IF4020 Kriptografi

Kriptografi Modern

Oleh: Rinaldi Munir

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2023

Pendahuluan

- Kriptografi modern adalah era kriptografi setelah penemuan komputer digital.
- Perkembangan teknologi komputer digital membuat ilmu kriptografi berkembang dengan pesat.
- Komputer digital merepresentasikan data dan informasi dalam biner.
- Algoritma kriptografi modern beroperasi dalam mode bit atau *byte* (bandingkan dengan algoritma kriptografi klasik beroperasi dalam mode karakter)

  → kunci, plainteks, cipherteks, diproses dalam rangkaian bit/byte

  → operasi **xor** paling banyak digunakan di dalam algoritmanya

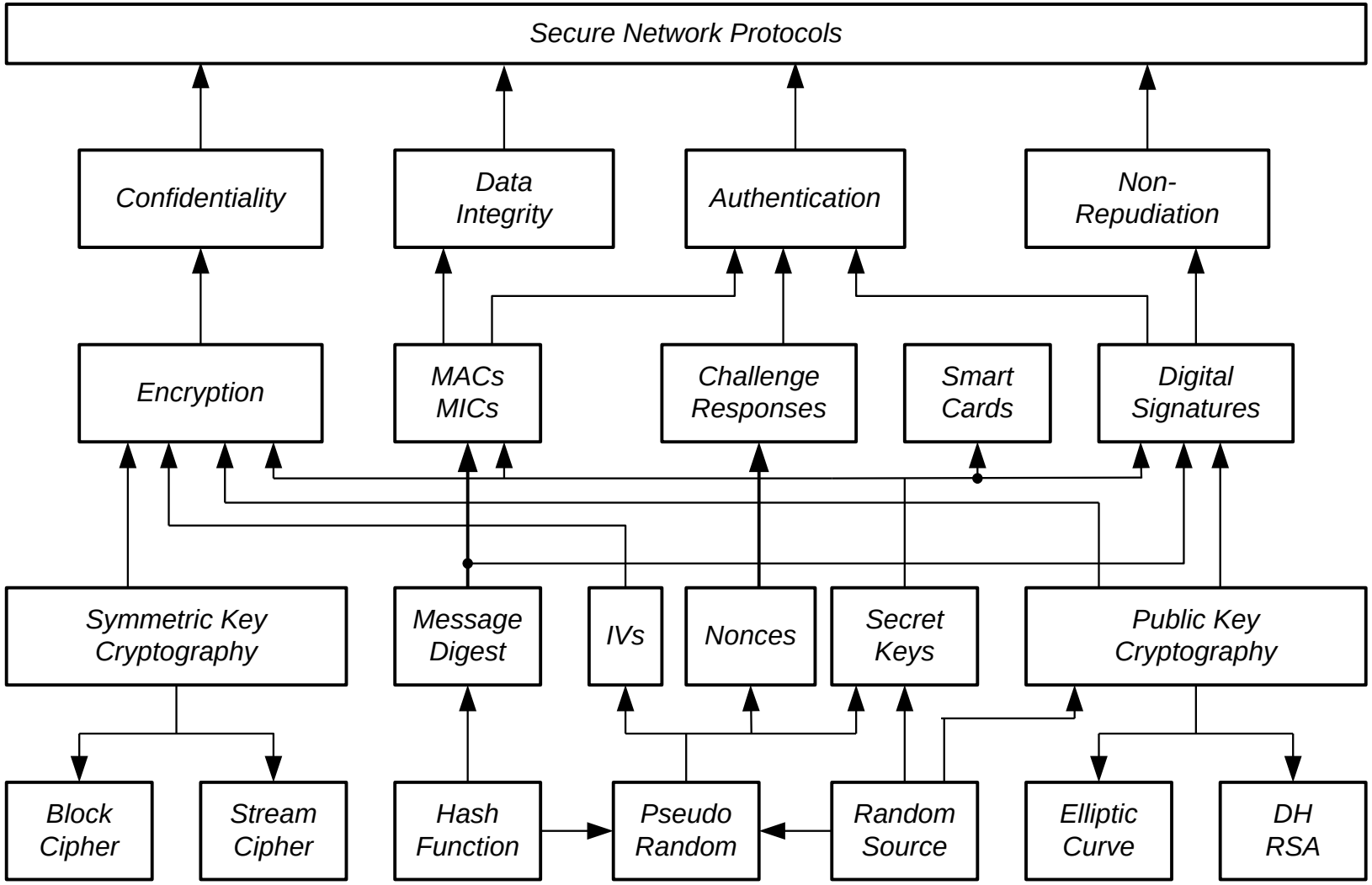
- Meskipun disebut kriptografi modern, namun algoritmanya tetap menggunakan dua teknik dasar dasar di dalam kriptografi klasik: **teknik substitusi** dan **teknik transposisi**,

- tetapi operasinya dibuat lebih kompleks, tidak sesederhana cipher klasik.

  Tujuannya: agar *cipher* modern lebih sulit dikriptanalisis
- Selain kedua teknik dasar tersebut, juga digunakan teknik lain seperti rotasi, kompresi, ekspansi, penjumlahan modulo, dan lain-lain.

- Kriptografi modern melahirkan konsep-konsep baru seperti algoritma kriptografi kunci-publik, fungsi *hash*, protokol kriptografi, tanda-tangan digital, pembangkit bilangan acak, skema pembagian kunci, dsb.

Diagram Blok Kriptografi Modern



Bit, Byte, dan Kode Heksadesimal

- Pesan di dalam *cipher* modern dienkripsi bit-per-bit atau byte-per-byte, atau dalam kelompok bit (byte).

  1 byte = 8 bit

- Pada beberapa algoritma kriptografi, pesan direpresentasikan dalam kode heksadesimal (Hex).

  1 kode hex = 4 bit

0000 = 0	0001 = 1	0010 = 2	0011 = 3
0100 = 4	0101 = 5	0110 = 6	0111 = 7
1000 = 8	1001 = 9	1010 = A	1011 = B
1100 = C	1101 = D	1110 = E	1111 = F

- Contoh: Pesan 100111010110 dalam kode Hex dengan cara membagi pesan menjadi blok 4-bit:

  1001 1101 0110 = 9D6

- Jika pesan diproses dalam kelompok bit, maka rangkaian bit pesan dibagi menjadi blok-blok bit berukuran sama.

- Contoh: Plainteks 10011101011000101111100001

  Bila dibagi menjadi blok 8-bit

  10011101 01100010 11100001

  atau dalam kode heksadesimal menjadi :

  9E 62 E1

- *Padding bits*: bit-bit tambahan jika ukuran blok terakhir tidak mencukupi panjang blok

- Contoh: Plainteks 100111010110

  Bila dibagi menjadi blok 5-bit:

  10011 10101 **000**10

Padding bits mengakibatkan ukuran cipherteks sedikit lebih besar daripada ukuran plainteks semula.

Operasi *XOR*

- Paling banyak digunakan di dalam *cipher* modern
- Notasi: $\oplus$
- Operasi:

$0 \oplus 0 = 0$ $\quad\quad$ $0 \oplus 1 = 1$

$1 \oplus 0 = 1$ $\quad\quad$ $1 \oplus 1 = 0$

- Operasi XOR = penjumlahan dalam modulus 2:

$0 \oplus 0 = 0 \Leftrightarrow 0 + 0 \pmod 2 = 0$

$0 \oplus 1 = 1 \Leftrightarrow 0 + 1 \pmod 2 = 1$

$1 \oplus 0 = 1 \Leftrightarrow 1 + 0 \pmod 2 = 1$

$1 \oplus 1 = 0 \Leftrightarrow 1 + 1 \pmod 2 = 0$

- Sifat-sifat operasi XOR:

  (i) $a \oplus a = 0$

  (ii) $a \oplus b = b \oplus a$

  (iii) $a \oplus (b \oplus c) = (a \oplus b) \oplus c$

Contoh:

(i) $1 \oplus 1 = 0$

(ii) $1 \oplus 0 = 0 \oplus 1 = 1$

(iii) $1 \oplus (0 \oplus 1) = (1 \oplus 0) \oplus 1 = 0$

Operasi XOR *Bitwise*

- Jika dua rangkaian dioperasikan dengan *XOR*, maka operasinya dilakukan dengan meng-*XOR*-kan setiap bit yang berkoresponden dari kedua rangkaian bit tersebut.

  Contoh: 10011 ⊕ 11001 = 01010
  yang dalam hal ini, hasilnya diperoleh sebagai berikut:

$$
\begin{array}{ccccccccccc}
1 & & 0 & & 0 & & 1 & & 1 & & \\
1 & & 1 & & 0 & & 0 & & 1 & \oplus & \\
\hline
1 \oplus 1 & & 0 \oplus 1 & & 0 \oplus 0 & & 1 \oplus 0 & & 1 \oplus 1 & & \\
0 & & 1 & & 0 & & 1 & & 0 & &
\end{array}
$$

Cipher Sederhana dengan operasi XOR

- Sama seperti *Vigenere Cipher*, tetapi dalam mode bit
- Setiap bit plainteks di-*XOR*-kan dengan setiap bit kunci.

Enkripsi: $C = P \oplus K$

Dekripsi: $P = C \oplus K$

Contoh:	plainteks	01100101		(karakter ‘e’)
	kunci	00110101	⊕	(karakter ‘5’)
	cipherteks	01010000		(karakter ‘P’)
	kunci	00110101	⊕	(karakter ‘5’)
	plainteks	01100101		(karakter ‘e’)

- Jika panjang bit-bit kunci lebih pendek daripada panjang bit-bit pesan, maka bit-bit kunci diulang penggunaannya secara periodik (seperti halnya pada Vigenere Cipher)

- Contoh:

  Plainteks : 10010010101110101010001110001

  Kunci : 11011011011011011011011011011

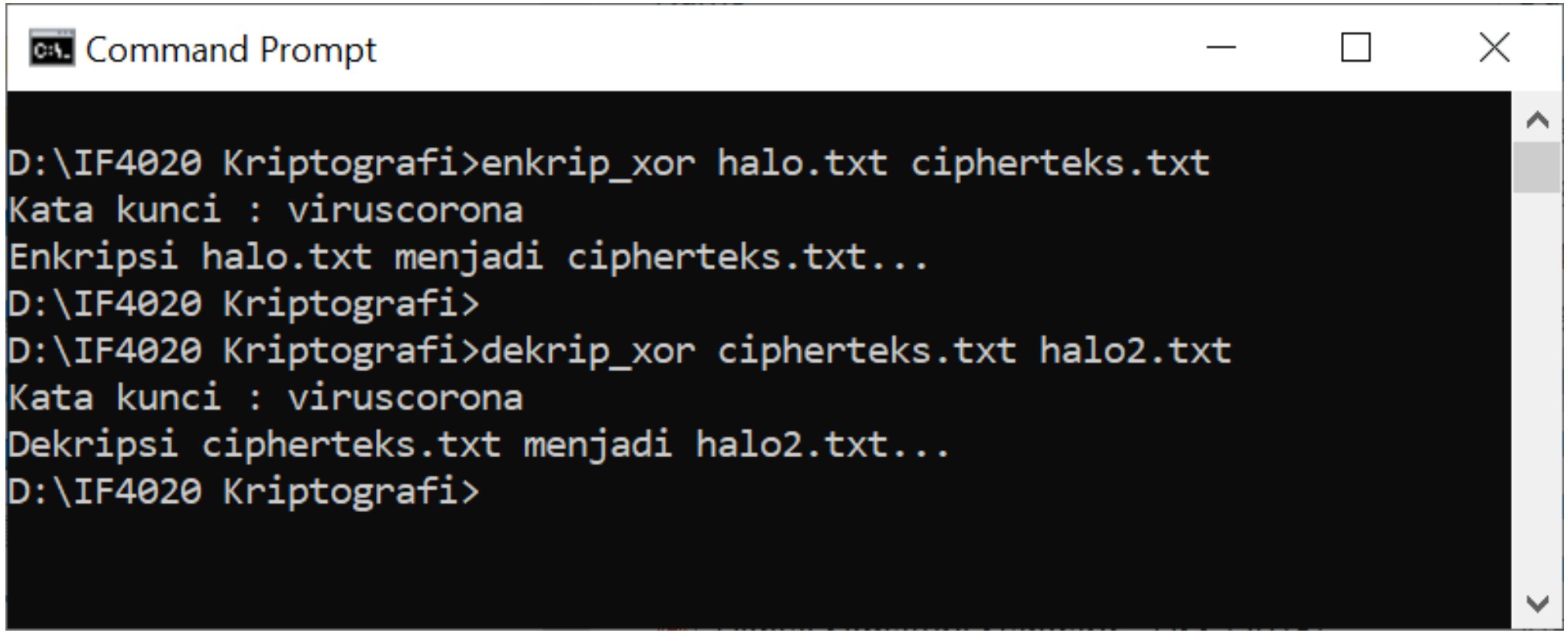
  Cipherteks: 01001001110101110001010101010

```
// Enkripsi sembarang berkas dengan
// algoritma XOR sederhana.
#include <iostream>
#include <string.h>
#include <fstream>
#include <stdlib.h>
using namespace std;

main(int argc, char *argv[])
{
 FILE *Fin, *Fout;
 char p, c;
 string K;
 int i;

 Fin = fopen(argv[1], "rb");
 if (Fin == NULL) {
   cout << "Berkas " << argv[1] <<"
tidak ada" << endl;
   exit(0);
 }

 Fout = fopen(argv[2], "wb");

 cout << "Kata kunci : "; cin >> K;
 cout <<"Enkripsi " << argv[1] << "
menjadi " << argv[2] << "...";
 i = 0;
 while (!feof(Fin)) {
   p = getc(Fin);
   c = p ^ K[i];  // operasi XOR
   putc(c, Fout);
   i = (i + 1) % K.length();
 }
 fclose(Fin);
 fclose(Fout);
}
```

(a) enkrip_xor.cpp

```
// Dekripsi sembarang berkas dengan
// algoritma XOR sederhana.
#include <iostream>
#include <string.h>
#include <stdlib.h>
#include <fstream>
using namespace std;

main(int argc, char *argv[])
{
 FILE *Fin, *Fout;
 char p, c;
 string K;
 int i;

 Fin = fopen(argv[1], "rb");
 if (Fin == NULL){
   cout << "Berkas " << argv[1] <<"
tidak ada" << endl;
   exit(0);
 }

 Fout = fopen(argv[2], "wb");

 cout << "Kata kunci : "; cin >> K;
 cout <<"Dekripsi " << argv[1] << "
menjadi " << argv[2] << "...";
 i = 0;
 while (!feof(Fin)) {
   c = getc(Fin);
   p = c ^ K[i];  // operasi XOR
   putc(p, Fout);
   i = (i + 1) % K.length();
 }
 fclose(Fin);
 fclose(Fout);
}
```

(b) dekrip_xor.cpp



- Cipher sederhana dengan XOR tidak aman, karena mudah dikriptanalisis dengan metode yang sama seperti metode Kasiski

Hasil *running* program cipher XOR sederhana:

Plainteks	Cipherteks *)
Pada wisuda sarjana baru, ternyata ada seorang wisudawan yang paling muda. Umurnya baru 21 tahun. Ini berarti dia masuk ITB pada umur 17 tahun. Zaman sekarang banyak sarjana masih berusia muda belia.	7 S S H IS A o S G H H KS= b EAYA FA. E S A G(:'y N - GPYE @ES2 E H b A H A S K

*) Beberapa karakter ASCII *unprintable*, sehingga tidak dapat dicetak

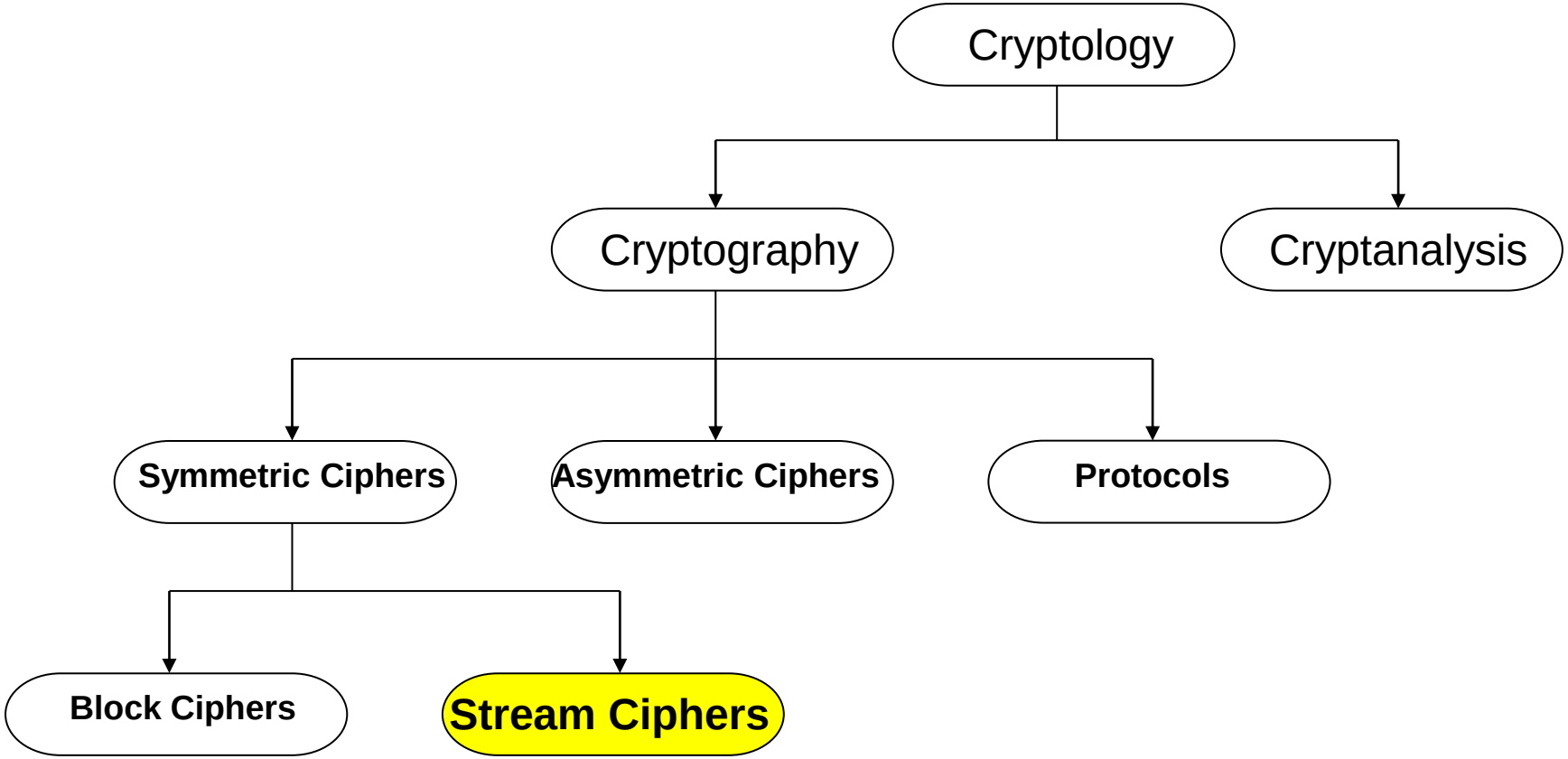
Kategori *cipher* berbasis bit

1. *Cipher* Alir (*Stream Cipher*)

   - beroperasi pada bit tunggal atau *byte* tunggal
   - enkripsi/dekripsi pesan secara bit per bit atau *byte* per *byte*

2. *Cipher* Blok (*Block Cipher*)

   - beroperasi pada blok bit atau blok *byte*
     (contoh: 64-bit/blok = 8 karakter/blok)
   - enkripsi/dekripsi pesan secara blok per blok bit atau blok per blok byte


Cryptology
Cryptography
Cryptanalysis
Symmetric Ciphers
Asymmetric Ciphers
Protocols
Block Ciphers
Stream Ciphers