Kwantowy algorytm Shora
- czyli jak łamać szyfry

Wiesław Płaczek

Zakład Zastosowań Metod Obliczeniowych

Instytut Fizyki Uniwersytetu Jagiellońskiego

Plan

- Szyfrowanie informacji.
- Kryptosystem RSA.
- Algorytm Shora – odsłona 1.
- Algorytm Shora – odsłona 2.
- Algorytm Shora – odsłona 3.
- Kwantowa transformacja Fouriera.
- Znajdowanie okresu funkcji.
- Podsumowanie.
- Literatura.

Szyfrowanie informacji

- *Powszechnie stosowana metoda ochrony informacji przesyłanej przez przez niepewne łącza.*
- Standard szyfrowania danych *(data-encryption standard – DES)* przyjęty przez National Bureau of Standards, USA:
  - Tajny klucz (lub klucze) ***k.***
  - Algorytm szyfrowania danych ***E_k***.
  - Algorytm deszyfrowania danych ***D_k***.
  - **Dla każdego komunikatu *m* muszą być spełnione warunki:**
  1. ***D_k(E_k(m)) = m.***
  2. Zarówno ***E_k*** jak i ***D_k*** można wykonać wydajnie.
  3. Bezpieczeństwo systemu zależy tylko od tajności klucza ***k***, a nie od tajności algorytmów ***E_k*** i ***D_k***.

Szyfrowanie informacji – c.d.

- **Podstawowy problem:**
- ***Jak bezpiecznie przekazywać klucze do nadawcy i odbiorcy?***
  - Np. „pięta Achillesowa” niemieckiej Enigmy.
- W środowiskach sieci komunikacyjnych wygodnym rozwiązaniem jest ***szyfrowanie z kluczem jawnym*** *(public-key encryption)*:
  - Dwa różne klucze: ***klucz jawny*** *(public key)* – do szyfrowania oraz ***klucz prywatny*** *(private key)* – do deszyfrowania.
  - Odbiorca informacji ujawnia publicznie klucz do szyfrowania, natomiast klucz do deszyfrowania jest jego ścisłą tajemnicą.
  - Dowolny nadawca może zaszyfrować swoją informację przy użyciu jawnego klucza, ale tylko odbiorca może ją odczytać przy pomocy prywatnego klucza do deszyfrowania.
- ***Czy da się skonstruować dobry algorytm realizujący ten schemat?***

Kryptosystem RSA

- Powszechnie stosowanym kryptosystemem z kluczem jawnym jest **metoda RSA** (Rivest, Shamir, Adleman: 1977) – oparta na **faktoryzacji** (rozkład na czynniki pierwsze) **liczb naturalnych**.

- Podstawy:
  - Niech:

    $N = p \cdot q$, gdzie p i q – liczby pierwsze (duże)

  - Funkcja Eulera:

    $\varphi(N)$ – ilość liczb mniejszych niż N względnie pierwszych z N:

    $$\varphi(N) = (p-1)(q-1)$$

  - Notacja:

    $a \equiv b \pmod N$ oznacza: $a \bmod N = b \bmod N$.

  - Twierdzenie Eulera:

    Jeżeli $a < N$ i a jest względnie pierwsze z N (nie mają wspólnych dzielników), to:

    $$a^{\varphi(N)} \equiv 1 \pmod N$$

Kryptosystem RSA – c.d.

- Uczestnicy komunikacji: **Alice** i **Bob**

1. **Bob** bierze dwie duże liczby pierwsze p i q, i wylicza: $N = p \cdot q$ oraz $\varphi(N)$.
2. **Bob** wybiera losowo liczbę $e < \varphi(N)$, która jest względnie pierwsza z $\varphi(N)$, tzn. $\mathrm{GCD}(e, \varphi(N)) = 1$ (GCD – *greatest common divisor*).
3. Następnie ogłasza publicznie (np. w CNN): e i N.
4. **Alice** zamienia swoją wiadomość na liczbę $a < N$ (dłuższe wiadomości można podzielić na odpowiednie fragmenty i każdy z nich szyfrować oddzielnie) i oblicza: $b = f(a) = a^e \bmod N$
5. Tak zaszyfrowaną wiadomość **Alice** wysyła do **Boba**.
6. **Bob** wylicza: →np. przy pomocy algorytmu Euklidesa dla $\mathrm{GCD}(e, \varphi(N))$

$$d = e^{-1} \bmod \varphi(N) \qquad \text{tzn.} \quad ed \equiv 1 \pmod{\varphi(N)}$$

7. Następnie **Bob** odszyfrowuje wiadomość **Alice** przy pomocy prywatnego klucza d:

$$f^{-1}(b) = b^d \bmod N = a^{ed} \bmod N = a \cdot [a^{\varphi(N)}]^m \bmod N, \quad (m \text{ - liczba naturalna})$$

$$= \underset{\text{tw. Eulera}}{} \quad a \bmod N = a \qquad (\text{bo } a < N).$$

Kryptosystem RSA – c.d.

- **Czy kryptosystem RSA jest bezpieczny?**
  - Aby odszyfrować wiadomość trzeba wyliczyć wartość klucza deszyfrującego d, a do tego potrzebna jest znajomość czynników pierwszych p i q liczby N.
  - **Rozkład dużych liczb naturalnych na czynniki pierwsze (faktoryzacja) jest klasycznie trudnym problemem!**
    - ✓ Najlepszy **klasyczny** algorytm, tzw. „sito ciała liczbowego" (J. Pollard, 1988) ma (sub)eksponencjalny czas faktoryzacji:

$$t_F \propto \exp[(64/9)^{1/3} (\ln N)^{1/3} (\ln \ln N)^{2/3}]$$

❑ W roku 1994 użyto 1600 szybkich stacji roboczych do faktoryzacji liczby **RSA129**, tzn. znalezienia dwóch dużych czynników pierwszych liczby **129**-cyfrowej – obliczenia zajęły 8 miesięcy (5000 MIPS lat).

✓ Można oszacować, że dla liczby **400**-cyfrowej analogiczne obliczenia zajęłyby:

$$t_F \approx 10^{10} \text{ lat} \qquad \approx \text{wiek wszechświata!}$$

➔ **Dla dostatecznie dużych N szyfr RSA jest praktycznie nie do złamania!**

Kryptosystem RSA – c.d.

Ale …

- W roku 1994 **Peter Shor** wynalazł algorytm kwantowy, tzn. taki, który może być zrealizowany tylko na komputerze kwantowym, dla którego czas faktoryzacji liczby naturalnej ***N*** wynosi:

$$t_F \propto (\ln N)^3$$

- Np. do faktoryzacji liczby **RSA129** komputer kwantowy z zegarem 100 MHz potrzebowałby tylko **kilka sekund**!

- … a dla liczby **400**-cyfrowej **niewiele ponad 1 minutę**!

Ze względu na wielomianową zależność czasu faktoryzacji od rozmiaru (liczby bitów) liczby naturalnej kryptosystem RSA nie jest bezpieczny względem algorytmu Shora!

- Kryptosystem RSA jest powszechnie używany w internecie (np. ssh).

Algorytm Shora – odsłona 1

- Szukamy nietrywialnych rozwiązań kongruencji Legendre'a:

$$x^2 \equiv 1 \pmod{N}$$

czyli: $(x+1)(x-1) \equiv 0 \pmod{N}$

dla $x \in [1,\ldots,N-1]$; (rozwiązania trywialne: $x=1$ i $x=N-1$)

Jeżeli: $N = m \cdot n$ i $\mathrm{GCD}(m,n) = 1,$

to na podstawie ***chińskiego twierdzenia o resztach*** dla kongruencji Legendre'a oprócz rozwiązań trywialnych, tzn. $x \equiv \pm 1 \pmod{N}$, istnieje także rozwiązanie nietrywialne, dla którego czynniki m i n rozdzielone są pomiędzy $(x-1)$ i $(x+1)$.

Zatem: $\mathrm{GCD}(x-1,N)$ i $\mathrm{GCD}(x+1,N) \Rightarrow m$ i $n.$

➢ Jak liczyć $\mathrm{GCD}(a,N)$?

✓ Np. ***algorytm Euklidesa***: $t \propto (\log N)^3.$

Algorytm Shora – odsłona 2

- **Jak znaleźć rozwiązanie kongruencji Legendre'a?**
  - Rozwiązanie kongruencji Legendre'a ma związek z okresem funkcji periodycznej:

$$f_{N,y}(a) = y^a \bmod N, \quad \text{gdzie } y \in \{1,\ldots,N-1\} \text{ i } \mathrm{GCD}(y,N) = 1.$$

    - Jeżeli *r* jest okresem powyższej funkcji, to zachodzi:

$$y^r \equiv 1 \pmod N$$

(tzn. r – tzw. rząd $y \bmod N$).

    - Jeżeli *r* – parzyste, to: $y^{r/2} \equiv x \pmod N$

      jest rozwiązaniem kongruencji Legendre'a: $x^2 \equiv 1 \pmod N$.

- Skąd bierzemy y ?
  - ✓ Wybieramy losowo ze zbioru $\{1,\ldots,N-1\}$.
  - ✓ Można pokazać, że aby znaleźć nietrywialne rozwiązanie dla r parzystego potrzeba wykonać $< 2\log N$ losowań.
  - ✓ Złożoność obliczeniowa $y^a \bmod N$: $\mathcal{O}((\log N)^3)$

Algorytm Shora – odsłona 3

- **Jak znaleźć okres funkcji** $f_{N,y}(a) = y^a \bmod N$?
  - Klasycznie – trudny problem (brak efektywnego algorytmu):

$$t \propto \exp(\log N)$$

  - Kwantowo – Shor pokazał, że okres funkcji można znaleźć wydajnie przy użyciu **kwantowej transformacji Fouriera**:

$$t_{\mathrm{QFT}} \propto (\log N)^2$$

  - Dyskretna transformacja Fouriera – klasycznie:

$$\mathrm{F} : (x_0, \ldots, x_{N-1}) \rightarrow (y_0, \ldots, y_{N-1}), \qquad x_i, y_i \in C \ \text{(liczby zespolone)}$$

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2\pi i j k / N}$$

    - Złożoność obliczeniowa dyskretnej transformacji Fouriera:
      - Naiwnie : $\propto [\exp(\ln N)]^2$
      - Tzw. szybka transformata Fouriera (FFT) : $\propto \ln N \exp(\ln N)$

Kwantowa transformacja Fouriera

- Kwantowa transformacja Fouriera (**QFT**) – operator unitarny działający na wektory w N-wymiarowej przestrzeni Hilberta.

Niech $\{|0>,...,|N-1>\}$ - baza ortonormalna w $H^{\otimes N}$,

$$\hat{F}_q : |j> \rightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i jk/N} |k>.$$

Niech $N = 2^n$ oraz $|j> = |j_1 j_2 ... j_n>,\ j_k \in \{0,1\}, k = 1,..., n.$

Ułamki binarne: $0.j_l j_{l+1} ... j_m \equiv \frac{j_l}{2} + \frac{j_{l+1}}{2^2} + ... + \frac{j_m}{2^{m-l+1}}.$

- Kwantową transformację Fouriera dla wektorów bazowych można przedstawić w postaci:

$$\hat{F}_q |j_1 ... j_n> = 2^{-\frac{n}{2}} (|0> + e^{2\pi i 0.j_n} |1>)(|0> + e^{2\pi i 0.j_{n-1} j_n} |1>)...$$

$$...(|0> + e^{2\pi i 0.j_1 j_2 ... j_n} |1>).$$

← n iloczynów

Kwantowa transformacja Fouriera

- **Czy da się wydajnie zaimplementować kwantową transformację Fouriera?**
  - Tak, ale tylko na **komputerze kwantowym!**
- **Co to jest komputer kwantowy?**
  - Komputer kwantowy jest to maszyna operująca na stanach kwantowych, tzw. **qubitach**, w oparciu o prawa mechaniki kwantowej.
  - Qubit – wektor w 2-wymiarowej przestrzeni Hilberta:

$$|\psi> = a\,|0> + b\,|1>, \qquad |a|^2 + |b|^2 = 1,$$

gdzie $a, b \in C$, a $\{|0>, |1>\}$ - baza ortonormalna w H^2

  - Stan n-qubitowy – wektor w przestrzeni Hilberta o wymiarze 2^n:

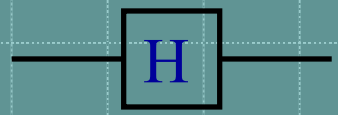
$$|\Psi> = \sum_{k=0}^{2^n-1} c_k\,|k>, \qquad \sum_{k=0}^{2^n-1} |c_k|^2 = 1, \quad c_k \in C.$$

Kwantowa transformacja Fouriera

- Obliczenia kwantowe – transformacje unitarne wektorów w przestrzeni Hilberta.
- Dowolny układ obliczeniowy można zbudować z kilku podstawowych elementów, tzw. bramek kwantowych.
- Bramki używane do kwantowej transformacji Fouriera:
  - Bramka **Hadamarda** – 1-qubitowa:

$$\hat{H}|x>=\frac{1}{\sqrt{2}}(|0>+e^{2\pi i 0.x}|1>),\ x\in\{0,1\}.$$

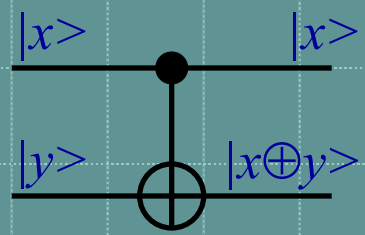
Ozn.



  - Bramka **Control-NOT (CN)** – 2 -qubitowa:

$$\hat{U}_{CN}|x,y>=|x,x\oplus y>,\ x,y\in\{0,1\}.$$

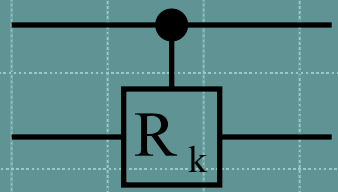
Ozn.



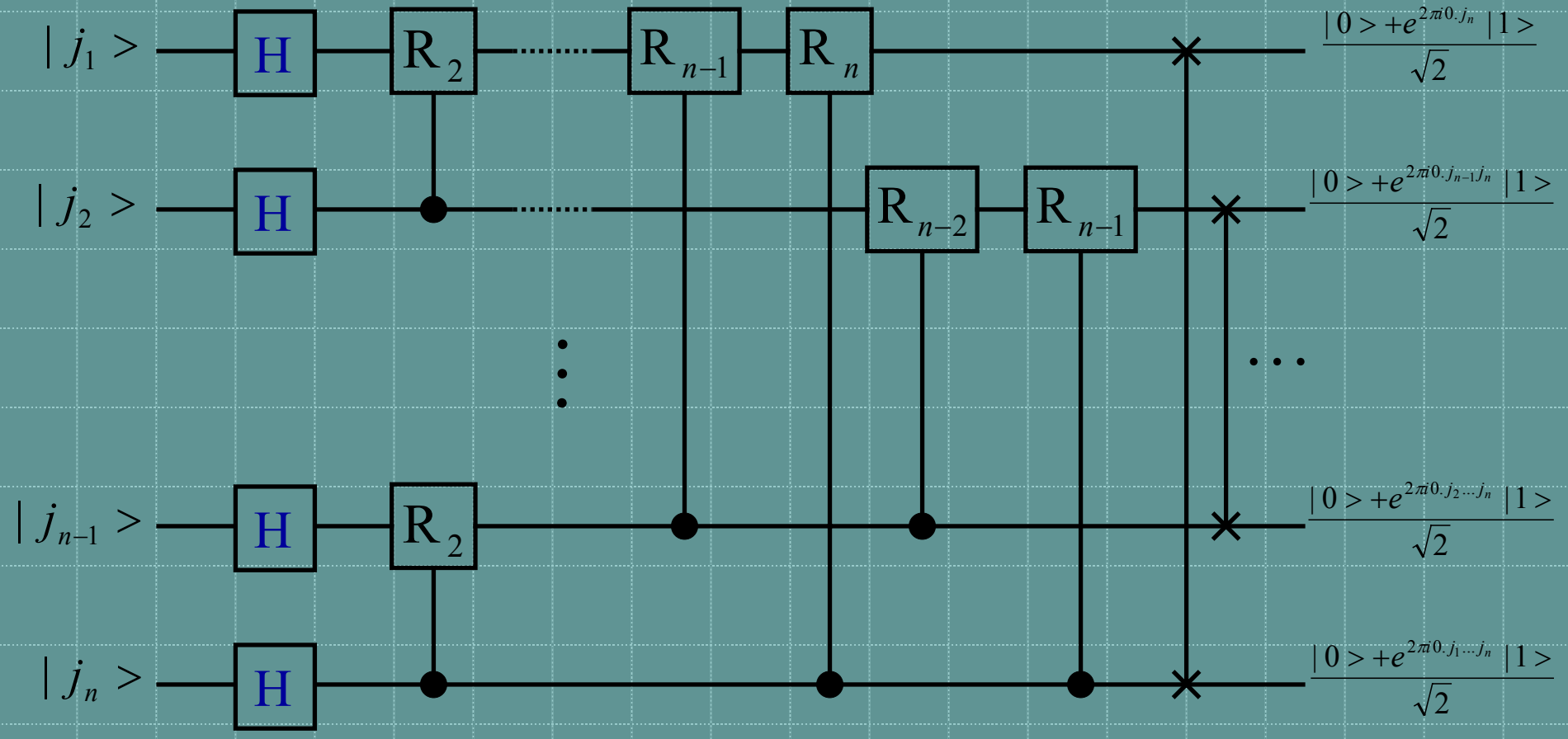
  - Bramka $\text{Control-}R_k$:

gdzie: $\hat{R}_k|x>=e^{2\pi i x/2^k}|x>,\ x\in\{0,1\}.$

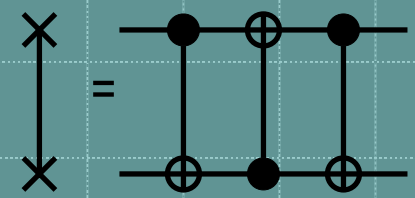
Ozn.



Układ do kwantowej transformacji Fouriera



Bramka **SWAP**



- Liczba bramek: $\frac{n(n+1)}{2}+\frac{n}{2}=\frac{n(n+2)}{2}=\mathcal{O}(n^2)$

Znajdowanie okresu funkcji

- Potrzeba dwóch rejestrów kwantowych:
  - Dla argumentu: t-qubitowy, gdzie t powinno być takie, że $N^2 \le 2^t < 2N^2$
  - Dla wartości funkcji: n-qubitowy, gdzie $n = \lceil \log_2 N \rceil$

$$|\psi_0> = \underbrace{|0...0>}_{t}\underbrace{|0...0>}_{n}$$

- Pierwszy rejestr wprowadzamy w stan superpozycyjny działając na każdy z t-qubitów bramką Hadamarda $\left(\hat{H}|0> = (|0> + |1>)/\sqrt{2}\right)$:

$$|\psi_1> = \frac{1}{\sqrt{K}} \sum_{j=0}^{K-1} |j>|0>, \qquad K = 2^t.$$

- Na drugi rejestr działamy bramką: $\hat{U}_f$ dla $f(j) = y^j \bmod N$

$$|\psi_2> = \frac{1}{\sqrt{K}} \sum_{j=0}^{K-1} |j>|y^j \bmod N>$$

- Mierzymy drugi rejestr $\rightarrow$ otrzymujemy pewną liczbę m

$\Rightarrow$ pierwszy rejestr jest superpozycją stanów $|j>$: $y^j \bmod N = m$

Znajdowanie okresu funkcji

- Niech r – okres funkcji $f(j)$ i załóżmy, że K jest podzielne przez r (dla uproszczenia analizy; algorytm działa również bez tego założenia, ale rachunki stają się bardziej skomplikowane):
- Liczba stanów, które „przeżyły pomiar” = $K/r \Rightarrow$ stan układu:

$$|\psi_3> = \sqrt{\frac{r}{K}} \sum_{k=0}^{K/r-1} |kr+l> |m>, \quad \text{gdzie } l \in \{0,\ldots,r-1\}.$$

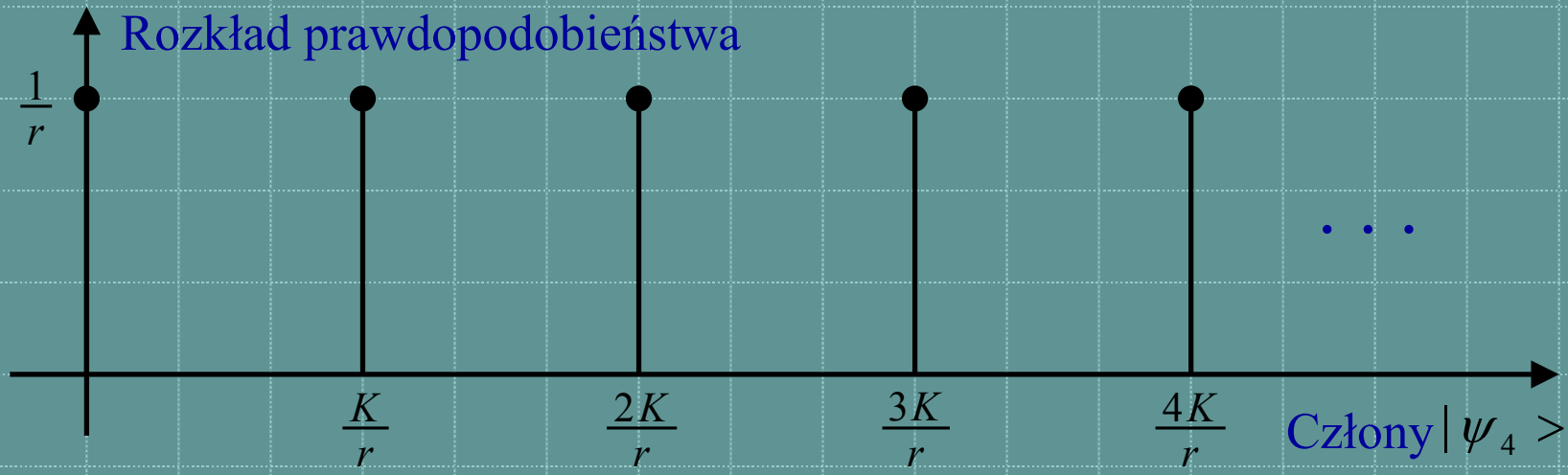
- Jak znaleźć r ?
- Do 1-go rejestru stosujemy odwrotną transformację Fouriera $\hat{F}_q^{\dagger}$:
  - W oparciu tożsamość:

$$\frac{1}{N} \sum_{j=0}^{N-1} e^{2\pi ijk/N} = \begin{cases} 1 \text{ dla } k \bmod N = 0, \\ 0 \text{ dla } k \bmod N \neq 0, \end{cases}$$

  - Dostajemy stan układu:

$$|\psi_4> = \frac{1}{\sqrt{r}} \sum_{\lambda=0}^{r-1} e^{-2\pi i \lambda l/r} \, |\lambda \tfrac{K}{r}> |m>, \quad \text{gdzie } \lambda = 0,\ldots,r-1.$$

Znajdowanie okresu funkcji



Podsumowanie

- Kwantowy algorytm Shora umożliwia faktoryzację liczb naturalnych w czasie wielomianowym, co stanowi zagrożenie dla kryptosystemu RSA!
- Shor podał również algorytm rozwiązujący problem dyskretnych logarytmów w czasie wielomianowym.
  - Mając daną liczbę pierwszą p oraz dwie liczby naturalne $g, y < p$ znaleźć najmniejszą liczbę naturalną x, taką że: $g^x \equiv y \pmod{p}$.
  - Również szeroko stosowany w kryptografii z kluczem jawnym.
- Roku 2001: grupa 6 eksperymentatorów z IBM i Uniwersytetu Stanford dokonała implementacji algorytmu Shora na komputerze kwantowym opartym o jądrowy rezonans magnetyczny.
  - Dokonano rozkładu na czynniki pierwsze liczby $15 = 3 \cdot 5$.
- Czy możliwa będzie kryptografia w dobie komputerów kwantowych?
- **Tak – mechanika kwantowa umożliwia bezpieczne przesyłanie klucza** (np. drogą teleportacji kwantowej).

Literatura

- P. Shor, *„Algorithms for Quantum Computation: Discrete Logarithm and Factoring”,* Proc. 35th Annual Symposium on Foundations of Computer Science (IEEE, Los Alamitos, CA, 1994), p. 124 and SIAM J. Comput. **26** (1997) 1484.
- A. Eckert, R. Jozsa, *„Quantum computation and Shor’s factoring algorithm”*, Rev. Mod. Phys. **68** (1996) 733.
- L.M.K. Vandersypen, M. Steffen, G. Breyta, C.S. Yannoni, M.H. Sherwood, I.L. Chuang, *„Experimental ralization of Shor’s quantum factoring algoritm using nuclear magnetic resonance”,* Nature **414** (2001) 883.
- C. Lavor, L.R.U. Manssur, R. Portugal, *„Shor’s Algorithm for Factoring Large Integers”,* quant-ph/0303175.
- A.J.G. Hey, ed., *„Feynman and Computation”*, Perseus Books, 1999.
- M.A. Nielsen, I.L. Chuang, *„Quantum Computation and Quantum Information”*, Cambridge University Press, 2000.
- J. Preskill, *„Lecture Notes for Physics 229: Quantum Information and Computation”*, http://www.theory.caltech.edu/~preskill/ph229.
- D.E. Knuth, *„Sztuka programowania”*, tom 2, WNT, Warszawa, 2002.