EIROPAS SOCIĀLAIS FONDS
EIROPAS SAVIENĪBA
IEGULDĪJUMS TAVĀ NĀKOTNĒ

Cryptography that is secure against quantum computers?

Andris Ambainis
University of Latvia

European Social Fund project “Datorzinātnes pielietojumi un tās saiknes ar kvantu fiziku” Nr.2009/0216/1DP/1.1.1.2.0/09/APIA/VIAA/044

Quantum computing

- New model of computation based on quantum physics.
- More powerful than conventional computing.

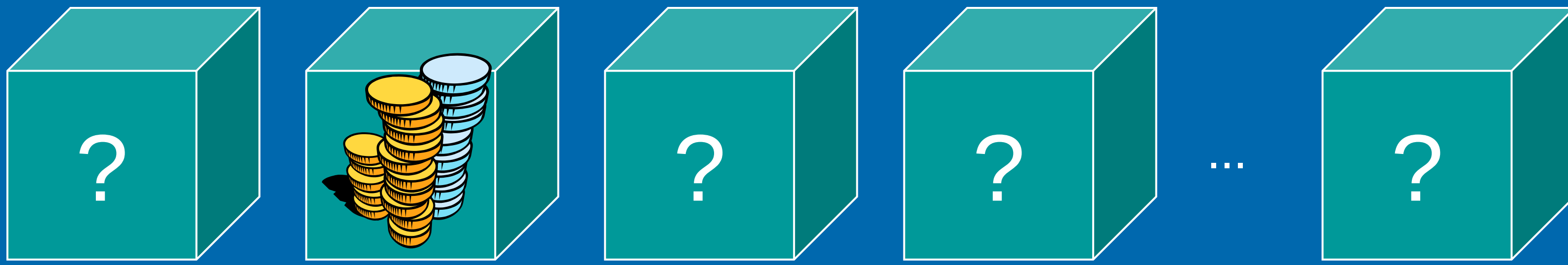
Factoring

- 6231540623 = 93599 * 66577.
- Find 6231540623?

- For large (300 digit) numbers conventional computers are too slow.

Shor, 1994: quantum computers can factor large numbers efficiently.

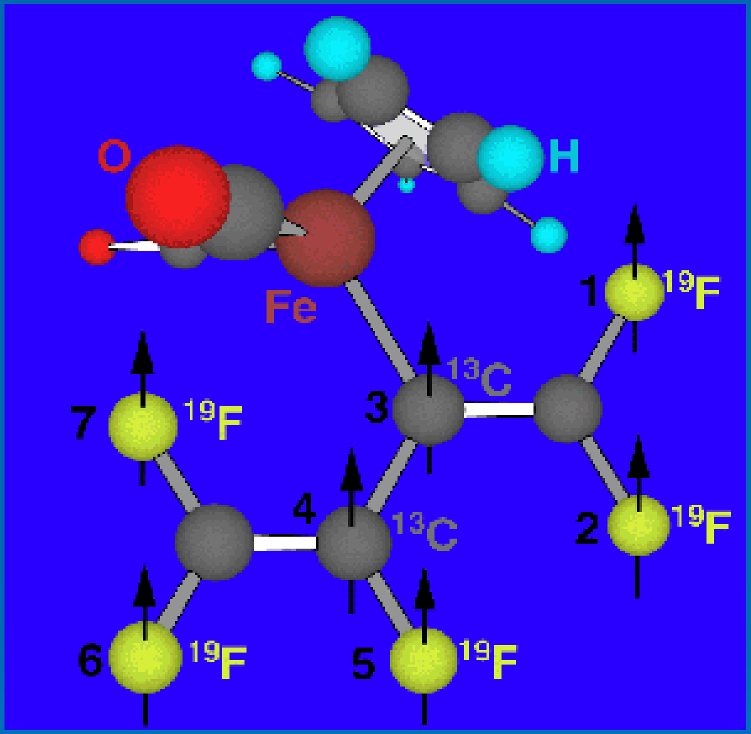
Quantum search



- N objects;
- Find an object with a certain property.

Grover, 1996: can be done in $O(\sqrt{N})$ quantum steps.

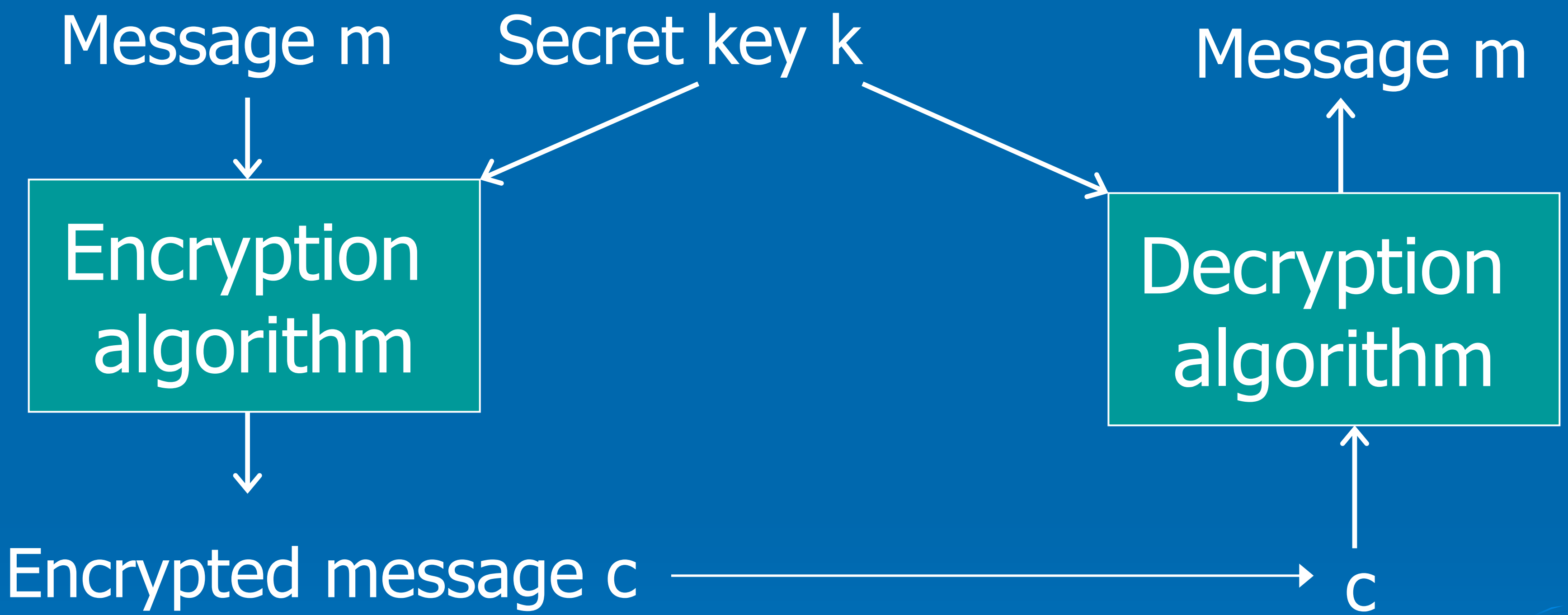
13 bit quantum computer (MIT/Waterloo, 2004)



- Quantum computer = molecule.
- Quantum bits = nuclear spins.
- Manipulate nuclear spins with magnetic field.

Post-quantum cryptography

Cryptography

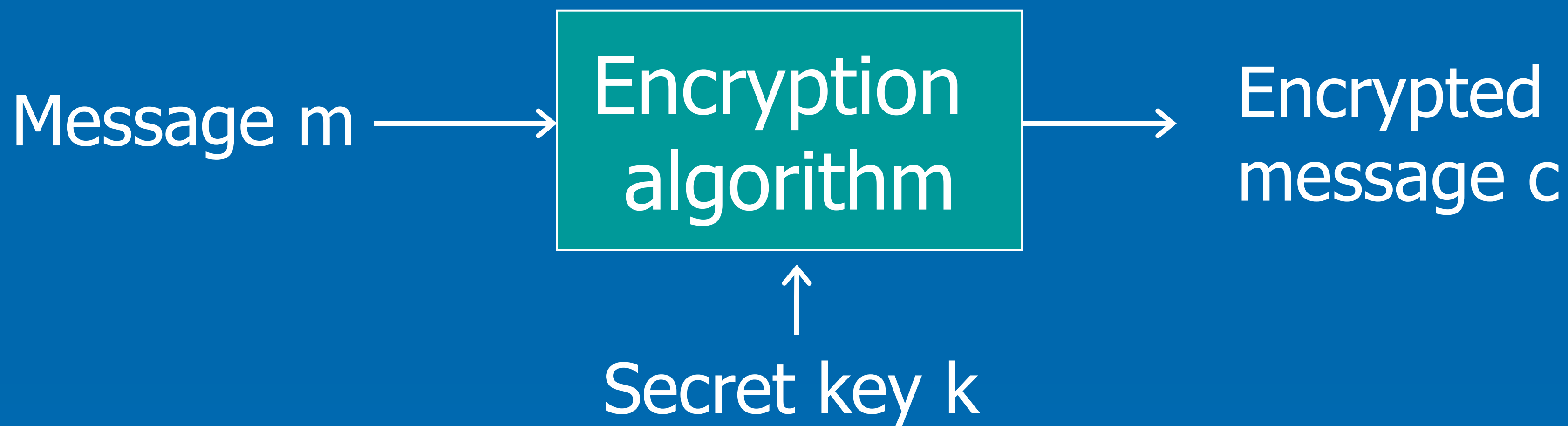


Symmetric cryptography: same key k for encryption and decryption

4-rotor Enigma, 1942

Codebreaking by exhaustive search

- For each k, test:



Classically: N steps;

Quantum (Grover): $O(\sqrt{N})$ steps.

Codebreaking by exhaustive search

- 64 bit key $\rightarrow N = 2^{64}$ secret keys.

$N = 2^{64} \approx 18,000,000,000,000,000,000.$
$\sqrt{N} = 2^{32} \approx 4,294,000,000.$

Is this a big advantage for quantum computers?

128 bit key $\rightarrow N = 2^{128}$, $\sqrt{N} = 2^{64}$.

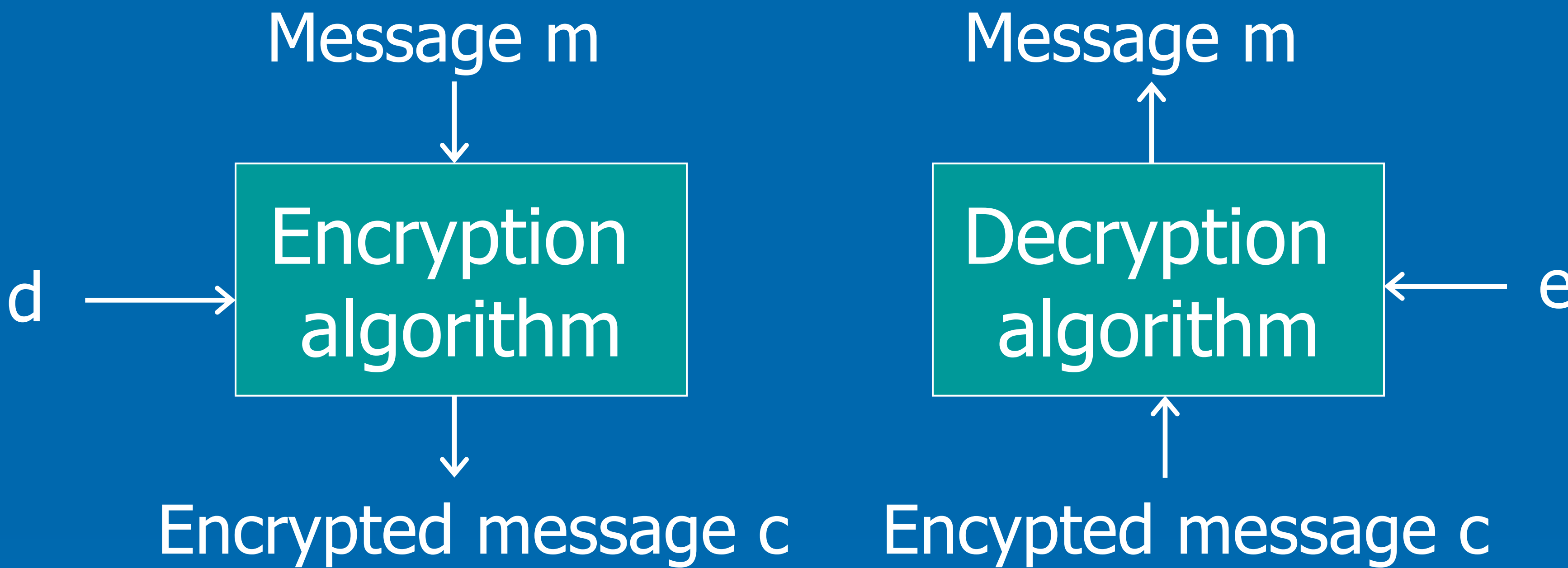
Cryptography

amazon.com

4252 1890 6767 1345

Where do we get a secret key?

Public-key cryptography (RSA, 1977)



One key for encryption – d, one for decryption – e.

Computing e from d – difficult.

Public key cryptography

e

Encrypt(4252 ..., e)

amazon.com

4252 1890 6767 1345

Eavesdropper does not have decryption key d

RSA

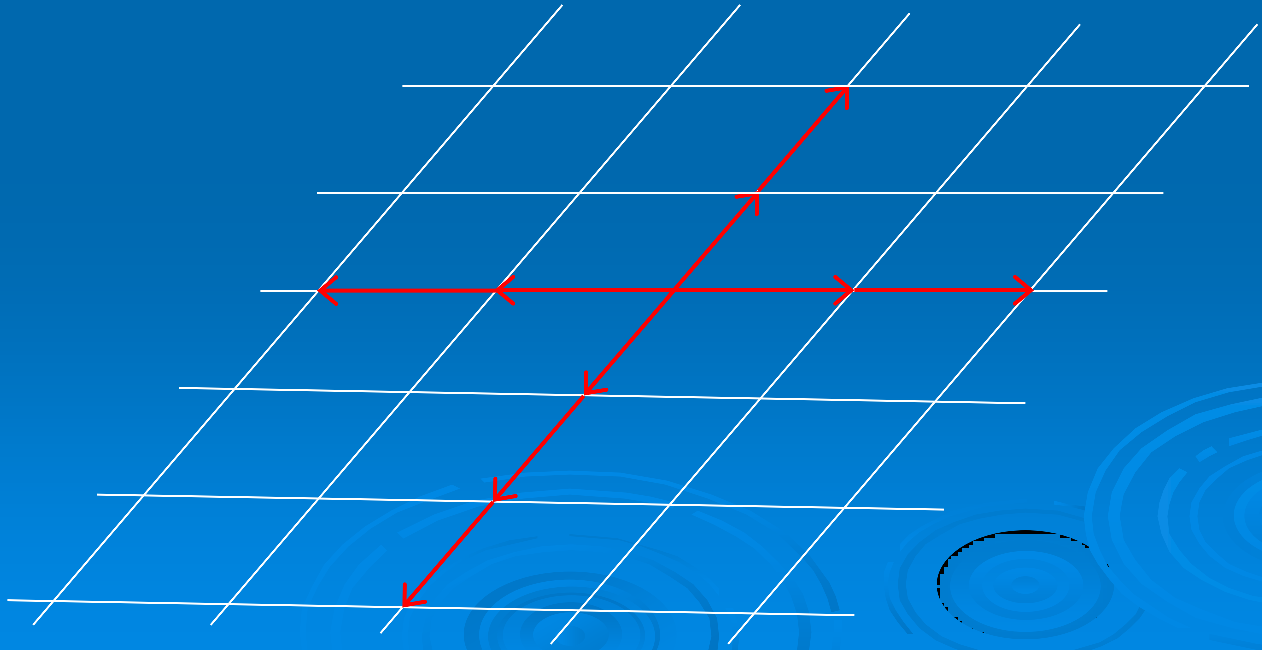
- Rivest, Shamir, Adleman, 1977;
- Computing decryption key d from encryption key e is roughly equivalent to factoring a large number.
- Factoring large (300-digit) number N = pq into p and q is very difficult.

Factoring becomes easy if we have a quantum computer.

Lattice-based cryptography

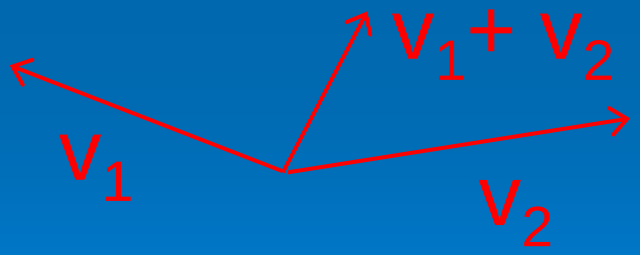
Lattices

- Set of vectors $v_1, ..., v_m$ in n dimensions;
- Lattice $L = \{ a_1 v_1 + ... + a_m v_m :$
  $a_1, ..., a_m$ - integers$\}$.



Lattices

- Lattice L = { $a_1v_1+...+a_mv_m$: $a_1, ..., a_m$ - integers}.
- Shortest vector problem (SVP): given $v_1, ..., v_m$, find the shortest vector in L.



Breaking a lattice-based cryptosystem ≈ SVP

Versions of SVP

- SVP: find the shortest vector v_{min} in L;
- γ-SVP: find a vector v: $||v|| \leq \gamma \, ||v_{min}||$;
- γ-Unique-SVP: find v_{min} if we are promised that $||v|| \geq \gamma \, ||v_{min}||$, unless $v = c \bullet v_{min}$.

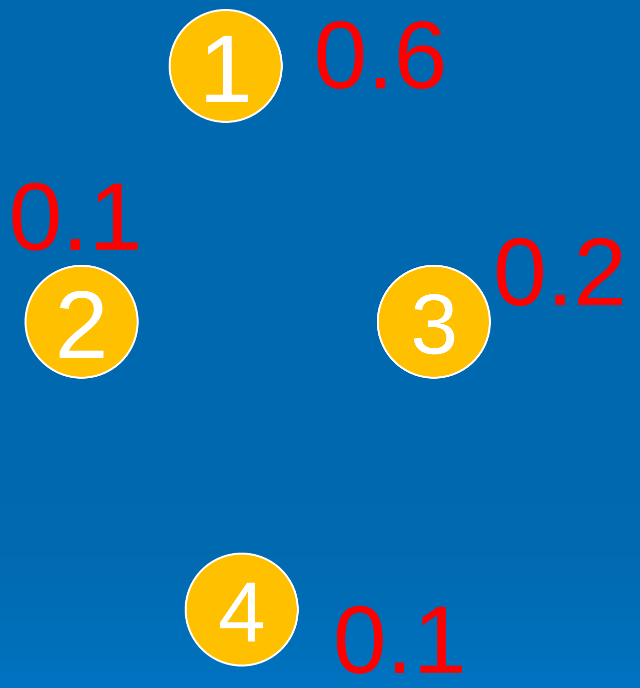
SVP is NP-hard;
Hardness of γ-SVP and γ-Unique-SVP depends on γ.

γ-Unique-SVP

- Task: find v_{min} if we are promised that $||v|| \geq \gamma\ ||v_{min}||$, unless $v = c \bullet v_{min}$.
- Lenstra-Lenstra-Lovasz, 1982: efficiently solvable if $\gamma = 2^n$.
- Thought to be difficult for classical algorithms if $\gamma = n^c$.
- Regev, 2002: idea for quantum algorithm.

Quantum computing: the model

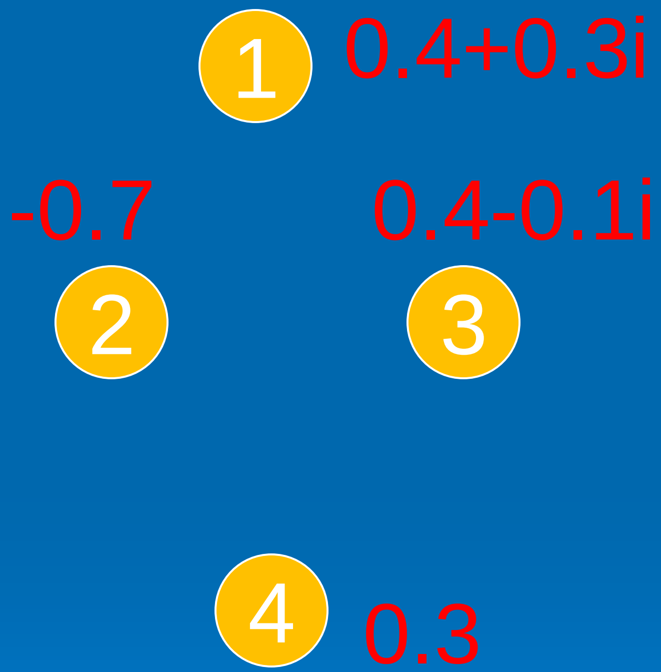
Probabilistic computation



- Probabilistic system with finite state space.
- Current state: probabilities p_i to be in state i.

$$\sum_i p_i = 1$$

Quantum computation

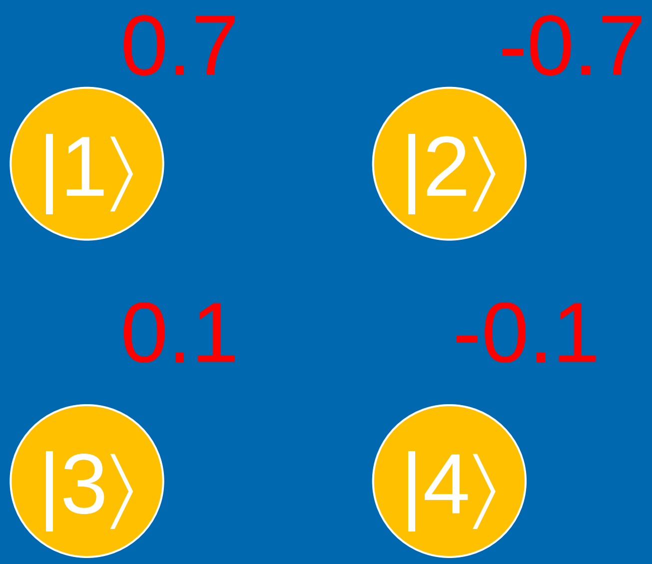


- Current state: amplitudes α_i to be in state i.

$$\sum_i |\alpha_i|^2 = 1$$

For most purposes, real (but negative) amplitudes suffice.

Notation

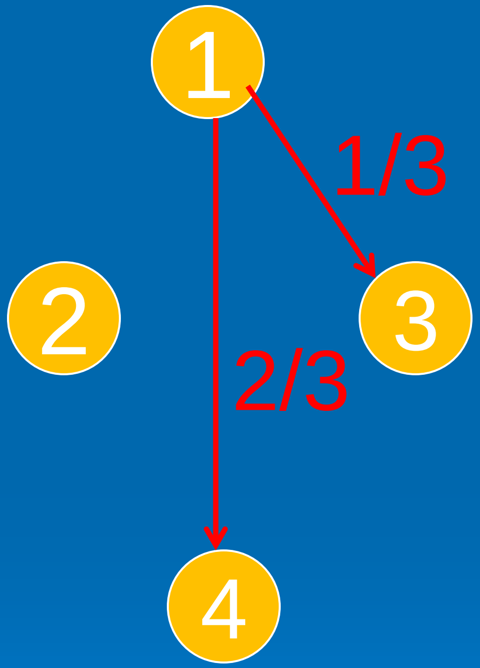


- Basis states $|1\rangle$, $|2\rangle$, $|3\rangle$, $|4\rangle$.

$$|\Psi\rangle = \begin{pmatrix} 0.7 \\ -0.7 \\ 0.1 \\ -0.1 \end{pmatrix}$$

$$|\Psi\rangle = 0.7\,|1\rangle - 0.7\,|2\rangle + 0.1\,|3\rangle - 0.1\,|4\rangle.$$

Probabilistic computation



- Pick the next state, depending on the current one.
- Transitions: r_{ij} - probabilities to move from i to j.

Probabilistic computation

- Probability vector $(p_1, \ldots, p_N)$.
- Transitions:

$$\begin{pmatrix} p'_1 \\ \ldots \\ p'_N \end{pmatrix} = \begin{pmatrix} r_{11} & \ldots & r_{1N} \\ \ldots & \ldots & \ldots \\ r_{N1} & \ldots & r_{NN} \end{pmatrix} \begin{pmatrix} p_1 \\ \ldots \\ p_N \end{pmatrix}$$

after the transition

transition probabilities

before the transition

Quantum computation

- Quantum state

$$\alpha_1|1\rangle + \alpha_2|2\rangle + \ldots + \alpha_N|N\rangle$$

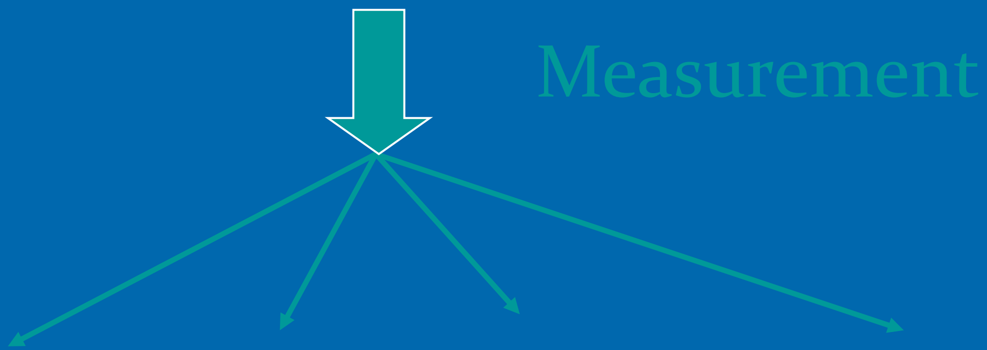
- Transitions

$$\begin{pmatrix} u_{11} & \ldots & u_{1n} \\ \ldots & \ldots & \ldots \\ u_{n1} & \ldots & u_{nn} \end{pmatrix} \begin{pmatrix} \beta_1 \\ \ldots \\ \beta_n \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \ldots \\ \alpha_n \end{pmatrix}$$

U-unitary (preserves $\Sigma_i |\alpha_i|^2 = 1$).

Measurements

$$|\Psi\rangle = \alpha_1 |1\rangle + \alpha_2 |2\rangle + \ldots + \alpha_M |M\rangle$$



	1	2	…	M
prob.	$|\alpha_1|^2$	$|\alpha_2|^2$		$|\alpha_M|^2$

Partial measurements

$$|\Psi\rangle = \alpha_{00}\,|00\rangle + \alpha_{01}\,|01\rangle + \alpha_{10}\,|10\rangle + \alpha_{11}\,|11\rangle$$

Measure the 1st bit

$$\alpha_{00}\,|00\rangle + \alpha_{01}\,|01\rangle$$

$$\alpha_{10}\,|10\rangle + \alpha_{01}\,|11\rangle$$

Quantum algorithm for unique-SVP?

Quantum algorithm for SVP?

- Set of vectors $v_1, ..., v_m$ in n dimensions;
- Lattice L = { $a_1v_1+...+a_mv_m$:
  $a_1, ..., a_m$ - integers}.
- Task: find v_{min} if we are promised that $||v|| \geq \gamma ||v_{min}||$, unless $v = c \bullet v_{min}$.

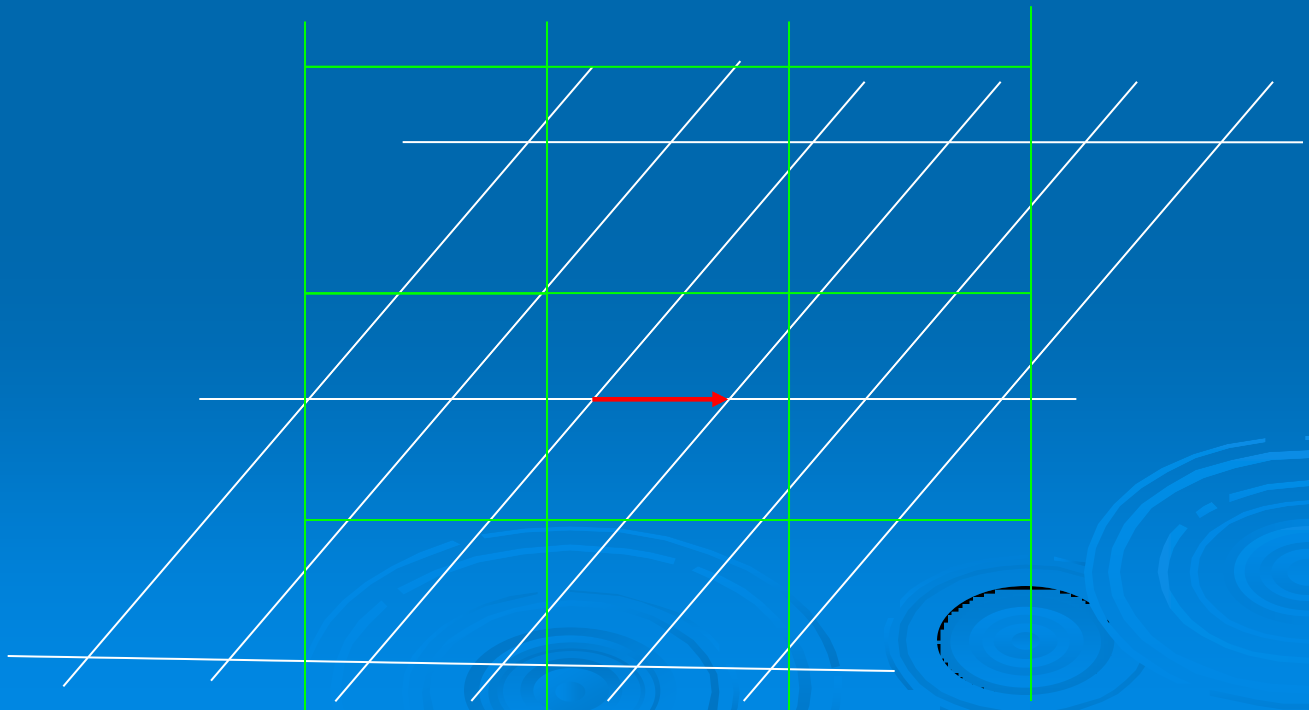
Step 1: prepare

$$\sum_{a_1,\ldots,a_n \in \{-M,\ldots,M\}} \left| a_1 x_1 + a_2 x_2 + \ldots + a_m x_m \right\rangle$$

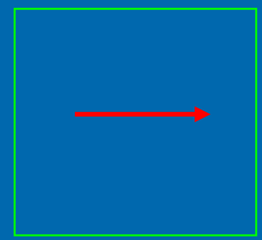
Quantum algorithm for SVP?

Step 2: measure the most significant bits of

$$\sum_{a_1,\ldots,a_n \in \{-M,\ldots,M\}} \left| a_1 x_1 + a_2 x_2 + \ldots + a_m x_m \right\rangle$$



Result



Quantum state:

$$|x\rangle + |x + v_{\min}\rangle$$

$$|x\rangle + |x + v_{\min}\rangle + |x + 2v_{\min}\rangle$$

Missing step

- How do we get v_{min} from

$$\left| x \right\rangle + \left| x + v_{\min} \right\rangle ?$$

Measuring the state gives x or $x+v_{min}$, but not v_{min}.

Period-finding

- Basis states |1⟩, |2⟩, ..., |N⟩.
- State

$$|x\rangle + |x+r\rangle + |x+2r\rangle + \ldots + |x+kr\rangle$$

Quantum Fourier Transform

One of numbers $\frac{N}{r}, \frac{2N}{r}, \ldots$

Fourier sampling

Open problems

- Can we extract v_{min} from

$$|x\rangle + |x + v_{\min}\rangle ?$$

- Fourier sampling provides enough information;
- Computing v_{min} from this information is difficult.

Hidden subgroup problem

Hidden Subgroup Problem (HSP)

- Group G, function F: $G \to S$.
- Promise: subgroup $H \subseteq G$ such that

  $F(x) = F(y) \leftrightarrow x = yz, z \in H$.

  (equivalent: $F(x) = F(y) \leftrightarrow x, y \in xH$)
- Task: find H.

Example: period-finding

- Group: G = Z (integers);
- Subgroup: H = k Z (integers divisible by k).
- Promise: f(x) = f(y) ↔ x= y + kx.

f(m) = f(m+k) = f(m+2k) = ...

- Task: find k.

Efficient quantum algorithm,
subroutine for factoring algorithm

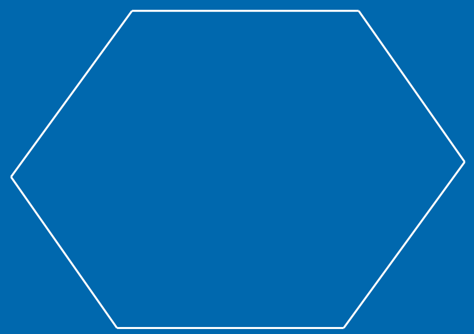
Hidden Subgroup Problem (HSP)

- Group G, function F: $G \rightarrow S$.
- Promise: subgroup $H \subseteq G$ such that

  $F(x) = F(y) \leftrightarrow x = yz,\ z \in H.$
- Task: find H.

- Abelian G: polynomial time quantum algorithms;
- Non-abelian G: open.

Dihedral HSP



- Group of symmetries of regular N-gon.
- (x, y), $x \in \{0, 1, ..., N-1\}$, $y \in \{0, 1\}$.
- The most difficult case: $H=\{(0, 0), (k, 1)\}$;
- Task: find k.
- Equivalent to $f(x, 0) = f((x+k) \bmod N, 1)$.

Hidden shift problem

Connection to SVP

- f(x, 0) = f((x+k) mod N, 1).

$$\sum_{x,y} |x, y\rangle \rightarrow \sum_{x,y} |x, y, f(x, y)\rangle$$

Measure f(x, y).

$$|x,0\rangle + |(x+k) \bmod N, 1\rangle$$

SVP:

$$|x\rangle + |x + v_{\min}\rangle$$

Complexity of dihedral HSP

- Promise: f(x, 0) = f((x+k) mod N, 1).
- Task: find k.

- Goal: $O(\log^c N)$ time quantum algorithm.

- Solvable with O(log N) evaluations of f.
- Solvable in time $2^{O\left(\sqrt{\log N}\right)}$

McEliece cryptosystem

McEliece cryptosystem

- Based on coding theory;
- Public key:

$$G = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

Matrix of an error-correcting code + some scrambling

- Private key: how G was generated.

McEliece: encryption

$$v = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \xrightarrow{\text{encode}} Gv = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} \xrightarrow{\text{+noise e}} Gv + e$$

Decoding Gv+e $\rightarrow$ v can be performed if we know the structure of G.

McEliece: decryption

- $G = P G' A$,
  - P – permutation matrix.
  - G' – generator matrix of efficiently decodable error correcting code;
  - A – invertible matrix;

$$Gv+e \xrightarrow{P^{-1}} G'A\,v+P^{-1}e \xrightarrow{\text{decoding}} A\,v$$

Quantum attack on McEliece

- Codebreaking: given G = PG'A and G', determine A and P.
- Reduces to a difficult instance of HSP.
- Define f(A', P', x): A' – invertible, P' – permutation matrix, x$\in${0, 1}:

$$f(A',P',x) = \begin{cases} P'G'A' & if \;\; x = 0 \\ P'GA' & if \;\; x = 1 \end{cases}$$

Quantum attack on McEliece

$$f(A',P',x)=\begin{cases} P'GA' & if\ \ x=0 \\ P'G'A' & if\ \ x=1 \end{cases}$$

G = PG'A

f(A', P', 0) = f(A'A, PP', 1);

Hidden shift problem: given such f, find A and P.

Quantum attack on McEliece

- HSP for a group that is more complicated than dihedral group.
- Dinh, Moore, Russell, 2010: Standard approach (Fourier sampling) fails to break McEliece, assuming that secret code has:

a) large automorphism group and

b) generator matrix with almost full rank.

Key size

- Key = k*n matrix

$$G = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

- Typical parameters: k = 3556, n = 4084.
- Encryption key = 1.5 Mbytes.

Attack by quantum search.
Can be defeated by increasing key size 4 times.

Summary

- Cryptosystems based on factoring and discrete logarithm are insecure against quantum computers;
- Alternatives:
  - Lattice-based crypto;
  - McEliece system;
  - Multivariate polynomials [Schulman, 2012].