Grover on SPEEDY

https://youtu.be/DWhlklFPenI

IT융합공학부 송경주

Grover's algorithm

Grover on SPEEDY

자원추정 및 강도평가

Grover's algorithm

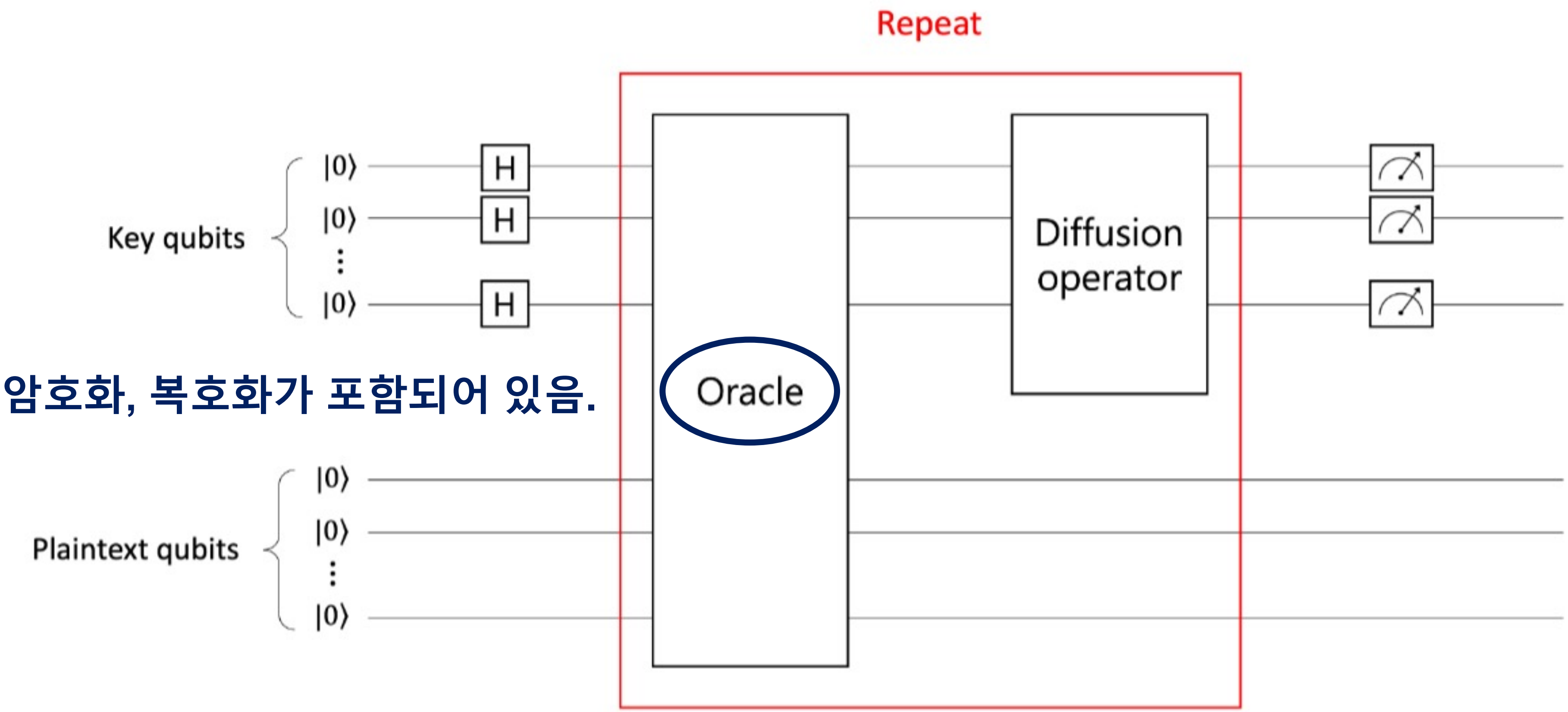
- Quantum algorithm 중 하나로 N개의 정렬되지 않은 데이터에 대해서 데이터를 검색할 복잡도를 $O(\sqrt{N})$로 만들어 줌. (N = 2^n)

정렬되지 않은 N개의 데이터가 주어지면 $\begin{cases} \text{일반컴퓨터 : } O(N) \text{ 복잡도} \\ \text{양자컴퓨터 : } O(\sqrt{N}) \text{ 복잡도} \end{cases}$

[Grover 진행]

1. $|0\rangle^{\otimes n}$ 준비.
2. 모든 큐비트에 H-gate를 적용하여 superposition 상태로 만듦.
3. 오라클을 적용하여 타켓 요소의 부호를 –로 만듦.
4. Grover diffusion operator을 적용시켜 타겟의 확률을 증폭시킴.
5. 3, 4 을 $\left\lfloor \frac{\pi}{4} 2^{\frac{n}{2}} \right\rfloor$번 반복 수행.

Grover’s algorithm



Oracle 안에는 암호화, 복호화가 포함되어 있음.

SPEEDY

- 2021 CHES에서 소개된 a family of ultra low-latency block ciphers.

- SPEEDY-r-6l (r : 라운드 수, 6l : input), 다양한 길이의 입력 가능.

- 192 길이의 입력(l=32)에 대해서 r=7일 때 완전한 보안을 달성한다고 소개됨.

- 6xl array 로 동작.

- 라운드 함수는 Subbox, ShiftColumns, MixColumns, AddRoundConstant, AddRoundKey 로 구성됨

SPEEDY

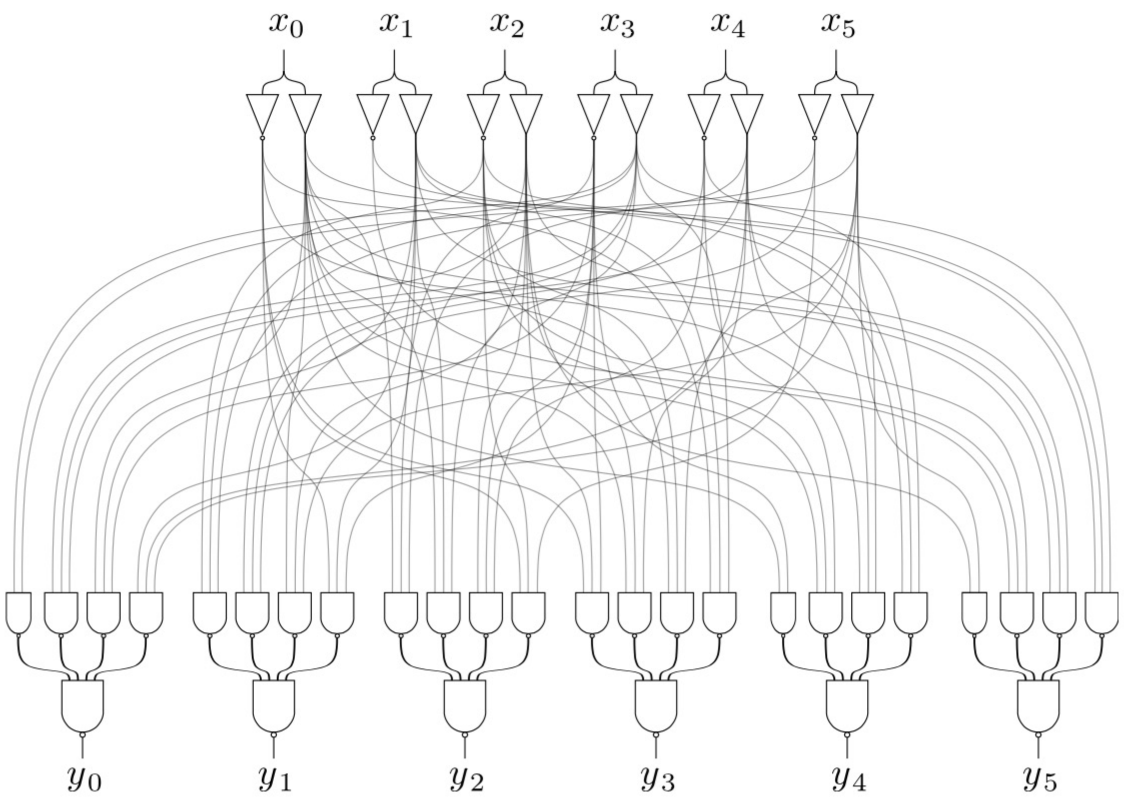
- SubBox(SB)
  - 6bit S-box.
  - 2-level NAND gates로 연산된다.
  - $l \times 6$ array 에서 각 열 6bit 씩 입력된다.

x_0x_1	$x_2x_3x_4x_5$															
	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.a	.b	.c	.d	.e	.f
0.	08	00	09	03	38	10	29	13	0c	0d	04	07	30	01	20	23
1.	1a	12	18	32	3e	16	2c	36	1c	1d	14	37	34	05	24	27
2.	02	06	0b	0f	33	17	21	15	0a	1b	0e	1f	31	11	25	35
3.	22	26	2a	2e	3a	1e	28	3c	2b	3b	2f	3f	39	19	2d	3d

$$
\begin{aligned}
y_0 &= (x_3 \wedge \neg x_5) \vee (x_3 \wedge x_4 \wedge x_2) \vee (\neg x_3 \wedge x_1 \wedge x_0) \vee (x_5 \wedge x_4 \wedge x_1),\\
y_1 &= (x_5 \wedge x_3 \wedge \neg x_2) \vee (\neg x_5 \wedge x_3 \wedge \neg x_4) \vee (x_5 \wedge x_2 \wedge x_0) \vee (\neg x_3 \wedge \neg x_0 \wedge x_1),\\
y_2 &= (\neg x_3 \wedge x_0 \wedge x_4) \vee (x_3 \wedge x_0 \wedge x_1) \vee (\neg x_3 \wedge \neg x_4 \wedge x_2) \vee (\neg x_0 \wedge \neg x_2 \wedge \neg x_5),\\
y_3 &= (\neg x_0 \wedge x_2 \wedge \neg x_3) \vee (x_0 \wedge x_2 \wedge x_4) \vee (x_0 \wedge \neg x_2 \wedge x_5) \vee (\neg x_0 \wedge x_3 \wedge x_1),\\
y_4 &= (x_0 \wedge \neg x_3) \vee (x_0 \wedge \neg x_4 \wedge \neg x_2) \vee (\neg x_0 \wedge x_4 \wedge x_5) \vee (\neg x_4 \wedge \neg x_2 \wedge x_1),\\
y_5 &= (x_2 \wedge x_5) \vee (\neg x_2 \wedge \neg x_1 \wedge x_4) \vee (x_2 \wedge x_1 \wedge x_0) \vee (\neg x_1 \wedge x_0 \wedge x_3).
\end{aligned}
$$

$$
\begin{aligned}
y_0 &= x_3 \oplus x_5x_3 \oplus x_5x_4x_3x_2 \oplus x_5x_4x_1 \oplus x_5x_4x_3x_2x_1 \oplus x_1x_0 \oplus x_5x_4x_1x_0 \oplus x_3x_1x_0 \oplus \\
&\quad x_5x_4x_3x_1x_0\\
y_1 &= x_3 \oplus x_4x_3 \oplus x_5x_4x_3 \oplus x_5x_3x_2 \oplus x_1 \oplus x_3x_1 \oplus x_5x_2x_0 \oplus x_1x_0 \oplus x_3x_1x_0\\
y_2 &= 1 \oplus x_5 \oplus x_5x_2 \oplus x_4x_2 \oplus x_3x_2 \oplus x_4x_3x_2 \oplus x_0 \oplus x_5x_0 \oplus x_4x_0 \oplus x_4x_3x_0 \oplus x_2x_0 \oplus \\
&\quad x_5x_2x_0 \oplus x_3x_1x_0\,,\\
y_3 &= x_2 \oplus x_3x_2 \oplus x_3x_1 \oplus x_5x_0 \oplus x_2x_0 \oplus x_5x_2x_0 \oplus x_4x_2x_0 \oplus x_3x_2x_0 \oplus x_3x_1x_0\\
y_4 &= x_5x_4 \oplus x_1 \oplus x_4x_1 \oplus x_2x_1 \oplus x_4x_2x_1 \oplus x_0 \oplus x_5x_4x_0 \oplus x_4x_3x_0 \oplus x_3x_2x_0 \oplus x_4x_3x_2x_0 \oplus \\
&\quad x_1x_0 \oplus x_4x_1x_0 \oplus x_2x_1x_0 \oplus x_4x_2x_1x_0\,,\\
y_5 &= x_4 \oplus x_5x_2 \oplus x_4x_2 \oplus x_4x_1 \oplus x_4x_2x_1 \oplus x_3x_0 \oplus x_4x_3x_0 \oplus x_5x_3x_2x_0 \oplus x_4x_3x_2x_0 \oplus \\
&\quad x_3x_1x_0 \oplus x_4x_3x_1x_0 \oplus x_2x_1x_0 \oplus x_5x_2x_1x_0 \oplus x_5x_3x_2x_1x_0 \oplus x_4x_3x_2x_1x_0\,.
\end{aligned}
$$

ANF 표현식



SPEEDY

$$
\begin{aligned}
y_0 &= x_3 \oplus x_5x_3 \oplus x_5x_4x_3x_2 \oplus x_5x_4x_1 \oplus x_5x_4x_3x_2x_1 \oplus x_1x_0 \oplus x_5x_4x_1x_0 \oplus x_3x_1x_0 \oplus \\
&\quad x_5x_4x_3x_1x_0 \\
y_1 &= x_3 \oplus x_4x_3 \oplus x_5x_4x_3 \oplus x_5x_3x_2 \oplus x_1 \oplus x_3x_1 \oplus x_5x_2x_0 \oplus x_1x_0 \oplus x_3x_1x_0 \\
y_2 &= 1 \oplus x_5 \oplus x_5x_2 \oplus x_4x_2 \oplus x_3x_2 \oplus x_4x_3x_2 \oplus x_0 \oplus x_5x_0 \oplus x_4x_0 \oplus x_4x_3x_0 \oplus x_2x_0 \oplus \\
&\quad x_5x_2x_0 \oplus x_3x_1x_0\,, \\
y_3 &= x_2 \oplus x_3x_2 \oplus x_3x_1 \oplus x_5x_0 \oplus x_2x_0 \oplus x_5x_2x_0 \oplus x_4x_2x_0 \oplus x_3x_2x_0 \oplus x_3x_1x_0 \\
y_4 &= x_5x_4 \oplus x_1 \oplus x_4x_1 \oplus x_2x_1 \oplus x_4x_2x_1 \oplus x_0 \oplus x_5x_4x_0 \oplus x_4x_3x_0 \oplus x_3x_2x_0 \oplus x_4x_3x_2x_0 \oplus \\
&\quad x_1x_0 \oplus x_4x_1x_0 \oplus x_2x_1x_0 \oplus x_4x_2x_1x_0\,, \\
y_5 &= x_4 \oplus x_5x_2 \oplus x_4x_2 \oplus x_4x_1 \oplus x_4x_2x_1 \oplus x_3x_0 \oplus x_4x_3x_0 \oplus x_5x_3x_2x_0 \oplus x_4x_3x_2x_0 \oplus \\
&\quad x_3x_1x_0 \oplus x_4x_3x_1x_0 \oplus x_2x_1x_0 \oplus x_5x_2x_1x_0 \oplus x_5x_3x_2x_1x_0 \oplus x_4x_3x_2x_1x_0\,.
\end{aligned}
$$

Input: $x_0, x_1, x_2, x_3, x_4, x_5$
Output: $y_0, y_1, y_2, y_3, y_4, y_5$

$y_0 \leftarrow$ CNOT(x_3, y_0)
Toffoli(x_5, x_3, y_0)
CCCCX(x_5, x_4, x_3, x_2, y_0)
CCCX(x_5, x_4, x_1, y_0)
CCCCCX(x_5, x_4, x_3, x_2, x_1, y_0)
Toffoli(x_1, x_0, y_0)
CCCCX(x_5, x_4, x_1, x_0, y_0)
CCCX(x_3, x_1, x_0, y_0)
CCCCCX(x_5, x_4, x_3, x_1, x_0, y_0)

$y_1 \leftarrow$ CNOT(x_3, y_1)
Toffoli(x_4, x_3, y_1)
CCCX(x_5, x_4, x_3, y_1)
CCCX(x_5, x_3, x_2, y_1)
CNOT(x_1, y_1)
Toffoli(x_3, x_1, y_1)
CCCX(x_5, x_2, x_0, y_1)
Toffoli(x_1, x_0, y_1)
CCCX(x_3, x_1, x_0, y_1)

$y_2 \leftarrow$ NOT(y_2)
CNOT(x_5, y_2)
Toffoli(x_5, x_2, y_2)
Toffoli(x_4, x_2, y_2)
Toffoli(x_3, x_2, y_2)
CCCX(x_4, x_3, x_2, y_2)
CNOT(x_0, y_2)
Toffoli(x_5, x_0, y_2)
Toffoli(x_4, x_0, y_2)
CCCX(x_4, x_3, x_0, y_2)
Toffoli(x_2, x_0, y_2)
CCCX(x_5, x_2, x_0, y_2)
CCCX(x_3, x_1, x_0, y_2)

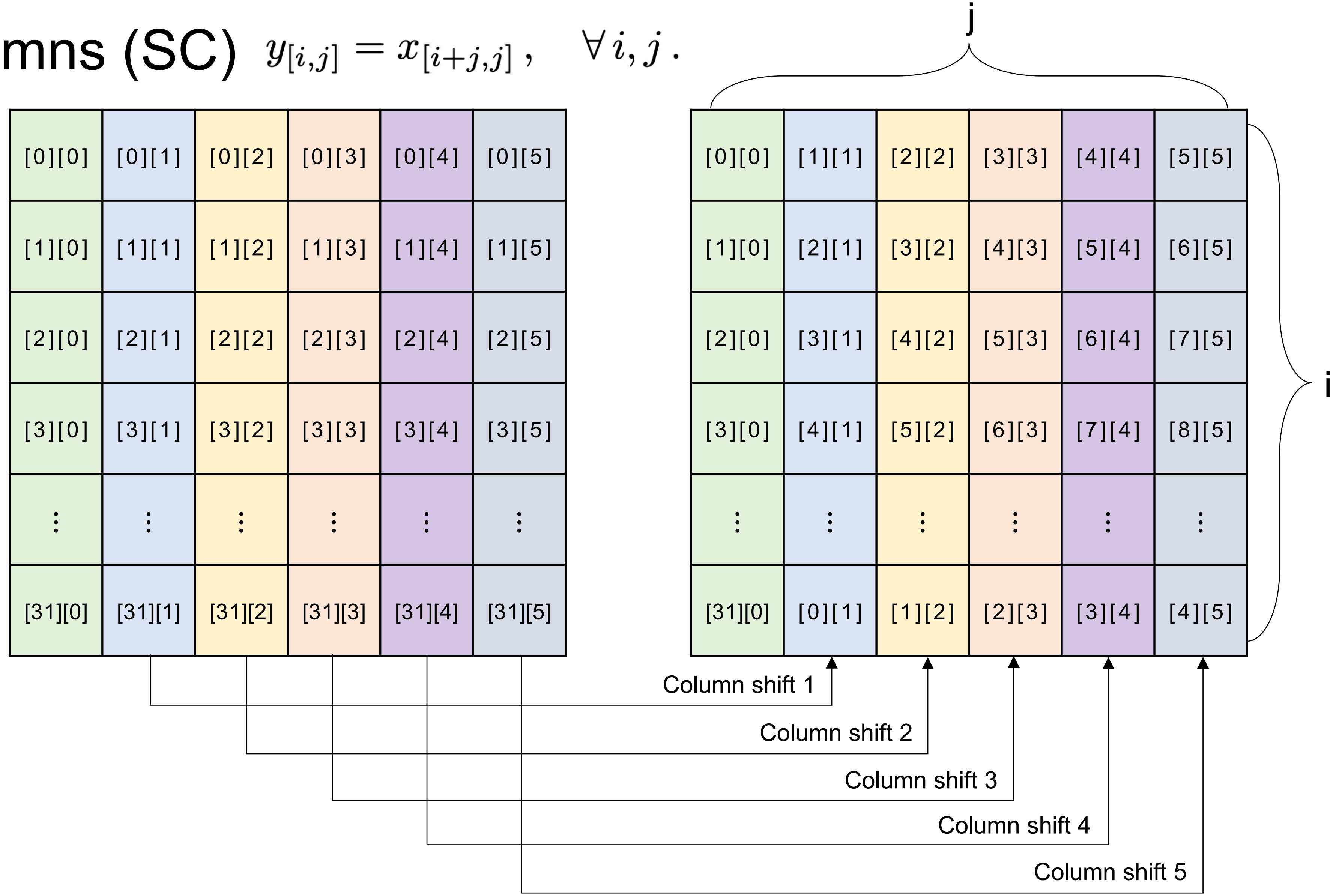
$y_3 \leftarrow$ CNOT(x_2,y_3)
Toffoli(x_3, x_2, y_3)
Toffoli(x_3, x_1, y_3)
Toffoli(x_5, x_0, y_3)
Toffoli(x_2, x_0, y_3)
CCCX(x_5, x_2, x_0, y_3)
CCCX(x_4, x_2, x_0, y_3)
CCCX(x_3, x_2, x_0, y_3)
CCCX(x_3, x_1, x_0, y_3)

$y_4 \leftarrow$ Toffoli(x_5, x_4, y_4)
CNOT(x_1, y_4)
Toffoli(x_4, x_1, y_4)
Toffoli(x_2, x_1, y_4)
CCCX(x_4, x_2, x_1, y_4)
CNOT(x_0, y_4)
CCCX(x_5, x_4, x_0, y_4)
CCCX(x_4, x_3, x_0, y_4)
CCCX(x_3, x_2, x_0, y_4)
CCCCX(x_4, x_3, x_2, x_0, y_4)
Toffoli(x_1, x_0, y_4)
CCCX(x_4, x_1, x_0, y_4)
CCCX(x_2, x_1, x_0, y_4)
CCCCX(x_4, x_2, x_1, x_0, y_4)

$y_5 \leftarrow$ CNOT(x_4, y_5)
Toffoli(x_5, x_2, y_5)
Toffoli(x_4, x_2, y_5)
Toffoli(x_4, x_1, y_5)
CCCX(x_4, x_2, x_1, y_5)
Toffoli(x_3, x_0, y_5)
CCCX(x_4, x_3, x_0, y_5)
CCCCX(x_5, x_3, x_2, x_0, y_5)
CCCCX(x_4, x_3, x_2, x_0, y_5)
CCCX(x_3, x_1, x_0, y_5)
CCCCX(x_4, x_3, x_1, x_0, y_5)
CCCX(x_2, x_1, x_0, y_5)
CCCCX(x_5, x_2, x_1, x_0, y_5)
CCCCCX(x_5, x_3, x_2, x_1, x_0, y_5)
CCCCCX(x_4, x_3, x_2, x_1, x_0, y_5)

SPEEDY

- ShiftColumns (SC) $y_{[i,j]} = x_{[i+j,j]}\ , \quad \forall\, i, j\ .$



SPEEDY

- ShiftColumns (SC) $y_{[i,j]} = x_{[i+j,j]}\,, \quad \forall i, j\,.$

Logical SWAP을 사용하여 별도의 게이트 비용 X

Input: $array = [x_0, x_1, \cdots, x_{192}]$
Output: $array = [x_0, x_7, x_{14}, \cdots, x_{29}]$

```
for i = 0 to 31 do
    for j = 0 to 5 do
        array[ ] ← array.append(x_(6*(i+j)+j))
    end for
end for
```

SPEEDY

- MixColumns (MC)

각 위치에 맞는 행들의 XOR연산을 수행한다. (이때 α는 정해진 상수)

$$y_{[i,j]} = x_{i,j} \oplus x_{[i+\alpha_1,j]} \oplus x_{[i+\alpha_2,j]} \oplus x_{[i+\alpha_3,j]} \oplus x_{[i+\alpha_4,j]} \oplus x_{[i+\alpha_5,j]} \oplus x_{[i+\alpha_6,j]}\,, \quad \forall i,j\,.$$

- AddRoundKey (A_{k_r})

$$y_{[i,j]} = x_{[i,j]} \oplus k_{r\,[i,j]}\,, \quad \forall i,j\,.$$

- AddRoundConstant (A_{c_r})

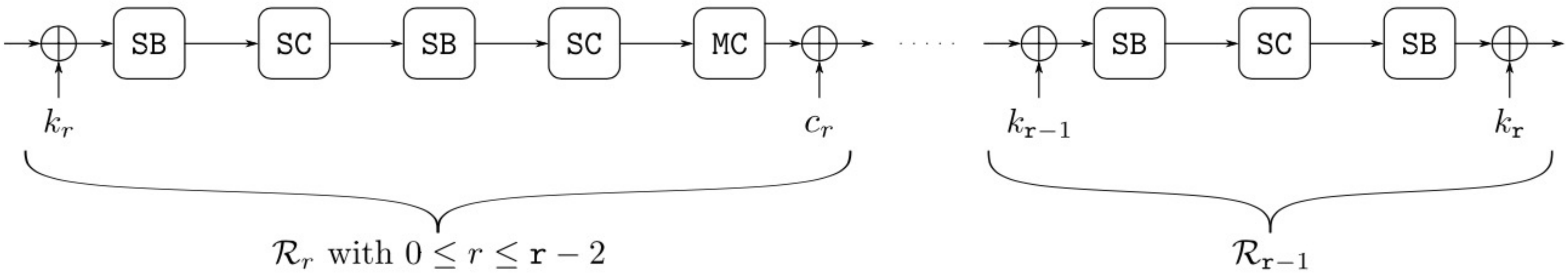
$$y_{[i,j]} = x_{[i,j]} \oplus c_{r\,[i,j]}\,, \quad \forall i,j\,.$$

SPEEDY

- Round Function (R_n) : 0~(r-1)라운드

  0 ~ (r-2)라운드는 A_{k_n}, SB, SC, MC, A_{c_n}순서로 동작,

  마지막 라운드(r-1)만 A_{k_n}, SB, SC, SB, $A_{k_{n+1}}$ 순서로 동작.



$$R_n = \begin{cases} A_{c_n} \circ MC \circ SC \circ SB \circ A_{k_n} & (0 < n < r-2) \\ A_{k_{n+1}} \circ SB \circ SC \circ SB \circ A_{k_n} & (n = r-1) \end{cases}$$

Grover 자원추정

r : 그루버 키 탐색 알고리즘 비용을 추정하기 위해 필요한 평문 쌍, $r = \left\lceil \frac{key\ size}{block\ size} \right\rceil$

$\times 2 \times r \times \left\lfloor \frac{\pi}{4} 2^{\frac{n}{2}} \right\rfloor$ = 그루버 알고리즘에 필요한 리소스

1 $\left\lfloor \frac{\pi}{4} 2^{96} \right\rfloor$

```
Gate counts:
    Allocate : 4224
    CCCCCX : 1792
    CCCCX : 3584
    CCCX : 10752
    CCX : 10304
    CX : 13632
    Deallocate : 4224
    Swap : 2688
    X : 1018
```

SPEEDY 암호화 리소스 측정값

Gates					
NOT	CNOT	Toffoli	CCCX	CCCCX	CCCCCX
1.56×2^{106}	1.30×2^{110}	1.97×2^{109}	1.03×2^{110}	1.37×2^{108}	1.37×2^{107}

NIST 기준 강도평가

Gates					
NOT	CNOT	Toffoli	CCCX	CCCCX	CCCCCX
1.56×2^{106}	1.30×2^{110}	1.97×2^{109}	1.03×2^{110}	1.37×2^{108}	1.37×2^{107}

Non-Clifford gate를 T+Clifford gate로 분해

r	Gates		Total gates	Total depth	Cost
	T	Clifford			
1	$1.48 \cdot 2^{115}$	$1.16 \cdot 2^{113}$	$1.77 \cdot 2^{115}$	$1.56 \cdot 2^{106}$	$1.38 \cdot 2^{222}$

Total gates × Total depth = Cost

Level 1	Any attack that breaks the relevant security definition must require computational resources comparable to or greater than those required for key search on a block cipher with a 128-bit key (e.g. AES 128)
Level 3	Any attack that breaks the relevant security definition must require computational resources comparable to or greater than those required for key search on a block cipher with a 192-bit key (e.g. AES 192)
Level 5	Any attack that breaks the relevant security definition must require computational resources comparable to or greater than those required for key search on a block cipher with a 256-bit key (e.g. AES 256)

Cipher	AES			SPEEDY
	128	192	256	7-192
Cost	2^{170}	2^{233}	2^{298}	2^{222}
Level	Level 1	Level 3	Level 5	Level 1

Q & A