Amsterdam University of Applied Sciences

Quantum Factorization: Shor's algorithm

Quantum Capita Selecta

Bernardo Villalba Frías, PhD
b.r.villalba.frias@hva.nl

RSA Cryptosystem

Amsterdam University of Applied Sciences

- Rivest–Shamir–Adleman (RSA)
- Asymmetric–key system
  - Encryption key (e, N) is public
  - Decryption key (d) is private (and different)
- Based on the difficulty of the factorization of the product of two large prime numbers
- RSA algorithm involves four steps:
  - Key generation
  - Key distribution
  - Encryption, and
  - Decryption

RSA Cryptosystem

- Alice encrypts a message M for Bob, which he has to decrypt
  - Encrytion:
    - Obtain Bob's authentic public key $(e,\ N)$
    - Pad M into m
    - Encode the message: $c = m^e \pmod{N}$
    - Send c to Bob
  - Decryption:
    - Use Bob's private key $(d,\ N)$
    - Decode the message: $m = c^d \pmod{N}$
    - De–pad m into M

- Toy example:

Encryption key: $(e,\ N)$

$(7,\ 15)$

Plain message: $B \to 2$

Encoded message: $2^7 \pmod{15}$

$128 \pmod{15}$

$8 \to H$

Decryption key: $(d,\ N)$

$(23,\ 15)$

$8^{23} \pmod{15}$

$590295810358705651712 \pmod{15}$

$2 \to B$

RSA Cryptosystem

- How do we generate the keys?

1. Pick 2 prime numbers: p and q

$$p = 3 \quad \wedge \quad q = 5$$

2. Obtain the modulus (N):

$$\begin{aligned} N &= p \times q \\ &= 3 \times 5 \\ &= 15 \end{aligned}$$

3. Compute the function Φ:

1	2	$\cancel{3}$	4	$\cancel{5}$
$\cancel{6}$	7	8	$\cancel{9}$	$\cancel{10}$
11	$\cancel{12}$	13	14	$\cancel{15}$

- Remove the numbers sharing a common factor with N
- Keep only coprime with N

$$C = \{1,\ 2,\ 4,\ 7,\ 8,\ 11,\ 13,\ 14\}$$

- How do we generate the keys?

3. Compute the function Φ:

$$\begin{aligned}\Phi &= |C| \\ &= 8\end{aligned} \qquad \begin{aligned}\Phi &= (p-1)\times(q-1) \\ &= (3-1)\times(5-1) \\ &= 2\times 4 \\ &= 8\end{aligned}$$

4. Choose the encryption key (e):

$$e = \begin{cases} 1 < e < \Phi \\ \text{coprime with } N \text{ and } \Phi \end{cases}$$

$$\begin{aligned}\text{coprime with } N &\rightarrow e \in \{1,\, 2,\, 4,\, 7,\, 8,\, 11,\, 13,\, 14\} \\ 1 < e < \Phi &\rightarrow e \in \{2,\, 4,\, 7\} \\ \text{coprime with } \Phi &\rightarrow e \in \{7\} \\ &\Rightarrow (e,\, N) = (7,\, 15)\end{aligned}$$

Amsterdam University of Applied Sciences

- How do we generate the keys?

5. Choose the decryption key (d):

$$(d \times e) \pmod{\Phi} = 1$$

$$7d \pmod{8} = 1$$

$$\begin{array}{rcccccccccc}
d & \rightarrow & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & \ldots \\
7d & \rightarrow & 7 & 14 & 21 & 28 & 35 & 42 & 49 & 56 & \ldots \\
7d \pmod{8} & \rightarrow & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 & \ldots \\
 & & & & & & & & \uparrow & & \\
 & & & & & & & & \text{pick} & &
\end{array}$$

$$\rightarrow d \in \{7,\ 15,\ 23,\ 31,\ 39,\ \ldots\}$$

$$\Rightarrow (d,\ N) = (23,\ 15)$$

RSA Cryptosystem

- What about a more realistic example?
  - RSA–64:

$$
\begin{aligned}
p &= 3970211251 \\
q &= 3873813143 \\
N &= 15379856524610271893 \\
\Phi &= 15379856516766247500 \\
e &= 15114048278816893619 \\
d &= 7635707568842743979
\end{aligned}
$$

RSA Cryptosystem

- Could you imagine RSA–4096?

RSA Cryptosystem

- How does it break?
  - Public key $(e,\ N)$
  - Factor N into p and q:

$$N = p \times q$$

  - Compute Φ:

$$\Phi = (p-1) \times (q-1)$$

  - Obtain d

$$d = \text{Inverse of } e \pmod{\Phi}$$

Shor's algorithm

Amsterdam University of Applied Sciences

- Quantum algorithm to find the prime factors of any given integer N
- It takes $\mathcal{O}((\log N)^3)$ time:
  - Solved in polynomial time by a quantum computer
  - Making RSA vulnerable to attacks
- It consists of two parts:
  - Reducing the factoring problem into an order–finding problem
  - Solve the order–finding problem (Quantum period finding)

Shor's algorithm

Amsterdam University of Applied Sciences

- Preliminary Theorem:
  - Suppose that:
    - N is a non–prime number
    - x is a solution of $x^2 \pmod{N} = 1$
    - $x \pmod{N} \neq 1$ and $x \pmod{N} \neq -1$
  - Then:
    - $\mathtt{gcd}(x-1,\ N)$ and $\mathtt{gcd}(x+1,\ N)$ are non–trivial factors of N

Shor's algorithm

1. Choose a random integer $a < N$, such that they are co–prime
2. Determine the unknown period r of the function:
$$f(x) = a^x \pmod N$$
3. If r is odd, then go to Step 1; otherwise go to Step 4
4. Since r is even:
$$((a^{\frac{r}{2}} - 1)(a^{\frac{r}{2}} + 1)) \pmod N = 0$$
If $(a^{\frac{r}{2}} + 1) \pmod N = 0$, then go to Step 1
If $(a^{\frac{r}{2}} + 1) \pmod N \neq 0$, then go to Step 5
5. Compute:
$$p = \gcd(a^{\frac{r}{2}} - 1,\, N)$$
$$q = \gcd(a^{\frac{r}{2}} + 1,\, N)$$

- Toy example:

$$N = 15$$
$$a = 2$$

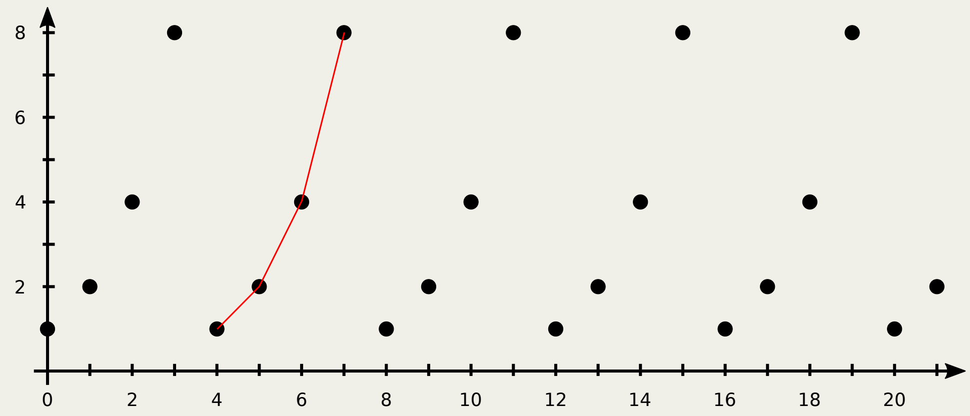
$$\begin{aligned}
a^r &\pmod{N} = 1 \\
2^0 &\pmod{15} = 1 \\
2^1 &\pmod{15} = 2 \\
2^2 &\pmod{15} = 4 \\
2^3 &\pmod{15} = 8 \\
2^4 &\pmod{15} = 1
\end{aligned}$$

$\Rightarrow r = 4$ (even)

$$(2^2 + 1) \pmod{15} \neq 0$$

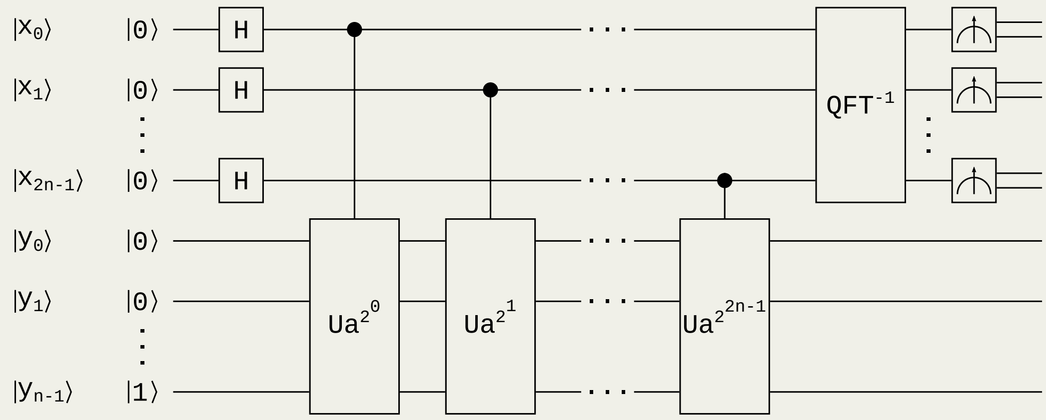
$$\begin{aligned}
p &= \mathtt{gcd}(a^{\frac{r}{2}} - 1,\ N) \\
&= \mathtt{gcd}(2^2 - 1,\ 15) \\
&= \mathtt{gcd}(3,\ 15) \\
&= 3
\end{aligned}$$

$$\begin{aligned}
q &= \mathtt{gcd}(a^{\frac{r}{2}} + 1,\ N) \\
&= \mathtt{gcd}(2^2 + 1,\ 15) \\
&= \mathtt{gcd}(5,\ 15) \\
&= 5
\end{aligned}$$



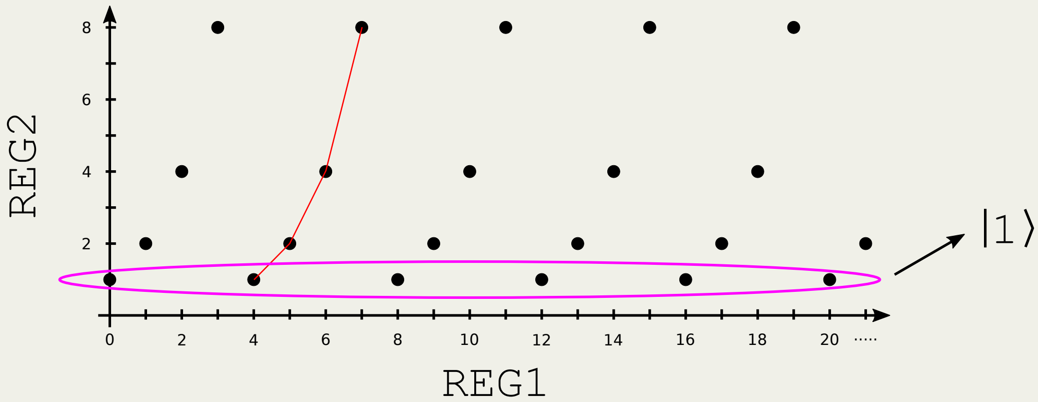
- Period finding:
  - Main building block of the Shor's algorithm
  - $f(x) = a^x \pmod N$ is periodic
  - Challenge: Find the period!... but, how?
    - Modular exponentiation
      - Resource expensive
    - Inverse Quantum Fourier Transform

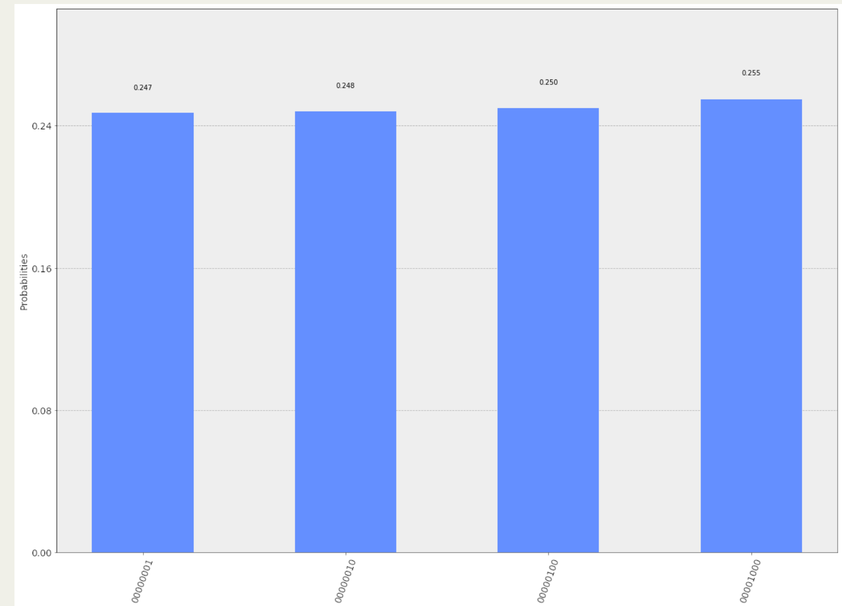
Shor's algorithm



- Creates a quantum register with 2 parts:
  1. $\texttt{REG1}$: Superposition of x in $a^x \pmod{N}$
     - $x \in [0,\ b-1]$, with $N^2 < 2^b < 2N^2 \rightarrow 2^b = 256$
  2. $\texttt{REG2}$: Possible outcomes of $a^x \pmod{N}$
- Measuring the second register ($\texttt{REG2}$) collapses:
  - $\texttt{REG2}$ into some observed value
  - $\texttt{REG1}$ into a state consistent with $\texttt{REG2}$

Shor's algorithm

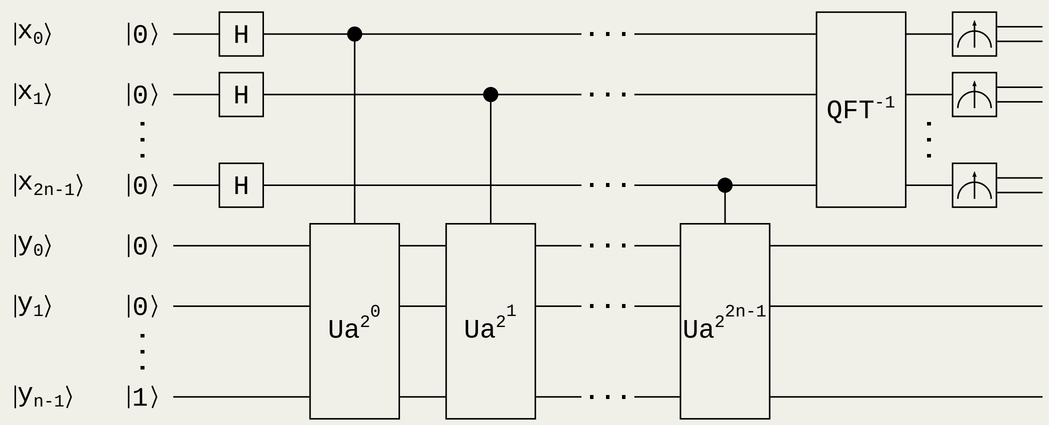




- Assume $\mathtt{REG2}$ collapses to $|1\rangle$
  - $\mathtt{REG1}$ will collapse to

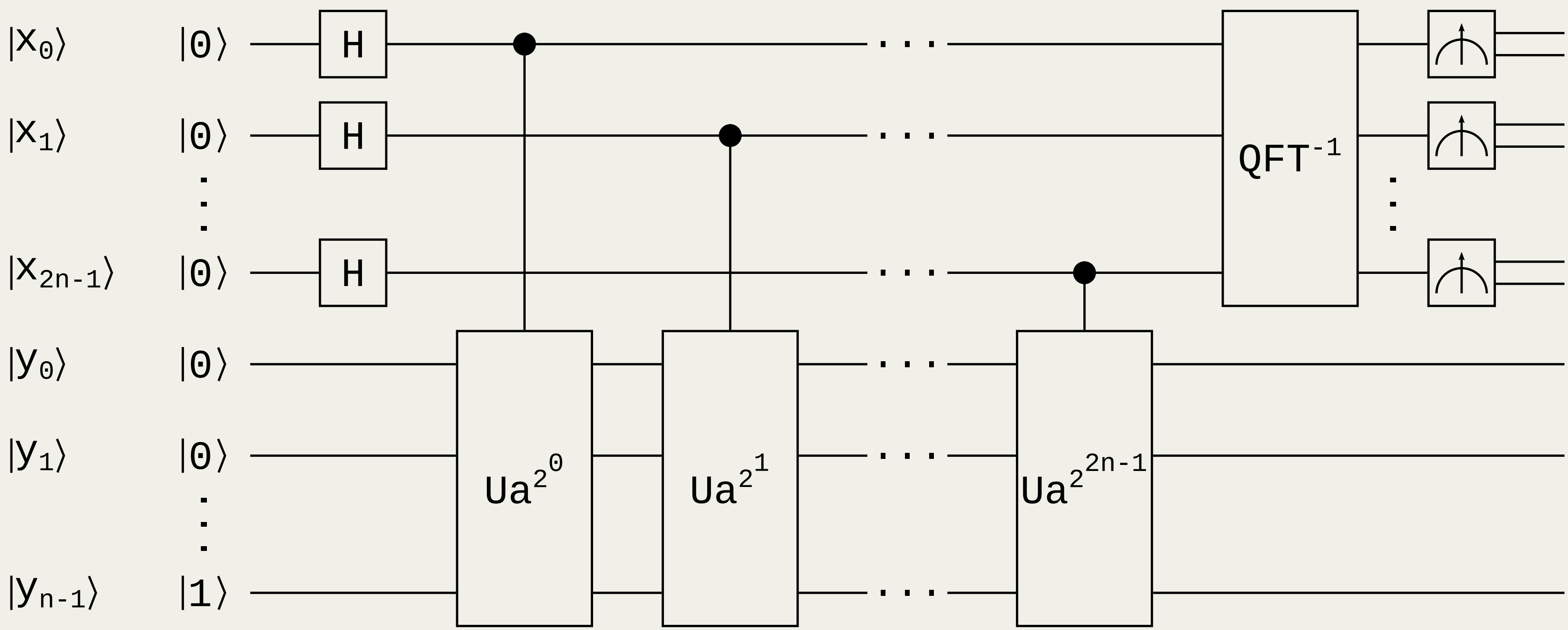
$$\frac{1}{\sqrt{64}}\left(|0\rangle + |4\rangle + |8\rangle + \ldots\right)$$

Shor's algorithm

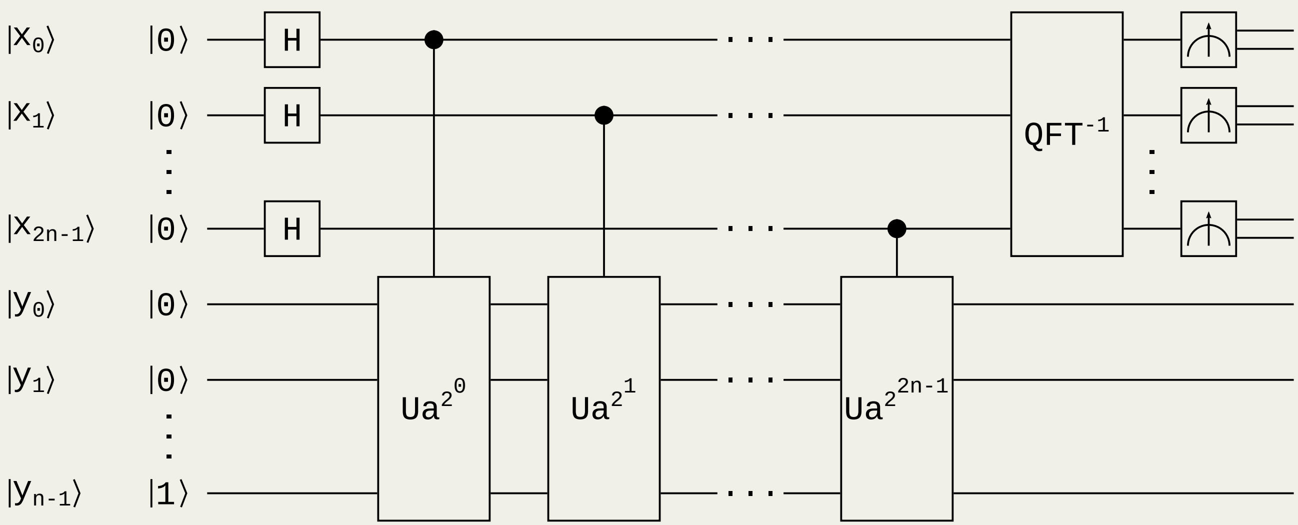


- Inverse Quantum Fourier Transform on REG1
  - Peak the probabilities amplitudes at integer multiples of the $\frac{2^b}{r}$
  - Measuring REG1 will collapse, with high probability, to a multiple of the inverse period
  - Additional processing required

Shor’s algorithm



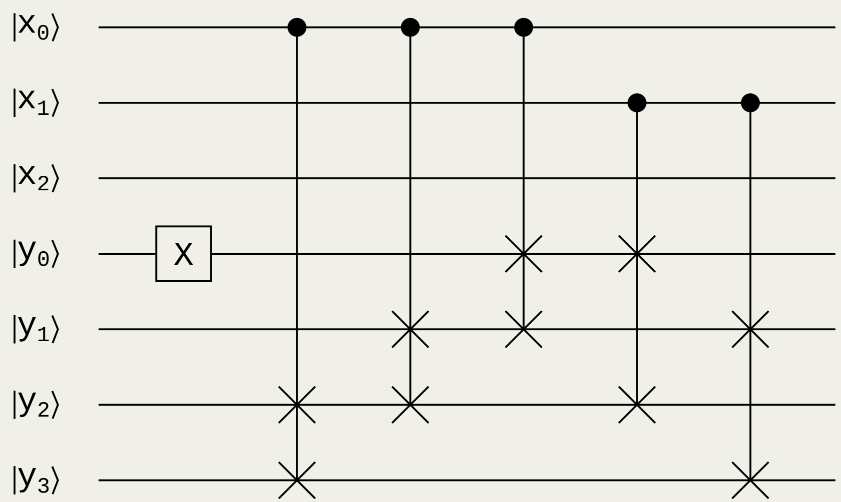
Shor's algorithm

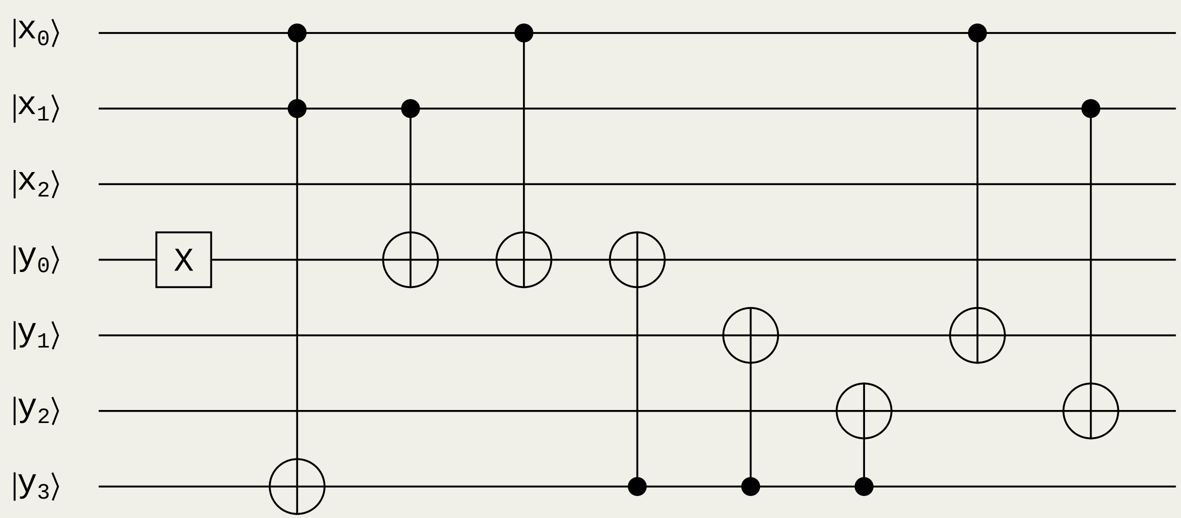


- Modular exponentiation:
  - $U_a : |x\rangle\,|0\rangle \rightarrow |x\rangle\,|a^x \pmod N)\rangle$
  - Repeated squaring: $x^n = (x^{\frac{n}{2}})^2$
  - Implementation depends on a
  - How do we build a generic circuit for a random a?
    - "Compiled" versions

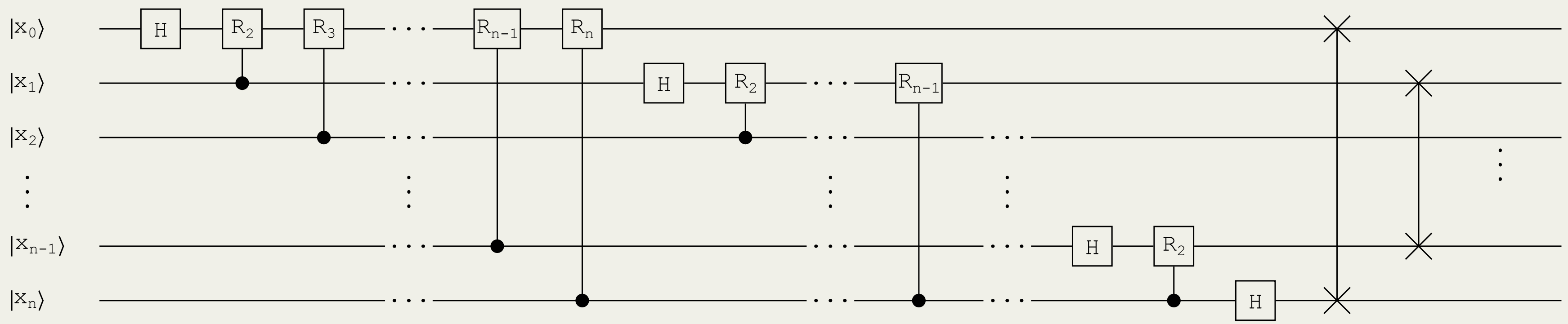
Shor's algorithm

- Compiled circuits for $N = 15$ and $a = 2$





- Quantum Fourier Transform:



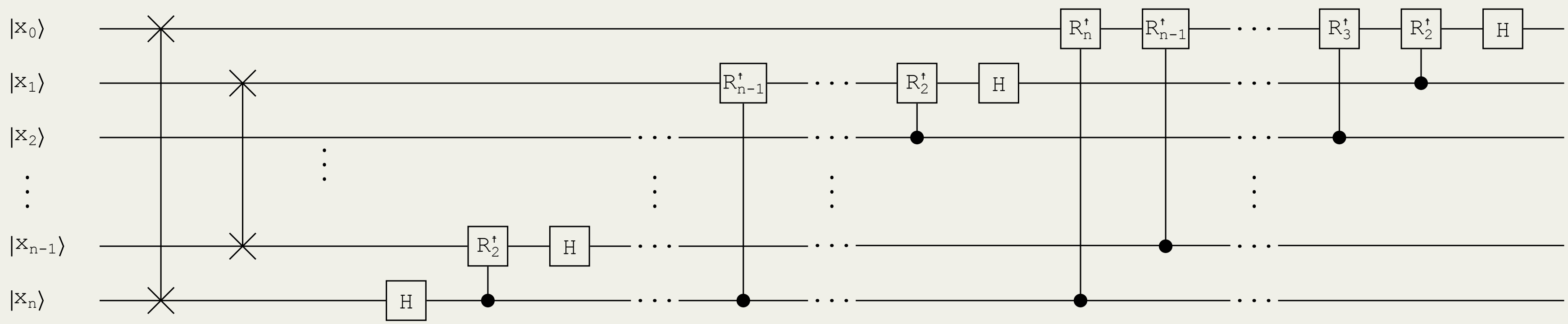
  - Several quantum algorithms use the QFT
  - Implemented by: Hadamard, CR$_m$ and Swap gates

$$CR_m = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{\lambda i} \end{bmatrix}, \text{ where } \lambda = \frac{2\pi}{2^m}$$

- But, we need the inverse!

Shor's algorithm

- Inverse Quantum Fourier Transform:



  - Flip the gates' order: from right to left
  - Also the gates should be inverted
    - $H = H^\dagger \rightarrow$ No problem!
    - $R_m \neq R_m^\dagger \rightarrow$ Obtain $R_m^\dagger$
      - λ has negative sign $(-\lambda)$
      - In Qiskit: `qc.cu1(−λ, ctrl, tgt)`

Shor's algorithm

- Continued fractions

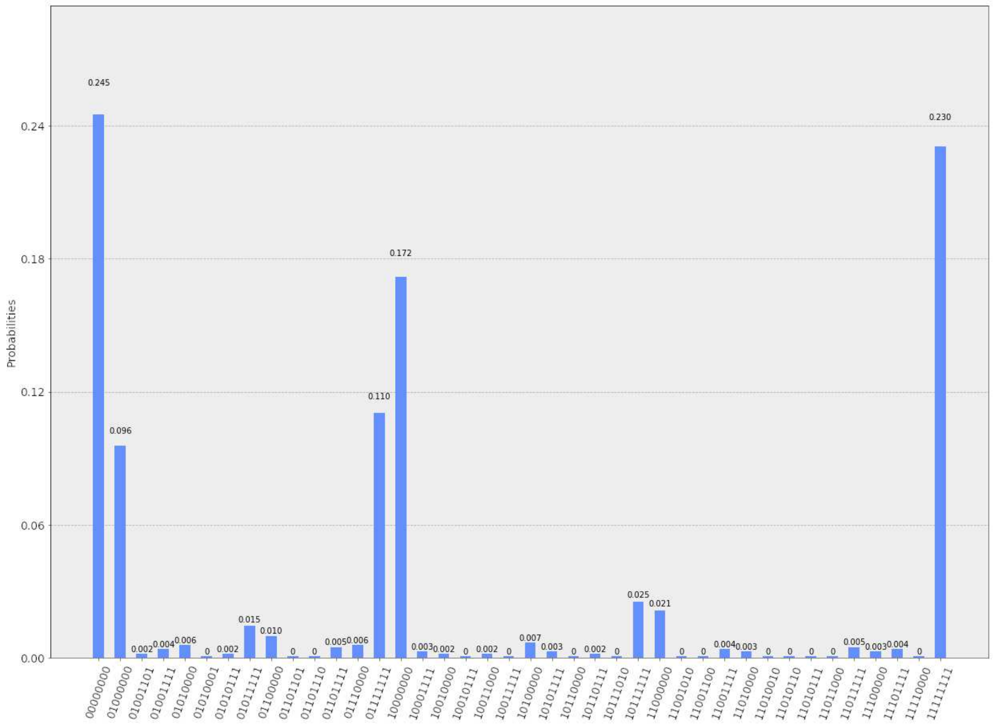
$$[a_0, \ldots, a_M] = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots + \cfrac{1}{a_M}}}}$$

- Any rational number can be represented as a continued fraction:

$$\begin{aligned}\frac{13}{5} &= 2 + \frac{3}{5} \\ &= 2 + \frac{1}{\frac{5}{3}} \\ &= 2 + \frac{1}{1 + \frac{2}{3}} \\ &= 2 + \frac{1}{1 + \frac{1}{\frac{3}{2}}} \\ &= 2 + \frac{1}{1 + \frac{1}{1 + \frac{1}{2}}}\end{aligned}$$

Shor's algorithm

- Continuous fractions for post–processing



- Probable measurements: $\{0, 64, 127, 128, 191, 255\}$
- Let's assume that we measured $s = 191$

Shor's algorithm

- Continuous fractions for post–processing
- Assume $s = 191$

$$
\begin{aligned}
\frac{256}{191} &= 1 + \frac{1}{\frac{191}{65}} \\
&= 1 + \frac{1}{2 + \frac{1}{\frac{65}{61}}} \\
&= 1 + \frac{1}{2 + \frac{1}{1+\frac{1}{\frac{61}{4}}}} \\
&= 1 + \frac{1}{2 + \frac{1}{1+\frac{1}{15+\frac{1}{4}}}}
\end{aligned}
$$

Shor's algorithm

- Continuous fractions for post–processing
- Assume $s = 191$
  - Coefficients: $[1,\ 2,\ 1,\ 15,\ 4]$
  - Compute the m–th convergent:

$$
\begin{aligned}
\text{3–rd} &= [1,\ 2,\ 1] \\
&= 1 + \frac{1}{2+1} \\
&= 1 + \frac{1}{3} \\
&= \frac{4}{3}
\end{aligned}
$$

Shor's algorithm

- Continuous fractions for post–processing
- Assume $s = 191$
  - Coefficients: $[1,\ 2,\ 1,\ 15,\ 4]$
  - m–th convergent: $\left[1,\ \frac{3}{2},\ \frac{4}{3},\ \frac{63}{47},\ \frac{256}{191}\right]$
- We are interested in the numerators:
  - Even
  - $a^x \pmod N = 1$
- Possible periods: $[4,\ 256]$
- Compute $\gcd(a^{\frac{r}{2}} \pm 1,\ N)$
  - $r = 4 \rightarrow p = 3$ and $q = 5$


Questions?