RC4 dan A5

Bahan Kuliah IF5054 Kriptografi

1

RC4

- Termasuk ke dalam *cipher* aliran (*stream cipher*)
- Dibuat oleh Ron Rivest (1987) dari Laboratorium *RSA*
- *RC* adalah singkatan dari *Ron's Code*). Versi lain megatakan *Rivest Cipher* .
- Digunakan sistem keamanan seperti:
  - protokol *SSL* (*Secure Socket Layer*).
  - WEP (Wired Equivalent Privacy)
  - WPA (Wi-fi Protect Access) untuk nirkabel

2

- RC4 awalnya rahasia
- Pada September 1994, RC4 dikirim secara anonim ke milis *Cypherpunks*
- Lalu dikirim ke *newsgroup* `sci.crypt` dan menyebar di internet
- Karena telah diketahui orang, RC4 bukan lagi rahasia dagang
- Status sekarang, implementasi tidak resmi adalah legal, tapi tidak boleh menggunakan nama RC4. Maka digunakan nama ARCFOUR untuk menghindari masalah *trademark*.

3

- *RC4* membangkitkan aliran kunci (*keystream*) yang kemudian di-XOR-kan dengan plainteks
- *RC4* memproses data dalam ukuran *byte*, bukan dalam bit.
- Untuk membangkitkan aliran kunci, *cipher* menggunakan status internal yang terdiri dari:
  - Permutasi angka 0 sampai 255 di dalam larik S_0, S_1, ..., S_{255}. Permutasi merupakan fungsi dari kunci U dengan panjang variabel.
  - Dua buah pencacah indeks, i dan j

4

Algoritma RC4:

1. Inisialisasi larik S: $S_0 = 0$, $S_1 = 1$, ..., $S_{255} = 255$

```
for i ← 0 to 255 do
  S[i] ← i
endfor
```

2. Jika panjang kunci $U < 256$, lakukan *padding* sehingga panjang kunci menjadi 256 *byte*.
   Contoh: U = "abc" (3 *byte*)
   Padding: U = "abcabcabc..." sampai panjang U mencapai 256 *byte*

5

3. Lakukan permutasi nilai-nilai di dalam larik S:

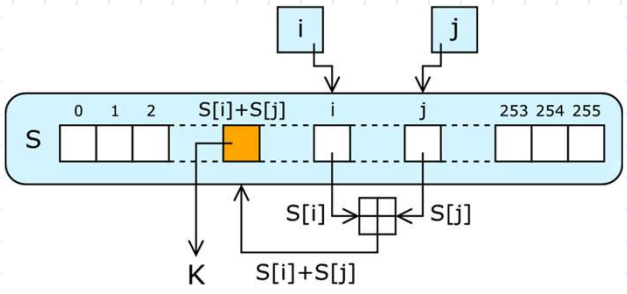
```
j ← 0
for i ← 0 to 255 do
   j ← (j + S[i] + U[i]) mod 256
   swap(S[i], S[j])
endfor
```

6

4. Bangkitkan aliran-kunci dan lakukan enkripsi:

```
i ← 0
j ← 0
for idx ← 0 to PanjangPlainteks – 1 do
   i ← (i + 1) mod 256
   j ← (j + S[i]) mod 256
   swap(S[i], S[j])
   t ← (S[i] + S[j]) mod 256
   K ← S[t]          (* keystream *)
   c ← K ⊕ P[idx]
endfor
```

7



8

- Sampai saat ini tidak ada yang dapat memecahkan RC4 sehinggat dapat dikatakan sangat kuat.
- Terdapat laporan versi kunci 40 bit dapat dipecahkan secara brute force.
- Kelemahan: *Padding* dapat menyebabkan kemungkinan nilai-nilai di dalam larik S ada yang sama.
- *RC4* juga mudah diserang dengan *known-plaintext attack*, dengan cara meng-XOR-kan dua set *byte* cipherteks (kelemahan umum pada *cipher*-aliran)

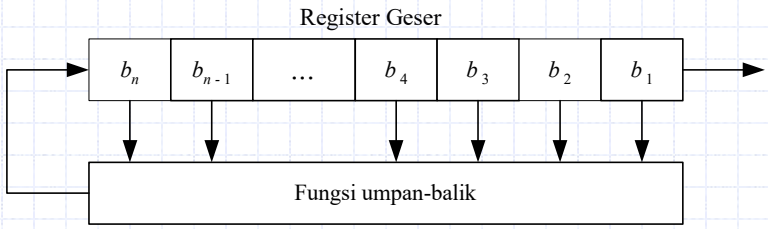
9

Linear Feedback Shift Register (LFSR)

- Banyak digunakan sebagai pembangkit aluran-kunci pada *cipher* aliran.
- Mangkus jika diimplementasikan sebagai *hardware*, lambat jika dalam bentuk *software*.

10

- Register geser umpan-balik (*feedback shift register*) atau *FSR* terdiri dari dua bagian:

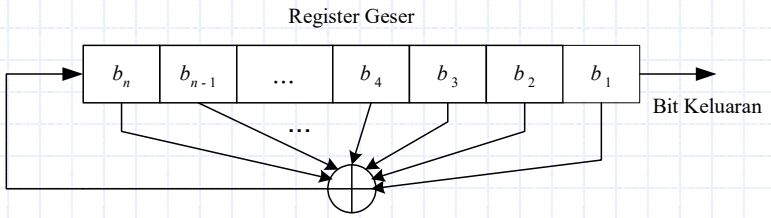


11

- Cara kerja:
  - Tiap kali sebuah bit dibutuhkan, semua bit di dalam register digeser 1 bit ke kanan
  - Bit paling kiri (b_n) dihitung sebagai fungsi bit-bit lain di dalam register tersebut.
  - Keluaran dari register geser adalah 1 bit (yaitu bit b_1 yang tergeser)

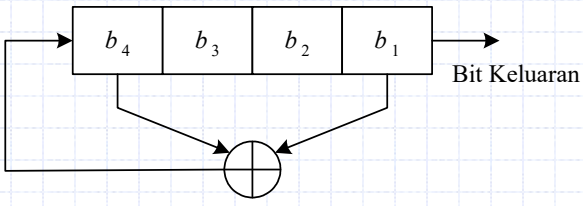
12

LFSR *n*-bit:



13

Contoh LFSR 4-bit:



$$b_4 = f(b_1, b_4) = b_1 \oplus b_4$$

14

i	Isi Register	Bit Keluaran
0	1 1 1 1	
1	0 1 1 1	1
2	1 0 1 1	1
3	0 1 0 1	1
4	1 0 1 0	1
5	1 1 0 1	0
6	0 1 1 0	1
7	0 0 1 1	0
8	1 0 0 1	1
9	0 1 0 0	1
10	0 0 1 0	0
11	0 0 0 1	0
12	1 0 0 0	1
13	1 1 0 0	0
14	1 1 1 0	0

15

A5

- *A5* : *cipher* aliran yang digunakan untuk mengenkripsi transmisi sinyal percakapan dari standard telepon seluler *GSM* (*Group Special Mobile*).
- Sinyal *GSM* dikirim sebagai barisan *frame*. Satu *frame* panjangnya 228 bit dan dikirim setiap 4,6 milidetik.
- *A5* digunakan untuk untuk menghasilkan aliran-kunci 228-bit yang kemudian di-*XOR*-kan dengan *frame*. Kunci eksternal panjangnya 64 bit.

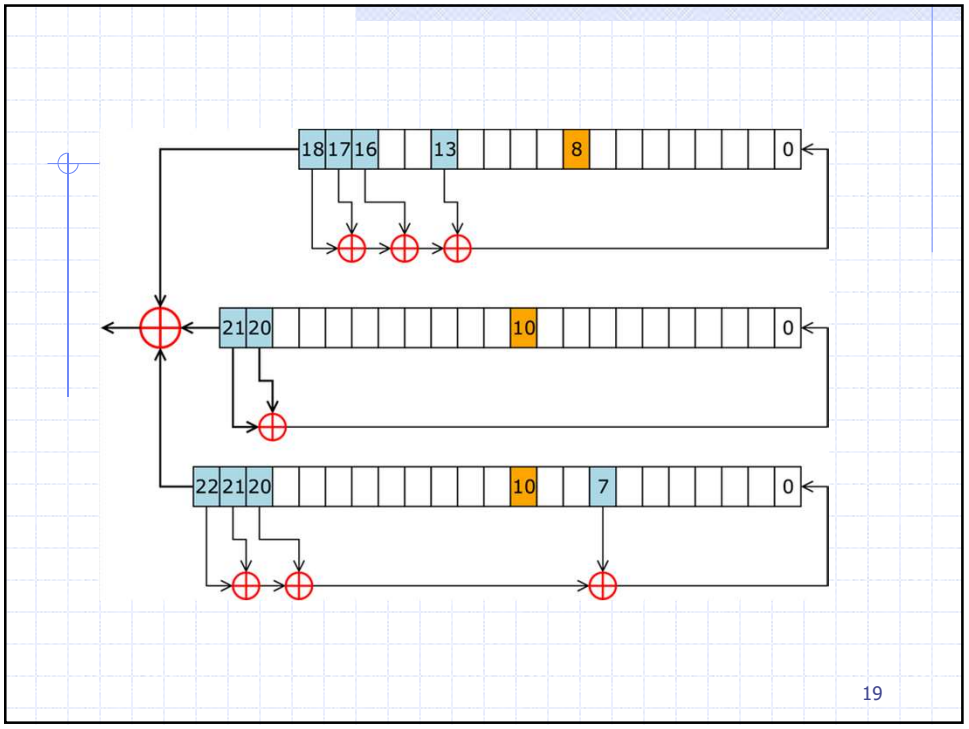
16

- GSM merupakan standard telepon seluler Eropa
- A5 Dibuat oleh Perancis
- Tidak semua operator GSM mengimplementasikan A5 (seperti di Indonesia)
- A5 ada dua versi:
  1. A5/1 : versi kuat A5, digunakan di Eropa
  2. A5/2 : versi ekspor, lebih lemah
- Algoritma A5/1 pada awalnya rahasia, tetapi pada tahun 1994 melalui *reverse engineering*, algoritmanya terbongkar.

17

- *A5* terdiri dari 3 buah *LFSR* , masing-masing panjangnya 19, 22, dan 23 bit (total = 19 + 22 + 23 = 64).
- Bit-bit di dalam register diindeks dimana bit paling tidak penting (*LSB*) diindeks dengan 0 (elemen paling kanan).
- Luaran (*output*) dari *A5* adalah hasil *XOR* dari ketiga buah *LFSR* ini.
- *A5* menggunakan tiga buah kendali detak (*clock*) yang variabel

18



19

- Register diinisialiasasi dengan kunci sesi (64 bit).
- Tiap register didetak (*clock*) berdasarkan bit pertengahannya (masing-masing: 8, 10, dan 10).
- Setiap register mempunyai bit pendetakan yang berbeda-beda.
- Pendetakan bergantung kesamaan bit tengah dengan mayorita bit-bit pendetakan.
- *Cipher* menghasilkan *keytream* yang panjangnya 228 bit untuk kemudian dienkripsi dengan meng-XOR-kan nya dengan setiap *frame*.

20