STMIK Amik Riau

MATAKULIAH
KEAMANAN PERANGKAT LUNAK

SI Channels

Pertemuan 14

PRAKTIKUM LANJUTAN KRIPTOGRAFI MODERN

SYAHRUL IMARDI, M.T

#14

Syahrul Imardi, MT

STMIK Amik Riau

MATAKULIAH

KEAMANAN PERANGKAT LUNAK

Praktikum (Lanjutan) KRIPTOGRAFI MODERN

P14

STMIK Amik Riau

MATAKULIAH
KEAMANAN PERANGKAT LUNAK

Syahrul Imardi, MT

P14 : Kriptografi Modern (Lanjutan) – Praktikum

Kriptografi Modern

(Bagian 4: Prinsip Perancangan Block Cipher)

Prinsip-prinsip Perancangan *Cipher* Blok

1. Prinsip *Confusion* dan *Diffusion* dari Shannon.
2. *Cipher* berulang (*iterated cipher*)
3. Jaringan Feistel (*Feistel Network*)
4. Kotak-S (*S-box*)

Prinsip *Confusion* dan *Diffusion* dari Shannon.

- Banyak algoritma kriptografi klasik yang telah berhasil dipecahkan karena distribusi statistik plainteks dalam suatu bahasa diketahui.

- Claude Shannon dalam makalah klasiknya tahun 1949, *Communication theory of secrecy systems*, memperkenalkan prinsip *confusion* dan *diffusion* untuk membuat serangan statistik menjadi rumit.

- Dua prinsip tersebut menjadi panduan dalam merancang algoritma kriptografi.

Confusion

- Prinsip ini menyembunyikan hubungan apapun yang ada antara plainteks, cipherteks, dan kunci.

- Prinsip *confusion* membuat kriptanalis frustasi untuk mencari pola-pola statistik yang muncul pada cipherteks.

- *One-Time Pad* adalah contoh algoritma yang *confuse*.

- *Confusion* dapat direalisasikan dengan menggunakan algoritma substitusi yang kompleks.

- DES mengimplementasikan substitusi dengan menggunakan kotak-S.

Diffusion

- Prinsip ini menyebarkan pengaruh satu bit plainteks atau kunci ke sebanyak mungkin cipherteks.

- Sebagai contoh, pengubahan kecil pada plainteks sebanyak satu atau dua bit menghasilkan perubahan pada cipherteks yang tidak dapat diprediksi.

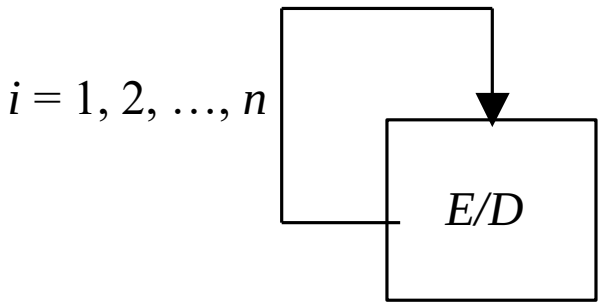
- Mode CBC dan CFB menggunakan prinsip ini

- Pada algoritma DES, *diffusion* direalisasikan dengan menggunakan operasi permutasi.

Cipher Berulang (*Iterated Cipher*)

- Fungsi transformasi sederhana yang mengubah plainteks menjadi cipherteks diulang sejumlah kali.

- Pada setiap putaran digunakan upa-kunci (*subkey*) atau kunci putaran (*round key*) yang dikombinasikan dengan plainteks.



- *Cipher* berulang dinyatakan sebagai

$$C_i = f(C_{i-1}, K_i)$$

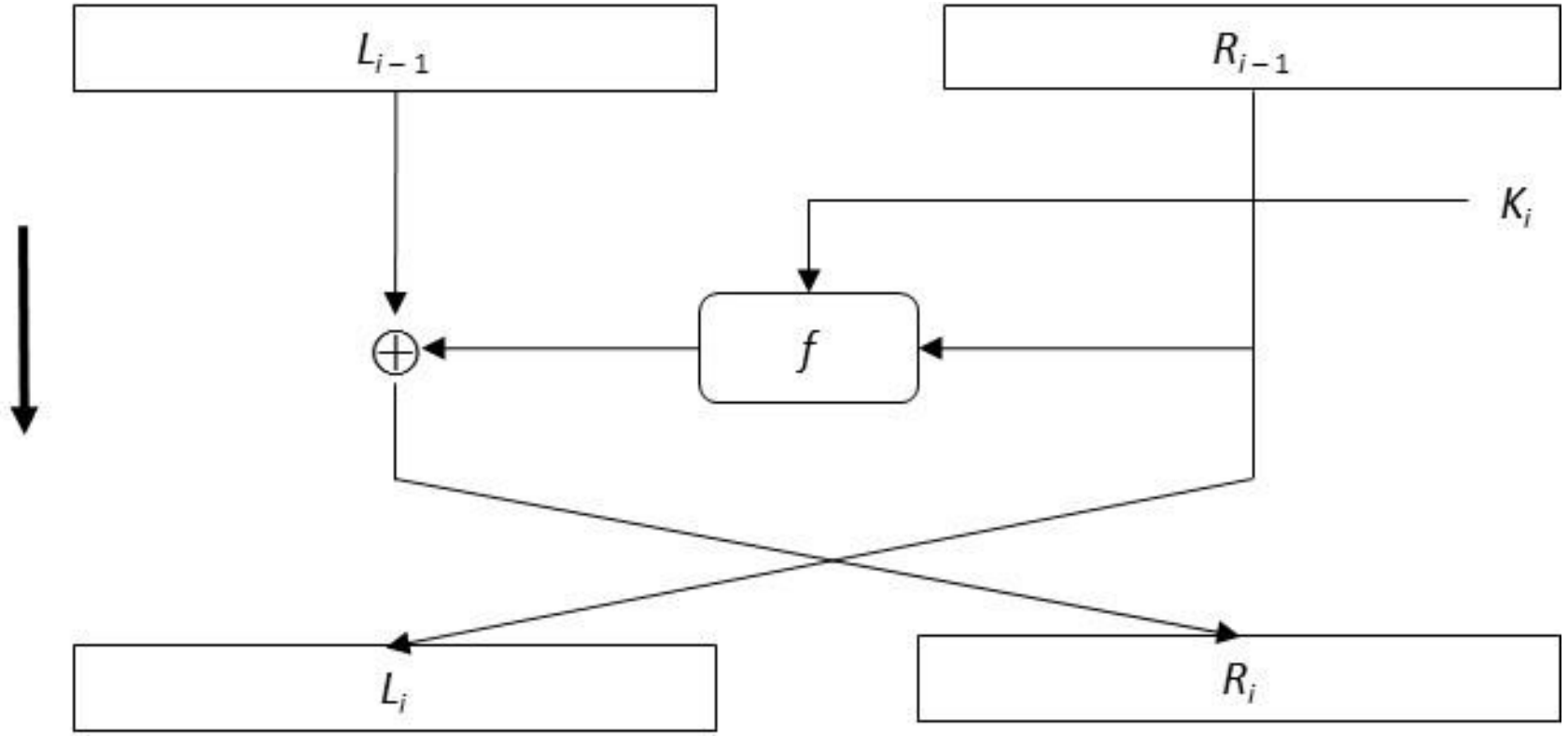
i = 1, 2, …, r (r adalah jumlah putaran).

K_i = upa-kunci (*subkey*) pada putaran ke-i

f = fungsi transformasi (di dalamnya terdapat operasi substitusi, permutasi, dan/atau ekspansi, kompresi).

Plainteks dinyatakan dengan C_0 dan cipherteks dinyatakan dengan C_r.

Jaringan Feistel (*Feistel Network*)



Jaringan *Feistel* pada enkripsi putaran ke-*i*

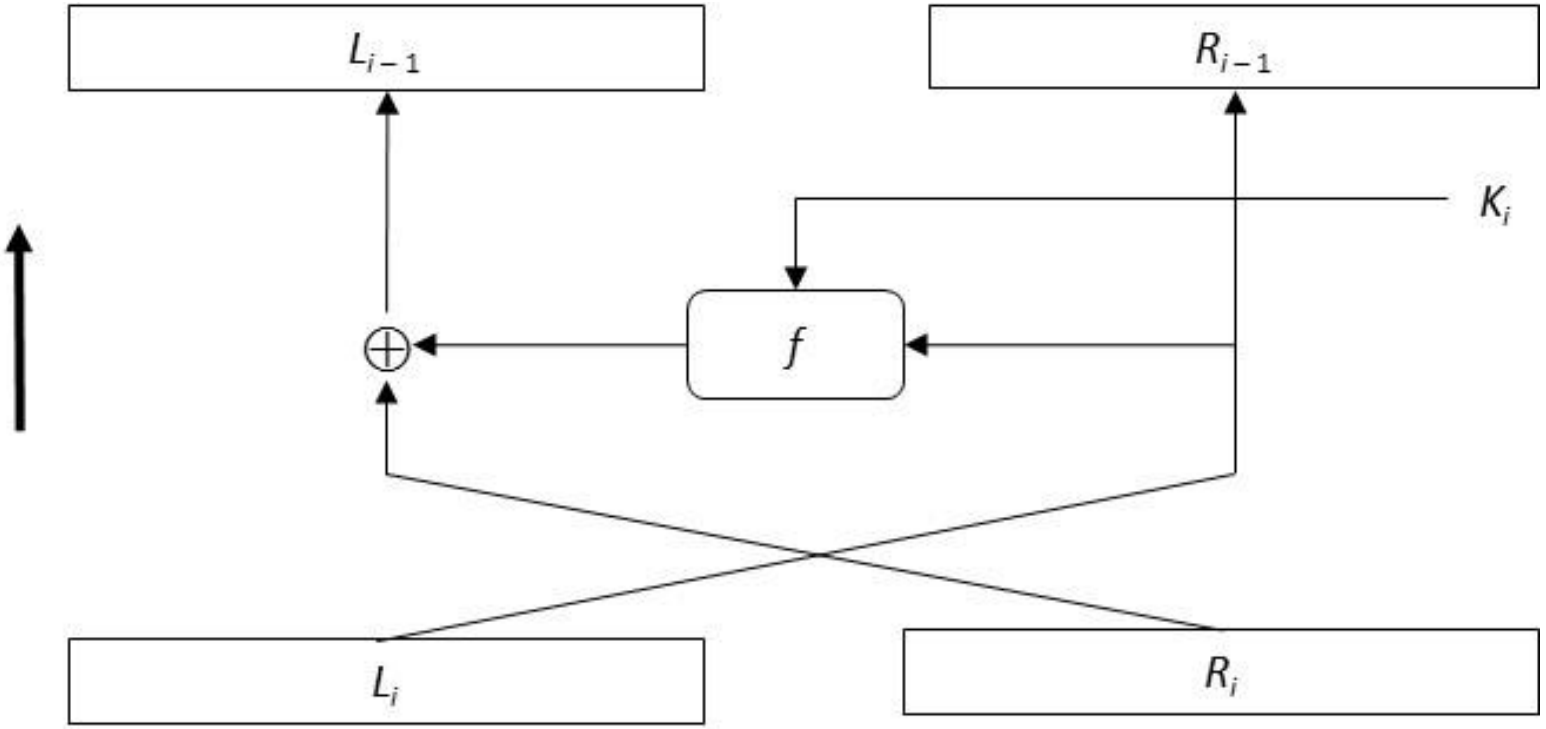
$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus f(R_{i-1,}, K_i)$$

- Jaringan *Feistel* banyak dipakai pada algoritma kriptografi *DES*, *LOKI*, *GOST*, *FEAL*, *Lucifer*, *Blowfish*, dan lain-lain karena model ini bersifat *reversible* untuk proses enkripsi dan dekripsi.

- Sifat *reversible* ini membuat kita tidak perlu membuat algoritma baru untuk mendekripsi cipherteks menjadi plainteks.

  Contoh: $L_{i-1} \oplus f(R_{i-1}, K_i) \oplus f(R_{i-1}, K_i) = L_{i-1}$

- Sifat *reversible* tidak bergantung pada fungsi *f* sehingga fungsi *f* dapat dibuat serumit mungkin.



Jaringan *Feistel* pada dekripsi putaran ke-*i*

$$R_{i-1} = L_i$$
$$L_{i-1} = R_i \oplus f(R_{i-1}, K_i) = R_i \oplus f(L_i, K_i)$$

Kotak-*S* (S-*box*)

- Kotak-*S* adalah matriks yang berisi substitusi sederhana yang memetakan satu atau lebih bit dengan satu atau lebih bit yang lain.

- Pada kebanyakan algoritma *cipher* blok, kotak-*S* memetakan *m* bit masukan menjadi *n* bit keluaran, sehingga kotak-*S* tersebut dinamakan kotak $m \times n$ *S-box*.

- Kotak-*S* merupakan satu-satunya langkah nirlanjar di dalam algoritma, karena operasinya adalah *look-up table*. Masukan dari operasi *look-up table* dijadikan sebagai indeks kotak-*S*, dan keluarannya adalah *entry* di dalam kotak-*S*.

Contoh: Kotak-*S* di dalam algoritma *DES* adalah 6 × 4 *S-box* yang berarti memetakan 6 bit masukan menjadi 4 bit keluaran. Salah satu kotak-*S* yang ada di dalam algoritma DES adalah sebagai berikut:

12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

Baris diberi nomor dari 0 sampai 3
Kolom diberi nomor dari 0 sampai 15

Masukan untuk proses substitusi adalah 6 bit,

$$b_1b_2b_3b_4b_5b_6$$

Nomor baris dari tabel ditunjukkan oleh *string* bit b_1b_6
(menyatakan 0 sampai 3 desimal)

Nomor kolom ditunjukkan oleh *string* bit $b_2b_3b_4b_5$
(menyatakan 0 sampai 15)

- Misalkan masukan adalah 110100

  Nomor baris tabel = 10 (baris 2)

  Nomor kolom tabel = 1010 (kolom 10)

  Jadi, substitusi untuk 110100 adalah *entry* pada baris 2 dan kolom 10, yaitu 0100 (atau 4 desimal).

- DES mempunyai 8 buah kotak-S

- Pada AES kotak S hanya ada satu buah:

hex		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

S-BOX

19	a0	9a	e9
3d	f4	c6	f8
e3	e2	8d	48
be	2b	2a	08

hex	y															
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

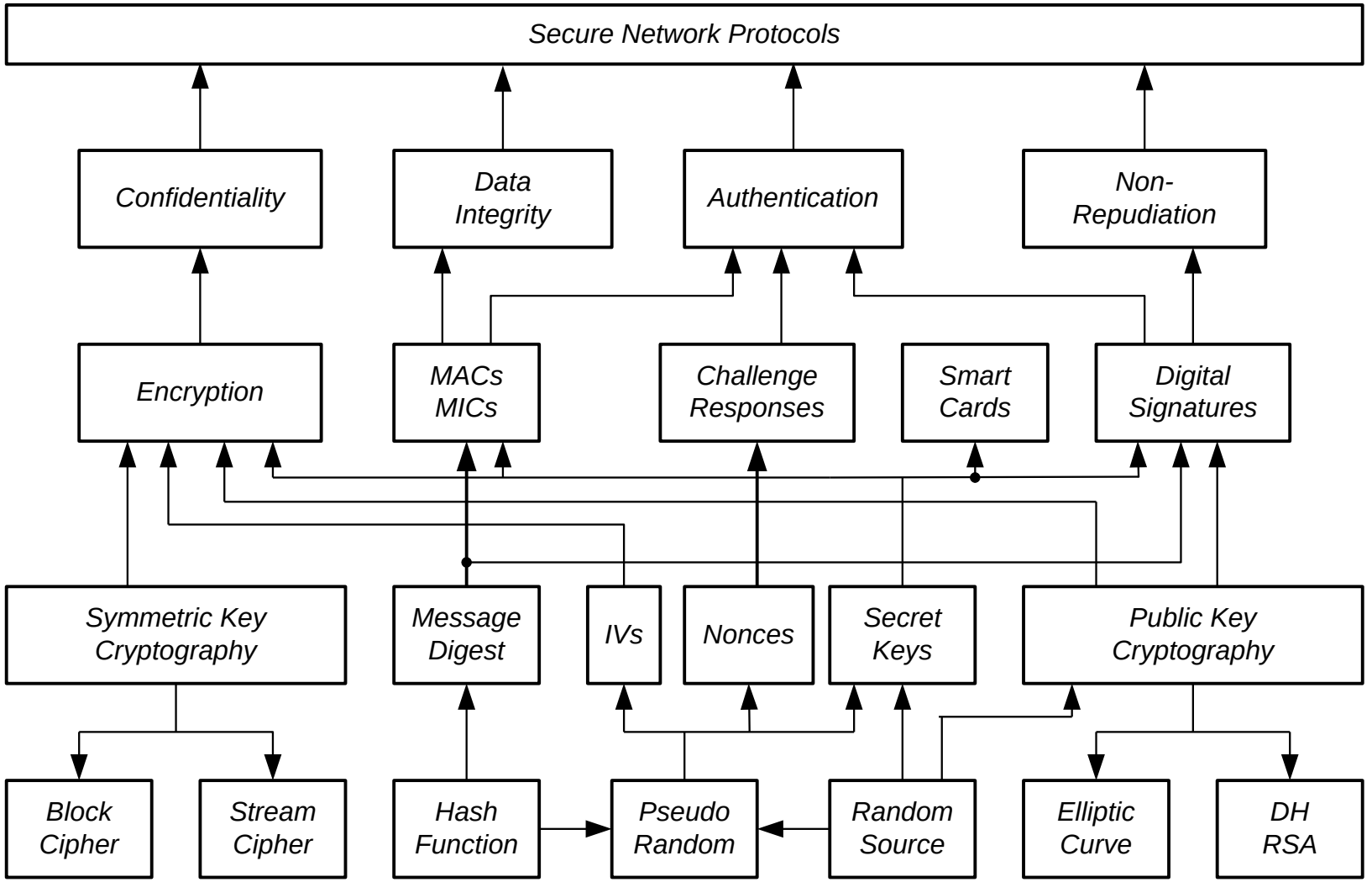
	a0	9a	e9
3d	f4	c6	f8
e3	e2	8d	48
be	2b	2a	08

19

d4

hex	y															
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Diagram Blok Kriptografi Modern



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?