Advanced Encryption Standard (AES)

Nikolay Kaleyski

Learning objectives

- What is the brief history behind **Rijndael** and **AES**
- What is the **basic structure** of AES
    - **Key addition**, **diffusion**, and **byte substitution** layers
    - **ShiftRows** and **MixColumns**
- How the **key schedule** of AES operates
- How **decryption** is performed

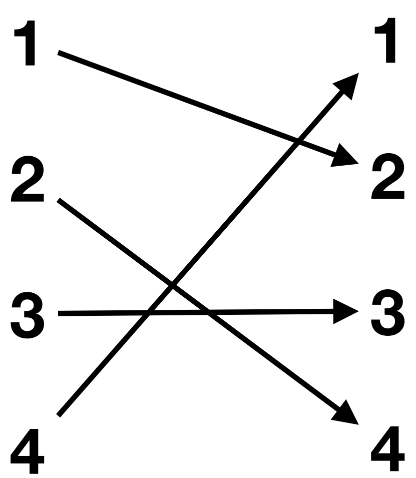
Background

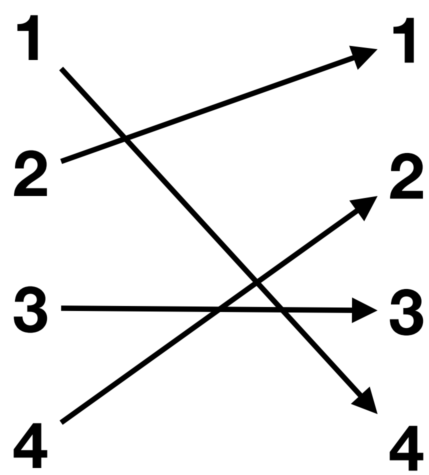
- The Data Encryption Standard (DES) is insecure
- NIST announced in 1997 that it would accept proposals for a new **Advanced Encryption Standard (AES)**
- **Open competition**, proposals evaluated by the community in three consecutive rounds
- The proposals were to be for block ciphers with **128-bit blocks** supporting **keys of length 128, 192 and 256**
- Fifteen proposals submitted: CAST-256, CRYPTON, DEAL, DFC, E2, FROG, HPC, LOKI97, MAGENTA, MARS, RC6, **Rijndael**, SAFER+, Serpent, and Twofish
- In 2000, **Rijndael** (co-designed by Vincent Rijmen and Joan Daemen from KU Leuven) won the competition and became AES
- AES is the most widely used symmetric cipher today
  - everyday communications (WiFi, SSH, Skype, etc.)
  - online banking, electronic commerce, transfer of confidential data, etc.

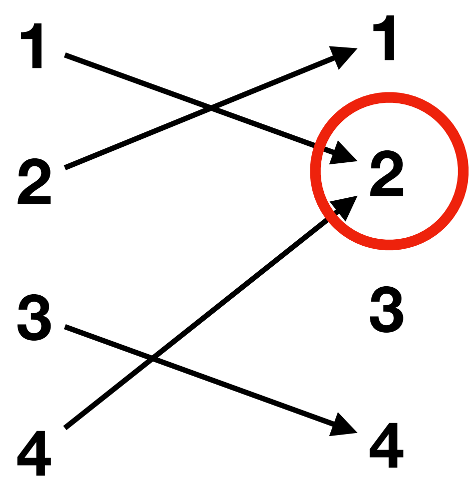
Overview

- Input and output are **128** bit blocks
- Key can be <u>**128**</u>, 192, or 256 bits long
- Like DES, AES encrypts its input in a number of rounds
    - 10 rounds for 128-bit key
    - 12 rounds for 192-bit key; 14 rounds for 256-bit key
- Based on a **substitution-permutation (SPN)** rather than a Feistel network
- AES (as opposed to Rijndael) only specifies encryption in the case of a **128-bit key**, so that the cipher has **10 rounds** in this case

Permutation



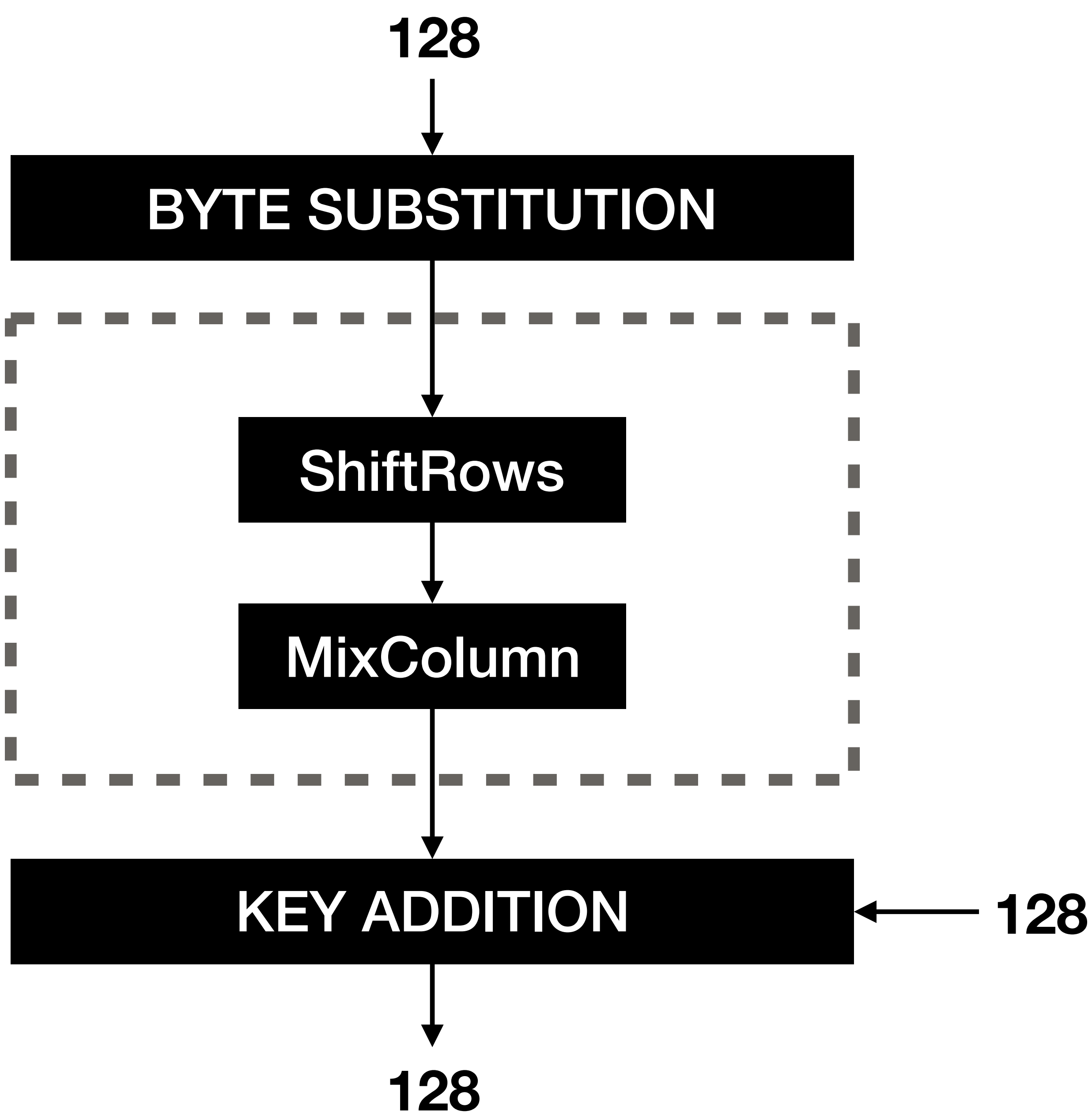




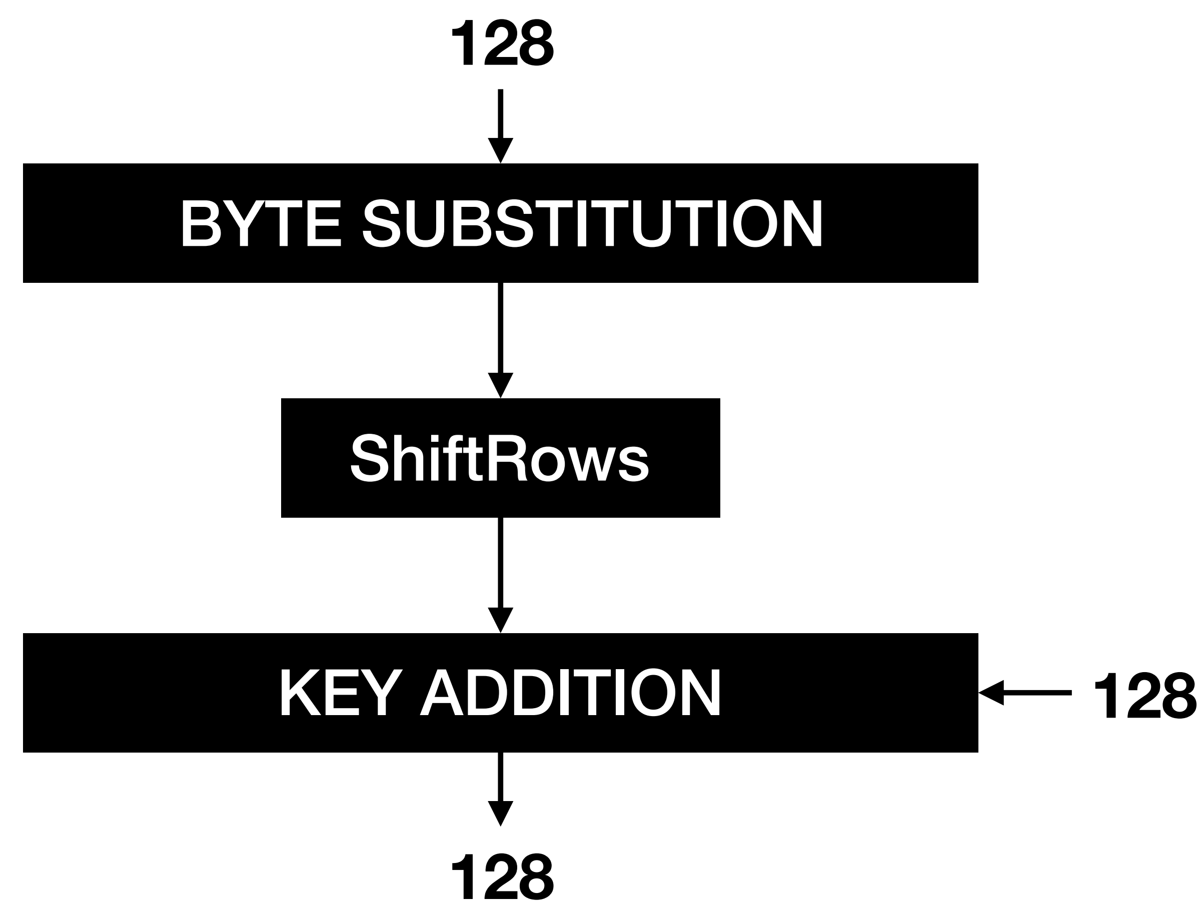
Overview

- Each round consists of **layers** providing confusion and diffusion
- All layers operate on **128-bit inputs/outputs**
  - **Key addition** layer: XOR’s the round key with the input
  - **Byte substitution** layer (S-Box): introduces confusion
  - **Diffusion layer**: consists of two sub-layers:
    - ShiftRows
    - MixColumn

Round operations



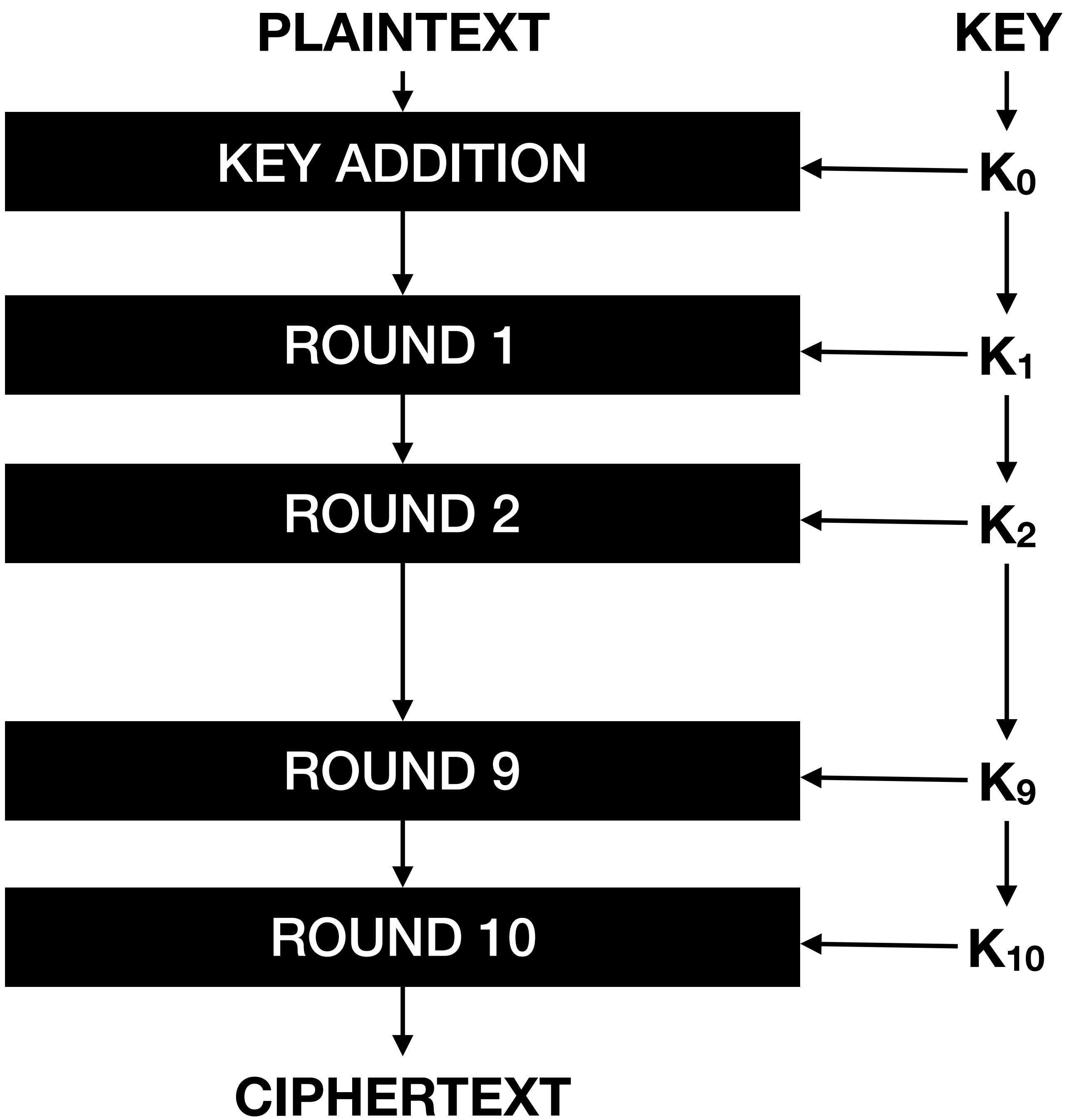
First (r-1) rounds



Last round

Last round does not do MixColumn

AES at a glance



- **Ten rounds** of encryption (byte substitution, diffusion, key addition)
- One extra layer of **key addition** before the first round
- **Eleven sub-keys** generated from the main key by the key schedule

Byte substitution

S	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	1	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	4	C7	23	C3	18	96	5	9A	7	12	80	E2	EB	27	B2	75
4	9	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	0	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	2	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	6	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	8
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	3	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

- The **AES S-box: (8,8)-function**
- The input can be written as a pair of hexadecimal digits, e.g. 00111010 = 3A
- The first digit selects the row and the second selects the column of the output
- Byte substitution can also be described mathematically as **BS(x) = M** $\times$ **x**$^{-1}$ **+ a** for a fixed matrix **M** and a fixed vector **a**

Diffusion

- **ShiftRows** and **MixColumn** sub-layers
    - (except in the last round, which has no MixColumn)
- The **128-bit input** is represented in matrix form is split into **16 bytes** and arranged in a **matrix**

$A0$	$A4$	$A8$	$A12$
$A1$	$A5$	$A9$	$A13$
$A2$	$A6$	$A10$	$A14$
$A3$	$A7$	$A11$	$A15$

ShiftRows

The **second**, **third** and **fourth** row are shifted **one**, **two** and **three** positions to the left, respectively

$A0$	$A4$	$A8$	$A12$
$A1$	$A5$	$A9$	$A13$
$A2$	$A6$	$A10$	$A14$
$A3$	$A7$	$A11$	$A15$

$\longrightarrow$

$A0$	$A4$	$A8$	$A12$
$A5$	$A9$	$A13$	$A1$
$A10$	$A14$	$A2$	$A6$
$A15$	$A3$	$A7$	$A11$

MixColumn

- Multiplies every column (as a vector of elements of $\mathbb{F}_{2^8}$) by a fixed matrix
- Every byte of the input affects four bytes of the output (diffusion)

$$\begin{bmatrix} B0 & B4 & B8 & B12 \\ B1 & B5 & B9 & B13 \\ B2 & B6 & B10 & B14 \\ B3 & B7 & B11 & B15 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \times \begin{bmatrix} A0 & A4 & A8 & A12 \\ A5 & A9 & A13 & A1 \\ A10 & A14 & A2 & A6 \\ A15 & A3 & A7 & A11 \end{bmatrix}$$

MixColumn

$$
\begin{array}{|c|c|c|c|}
\hline
B0 & B4 & B8 & B12 \\ \hline
B1 & B5 & B9 & B13 \\ \hline
B2 & B6 & B10 & B14 \\ \hline
B3 & B7 & B11 & B15 \\ \hline
\end{array}
=
\begin{array}{|c|c|c|c|}
\hline
2 & 3 & 1 & 1 \\ \hline
1 & 2 & 3 & 1 \\ \hline
1 & 1 & 2 & 3 \\ \hline
3 & 1 & 1 & 2 \\ \hline
\end{array}
\times
\begin{array}{|c|c|c|c|}
\hline
A0 & A4 & A8 & A12 \\ \hline
A5 & A9 & A13 & A1 \\ \hline
A10 & A14 & A2 & A6 \\ \hline
A15 & A3 & A7 & A11 \\ \hline
\end{array}
$$

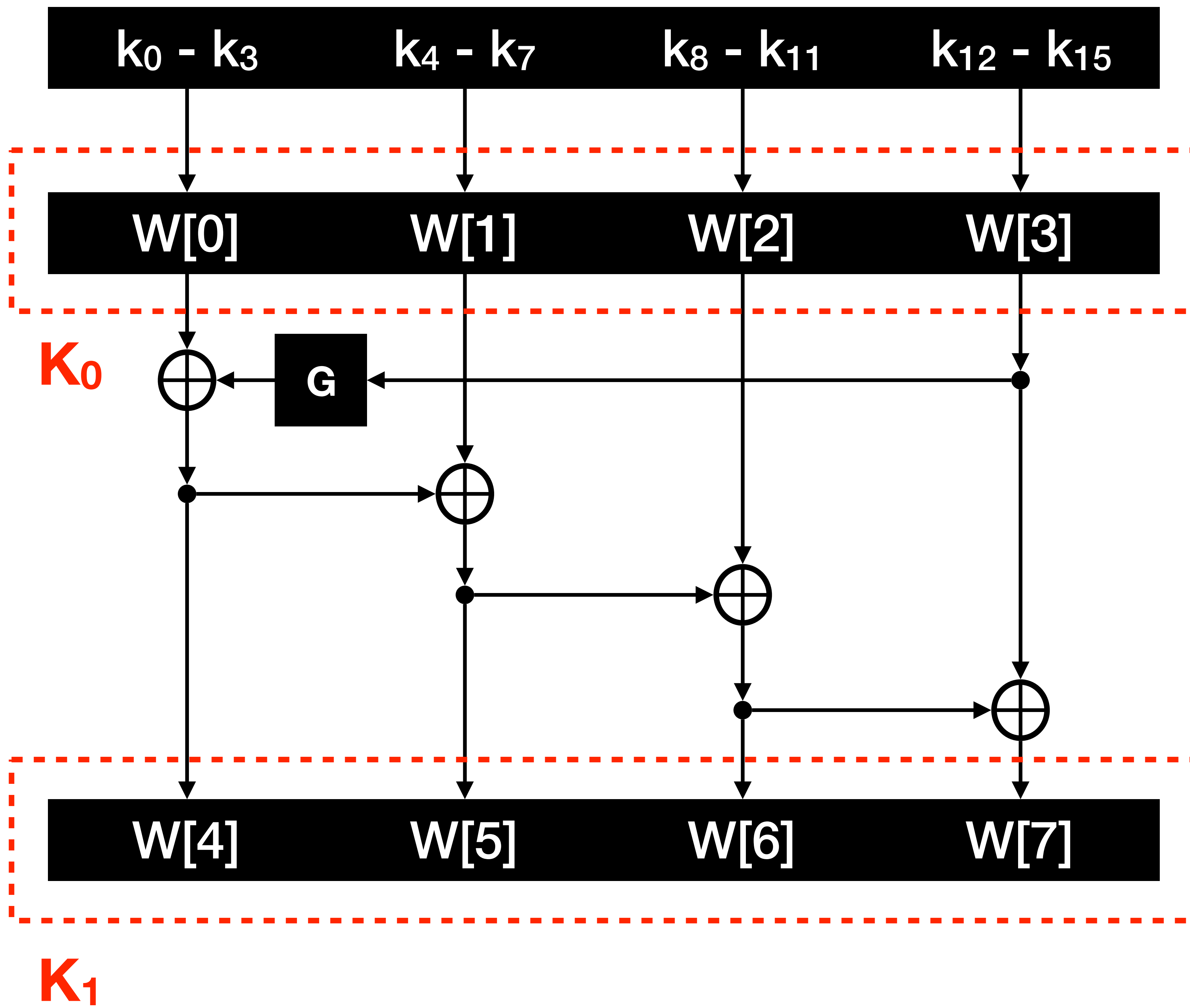
B0 = 2 x A0 + 3 x A5 + 1 x A10 + 1 x A15

B6 = 1 x A4 + 1 x A9 + 2 x A14 + 3 x A3

B11 = 3 x A8 + 1 x A13 + 1 x A2 + 2 x A7

Key addition and key schedule

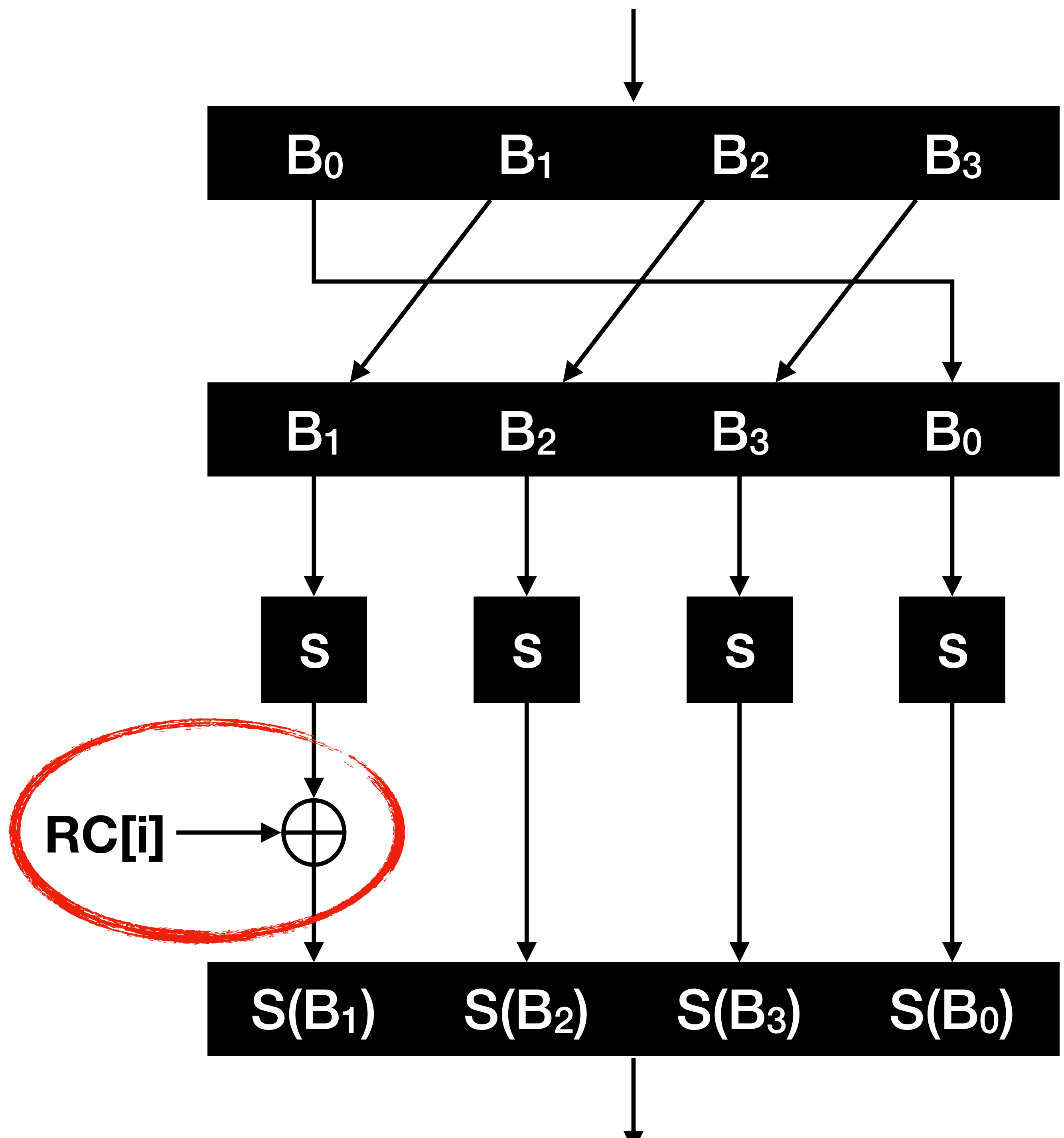
- **Key addition**: XOR with round key
- Main key split into 16 bytes **k_0, k_1, …, k_{15}** which are fed as input to a round network for generating round keys
- The same process is repeated to generate K_1, K_2, …, K_{10}



Key addition and key schedule

- The function G shifts the four bytes in its input to the left and runs each of them through the AES S-box

- One of the bytes is XOR-ed with a *round coefficient*



Key addition and key schedule

The round coefficient for the *i*-th round is α^i as an element of $\mathbb{F}_{2^8}$

RC	00000001	RC_6	0010000
RC	00000010	RC_7	0100000
RC	00000100	RC_8	1000000
RC	00001000	RC_9	00011011
RC	00010000	RC_1	00110110

Decryption

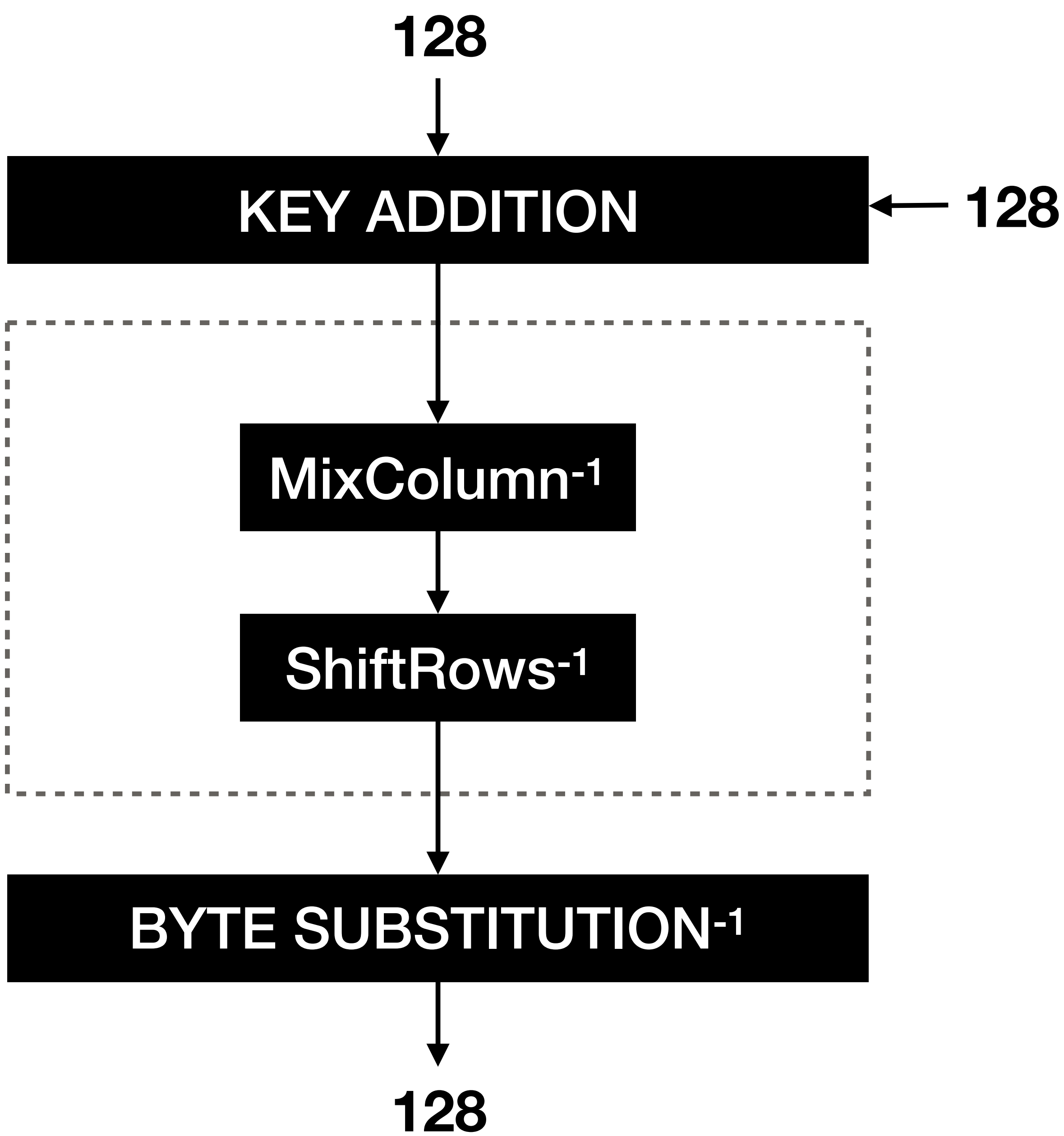
- Like with DES, the encryption is “reversed” round by round, with the first round of decryption “cancelling” the last round of encryption, etc.

- Unlike DES, encryption is not the same as decryption, i.e. it is not sufficient to simply reverse the key schedule

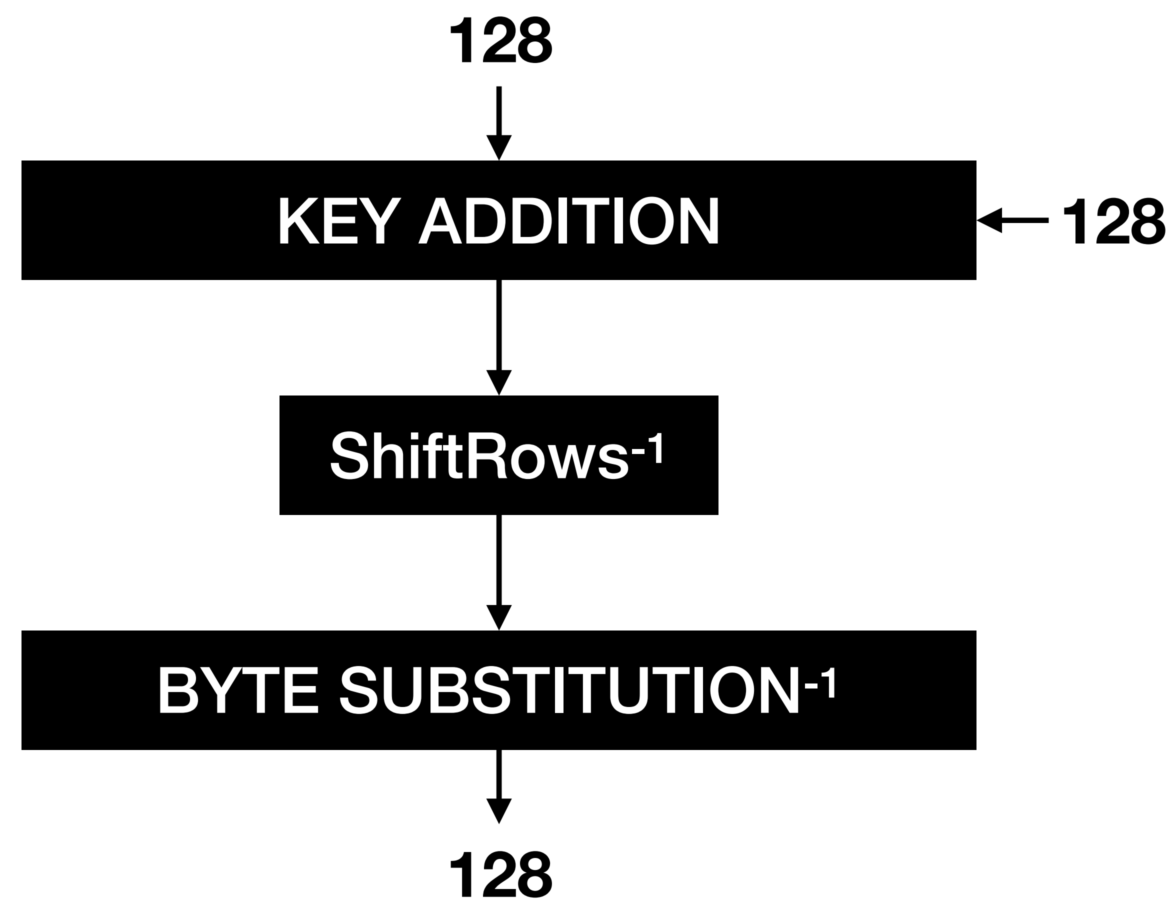
- Each of the four layers (byte substitution, ShiftRows, MixColumn, and key addition) defines an inverse operation

- The inverse operations are applied in the reverse order with respect to encryption

Decryption round operations



Last (r-1) rounds



First round

Inverse MixColumn

- Recall that the *MixColumn* layer multiplied its input (partitioned into 16 byte-sized blocks) by a matrix
- To reverse this operation, we multiply by the **inverse matrix**

$$\begin{array}{|c|c|c|c|}\hline B0 & B4 & B8 & B12 \\ \hline B1 & B5 & B9 & B13 \\ \hline B2 & B6 & B10 & B14 \\ \hline B3 & B7 & B11 & B15 \\ \hline\end{array} = \begin{array}{|c|c|c|c|}\hline E & B & D & 9 \\ \hline 9 & E & B & D \\ \hline D & 9 & E & B \\ \hline B & D & 9 & E \\ \hline\end{array} \times \begin{array}{|c|c|c|c|}\hline A0 & A4 & A8 & A12 \\ \hline A1 & A5 & A9 & A13 \\ \hline A2 & A6 & A10 & A14 \\ \hline A3 & A7 & A11 & A15 \\ \hline\end{array}$$

Inverse ShiftRows

Reverse ShiftRows: shift in the opposite direction

B0	B4	B8	B12
B1	B5	B9	B13
B2	B6	B10	B14
B3	B7	B11	B15

B0	B4	B8	B12
B13	B1	B5	B9
B10	B14	B2	B6
B7	B11	B15	B3

Inverse byte substitution

- The Rijndael S-box is **bijective** so it can be inverted
- Recall byte substitution could be written in the form of $\mathbf{BS(x) = Mx^{-1} + a}$
- From here we can express $\mathbf{x^{-1} = M^{-1}BS(x) + M^{-1}a}$, and hence $\mathbf{x = (M^{-1}BS(x) + M^{-1}a)^{-1}}$
- Alternatively, the inverse operation can be directly derived from the lookup table of the forward function

Inverse byte substitution

S	**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	1	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	4	C7	23	C3	18	96	5	9A	7	12	80	E2	EB	27	B2	75
4	9	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	0	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	2	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	6	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	8
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	3	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

S^{-1}	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
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

Inverse key schedule

- Like in DES, the round keys have to be applied in the **reverse order** during decryption

- The easiest way to do this is to simply **pre-compute all the keys** and keep the in memory at once

- The decrease in efficiency is negligible

What is missing?

$$\mathbb{F}_{2^8}$$

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

Conclusion

- Rijndael is a 128-bit block cipher based on an SPN and offers several variants in terms of key length
- The 128-bit key variant is the Advanced Encryption Standard (AES)
- An SPN can contain a single S-box, but that S-box must be a permutation
- Decryption in an SPN is not the same as encryption, and a separate implementation is needed
- AES remains a secure and very widely used cipher today

Thank you for your attention!