

IV. Practical constructions of PRPs

- **consider constructions widely used in practice**
- **very efficient, but security only heuristic**
- **consider theoretical constructions satisfying definition of pseudorandom permutations in next chapter**
- **see most important design principles and techniques for practical constructions**
- **Feistel ciphers and DES**
- **SPNs and AES (high level)**

Block ciphers

Definition 4.1 A block cipher is an efficient, keyed permutation

$$F : \{0,1\}^l \times \{0,1\}^n \rightarrow \{0,1\}^n .$$

n is called the block length, l is called the key length.

permutation $\forall k \in \{0,1\}^l$

$$\begin{aligned} F_k : \{0,1\}^n &\rightarrow \{0,1\}^n \\ \mathbf{x} &\mapsto F(\mathbf{k}, \mathbf{x}) = F_k(\mathbf{x}) \end{aligned}$$

is a bijection

efficient there is a practical algorithm A that on input

$k \in \{0,1\}^l$ and $\mathbf{x} \in \{0,1\}^n$ outputs $F_k(\mathbf{x})$.

Use and security of block ciphers

- **block ciphers used as building blocks for encryption schemes not as encryption schemes**
- **security only heuristic, but based on principles**
- **heuristic security: best known techniques cannot break block cipher**
- **principles: „algorithms will be judged on the following factors:**
 - **the extent to which the algorithm output is indistinguishable from a random permutation on the input block.“**

(NIST on AES selection criteria)

Attacks on block ciphers

ciphertext-only attack

attacker only given series of outputs $\{F_k(x_i)\}$ for unknown $\{x_i\}$

known-plaintext attack

attacker given pairs of input/output $\{(F_k(x_i), x_i)\}$

chosen-plaintext attack

attacker given pairs of input/output $\{(F_k(x_i), x_i)\}$ for $\{x_i\}$ of his choice

chosen-ciphertext attack

attacker given pairs $\{(F_k(x_i), x_i)\}$
 $\{(F_k^{-1}(y_i), y_i)\}$ for $\{x_i\}, \{y_i\}$ of his choice

Goals of adversaries

- **key recovery**
- **distinguish from (pseudo-) random permutation**

Confusion and diffusion

Confusion The relation between key/input and output should be complex, i.e. even small changes in key/input should lead to unpredictable changes in output.

Diffusion Small changes in key/input should lead to many changes in output.

Confusion and diffusion help to protect against

- linear cryptanalysis
- differential cryptanalysis.

Design of block ciphers

Design principles

- **simple operations (efficiency)**
- **at least one non-linear operation (security)**
- **several rounds (security and efficiency) that result in confusion and diffusion**
- **2 important structures**
 - **Feistel-ciphers (DES)**
 - **substitution-permutation networks (AES).**

Feistel ciphers

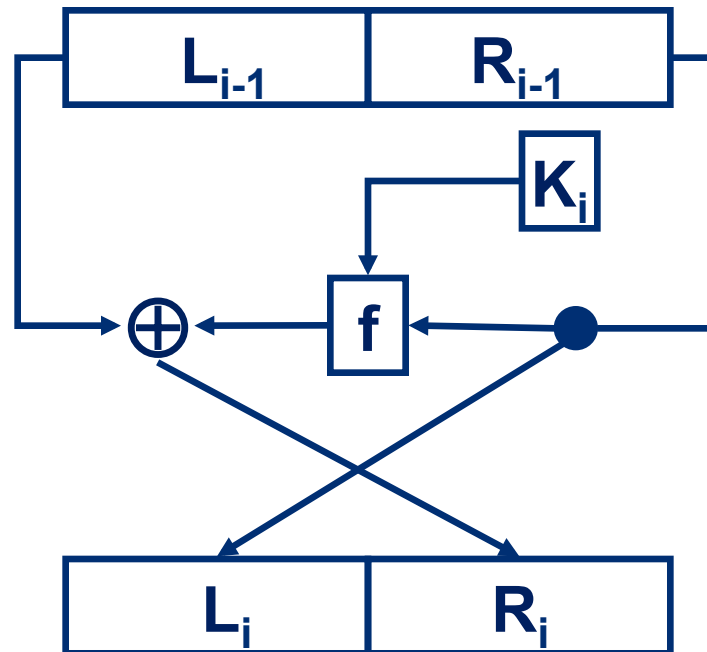
Feistel ciphers defined by

1. input space $P = \{0,1\}^n$, output space $C = \{0,1\}^n$,
key space $K \subseteq \{0,1\}^m$, where n is even, $n=2t$.
2. number of rounds $r > 1$.
3. KeySchedule: $\{0,1\}^m \rightarrow \{0,1\}^{r \cdot t}$
 $K \mapsto K_1, \dots, K_r$
4. round function $f : \{0,1\}^l \times \{0,1\}^t \rightarrow \{0,1\}^t$

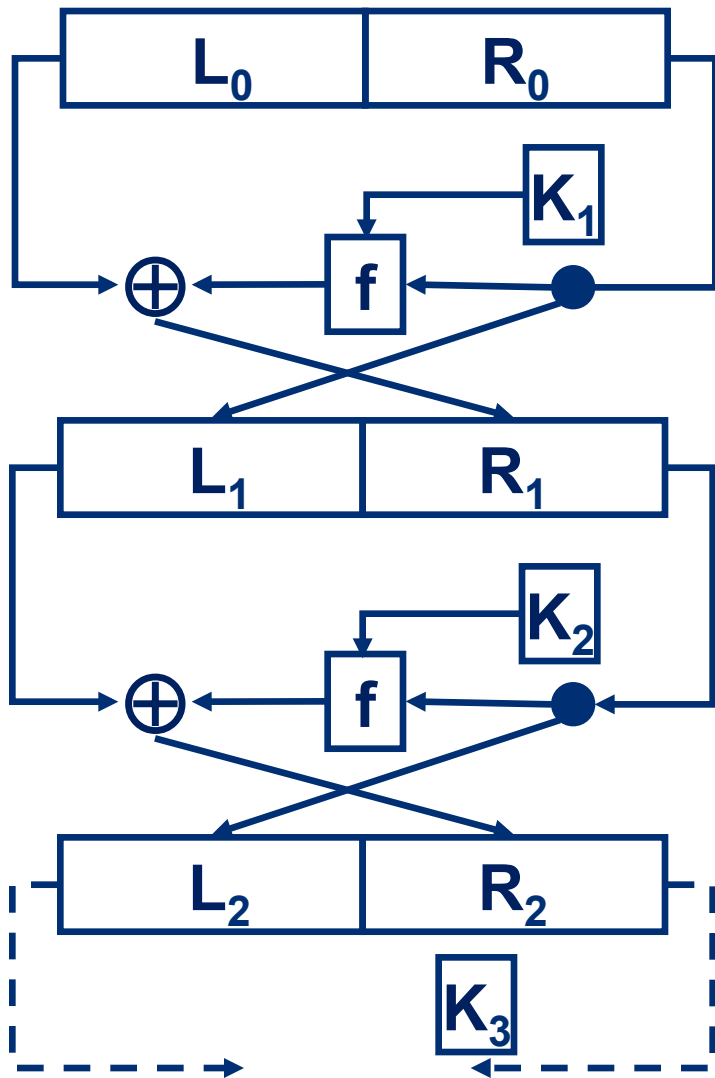
Single round

$(L_{i-1}, R_{i-1}) \mapsto (L_i, R_i)$, where

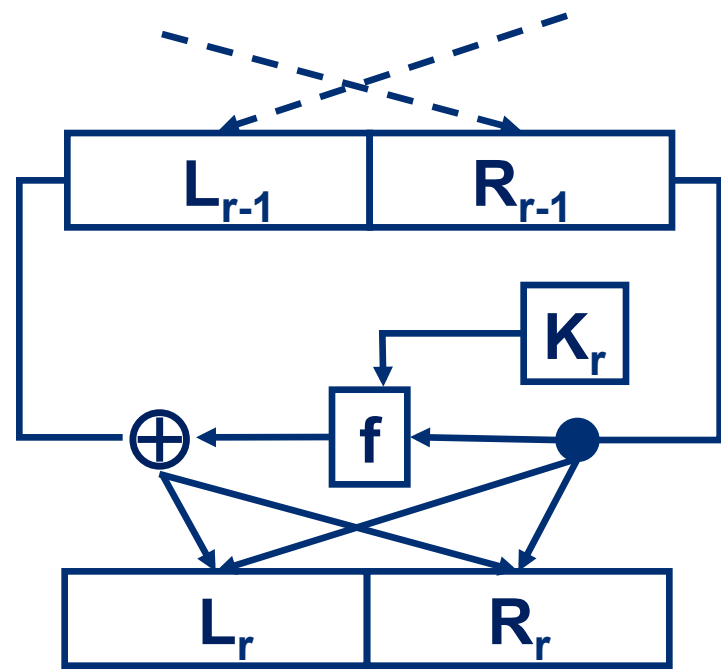
- $L_{i-1}, L_i, R_{i-1}, R_i \in \{0, 1\}^t$
- $L_i = R_{i-1}, R_i = L_{i-1} \oplus f(K_i, R_{i-1})$



Overall structure



input $p=(L_0,R_0)$
output $c=(R_r,L_r)$



DES

- **DES designed 1973/74 by NSA and IBM**
- **1975 introduced as standard of NIST**
- **1999 DeepCrack breaks DES in 22 hours**
- **DES replaced by 3DES as standard**
- **2002 DES replaced by AES (Advanced Encryption Standard) as standard of NIST**
- **3DES still in use today**

DES setup

DES is a Feistel cipher

1. input space $P = \{0,1\}^{64}$, output space $C = \{0,1\}^{64}$,
key space $K \subseteq \{0,1\}^{64}$.

2. number of rounds $r = 16$.

3. KeySchedule: $\{0,1\}^{64} \rightarrow \{0,1\}^{16 \cdot 48}$
 $K \mapsto K_1, \dots, K_{16}$

4. round function $f : \{0,1\}^{48} \times \{0,1\}^{32} \rightarrow \{0,1\}^{32}$

DES key space

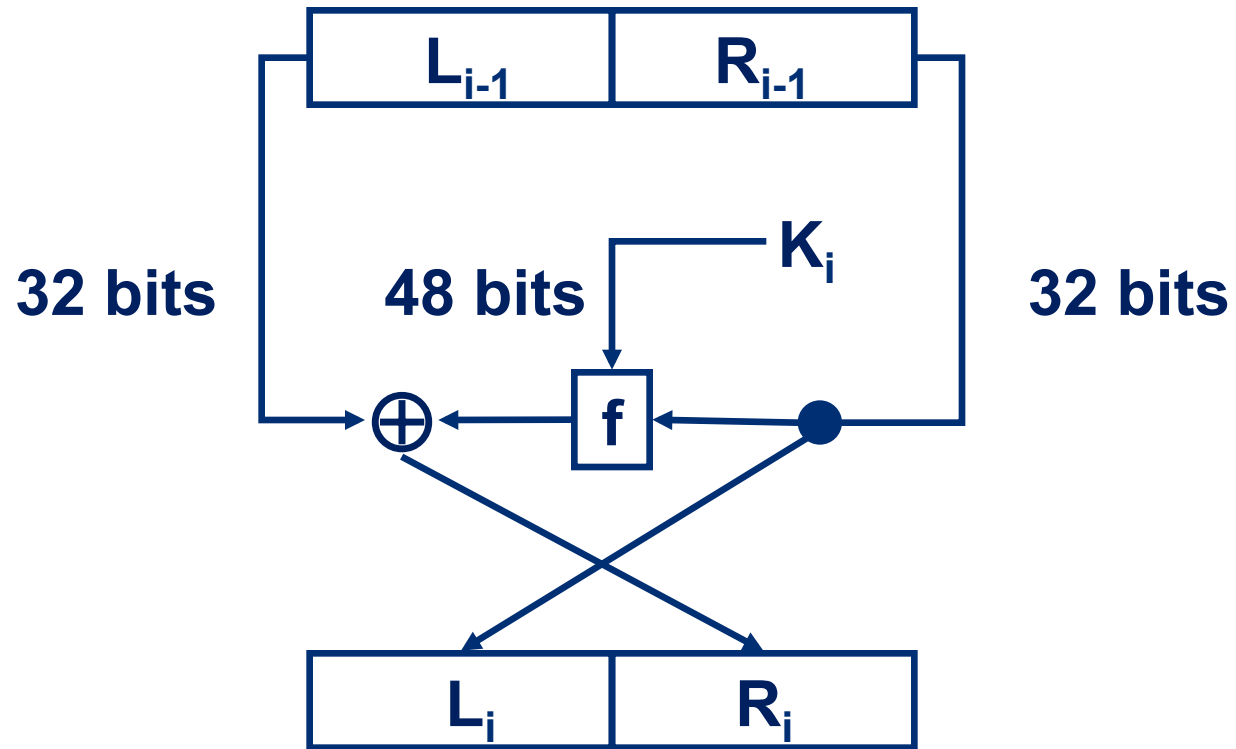
$$K = \left\{ (b_1, \dots, b_{64}) \in \{0, 1\}^{64} : \sum_{i=1}^8 b_{8u+i} = 1 \pmod{2}, 0 \leq u \leq 7 \right\}$$

Hence $|K| = 2^{56}$.

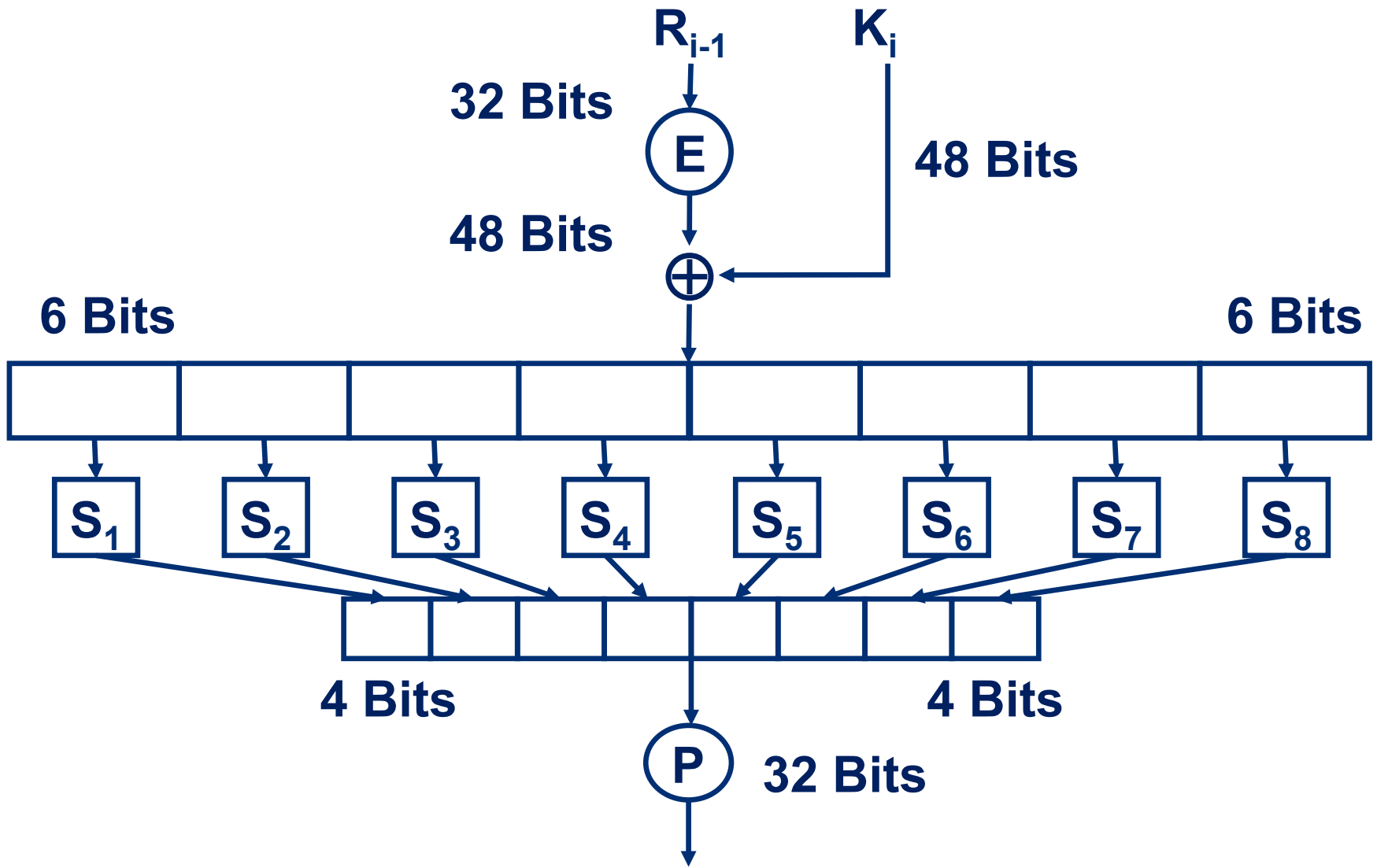
valid DES key

0	0	0	1	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	1	0	1	1	1
0	1	1	1	1	0	0	1
1	0	0	1	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	1	1	1	1	1
1	1	1	1	0	0	0	1

Single round



Round function



$$f(K_i, R_{i-1}) = P(S(E(R_{i-1}) \oplus K_i))$$

Expansion and permutation

expansion

$$\begin{aligned} \mathbf{E}: \{0,1\}^{32} &\rightarrow \{0,1\}^{48} \\ \mathbf{R}_1 \cdots \mathbf{R}_{32} &\mapsto \mathbf{R}_{32} \mathbf{R}_1 \cdots \mathbf{R}_{32} \mathbf{R}_1 \end{aligned}$$

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

permutation

$$\begin{aligned} \mathbf{P}: \{0,1\}^{32} &\rightarrow \{0,1\}^{32} \\ \mathbf{R}_1 \cdots \mathbf{R}_{32} &\mapsto \mathbf{R}_{16} \cdots \mathbf{R}_{25} \end{aligned}$$

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

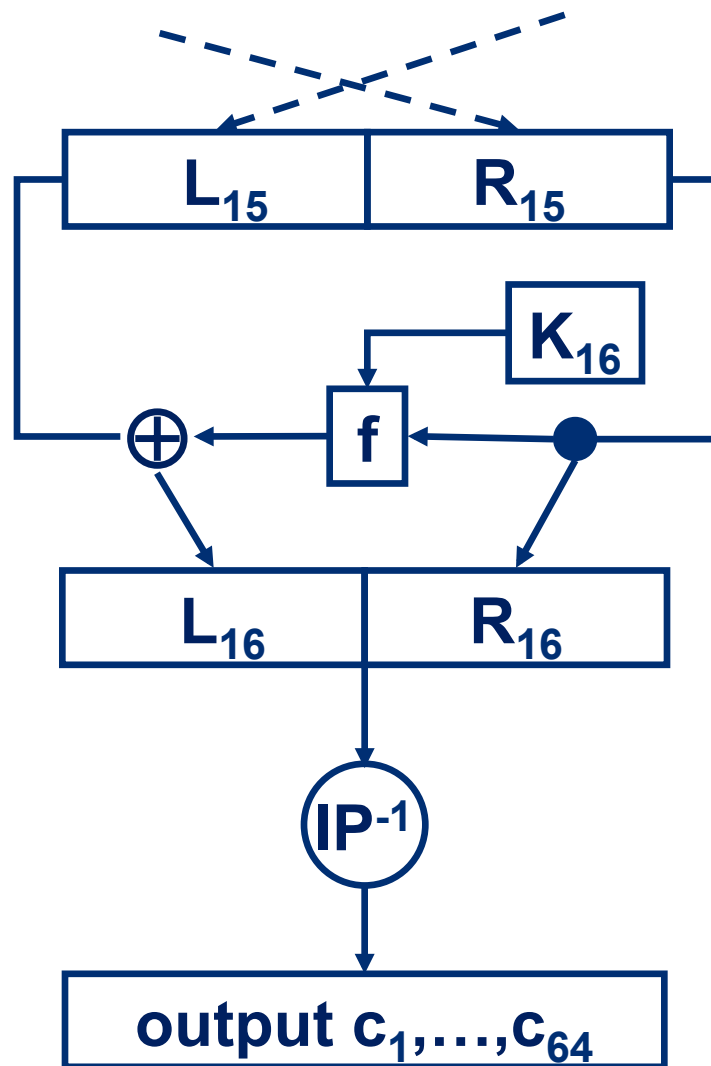
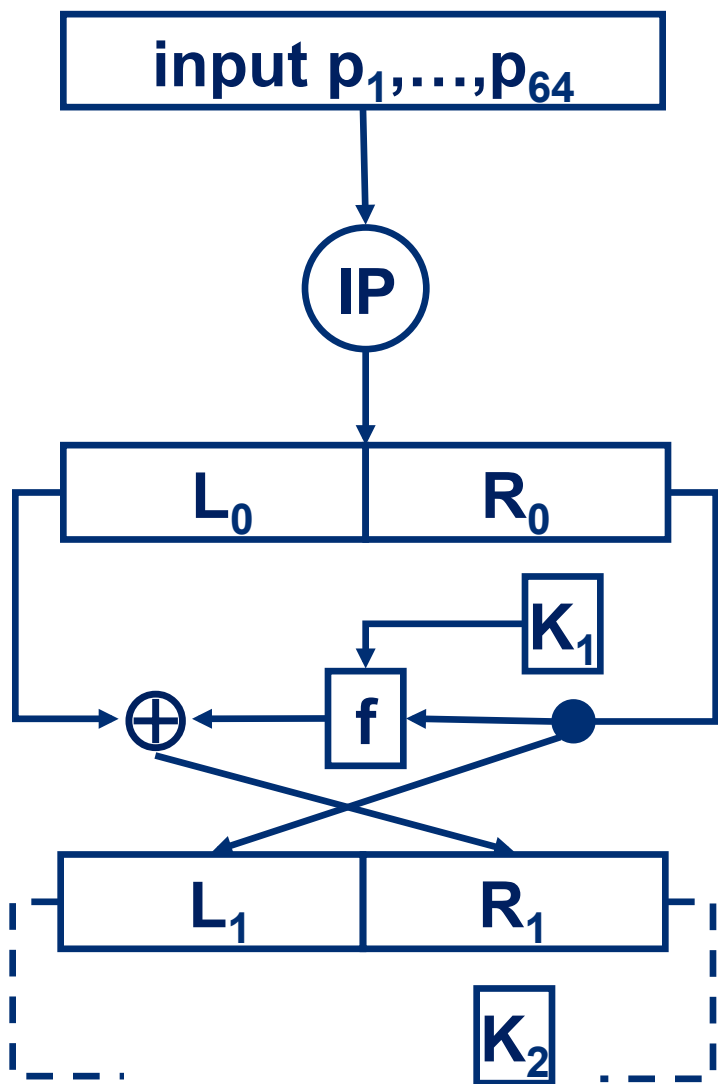
S-boxes

$$S_i: \{0,1\}^6 \rightarrow \{0,1\}^4, i = 1, \dots, 8$$
$$b_1 \dots b_6 \mapsto S_i(b_1 b_6, b_2 b_3 b_4 b_5)$$

1. interpret $b_1 b_6$ as row index, $b_2 b_3 b_4 b_5$ as column index.
2. output is binary representation of entry in position $(b_1 b_6, b_2 b_3 b_4 b_5)$ of i -th S-Box.

S_1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

DES



Initial permutation

$$\begin{aligned} \mathbf{IP} : \{0,1\}^{64} &\rightarrow \{0,1\}^{64} \\ \mathbf{p}_1 \cdots \mathbf{p}_{64} &\mapsto \mathbf{p}_{58} \mathbf{p}_{50} \cdots \mathbf{p}_7 \end{aligned}$$

$$\begin{aligned} \mathbf{IP}^{-1} : \{0,1\}^{64} &\rightarrow \{0,1\}^{64} \\ \mathbf{v}_1 \cdots \mathbf{v}_{64} &\mapsto \mathbf{v}_{40} \mathbf{v}_8 \cdots \mathbf{v}_{25} \end{aligned}$$

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

S-boxes

Lemma 4.2 For $i = 1, \dots, 8$ and for all $u, v \in \{0, 1\}^6$, that differ in exactly one position, $S_i(u)$ and $S_i(v)$ differ in at least two positions.

Example $i = 1, u = 101101, v = 111101$

$$S_1(u) = 0001, S_1(v) = 0110$$

S_1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Permutation

Lemma 4.3 View $u, v \in \{0, 1\}^{32}$ as sequence of eight 4-bit sequences. If u, v differ in two positions in a single 4-bit string, then $P(u)$ and $P(v)$ differ in two 4-bit sequences.

Example $u = 0^{32}, v = 110^{30}$

$$P(u) = 0^{32}$$

$$P(v) = 0^8 10^7 10^{15}$$

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Diffusion in DES

Lemma 4.2 For $i = 1, \dots, 8$ and for all $u, v \in \{0, 1\}^6$, that differ in exactly one position, $S_i(u)$ and $S_i(v)$ differ in at least two positions.

Lemma 4.3 View $u, v \in \{0, 1\}^{32}$ as sequence of eight 4-bit sequences. If u, v differ in two positions in a single 4-bit string, then $P(u)$ and $P(v)$ differ in two 4-bit sequences.

Diffusion and avalanche effect

inputs p and q differ in $\Delta=1$ positions,
difference in left half

$\Delta=1$	$\Delta=0$
------------	------------

input

$\Delta=0$	$\Delta=1$
------------	------------

after round 1

$\Delta=1$	$\Delta=2$
------------	------------

after round 2

$\Delta=2$	$\Delta=4$
------------	------------

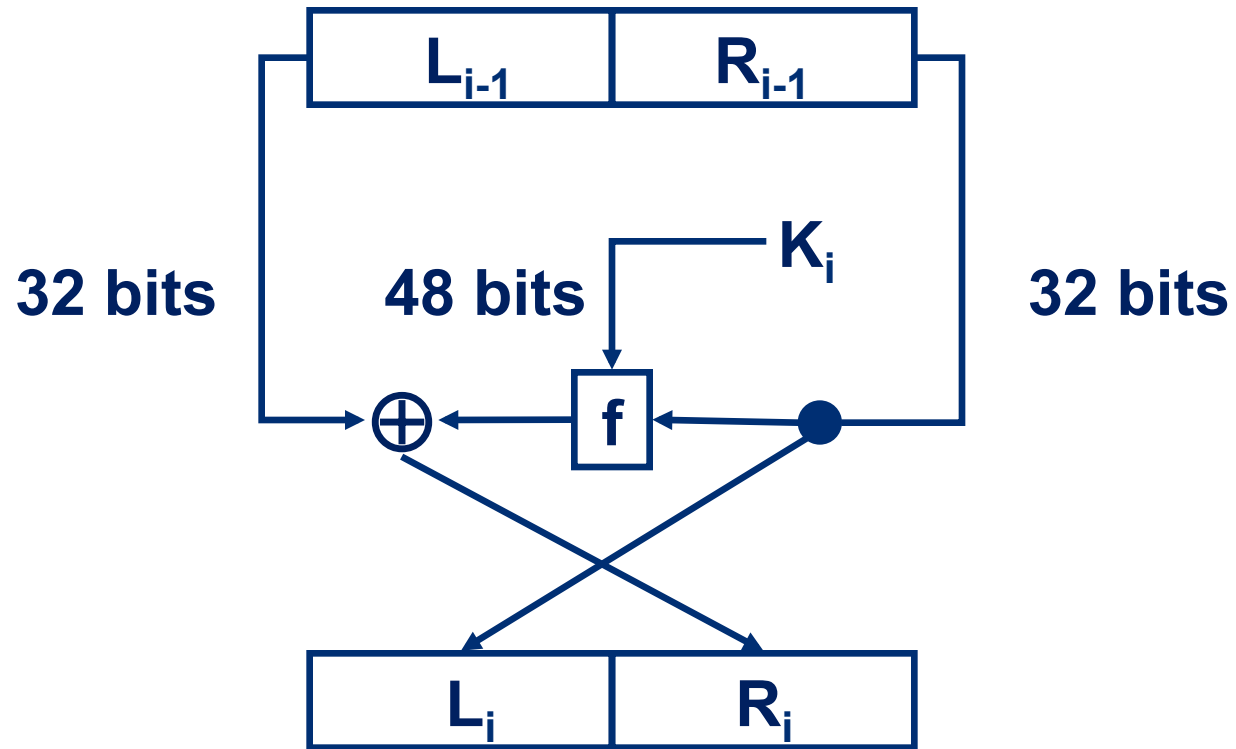
after round 3

$\Delta=4$	$\Delta=8$
------------	------------

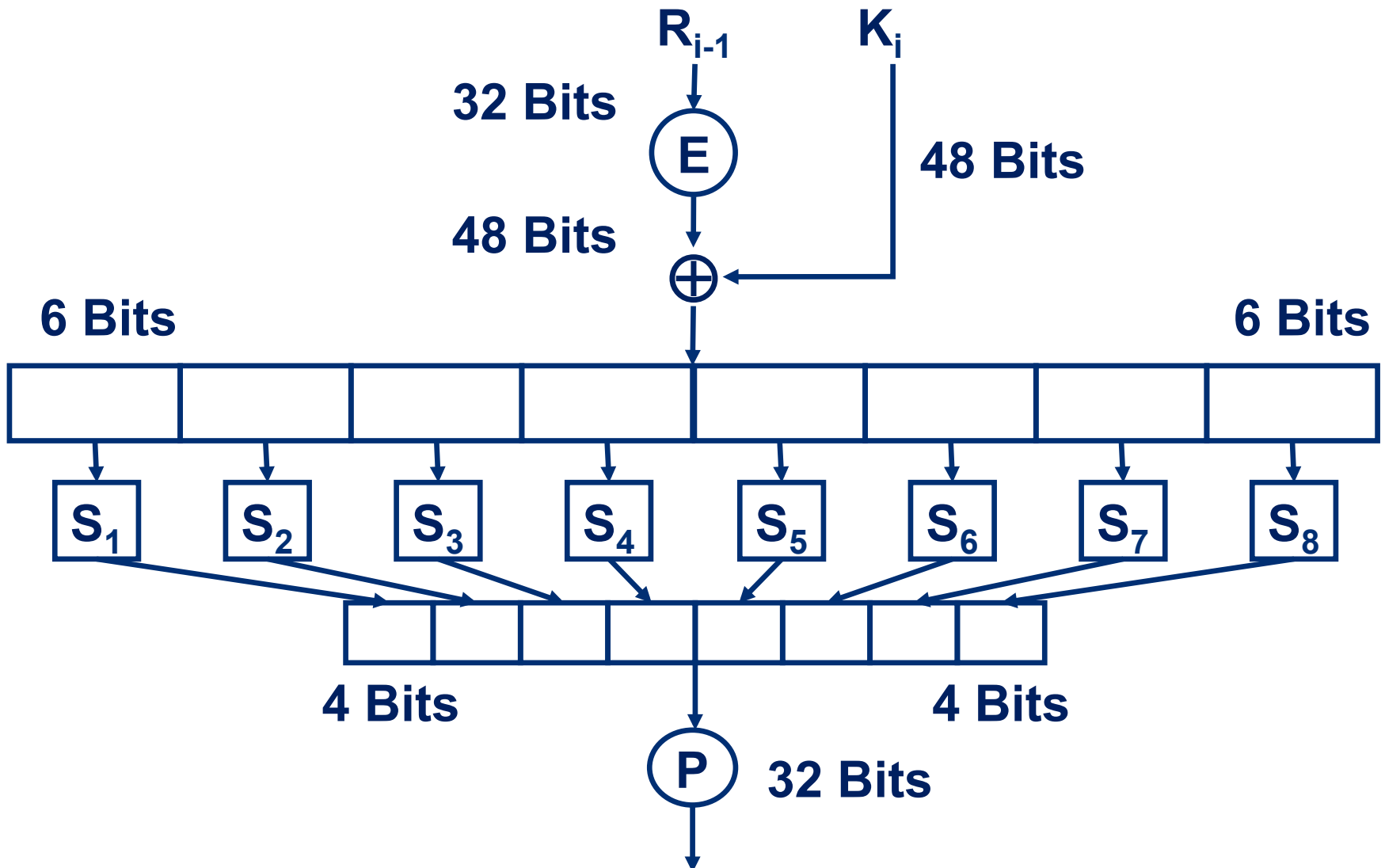
after round 4

avalanche effect

Single round



Round function



$$f(K_i, R_{i-1}) = P(S(E(R_{i-1}) \oplus K_i))$$

DES variants

$$\mathbf{2DES} \quad \mathbf{P} = \mathbf{C} = \{0,1\}^{64}, \mathbf{K} \subseteq \{0,1\}^{64} \times \{0,1\}^{64}$$

$$\mathbf{k} \in \mathbf{K}, \mathbf{k} = (\mathbf{k}_1, \mathbf{k}_2)$$

$$\mathbf{2DES}_{\mathbf{k}}(\mathbf{p}) = \mathbf{DES}_{\mathbf{k}_2}(\mathbf{DES}_{\mathbf{k}_1}(\mathbf{p}))$$

$$\mathbf{3DES} \quad \mathbf{P} = \mathbf{C} = \{0,1\}^{64}, \mathbf{K} \subseteq \{0,1\}^{64} \times \{0,1\}^{64} \times \{0,1\}^{64}$$

$$\mathbf{k} \in \mathbf{K}, \mathbf{k} = (\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3)$$

$$\mathbf{3DES}_{\mathbf{k}}(\mathbf{p}) = \mathbf{DES}_{\mathbf{k}_3}(\mathbf{DES}_{\mathbf{k}_2}^{-1}(\mathbf{DES}_{\mathbf{k}_1}(\mathbf{p})))$$

- **2DES** not as secure as key length indicates.
- **3DES** still considered secure, but too slow.

Substitution-permutation networks

SPNs defined by

1. input space $P = \{0,1\}^n$, output space $C = \{0,1\}^n$,
key space $K = \{0,1\}^n$, where $n = t \cdot b$.

2. round number $r > 1$.

3. KeySchedule: $\{0,1\}^n \rightarrow \{0,1\}^{(r+1) \cdot n}$
 $K \mapsto K_0, \dots, K_r$

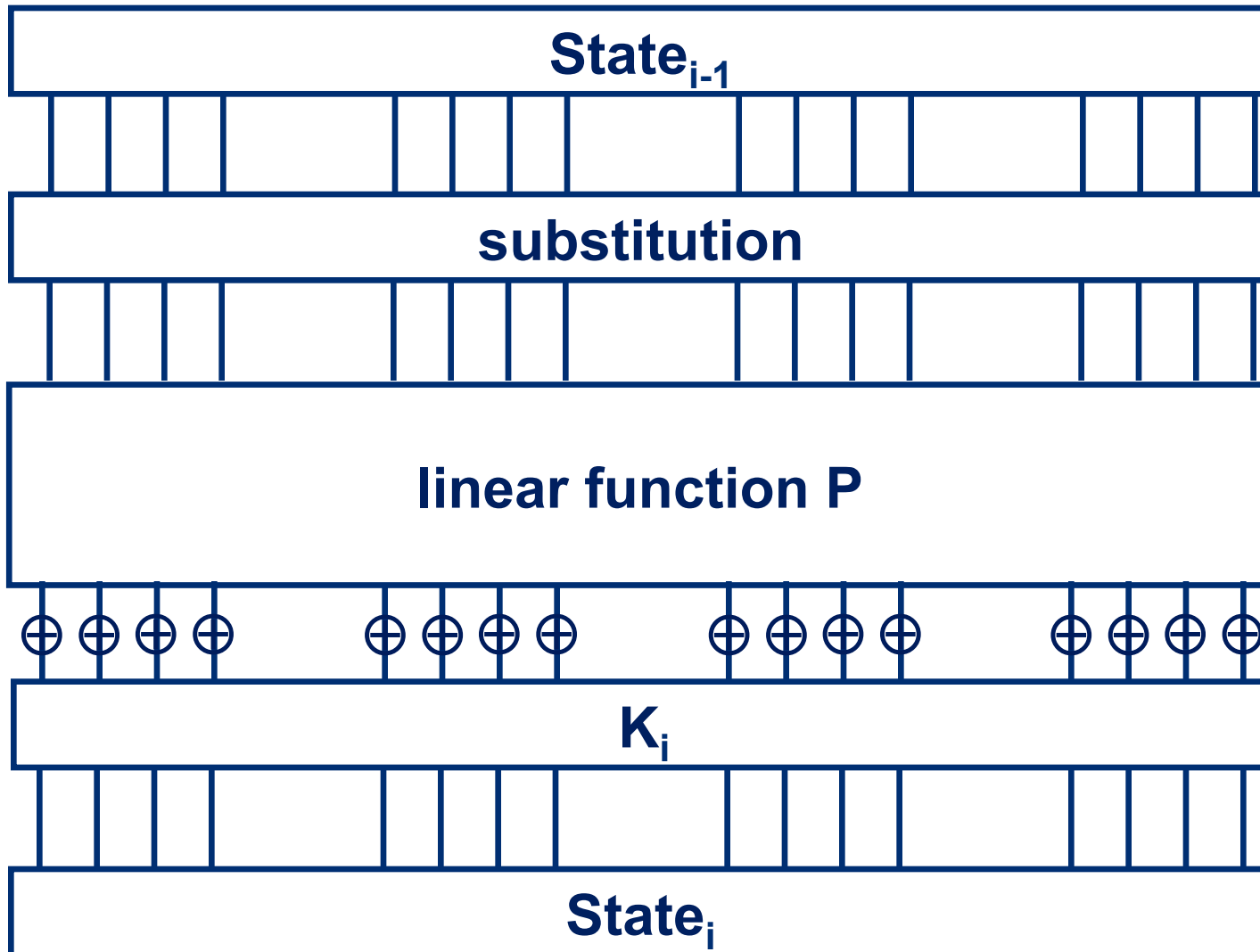
4. bijective substitutions $S_i : \{0,1\}^b \rightarrow \{0,1\}^b, i = 1, \dots, t$

5. bijective linear function $P : \{0,1\}^n \rightarrow \{0,1\}^n$.

(P sometimes permutation on n bits, hence the name) ²⁷

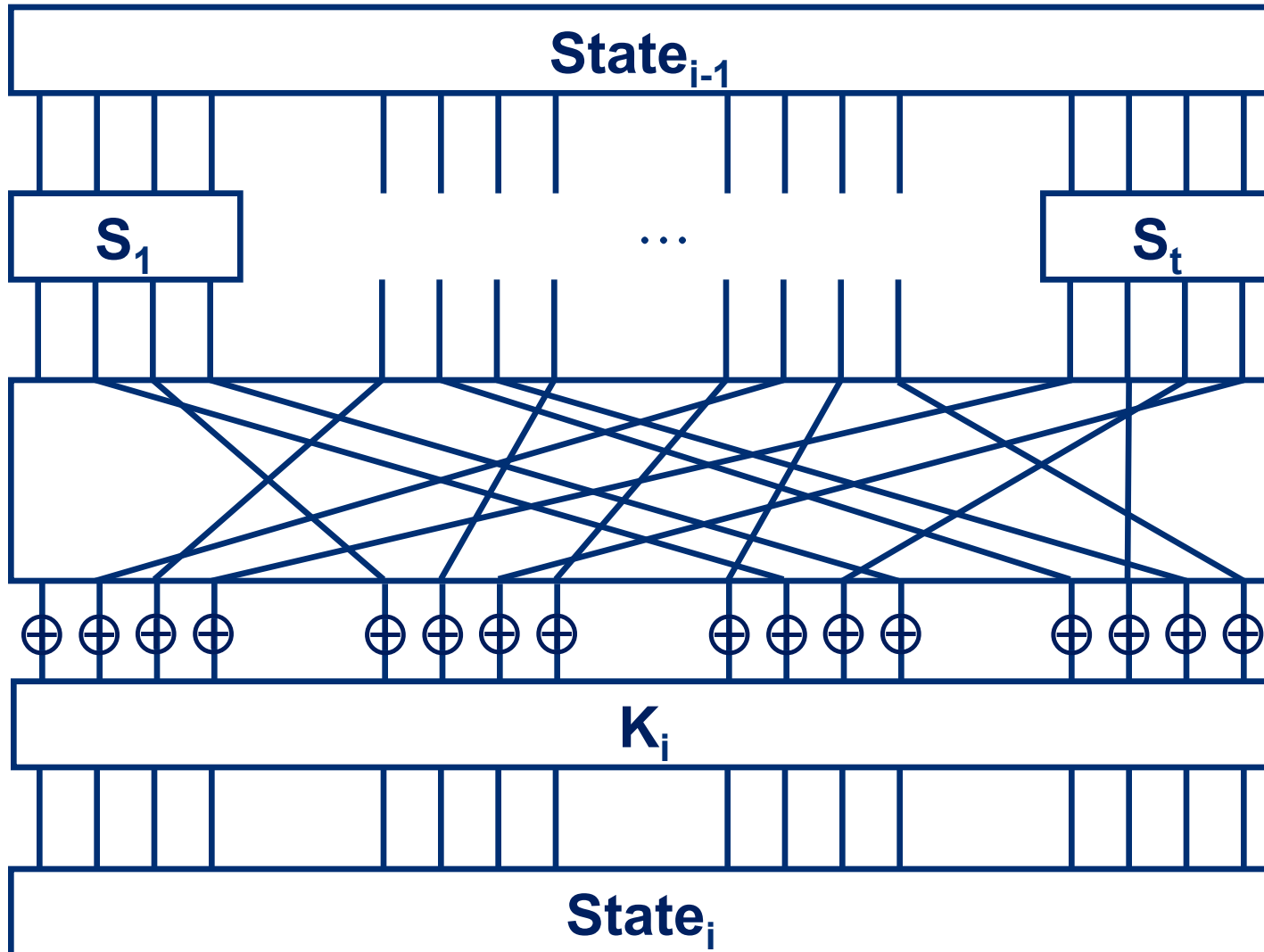
SPN – single round

$$\text{State}_i := P(S(\text{State}_{i-1})) \oplus K_i, \quad \text{State}_j \in \{0,1\}^n$$



SPN – single round

$$\text{State}_i := P(S(\text{State}_{i-1})) \oplus K_i, \quad \text{State}_j \in \{0,1\}^n$$



SPN initialization

- input $p \in \{0,1\}^n$
- $\text{State}_0 = p \oplus K_0$
- called **whitening**
- introduced so that all states depend on the secret key

Advanced Encryption Standard (AES)

- 1997 NIST call for proposals for successor of DES.**
- 1998 15 proposals submitted.**
- 1999 selection of 5 candidates (MARS, RC6, Rijndael, Serpent, Twofish).**
- 2. Oktober 2000 announcement of Rijndael as winner.**
- 26. November 2001 AES-Rijndael replaces DES as NIST standard.**
- AES-Rijndael designed by Joan Daemen and Vincent Rijmen.**

AES parameters

AES is SPN cipher with

1. input space $P = \{0,1\}^{128}$, output space $C = \{0,1\}^{128}$,
key space $K = \{0,1\}^{128}$, where $b=8,t=16$.
2. round number $r = 10$.
3. KeySchedule: $\{0,1\}^{128} \rightarrow \{0,1\}^{11 \cdot 128}$
 $K \mapsto K_0, \dots, K_{10}$
4. bijective substitution $S : \{0,1\}^8 \rightarrow \{0,1\}^8$
5. linear bijection $P : \{0,1\}^{128} \rightarrow \{0,1\}^{128}$.

States in AES

$$\text{State}_j = \begin{array}{|c|c|c|c|} \hline \mathbf{B}_{00} & \mathbf{B}_{01} & \mathbf{B}_{02} & \mathbf{B}_{03} \\ \hline \mathbf{B}_{10} & \mathbf{B}_{11} & \mathbf{B}_{12} & \mathbf{B}_{13} \\ \hline \mathbf{B}_{20} & \mathbf{B}_{21} & \mathbf{B}_{22} & \mathbf{B}_{23} \\ \hline \mathbf{B}_{30} & \mathbf{B}_{31} & \mathbf{B}_{32} & \mathbf{B}_{33} \\ \hline \end{array}$$

$$\mathbf{B}_{ij} \in \{0,1\}^8, i, j \in \{0,1,2,3\}$$

AES round structure

State \leftarrow **p**

State \leftarrow **AddRoundKey(State, K₀)**

For **i=1** **to** **9** **do**

State \leftarrow **SubBytes(State)**

State \leftarrow **ShiftRows(State)**

State \leftarrow **MixColumns(State)**

} **function P**

State \leftarrow **AddRoundKey(State, K_i)**

State \leftarrow **SubBytes(State)**

State \leftarrow **ShiftRows(State)**

State \leftarrow **AddRoundKey(State, K₁₀)**

return **State**

Byte interpretation

$$\{0,1\}^8 = \{0,1\}^4 \times \{0,1\}^4$$

$$\{0,1\}^4 \doteq \{0,1,2,\dots,15\}$$

write elements in $\{0,1,2,\dots,15\}$ in hexadecimal representation.

$$\{0,1,2,\dots,14,15\} \doteq \{0,1,2,\dots,9,A,B,C,D,E,F\}$$

Fact There is a field \mathbb{F}_{256} with 256 elements.

$256 = 2^8 \Rightarrow$ identify elements in \mathbb{F}_{256} with pairs of hexadecimal numbers

SubBytes

B_{00}	B_{01}	B_{02}	B_{03}
B_{10}	B_{11}	B_{12}	B_{13}
B_{20}	B_{21}	B_{22}	B_{23}
B_{30}	B_{31}	B_{32}	B_{33}

SubBytes

$S(B_{00})$	$S(B_{01})$	$S(B_{02})$	$S(B_{03})$
$S(B_{10})$	$S(B_{11})$	$S(B_{12})$	$S(B_{13})$
$S(B_{20})$	$S(B_{21})$	$S(B_{22})$	$S(B_{23})$
$S(B_{30})$	$S(B_{31})$	$S(B_{32})$	$S(B_{33})$

$S : \{0,1\}^8 \rightarrow \{0,1\}^8$ non-linear

SubBytes

	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	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

ShiftRows

B_{00}	B_{01}	B_{02}	B_{03}
B_{10}	B_{11}	B_{12}	B_{13}
B_{20}	B_{21}	B_{22}	B_{23}
B_{30}	B_{31}	B_{32}	B_{33}

ShiftRows

B_{00}	B_{01}	B_{02}	B_{03}
B_{11}	B_{12}	B_{13}	B_{10}
B_{22}	B_{23}	B_{20}	B_{21}
B_{33}	B_{30}	B_{31}	B_{32}

MixColumns

B_{00}	B_{01}	B_{02}	B_{03}
B_{10}	B_{11}	B_{12}	B_{13}
B_{20}	B_{21}	B_{22}	B_{23}
B_{30}	B_{31}	B_{32}	B_{33}

MixColumns

02	03	01	01
01	02	03	01
01	01	02	03
03	01	01	02

.

B_{00}	B_{01}	B_{02}	B_{03}
B_{10}	B_{11}	B_{12}	B_{13}
B_{20}	B_{21}	B_{22}	B_{23}
B_{30}	B_{31}	B_{32}	B_{33}

Arithmetic operations performed in \mathbb{F}_{256} !

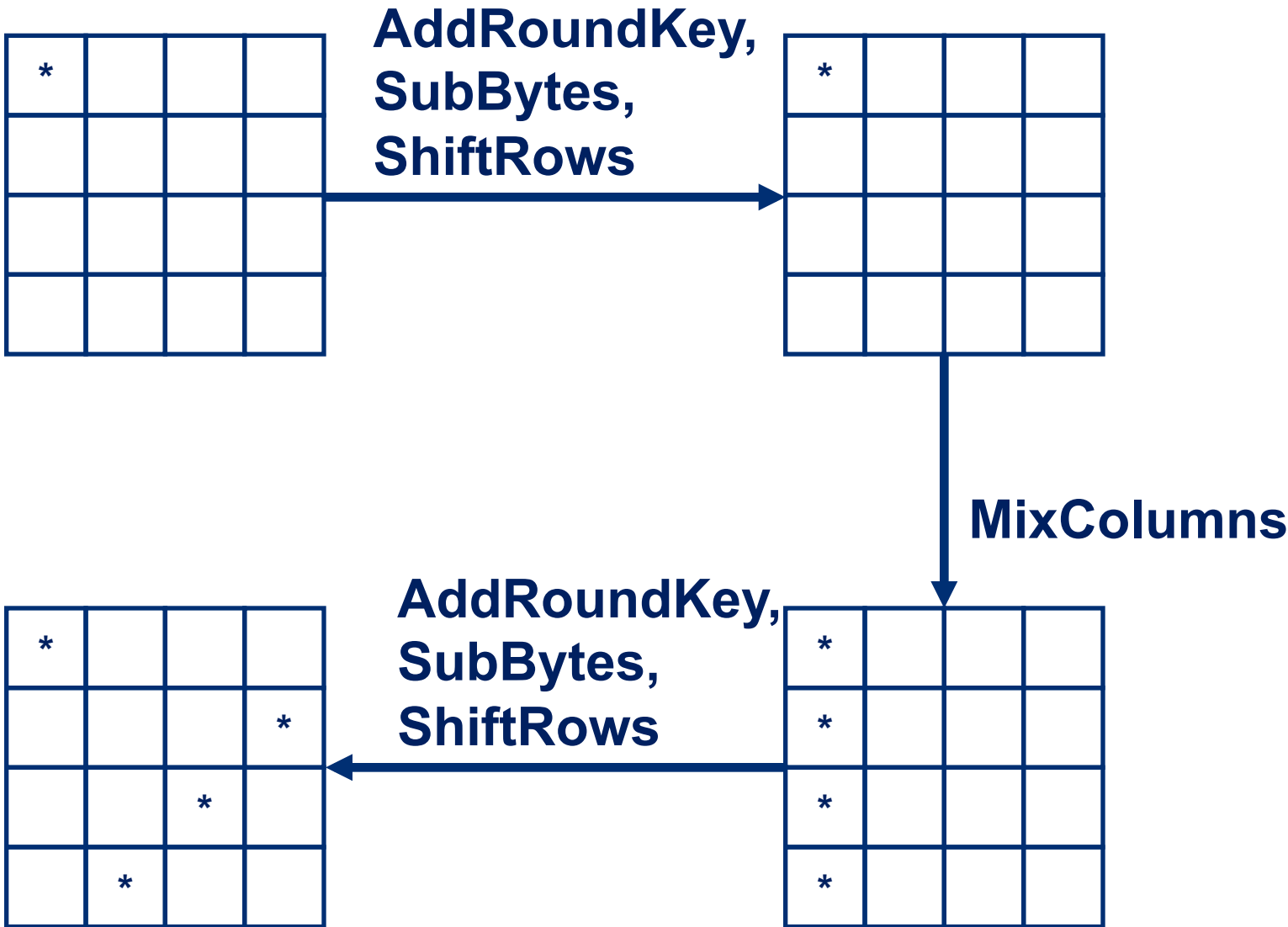
Diffusion in AES

Diffusion Small changes in key/input should lead to many changes in output.

Lemma 4.4 If a vector consisting of four bytes has t non-zero bytes, then its image under MixColumns has at least $5 - t$ non-zero bytes, $1 \leq t \leq 4$.

MDS property (maximum distance separable)

Diffusion in AES



* := positions with different bytes in two states

Diffusion in AES



***:= positions with different bytes in two states**

Inversion in AES

State \leftarrow p

State \leftarrow AddRoundKey(State, K_{10})

For i=9 **downto** 1 **do**

State \leftarrow InvShiftRows(State)

State \leftarrow InvSubBytes(State)

State \leftarrow AddRoundKey(State, K_i)

State \leftarrow InvMixColumns(State)

State \leftarrow InvShiftRows(State)

State \leftarrow InvSubBytes(State)

State \leftarrow AddRoundKey(State, K_0)

return State

Inv* := inverse operation to *