Studi Mengenai Unbalanced Feistel Network

Oleh : Muhammad Arif Romdhoni (13503108)

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

E-mail : arif.romdhoni@gmail.com

Abstraksi

Jaringan Feistel merupakan salah satu algoritma kriptografi yang diperkenalkan oleh Horst Feistel, seorang ilmuwan IBM pada saat itu. Algoritma ini menggunakan fungsi F untuk melakukan operasi pada setengah blok saja, yang nantinya akan dilakukan beberapa kali perputaran sehingga seluruh blok dikenai fungsi tersebut. Algoritma ini telah diaplikasikan ke dalam beberapa algoritma yang lain, seperti DES, Khufu, Elastic Cipher Block, dan lain sebagainya. Beberapa algoritma menggunakan jaringan Feistel konvensional, yakni yang diperkenalkan oleh Feistel pertama kali. Beberapa algoritma lain melakukan modifikasi atas jaringan Feistel tersebut. Salah satunya ialah *Unbalanced Feistel Network*. Dinamakan *Unbalanced Feistel Network* karena jaringan ini mirip dengan jaringan Feistel, tetapi pembagian jumlah bit untuk setiap subblok dalam putaran tidak sama. Di dalam paper ini akan dikemukakan hasil studi atas algoritma ini.

Keyword: Jaringan Feistel, *Unbalanced Feistel Network*, *Cipher Block*

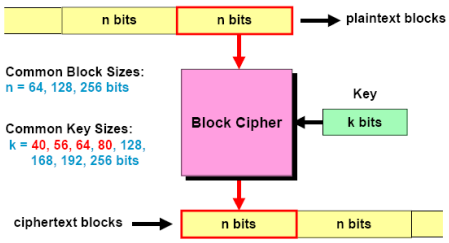
1. Pendahuluan

Ilmu kriptografi telah berkembang sejak lama. Di dalam ilmu kriptografi konvensional, kita menemukan adanya metode substitusi dan transposisi. Pada zaman dahulu, algoritma yang hanya menggunakan dua metode tersebut dapat dikatakan cukup kuat, walaupun hanya berbasiskan karakter abjad dan angka. Akan tetapi, seiring dengan perkembangan zaman dan teknologi, algoritma dari kedua metode tersebut mulai diketahui kelemahan dan kekurangannya sehingga orang berpikir untuk mendapatkan cara enkripsi data yang kuat. Walaupun demikian, kedua metode tersebut tidak lantas dibuang begitu saja, tetapi sebagai dasar munculnya kriptografi modern di masa-masa selanjutnya.

Kriptografi yang ada saat ini juga berdasarkan metode substitusi dan transposisi, akan tetapi lebih kompleks dan lebih sulit untuk diterka. Hal ini dikarenakan operasi yang ada saat ini berdasarkan bit, bukan lagi karakter sebagaimana kriptografi klasik.

Salah satu kriptografi yang ada saat ini ialah kriptografi blok. Dinamakan demikian karena kriptografi ini bekerja dalam blok-blok bit. Contoh penggambaran kriptografi blok adalah sebagai berikut:

Gambar 1 Skema umum kriptografi blok



Kriptografi blok ini banyak instansnya, seperti DES, Blowfish, dan RC6. Dan kriptografi blok ini juga telah banyak digunakan di berbagai tempat di dunia untuk mengenkripsi data.

Saat ini telah berkembang beberapa metode untuk membentuk algoritma kriptografi blok ini. Salah satu yang sering digunakan ialah jaringan Feistel.

2. Jaringan Feistel

Jaringan Feistel pertama kali diperkenalkan oleh Horst Feistel, seorang ilmuwan IBM. Metode ini digunakan dalam salah satu algoritma kriptografi blok temuannya, yakni Lucifer.

Algoritma ini menggunakan putaran-putaran kunci untuk menghasilkan output. Putaran-putaran ini mengaplikasikan prinsip dari Shannon, yakni *diffusion* dan *confusion*.

Di dalam jaringan Feistel, blok plainteks di dibagi menjadi dua bagian yang sama besar. Kemudian salah satu blok dikenakan fungsi f dan diputar. Demikian seterusnya selama beberapa kali putaran sehingga menghasilkan blok cipher yang lebih acak.

Inti dari jaringan Feistel ialah fungsi f, yang memetakan sebuah input bits menjadi output bits. Blok hasil dari fungsi f tersebut kemudian diputar. Kemudian dilakukan fungsi kembali selama beberapa kali. Untuk melakukan tugasnya, fungsi f membutuhkan adanya kunci internal. Kunci internal ini dibangkitkan dari kunci eksternal yang sebelumnya telah didefinisikan.

Fungsi f sendiri dapat diekspresikan sebagai berikut:

$$F : \{0,1\}^{n/2} \times \{0,1\}^{k} \rightarrow \{0, 1\}^{n/2}$$

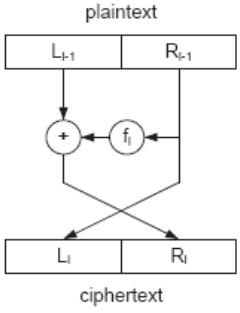
Di mana bagian pertama menyatakan bits input yang dibutuhkan oleh fungsi F tersebut, yang kedua merupakan bits kunci, dan kemudian menghasilkan bits output yang sama besar dengan bit input. Variabel *n* menyatakan panjang blok, berarti input yang diperlukan oleh jaringan Feistel ini ½ blok. Variabel *k* menyatakan panjang kunci. Sedangkan {0,1} menyatakan nilai-nilai bit yang mungkin.

Satu putaran dalam jaringan Feistel dapat didefinisikan sebagai berikut:

$$X_{i+1} = (F_{ki}(msb_{n/2}(X_i)) \oplus lsb_{n/2}(X_i)) \\ || \; msb_{n/2}(X_i)$$

X_{i+1} merupakan output dari putaran tersebut, X_i ialah input putaran, k_i merupakan kunci, *n* adalah panjang blok, lsb_u dan msb_u ialah barisan bit paling rendah dan tinggi pada blok. Tanda XOR menyatakan penjumlahan dalam modulo 2, dan || menyatakan konkatensi. Apabila dituangkan ke dalam gambar, putaran di atas akan berupa:

Gambar 2 Satu putaran di dalam Feistel



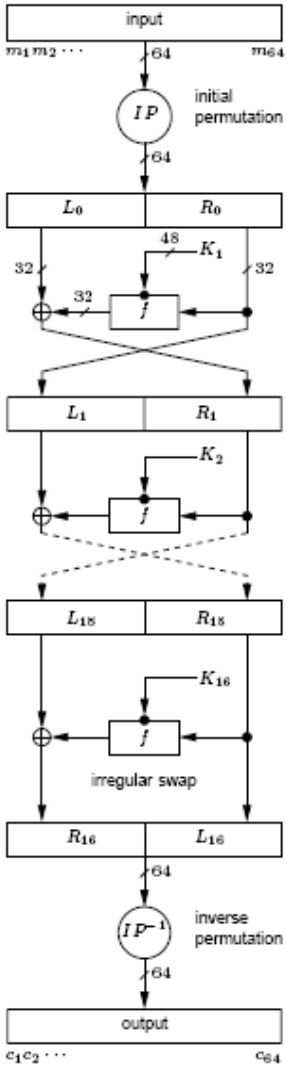
Dari gambar di atas, dapat kita simpulkan bahwa output adalah cipherteks di bagian bawah. Input berupa plainteks di sebelah atas.

Sedangkan fungsi yang bekerja berada di tengah, beroperasi dengan mode operasi XOR. Kunci diperlukan di dalam operasi XOR tersebut sehingga menghasilkan nilai fungsi yang berbeda.

Kekuatan penggunaan jaringan Feistel ini berada pada putaran di fungsi f. Sejumlah putaran yang dibutuhkan supaya aman dari serangan ditentukan oleh fungsi f tersebut.

Salah satu algoritma yang mengaplikasikan jaringan Feistel ialah DES. Gambaran putaran lengkap jaringan Feistel dalam DES:

Gambar 3 DES Putaran Lengkap



Algoritma ini dianggap kuat pada masanya, akan tetapi saat ini sudah lebih lemah karena performansi komputasi yang semakin cepat.

3. Unbalanced Feistel Network

Sebagaimana jaringan Feistel, *Unbalanced Feistel Network* juga memiliki sejumlah putaran kunci di mana salah satu bagian blok melakukan putaran atas bagian blok yang lain. Bedanya, di dalam Unbalanced Feistel Network jumlah bit di satu bagian blok dengan jumlah bit di bagian yang lain tidak sama.

Secara umum, hal ini dapat didefinisikan sebagai berikut:

Dalam sebuah putaran *s*-bit atas *t*-bit dalam jaringan Feistel, maka disebut *Unbalanced Feistel Network* jika memenuhi:

$$X_{i+1} = (F(msb_s(X_i), k_i) \oplus lsb_t(X_i)) \\ || \; msb_s(X_i)$$

Di mana $msb_s(X_i)$ merupakan blok sumber dan $lsb_t(X_i)$ merupakan blok target. Apabila s > t, *Unbalanced Feistel Network* yang terbentuk dinamakan *source heavy*. Sedangkan apabila t>s, maka disebut *target heavy*.

Fungsi F dinyatakan sebagai koleksi 2^k pemetaan dari *s*-bit ke *t*-bit. Jika diketahui bahwa subblok b merupakan *gcd* dari *s, t,* dan *n* maka fungsi F dapat dinyatakan juga sebagai pemetaan dari *s/b* subblok ke *t/b* subblok.

Berdasarkan fungsi yang digunakan dalam setiap putaran, *Unbalanced Feistel Network* dibagi menjadi dua bagian, yakni:

1. Homogen
2. Heterogen

Disebut homogen apabila fungsi F identik pada setiap putaran kecuali untuk putaran kunci. Sedangkan heterogen jika fungsi F yang digunakan tidak selalu sama untuk setiap putaran kecuali untuk putaran kunci.

Ada keuntungan menggunakan *Unbalanced Feistel Network* yang heterogen. Karena fungsi yang digunakan dari satu putaran ke putaran yang lain berbeda, maka pencarian karakteristik atau pola dari algoritma tersebut akan sulit untuk ditemukan pada cipher di tahapan yang berbeda.

Akan tetapi, metode ini juga memiliki kompleksitas yang tinggi dan biaya yang cukup mahal dalam pengaplikasian dan analisanya. Sedangkan biaya yang dibutuhkan untuk mengaplikasikan *Unbalanced Feistel Network* yang homogen lebih murah, implementasi perangkat lunaknya juga lebih mudah, dan mudah untuk mengoreksi kesalahan kode.

Oleh karena itu, di dalam makalah ini mengacu kepada *Unbalanced Feistel Network* yang homogen.

Sedangkan berdasarkan penggunaan bit dalam blok pada setiap putaran, *Unbalanced Feistel Network* dapat dinyatakan komplit dan tidak komplit.

Dinyatakan komplit apabila $s+t=n$, di mana setiap bit di dalam blok digunakan dalam putaran baik sebagai sumber maupun sebagai target. Adapun jika $s+t<n$, maka disebut tidak komplit. Di dalam *Unbalanced Feistel Network* yang tidak komplit, terdapat $n-s-t=z$ bit yang tidak digunakan, disebut sebagai blok *null*.

Dilihat dari perubahan struktur pada setiap kali putaran, *Unbalanced Feistel Network* dapat dinyatakan sebagai konsisten dan tidak konsisten.

Disebut sebagai konsisten apabila besar *s, t, n,* dan *z* di atas konstan untuk cipher yang sama. Akan tetapi, di dalam *Unbalanced Feistel Network*, besar jumlah bit sumber dan target dapat pula meningkat atau menurun ketika enkripsi. Ini disebut sebagai tidak konsisten.

Perlu dicatat bahwa *Unbalanced Feistel Network* yang tidak konsisten selalu heterogen. Sedangkan *Unbalanced Feistel Network* yang heterogen belum tentu tidak konsisten.

Karena pembahasan *Unbalanced Feistel Network* yang tidak konsisten atau tidak komplit cukup rumit, sehingga di dalam makalah ini hanya dibatasi untuk *Unbalanced Feistel Network* yang konsisten, komplit, dan homogen.

4. Cycles dan Rotation

Cycle adalah jumlah putaran yang dibutuhkan masing-masing bit untuk mencicipi berada di sumber dan target minimal sekali. Sedangkan *Rotation* ialah jumlah putaran yang diperlukan masing-masing bit untuk kembali ke posisi awalnya.

Aturan:

Sebuah *cycle* C pada sebuah s:t *Unbalanced Feistel Network* dinyatakan sebagai:

$$C = \left\lceil \frac{n}{\min(s,t)} \right\rceil$$

Aturan:

Sebuah *rotation* G pada sebuah s:t *Unbalanced Feistel Network* dinyatakan sebagai:

$$G = \left\lceil \frac{n}{\gcd(s,t)} \right\rceil$$

Jika C=G, maka *Unbalanced Feistel Network* yang terbentuk disebut genap. Jika sebaliknya, maka disebut ganjil. Sebuah *Unbalanced Feistel Network* disebut prima jika G = n.

Perlu dicatat bahwa jaringan Feistel konvensional, seperti DES atau Blowfish dapat dilihat sebagai kasus khusus dari *Unbalanced Feistel Network* yang genap dan komplit.

5. Tingkat Kekacauan

Seseorang yang mengetahui informasi mengenai X_i, tetapi tidak mengetahui kunci putaran k_i, akan kesulitan mengetahui tentang X_{i+1}. Proses di mana orang tersebut kehilangan informasi mengenai urutan nilai X_i disebut kekacauan.

Diberikan informasi mengenai X_i, maka akan terdapat X_{i+t} di mana seseorang yang tidak memiliki pengetahuan tentang kunci putaran, tidak mengetahui nilai X_{i+t} tersebut.

Di dalam jaringan Feistel, bit j dari blok dapat dikaburkan hanya ketika bit j muncul di blok target dari putaran tersebut. Ini berarti bahwa kemungkinan bit j dapat dikaburkan setiap *cycle* dapat tidak lebih besar daripada kemungkinan bit j muncul di blok sumber per *cycle*. Ini disebut tingkat kekacauan, disimbolkan dengan R_c.

Bahwa tingkat kekacauan dari sebuah *Unbalanced Fiestel Network* yang konsisten adalah jumlah minimum per *cycle* sebuah bit dapat muncul di blok target.

Aturan:

Untuk sebuah *s:t Unbalanced Fiestel Network*, tingkat kekacauannya adalah

$$R_C = \frac{t}{n}$$

Perlu dicatat bahwa tidak akan pernah seseorang kehilangan seluruh informasi mengenai suatu blok di *Unbalanced Feistel Network* lebih kecil daripada $1/R_c$.

Nilai R_C yang semakin besar menunjukkan ketahanannya atas linear kriptanalisis selama variabel yang lain bernilai konstan.

Jika sebuah *Unbalanced Feistel Network* yang komplit dan genap memiliki C putaran per *cycle*, maka fraksi putaran per *cycle* yang aktif pada serangan linear minimal $R_c = t/n$. Hal ini dikarenakan R_C memiliki ukuran jumlah minimum per *cycle* di mana satu bit atau beberapa bit mungkin dapat muncul di blok target.

Ambil p sebagai bias dari kemungkinan perkiraan terbaik untuk putaran saat ini yang ada di bawah serangan linear. Maka bias keluaran dari karakteristik linear nontrivial yang melewati C putaran paling banyak ialah $2^{(CR_C-1)} p^{CR_C}$.

Mungkin juga, apabila sebuah bit string dengan suatu ketidakpastian seluruhnya dilakukan XOR atas bit-bit perkiraan, sehingga bias perkiraan menjadi berkurang. Dengan mengasumsikan bahwa tidak ada perkiraan yang sempurna dari subset keluaran fungsi F, hal ini mengimplikasikan bahwa tingginya tingkat kekacauan mengarah kepada bias yang semakin kecil untuk perkiraan linear dikirim melalui *cycle* yang penuh. Hal ini penting untuk dicatat, bahwa tingkat kekacauan yang tinggi tidak menjamin ketahanan atas linear kriptanalisis.

6. Tingkat Difusi

Perubahan pada X_i mungkin memiliki kesempatan untuk mengubah setiap bit pada X_{i+t}, untuk beberapa nilai t. Proses di mana sebuah bit di dalam blok yang memiliki kesempatan untuk mempengaruhi bit-bit yang lan di dalam blok disebut difusi.

Di dalam Feistel cipher, waktu dimana bit j dapat mempengaruhi bit-bit yang lain di dalam blok adalah ketika bit j berada di blok sumber. Ini secara natural menunjukkan ide dari tingkat difusi.

Tingkat difusi adalah jumlah terkecil *cycle* di mana sebuah bit dapat memiliki kesempatan untuk mempengaruhi bit-bit yang lain di dalam blok tersebut.

Untuk sebuah *s:t Unbalanced Feistel Network* tingkat difusi disimbolkan dengan:

$$R_d <= \frac{s}{n}$$

Sebagai sebuah bit tunggal yang berada di dalam *Unbalanced Feistel Network* yang komplit dan genap, serta *source heavy*, merupakan input bagi fungsi F sebanyak C-1 kali, setelah ia mengenkripsi dirinya sendiri. Setiap kali ia digunakan secara berbeda, di mana setiap input ke fungsi F bernilai unik. Setiap bit akan digunakan oleh n+s-1 bit lainnya, bahkan mungkin lebih. Setelah satu

cycle, setiap bit disebarkan sebanyak C-1 kali melalui blok, menggunakan C-1 aplikasi yang berbeda dari fungsi F.

Tingkat difusi merupakan ukuran berapa banyak per *cycle* setiap bit digunakan untuk melakukan enkripsi atas bit-bit yang lain. Hal ini dibatasi oleh proporsi blok yang digunakan sebagai input oleh fungsi F. DES memiliki tingkat difusi yang lebih rendah, yang merupakan karakteristik fungsi F algoritma tersebut. Baik blowfish maupun CAST memiliki tingkat difusi yang tinggi untuk jaringan Feistel konvensional : 1/2.

Perlu dicatat bahwa bit-bit yang berada dalam subblok yang sama tidak dapat mempengaruhi bit-bit yang lain secara langsung. Bit-bit ini dapat saling mempengaruhi hanya dengan mempengaruhi bit-bit lain yang tidak berada dalam subblok yang sama, yang kemudian akan mempengaruhi bit-bit di dalam subblok yang sama.

Nilai R_d yang bertambah akan meningkatkan ketahanan akan diferensial kriptanalisis, ketika seluruh variabel yang lain adalah konstan.

Sebuah putaran yang aktif di bawah serangan diferensial merupakan putaran di mana terdapat perbedaan non zero input ke fungsi F.

Jika sebuah *Unbalanced Feistel Network* yang komplit dan genap memiliki C putaran per *cycle*, maka fraksi putaran per *cycle* yang aktif di dalam serangan diferensial minimal R_d = s/n.

Hal ini mengikuti definisi dari sebuah putaran aktif di bawah serangan diferensial dan dari definisi tingkat difusi.

Misalkan p sebagai kemungkinan terbesar untuk sebuah karakteristik nontrivial untuk mempengaruhi sebuah putaran. Kemudian kemungkinan untuk karakteristik nontrivial melalui putaran-putaran konsekutif maksimal $p^{(CR)}$, asumsi bahwa karakteristik-karakteristik tersebut digabungkan dengan mengalikan probabilitas mereka.

Sebuah karakteristik diferensial memiliki probabilitas yang dihitung dengan mengalikan seluruh probabilitas karakteristik konstituennya. Tidak ada karakteristik putaran C yang dapat lebih sedikit dari karakteristik-karakteristik satu putaran aktif $C_{d,}$ dan karakteristik-karakteristk putaran yang tidak aktif memiliki probabilitas satu. Karena tidak ada karakteristik putaran aktif yang dapat memiliki probabilitas melebihi p, jelas tidak mungkin suatu karakteristik putaran C memiliki nilai lebih tinggi daripada probabilitas p^{CR_d}.

Mungkin saja, setiap waktu di mana perbedaan input muncul di dalam blok sumber, kecuali perbedaan output terjadi dengan probabilitas satu, karakteristik diferensial menjadi berkurang kemungkinannya untuk sukses melewati sebuah *cycle*. Bagaimanapun, penting untuk dicatat bahwa tingkat difusi yang tinggi saja tidak cukup untuk menahan serangan diferensial. Properti-properti diferensial fungsi F juga harus dimasukkan ke dalam akun. Tambahan, serangan diferensial pada *source-heavy Unbalanced Feistel Network* (dengan tingkat difusi yang relatif tinggi) cukup kompleks karena pada kenyataannya masukan untuk putaran-putaran yang sukses berhubungan dekat. Hal ini akan didiskusikan di bawah ini.

7. Contoh Formulasi

Walaupun sebagian besar desain block cipher menggunakan jaringan Feistel konvensional, terdapat beberapa literatur yang menggunakan *Unbalanced Feistel Network*.

[1] MacGuffin

MacGuffin merupakan 48:16 *Unbalanced Feistel Network* (b=16 dan n = 64) yang dirancang untuk memperkenalkan konsep *Unbalanced Feistel Network*. Fungsi F sangat mirip dengan DES, memiliki permutasi, XOR untuk 48 bit, sebuah substitusi S-Box, dan permutasi lain. Ukuran S-box 6x2 dan memiliki kerentanan untuk serangan diferensial.

[2] BEAR and LION

BEAR dan LION merupakan konstruksi blok cipher yang didesain oleh Ross Anderson dan Eli Biham, yang dapat digunakan untuk membangun tiga putaran *Unbalanced Feistel Network* heterogen yang tidak konsisten tanpa fungsi hash untuk kunci dengan output n bit, serta cipher stream dengan kunci n-bit. BEAR menggunakan fungsi hash sebagai fungsi F untuk putaran pertama, yang pada umumnya *source-heavy*, dan kemudian menggunakan stream cipher untuk putaran yang kedua, yang merupakan *target heavy*. Putaran yang ketiga adalah aplikasi *source heavy* yang lain dari fungsi hash. LION menggunakan stream cipher untuk putaran pertama dan ketiga, yang *target heavy*. dan fungsi hash untuk putaran yang kedua. Keduanya memiliki properti bahwa untuk fungsi hash dan stream cipher aman, maka hasil yang diperoleh juga aman.

[3] MD4 Family of Hash Function

Struktur dasar untuk fungsi-fungsi hash ini adalah algoritma blok. Davies Meyes memperbaiki fungsi dan mengubahnya ke fungsi hash satu fungsi. Di dalam SHA, misalnya, algoritma blok memiliki 80 putaran, 128:32 *Unbalanced Feistel Network* yang komplit dan genap. MD4 memiliki 48 putaran 96:32 *Unbalanced Feistel Network*, MD5 mempunyai 64 putaran 96:32 *Unbalanced Feistel Network*.

[4] GDES

GDES merupakan varian dari DES di mana hasil keluaran dari fungsi F digabungkan dengan dirinya sendiri beberapa kali dan dilakukan XOR dengan blok target yang lebih besar. Ini didefinisikan untuk parameter yang lebih luas dan merupakan 32:32q *Unbalanced Feistel Network*. Akan tetapi, jumlah putaran yang direkomendasikan terlalu kecil dan cipher yang dihasilkan berpeluang mendapat serangan diferensial.

[5] Khufu / Khafre

Khufu dan Khafre, keduanya merupakan *target heavy Unbalanced Feistel Network* yang tidak komplete, genap, dan heterogen. s=8, t=32, b=8, dan n=64. Satu *cycle* setara dengan sebuah oktet dari terminologi Merkle. Catatan bahwa di Khufu, subblok-sublok digeser sedikit berbeda dengan notasi-notasi ini, dan juga bahwa masing-masing blok sumber putaran diambil dari blok target sebelumnya.

[6] REDOC III

REDOC III merupakan suatu *target-heavy* UFN dengan n= 80, s= 8, dan t= 72. Notasi dalam REDOC III dokumentasi adalah sedikit banyak berbeda dibanding di sini, sebab REDOC III dirancang untuk bergeser di sekitar blok target dan sumber.

[7] Non Linear Feedback Shift Register

Kasus dari suatu *source heavy* UFN yang paling ekstrim (n-1):1 mengenkripsi target satu bit menggunakan sisa dari bit blok sebagai bit sumber. Ini merupakan inti dari Non Linear Feedback Shift Regitster. Kasus paling ekstrim dari *target heavy* UFN 1:(n-1) mengenkripsi seluruhnya kecuali satu bit dari satu blok, menggunakan nilai yang ditentukan oleh sumber bit tunggal. Ini dapat dilihat sebagai NLFSR dalam format Galois. Catatan bahwa jenis konstruksi ini selalu rapuh, sehingga mudah dihancurkan.

Daftar Pustaka

[1] Bruce Schneier, John Kelsey, *Unbalanced Feistel Network and Block-Cipher Design.*

[2] Dr. Andreas Steffen, *Secure Network Communication Part I Introduction to Cryptography*, 2002.