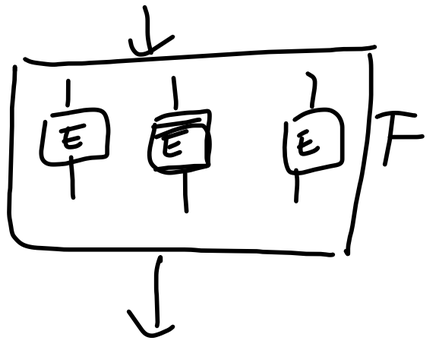


Block-Cipher

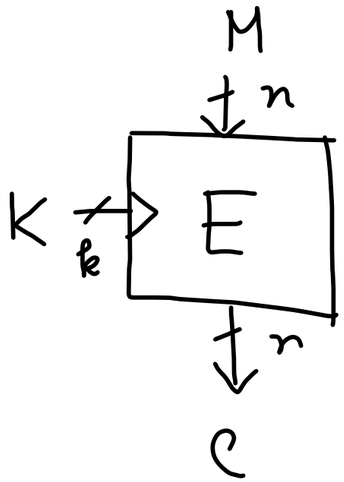
$E: \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ $\quad$ $n \to$ block-size

$\forall k$, $E_k(\cdot)$ Should be a permutation.

F^E is xyz secure if E is abc secure

- xyz: IND-CPA, IND-CCA, AE-secure
- abc: PRP, SPRP, PRF (hard to distinguish)

Designing Block-Ciphers



Security → Hard to distinguish. (PRP)
- (CPA / KPA / CCA)

Shannon (1949)
- Confusion (from (PT, CT) pairs, it is difficult to get any information about key)
- Diffusion (even if you change one-bit in the i/p, it should affect all the ciphertext bits)

{ Properties you should keep in mind

Confusion → Non-linear function

Diffusion → linear function

Non-linear functions

- Boolean functions
- field multiplication
- field inverse
- integer addition

Linear Functions

- XOR
- Rotation, Shift
- Matrix Multiplication

[Ref: Sasaki's Book Chapter 1]

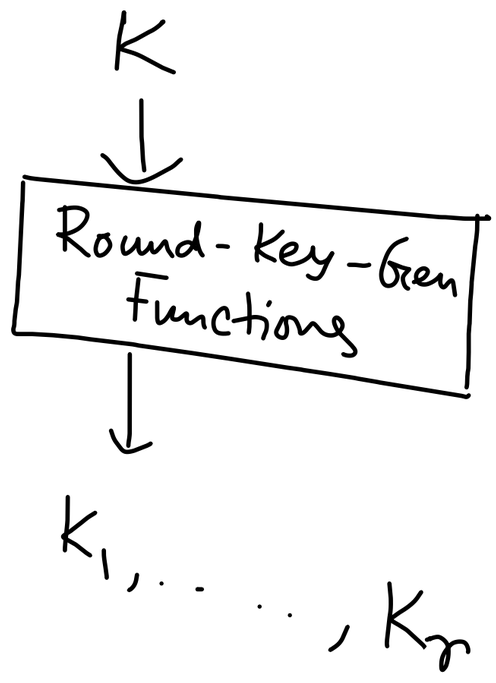
$\{M_1 M_2 M_3 M_4$
$K_1 K_2 K_3 K_4$
$C_1 C_2 C_3 C_4$

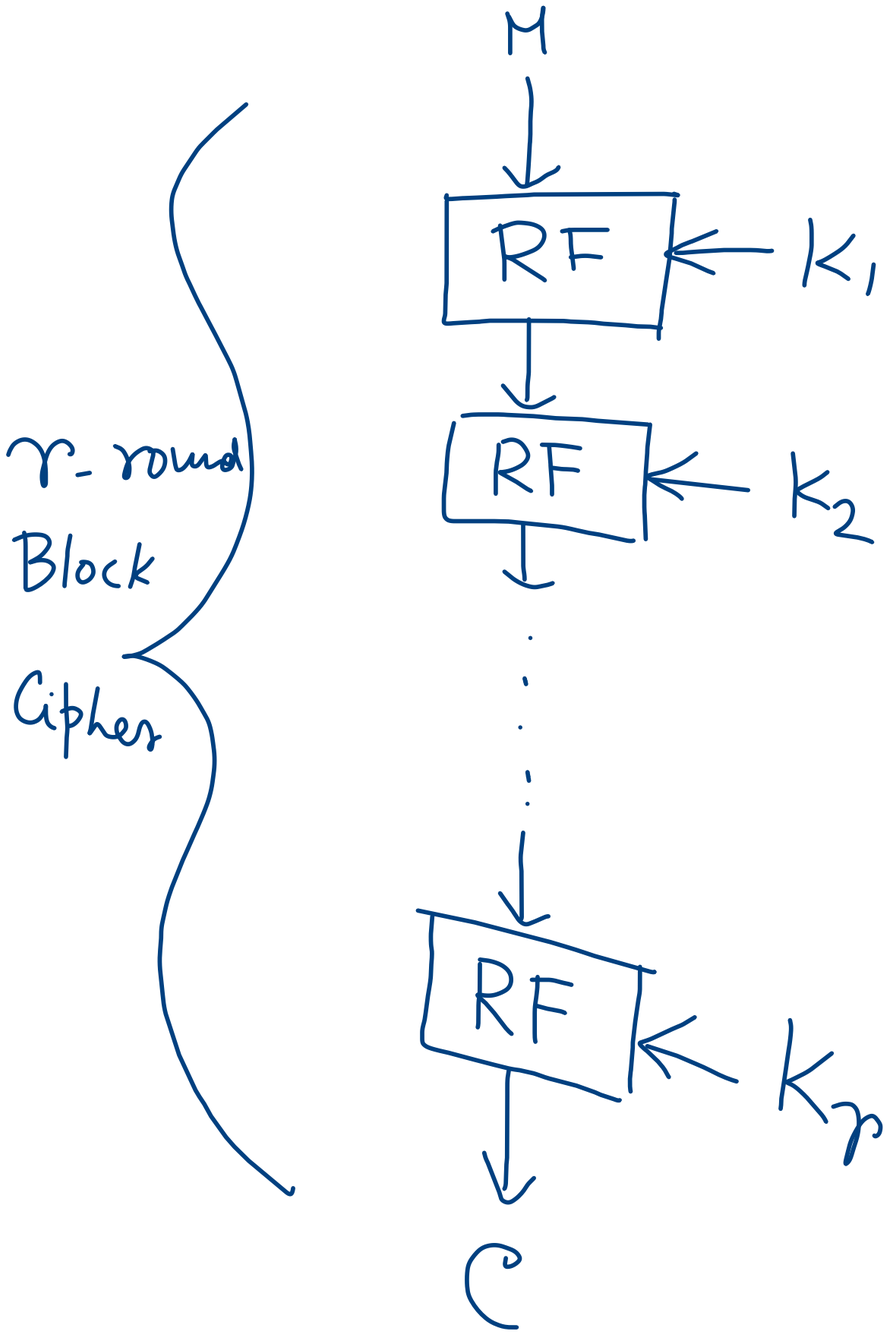
$C_1 = (M_1 + M_3) + (K_2 + K_3)$

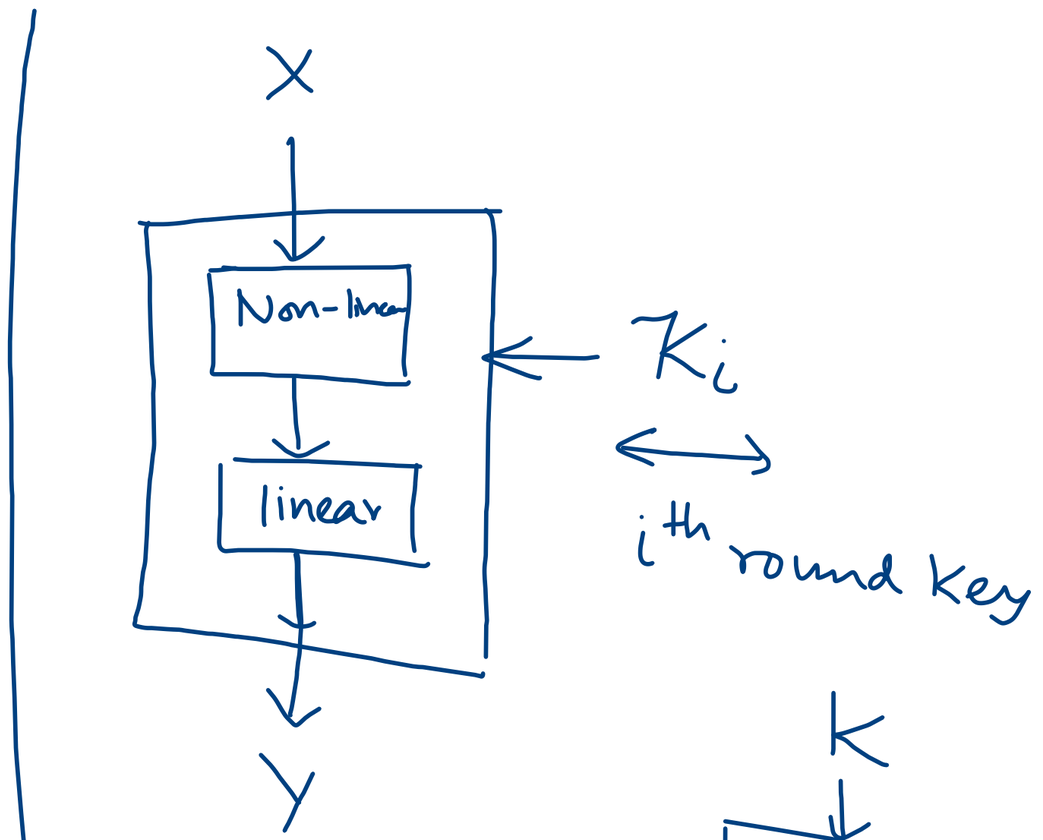
$C_1 = (M_1 + M_3) + K_2 K_3 K_4 + K_1 K_2 K_4 + K_2 K_3$

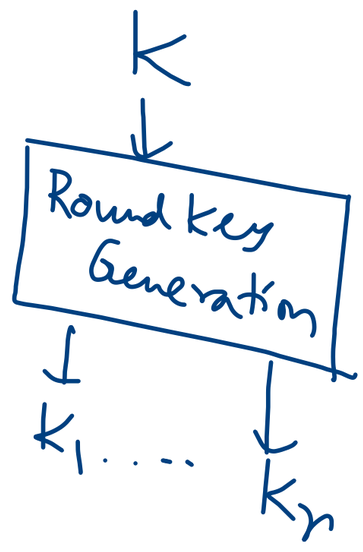
Constructing Block-Cipher

- Iterative Structure
  - Apply "Round Functions" Several times.
- Round Function
  - Non-linear function (Substitution)
  - Linear functions (Permutation)
  - Add round-keys

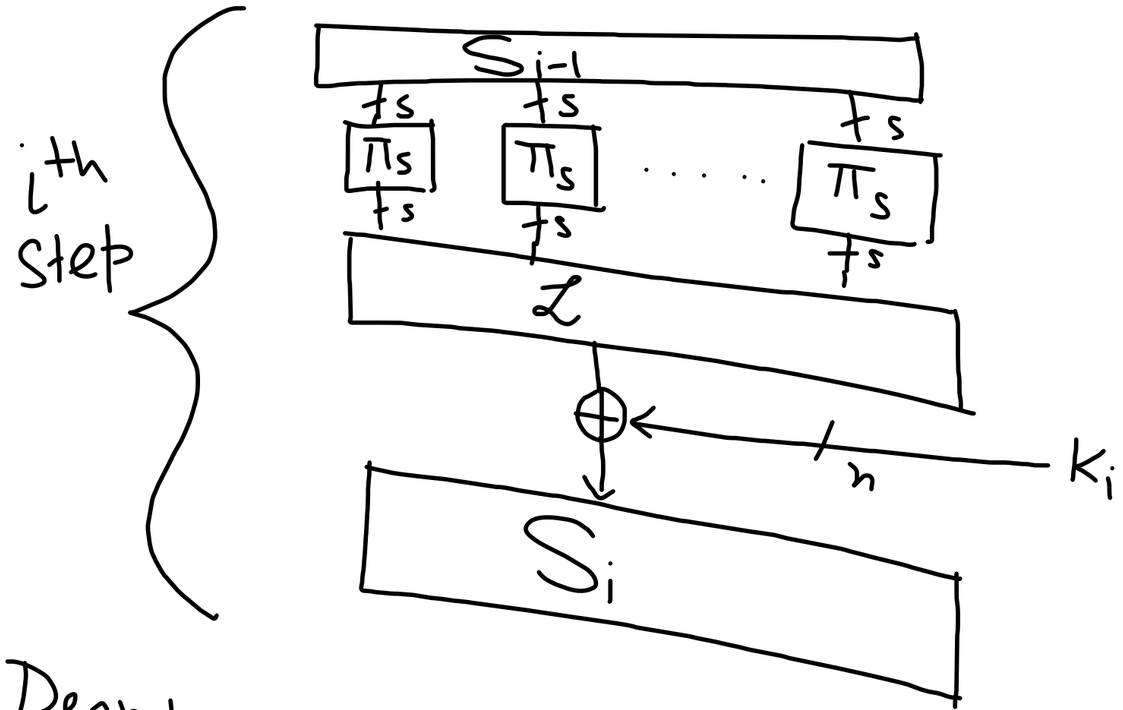



M
RF
K_1
RF
K_2
r- round
Block
Cipher
RF
K_r
C


X
Non-linear
linear
K_i
i^{th} round key
Y


K
Round key
Generation
K_1 ....
K_r

SPN (Substitute/Permutation Network) $\Leftarrow$ (AES)



Decryption required π_S^{-1}

$n = s \cdot t$

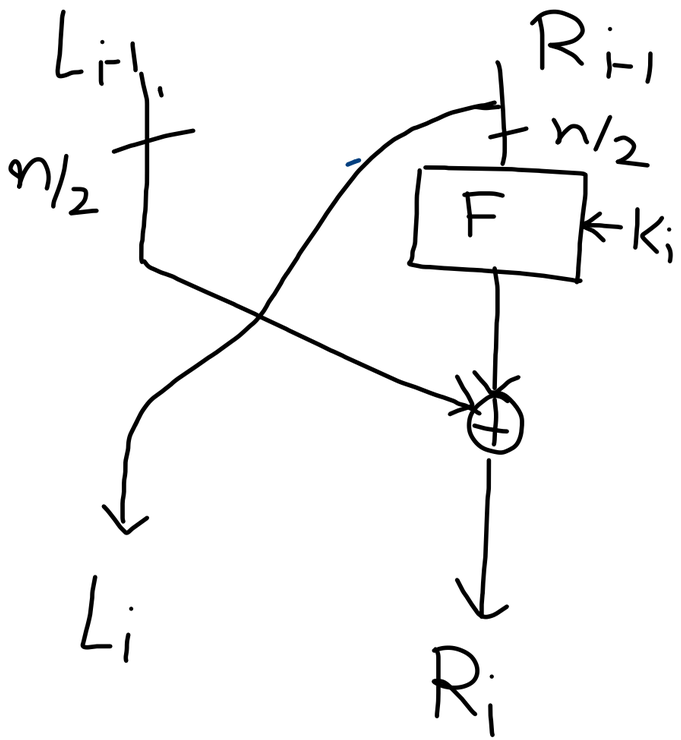
$\pi_s : \{0,1\}^s \rightarrow \{0,1\}^s$

↳ Non-linear

$\mathcal{L} : \{0,1\}^n \rightarrow \{0,1\}^n$

↳ Linear

Feistel



- F^{-1} not needed in decryption
- Non-linear function is applied to partial state.

⇐ DES

⇒ Non-linear function

$F: \{0,1\}^k \times \{0,1\}^{n/2} \rightarrow \{0,1\}^{n/2}$

↳ Size $n/2$

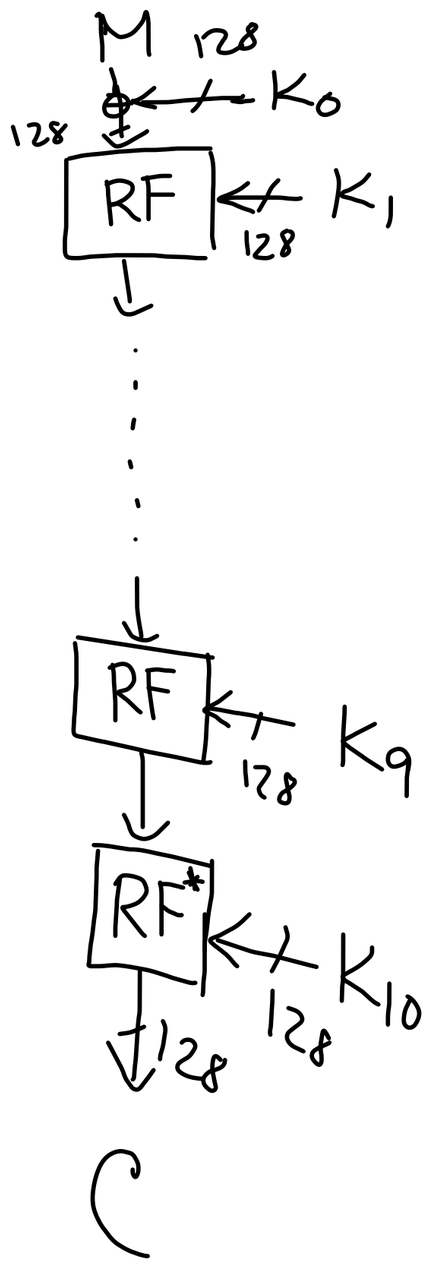
AES (Advanced Encryption Standard)

Rijndael

(Rijmen, daeman)

AES

- Block size: 128
- Key size: 128 / 192 / 256



AES

RF

- Sub-Bytes (using SBox) ↕ Non-linear
- Shift Row
- Mix-Column ↕ Linear
- Add Sub-key ↕ round key addition

RF*

Same as RF except Mix-Column

Plaintext

M_0	M_4	M_8	M_{12}
M_1	M_5	M_9	M_{13}
M_2	M_6	M_{10}	M_{14}
M_3	M_7	M_{11}	M_{15}

$$M = M_0 \| M_1 \| \cdots \| M_{15}$$
$$|M_i| = 8 \;,\; \forall i = 0(1)15$$

Sub-Bytes

- All operations are defined in $GF(2^8)$.
- Primitive Polynomial $\Rightarrow (x^8 + x^4 + x^2 + x + 1)$

X_0	X_4	X_8	X_{12}
X_1	X_5	X_9	X_{13}
X_2	X_6	X_{10}	X_{14}
X_3	X_7	X_{11}	X_{15}

State

Sub-Bytes →

$SB(X_0)$			
$SB(X_1)$			
$SB(X_2)$			
$SB(X_3)$			

SB → Non-linear function

$SB : \{0,1\}^8 \rightarrow \{0,1\}^8$

Plaintext

M_0	M_4	M_8	M_{12}
M_1	M_5	M_9	M_{13}
M_2	M_6	M_{10}	M_{14}
M_3	M_7	M_{11}	M_{15}

$$M = M_0 \| M_1 \| \cdots \| M_{15}$$
$$|M_i| = 8 \,, \quad \forall i = 0(1)15$$

- All operations are defined in $GF(2^8)$.
- Primitive Polynomial $\Rightarrow x^8 + x^4 + x^3 + x + 1$

Sub-Bytes

X_0	X_4	X_8	X_{12}
X_1	X_5	X_9	X_{13}
X_2	X_6	X_{10}	X_{14}
X_3	X_7	X_{11}	X_{15}

State

Sub-Bytes

$SB(X_0)$			
$SB(X_1)$			
$SB(X_2)$			
$SB(X_3)$			

$SB \rightarrow$ Non-linear function

$SB : \{0,1\}^8 \rightarrow \{0,1\}^8$

Shift Row

0 →
1 →
2 →
3 →

X_0	X_4	X_8	X_{12}
X_1	X_5	X_9	X_{13}
X_2	X_6	X_{10}	X_{14}
X_3	X_7	X_{11}	X_{15}

Shift Row

(Shift i^{th} row left by i steps)

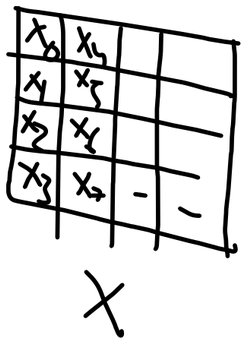
X_0	X_4	X_8	X_{12}
X_5	X_9	X_{13}	X_1
X_{10}	X_{14}	X_2	X_6
X_{15}	X_3	X_7	X_{11}

$X_0 = x^4 + x^3 + 1$

$2 = x$

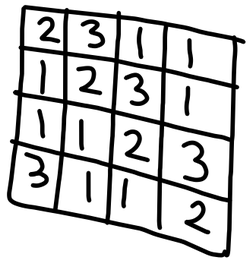
$X_0 \odot 2 = x^5 + x^4 + x$

Mix-Column

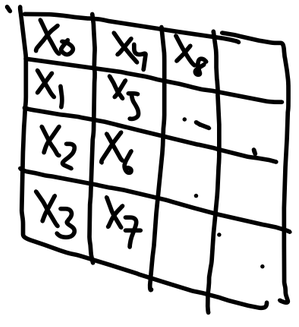


X

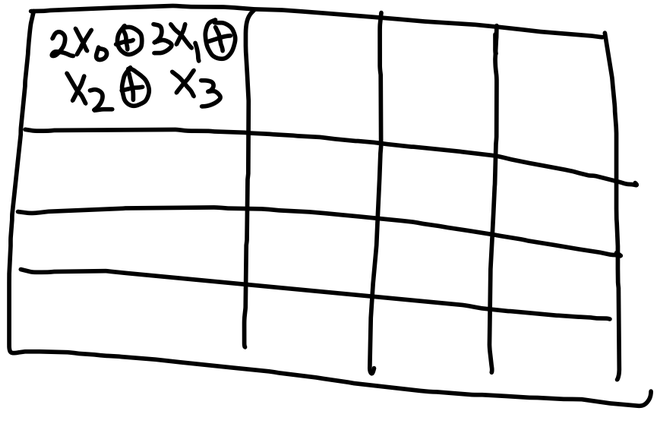
Mix Column



X



=



Add Round Key

i^{th} round Key

X_0	X_4	X_8	X_{12}
X_1	X_5	X_9	X_{13}
X_2	X_6	X_{10}	X_{14}
X_3	X_7	X_{11}	X_{15}

$\oplus$

K_0^i			
K_1^i			
K_2^i			
K_3^i	-	- .	K_{15}^i

$=$

$X_0 \oplus K_0^i$			
$X_1 \oplus K_1^i$			

- Why $\begin{pmatrix} 2&3&1&1 \\ 1&2&3&1 \\ 1&1&2&3 \\ 3&1&1&2 \end{pmatrix}$ used as Mix-Column?
- Why both Shift Row & Mix-Column?
- Rationale behind the S-Box
- Why 10 rounds?
- Why mix-column avoided in the last round?

SBox (Substitution Box)

I/P	O/P
00	— —
⋮	
FF	. .

Look-up table based implementation.

SB

AES SubByte Operation

\	0	1	. . .	F
0	63			
⋮				
F		*		

$GF(2^n)$

$n=8 \Rightarrow GF(2^8)$

Addition → Linear

Multiplication → Non-Linear

Inverse → Non-linear

binary	integer	Hex	Polynomial
1 100 0011	$2^0+2^1+2^6+2^7$ = 195	C3	x^7+x^6+x+1

Primitive Polynomial } $(x^8+x^4+x^3+x+1)$

Hex

0000 → 0
0001 → 1
0010 → 2
0011 → 3
⋮
1001 → 9
1010 → A
1011 → B
1100 → C
1101 → D
1110 → E
1111 → F

$8E \circ 90$

$8E \to 1000\,1110$

$90 \to 1001\,0000$

$(x^7 + x^3 + x^2 + x) \cdot (x^7 + x^4)$

$= x^{14} + x^{11} + x^{10} + x^7 + x^9 + x^6 + x^8 + x^5$

$= x^{14} + x^{11} + x^{10} + x^9 + x^8 + (x^7 + x^6 + x^5)$

$= \ldots \quad (x^4 + x^3 + x + 1)(x^5 + x^4 + x^2 + x)$

$x^8 + x^4 + x^3 + x + 1 = 0$

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

Inverse of α in $GF(2^8)$?

$$\alpha^{-1} \equiv \alpha^{2^8-2}$$
$$\equiv \alpha^{254}$$
$$\Longleftrightarrow$$

(Non-linear function)

Addition

$10 \oplus 11$

$= 1$

$$\begin{array}{r} 1010 \\ 1011 \\ \hline 0001 \end{array}$$

$10 + 11 = 21$ (Integer addition)

$1010 \quad 1011 \quad 10101$

Linear Functions

$x_1\ x_2\ x_3\ x_4$

↓ Rotation

$x_3\ x_4\ x_1\ x_2$

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = M \circ \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

↝ Matrix Multiplication