Stream Cipher

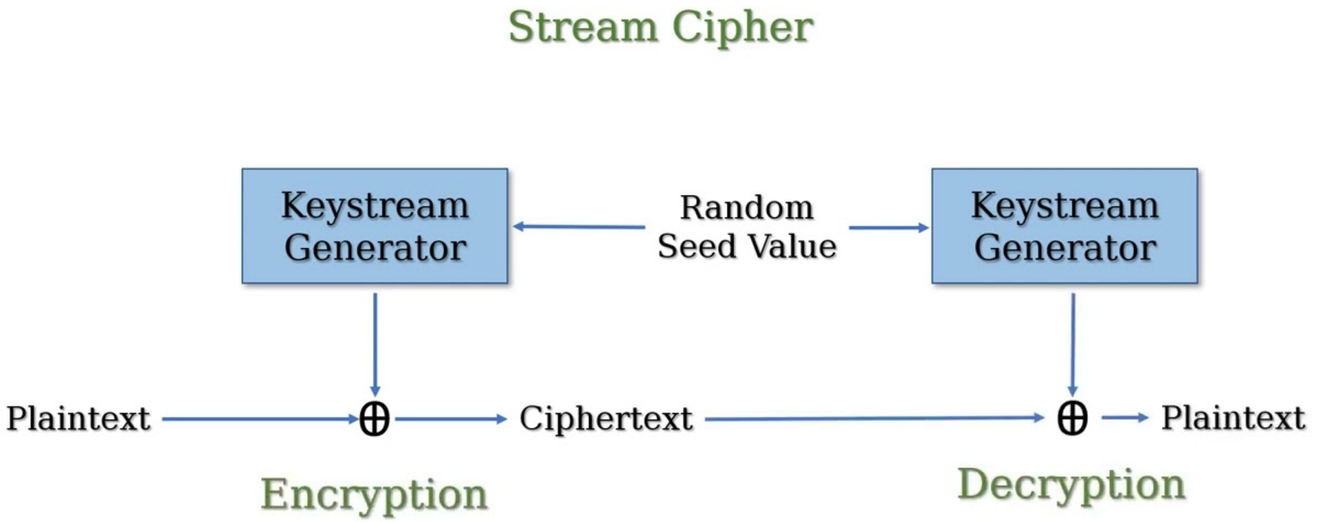
Bahan Kuliah Keamanan Data

Sevi **Nurafni**

Fakultas Sains dan Teknologi
Universitas Koperasi Indonesia 2025

- Mengenkripsi plainteks menjadi cipherteks setiap bit per bit dengan bit-bit kunci (keystream) atau byte per byte
- Diperkenalkan oleh Vernam melalui algoritma Vernam Cipher
- Vernam cipher diadopsi dari one-time pad cipher, yang dalam hal ini diganti dengan bit (0 atau 1)

Diagram Cipher Stream



The Security Buddy
https://www.thesecuritybuddy.com/

- Keystream dibangkitkan oleh keystream generator
- Keystream di-XOR-kan dengan bit-bit plainteks menghasilkan aliran bit-bit cipherteks:

$$c_i = p_i \oplus k_i$$

- Di sisi penerima dibangkitkan keystream yang sama untuk mendekripsi aliran bit-bit cipherteks:

$$p_i = c_i \oplus k_i$$

Contoh:

Plainteks:	1100101010100110001	
Keystream:	1000110000101001101 ⊕	Enkripsi
Cipherteks:	0100011010001111100	
Keystream:	1000110000101001101 ⊕	Dekripsi
Plainteks:	1100101010100110001	

- Keamanan stream cipher bergantung seluruhnya pada keystream generator.
- Tinjau 3 kasus yang dihasilkan oleh keystream generator:
  1. Keystream seluruhnya 0
  2. Keystream berulang secara periodik
  3. Keystream benar-benar acak

Kasus 1: jika pembangkit mengeluarkan keystream yang seluruhnya nol

1. Keystream: 0000000000000000…

- Maka cipherteks = plainteks
- Sebab $c_i = p_i \oplus 0 = p_i$
- Proses enkripsi menjadi tidak berarti

Kasus 2: jika pembangkit mengeluarkan keystream yang berulang secara periodik

2. Keystream: 1101101101101101101101101…

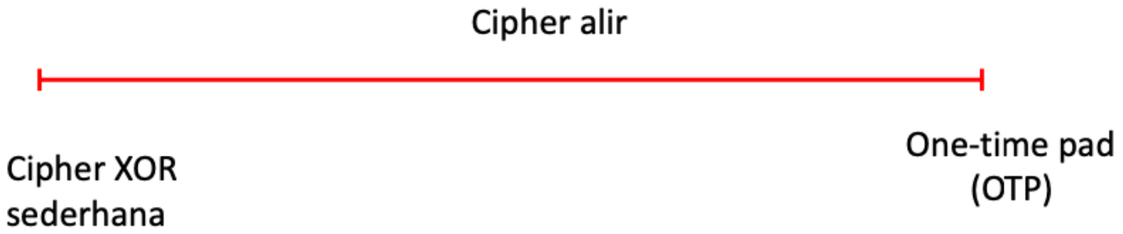
- Maka algoritma enkripsinya = cipher XOR sederhana yang memiliki tingkat keamanan yang rendah

Kasus 3: jika pembangkit mengeluarkan keystream benar-benar acak

3. Keystream: 01101010010101110011010110010…

- Maka algoritma enkripsinya = one-time pad dengan tingkat keamanan yang sempurna
- Pada kasus ini, panjang keystream = panjang plainteks, dan kita mendapatkan stream cipher sebagai unbreakable cipher

- Kesimpulan: tingkat keamanan stream cipher terletak antara cipher XOR sederhana dengan one-time pad.



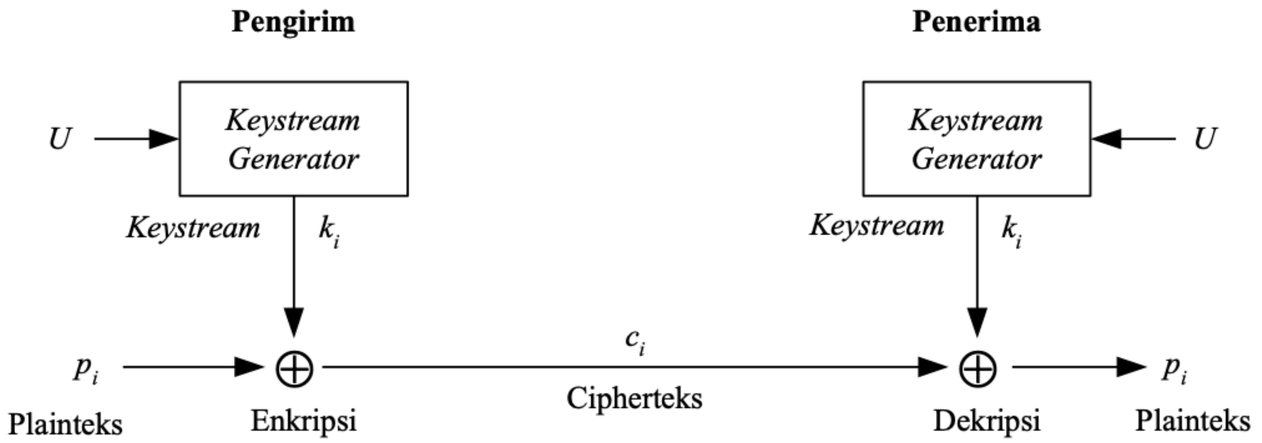
- Semakin acak keluaran yang dihasilkan oleh pembangkit keystream, semakin sulit kriptanalis memecahkan cipherteks

Keystream Generator

- Keystream Generator diimplementasikan sebagai prosedur yang sama di sisi pengirim dan penerima
- Keystream generator dapat membangkitkan keystream berbasis bit per bit atau dalam bentuk blok-blok bit
- Jika keystream berbentuk blok-blok bit, cipher blok dapat digunakan untuk memperoleh stream cipher

Keystream Generator

- Keystream Generator menerima masukan sebagai kunci U. Luaran dari prosedur merupakan fungsi dari U. Pengirim dan penerima harus memiliki kunci U yang sama. Kunci U ini harus dijaga kerahasiannya.
- Keystream generator menghasilkan bit-bit kunci yang di-XOR-kan dengan bit plair



Keystream Generator

- Contoh: $U = 1111$

  (U adalah kunci 4-bit yang dipilih sembarang, kecuali 0000)

  Algoritma sederhana memperoleh keystream:

  XOR-kan bit ke-1 dan bit ke-4 dari empat bit sebelumnya:

  111101011001000

  Dan akan berulang setiap 15 bit

- Secara umum, jika panjang kunci U adalah n bit, maka bit-bit kunci tidak akan berulang sampai $2^n - 1$ bit

Serangan pada Stream Cipher

Known-plaintext attack

- Kriptanalis mengetahui potongan P dan C yang berkoresponden.
- Hasil: K untuk potongan P tersebut karena

$$
\begin{aligned}
P \oplus C &= P \oplus (P \oplus K) \\
&= (P \oplus P) \oplus K \\
&= 0 \oplus K \\
&= K
\end{aligned}
$$

Contoh

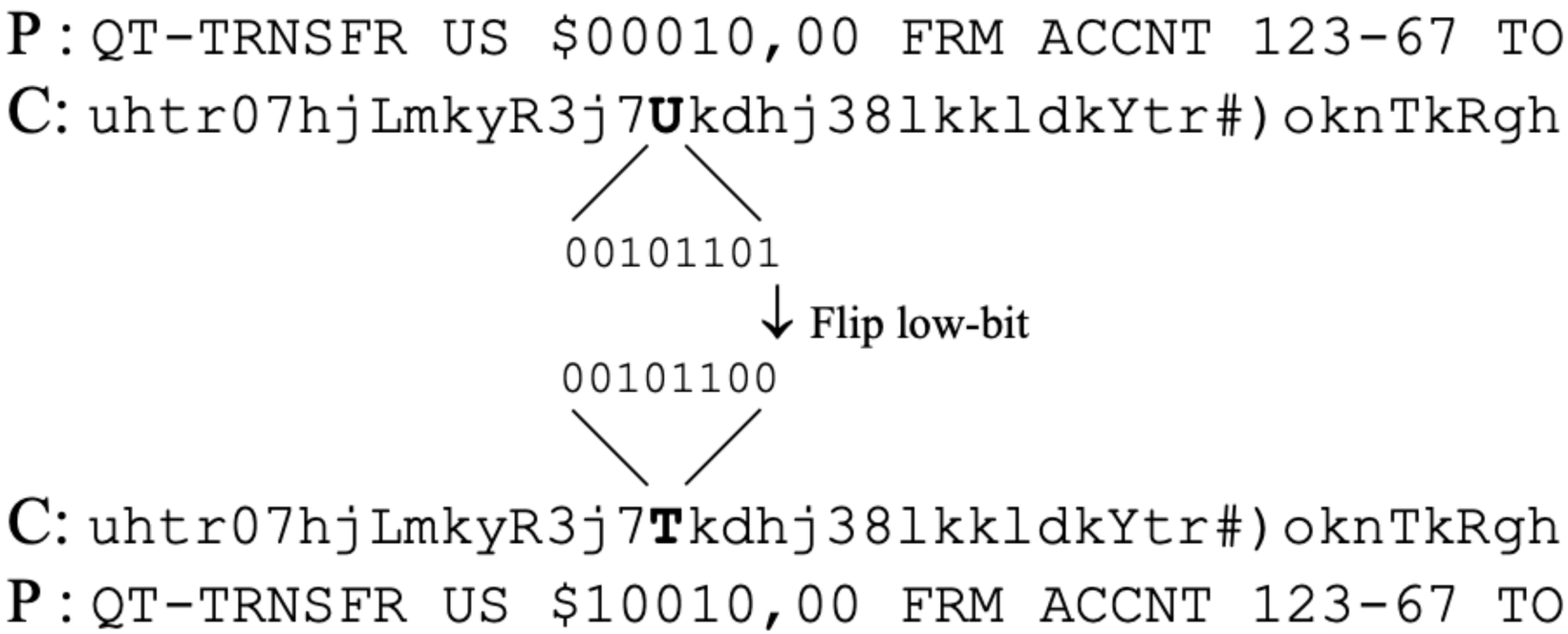
P	01100101		(karakter ‘e’)
K	00110101	$\oplus$	(karakter ‘5’)
C	01010000		(karakter ‘P’)
P	01100101	$\oplus$	(karakter ‘e’)
K	00110101		(karakter ‘5’)

Flip-bit attack

- Tujuan: mengubah bit cipherteks tertentu sehingga hasil deskripsinya berubah.

- Pengubahan dilakukan dengan membalikkan (flip) bit tertentu (0 menjadi 1 atau 1 menjadi 0)

Contoh



- Pengubahan 1 bit U dari cipherteks sehingga menjadi T.
- Hasil dekripsi: $10,00 menjadi $10010,00

Flip-bit attack

- Pengubah pesan tidak perlu mengetahui kunci, hanya perlu mengetahui posisi pesan yang diminati saja.

- Serangan semacam ini memnfaatkan karakteristik stream cipher yang sudah disebutkan di atas, bahwa kesalahan 1-bit pada cipherteks hanya menghasilkan kesalahan 1-bit pada plainteks hasil dekripsi.

Aplikasi *Stream Cipher*

- Stream cipher cocok untuk mengenkripsi aliran data yang terus menerus melalui saluran komunikasi, misalnya
- Mengenkripsi data pada saluran yang menghubungkan antara dua kompute
- Mengenkripsi suara pada jaringan telepon mobile GSM

- Alasan: jika bit cipherteks yang diterima mengandung kesalaham, maka hal ini hanya menghasilkan 1 bit kesalah pada waktu dekripsi, karena tiap bit plainteks ditentukan hanya 1 bit cipherteks

SELAMAT BELAJAR