#11

Syahrul Imardi, MT

STMIK Amik Riau

MATAKULIAH

KEAMANAN PERANGKAT LUNAK

KRIPTOGRAFI MODERN (Lanjutan)

P11

STMIK
Amik Riau

MATAKULIAH
KEAMANAN PERANGKAT LUNAK

Syahrul Imardi, MT

P11 : Kriptografi Modern

Pendahuluan

- Beroperasi dalam mode bit (algoritma kriptografi klasik beroperasi dalam mode karakter)
- kunci, plainteks, cipherteks, diproses dalam rangkaian bit
- operasi bit xor paling banyak digunakan

Pendahuluan

- Tetap menggunakan gagasan pada algoritma klasik: substitusi dan transposisi, tetapi lebih rumit (sangat sulit dipecahkan)
- Perkembangan algoritma kriptografi modern didorong oleh penggunaan komputer digital untuk keamanan pesan.
- Komputer digital merepresentasikan data dalam biner.

Algoritma Enkripsi dengan rangkaian bit

- Pesan (dalam bentuk rangkaian bit) dipecah menajdi beberapa blok

- Contoh: Plainteks `100111010110`

  Bila dibagi menjadi blok 4-bit

  `1001  1101  0110`

  8-4-2-1 8-4-2-1 8-4-2-1

  maka setiap blok menyatakan `0` sampai `15`**:**

  `9  13  6`

Algoritma Enkripsi dengan rangkaian bit

Bila plainteks dibagi menjadi blok 3-bit:

100 111 010 110

4-2-1 4-2-1 4-2-1 4-2-1

maka setiap blok menyatakan 0 sampai 7 **:**

4 7 2 6

Jenis Algoritma Kriptografi

- Algoritma Simetri

  a. Blok Chiper : DES, IDEA, AES

  b. Stream Chiper : OTP, A5 dan RC4
- Algoritma Asimetri : RSA, DH, ECC, DSA
- Fungsi Hash : MD5, SHA1

- Dalam presentasi kami menggunakan Algoritma AES, RSA dan MD5

ALGORITMA SIMETRI : BLOK CHIPER

AES (Advanced Encryption Standard)

KELOMPOK 5	KELOMPOK 2	KELOMPOK 4	KELOMPOK 3	KELOMPOK 1
1				
2				
3				
4				
5				
6				

Kel I : DES
Kel II : AES
Kel III : RSA
Kel IV : MD5
Kel V : SHA1

Nama File : Nobp_Nama_Kelas_MK.Doc kirim ke GCR paling lambat : Selasa / 01 Juni 2021, Jam 23:59:59 WIB. (UTS Take-home) setelah dikirim

AES (Advanced Encryption Standard)

- DES dianggap sudah tidak aman.
- Perlu diusulkan standard algoritma baru sebagai pengganti DES.
- National Institute of Standards and Technology (NIST) mengusulkan kepada Pemerintah Federal AS untuk sebuah standard kriptografi kriptografi yang baru.
- NIST mengadakan lomba membuat standard algoritma kriptografi yang baru. Standard tersebut kelak diberi nama Advanced Encryption Standard (AES).

AES (Advanced Encryption Standard)

- Pada bulan Oktober 2000, NIST mengumumkan untuk memilih Rijndael (dibaca: Rhine-doll)
- Pada bulan November 2001, Rijndael ditetapkan sebagai AES
- Diharapkan Rijndael menjadi standard kriptografi yang dominan paling sedikit selama 10 tahun.

AES (Advanced Encryption Standard)

- Tidak seperti *DES* yang berorientasi bit, *Rijndael* beroperasi dalam orientasi *byte.*
- Setiap putaran mengunakan kunci internal yang berbeda (disebut *round key*).
- *Enciphering* melibatkan operasi substitusi dan permutasi.
- Karena *AES* menetapkan panjang kunci adalah 128, 192, dan 256, maka dikenal *AES*-128, *AES*-192, dan *AES*-256

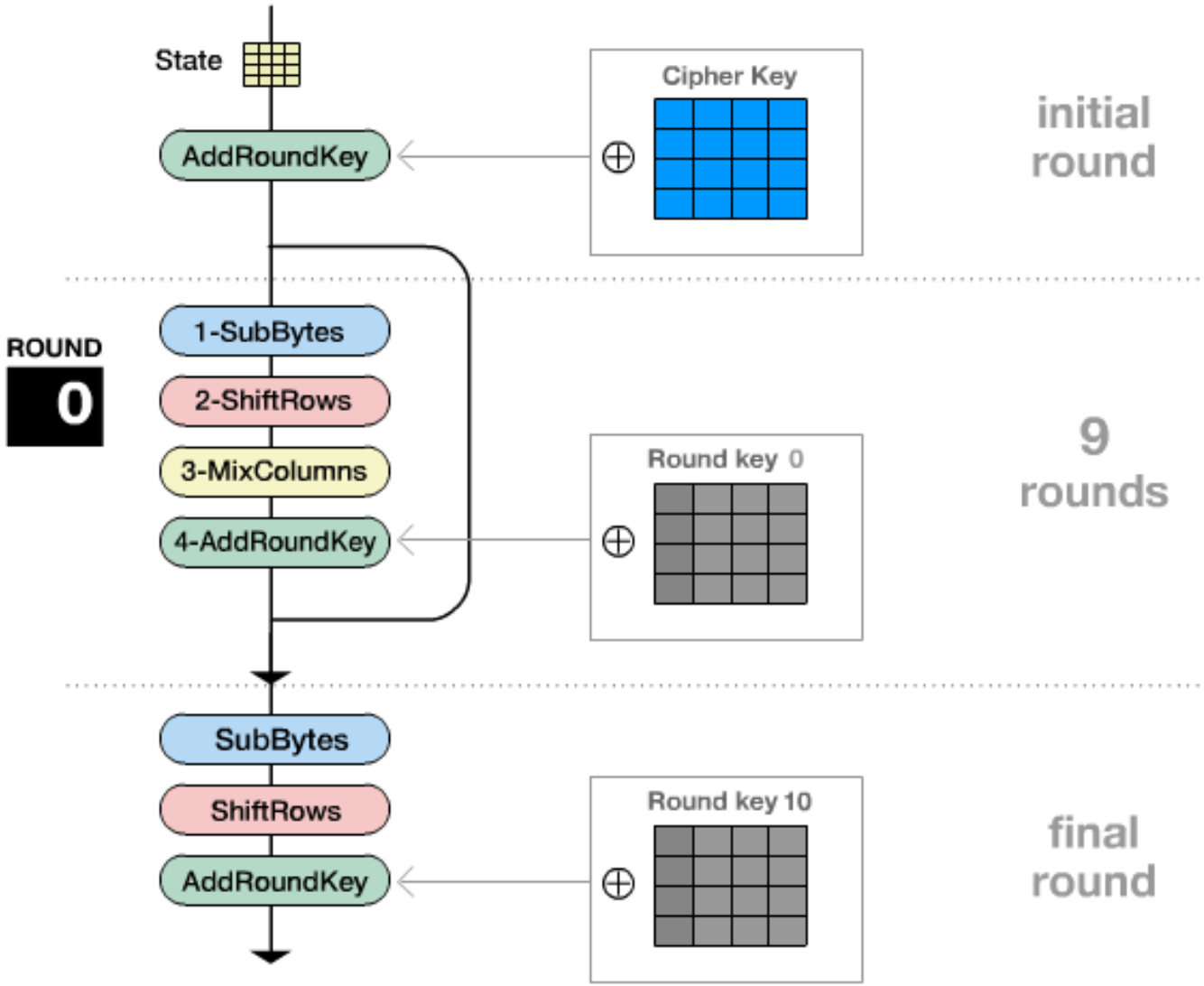
	Panjang Kunci (*Nk words*)	Ukuran Blok (*Nb words*)	Jumlah Putaran (*Nr*)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Catatan: 1 *word* = 32 bit

AES (Advanced Encryption Standard)

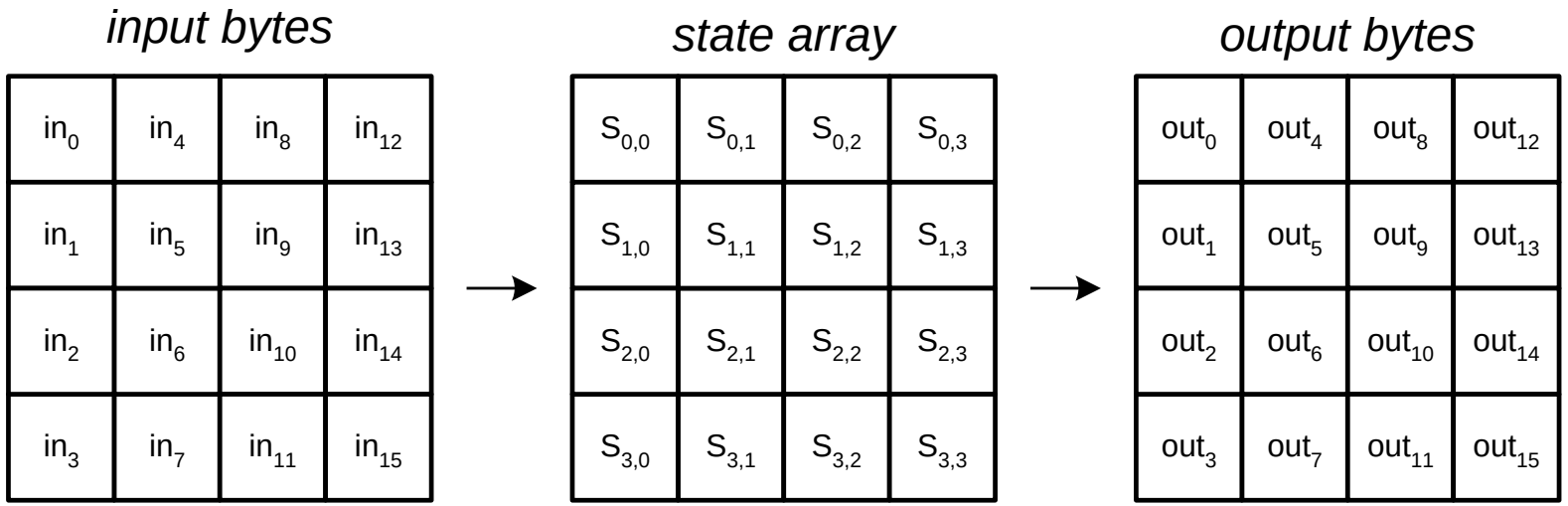
- Garis besar Algoritma *Rijndael* yang beroperasi pada blok 128-bit dengan kunci 128-bit adalah sebagai berikut (di luar proses pembangkitan *round key*):
  - *AddRoundKey*: melakukan *XOR* antara *state* awal (plainteks) dengan *cipher key*. Tahap ini disebut juga *initial round*.
  - Putaran sebanyak *Nr* – 1 kali. Proses yang dilakukan pada setiap putaran adalah:
    - *SubBytes*: substitusi *byte* dengan menggunakan tabel substitusi (*S-box*).
    - *ShiftRows*: pergeseran baris-baris *array state* secara *wrapping*.
    - *MixColumns*: mengacak data di masing-masing kolom *array state.*
    - *AddRoundKey*: melakukan XOR antara *state* sekarang *round key*.
  - *Final round*: proses untuk putaran terakhir:
    - *SubBytes*
    - *ShiftRows*
    - *AddRoundKey*

AES (Advanced Encryption Standard)



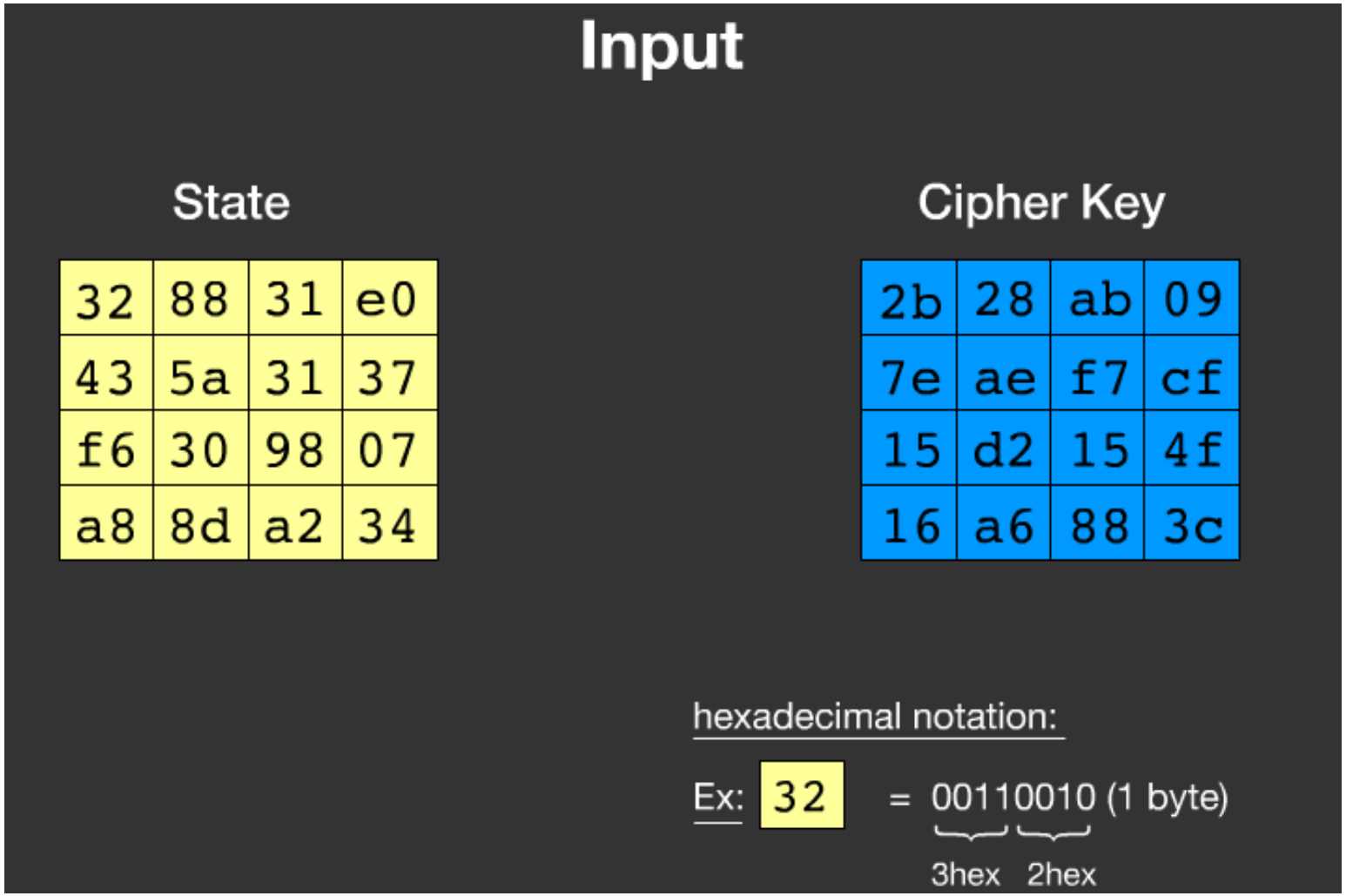
AES (Advanced Encryption Standard)

- Selama kalkulasi plainteks menjadi cipherteks, status sekarang dari data disimpan di dalam *array of bytes* dua dimensi, `state`, yang berukuran `NROWS` × `NCOLS`.
- Untuk blok data 128-bit, ukuran `state` adalah 4 × 4.
- Elemen *array* `state` diacu sebagai $S[r,c]$, $0 \leq r < 4$ dan $0 \leq c < Nb$ (*Nb* adalah panjang blok dibagi 32.
- Pada *AES*-128, *Nb* = 128/32 = 4)



AES (Advanced Encryption Standard)

- Contoh: (elemen state dan kunci dalam notasi HEX)



ALGORITMA ASIMETRI

RSA

RSA

- Ditemukan oleh tiga orang yaitu **R**on Rivest, Adi **S**hamir, dan Leonard **A**dleman yang kemudian disingkat menjadi RSA.
- Termasuk algritma asimetri karena mempunyai dua kunci, yaitu kunci publik dan kunci privat.
- Algoritma kunci-publik yang paling terkenal dan paling banyak aplikasinya.
- Ditemukan oleh tiga peneliti dari MIT (Massachussets Institute of Technology), yaitu Ron Rivest, Adi Shamir, dan Len Adleman, pada tahun 1976.
- Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima.

RSA

Pembangkitan pasangan kunci

1. Pilih dua bilangan prima, *a* dan *b* (rahasia)
2. Hitung $n = a\ b$. Besaran *n* tidak perlu dirahasiakan.
3. Hitung $\phi(n) = (a - 1)(b - 1)$.
4. Pilih sebuah bilangan bulat untuk kunci publik, sebut namanya *e*, yang relatif prima terhadap $\phi(n)$.
5. Hitung kunci dekripsi, *d*, melalui $ed \equiv 1$ (**mod** *m*) atau $d \equiv e^{-1} \bmod (\phi(n))$

Hasil dari algoritma di atas:

- Kunci publik adalah pasangan (*e*, *n*)
- Kunci privat adalah pasangan (*d*, *n*)

Catatan: *n* tidak bersifat rahasia, namun ia diperlukan pada perhitungan enkripsi/dekripsi

RSA

Kunci Publik

- Misalkan a = 47 dan b = 71 (keduanya prima), maka dapat dihitung:

  $n = a \times b = 3337$

  $\phi(n) = (a - 1) \times (b - 1) = 46 \times 70 = 3220$.
- Pilih kunci publik e = 79 (yang relatif prima dengan 3220 karena pembagi bersama terbesarnya adalah 1).
- Hapus a dan b dan kunci publiknya adalah n=3337 dan e=79

Kunci Privat

- Selanjutnya akan dihitung kunci privat d dengan kekongruenan:

$$e \times d \equiv 1 \pmod{m} = =>$$

Dengan mencoba nilai-nilai k = 1, 2, 3, …, diperoleh nilai d yang bulat adalah 1019. Ini adalah kunci privat (untuk dekripsi).

$$d = \frac{1 + (k \times 3220)}{79}$$

RSA

- Misalkan plainteks M = HARI INI

  atau dalam ASCII: 726582732737873

  Pecah M menjadi blok yang lebih kecil (misal 3 digit):

  $m_1 = 726$ $m_4 = 273$

  $m_2 = 582$ $m_5 = 787$

  $m_3 = 733$ $m_6 = 003$

  (Perhatikan, m_i masih terletak di dalam antara 0 sampai $n - 1$)

RSA

- *Enkripsi setiap blok:*

  $c_1 = 726^{79} \bmod 3337 = 215$

  $c_2 = 582^{79} \bmod 3337 = 776$, dst

  Chiperteks *C* = 215 776 1743 933 1731 158.

- *Dekripsi (menggunakan kunci privat d = 1019)*

  $m_1 = 215^{1019} \bmod 3337 = 726$

  $m_2 = 776^{1019} \bmod 3337 = 582$ dst untuk sisi blok lainnya

  Plainteks *M* = 726582732737873 yang dalam ASCII karakternya adalah HARI INI.

RSA

- ***Kekuatan dan Keamanan RSA***
  - Kekuatan algoritma *RSA* terletak pada tingkat kesulitan dalam memfaktorkan bilangan non prima menjadi faktor primanya, yang dalam hal ini $n = a \times b$.
  - Sekali n berhasil difaktorkan menjadi a dan b, maka $\phi(n) = (a - 1)\times(b - 1)$ dapat dihitung. Selanjutnya, karena kunci enkripsi e diumumkan (tidak rahasia), maka kunci dekripsi d dapat dihitung dari persamaan $ed \equiv 1 \pmod{n}$.
  - Penemu algoritma RSA menyarankan nilai a dan b panjangnya lebih dari 100 digit. Dengan demikian hasil kali n = a □ b akan berukuran lebih dari 200 digit.
  - Menurut Rivest dan kawan-kawan, usaha untuk mencari faktor bilangan 200 digit membutuhkan waktu komputasi selama 4 milyar tahun! (dengan asumsi bahwa algoritma pemfaktoran yang digunakan adalah algoritma yang tercepat saat ini dan komputer yang dipakai mempunyai kecepatan 1 milidetik).

Referensi utama :

>> Michael Felderer , Riccardo Scandariato (editor) - Exploring Security in Software Architecture and Design, 2018.

>> Nancy R. Mead, Carol Woody - Cyber Security Engineering_ A Practical Approach for Systems and Software Assurance-Addison-Wesley Professional (2016)

>> James Helfrich - Security for Software Engineers-CRC Press (2019)

>> Pete Loshin - Simple Steps to Data Encryption_ A Practical Guide to Secure Computing-Syngress (2013)

>> Tevfik Bultan,Fang Yu,Muath Alkhalaf,Abdulbaki Aydin (auth.) - String Analysis for Software Verification and Security (2017)


Any
Questions?