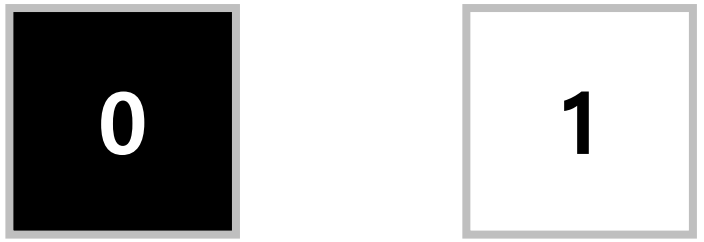
양자컴퓨팅 아는 척하기

최유빈(sean9892)

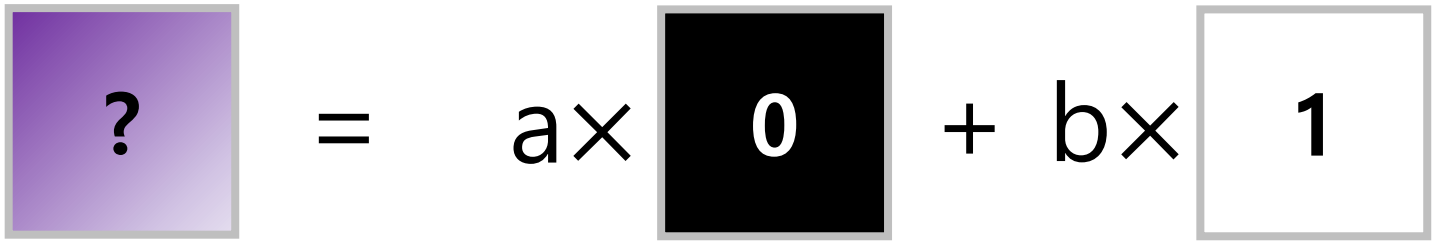
목차

- **Part I. What is “Qubit”?**
  - Qubit의 수학적 표현
  - 양자 게이트
- **Part II. Shor’s Algorithm**
  - 개요
  - 주기탐색 알고리즘
- **Part III. Quantum Cryptology**

Part I. What is “Qubit”?



Part I. What is “Qubit”?



Part I. What is “Qubit”?

$$|\psi\rangle = a|0\rangle + b|1\rangle$$

$$\langle\psi|\psi\rangle = a^2 + b^2 = 1$$

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Bra : $\langle v| = [v_1 \quad v_2 \quad \cdots \quad v_n]$

Ket : $|v\rangle = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$

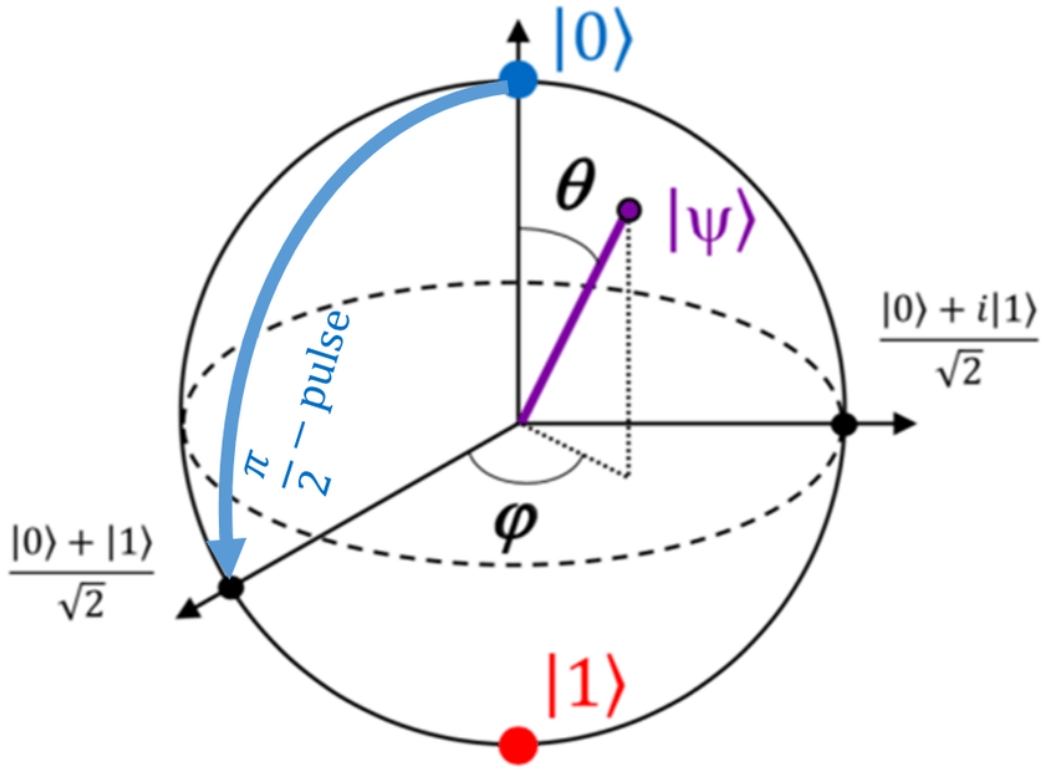
Part I. What is “Qubit”?

$$|\psi\rangle = a|0\rangle + b|1\rangle = Ae^{i\theta}|0\rangle + Be^{i\phi}|1\rangle$$

$$|\psi\rangle = A|0\rangle + Be^{i\phi}|1\rangle$$

$$A^2 + \left|Be^{i\phi}\right|^2 = A^2 + B^2 = 1$$

Part I. What is “Qubit”?



$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle$$

Part I. What is “Qubit”?

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\longrightarrow \quad |x_1\rangle_1 \otimes |x_2\rangle_2 \cdots$$

Part I. What is “Qubit”?

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$|x_1\rangle_1 \otimes |x_2\rangle_2 \cdots$$

Tensor Product

Part I. What is “Qubit”?

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\longrightarrow \quad |x_1\rangle_1 \otimes |x_2\rangle_2 \cdots$$

Tensor Product

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}B & a_{n2}B & \cdots & a_{nn}B \end{bmatrix}$$

Part I. What is “Qubit”?

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\longrightarrow \quad |x_1\rangle_1 \otimes |x_2\rangle_2 \cdots$$

Tensor Product

$$\begin{bmatrix} 1 & 3 \\ 0 & 2 \end{bmatrix} \otimes \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 3 & 9 \\ 0 & 6 \\ 2 & 6 \\ 0 & 4 \end{bmatrix}$$

Part I. What is “Qubit”?

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$|x_1\rangle_1 \otimes |x_2\rangle_2 \cdots$$

Tensor Product

$|n\rangle$: 위에서부터 n번째(0-based) element만이 1

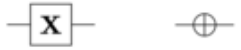
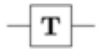
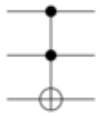
Part I. What is “Qubit”?

Unitary Matrix $UU^* = I$

Pauli-X (X)  $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

Controlled Not (CNOT, CX)  $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

Hadamard (H)  $\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

Operator	Gate(s)	Matrix
Pauli-X (X)		$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y (Y)		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z (Z)		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard (H)		$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Phase (S, P)		$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$\pi/8$ (T)		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$
Controlled Not (CNOT, CX)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
Controlled Z (CZ)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$
SWAP		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Toffoli (CCNOT, CCX, TOFF)		$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

Part II. Shor's Algorithm

정수 N의 비자명 약수를 비트 수에 대한 다항 시간 복잡도에 구하는 알고리즘

1. 임의의 정수 1<a<N를 선택한다.
   gcd(a,N)을 계산하고, 1이 아니라면 gcd(a,N)을 반환한다.

2. **함수 $f(x) = a^x mod\ N$의 주기 r을 계산한다.**

3. r이 홀수라면 1로 되돌아가 다시 시행한다.

4. $a^{\frac{r}{2}} \equiv -1\ (mod\ N)$이라면 1로 되돌아가 다시 시행한다.

5. $\gcd\left(a^{\frac{r}{2}} - 1, N\right), \gcd\left(a^{\frac{r}{2}} + 1, N\right)$을 반환한다.

Part II. Shor's Algorithm

$$a^r \equiv 1 \ (mod \ N)$$

$$\left(a^{\frac{r}{2}}\right)^2 - 1 \equiv 0 \ (mod \ N)$$

$$\left(a^{\frac{r}{2}} + 1\right)\left(a^{\frac{r}{2}} - 1\right) = mN$$

$$\Rightarrow \gcd\left(a^{\frac{r}{2}} - 1, N\right), \gcd\left(a^{\frac{r}{2}} + 1, N\right)$$

Part II. Shor's Algorithm

$$N^2 < Q = 2^q < 2N^2 \Rightarrow \frac{Q}{r} > N$$

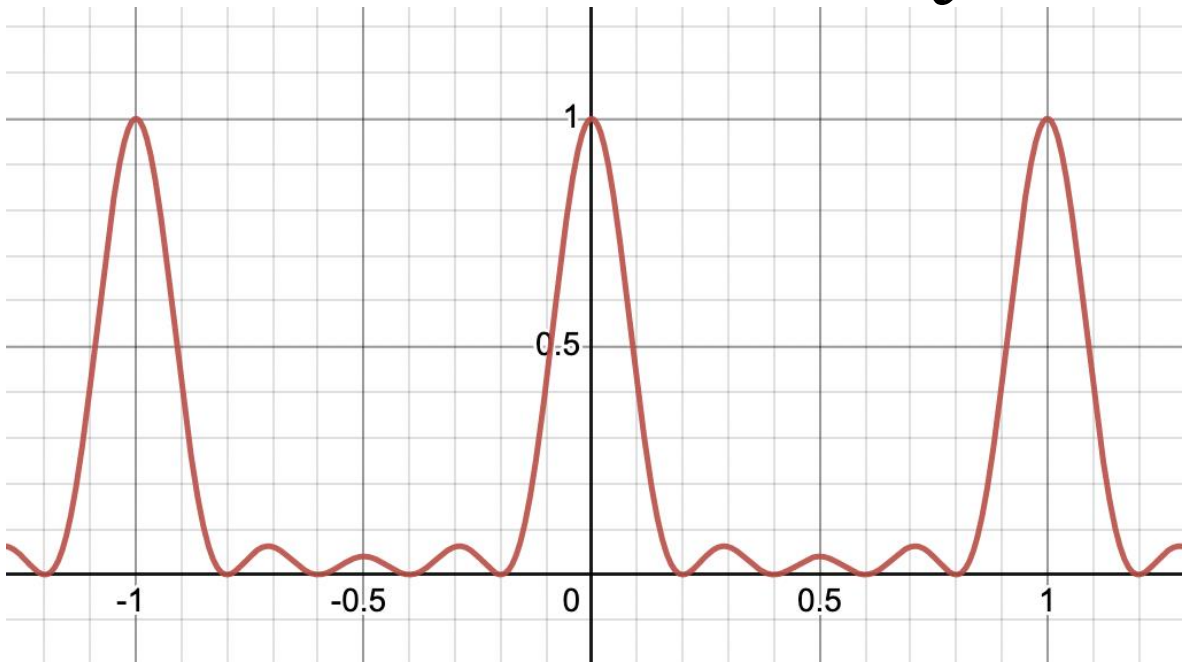
1. q개의 qubits를 $\frac{1}{\sqrt{Q}}\sum_{x=0}^{Q-1}|x\rangle = \left(\frac{1}{\sqrt{2}}\sum_{x_1=0}^{1}|x_1\rangle\right)\otimes\left(\frac{1}{\sqrt{2}}\sum_{x_2=0}^{1}|x_2\rangle\right)\otimes\cdots$로 초기화
   Hadamard Gate를 사용하여 수행 가능

2. 함수 f를 quantum logic gate로 구성(repeated squaring)
   $\mathrm{U_f}\frac{1}{\sqrt{Q}}\sum_{x=0}^{Q-1}|x,0^n\rangle = \frac{1}{\sqrt{Q}}\sum_{x=0}^{Q-1}|x,f(x)\rangle$

3. 앞의 q개의 qubits에 QFT를 적용
   $\frac{1}{Q}\sum_{x=0}^{Q-1}\sum_{y=0}^{Q-1}e^{\frac{2\pi ixy}{Q}}|y,f(x)\rangle = \frac{1}{Q}\sum_{z=0}^{Q-1}\sum_{y=0}^{Q-1}\left[\sum_{f(x)=z}e^{\frac{2\pi ixy}{Q}}\right]|y,z\rangle$

Part II. Shor's Algorithm

$$\Pr(|y,z\rangle) = \frac{1}{Q^2} \frac{\sin^2 \frac{\pi m r y}{Q}}{\sin^2 \frac{\pi r y}{Q}}$$



Part II. Shor's Algorithm

$$\frac{yr}{Q} \approx c \in \mathbb{Z}$$

$$\frac{y}{Q} \approx \frac{c}{r}$$

$$\Rightarrow \frac{d}{s} \approx \frac{c}{r}$$

Part III. Quantum Cryptology

Part II. Shor's Algorithm

정수 N의 비자명 약수를 비트 수에 대한 다항 시간 복잡도에 구하는 알고리즘

1. 임의의 정수 1<a<N를 선택한다.
   gcd(a,N)을 계산하고, 1이 아니라면 gcd(a,N)을 반환한다.
2. **함수 $f(x) = a^x mod\ N$의 주기 r을 계산한다.**
3. r이 홀수라면 1로 되돌아가 다시 시행한다.
4. $a^{\frac{r}{2}} \equiv -1\ (mod\ N)$이라면 1로 되돌아가 다시 시행한다.
5. $\gcd\left(a^{\frac{r}{2}} - 1, N\right), \gcd\left(a^{\frac{r}{2}} + 1, N\right)$을 반환한다.

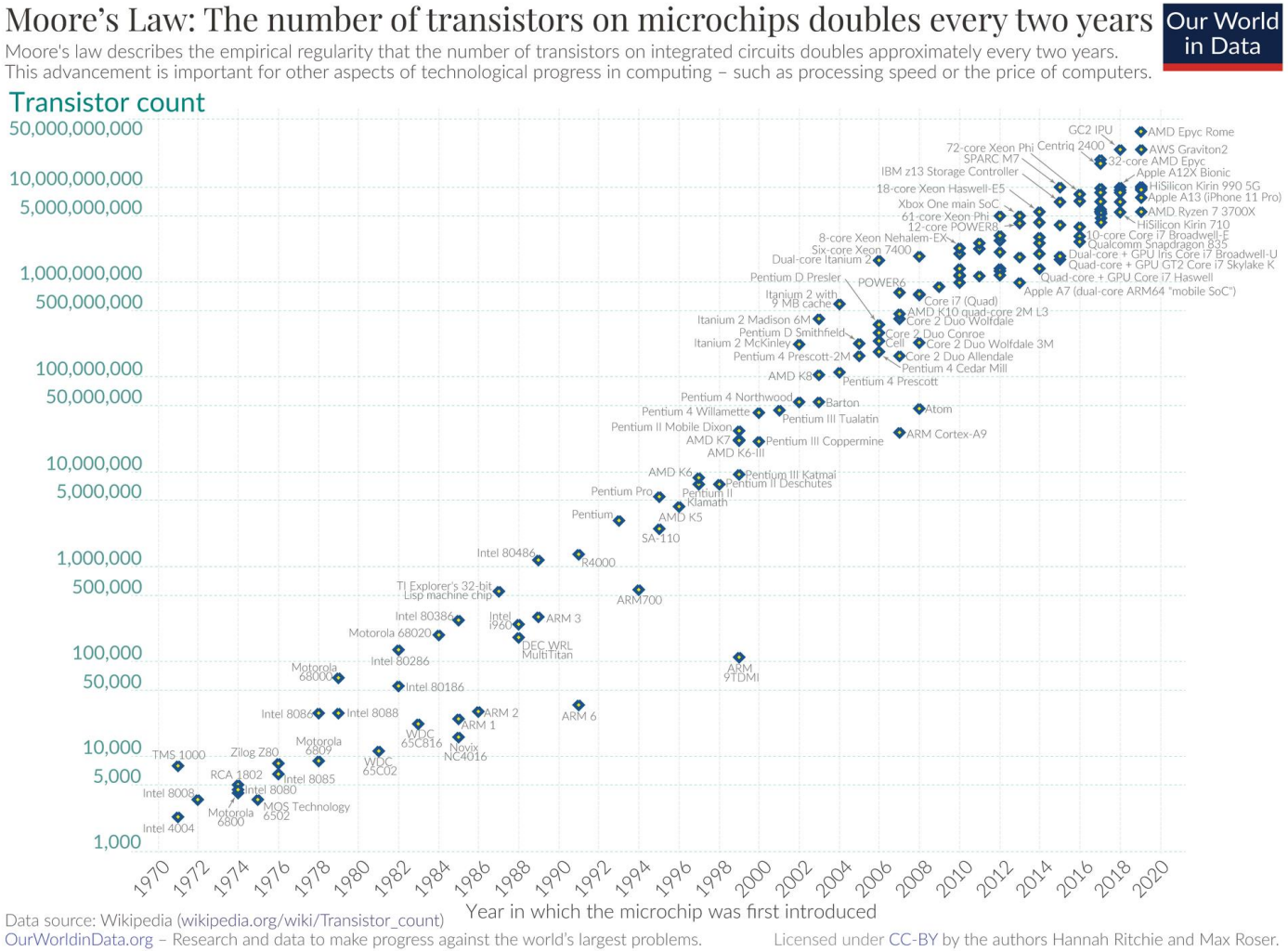
Part III. Quantum Cryptology

2. 함수 $f(x) = a^x mod\ N$의 주기 r을 계산한다.

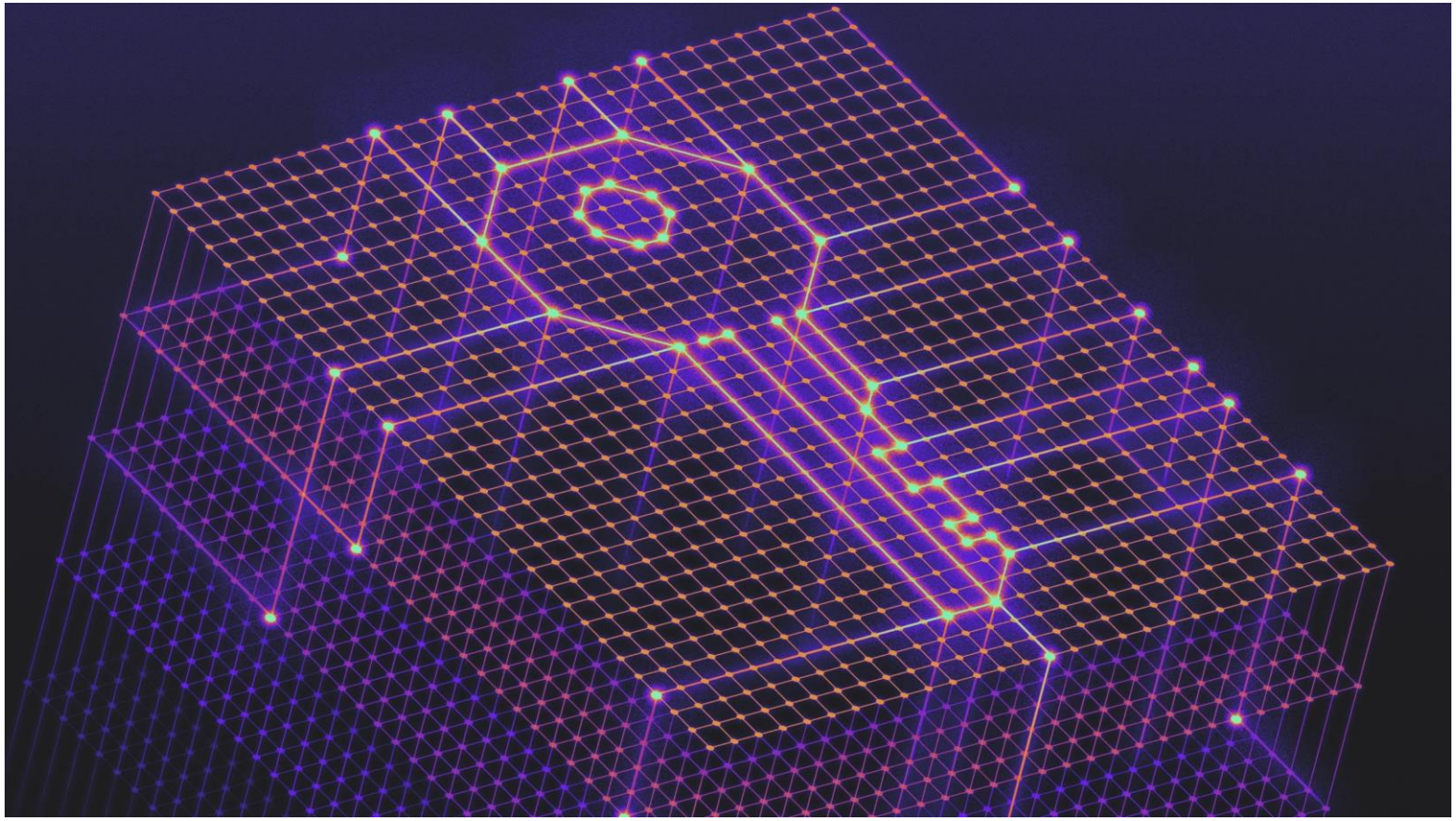
Part III. Quantum Cryptology

Part III. Quantum Cryptology

Part III. Quantum Cryptology



Part III. Quantum Cryptology



Thank You for Listening

양자컴퓨터 아는 척하기