密 码 学

第四讲　高级数据加密标准(AES)

王后珍

武汉大学国家网络安全学院

空天信息安全与可信计算教育部重点实验室

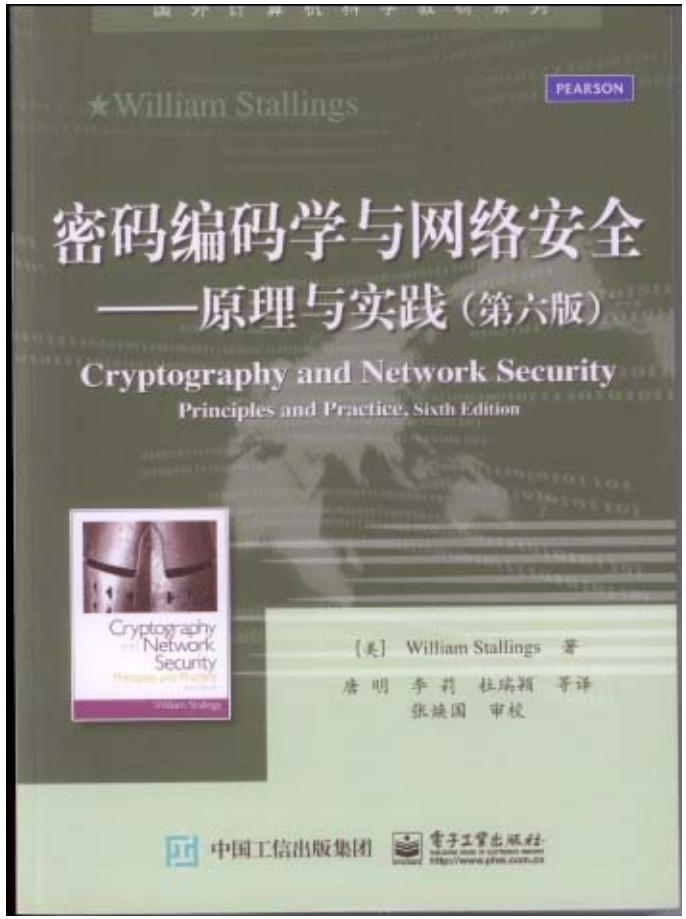
目 录

目 录

教材与主要参考书

教材

参考书



本讲内容

一、AES的概况
二、算法框图
三、数学基础
四、AES的基本变换
五、轮密钥的产生
六、 AES的加密算法
七、AES的基本逆变换
八、 AES的解密算法
九、AES的实现
十、AES的安全性

一、AES的概况

1、AES产生的背景

①1984年12月里根总统下令由国家安全局（NSA）研制新密码标准，以取代DES。

②1991年新密码开始试用并征求意见。

- 不公开算法，只提供芯片；
- 新密码设计成双刃剑。它是安全的，但通过法律允许可破译监听；
- 民众要求公开算法，并去掉法律监督。

③1994年颁布新密码标准（EES）。

④1995年5月贝尔实验室的博士生M.Blaze在PC 机上用45分钟攻击法律监督字段获得成功。

⑤1995年7月美国政府放弃用EES加密数据。

⑥1997年美国政府向社会公开征AES。

一、AES的概况

- 1997年4月15日，NIST发起征集高级加密标准（Advanced Encryption Standard）AES的活动，活动目的是确定一个非保密的、可以公开技术细节的、全球免费使用的分组密码算法，作为新的数据加密标准。
- 对AES的基本要求是：比三重DES快、至少与三重DES一样安全；无类别的；可公开的；无特权的；数据分组长度为128比特；密钥长度为128/192/256比特。

一、AES的概况

- 1998年6月NIST共收到21个提交的算法，在同年的8月首届AES会议上指定了15个候选算法。

- 1999年3月22日第二次AES会议上，将候选名单减少为5个，这5个算法是MARS，RC6，Rijndael，SERPENT，和Twofish。

一、AES的概况

- MARS（由IBM公司研究部门的一个庞大团队发布，对它的评价是算法复杂、速度快、安全性高）
- RC6（由RSA实验室发布，对它的评价是极简单、速度极快、安全性低）
- Rijndael（由Joan Daemen和 Vincent Rijmen 两位比利时密码专家发布，对它的评价是算法简洁、速度快、安全性好）
- Serpent（由Ross Anderson , Eli Biham 和 Lars Knudsen 发布，对它的评价是算法简洁、速度慢、安全性极高）
- Twofish（由Counterpane公司一个庞大的团队发布，对它的评价是算法复杂、速度极快、安全性高）

一、AES的概况

- 从全方位考虑，Rijndael汇聚了安全，性能，效率，易用和灵活等优点，使它成为AES最合适的选择
- 2000年10月NIST宣布Rijndael算法被选为高级加密标准
- 2001年11月发布为联邦信息处理标准(Federal Information Processing Standard，FIPS), 用于美国政府组织保护敏感信息的一种特殊的加密算法，即FIPS PUB 197标准

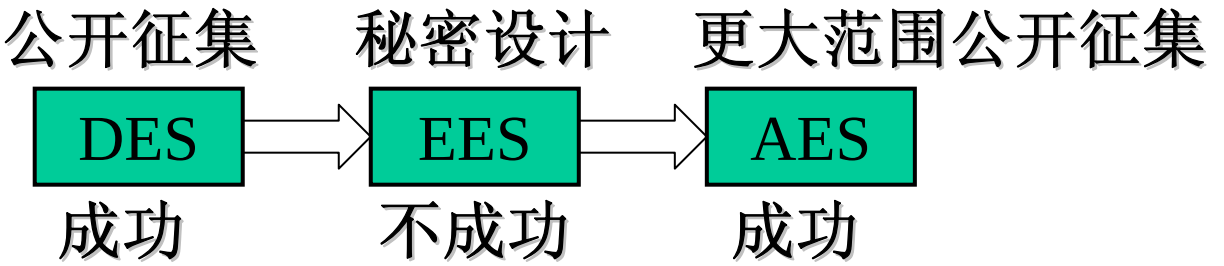
DES和AES的比较

	DES	AES
日期	1976年	1999年
分组大小	64b	128b、192b、256b
密钥长度	56b(有效长度)	128b、192b、256b(可能更长)
加密原语	代替、置换	代替、移位、位混合
算法	公开	公开
设计基本原理	未公开	公开
选择过程	保密	保密，但接受公开评论
来源	IBM, 由NSA加强	比利时密码学家

一、AES的概况

3、历史经验

- 从DES到AES，反映了美国商用密码政策的变化



- 这说明
  - 商用密码应当坚持公开设计原则，公布算法的政策，这是商用密码的客观规律。
  - 群众的力量是伟大的。

一、AES的概况

4、AES的设计要求

①安全性：可以抵抗目前所有已知的攻击；

②实用性：适应各种应用环境，加解密速度快；

③扩展性：分组长度和密钥长度可扩展，可以适应社会对保密性不断提高的需求。

一、AES的概况

5、整体特点

①分组密码

- 明文和密文长度128位，密钥长度可变（128/192/256等，现在选用 128 位）。

②面向二进制的密码算法

- 能够加解密任何形式的计算机数据。

③不是对合运算

- 加解密使用不同的算法。

④综合运用多种密码技术

- 置换、代替、代数

⑤整体结构

- SP结构，基本轮函数迭代，迭代轮数可变（≥10）

一、AES的概况

6、应用

①许多国际组织采用为标准

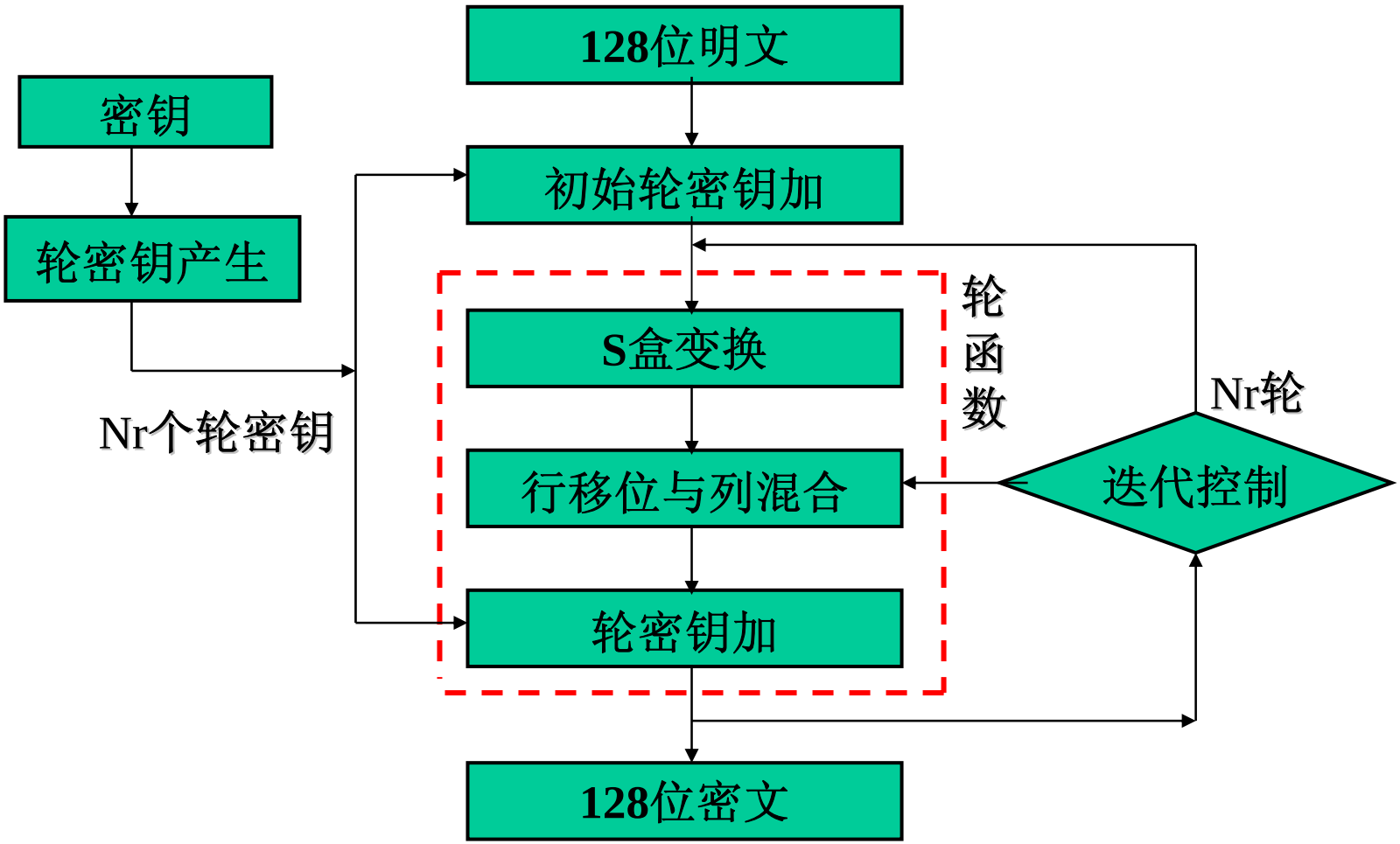
②产品形式：各种软件和硬件形式

③应用范围逐渐扩大

7、结论

- 十多年来的实际应用证明AES是安全的。
- AES还要继续接受更严峻的考验。
- 我们相信：经过全世界广泛分析的AES是不负众望的。

二、算法框图


128位明文
密钥
初始轮密钥加
轮密钥产生
轮函数
S盒变换
Nr轮
Nr个轮密钥
行移位与列混合
迭代控制
轮密钥加
128位密文

武漢大学

三、数学基础

1、AES的基础域是有限域 $GF(2^8)$

- 一个字节的全体256种取值构成一个$GF(2^8)$
- 一个$GF(2)$上的8次既约多项式可生成一个 $GF(2^8)$
- $GF(2^8)$的全体元素构成加法交换群、线性空间。
- $GF(2^8)$的非零元素构成乘法循环群。
- $GF(2^8)$中的元素有多种表示：

字节： $GF(2^8) = \{(a_7, a_6, \ldots a_1, a_0) \mid a_i \in GF(2)\}$

多项式形式： $GF(2^8) = \{a_7x^7 + \ldots + a_1x + a_0 \mid a_i \in GF(2)\}$

指数形式：$GF(2^8)^* = \{\alpha^0, \alpha^1 \ldots \alpha^{254}\}$， α是一个本原元

对数形式：$GF(2^8)^* = \{0, 1 \ldots 254\}$

- $GF(2^8)$的特征为 2 。

三、数学基础

2、 AES的$GF(2^8)$表示

- **AES采用$GF(2)$上既约多项式$m(x)$生成$GF(2^8)$：**

  $$m(x) = x^8+x^4+x^3+x+1$$

- **$GF(2^8)$的元素采用$GF(2)$上的多项式表示。**

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：字节57＝01010111的多项式表示：

$$01010111 \Longleftrightarrow x^6+x^4+x^2+x+1$$

三、数学基础

2、 AES的$GF(2^8)$表示

●加法：两个元素多项式的系数按位模 2加

例2：57＋83＝D4

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

●乘法：两个元素多项式相乘，模 $m(x)$

例3：57×83＝C1

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

■乘法单位元：字节01　多项式 1

■乘法逆元：

■ 设$a(x)$的逆元为$b(x)$，则 $a(x)b(x)=1 \bmod m(x)$ 。

■ 可根据Euclid算法可求出$b(x)$ 。

$$\begin{array}{l} x^3 + x^2 \quad\quad + 2 \\ +\ (x^2 - x + 1) \\ \hline x^3 + 2x^2 - x + 3 \end{array}$$

(a) 加法

$$\begin{array}{l} x^3 + x^2 \quad\quad + 2 \\ -\ (x^2 - x + 1) \\ \hline x^3 \quad\quad + x + 1 \end{array}$$

(b) 减法

$$\begin{array}{l} \quad\quad x^3 + x^2 \quad\quad + 2 \\ \quad\quad \times\ (x^2 - x + 1) \\ \hline \quad\quad x^3 + x^2 \quad\quad + 2 \\ \quad - x^4 - x^3 \quad\quad - 2x \\ x^5 + x^4 \quad\quad + 2x^2 \\ \hline x^5 \quad\quad\quad + 3x^2 - 2x + 2 \end{array}$$

(c) 乘法

$$\begin{array}{r} \quad\quad\quad\quad x + 2 \\ x^2 - x + 1 \,\big/\, x^3 + x^2 \quad\quad + 2 \\ \quad\quad x^3 - x^2 + x \\ \hline \quad\quad\quad\quad 2x^2 - x + 2 \\ \quad\quad\quad\quad 2x^2 - 2x + 2 \\ \hline \quad\quad\quad\quad\quad x \end{array}$$

(d) 除法

武漢大学

系数在二元域中的多项式运算

$$
\begin{array}{rrrrrrrr}
x^7 & +x^5 & +x^4 & +x^3 & & +x & +1 \\
 & & & +(x^3 & & +x & +1) \\
\hline
x^7 & +x^5 & +x^4 & & & &
\end{array}
$$

(a) 加法

$$
\begin{array}{rrrrrrrr}
x^7 & +x^5 & +x^4 & +x^3 & & +x & +1 \\
 & & & -(x^3 & & +x & +1) \\
\hline
x^7 & +x^5 & +x^4 & & & &
\end{array}
$$

(b) 减法

$$
\begin{array}{rrrrrrrrrr}
 & & x^7 & +x^5 & +x^4 & +x^3 & & +x & +1 \\
 & & & & & \times(x^3 & & +x & +1) \\
\hline
 & & x^7 & +x^5 & +x^4 & +x^3 & & +x & +1 \\
 & x^8 & & +x^6 & +x^5 & +x^4 & +x^2 & +x & \\
x^{10} & +x^8 & +x^7 & +x^6 & & +x^4 & +x^3 & & \\
\hline
x^{10} & & & & & +x^4 & +x^2 & & +1
\end{array}
$$

(c) 乘法

$$
\begin{array}{r|l}
 & x^4+1 \\
\cline{2-2}
x^3+x+1 & x^7+x^5+x^4+x^3+x+1 \\
 & x^7+x^5+x^4 \\
\cline{2-2}
 & x^3+x+1 \\
 & x^3+x+1 \\
\cline{2-2}
\end{array}
$$

(d) 除法

三、数学基础

2、 AES的$GF(2^8)$表示

● ***x*乘法 *x*time：用 *x*乘 $GF(2^8)$的元素**

例：

xtime(57) $= x(x^6+x^4+x^2+x+1) = x^7+x^5+x^3+x^2+x$

xtime(83) $= x(x^7+x+1) = x^8+x^7+x \bmod m(x)$

$= x^7+x^4+x^3+1 \bmod m(x)$

■若x^7的系数＝**0**，则为简单相乘：系数左移。

■若x^7的系数＝**1**，则乘后取模$m(x)$，即乘后减去 $x^8+x^4+x^3+x+1$ 。

三、数学基础

3、 AES的字表示与运算

●AES数据处理的单位是字节和字

■一个字＝4个字节=32比特

■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

如： $(57x^3+83x^2+4Ax+D1)+(Ax^3+B3x^2+EF)$

$= 5Dx^3+30x^2+4Ax+3F$

■字乘法：设a和c是两个字，$a(x)$和$c(x)$是其字多项式，AES定义a和c的乘积b为

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

三、数学基础

3、 AES的字表示与运算

●字乘法：

设 $a(x)=a_3x^3 + a_2x^2 + a_1x + a_0$

$c(x)=c_3x^3 + c_2x^2 + c_1x + c_0$

$b(x)=b_3x^3 + b_2x^2 + b_1x + b_0$

则，$b(x)= a(x)c(x) \bmod x^4+1$为：

$$b_0 = a_0 c_0 + a_3 c_1 + a_2 c_2 + a_1 c_3$$
$$b_1 = a_1 c_0 + a_0 c_1 + a_3 c_2 + a_2 c_3$$
$$b_2 = a_2 c_0 + a_1 c_1 + a_0 c_2 + a_3 c_3$$
$$b_3 = a_3 c_0 + a_2 c_1 + a_1 c_2 + a_0 c_3$$

三、数学基础

3、 AES的字表示与运算

●字乘法：写成矩阵形式

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

注意：

- x^4+1是可约多项式，字$c(x)$不一定有逆；
- 但AES选择的$c(x)$有逆， $c(x) = 03x^3+01x^2+01x+02$；
- $c(x)$有逆的条件是 $(x^4+1, c(x)) = 1$ 。

三、数学基础

3、 AES的字表示与运算

●字的x乘法：设$b(x)$是一个字，

$$p(x)=xb(x) \bmod x^4+1$$

●写成矩阵形式：

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

注意：

■因为模x^4+1，字的x乘法相当于按字节循环移位。

四、AES的基本变换

1、AES的数据处理方式

- 按字节处理
- 按字处理
- 按状态处理

2、状态

- 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。

四、AES的基本变换

2、状态

■符号： Nb—明密文所含的字数。

Nk—密钥所含的字数。

Nr—迭代轮数。

■ 例：Nb＝4时的状态

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

Nk＝4时的密钥数组

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

四、AES的基本变换

2、状态

■Nb、Nk、Nr之间的关系：

Nr	Nb=4	Nb=6	Nb=8
Nk=4	10	12	14
Nk=6	12	12	14
Nk=8	14	14	14

四、AES的基本变换

3、轮变换：加密轮函数

①标准轮变换：

```
Round(State,RoundKey)
{
ByteSub(State);          S盒变换
ShiftRow(State);         行移位变换
MixColumn(State);        列混合变换
AddRoundKey(State,RoundKey)     轮密钥加变换
}
```

四、AES的基本变换

3、轮变换—加密轮函数

②最后一轮的轮变换：非标准轮变换

```
Round(State,RounKey)
{
ByteSub(State);   S盒变换
ShiftRow(State); 行移位变换
AddRoundKey(State,RoundKey)  轮密钥加变换
}
```

注：最后一轮的轮变换中没有列混合变换。

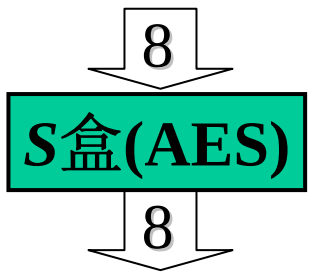
四、AES的基本变换

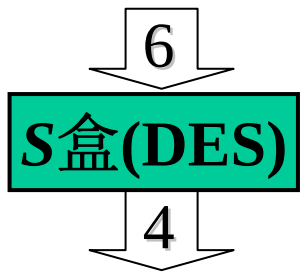
4、*S*盒变换 ByteSub(State)

①*S*盒变换是AES的唯一的非线性变换，是AES安全的关键，起密码学的混淆作用，

②AES使用16个相同的*S*盒，DES使用8个不相同的*S*盒。

③AES的*S*盒有8位输入8位输入，是一种非线性置换。DES的*S*盒有6位输入4位输出，是一种非线性压缩。





四、AES的基本变换

4、*S*盒变换 ByteSub(State)

④*S*盒变换：

■第一步：将输入字节用其$GF(2^8)$上的逆来代替；

- 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
- 用该逆元素代替原输入字节

■第二步：对上面的结果作如下的仿射变换：

（以$x_0 - x_7$作输入，以$y_0 - y_7$作输出）

四、AES的基本变换

4、*S*盒变换 ByteSub(State)

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

四、AES的基本变换

4、S盒变换 ByteSub(State)

- 注意：
  - S盒变换的第一步是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。
  - 系数矩阵中每列都含有 5个1，说明改变输入中的任意一位，将影响输出中的 5位发生变化。
  - 系数矩阵中每行都含有 5个1，说明输出中的每一位，都与输入中的 5位相关。

四、AES的基本变换

5、行移位变换 ShiftRow(State)

①行移位变换对状态的行进行循环移位。

②第 0行不移位，第1行移 C1字节，第2行移 C2字节，第3行移 C3字节。

③ C1，C2，C3按表取值

Nb	C1	C2	C3
4	1	2	3
6	1	2	3

④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。

四、AES的基本变换

6、列混合变换 MixColumn(State)

①列混合变换把状态的列视为$GF(2^8)$上的多项式$a(x)$，乘以一个固定的多项式$c(x)$，并模 x^4+1:

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

其中，$c(x)=03x^3+01x^2+01x+02$

②列混合变换属于线性变换，起扩散作用。

③$c(x)$与x^4+1互素，从而保证$c(x)$存在逆多项式$d(x)$，满足$c(x)d(x)=1 \mod x^4+1$ 。只有逆多项式$d(x)$存在，才能正确进行解密。

四、AES的基本变换

6、列混合变换 MixColumn(State)

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

其中，$b(x)=b_3x^3+b_2x^2+b_1x+b_0$

$a(x)=a_3x^3+a_2x^2+a_1x+a_0$

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

●写成矩阵形式

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

四、AES的基本变换

7、轮密钥加变换 AddRoundKey()

① 把轮密钥与状态进行模2相加。

② 轮密钥根据密钥产生算法产生。

③ 轮密钥长度等于数据块长度。

五、轮密钥的生成

①加密迭代中每一轮需要一个轮密钥参与加密。

②轮密钥根据密钥产生算法通过用户密钥得到。

③密钥产生分两步进行：

- 密钥扩展
- 轮密钥选择

④密钥扩展将用户密钥扩展为一个扩展密钥。

⑤密钥选择从扩展密钥中选出轮密钥。

五、轮密钥的生成

1、密钥扩展

①密钥扩展产生扩展密钥。

②用一个字元素的一维数组W[Nb*(Nr+1)]表示扩展密钥。

③用户密钥放在该数组最开始的Nk个字中。

④其它的字由它前面的字经过处理后得到。

⑤分 Nk≤6 和 Nk>6 两种密钥扩展算法。

五、轮密钥的生成

1、密钥扩展

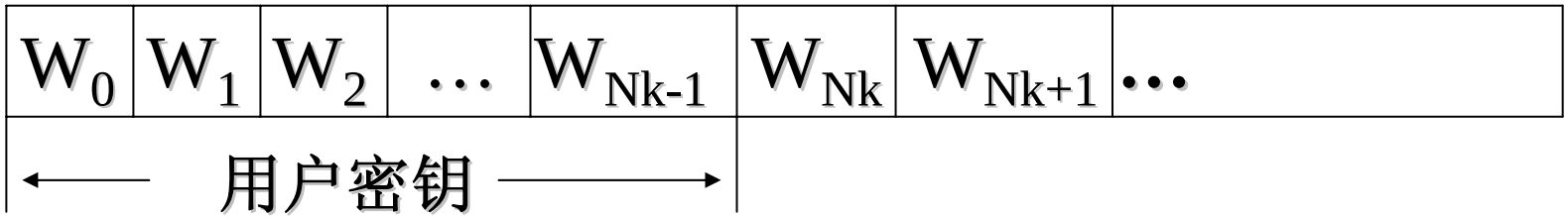
(1) Nk≤6 的密钥扩展

①最前面的Nk个字是由用户密钥填充的。

②之后的每一个字W[*j*]等于前面的字W[*j*-1]与Nk 个位置之前的字W[*j*-Nk]的异或。

③而且对于Nk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 。

五、轮密钥的生成



- 当 *j* 不是Nk的整数倍时:

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

- 当 *j* 是Nk的整数倍时:

$$W_j = W_{j-Nk} \oplus \text{ByteSub}(\text{Rotl}(W_{j-1})) \oplus \text{Rcon}[j/Nk];$$

五、轮密钥的生成

- 举例：Nk=4

W_0	W_1	W_2	W_3	W_4	W_5	W_6	…
←— 用户密钥 —→							

- 当 j =5时，j不是Nk=4的整数倍:

$$W_5 = W_1 \oplus W_4$$

- 当 j=4时，j 是Nk=4的整数倍:

$$W_4 = W_0 \oplus \text{ByteSub}(\text{Rotl}(W_3)) \oplus \text{Rcon}[1];$$

五、轮密钥的生成

- 说明：
  - Rotl是一个字里的字节循环左移函数，

设 字W ＝（A，B，C，D），

则， Rotl（W）＝（B，C，D，A）。

  - 轮常数Rcon与Nk无关，且定义为：

Rcon[*i*] = (RC [*i*],‘00’，‘00’，‘00’)

RC[0] = ‘01’

RC[*i*] = xtime(RC[*i*-1])

五、轮密钥的生成

1、密钥扩展

(2) Nk>6 的密钥扩展

●说明：

■ Nk>6 的密钥扩展与Nk≤6 的密钥扩展基本相同，不同之处在于：如果*j*被Nk除的余数＝4，则在异或之前，对W[j-1]进行ByteSub变换。

■增加ByteSub变换，是因为当Nk>6时密钥很长，仅仅对Nk的整数倍的位置处的字进行ByteSub变换，就显得ByteSub变换的密度较稀，安全程度不够强。

五、轮密钥的生成

2、轮密钥选择

- 根据分组的大小，依次从扩展密钥中取出轮密钥。
- 前面的Nb个字作为轮密钥0，接下来的Nb个字作为轮密钥1。…

W_0	W_1	…	W_{Nb-1}	W_{Nb}	…	W_{2Nb-1}	…	
轮密钥0				轮密钥1				

六、AES的加密算法

- AES的加密算法由以下部分组成：

①一个初始轮密钥加变换。

② Nr-1轮的标准轮变换。

③最后一轮的非标准轮变换。

加密算法：

```
Encryption（State,CipherKey）
 { KeyExpansion(CipherKey, RoundKey)
     AddRoundKey(State, RoundKey)
     For(I=1;I<Nr;I++)
     Round(State, RoundKey)
        {ByteSub(State);
         ShiftRow(State);
         MixColumn(State);
         AddRoundKey(State, RoundKey)}
      FinalRound(State, RoundKey)
        { ByteSub(State);
          ShiftRow(State);
          AddRoundKey(State, RoundKey)； }
}
```

注意：

●第一步和最后一步都用了轮密钥加，因为任何没有密钥参与的变换都是容易被攻破的。

●这一点比DES好，DES的IP和IP^{-1}都没有密钥参与。

七、AES的基本逆变换

- AES的加密算法不是对合运算，解密算法与加密算法不同。
- 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。

七、AES的基本逆变换

- **AES的各个基本变换都是可逆的。**

1、轮密钥加变换的逆就是其本身

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

2、行移位变换的逆是状态的后三行分别移位Nb-C1，Nb-C2，Nb-C3个字节。

七、AES的基本逆变换

3、列混合变换的逆

●因为列混合变换是把状态的每一列都乘以一个固定的多项式**c(x)**：

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

●所以列混合变换的逆就是状态的每列都乘以$c(x)$的逆多项式$d(x)$：

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

● $c(x)=03x^3+01x^2+01x+02$

● $d(x)=0Bx^3+0Dx^2+09x+0E$

七、AES的基本逆变换

4、S盒变换的逆

●第一步：首先进行逆仿射变换；

●第二步：对于第一步的结果字节，用其在$GF(2^8)$中的逆元素来代替。

- ■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
- ■用该逆元素代替原输入字节

七、AES的基本逆变换

- S盒的逆仿射变换：

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

七、AES的基本逆变换

5、解密的密钥扩展

- 解密的密钥扩展与加密的密钥扩展不同；
- 解密的密钥扩展定义如下：

①加密算法的密钥扩展。

②把InvMixColumn应用到除第一和最后一轮外的所有轮密钥上。

七、AES的基本逆变换

6、逆轮变换

●标准逆轮变换

```
Inv_Round(State,Inv_RoundKey)
{  Inv_ByteSub(State);
   Inv_ShiftRow(State);
   Inv_MixColunm(State);
   AddRoundKey(State,Inv_RoundKey);
}
```

七、AES的基本逆变换

6、逆轮变换

●最后一轮的逆变换：非标准逆轮变换

```
Inv_FinalRound(State,Inv_RoundKey)
{  Inv_ByteSub(State);
   Inv_ShiftRow(State);
   AddRoundKey(State,Inv_RoundKey);
}
```

八、AES的解密算法

- 加密算法不是对合运算：

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

- 解密算法的结构与加密算法的结构相同
- 解密中的变换为加密算法变换的逆变换，且密钥扩展策略稍有不同。

解密算法：

```
Decryption(State,CipherKey)
{  Inv_KeyExpansion(CipherKey,Inv_ RoundKey);
AddRoundKey(State,Inv_ RoundKey);
For(I=1;I<Nr;I++)
   Inv_Round(State, Inv_ RoundKey);
        {Inv_ByteSub(State);
        Inv_ShiftRow(State);
        Inv_MixColumn(State);
        AddRoundKey(State, Inv_ RoundKey ；  }
Inv_FinalRound(State,Inv_RoundKey)
        { InvByteSub(State);
        InvShiftRow(State);
        AddRoundKey(State, Inv_ RoundKey )；  }
}
```

九、AES的实现

- 适应多种环境，高效，方便是**AES**的突出优点。
- 由于**AES**的基本运算由ByteSub、MixColumn、ShiftRow和AddRoundKey变换构成，因此**AES**的实现主要是这些变换的实现。
- 其中ShiftRow和AddRoundKey的实现比较容易，因此主要是ByteSub和MixColumn变换的实现问题。
- 有了这些基本运算的实现，便可以有效地实现整个**AES**。

九、AES的实现

- 实现方法：
  - 软件
  - 硬件
- 软件方法：
  - 基于算法描述
  - 基于查表

九、AES的实现

1、基于算法描述的软件实现

- AES的算法描述是一种程序化的描述，便于实现。
- AES的四种基本变换都比较简单，便于实现。
- 用 C语言仿照算法描述，可方便地实现。但这种实现的速度不是最快的！

九、AES的实现

2、基于查表的软件实现

- 用查表实现算法是一种高效的软件实现方法。
- 时空折换是信息科学的基本策略。
- 用查表实现算法，就是用空间换取时间。
- 目前计算机系统的存储空间大、而且便宜，为查表实现算法提供了物资基础。

九、AES的实现

2、基于查表的软件实现

①S盒的查表实现

- 实现S盒变换的最快方法是，直接计算出S盒的变换的结果，并造表存储，使用时直接查表。因为ByteSub变换是字节函数，所以表的规模不大，只含256个字节元素。
- 注意：解密时用的是逆S盒，因此共需要造两个S盒表。

九、AES的实现

②列混合变换 MixColumn的查表实现

- $b(x)= a(x)c(x) \bmod x^4+1$
- 写成矩阵形式

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

- 主要运算是$GF(2^8)$上的乘法
- 对于输入(a_0,a_1,a_2,a_3)造表，表太大，很困难。
- 分而治之：因为系数是固定的，对一个字节a_i造表，查4次表，完成列混合变换。

九、AES的实现

- 逆列混合变换 Inv_MixColumn的查表实现
- 因为解密时需要逆列混合变换
- $a(x)= b(x)d(x) \bmod x^4+1$
- $d(x)= 0Bx^3+0Dx^2+09x+0E$
- 写成矩阵形式

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

- 同样可造表查表实现逆列混合变换。

九、AES的实现

③轮函数的查表实现

●整个轮函数都可以通过查表和一些简单的运算来实现。

●教科书上给出了利用查表实现轮函数的完整方法，请认真阅读教科书

十、AES的安全性

- 数学分析：有文献利用不同方法给出了8轮AES-128、8轮AES-192、9轮AES-256、14轮AES-256的密码分析。
- 侧信道分析：有文献给出了分别使用10和15个样本恢复出AES-192的密钥和AES-256的密钥。
- AES的密码算法已经设计得相当好，但AES密码算法也有自己弱点。研究表明，列混合的扩散度不够，密钥扩展的非线性不够，而且缺少抵抗侧信道分析设计。
- 上述攻击方法都还不能对AES构成本质的威胁。美国NIST继续支持AES。

谢　谢！