字节代换层(S-盒)　状态中的每个元素都使用具有特殊数学属性的查找表进行非线性变换。这种方法将混淆引入数据中，即它可以保证对单个状态位的修改可以迅速传播到整个数据路径中。

扩散层　为所有状态位提供扩散。它由两个子层组成，每个子层都执行线性操作：

- ShiftRows(行移位变化)层在位级别进行数据置换。
- MixColumn(列混淆变换)层是一个混淆操作，它合并(混合)了长度为四个字节的分组。

与 DES 类似，AES 密钥编排也从原始 AES 密钥中计算出轮密钥或子密钥($k_0, k_1, \ldots k_{n_r}$)。

在进一步描述各层的内部功能(第 4.4 节)前，我们首先要介绍一个新的数学概念，即伽罗瓦域(Galois field)。AES 层内的所有操作都需要伽罗瓦域的计算。

4.3　一些数学知识：伽罗瓦域简介

AES 的绝大多数层内都会用到伽罗瓦域运算，尤其是在 S-盒层和 MixColumn 层。因此，为了更深入地理解 AES 的内部结构，在继续学习第 4.4 节的算法前我们首先需要简单地介绍一下伽罗瓦域。初步理解 AES 并不需要有伽罗瓦域的背景知识，所以读者可以跳过此节的内容。

4.3.1　有限域的存在性

有限域有时也称为伽罗瓦域，它指的是拥有有限个元素的集合。大致来讲，伽罗瓦域是一个由有限个元素组成的集合，在这个集合内可以执行加、减、乘和逆操作。在介绍域的定义前，我们首先需要理解一个更简单的代数结构概念，即群。图 4-2 显示了 AES 的加密框图。

定义 4.3.1　群(Group)

群指的是元素集合 G 及 G 内任意两个元素的联合操作$\circ$的集合。群具有以下特性：

1. 群操作$\circ$是封闭的，即对所有的 $a,b, \in G$，$a \circ b = c \in G$ 始终成立。
2. 群操作是可结合的，即对所有的 $a,b,c \in G$ 都有 $a \circ (b \circ c) = (a \circ b) \circ c$。
3. 存在一个元素 $1 \in G$，对所有 $a \in G$ 都满足 $a \circ 1 = 1 \circ a$。此元素 1 称为中性元(或单位元)。
4. 对每个 $a \in G$，都存在一个元素 $a^{-1} \in G$，使得 $a \circ a^{-1} = a^{-1} \circ a = 1$，而 a^{-1} 就称为 a 的逆元。
5. 在上面的特性的基础上，如果对所有的 a，$b \in G$ 都有 $a \circ b = b \circ a$，则称此群为阿贝尔群(或交换群)。

概括地讲，群就是一个操作及对应逆操作的集合。如果该操作为加法，其逆操作就是减法；如果该操作是乘法，其逆操作则为除法(或与其逆元的乘法)。

示例 4.1 整数集合 $\mathbb{Z}_m = \{0,1,...,m-1\}$ 与操作加法模 m 组成了一个中性元为 0 的群。每个元素 a 都存在一个逆元-a，使得 $a + (-a) = 0 \bmod m$。请注意，这个整数集合与乘法操作并不能构成群，因此绝大多数的元素 a 都不存在满足 $aa^{-1} = 1 \bmod m$ 的逆元。

◇

为使一个结构同时支持四种基本算术运算(即加、减、乘、除)，我们需要一个同时包含加法与乘法群的集合，这也是我们常说的域(field)。

定义 4.3.2 域(field)

域 F 是拥有以下特性的元素的集合。

- F 中的所有元素形成一个加法群，对应的群操作为“+”，中性元为 0。
- F 中除 0 外的所有元素构成了一个乘法群，对应的群操作为“×”，中性元为 1。
- 当混合使用这两种群操作时，分配定理始终成立，即对所有的 $a, b, c \in F$，都有 $a(b+c) = (ab) + (ac)$。

示例 4.2 实数集合 $\mathbb{R}$ 是一个域，其加法群中的中性元为 0，乘法群中的中性元为 1。每个实数 a 都有一个加法逆元，称为-a；并且每个非零元素 a 都有一个乘法逆元 1/a。

◇

在密码编码学中，我们基本上只对拥有有限个元素的域感兴趣，而这样的域也称为有限域或伽罗瓦域。域所包含元素的个数称为域的阶或基。下面这个定理非常重要：

定理 4.3.1 只有当 m 是一个素数幂时，即 $m = p^n$(其中 n 为正整数，p 为素数)，阶为 m 的域才存在。P 称为这个有限域的特征。

此定理意味着有限域的元素个数可以为 11 或 81(因为 $81 = 3^4$)或 256(因为 $256=2^8$，并且 2 是素数)等等。但是，拥有 12 个元素的有限域是不存在的，因为 $12 = 2^2 \cdot 3$，因此 12 也不是一个素数幂。本章剩余部分将介绍有限域的构建方式；更重要的是介绍如何在有限域内进行算术运算，这才是我们的最终目的。

4.3.2　素域

有限域最直观的例子就是阶为素数的域，即 $n=1$ 的域。域 $GF(p)$ 的元素可以用整数 0，1，…，p-1 来表示。域的两种操作就是模整数加法和整数乘法模 p。

> **定理 4.3.2**　假设 p 是一个素数，整数环 $\mathbb{Z}_p$ 表示为 $GF(p)$，也称为是拥有素数个元素的素数域或伽罗瓦域。$GF(p)$ 中所有的非零元素都存在逆元，$GF(p)$ 内的算术运算都是模 p 实现的。

这意味着第 1.4.2 节介绍的整数环 $\mathbb{Z}_m$，即运算为整数模加法和模乘法的整数且 m 也正好是素数，$\mathbb{Z}_m$ 不仅是一个环，而且也是一个有限域。

为了在素域中进行算术运算，我们必须遵循整数环的以下规则：加法和乘法都是通过模 p 实现的；任何一个元素 a 的加法逆元由 $a + (-a) = 0 \bmod p$ 给出；任何一个非零元素 a 的乘法逆元定义为 $a \cdot a^{-1} = 1$。下面列举一个素域的示例。

示例 4.3　对于有限域 $GF(5) = \{0, 1, 2, 3, 4\}$，下面的表格描述了如何计算两个元素之间的加法和乘法结果，以及求解域元素的加法逆元和乘法逆元的方法。利用这些表格，我们可以在不明确使用模约简的情况下完成该域内的所有计算。

加法

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

加法逆元

-0=0
-1=4
-2=3
-3=2
-4=1

乘法

×	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

乘法逆元

0^{-1} 不存在
$1^{-1}=1$
$2^{-1}=3$
$3^{-1}=2$
$4^{-1}=4$

◇

$GF(2)$是一个非常重要的素域，它也是存在的最小的有限域。下面来看该域对应的乘

法表和加法表。

示例 4.4 考虑一个小的有限域 $GF(2) = \{0,1\}$。算术运算是通过模 2 实现的，并得到以下算术表：

加法

+	0	1
0	0	1
1	1	0

乘法

×	0	1
0	0	0
1	0	1

从第 2 章的序列密码可知，$GF(2)$的加法，即模 2 加法，与异或(XOR)门等价。而从上面的示例可以看出 $GF(2)$乘法与逻辑与(AND)门等价。域 $GF(2)$对 AES 而言至关重要。

4.3.3 扩展域 GF(2^m)

在 AES 中包含 256 个元素的有限域可以表示为 $GF(2^8)$。选择这个有限域的原因在于，该域中的每个元素都可以用一个字节表示。在 S-盒和 MixColumn 变换中，AES 将内部数据路径的每个字节均表示为域 $GF(2^8)$中的一个元素，并利用此有限域中的算术运算操作数据。

然而，如果有限域的阶不是素数，2^8 很显然也不是一个素数，则此有限域内的加法和乘法操作就不能用整数加法模 2^8 和乘法模 2^8 表示。$m > 1$ 的域称为扩展域。为了处理扩展域，我们需要(1)使用不同的符号表示此域内的元素；(2)使用不同的规则执行此域内元素的算术运算。我们将在下面看到，扩展域的元素可以用多项式表示；并且扩展域内的计算也可以通过某种多项式运算得到。

在扩展域 $GF(2^m)$中，元素并不是用整数表示的，而是用系数为域 $GF(2)$中元素的多项式表示。这个多项式最大的度为 m-1，所以，每个元素共有 m 个系数。在 AES 使用的域 $GF(2^8)$中，每个元素 $A \in GF(2^8)$都可表示为：

$$A(x) = a_7x^7 + \cdots + a_1x + a_0, \quad a_i \in GF(2) = \{0,1\}。$$

请注意：这样的多项式共有 $256 = 2^8$ 个。这 256 个多项式的集合就是有限域 $GF(2^8)$。我们注意到，每个多项式都可以按一个 8 位向量的数字形式存储：

$$A = (a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0)。$$

尤其是诸如 x^7、x^6 等因子都无需存储，因为从位的位置就可以清楚地地判断出每个系数对应的幂 x^i。

4.3.4　GF(2^m)内的加法与减法

下面来看一下扩展域中的加法与减法。AES 的密钥加法层使用了加法；事实证明，扩展域中的加法与减法操作都十分简单：通过标准的多项式加法和减法即可得到，即仅需将 x 幂次相同的系数进行相加或相减即可。而且这些系数的加法或减法操作都是在底层域 $GF(2)$内完成的。

定义 4.3.3　扩展域中的加法与减法

假设 $A(x)$，$B(x) \in GF(2^m)$，计算两个元素之和的方法为：

$$C(x) = A(x) + B(x) = \sum_{i=0}^{m-1} c_i x^i \ , \quad c_i \equiv a_i + b_i \bmod 2$$

而两个元素之差的计算方式为：

$$C(x) = A(x) - B(x) = \sum_{i=0}^{m-1} c_i x^i \ , \quad c_i \equiv a_i - b_i \equiv a_i + b_i \bmod 2 \text{。}$$

注意：上面系数执行的是模 2 加法(或减法)。从第 2 章可知，模 2 加法与模 2 减法其实是相同的。此外，模 2 加法与按位 XOR 或等价。下面列举 AES 中使用的域 $GF(2^8)$的一个示例：

示例 4.5　下面是计算 $GF(2^8)$中两个元素之和 $C(x) = A(x) + B(x)$的方法：

$$\begin{array}{l} A(x)=x^7+x^6+x^4+\quad 1 \\ \underline{B(x)=\quad\quad x^4+x^2+1} \\ C(x)=x^7+x^6+\quad x^2 \end{array}$$

◇

注意：如果计算上例中两个多项式的差 $A(x)-B(x)$，将得到与其和相同的结果。

4.3.5　GF(2^m)内的乘法

GF(2^8)内的乘法是 AES MixColumn 变换的核心操作。首先使用标准多项式乘法准则将有限域 $GF(2^m)$的两个元素(使用多项式表示)相乘：

$$A(x) \cdot B(x) = (a_{m-1}x^{m-1} + \cdots + a_0) \cdot (b_{m-1}x^{m-1} + \cdots + b_0)$$

$$C'(x) = c'_{2m-2}\ x^{2m-2} + \cdots + c'_0 \text{，}$$

其中，

$$c'_0 = a_0 b_0 \bmod 2$$

$$c_1' = a_0 b_1 + a_1 b_0 \bmod 2$$

$$\vdots$$

$$c_{2m-2}' = a_{m-1} b_{m-1} \bmod 2 。$$

注意：所有系数 a_i、b_i 和 c_i 都是 $GF(2)$中的元素，并且系数的算术运算都是在 $GF(2)$内完成的。通常，乘积多项式 $C(x)$的度会大于 m-1，因此需要进行化简。而化简的基本思想与素域内的乘法情况相似：在 $GF(p)$中，将两个整数相乘得到的结果除以一个素数，并且只考虑最后的余数。而扩展域中进行的操作为：将两个多项式相乘的结果除以一个多项式，并且只考虑多项式除法得到的余数。模约简需要不可约多项式。回顾第 2.3.1 节可知，不可约多项式大致可以看作为素数，即它们仅有的因子就是 1 和多项式本身。

定义 4.3.4　扩展域乘法

假设 $A(x)$，$B(x) \in GF(2^m)$，且

$$P(x) \equiv \sum_{i=0}^{m} p_i x^i , \quad p_i \in GF(2)$$

是一个不可约多项式。两个元素 $A(x)$和 $B(x)$的乘法运算为：

$$C(x) \equiv A(x) \cdot B(x) \bmod P(x) 。$$

因此，每个域 $GF(2^m)$都需要一个度为 m、且系数来自 $GF(2)$的不可约多项式 $P(x)$。注意：不是所有的多项式都是不可约多项式。例如，多项式 $x^4 + x^3 + x + 1$ 是可约的，因为

$$x^4 + x^3 + x + 1 = (x^2 + x + 1)(x^2 + 1)$$

因此，它不能用来构建扩展域 $GF(2^4)$。由于本原多项式是一种特殊的不可约多项式，表 2-3 中所列出来的多项式都可以用来构建域 $GF(2^m)$。AES 使用的不可约多项式为

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

它是 AES 规范的一部分。

示例 4.6　我们想要将域 $GF(2^4)$中的两个多项式 $A(x) = x^3 + x^2 + 1$ 和 $B(x) = x^2 + x$ 相乘，此伽罗瓦域内的不可约多项式为：

$$P(x) = x^4 + x + 1。$$

普通多项式乘积的计算方式为：

$$C'(x) = A(x) \cdot B(x) = x^5 + x^3 + x^2 + x。$$

现在可以使用之前学习的多项式除法化简 $C'(x)$。然而，有时对主项 x^4 和 x^5 分别化简会更容易计算：

$$x^4 = 1 \cdot P(x) + (x+1)$$
$$x^4 \equiv x+1 \bmod P(x)$$
$$x^5 \equiv x^2 + x \bmod P(x)。$$

下面只需将 x^5 化简后的表达式插入到中间结果 $C'(x)$ 中：

$$C(x) \equiv x^5 + x^3 + x^2 + x \bmod P(x)$$
$$C(x) \equiv (x^2 + x) + (x^3 + x^2 + x) = x^3$$
$$A(x) \cdot B(x) \equiv x^3。$$

◇

注意，切勿将 $GF(2^m)$内的乘法与整数乘法混为一谈，在考虑伽罗瓦域软件实现时尤其如此。回顾前面的内容可知，多项式(即域元素)通常都是以位向量的方式存储于计算机中。而从前一个例子中的乘法可以看出，下面的非典型操作也是在位级别执行的：

$$\begin{array}{ccccc} A & \cdot & B & = & C \\ (x^3+x^2+1) & \cdot & (x^2+x) & = & x^3 \\ (1\ 1\ 0\ 1) & \cdot & (0\ 1\ 1\ 0) & = & (1\ 0\ 0\ 0)。 \end{array}$$

这种计算方式与整数算术运算完全不同。如果将多项式表示为整数，即$(1101)_2 = 13_{10}$，$(0110)_2 = 6_{10}$，则结果为$(1001110)_2 = 78_{10}$。显然，这个结果与伽罗瓦域中的乘积不同。因此，尽管我们可将域元素表示为整型数据类型，却不能使用整数算术运算来计算。

4.3.6　GF(2^m)内的逆操作

$GF(2^8)$中的逆操作是字节代换变换的核心操作，而字节代换变换包含了 AES 的 S-盒。给定一个有限域 $GF(2^m)$与其对应的不可约简化多项式 $P(x)$，任何一个非零元素 $A \in GF(2^m)$的逆元定义为：

$$A^{-1}(x) \cdot A(x) = 1 \bmod P(x)。$$

对小型域而言——实际上它通常指是元素个数不超过 2^{16} 的域——一般使用查找表就已足够，该查找表包含了使用预计算得到的域内所有元素的逆元。表 4-2 显示了 AES 的 S-盒中使用的所有值。此表包含了 $GF(2^8)$模 $P(x) = x^8 + x^4 + x^3 + x + 1$(十六进制表示)的所有逆元。其中一个特例就是域元素 0 的项，因为它的逆元并不存在。然而，AES 的 S-盒需要一个定义了每个可能输入值的代换表。因此，设计者将 S-盒定义为输入值 0 对应的输出值也为 0。

表 4-2 AES S-盒内使用的字节 xy 对应的 $GF(2^8)$中的乘法逆元表

	Y															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00	01	8D	F6	CB	52	7B	Dl	E8	4F	29	C0	B0	El	E5	C7
1	74	B4	AA	4B	99	2B	60	5F	58	3F	FD	CC	FF	40	EE	B2
2	3A	6E	5A	Fl	55	4D	A8	C9	Cl	0A	98	15	30	44	A2	C2
3	2C	45	92	6C	F3	39	66	42	F2	35	20	6F	77	BB	59	19
4	1D	FE	37	67	2D	31	F5	69	A7	64	AB	13	54	25	E9	09
5	ED	5C	05	CA	4C	24	87	BF	18	3E	22	F0	51	EC	61	17
6	16	5E	AF	D3	49	A6	36	43	F4	47	91	DF	33	93	21	3B
7	79	B7	97	85	10	B5	BA	3C	B6	70	D0	06	Al	FA	81	82
X 8	83	7E	7F	80	96	73	BE	56	98	9E	95	D9	F7	02	B9	A4
9	DE	6A	32	6D	D8	8A	84	72	2A	14	9F	88	F9	DC	89	9A
A	FB	7C	2E	C3	8F	B8	65	48	26	C8	12	4A	CE	E7	D2	62
B	0C	E0	1F	EF	11	75	78	71	A5	8E	76	3D	BD	BC	86	57
C	0B	28	2F	A3	DA	D4	E4	0F	A9	27	53	04	1B	FC	AC	E6
D	7A	07	AE	63	C5	DB	E2	EA	94	8B	C4	D5	9D	F8	90	6B
E	B1	0D	D6	EB	C6	0E	CF	AD	08	4E	D7	E3	5D	50	1E	B3
F	5B	23	38	34	68	46	03	8C	DD	9C	7D	A0	CD	lA	41	1C

示例 4.7 从表 4-2 可知，

$$x^7 + x^6 + x = (1100\ 0010)_2 = (C2)_{hex} = (xy)$$

的逆元可以从第 C 行，第 2 列的元素得到：

$$(2F)_{hex} = (0010\ 1111)_2 = x^5 + x^3 + x^2 + x + 1。$$

可使用以下乘法对上述结果进行验证：

$$(x^7 + x^6 + x) \cdot (x^5 + x^3 + x^2 + x + 1) \equiv 1 \bmod P(x)。$$

◇

注意：以上的表并不包含 S-盒本身，S-盒比较复杂，将在第 4.4.1 节中予以介绍。

除使用查找表外，另一种方法就是直接计算逆元。计算乘法逆元的主要算法就是扩展的欧几里得算法，这部分内容将在第 6.3.1 节进行介绍。