Advanced Encryption Standard (AES)

Yih-Kuen Tsay

Department of Information Management
National Taiwan University

The Origin of AES

- A symmetric cipher intended to replace DES and 3DES (DES is slow and 3DES is three times as slow. Both use a 64-bit block size; a larger block size would be more secure.)
- A call for proposals for a new Advanced Encryption Standard issued in 1997 by NIST
- Selected algorithm: Rijndael, designed by Joan Daemen and Vincent Rijmen from Belgium.
- Published as FIPS PUB 197 in November, 2001
- Block size: 128 bits
- Key lengths: 128, 192, and 256 bits

Arithmetic Operations in AES

- All operations are performed on 8-bit bytes as elements in $GF(2^8)$ with $m(x) = x^8 + x^4 + x^3 + x + 1$.
- The addition of two bytes is the bitwise XOR operation.

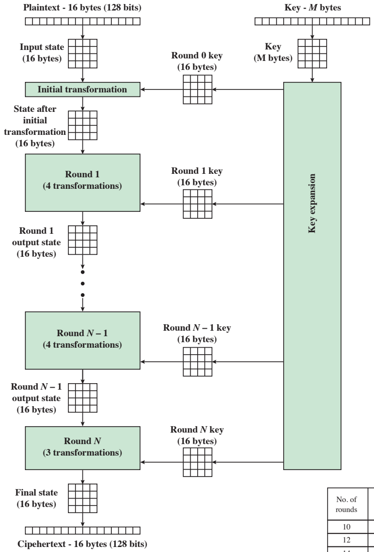
$$\begin{array}{rr} & 01010111 \\ \oplus & 10000011 \\ \hline & 11010100 \end{array}$$

- The multiplication of two bytes is the multiplication in $GF(2^8)$.

$$\begin{array}{rl} & (b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0) \times (00000010) \\ = & \begin{cases} (b_6 b_5 b_4 b_3 b_2 b_1 b_0 0) & \text{if } b_7 = 0 \\ (b_6 b_5 b_4 b_3 b_2 b_1 b_0 0) \oplus (00011011) & \text{if } b_7 = 1 \end{cases} \end{array}$$

- Multiplication by a nonzero byte b and then by its multiplicative inverse b^{-1} gives back the original byte.

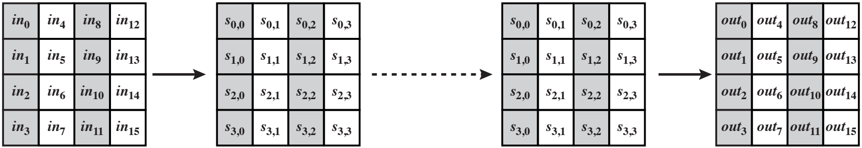
AES Encryption Process



No. of rounds	Key Length (bytes)
10	16
12	24
14	32

Source: Figure 5.1, Stallings 2010

AES Data Structures



(a) Input, state array, and output

(b) Key and expanded key

Source: Figure 5.2, Stallings 2010

AES Parameters

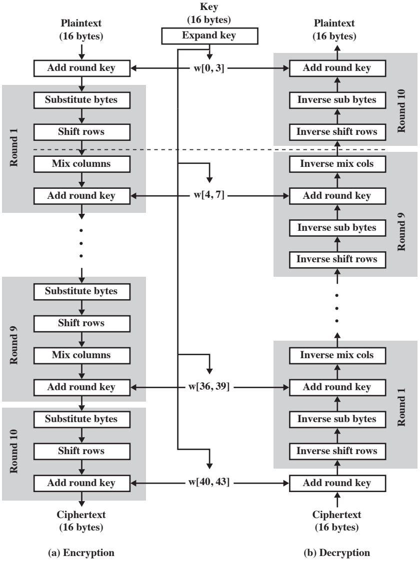
Key Size (words/bytes/bits)	4/16/128	6/24/192	8/32/256
Plaintext Block Size (words/bytes/bits)	4/16/128	4/16/128	4/16/128
Number of Rounds	10	12	14
Round Key Size (words/bytes/bits)	4/16/128	4/16/128	4/16/128
Expanded Key Size (words/bytes)	44/176	52/208	60/240

Source: Table 5.1, Stallings 2010

About the AES Cipher

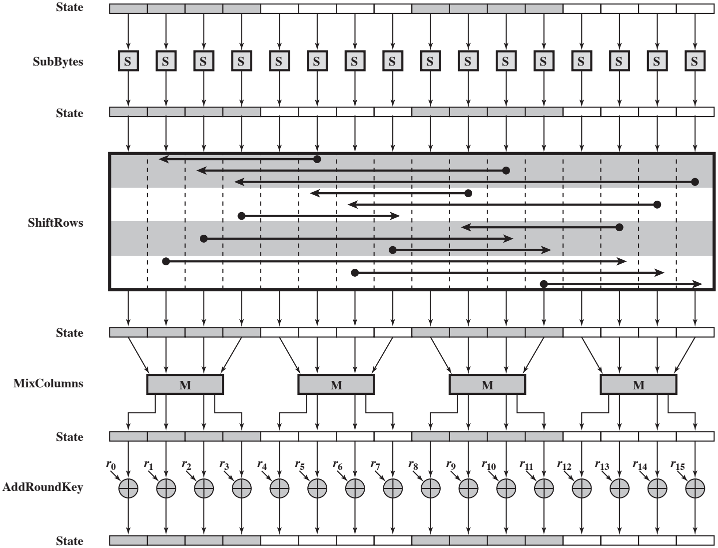
- Not a Feistel structure; entire data block processed in each round
- Input key expanded into 11 round keys of the same length
- Four stages used: **Substitute bytes**, **Shift rows** (the only permutation), **Mix columns**, **Add round key**
- Decryption algorithm different from encryption algorithm
- Correctness easy to verify.

AES Encryption and Decryption



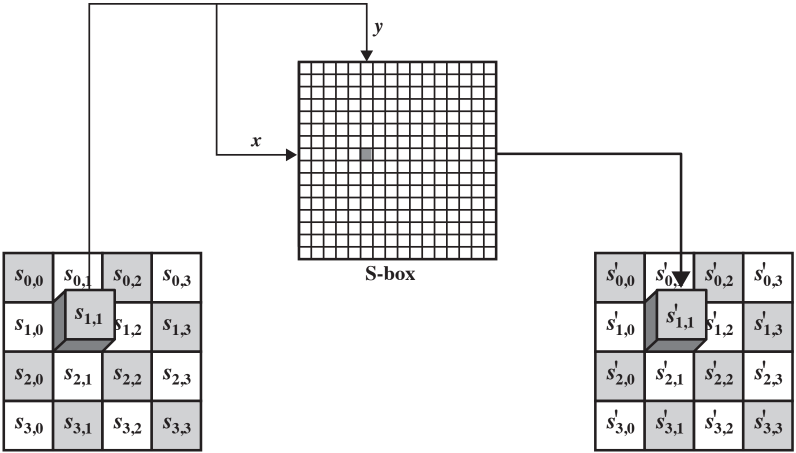
Source: Figure 5.3, Stallings 2010

AES Encryption Round



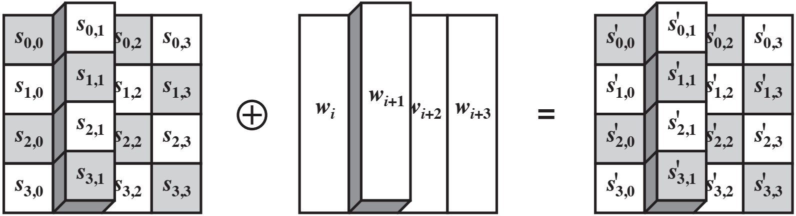
Source: Figure 5.4, Stallings 2010

AES Byte-Level Operations



Source: Figure 5.5(a), Stallings 2010

AES Byte-Level Operations (cont.)



Source: Figure 5.5(b), Stallings 2010

AES S-Boxes

		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

Source: Table 5.2, Stallings 2010

AES Inverse S-Boxes

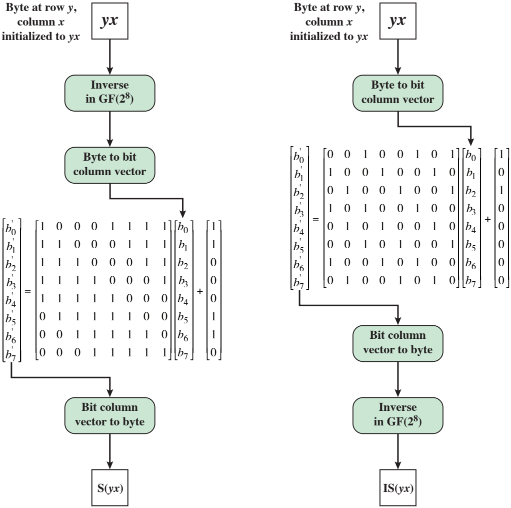
		y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
	1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
	2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
	3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
	4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	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

Source: Table 5.2, Stallings 2010

An Example of `SubBytes`

EA	04	65	85		87	F2	4D	97
83	45	5D	96	→	EC	6E	4C	90
5C	33	98	B0		4A	C3	46	E7
F0	2D	AD	C5		8C	D8	95	A6

Construction of S-Box and IS-Box



(a) Calculation of byte at row y, column x of S-box

(a) Calculation of byte at row y, column x of IS-box

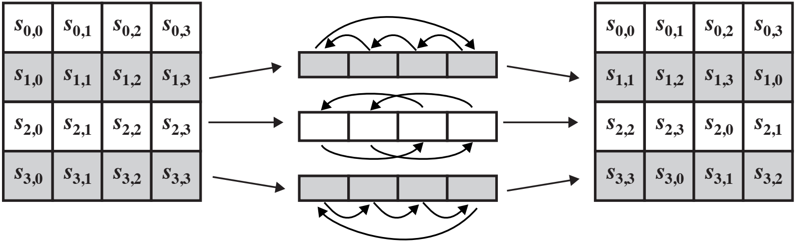
Source: Figure 5.6, Stallings 2010

Construction of the S-Box

- Initialization: 1st row: $\{00\}, \{01\}, \{02\}, \cdots, \{0F\}$; 2nd row: $\{10\}, \{11\}, \{12\}, \cdots, \{1F\}$; etc.
- Replace each byte with its multiplicative inverse; the value $\{00\}$ is mapped to itself.
- Apply the following (invertible) transformation:

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

Shift Rows



Source: Figure 5.7(a), Stallings 2010

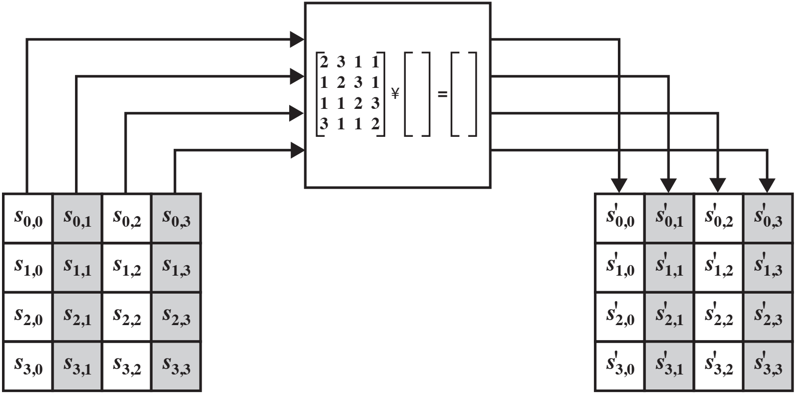
An Example of `ShiftRows`

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

$\longrightarrow$

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

Mix Columns



Source: Figure 5.7(b), Stallings 2010

An Example of `MixColumns`

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

$\longrightarrow$

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

$$
\begin{array}{lcl}
(\{02\} \bullet \{87\}) & = & 00010101 \\
(\{03\} \bullet \{6E\}) & = & 10110010 \\
\{46\} & = & 01000110 \\
\{A6\} & = & 10100110 \\
\hline
 & & 01000111 \ (= \{47\})
\end{array}
$$

InvMixColumns

IM NTU

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

An Example of AddRoundKey

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

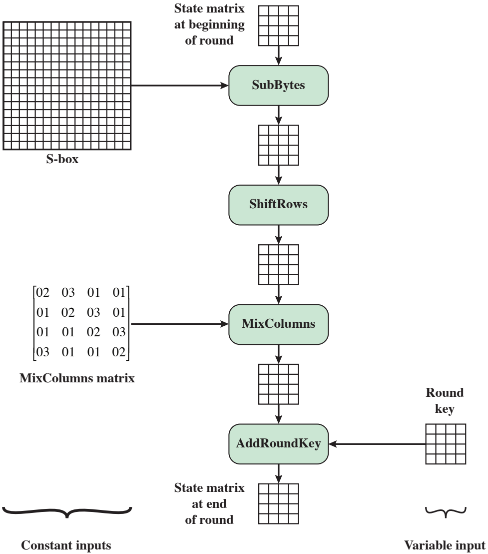
$\oplus$

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	D2

Inputs for Single AES Round



Source: Figure 5.8, Stallings 2010

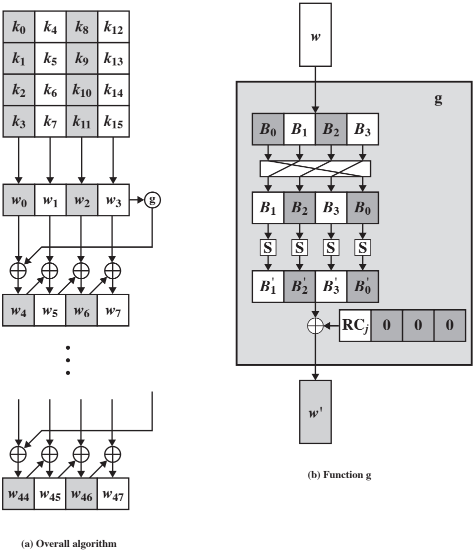
Key Expansion

```
KeyExpansion (byte key[16],word w[44])
{
   word temp
   for (i = 0; i < 4; i++)
      w[i] = (key[4*i],key[4*i+1],key[4*i+2],key[4*i+3]);
   for (i = 4; i < 44; i++)
   {  temp = w[i-1];
      if (i mod 4 = 0) temp = SubWord(RotWord(temp))⊕Rcon[i/4];
      w[i] = w[i-4]⊕temp
   }
}
```

Rcon[j] = (RC[j],0,0,0), with RC[1]=1, RC[j]=2•RC[$j-1$]

j	1	2	3	4	5	6	7	8	9	10
$RC[j]$	01	02	04	08	10	20	40	80	1B	36

AES Key Expansion



Source: Figure 5.9, Stallings 2010

An Example of Key Expansion

Suppose the round key (Words 32, 33, 34, and 35) for Round 8 is

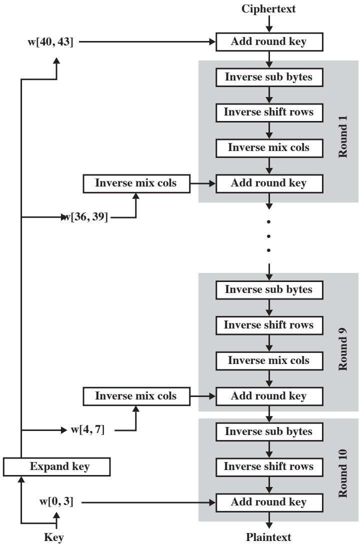
EA D2 73 21 B5 8D BA D2 31 2B F5 60 7F 8D 29 2F.

The first 4 bytes (Word 36) of the round key for round 9 are calculated as follows:

i	temp	RotWord	SubWord	Rcon(9)
36	7F8D292F	8D292F7F	5DA515D2	1B000000

XOR	$w[i-4]$	w[i]
46A515D2	EAD27321	AC7766F3

Equivalent Inverse Cipher



Source: Figure 5.10, Stallings 2010

Equivalent Inverse Cipher (cont.)

- Interchanging **InvShiftRows** and **InvSubBytes**:

  $$\text{InvShiftRows}[\text{InvSubBytes}(S_i)] = \text{InvSubBytes}[\text{InvShiftRows}(S_i)]$$

- Interchanging **AddRoundKey** and **InvMixColumns**:
  For a given state S_i and a given round key w_j,

  $$\text{InvMixColumns}(S_i \oplus w_j)$$
  $$= [\text{InvMixColumns}(S_i)] \oplus [\text{InvMixColumns}(w_j)]$$

Implementation in 32-Bit Processes

$$
\begin{bmatrix} e_{0,j} \\ e_{1,j} \\ e_{2,j} \\ e_{3,j} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S[a_{0,j}] \\ S[a_{1,j+1}] \\ S[a_{2,j+2}] \\ S[a_{3,j+3}] \end{bmatrix} \oplus \begin{bmatrix} k_{0,j} \\ k_{1,j} \\ k_{2,j} \\ k_{3,j} \end{bmatrix} =
$$

$$
\left(\begin{bmatrix} 02 \\ 01 \\ 01 \\ 03 \end{bmatrix} \bullet S[a_{0,j}] \right) \oplus \left(\begin{bmatrix} 03 \\ 02 \\ 01 \\ 01 \end{bmatrix} \bullet S[a_{1,j+1}] \right) \oplus \left(\begin{bmatrix} 01 \\ 03 \\ 02 \\ 01 \end{bmatrix} \bullet S[a_{2,j+2}] \right) \oplus
$$

$$
\left(\begin{bmatrix} 01 \\ 01 \\ 03 \\ 02 \end{bmatrix} \bullet S[a_{3,j+3}] \right) \oplus \begin{bmatrix} k_{0,j} \\ k_{1,j} \\ k_{2,j} \\ k_{3,j} \end{bmatrix}
$$

Implementation in 32-Bit Processes (cont.)

To facilitate the preceding calculation, four tables may be defined:

$$T_0(x) = \left(\begin{bmatrix} 02 \\ 01 \\ 01 \\ 03 \end{bmatrix} \bullet S[x] \right); \quad T_1(x) = \left(\begin{bmatrix} 03 \\ 02 \\ 01 \\ 01 \end{bmatrix} \bullet S[x] \right)$$

$$T_2(x) = \left(\begin{bmatrix} 01 \\ 03 \\ 02 \\ 01 \end{bmatrix} \bullet S[x] \right); \quad T_3(x) = \left(\begin{bmatrix} 01 \\ 01 \\ 03 \\ 02 \end{bmatrix} \bullet S[x] \right)$$

These tables can be pre-computed.