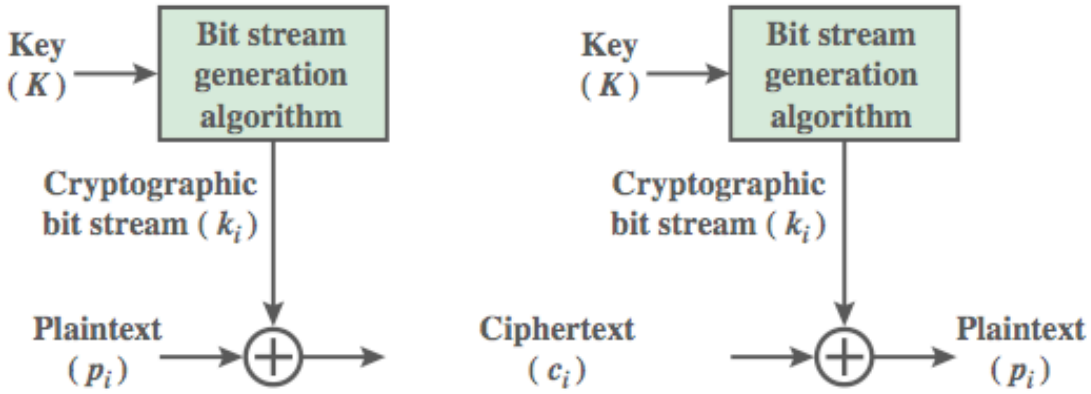
PENYANDIAN BLOCK

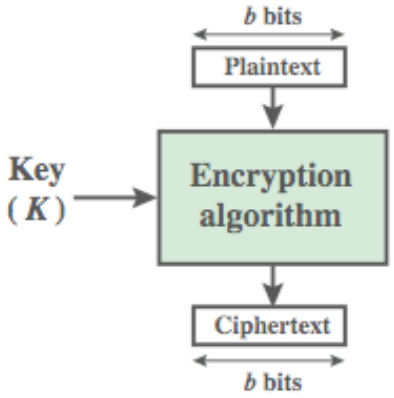
KULIAH KRIPTOGRAFI DAN KEAMANAN JARINGAN

BLOCK CIPHER VS STREAM CIPHER

- **Stream cipher: cipher yang mengenkripsi/mendekripsi aliran data digital satu bit atau satu byte sekali waktu.**
  - Menggunakan bit-stream generator yang menghasilkan bit stream berdasarkan kunci yang digunakan → setiap pengguna harus memperoleh informasi mengenai generating key dan masing-masing dapat menghasilkan keystream.
  - Plaintext di-xor-kan dengan bit stream untuk menghasilkan ciphertext
- **Block cipher: cipher yang mengenkripsi/mendekripsi data dalam blok-blok.**
  - Ukuran: 64 atau 128 bit atau yang lain.
  - Pengguna mendapatkan informasi tentang kunci enkripsi simetri.
- **Penyandian simetris yang ada sekarang lebih banyak menggunakan block cipher:**
  - Lebih mudah dianalisis.
  - Aplikasi lebih luas.



(a) Stream Cipher Using Algorithmic Bit Stream Generator



(b) Block Cipher

Stallings (2011)

MOTIVASI STRUKTUR PENYANDIAN FEISTEL

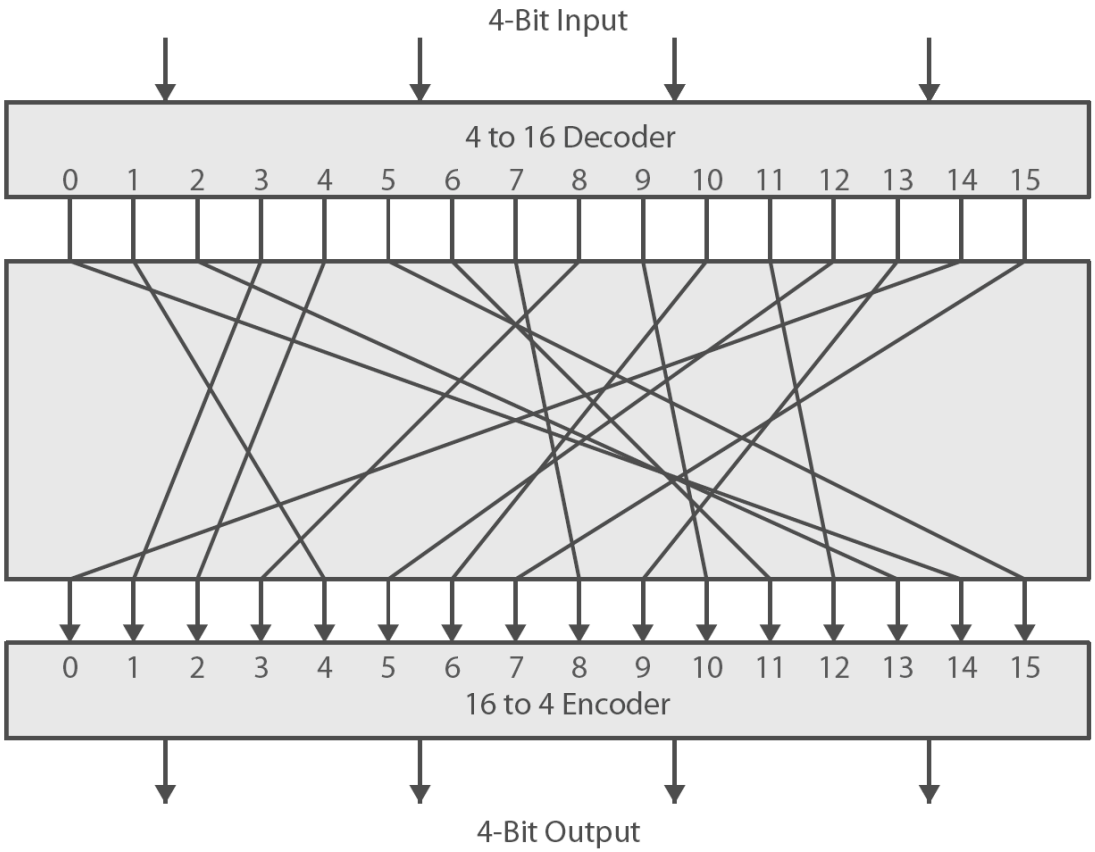
- **Block cipher beroperasi pada blok plaintext n bit untuk mendapatkan ciphertext sepanjang n bit.**
- **Terdapat 2^n blok plaintext yang mungkin.**
- **Agar reversible (bisa didekripsi), masing-masing harus menghasilkan blok ciphertext yang unik → transformasi yang reversible/non-singular.**
- **Jumlah transformasi yang mungkin: $2^n!$**

Plaintext	Ciphertext
00	11
01	10
10	00
11	01

Plaintext	Ciphertext
00	11
01	10
10	01
11	00

PENYANDIAN BLOCK IDEAL

- **Feistel: memungkinkan jumlah maksimum kemungkinan pemetaan enkripsi dari blok plaintext**



MASALAH PENYANDIAN BLOCK IDEAL

- **Jika n kecil, ekuivalen dengan penyandian substitusi klasik → rawan terhadap analisis statistik.**
  - Jika n besar, resiko dapat dihindari.
- **Tidak praktis untuk ukuran blok yang besar.:**
  - Pemetaannya sendiri memuat kunci.
  - Untuk n = 4, ukuran kunci: 4 (bit) X 16 (baris) = 64 bit.
  - Secara umum untuk untuk panjang bit n, diperlukan kunci k sepanjang n x 2^n bit.
  - Jadi untuk n = 64, panjang key ≈ 10^{21} bit.
- **Feistel: pendekatan/aproksimasi terhadap sistem ideal block cipher untuk n besar.**

PENYANDIAN FEISTEL

- **Dipublikasikan pada tahun 1973.**
- **Mendekati ideal block cipher dengan memanfaatkan konsep product cipher: eksekusi 2 atau lebih cipher sederhana secara berurutan, dengan hasil akhir yang lebih kuat secara kriptografis daripada cipher komponennya.**
- **Prinsip: membangun block cipher dengan panjang kunci k bit dan blok dengan panjang n bit yang memungkinkan jumlah transformasi sebanyak 2^k, bukan 2^n!**
- **Menggunakan:**
  - Substitusi: tiap elemen atau kelompok elemen plaintext diganti secara unik oleh elemen atau kelompok elemen cipertext.
  - Permutasi: sederetan elemen dari plaintext diganti dengan permutasi dari deretan tersebut. Tidak ada penambahan atau penghapusan elemen, hanya urutan yang diubah.
- **Merupakan aplikasi praktis dari confusion and diffusion.**

DIFFUSION DAN CONFUSION

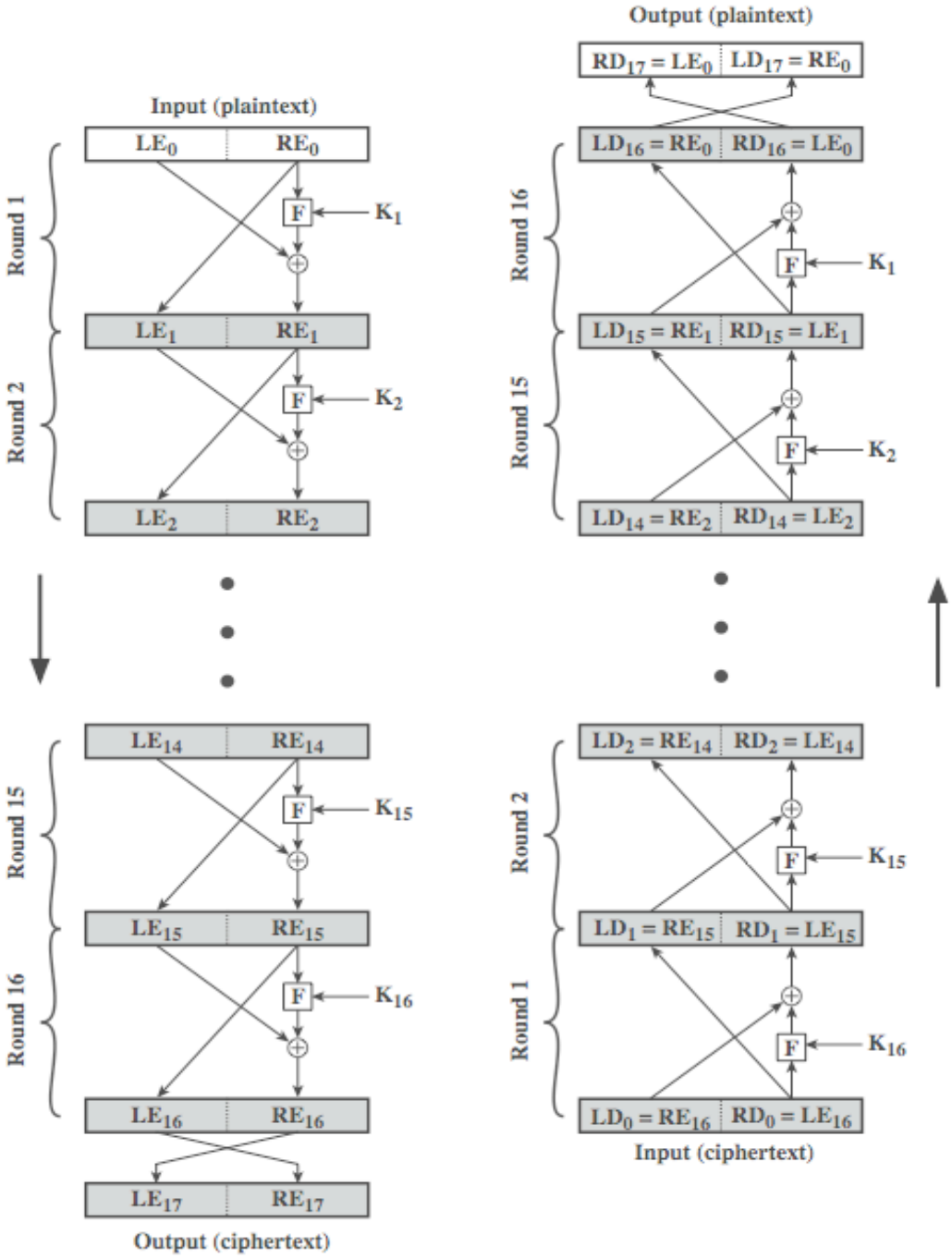
- **Diperkenalkan oleh Claude Shannon (1949).**
- **Tujuan utama: mencegah kriptoanalisis berdasarkan analisis statistik.**
  - Misalkan cryptanalyst mengetahui karakteristik statistik dari plaintext. Jika ciphertext merefleksikan karakteristik tersebut, cryptanalyst mungkin bisa mendapatkan kunci enkripsi atau sebagian kunci.
- **Metode yang dipakai: diffusion (difusi) dan confusion.**
- **Diffusion:**
  - Struktur statistik plaintext dihilangkan dengan ciphertext yang memiliki statistik dengan jangkauan besar.
  - Setiap digit plaintext mempengaruhi nilai dari digit ciphertext, atau setiap digit ciphertext dipengaruhi oleh banyak digit plaintext.

- Contoh: untuk mengenkripsi pesan M = m_1, m_2, …, digunakan rumus: $y_n = (\Sigma_i \ m_{n+i}) \bmod 26$, i dari 1 sampai k.
  - Frekuensi huruf menjadi lebih seragam.
- Dalam blok cipher biner, dapat dilakukan dengan berulangkali melakukan permutasi pada data, kemudian mengaplikasikan fungsi pada permutasi tersebut → beberapa bit dari posisi yang berbeda berkontribusi terhadap 1 bit ciphertext.

- **Confusion:**
  - Membuat agar hubungan antara statistik dari ciphertext dan nilai dari kunci enkripsi menjadi sekompleks mungkin.
  - Meskipun attacker mendapatkan statistik dari ciphertext, dia tidak dapat menggunakannya untuk mendapatkan kunci.
  - Menggunakan algoritma substitusi yang kompleks.

STRUKTUR FEISTEL CIPHER

- **Input: blok plaintext dengan ukuran 2w bit dan key K.**
- **Blok plaintext dibagi 2: L_0 dan R_0, masing-masing melalui n tahap pemrosesan, kemudian digabung untuk mendapatkan blok ciphertext.**
- **Tiap tahap (round) ke-i mendapatkan input L_{i-1} dan R_{i-1}, dan sub-key K_i yang diturunkan dari K.**
- **Substitusi terhadap bagian kiri data dilakukan pada setiap tahap.**
- **Fungsi F mengolah blok sebelah kanan (w bit) dan sub-key (y bit), dapat ditulis: $F(RE_{i,}, K_{i+1})$.**
- **Setelah tahap terakhir, data kiri dan kanan dipertukarkan.**



- **Proses dekripsi:**
  - Sama dengan enkripsi.
  - Aturan: gunakan ciphertext sebagai input, tetapi K_i digunakan dengan urutan terbalik.

PENGGUNAAN FEISTEL CIPHER

- **Bergantung pada:**
  - Ukuran blok:
    - Makin besar maka makin aman, tetapi mengurangi kecepatan enkripsi/dekripsi.
    - Ukuran 64 bit sudah dianggap aman.
  - Ukuran kunci:
    - Makin besar maka makin aman, tetapi mengurangi kecepatan enkripsi/dekripsi.
    - Ukuran 64 bit sudah tidak aman, yang umum 128 bit.
  - Jumlah tahapan:
    - Satu kali tidak aman, tetapi dengan berkali-kali menjadi lebih aman.
    - Biasanya 16 kali.
  - Algoritma pembangkit sub-key: Kompleksitas yang tinggi akan menyulitkan cyriptanalyst.
  - Fungsi F: Kompleksitas yang tinggi akan menyulitkan cyriptanalyst.

PRINSIP DESAIN BLOK CIPHER

- **Tiga aspek penting:**
  - Jumlah tahap
  - Fungsi F
  - Key scheduling algorithm
- **Jumlah tahap:**
  - Semakin banyak jumlah tahap, semakin sulit untuk dilakukan kriptoanalisis, meski F tidak terlalu kuat.
- **Desain fungsi F:**
  - Semakin tidak linier semakin baik.
  - Properti avalanche yang baik
    - Avalanche effect: perubahan pada 1 bit dari plaintext harus mengakibatkan perubahan pada beberapa bit ciphertext.
  - Bit independence criterion (BIC): output pada bit ke-j dan k harus berubah secara independen jika satu bit tunggal ke-i diubah.

- **Key scheduling algorithm:**
  - Dipilih sub-keys agar tidak dapat dilakukan deduksi terhadap masing-masing sub-key dan key utama.
  - Belum ada prinsip umum.