

IQI 04, Seminar 9

Produced with pdflatex and xfig

- Factoring algorithms.
- Quantum order finding.
- Quantum Fourier transform.

E. “Manny” Knill: knill@boulder.nist.gov

Composite and Prime Numbers

- $\mathbb{N} = \{0, 1, 2, 3, \dots\}$, the set of natural numbers.

Composite and Prime Numbers

- $\mathbb{N} = \{0, 1, 2, 3, \dots\}$, the set of natural numbers.
- $f|n$ means f divides n .

Composite and Prime Numbers

- $\mathbb{N} = \{0, 1, 2, 3, \dots\}$, the set of natural numbers.
- $f|n$ means f divides n .
- f is a *proper factor* of n if $1 < f < n$ and $f|n$.

Composite and Prime Numbers

- $\mathbb{N} = \{0, 1, 2, 3, \dots\}$, the set of natural numbers.
- $f|n$ means f divides n .
- f is a *proper factor* of n if $1 < f < n$ and $f|n$.
- p is a prime if $1 < p$ and p has no proper factors.

Composite and Prime Numbers

- $\mathbb{N} = \{0, 1, 2, 3, \dots\}$, the set of natural numbers.
- $f|n$ means f divides n .
- f is a *proper factor* of n if $1 < f < n$ and $f|n$.
- p is a prime if $1 < p$ and p has no proper factors.
- Examples:
 - Primes: 2, 3, 5, 7, 11, 13, 17, ...

Composite and Prime Numbers

- $\mathbb{N} = \{0, 1, 2, 3, \dots\}$, the set of natural numbers.
- $f|n$ means f divides n .
- f is a *proper factor* of n if $1 < f < n$ and $f|n$.
- p is a prime if $1 < p$ and p has no proper factors.
- Examples:
 - Primes: 2, 3, 5, 7, 11, 13, 17, ...
 - What divides 15? $15 = 3 * 5$ so $3|15, 5|15$.

Composite and Prime Numbers

- $\mathbb{N} = \{0, 1, 2, 3, \dots\}$, the set of natural numbers.
- $f|n$ means f divides n .
- f is a *proper factor* of n if $1 < f < n$ and $f|n$.
- p is a prime if $1 < p$ and p has no proper factors.
- Examples:
 - Primes: 2, 3, 5, 7, 11, 13, 17, ...
 - What divides 15? $15 = 3 * 5$ so $3|15, 5|15$.
 - Which of the following are prime?

23 :

Composite and Prime Numbers

- $\mathbb{N} = \{0, 1, 2, 3, \dots\}$, the set of natural numbers.
- $f|n$ means f divides n .
- f is a *proper factor* of n if $1 < f < n$ and $f|n$.
- p is a prime if $1 < p$ and p has no proper factors.
- Examples:
 - Primes: 2, 3, 5, 7, 11, 13, 17, ...
 - What divides 15? $15 = 3 * 5$ so $3|15, 5|15$.
 - Which of the following are prime?
23 : prime

Composite and Prime Numbers

- $\mathbb{N} = \{0, 1, 2, 3, \dots\}$, the set of natural numbers.
- $f|n$ means f divides n .
- f is a *proper factor* of n if $1 < f < n$ and $f|n$.
- p is a prime if $1 < p$ and p has no proper factors.
- Examples:
 - Primes: 2, 3, 5, 7, 11, 13, 17, ...
 - What divides 15? $15 = 3 * 5$ so $3|15, 5|15$.
 - Which of the following are prime?
23 : prime
21 :

Composite and Prime Numbers

- $\mathbb{N} = \{0, 1, 2, 3, \dots\}$, the set of natural numbers.
- $f|n$ means f divides n .
- f is a *proper factor* of n if $1 < f < n$ and $f|n$.
- p is a prime if $1 < p$ and p has no proper factors.
- Examples:
 - Primes: 2, 3, 5, 7, 11, 13, 17, ...
 - What divides 15? $15 = 3 * 5$ so $3|15, 5|15$.
 - Which of the following are prime?
 - 23 : prime
 - 21 : $3 * 7$

Composite and Prime Numbers

- $\mathbb{N} = \{0, 1, 2, 3, \dots\}$, the set of natural numbers.
- $f|n$ means f divides n .
- f is a *proper factor* of n if $1 < f < n$ and $f|n$.
- p is a prime if $1 < p$ and p has no proper factors.
- Examples:
 - Primes: 2, 3, 5, 7, 11, 13, 17, ...
 - What divides 15? $15 = 3 * 5$ so $3|15, 5|15$.
 - Which of the following are prime?

23 : prime

21 : $3 * 7$

47 :

Composite and Prime Numbers

- $\mathbb{N} = \{0, 1, 2, 3, \dots\}$, the set of natural numbers.
- $f|n$ means f divides n .
- f is a *proper factor* of n if $1 < f < n$ and $f|n$.
- p is a prime if $1 < p$ and p has no proper factors.
- Examples:
 - Primes: 2, 3, 5, 7, 11, 13, 17, ...
 - What divides 15? $15 = 3 * 5$ so $3|15, 5|15$.
 - Which of the following are prime?

23	:	prime
21	:	$3 * 7$
47	:	prime

Composite and Prime Numbers

- $\mathbb{N} = \{0, 1, 2, 3, \dots\}$, the set of natural numbers.
- $f|n$ means f divides n .
- f is a *proper factor* of n if $1 < f < n$ and $f|n$.
- p is a prime if $1 < p$ and p has no proper factors.
- Examples:
 - Primes: 2, 3, 5, 7, 11, 13, 17, ...
 - What divides 15? $15 = 3 * 5$ so $3|15, 5|15$.
 - Which of the following are prime?

23	:	prime
21	:	$3 * 7$
47	:	prime
51	:	

Composite and Prime Numbers

- $\mathbb{N} = \{0, 1, 2, 3, \dots\}$, the set of natural numbers.
- $f|n$ means f divides n .
- f is a *proper factor* of n if $1 < f < n$ and $f|n$.
- p is a prime if $1 < p$ and p has no proper factors.
- Examples:
 - Primes: 2, 3, 5, 7, 11, 13, 17, ...
 - What divides 15? $15 = 3 * 5$ so $3|15, 5|15$.
 - Which of the following are prime?

23	:	prime
21	:	$3 * 7$
47	:	prime
51	:	$3 * 17$

Composite and Prime Numbers

- $\mathbb{N} = \{0, 1, 2, 3, \dots\}$, the set of natural numbers.
 - $f|n$ means f divides n .
 - f is a *proper factor* of n if $1 < f < n$ and $f|n$.
 - p is a prime if $1 < p$ and p has no proper factors.
 - Examples:
 - Primes: 2, 3, 5, 7, 11, 13, 17, ...
 - What divides 15? $15 = 3 * 5$ so $3|15, 5|15$.
 - Which of the following are prime?
- | | | |
|----|---|----------|
| 23 | : | prime |
| 21 | : | $3 * 7$ |
| 47 | : | prime |
| 51 | : | $3 * 17$ |
| 91 | : | |

Composite and Prime Numbers

- $\mathbb{N} = \{0, 1, 2, 3, \dots\}$, the set of natural numbers.
- $f|n$ means f divides n .
- f is a *proper factor* of n if $1 < f < n$ and $f|n$.
- p is a prime if $1 < p$ and p has no proper factors.
- Examples:
 - Primes: 2, 3, 5, 7, 11, 13, 17, ...
 - What divides 15? $15 = 3 * 5$ so $3|15, 5|15$.
 - Which of the following are prime?

23 : prime

21 : $3 * 7$

47 : prime

51 : $3 * 17$

91 : $7 * 13$

Factoring

- The factoring problem:

Given: $N \in \mathbb{N}, N > 1, N$ not a prime.

Problem: Write N as a product $N = p * q, p > 1, q > 1$.



Factoring

- The factoring problem:

Given: $N \in \mathbb{N}, N > 1, N$ not a prime.

Problem: Write N as a product $N = p * q, p > 1, q > 1$.

- Examples:

49 :



Factoring

- The factoring problem:

Given: $N \in \mathbb{N}, N > 1, N$ not a prime.

Problem: Write N as a product $N = p * q, p > 1, q > 1$.

- Examples:

$$49 : 49 = 7 * 7$$



Factoring

- The factoring problem:

Given: $N \in \mathbb{N}, N > 1, N$ not a prime.

Problem: Write N as a product $N = p * q, p > 1, q > 1$.

- Examples:

$$49 : 49 = 7 * 7$$

$$24 :$$



Factoring

- The factoring problem:

Given: $N \in \mathbb{N}, N > 1, N$ not a prime.

Problem: Write N as a product $N = p * q, p > 1, q > 1$.

- Examples:

$$49 : 49 = 7 * 7$$

$$24 : 24 = 8 * 3 = 4 * 6 = \dots = 2^3 * 3$$



Factoring

- The factoring problem:

Given: $N \in \mathbb{N}, N > 1, N$ not a prime.

Problem: Write N as a product $N = p * q, p > 1, q > 1$.

- Examples:

$$49 : 49 = 7 * 7$$

$$24 : 24 = 8 * 3 = 4 * 6 = \dots = 2^3 * 3$$

$$42 :$$



Factoring

- The factoring problem:

Given: $N \in \mathbb{N}, N > 1, N$ not a prime.

Problem: Write N as a product $N = p * q, p > 1, q > 1$.

- Examples:

$$49 : 49 = 7 * 7$$

$$24 : 24 = 8 * 3 = 4 * 6 = \dots = 2^3 * 3$$

$$42 : 42 = 2 * 21 = 6 * 7 = \dots = 2 * 3 * 7$$



Factoring

- The factoring problem:

Given: $N \in \mathbb{N}, N > 1, N$ not a prime.

Problem: Write N as a product $N = p * q, p > 1, q > 1$.

- Examples:

$$49 : 49 = 7 * 7$$

$$24 : 24 = 8 * 3 = 4 * 6 = \dots = 2^3 * 3$$

$$42 : 42 = 2 * 21 = 6 * 7 = \dots = 2 * 3 * 7$$

- Input size: Number of binary digits of $N, n = \lfloor \log_2(N) \rfloor + 1$.



Factoring

- The factoring problem:

Given: $N \in \mathbb{N}, N > 1, N$ not a prime.

Problem: Write N as a product $N = p * q, p > 1, q > 1$.

- Examples:

$$49 : 49 = 7 * 7$$

$$24 : 24 = 8 * 3 = 4 * 6 = \dots = 2^3 * 3$$

$$42 : 42 = 2 * 21 = 6 * 7 = \dots = 2 * 3 * 7$$

- Input size: Number of binary digits of $N, n = \lfloor \log_2(N) \rfloor + 1$.
- Complexity of factoring.
 - Best classical algorithm: $\simeq e^{1.93 \ln(N)^{1/3} \ln \ln(N)^{1-1/3}}$
Number field sieve, complexity based on number density heuristics.



Factoring

- The factoring problem:

Given: $N \in \mathbb{N}, N > 1, N$ not a prime.

Problem: Write N as a product $N = p * q, p > 1, q > 1$.

- Examples:

$$49 : 49 = 7 * 7$$

$$24 : 24 = 8 * 3 = 4 * 6 = \dots = 2^3 * 3$$

$$42 : 42 = 2 * 21 = 6 * 7 = \dots = 2 * 3 * 7$$

- Input size: Number of binary digits of $N, n = \lfloor \log_2(N) \rfloor + 1$.
- Complexity of factoring.
 - Best classical algorithm: $\simeq e^{1.93 \ln(N)^{1/3} \ln \ln(N)^{1-1/3}}$
Number field sieve, complexity based on number density heuristics.
Note: This is *subexponential* but *superpolynomial*.



Factoring

- The factoring problem:

Given: $N \in \mathbb{N}, N > 1, N$ not a prime.

Problem: Write N as a product $N = p * q, p > 1, q > 1$.

- Examples:

$$49 : 49 = 7 * 7$$

$$24 : 24 = 8 * 3 = 4 * 6 = \dots = 2^3 * 3$$

$$42 : 42 = 2 * 21 = 6 * 7 = \dots = 2 * 3 * 7$$

- Input size: Number of binary digits of $N, n = \lfloor \log_2(N) \rfloor + 1$.
- Complexity of factoring.
 - Best classical algorithm: $\simeq e^{1.93 \ln(N)^{1/3} \ln \ln(N)^{1-1/3}}$
Number field sieve, complexity based on number density heuristics.
Note: This is *subexponential* but *superpolynomial*.
 - Best quantum algorithm: $\tilde{O}(n^2)$.
Shor's algorithm. $\tilde{O}(f(n))$ means that for some $k, O(f(n) \log(n)^k)$



Trial Division Algorithm

- TRIALDIVISIONFACTOR(N)

Input: $N > 1$

Output: If N is prime, 1, else a proper factor f of N .

```
for  $f = 2$  to  $\lfloor \sqrt{N} \rfloor$ 
    if  $f|N$  then return  $f$ 
end
return 1
```



Trial Division Algorithm

- TRIALDIVISIONFACTOR(N)

Input: $N > 1$

Output: If N is prime, 1, else a proper factor f of N .

```
for  $f = 2$  to  $\lfloor \sqrt{N} \rfloor$ 
    if  $f|N$  then return  $f$ 
end
return 1
```

- Complexity: $\lfloor \sqrt{N} \rfloor - 1$ trial divisions in the worst case.
If f is the smallest factor, it requires $f - 1$ trial divisions.



Trial Division Algorithm

- TRIALDIVISIONFACTOR(N)

Input: $N > 1$

Output: If N is prime, 1, else a proper factor f of N .

```
for  $f = 2$  to  $\lfloor \sqrt{N} \rfloor$ 
    if  $f|N$  then return  $f$ 
end
return 1
```

- Complexity: $\lfloor \sqrt{N} \rfloor - 1$ trial divisions in the worst case.
If f is the smallest factor, it requires $f - 1$ trial divisions.
- Division with remainder:

Input: $1 \leq f < g$

Output: $\lfloor \frac{g}{f} \rfloor$ and the remainder $g \% f = g - \lfloor \frac{g}{f} \rfloor f$



Trial Division Algorithm

- TRIALDIVISIONFACTOR(N)

Input: $N > 1$

Output: If N is prime, 1, else a proper factor f of N .

```
for  $f = 2$  to  $\lfloor \sqrt{N} \rfloor$ 
    if  $f|N$  then return  $f$ 
end
return 1
```

- Complexity: $\lfloor \sqrt{N} \rfloor - 1$ trial divisions in the worst case.
If f is the smallest factor, it requires $f - 1$ trial divisions.
- Division with remainder:

Input: $1 \leq f < g$

Output: $\lfloor \frac{g}{f} \rfloor$ and the remainder $g \% f = g - \lfloor \frac{g}{f} \rfloor f$

– Complexity: $O(\log(g)^3)$ for long division,
 $\tilde{O}(\log(g)^2)$ for the best algorithm known.



Modular Arithmetic

- $\mathbb{Z}_q = \{\underline{0}, \dots, \underline{q-1}\}$ with modular addition, multiplication rules:
 1. Add/multiply as integers,
 2. then *reduce modulo q* (remainder after division by q).



Modular Arithmetic

- $\mathbb{Z}_q = \{\underline{0}, \dots, \underline{q-1}\}$ with modular addition, multiplication rules:
 1. Add/multiply as integers,
 2. then *reduce modulo q* (remainder after division by q).

Notation: $a \bmod q = \underline{a\%q}$. $a = b \bmod q$ means $a\%q = b\%q$.



Modular Arithmetic

- $\mathbb{Z}_q = \{\underline{0}, \dots, \underline{q-1}\}$ with modular addition, multiplication rules:
 1. Add/multiply as integers,
 2. then *reduce modulo q* (remainder after division by q).

Notation: $a \bmod q = \underline{a \% q}$. $a = b \bmod q$ means $a \% q = b \% q$.

Example: $16 = 9 = 2 \bmod 7$.



Modular Arithmetic

- $\mathbb{Z}_q = \{\underline{0}, \dots, \underline{q-1}\}$ with modular addition, multiplication rules:
 1. Add/multiply as integers,
 2. then *reduce modulo q* (remainder after division by q).

Notation: $a \bmod q = \underline{a \% q}$. $a = b \bmod q$ means $a \% q = b \% q$.

Example: $16 = 9 = 2 \bmod 7$.

- Modular identities:

$$f + g = f \% q + g \% q \pmod{q}$$



Modular Arithmetic

- $\mathbb{Z}_q = \{\underline{0}, \dots, \underline{q-1}\}$ with modular addition, multiplication rules:
 1. Add/multiply as integers,
 2. then *reduce modulo q* (remainder after division by q).

Notation: $a \bmod q = \underline{a \% q}$. $a = b \bmod q$ means $a \% q = b \% q$.

Example: $16 = 9 = 2 \bmod 7$.

- Modular identities:

$$f + g = f \% q + g \% q \bmod q$$

$$f + g = d_f * q + f \% q + d_g * q + g \% q = f \% q + g \% q + (d_f + d_g) * q$$



Modular Arithmetic

- $\mathbb{Z}_q = \{\underline{0}, \dots, \underline{q-1}\}$ with modular addition, multiplication rules:
 1. Add/multiply as integers,
 2. then *reduce modulo q* (remainder after division by q).

Notation: $a \bmod q = \underline{a \% q}$. $a = b \bmod q$ means $a \% q = b \% q$.

Example: $16 = 9 = 2 \bmod 7$.

- Modular identities:

$$f + g = f \% q + g \% q \bmod q$$

$$f + g = d_f * q + f \% q + d_g * q + g \% q = f \% q + g \% q + (d_f + d_g) * q$$

$$f * g = (f \% q) * (g \% q) \bmod q$$



Modular Arithmetic

- $\mathbb{Z}_q = \{\underline{0}, \dots, \underline{q-1}\}$ with modular addition, multiplication rules:
 1. Add/multiply as integers,
 2. then *reduce modulo q* (remainder after division by q).

Notation: $a \bmod q = \underline{a \% q}$. $a = b \bmod q$ means $a \% q = b \% q$.

Example: $16 = 9 = 2 \bmod 7$.

- Modular identities:

$$f + g = f \% q + g \% q \bmod q$$

$$f + g = d_f * q + f \% q + d_g * q + g \% q = f \% q + g \% q + (d_f + d_g) * q$$

$$f * g = (f \% q) * (g \% q) \bmod q$$

$$f * g = (d_f * q + f \% q) * (d_g * q + g \% q) = (f \% q) * (g \% q) + (\dots) * q$$



Modular Arithmetic

- $\mathbb{Z}_q = \{\underline{0}, \dots, \underline{q-1}\}$ with modular addition, multiplication rules:
 1. Add/multiply as integers,
 2. then *reduce modulo q* (remainder after division by q).

Notation: $a \bmod q = \underline{a \% q}$. $a = b \bmod q$ means $a \% q = b \% q$.

Example: $16 = 9 = 2 \bmod 7$.

- Modular identities:

$$f + g = f \% q + g \% q \bmod q$$

$$f + g = d_f * q + f \% q + d_g * q + g \% q = f \% q + g \% q + (d_f + d_g) * q$$

$$f * g = (f \% q) * (g \% q) \bmod q$$

$$f * g = (d_f * q + f \% q) * (d_g * q + g \% q) = (f \% q) * (g \% q) + (\dots) * q$$

- \mathbb{Z}_q with addition and multiplication modulo q is a *ring*.



Largest Common Divisors

- The largest common divisor $\gcd(f, g)$ of f and g is the maximum d such that $d|f$ and $d|g$.



Largest Common Divisors

- The largest common divisor $\gcd(f, g)$ of f and g is the maximum d such that $d|f$ and $d|g$.

- $\text{EUCLID}(f, g)$

Input: $f > g > 1$

Output: $d, k, l \in \mathbb{Z}$ such that $kf + lg = \gcd(f, g) = d$.

$f_1 \leftarrow f; k_1 \leftarrow 1; l_1 \leftarrow 0$

$f_2 \leftarrow g; k_2 \leftarrow 0; l_2 \leftarrow 1$

C: *Each f_c is divisible by d . The last non-zero f_c is equal d .*

$c = 0;$

repeat

$c \leftarrow c + 1;$

$x \leftarrow \lfloor f_c / f_{c+1} \rfloor$

$f_{c+2} \leftarrow f_c \% f_{c+1};$

(* $f_{c+2} = f_c - xf_{c+1} \Rightarrow d|f_{c+2}$. *)

$k_{c+2} \leftarrow k_c - xk_{c+1}$

$l_{c+2} \leftarrow l_c - xl_{c+1}$

until $f_{c+2} = 0$

return $f_{c+1}, k_{c+1}, l_{c+1}$



Largest Common Divisors

- The largest common divisor $\gcd(f, g)$ of f and g is the maximum d such that $d|f$ and $d|g$.
- $\text{EUCLID}(f, g)$

Input: $f > g > 1$

Output: $d, k, l \in \mathbb{Z}$ such that $kf + lg = \gcd(f, g) = d$.

$f_1 \leftarrow f; k_1 \leftarrow 1; l_1 \leftarrow 0$

$f_2 \leftarrow g; k_2 \leftarrow 0; l_2 \leftarrow 1$

C: *Each f_c is divisible by d . The last non-zero f_c is equal d .*

$c = 0;$

repeat

$c \leftarrow c + 1;$

$x \leftarrow \lfloor f_c / f_{c+1} \rfloor$

$f_{c+2} \leftarrow f_c \% f_{c+1};$

(* $f_{c+2} = f_c - xf_{c+1} \Rightarrow d|f_{c+2}$. *)

$k_{c+2} \leftarrow k_c - xk_{c+1}$

$l_{c+2} \leftarrow l_c - xl_{c+1}$

until $f_{c+2} = 0$

return $f_{c+1}, k_{c+1}, l_{c+1}$

Example: $f = 132, g = 111$.



Largest Common Divisors

- The largest common divisor $\gcd(f, g)$ of f and g is the maximum d such that $d|f$ and $d|g$.

- $\text{EUCLID}(f, g)$

Input: $f > g > 1$

Output: $d, k, l \in \mathbb{Z}$ such that $kf + lg = \gcd(f, g) = d$.

$\Rightarrow f_1 \leftarrow f; k_1 \leftarrow 1; l_1 \leftarrow 0$

$\Rightarrow f_2 \leftarrow g; k_2 \leftarrow 0; l_2 \leftarrow 1$

C: *Each f_c is divisible by d . The last non-zero f_c is equal d .*

$c = 0;$

repeat

$c \leftarrow c + 1;$

$x \leftarrow \lfloor f_c / f_{c+1} \rfloor$

$f_{c+2} \leftarrow f_c \% f_{c+1};$

(* $f_{c+2} = f_c - xf_{c+1} \Rightarrow d|f_{c+2}$. *)

$k_{c+2} \leftarrow k_c - xk_{c+1}$

$l_{c+2} \leftarrow l_c - xl_{c+1}$

until $f_{c+2} = 0$

return $f_{c+1}, k_{c+1}, l_{c+1}$

Example: $f = 132, g = 111.$



Largest Common Divisors

- The largest common divisor $\gcd(f, g)$ of f and g is the maximum d such that $d|f$ and $d|g$.

- $\text{EUCLID}(f, g)$

Input: $f > g > 1$

Output: $d, k, l \in \mathbb{Z}$ such that $kf + lg = \gcd(f, g) = d$.

$\Rightarrow f_1 \leftarrow f; k_1 \leftarrow 1; l_1 \leftarrow 0$

$\Rightarrow f_2 \leftarrow g; k_2 \leftarrow 0; l_2 \leftarrow 1$

C: *Each f_c is divisible by d . The last non-zero f_c is equal d .*

$c = 0;$

repeat

$c \leftarrow c + 1;$

$x \leftarrow \lfloor f_c / f_{c+1} \rfloor$

$f_{c+2} \leftarrow f_c \% f_{c+1};$

(* $f_{c+2} = f_c - xf_{c+1} \Rightarrow d|f_{c+2}$. *)

$k_{c+2} \leftarrow k_c - xk_{c+1}$

$l_{c+2} \leftarrow l_c - xl_{c+1}$

until $f_{c+2} = 0$

return $f_{c+1}, k_{c+1}, l_{c+1}$

Example: $f = 132, g = 111.$

$$f_1 = 132 = 1*f + 0*g$$

$$f_2 = 111 = 0*f + 1*g$$



Largest Common Divisors

- The largest common divisor $\gcd(f, g)$ of f and g is the maximum d such that $d|f$ and $d|g$.
- $\text{EUCLID}(f, g)$

Input: $f > g > 1$

Output: $d, k, l \in \mathbb{Z}$ such that $kf + lg = \gcd(f, g) = d$.

$f_1 \leftarrow f; k_1 \leftarrow 1; l_1 \leftarrow 0$

$f_2 \leftarrow g; k_2 \leftarrow 0; l_2 \leftarrow 1$

C: *Each f_c is divisible by d . The last non-zero f_c is equal d .*

$c = 0;$

repeat

$c \leftarrow c + 1;$

$\Rightarrow x \leftarrow \lfloor f_c / f_{c+1} \rfloor$

$f_{c+2} \leftarrow f_c \% f_{c+1};$

(* $f_{c+2} = f_c - xf_{c+1} \Rightarrow d|f_{c+2}$. *)

$k_{c+2} \leftarrow k_c - xk_{c+1}$

$l_{c+2} \leftarrow l_c - xl_{c+1}$

until $f_{c+2} = 0$

return $f_{c+1}, k_{c+1}, l_{c+1}$

Example: $f = 132, g = 111.$

$$f_1 = 132 = 1*f + 0*g$$

$$f_2 = 111 = 0*f + 1*g$$



Largest Common Divisors

- The largest common divisor $\gcd(f, g)$ of f and g is the maximum d such that $d|f$ and $d|g$.
- $\text{EUCLID}(f, g)$

Input: $f > g > 1$

Output: $d, k, l \in \mathbb{Z}$ such that $kf + lg = \gcd(f, g) = d$.

$f_1 \leftarrow f; k_1 \leftarrow 1; l_1 \leftarrow 0$

$f_2 \leftarrow g; k_2 \leftarrow 0; l_2 \leftarrow 1$

C: *Each f_c is divisible by d . The last non-zero f_c is equal d .*

$c = 0;$

repeat

$c \leftarrow c + 1;$

$\Rightarrow x \leftarrow \lfloor f_c / f_{c+1} \rfloor$

$f_{c+2} \leftarrow f_c \% f_{c+1};$

(* $f_{c+2} = f_c - xf_{c+1} \Rightarrow d|f_{c+2}$. *)

$k_{c+2} \leftarrow k_c - xk_{c+1}$

$l_{c+2} \leftarrow l_c - xl_{c+1}$

until $f_{c+2} = 0$

return $f_{c+1}, k_{c+1}, l_{c+1}$

Example: $f = 132, g = 111.$

$$f_1 = 132 = 1*f + 0*g$$

$$f_2 = 111 = 0*f + 1*g$$

$x=1$



Largest Common Divisors

- The largest common divisor $\gcd(f, g)$ of f and g is the maximum d such that $d|f$ and $d|g$.
- $\text{EUCLID}(f, g)$

Input: $f > g > 1$

Output: $d, k, l \in \mathbb{Z}$ such that $kf + lg = \gcd(f, g) = d$.

$f_1 \leftarrow f; k_1 \leftarrow 1; l_1 \leftarrow 0$

$f_2 \leftarrow g; k_2 \leftarrow 0; l_2 \leftarrow 1$

C: *Each f_c is divisible by d . The last non-zero f_c is equal d .*

$c = 0;$

repeat

$c \leftarrow c + 1;$

$x \leftarrow \lfloor f_c / f_{c+1} \rfloor$

$\Rightarrow f_{c+2} \leftarrow f_c \% f_{c+1};$
($f_{c+2} = f_c - xf_{c+1} \Rightarrow d|f_{c+2}$. *)*

$k_{c+2} \leftarrow k_c - xk_{c+1}$

$l_{c+2} \leftarrow l_c - xl_{c+1}$

until $f_{c+2} = 0$

return $f_{c+1}, k_{c+1}, l_{c+1}$

Example: $f = 132, g = 111.$

$$f_1 = 132 = 1*f + 0*g$$

$$f_2 = 111 = 0*f + 1*g$$

$x=1$



Largest Common Divisors

- The largest common divisor $\gcd(f, g)$ of f and g is the maximum d such that $d|f$ and $d|g$.
- $\text{EUCLID}(f, g)$

Input: $f > g > 1$

Output: $d, k, l \in \mathbb{Z}$ such that $kf + lg = \gcd(f, g) = d$.

$f_1 \leftarrow f; k_1 \leftarrow 1; l_1 \leftarrow 0$

$f_2 \leftarrow g; k_2 \leftarrow 0; l_2 \leftarrow 1$

C: *Each f_c is divisible by d . The last non-zero f_c is equal d .*

$c = 0;$

repeat

$c \leftarrow c + 1;$

$x \leftarrow \lfloor f_c / f_{c+1} \rfloor$

$\Rightarrow f_{c+2} \leftarrow f_c \% f_{c+1};$

(* $f_{c+2} = f_c - xf_{c+1} \Rightarrow d|f_{c+2}$. *)

$k_{c+2} \leftarrow k_c - xk_{c+1}$

$l_{c+2} \leftarrow l_c - xl_{c+1}$

until $f_{c+2} = 0$

return $f_{c+1}, k_{c+1}, l_{c+1}$

Example: $f = 132, g = 111.$

$$f_1 = 132 = 1*f + 0*g$$

$$f_2 = 111 = 0*f + 1*g$$

$$f_3 = 21$$

$x=1$



Largest Common Divisors

- The largest common divisor $\gcd(f, g)$ of f and g is the maximum d such that $d|f$ and $d|g$.
- $\text{EUCLID}(f, g)$

Input: $f > g > 1$

Output: $d, k, l \in \mathbb{Z}$ such that $kf + lg = \gcd(f, g) = d$.

$f_1 \leftarrow f; k_1 \leftarrow 1; l_1 \leftarrow 0$

$f_2 \leftarrow g; k_2 \leftarrow 0; l_2 \leftarrow 1$

C: *Each f_c is divisible by d . The last non-zero f_c is equal d .*

$c = 0;$

repeat

$c \leftarrow c + 1;$

$x \leftarrow \lfloor f_c / f_{c+1} \rfloor$

$f_{c+2} \leftarrow f_c \% f_{c+1};$

(* $f_{c+2} = f_c - xf_{c+1} \Rightarrow d|f_{c+2}$. *)

$\Rightarrow k_{c+2} \leftarrow k_c - xk_{c+1}$

$\Rightarrow l_{c+2} \leftarrow l_c - xl_{c+1}$

until $f_{c+2} = 0$

return $f_{c+1}, k_{c+1}, l_{c+1}$

Example: $f = 132, g = 111.$

$$f_1 = 132 = 1*f + 0*g$$

$$f_2 = 111 = 0*f + 1*g$$

$$f_3 = 21$$

$x=1$



Largest Common Divisors

- The largest common divisor $\gcd(f, g)$ of f and g is the maximum d such that $d|f$ and $d|g$.
- $\text{EUCLID}(f, g)$

Input: $f > g > 1$

Output: $d, k, l \in \mathbb{Z}$ such that $kf + lg = \gcd(f, g) = d$.

$f_1 \leftarrow f; k_1 \leftarrow 1; l_1 \leftarrow 0$

$f_2 \leftarrow g; k_2 \leftarrow 0; l_2 \leftarrow 1$

C: *Each f_c is divisible by d . The last non-zero f_c is equal d .*

$c = 0;$

repeat

$c \leftarrow c + 1;$

$x \leftarrow \lfloor f_c / f_{c+1} \rfloor$

$f_{c+2} \leftarrow f_c \% f_{c+1};$

(* $f_{c+2} = f_c - xf_{c+1} \Rightarrow d|f_{c+2}$. *)

$\Rightarrow k_{c+2} \leftarrow k_c - xk_{c+1}$

$\Rightarrow l_{c+2} \leftarrow l_c - xl_{c+1}$

until $f_{c+2} = 0$

return $f_{c+1}, k_{c+1}, l_{c+1}$

Example: $f = 132, g = 111$.

$$f_1 = 132 = 1*f + 0*g$$

$$f_2 = 111 = 0*f + 1*g \quad x=1$$

$$f_3 = 21 = 1*f - 1*g$$



Largest Common Divisors

- The largest common divisor $\gcd(f, g)$ of f and g is the maximum d such that $d|f$ and $d|g$.
- $\text{EUCLID}(f, g)$

Input: $f > g > 1$

Output: $d, k, l \in \mathbb{Z}$ such that $kf + lg = \gcd(f, g) = d$.

$$f_1 \leftarrow f; k_1 \leftarrow 1; l_1 \leftarrow 0$$

$$f_2 \leftarrow g; k_2 \leftarrow 0; l_2 \leftarrow 1$$

C: Each f_c is divisible by d . The last non-zero f_c is equal d .

$$c = 0;$$

repeat

$$c \leftarrow c + 1;$$

$$\Rightarrow x \leftarrow \lfloor f_c / f_{c+1} \rfloor$$

$$f_{c+2} \leftarrow f_c \% f_{c+1};$$

$$(* f_{c+2} = f_c - xf_{c+1} \Rightarrow d|f_{c+2}. *)$$

$$k_{c+2} \leftarrow k_c - xk_{c+1}$$

$$l_{c+2} \leftarrow l_c - xl_{c+1}$$

until $f_{c+2} = 0$

return $f_{c+1}, k_{c+1}, l_{c+1}$

Example: $f = 132, g = 111$.

$$f_1 = 132 = 1*f + 0*g$$

$$f_2 = 111 = 0*f + 1*g \quad x=1$$

$$f_3 = 21 = 1*f - 1*g$$



Largest Common Divisors

- The largest common divisor $\gcd(f, g)$ of f and g is the maximum d such that $d|f$ and $d|g$.
- $\text{EUCLID}(f, g)$

Input: $f > g > 1$

Output: $d, k, l \in \mathbb{Z}$ such that $kf + lg = \gcd(f, g) = d$.

$$f_1 \leftarrow f; k_1 \leftarrow 1; l_1 \leftarrow 0$$

$$f_2 \leftarrow g; k_2 \leftarrow 0; l_2 \leftarrow 1$$

C: Each f_c is divisible by d . The last non-zero f_c is equal d .

$$c = 0;$$

repeat

$$c \leftarrow c + 1;$$

$$\Rightarrow x \leftarrow \lfloor f_c / f_{c+1} \rfloor$$

$$f_{c+2} \leftarrow f_c \% f_{c+1};$$

$$(* f_{c+2} = f_c - xf_{c+1} \Rightarrow d|f_{c+2}. *)$$

$$k_{c+2} \leftarrow k_c - xk_{c+1}$$

$$l_{c+2} \leftarrow l_c - xl_{c+1}$$

until $f_{c+2} = 0$

return $f_{c+1}, k_{c+1}, l_{c+1}$

Example: $f = 132, g = 111$.

$$f_1 = 132 = 1*f + 0*g$$

$$f_2 = 111 = 0*f + 1*g \quad x=1$$

$$f_3 = 21 = 1*f - 1*g \quad x=5$$



Largest Common Divisors

- The largest common divisor $\gcd(f, g)$ of f and g is the maximum d such that $d|f$ and $d|g$.
- $\text{EUCLID}(f, g)$

Input: $f > g > 1$

Output: $d, k, l \in \mathbb{Z}$ such that $kf + lg = \gcd(f, g) = d$.

$f_1 \leftarrow f; k_1 \leftarrow 1; l_1 \leftarrow 0$

$f_2 \leftarrow g; k_2 \leftarrow 0; l_2 \leftarrow 1$

C: *Each f_c is divisible by d . The last non-zero f_c is equal d .*

$c = 0;$

repeat

$c \leftarrow c + 1;$

$x \leftarrow \lfloor f_c / f_{c+1} \rfloor$

$\Rightarrow f_{c+2} \leftarrow f_c \% f_{c+1};$

(* $f_{c+2} = f_c - xf_{c+1} \Rightarrow d|f_{c+2}$. *)

$k_{c+2} \leftarrow k_c - xk_{c+1}$

$l_{c+2} \leftarrow l_c - xl_{c+1}$

until $f_{c+2} = 0$

return $f_{c+1}, k_{c+1}, l_{c+1}$

Example: $f = 132, g = 111.$

$$f_1 = 132 = 1*f + 0*g$$

$$f_2 = 111 = 0*f + 1*g \quad x=1$$

$$f_3 = 21 = 1*f - 1*g \quad x=5$$



Largest Common Divisors

- The largest common divisor $\gcd(f, g)$ of f and g is the maximum d such that $d|f$ and $d|g$.
- $\text{EUCLID}(f, g)$

Input: $f > g > 1$

Output: $d, k, l \in \mathbb{Z}$ such that $kf + lg = \gcd(f, g) = d$.

$f_1 \leftarrow f; k_1 \leftarrow 1; l_1 \leftarrow 0$

$f_2 \leftarrow g; k_2 \leftarrow 0; l_2 \leftarrow 1$

C: *Each f_c is divisible by d . The last non-zero f_c is equal d .*

$c = 0;$

repeat

$c \leftarrow c + 1;$

$x \leftarrow \lfloor f_c / f_{c+1} \rfloor$

$\Rightarrow f_{c+2} \leftarrow f_c \% f_{c+1};$
(* $f_{c+2} = f_c - xf_{c+1} \Rightarrow d|f_{c+2}$. *)

$k_{c+2} \leftarrow k_c - xk_{c+1}$

$l_{c+2} \leftarrow l_c - xl_{c+1}$

until $f_{c+2} = 0$

return $f_{c+1}, k_{c+1}, l_{c+1}$

Example: $f = 132, g = 111.$

$$f_1 = 132 = 1*f + 0*g$$

$$f_2 = 111 = 0*f + 1*g \quad x=1$$

$$f_3 = 21 = 1*f - 1*g \quad x=5$$

$$f_4 = 6$$



Largest Common Divisors

- The largest common divisor $\gcd(f, g)$ of f and g is the maximum d such that $d|f$ and $d|g$.
- $\text{EUCLID}(f, g)$

Input: $f > g > 1$

Output: $d, k, l \in \mathbb{Z}$ such that $kf + lg = \gcd(f, g) = d$.

$f_1 \leftarrow f; k_1 \leftarrow 1; l_1 \leftarrow 0$

$f_2 \leftarrow g; k_2 \leftarrow 0; l_2 \leftarrow 1$

C: *Each f_c is divisible by d . The last non-zero f_c is equal d .*

$c = 0;$

repeat

$c \leftarrow c + 1;$

$x \leftarrow \lfloor f_c / f_{c+1} \rfloor$

$f_{c+2} \leftarrow f_c \% f_{c+1};$

(* $f_{c+2} = f_c - xf_{c+1} \Rightarrow d|f_{c+2}$. *)

$\Rightarrow k_{c+2} \leftarrow k_c - xk_{c+1}$

$\Rightarrow l_{c+2} \leftarrow l_c - xl_{c+1}$

until $f_{c+2} = 0$

return $f_{c+1}, k_{c+1}, l_{c+1}$

Example: $f = 132, g = 111$.

$$f_1 = 132 = 1*f + 0*g$$

$$f_2 = 111 = 0*f + 1*g \quad x=1$$

$$f_3 = 21 = 1*f - 1*g \quad x=5$$

$$f_4 = 6$$

Largest Common Divisors

- The largest common divisor $\gcd(f, g)$ of f and g is the maximum d such that $d|f$ and $d|g$.
- $\text{EUCLID}(f, g)$

Input: $f > g > 1$

Output: $d, k, l \in \mathbb{Z}$ such that $kf + lg = \gcd(f, g) = d$.

$$f_1 \leftarrow f; k_1 \leftarrow 1; l_1 \leftarrow 0$$

$$f_2 \leftarrow g; k_2 \leftarrow 0; l_2 \leftarrow 1$$

C: *Each f_c is divisible by d . The last non-zero f_c is equal d .*

$$c = 0;$$

repeat

$$c \leftarrow c + 1;$$

$$x \leftarrow \lfloor f_c / f_{c+1} \rfloor$$

$$f_{c+2} \leftarrow f_c \% f_{c+1};$$

$$(* f_{c+2} = f_c - x f_{c+1} \Rightarrow d | f_{c+2}. *)$$

$$\Rightarrow k_{c+2} \leftarrow k_c - x k_{c+1}$$

$$\Rightarrow l_{c+2} \leftarrow l_c - x l_{c+1}$$

until $f_{c+2} = 0$

return $f_{c+1}, k_{c+1}, l_{c+1}$

Example: $f = 132, g = 111.$

$$f_1 = 132 = 1*f + 0*g$$

$$f_2 = 111 = 0*f + 1*g \quad x=1$$

$$f_3 = 21 = 1*f - 1*g \quad x=5$$

$$f_4 = 6 = -5*f + 6*g$$



Largest Common Divisors

- The largest common divisor $\gcd(f, g)$ of f and g is the maximum d such that $d|f$ and $d|g$.
- $\text{EUCLID}(f, g)$

Input: $f > g > 1$

Output: $d, k, l \in \mathbb{Z}$ such that $kf + lg = \gcd(f, g) = d$.

$$f_1 \leftarrow f; k_1 \leftarrow 1; l_1 \leftarrow 0$$

$$f_2 \leftarrow g; k_2 \leftarrow 0; l_2 \leftarrow 1$$

C: Each f_c is divisible by d . The last non-zero f_c is equal d .

$$c = 0;$$

repeat

$$c \leftarrow c + 1;$$

$$\Rightarrow x \leftarrow \lfloor f_c / f_{c+1} \rfloor$$

$$f_{c+2} \leftarrow f_c \% f_{c+1};$$

$$(* f_{c+2} = f_c - x f_{c+1} \Rightarrow d | f_{c+2}. *)$$

$$k_{c+2} \leftarrow k_c - x k_{c+1}$$

$$l_{c+2} \leftarrow l_c - x l_{c+1}$$

until $f_{c+2} = 0$

return $f_{c+1}, k_{c+1}, l_{c+1}$

Example: $f = 132, g = 111$.

$$f_1 = 132 = 1*f + 0*g$$

$$f_2 = 111 = 0*f + 1*g \quad x=1$$

$$f_3 = 21 = 1*f - 1*g \quad x=5$$

$$f_4 = 6 = -5*f + 6*g$$



Largest Common Divisors

- The largest common divisor $\gcd(f, g)$ of f and g is the maximum d such that $d|f$ and $d|g$.
- $\text{EUCLID}(f, g)$

Input: $f > g > 1$

Output: $d, k, l \in \mathbb{Z}$ such that $kf + lg = \gcd(f, g) = d$.

$f_1 \leftarrow f; k_1 \leftarrow 1; l_1 \leftarrow 0$

$f_2 \leftarrow g; k_2 \leftarrow 0; l_2 \leftarrow 1$

C: *Each f_c is divisible by d . The last non-zero f_c is equal d .*

$c = 0;$

repeat

$c \leftarrow c + 1;$

$\Rightarrow x \leftarrow \lfloor f_c / f_{c+1} \rfloor$

$f_{c+2} \leftarrow f_c \% f_{c+1};$

(* $f_{c+2} = f_c - xf_{c+1} \Rightarrow d|f_{c+2}$. *)

$k_{c+2} \leftarrow k_c - xk_{c+1}$

$l_{c+2} \leftarrow l_c - xl_{c+1}$

until $f_{c+2} = 0$

return $f_{c+1}, k_{c+1}, l_{c+1}$

Example: $f = 132, g = 111.$

$f_1 = 132$	$=$	$1*f + 0*g$	
$f_2 = 111$	$=$	$0*f + 1*g$	$x=1$
$f_3 = 21$	$=$	$1*f - 1*g$	$x=5$
$f_4 = 6$	$=$	$-5*f + 6*g$	$x=3$



Largest Common Divisors

- The largest common divisor $\gcd(f, g)$ of f and g is the maximum d such that $d|f$ and $d|g$.
- $\text{EUCLID}(f, g)$

Input: $f > g > 1$

Output: $d, k, l \in \mathbb{Z}$ such that $kf + lg = \gcd(f, g) = d$.

$f_1 \leftarrow f; k_1 \leftarrow 1; l_1 \leftarrow 0$

$f_2 \leftarrow g; k_2 \leftarrow 0; l_2 \leftarrow 1$

C: *Each f_c is divisible by d . The last non-zero f_c is equal d .*

$c = 0;$

repeat

$c \leftarrow c + 1;$

$x \leftarrow \lfloor f_c / f_{c+1} \rfloor$

$\Rightarrow f_{c+2} \leftarrow f_c \% f_{c+1};$
($f_{c+2} = f_c - xf_{c+1} \Rightarrow d|f_{c+2}$. *)*

$k_{c+2} \leftarrow k_c - xk_{c+1}$

$l_{c+2} \leftarrow l_c - xl_{c+1}$

until $f_{c+2} = 0$

return $f_{c+1}, k_{c+1}, l_{c+1}$

Example: $f = 132, g = 111.$

$f_1 = 132$	$=$	$1*f + 0*g$	
$f_2 = 111$	$=$	$0*f + 1*g$	$x=1$
$f_3 = 21$	$=$	$1*f - 1*g$	$x=5$
$f_4 = 6$	$=$	$-5*f + 6*g$	$x=3$



Largest Common Divisors

- The largest common divisor $\gcd(f, g)$ of f and g is the maximum d such that $d|f$ and $d|g$.
- $\text{EUCLID}(f, g)$

Input: $f > g > 1$

Output: $d, k, l \in \mathbb{Z}$ such that $kf + lg = \gcd(f, g) = d$.

$f_1 \leftarrow f; k_1 \leftarrow 1; l_1 \leftarrow 0$

$f_2 \leftarrow g; k_2 \leftarrow 0; l_2 \leftarrow 1$

C: *Each f_c is divisible by d . The last non-zero f_c is equal d .*

$c = 0;$

repeat

$c \leftarrow c + 1;$

$x \leftarrow \lfloor f_c / f_{c+1} \rfloor$

$\Rightarrow f_{c+2} \leftarrow f_c \% f_{c+1};$
(* $f_{c+2} = f_c - xf_{c+1} \Rightarrow d|f_{c+2}$. *)

$k_{c+2} \leftarrow k_c - xk_{c+1}$

$l_{c+2} \leftarrow l_c - xl_{c+1}$

until $f_{c+2} = 0$

return $f_{c+1}, k_{c+1}, l_{c+1}$

Example: $f = 132, g = 111.$

$f_1 = 132$	$=$	$1*f + 0*g$	
$f_2 = 111$	$=$	$0*f + 1*g$	$x=1$
$f_3 = 21$	$=$	$1*f - 1*g$	$x=5$
$f_4 = 6$	$=$	$-5*f + 6*g$	$x=3$
$f_5 = 3$			



Largest Common Divisors

- The largest common divisor $\gcd(f, g)$ of f and g is the maximum d such that $d|f$ and $d|g$.
- $\text{EUCLID}(f, g)$

Input: $f > g > 1$

Output: $d, k, l \in \mathbb{Z}$ such that $kf + lg = \gcd(f, g) = d$.

$f_1 \leftarrow f; k_1 \leftarrow 1; l_1 \leftarrow 0$

$f_2 \leftarrow g; k_2 \leftarrow 0; l_2 \leftarrow 1$

C: *Each f_c is divisible by d . The last non-zero f_c is equal d .*

$c = 0;$

repeat

$c \leftarrow c + 1;$

$x \leftarrow \lfloor f_c / f_{c+1} \rfloor$

$f_{c+2} \leftarrow f_c \% f_{c+1};$

(* $f_{c+2} = f_c - xf_{c+1} \Rightarrow d|f_{c+2}$. *)

$\Rightarrow k_{c+2} \leftarrow k_c - xk_{c+1}$

$\Rightarrow l_{c+2} \leftarrow l_c - xl_{c+1}$

until $f_{c+2} = 0$

return $f_{c+1}, k_{c+1}, l_{c+1}$

Example: $f = 132, g = 111$.

$$f_1 = 132 = 1*f + 0*g$$

$$f_2 = 111 = 0*f + 1*g \quad x=1$$

$$f_3 = 21 = 1*f - 1*g \quad x=5$$

$$f_4 = 6 = -5*f + 6*g \quad x=3$$

$$f_5 = 3$$

Largest Common Divisors

- The largest common divisor $\gcd(f, g)$ of f and g is the maximum d such that $d|f$ and $d|g$.
- $\text{EUCLID}(f, g)$

Input: $f > g > 1$

Output: $d, k, l \in \mathbb{Z}$ such that $kf + lg = \gcd(f, g) = d$.

$f_1 \leftarrow f; k_1 \leftarrow 1; l_1 \leftarrow 0$

$f_2 \leftarrow g; k_2 \leftarrow 0; l_2 \leftarrow 1$

C: *Each f_c is divisible by d . The last non-zero f_c is equal d .*

$c = 0;$

repeat

$c \leftarrow c + 1;$

$x \leftarrow \lfloor f_c / f_{c+1} \rfloor$

$f_{c+2} \leftarrow f_c \% f_{c+1};$

(* $f_{c+2} = f_c - xf_{c+1} \Rightarrow d|f_{c+2}$. *)

$\Rightarrow k_{c+2} \leftarrow k_c - xk_{c+1}$

$\Rightarrow l_{c+2} \leftarrow l_c - xl_{c+1}$

until $f_{c+2} = 0$

return $f_{c+1}, k_{c+1}, l_{c+1}$

Example: $f = 132, g = 111$.

$$f_1 = 132 = 1*f + 0*g$$

$$f_2 = 111 = 0*f + 1*g \quad x=1$$

$$f_3 = 21 = 1*f - 1*g \quad x=5$$

$$f_4 = 6 = -5*f + 6*g \quad x=3$$

$$f_5 = 3 = 16*f - 19*g$$



Largest Common Divisors

- The largest common divisor $\gcd(f, g)$ of f and g is the maximum d such that $d|f$ and $d|g$.
- $\text{EUCLID}(f, g)$

Input: $f > g > 1$

Output: $d, k, l \in \mathbb{Z}$ such that $kf + lg = \gcd(f, g) = d$.

$$f_1 \leftarrow f; k_1 \leftarrow 1; l_1 \leftarrow 0$$

$$f_2 \leftarrow g; k_2 \leftarrow 0; l_2 \leftarrow 1$$

C: *Each f_c is divisible by d . The last non-zero f_c is equal d .*

$$c = 0;$$

repeat

$$c \leftarrow c + 1;$$

$$\Rightarrow x \leftarrow \lfloor f_c / f_{c+1} \rfloor$$

$$f_{c+2} \leftarrow f_c \% f_{c+1};$$

$$(* f_{c+2} = f_c - x f_{c+1} \Rightarrow d | f_{c+2}. *)$$

$$k_{c+2} \leftarrow k_c - x k_{c+1}$$

$$l_{c+2} \leftarrow l_c - x l_{c+1}$$

until $f_{c+2} = 0$

return $f_{c+1}, k_{c+1}, l_{c+1}$

Example: $f = 132, g = 111.$

$$f_1 = 132 = 1*f + 0*g$$

$$f_2 = 111 = 0*f + 1*g \quad x=1$$

$$f_3 = 21 = 1*f - 1*g \quad x=5$$

$$f_4 = 6 = -5*f + 6*g \quad x=3$$

$$f_5 = 3 = 16*f - 19*g$$



Largest Common Divisors

- The largest common divisor $\gcd(f, g)$ of f and g is the maximum d such that $d|f$ and $d|g$.
- $\text{EUCLID}(f, g)$

Input: $f > g > 1$

Output: $d, k, l \in \mathbb{Z}$ such that $kf + lg = \gcd(f, g) = d$.

$$f_1 \leftarrow f; k_1 \leftarrow 1; l_1 \leftarrow 0$$

$$f_2 \leftarrow g; k_2 \leftarrow 0; l_2 \leftarrow 1$$

C: *Each f_c is divisible by d . The last non-zero f_c is equal d .*

$$c = 0;$$

repeat

$$c \leftarrow c + 1;$$

$$\Rightarrow x \leftarrow \lfloor f_c / f_{c+1} \rfloor$$

$$f_{c+2} \leftarrow f_c \% f_{c+1};$$

$$(* f_{c+2} = f_c - x f_{c+1} \Rightarrow d | f_{c+2}. *)$$

$$k_{c+2} \leftarrow k_c - x k_{c+1}$$

$$l_{c+2} \leftarrow l_c - x l_{c+1}$$

until $f_{c+2} = 0$

return $f_{c+1}, k_{c+1}, l_{c+1}$

Example: $f = 132, g = 111.$

$f_1 = 132$	$=$	$1*f + 0*g$	
$f_2 = 111$	$=$	$0*f + 1*g$	$x=1$
$f_3 = 21$	$=$	$1*f - 1*g$	$x=5$
$f_4 = 6$	$=$	$-5*f + 6*g$	$x=3$
$f_5 = 3$	$=$	$16*f - 19*g$	$x=2$



Largest Common Divisors

- The largest common divisor $\gcd(f, g)$ of f and g is the maximum d such that $d|f$ and $d|g$.
- $\text{EUCLID}(f, g)$

Input: $f > g > 1$

Output: $d, k, l \in \mathbb{Z}$ such that $kf + lg = \gcd(f, g) = d$.

$$f_1 \leftarrow f; k_1 \leftarrow 1; l_1 \leftarrow 0$$

$$f_2 \leftarrow g; k_2 \leftarrow 0; l_2 \leftarrow 1$$

C: Each f_c is divisible by d . The last non-zero f_c is equal d .

$$c = 0;$$

repeat

$$c \leftarrow c + 1;$$

$$x \leftarrow \lfloor f_c / f_{c+1} \rfloor$$

$$\Rightarrow f_{c+2} \leftarrow f_c \% f_{c+1};$$

(* $f_{c+2} = f_c - xf_{c+1} \Rightarrow d|f_{c+2}$. *)

$$k_{c+2} \leftarrow k_c - xk_{c+1}$$

$$l_{c+2} \leftarrow l_c - xl_{c+1}$$

until $f_{c+2} = 0$

return $f_{c+1}, k_{c+1}, l_{c+1}$

Example: $f = 132, g = 111$.

$f_1 = 132$	$=$	$1*f + 0*g$	
$f_2 = 111$	$=$	$0*f + 1*g$	$x=1$
$f_3 = 21$	$=$	$1*f - 1*g$	$x=5$
$f_4 = 6$	$=$	$-5*f + 6*g$	$x=3$
$f_5 = 3$	$=$	$16*f - 19*g$	$x=2$



Largest Common Divisors

- The largest common divisor $\gcd(f, g)$ of f and g is the maximum d such that $d|f$ and $d|g$.
- $\text{EUCLID}(f, g)$

Input: $f > g > 1$

Output: $d, k, l \in \mathbb{Z}$ such that $kf + lg = \gcd(f, g) = d$.

$$f_1 \leftarrow f; k_1 \leftarrow 1; l_1 \leftarrow 0$$

$$f_2 \leftarrow g; k_2 \leftarrow 0; l_2 \leftarrow 1$$

C: Each f_c is divisible by d . The last non-zero f_c is equal d .

$$c = 0;$$

repeat

$$c \leftarrow c + 1;$$

$$x \leftarrow \lfloor f_c / f_{c+1} \rfloor$$

$$\Rightarrow f_{c+2} \leftarrow f_c \% f_{c+1};$$

(* $f_{c+2} = f_c - xf_{c+1} \Rightarrow d|f_{c+2}$. *)

$$k_{c+2} \leftarrow k_c - xk_{c+1}$$

$$l_{c+2} \leftarrow l_c - xl_{c+1}$$

until $f_{c+2} = 0$

return $f_{c+1}, k_{c+1}, l_{c+1}$

Example: $f = 132, g = 111$.

$f_1 = 132$	$=$	$1*f + 0*g$	
$f_2 = 111$	$=$	$0*f + 1*g$	$x=1$
$f_3 = 21$	$=$	$1*f - 1*g$	$x=5$
$f_4 = 6$	$=$	$-5*f + 6*g$	$x=3$
$f_5 = 3$	$=$	$16*f - 19*g$	$x=2$
$f_6 = 0$			



Largest Common Divisors

- The largest common divisor $\gcd(f, g)$ of f and g is the maximum d such that $d|f$ and $d|g$.
- $\text{EUCLID}(f, g)$

Input: $f > g > 1$

Output: $d, k, l \in \mathbb{Z}$ such that $kf + lg = \gcd(f, g) = d$.

$f_1 \leftarrow f; k_1 \leftarrow 1; l_1 \leftarrow 0$

$f_2 \leftarrow g; k_2 \leftarrow 0; l_2 \leftarrow 1$

C: *Each f_c is divisible by d . The last non-zero f_c is equal d .*

$c = 0;$

repeat

$c \leftarrow c + 1;$

$x \leftarrow \lfloor f_c / f_{c+1} \rfloor$

$f_{c+2} \leftarrow f_c \% f_{c+1};$

(* $f_{c+2} = f_c - xf_{c+1} \Rightarrow d|f_{c+2}$. *)

$k_{c+2} \leftarrow k_c - xk_{c+1}$

$l_{c+2} \leftarrow l_c - xl_{c+1}$

until $f_{c+2} = 0$

\Rightarrow **return** $f_{c+1}, k_{c+1}, l_{c+1}$

Example: $f = 132, g = 111.$

$f_1 = 132$	$=$	$1*f + 0*g$	
$f_2 = 111$	$=$	$0*f + 1*g$	$x=1$
$f_3 = 21$	$=$	$1*f - 1*g$	$x=5$
$f_4 = 6$	$=$	$-5*f + 6*g$	$x=3$
$f_5 = 3$	$=$	$16*f - 19*g$	$x=2$
$f_6 = 0$			



The Structure of Z_q

- a is *relatively prime* to q , $(a, q) = 1$, if $\gcd(a, q) = 1$.



The Structure of Z_q

- a is *relatively prime* to q , $(a, q) = 1$, if $\gcd(a, q) = 1$.
- If $(a, q) = 1$, then for some b , $ba \equiv 1 \pmod{q}$.



The Structure of Z_q

- a is *relatively prime* to q , $(a, q) = 1$, if $\gcd(a, q) = 1$.
 - If $(a, q) = 1$, then for some b , $ba \equiv 1 \pmod{q}$.
- Proof.** Choose k, l , such that $ka + lq = 1$. Then $b = k \pmod{q}$.



The Structure of Z_q

- a is *relatively prime* to q , $(a, q) = 1$, if $\gcd(a, q) = 1$.
- If $(a, q) = 1$, then for some b , $ba \equiv 1 \pmod{q}$.
Proof. Choose k, l , such that $ka + lq = 1$. Then $b = k \pmod{q}$.
 - a is invertible \pmod{q} , so write $b = a^{-1} \pmod{q}$.



The Structure of Z_q

- a is *relatively prime* to q , $(a, q) = 1$, if $\gcd(a, q) = 1$.
 - If $(a, q) = 1$, then for some b , $ba \equiv 1 \pmod{q}$.
Proof. Choose k, l , such that $ka + lq = 1$. Then $b = k \pmod{q}$.
 - a is invertible \pmod{q} , so write $b = a^{-1} \pmod{q}$.
- Example:** $q = 21$, $a = 5$. Then $17 * 5 = 85 = 1 \pmod{21}$.



The Structure of Z_q

- a is *relatively prime* to q , $(a, q) = 1$, if $\gcd(a, q) = 1$.
- If $(a, q) = 1$, then for some b , $ba \equiv 1 \pmod{q}$.
Proof. Choose k, l , such that $ka + lq = 1$. Then $b = k \pmod{q}$.
 - a is invertible \pmod{q} , so write $b = a^{-1} \pmod{q}$.
- **Example:** $q = 21$, $a = 5$. Then $17 * 5 = 85 = 1 \pmod{21}$.
- $(a, q) = 1$ and $(b, q) = 1$ implies $(a * b, q) = 1$.



The Structure of Z_q

- a is *relatively prime* to q , $(a, q) = 1$, if $\gcd(a, q) = 1$.
- If $(a, q) = 1$, then for some b , $ba \equiv 1 \pmod{q}$.
Proof. Choose k, l , such that $ka + lq = 1$. Then $b = k \pmod{q}$.
 - a is invertible \pmod{q} , so write $b = a^{-1} \pmod{q}$.
- **Example:** $q = 21$, $a = 5$. Then $17 * 5 = 85 \equiv 1 \pmod{21}$.
- $(a, q) = 1$ and $(b, q) = 1$ implies $(a * b, q) = 1$.
- The set $\mathbb{Z}_q^* = \{a \mid (a, q) = 1\}$ is a group under multiplication.



The Structure of Z_q

- a is *relatively prime* to q , $(a, q) = 1$, if $\gcd(a, q) = 1$.
- If $(a, q) = 1$, then for some b , $ba \equiv 1 \pmod{q}$.
Proof. Choose k, l , such that $ka + lq = 1$. Then $b = k \pmod{q}$.
 - a is invertible \pmod{q} , so write $b = a^{-1} \pmod{q}$.
- **Example:** $q = 21$, $a = 5$. Then $17 * 5 = 85 \equiv 1 \pmod{21}$.
- $(a, q) = 1$ and $(b, q) = 1$ implies $(a * b, q) = 1$.
- The set $\mathbb{Z}_q^* = \{a \mid (a, q) = 1\}$ is a group under multiplication.
- Euler ϕ function: $\phi(q) = |\mathbb{Z}_q^*|$.



The Structure of Z_q

- a is *relatively prime* to q , $(a, q) = 1$, if $\gcd(a, q) = 1$.
- If $(a, q) = 1$, then for some b , $ba \equiv 1 \pmod{q}$.
Proof. Choose k, l , such that $ka + lq = 1$. Then $b = k \pmod{q}$.
 - a is invertible \pmod{q} , so write $b = a^{-1} \pmod{q}$.**Example:** $q = 21$, $a = 5$. Then $17 * 5 = 85 \equiv 1 \pmod{21}$.
- $(a, q) = 1$ and $(b, q) = 1$ implies $(a * b, q) = 1$.
- The set $\mathbb{Z}_q^* = \{a \mid (a, q) = 1\}$ is a group under multiplication.
- Euler ϕ function: $\phi(q) = |\mathbb{Z}_q^*|$.
Examples: - $\phi(6) = 2$ because 1, 5 are relatively prime to 6.



The Structure of Z_q

- a is *relatively prime* to q , $(a, q) = 1$, if $\gcd(a, q) = 1$.
- If $(a, q) = 1$, then for some b , $ba \equiv 1 \pmod{q}$.
Proof. Choose k, l , such that $ka + lq = 1$. Then $b = k \pmod{q}$.
 - a is invertible \pmod{q} , so write $b = a^{-1} \pmod{q}$.
- **Example:** $q = 21$, $a = 5$. Then $17 * 5 = 85 \equiv 1 \pmod{21}$.
- $(a, q) = 1$ and $(b, q) = 1$ implies $(a * b, q) = 1$.
- The set $\mathbb{Z}_q^* = \{a \mid (a, q) = 1\}$ is a group under multiplication.
- Euler ϕ function: $\phi(q) = |\mathbb{Z}_q^*|$.
Examples:
 - $\phi(6) = 2$ because 1, 5 are relatively prime to 6.
 - For p a prime, $\phi(p) = p - 1$.



The Structure of Z_q

- a is *relatively prime* to q , $(a, q) = 1$, if $\gcd(a, q) = 1$.
- If $(a, q) = 1$, then for some b , $ba \equiv 1 \pmod{q}$.
Proof. Choose k, l , such that $ka + lq = 1$. Then $b = k \pmod{q}$.
 - a is invertible \pmod{q} , so write $b = a^{-1} \pmod{q}$.**Example:** $q = 21$, $a = 5$. Then $17 * 5 = 85 \equiv 1 \pmod{21}$.
- $(a, q) = 1$ and $(b, q) = 1$ implies $(a * b, q) = 1$.
- The set $\mathbb{Z}_q^* = \{a \mid (a, q) = 1\}$ is a group under multiplication.
- Euler ϕ function: $\phi(q) = |\mathbb{Z}_q^*|$.
Examples:
 - $\phi(6) = 2$ because 1, 5 are relatively prime to 6.
 - For p a prime, $\phi(p) = p - 1$.
- For $a \in \mathbb{Z}_q^*$, $a^{\phi(q)} \equiv 1 \pmod{q}$.

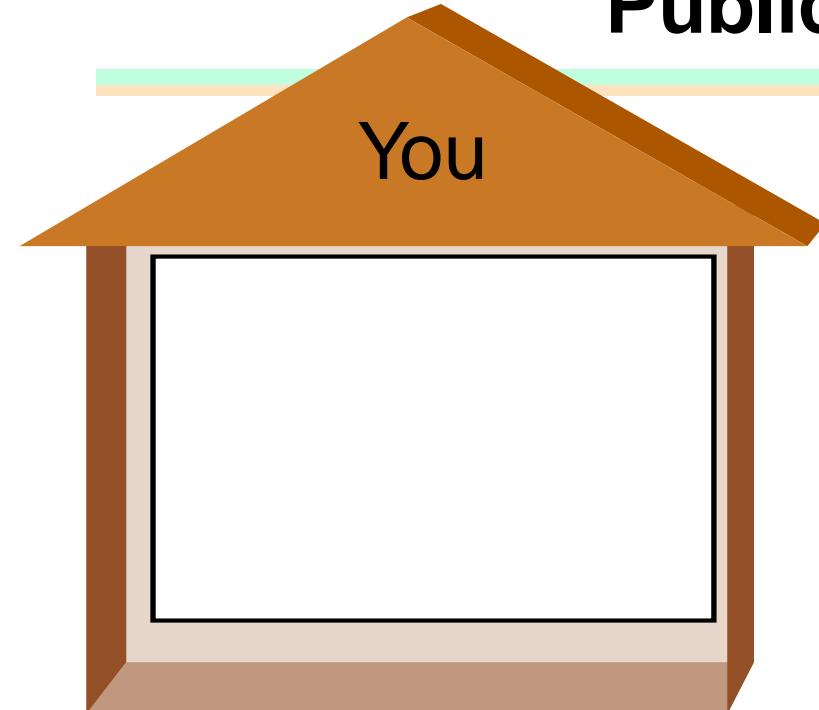


The Structure of Z_q

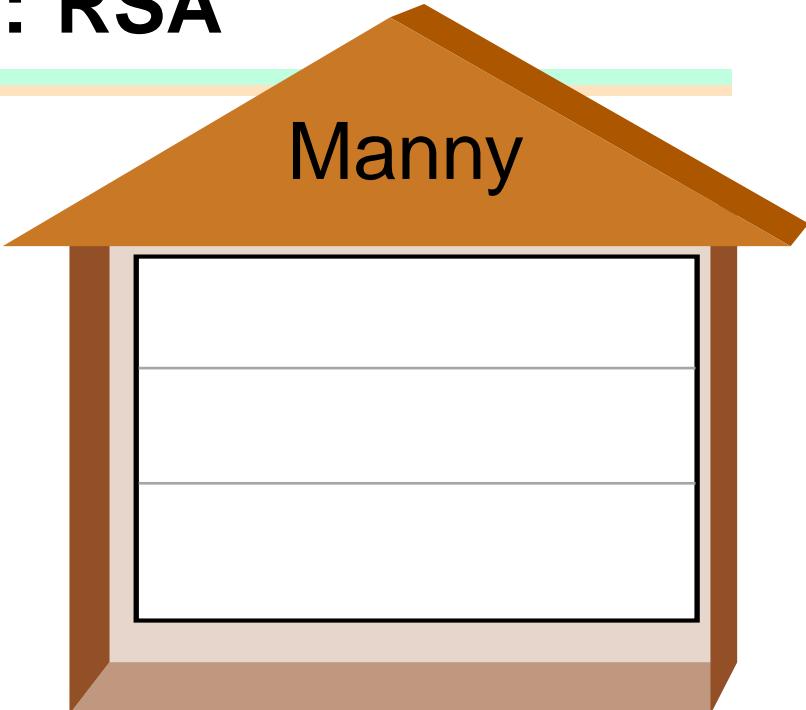
- a is *relatively prime* to q , $(a, q) = 1$, if $\gcd(a, q) = 1$.
- If $(a, q) = 1$, then for some b , $ba \equiv 1 \pmod{q}$.
Proof. Choose k, l , such that $ka + lq = 1$. Then $b = k \pmod{q}$.
 - a is invertible \pmod{q} , so write $b = a^{-1} \pmod{q}$.**Example:** $q = 21$, $a = 5$. Then $17 * 5 = 85 \equiv 1 \pmod{21}$.
- $(a, q) = 1$ and $(b, q) = 1$ implies $(a * b, q) = 1$.
- The set $\mathbb{Z}_q^* = \{a \mid (a, q) = 1\}$ is a group under multiplication.
- Euler ϕ function: $\phi(q) = |\mathbb{Z}_q^*|$.
Examples:
 - $\phi(6) = 2$ because 1, 5 are relatively prime to 6.
 - For p a prime, $\phi(p) = p - 1$.
- For $a \in \mathbb{Z}_q^*$, $a^{\phi(q)} \equiv 1 \pmod{q}$.
Proof. Basic group theory.



Public Key Crypto: RSA

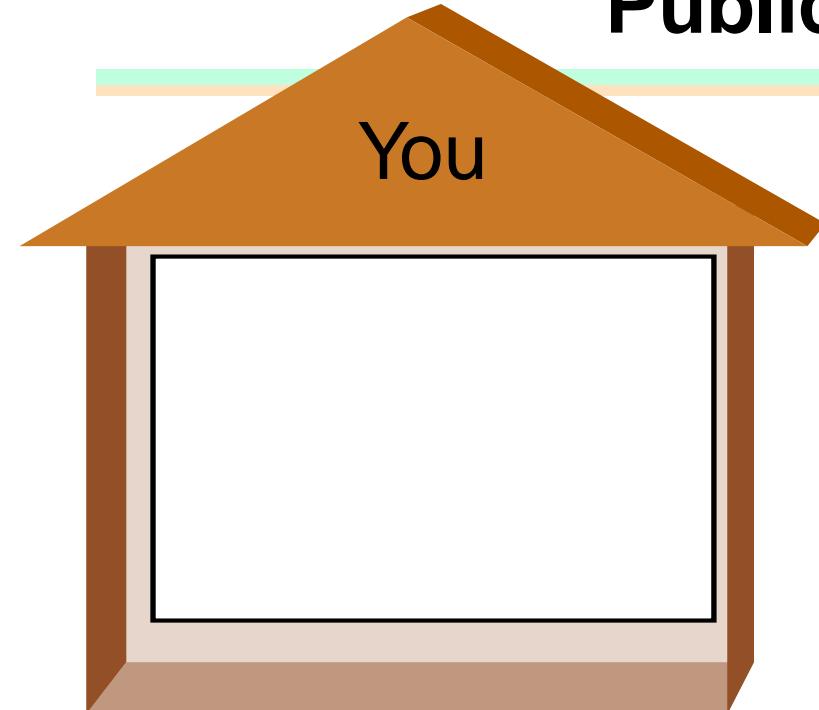


You

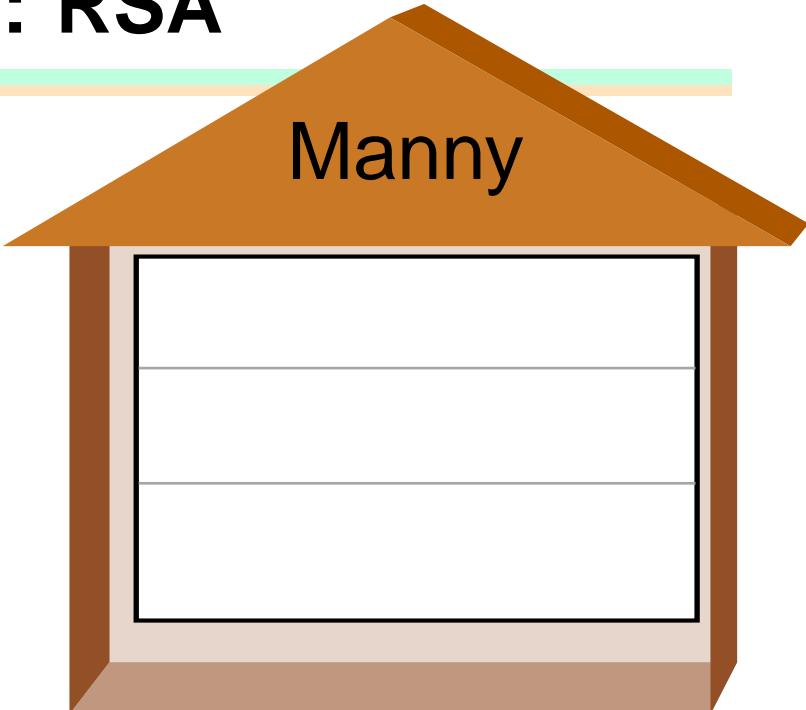


Manny

Public Key Crypto: RSA



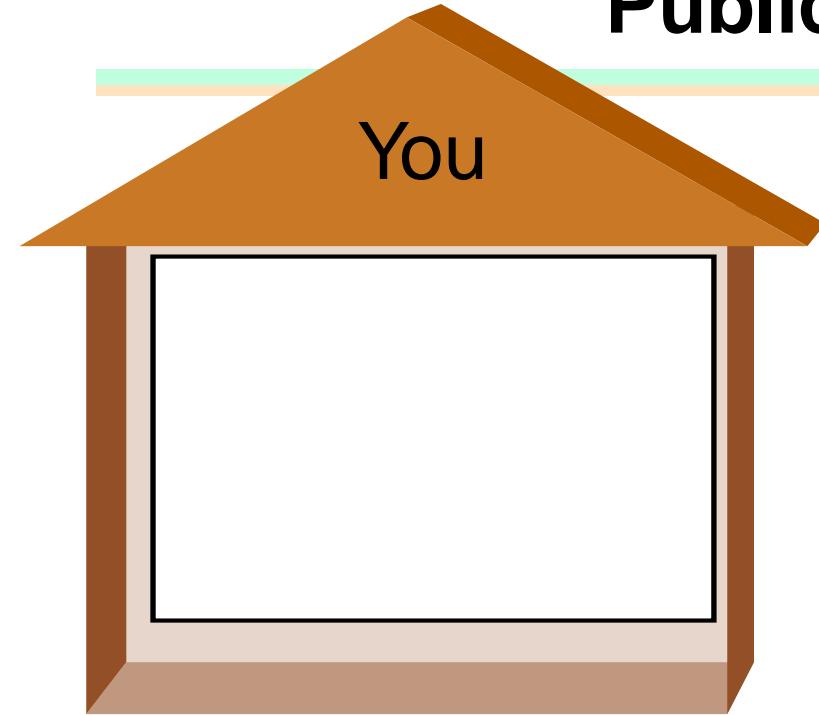
You



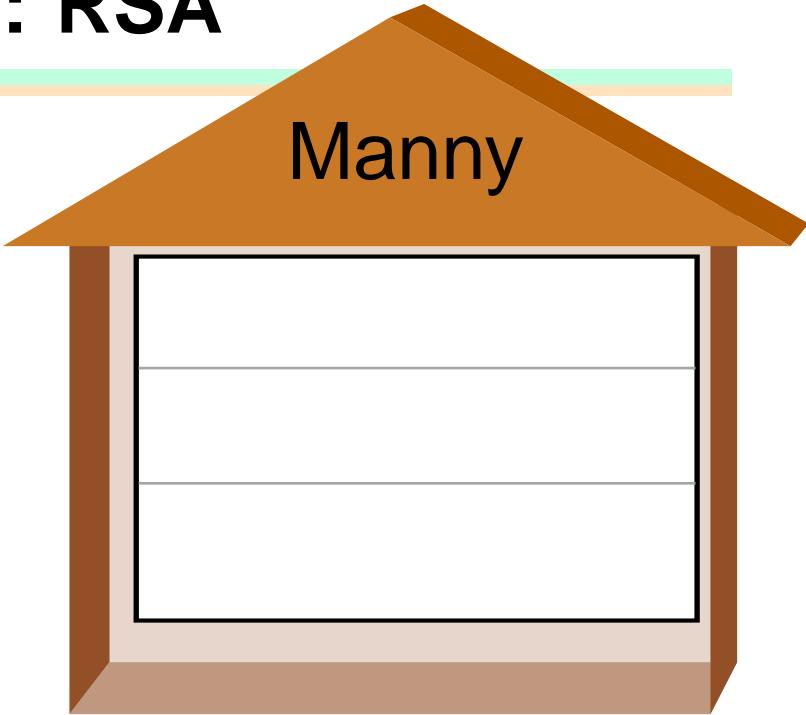
Manny

- Pick two large primes p, q .

Public Key Crypto: RSA



You

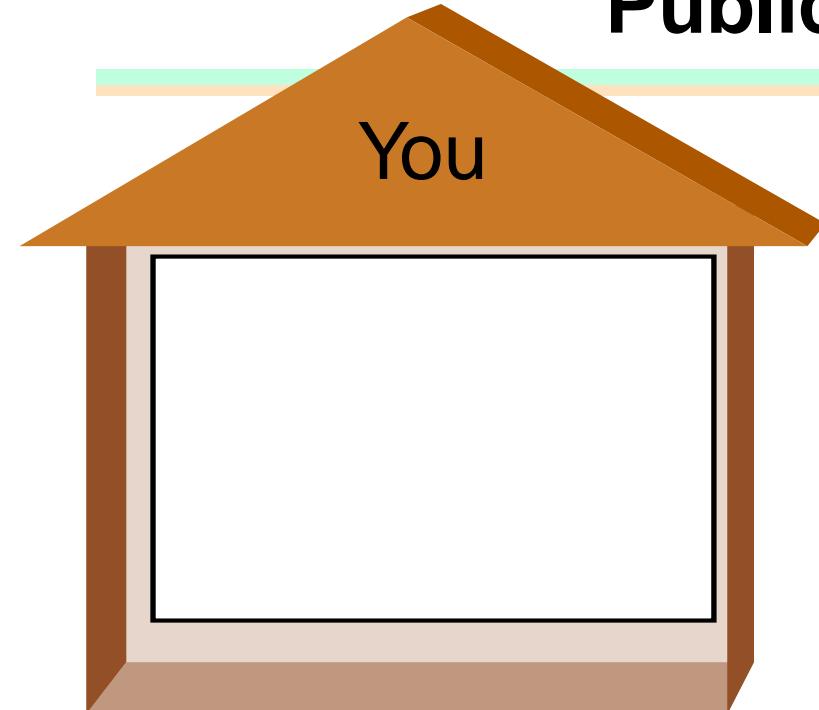


Manny

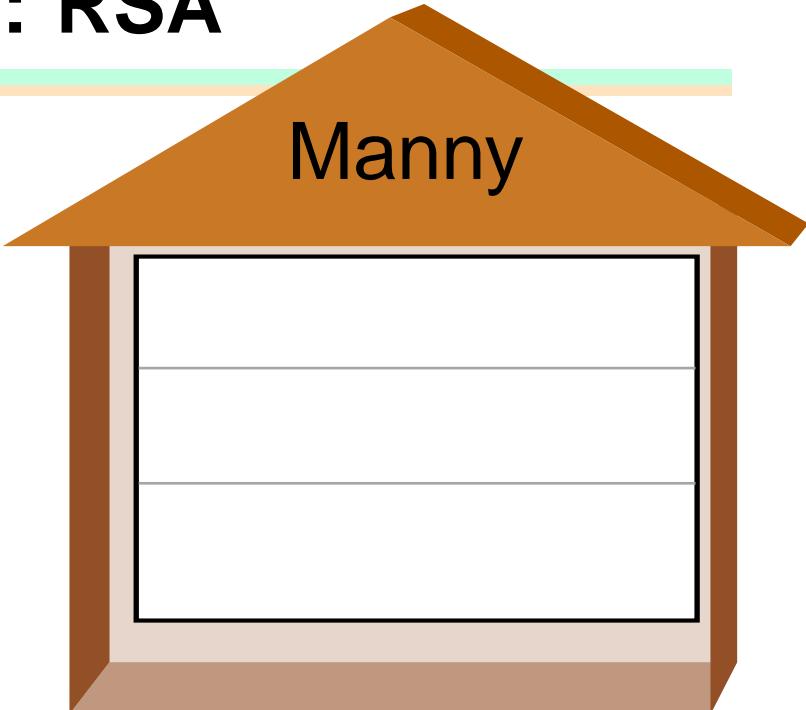
- Pick two large primes p, q .

$$p=3, q=11.$$

Public Key Crypto: RSA



You

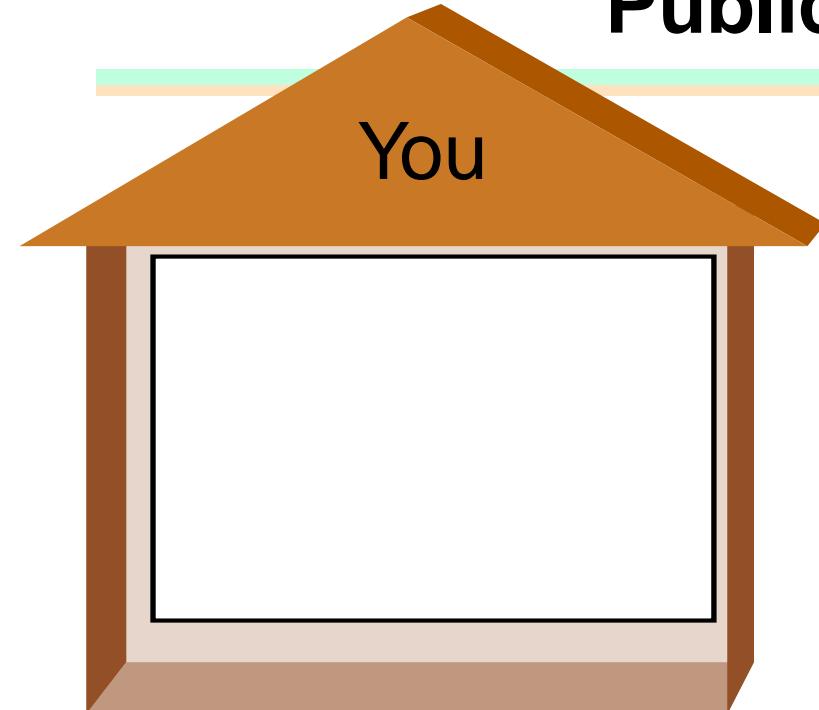


Manny

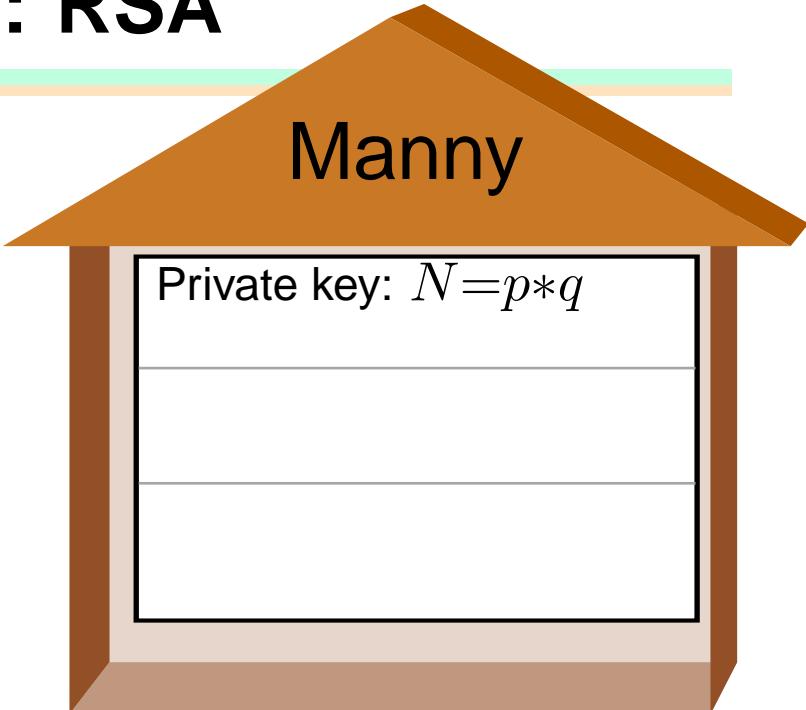
- Pick two large primes p, q .
- $N = p * q$

$$p=3, q=11.$$

Public Key Crypto: RSA



You



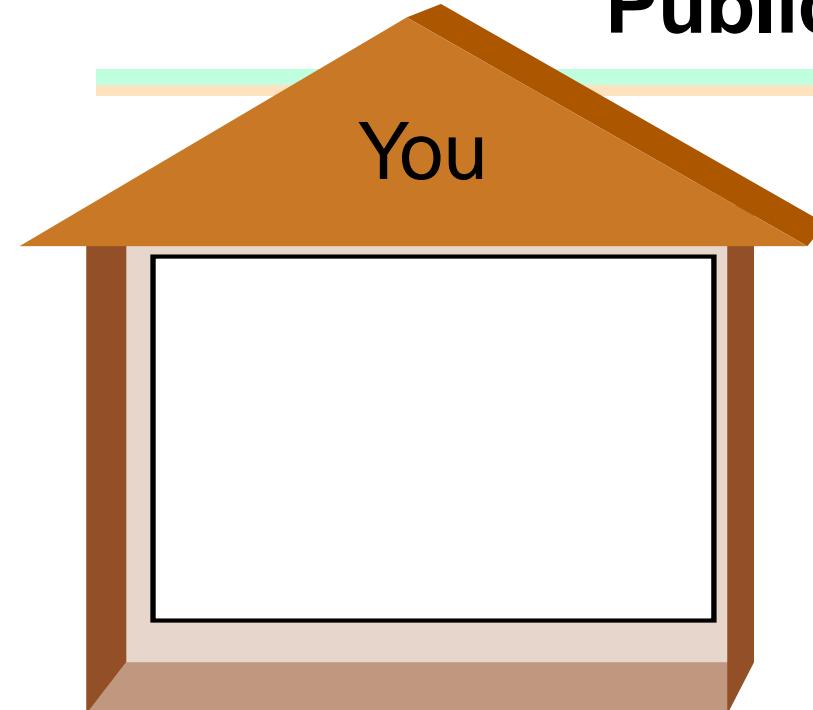
Manny

Private key: $N=p*q$

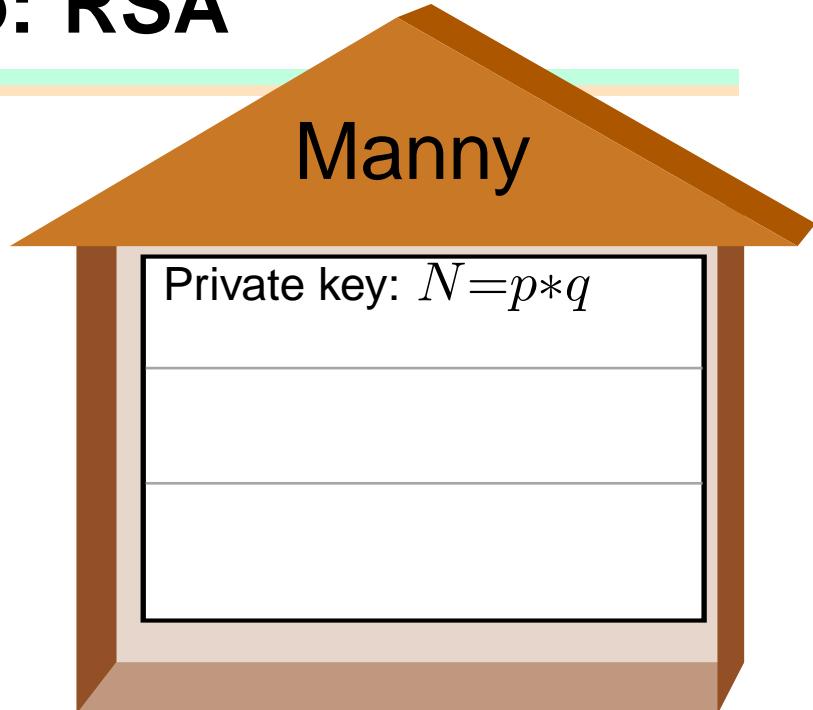
- Pick two large primes p, q .
- $N = p * q$

$$p=3, q=11.$$

Public Key Crypto: RSA



You



Manny

Private key: $N=p*q$

- Pick two large primes p, q .
- $N = p * q$

$$p=3, q=11.
N=33$$

Public Key Crypto: RSA

You

Manny

Private key: $N=p*q$

$\phi(N)=(p-1)*(q-1)$

- Pick two large primes p, q .
- $N = p * q$
- $\phi(N) = (p - 1) * (q - 1)$

$$p=3, q=11.
N=33$$

Public Key Crypto: RSA

You

Manny

Private key: $N=p*q$

$\phi(N)=(p-1)*(q-1)$

- Pick two large primes p, q .
- $N = p * q$
- $\phi(N) = (p - 1) * (q - 1)$

$$p=3, q=11.$$

$$N=33$$

$$\phi(33)=2 * 10=20$$

Public Key Crypto: RSA

You

Manny

Private key: $N=p*q$

$\phi(N)=(p-1)*(q-1)$

- Pick two large primes p, q .
- $N = p * q$
- $\phi(N) = (p - 1) * (q - 1)$
- Exponent x , $(x, \phi(N)) = 1$.

$$p=3, q=11.$$

$$N=33$$

$$\phi(33)=2 * 10=20$$

Public Key Crypto: RSA

You

Manny

Private key: $N=p*q$

Public key: N, x

$\phi(N)=(p-1)*(q-1)$

- Pick two large primes p, q .
- $N = p * q$
- $\phi(N) = (p - 1) * (q - 1)$
- Exponent $x, (x, \phi(N)) = 1$.

$$p=3, q=11.$$

$$N=33$$

$$\phi(33)=2 * 10=20$$

Public Key Crypto: RSA

You

Manny

Private key: $N=p*q$

Public key: N, x

$\phi(N)=(p-1)*(q-1)$

- Pick two large primes p, q .
- $N = p * q$
- $\phi(N) = (p - 1) * (q - 1)$
- Exponent $x, (x, \phi(N)) = 1$.

$$p=3, q=11.$$

$$N=33$$

$$\phi(33)=2 * 10=20$$

$$x=3$$

Public Key Crypto: RSA

You

Manny

Private key: $N=p*q$

Public key: N, x

$\phi(N)=(p-1)*(q-1)$

$y=x^{-1} \bmod \phi(N)$

- Pick two large primes p, q .
- $N = p * q$
- $\phi(N) = (p - 1) * (q - 1)$
- Exponent x , $(x, \phi(N)) = 1$.
- Compute $y = x^{-1} \bmod \phi(N)$.

$$p=3, q=11.$$

$$N=33$$

$$\phi(33)=2 * 10=20$$

$$x=3$$

Public Key Crypto: RSA

You

Manny

Private key: $N=p*q$

Public key: N, x

$\phi(N)=(p-1)*(q-1)$

$y=x^{-1} \bmod \phi(N)$

- Pick two large primes p, q .
- $N = p * q$
- $\phi(N) = (p - 1) * (q - 1)$
- Exponent $x, (x, \phi(N)) = 1$.
- Compute $y = x^{-1} \bmod \phi(N)$.

$$p=3, q=11.$$

$$N=33$$

$$\phi(33)=2 * 10=20$$

$$x=3$$

$$y=3^{-1}=7 \bmod \phi(N)$$

Public Key Crypto: RSA

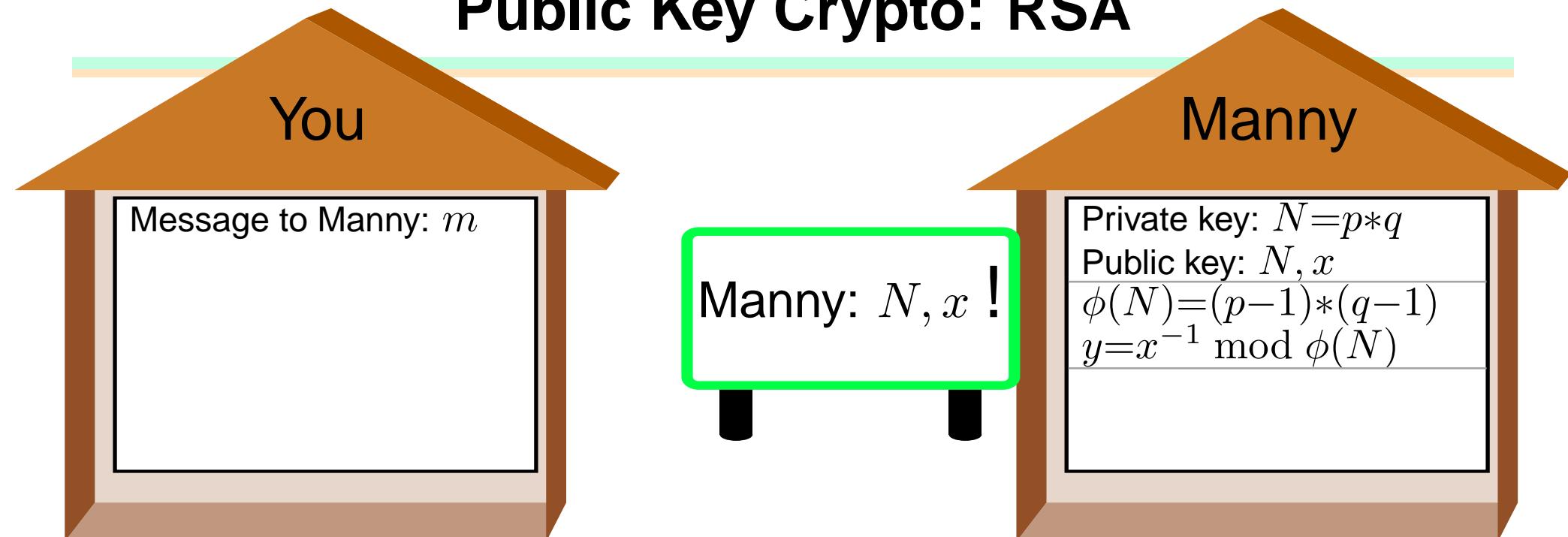


- Pick two large primes p, q .
- $N = p * q$
- $\phi(N) = (p - 1) * (q - 1)$
- Exponent x , $(x, \phi(N)) = 1$.
- Compute $y = x^{-1} \bmod \phi(N)$.

$$\begin{aligned} p &= 3, q = 11. \\ N &= 33 \\ \phi(33) &= 2 * 10 = 20 \\ x &= 3 \\ y &= 3^{-1} = 7 \bmod \phi(N) \end{aligned}$$



Public Key Crypto: RSA



- Pick two large primes p, q .
- $N = p * q$
- $\phi(N) = (p - 1) * (q - 1)$
- Exponent x , $(x, \phi(N)) = 1$.
- Compute $y = x^{-1} \bmod \phi(N)$.

$$\begin{aligned} p &= 3, q = 11. \\ N &= 33 \\ \phi(33) &= 2 * 10 = 20 \\ x &= 3 \\ y &= 3^{-1} = 7 \bmod \phi(N) \end{aligned}$$

Public Key Crypto: RSA



- Pick two large primes p, q .
- $N = p * q$
- $\phi(N) = (p - 1) * (q - 1)$
- Exponent x , $(x, \phi(N)) = 1$.
- Compute $y = x^{-1} \bmod \phi(N)$.

$$\begin{aligned} p &= 3, q = 11. \\ N &= 33 \\ \phi(33) &= 2 * 10 = 20 \\ x &= 3 \\ y &= 3^{-1} = 7 \bmod \phi(N) \end{aligned}$$

Public Key Crypto: RSA

You

Message to Manny: m

Cypher text:

$$c = m^x \bmod N$$

Example:

$$m=4$$

$$c=4^3=64=31 \bmod 33$$

Manny

Private key: $N=p*q$

Public key: N, x

$$\phi(N)=(p-1)*(q-1)$$

$$y=x^{-1} \bmod \phi(N)$$

Manny: $N, x !$

- Pick two large primes p, q .
- $N = p * q$
- $\phi(N) = (p - 1) * (q - 1)$
- Exponent x , $(x, \phi(N)) = 1$.
- Compute $y = x^{-1} \bmod \phi(N)$.

$$p=3, q=11.$$

$$N=33$$

$$\phi(33)=2 * 10=20$$

$$x=3$$

$$y=3^{-1}=7 \bmod \phi(N)$$

Public Key Crypto: RSA

You

Message to Manny: m

Cypher text:

$$c = m^x \bmod N$$

Example:

$$m=4$$

$$c=4^3=64=31 \bmod 33$$

c

To:
Manny

Manny

Private key: $N=p*q$

Public key: N, x

$$\phi(N)=(p-1)*(q-1)$$

$$y=x^{-1} \bmod \phi(N)$$

Manny: $N, x !$

- Pick two large primes p, q .
- $N = p * q$
- $\phi(N) = (p - 1) * (q - 1)$
- Exponent x , $(x, \phi(N)) = 1$.
- Compute $y = x^{-1} \bmod \phi(N)$.

$$p=3, q=11.$$

$$N=33$$

$$\phi(33)=2 * 10=20$$

$$x=3$$

$$y=3^{-1}=7 \bmod \phi(N)$$

Public Key Crypto: RSA

You

Message to Manny: m

Cypher text:

$$c = m^x \bmod N$$

Example:

$$m=4$$

$$c=4^3=64=31 \bmod 33$$

c

To:
Manny

Manny: N, x !

Private key: $N=p*q$

Public key: N, x

$$\phi(N)=(p-1)*(q-1)$$

$$y=x^{-1} \bmod \phi(N)$$

- Pick two large primes p, q .
- $N = p * q$
- $\phi(N) = (p - 1) * (q - 1)$
- Exponent x , $(x, \phi(N)) = 1$.
- Compute $y = x^{-1} \bmod \phi(N)$.

$$p=3, q=11.$$

$$N=33$$

$$\phi(33)=2 * 10=20$$

$$x=3$$

$$y=3^{-1}=7 \bmod \phi(N)$$

Public Key Crypto: RSA

You

Message to Manny: m

Cypher text:

$$c = m^x \bmod N$$

Example:

$$m=4$$

$$c=4^3=64=31 \bmod 33$$

Manny

Private key: $N=p*q$

Public key: N, x

$$\phi(N)=(p-1)*(q-1)$$

$$y=x^{-1} \bmod \phi(N)$$

Manny: $N, x !$

c To:
Manny

- Pick two large primes p, q .
- $N = p * q$
- $\phi(N) = (p - 1) * (q - 1)$
- Exponent x , $(x, \phi(N)) = 1$.
- Compute $y = x^{-1} \bmod \phi(N)$.

$p=3, q=11.$
 $N=33$
 $\phi(33)=2 * 10=20$
 $x=3$
 $y=3^{-1}=7 \bmod \phi(N)$

Public Key Crypto: RSA

You

Message to Manny: m

Cypher text:

$$c = m^x \bmod N$$

Example:

$$m=4$$

$$c=4^3=64=31 \bmod 33$$

Manny

Private key: $N=p*q$

Public key: N, x

$$\phi(N)=(p-1)*(q-1)$$

$$y=x^{-1} \bmod \phi(N)$$

Manny: $N, x !$

c	To: Manny
-----	--------------

- Pick two large primes p, q .
- $N = p * q$
- $\phi(N) = (p - 1) * (q - 1)$
- Exponent x , $(x, \phi(N)) = 1$.
- Compute $y = x^{-1} \bmod \phi(N)$.

$$p=3, q=11.$$
$$N=33$$
$$\phi(33)=2 * 10=20$$
$$x=3$$
$$y=3^{-1}=7 \bmod \phi(N)$$

Public Key Crypto: RSA

You

Message to Manny: m

Cypher text:

$$c = m^x \bmod N$$

Example:

$$m=4$$

$$c=4^3=64=31 \bmod 33$$

Manny

Private key: $N=p*q$

Public key: N, x

$$\phi(N)=(p-1)*(q-1)$$

$$y=x^{-1} \bmod \phi(N)$$

Manny: $N, x !$

c

To:
Manny

- Pick two large primes p, q .
- $N = p * q$
- $\phi(N) = (p - 1) * (q - 1)$
- Exponent x , $(x, \phi(N)) = 1$.
- Compute $y = x^{-1} \bmod \phi(N)$.

$$p=3, q=11.$$

$$N=33$$

$$\phi(33)=2 * 10=20$$

$$x=3$$

$$y=3^{-1}=7 \bmod \phi(N)$$

Public Key Crypto: RSA

You

Message to Manny: m

Cypher text:

$$c = m^x \bmod N$$

Example:

$$m=4$$

$$c=4^3=64=31 \bmod 33$$

Manny

Private key: $N=p*q$

Public key: N, x

$$\phi(N)=(p-1)*(q-1)$$

$$y=x^{-1} \bmod \phi(N)$$

Manny: $N, x !$

c

To:
Manny

- Pick two large primes p, q .
- $N = p * q$
- $\phi(N) = (p - 1) * (q - 1)$
- Exponent x , $(x, \phi(N)) = 1$.
- Compute $y = x^{-1} \bmod \phi(N)$.

$$p=3, q=11.$$

$$N=33$$

$$\phi(33)=2 * 10=20$$

$$x=3$$

$$y=3^{-1}=7 \bmod \phi(N)$$

Public Key Crypto: RSA

You

Message to Manny: m

Cypher text:

$$c = m^x \bmod N$$

Example:

$$m=4$$

$$c=4^3=64=31 \bmod 33$$

Manny

Private key: $N=p*q$

Public key: N, x

$$\phi(N)=(p-1)*(q-1)$$

$$y=x^{-1} \bmod \phi(N)$$

$$c^y \bmod N =$$

$$m^{xy} \bmod N = m$$

c

To:
Manny

- Pick two large primes p, q .
- $N = p * q$
- $\phi(N) = (p - 1) * (q - 1)$
- Exponent $x, (x, \phi(N)) = 1$.
- Compute $y = x^{-1} \bmod \phi(N)$.

$$p=3, q=11.$$

$$N=33$$

$$\phi(33)=2 * 10=20$$

$$x=3$$

$$y=3^{-1}=7 \bmod \phi(N)$$

Public Key Crypto: RSA

You

Message to Manny: m

Cypher text:

$$c = m^x \bmod N$$

Example:

$$m=4$$

$$c=4^3=64=31 \bmod 33$$

Manny

Private key: $N=p*q$

Public key: N, x

$$\phi(N)=(p-1)*(q-1)$$

$$y=x^{-1} \bmod \phi(N)$$

$$c^y \bmod N =$$

$$m^{xy} \bmod N = m$$

c

To:
Manny

- Pick two large primes p, q .
- $N = p * q$
- $\phi(N) = (p - 1) * (q - 1)$
- Exponent $x, (x, \phi(N)) = 1$.
- Compute $y = x^{-1} \bmod \phi(N)$.
- $(\text{Order of } m) \mid \phi(N)!$

$$p=3, q=11.$$

$$N=33$$

$$\phi(33)=2 * 10=20$$

$$x=3$$

$$y=3^{-1}=7 \bmod \phi(N)$$

Public Key Crypto: RSA

You

Message to Manny: m

Cypher text:

$$c = m^x \bmod N$$

Example:

$$m=4$$

$$c=4^3=64=31 \bmod 33$$

Manny

Private key: $N=p*q$

Public key: N, x

$$\phi(N)=(p-1)*(q-1)$$

$$y=x^{-1} \bmod \phi(N)$$

$$c^y \bmod N =$$

$$m^{xy} \bmod N = m$$

c

To:
Manny

- Pick two large primes p, q .
- $N = p * q$
- $\phi(N) = (p - 1) * (q - 1)$
- Exponent $x, (x, \phi(N)) = 1$.
- Compute $y = x^{-1} \bmod \phi(N)$.
- $(\text{Order of } m) \mid \phi(N)!$

$$p=3, q=11.$$

$$N=33$$

$$\phi(33)=2 * 10=20$$

$$x=3$$

$$y=3^{-1}=7 \bmod \phi(N)$$

$$c^y=31^7 \bmod N$$

Public Key Crypto: RSA

You

Message to Manny: m

Cypher text:

$$c = m^x \bmod N$$

Example:

$$m=4$$

$$c=4^3=64=31 \bmod 33$$

Manny

Private key: $N=p*q$

Public key: N, x

$$\phi(N)=(p-1)*(q-1)$$

$$y=x^{-1} \bmod \phi(N)$$

$$c^y \bmod N =$$

$$m^{xy} \bmod N = m$$

c

To:
Manny

- Pick two large primes p, q .
- $N = p * q$
- $\phi(N) = (p - 1) * (q - 1)$
- Exponent $x, (x, \phi(N)) = 1$.
- Compute $y = x^{-1} \bmod \phi(N)$.
- $(\text{Order of } m) \mid \phi(N)!$

$$p=3, q=11.$$

$$N=33$$

$$\phi(33)=2 * 10=20$$

$$x=3$$

$$y=3^{-1}=7 \bmod \phi(N)$$

$$c^y=31^7 \bmod N$$

$$=31 * 31^2 * 31^4 \bmod N$$

Public Key Crypto: RSA

You

Message to Manny: m

Cypher text:

$$c = m^x \bmod N$$

Example:

$$m=4$$

$$c=4^3=64=31 \bmod 33$$

$$31=-2 \bmod 33$$

Manny: $N, x !$

Private key: $N=p*q$

Public key: N, x

$$\phi(N)=(p-1)*(q-1)$$

$$y=x^{-1} \bmod \phi(N)$$

$$c^y \bmod N =$$

$$m^{xy} \bmod N = m$$

c

To:
Manny

$$p=3, q=11.$$

$$N=33$$

$$\phi(33)=2 * 10=20$$

$$x=3$$

$$y=3^{-1}=7 \bmod \phi(N)$$

$$c^y=31^7 \bmod N$$

$$=31 * 31^2 * 31^4 \bmod N$$

- Pick two large primes p, q .
- $N = p * q$
- $\phi(N) = (p - 1) * (q - 1)$
- Exponent $x, (x, \phi(N)) = 1$.
- Compute $y = x^{-1} \bmod \phi(N)$.
- $(\text{Order of } m) \mid \phi(N)!$

Public Key Crypto: RSA

You

Message to Manny: m

Cypher text:

$$c = m^x \bmod N$$

Example:

$$m=4$$

$$c=4^3=64=31 \bmod 33$$

$$31=-2 \bmod 33$$

$$(-2)^2=4 \bmod 33$$

Manny

Private key: $N=p*q$

Public key: N, x

$$\phi(N)=(p-1)*(q-1)$$

$$y=x^{-1} \bmod \phi(N)$$

$$c^y \bmod N =$$

$$m^{xy} \bmod N = m$$

c

To:
Manny

$$p=3, q=11.$$

$$N=33$$

$$\phi(33)=2 * 10=20$$

$$x=3$$

$$y=3^{-1}=7 \bmod \phi(N)$$

$$c^y=31^7 \bmod N$$

$$=31 * 31^2 * 31^4 \bmod N$$

- Pick two large primes p, q .
- $N = p * q$
- $\phi(N) = (p - 1) * (q - 1)$
- Exponent $x, (x, \phi(N)) = 1$.
- Compute $y = x^{-1} \bmod \phi(N)$.
- $(\text{Order of } m) \mid \phi(N)!$

Public Key Crypto: RSA

You

Message to Manny: m

Cypher text:

$$c = m^x \bmod N$$

Example:

$$m=4$$

$$c=4^3=64=31 \bmod 33$$

Manny: $N, x !$

Private key: $N=p*q$

Public key: N, x

$$\phi(N)=(p-1)*(q-1)$$

$$y=x^{-1} \bmod \phi(N)$$

$$c^y \bmod N =$$

$$m^{xy} \bmod N = m$$

c

To:
Manny

$$31=-2 \bmod 33$$

$$(-2)^2=4 \bmod 33$$

$$(-2)^4=4^2$$

$$=16 \bmod 33$$

- Pick two large primes p, q .
- $N = p * q$
- $\phi(N) = (p - 1) * (q - 1)$
- Exponent x , $(x, \phi(N)) = 1$.
- Compute $y = x^{-1} \bmod \phi(N)$.
- $(\text{Order of } m) \mid \phi(N)!$

$$p=3, q=11.$$

$$N=33$$

$$\phi(33)=2 * 10=20$$

$$x=3$$

$$y=3^{-1}=7 \bmod \phi(N)$$

$$c^y=31^7 \bmod N$$

$$=31 * 31^2 * 31^4 \bmod N$$

Public Key Crypto: RSA

You

Message to Manny: m

Cypher text:

$$c = m^x \bmod N$$

Example:

$$m=4$$

$$c=4^3=64=31 \bmod 33$$

$$31=-2 \bmod 33$$

$$(-2)^2=4 \bmod 33$$

$$(-2)^4=4^2$$

$$=16 \bmod 33$$

$$(-2)^7=-2 * 4 * 16 \bmod 33$$

Manny: $N, x !$

Private key: $N=p*q$

Public key: N, x

$$\phi(N)=(p-1)*(q-1)$$

$$y=x^{-1} \bmod \phi(N)$$

$$c^y \bmod N =$$

$$m^{xy} \bmod N = m$$

c

To:
Manny

$$p=3, q=11.$$

$$N=33$$

$$\phi(33)=2 * 10=20$$

$$x=3$$

$$y=3^{-1}=7 \bmod \phi(N)$$

$$c^y=31^7 \bmod N$$

$$=31 * 31^2 * 31^4 \bmod N$$

- Pick two large primes p, q .
- $N = p * q$
- $\phi(N) = (p - 1) * (q - 1)$
- Exponent $x, (x, \phi(N)) = 1$.
- Compute $y = x^{-1} \bmod \phi(N)$.
- $(\text{Order of } m) \mid \phi(N)!$

Public Key Crypto: RSA

You

Message to Manny: m

Cypher text:

$$c = m^x \bmod N$$

Example:

$$m=4$$

$$c=4^3=64=31 \bmod 33$$

$$31=-2 \bmod 33$$

$$(-2)^2=4 \bmod 33$$

$$(-2)^4=4^2$$

$$=16 \bmod 33$$

$$(-2)^7=-2 * 4 * 16 \bmod 33$$

$$=-2 * 64 \bmod 33$$

Manny: $N, x !$

Private key: $N=p*q$

Public key: N, x

$$\phi(N)=(p-1)*(q-1)$$

$$y=x^{-1} \bmod \phi(N)$$

$$c^y \bmod N =$$

$$m^{xy} \bmod N = m$$

c

To:
Manny

$$p=3, q=11.$$

$$N=33$$

$$\phi(33)=2 * 10=20$$

$$x=3$$

$$y=3^{-1}=7 \bmod \phi(N)$$

$$c^y=31^7 \bmod N$$

$$=31 * 31^2 * 31^4 \bmod N$$

- Pick two large primes p, q .
- $N = p * q$
- $\phi(N) = (p - 1) * (q - 1)$
- Exponent $x, (x, \phi(N)) = 1$.
- Compute $y = x^{-1} \bmod \phi(N)$.
- $(\text{Order of } m) \mid \phi(N)!$

Public Key Crypto: RSA

You

Message to Manny: m

Cypher text:

$$c = m^x \bmod N$$

Example:

$$m=4$$

$$c=4^3=64=31 \bmod 33$$

$$31=-2 \bmod 33$$

$$(-2)^2=4 \bmod 33$$

$$(-2)^4=4^2$$

$$=16 \bmod 33$$

$$(-2)^7=-2 * 4 * 16 \bmod 33$$

$$=-2 * 64 \bmod 33$$

$$=-2 * 31 \bmod 33$$

- Pick two large primes p, q .
- $N = p * q$
- $\phi(N) = (p - 1) * (q - 1)$
- Exponent $x, (x, \phi(N)) = 1$.
- Compute $y = x^{-1} \bmod \phi(N)$.
- $(\text{Order of } m) \mid \phi(N)!$

Manny

Private key: $N=p*q$

Public key: N, x

$$\phi(N)=(p-1)*(q-1)$$

$$y=x^{-1} \bmod \phi(N)$$

$$c^y \bmod N =$$

$$m^{xy} \bmod N = m$$

c

To:
Manny

$$p=3, q=11.$$

$$N=33$$

$$\phi(33)=2 * 10=20$$

$$x=3$$

$$y=3^{-1}=7 \bmod \phi(N)$$

$$c^y=31^7 \bmod N$$

$$=31 * 31^2 * 31^4 \bmod N$$

Public Key Crypto: RSA

You

Message to Manny: m

Cypher text:

$$c = m^x \bmod N$$

Example:

$$m=4$$

$$c=4^3=64=31 \bmod 33$$

Manny: $N, x !$

Private key: $N=p*q$

Public key: N, x

$$\phi(N) = (p-1)*(q-1)$$

$$y = x^{-1} \bmod \phi(N)$$

$$c^y \bmod N =$$

$$m^{xy} \bmod N = m$$

c

To:
Manny

$$31 = -2 \bmod 33$$

$$(-2)^2 = 4 \bmod 33$$

$$(-2)^4 = 4^2$$

$$= 16 \bmod 33$$

$$(-2)^7 = -2 * 4 * 16 \bmod 33$$

$$= -2 * 64 \bmod 33$$

$$= -2 * 31 \bmod 33$$

$$= -62 \bmod 33$$

- Pick two large primes p, q .
- $N = p * q$
- $\phi(N) = (p-1)*(q-1)$
- Exponent $x, (x, \phi(N)) = 1$.
- Compute $y = x^{-1} \bmod \phi(N)$.
- $(\text{Order of } m) \mid \phi(N)!$

$$p=3, q=11.$$

$$N=33$$

$$\phi(33)=2*10=20$$

$$x=3$$

$$y=3^{-1}=7 \bmod \phi(N)$$

$$c^y = 31^7 \bmod N$$

$$= 31 * 31^2 * 31^4 \bmod N$$

Public Key Crypto: RSA

You

Message to Manny: m

Cypher text:

$$c = m^x \bmod N$$

Example:

$$m=4$$

$$c=4^3=64=31 \bmod 33$$

$$31=-2 \bmod 33$$

$$(-2)^2=4 \bmod 33$$

$$(-2)^4=4^2$$

$$=16 \bmod 33$$

$$(-2)^7=-2 * 4 * 16 \bmod 33$$

$$=-2 * 64 \bmod 33$$

$$=-2 * 31 \bmod 33$$

$$=-62 \bmod 33$$

$$=4 \bmod 33$$

Manny

Private key: $N=p*q$

Public key: N, x

$$\phi(N)=(p-1)*(q-1)$$

$$y=x^{-1} \bmod \phi(N)$$

$$c^y \bmod N=$$

$$m^{xy} \bmod N=m$$

c

To:
Manny

$$p=3, q=11.$$

$$N=33$$

$$\phi(33)=2 * 10=20$$

$$x=3$$

$$y=3^{-1}=7 \bmod \phi(N)$$

$$c^y=31^7 \bmod N$$

$$=31 * 31^2 * 31^4 \bmod N$$

- Pick two large primes p, q .
- $N = p * q$
- $\phi(N) = (p - 1) * (q - 1)$
- Exponent $x, (x, \phi(N)) = 1$.
- Compute $y = x^{-1} \bmod \phi(N)$.
- $(\text{Order of } m) \mid \phi(N)!$

Factor Finding with Square Roots of Unity

- Let N be composite.
- Find x such that $x^2 = 1 \pmod{N}$ and $x \neq \pm 1 \pmod{N}$.



Factor Finding with Square Roots of Unity

- Let N be composite.
- Find x such that $x^2 \equiv 1 \pmod{N}$ and $x \not\equiv \pm 1 \pmod{N}$.
 - $x^2 - 1 = (x - 1)(x + 1) \equiv 0 \pmod{N}$.



Factor Finding with Square Roots of Unity

- Let N be composite.
- Find x such that $x^2 = 1 \pmod{N}$ and $x \neq \pm 1 \pmod{N}$.
 - $x^2 - 1 = (x - 1)(x + 1) = 0 \pmod{N}$.
 - Hence either $N > \gcd(x - 1, N) > 1$ or
 $N > \gcd(x + 1, N) > 1$.



Factor Finding with Square Roots of Unity

- Let N be composite.
- Find x such that $x^2 = 1 \pmod{N}$ and $x \neq \pm 1 \pmod{N}$.
 - $x^2 - 1 = (x - 1)(x + 1) = 0 \pmod{N}$.
 - Hence either $N > \gcd(x - 1, N) > 1$ or
 $N > \gcd(x + 1, N) > 1$.
 - N can be factored.



Factor Finding with Square Roots of Unity

- Let N be composite.
- Find x such that $x^2 = 1 \pmod{N}$ and $x \neq \pm 1 \pmod{N}$.
 - $x^2 - 1 = (x - 1)(x + 1) = 0 \pmod{N}$.
 - Hence either $N > \gcd(x - 1, N) > 1$ or
 $N > \gcd(x + 1, N) > 1$.
 - N can be factored.
 - x is a *non-trivial squareroot of unity modulo N* .



Factor Finding with Square Roots of Unity

- Let N be composite.
- Find x such that $x^2 = 1 \pmod{N}$ and $x \neq \pm 1 \pmod{N}$.
 - $x^2 - 1 = (x - 1)(x + 1) = 0 \pmod{N}$.
 - Hence either $\gcd(x - 1, N) > 1$ or $\gcd(x + 1, N) > 1$.
 - N can be factored.
 - x is a *non-trivial squareroot of unity modulo N* .
- Suppose N is odd, not a prime power.
Then nontrivial squareroots of unity modulo N exist.



Factor Finding with Square Roots of Unity

- Let N be composite.
- Find x such that $x^2 = 1 \pmod{N}$ and $x \neq \pm 1 \pmod{N}$.
 - $x^2 - 1 = (x - 1)(x + 1) = 0 \pmod{N}$.
 - Hence either $\gcd(x - 1, N) > 1$ or $\gcd(x + 1, N) > 1$.
 - N can be factored.
 - x is a *non-trivial squareroot of unity modulo N* .
- Suppose N is odd, not a prime power.
Then nontrivial squareroots of unity modulo N exist.
- Examples:
 $N = 15$. $\sqrt{1} \pmod{15}$: 1, 4, 11, 14. $\gcd(4 - 1, 15) = 3$, $\gcd(4 + 1, 15) = 5$.



Factor Finding with Square Roots of Unity

- Let N be composite.
- Find x such that $x^2 = 1 \pmod{N}$ and $x \neq \pm 1 \pmod{N}$.
 - $x^2 - 1 = (x - 1)(x + 1) = 0 \pmod{N}$.
 - Hence either $\gcd(x - 1, N) > 1$ or $\gcd(x + 1, N) > 1$.
 - N can be factored.
 - x is a *non-trivial squareroot of unity modulo N* .
- Suppose N is odd, not a prime power.
Then nontrivial squareroots of unity modulo N exist.
- Examples:
 - $N = 15$. $\sqrt{1} \pmod{15}$: 1, 4, 11, 14. $\gcd(4 - 1, 15) = 3$, $\gcd(4 + 1, 15) = 5$.
 - $N = 35$. $\sqrt{1} \pmod{35}$: 1, 6, 29, 34. $\gcd(29 - 1, 35) = 7$, $\gcd(29 + 1, 35) = 5$.



Factor Finding with Square Roots of Unity

- Let N be composite.
- Find x such that $x^2 = 1 \pmod{N}$ and $x \neq \pm 1 \pmod{N}$.
 - $x^2 - 1 = (x - 1)(x + 1) = 0 \pmod{N}$.
 - Hence either $\gcd(x - 1, N) > 1$ or $\gcd(x + 1, N) > 1$.
 - N can be factored.
 - x is a *non-trivial squareroot of unity modulo N* .
- Suppose N is odd, not a prime power.
Then nontrivial squareroots of unity modulo N exist.
- Examples:
 - $N = 15$. $\sqrt{1} \pmod{15}$: 1, 4, 11, 14. $\gcd(4 - 1, 15) = 3$, $\gcd(4 + 1, 15) = 5$.
 - $N = 35$. $\sqrt{1} \pmod{35}$: 1, 6, 29, 34. $\gcd(29 - 1, 35) = 7$, $\gcd(29 + 1, 35) = 5$.
- Easy to factor: $\begin{cases} \text{Even numbers: } N = 2M. \\ \text{Power numbers: } N = M^k, k > 1 \end{cases}$



From Order to Square Roots of Unity

FACTOR(N)

Input: $N > 1$, N odd, not a prime power.

Output: A nontrivial factor f of N .

$f \leftarrow 1$

while $f = 1$

$x \leftarrow \text{rand}(2, N - 1)$; $f \leftarrow \text{gcd}(x, N)$

if $f > 1$ **then return** f

$o \leftarrow \text{ORDER}(x \bmod N)$

if $2|o \& x^{o/2} \bmod N \notin \{\pm 1\}$

$f \leftarrow \max(\text{gcd}(x^{o/2} - 1, N), \text{gcd}(x^{o/2} + 1, N))$

end

end

return f



From Order to Square Roots of Unity

FACTOR(N)

Input: $N > 1$, N odd, not a prime power.

Output: A nontrivial factor f of N .

$f \leftarrow 1$

while $f = 1$

$x \leftarrow \text{rand}(2, N - 1)$; $f \leftarrow \text{gcd}(x, N)$

if $f > 1$ **then return** f

$o \leftarrow \text{ORDER}(x \bmod N)$

if $2|o \& x^{o/2} \bmod N \notin \{\pm 1\}$

$f \leftarrow \max(\text{gcd}(x^{o/2} - 1, N), \text{gcd}(x^{o/2} + 1, N))$

end

end

return f

$N=35$



From Order to Square Roots of Unity

FACTOR(N)

Input: $N > 1$, N odd, not a prime power.

Output: A nontrivial factor f of N .

⇒ $f \leftarrow 1$

while $f = 1$

$x \leftarrow \text{rand}(2, N - 1)$; $f \leftarrow \text{gcd}(x, N)$

if $f > 1$ **then return** f

$o \leftarrow \text{ORDER}(x \bmod N)$

if $2|o \& x^{o/2} \bmod N \notin \{\pm 1\}$

$f \leftarrow \max(\text{gcd}(x^{o/2} - 1, N), \text{gcd}(x^{o/2} + 1, N))$

end

end

return f

$N=35$

From Order to Square Roots of Unity

FACTOR(N)

Input: $N > 1$, N odd, not a prime power.

Output: A nontrivial factor f of N .

```
 $f \leftarrow 1$ 
 $\Rightarrow \textbf{while } f = 1$ 
     $x \leftarrow \text{rand}(2, N - 1); f \leftarrow \text{gcd}(x, N)$ 
    if  $f > 1$  then return  $f$ 
     $o \leftarrow \text{ORDER}(x \bmod N)$ 
    if  $2|o \& x^{o/2} \bmod N \notin \{\pm 1\}$ 
         $f \leftarrow \max(\text{gcd}(x^{o/2} - 1, N), \text{gcd}(x^{o/2} + 1, N))$ 
    end
end
return  $f$ 
```

$N=35$

From Order to Square Roots of Unity

FACTOR(N)

Input: $N > 1$, N odd, not a prime power.

Output: A nontrivial factor f of N .

$N=35$

```
 $f \leftarrow 1$ 
while  $f = 1$ 
     $\Rightarrow x \leftarrow \text{rand}(2, N - 1); f \leftarrow \text{gcd}(x, N)$ 
        if  $f > 1$  then return  $f$ 
         $o \leftarrow \text{ORDER}(x \bmod N)$ 
        if  $2|o \& x^{o/2} \bmod N \notin \{\pm 1\}$ 
             $f \leftarrow \max(\text{gcd}(x^{o/2} - 1, N), \text{gcd}(x^{o/2} + 1, N))$ 
    end
end
return  $f$ 
```

From Order to Square Roots of Unity

FACTOR(N)

Input: $N > 1$, N odd, not a prime power.

Output: A nontrivial factor f of N .

$f \leftarrow 1$

while $f = 1$

$\Rightarrow x \leftarrow \text{rand}(2, N - 1); f \leftarrow \text{gcd}(x, N)$

if $f > 1$ **then return** f

$o \leftarrow \text{ORDER}(x \bmod N)$

if $2|o \& x^{o/2} \bmod N \notin \{\underline{1}, \underline{-1}\}$

$f \leftarrow \max(\text{gcd}(x^{o/2} - 1, N), \text{gcd}(x^{o/2} + 1, N))$

end

end

return f

$N=35$

$x=9$

From Order to Square Roots of Unity

FACTOR(N)

Input: $N > 1$, N odd, not a prime power.

Output: A nontrivial factor f of N .

$f \leftarrow 1$

while $f = 1$

$x \leftarrow \text{rand}(2, N - 1)$; $f \leftarrow \text{gcd}(x, N)$

\Rightarrow **if** $f > 1$ **then return** f

$o \leftarrow \text{ORDER}(x \bmod N)$

if $2|o$ & $x^{o/2} \bmod N \notin \{\pm 1\}$

$f \leftarrow \max(\text{gcd}(x^{o/2} - 1, N), \text{gcd}(x^{o/2} + 1, N))$

end

end

return f

$N=35$

$x=9$

From Order to Square Roots of Unity

FACTOR(N)

Input: $N > 1$, N odd, not a prime power.

Output: A nontrivial factor f of N .

$f \leftarrow 1$

while $f = 1$

$x \leftarrow \text{rand}(2, N - 1)$; $f \leftarrow \text{gcd}(x, N)$

if $f > 1$ **then return** f

$\Rightarrow o \leftarrow \text{ORDER}(x \bmod N)$

if $2|o \& x^{o/2} \bmod N \notin \{\pm 1\}$

$f \leftarrow \max(\text{gcd}(x^{o/2} - 1, N), \text{gcd}(x^{o/2} + 1, N))$

end

end

return f

$N=35$

$x=9$

From Order to Square Roots of Unity

FACTOR(N)

Input: $N > 1$, N odd, not a prime power.

Output: A nontrivial factor f of N .

$f \leftarrow 1$

while $f = 1$

$x \leftarrow \text{rand}(2, N - 1)$; $f \leftarrow \text{gcd}(x, N)$

if $f > 1$ **then return** f

$\Rightarrow o \leftarrow \text{ORDER}(x \bmod N)$

if $2|o \& x^{o/2} \bmod N \notin \{\underline{1}, \underline{-1}\}$

$f \leftarrow \max(\text{gcd}(x^{o/2} - 1, N), \text{gcd}(x^{o/2} + 1, N))$

end

end

return f

$$N=35$$

$$x=9$$

$$x^2 = x * 9$$



From Order to Square Roots of Unity

FACTOR(N)

Input: $N > 1$, N odd, not a prime power.

Output: A nontrivial factor f of N .

$f \leftarrow 1$

while $f = 1$

$x \leftarrow \text{rand}(2, N - 1)$; $f \leftarrow \text{gcd}(x, N)$

if $f > 1$ **then return** f

$\Rightarrow o \leftarrow \text{ORDER}(x \bmod N)$

if $2|o \& x^{o/2} \bmod N \notin \{\pm 1\}$

$f \leftarrow \max(\text{gcd}(x^{o/2} - 1, N), \text{gcd}(x^{o/2} + 1, N))$

end

end

return f

$$N=35$$

$$x=9$$

$$x^2 = x * 9 = 11 \bmod 35$$



From Order to Square Roots of Unity

FACTOR(N)

Input: $N > 1$, N odd, not a prime power.

Output: A nontrivial factor f of N .

$f \leftarrow 1$

while $f = 1$

$x \leftarrow \text{rand}(2, N - 1)$; $f \leftarrow \text{gcd}(x, N)$

if $f > 1$ **then return** f

$\Rightarrow o \leftarrow \text{ORDER}(x \bmod N)$

if $2|o \& x^{o/2} \bmod N \notin \{\underline{1}, \underline{-1}\}$

$f \leftarrow \max(\text{gcd}(x^{o/2} - 1, N), \text{gcd}(x^{o/2} + 1, N))$

end

end

return f

$$N=35$$

$$x=9$$

$$x^2 = x * 9 = 11 \bmod 35$$

$$x^3 = x * 11$$

From Order to Square Roots of Unity

FACTOR(N)

Input: $N > 1$, N odd, not a prime power.

Output: A nontrivial factor f of N .

$f \leftarrow 1$

while $f = 1$

$x \leftarrow \text{rand}(2, N - 1)$; $f \leftarrow \text{gcd}(x, N)$

if $f > 1$ **then return** f

$\Rightarrow o \leftarrow \text{ORDER}(x \bmod N)$

if $2|o \& x^{o/2} \bmod N \notin \{\pm 1\}$

$f \leftarrow \max(\text{gcd}(x^{o/2} - 1, N), \text{gcd}(x^{o/2} + 1, N))$

end

end

return f

$$N=35$$

$$x=9$$

$$x^2 = x * 9 = 11 \bmod 35$$

$$x^3 = x * 11 = 29 \bmod 35$$



From Order to Square Roots of Unity

FACTOR(N)

Input: $N > 1$, N odd, not a prime power.

Output: A nontrivial factor f of N .

$f \leftarrow 1$

while $f = 1$

$x \leftarrow \text{rand}(2, N - 1)$; $f \leftarrow \text{gcd}(x, N)$

if $f > 1$ **then return** f

$\Rightarrow o \leftarrow \text{ORDER}(x \bmod N)$

if $2|o \& x^{o/2} \bmod N \notin \{\underline{1}, \underline{-1}\}$

$f \leftarrow \max(\text{gcd}(x^{o/2} - 1, N), \text{gcd}(x^{o/2} + 1, N))$

end

end

return f

$$N=35$$

$$x=9$$

$$x^2 = x * 9 = 11 \bmod 35$$

$$x^3 = x * 11 = 29 \bmod 35$$

$$x^4 = x * 29$$

From Order to Square Roots of Unity

FACTOR(N)

Input: $N > 1$, N odd, not a prime power.

Output: A nontrivial factor f of N .

$f \leftarrow 1$

while $f = 1$

$x \leftarrow \text{rand}(2, N - 1); f \leftarrow \text{gcd}(x, N)$

if $f > 1$ **then return** f

$\Rightarrow o \leftarrow \text{ORDER}(x \bmod N)$

if $2|o \& x^{o/2} \bmod N \notin \{\underline{1}, \underline{-1}\}$

$f \leftarrow \max(\text{gcd}(x^{o/2} - 1, N), \text{gcd}(x^{o/2} + 1, N))$

end

end

return f

$$N=35$$

$$x=9$$

$$x^2 = x * 9 = 11 \bmod 35$$

$$x^3 = x * 11 = 29 \bmod 35$$

$$x^4 = x * 29 = 16 \bmod 35$$



From Order to Square Roots of Unity

FACTOR(N)

Input: $N > 1$, N odd, not a prime power.

Output: A nontrivial factor f of N .

$f \leftarrow 1$

while $f = 1$

$x \leftarrow \text{rand}(2, N - 1)$; $f \leftarrow \text{gcd}(x, N)$

if $f > 1$ **then return** f

$\Rightarrow o \leftarrow \text{ORDER}(x \bmod N)$

if $2|o \& x^{o/2} \bmod N \notin \{\underline{1}, \underline{-1}\}$

$f \leftarrow \max(\text{gcd}(x^{o/2} - 1, N), \text{gcd}(x^{o/2} + 1, N))$

end

end

return f

$$N=35$$

$$x=9$$

$$x^2 = x * 9 = 11 \bmod 35$$

$$x^3 = x * 11 = 29 \bmod 35$$

$$x^4 = x * 29 = 16 \bmod 35$$

$$x^5 = x * 16$$



From Order to Square Roots of Unity

FACTOR(N)

Input: $N > 1$, N odd, not a prime power.

Output: A nontrivial factor f of N .

$f \leftarrow 1$

while $f = 1$

$x \leftarrow \text{rand}(2, N - 1)$; $f \leftarrow \text{gcd}(x, N)$

if $f > 1$ **then return** f

$\Rightarrow o \leftarrow \text{ORDER}(x \bmod N)$

if $2|o \& x^{o/2} \bmod N \notin \{\underline{1}, \underline{-1}\}$

$f \leftarrow \max(\text{gcd}(x^{o/2} - 1, N), \text{gcd}(x^{o/2} + 1, N))$

end

end

return f

$$N=35$$

$$x=9$$

$$x^2 = x * 9 = 11 \bmod 35$$

$$x^3 = x * 11 = 29 \bmod 35$$

$$x^4 = x * 29 = 16 \bmod 35$$

$$x^5 = x * 16 = 4 \bmod 35$$



From Order to Square Roots of Unity

FACTOR(N)

Input: $N > 1$, N odd, not a prime power.

Output: A nontrivial factor f of N .

$f \leftarrow 1$

while $f = 1$

$x \leftarrow \text{rand}(2, N - 1)$; $f \leftarrow \text{gcd}(x, N)$

if $f > 1$ **then return** f

$\Rightarrow o \leftarrow \text{ORDER}(x \bmod N)$

if $2|o \& x^{o/2} \bmod N \notin \{\underline{1}, \underline{-1}\}$

$f \leftarrow \max(\text{gcd}(x^{o/2} - 1, N), \text{gcd}(x^{o/2} + 1, N))$

end

end

return f

$$N=35$$

$$x=9$$

$$x^2 = x * 9 = 11 \bmod 35$$

$$x^3 = x * 11 = 29 \bmod 35$$

$$x^4 = x * 29 = 16 \bmod 35$$

$$x^5 = x * 16 = 4 \bmod 35$$

$$x^6 = x * 4$$



From Order to Square Roots of Unity

FACTOR(N)

Input: $N > 1$, N odd, not a prime power.

Output: A nontrivial factor f of N .

$f \leftarrow 1$

while $f = 1$

$x \leftarrow \text{rand}(2, N - 1)$; $f \leftarrow \text{gcd}(x, N)$

if $f > 1$ **then return** f

$\Rightarrow o \leftarrow \text{ORDER}(x \bmod N)$

if $2|o \& x^{o/2} \bmod N \notin \{\underline{1}, \underline{-1}\}$

$f \leftarrow \max(\text{gcd}(x^{o/2} - 1, N), \text{gcd}(x^{o/2} + 1, N))$

end

end

return f

$$N=35$$

$$x=9$$

$$x^2 = x * 9 = 11 \bmod 35$$

$$x^3 = x * 11 = 29 \bmod 35$$

$$x^4 = x * 29 = 16 \bmod 35$$

$$x^5 = x * 16 = 4 \bmod 35$$

$$x^6 = x * 4 = 1 \bmod 35$$



From Order to Square Roots of Unity

FACTOR(N)

Input: $N > 1$, N odd, not a prime power.

Output: A nontrivial factor f of N .

$f \leftarrow 1$

while $f = 1$

$x \leftarrow \text{rand}(2, N - 1)$; $f \leftarrow \text{gcd}(x, N)$

if $f > 1$ **then return** f

$\Rightarrow o \leftarrow \text{ORDER}(x \bmod N)$

if $2|o \& x^{o/2} \bmod N \notin \{\underline{1}, \underline{-1}\}$

$f \leftarrow \max(\text{gcd}(x^{o/2} - 1, N), \text{gcd}(x^{o/2} + 1, N))$

end

end

return f

$$N=35$$

$$x=9$$

$$x^2 = x * 9 = 11 \bmod 35$$

$$x^3 = x * 11 = 29 \bmod 35$$

$$x^4 = x * 29 = 16 \bmod 35$$

$$x^5 = x * 16 = 4 \bmod 35$$

$$x^6 = x * 4 = 1 \bmod 35$$

$$o = 6$$



From Order to Square Roots of Unity

FACTOR(N)

Input: $N > 1$, N odd, not a prime power.

Output: A nontrivial factor f of N .

$f \leftarrow 1$

while $f = 1$

$x \leftarrow \text{rand}(2, N - 1)$; $f \leftarrow \text{gcd}(x, N)$

if $f > 1$ **then return** f

$o \leftarrow \text{ORDER}(x \bmod N)$

\Rightarrow **if** $2|o \& x^{o/2} \bmod N \notin \{\underline{1}, \underline{-1}\}$

$f \leftarrow \max(\text{gcd}(x^{o/2} - 1, N), \text{gcd}(x^{o/2} + 1, N))$

end

end

return f

$$N=35$$

$$x=9$$

$$x^2 = x * 9 = 11 \bmod 35$$

$$x^3 = x * 11 = 29 \bmod 35$$

$$x^4 = x * 29 = 16 \bmod 35$$

$$x^5 = x * 16 = 4 \bmod 35$$

$$x^6 = x * 4 = 1 \bmod 35$$

$$o = 6$$



From Order to Square Roots of Unity

FACTOR(N)

Input: $N > 1$, N odd, not a prime power.

Output: A nontrivial factor f of N .

$f \leftarrow 1$

while $f = 1$

$x \leftarrow \text{rand}(2, N - 1)$; $f \leftarrow \text{gcd}(x, N)$

if $f > 1$ **then return** f

$o \leftarrow \text{ORDER}(x \bmod N)$

if $2|o \& x^{o/2} \bmod N \notin \{\pm 1\}$

$\Rightarrow f \leftarrow \max(\text{gcd}(x^{o/2} - 1, N), \text{gcd}(x^{o/2} + 1, N))$

end

end

return f

$$N=35$$

$$x=9$$

$$x^2 = x * 9 = 11 \bmod 35$$

$$x^3 = x * 11 = 29 \bmod 35$$

$$x^4 = x * 29 = 16 \bmod 35$$

$$x^5 = x * 16 = 4 \bmod 35$$

$$x^6 = x * 4 = 1 \bmod 35$$

$$o = 6$$



From Order to Square Roots of Unity

FACTOR(N)

Input: $N > 1$, N odd, not a prime power.

Output: A nontrivial factor f of N .

$f \leftarrow 1$

while $f = 1$

$x \leftarrow \text{rand}(2, N - 1)$; $f \leftarrow \text{gcd}(x, N)$

if $f > 1$ **then return** f

$o \leftarrow \text{ORDER}(x \bmod N)$

if $2|o \& x^{o/2} \bmod N \notin \{\pm 1\}$

$\Rightarrow f \leftarrow \max(\text{gcd}(x^{o/2} - 1, N), \text{gcd}(x^{o/2} + 1, N))$

end

end

return f

$$N=35$$

$$x=9$$

$$x^2 = x * 9 = 11 \bmod 35$$

$$x^3 = x * 11 = 29 \bmod 35$$

$$x^4 = x * 29 = 16 \bmod 35$$

$$x^5 = x * 16 = 4 \bmod 35$$

$$x^6 = x * 4 = 1 \bmod 35$$

$$o = 6$$

$$x^{o/2} = x^3 = 29 \bmod 35$$



From Order to Square Roots of Unity

FACTOR(N)

Input: $N > 1$, N odd, not a prime power.

Output: A nontrivial factor f of N .

$f \leftarrow 1$

while $f = 1$

$x \leftarrow \text{rand}(2, N - 1)$; $f \leftarrow \text{gcd}(x, N)$

if $f > 1$ **then return** f

$o \leftarrow \text{ORDER}(x \bmod N)$

if $2|o \& x^{o/2} \bmod N \notin \{\underline{1}, \underline{-1}\}$

$\Rightarrow f \leftarrow \max(\text{gcd}(x^{o/2} - 1, N), \text{gcd}(x^{o/2} + 1, N))$

end

end

return f

$$N=35$$

$$x=9$$

$$x^2 = x * 9 = 11 \bmod 35$$

$$x^3 = x * 11 = 29 \bmod 35$$

$$x^4 = x * 29 = 16 \bmod 35$$

$$x^5 = x * 16 = 4 \bmod 35$$

$$x^6 = x * 4 = 1 \bmod 35$$

$$o = 6$$

$$x^{o/2} = x^3 = 29 \bmod 35$$

$$\text{gcd}(x^{o/2} - 1, 35) = 7$$



From Order to Square Roots of Unity

FACTOR(N)

Input: $N > 1$, N odd, not a prime power.

Output: A nontrivial factor f of N .

$f \leftarrow 1$

while $f = 1$

$x \leftarrow \text{rand}(2, N - 1)$; $f \leftarrow \text{gcd}(x, N)$

if $f > 1$ **then return** f

$o \leftarrow \text{ORDER}(x \bmod N)$

if $2|o \& x^{o/2} \bmod N \notin \{\underline{1}, -\underline{1}\}$

$\Rightarrow f \leftarrow \max(\text{gcd}(x^{o/2} - 1, N), \text{gcd}(x^{o/2} + 1, N))$

end

end

return f

$N=35$

$x=9$

$x^2 = x * 9 = 11 \bmod 35$

$x^3 = x * 11 = 29 \bmod 35$

$x^4 = x * 29 = 16 \bmod 35$

$x^5 = x * 16 = 4 \bmod 35$

$x^6 = x * 4 = 1 \bmod 35$

$o = 6$

$x^{o/2} = x^3 = 29 \bmod 35$

$\text{gcd}(x^{o/2} - 1, 35) = 7$

$\text{gcd}(x^{o/2} + 1, 35) = 5$



From Order to Square Roots of Unity

FACTOR(N)

Input: $N > 1$, N odd, not a prime power.

Output: A nontrivial factor f of N .

$f \leftarrow 1$

while $f = 1$

$x \leftarrow \text{rand}(2, N - 1)$; $f \leftarrow \text{gcd}(x, N)$

if $f > 1$ **then return** f

$o \leftarrow \text{ORDER}(x \bmod N)$

if $2|o \& x^{o/2} \bmod N \notin \{\pm 1\}$

$f \leftarrow \max(\text{gcd}(x^{o/2} - 1, N), \text{gcd}(x^{o/2} + 1, N))$

end

end

⇒ **return** f

$$N=35$$

$$x=9$$

$$x^2 = x * 9 = 11 \bmod 35$$

$$x^3 = x * 11 = 29 \bmod 35$$

$$x^4 = x * 29 = 16 \bmod 35$$

$$x^5 = x * 16 = 4 \bmod 35$$

$$x^6 = x * 4 = 1 \bmod 35$$

$$o = 6$$

$$x^{o/2} = x^3 = 29 \bmod 35$$

$$\text{gcd}(x^{o/2} - 1, 35) = 7$$

$$\text{gcd}(x^{o/2} + 1, 35) = 5$$



From Order to Square Roots of Unity

FACTOR(N)

Input: $N > 1$, N odd, not a prime power.

Output: A nontrivial factor f of N .

$f \leftarrow 1$

while $f = 1$

$x \leftarrow \text{rand}(2, N - 1)$; $f \leftarrow \text{gcd}(x, N)$

if $f > 1$ **then return** f

$o \leftarrow \text{ORDER}(x \bmod N)$

if $2|o \& x^{o/2} \bmod N \notin \{\pm 1\}$

$f \leftarrow \max(\text{gcd}(x^{o/2} - 1, N), \text{gcd}(x^{o/2} + 1, N))$

end

end

⇒ **return** f

- Prob($1 < x < N$ has even order o and $x^{o/2} \notin \{\pm 1\}$) $\geq 1/2$.

$$N=35$$

$$x=9$$

$$x^2 = x * 9 = 11 \bmod 35$$

$$x^3 = x * 11 = 29 \bmod 35$$

$$x^4 = x * 29 = 16 \bmod 35$$

$$x^5 = x * 16 = 4 \bmod 35$$

$$x^6 = x * 4 = 1 \bmod 35$$

$$o = 6$$

$$x^{o/2} = x^3 = 29 \bmod 35$$

$$\text{gcd}(x^{o/2} - 1, 35) = 7$$

$$\text{gcd}(x^{o/2} + 1, 35) = 5$$



Quantum Order Finding

- **Problem:** Given $N, x, (x, N) = 1$. ORDER($x \bmod N$)?



Quantum Order Finding

- **Problem:** Given $N, x, (x, N) = 1$. $\text{ORDER}(x \bmod N)$?
- Consider $M_x : y \mapsto x * y \bmod N$. Let $o = \text{ORDER}(x \bmod N)$.



Quantum Order Finding

- **Problem:** Given $N, x, (x, N) = 1$. $\text{ORDER}(x \bmod N)$?
- Consider $M_x : y \mapsto x * y \bmod N$. Let $o = \text{ORDER}(x \bmod N)$.
 - M_x is invertible on \mathbb{Z}_N .



Quantum Order Finding

- **Problem:** Given $N, x, (x, N) = 1$. $\text{ORDER}(x \bmod N)$?
- Consider $M_x : y \mapsto x * y \bmod N$. Let $o = \text{ORDER}(x \bmod N)$.
 - M_x is invertible on \mathbb{Z}_N^* .
 - M_x 's cycles on \mathbb{Z}_N^* are of length o .



Quantum Order Finding

- **Problem:** Given $N, x, (x, N) = 1$. $\text{ORDER}(x \bmod N)$?
- Consider $M_x : y \mapsto x * y \bmod N$. Let $o = \text{ORDER}(x \bmod N)$.
 - M_x is invertible on \mathbb{Z}_N^* .
 - M_x 's cycles on \mathbb{Z}_N^* are of length o .

$$1 \xrightarrow{M_x} x \xrightarrow{M_x} x^2 \xrightarrow{M_x} \dots \xrightarrow{M_x} x^{o-1} \xrightarrow{M_x} 1$$

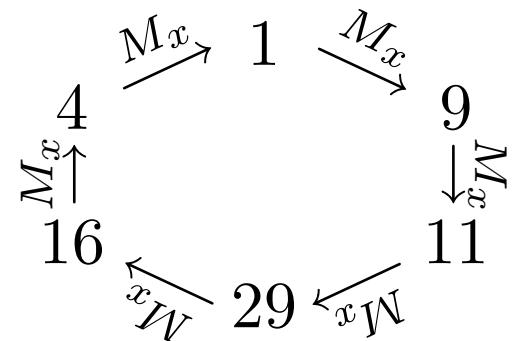


Quantum Order Finding

- **Problem:** Given $N, x, (x, N) = 1$. $\text{ORDER}(x \bmod N)$?
- Consider $M_x : y \mapsto x * y \bmod N$. Let $o = \text{ORDER}(x \bmod N)$.
 - M_x is invertible on \mathbb{Z}_N .
 - M_x 's cycles on \mathbb{Z}_N^* are of length o .

$$1 \xrightarrow{M_x} x \xrightarrow{M_x} x^2 \xrightarrow{M_x} \dots \xrightarrow{M_x} x^{o-1} \xrightarrow{M_x} 1$$

Example: $N = 35, x = 9, o(x) = 6$.

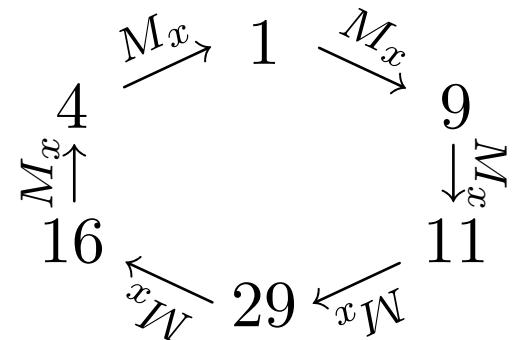


Quantum Order Finding

- **Problem:** Given $N, x, (x, N) = 1$. $\text{ORDER}(x \bmod N)$?
- Consider $M_x : y \mapsto x * y \bmod N$. Let $o = \text{ORDER}(x \bmod N)$.
 - M_x is invertible on \mathbb{Z}_N .
 - M_x 's cycles on \mathbb{Z}_N^* are of length o .

$$1 \xrightarrow{M_x} x \xrightarrow{M_x} x^2 \xrightarrow{M_x} \dots \xrightarrow{M_x} x^{o-1} \xrightarrow{M_x} 1$$

Example: $N = 35, x = 9, o(x) = 6$.



- $M_x^k = M_{x^k}$ is efficiently implementable. To compute x^k :

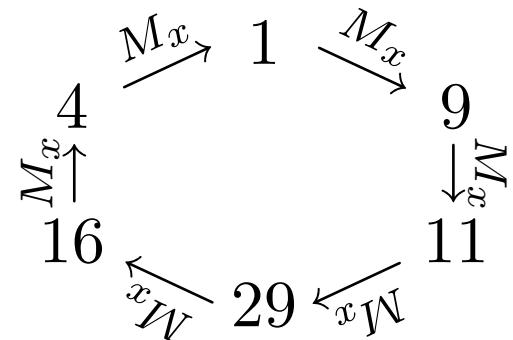


Quantum Order Finding

- **Problem:** Given $N, x, (x, N) = 1$. $\text{ORDER}(x \bmod N)$?
- Consider $M_x : y \mapsto x * y \bmod N$. Let $o = \text{ORDER}(x \bmod N)$.
 - M_x is invertible on \mathbb{Z}_N .
 - M_x 's cycles on \mathbb{Z}_N^* are of length o .

$$1 \xrightarrow{M_x} x \xrightarrow{M_x} x^2 \xrightarrow{M_x} \dots \xrightarrow{M_x} x^{o-1} \xrightarrow{M_x} 1$$

Example: $N = 35, x = 9, o(x) = 6$.



- $M_x^k = M_{x^k}$ is efficiently implementable. To compute x^k :
 1. Write $k = k_0 k_1 \dots k_b$ in reverse binary.

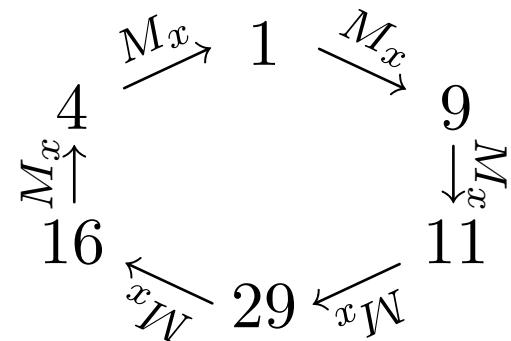


Quantum Order Finding

- **Problem:** Given $N, x, (x, N) = 1$. $\text{ORDER}(x \bmod N)$?
- Consider $M_x : y \mapsto x * y \bmod N$. Let $o = \text{ORDER}(x \bmod N)$.
 - M_x is invertible on \mathbb{Z}_N .
 - M_x 's cycles on \mathbb{Z}_N^* are of length o .

$$1 \xrightarrow{M_x} x \xrightarrow{M_x} x^2 \xrightarrow{M_x} \dots \xrightarrow{M_x} x^{o-1} \xrightarrow{M_x} 1$$

Example: $N = 35, x = 9, o(x) = 6$.



- $M_x^k = M_{x^k}$ is efficiently implementable. To compute x^k :
 1. Write $k = k_0 k_1 \dots k_b$ in reverse binary.
 2. Compute $x^{2^j} \bmod N, j = 0, \dots, b$ by repeated squaring.

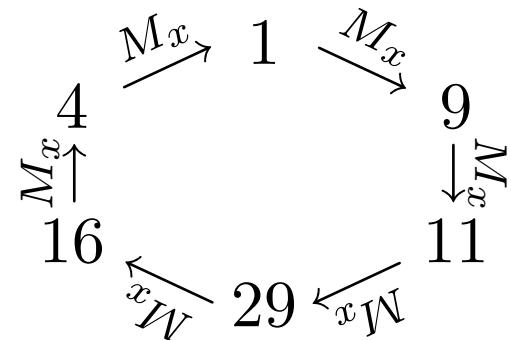


Quantum Order Finding

- **Problem:** Given $N, x, (x, N) = 1$. $\text{ORDER}(x \bmod N)$?
- Consider $M_x : y \mapsto x * y \bmod N$. Let $o = \text{ORDER}(x \bmod N)$.
 - M_x is invertible on \mathbb{Z}_N .
 - M_x 's cycles on \mathbb{Z}_N^* are of length o .

$$1 \xrightarrow{M_x} x \xrightarrow{M_x} x^2 \xrightarrow{M_x} \dots \xrightarrow{M_x} x^{o-1} \xrightarrow{M_x} 1$$

Example: $N = 35, x = 9, o(x) = 6$.



- $M_x^k = M_{x^k}$ is efficiently implementable. To compute x^k :
 1. Write $k = k_0 k_1 \dots k_b$ in reverse binary.
 2. Compute $x^{2^j} \bmod N, j = 0, \dots, b$ by repeated squaring.
 3. $x^k = x^{k_0 2^0} \dots x^{k_b 2^b}$.



Quantum Order Finding

- **Problem:** Given $N, x, (x, N) = 1$. $\text{ORDER}(x \bmod N)$?
- Consider $M_x : y \mapsto x * y \bmod N$. Let $o = \text{ORDER}(x \bmod N)$.
 - M_x is invertible on \mathbb{Z}_n^* , has cycles of length o and M_x^k is efficient.



Quantum Order Finding

- **Problem:** Given $N, x, (x, N) = 1$. $\text{ORDER}(x \bmod N)$?
- Consider $M_x : y \mapsto x * y \bmod N$. Let $o = \text{ORDER}(x \bmod N)$.
 - M_x is invertible on \mathbb{Z}_n^* , has cycles of length o and M_x^k is efficient.
- Quantum extension of M_x .
 - Let S be an n -qubit register,
Logical states: $|0\rangle_S, \dots, |2^n - 1\rangle_S$, $2^n > N$.



Quantum Order Finding

- **Problem:** Given $N, x, (x, N) = 1$. $\text{ORDER}(x \bmod N)$?
- Consider $M_x : y \mapsto x * y \bmod N$. Let $o = \text{ORDER}(x \bmod N)$.
 - M_x is invertible on \mathbb{Z}_n^* , has cycles of length o and M_x^k is efficient.
- Quantum extension of M_x .
 - Let S be an n -qubit register,
Logical states: $|0\rangle_S, \dots, |2^n - 1\rangle_S$, $2^n > N$.
 - $M_x|y\rangle_S = |x * y \bmod N\rangle_S$ for $0 \leq y < N$.
 $(M_x)^k \sum_{y=0}^{N-1} \alpha_y |y\rangle_S = \sum_{y=0}^{N-1} \alpha_y |x^k * y \bmod N\rangle_S$.



Quantum Order Finding

- **Problem:** Given $N, x, (x, N) = 1$. $\text{ORDER}(x \bmod N)$?
- Consider $M_x : y \mapsto x * y \bmod N$. Let $o = \text{ORDER}(x \bmod N)$.
 - M_x is invertible on \mathbb{Z}_n^* , has cycles of length o and M_x^k is efficient.
- Quantum extension of M_x .
 - Let S be an n -qubit register,
Logical states: $|0\rangle_S, \dots, |2^n - 1\rangle_S$, $2^n > N$.
 - $M_x|y\rangle_S = |x * y \bmod N\rangle_S$ for $0 \leq y < N$.
 $(M_x)^k \sum_{y=0}^{N-1} \alpha_y |y\rangle_S = \sum_{y=0}^{N-1} \alpha_y |x^k * y \bmod N\rangle_S$.
- Eigenvalues of M_x ?



Quantum Order Finding

- **Problem:** Given $N, x, (x, N) = 1$. $\text{ORDER}(x \bmod N)$?
- Consider $M_x : y \mapsto x * y \bmod N$. Let $o = \text{ORDER}(x \bmod N)$.
 - M_x is invertible on \mathbb{Z}_n^* , has cycles of length o and M_x^k is efficient.
- Quantum extension of M_x .
 - Let S be an n -qubit register,
Logical states: $|0\rangle_S, \dots, |2^n - 1\rangle_S$, $2^n > N$.
 - $M_x|y\rangle_S = |x * y \bmod N\rangle_S$ for $0 \leq y < N$.
 $(M_x)^k \sum_{y=0}^{N-1} \alpha_y |y\rangle_S = \sum_{y=0}^{N-1} \alpha_y |x^k * y \bmod N\rangle_S$.
- Eigenvalues of M_x ?
Let $\omega = e^{i\frac{2\pi}{o}}$. Abbreviate $|m\rangle = |m \bmod N\rangle_S$ and define
 $|\psi_l\rangle_S = \sum_{k=0}^{o-1} (\omega^{-l})^k |x^k\rangle$... normalization of $\frac{1}{\sqrt{o}}$ omitted.

Quantum Order Finding

- **Problem:** Given $N, x, (x, N) = 1$. $\text{ORDER}(x \bmod N)$?
- Consider $M_x : y \mapsto x * y \bmod N$. Let $o = \text{ORDER}(x \bmod N)$.
 - M_x is invertible on \mathbb{Z}_n^* , has cycles of length o and M_x^k is efficient.
- Quantum extension of M_x .
 - Let S be an n -qubit register,
Logical states: $|0\rangle_S, \dots, |2^n - 1\rangle_S$, $2^n > N$.
 - $M_x|y\rangle_S = |x * y \bmod N\rangle_S$ for $0 \leq y < N$.
 $(M_x)^k \sum_{y=0}^{N-1} \alpha_y |y\rangle_S = \sum_{y=0}^{N-1} \alpha_y |x^k * y \bmod N\rangle_S$.
- Eigenvalues of M_x ?
Let $\omega = e^{i\frac{2\pi}{o}}$. Abbreviate $|m\rangle = |m \bmod N\rangle_S$ and define
 $|\psi_l\rangle_S = \sum_{k=0}^{o-1} (\omega^{-l})^k |x^k\rangle$... normalization of $\frac{1}{\sqrt{o}}$ omitted.
 $M_x|\psi_l\rangle = M_x(|x^0\rangle + \omega^{-l}|x^1\rangle + \dots + \omega^{-l(o-2)}|x^{o-2}\rangle + \omega^{-l(o-1)}|x^{o-1}\rangle)$



Quantum Order Finding

- **Problem:** Given $N, x, (x, N) = 1$. $\text{ORDER}(x \bmod N)$?
- Consider $M_x : y \mapsto x * y \bmod N$. Let $o = \text{ORDER}(x \bmod N)$.
 - M_x is invertible on \mathbb{Z}_n^* , has cycles of length o and M_x^k is efficient.
- Quantum extension of M_x .
 - Let S be an n -qubit register,
Logical states: $|0\rangle_S, \dots, |2^n - 1\rangle_S$, $2^n > N$.
 - $M_x|y\rangle_S = |x * y \bmod N\rangle_S$ for $0 \leq y < N$.
 $(M_x)^k \sum_{y=0}^{N-1} \alpha_y |y\rangle_S = \sum_{y=0}^{N-1} \alpha_y |x^k * y \bmod N\rangle_S$.
- Eigenvalues of M_x ?
Let $\omega = e^{i\frac{2\pi}{o}}$. Abbreviate $|m\rangle = |m \bmod N\rangle_S$ and define
 $|\psi_l\rangle_S = \sum_{k=0}^{o-1} (\omega^{-l})^k |x^k\rangle$... normalization of $\frac{1}{\sqrt{o}}$ omitted.
$$M_x|\psi_l\rangle = M_x \left(|x^0\rangle + \omega^{-l}|x^1\rangle + \dots + \omega^{-l(o-2)}|x^{o-2}\rangle + \omega^{-l(o-1)}|x^{o-1}\rangle \right)$$
$$= |x^1\rangle + \dots + \omega^{-l(o-2)}|x^{o-1}\rangle + \omega^{-l(o-1)}|x^o\rangle$$



Quantum Order Finding

- **Problem:** Given $N, x, (x, N) = 1$. $\text{ORDER}(x \bmod N)$?
- Consider $M_x : y \mapsto x * y \bmod N$. Let $o = \text{ORDER}(x \bmod N)$.
 - M_x is invertible on \mathbb{Z}_n^* , has cycles of length o and M_x^k is efficient.
- Quantum extension of M_x .
 - Let S be an n -qubit register,
 - Logical states: $|0\rangle_S, \dots, |2^n - 1\rangle_S, 2^n > N$.
 - $M_x|y\rangle_S = |x * y \bmod N\rangle_S$ for $0 \leq y < N$.
 $(M_x)^k \sum_{y=0}^{N-1} \alpha_y |y\rangle_S = \sum_{y=0}^{N-1} \alpha_y |x^k * y \bmod N\rangle_S$.

- Eigenvalues of M_x ?

Let $\omega = e^{i\frac{2\pi}{o}}$. Abbreviate $|m\rangle = |m \bmod N\rangle_S$ and define

$$|\psi_l\rangle_S = \sum_{k=0}^{o-1} (\omega^{-l})^k |x^k\rangle \quad \dots \text{normalization of } \frac{1}{\sqrt{o}} \text{ omitted.}$$

$$\begin{aligned} M_x |\psi_l\rangle_S &= M_x \left(|x^0\rangle + \omega^{-l} |x^1\rangle + \dots + \omega^{-l(o-2)} |x^{o-2}\rangle + \omega^{-l(o-1)} |x^{o-1}\rangle \right) \\ &= \quad \quad \quad |x^1\rangle + \dots \quad \quad \quad + \omega^{-l(o-2)} |x^{o-1}\rangle + \omega^{-l(o-1)} |x^o\rangle \\ &= \omega^l \left(\quad \quad \omega^{-l} |x^1\rangle + \dots \quad \quad \quad + \omega^{-l(o-1)} |x^{o-1}\rangle + \quad \quad \omega^{-lo} |x^o\rangle \right) \end{aligned}$$



Quantum Order Finding

- **Problem:** Given $N, x, (x, N) = 1$. ORDER($x \bmod N$)?
- Consider $M_x : y \mapsto x * y \bmod N$. Let $o = \text{ORDER}(x \bmod N)$.
 - M_x is invertible on \mathbb{Z}_n^* , has cycles of length o and M_x^k is efficient.
- Quantum extension of M_x .
 - Let S be an n -qubit register,
Logical states: $|0\rangle_S, \dots, |2^n - 1\rangle_S$, $2^n > N$.
 - $M_x|y\rangle_S = |x * y \bmod N\rangle_S$ for $0 \leq y < N$.
 $(M_x)^k \sum_{y=0}^{N-1} \alpha_y |y\rangle_S = \sum_{y=0}^{N-1} \alpha_y |x^k * y \bmod N\rangle_S$.
- Eigenvalues of M_x ?
Let $\omega = e^{i\frac{2\pi}{o}}$. Abbreviate $|m\rangle = |m \bmod N\rangle_S$ and define

$$|\psi_l\rangle_S = \sum_{k=0}^{o-1} (\omega^{-l})^k |x^k\rangle \quad \dots \text{ normalization of } \frac{1}{\sqrt{o}} \text{ omitted.}$$

$$M_x |\psi_l\rangle$$

$$\begin{aligned} &= |x^1\rangle + \dots + \omega^{-l(o-2)} |x^{o-1}\rangle + \omega^{-l(o-1)} |x^o\rangle \\ &= \omega^l \left(\omega^{-l} |x^1\rangle + \dots + \omega^{-l(o-1)} |x^{o-1}\rangle + \omega^{-lo} |x^o\rangle \right) \end{aligned}$$

Quantum Order Finding

- **Problem:** Given $N, x, (x, N) = 1$. ORDER($x \bmod N$)?
- Consider $M_x : y \mapsto x * y \bmod N$. Let $o = \text{ORDER}(x \bmod N)$.
 - M_x is invertible on \mathbb{Z}_n^* , has cycles of length o and M_x^k is efficient.
- Quantum extension of M_x .
 - Let S be an n -qubit register,
Logical states: $|0\rangle_S, \dots, |2^n - 1\rangle_S$, $2^n > N$.
 - $M_x|y\rangle_S = |x * y \bmod N\rangle_S$ for $0 \leq y < N$.
 $(M_x)^k \sum_{y=0}^{N-1} \alpha_y |y\rangle_S = \sum_{y=0}^{N-1} \alpha_y |x^k * y \bmod N\rangle_S$.

- Eigenvalues of M_x ?

Let $\omega = e^{i\frac{2\pi}{o}}$. Abbreviate $|m\rangle = |m \bmod N\rangle_S$ and define

$$|\psi_l\rangle_S = \sum_{k=0}^{o-1} (\omega^{-l})^k |x^k\rangle \quad \dots \text{normalization of } \frac{1}{\sqrt{o}} \text{ omitted.}$$

$$\begin{aligned} M_x |\psi_l\rangle &= |x^1\rangle + \dots + \omega^{-l(o-2)} |x^{o-1}\rangle + \omega^{-l(o-1)} |x^o\rangle \\ &= \omega^l \left(\omega^{-l} |x^1\rangle + \dots + \omega^{-l(o-1)} |x^{o-1}\rangle + \omega^{-lo} |x^o\rangle \right) \end{aligned}$$



Quantum Order Finding

- **Problem:** Given $N, x, (x, N) = 1$. $\text{ORDER}(x \bmod N)$?
- Consider $M_x : y \mapsto x * y \bmod N$. Let $o = \text{ORDER}(x \bmod N)$.
 - M_x is invertible on \mathbb{Z}_n^* , has cycles of length o and M_x^k is efficient.
- Quantum extension of M_x .
 - Let S be an n -qubit register,
 - Logical states: $|0\rangle_S, \dots, |2^n - 1\rangle_S, 2^n > N$.
 - $M_x|y\rangle_S = |x * y \bmod N\rangle_S$ for $0 \leq y < N$.
 $(M_x)^k \sum_{y=0}^{N-1} \alpha_y |y\rangle_S = \sum_{y=0}^{N-1} \alpha_y |x^k * y \bmod N\rangle_S$.

- Eigenvalues of M_x ?

Let $\omega = e^{i\frac{2\pi}{o}}$. Abbreviate $|m\rangle = |m \bmod N\rangle_S$ and define

$$|\psi_l\rangle_S = \sum_{k=0}^{o-1} (\omega^{-l})^k |x^k\rangle \quad \dots \text{normalization of } \frac{1}{\sqrt{o}} \text{ omitted.}$$

$$\begin{aligned} M_x |\psi_l\rangle &= |x^1\rangle + \dots + \omega^{-l(o-2)} |x^{o-1}\rangle + \omega^{-l(o-1)} |x^o\rangle \\ &= \omega^l \left(\omega^{-l} |x^1\rangle + \dots + \omega^{-l(o-1)} |x^{o-1}\rangle + \omega^{-lo} |x^o\rangle \right) \\ &= \omega^l \left(|x^0\rangle + \omega^{-l} |x^1\rangle + \dots + \omega^{-l(o-1)} |x^{o-1}\rangle \right) \end{aligned}$$



Quantum Order Finding

- **Problem:** Given $N, x, (x, N) = 1$. $\text{ORDER}(x \bmod N)$?
- Consider $M_x : y \mapsto x * y \bmod N$. Let $o = \text{ORDER}(x \bmod N)$.
 - M_x is invertible on \mathbb{Z}_n^* , has cycles of length o and M_x^k is efficient.
- Quantum extension of M_x .
 - Let S be an n -qubit register,
 - Logical states: $|0\rangle_S, \dots, |2^n - 1\rangle_S$, $2^n > N$.
 - $M_x|y\rangle_S = |x * y \bmod N\rangle_S$ for $0 \leq y < N$.

$$(M_x)^k \sum_{y=0}^{N-1} \alpha_y |y\rangle_S = \sum_{y=0}^{N-1} \alpha_y |x^k * y \bmod N\rangle_S.$$

- Eigenvalues of M_x ?

Let $\omega = e^{i\frac{2\pi}{o}}$. Abbreviate $|m\rangle = |m \bmod N\rangle_S$ and define

$$|\psi_l\rangle_S = \sum_{k=0}^{o-1} (\omega^{-l})^k |x^k\rangle \quad \dots \text{normalization of } \frac{1}{\sqrt{o}} \text{ omitted.}$$

$$M_x |\psi_l\rangle$$

$$\begin{aligned} &= \omega^l \left(\omega^{-l} |x^1\rangle + \dots + \omega^{-l(o-1)} |x^{o-1}\rangle + \omega^{-lo} |x^o\rangle \right) \\ &= \omega^l \left(|x^0\rangle + \omega^{-l} |x^1\rangle + \dots + \omega^{-l(o-1)} |x^{o-1}\rangle \right) \end{aligned}$$



Quantum Order Finding

- **Problem:** Given $N, x, (x, N) = 1$. $\text{ORDER}(x \bmod N)$?
- Consider $M_x : y \mapsto x * y \bmod N$. Let $o = \text{ORDER}(x \bmod N)$.
 - M_x is invertible on \mathbb{Z}_n^* , has cycles of length o and M_x^k is efficient.
- Quantum extension of M_x .
 - Let S be an n -qubit register,
Logical states: $|0\rangle_S, \dots, |2^n - 1\rangle_S$, $2^n > N$.
 - $M_x|y\rangle_S = |x * y \bmod N\rangle_S$ for $0 \leq y < N$.
 $(M_x)^k \sum_{y=0}^{N-1} \alpha_y |y\rangle_S = \sum_{y=0}^{N-1} \alpha_y |x^k * y \bmod N\rangle_S$.

- Eigenvalues of M_x ?

Let $\omega = e^{i\frac{2\pi}{o}}$. Abbreviate $|m\rangle = |m \bmod N\rangle_S$ and define

$$|\psi_l\rangle_S = \sum_{k=0}^{o-1} (\omega^{-l})^k |x^k\rangle \quad \dots \text{normalization of } \frac{1}{\sqrt{o}} \text{ omitted.}$$

$$\begin{aligned} M_x |\psi_l\rangle &= \omega^l \left(\omega^{-l} |x^1\rangle + \dots + \omega^{-l(o-1)} |x^{o-1}\rangle + \omega^{-lo} |x^o\rangle \right) \\ &= \omega^l \left(|x^0\rangle + \omega^{-l} |x^1\rangle + \dots + \omega^{-l(o-1)} |x^{o-1}\rangle \right) \end{aligned}$$



Quantum Order Finding

- **Problem:** Given $N, x, (x, N) = 1$. $\text{ORDER}(x \bmod N)$?
- Consider $M_x : y \mapsto x * y \bmod N$. Let $o = \text{ORDER}(x \bmod N)$.
 - M_x is invertible on \mathbb{Z}_n^* , has cycles of length o and M_x^k is efficient.
- Quantum extension of M_x .
 - Let S be an n -qubit register,
Logical states: $|0\rangle_S, \dots, |2^n - 1\rangle_S$, $2^n > N$.
 - $M_x|y\rangle_S = |x * y \bmod N\rangle_S$ for $0 \leq y < N$.
 $(M_x)^k \sum_{y=0}^{N-1} \alpha_y |y\rangle_S = \sum_{y=0}^{N-1} \alpha_y |x^k * y \bmod N\rangle_S$.
- Eigenvalues of M_x ?

Let $\omega = e^{i\frac{2\pi}{o}}$. Abbreviate $|m\rangle = |m \bmod N\rangle_S$ and define

$$|\psi_l\rangle_S = \sum_{k=0}^{o-1} (\omega^{-l})^k |x^k\rangle \quad \dots \text{normalization of } \frac{1}{\sqrt{o}} \text{ omitted.}$$

$$\begin{aligned} M_x |\psi_l\rangle &= \omega^l \left(\omega^{-l} |x^1\rangle + \dots + \omega^{-l(o-1)} |x^{o-1}\rangle + \omega^{-lo} |x^o\rangle \right) \\ &= \omega^l \left(|x^0\rangle + \omega^{-l} |x^1\rangle + \dots + \omega^{-l(o-1)} |x^{o-1}\rangle \right) \\ &= \omega^l |\psi_l\rangle \end{aligned}$$



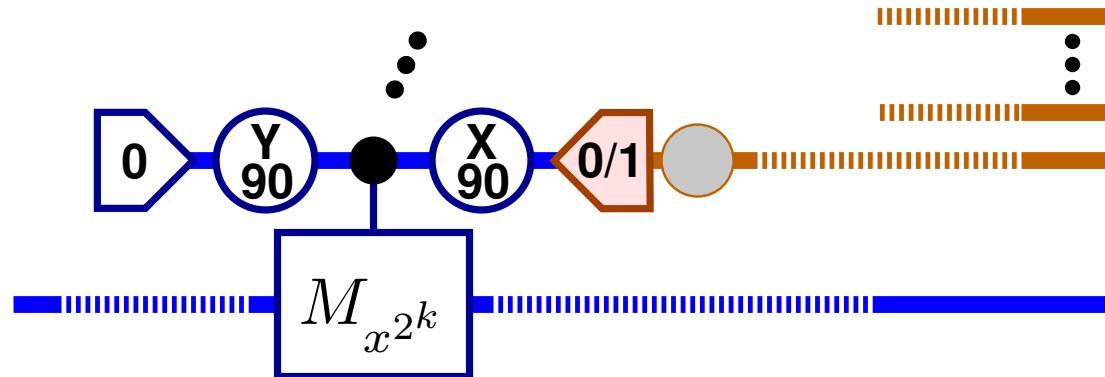
From Eigenvalues to Order

- Given operators $M_{x^k}|y\rangle = |x^k * y\rangle$, $M_{x^k}|\psi_l\rangle = e^{i\frac{2\pi lk}{o}}$. Determine o ?



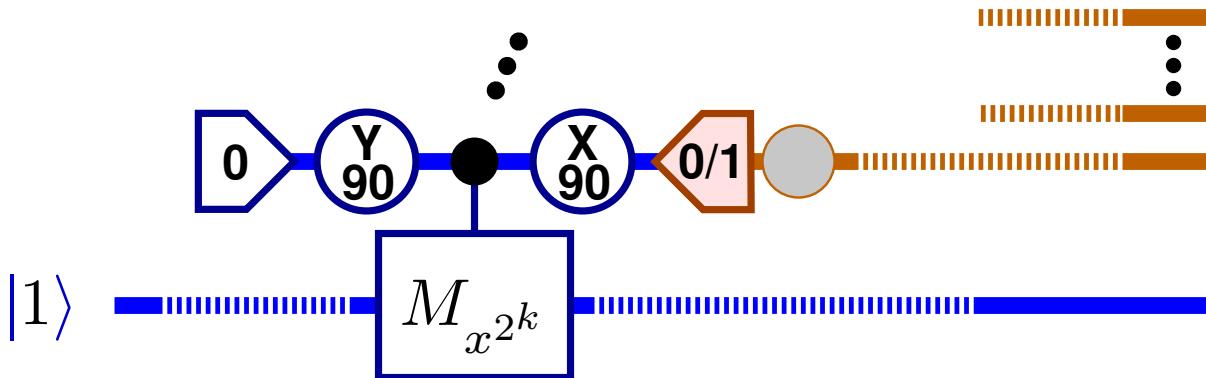
From Eigenvalues to Order

- Given operators $M_{x^k}|y\rangle = |x^k * y\rangle$, $M_{x^k}|\psi_l\rangle = e^{i\frac{2\pi lk}{o}}$. Determine o ?
- Use phase estimation with input state $|1\rangle$.



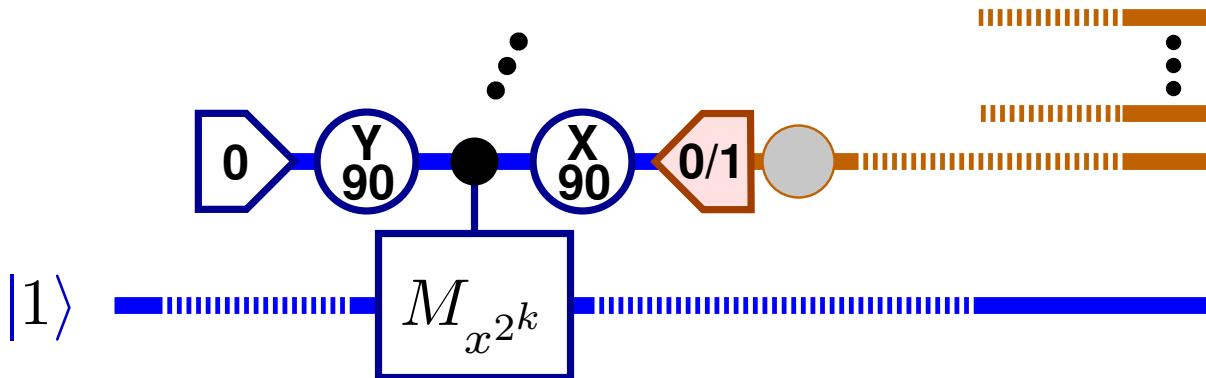
From Eigenvalues to Order

- Given operators $M_{x^k}|y\rangle = |x^k * y\rangle$, $M_{x^k}|\psi_l\rangle = e^{i\frac{2\pi lk}{o}}$. Determine o ?
- Use phase estimation with input state $|1\rangle$.



From Eigenvalues to Order

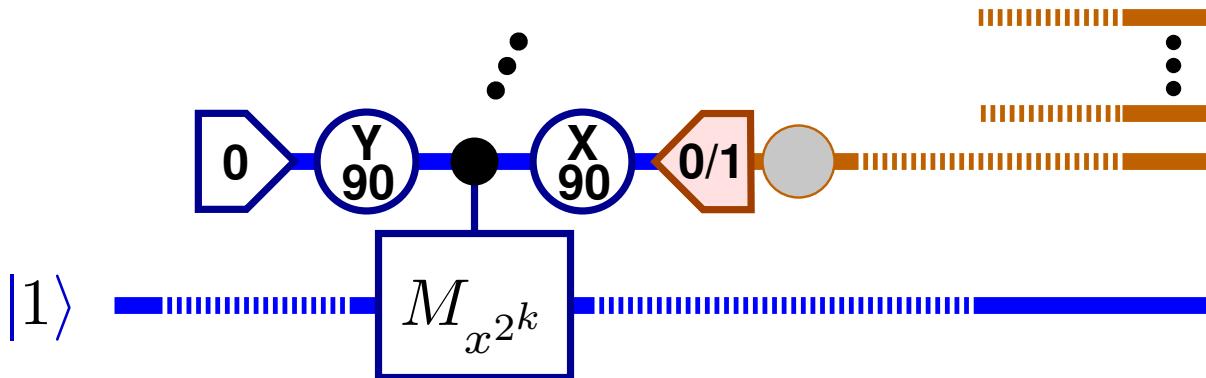
- Given operators $M_{x^k}|y\rangle = |x^k * y\rangle$, $M_{x^k}|\psi_l\rangle = e^{i\frac{2\pi lk}{o}}$. Determine o ?
- Use phase estimation with input state $|1\rangle$.



$$|1\rangle \stackrel{?}{=} \frac{1}{\sqrt{o}} \sum_{l=0}^{o-1} |\psi_l\rangle$$

From Eigenvalues to Order

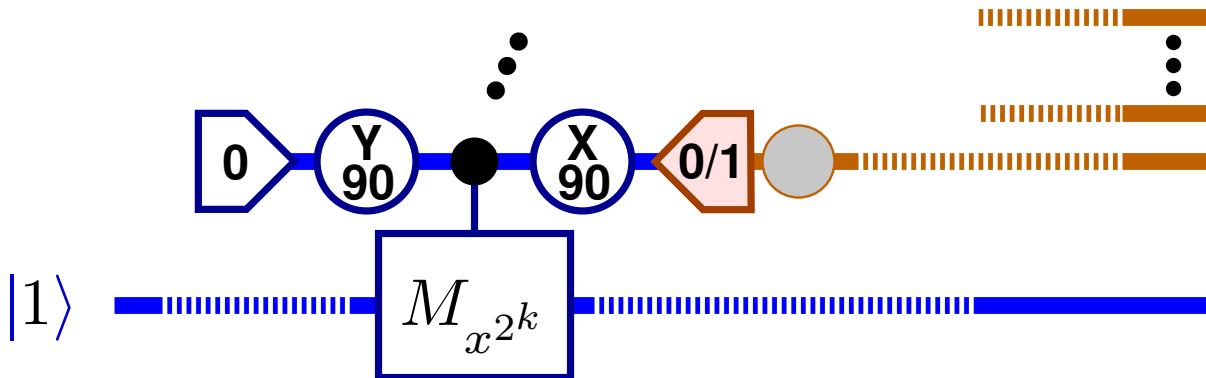
- Given operators $M_{x^k}|y\rangle = |x^k * y\rangle$, $M_{x^k}|\psi_l\rangle = e^{i\frac{2\pi lk}{o}}$. Determine o ?
- Use phase estimation with input state $|1\rangle$.



$$\begin{aligned} |1\rangle &\stackrel{?}{=} \frac{1}{\sqrt{o}} \sum_{l=0}^{o-1} |\psi_l\rangle \\ &= \frac{1}{o} \sum_{l=0}^{o-1} \sum_{k=0}^{o-1} \omega^{-lk} |k\rangle \end{aligned}$$

From Eigenvalues to Order

- Given operators $M_{x^k}|y\rangle = |x^k * y\rangle$, $M_{x^k}|\psi_l\rangle = e^{i\frac{2\pi lk}{o}}$. Determine o ?
- Use phase estimation with input state $|1\rangle$.

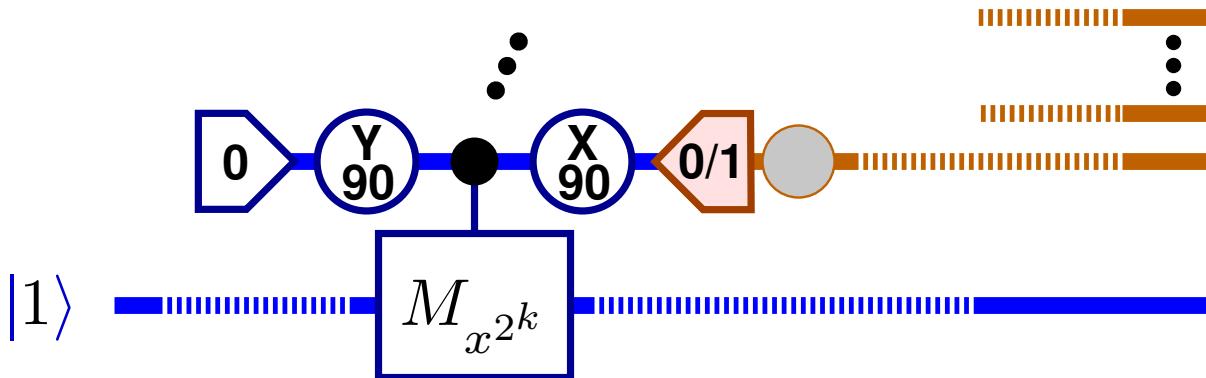


$$\begin{aligned} |1\rangle &\stackrel{?}{=} \frac{1}{\sqrt{o}} \sum_{l=0}^{o-1} |\psi_l\rangle \\ &= \frac{1}{o} \sum_{l=0}^{o-1} \sum_{k=0}^{o-1} \omega^{-lk} |k\rangle \\ &= \frac{1}{o} \sum_{k=0}^{o-1} |k\rangle \sum_{l=0}^{o-1} \omega^{-lk} \end{aligned}$$



From Eigenvalues to Order

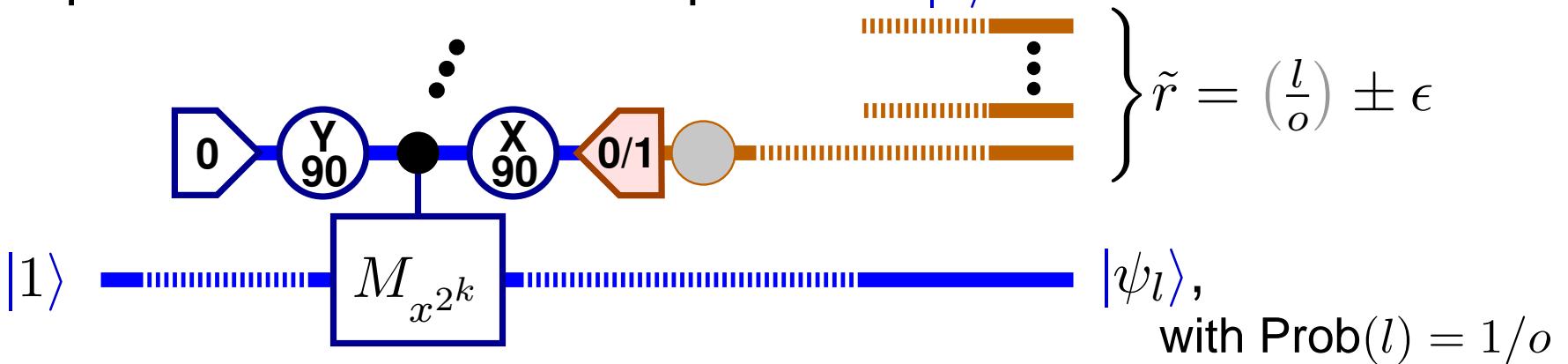
- Given operators $M_{x^k}|y\rangle = |x^k * y\rangle$, $M_{x^k}|\psi_l\rangle = e^{i\frac{2\pi lk}{o}}$. Determine o ?
- Use phase estimation with input state $|1\rangle$.



$$\begin{aligned} |1\rangle &\stackrel{?}{=} \frac{1}{\sqrt{o}} \sum_{l=0}^{o-1} |\psi_l\rangle \\ &= \frac{1}{o} \sum_{l=0}^{o-1} \sum_{k=0}^{o-1} \omega^{-lk} |k\rangle \\ &= \frac{1}{o} \sum_{k=0}^{o-1} |k\rangle \sum_{l=0}^{o-1} \omega^{-lk} \\ &= |1\rangle \end{aligned}$$

From Eigenvalues to Order

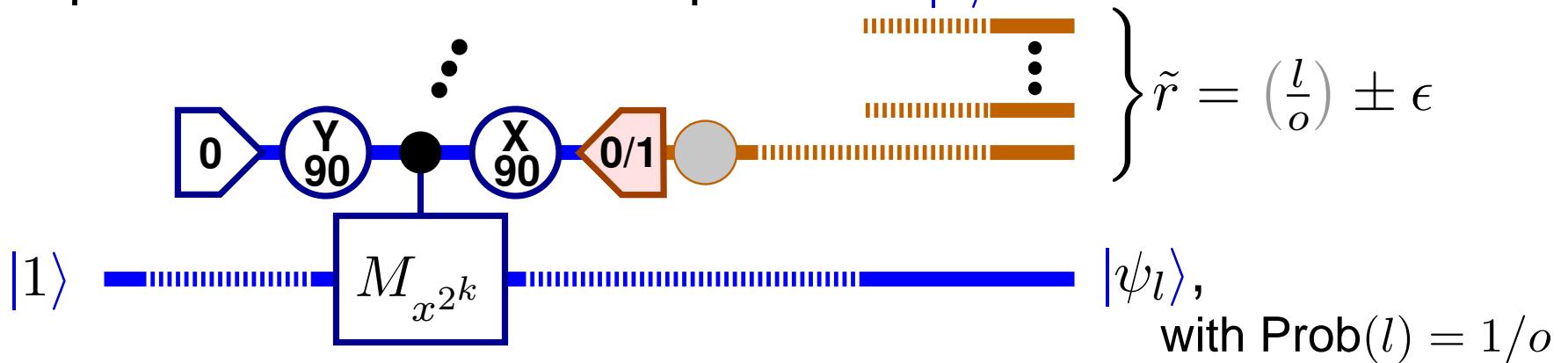
- Given operators $M_{x^k}|y\rangle = |x^k * y\rangle$, $M_{x^k}|\psi_l\rangle = e^{i\frac{2\pi lk}{o}}$. Determine o ?
- Use phase estimation with input state $|1\rangle$.



$$\begin{aligned}
 |1\rangle &\stackrel{?}{=} \frac{1}{\sqrt{o}} \sum_{l=0}^{o-1} |\psi_l\rangle \\
 &= \frac{1}{o} \sum_{l=0}^{o-1} \sum_{k=0}^{o-1} \omega^{-lk} |k\rangle \\
 &= \frac{1}{o} \sum_{k=0}^{o-1} |k\rangle \sum_{l=0}^{o-1} \omega^{-lk} \\
 &= |1\rangle
 \end{aligned}$$

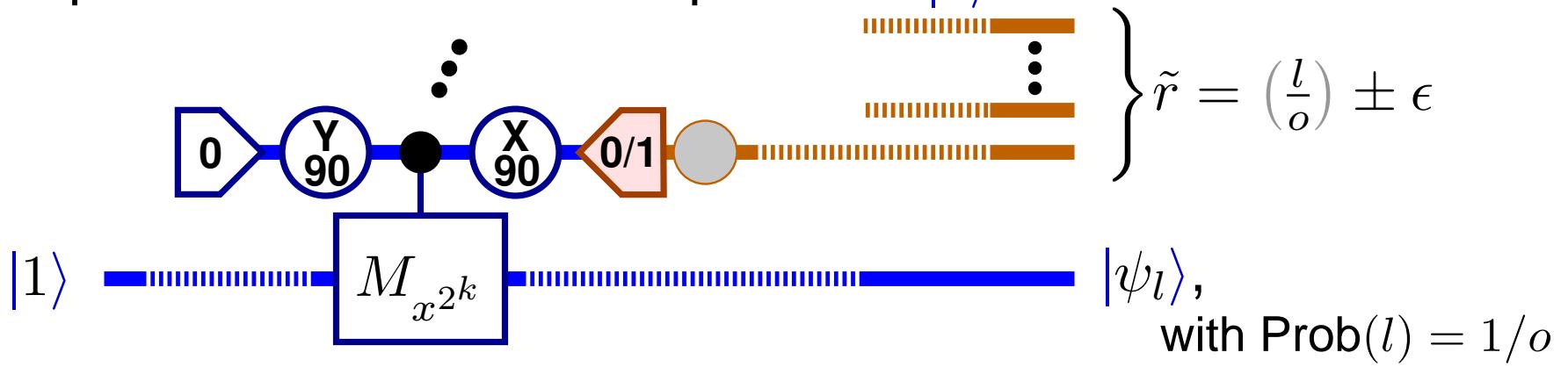
From Eigenvalues to Order

- Given operators $M_{x^k}|y\rangle = |x^k * y\rangle$, $M_{x^k}|\psi_l\rangle = e^{i\frac{2\pi lk}{o}}$. Determine o ?
- Use phase estimation with input state $|1\rangle$.



From Eigenvalues to Order

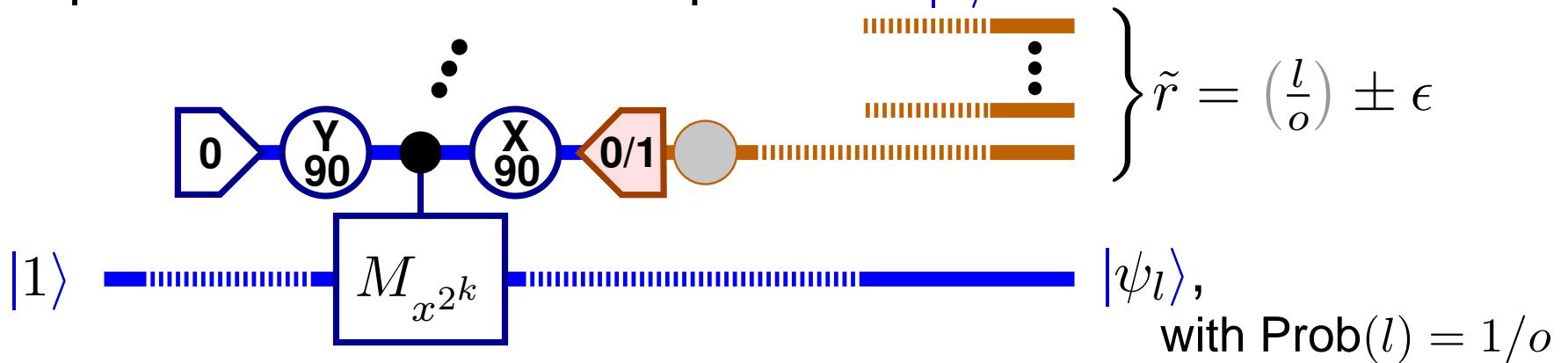
- Given operators $M_{x^k}|y\rangle = |x^k * y\rangle$, $M_{x^k}|\psi_l\rangle = e^{i\frac{2\pi lk}{o}}$. Determine o ?
- Use phase estimation with input state $|1\rangle$.



- Inferring o from \tilde{r} . Known: $1 < o < N$.

From Eigenvalues to Order

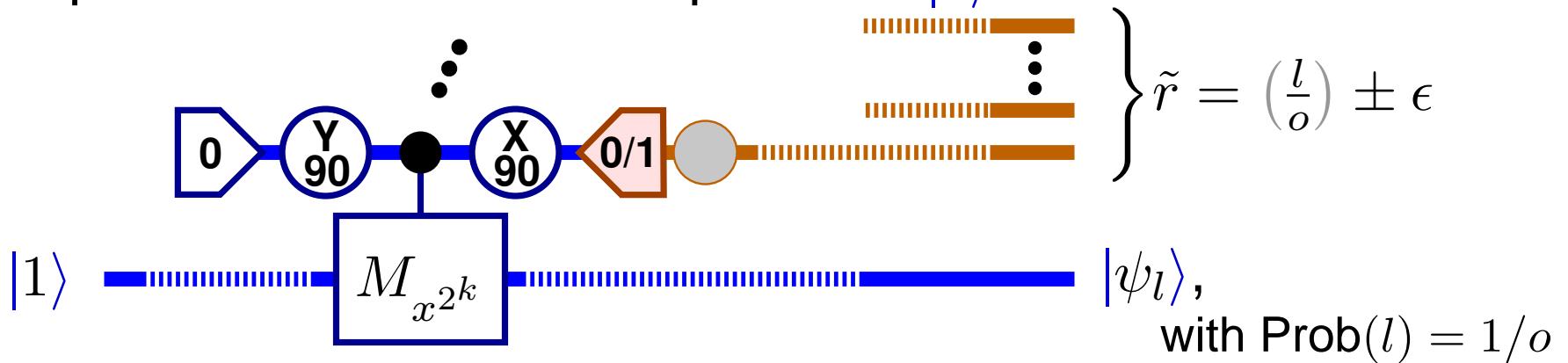
- Given operators $M_{x^k}|y\rangle = |x^k * y\rangle$, $M_{x^k}|\psi_l\rangle = e^{i\frac{2\pi lk}{o}}$. Determine o ?
- Use phase estimation with input state $|1\rangle$.



- Inferring o from \tilde{r} . Known: $1 < o < N$.
 - Determine p/q with $q < N$, $(p, q) = 1$, $|p/q - \tilde{r}|$ minimal.

From Eigenvalues to Order

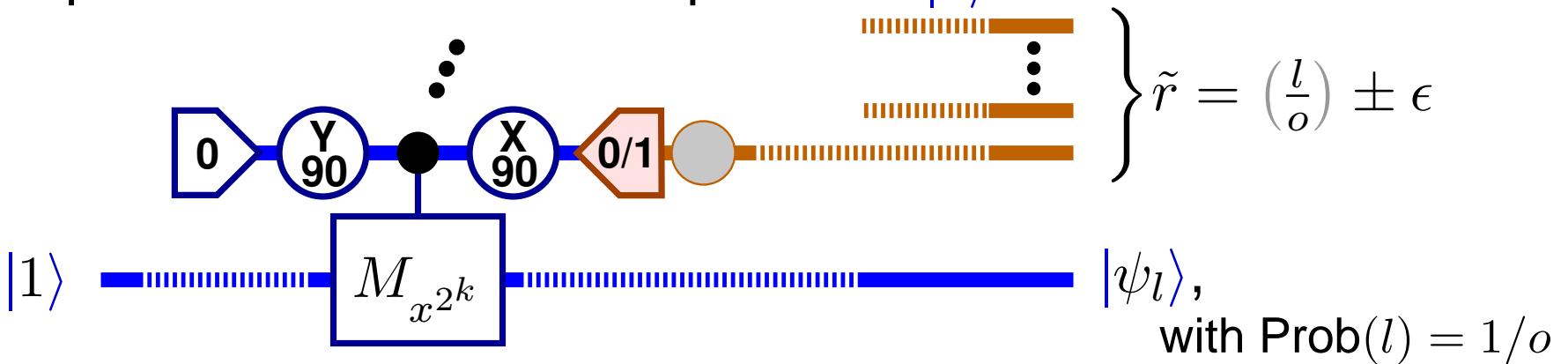
- Given operators $M_{x^k}|y\rangle = |x^k * y\rangle$, $M_{x^k}|\psi_l\rangle = e^{i\frac{2\pi lk}{o}}$. Determine o ?
- Use phase estimation with input state $|1\rangle$.



- Inferring o from \tilde{r} . Known: $1 < o < N$.
 - Determine p/q with $q < N$, $(p, q) = 1$, $|p/q - \tilde{r}|$ minimal.
 - Can be done efficiently by the *continued fraction algorithm*.

From Eigenvalues to Order

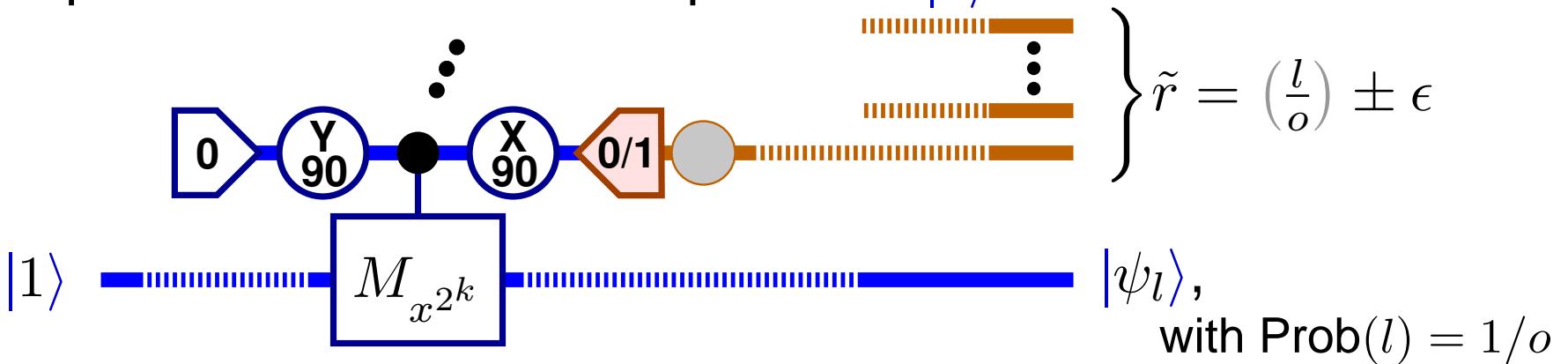
- Given operators $M_{x^k}|y\rangle = |x^k * y\rangle$, $M_{x^k}|\psi_l\rangle = e^{i\frac{2\pi lk}{o}}$. Determine o ?
- Use phase estimation with input state $|1\rangle$.



- Inferring o from \tilde{r} . Known: $1 < o < N$.
 - Determine p/q with $q < N$, $(p, q) = 1$, $|p/q - \tilde{r}|$ minimal.
 - Can be done efficiently by the *continued fraction algorithm*.
 - If $\epsilon < 1/N^2$, $q|o$ with high confidence.

From Eigenvalues to Order

- Given operators $M_{x^k}|y\rangle = |x^k * y\rangle$, $M_{x^k}|\psi_l\rangle = e^{i\frac{2\pi lk}{o}}$. Determine o ?
- Use phase estimation with input state $|1\rangle$.

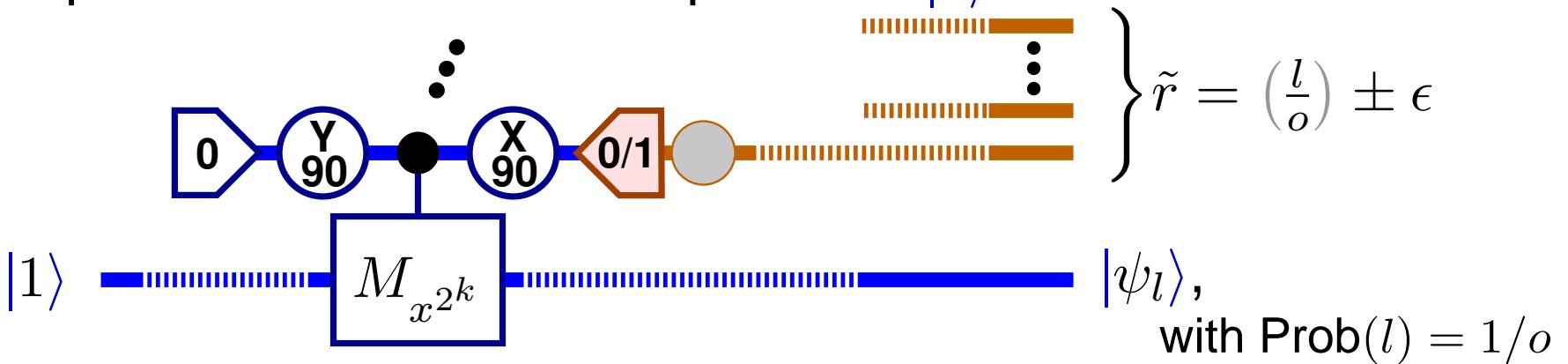


- Inferring o from \tilde{r} . Known: $1 < o < N$.
 - Determine p/q with $q < N$, $(p, q) = 1$, $|p/q - \tilde{r}|$ minimal.
 - Can be done efficiently by the *continued fraction algorithm*.
 - If $\epsilon < 1/N^2$, $q|o$ with high confidence.

$q = o$ if $(l, o) = 1$. This happens with probability $> c/\ln \ln(N)$.

From Eigenvalues to Order

- Given operators $M_{x^k}|y\rangle = |x^k * y\rangle$, $M_{x^k}|\psi_l\rangle = e^{i\frac{2\pi lk}{o}}$. Determine o ?
- Use phase estimation with input state $|1\rangle$.

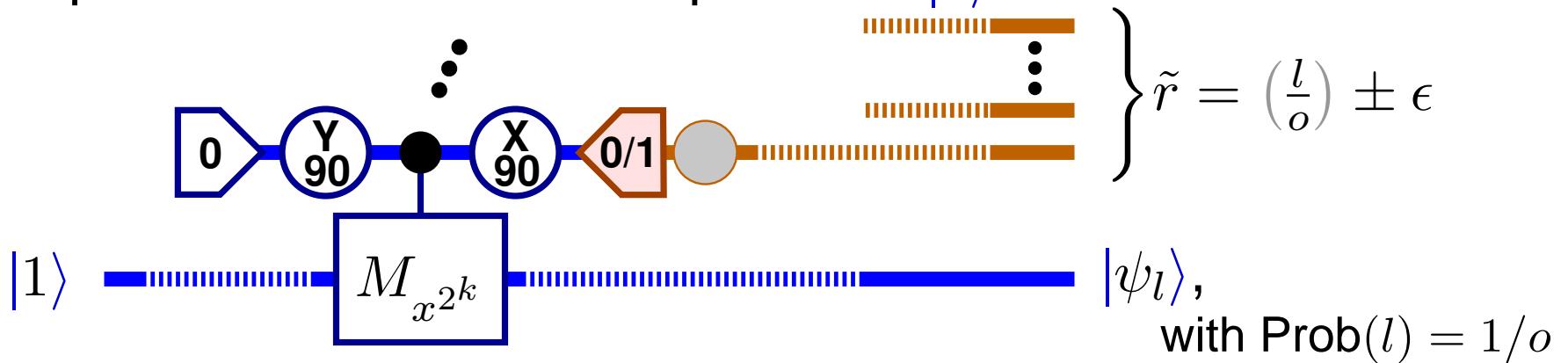


- Inferring o from \tilde{r} . Known: $1 < o < N$.
 - Determine p/q with $q < N$, $(p, q) = 1$, $|p/q - \tilde{r}|$ minimal.
 - Can be done efficiently by the *continued fraction algorithm*.
 - If $\epsilon < 1/N^2$, $q|o$ with high confidence.

$q = o$ if $(l, o) = 1$. This happens with probability $> c/\ln \ln(N)$.
 - Check whether q is the order of x .

From Eigenvalues to Order

- Given operators $M_{x^k}|y\rangle = |x^k * y\rangle$, $M_{x^k}|\psi_l\rangle = e^{i\frac{2\pi lk}{o}}$. Determine o ?
- Use phase estimation with input state $|1\rangle$.



- Inferring o from \tilde{r} . Known: $1 < o < N$.
 - Determine p/q with $q < N$, $(p, q) = 1$, $|p/q - \tilde{r}|$ minimal.
 - Can be done efficiently by the *continued fraction algorithm*.
 - If $\epsilon < 1/N^2$, $q|o$ with high confidence.
 $q = o$ if $(l, o) = 1$. This happens with probability $> c/\ln \ln(N)$.
 - Check whether q is the order of x .
 - If not, try phase estimation again.

A Phase Estimation Step

- $\text{PHASESTEP}(P, k)$

Input: $k \geq 1$, black box $P = e^{-ix\pi\sigma_z}$, x unknown, efficient P^{2^l} .

Output: 1. Modified black box $P' = e^{i\frac{b\pi}{2^k}\sigma_z}P$ for $b = 0$ or $b = 1$.
2. b , which approximates the k 'th bit of x .



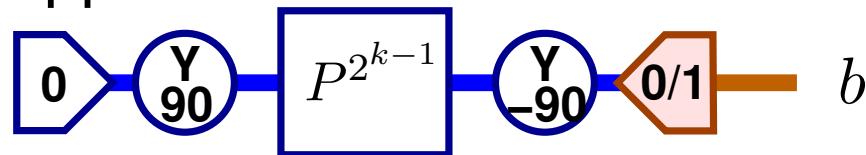
A Phase Estimation Step

- PHASESTEP(P, k)

Input: $k \geq 1$, black box $P = e^{-ix\pi\sigma_z}$, x unknown, efficient P^{2^l} .

Output: 1. Modified black box $P' = e^{i\frac{b\pi}{2^k}\sigma_z}P$ for $b = 0$ or $b = 1$.
2. b , which approximates the k 'th bit of x .

1. Implement



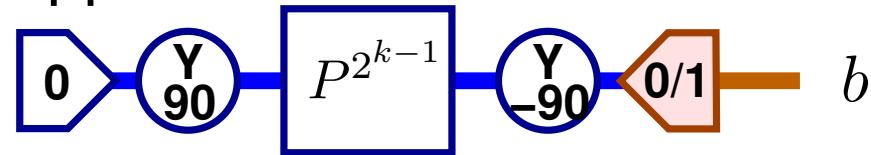
A Phase Estimation Step

- PHASESTEP(P, k)

Input: $k \geq 1$, black box $P = e^{-ix\pi\sigma_z}$, x unknown, efficient P^{2^l} .

Output: 1. Modified black box $P' = e^{i\frac{b\pi}{2^k}\sigma_z}P$ for $b = 0$ or $b = 1$.
2. b , which approximates the k 'th bit of x .

1. Implement



2. Return $P' = e^{i\frac{b\pi}{2^{k+1}}\sigma_z}P$.



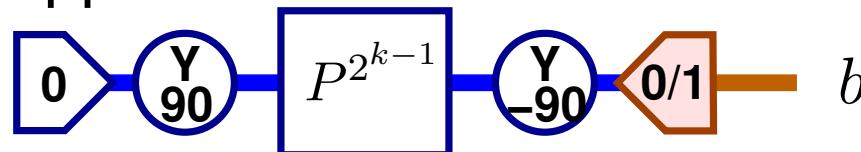
A Phase Estimation Step

- PHASESTEP(P, k)

Input: $k \geq 1$, black box $P = e^{-ix\pi\sigma_z}$, x unknown, efficient P^{2^l} .

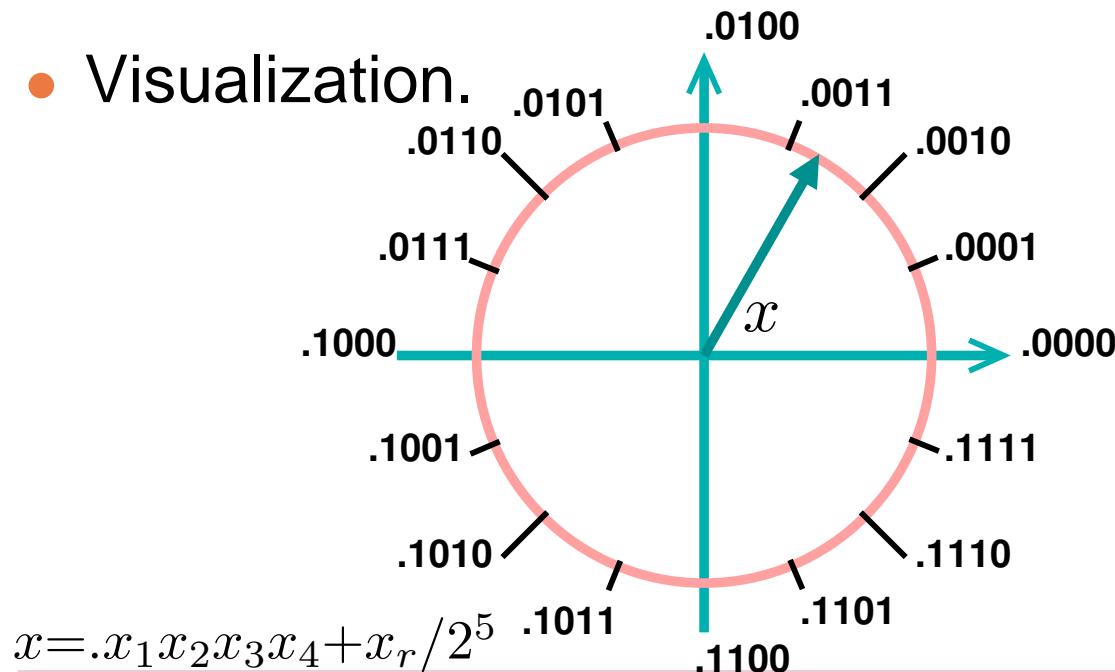
Output: 1. Modified black box $P' = e^{i\frac{b\pi}{2^k}\sigma_z}P$ for $b = 0$ or $b = 1$.
2. b , which approximates the k 'th bit of x .

1. Implement



2. Return $P' = e^{i\frac{b\pi}{2^{k+1}}\sigma_z}P$.

- Visualization.



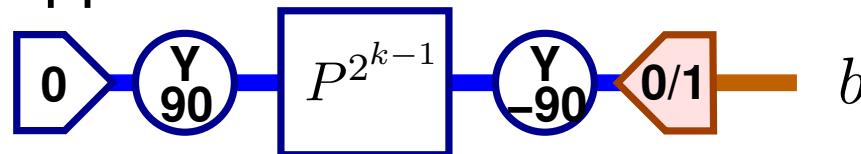
A Phase Estimation Step

- PHASESTEP(P, k)

Input: $k \geq 1$, black box $P = e^{-ix\pi\sigma_z}$, x unknown, efficient P^{2^l} .

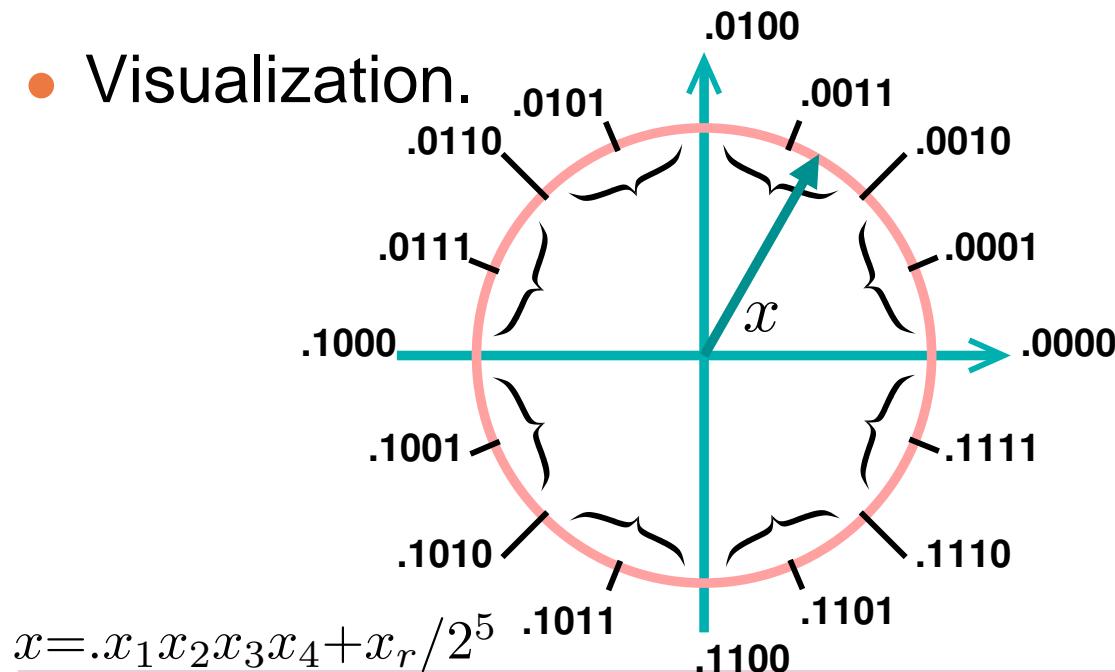
Output: 1. Modified black box $P' = e^{i\frac{b\pi}{2^k}\sigma_z}P$ for $b = 0$ or $b = 1$.
2. b , which approximates the k 'th bit of x .

1. Implement



2. Return $P' = e^{i\frac{b\pi}{2^{k+1}}\sigma_z}P$.

- Visualization.



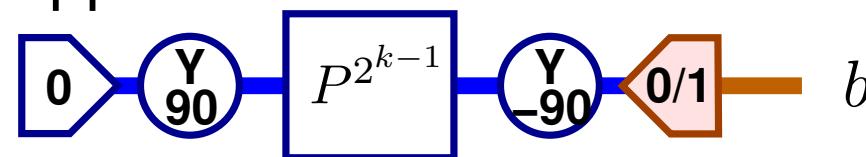
A Phase Estimation Step

- PHASESTEP(P, k)

Input: $k \geq 1$, black box $P = e^{-ix\pi\sigma_z}$, x unknown, efficient P^{2^l} .

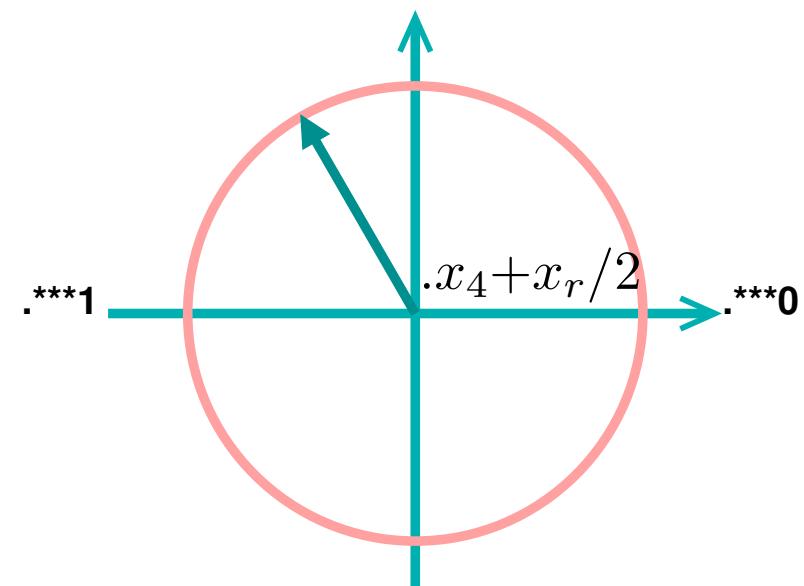
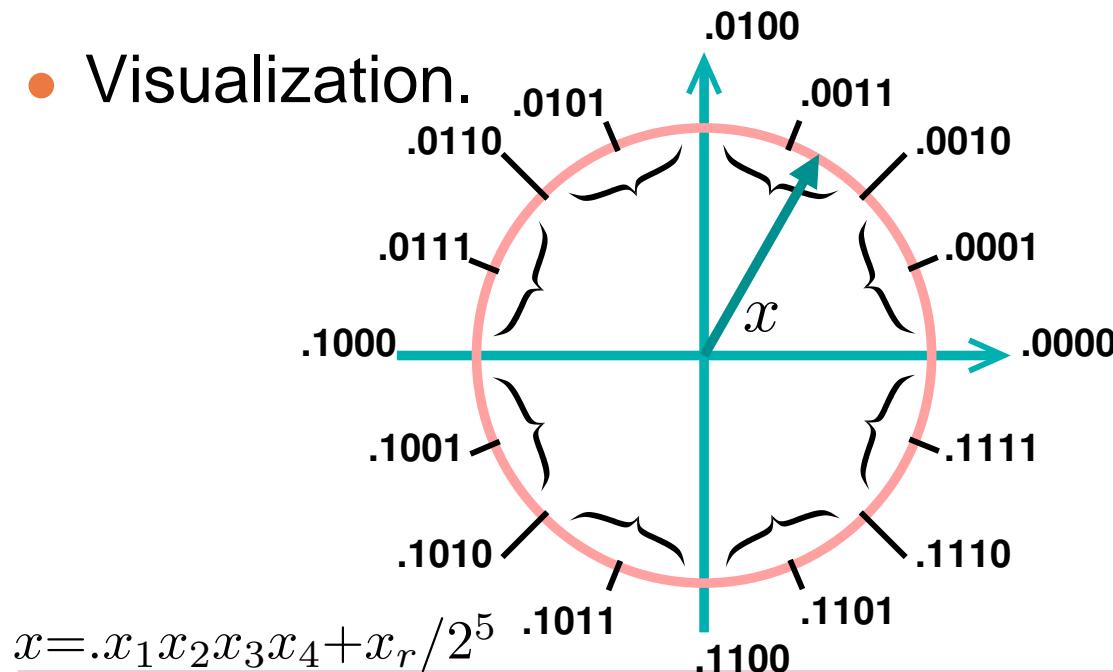
Output: 1. Modified black box $P' = e^{i\frac{b\pi}{2^k}\sigma_z}P$ for $b = 0$ or $b = 1$.
2. b , which approximates the k 'th bit of x .

1. Implement



2. Return $P' = e^{i\frac{b\pi}{2^{k+1}}\sigma_z}P$.

- Visualization.



Efficient Phase Estimation

PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

$P_k \leftarrow P$

for $l = k$ **to** 1

$(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$

end



Efficient Phase Estimation

PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

$P_k \leftarrow P$

for $l = k$ **to** 1

$(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$

end

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.



Efficient Phase Estimation

PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

$P_k \leftarrow P$

for $l = k$ **to** 1

$(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$

end

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.
 - Overall probability of optimal x_l for each l :



Efficient Phase Estimation

PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

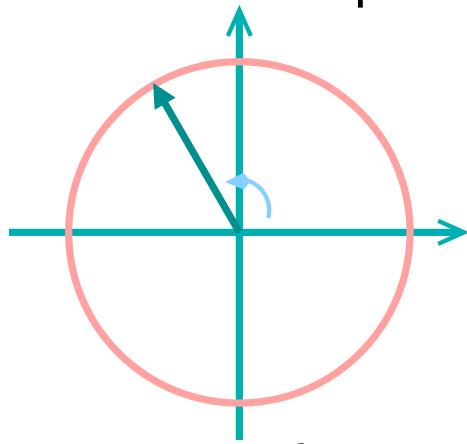
$$P_k \leftarrow P$$

for $l = k$ **to** 1

$$(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$$

end

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.
 - Overall probability of optimal x_l for each l :



Efficient Phase Estimation

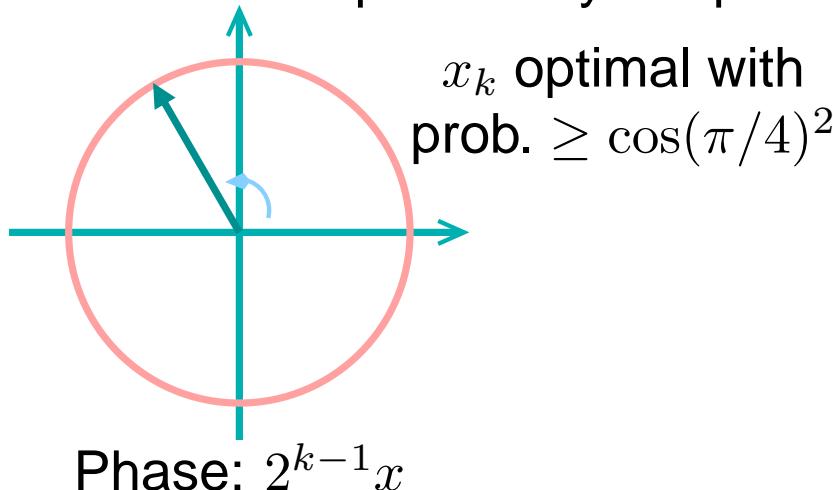
PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

```
 $P_k \leftarrow P$ 
for  $l = k$  to 1
     $(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$ 
end
```

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.
 - Overall probability of optimal x_l for each l :



Efficient Phase Estimation

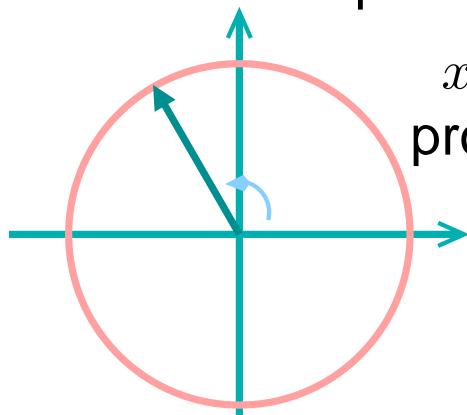
PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

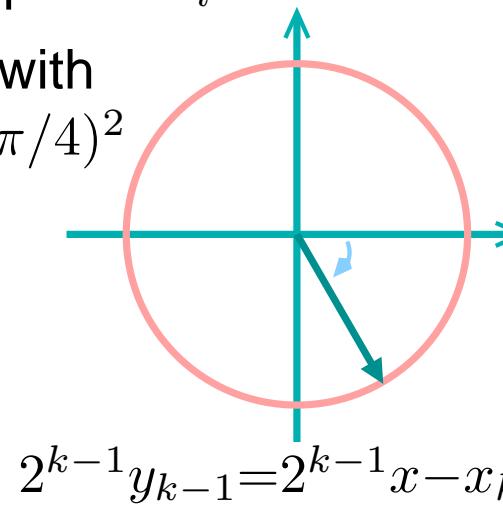
Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

```
 $P_k \leftarrow P$ 
for  $l = k$  to 1
     $(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$ 
end
```

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.
 - Overall probability of optimal x_l for each l :



x_k optimal with
prob. $\geq \cos(\pi/4)^2$



Efficient Phase Estimation

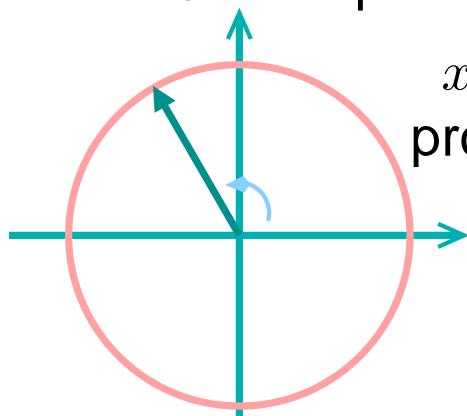
PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

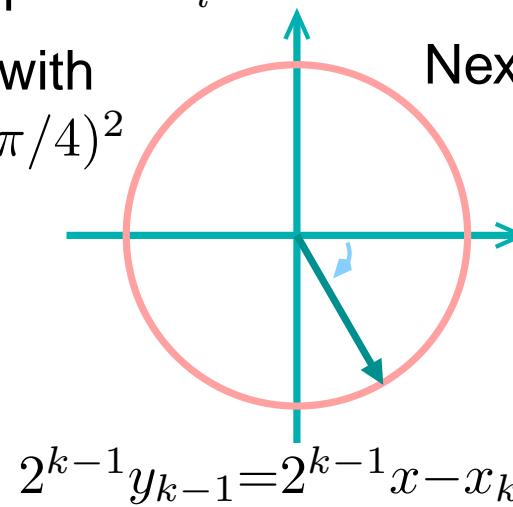
Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

```
 $P_k \leftarrow P$ 
for  $l = k$  to 1
     $(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$ 
end
```

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.
 - Overall probability of optimal x_l for each l :



x_k optimal with
prob. $\geq \cos(\pi/4)^2$



Next step, half angle:

Phase: $2^{k-1}x$

$2^{k-1}y_{k-1} = 2^{k-1}x - x_k/2$

Efficient Phase Estimation

PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

$P_k \leftarrow P$

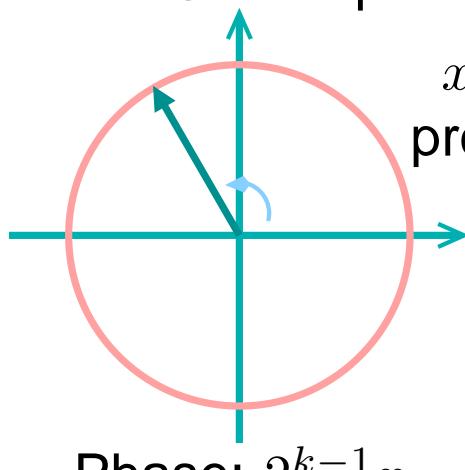
for $l = k$ **to** 1

$(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$

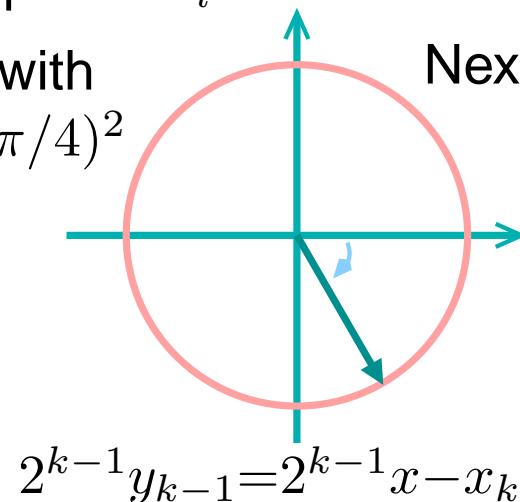
end

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.

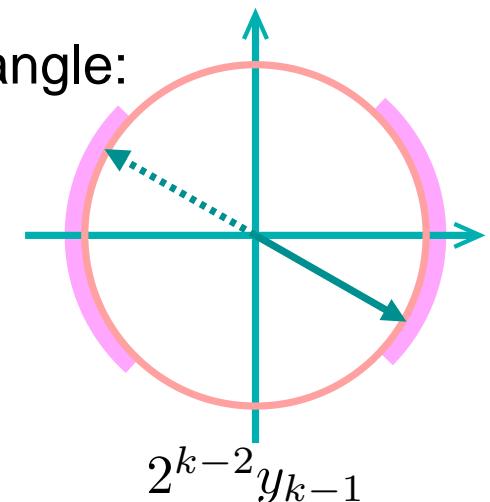
- Overall probability of optimal x_l for each l :



x_k optimal with
prob. $\geq \cos(\pi/4)^2$



Next step, half angle:



Efficient Phase Estimation

PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

$P_k \leftarrow P$

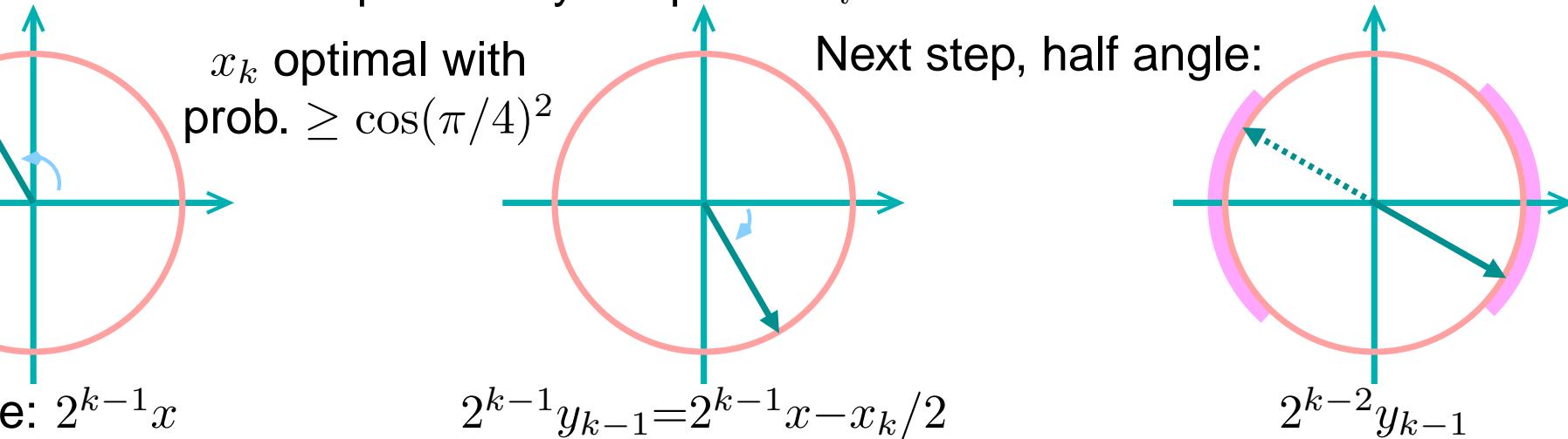
for $l = k$ **to** 1

$(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$

end

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.

- Overall probability of optimal x_l for each l :



Efficient Phase Estimation

PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

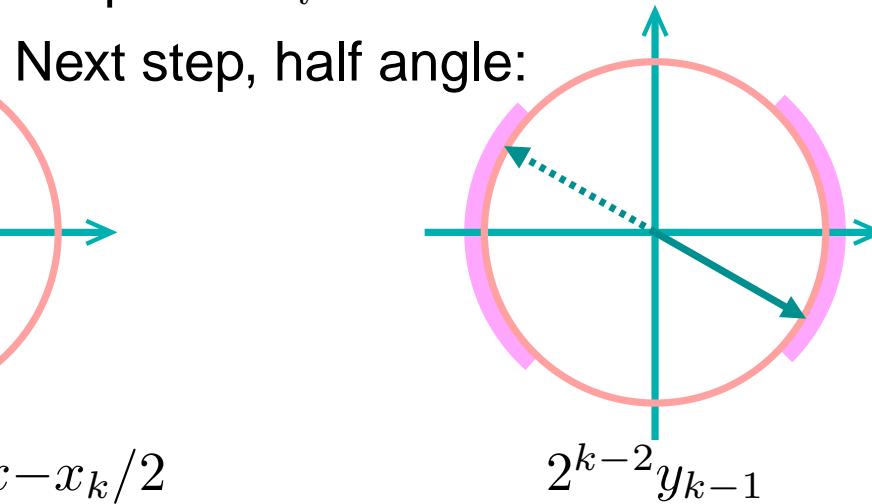
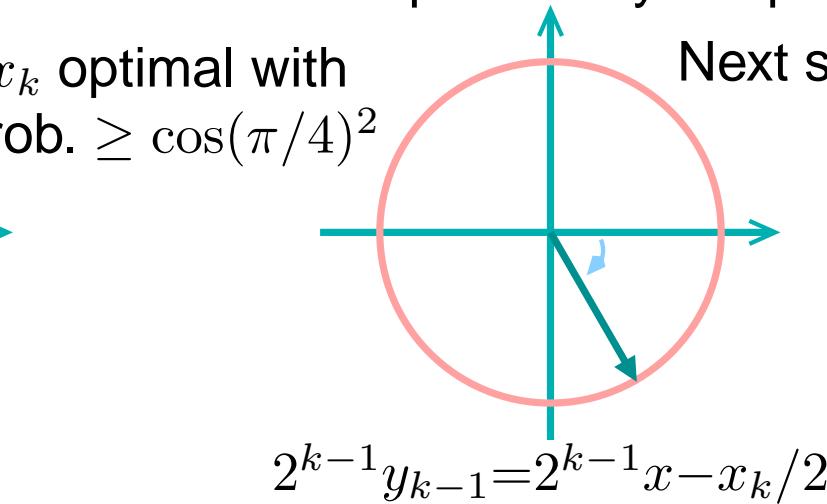
$$P_k \leftarrow P$$

for $l = k$ **to** 1

$$(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$$

end

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.
 - Overall probability of optimal x_l for each l :



Efficient Phase Estimation

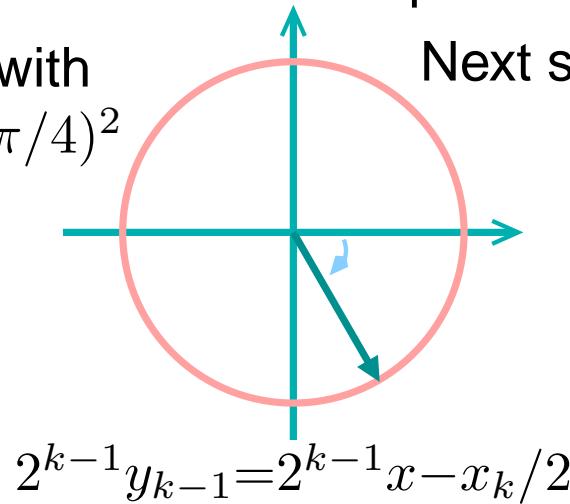
PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

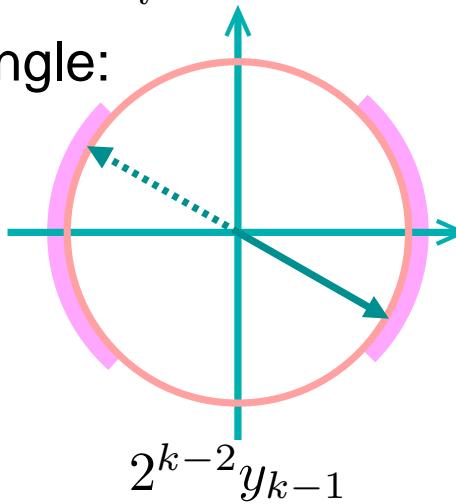
Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

```
 $P_k \leftarrow P$ 
for  $l = k$  to 1
     $(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$ 
end
```

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.
 - Overall probability of optimal x_l for each l :



Next step, half angle:



Efficient Phase Estimation

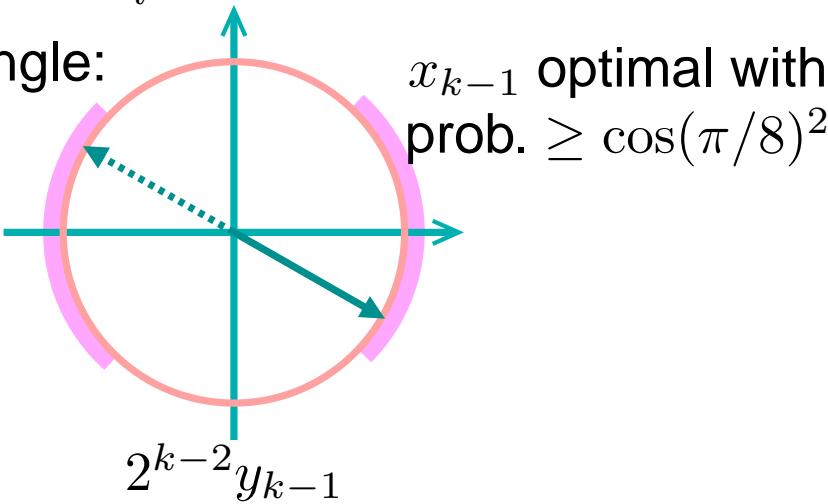
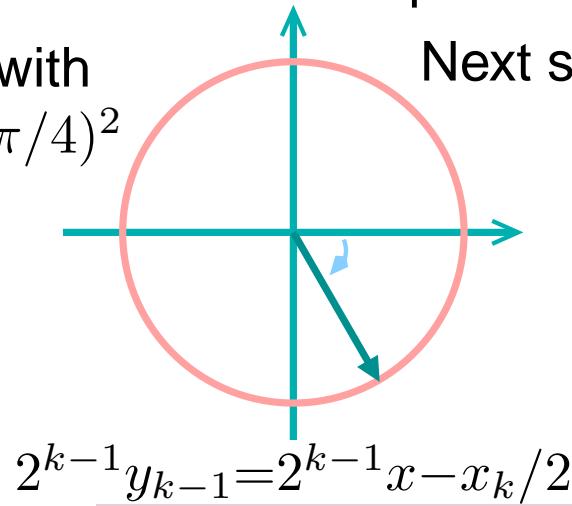
PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

```
 $P_k \leftarrow P$ 
for  $l = k$  to 1
     $(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$ 
end
```

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.
 - Overall probability of optimal x_l for each l :



Efficient Phase Estimation

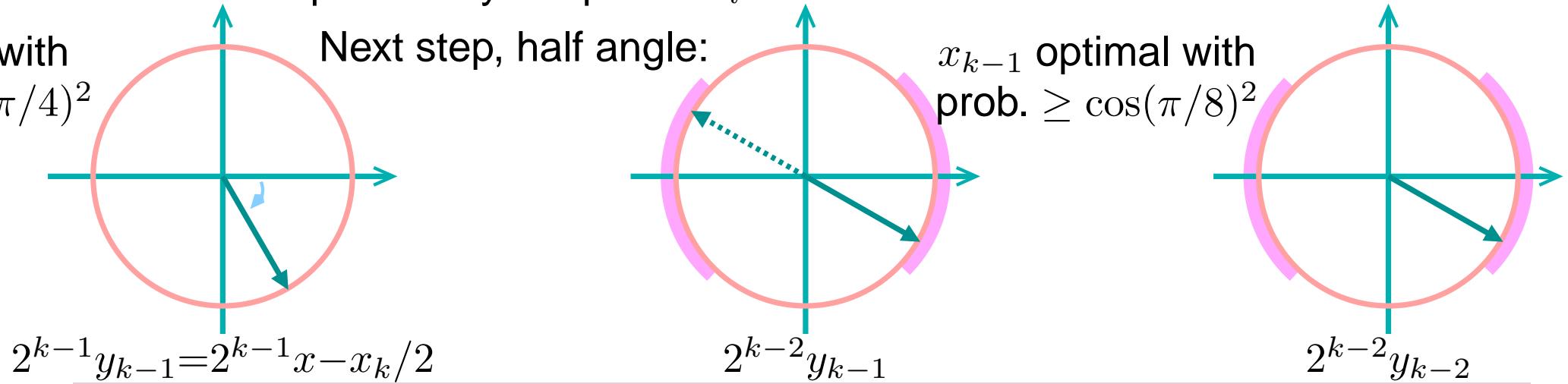
PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

```
 $P_k \leftarrow P$ 
for  $l = k$  to 1
     $(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$ 
end
```

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.
 - Overall probability of optimal x_l for each l :



Efficient Phase Estimation

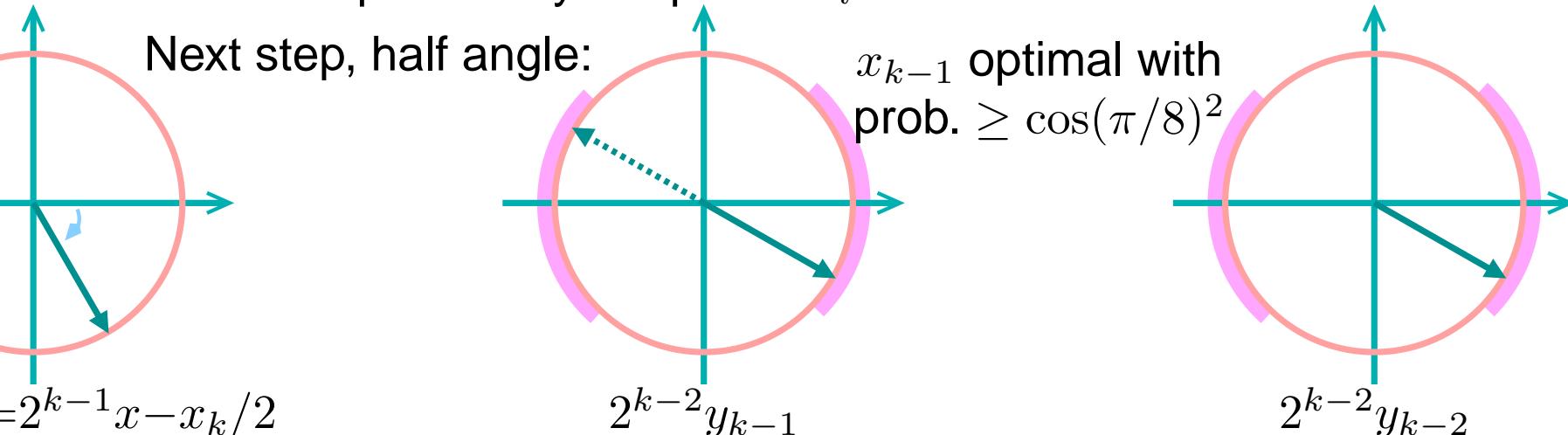
PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

```
 $P_k \leftarrow P$ 
for  $l = k$  to 1
   $(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$ 
end
```

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.
 - Overall probability of optimal x_l for each l :



Efficient Phase Estimation

PHASEEST(P, k)

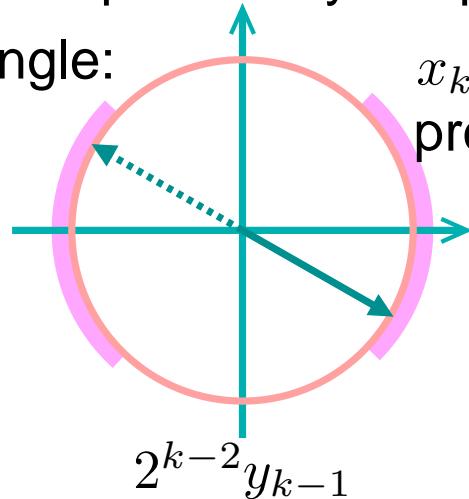
Input: An unknown z -rotation P , with efficient P^{2^l} .

Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

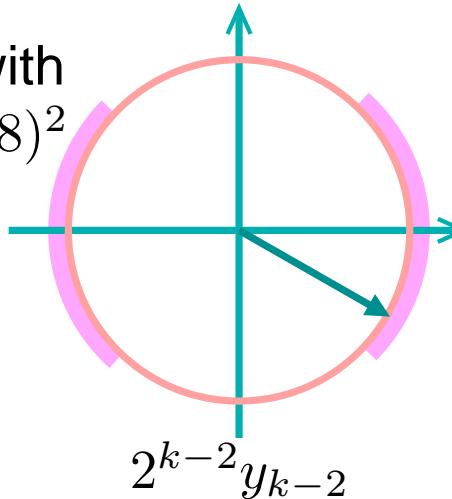
```
 $P_k \leftarrow P$ 
for  $l = k$  to 1
   $(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$ 
end
```

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.
 - Overall probability of optimal x_l for each l :

First step, half angle:



x_{k-1} optimal with
prob. $\geq \cos(\pi/8)^2$



Efficient Phase Estimation

PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

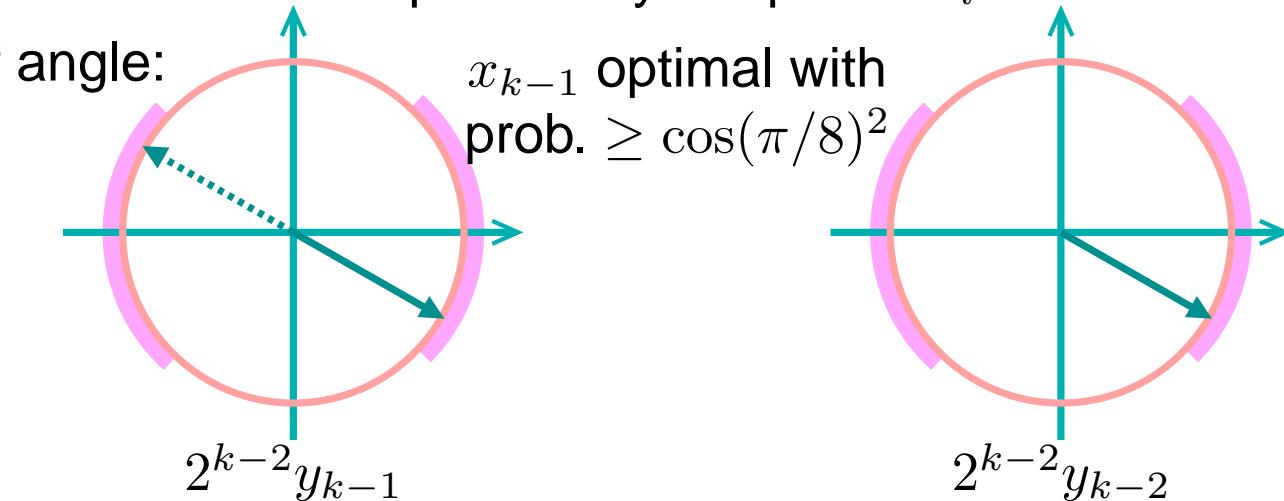
$P_k \leftarrow P$

for $l = k$ **to** 1

$(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$

end

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.
 - Overall probability of optimal x_l for each l :



Efficient Phase Estimation

PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

$P_k \leftarrow P$

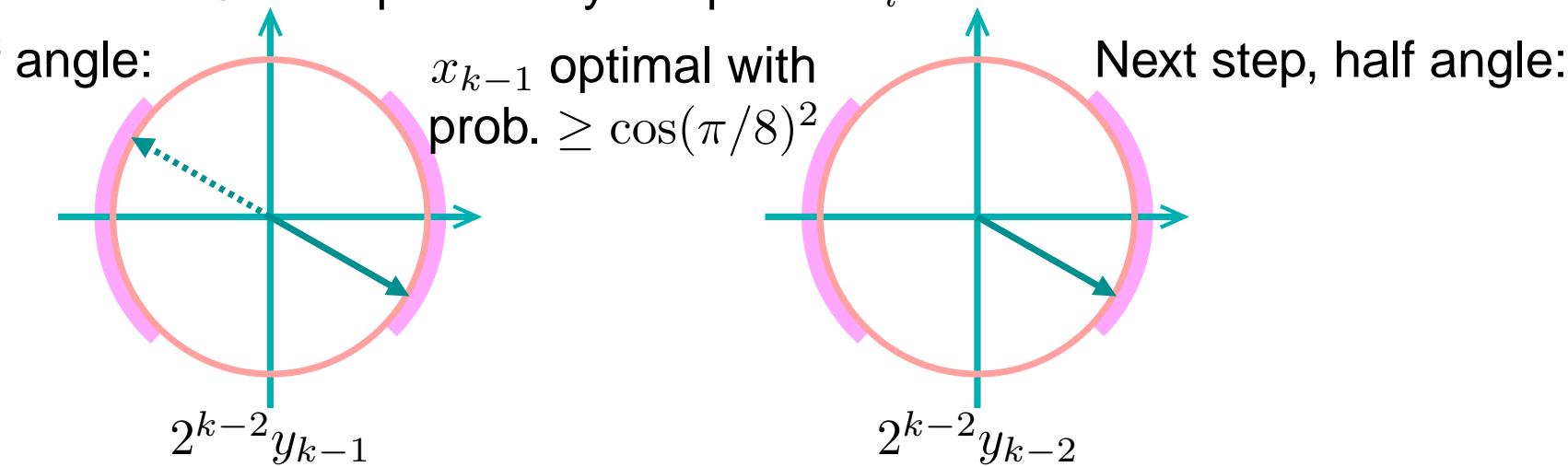
for $l = k$ **to** 1

$(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$

end

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.

- Overall probability of optimal x_l for each l :



Efficient Phase Estimation

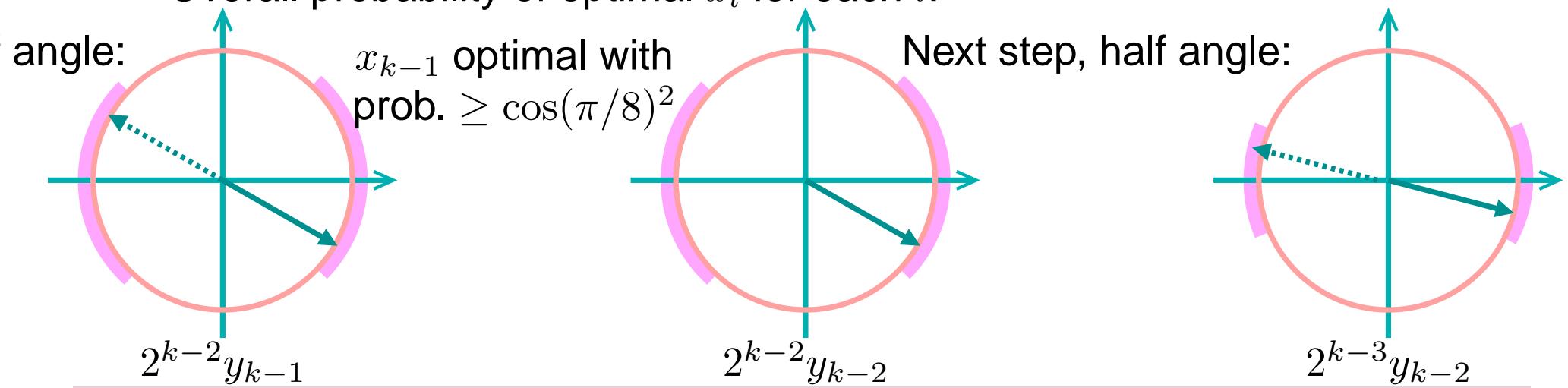
PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

```
 $P_k \leftarrow P$ 
for  $l = k$  to 1
   $(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$ 
end
```

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.
 - Overall probability of optimal x_l for each l :



Efficient Phase Estimation

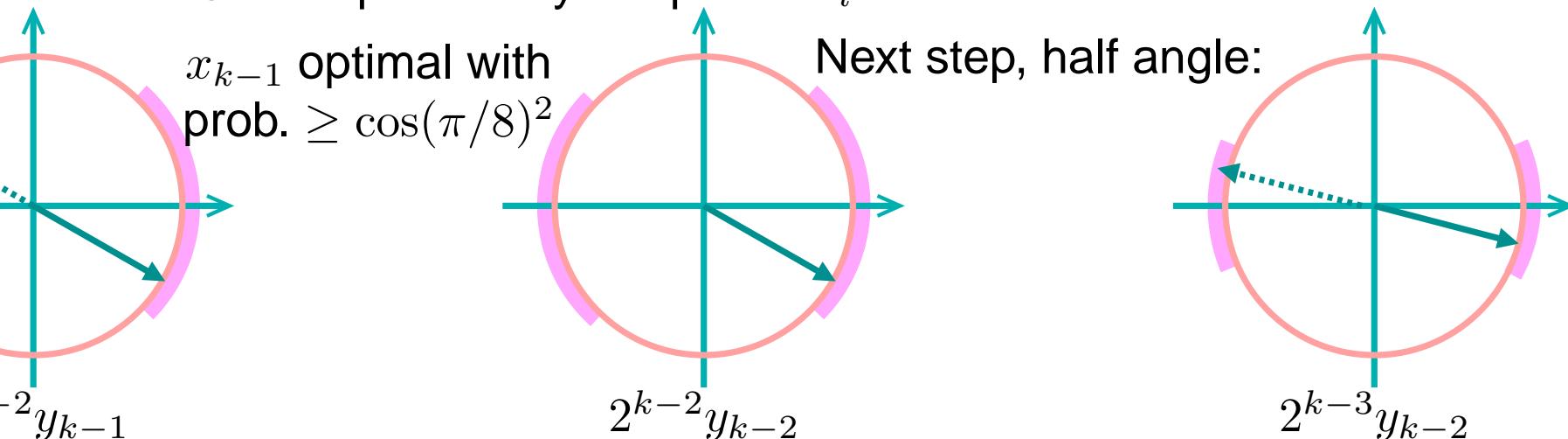
PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

```
 $P_k \leftarrow P$ 
for  $l = k$  to 1
     $(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$ 
end
```

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.
 - Overall probability of optimal x_l for each l :



Efficient Phase Estimation

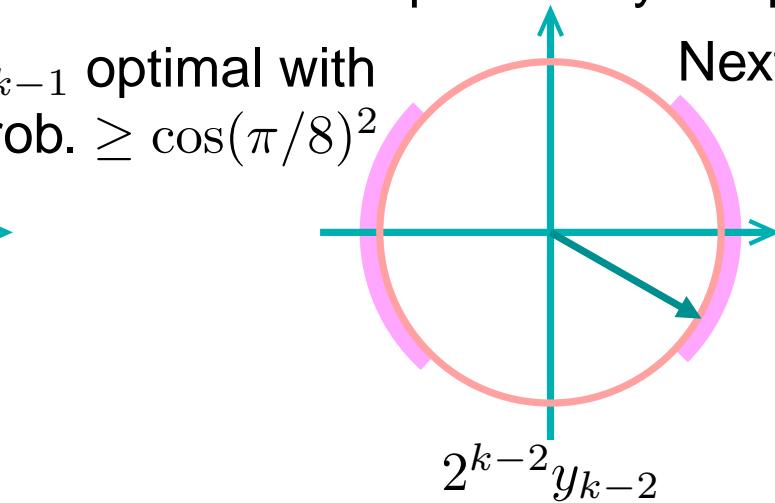
PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

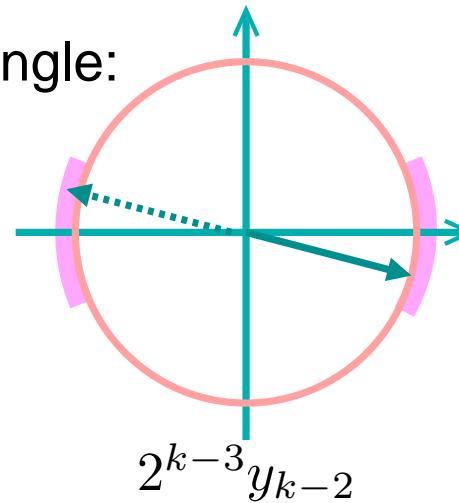
Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

```
 $P_k \leftarrow P$ 
for  $l = k$  to 1
     $(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$ 
end
```

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.
 - Overall probability of optimal x_l for each l :



Next step, half angle:



Efficient Phase Estimation

PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

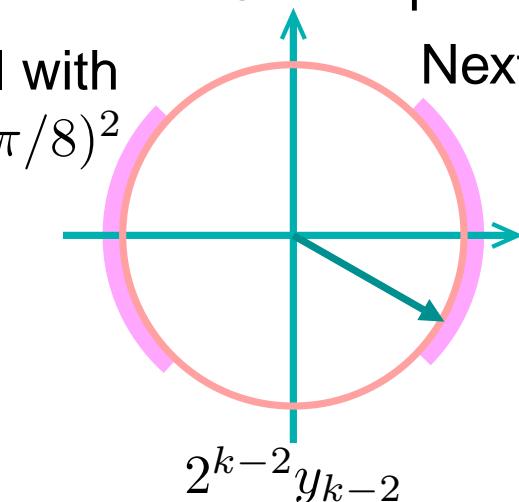
$P_k \leftarrow P$

for $l = k$ **to** 1

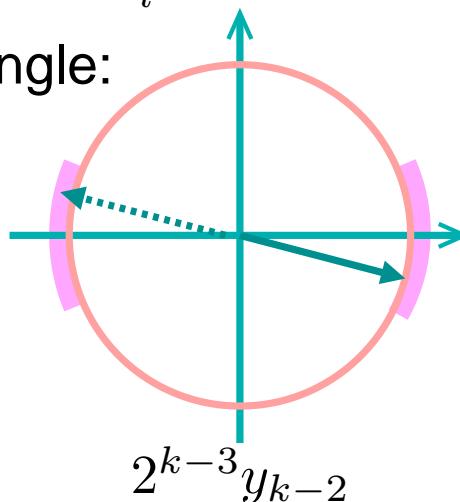
$(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$

end

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.
 - Overall probability of optimal x_l for each l :



Next step, half angle:



Efficient Phase Estimation

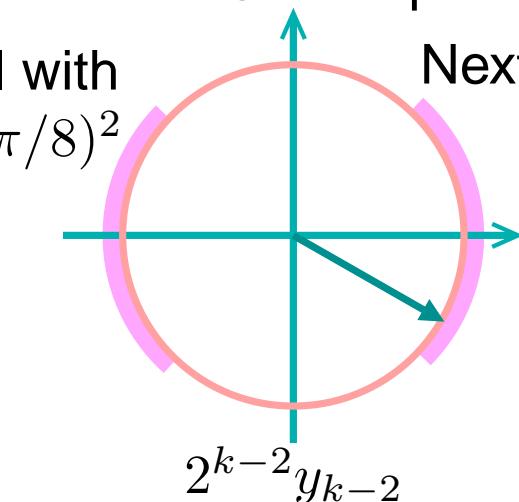
PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

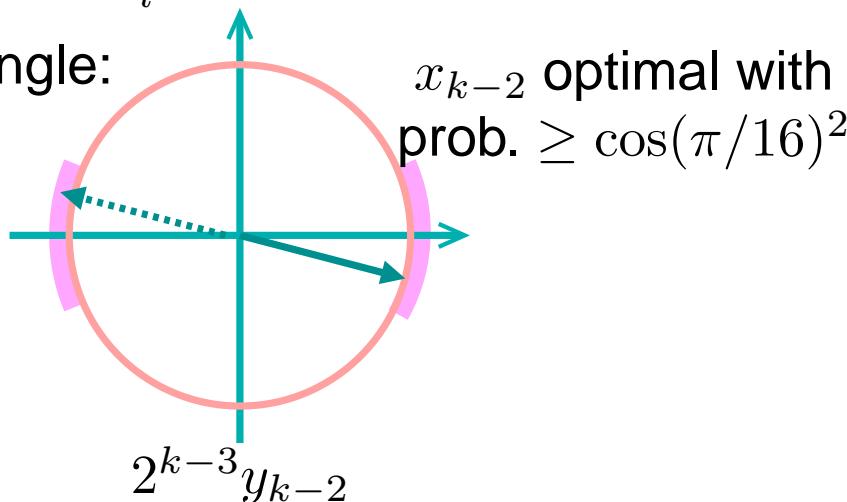
Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

```
 $P_k \leftarrow P$ 
for  $l = k$  to 1
     $(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$ 
end
```

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.
 - Overall probability of optimal x_l for each l :



Next step, half angle:



Efficient Phase Estimation

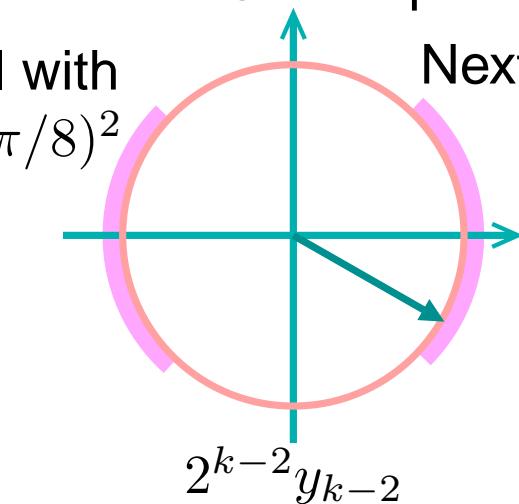
PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

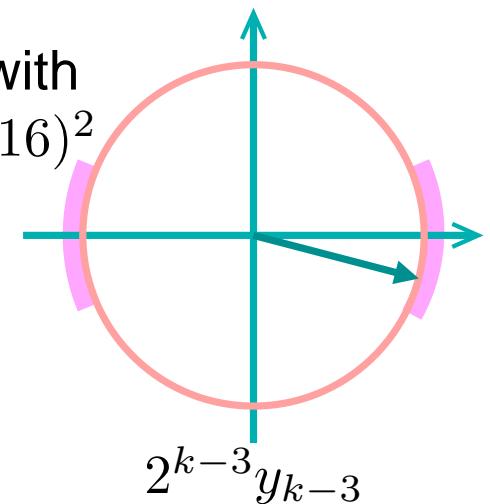
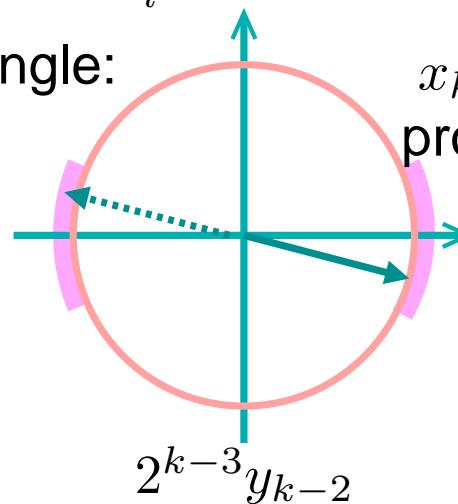
Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

```
 $P_k \leftarrow P$ 
for  $l = k$  to 1
   $(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$ 
end
```

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.
 - Overall probability of optimal x_l for each l :



Next step, half angle:



Efficient Phase Estimation

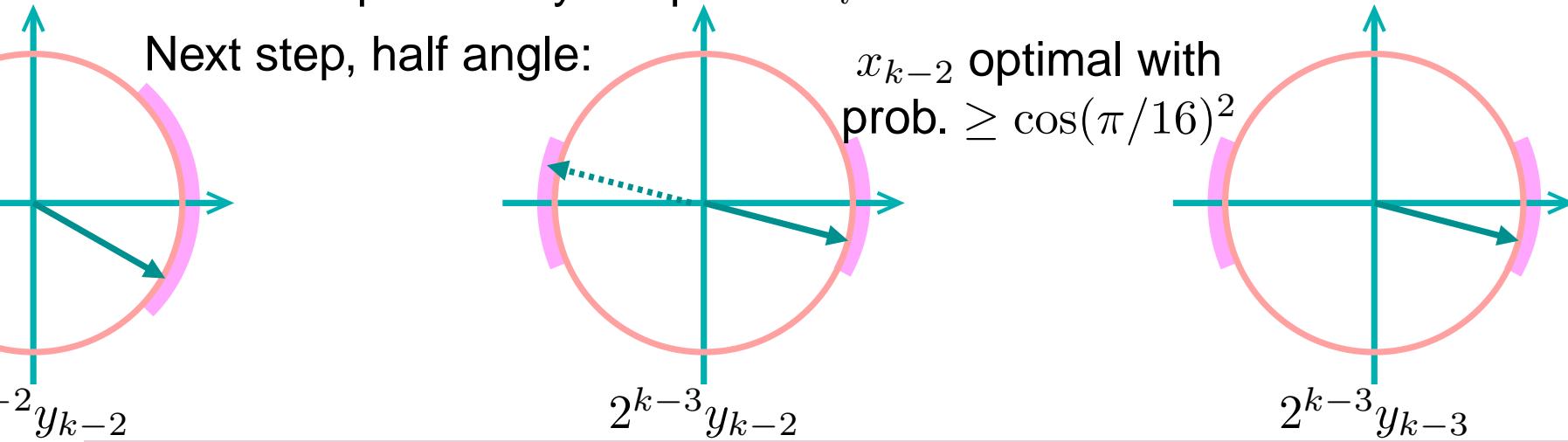
PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

```
 $P_k \leftarrow P$ 
for  $l = k$  to 1
     $(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$ 
end
```

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.
 - Overall probability of optimal x_l for each l :



Efficient Phase Estimation

PHASEEST(P, k)

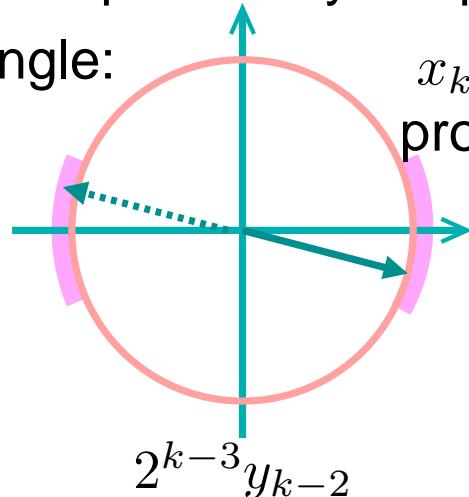
Input: An unknown z -rotation P , with efficient P^{2^l} .

Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

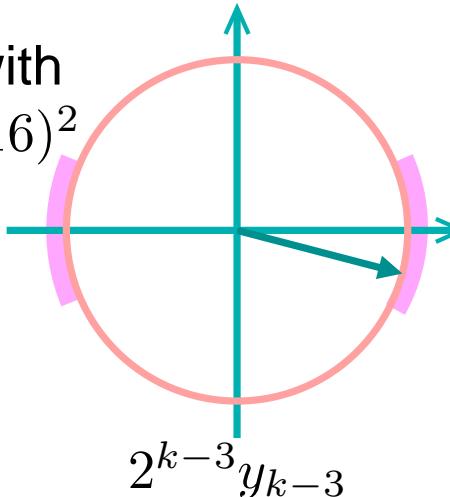
```
 $P_k \leftarrow P$ 
for  $l = k$  to 1
   $(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$ 
end
```

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.
 - Overall probability of optimal x_l for each l :

First step, half angle:



x_{k-2} optimal with
prob. $\geq \cos(\pi/16)^2$



Efficient Phase Estimation

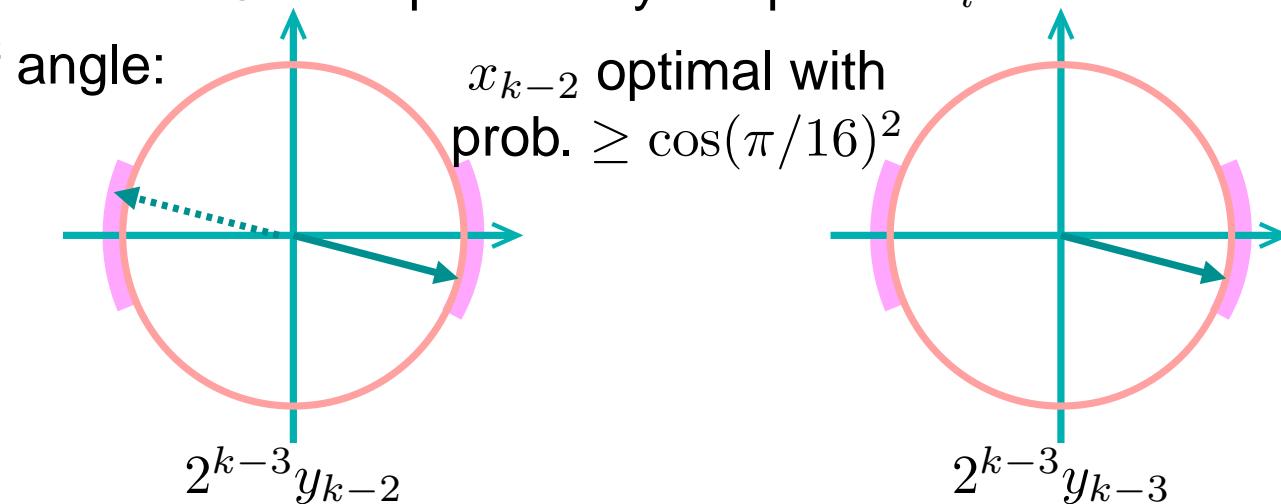
PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

```
 $P_k \leftarrow P$ 
for  $l = k$  to 1
     $(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$ 
end
```

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.
 - Overall probability of optimal x_l for each l :



Efficient Phase Estimation

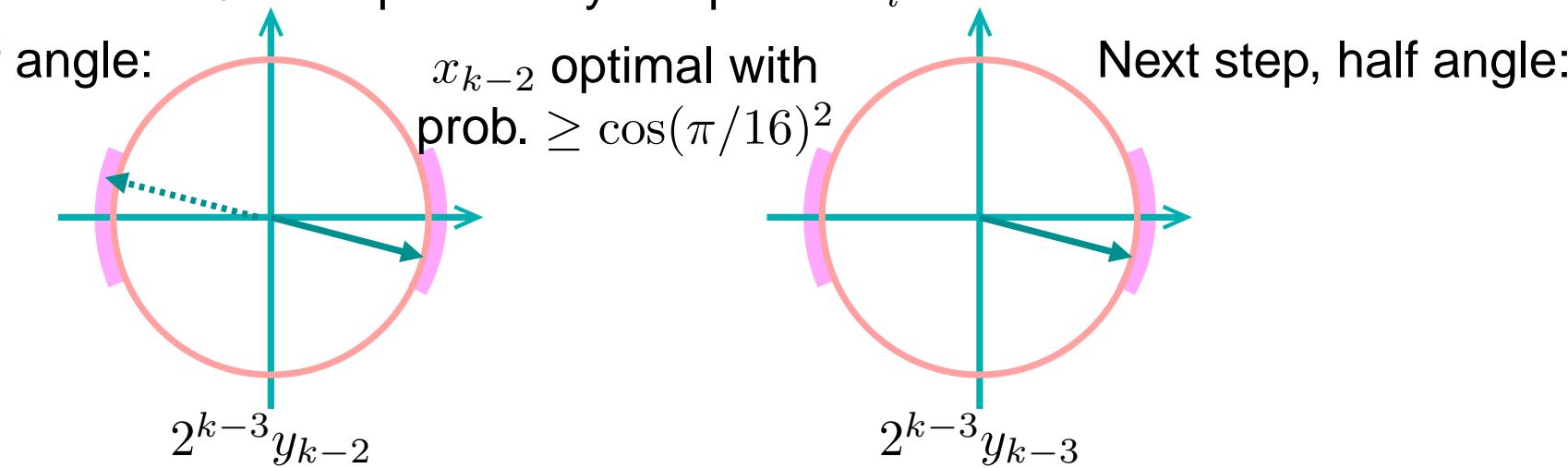
PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

```
 $P_k \leftarrow P$ 
for  $l = k$  to 1
     $(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$ 
end
```

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.
 - Overall probability of optimal x_l for each l :



Efficient Phase Estimation

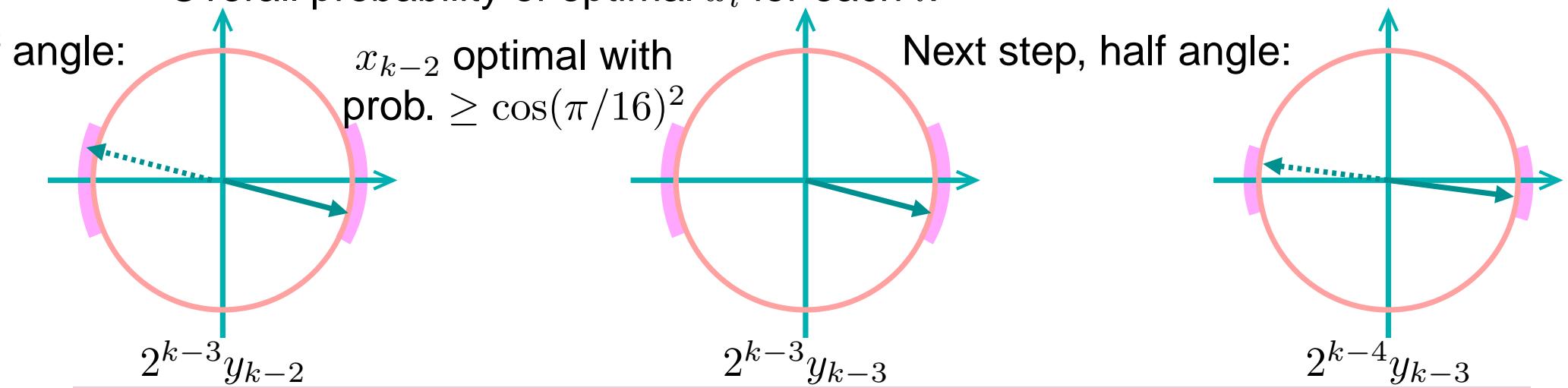
PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

```
 $P_k \leftarrow P$ 
for  $l = k$  to 1
     $(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$ 
end
```

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.
 - Overall probability of optimal x_l for each l :



Efficient Phase Estimation

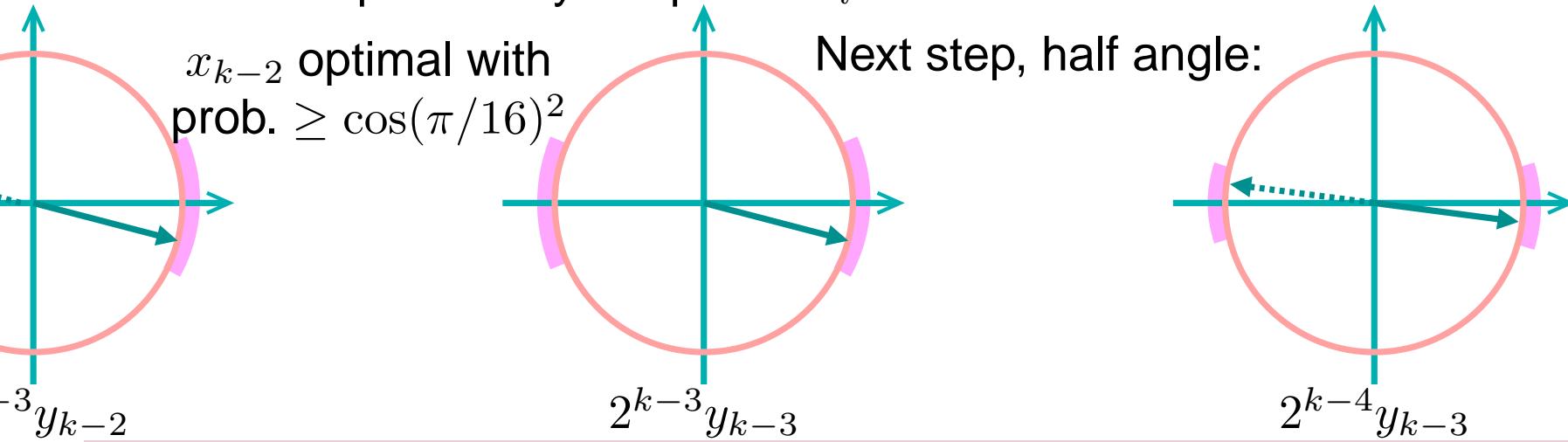
PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

```
 $P_k \leftarrow P$ 
for  $l = k$  to 1
   $(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$ 
end
```

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.
 - Overall probability of optimal x_l for each l :



Efficient Phase Estimation

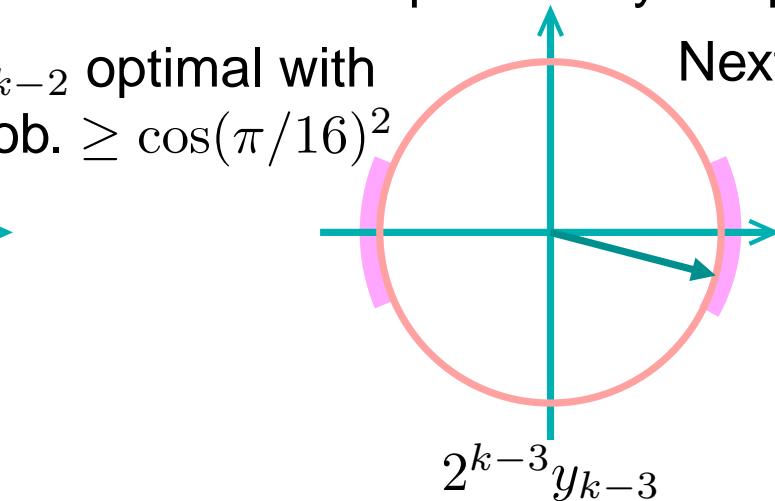
PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

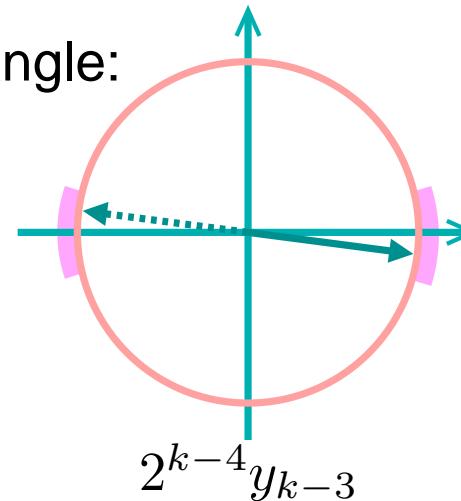
Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

```
 $P_k \leftarrow P$ 
for  $l = k$  to 1
     $(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$ 
end
```

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.
 - Overall probability of optimal x_l for each l :



Next step, half angle:



Efficient Phase Estimation

PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

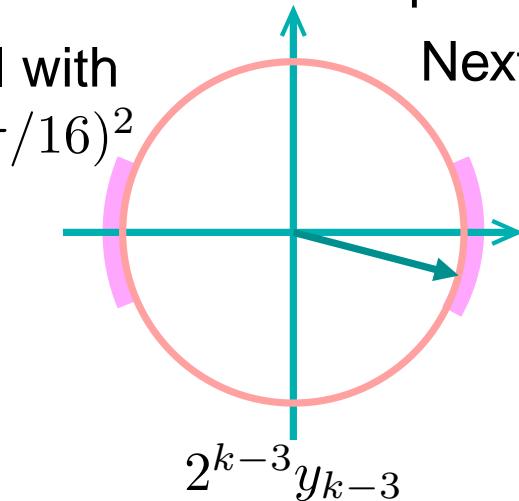
$P_k \leftarrow P$

for $l = k$ **to** 1

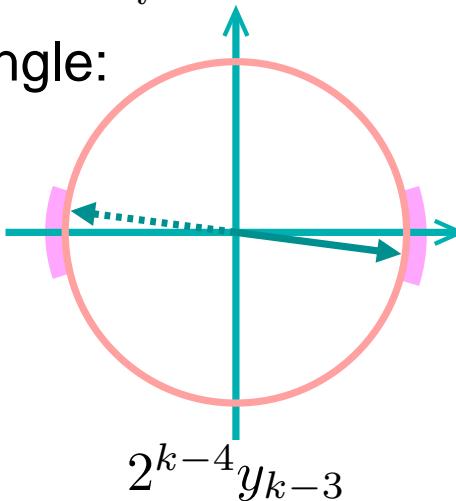
$(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$

end

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.
 - Overall probability of optimal x_l for each l :



Next step, half angle:



Efficient Phase Estimation

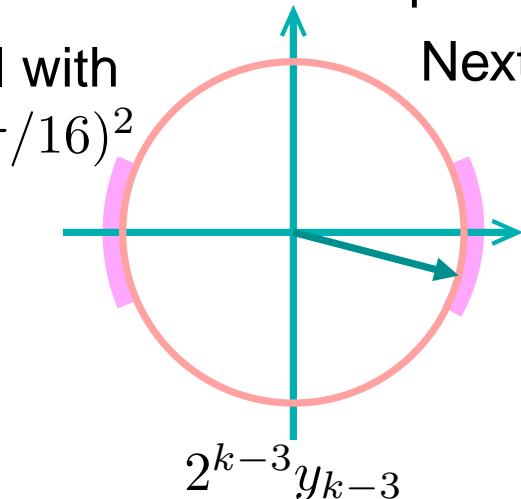
PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

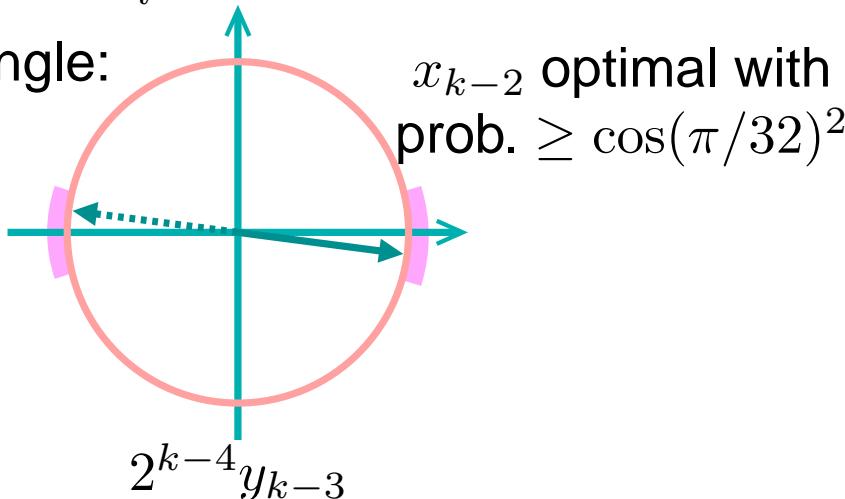
Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

```
 $P_k \leftarrow P$ 
for  $l = k$  to 1
     $(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$ 
end
```

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.
 - Overall probability of optimal x_l for each l :



Next step, half angle:



Efficient Phase Estimation

PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

$$P_k \leftarrow P$$

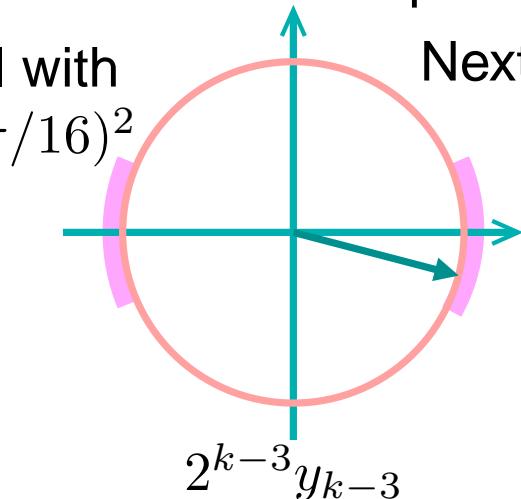
for $l = k$ **to** 1

$$(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$$

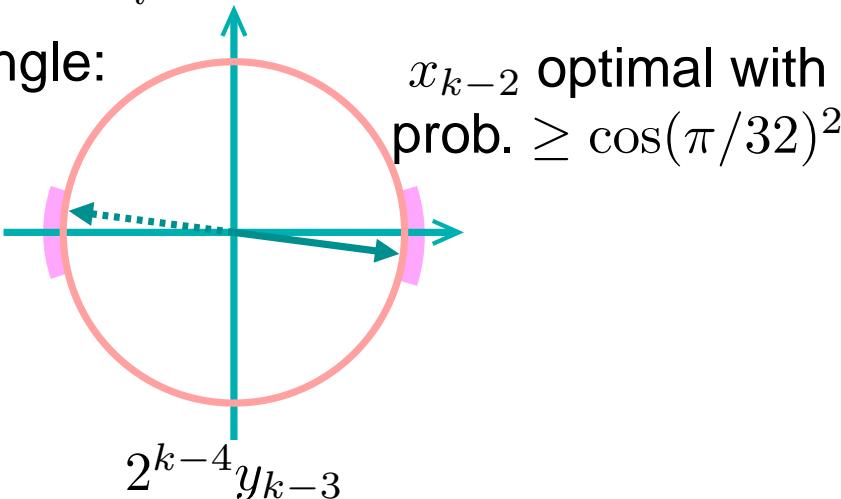
end

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.

- Overall probability of optimal x_l for each l :



Next step, half angle:



etc.



Efficient Phase Estimation

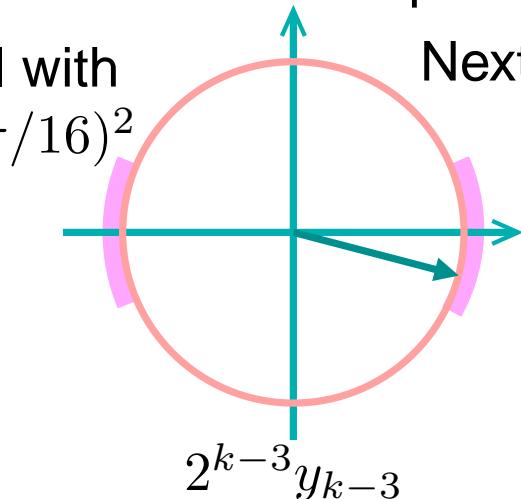
PHASEEST(P, k)

Input: An unknown z -rotation P , with efficient P^{2^l} .

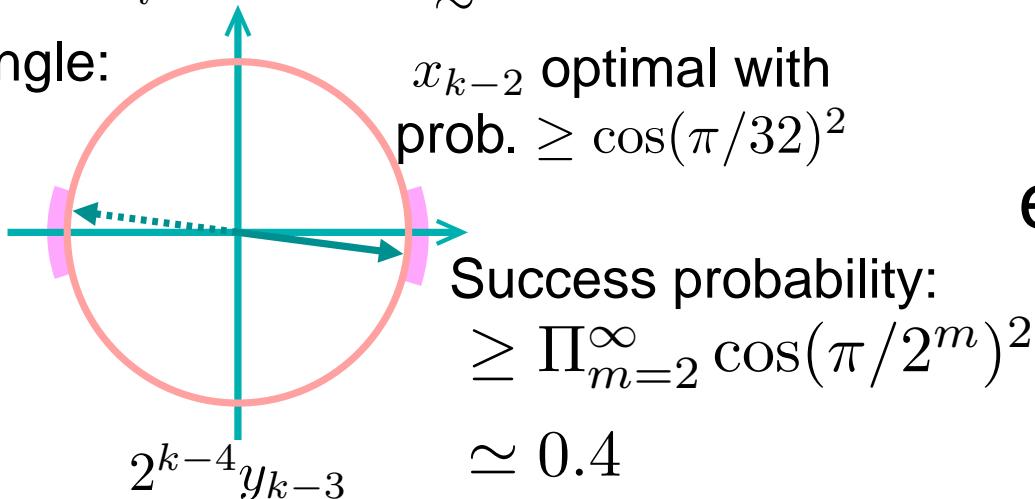
Output: $\tilde{x} = .x_1 \dots x_k$, the approximate angle of P in binary.

```
 $P_k \leftarrow P$ 
for  $l = k$  to 1
     $(P_{l-1}, x_l) \leftarrow \text{PHASESTEP}(P_l, l)$ 
end
```

- **Analysis.** Define $P = e^{-ix\pi\sigma_z}$, $P_l = e^{-iy_l\pi\sigma_z}$.
 - Overall probability of optimal x_l for each l : $\gtrsim .4$

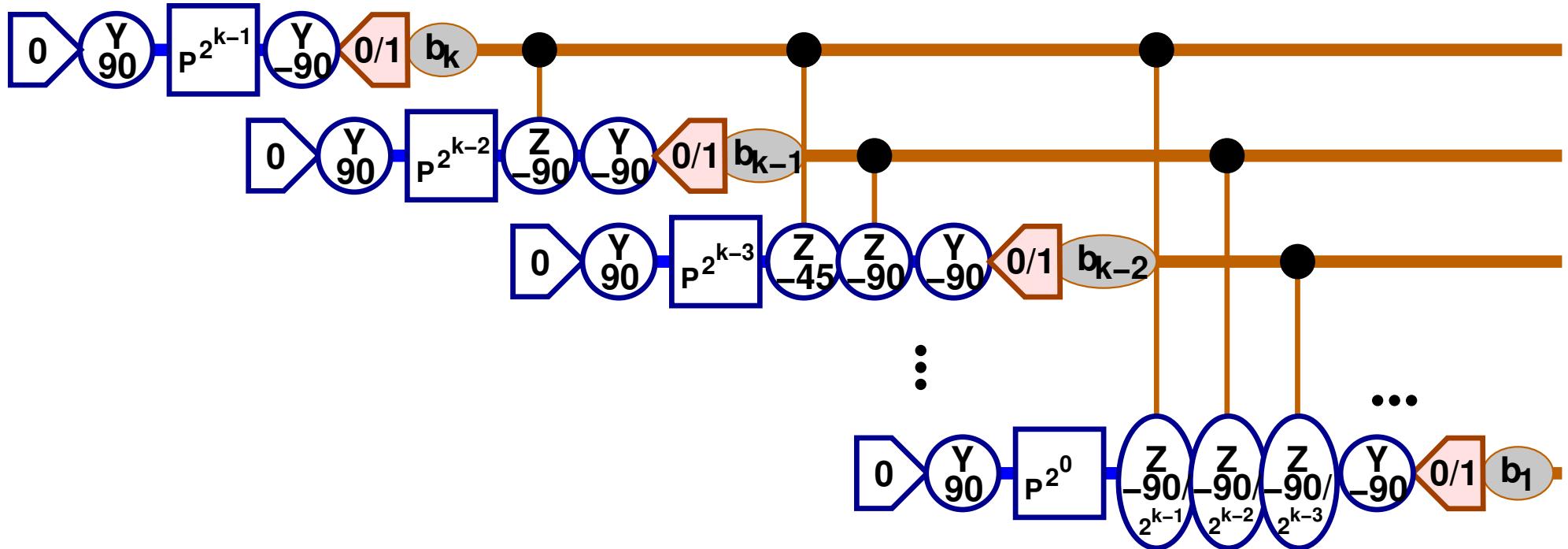


Next step, half angle:



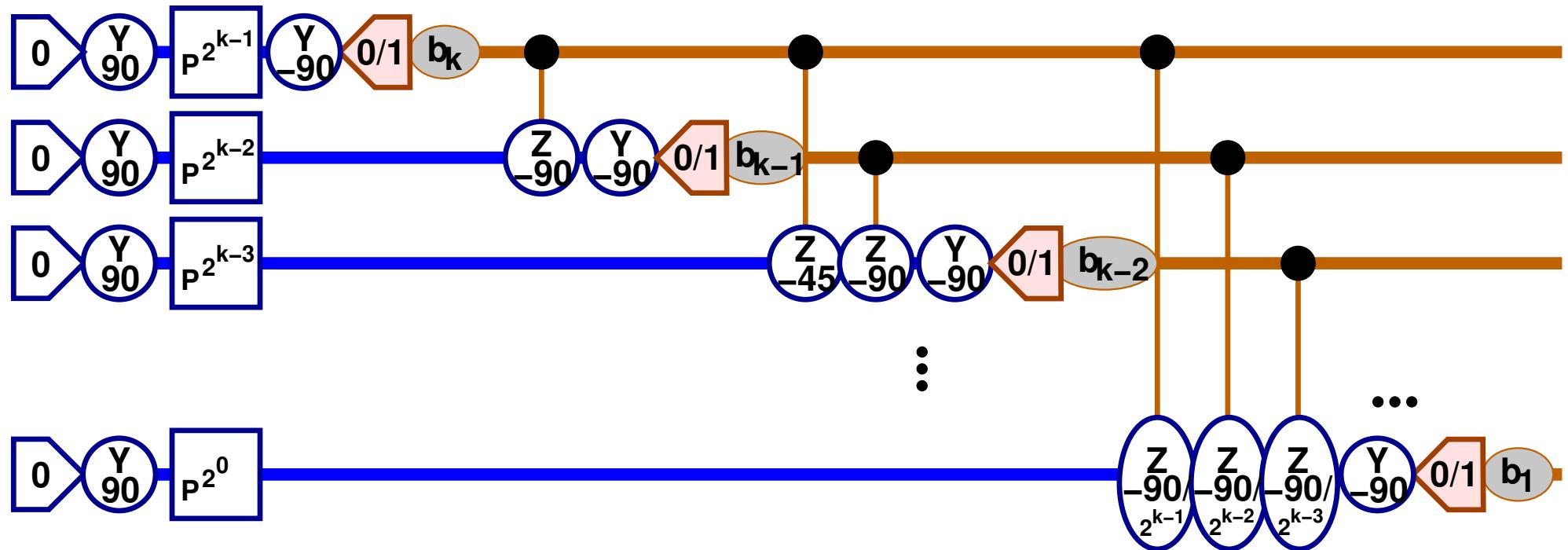
Quantum Fourier Transform

- From the full phase estimation network to the QFT.



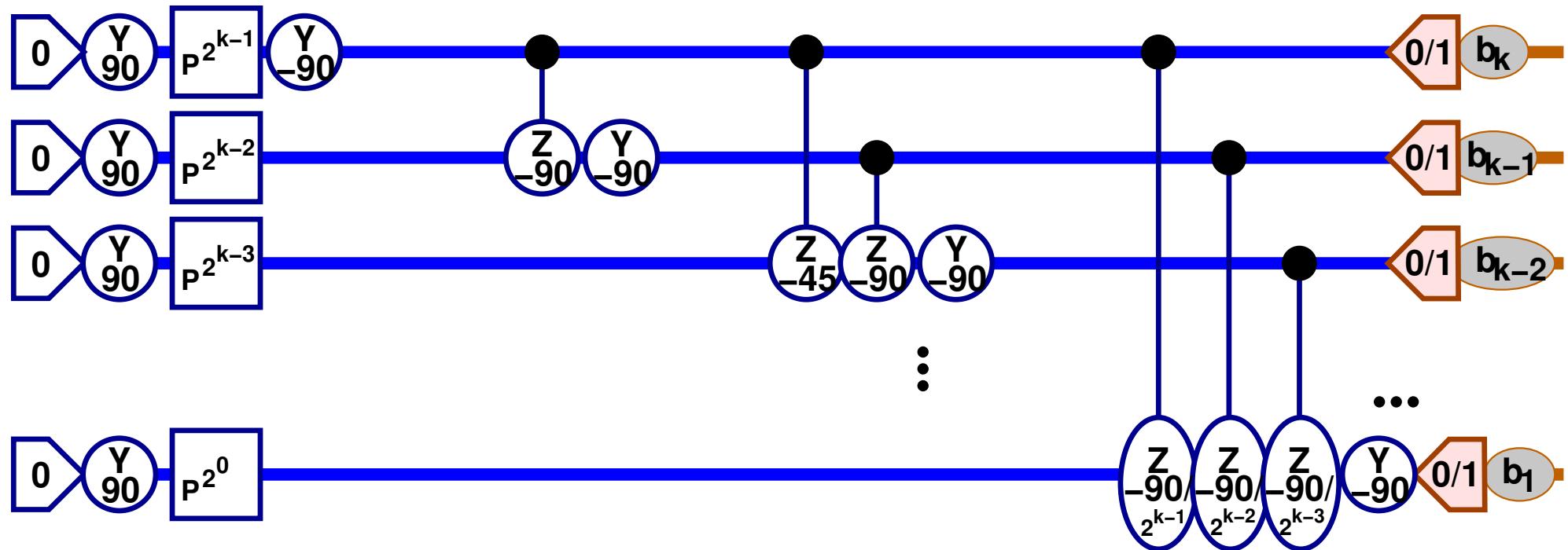
Quantum Fourier Transform

- From the full phase estimation network to the QFT.



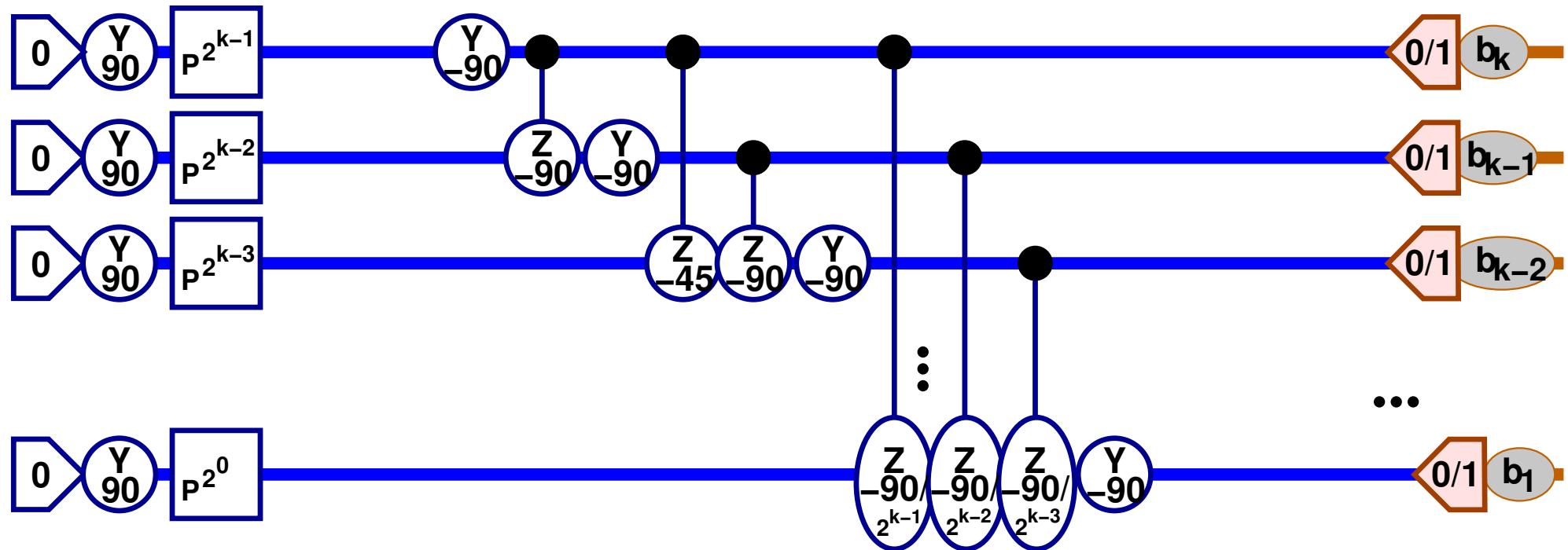
Quantum Fourier Transform

- From the full phase estimation network to the QFT.



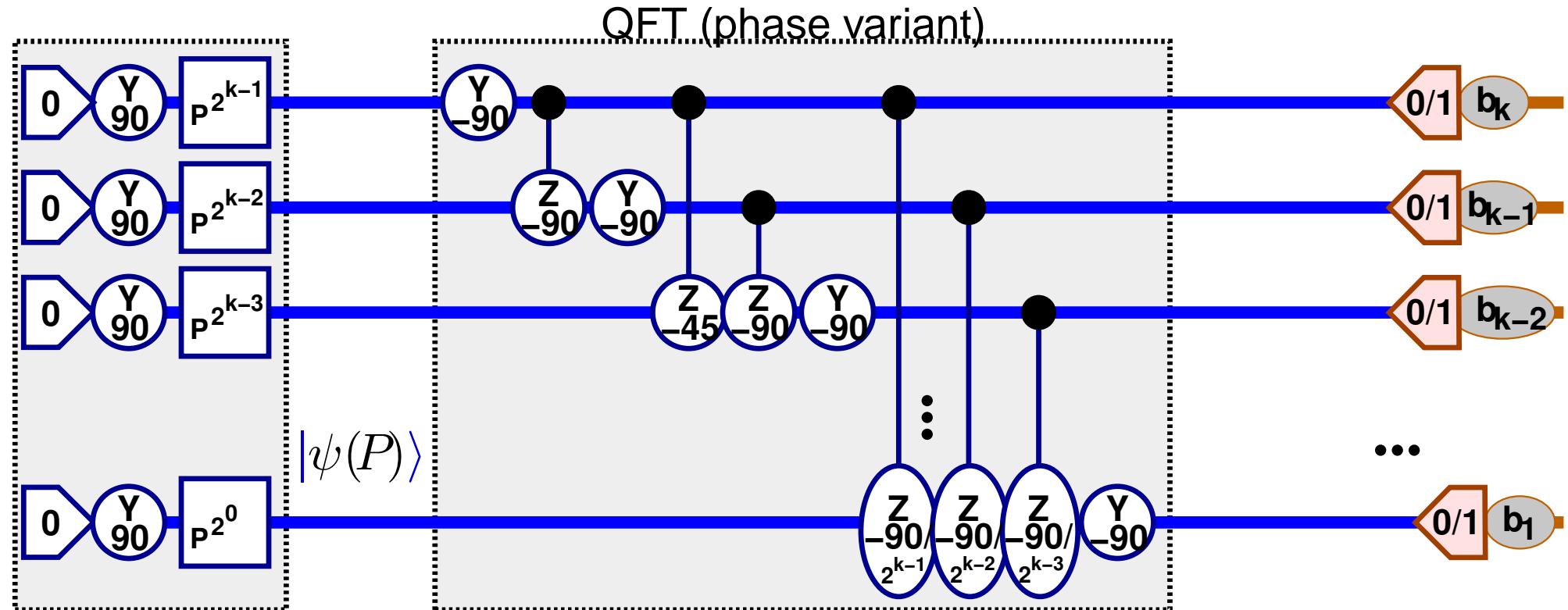
Quantum Fourier Transform

- From the full phase estimation network to the QFT.



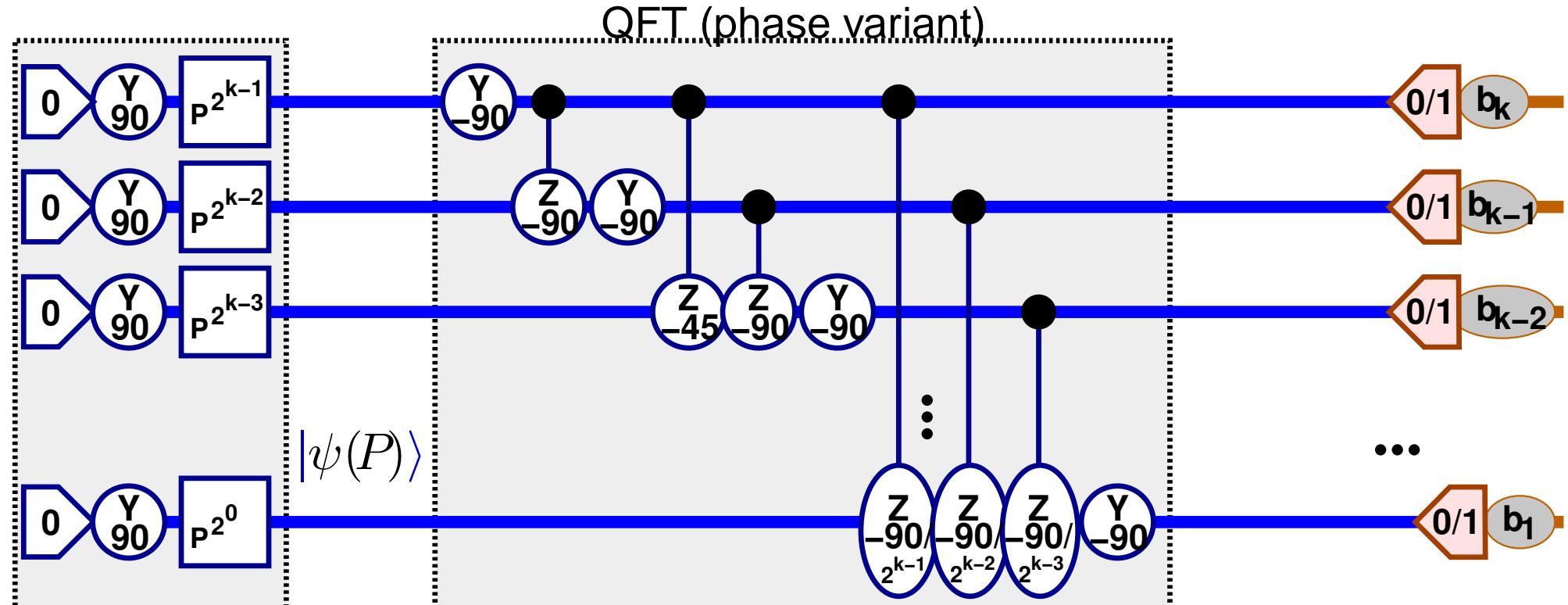
Quantum Fourier Transform

- From the full phase estimation network to the QFT.



Quantum Fourier Transform

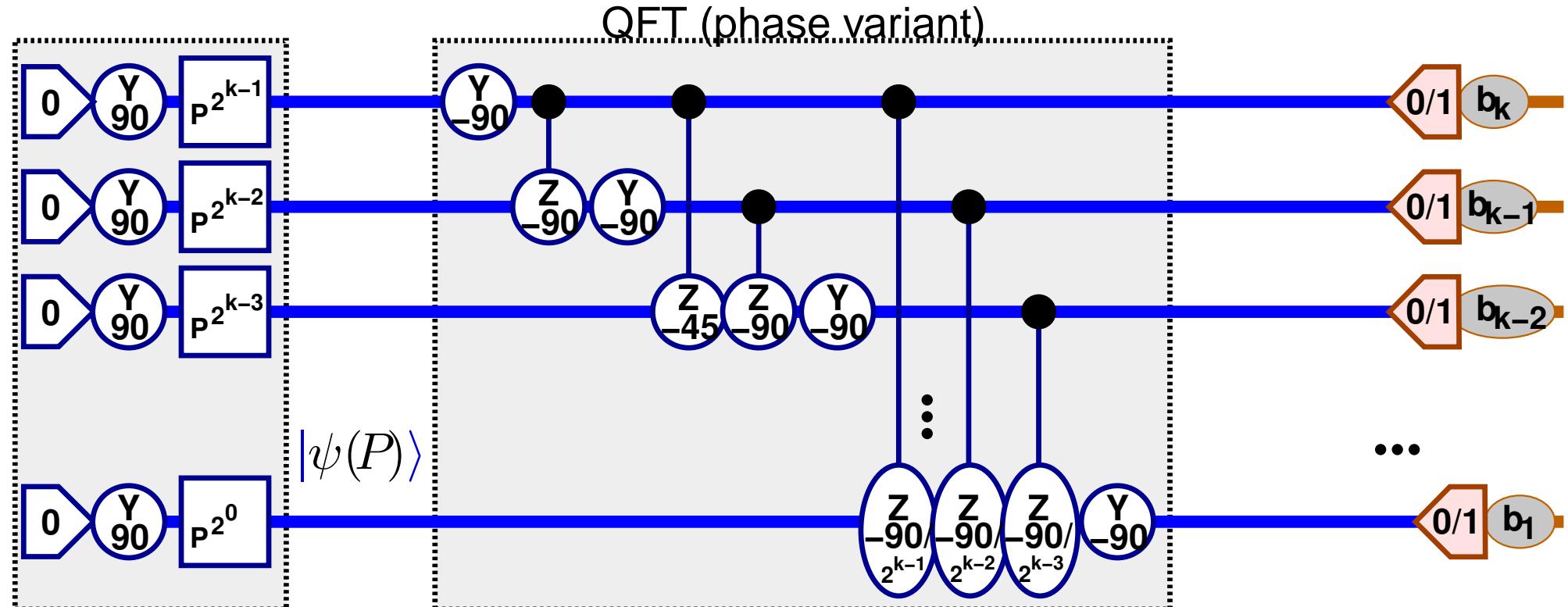
- From the full phase estimation network to the QFT.



- If $P = e^{-il\pi/2^k}$, then $|\psi(P)\rangle$ is the l 'th Fourier state modulo 2^k .

Quantum Fourier Transform

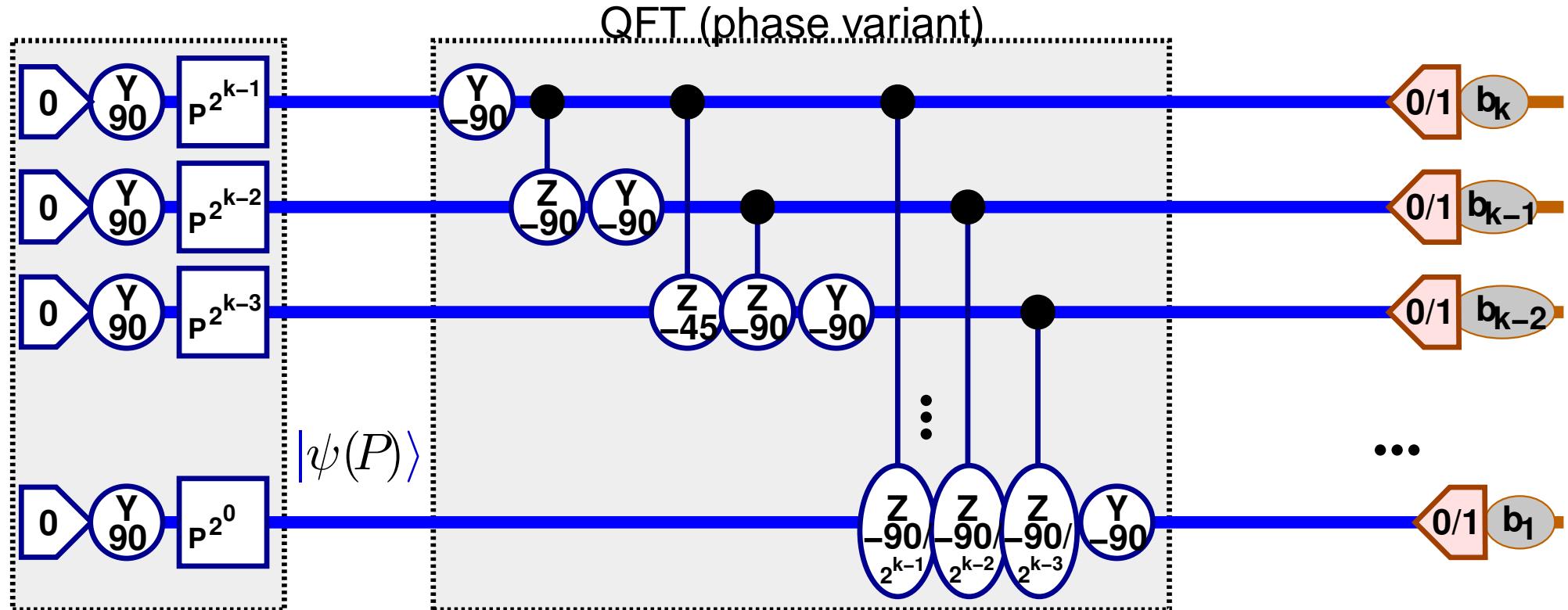
- From the full phase estimation network to the QFT.



- If $P = e^{-il\pi/2^k}$, then $|\psi(P)\rangle$ is the l 'th Fourier state modulo 2^k . Hence QFT transforms Fourier states to logical states.

Quantum Fourier Transform

- From the full phase estimation network to the QFT.



- If $P = e^{-il\pi/2^k}$, then $|\psi(P)\rangle$ is the l 'th Fourier state modulo 2^k . Hence QFT transforms Fourier states to logical states.
⇒ QFT is a phase variant of the Fourier transform.

Contents

Title: IQI 04, Seminar 9.....	0	
Composite and Prime Numbers	top	1
Factoring.....	top	2
Trial Division Algorithm.....	top	3
Modular Arithmetic	top	4
Largest Common Divisors.....	top	5
The Structure of Z_q	top	6
Public Key Crypto: RSA.....	top	7
Factor Finding with Square Roots of Unity.....	top	8
From Order to Square Roots of Unity	top	9
Quantum Order Finding I.....	top	10
Quantum Order Finding II	top	11
From Eigenvalues to Order.....	top	12
A Phase Estimation Step.....	top	13
Efficient Phase Estimation I	top	14
Quantum Fourier Transform	top	15
References		17



References

- [1] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26:1484–1509, 1997.
- [2] D. Beckman, A. N. Chari, S. Devabhaktuni, and J. Preskill. Efficient networks for quantum factoring. *Phys. Rev. A*, 54:1034–, 1996.
- [3] A. Ekert and R. Jozsa. Notes on quantum factoring. *Reviews of Modern Physics*, 68:733–753, 1996.
- [4] M. E. Briggs. *An Introduction to the General Number Field Sieve*. PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 1998.
- [5] A. K. Lenstra. Integer factoring. *Designs, Codes and Cryptography*, 19:101–128, 2000.

