现代密码学理论与实践
5 高级加密标准AES

苗付友，黄文超
主页：http://staff.ustc.edu.cn/~huangwc/crypto.html

mfy@ustc.edu.cn，huangwc@ustc.edu.cn

高级加密标准AES要点

- AES是一种**分组密码**，用以取代DES的商业应用。其**分组长度**为128位，**密钥长度**为128位、192位或256位
- AES**没有使用Feistel**结构。每轮由**四个独立的运算**组成：字节代换、置换、有限域上的算术运算，以及与密钥的异或运算

本章内容

1. AES概述
2. The AES Cipher – Rijndael
3. AES 的一轮加密过程
4. 安全性分析

1. AES概述
动机

- **DES的不安全**
- 建议用3DES，密钥168位，抵御密码分析攻击
  - 但是3DES用软件**实现速度较慢**，分组短，仅64位

1. AES概述

起源

- 美国国家标准技术协会NIST在1997年征集新标准，**要求分组**128位，**密钥**128、192或256
- 15候选算法在1998年6月通过了第一轮评估，仅有5个候选算法在1999年8月通过了第二轮评估
- 2000年10月，NIST选择Rijndael作为AES算法， Rijndael的作者是比利时的密码学家Joan Daemen博士和Vincent Rijmen博士
- 2001年11月，NIST完成评估并发布了最终标准 FIPS PUB 197

1. AES概述
初选的评估准则

- 安全性
  - 实际安全、随机性（观察等价）、可靠性（数学基础）、其它安全因素（公众提出的攻击方法）
- 成本
  - 专利要求、计算效率、存储空间要求
- 算法和执行特征
  - 灵活性（长度、多平台、多用性）、软 / 硬件、简洁性

1. AES概述
最终评估的准则

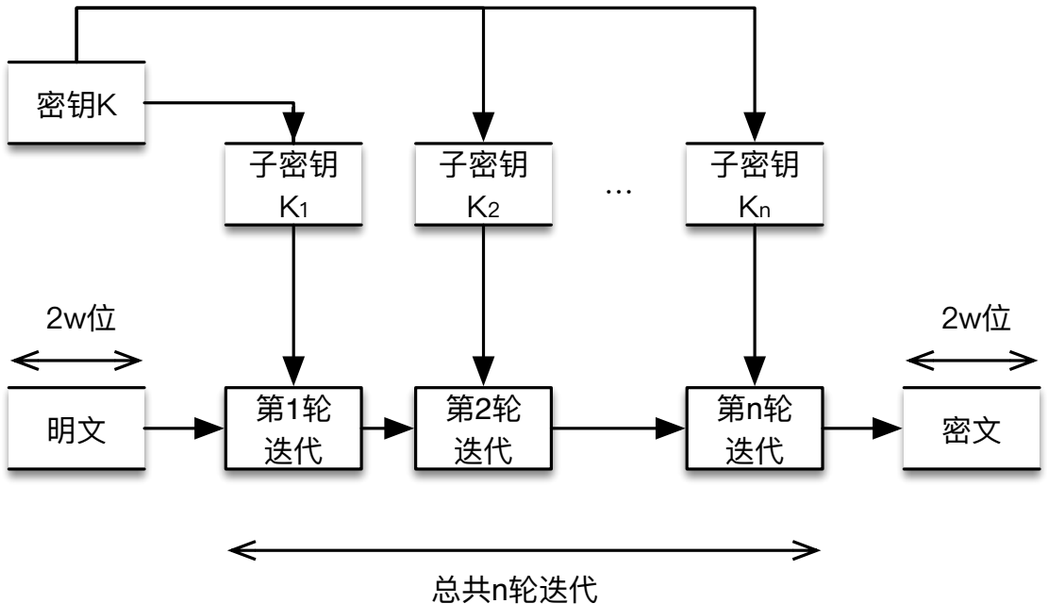
- **一般安全性** — 依赖于密码学界的公共安全分析
- **软件实现** — 软件执行速度，跨平台执行能力及密钥长度改变时速度变化
- **受限空间环境** — 在诸如智能卡中的应用
- **硬件实现** — 硬件实现时能够提高执行速度或缩短代码长度
- **抵御密码分析攻击** — 计时攻击、能量分析攻击
- **密钥灵活性** — 快速改变密钥长度的能力
- **其他的多功能性和灵活性**
- **指令级并行执行的潜力**

2 AES 密码
回顾

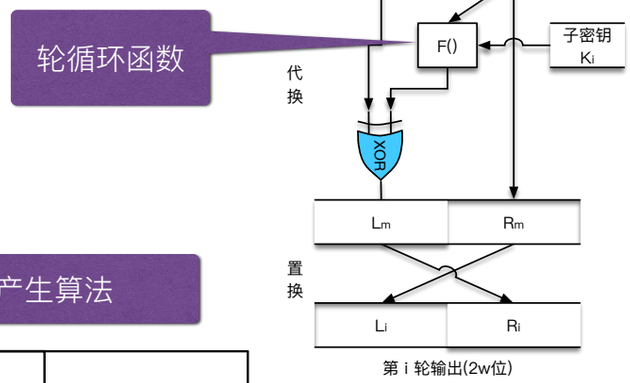
- 1973年，Feistel提出使用**乘积密码**的概念**逼近理想分组**密码
  - **乘积密码**：依次使用两个或两个以上的基本密码，所得结果的密码强度将**强于**所有**单个密码**的强度
  - Feistel建议交替使用**代换**和**置换**
    - 对应于1949年Shannon提出的**混淆**和**扩散**
    - 当前使用的**大多数重要对称分组密码的基本结构**

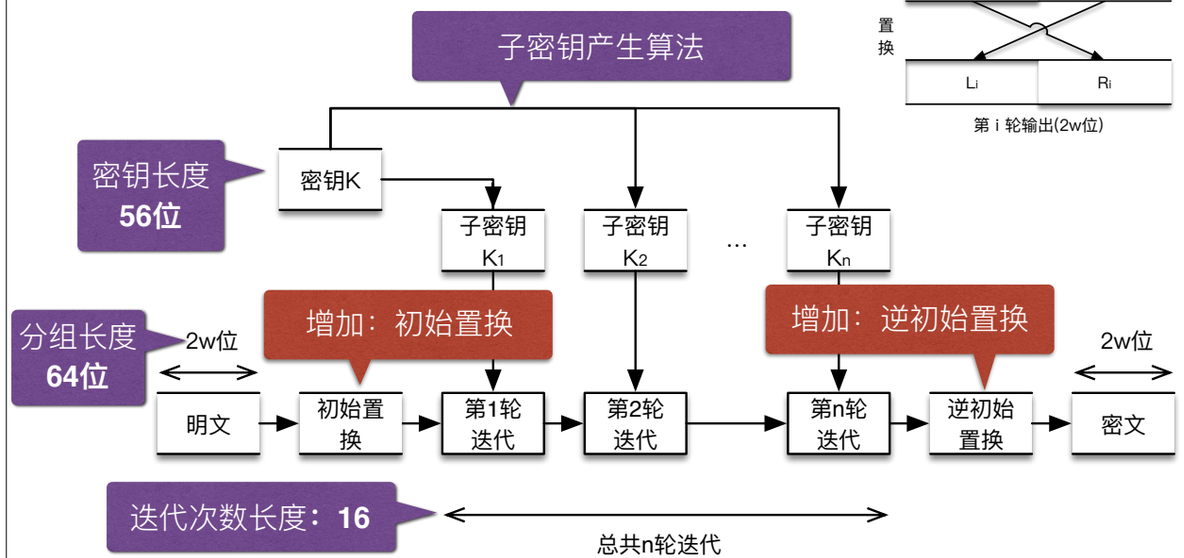
2 AES 密码

回顾

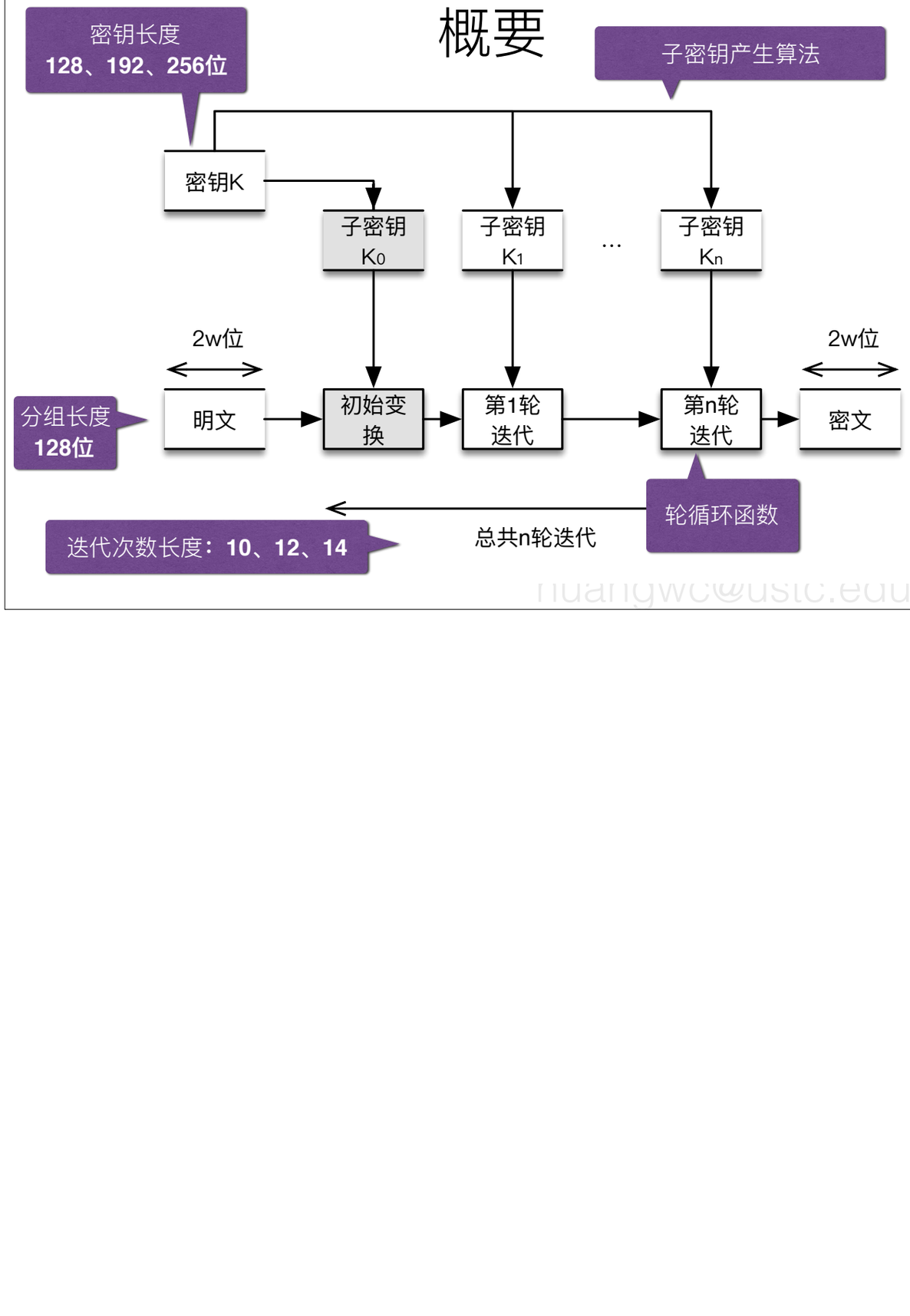


DES方案

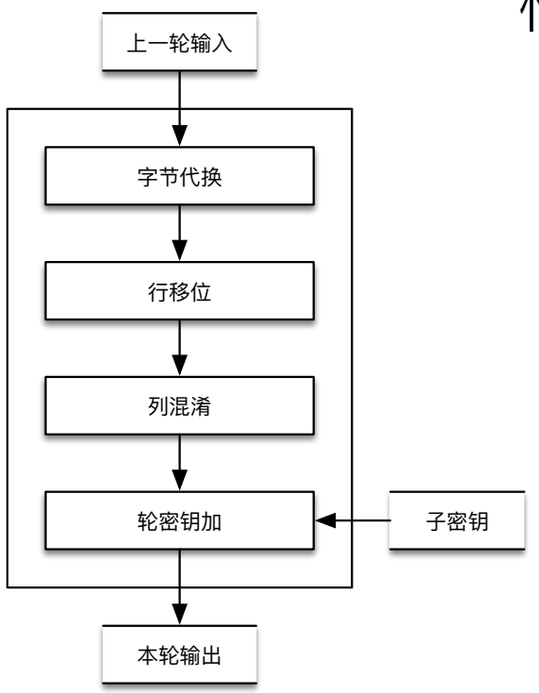

轮循环函数
第 i 轮输入(2w位)
Li-1
Ri-1
F()
子密钥
Ki
代
换
BOX
Lm
Rm
置
换
Li
Ri
第 i 轮输出(2w位)

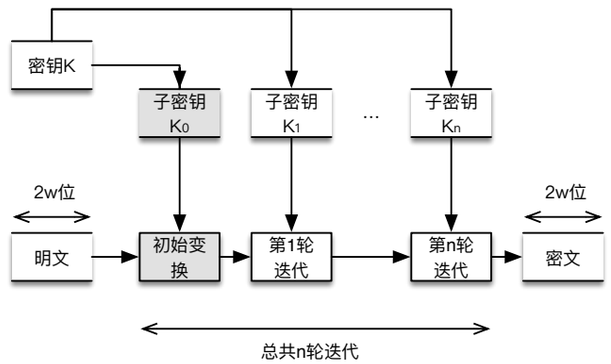

子密钥产生算法
密钥长度
56位
密钥K
子密钥
K1
子密钥
K2
...
子密钥
Kn
分组长度
64位
2w位
增加：初始置换
增加：逆初始置换
2w位
明文
初始置
换
第1轮
迭代
第2轮
迭代
第n轮
迭代
逆初始
置换
密文
迭代次数长度：16
总共n轮迭代

2 AES 密码概要

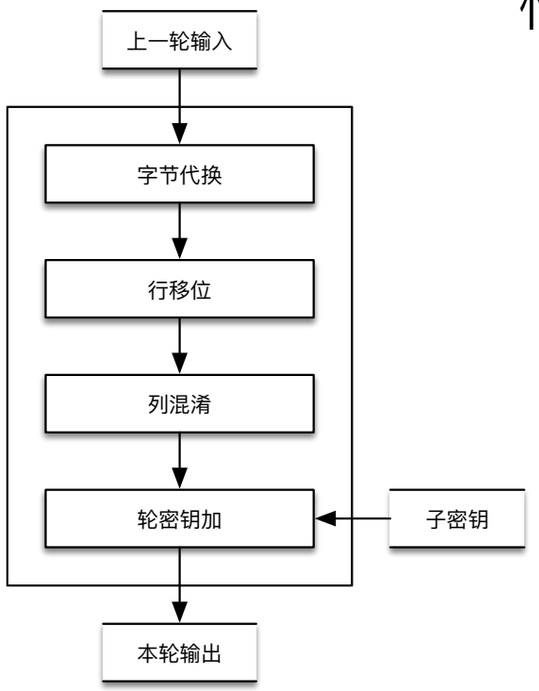


2 AES 密码概要

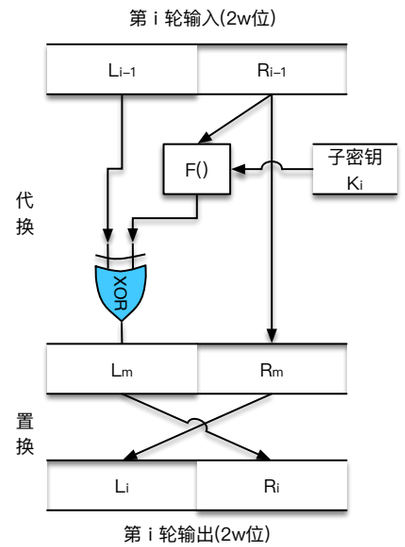




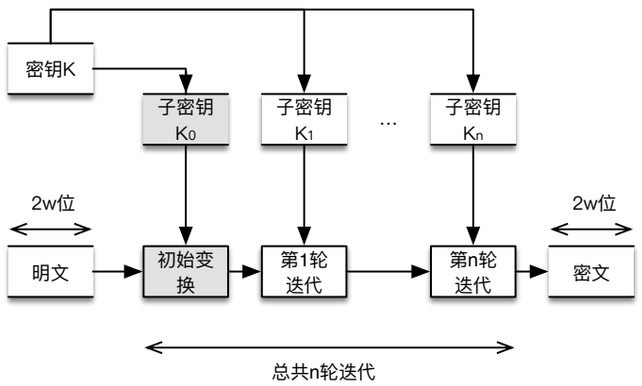
2 AES 密码概要



- 不是Feistel 结构

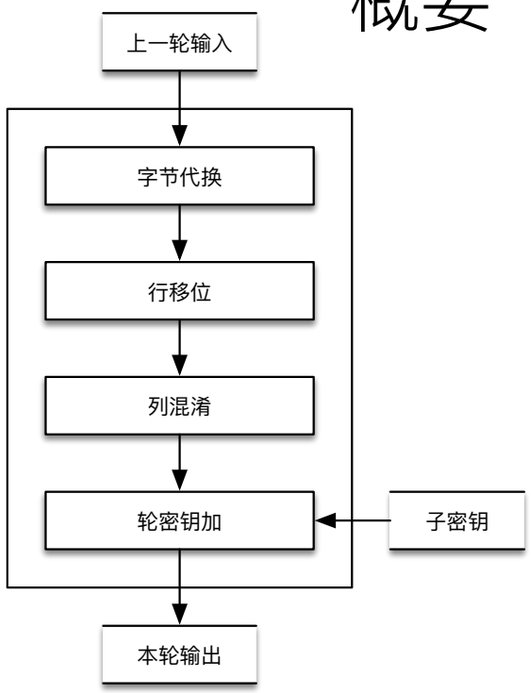


2 AES 密码
概要（128位）



- n=**10**时
  - K被扩展成**11**个128位的子密钥 （16字节）

2 AES 密码概要（128位）



- 迭代过程
  - 输入
    - state数组 （16字节）
    - 子密钥 （16字节）
  - 输出
    - state数组(16字节)

2 AES 密码
概要（128位）

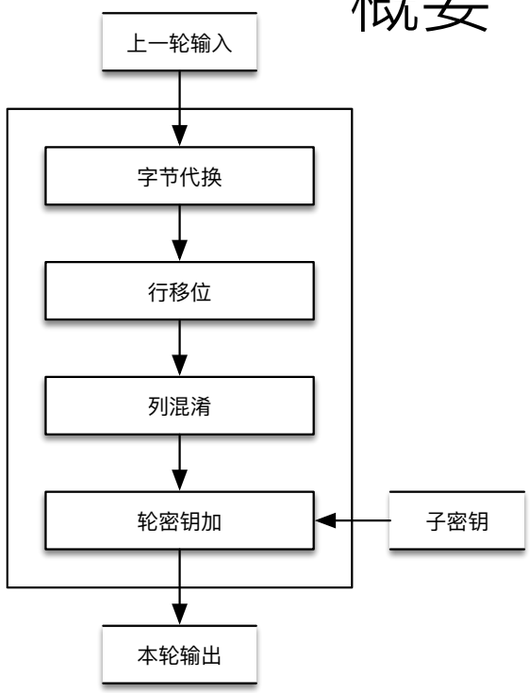
- 运算法则
  - GF(2^8)算术
  - 不可约多项式 $m(x) = x^8 + x^4 + x^3 + x + 1$
  - 例： $A = (a_7 a_6 \dots a_1 a_0)$ $B = (b_7 b_6 \dots b_1 b_0)$

$$A + B = (c_7 c_6 \dots c_1 c_0) \qquad c_i = a_i \oplus b_i$$

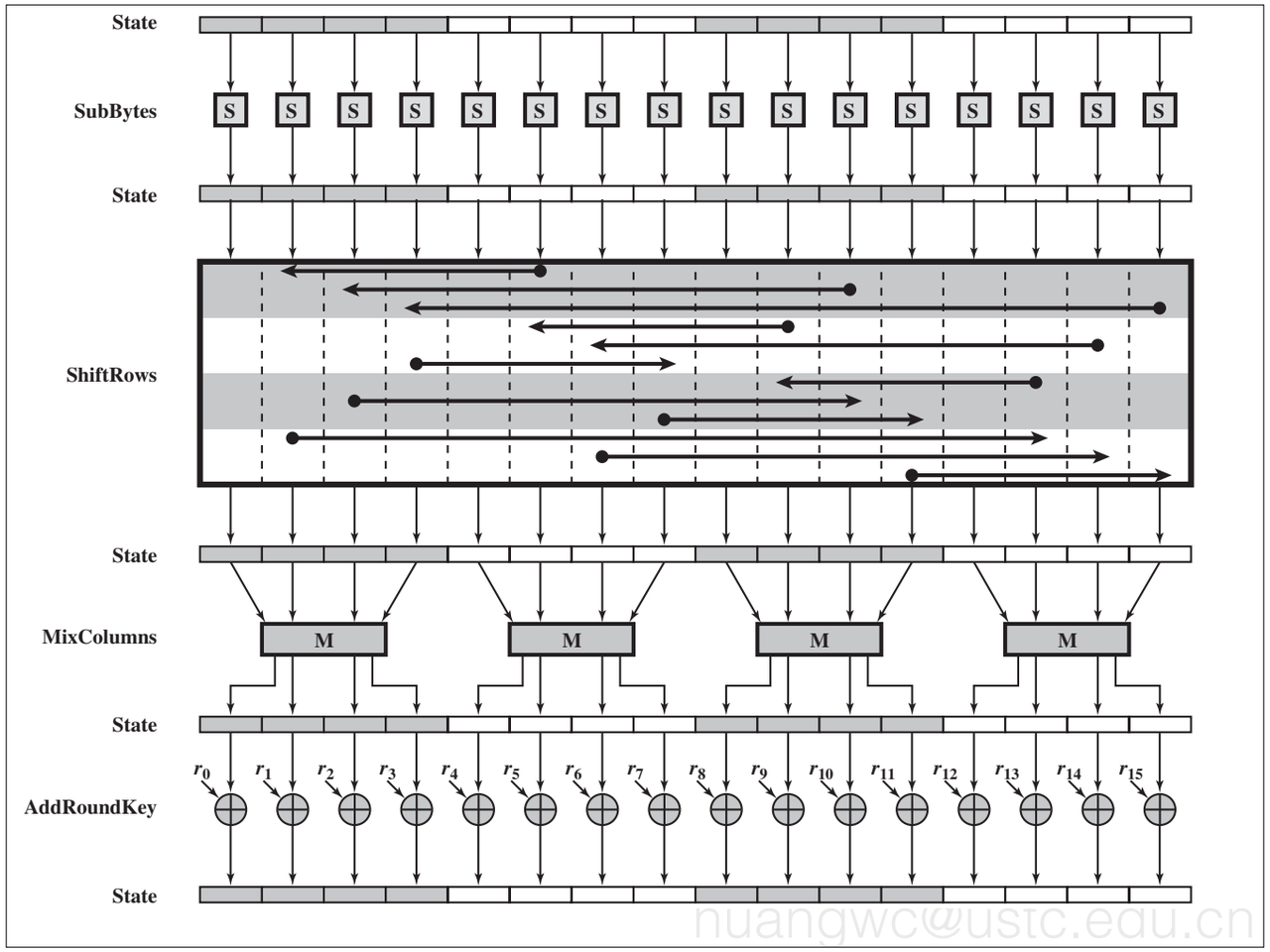
$$\{02\} \cdot A = (a_6 \dots a_1 a_0 0), \text{if } a_7 = 0$$

$$\{02\} \cdot A = (a_6 \dots a_1 a_0 0) \oplus (00011011), \text{if } a_7 = 1$$

2 AES 密码概要（128位）

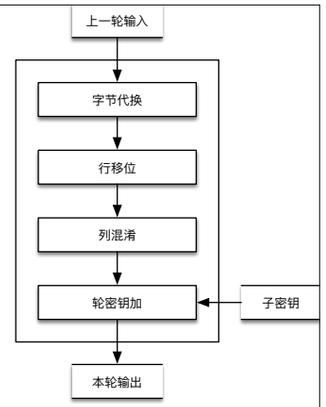


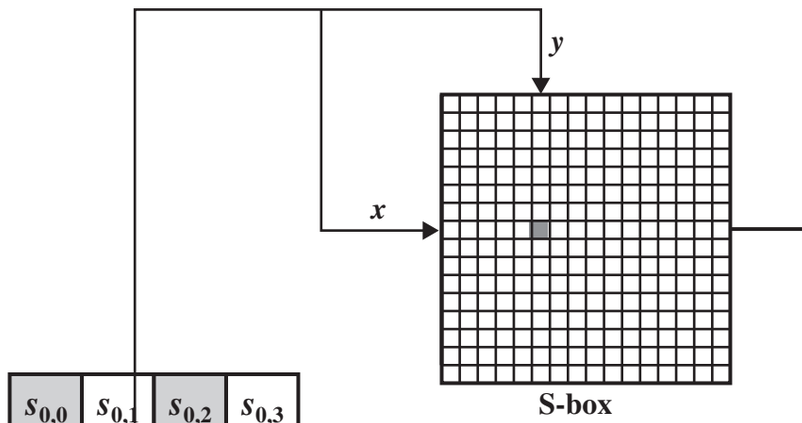
- 其它特点
  - 算法简单
  - 仅在轮密钥加时使用密钥
  - 每个阶段均可逆
  - AES解密算法与加密算法不同
  - 最后一轮只包含前三步

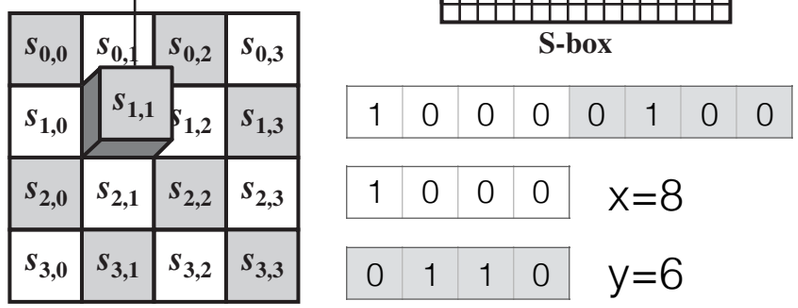


3 一轮加密过程

字节代换



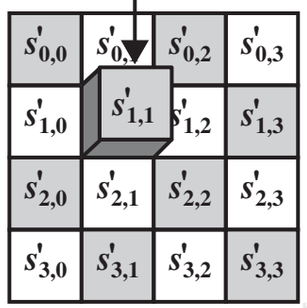




1 0 0 0 0 1 0 0

1 0 0 0 x=8

0 1 1 0 y=6



1	0	0	0	x=8	0	1	1	0	y=6	0	1	0	0	0	1	0	0	s=44

代换

x \ *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

0	1	0	0	0	1	0	0	s=44	1	0	0	0	x=8	0	1	1	0	y=6

逆代换

x \ y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
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

3 一轮加密过程
字节代换

- 怎样求逆代换? —— S-Box构建方式

EA	04	65	85
83	45	5D	96
5C	33	98	B0
F0	2D	AD	C5

→

87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

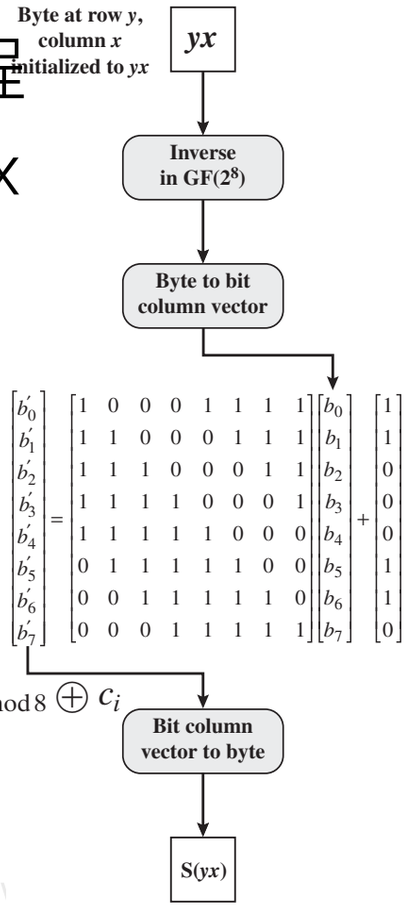
3 一轮加密过程

字节代换-SBox

1. 初始化：构建16*16的盒子。对第y行，第x列的位置，设定初值{yx}
2. 求{yx}在GF(2^8)下的逆
3. 矩阵变换(**可逆**)

$$b_i' = b_i \oplus b_{(i+4) \bmod 8} \oplus b_{(i+5) \bmod 8} \oplus b_{(i+6) \bmod 8} \oplus b_{(i+7) \bmod 8} \oplus c_i$$

5. 将列向量转为字节，并填充至(y,x)

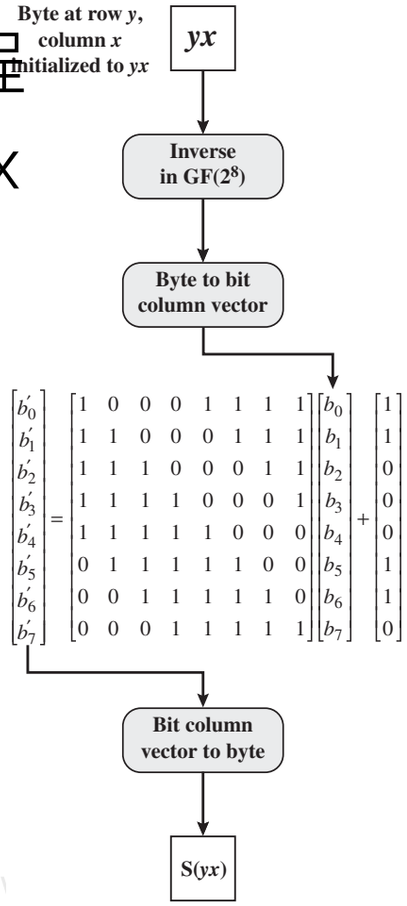


$$\begin{bmatrix} b_0' \\ b_1' \\ b_2' \\ b_3' \\ b_4' \\ b_5' \\ b_6' \\ b_7' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

3 一轮加密过程
字节代换-SBox

例子

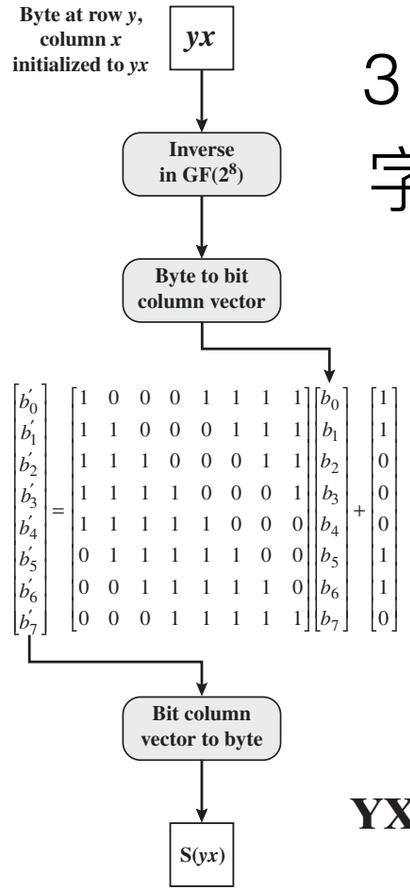
1. 取输入{95} （第9行，第5列）
2. {95}的逆为{8A}={10001010}
3. M*[0 1 0 1 0 0 0 1]'+[1 1 0 0 0 1 1 0]'
4. 结果={2A}



$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

3 一轮加密过程

字节代换-SBox



$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

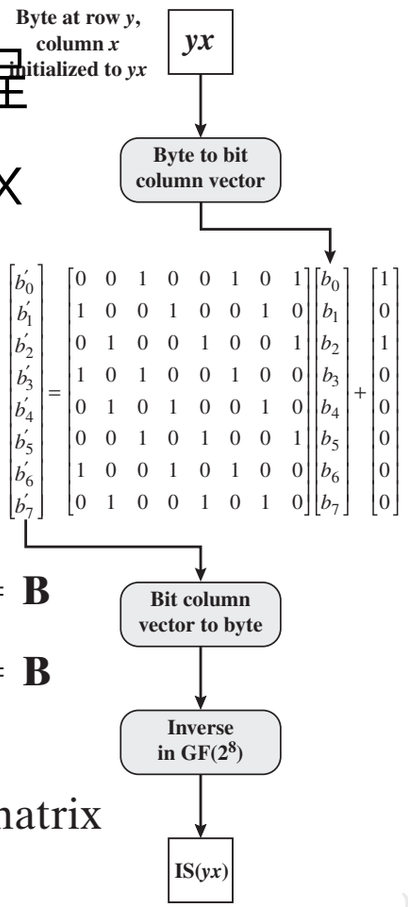
$$\mathbf{B}' = \mathbf{XB} \oplus \mathbf{C}$$

$$\mathbf{Y}(\mathbf{XB} \oplus \mathbf{C}) \oplus \mathbf{D} = \mathbf{B}$$

$$\mathbf{YXB} \oplus \mathbf{YC} \oplus \mathbf{D} = \mathbf{B}$$

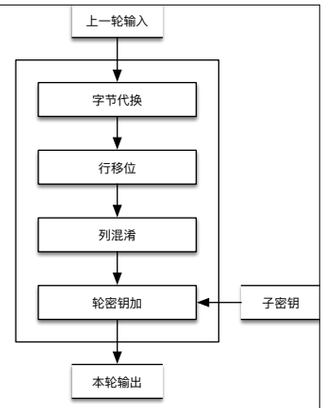
$$\mathbf{YC} = \mathbf{D}$$

YX equals the identity matrix



$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

3 一轮加密过程
字节代换-SBox



- S-Box的特点
  - 防止已有的各种密码分析攻击
  - 输出位和输入位的相关性很低

$$b_i' = b_i \oplus b_{(i+4) \bmod 8} \oplus b_{(i+5) \bmod 8} \oplus b_{(i+6) \bmod 8} \oplus b_{(i+7) \bmod 8} \oplus c_i$$

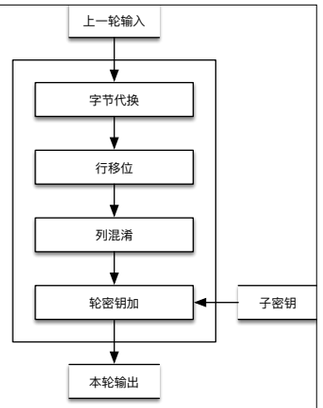
  - 输出值不是输入值的线性数学函数——乘法逆

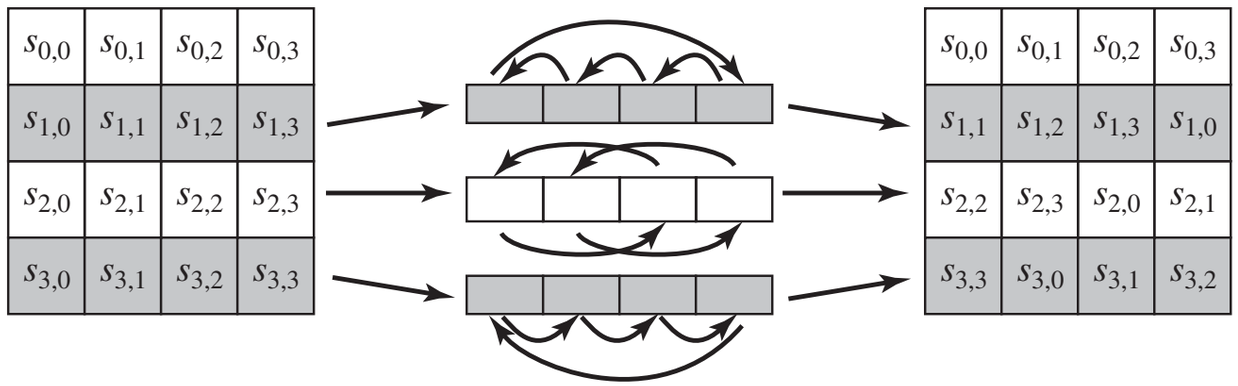
2. {95}的逆为{8A}={10001010}

  - 没有不动点

3 一轮加密过程

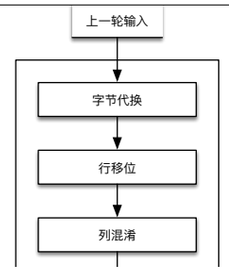
行移位





3 一轮加密过程

列混淆



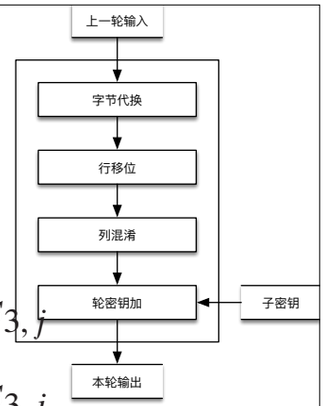
$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \times \begin{bmatrix} \quad \\ \quad \\ \quad \\ \quad \end{bmatrix} = \begin{bmatrix} \quad \\ \quad \\ \quad \\ \quad \end{bmatrix}$$

$s_{0,0}$	$s_{0,1}$	$s_{0,2}$	$s_{0,3}$
$s_{1,0}$	$s_{1,1}$	$s_{1,2}$	$s_{1,3}$
$s_{2,0}$	$s_{2,1}$	$s_{2,2}$	$s_{2,3}$
$s_{3,0}$	$s_{3,1}$	$s_{3,2}$	$s_{3,3}$

$s'_{0,0}$	$s'_{0,1}$	$s'_{0,2}$	$s'_{0,3}$
$s'_{1,0}$	$s'_{1,1}$	$s'_{1,2}$	$s'_{1,3}$
$s'_{2,0}$	$s'_{2,1}$	$s'_{2,2}$	$s'_{2,3}$
$s'_{3,0}$	$s'_{3,1}$	$s'_{3,2}$	$s'_{3,3}$

3 一轮加密过程

列混淆



$$s'_{0,j} = (2 \bullet s_{0,j}) \oplus (3 \bullet s_{1,j}) \oplus s_{2,j} \oplus s_{3,j}$$

$$s'_{1,j} = s_{0,j} \oplus (2 \bullet s_{1,j}) \oplus (3 \bullet s_{2,j}) \oplus s_{3,j}$$

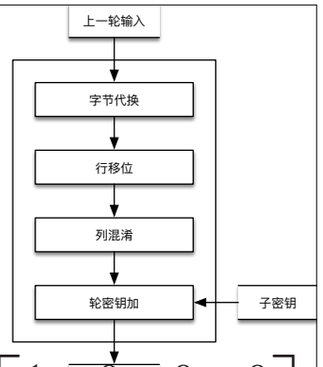
$$s'_{2,j} = s_{0,j} \oplus s_{1,j} \oplus (2 \bullet s_{2,j}) \oplus (3 \bullet s_{3,j})$$

$$s'_{3,j} = (3 \bullet s_{0,j}) \oplus s_{1,j} \oplus s_{2,j} \oplus (2 \bullet s_{3,j})$$

$$\{02\} \cdot A = (a_6 \ldots a_1 a_0 0), \text{if } a_7 = 0$$

$$\{02\} \cdot A = (a_6 \ldots a_1 a_0 0) \oplus (00011011), \text{if } a_7 = 1$$

3 一轮加密过程
列混淆



- 逆变换

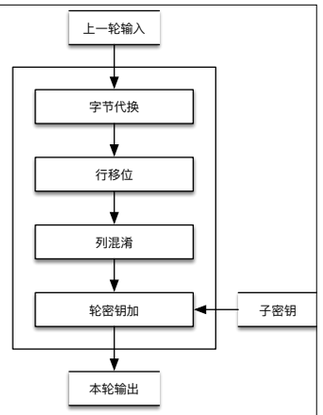
$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix}$$

- 加密和解密的运算复杂度

3 一轮加密过程

轮秘钥加



47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

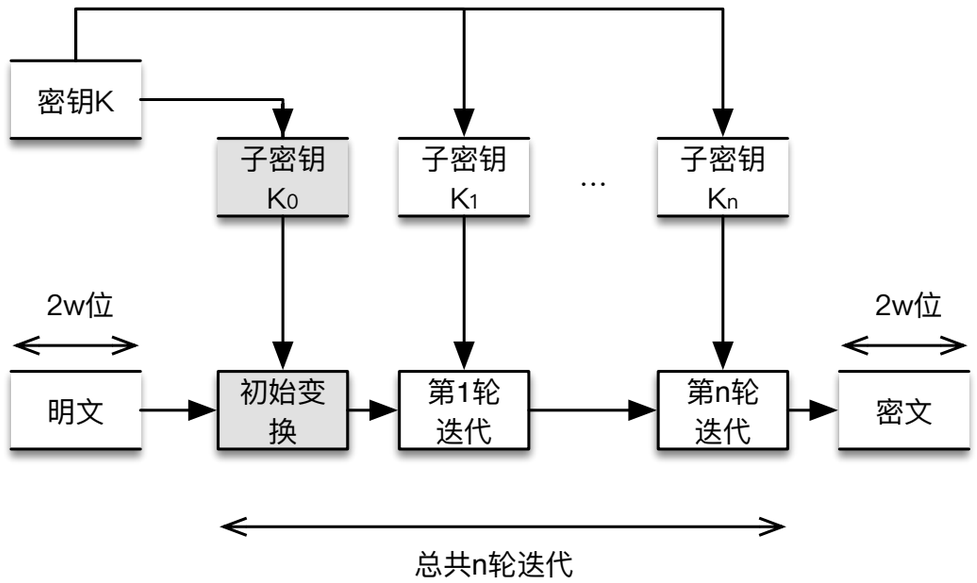
$\oplus$

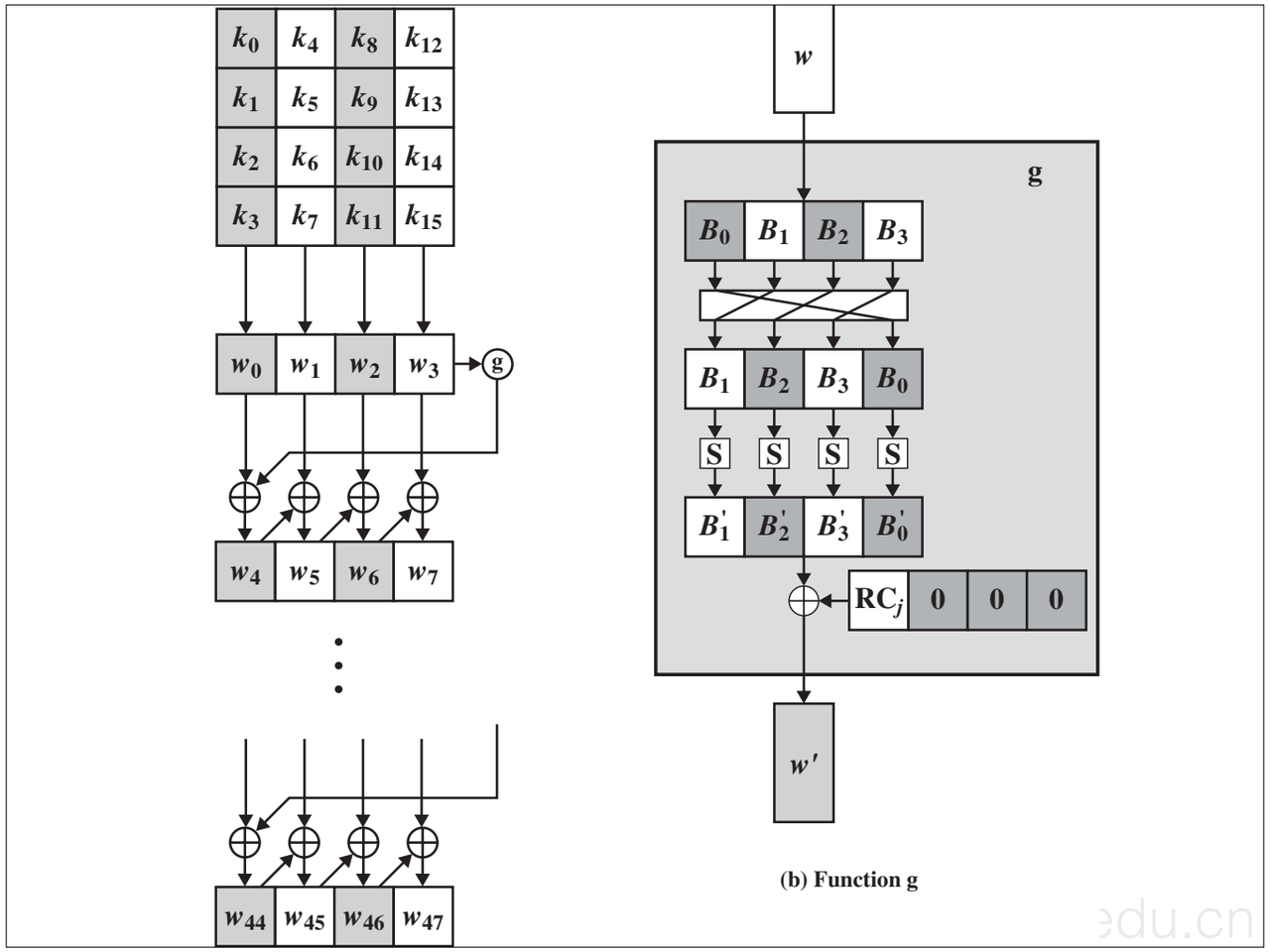
AC	19	28	57
77	FA	D1	5C
66	DC	29	00
F3	21	41	6A

=

EB	59	8B	1B
40	2E	A1	C3
F2	38	13	42
1E	84	E7	D6

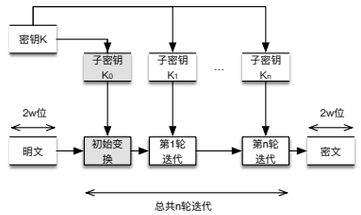
3 一轮加密过程
秘钥扩展算法




k_0 k_4 k_8 k_{12}
k_1 k_5 k_9 k_{13}
k_2 k_6 k_{10} k_{14}
k_3 k_7 k_{11} k_{15}
w_0 w_1 w_2 w_3
g
w_4 w_5 w_6 w_7
w_{44} w_{45} w_{46} w_{47}
w
g
B_0 B_1 B_2 B_3
B_1 B_2 B_3 B_0
S S S S
B'_1 B'_2 B'_3 B'_0
RC_j 0 0 0
w'

(b) Function g

3 一轮加密过程
秘钥扩展算法



- 基本原理
  - 如果已知密钥或轮密钥的部分位，不能推算出其他位
  - 变换可逆（由**任意**连续Nk个字可推出整个扩展密钥）
  - 各种处理器均可使用
  - 使用轮常量RC_j,来消除对称性
  - 密钥的很多位能影响到轮密钥的许多位
  - 足够的非线性
  - 易于描述

4 分析
防御能力

- 雪崩效应
  - 当明文改变1位，大约1半密文被改变
  - 当密钥改变1位，大约1半密文被改变

4 分析
密码学攻击

- **理论**上，复杂度少于brute-force(暴力) 攻击的方法都算成功的攻击
  - 2002年，XSL attack
    - eXtended Sparse Linearization (XSL) attack
    - 用于针对分组密码的攻击方式
    - 具有攻击AES的潜在可能性
    - 不过目前没有找到可行方案

4 分析
密码学攻击-XSL攻击

- XSL攻击方案 （**一种已知明文攻击，**且数量要求较少）
  - 分析密文的内部结构
  - 解析出一套**多元二次方程组**: quadratic simultaneous equations， 如果AES为128位加密，则有
    - 8000个等式
    - 1600个变量
  - 解方程组，求出密钥（eXtended Sparse Linearization）

4 分析
密码学攻击-XSL攻击

- 多元二次方程组求解：
  - NP难问题
  - 1999年，HFE加密方案，可以将方程组分解至Overdetermined system (方程数大于变量数)
    - 思路：Linearization: 将每个二次项转换为一个独立变量，然后求解线性方程组
    - 挑战：方程数太少
    - 方法：**re-Linearization**：linearization后，加入额外的非线性方程，再次线性化

4 分析
密码学攻击-XSL攻击

- 多元二次方程组求解：
  - 2000年，改进方案XL(eXtended Linearization)
    - 思路：增加线性方程的个数
      - 方法：略 (multiplying them with all monomials of a certain degree)
    - 缺陷：针对AES加密无效
    - 改进方案: XSL
      - 思路：利用这些新增方程的特殊结构

4 分析
密码学攻击-XSL攻击

- 产生XSL攻击的主要原因
  - S-box的求逆过程过于简单

The method has some merit, and is worth investigating, but it does not break Rijndael as it stands.

The XSL attack is not an attack. It is a dream.

XSL may be a dream. It may also be a very bad dream and turn into a nightmare.

1. 初始化：构建16*16的盒子。对第y行，第x列的位置，设定初值{yx}
2. **求{yx}在GF(2^8)下的逆**
3. 矩阵变换
4. 将列向量转为字节，并填充至(y,x)

4 分析
密码学攻击-XSL攻击

- 产生XSL攻击的主要原因
  - S-box的求逆过程过于简单

We have one criticism of AES: we don't quite trust the security…

What concerns us the most about AES is its simple algebraic structure…

No other block cipher we know of has such a simple algebraic representation.

1. 初始化：构建16*16的盒子。对第y行，第x列的位置，设定初值{yx}
2. **求{yx}在GF(2^8)下的逆**
3. 矩阵变换
4. 将列向量转为字节，并填充至(y,x)

4 分析
密码学攻击

- 2009年
  - 攻击复杂度：2^{119}
  - 思路：利用密钥扩展算法的简易性
- 2009年12月
  - 攻击复杂度：$2^{99.5}$
- 等等。。。
- 上述攻击并不具备实用性

4 分析
密码学攻击

- 理想的加密算法
  - 抵御所有攻击?
  - 诸多限制
    - 分组大小、密钥大小
    - 理想分组密码

4 分析
旁路(side channel)攻击

- 计时攻击
- 能量分析攻击

4 分析
旁路(side channel)攻击

- 2005.4 Cache-timing attack
  - OpenSSL的 AES实现
  - 需求: 200milllion 的选择明文
- 2005.10 多个cache-timing attack
  - 其中1种攻击的需求: 800个加密操作， 65ms

In April 2005, D.J. Bernstein announced a cache-timing attack that he used to break a custom server that used OpenSSL's AES encryption.[31] The attack required over 200 million chosen plaintexts.[32] The custom server was designed to give out as much timing information as possible (the server reports back the number of machine cycles taken by the encryption operation); however, as Bernstein pointed out, "reducing the precision of the server's timestamps, or eliminating them from the server's responses, does not stop the attack: the client simply uses round-trip timings based on its local clock, and compensates for the increased noise by averaging over a larger number of samples."[31]

In October 2005, Dag Arne Osvik, Adi Shamir and Eran Tromer presented a paper demonstrating several cache-timing attacks against AES.[33] One attack was able to obtain an entire AES key after only 800 operations triggering encryptions, in a total of 65 milliseconds. This attack requires the attacker to be able to run programs on the same system or platform that is performing AES.

4 分析
旁路(side channel)攻击

- 2009年
  - 针对某些硬件实现的攻击
  - Differential fault analysis
  - 复杂度2^{32}
- 2010年
  - AES 128位破解
  - 特点：不需要明文或密文， 接近实时地破解
  - 需求： 在进行加密的机器上运行**无特权**的程序

In November 2010 Endre Bangerter, David Gullasch and Stephan Krenn published a paper which described a practical approach to a "near real time" recovery of secret keys from AES-128 without the need for either cipher text or plaintext. The approach also works on AES-128 implementations that use compression tables, such as OpenSSL.[35] Like some earlier attacks this one requires the ability to run unprivileged code on the system performing the AES encryption, which may be achieved by malware infection far more easily than commandeering the root account.[36]

In November 2010 Endre Bangerter, David Gullasch and Stephan Krenn published a paper which described a practical approach to a "near real time" recovery of secret keys from AES-128 without the need for either cipher text or plaintext. The approach also works on AES-128 implementations that use compression tables, such as OpenSSL.[35] Like some earlier attacks this one requires the ability to run unprivileged code on the system performing the AES encryption, which may be achieved by malware infection far more easily than commandeering the root account.[36]

4 分析
旁路(side channel)攻击

- 2016年
  - 攻击128 bit AES加密
  - 需求：
    - 6-7个明文/密文对
    - 标准用户权限
    - 运行1分钟时间

4 分析
旁路(side channel)攻击

- 防范方式：
  - CPU 专用的AES指令集
  - 等等

Many modern CPUs have built-in hardware instructions for AES, which would protect against timing-related side-channel attacks.[38][39]

作业

- 无