密 码 学

第四章 分组密码

网络空间安全学院
胡 伟　朱丹
weihu/zhudan@nwpu.edu.cn

知识回顾 – AES概况

- 1997年美国国家标准与技术研究所（NIST）向社会公开征集高级数据加密标准（AES, Advanced Encryption Standard）
- 基本要求：比3-DES快，至少和3-DES一样安全，数据分组长度为128位，密钥长度支持128/192/256位
  - 第一轮：1998年8月20日从应征的21个算法中选出15个
  - 第二轮：1999年8月又选出其中5个候选算法（RC6，Rijndael，SERPENT，Twofish和MARS）
  - 第三轮：2000年10月2日再选出1个算法（Rijndael）
- 2001年11月26日NIST接受其作为标准
- 2001年12月4日正式公布为联邦标准：FIPS - 197

AES英文全称

AES设计要求：安全性（由密钥决定）、分组长度、密钥长度

知识回顾 – AES特点

- 分组密码：明文和密文分组长度为128位，密钥长度可为128/192/256位
- 基本轮函数迭代，轮数可为10/12/14（与密钥长度对应）
- 整体结构：S-P网络结构
- 综合运用多种密码技术：置换、替代、代数
- 不是对合运算：加解密算法存在差异
- 与DES类似，属于面向二进制的密码：便于计算机实现

- AES密码算法的参数：分组长度、密钥长度、轮数、网络结构
- AES算法不是对合结构

知识回顾 – AES数学基础

- AES基于有限域GF(2^8)
- 有限域GF(2^8)上元素的GF(2)多项式表示
  - 字节$B = b_7b_6b_5b_4b_3b_2b_1b_0$可表示成GF(2)上的多项式

$$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0$$

  - 例如，0x57对应的二进制数为，01010111
  - 相应的多项式为$x^6 + x^4 + x^2 + x + 1$

知识回顾 – AES数学基础

- 有限域GF(2^8)上的加法
  - 对应多项式系数的模2加（异或）
  - 结果仍为GF(2^8)上的元素（次数不超过7的多项式）
  - 例，0x57 + 0x83 = ?

$$01010111 \oplus \quad 10000011 = 11010100$$

$$(x^6+x^4+x^2+x+1) \oplus \ (x^7+x+1) = x^7+x^6+x^4+x^2$$

0x57 + 0x83 = 0xD4

知识回顾 – AES数学基础

- 有限域GF(2^8)上的乘法
  - 先确定一个8次不可约多项式$m(x)$
  - AES选择$m(x) = x^8+x^4+x^3+x+1$，其16进制表示为0x11B
  - 多项式乘法对$m(x)$取模，结果仍为GF(2^8)上的元素（次数不超过7的多项式）
  - 例，0x57×0x83 = ?

$$(x^6+x^4+x^2+x+1)\otimes (x^7+x+1) = x^7+x^6+1 \bmod m(x)$$

0x57 × 0x83 = 0xC1

有限域上的乘法：多项式乘法展开，然后对$m(x)$取模，保证结果的次数不超过7，任然在GF(2^8)上加法或者减法均为异或操作

知识回顾 – AES数学基础

- 有限域GF(2^8)上的x乘法($x$time)，定义为

$$x \otimes (b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0)$$
$$= b_7x^8 + b_6x^7 + b_5x^6 + b_4x^5 + b_3x^4 + b_2x^3 + b_1x^2 + b_0x$$

- 计算规则：
  - 若$b_7 = 0$，次数不超过7，直接得到结果
  - 否则，乘法结果减去$m(x)$，即与$m(x)$做异或
- 通过系数直接计算
  - $B = b_7b_6b_5b_4b_3b_2b_1b_0$左移一位，最低位补0（乘2）
  - 若$b_7 = 0$，直接得到结果
  - 否则，$b_6b_5b_4b_3b_2b_1b_00$再与0x1B做异或
  - x的更高次的乘法可以重复应用$x$time实现

有限域上的乘法：多项式乘法展开，然后对$m(x)$取模，保证结果的次数不超过7，任然在GF(2^8)上加法或者减法均为异或操作

知识回顾 – AES数学基础

- AES数据处理的单位是字节（byte）、字（word）和状态（state）
  - 一个字 = 4个字节 = 32位，状态为128位
  - 一个字可表示为系数取自GF(2^8)上的次数低于4次的多项式
  - 例，字：57 83 4A D1 -- $57x^3 + 83x^2 + 4Ax + D1$

- 字加法：两多项式系数按位模2加
  - 例，$(57x^3 + 83x^2 + 4Ax + D1) + (Ax^3 + B3x^2 + EF)$
    $= 5Dx^3 + 30x^2 + 4Ax + 3E$

- 状态（128位）
  - 加解密过程中的中间数据
  - 以字节为元素的矩阵或二维数组

状态矩阵

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

知识回顾 – AES数学基础

- 字乘法：设a和c是两个字，$a(x)$和$c(x)$为对应的字多项式，AES定义a和c的乘积b为

$$b(x) = a(x)c(x) \bmod x^4 + 1$$

- 假设

$$a(x) = a3x^3 + a2x^2 + a1x + a0$$
$$c(x) = c3x^3 + c2x^2 + c1x + c0$$
$$b(x) = b3x^3 + b2x^2 + b1x + b0$$

- 则，$b(x)=a(x)c(x) \bmod x^4 + 1$为

$$b0 = a0c0 + a3c1 + a2c2 + a1c3$$ 四次项和常量
$$b1 = a1c0 + a0c1 + a3c2 + a2c3$$ 一次项
$$b2 = a2c0 + a1c1 + a0c2 + a3c3$$ 二次项
$$b3 = a3c0 + a2c1 + a1c2 + a0c3$$ 三次项

- 模多项式是4次多项式$x^4 + 1$，因此，相乘结果为4次的项取模约简为常量
- b0~b3分别对应取模约简后的常量至3次项

知识回顾 – AES数学基础

- 字乘法的矩阵表示
- $x^4 + 1 = (x^2 + 1)(x^2 + 1)$，是可约多项式，字$c(x)$不一定存在逆元
- $c(x)$存在逆元的条件是 $(x^4+1, c(x)) = 1$
- AES选择的$c(x)$有逆，$c(x) = 03x^3 + 01x^2 + 01x + 02$

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} c_0 & c_3 & c_2 & c_1 \\ c_1 & c_0 & c_3 & c_2 \\ c_2 & c_1 & c_0 & c_3 \\ c_3 & c_2 & c_1 & c_0 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

- 将上页的多项式改写为矩阵形式
- AES选择的c(x)是固定的

知识回顾 – AES数学基础

- 字的x乘法：设$b(x)$是一个字，

$$\begin{aligned} p(x) &= xb(x) \bmod x^4 + 1 \\ &= b_3x^4 + b_2x^3 + b_1x^2 + b_0x \bmod x^4 + 1 \\ &= b_2x^3 + b_1x^2 + b_0x + b_3 \end{aligned}$$

- 因为模$x^4 + 1$，字的x乘法相当于按字节循环移位

- 写成矩阵形式

$$\begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{pmatrix} = \begin{pmatrix} 00 & 00 & 00 & 01 \\ 01 & 00 & 00 & 00 \\ 00 & 01 & 00 & 00 \\ 00 & 00 & 01 & 00 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

字的x乘法，相当于按照字节循环移位

练 习 题

- $x$time是AES密码算法中的基本运算，$x\text{time}(2\text{A}) = \_\_$，$x\text{time}(86) = \_\_$。
- 选择$m(x) = x^8+x^4+x^3+x+1$，计算有限域GF(2^8)上的乘法87×15。

章节安排

Outline

AES算法概述

AES数学基础

AES加解密算法

AES密钥扩展算法

AES安全性分析

4.9(1) AES的处理方式

- 常用符号
  - N_b：明密文所含的字数（$N_b = 4$）
  - N_k：密钥所含的字数 （$N_k = 4, 6, 8$）
  - N_r：迭代轮数（$N_r = 10, 12, 14$）

$N_b = 4$时，状态矩阵为

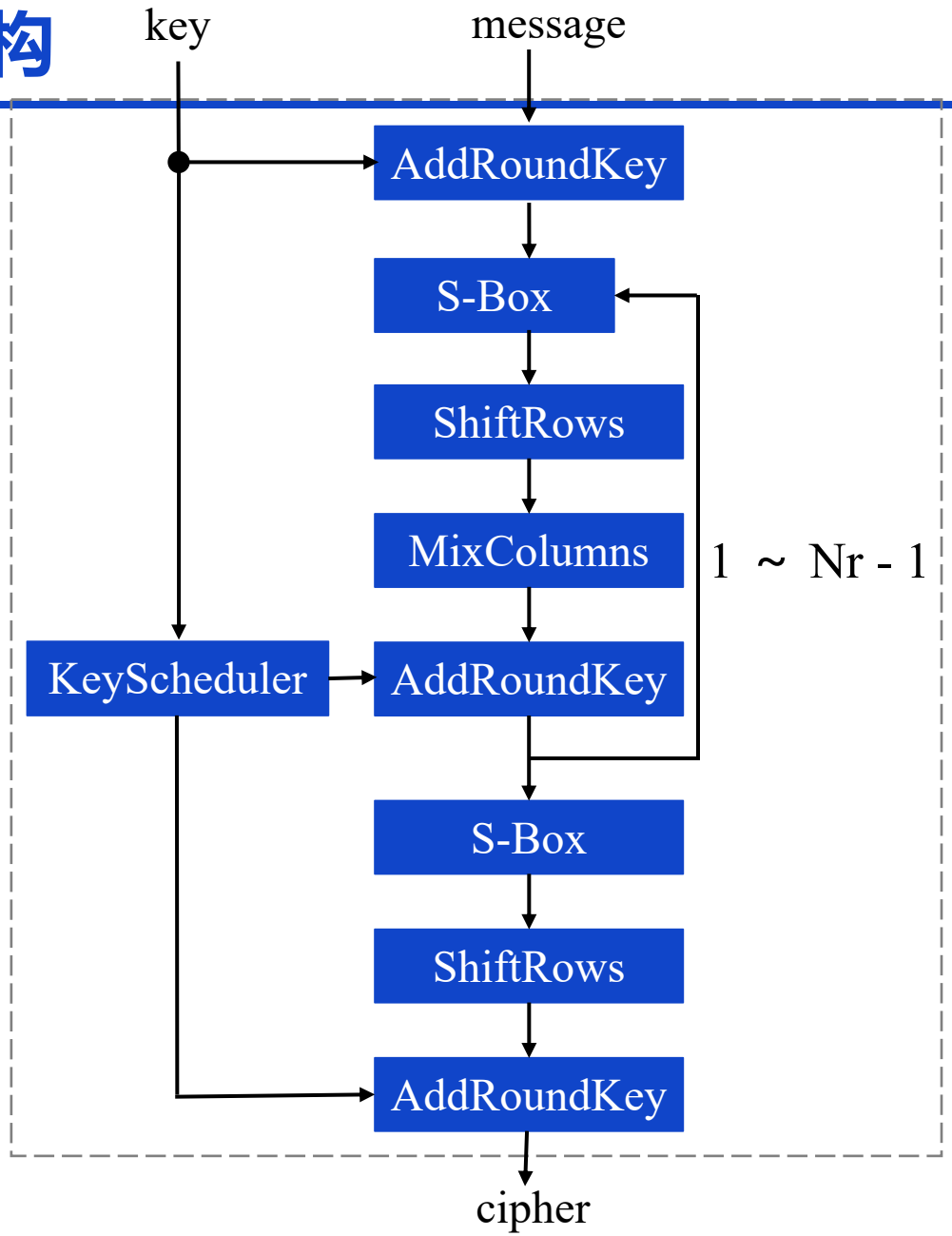
$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$

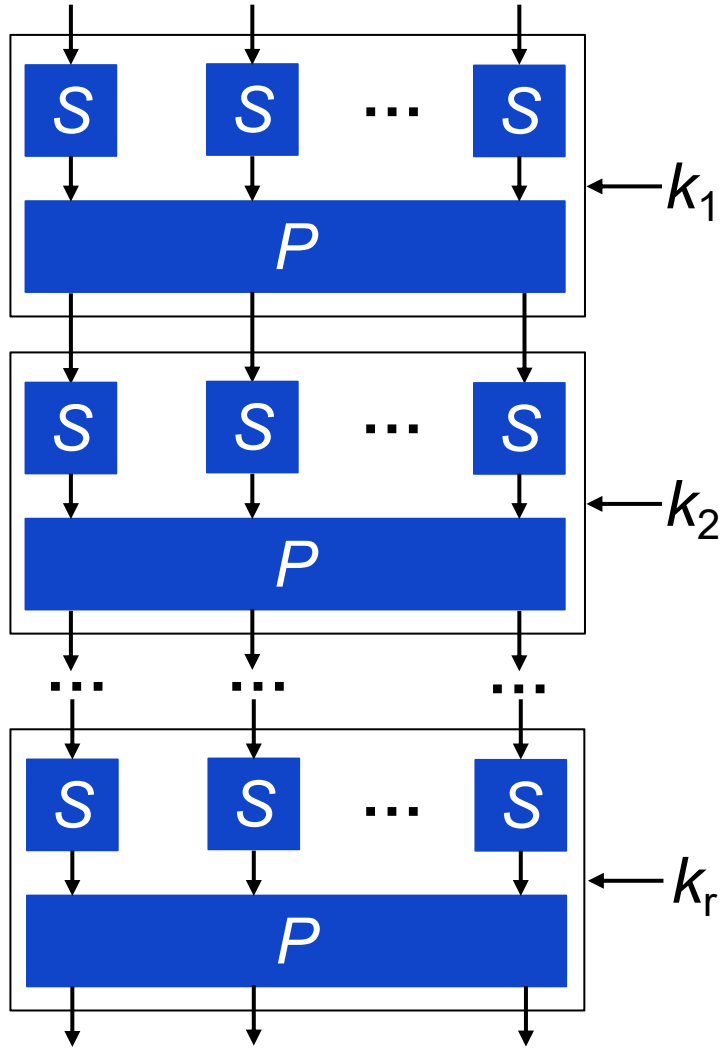
$N_k = 4$时，密钥矩阵为

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

4.9(2) AES的基本结构

- S盒变换 – S-Box
- 行移位 – ShiftRows
- 列混合 – MixColumns
- 密钥扩展 – KeyScheduler
- 加轮密钥 – AddRoundKey

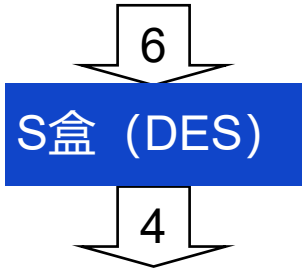


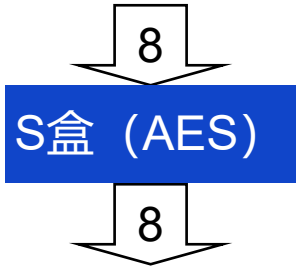


4.9(2) AES的基本结构-S盒

- AES的S盒变换 SubByte (以字节为单位)
- S盒变换是AES的唯一的非线性变换，是AES安全的关键

	S盒数量	S盒规模	S盒功能	各S盒是否相同
DES	8	6输入4输出	非线性压缩	不同
AES	16	8输入8输出	非线性置换	相同





- AES的S盒对其安全性的作用
- AES和DES算法S盒特点的对比

4.9(2) AES的基本结构-S盒

- S盒变换的特点
  - 把输入字节看成GF(2^8)上的元素
  - 求出其在GF(2^8)上的逆元素，00变换为自身

设$a(x)$的逆元为$b(x)$，

则 $a(x)b(x) = 1 \bmod m(x)$

其中，$m(x) = x^8+x^4+x^3+x+1$

- 第二步：在GF(2)上对上面的结果作如下的仿射变换

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

AES的S盒通过有限域运算的实现方式：求逆元+仿射变换

4.9(2) AES的基本结构-S盒

- 第一步把字节的值用它的乘法逆来代替，是一种非线性变换
- 第二步是仿射运算，是线性变换
- 由于系数矩阵中每列都含有 5个1，这说明改变输入中的任意一位，将影响输出中的5位
- 由于系数矩阵中每行都含有 5个1，这说明输出中的任意一位，都与输入中的5位相关

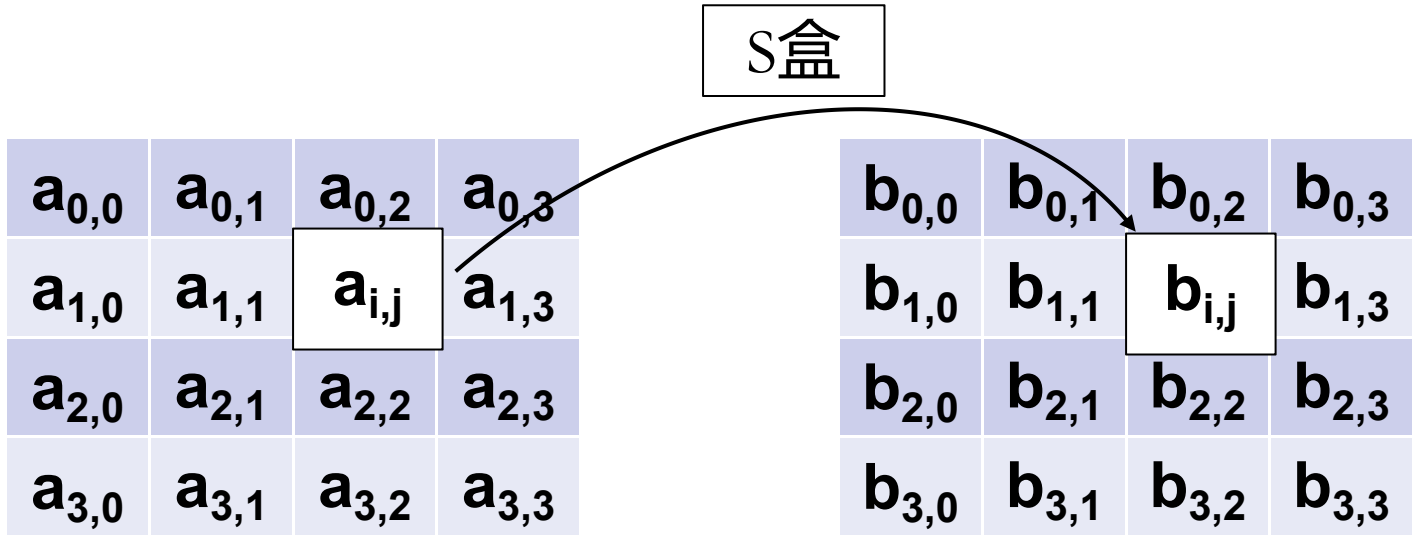
$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\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

AES S盒的一些特性，体现非线性、混淆

4.9(2) AES的基本结构-S盒

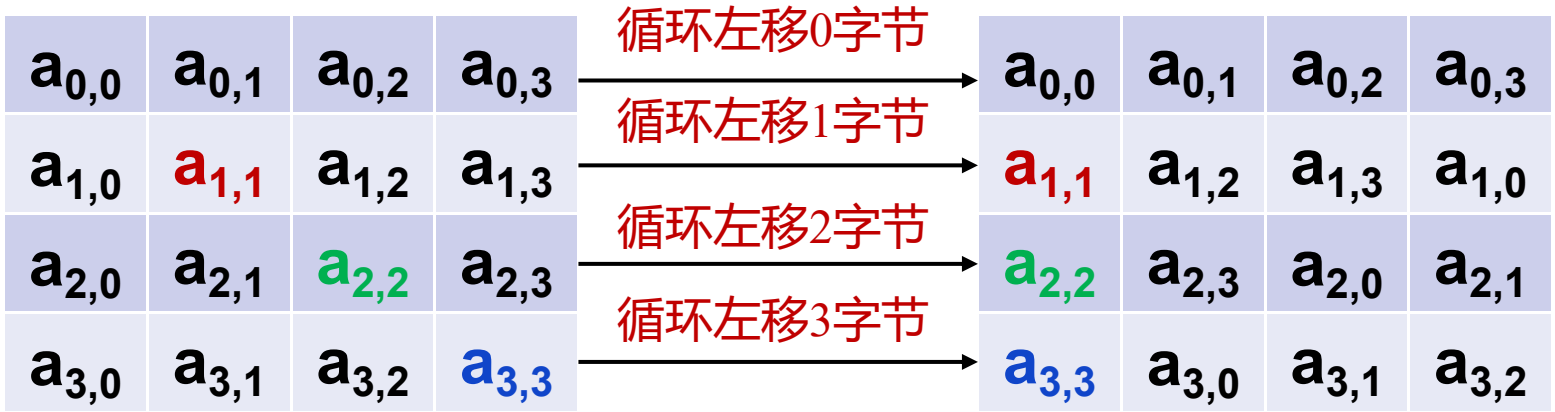
- 通常采用查找表来实现
  - 256字节数组
  - case语句

	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



4.9(2) AES的基本结构-行移位

- 行移位变换 **ShiftRow(128位)**
  - 行移位变换对状态矩阵的行进行循环左移
  - 第 0行不移位，第1行移 1字节，第2行移 2字节，第3行移3字节
  - 行移位变换属于置换，属于线性变换，本质在于把数据打乱重排，起扩散作用



行移位是字节级别的循环左移

4.9(2) AES的基本结构-列混合

- 列混合变换 MixColumn (128位)，属于线性变换，起扩散作用
- 把状态的列视为GF(2^8)上的多项式$a(x)$，乘以一个固定的多项式$c(x)$，然后模x^4+1：

$$b(x) = a(x)c(x) \bmod x^4 + 1$$

- 其中，$c(x) = 03x^3 + 01x^2 + 01x + 02$
- $c(x)$与$x^4 + 1$互素，从而保证$c(x)$存在逆多项式$d(x)$，使得$c(x)d(x) = 1 \mod x^4+1$
- 只有逆多项式$d(x)$存在，才能正确进行解密

- 列混合的特性：线性置换、起到扩散作用
- 列混合的字乘法定义

4.9(2) AES的基本结构-列混合

AES数学基础

- 字乘法：设a和c是两个字，$a(x)$和$c(x)$为对应的字多项式，AES定义a和c的乘积b为

$$b(x) = a(x)c(x) \bmod x^4 + 1$$

- 假设

$$a(x) = a3x^3 + a2x^2 + a1x + a0$$
$$c(x) = c3x^3 + c2x^2 + c1x + c0$$
$$b(x) = b3x^3 + b2x^2 + b1x + b0$$

- 则，$b(x)=a(x)c(x) \bmod x^4 + 1$为

$$b0 = a0c0 + a3c1 + a2c2 + a1c3$$ 四次项和常量
$$b1 = a1c0 + a0c1 + a3c2 + a2c3$$ 一次项
$$b2 = a2c0 + a1c1 + a0c2 + a3c3$$ 二次项
$$b3 = a3c0 + a2c1 + a1c2 + a0c3$$ 三次项

4.9(2) AES的基本结构-列混合

- 列混合变换 MixColumn (128位)，属于线性变换，起扩散作用
- 把状态的列视为GF(2^8)上的多项式$a(x)$，乘以一个固定的多项式$c(x)$，并模 x^4+1：

$$b(x) = a(x)c(x) \bmod x^4 + 1$$

- 其中，$c(x) = c_3x^3 + c_2x^2 + c_1x + c_0 = 03x^3 + 01x^2 + 01x + 02$
- 写成矩阵形式

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

$$b_0 = a_0c_0 + a_3c_1 + a_2c_2 + a_1c_3$$
$$b_1 = a_1c_0 + a_0c_1 + a_3c_2 + a_2c_3$$
$$b_2 = a_2c_0 + a_1c_1 + a_0c_2 + a_3c_3$$
$$b_3 = a_3c_0 + a_2c_1 + a_1c_2 + a_0c_3$$

四次项和常量

一次项

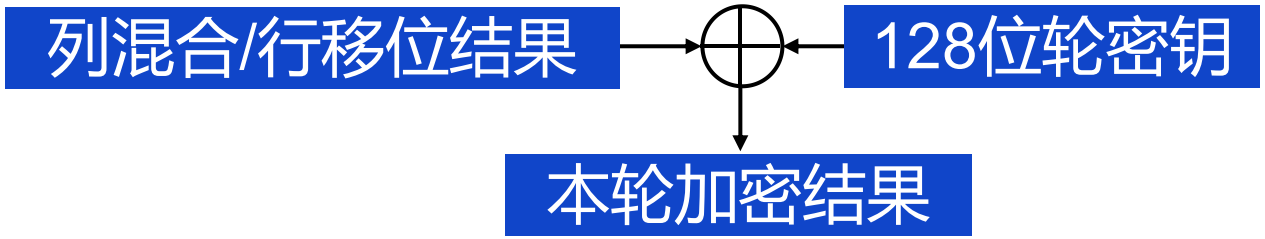
二次项

三次项

- 列混合的特性：线性置换、起到扩散作用
- 列混合的数学定义：字乘法

4.9(2) AES的基本结构-加轮密钥

- 轮密钥加变换AddRoundKey(128位)
  - 把轮密钥与状态进行模2加
  - 轮密钥根据密钥产生算法产生
  - 轮密钥长度等于数据分组长度



轮密钥加是模2加，即异或运算

4.9(3) AES逆变换

- AES不是对合运算，解密算法与加密算法不同

$$(AES)^{-1} \neq AES$$

- AES的巧妙之处：虽然解密算法与加密算法不同，但是解密算法与加密算法的结构相同
- 把加密算法的基本运变换成逆变换，便得到解密算法，密钥扩展策略稍有不同

- AES的结构特点：不是对合结构，但是加解密结构可以是对称的
- AES加解密中存在正变换和逆变换的概念

4.9(3) AES逆变换

- 轮密钥加变换的逆就是其本身

$$(AddRoundKey)^{-1} = AddRoundKey$$

- 行移位变换的逆是状态的后三行分别循环左移3，2，1个字节（或循环右移1，2，3个字节）

- 列混合变换把状态的每一列都乘以一个多项式$c(x)$：

$$b(x) = a(x)c(x) \bmod x^4 + 1$$

- 列混合变换的逆就是状态的每列都乘以$c(x)$的逆多项式$d(x)$：

$$d(x) = (c(x))^{-1} \bmod x^4 + 1$$

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

$$d(x) = 0Bx^3 + 0Dx^2 + 09x + 0E$$

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

- 轮密钥加的逆运算是它本身；行移位的逆运算是按照一定规则移回来
- 列混合的逆运算是在正变换的两边乘以c(x)的逆多项式，字乘法的逆，也可写成矩阵乘法形式

4.9(3) AES逆变换

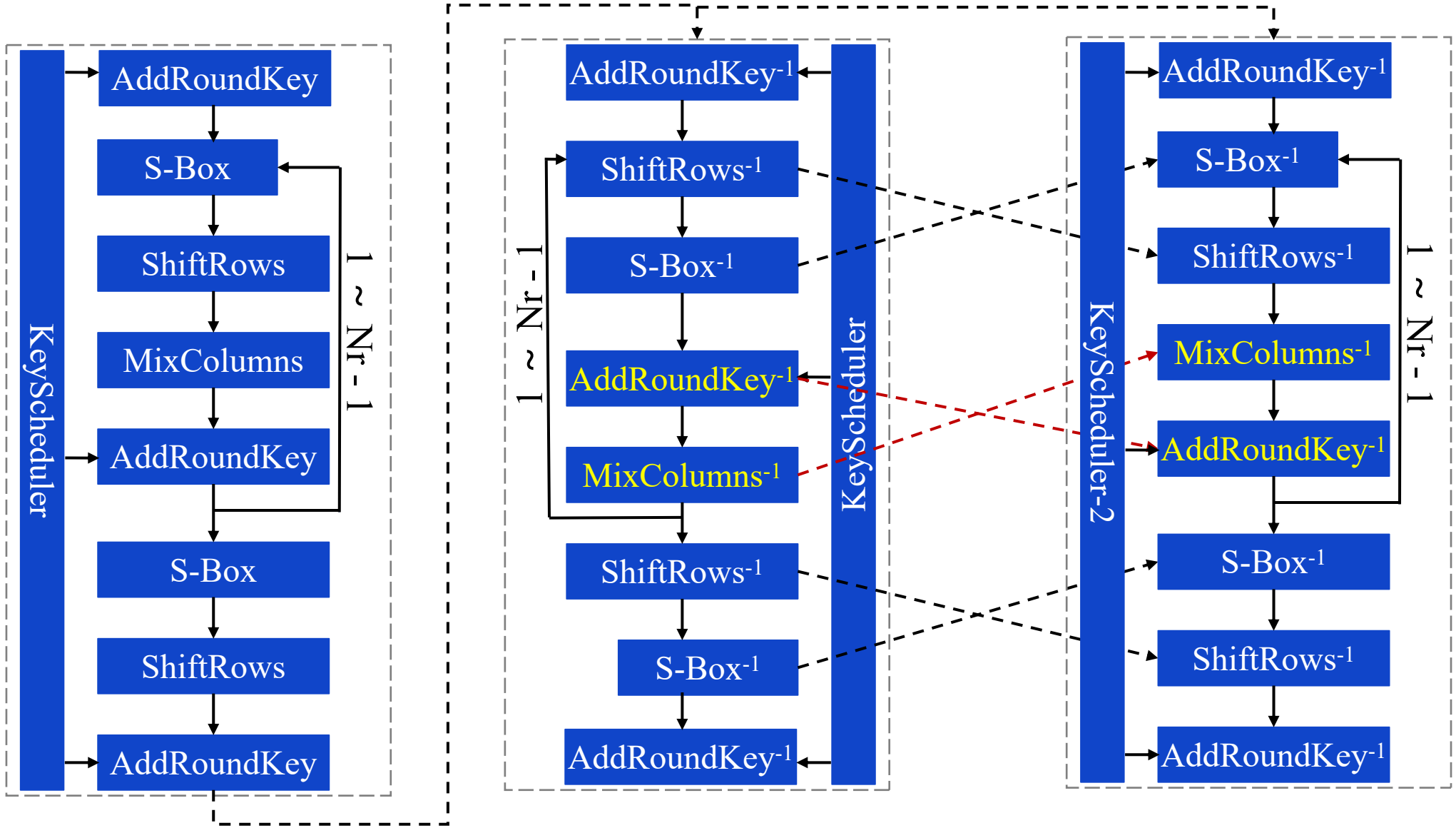
- S盒变换的逆
- 第一步：首先进行逆仿射变换
- 第二步：再把每个字节用其在GF(2^8)中的逆来代替

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

S盒的逆，根据S盒在有限域上的定义，把计算过程逆过来，先做逆仿射变换，再求逆元

4.9(4) AES解密算法

AES Decryption



- AES解密的两种结构
- 为什么AES加密和解密过程是可以对称的

章节安排

Outline

AES算法概述

AES数学基础

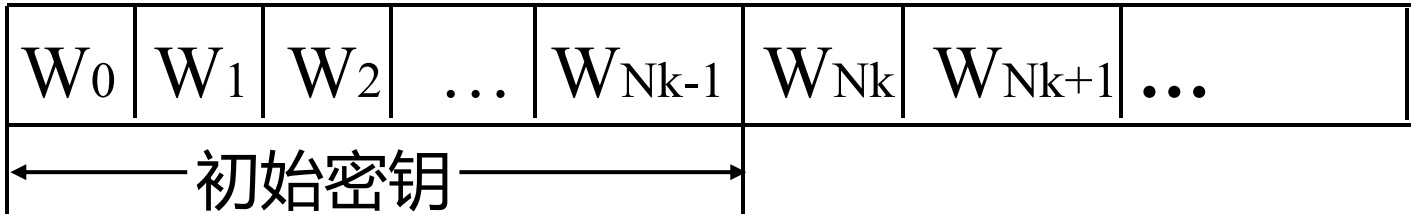
AES加解密算法

AES密钥扩展算法

AES安全性分析

4.10(1) AES密钥扩展算法

- 密钥扩展（$N_k \leq 6$ 的密钥扩展）
  - 最前面的N_k个字由用户密钥填充
  - 之后每个字W[j]等于W[j-1]与N_k个位置之前的字W[j - N_k]的异或
  - 对于N_k的整数倍的位置处的字，在异或之前，对W[j-1]进行Rotl变换和ByteSub变换，再异或一个轮常数Rcon



当 j 是N_k的整数倍时: $W_j = W_{j\text{-}Nk} \oplus \text{ByteSub}(\text{Rotl}(W_{j-1})) \oplus \text{Rcon}[j/N_k]$

否则： $W_j = W_{j\text{-}Nk} \oplus W_{j-1}$

- 初始密钥先按字填充
- 当前密钥字和两个字相关；根据当前密钥字位置下标与Nk的关系，确定扩展规则

4.10(1) AES密钥扩展算法

- Rotl是一个字里的字节循环左移函数，设

$$W = (A, B, C, D),$$

$$\text{Rotl}(W) = (B, C, D, A)$$

- 轮常数Rcon与N_k无关，且定义为：
  - Rcon[i] = (RC [i], ‘00’, ‘00’, ‘00’)
  - RC[0] = ‘01’
  - RC[i] = $x$time(RC[i-1])

- Rotl是循环左移一个字节
- 轮常数Rcon主要是RC，依次是01，02，04，08……每次倍乘

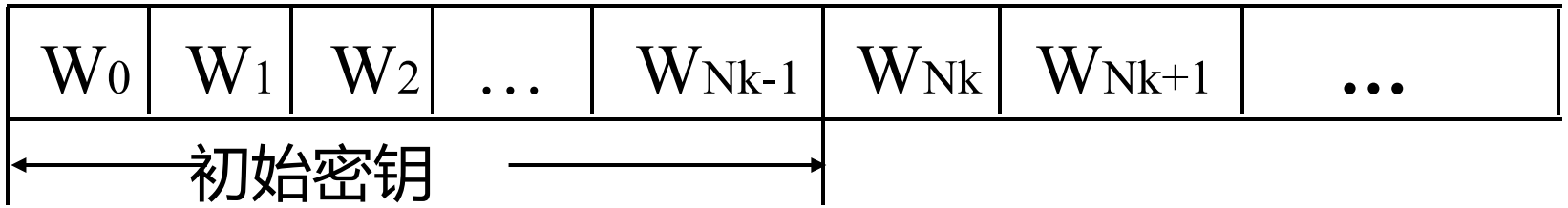
4.10(1) AES密钥扩展算法

- $N_k > 6$ 的密钥扩展
  - 增加：$N_k > 6$的密钥扩展与$N_k \leq 6$的密钥扩展不同之处在于：如果*j*被N_k除的余数为4，则在异或之前，对W[j-1]进行SubBytes变换
  - 当$N_k > 6$时密钥很长，仅仅对N_k的整数倍的位置处的字进行SubBytes变换，就显得 SubBytes 变换的密度较稀，安全程度不够强

Nk为8时，轮密钥的数量较多，需要增加S盒运算的密度，以提升密钥的安全性（涉及到密钥比特的分布和随机性）

4.10(1) AES密钥扩展算法

- $N_k > 6$ 的密钥扩展
  - 增加：$N_k > 6$的密钥扩展与$N_k \leq 6$的密钥扩展不同之处在于：如果j被N_k除的余数为4，则在异或之前，对W[j-1]进行SubBytes变换



当 j 是N_k的整数倍时: $W_j = W_{j\text{-}Nk} \oplus \text{ByteSub}(\text{Rotl}(W_{j-1})) \oplus \text{Rcon}[j/N_k]$

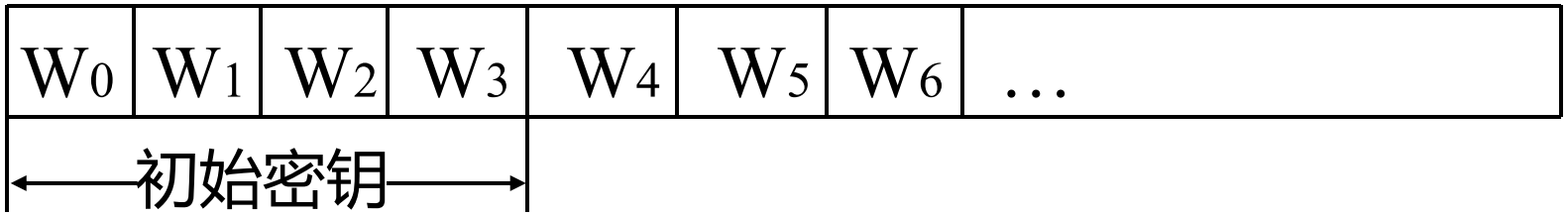
当j被N_k除的余数为4: $W_j = W_{j\text{-}Nk} \oplus \text{ByteSub}(W_{j-1})$

否则: $W_j = W_{j\text{-}Nk} \oplus W_{j-1}$

- 初始密钥先按字填充
- 当前密钥字和两个字相关；根据当前密钥字位置下标与Nk的关系，确定扩展规则；注意增加了一条规则

4.10(1) AES密钥扩展算法

- 举例：$N_k = 4$



当 $j = 5$时，j不是$N_k = 4$的整数倍：

$$W_5 = W_1 \oplus W_4$$

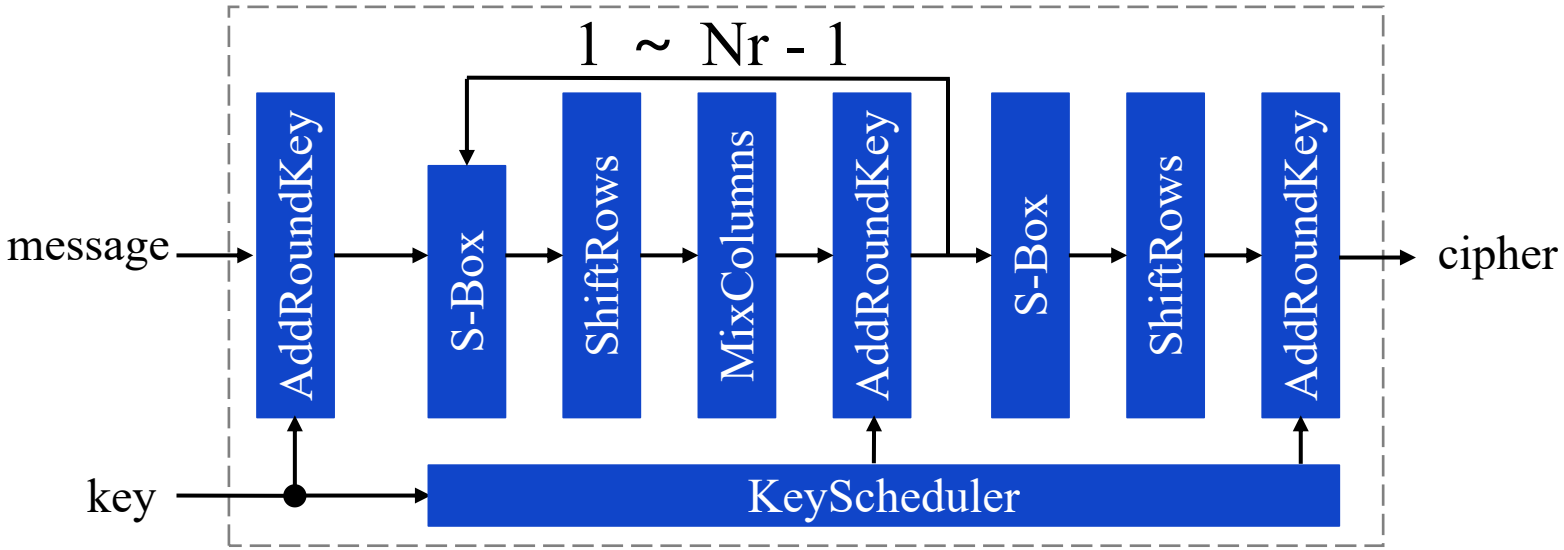
当 $j = 4$时，j是$N_k = 4$的整数倍：

$$W4 = W0 \oplus \text{ByteSub}(\text{Rotl}(W3)) \oplus \text{Rcon}[1]$$

- 初始密钥先按字填充
- 当前密钥字和两个字相关；根据当前密钥字位置下标与Nk的关系，确定扩展规则

4.10(3) AES解密密钥扩展算法

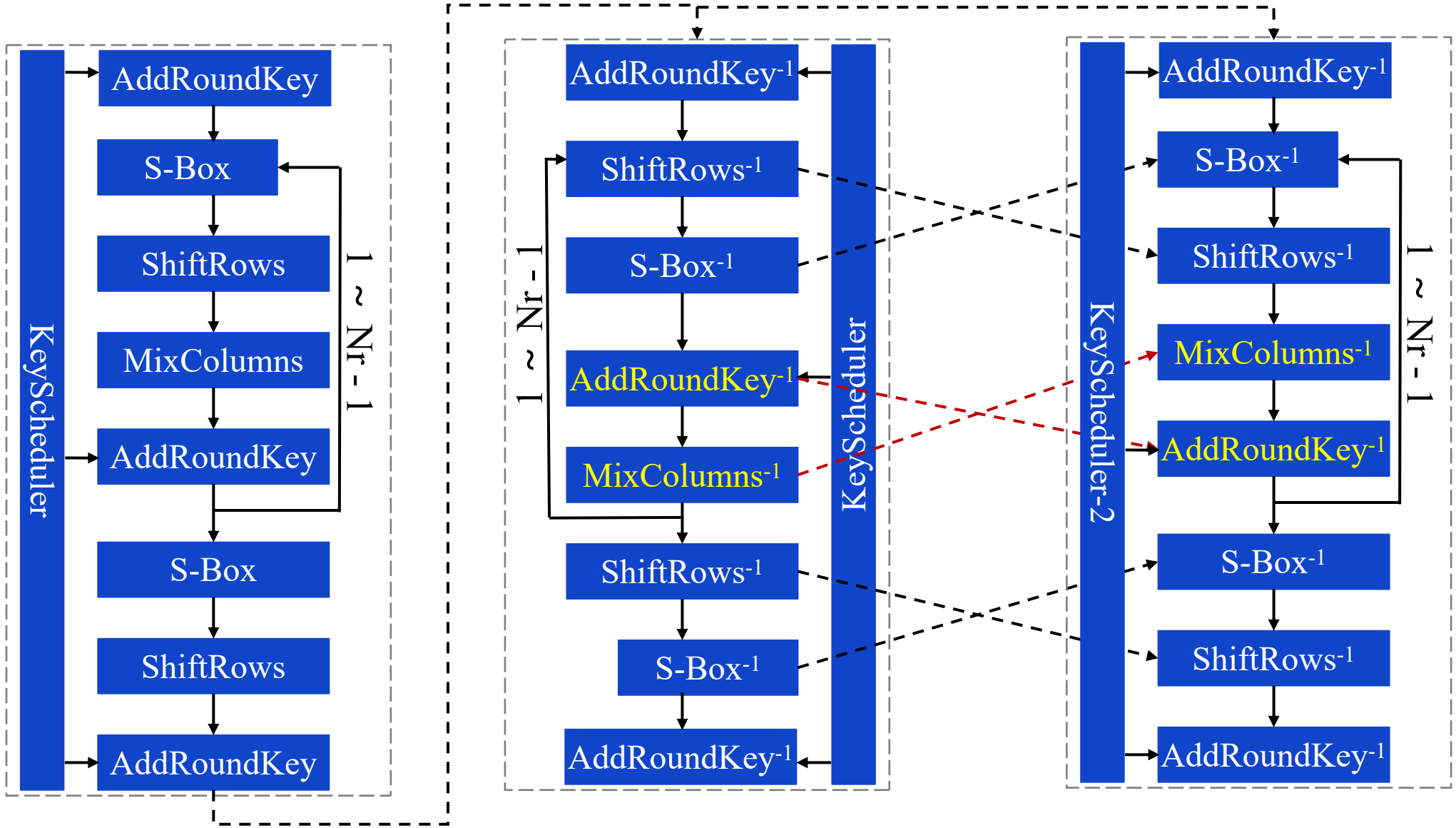
- 第一种解密方法解密的密钥扩展与加密的密钥扩展
- 第二种解密方法解密的密钥扩展与加密的密钥扩展不同，定义如下
  - 加密算法的密钥扩展
  - 把InvMixColumn应用到除第一和最后一轮外的所有轮密钥



InvMixColumn应用到除第一和最后一轮外的所有轮密钥，这里指的是采用与加密完成对称结构的解密算法时

4.10(2) AES解密算法

AES Decryption



- AES解密的两种结构
- 为什么AES加密和解密过程是可以对称的，此时，需将InvMixColumn应用到除第一和最后一轮外的所有轮密钥

章节安排

Outline

AES算法概述

AES数学基础

AES加解密算法

AES密钥扩展算法

AES安全性分析

4.11 AES安全性

- 安全性：
  - AES仍然是目前主流的数据加密标准
  - AES主要的安全威胁来源于侧信道攻击
  - 主要是能量侧信道、电磁侧信道和故障注入攻击
  - 无弱密钥，128位AES的密钥空间可达2^{128}

课 后 作 业

- FIPS 197 Advance Encryption Standard (AES), https://www.nist.org/nist_plugins/content/content.php?content.39
- Mini-AES, https://doc.sagemath.org/html/en/reference/cryptography/sage/crypto/block_cipher/miniaes.html
- R. C.-W. Phan. Mini advanced encryption standard (mini-AES): a testbed for cryptanalysis students. Cryptologia, 26(4):283–306, 2002
- 编程实现AES算法（密钥长度128位）


感谢聆听!
THANK YOU FOR YOUR ATTENTION!