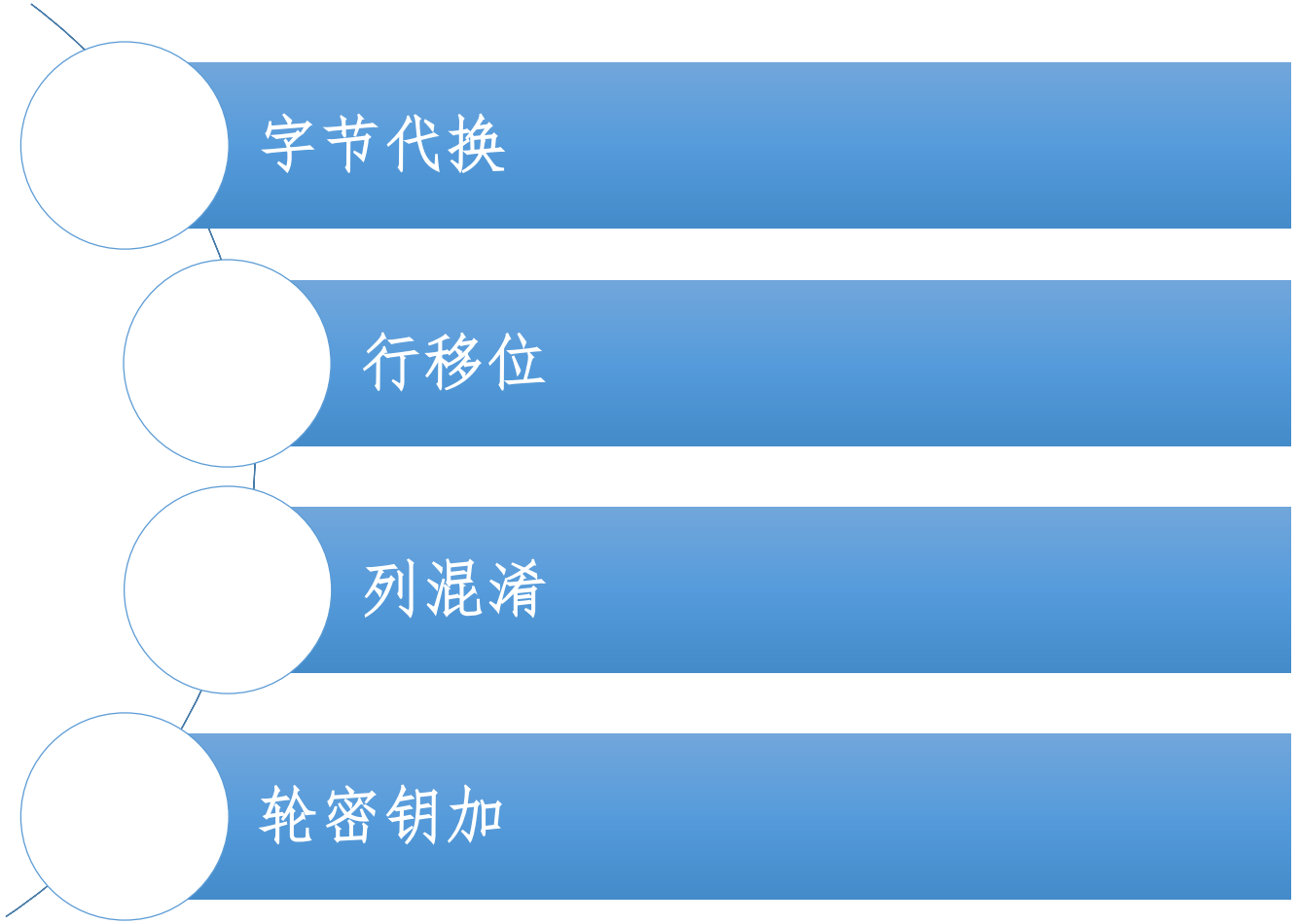
现代密码学

第二十八讲 AES的轮函数

信息与软件工程学院

第二十八讲 AES的轮函数



字节代换

- 非线性代换，独立地对状态的每个字节进行，并且代换表(S盒)可逆，记为**ByteSub(State),**分两步

(1) 将字节作为**GF(2^8)**上的元素映射到自己的逆元

(2) 将字节做**GF(2)**上的仿射变换

即

$$y = Ax^{-1} + B$$

其中**A**是一个**GF**（2）上**8X8**的可逆矩阵，**B**是**GF**（2）上一个**8**位列向量

字节代换

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

AES的S盒

		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

AES的S盒的使用：输入8a，输出7e，即7e=S（8a）

		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

逆字节代换InvSubBytes()

- 逆字节替代变换是字节替代变换的逆变换，在状态的每个字节上应用逆S盒
  - 这是通过应用字节替代变换中的仿射变换的逆变换，再对所得结果应用有限域的乘法逆运算得到的

即

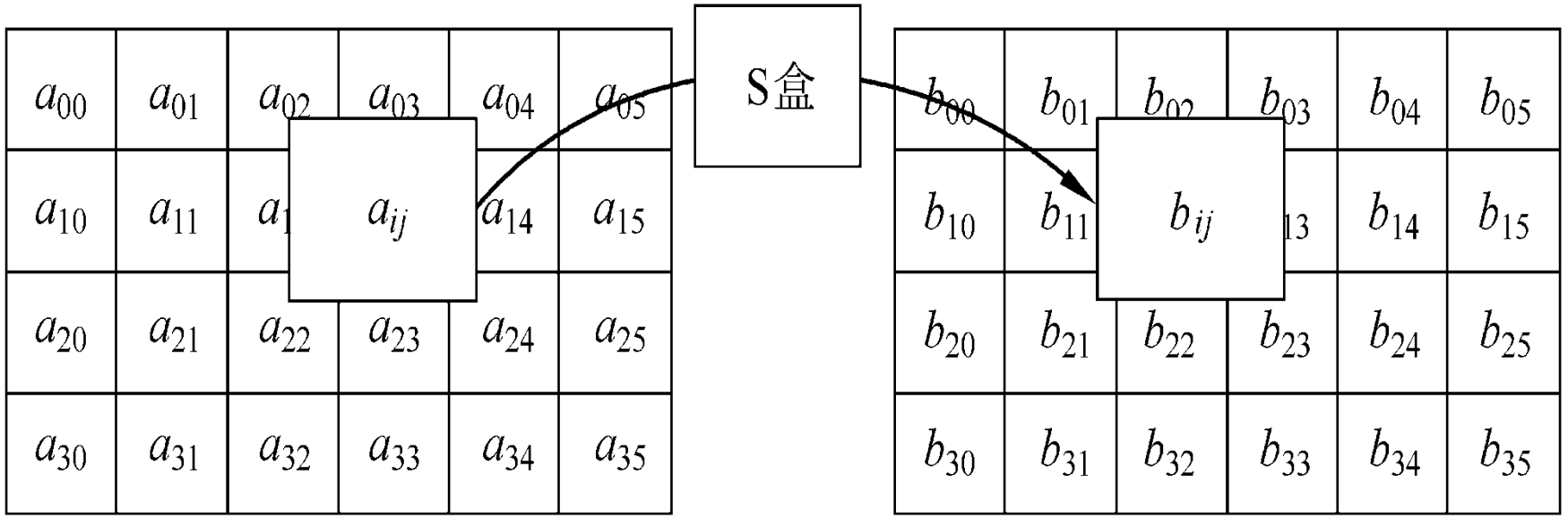
$$y = A^{-1}(x - B)$$

AES的逆S盒

		y														
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	08	2e	A1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	B6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
	7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
	a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
	b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
	d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
	e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

字节代换示意图

- 上述S-盒对状态的所有字节所做的变换记为ByteSub（State）



第二十八讲 AES的轮函数

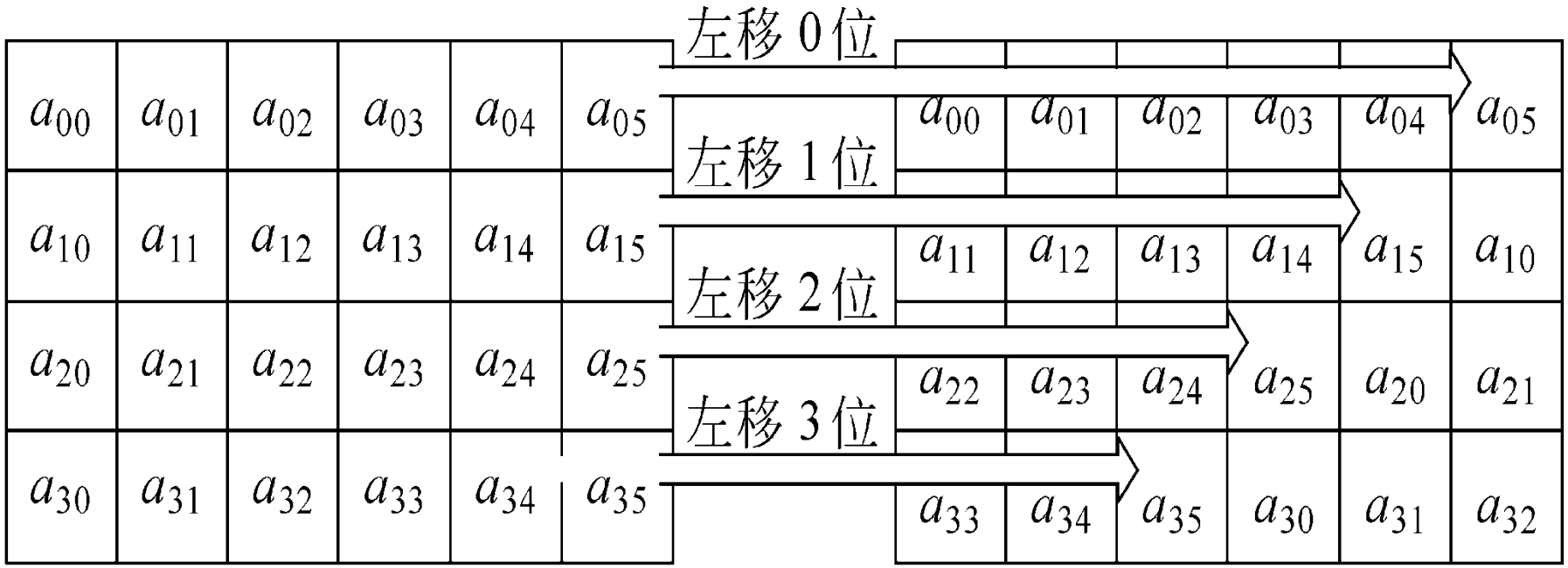
- 字节代换
- 行移位
- 列混淆
- 轮密钥加

行移位

- 将状态阵列的各行进行循环移位，不同行的移位量不同
- 0行：不动
- 1行：循环左移C1字节
- 2行：循环左移C2字节
- 3行：循环左移C3字节
- 记为：ShiftRow(State)

N_b	C1	C2	C3
4	1	2	3
6	1	2	3
8	1	3	4

行移位示意图


左移 0 位
左移 1 位
左移 2 位
左移 3 位
a_{00} a_{01} a_{02} a_{03} a_{04} a_{05}
a_{10} a_{11} a_{12} a_{13} a_{14} a_{15}
a_{20} a_{21} a_{22} a_{23} a_{24} a_{25}
a_{30} a_{31} a_{32} a_{33} a_{34} a_{35}
a_{00} a_{01} a_{02} a_{03} a_{04} a_{05}
a_{11} a_{12} a_{13} a_{14} a_{15} a_{10}
a_{22} a_{23} a_{24} a_{25} a_{20} a_{21}
a_{33} a_{34} a_{35} a_{30} a_{31} a_{32}

逆行移位InvShiftRows()

- 逆行移位变换是行移位变换的逆变换
  - 它对状态的每一行进行循环右移，
  - 第0行保持不变
  - 第1行循环右移C_1个字节
  - 第2行循环右移C_2个字节
  - 第3行循环右移C_3个字节

第二十八讲 AES的轮函数

- 字节代换
- 行移位
- 列混淆
- 轮密钥加

列混淆

- 将每列视为**GF(2^8)**上多项式，与固定的多项式**c(x)**进行模**x^4+1**乘法，记为⊗,要求**c(x)**模**x^4+1**可逆。
- 表示为**MixColumn(State)**

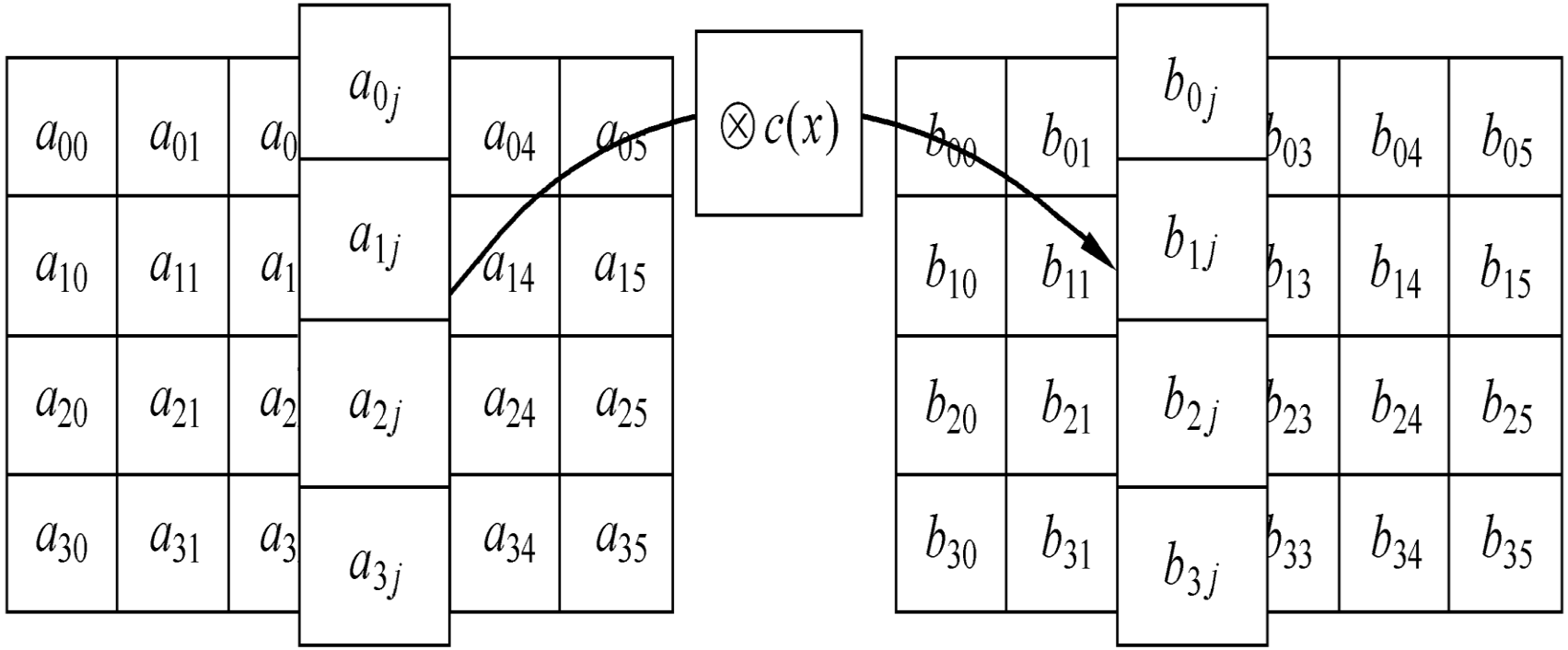
$$c(x)='03'x^3+'01'x^2+'01'x+'02'$$

列混淆的矩阵表示

列混淆运算也可写为矩阵乘法。设 $b(x)=c(x)\otimes a(x)$，则

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

列混淆运算示意图


a_{00} a_{01} a_{0} a_{0j} a_{04} a_{05}
a_{10} a_{11} a_{1} a_{1j} a_{14} a_{15}
a_{20} a_{21} a_{2} a_{2j} a_{24} a_{25}
a_{30} a_{31} a_{3} a_{3j} a_{34} a_{35}
$\otimes c(x)$
b_{00} b_{01} b_{0j} b_{03} b_{04} b_{05}
b_{10} b_{11} b_{1j} b_{13} b_{14} b_{15}
b_{20} b_{21} b_{2j} b_{23} b_{24} b_{25}
b_{30} b_{31} b_{3j} b_{33} b_{34} b_{35}

逆列混淆InvMixColumns()

- 逆列混淆变换是列混淆变换的逆
- 它将状态矩阵中的每一列视为系数在$GF(2^8)$上的次数小于**4**的多项式与同一个固定的多项式 $d(x)$相乘。 $d(x)$满足

$$(\text{‘03’}x^3+\text{‘01’}x^2+\text{‘01’}x+\text{‘02’})\otimes d(x)=\text{‘01’}$$

由此可得

$$d(x)=\text{‘0B’}x^3+\text{‘0D’}x^2+\text{‘09’}x+\text{‘0E’}$$

逆列混淆的矩阵形式

- 同样，逆列混淆可以写成矩阵乘法形式

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_2 \end{pmatrix} = \begin{pmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

第二十八讲 AES的轮函数

- 字节代换
- 行移位
- 列混淆
- 轮密钥加

轮密钥加

- 轮密钥与状态进行逐比特异或。
- 轮密钥由种子密钥通过密钥编排算法得到
- 轮密钥长度与分组长度相同
- 表示为AddRoundKey(State,RoundKey)

感 谢 聆 听!

xynie@uestc.edu.cn