾ memberikan persamaan

$$V_{2,6} \oplus V_{2,8} \oplus P_5 \oplus P_7 \oplus P_8 \oplus$$
$$K_{1,5} \oplus K_{1,7} \oplus K_{1,8} \oplus K_{2,6} = 0 \quad (3)$$

yang berlaku dengan kemungkinan sebesar ½+12(3/4-1/2)(1/4-1/2) = 3/8 (yaitu, dengan bias sebesar -1/8), yang dihitung memanfaatkan *Lemma Pilling-Up*. Catat bahwa diasumsikan bahwa tiap pendekatan terhadap S-box adalah saling bebas yang pada kenyataannya hal ini tidak sepenuhnya benar namun pada prakteknya karena asumsi inilah serangan-serangan terhadap kebanyakan cipher berhasil dilakukan.

Pada putaran ketiga, persamaan lanjarnya adalah

$$V_{3,6} \oplus V_{3,8} = U_{3,6}$$

dengan kemungkinan ¼ dan

$$V_{3,14} \oplus V_{3,16} = U_{3,14}$$

dengan kemungkinan ¼. Dengan $U_{3,6} = V_{2,6} \oplus K_{3,6}$ dan $U_{3,14} = V_{2,8} \oplus K_{3,14}$, diperoleh

$$V_{3,6} \oplus V_{3,8} \oplus V_{3,14} \oplus V_{3,16} \oplus V_{2,6} \oplus K_{3,6}$$
$$\oplus V_{2,8} \oplus K_{3,14} = 0 \quad (4)$$

dengan kemungkinan $1/2+2(1/4-1/2)^2 = 5/8$ (yaitu dengan bias sebesar +1/8). Lagi-lagi, kita menggunakan *Lemma Pilling-Up*.

Dengan mengombinasikan persamaan (3) dan (4) untuk menggabungkan keempat pendekatan S-box, kita peroleh

$$V_{3,6} \oplus V_{3,8} \oplus V_{3,14} \oplus V_{3,16} \oplus P_5 \oplus P_7 \oplus P_8 \oplus K_{1,5} \oplus K_{1,7} \oplus K_{1,8} \oplus K_{2,6} \oplus K_{3,6} \oplus K_{3,14} = 0$$

Karena $U_{4,6}=V_{3,6} \oplus K_{4,6}$, $U_{4,8}=V_{3,14} \oplus K_{4,8}$, $U_{4,14}=V_{3,8} \oplus K_{4,14}$, dan $U_{4,16}=V_{3,16} \oplus K_{4,16}$, kita bisa merampatkan persamaan pendekatan tersebut menjadi

$$U_{4,6} \oplus U_{4,8} \oplus U_{4,14} \oplus U_{4,16} \oplus P_5 \oplus P_7 \oplus P_8 \oplus \Sigma K = 0$$

dengan

$$\Sigma K = K_{1,5} \oplus K_{1,7} \oplus K_{1,8} \oplus K_{2,6} \oplus K_{3,6} \oplus K_{3,14} \oplus K_{4,6} \oplus K_{4,8} \oplus K_{4,14} \oplus K_{4,16}$$

dimana ΣK bernilai tetap yaitu 0 atau 1 tergantung pada nilai kunci cipher. Dengan mengaplikasikan *Lemma Pilling-Up*, ekspresi lanjar diatas akan berlaku dengan kemungkinan $1/2+2^3(3/4–1/2)(1/4–1/2)^3 = 15/32$ (yaitu, dengan bias sebesar –1/32).

Karena ΣK adalah tetap, maka

$$U_{4,6} \oplus U_{4,8} \oplus U_{4,14} \oplus U_{4,16} \oplus P_5 \oplus P_7 \oplus P_8 = 0 \quad (5)$$

akan berlaku dengan kemungkinan sebesar 15/32 atau (1-15/32)=17/32 tergantung apakah ΣK=0 atau 1. Dengan kata lain, kita telah memperoleh pendekatan lanjar untuk tiga putaran pertama dari cipher dengan tingkatan bias sebesar 1/32. Selanjutnya akan dibahas bagaimana bias ini dipakai untuk menentukan beberapa nilai bit dari kunci.

### 3.5 Mengekstraksu Bit-bit Kunci

Setelah pendekatan lanjar untuk R-1putaran diketahui pada cipher yang memiliki R putaran, dengan kemungkinan bias lanjar yang cukup besar, serangan terhadap cipher bisa dilakukan untuk memperoleh bit-bit dari upa kunci terakhir yang digunakan terakhir. Dalam konteks SIMPI, dimungkinkan untuk mengekstreaksi bit-bit dari upa kunci $K_5$ bila telah diketahui pendekatan untuk tiga putaran pertama. Untuk selanjutnya, bit-bit dari upa kunci terakhir yang akan dicari nilainya disebut sebagai bit-bit upa kunci target. Jadi, bit-bit upa kunci target adalah bit-bit dari upa kunci terakhir yang berhubungan dengan operasi S-box pada putaran terakhir yang dipengaruhi oleh bit-bit data yang terlibat dalam pendekatan lanjar. Proses yang akan dijelaskan selanjutnya melibatkan pendekripsian parsial cipherteks pada putaran terakhir cipher. Dalam hal ini, untuk semua kemungkinan nilai bagi bit-bit upa kunci target, cipherteks dikenai operasi XOR dengan bit-bit upa kunci target yang bersesuaian dan hasilnya dilewatkan ke operasi substitusi S-box secara terbalik(nilai masukan menjadi keluaran, nilai keluaran menjadi masukan). Hal ini dilakukan untuk semua pasangan sampel plainteks/cipherteks dimana sebuah variabel penghitung disediakan untuk tiap kemungkinan nilai dari bit-bit upa kunci target. Variabel penghitung ini nilainya ditambah satu bila ekspresi lanjar bernilai benar untuk bit-bit yang akan memasuki S-box putaran terakhir (bit-bit ini diperoleh dengan melakukan dekripsi parsial yang telah dijelaskan sebelumnya) dan bit-bit plainteksnya. Kombinasi nilai dari bit-bit upa kunci target yang memiliki nilai penghitung yang paling menyimpang jauh dari setengah jumlah sampel plainteks/cipherteks adalah diasumsikan sebagai nilai yang benar untuk bit-bit upa kunci target. Asumsi ini adalah berlaku karena diasumsikan bahwa nilai bit-bit upa kunci yang benar akan mengakibatkan pendekatan lanjar berlaku dengan kemungkinan yang meyimpang paling jauh dari ½(apakah itu diatas atau dibawah 1/2 tergantung apakah pendekatan yang terbaik tersebut adalah lanjar atau *affine* dan hal ini bergantung pada nilai bit-bit upa kunci yang tidak diketahui yang secara implisit diibatkan dalam ekspresi lanjar). Nilai upa kunci yang salah diasumsikan menghasilkan nilai keluaran yang cukup acak pada bit-bit yang memasuki S-box pad putaran terakhir dan akibanya, eksprsi lanjar akan berlaku dengan kemungkinan mendekati ½.

Sekarang konsep ini akan diterapkan pada SIMPI. Ekspresi lanjar (5) mempengaruhi masukan untuk S-box $S_{42}$ dan $S_{44}$ pada putaran terakhir. Untuk tiap sampel plainteks/ cipherteks, dicoba semua 256 nilai yang mungkin untuk bit-bit upa kunci target $[K_{5,5}…K_{5,8}, K_{5,13}…K_{5,16}]$. Untuk tiap nilai bit-bit upa kunci target, kita akan menambah nilai variabel penghitungnya dengan satu ketika persamaan (5) berlaku, dimana kita menentukan nilai dari $[U_{4,5}…U_{4,8}, U_{4,13}…U_{4,16}]$ dengan mengoperasikannya terhadap bit-bit upa kunci target dan S-box $S_{24}$ dan $S_{44}$. Nilai bit-bit upa kunci target yang memiliki nilai penghitung yang paling meyimpang dari nilai separuh jumlah sampel plainteks/cipherteks diasumsikan sebagai nilai bit-bit upa kunci target yang benar. Apakah penyimpangan tersebut bernilai positif ataukah negatif tergantung pada nilai dari bit-bit upa kunci yang terlibat dalam ΣK. Ketika ΣK = 0, pendekatan lanjar (5) akan berlaku dengan kemungkinan <1/2 dan ketika ΣK = 1, (5) akan berlaku dengan kemugkinan >½.

Simulasi penyerangan terhadap SIMPI dilakukan dengan menggunakan 10000 known plainteks/cipherteks dan dengan mengikuti proses kriptanalisis yang telah dijelaskan untuk nilai bit-bit upa kunci $[K_{5,5}\ldots K_{5,8}] = [0010]$ dan $[K_{5,13}\ldots K_{5,16}] = [0100]$. Seperti yang diharapkan, nilai penghitung yang paling meyimpang dari nilai 5000 ternyata berkorespondensi dengan nilai bit-bit upa kunci target [2,4], yang mengindikasikan bahwa penyerangan telah berhasil memperoleh bit-bit upa kunci target. Tabel 5 memperlihatkan sebagian hasil dari data yang diambil dari nilai penghitung tiap nilai upa kunci target. (data yang lengkap melibatkan 256 entri, masing-masing untuk tiap kemungkinan nilai bit-bit upa kunci target). Entri pada tabel menunjukkan derajat bias yang dihitung dengan

$$|\text{bias}| = |\text{count} - 5000| / 10000$$

dengan count adalah nilai variabel penghitung untuk tiap nilai bit-bit upa kunci target.

Sebagaimana dapat diamati dari hasil parsial dari tabel, nilai bias terbesar terjadi untuk nilai bit-bit upa kunci $[K_{5,5}\ldots K_{5,8}, K_{5,13}\ldots K_{5,16}] = [2,4]$ dan hasil ini adalah benar sebagai nilai sebenarnya dari bit-bit upa kunci target.

| *bit up kunci* $[K_{5,5}\ldots K_{5,8}, K_{5,13}\ldots K_{5,16}]$ | \| bias \| | *bit up kunci* $[K_{5,5}\ldots K_{5,8}, K_{5,13}\ldots K_{5,16}]$ | \| bias \| |
|---|---|---|---|
| 1 C | 0.0031 | 2 A | 0.0044 |
| 1 D | 0.0078 | 2 B | 0.0186 |
| 1 E | 0.0071 | 2 C | 0.0094 |
| 1 F | 0.0170 | 2 D | 0.0053 |
| 2 0 | 0.0025 | 2 E | 0.0062 |
| 2 1 | 0.0220 | 2 F | 0.0133 |
| 2 2 | 0.0211 | 3 0 | 0.0027 |
| 2 3 | 0.0064 | 3 1 | 0.0050 |
| **2 4** | **0.0336** | 3 2 | 0.0075 |
| 2 5 | 0.0106 | 3 3 | 0.0162 |
| 2 6 | 0.0096 | 3 4 | 0.0218 |
| 2 7 | 0.0074 | 3 5 | 0.0052 |
| 2 8 | 0.0224 | 3 6 | 0.0056 |
| 2 9 | 0.0054 | 3 7 | 0.0048 |

Tabel 5. Sebagian Hasil Percobaan Serangan Lanjar terhadap SIMPI

Nilai bias percobaan sebesar 0,0336 adalah sangat mendekati nilai bias yang diharapkan secara teoritis, yaitu 1/32 = 0,03125. Perlu dicatat bahwa meskipun nilai bit-bit upa kunci target yang benar secara jelas memiliki bias yang terbesar, nilai bias besar lainnya yang mungkin muncul mengindikasikan bahwa pengecekan dari bit-bit upa kunci target yang salah tidak secara tepat berpadanan dengan membandingkan data acak terhadap ekspresi lanjar (yang mana biasnya bisa saja sangat mendekati nol). Ketidakkonsistensian bias dalam percobaan dapat muncul karena beberapa alasan termasuk diantaranya karena properti dari S-box yang mempengaruhi dekripsi parsial untuk nilai upa kunci yang berbeda., ketidak akuratan atas asumsi mengenai kesalingbebasan yang diperlukan agar bisa menggunakan *Lemma Pilling-Up*, dan karena pengaruh dari *linear hulls* yang akan dijelaskan selanjutnya.

### 3.6 Kompleksitas Serangan

S-box yang terlibat dalam pendekatan lanjar disebut sebagai S-box aktif. Pada gambar 3, keempat S-box pada putaran satu sampai tiga yang terlewati garis tebal adalah S-box aktif. Besarnya kemungkinan suatu ekspresi lanjar cipher berlaku berhubungan dengan kemungkinan bias lanjar pada tiap S-box aktif dan jumlah S-box yang aktif. Secara umum, semakin besar tingkatan nilai bias dalam S-box, semakin besar pula derajat nilai bias untuk keseluruhan ekspresi pendekatan untuk cipher. Semakin sedikit jumlah S-box yang aktif, semakin besar derajat nilai bias untuk keseluruhan ekspresi lanjar cipher.

Misalkan $\varepsilon$ merepresentasikan nilai bias terhadap ½ dari kemungkinan bahwa ekspresi lanjar untuk keseluruhan cipher berlaku. Dalam makalahnya, Matsui menunjukkan bahwa jumlah known palinteks yang diperlukan dalam serangan adalah sebanding dengan $\varepsilon^{-2}$, dan bila $N_L$ melambangkan jumlah known palinteks yang diperlukan dalam serangan, adalah beralasan untuk mengaproksimasi $N_L$ dengan

$$N_L \approx 1/\varepsilon^2$$

Dalam prakteknya, dibutuhkan known plainteks sejumlah beberapa kelipatan kecil $\varepsilon^{-2}$. Meskipun kompleksitas kriptanalisis dapat dikarakteristikkan dari sisi waktu dan ruang (atau memori), kita akan mengacu pada jumlah data yang diperlukan untuk menyusun serangan bila berbicara mengenai kompleksitas kriptanalisis. Yaitu, diasumsikan bahwa jika penyerang bisa mendapatkan $N_L$ plainteks, serangan terhadap cipher bisa disusun.

Karena nilai bias diperoleh dengan memanfaatkan *Lemma Pilling-Up* dimana tiap variabel yang digunakan dalam perkalian mengacu pada pendekatan S-box, nilai bias tergantung pada nilai-nilai bias dari pendekatan-pendekatan lanjar terhadap S-box dan jumlah S-box aktif yang terlibat. Strategi yang umum dilakukan dalam meningkatkan keamanan cipher terhadap kriptanalisi lanjar

berfokus pada pengoptimisasian S-box (meminimalkan nilai bias yang terbesar) dan pencarian struktur cipher yang akan memaksimalkan jumlah S-box yang aktif. Prinsip desain dari Rijndael adalah contoh yang bagus dalam desain cipher yang aman dari serangan lanjar.

Kita perlu waspada bahwa konsep pembuktian keamanan cipher terhadap kriptanalisis lanjar biasanya berpedoman pada tidak adanya pendekatan lanjar yang berkemungkinan besar. Penghitungan kemungkinan dari pendekatan lanjar didasarkan pada asumsi bahwa tiap pendekaan untuk S-box adalah saling bebas (sehingga *Lemma Pilling-Up* bisa digunakan) dan asumsi bahwa satu skenario pendekatan lanjar (misalnya satu set S-box aktif tertentu) adalah cukup untuk memperoleh ekspresi lanjar terbaik yang menghubungkan bit-bit plainteks dengan bit-bit data masukan pada putaran terakhir. Kenyataannya adalah pendekatan-pendekatan S-box adalah tidak saling bebas dan hal ini bisa berdampak serius pada perhitungan kemungkinan bias. Juga, skenario pendekatan lanjar yang melibatkan plainteks dan bit-bit masukan untuk putaran terakhir yang sama dengan S-box aktif yang berbeda bisa saling dikombinasikan untuk memperoleh nilai kemungkinan lanjar yang lebih besar daripada hanya digunakan satu set S-box aktif tertentu. Konsep ini disebut sebgai *linear hull* [16]. Sebagai contoh, sejumlah skenario pendekatan lanjar mungkin masing-masing hanya memiliki nilai bias yang sangat kecil dan dari tiap skenario ini mengindikasikan bahwa cipher tersebut kebal terhadap serangan lanjar namun bila skenario-skenario ini digabungkan, hasil ekspresi lanjar terhadap plainteks dan bit-bit masukan untuk putaran trakhir akan bisa memiliki nilai bias yang sangat tinggi. Meskipun demikian, pendekatan yang diberikan dalam makalah ini nampaknya cukup bagus digunakan dalam berbagai cipher karena asumsi kesaling-bebasan adalah pendekatan yang cukup bagus dan ketika satu skenario pendekatan lanjar pada pemilihan S-box aktif memiliki nilai bias yang tinggi, hal ini akan mendominasi *linear hull*.

## 4. Kriptanalisis Differensial

Bagian ini berfokus pada konsep kriptanalisis diferensial dan aplikasinya pada SIMPI.

### 4.1 Gambaran Umum Serangan

Kriptanalisis diferensial mengeksploitasi tingginya kemungkinan munculnya *perbedaan* tertentu pada plainteks dan data pada masukan untuk putaran terakhir cipher. Misalkan suatu sistem dengan masukan $X = [X_1\ X_2\ \dots\ X_n]$ dan keluaran $Y = [Y_1\ Y_2\ \dots\ Y_n]$ dan kedua masukan untuk sistem adalah $X'$ dan $X''$ serta keluarannya masing-masing adalah $Y'$ dan $Y''$. *Perbedaan* masukan didefinisikan sebagai $\Delta X = X' \oplus X''$ dengan "$\oplus$ " melambangkan operasi bit XOR pada rangkaian $n$-bit, maka

$$\Delta X = [\Delta X_1\ \Delta X_2\ \dots\ \Delta X_n]$$

dengan $\Delta X_i = X_i' \oplus X_i''$ dengan $X_i'$ dan $X_i''$ merepresentasikan bit ke-i dari $X'$ dan $X''$. Serupa pula, $\Delta Y = Y' \oplus Y''$ adalah *perbedaan* dari keluaran dan

$$\Delta Y = [\Delta Y_1\ \Delta Y_2\ \dots\ \Delta Y_n]$$

dengan $\Delta Y_i = Y_i' \oplus Y_i''$.

Dalam cipher pengacak yang ideal, kemungkinan bahwa suatu *perbedaan* keluaran $\Delta Y$ tertentu muncul diberikan suatu *perbedaan* masukan $\Delta X$ tertentu adalah $1/2^n$ dengan $n$ adalah jumlah bit pada $X$. Kriptanalisis diferensial berusaha mengeksploitasi skenario dimana suatu $\Delta Y$ tertentu muncul dengan kemungkinan yang sangat besar diberikan suatu *perbedaan* masukan $\Delta X$ (yaitu jauh lebih besar daripada $1/2^n$). Pasangan $(\Delta X, \Delta Y)$ disebut sebagai sebuah *diferensial*.

Kriptanalisis diferensial.adalah serangan *chosen plainteks*, yang berarti penyerang bisa memilih masukan dan memeriksa keluaran dalam usaha memperoleh kunci. Dalam kriptanalisis diferensial, penyerang akan memilih pasangan-pasangan masukan $X'$ dan $X''$, yang memenuhi $\Delta X$ tertentu, dengan mengetahui bahwa untuk $\Delta X$ tersebut, nilai $\Delta Y$ tertentu muncul dengan kemungkinan yang besar.

Dalam makalah ini, disajikan langkah penyusunan suatu differensial $(\Delta X, \Delta Y)$ yang melibatkan bit-bit plainteks yang direpresentasi-kan sebagai $X$ dan masukan untuk putaran terakhir cipher yang direpresentasikan sebagai $Y$. Selanjutnya akan diperiksa karakteristik diferensial yang berkemungkinan besar muncul dimana suatu karaktersitik differensial adalah suatu urutan *perbedaan* masukan dan keluaran pada putaran sehingga *perbedaan* keluaran suatu putaran berkorespondensi dengan *perbedaan* masukan untuk putaran selanjutnya. Menggunakan karakteristik *perbedaan* yang berkemungkinaan besar akan memberikan jalan dalam mengeksploitasi informasi yang memasuki putaran terakhir untuk memperoleh bit-bit upa kunci yang dipakai pada putaran terakhir.

Serupa dengan kriptanalisis lanjar, untuk mnyusun karaktristik *perbedaan*, diperiksa

properti dari masing-masing S-box dan menggunakannya untuk menentukan karakterisik diferensial cipher secara keseluruhan yaitu mengamati perbedaan masukan dan keluaran pada S-box untuk menentukan pasangan diferensial yang kemungkinannya besar muncul. Dengan mengkombinasikan pasangan diferensial S-box dari putaran ke putaran selanjutnya sehingga bit-bit diferensial keluaran yang tidak nol dari satu putaran berkoresponden dengan bit-bit diferensial masukan yang tidak nol pada putaran selanjutnya membuat kita bisa menemukan diferensial yang kemungkinannya besar yang terdiri atas *perbedaan* plainteks dan *perbedaan* masukan untuk putaran terakhir. Bit-bit upa kunci akan tidak terlibat dalam ekpresi *perbedaan* karena bit-bit ini terlibat pada kedua data sehingga dengan melibatkan pengaruhnya pada perbedaan berarti akan mengXOR-kan bit-bit upa kunci dengan dirinya sendiri yang hasilnya tentu saja adalah nol.

### 4.2 Menganalisis bagian-bagian cipher

Pertama-tama akan diperiksa pasangan diferensial suatu S-box. Perhatikan representasi S-box 4x4 pada gambar 2 dengan masukan $X = [X1\ X_2\ X_3\ X_4]$ dan keluaran $Y = [Y_1\ Y_2\ Y_3\ Y_4]$. Semua pasangan diferensial suatu S-box, $(\Delta X, \Delta Y)$ bisa diperiksa dan kemungkinan munculnya $\Delta Y$ diberikan $\Delta X$ dapat diturunkan dengan memperhatikan pasangan masukan $(X', X'')$ sehingga $X' \oplus X'' = \Delta X$. Karena urutan pasangan tidak berpengaruh, untuk S-box 4x4 hanya perlu diperhatikan 16 nilai untuk $X'$ dan nilai $\Delta X$ akan memberikan batasan untuk nilai $X''$ yaitu $X'' = X' \oplus \Delta X$.

Dengan menggunakan S-box pada SIMPI, dapat diturunkan nilai $\Delta Y$ untuk tiap pasangan masukan $(X', X'' = X' \oplus \Delta X)$. Sebagai contoh, nilai biner dari $X$, $Y$ dan nilai $\Delta Y$ yang bersesuaian diberikan pasangan masukan $(X, X \oplus \Delta X)$ dapat dilihat pada tabel 6 untuk nilai $\Delta X$ = 1011(heksa B), 1000(heksa 8), dan 0100 (heksa 4). Tiga kolom terakhir tabel merepresentasikan nilai $\Delta Y$ untuk suatu nilai $X$(pada baris) dan $\Delta X$(pada kolom). Dari tabel, kita bisa melihat jumlah munculnya $\Delta Y$= 0010 untuk $\Delta X$ = 1011 adalah 8 dari 16 kemungkinan kemunculan (yaitu kemungkinannya adalah 8/16); jumlah kemunculan $\Delta Y$ = 1011 dengan $\Delta X$ = 1000 adalah 4 dari 16; kemungkinan munculnya $\Delta Y$ = 1010 diberikan $\Delta X$ = 0100 adalah 0. Jika S-box adalah "ideal", jumlah kemunculan nilai pasangan diferensial masing-masing adalah satu agar memberikan kemungkinan 1/16 bagi munculnya $\Delta Y$ tertentu diberikan $\Delta X$ (dalam hal ini, S-box yang "ideal" ini secara matematis tidaklah mungkin ada).

| $X$ | $Y$ | $\Delta Y$ | | |
|---|---|---|---|---|
| | | $\Delta X$ = 1011 | $\Delta X$ = 1000 | $\Delta X$ = 0100 |
| 0000 | 1110 | 0010 | 1101 | 1100 |
| 0001 | 0100 | 0010 | 1110 | 1011 |
| 0010 | 1101 | 0111 | 0101 | 0110 |
| 0011 | 0001 | 0010 | 1011 | 1001 |
| 0100 | 0010 | 0101 | 0111 | 1100 |
| 0101 | 1111 | 1111 | 0110 | 1011 |
| 0110 | 1011 | 0010 | 1011 | 0110 |
| 0111 | 1000 | 1101 | 1111 | 1001 |
| 1000 | 0011 | 0010 | 1101 | 0110 |
| 1001 | 1010 | 0111 | 1110 | 0011 |
| 1010 | 0110 | 0010 | 0101 | 0110 |
| 1011 | 1100 | 0010 | 1011 | 1011 |
| 1100 | 0101 | 1101 | 0111 | 0110 |
| 1101 | 1001 | 0010 | 0110 | 0011 |
| 1110 | 0000 | 1111 | 1011 | 0110 |
| 1111 | 0111 | 0101 | 1111 | 1011 |

Tabel 6. Sampel pasangan Diferensial dari S-box

Kita bisa mentabulasikan data lengkap suatu S-box dalam tabel distribusi diferensial dengan baris merepresentasikan nilai $\Delta X$ (dalam heksadesimal) dan koom merepresentasikan nilai $\Delta Y$ (dalam heksadesimal). Tabel distribusi diferensial untuk S-box pada tabel 1 diberikan pada tabel 7. Tiap elemen tabel menunjukkan jumlah kemunculan perbedaan keluaran $\Delta Y$ tertentu diberikan perbedaan masukan $\Delta X$. Perhatikan bahwa selain kasus ($\Delta X$ = 0, $\Delta Y$ = 0), nilai terbesar dalam tabel adalah 8 yang berkoresponden dengan $\Delta X$ = B dan $\Delta Y$ = 2. Kemungkinan bahwa $\Delta Y$ = 2 diberikan sejumlah pasangan nilai masukan yang memenuhi $\Delta X$ = B adalah 8/16. Nilai terkecil dalam tabel adalah nol dan muncul untuk banyak pasangan diferensial. Dalam kasus ini, kemungkinan nilai $\Delta Y$ muncul diberikan nilai $\Delta X$ adalah nol.

Terdapat beberapa properti umum dari tabel distribusi perbedaan yang perlu untuk disebutkan. Pertama, perlu dicatat bahwa jumlah semua elemen dalam tiap baris adalah $2^n$ = 16; serupa pula, jumlah semua elemen tiap kolom adalah $2^n$ = 16. Semua nilai elemen adalah genap: hal ini terjadi karena suatu nilai pasangan masukan (atau keluaran) direpresentasikan sebagai $(X', X'')$ memiliki $\Delta X$ yang sama dengan pasangan $(X'', X')$ karena $\Delta X = X' \oplus X'' = X'' \oplus X'$. Juga, perbedaan masukan $\Delta X$ = 0 harus menghasilkan perbedaan keluaran $\Delta Y$ = 0 karena sifat pemetaan satu ke satu dari

S-box. Hal ini menjelaskan mengapa pojok kanan atas tabel memiliki nilai $2^n = 16$ dan semua nilai lainnya pada baris pertama dan kolom pertama adalah nol. Jika kita bisa membuat S-box yang ideal, yang tidak memberikan informasi diferensial mengenai keluaran bila diberikan nilai masukan, S-box ini akan menyebabkan semua elemen tabel bernilai satu agar kemungkinan kemunculan suatu $\Delta Y$ tertentu diberikan suatu $\Delta X$ tertentu haruslah $1/2^n = 1/16$. Namun karena peroperti yang disebutkan sebelumnya harus juga dipenuhi, S-box ideal ini tidaklah mungkin tecapai.

| perbedaan masukan | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 2 | 0 | 2 | 4 | 0 | 4 | 2 | 0 | 0 |
| 2 | 0 | 0 | 0 | 2 | 0 | 6 | 2 | 2 | 0 | 2 | 0 | 0 | 0 | 0 | 2 | 0 |
| 3 | 0 | 0 | 2 | 0 | 2 | 0 | 0 | 0 | 0 | 4 | 2 | 0 | 2 | 0 | 0 | 4 |
| 4 | 0 | 0 | 0 | 2 | 0 | 0 | 6 | 0 | 0 | 2 | 0 | 4 | 2 | 0 | 0 | 0 |
| 5 | 0 | 4 | 0 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 4 | 0 | 2 | 0 | 0 | 2 |
| 6 | 0 | 0 | 0 | 4 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 |
| 7 | 0 | 0 | 2 | 2 | 2 | 0 | 2 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 4 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 4 | 0 | 4 | 2 | 2 |
| 9 | 0 | 2 | 0 | 0 | 2 | 0 | 0 | 4 | 2 | 0 | 2 | 2 | 2 | 0 | 0 | 0 |
| A | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 6 | 0 | 0 | 2 | 0 | 0 | 4 | 0 |
| B | 0 | 0 | 8 | 0 | 0 | 2 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 2 |
| C | 0 | 2 | 0 | 0 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 2 | 0 | 6 | 0 | 0 |
| D | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 4 | 2 | 0 | 2 | 0 | 2 | 0 | 2 | 0 |
| E | 0 | 0 | 2 | 4 | 2 | 0 | 0 | 0 | 6 | 0 | 0 | 0 | 0 | 0 | 2 | 0 |
| F | 0 | 2 | 0 | 0 | 6 | 0 | 0 | 0 | 0 | 4 | 0 | 2 | 0 | 0 | 2 | 0 |

Tabel 7. Tabel Distribusi Perbedaan pada SIMPI

Sebelum berlanjut membicarakan pengkombinasian pasangan-pasangan *perbedaan* S-box untuk memperoleh sebuah karakteris-tik diferensial dan perkiraan diferensial yang bagus untuk digunakan dalam serangan, kharus diketahui pengaruh kunci pada diferensial S-box. Perhatikan gambar 4 dimana masukan untuk S-box tak berkunci adalah $X$ dan keluarannya adalah $Y$. Namun dalam struktur cipher ini, kita harus memperhatikan kunci yang dikombinasikan dengan masukan untuk S-box. Dalam hal ini, jika dimisalkan masukan untuk S-box berkunci adalah $W = [W_1\ W_2\ W_3\ W_4]$ , perbedaan masukan untuk S-box berkunci ini adalah

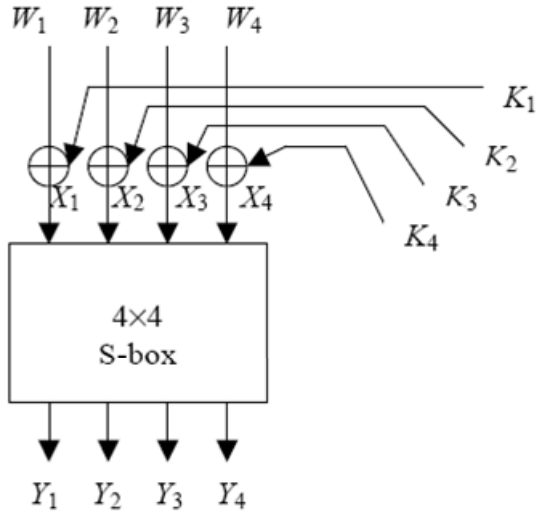
$$\Delta W = [W_1' \oplus W_1''\ \ W_2 \oplus W_2''\ \ \ldots\ W_n' \oplus W_n'']$$

dengan $W' = [W_1'\ W_2'\ \ldots\ W_n']$ dan $W'' = [W_1''\ W_2''\ \ldots\ W_n'']$ merepresentasikan kedua nilai masukan.

Karena bit-bit kunci adalah sama untuk kedua $W'$ dan $W''$,

$$\Delta W_i = W_i' \oplus W_i'' = (Xi' \oplus Ki) \oplus (X_i'' \oplus K_i)$$
$$= X_i' \oplus X_i'' = \Delta X_i$$

karena $Ki \oplus Ki = 0$. Jadi, bit-bit kunci tidak memiliki pengaruh pada nilai perbedaan masukan dan bisa diacuhkan; .dengan kata lain, S-box berkunci memiliki tabel distribusi perbedaan yang sama dengan S-box tak berkunci.



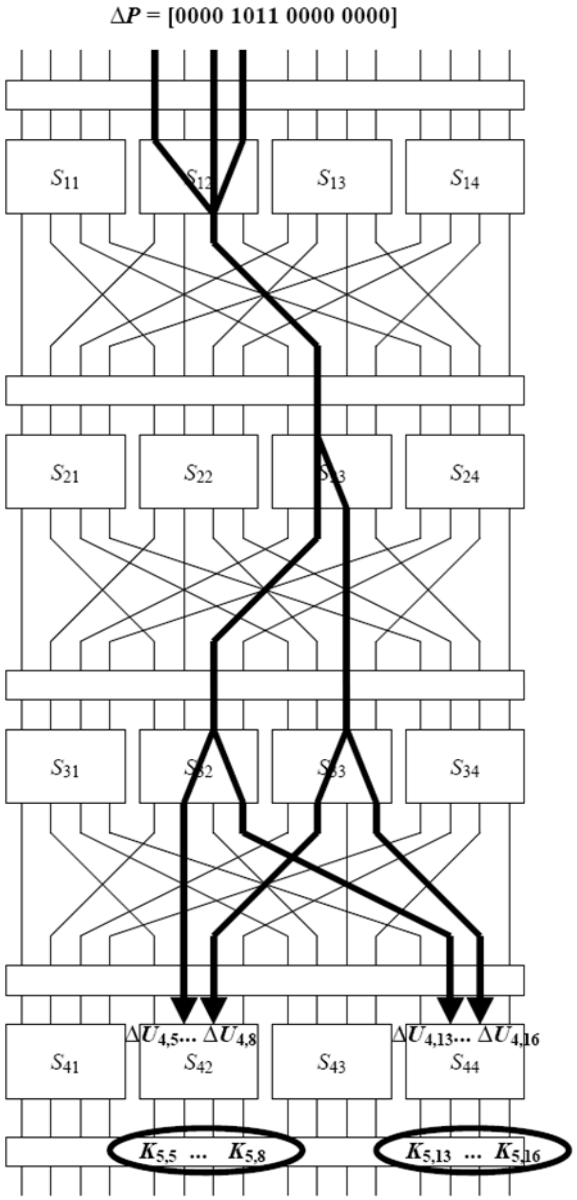

gambar 4. S-box dengan kunci

### 4.3 Menyusun Karakteristik Diferensial

Sekali informasi diferensial diperoleh untuk S-box dalam cipher berbasiskan Substitusi-Permutasi, kita memiliki data untuk selanjutnya menyusun suatu karakteristik diferensial yang berguna untuk keseluruhan cipher. Hal ini bisa diperoleh dengan menggabungkan pasangan perbedaan yang sesuai dari S-box S-box. Dengan menyusun suatu karakteristik diferensial dari pasangan-pasangan perbedaan S-box tertentu dari tiap putaran sehingga diferensial tersebut melibatkan bit-bit plainteks dan bit-bit data untuk masukan S-box putaran terakhir, dimungkinkan dilakukan penyerangan terhadap cipher untuk mengekstraksi sebagian bit-bit upa kunci yang dipakai pada putaran terakhir. Kita ilustrasikan penyusunan karakteristik diferensial dengan sebuah contoh.

Misalkan suatu karakteristik diferensial melibatkan $S_{12}$, $S_{23}$, $S_{32}$, dan $S_{33}$ pada SIMPI. Seperti halnya dengan kriptanalisis lanjar, karaktersitik ini dapat dilihat pada gambar 5. Diagarm tersebut mengilustrasikan pengaruh dari perbedaan yang tidak nol seiring merambat dalam cipher, sehingga S-box yang terlibat (S-box aktif yaitu yang tidak memiliki diferensial bernilai nol) jelas terlihat. Perlu dicatat bahwa dengan melakukan proses ini kita menyusun karakteristik diferensial untuk tiga putarn petama (bukannya empat putaran). Hal ini akan

berguna untuk memperooleh bit-bit upa kunci terakhir.




Gambar 5. Sampel Karakteristik Differensial pada SIMPI

Disunakan pasangan diferensial S-box berikut:

- $S_{12}$: $\Delta X = \mathrm{B} \rightarrow \Delta Y = 2$ dengan kemungkinan 8/16
- $S_{23}$: $\Delta X = 4 \rightarrow \Delta Y = 6$ dengan kemungkinan 6/16
- $S_{32}$: $\Delta X = 2 \rightarrow \Delta Y = 5$ dengan kemungkinan 6/16
- $S_{33}$: $\Delta X = 2 \rightarrow \Delta Y = 5$ dengan kemungkinan 6/16

S-box lainnya memiliki perbedaan masukan nol yang berakibat pada perbedaan keluaran yang juga bernilai nol.

*Perbedaan* masukan untuk cipher adalah sama dengan *perbedaan* masukan untuk putaran pertama dan diberikan sebagai :

$\Delta P = \Delta U = [0000\ 1011\ 0000\ 0000]$

dimana digunakan notasi yang sama seperti pada kriptanalisis lanjar sebelumnya, yaitu $U_i$ merepresentasikan msukan untuk S-box pada putaran ke-i dan $V_i$ merepresenastikan keluaran dari S-box pada putaran ke-i. Jadi, $\Delta U_i$ dan $\Delta V_i$ merepresentasikan perbedaannya masing-masing. Sebagai hasilnya,

$\Delta V_1 = [0000\ 0010\ 0000\ 0000]$

Dengan memperhatikan pasangan perbedaan untuk $S_{12}$ diatas dan dengan mengikuti pemutasi pada putaran pertama diperoleh:

$\Delta U_2 = [0000\ 0000\ 0100\ 0000]$

dengan kemungkinan 8/16 = ½ diberikan perbedaan plainteks ΔP.
Selanjutnya, diferensial putaran kedua menggunakan pasangan diferensial untuk $S_{23}$menghasilkan

$\Delta V_2 = [0000\ 0000\ 0110\ 0000]$

dan permutasi pada putaran kedua menghasilkan

$\Delta U_3 = [0000\ 0010\ 0010\ 0000]$

dengan kemungkinan 6/16 diberikan $\Delta U_2$ dan kemungkinan 8/16 x 6/16 = 3/16 diberikan ΔP.

Dalam menentukan kemungkinan diberikan perbedaan plainteks ΔP, diasumsikan bahwa diferensial pada putaran pertama adalah bebas terhadap diferensial pada putaran kedua, sehingga kemungkinan keduanya muncul ditentukan dengan hasil perkalian dari kemung-kinannya.

Selanjutnya, kita bisa menggunakan perbedaan pada S-box pada putaran ketiga, $S_{32}$ dan $S_{33}$, dan permutsai pada putaran ini untuk memperoleh

$\Delta V_3 = [0000\ 0101\ 0101\ 0000]$

dan

$\Delta U_4 = [0000\ 0110\ 0000\ 0110]$ (6)

dengan kemungkinan $(6/16)^2$ diberikan $\Delta U_3$, dan kemungkinan $8/16 \times 6/16 \times (6/16)^2 = 27/1024$ diberikan perbedaan plainteks ΔP, dimana digunakan lagi asumsi kesaling bebasan antara pasangan-pasangan perbedaan S-box di semua putaran.

Selama proses kriptanalisis, akan dilakukan pengenkripsian banyak pasangan palinteks

dengan $\Delta P$ = [0000 1011 0000 0000]. Dengan kemungkinan tinggi yaitu 27/1024, karakteristik diferensial akan muncul. Pasangan yang demikian untuk $\Delta P$ disebut sebagai pasangan-pasangan yang benar. Pasangan perbedaan plainteks dimana karakteristikny tidak muncul disebut sebagai pasangan-pasangan yang salah.

### 4.4 Mengekstraksi Bit-Bit Kunci

Sekali karakteristik diferensial untuk R-1 putaran untuk cipher R putaran diketahui dengan kemungkinan yang cukup besar, dimungkinkan dilakukan serangan tehadap cipher untuk memperoleh bit-bit dari upa kunci terakhir yang dalam hal SIMPI adalah mengekstraksi bit-bit dari upa kunci $K_5$. Proses untuk memperolehnya melibatkan pendekripsian parsial cipherteks pada putaran terakhir cipher dan memeriksa masukan bagi putaran terakhir untuk menentukan bahwa pasangan yang benar telah muncul. Bit-bit upa kunci yang dipakai setelah putaran terakhir yang dipengaruhi oleh perbedaan tidak nol pada keluaran diferensial selanjutnya disebut sebagai sebagai bit-bit upa kunci target. Pendekripsian parsial pada putaran terakhir melibatkan semua S-box putaran terakhir yang dipengaruhi oleh perbedaan tak nol dalam diferensialnya, operasi XOR pada cipherteks dengan bit-bit upa kunci target dan melewatkan data ecara terbalik melalui S-box(masukan menjadi keluaran, keluaran menjadi masukan), dimana semua kemungkinan nilai dari bit-bit upa kunci target dicobakan.

Dekripsi parsial dilakukan pada tiap pasangan cipherteks yang bersesuaian dengan pasangan plainteks yang digunakan untuk memperoleh perbedaan masukan $\Delta P$ untuk semua kemungkinan nilai bit-bit upa kunci target. Sebuah variabel penghitung diberikan untuk tiap nilai bit-bit upa kunci target. Penghitung ini akan ditambah satu nilainya bila *perbedaan* untuk masukan ke putaran terakhir bersesuaian dengan nilai yang diharapkan oleh karakteristik diferensial. Niali bit-bit upa kunci yang memiliki penghitung bernilai terbesar diasumsikan sebagai nilai bit-bit upa kunci target yang benar. Hal ini berlaku karena diasumsikan bahwa nilai bit-bit upa kunci target yang benar akan mengakibatkan perbedaan masukan ke putaran terakhir yang sesuai dengan karakteristik akan sering muncul (yaitu munculnya pasangan yang benar) karena karakteristik memiliki kemungkina kemunculan yang besar.. (Bilaa suatu pasangan yang salah muncul, meskipun dengan dekripsi parsial melibatkan upa kunci yang benar, nilai dari penghitung untuk nilai upa kunci yang benar akan tidak ditambah). Nilai upa kunci yang salah diasumsikan akan memunculkan keluaran yang relatif acak pada bit-bit yang memasuki S-box putaran terakhir dan sebagai hasilnya, *perbedaan* yang diharapkan oleh karakteristik akan berkemungkinan kecil muncul.

Perhatikan serangan pada SIMPI, karakteristik diferensial mempengaruhi masukan untuk S-box $S_{42}$, $S_{44}$ pada putaran terakhir. Utuk tiap pasangan cipherteks, dicobakan semua 256 kemungkinan nilai untuk [$K_{5,5}$…$K_{5,8}$, $K_{5,13}$ … $K_{5,16}$]. Untuk tiap nilai bit-bit upa kunci, penghitungnya akan ditambah nilainya bila perbedaan maukan untuk putaran terakhir ([$\Delta U_{4,5}$… $\Delta U_{4,8}$, $\Delta U_{4,13}$ … $\Delta U_{4,16}$]) yang diperoleh dengan mendekripsi secara parsial adalah sama dengan (6). Untuk tiap nilai bit-bti upa kunci, penghitung memberikan jumlah kemunculan perbedaan yang konsisten dengan pasangan-pasangan yang benar (dengan asumsi upa kunci parsial adalah bernilai benar). Penghitung yang mmiliki nilai terbesar mengindikasikan bahwa bit-bit upa kunci yang bersesuaian adalah bit-bitupa kunci target yang benar karena diasumsikan kbahwa kita sedang mengharapkan kemunculan tingkat tinggi dari pasangan yang benar.

Perlu dicatat bahwa kita tidak perlu melakukan dekripsi parsial untuk tiap pasangan cipherteks . Karena perbedaan masukan untuk putaran terakhir hanya memeprngaruhi dua S-box, maka ketika karakteristik muncul (yaitu untuk pasangan yang benar), perbedaan bit-bit cipherteks yagn berkorespondensi dengan S-box $_{41}$dan S-box $_{43}$ haruslah nol. Jadi, bisa disaring banyak pasangan yang salah dengan mengacuhkan pasangan-pasangan cipherteks yang tidak menghasilkan nilai-nilai nol pada upa-blok-yang-tepat pada perbedaan cipherteks. Dalam hal ini, karena pasangan cipherteks tidak bisa menjadi pasangan yang benar, tidak perlu kita memeriksa [$\Delta U_{4,5}$ … $\Delta U_{4,8}$, $\Delta U_{4,13}$ … $\Delta U_{4,16}$].

Simulasi penyerangan diferensial trhadap SIMPI dilakukan dengan 5000 pasangan *chosen plainteks/ cipherteks* (yaitu 1000 enkripsi dengan pasangan plainteks yang memenuhi $\Delta P$ = [0000 1011 0000 0000]) dan dengan mengikuti proses yang telah dijelaskan diatas. Bit-bit upa kunci target yang benar adalah [$K_{5,5}$ … $K_{5,8}$, $K_{5,13}$ … $K_{5,16}$] = [0010, 0100] = [2,4]. Seperti yang diharapkan, penghitung bernilai terbesar dimiiki oleh upa kunci parsial bernilai [2,4] yang menunjukkan bahwa penyerangan telah berhasil mmeperoleh bit-bit upa kunci. Tabel 8 memperlihatkan sebagian hasil dari data yang diambil dari penghitung pada upa kunci.( Data yang lengkap

melibatkan 256 entri, masing-masing untuk tiap nilai dari bit-bit upa kunci). Nilai pada tabel mengindikasikan kemungkinan munculnya pasangan-pasangan yang tepat untuk masing-msaing kandidat upa kunci, diperoleh dengan

prob = count / 5000

dengan count adalah penghitung yang dimiliki tiap kemungkinan nilai bit-bit upa kunci.

Seperti yang dapat dilihat dari sampel hasil pada tabel, kemungkinan yang terbesar muncul untuk nilai bit-bit upa kunci [$K_{5,5}$ … $K_{5,8}$, $K_{5,13}$ … $K_{5,26}$] = [2,4] dan ini sesuai dengan kenyataan nilai upa kunci ini adalah nilai upa kunci yang benar.

Dalam percobaan ini,diharapkan kemungkinan kemunculan pasangan yang tepat sebesar $p_D$ = 27/1024 = 0,0264 dan dari hasil percobaan, kemungkinan pada upa-kunci-yang-benar [2,4] memberikan $p_D$ = 0,0244. Perlu dicatat bahwa kadang-kadang nilai penghitung yang besar lainnya muncul untuk nilai bit-bit upa kunci target yang salah. Hal ini mengindikaikan bahwa pemeriksaan terhadap upa kunci target yang salah adalah tidak secara tepat sepadan dengan membandingkan perbedaan yang acak dengan nilai perbedaan yang diharapkan. Terdapat beberapa faktor yang memperngaruhi nilai penghitung berbeda dengan prediksi teoritis diantaranya karena properti S-box mempengaruhi dekripsi parsial untuk upa kunci yang berbeda, ketidakakuratan asumsi kesalingbebasan yang diperlukan dalam penentuan kemungkinan karaktreistik, dan konsep bahwa diferensial bisa dibentuk dari banyak karakteristik diferensial (untuk selanjutnya dibahas).

| *bit up kunci* [$K_{5,5}$...$K_{5,8}$, $K_{5,13}$...$K_{5,16}$] | prob | *bit up kunci* [$K_{5,5}$...$K_{5,8}$, $K_{5,13}$...$K_{5,16}$] | prob |
|---|---|---|---|
| 1 C | 0.0000 | 2 A | 0.0032 |
| 1 D | 0.0000 | 2 B | 0.0022 |
| 1 E | 0.0000 | 2 C | 0.0000 |
| 1 F | 0.0000 | 2 D | 0.0000 |
| 2 0 | 0.0000 | 2 E | 0.0000 |
| 2 1 | 0.0136 | 2 F | 0.0000 |
| 2 2 | 0.0068 | 3 0 | 0.0004 |
| 2 3 | 0.0068 | 3 1 | 0.0000 |
| **2 4** | **0.0244** | 3 2 | 0.0004 |
| 2 5 | 0.0000 | 3 3 | 0.0004 |
| 2 6 | 0.0068 | 3 4 | 0.0000 |
| 2 7 | 0.0068 | 3 5 | 0.0004 |
| 2 8 | 0.0030 | 3 6 | 0.0000 |
| 2 9 | 0.0024 | 3 7 | 0.0008 |

Gambar 8. Hasil Percobaan Serangan Diferensia

### 3.5 Kompleksitas Serangan

Dalam kriptanalisis diferensial, kita mngacu pada S-bok yang memiliki karakteristik dimana perbedaan masukan tidak nol (sehingga perbedaan keluarannya juga tidak nol) sebagai S-box aktif. Secara umum, semakin besar kemungkinan diferensial dari S-box S-box yang aktif, semakin besar kemungkinan karakteristik-nya. Seperti pada kriptanalisis lanjar, dalam membicarakan mengenai kompleksitas kriptanalisis, yang diacu adalah banyak data yang diperlukan untuk menyusun serangan. Yaitu, diasumsikan jika penyerang bisa memperoleh $N_D$ plainteks, serangan bisa dilakukan.

Secara umum, adalah sangat rumit untuk menentukan dengan tepat jumlah pasangan *chosen plainteks* yang diperlukan untuk menyusun serangan. Pedoman yang cukup bagus untuk menentukan jumlah pasangan chosen plainteks $N_D$ yang diperlukan untuk mengenali pasangan-pasangan yang benar ketika mencoba kandidat upa kunci adalah

$$N_D \approx c / p_D \qquad (7)$$

Dengan $p_D$ adalah karakteristik diferensial untuk R-1 putaran dari cipher R putaran dan c adalah konstanta yang kecil. Dengan mengasumsikan bahwa kemunculan pasangan-pasangan perbedaan pada tiap S-box aktif adalah saling bebas, kemungkinan karakteristik diferensial diberikan dengan

$$p_D = \prod_{i=1}^{\gamma} \beta_i \qquad (8)$$

dengan jumalh S-box aktif adalah $\gamma$ dan kemunculan pasangan perbedaan tertentu pada S-box ke-i dari karakteristik memiliki kemung-kinan yang direpresentasikan dengan $\beta_i$.

Persamaan (7) mnegindikasikan bahwa sedikit kemunculan dari pasangan yang tepat adalah cukup untuk menyimpulkan penghitung pada nilai bit-bit upa target yang benar nilainya jauh lebih besar daripada penghitung pada nilai bit-bit upa kunci target yang salah. Karena pasangan yang benar diharapkan untuk muncul kira-kira setiap $1/p_D$ pasangaan yang diperikas, biasanya diperlukan sejumlah kelipatan dari $1/p_d$ pasangan–pasangan chosen plainteks untuk menyusun penyerangan yang berhasil.

Pendekatan untuk melindungi cipher terhadap kriptanalisis diferensial biasanya berfokus pada peoperti S-box (yaitu meminimalkan kemung-kinan pasangan perbedaan pada suatu S-box) dan mendesain struktur untuk memaksimalkan jumlah S-box aktif. Lagi-lagi, Rijndael adalah

contoh dari desain cipher yang memberikan perlindungan tinggai terhadap kriptanalisis diferensial.

Seperti pada kriptanalisis lanjar, kewaspadaan haruslah diperhatikan dalam membuktikan kekebalan cipher terhadap kriptanalisis diferensial. Penghitungan pada kemungkinan karakteristik diferensial berpedoman pada asumsi kesalingbebasan S-box yang terlibat dalam pendekatan, yang pada kenyataannya tidak benar: terdapat ketergantungan antara data yang memasuki S-box yang berbeda. Jadi, kemungkinan $p_D$ adalah perkiraan saja meskipun dalam prakteknya pada banyak cipher, pendekatan ini terbukti cukup baik..

Karakteristik-karakteristik diferensial yang berbeda dengan perbedaan masukan dan keluaran yang sama (yaitu diferensial yang sama) dapat dikombinasikan untuk menghasilkan kemungkinan diferensial yang lebih besar daripada bila hanya menggunakan satu karakteristik diferensial saja [17] (Hal ini analog dengan konsep *linear hulls*). Untuk membuktikan keamanan terhadap kriptanalisis diferensial, perlu dibuktikan bahwa semua kemungkinan nilai karakteristik diferensial adalah dibawah suatu batasan yang dapat dibuktikan keamanannya.. Namun pada umumnya, ketika suatu karakteristik diferensial memiliki kemungkinan yang tinggi, ini akan mendominasi kemunculan diferensial dan pada prakteknya kemungkinan karakteristik memberikan pendekatan yang cukup bagus pada kemungkinan diferensial.

## 5. Simpulan

Dalam makalah ini telah disajikan konsep-konsep dasar dari kriptanalisis lanjar dan diferensial serta aplikasinya pada SIMPI, seuah cupher blok sederhana yang berbasiskan substitusi-permutasi.. SIMPI tidak ditujukan untuk digunakan dalam praktek nyata , namun, struktur cipher ini berguna untuk mengetahu isejuh mana penerapan kriptanalisis lanjar dan diferensial bisa diterapkan. Aplikasi dari konsep kriptanalisis lanjar dan diferensial pada SIMPI diharapkan dapat membantu memahami cara kerja kedua serangan ini.

## 6. Daftar Psutaka


[1] M. Matsui, "Linear Cryptanalysis Method for DES Cipher", *Advances in Cryptology - EUROCRYPT '93*, Springer-Verlag, pp. 386-397, 1994.

[2] E. Biham and A. Shamir, "Differential Cryptanalysis of DES-like Cryptosystems", 1991.

[3] U.S. DEPARTMENT OF COMMERCE / National Institute of Standards and Technology, "Data Encryption Standard (DES)", *FederalInformation Processing S tandards Publication*, Reaffirmed 1999 October 25.

[4] M. Matsui, "The First Experimental Cryptanalysis of the Data Encryption Standard", *Advances in Cryptology - CRYPTO '94*, Springer-Verlag, pp. 1-11, 1994.

[5] E. Biham and A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.

[6] National Institute of Standards, Advanced Encryption Standard (AES) web site: www.nist.gov/aes.

[7] J. Daemen and V. Rjimen, "AES Proposal: Rijndael", *First Advanced Encryption Standard (AES) Conference*, California, Aug. 1998.

[8] H.M. Heys and S.E. Tavares, "Substitution-Permutation Networks Resistant to Differential and Linear Cryptanalysis", *Journal of Cryptology*, vol. 9, no.1, pp. 1-19, 1996.

[9] L. Keliher, "Linear and Differential Cryptanalysis of SPNs", unpublished.

[10] L. Knudsen, "Block Ciphers: A Survey", *State of the Art in Applied Cryptography: Course on Computer Security and Industrial Cryptography*, Springer-Verlag, pp. 18-48, 1998.

[11] D.R. Stinson, *Cryptography: Theory and Practice*, CRC Press, 1995.

[12] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd ed., John Wiley & Sons, 1995.

[13] A. J. Menezes, P.C. van Oorschot, and S.A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997.

[14] W. Stallings, *Cryptography and Network Security: Principles and Practices*, 3rd ed., Prentice Hall, Forth Indian Reprint 2004.

[15] H. Feistel, "Cryptography and Computer Privacy", *Scientific American*, vol. 228, no. 5, pp. 15-23, 1973.

[16] K. Nyberg, "Linear Approximations of Block Ciphers", *Advances in Cryptology - EUROCRYPT '94*, Springer-Verlag, pp. 439-444, 1995.

[17] X. Lai, J.L. Massey, and S. Murphy, "Markov Ciphers and Differential Cryptanalysis", *Advances in Cryptology - EUROCRYPT '91*, Springer-Verlag, pp. 17-38, 1991.

[18] E. Biham, "On Matsui's Linear Cryptanalysis", *Advances in Cryptology - EUROCRYPT '94*, Springer-Verlag, pp. 341-355, 1995.

[19] F. Chabaud and S. Vaudenay, "Links Between Differential and Linear Cryptanalysis", *Advances in Cryptology* -

*EUROCRYPT '94*, Springer-Verlag, pp. 356-365, 1995.
[20] M. Hellman and S. Langford, "Differential-Linear Cryptanalysis", *Advances in Cryptology - CRYPTO '94*, Springer-Verlag, pp. 26-39, 1994.
[21] L.R. Knudsen, "Truncated and Higher Order Differentials", *Fast Software Encryption (Lecture Notes in Computer Science no. 1008)*, Springer-Verlag, pp. 196-211, 1995.
[22] E. Biham, A. Biryukov, and A. Shamir, " Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials", *Advances in Cryptology - EUROCRYPT '99*, Springer-Verlag, pp. 55-64, 1996.
[23] M.J.B. Robshaw and B.S. Kaliski, "Linear Cryptanalysis Using Multiple Approximations", *Advances in Cryptology - CRYPTO '94*, Springer-Verlag, pp. 1-11, 1994.
[24] L. Knudsen and M.J.B. Robshaw, "Nonlinear Approximations in Linear Cryptanalysis", *Advances in Cryptology - EUROCRYPT '96*, Springer-Verlag, pp. 224-236, 1996.
[25] K. Nyberg, "Differentially Uniform Mappings for Cryptography, *Advances in Cryptology - EUROCRYPT '93*, Springer-Verlag, pp. 55-64, 1994.
[26] E. De Win, A. Bosselaers, B. Preneel, J. Daemen, and V. Rijmen, "The Cipher SHARK", *Fast Software Encryption*, Springer-Verlag, pp. 99-112, 1996.
[27] A.M. Youssef, S. Mister, and S.E. Tavares, "On the Design of Linear Transformations for Substitution Permutation Encryption Networks", *Workshop on Selected Areas of Cryptography (SAC '96): Workshop Record*, pp. 40-48, 1997.
[28] M. Matsui, "On Correlation Between the Order of S-boxes and the Strength of DES", *Advances in Cryptology - EUROCRYPT '94*, Springer-Verlag, pp. 366-375, 1995.