This is the entry point to the paper-based Manual_AES algorithm. Before proceeding further, please check that you have all required forms in the required number, as the kit formulation depends on key length, plaintext message length and mode of operation. Following is an example of supported mode of operation with inventory of page to print :

Cipher Configuration	AES 128 bit, 16 char plaintext, ECB 1-block
Round number	10
Required forms	• KeySchedule::128::Main : 1 copy of 1 page • KeySchedule::128::3to10 : 1 copy of 3 pages • PlainTextInput (contains first round) : 1 copy of 1 page • EncryptRound : **8 copy** of 1 page • FinalRound : 1 copy of 1 page
ASCII Key length	16 char (128 bits)

Please make sure to also have the annex documents, required for all modes :

- ASCII converter table w/ nibble XOR table (Annex::ASCII)
- AES S-Box Direct and Reverse (Annex::S-Box)
- Rijndael Field Multiplication Lookup Table (Annex::Galois - 2 pages)
- (Optional) Rijndael RCON table

User-Defined Parameters

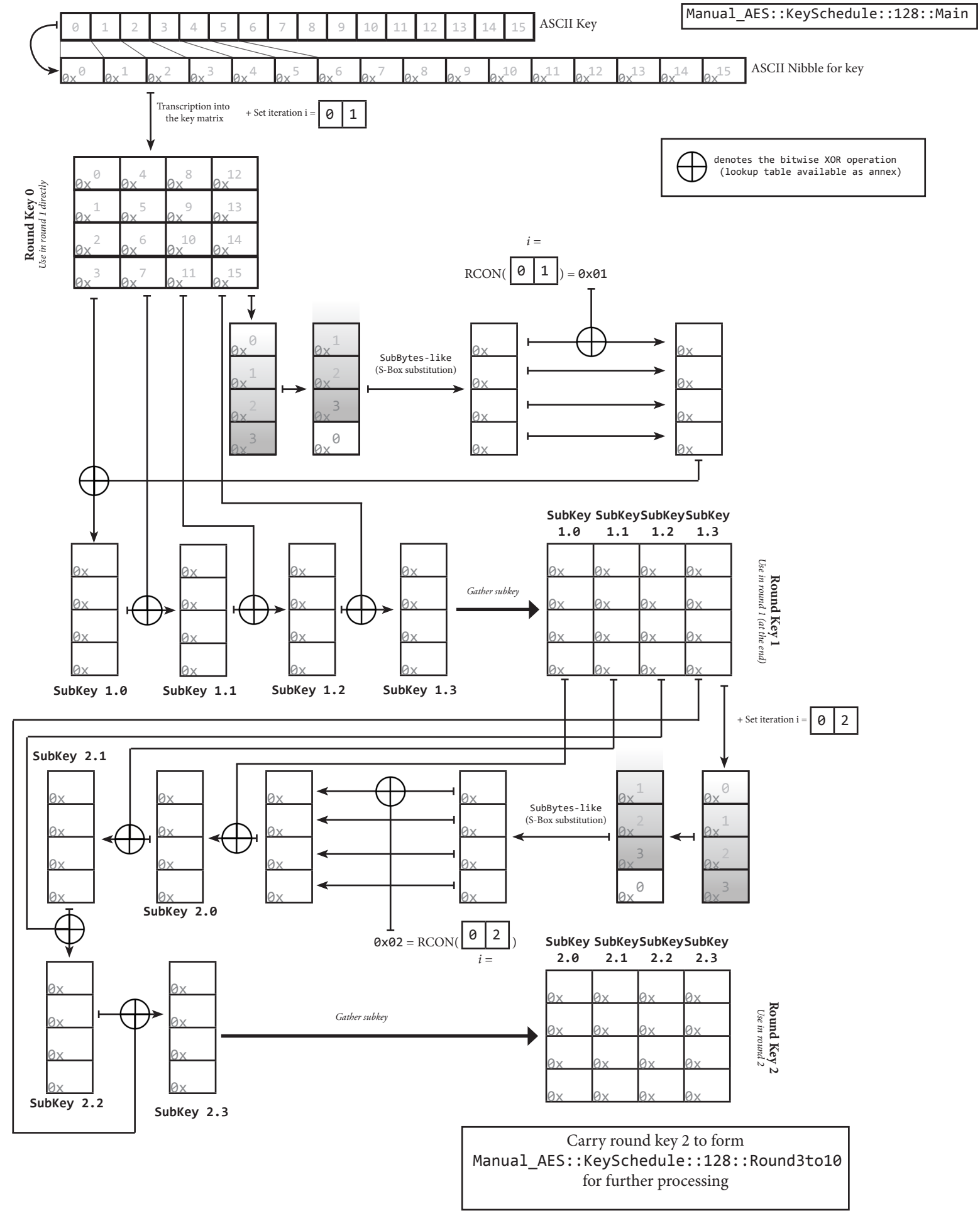
Key Length : | | | | bit (128/192/256)

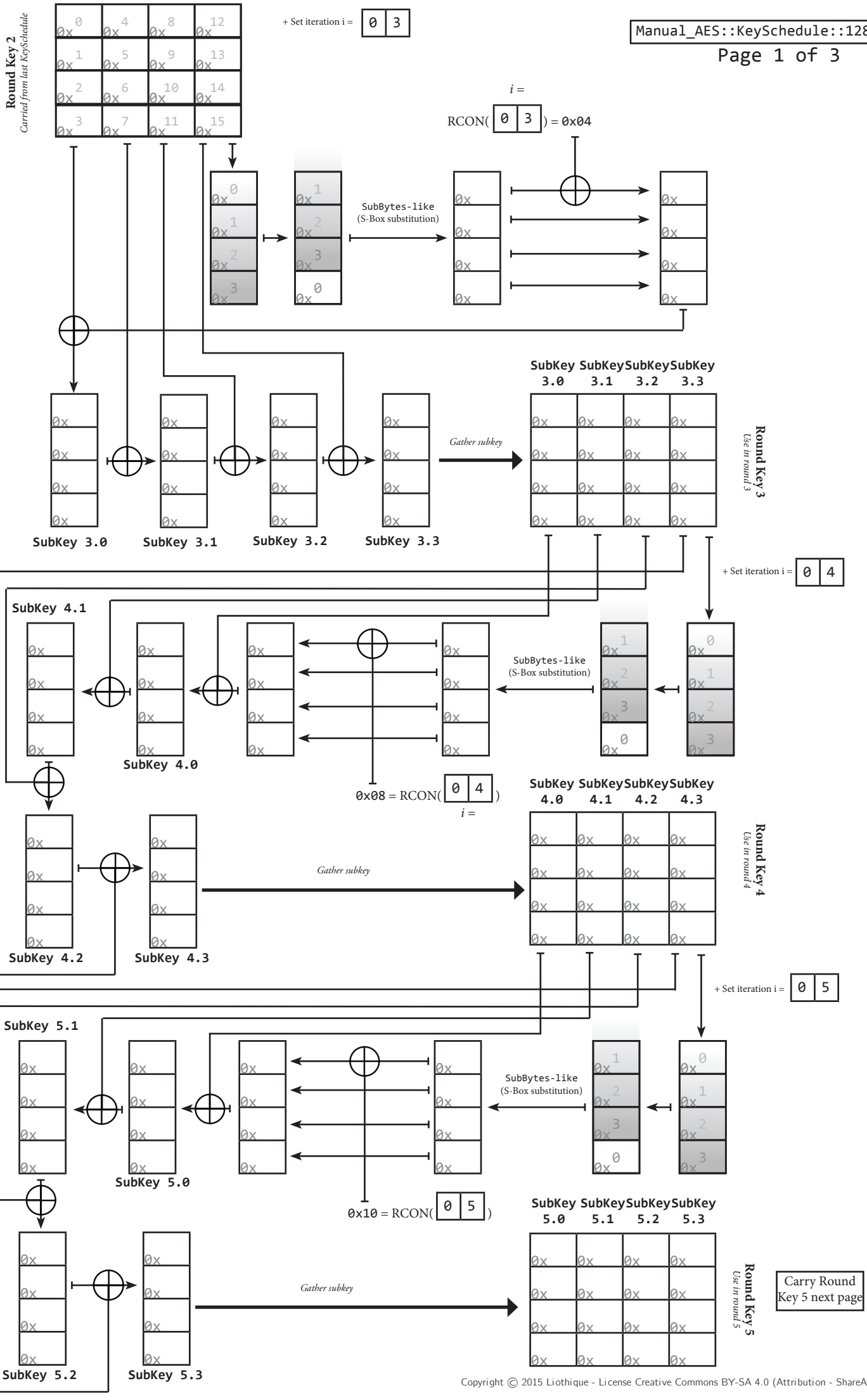
ASCII Key :

128 bit *192 bit* *256 bit*

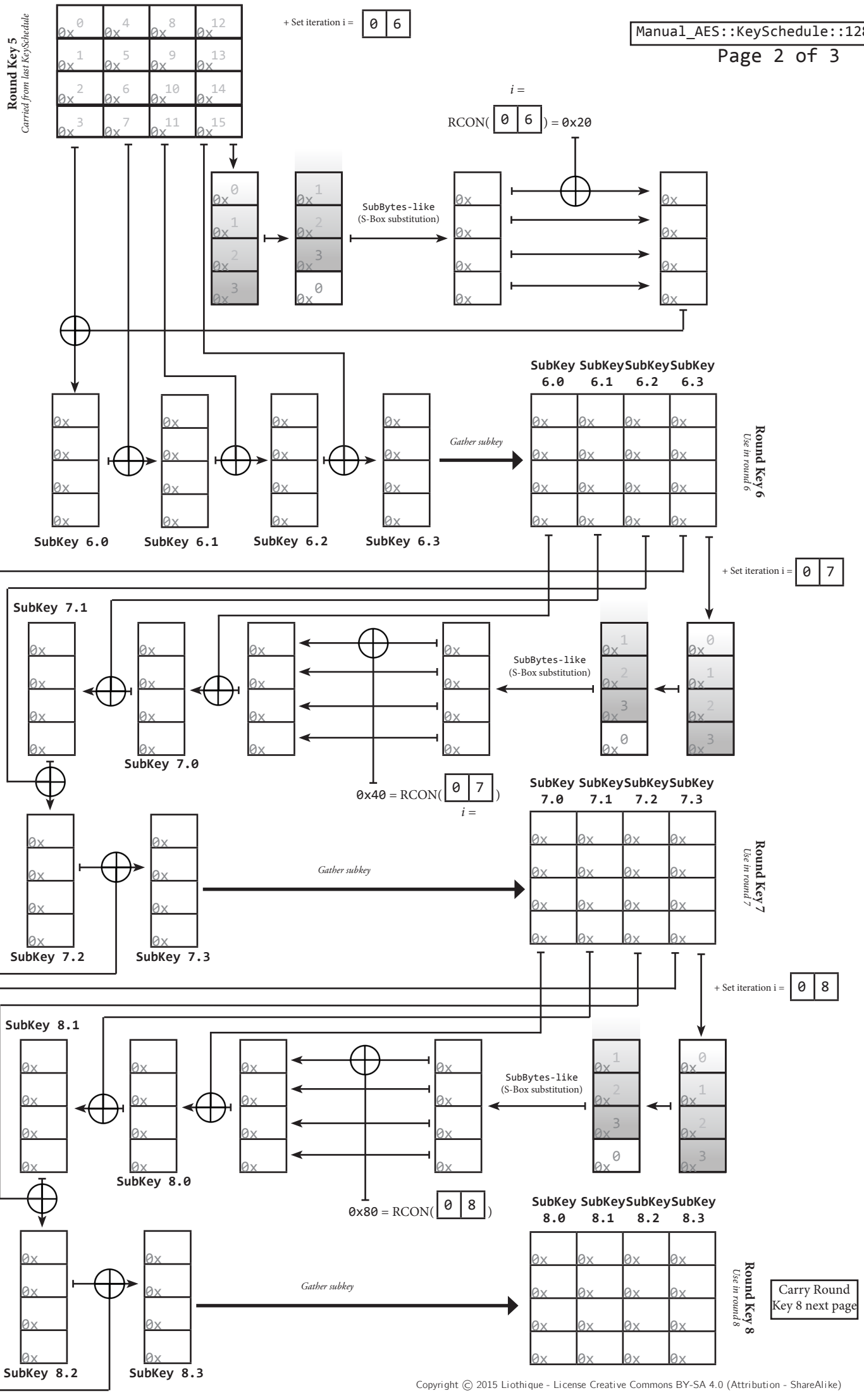
How to proceed ?

Start with Manual_AES::KeySchedule::128::Main and follow the instructions from there to Manual_AES::FinalRound. A example run could be : KeySchedule::Main > Key::Schedule::3to10 > PlainTextInput > EncryptRound 8 times > FinalRound

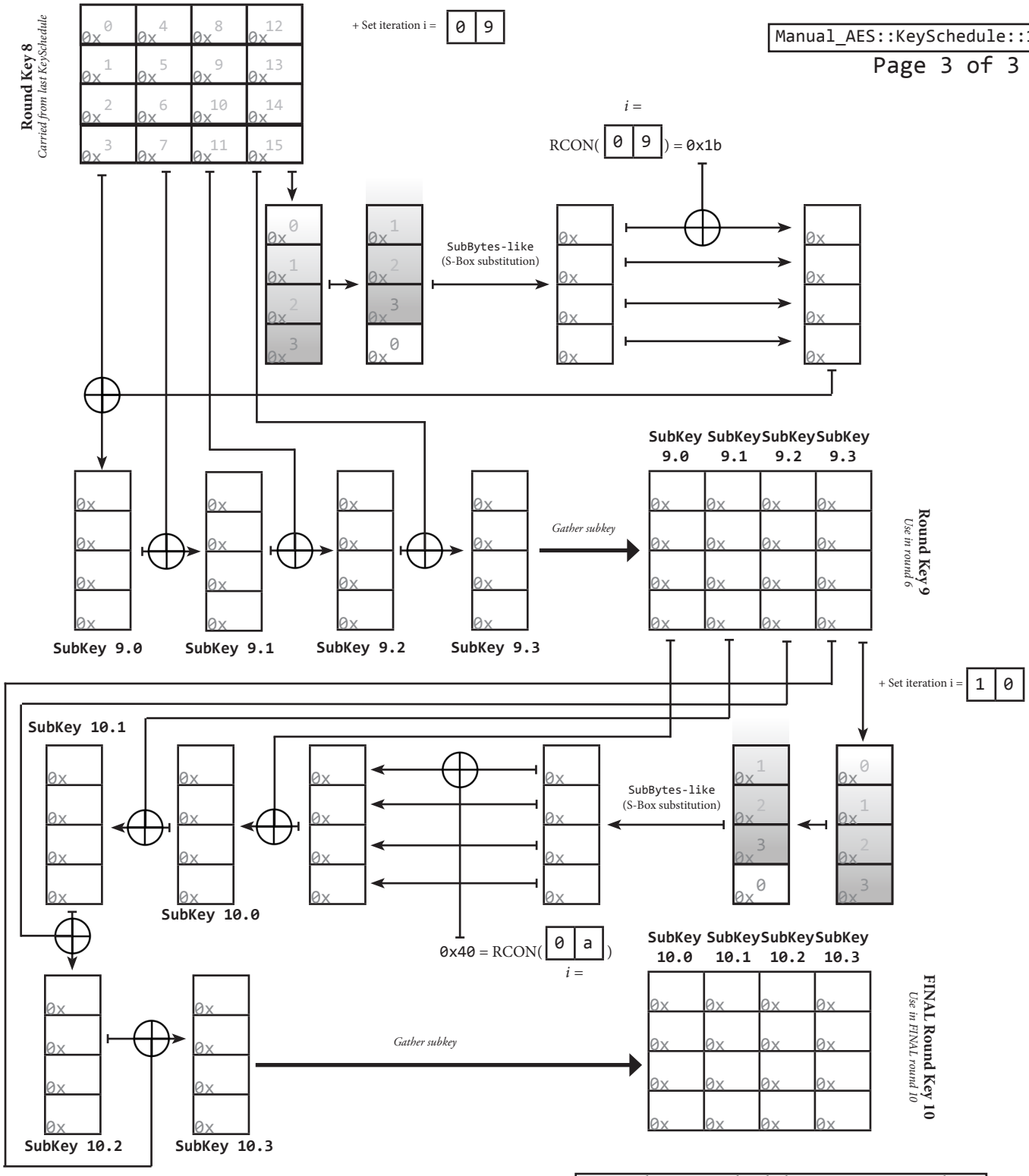




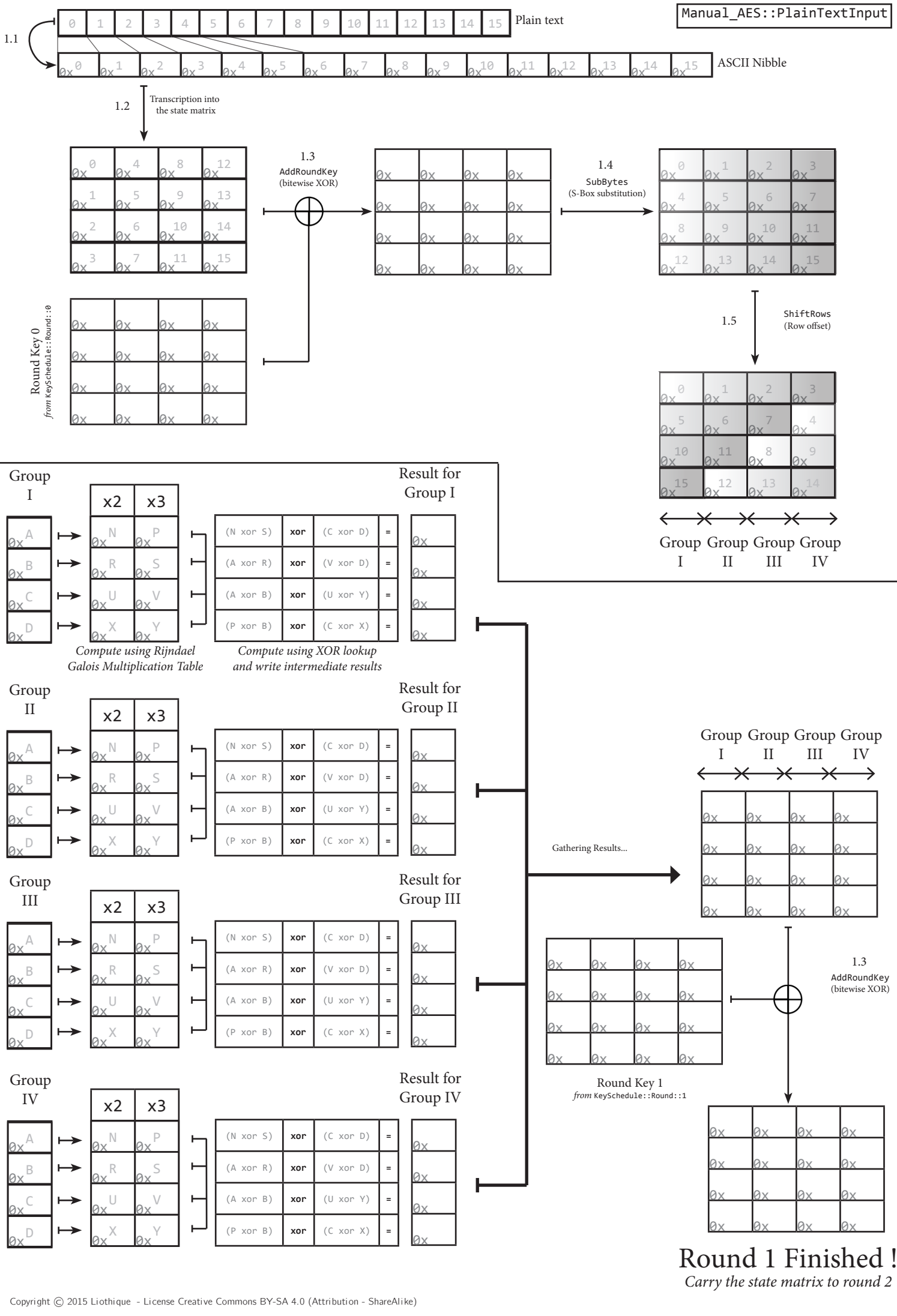
Carry Round Key 5 next page



Carry Round Key 8 next page



Round Key Scheduling now complete. Proceeds to `Manual_AES::PlainTextInput` for encryption rounds.

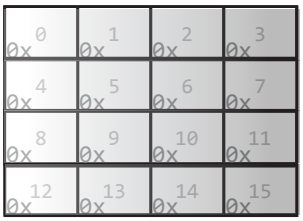


Round n = [ ][ ] (n > 1)

Manual_AES::EncryptRound

State Matrix
(Carried from previous round n-1)

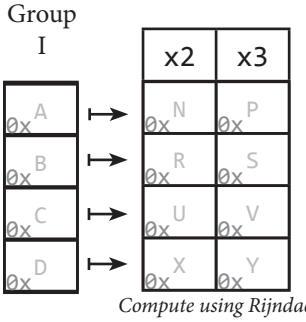
1.4
SubBytes
(S-Box substitution)



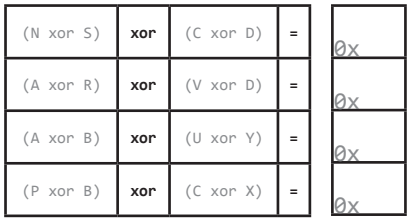
1.5
ShiftRows
(Row offset)

Group I | Group II | Group III | Group IV

Group I



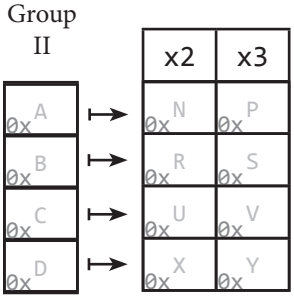
Compute using Rijndael Galois Multiplication Table

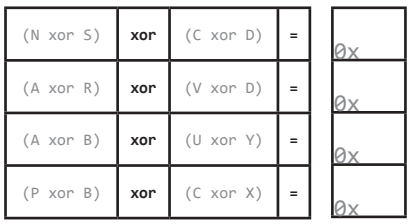


Compute using XOR lookup and write intermediate results

Result for Group I

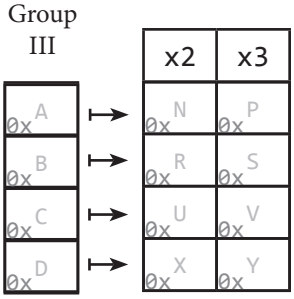
Group II

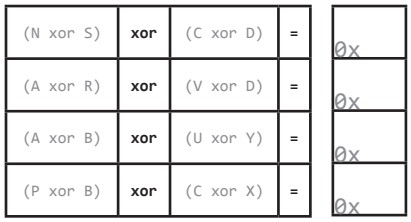




Result for Group II

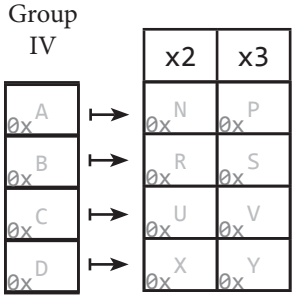
Group III

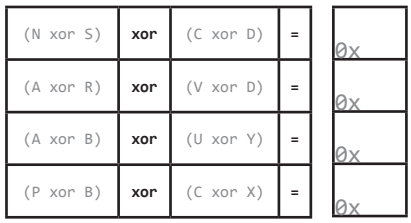




Result for Group III

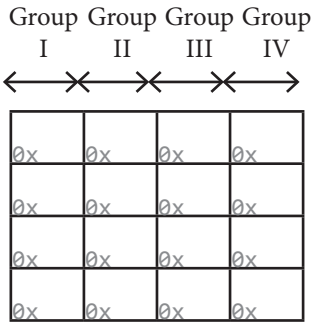
Group IV



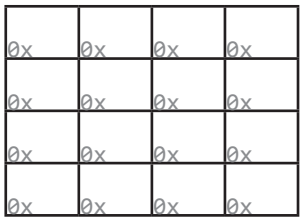


Result for Group IV

Gathering Results...



Group I | Group II | Group III | Group IV



Round Key n
from KeySchedule::Round::n

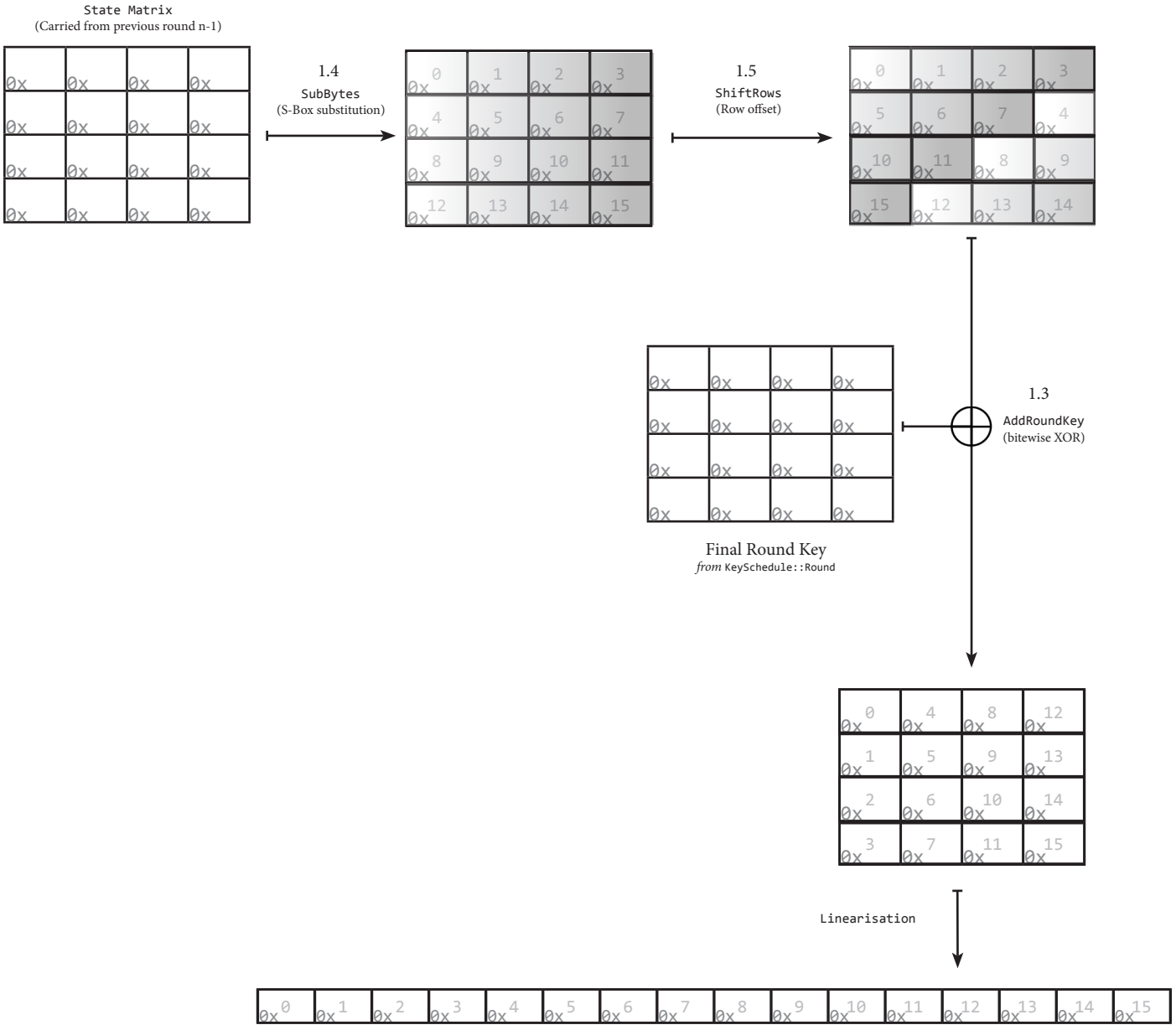
1.3
AddRoundKey
(bitewise XOR)

Round n Finished !

Carry the state matrix to round n+1 or FinalRound if appropriate

Round n = 10 (AES 128 bit), 12 (AES 192 bit), 14 (AES 256 bit)

`Manual_AES::FinalRound`



Final Round Finished !

Encryption is successul.

Annex

- ASCII converter table w/ nibble XOR table (`Annex::ASCII`)
- AES S-Box Direct and Reverse (`Annex::S-Box`)
- Rijndael Field Multiplication Lookup Table (`Manual_AES::Annex:Galois - 2 pages`)
- (Optional) Rijndael RCON table

Nibble					0x0•			0x1•			0x2•			0x3•			0x4•			0x5•			0x6•			0x7•		
B7					0			0			0			0			1			1			1			1		
	B6					0			0			1			1			0			0			1			1	
		B5					0			1			0			1			0			1			0			1
Nibble	B4	B3	B2	B1																								
0x•0	0	0	0	0	*NULL*						*SPACE*			0			@			P			`			p		
0x•1	0	0	0	1							!			1			**A**			**Q**			a			q		
0x•2	0	0	1	0							“			2			**B**			**R**			b			r		
0x•3	0	0	1	1							#			3			**C**			**S**			c			s		
0x•4	0	1	0	0							$			4			**D**			**T**			d			t		
0x•5	0	1	0	1							%			5			**E**			**U**			e			u		
0x•6	0	1	1	0							&			6			**F**			**V**			f			v		
0x•7	0	1	1	1	*BELL*						‘			7			**G**			**W**			g			w		
0x•8	1	0	0	0							(			8			**H**			**X**			h			x		
0x•9	1	0	0	1							)			9			**I**			**Y**			i			y		
0x•a	1	0	1	0	*LF*						*			:			**J**			**Z**			j			z		
0x•b	1	0	1	1				*ESC*			+			;			**K**			[			k			{		
0x•c	1	1	0	0							,			<			**L**			\			l			|		
0x•d	1	1	0	1	*CR*						-			=			**M**			]			m			}		
0x•e	1	1	1	0							.			>			**N**			^			n			~		
0x•f	1	1	1	1							/			?			**O**			_			o			*DEL*		

Lookup table for bitwise XOR on hexadecimal

To XOR two nibble proceed “digit” per “digit” (4 bit per 4 bit) with this table

XOR	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	0	3	2	5	4	7	6	9	8	B	A	D	C	F	E
2	2	3	0	1	6	7	4	5	A	B	8	9	E	F	C	D
3	3	2	1	0	7	6	5	4	B	A	9	8	F	E	D	C
4	4	5	6	7	0	1	2	3	C	D	E	F	8	9	A	B
5	5	4	7	6	1	0	3	2	D	C	F	E	9	8	B	A
6	6	7	4	5	2	3	0	1	E	F	C	D	A	B	8	9
7	7	6	5	4	3	2	1	0	F	E	D	C	B	A	9	8
8	8	9	A	B	C	D	E	F	0	1	2	3	4	5	6	7
9	9	8	B	A	D	C	F	E	1	0	3	2	5	4	7	6
A	A	B	8	9	E	F	C	D	2	3	0	1	6	7	4	5
B	B	A	9	8	F	E	D	C	3	2	1	0	7	6	5	4
C	C	D	E	F	8	9	A	B	4	5	6	7	0	1	2	3
D	D	C	F	E	9	8	B	A	5	4	7	6	1	0	3	2
E	E	F	C	D	A	B	8	9	6	7	4	5	2	3	0	1
F	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0

Revsersable substitution box, to be used nibble by nibble (e.g. `0x64` becomes `0x43`)

Direct S-Box (Encryption)

	0x•0	0x•1	0x•2	0x•3	0x•4	0x•5	0x•6	0x•7	0x•8	0x•9	0x•a	0x•b	0x•c	0x•d	0x•e	0x•f
0x0•	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
0x1•	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
0x2•	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
0x3•	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
0x4•	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
0x5•	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
0x6•	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
0x7•	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
0x8•	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
0x9•	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
0xa•	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
0xb•	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
0xc•	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
0xd•	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
0xe•	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
0xf•	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Reverse S-Box (Decryption)

	0x•0	0x•1	0x•2	0x•3	0x•4	0x•5	0x•6	0x•7	0x•8	0x•9	0x•a	0x•b	0x•c	0x•d	0x•e	0x•f
0x0•	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
0x1•	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
0x2•	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
0x3•	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
0x4•	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
0x5•	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
0x6•	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
0x7•	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
0x8•	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
0x9•	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
0xa•	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
0xb•	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
0xc•	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
0xd•	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
0xe•	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
0xf•	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

The MixColumn step takes place in the Rijndael Galois field; We use lookup table instead of the real Galois multiplication operator. For example, `0x8b` multiplied by 2 is `0x0d`.

	0x•0	0x•1	0x•2	0x•3	0x•4	0x•5	0x•6	0x•7	0x•8	0x•9	0x•a	0x•b	0x•c	0x•d	0x•e	0x•f
0x0•	00	02	04	06	08	0a	0c	0e	10	12	14	16	18	1a	1c	1e
0x1•	20	22	24	26	28	2a	2c	2e	30	32	34	36	38	3a	3c	3e
0x2•	40	42	44	46	48	4a	4c	4e	50	52	54	56	58	5a	5c	5e
0x3•	60	62	64	66	68	6a	6c	6e	70	72	74	76	78	7a	7c	7e
0x4•	80	82	84	86	88	8a	8c	8e	90	92	94	96	98	9a	9c	9e
0x5•	a0	a2	a4	a6	a8	aa	ac	ae	b0	b2	b4	b6	b8	ba	bc	be
0x6•	c0	c2	c4	c6	c8	ca	cc	ce	d0	d2	d4	d6	d8	da	dc	de
0x7•	e0	e2	e4	e6	e8	ea	ec	ee	f0	f2	f4	f6	f8	fa	fc	fe
0x8•	1b	19	1f	1d	13	11	17	15	0b	09	0f	0d	03	01	07	05
0x9•	3b	39	3f	3d	33	31	37	35	2b	29	2f	2d	23	21	27	25
0xa•	5b	59	5f	5d	53	51	57	55	4b	49	4f	4d	43	41	47	45
0xb•	7b	79	7f	7d	73	71	77	75	6b	69	6f	6d	63	61	67	65
0xc•	9b	99	9f	9d	93	91	97	95	8b	89	8f	8d	83	81	87	85
0xd•	bb	b9	bf	bd	b3	b1	b7	b5	ab	a9	af	ad	a3	a1	a7	a5
0xe•	db	d9	df	dd	d3	d1	d7	d5	cb	c9	cf	cd	c3	c1	c7	c5
0xf•	fb	f9	ff	fd	f3	f1	f7	f5	eb	e9	ef	ed	e3	e1	e7	e5

	0x•0	0x•1	0x•2	0x•3	0x•4	0x•5	0x•6	0x•7	0x•8	0x•9	0x•a	0x•b	0x•c	0x•d	0x•e	0x•f
0x0•	00	03	06	05	0c	0f	0a	09	18	1b	1e	1d	14	17	12	11
0x1•	30	33	36	35	3c	3f	3a	39	28	2b	2e	2d	24	27	22	21
0x2•	60	63	66	65	6c	6f	6a	69	78	7b	7e	7d	74	77	72	71
0x3•	50	53	56	55	5c	5f	5a	59	48	4b	4e	4d	44	47	42	41
0x4•	c0	c3	c6	c5	cc	cf	ca	c9	d8	db	de	dd	d4	d7	d2	d1
0x5•	f0	f3	f6	f5	fc	ff	fa	f9	e8	eb	ee	ed	e4	e7	e2	e1
0x6•	a0	a3	a6	a5	ac	af	aa	a9	b8	bb	be	bd	b4	b7	b2	b1
0x7•	90	93	96	95	9c	9f	9a	99	88	8b	8e	8d	84	87	82	81
0x8•	9b	98	9d	9e	97	94	91	92	83	80	85	86	8f	8c	89	8a
0x9•	ab	a8	ad	ae	a7	a4	a1	a2	b3	b0	b5	b6	bf	bc	b9	ba
0xa•	fb	f8	fd	fe	f7	f4	f1	f2	e3	e0	e5	e6	ef	ec	e9	ea
0xb•	cb	c8	cd	ce	c7	c4	c1	c2	d3	d0	d5	d6	df	dc	d9	da
0xc•	5b	58	5d	5e	57	54	51	52	43	40	45	46	4f	4c	49	4a
0xd•	6b	68	6d	6e	67	64	61	62	73	70	75	76	7f	7c	79	7a
0xe•	3b	38	3d	3e	37	34	31	32	23	20	25	26	2f	2c	29	2a
0xf•	0b	08	0d	0e	07	04	01	02	13	10	15	16	1f	1c	19	1a

	0x•0	0x•1	0x•2	0x•3	0x•4	0x•5	0x•6	0x•7	0x•8	0x•9	0x•a	0x•b	0x•c	0x•d	0x•e	0x•f
0x0•	00	09	12	1b	24	2d	36	3f	48	41	5a	53	6c	65	7e	77
0x1•	90	99	82	8b	b4	bd	a6	af	d8	d1	ca	c3	fc	f5	ee	e7
0x2•	3b	32	29	20	1f	16	0d	04	73	7a	61	68	57	5e	45	4c
0x3•	ab	a2	b9	b0	8f	86	9d	94	e3	ea	f1	f8	c7	ce	d5	dc
0x4•	76	7f	64	6d	52	5b	40	49	3e	37	2c	25	1a	13	08	01
0x5•	e6	ef	f4	fd	c2	cb	d0	d9	ae	a7	bc	b5	8a	83	98	91
0x6•	4d	44	5f	56	69	60	7b	72	05	0c	17	1e	21	28	33	3a
0x7•	dd	d4	cf	c6	f9	f0	eb	e2	95	9c	87	8e	b1	b8	a3	aa
0x8•	ec	e5	fe	f7	c8	c1	da	d3	a4	ad	b6	bf	80	89	92	9b
0x9•	7c	75	6e	67	58	51	4a	43	34	3d	26	2f	10	19	02	0b
0xa•	d7	de	c5	cc	f3	fa	e1	e8	9f	96	8d	84	bb	b2	a9	a0
0xb•	47	4e	55	5c	63	6a	71	78	0f	06	1d	14	2b	22	39	30
0xc•	9a	93	88	81	be	b7	ac	a5	d2	db	c0	c9	f6	ff	e4	ed
0xd•	0a	03	18	11	2e	27	3c	35	42	4b	50	59	66	6f	74	7d
0xe•	a1	a8	b3	ba	85	8c	97	9e	e9	e0	fb	f2	cd	c4	df	d6
0xf•	31	38	23	2a	15	1c	07	0e	79	70	6b	62	5d	54	4f	46

	0x•0	0x•1	0x•2	0x•3	0x•4	0x•5	0x•6	0x•7	0x•8	0x•9	0x•a	0x•b	0x•c	0x•d	0x•e	0x•f
0x0•	00	0b	16	1d	2c	27	3a	31	58	53	4e	45	74	7f	62	69
0x1•	b0	bb	a6	ad	9c	97	8a	81	e8	e3	fe	f5	c4	cf	d2	d9
0x2•	7b	70	6d	66	57	5c	41	4a	23	28	35	3e	0f	04	19	12
0x3•	cb	c0	dd	d6	e7	ec	f1	fa	93	98	85	8e	bf	b4	a9	a2
0x4•	f6	fd	e0	eb	da	d1	cc	c7	ae	a5	b8	b3	82	89	94	9f
0x5•	46	4d	50	5b	6a	61	7c	77	1e	15	08	03	32	39	24	2f
0x6•	8d	86	9b	90	a1	aa	b7	bc	d5	de	c3	c8	f9	f2	ef	e4
0x7•	3d	36	2b	20	11	1a	07	0c	65	6e	73	78	49	42	5f	54
0x8•	f7	fc	e1	ea	db	d0	cd	c6	af	a4	b9	b2	83	88	95	9e
0x9•	47	4c	51	5a	6b	60	7d	76	1f	14	09	02	33	38	25	2e
0xa•	8c	87	9a	91	a0	ab	b6	bd	d4	df	c2	c9	f8	f3	ee	e5
0xb•	3c	37	2a	21	10	1b	06	0d	64	6f	72	79	48	43	5e	55
0xc•	01	0a	17	1c	2d	26	3b	30	59	52	4f	44	75	7e	63	68
0xd•	b1	ba	a7	ac	9d	96	8b	80	e9	e2	ff	f4	c5	ce	d3	d8
0xe•	7a	71	6c	67	56	5d	40	4b	22	29	34	3f	0e	05	18	13
0xf•	ca	c1	dc	d7	e6	ed	f0	fb	92	99	84	8f	be	b5	a8	a3

	0x•0	0x•1	0x•2	0x•3	0x•4	0x•5	0x•6	0x•7	0x•8	0x•9	0x•a	0x•b	0x•c	0x•d	0x•e	0x•f
0x0•	00	0d	1a	17	34	39	2e	23	68	65	72	7f	5c	51	46	4b
0x1•	d0	dd	ca	c7	e4	e9	fe	f3	b8	b5	a2	af	8c	81	96	9b
0x2•	bb	b6	a1	ac	8f	82	95	98	d3	de	c9	c4	e7	ea	fd	f0
0x3•	6b	66	71	7c	5f	52	45	48	03	0e	19	14	37	3a	2d	20
0x4•	6d	60	77	7a	59	54	43	4e	05	08	1f	12	31	3c	2b	26
0x5•	bd	b0	a7	aa	89	84	93	9e	d5	d8	cf	c2	e1	ec	fb	f6
0x6•	d6	db	cc	c1	e2	ef	f8	f5	be	b3	a4	a9	8a	87	90	9d
0x7•	06	0b	1c	11	32	3f	28	25	6e	63	74	79	5a	57	40	4d
0x8•	da	d7	c0	cd	ee	e3	f4	f9	b2	bf	a8	a5	86	8b	9c	91
0x9•	0a	07	10	1d	3e	33	24	29	62	6f	78	75	56	5b	4c	41
0xa•	61	6c	7b	76	55	58	4f	42	09	04	13	1e	3d	30	27	2a
0xb•	b1	bc	ab	a6	85	88	9f	92	d9	d4	c3	ce	ed	e0	f7	fa
0xc•	b7	ba	ad	a0	83	8e	99	94	df	d2	c5	c8	eb	e6	f1	fc
0xd•	67	6a	7d	70	53	5e	49	44	0f	02	15	18	3b	36	21	2c
0xe•	0c	01	16	1b	38	35	22	2f	64	69	7e	73	50	5d	4a	47
0xf•	dc	d1	c6	cb	e8	e5	f2	ff	b4	b9	ae	a3	80	8d	9a	97

	0x•0	0x•1	0x•2	0x•3	0x•4	0x•5	0x•6	0x•7	0x•8	0x•9	0x•a	0x•b	0x•c	0x•d	0x•e	0x•f
0x0•	00	0e	1c	12	38	36	24	2a	70	7e	6c	62	48	46	54	5a
0x1•	e0	ee	fc	f2	d8	d6	c4	ca	90	9e	8c	82	a8	a6	b4	ba
0x2•	db	d5	c7	c9	e3	ed	ff	f1	ab	a5	b7	b9	93	9d	8f	81
0x3•	3b	35	27	29	03	0d	1f	11	4b	45	57	59	73	7d	6f	61
0x4•	ad	a3	b1	bf	95	9b	89	87	dd	d3	c1	cf	e5	eb	f9	f7
0x5•	4d	43	51	5f	75	7b	69	67	3d	33	21	2f	05	0b	19	17
0x6•	76	78	6a	64	4e	40	52	5c	06	08	1a	14	3e	30	22	2c
0x7•	96	98	8a	84	ae	a0	b2	bc	e6	e8	fa	f4	de	d0	c2	cc
0x8•	41	4f	5d	53	79	77	65	6b	31	3f	2d	23	09	07	15	1b
0x9•	a1	af	bd	b3	99	97	85	8b	d1	df	cd	c3	e9	e7	f5	fb
0xa•	9a	94	86	88	a2	ac	be	b0	ea	e4	f6	f8	d2	dc	ce	c0
0xb•	7a	74	66	68	42	4c	5e	50	0a	04	16	18	32	3c	2e	20
0xc•	ec	e2	f0	fe	d4	da	c8	c6	9c	92	80	8e	a4	aa	b8	b6
0xd•	0c	02	10	1e	34	3a	28	26	7c	72	60	6e	44	4a	58	56
0xe•	37	39	2b	25	0f	01	13	1d	47	49	5b	55	7f	71	63	6d
0xf•	d7	d9	cb	c5	ef	e1	f3	fd	a7	a9	bb	b5	9f	91	83	8d

RCON (Finite field exponentiation in the Rijndael field) used in the key schedule

	0x•0	0x•1	0x•2	0x•3	0x•4	0x•5	0x•6	0x•7	0x•8	0x•9	0x•a	0x•b	0x•c	0x•d	0x•e	0x•f
0x0•	**8d**	**01**	**02**	**04**	**08**	**10**	**20**	**40**	**80**	**1b**	**36**	6c	d8	ab	4d	9a
0x1•	2f	5e	bc	63	c6	97	35	6a	d4	b3	7d	fa	ef	c5	91	39
0x2•	72	e4	d3	bd	61	c2	9f	25	4a	94	33	66	cc	83	1d	3a
0x3•	74	e8	cb	8d	01	02	04	08	10	20	40	80	1b	36	6c	d8
0x4•	ab	4d	9a	2f	5e	bc	63	c6	97	35	6a	d4	b3	7d	fa	ef
0x5•	c5	91	39	72	e4	d3	bd	61	c2	9f	25	4a	94	33	66	cc
0x6•	83	1d	3a	74	e8	cb	8d	01	02	04	08	10	20	40	80	1b
0x7•	36	6c	d8	ab	4d	9a	2f	5e	bc	63	c6	97	35	6a	d4	b3
0x8•	7d	fa	ef	c5	91	39	72	e4	d3	bd	61	c2	9f	25	4a	94
0x9•	33	66	cc	83	1d	3a	74	e8	cb	8d	01	02	04	08	10	20
0xa•	40	80	1b	36	6c	d8	ab	4d	9a	2f	5e	bc	63	c6	97	35
0xb•	6a	d4	b3	7d	fa	ef	c5	91	39	72	e4	d3	bd	61	c2	9f
0xc•	25	4a	94	33	66	cc	83	1d	3a	74	e8	cb	8d	01	02	04
0xd•	08	10	20	40	80	1b	36	6c	d8	ab	4d	9a	2f	5e	bc	63
0xe•	c6	97	35	6a	d4	b3	7d	fa	ef	c5	91	39	72	e4	d3	bd
0xf•	61	c2	9f	25	4a	94	33	66	cc	83	1d	3a	74	e8	cb	8d