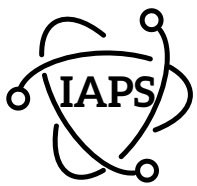
Lecture 19

QUANTUM CRYPTOGRAPHY. PART 4 - Prime number factorization

of the course "Fundamentals of Quantum Computing"

(by  *and* QUANTERALL*)*

Stoyan Mishev



INSTITUTE *for* ADVANCED PHYSICAL STUDIES

October 21, 2022

Plan

RSA

Shor's algorithm

RSA

R.L. Rivest, A. Shamir, L.M. Adleman, Commun. ACM 51(2), 2738 (1978)

$$m = \sum_{j=0}^{n_M-1} s_j 2^j \in \{0, \ldots, N_M\}$$

$$m = \sum_{j=0}^{n_M - 1} s_j 2^j \in \{0, \ldots, N_M\}$$

The Receiver

- picks two primes $\mathbf{p} \neq \mathbf{q}$ with $p, q > N_M$;
- finds an integer $\mathbf{a}$: $\gcd(a, (p-1)(q-1)) = 1$;
- calculate $\mathbf{N} = p.q \in \mathbb{N}$;
- publishes the public key $k_{\text{pub}} = (a, N)$.

$$m = \sum_{j=0}^{n_M - 1} s_j 2^j \in \{0, \ldots, N_M\}$$

The Receiver

- picks two primes $\mathbf{p} \neq \mathbf{q}$ with $p, q > N_M$;
- finds an integer $\mathbf{a}$: $\gcd(a, (p-1)(q-1)) = 1$;
- calculate $\mathbf{N} = p.q \in \mathbb{N}$;
- publishes the public key $k_{\text{pub}} = (a, N)$.

Any Sender

- encrypts his plaintext $m \leq N_M < N$ by calculating $e(k_{\text{pub}}, m) := m^a \mod N$
- and sends the ciphertext $c = e(k_{\text{pub}}, m)$ on public channels to the Receiver.

$$m = \sum_{j=0}^{n_M - 1} s_j 2^j \in \{0, \ldots, N_M\}$$

The Receiver

- picks two primes $\mathbf{p} \neq \mathbf{q}$ with $p, q > N_M$;
- finds an integer $\mathbf{a}$: $\gcd(a, (p-1)(q-1)) = 1$;
- calculate $\mathbf{N} = p.q \in \mathbb{N}$;
- publishes the public key $k_{\text{pub}} = (a, N)$.

Any Sender

- encrypts his plaintext $m \leq N_M < N$ by calculating $e(k_{\text{pub}}, m) := m^a \mod N$
- and sends the ciphertext $c = e(k_{\text{pub}}, m)$ on public channels to the Receiver.

The Receiver

- finds $\mathbf{b}$: $a.b \mod (p-1)(q-1) = 1$;
- uses $k_{\text{priv}} = (\mathbf{b}, N)$ as the private key and decrypt $d(k_{\text{priv}}, c) := c^b \mod N$. Then $d(k_{\text{priv}}, e(k_{\text{pub}}, m)) = m$!

Theorem 6.6 *Let p and q be two different primes and let $m \in \mathbb{N}$ with $m < \min\{q, p\}$. Moreover, let $a, b \in \mathbb{N}$ be such that*

$$ab \bmod (p-1)(q-1) = 1\,.$$

Then we have

$$m^{ab} \bmod pq = m\,.$$

$$
\begin{aligned}
\text{RSA-768} = {} & 12301866845301177551304949583849627207728535695 \\
& 95334792197322452151726400507263657518745202199 \\
& 78646938995647494277406384592519255732630345373 \\
& 15482685079170261221429134616704292143116022212 \\
& 40479274737794080665351419597459856902143413 \\
= {} & 33478071698956898786044169848212690817704794983 \\
& 71376856891243138898288379387800228761471165253 \\
& 17430877378144679994 89 \\
\times {} & 36746043666799590428244633799627952632279158164 \\
& 34308764267603228381573966651127923337341714339 \\
& 681027009279873630 8917 .
\end{aligned}
$$

$\sim$ 2000 CPU-years (on "2.2 GHz 2 GB RAM Opteron")

Shor's algorithm

P. Shor, in Proceedings of the 35th Annual Symposium on Foundations of Computer Science (IEEE Computer Society Press, 1994), pp. 124–134

The number of computational steps SSHOR(N) in SHOR's algorithm to factorize N satisfies

$$S_{\mathrm{SHOR}}(N) \in O\big((\log_2 N)^3 \log_2 \log_2 N\big) \qquad \text{for } N \to \infty$$

The factorization of a number of the order of 10^{1000} with a quantum computer would thus require a number of steps of the order of 10^9.

The number of computational steps SSHOR(N) in SHOR's algorithm to factorize N satisfies

$$S_{\mathrm{SHOR}}(N) \in O\big((\log_2 N)^3 \log_2 \log_2 N\big) \qquad \text{for } N \to \infty$$

The factorization of a number of the order of 10^{1000} with a quantum computer would thus require a number of steps of the order of 10^9.
The factorization of a number N is equivalent to finding the period of a given function

The number of computational steps SSHOR(N) in SHOR's algorithm to factorize N satisfies

$$S_{\mathrm{SHOR}}(N) \in O\big((\log_2 N)^3 \log_2 \log_2 N\big) \qquad \text{for } N \to \infty$$

The factorization of a number of the order of 10^{1000} with a quantum computer would thus require a number of steps of the order of 10^9.
The factorization of a number N is equivalent to finding the period of a given function and finding this period can be accelerated with the help of a quantum algorithm.

The number of computational steps SSHOR(N) in SHOR's algorithm to factorize N satisfies

$$S_{\mathrm{SHOR}}(N) \in O\big((\log_2 N)^3 \log_2 \log_2 N\big) \qquad \text{for } N \to \infty$$

The factorization of a number of the order of 10^{1000} with a quantum computer would thus require a number of steps of the order of 10^9.
The factorization of a number N is equivalent to finding the period of a given function and finding this period can be accelerated with the help of a quantum algorithm.
Notes:

- N must contain at least two **distinct** prime factors, i.e if p is a prime and $N = p^{\nu_p}$ then the Shor's algorithm is not "reliable enough";
- we consider N to be an odd integer.

The factorization of a number N is equivalent to finding the period of a function and finding this period can be accelerated with the help of a quantum algorithm.

The factorization of a number N is equivalent to finding the period of a function and finding this period can be accelerated with the help of a quantum algorithm.

The function $f_{b,N}(n)$ is periodic:

$$\begin{aligned} f_{b,N} : \mathbb{N}_0 &\longrightarrow \mathbb{N}_0 \\ n &\longmapsto f_{b,N}(n) := b^n \bmod N \end{aligned}$$

and the period of a function is:

$$r := min\{m \in N | f(n+m) = f(n), \forall n \in \mathbb{N}_0\}$$

The period of the function $f_{b,N}(n)$ is order of (b mod N), i.e. $ord_N(b) := \min\{m \in N | b^m \bmod N = 1\}$.

The factorization of a number N is equivalent to finding the period of a function and finding this period can be accelerated with the help of a quantum algorithm.
The function $f_{b,N}(n)$ is periodic:

$$\begin{aligned} f_{b,N} : \mathbb{N}_0 &\longrightarrow \mathbb{N}_0 \\ n &\longmapsto f_{b,N}(n) := b^n \bmod N \end{aligned}$$

and the period of a function is:

$$r := min\{m \in N | f(n+m) = f(n), \forall n \in \mathbb{N}_0\}$$

The period of the function $f_{b,N}(n)$ is order of (b mod N), i.e. $ord_N(b) := \min\{m \in N | b^m \bmod N = 1\}$.
b is a natural number $b < N$ which has no common divisor with N

The factorization of a number N is equivalent to finding the period of a function and finding this period can be accelerated with the help of a quantum algorithm.
The function $f_{b,N}(n)$ is periodic:

$$\begin{aligned} f_{b,N} : \mathbb{N}_0 &\longrightarrow \mathbb{N}_0 \\ n &\longmapsto f_{b,N}(n) := b^n \bmod N \end{aligned}$$

and the period of a function is:

$$r := min\{m \in N | f(n+m) = f(n), \forall n \in \mathbb{N}_0\}$$

The period of the function $f_{b,N}(n)$ is order of (b mod N), i.e. $ord_N(b) := \min\{m \in N | b^m \bmod N = 1\}$.
b is a natural number $b < N$ which has no common divisor with N (otherwise, the problem is solved).
With the Shor's algorithm we can determine this period in a number of computational steps, which for $N \to \infty$ grows asymptotically as $O((\log_2 N)^3)$.

If the period is **odd**, we choose a different b with $\gcd(b, N) = 1$ and again determine the period of $f_{b,N}$. This is repeated until a b is found such that $f_{b,N}$ has an **even** period $r \in \mathbb{N}$!

If the period is **odd**, we choose a different b with $\gcd(b, N) = 1$ and again determine the period of $f_{b,N}$. This is repeated until a b is found such that $f_{b,N}$ has an **even** period $r \in \mathbb{N}$!
Let r is an even period of $f_{b,N}$, i.e. $b^r \bmod N = 1 =>$
$(b^r - 1) \bmod N = 0 =>$

$$(b^{r/2} + 1)(b^{r/2} - 1) \bmod N = 0.$$

If the period is **odd**, we choose a different b with $\gcd(b, N) = 1$ and again determine the period of $f_{b,N}$. This is repeated until a b is found such that $f_{b,N}$ has an **even** period $r \in \mathbb{N}$!
Let r is an even period of $f_{b,N}$, i.e. $b^r \bmod N = 1 =>$
$(b^r - 1) \bmod N = 0 =>$

$$(b^{r/2} + 1)(b^{r/2} - 1) \bmod N = 0.$$

$=>$ N and $(b^{r/2} + 1)$ $(b^{r/2} - 1)$ have common divisors and via the Euclid's algorithm to N and $(b^{r/2} + 1)$ or to N and $(b^{r/2} - 1)$ we obtain a factor of N.

1. Preparation of the input register and the initial state
2. Exploiting massive quantum parallelism
3. Application of the quantum FOURIER transform
4. Probability in measurement of the input register
5. Probability to find r as the denominator in the continued fraction approximation
6. Aggregation of the number of computational steps

NEXT LECTURE

OCTOBER 28, 2022

THANK YOU FOR YOUR ATTENTION!

БЛАГОДАРЯ ЗА ВНИМАНИЕТО!