Quantum Information Set Decoding Algorithms

Ghazal Kachigar Jean-Pierre Tillich

Institut de Mathématiques de Bordeaux, Université de Bordeaux

Inria, EPI SECRET

PQCrypto, Utrecht - 27/06/2017

Code-based Cryptography: good candidate for quantum-resistant cryptography

-H: full-rank $(n-k) \times n$ binary matrix
-$\mathcal{C} = \{c \in \mathbb{F}_2^n : Hc^T = 0\}$ code of length n and dimension $n-k$
-w: public parameter

Syndrome Decoding Problem (NP-hard)

Given $s = He^T$, find e of weight w.

Best classical **generic decoding** algorithms rely on the Information Set Decoding (ISD) technique.

Correcting an error of weight w in a code of length n and dimension k using an ISD algorithm has cost $\tilde{O}(2^{\alpha(\frac{k}{n},\frac{w}{n})n})$.

Author(s)	Year	$\max_{0 \leq R \leq 1} \alpha(R, \omega_{\mathrm{GV}})$
Prange	1962	0.1207
Dumer	1991	0.1164
May, Meurer and Thomae	2011	0.1114
Becker, Joux, May, Meurer	2012	0.1019
May, Ozerov	2015	0.0966

ω_{GV} : Gilbert-Varshamov bound

Question [Overbeck & Sendrier, 2009]

How much better can we do if we have access to quantum computers ?

One tool: **Grover's search algorithm**

Unstructured Search Problem

Given a set $\mathcal{E}$ and a function $f : \mathcal{E} \to \{0, 1\}$, find an $x \in \mathcal{E}$ such that $f(x) = 1$.

How many **queries** to f are needed to solve this problem?

ε : proportion of elements x of $\mathcal{E}$ such that $f(x) = 1$
T_f : average execution time of f

- Grover's search algorithm make $O(\frac{1}{\sqrt{\varepsilon}})$ queries and this is **optimal**.
- **Time complexity** of Grover Search: $O(\frac{T_f}{\sqrt{\varepsilon}})$

Prange's Algorithm (1962) and Bernstein's Algorithm (2009) (1/2)

Recall: Syndrome Decoding Problem

Given $s = He^T$ where H is a full-rank $(n-k) \times n$ binary matrix, find e of Hamming weight w.

Main idea: if the w errors are among $n-k$ known positions, problem reduces to solving a linear system in $n-k$ variables.

Prange's algorithm

(1) loop over possible sets $\mathscr{S}$ of size $n-k$
(2) solve linear system for each $\mathscr{S}$ to get an error vector
(3) check if its Hamming weight is w

Proportion p of good sets $\mathscr{S}$: $\Omega\left(\frac{\binom{n-k}{w}}{\binom{n}{w}}\right)$.

Bernstein's algorithm: use Grover Search to find a good set $\mathscr{S}$.

Prange's Algorithm (1962) and Bernstein's Algorithm (2009) (2/2)

Complexity of Prange's algorithm

Cost of (1): $\frac{1}{p} = O\left(\frac{\binom{n}{w}}{\binom{n-k}{w}}\right)$

Cost of (2) and (3): polynomial in n

Total cost:

$$\tilde{O}\left(2^{\alpha_{\text{Prange}}(R,\omega)n}\right)$$

where $R \triangleq \frac{k}{n}$

$\omega \triangleq \frac{w}{n}$

$$\alpha_{\text{Prange}}(R,\omega) = H_2(\omega) - (1-R)H_2\left(\frac{\omega}{1-R}\right)$$

Complexity of Bernstein's algorithm

Cost of (1) becomes $\frac{1}{\sqrt{p}}$

Thus $\alpha_{\text{Bernstein}} = \frac{\alpha_{\text{Prange}}}{2}$

Our results

Question [Overbeck & Sendrier, 2009]

How much better can we do if we have access to quantum computers ?

Author(s)	$\max_{0 \leq R \leq 1} \alpha(R, \omega_{\mathrm{GV}})$
Prange (1962)	0.1207
Bernstein (2009)	0.06035
Our first algorithm (SSQW)	**0.05970**
Our second algorithm (MMTQW)	**0.05869**

ω_{GV} : Gilbert-Varshamov bound

New tool: **Quantum Walk algorithms**

Graph Search Problem

Given a graph $G = (\mathcal{V}, \mathcal{E})$ and a set of vertices $\mathcal{M} \subset \mathcal{V}$, called the set of *marked elements*, find an $x \in \mathcal{M}$.

- Grover Search: graph search on K_n with $1_{\mathcal{M}} = f$.
- Useful point of view for problems with slightly more structure (less edges).
- Can be solved using a Random Walk (discrete-time Markov chain).

Algorithm 1: *RandomWalk*

Input: $G = (\mathcal{E}, \mathcal{V})$, $\mathcal{M} \subset \mathcal{V}$, initial probability distribution v
Output: An element $e \in \mathcal{M}$
SETUP : Sample a vertex x according to v and initialise the data structure.
repeat
  CHECK : **if** *current vertex x is marked* **then**
    return x
else
    repeat
      UPDATE : *Take one step of the random walk and update data structure accordingly.*
    until *x is sampled according to a distribution close enough to the uniform distribution*

T_s: cost of Setup
T_c: cost of Check
T_u: cost of Update
ε: $\frac{|\mathcal{M}|}{|\mathcal{V}|}$ (proportion of marked elements)
δ: spectral gap (a parameter of the graph)

Cost of Quantum Walk [Magniez, Nayak, Roland & Santha 2007]

$$T_s + \frac{1}{\sqrt{\varepsilon}}\left(T_c + \frac{1}{\sqrt{\delta}}T_u\right)$$

Recall: Prange's algorithm looked for sets $\mathscr{S}$ of size $(n-k)$ where all error positions would be.

Idea: Take $\mathscr{S}$ to be of size $n-k-\ell$ and allow p of the w errors to be outside $\mathscr{S}$
There are $P_{\ell,p} \triangleq \frac{\binom{k+\ell}{p}\binom{n-k-\ell}{w-p}}{\binom{n}{w}}$ such sets.
There exists U such that

$$UH = \begin{pmatrix} H' & 0_\ell \\ H" & I_{n-k-\ell} \end{pmatrix}$$

To find e, solve a new Syndrome Decoding Problem $s' = H'e'^T$ where e' is of weight p (cost T).

Cost of Generalised Quantum ISD Algorithms

$$O\left(\frac{T}{\sqrt{P_{\ell,p}}}\right)$$

k-sum Problem

$\mathcal{G}$: an Abelian group, $\mathcal{E}$: an arbitrary set, $f : \mathcal{E} \to \mathcal{G}$
k subsets $\mathcal{V}_0, \mathcal{V}_1, \ldots, \mathcal{V}_{k-1}$ of $\mathcal{E}$, $g : \mathcal{E}^k \to \{0,1\}$, S an element of $\mathcal{G}$

Find a solution $(v_0, \ldots, v_{k-1}) \in \mathcal{V}_0 \times \cdots \times \mathcal{V}_{k-1}$ such that

(i) $f(v_0) + f(v_1) \cdots + f(v_{k-1}) = S$ (subset-sum condition);
(ii) $g(v_0, \ldots, v_{k-1}) = 0$

Dumer's algorithm

$$\begin{aligned}
\mathcal{G} &= \mathbb{F}_2^{\ell}, \quad \mathcal{E} = \mathbb{F}_2^{k+\ell}, \quad f(v) = H' v^T \\
\mathcal{V}_0 &= \{(e_0, 0_{(k+\ell)/2}) \in \mathbb{F}_2^{k+\ell} : e_0 \in \mathbb{F}_2^{(k+\ell)/2}, \ |e_0| = p/2\} \\
\mathcal{V}_1 &= \{(0_{(k+\ell)/2}, e_1) \in \mathbb{F}_2^{k+\ell} : e_1 \in \mathbb{F}_2^{(k+\ell)/2}, \ |e_1| = p/2\}
\end{aligned}$$

$g(v_0, v_1) = 0$ if and only if the e resulting from $e' = v_0 + v_1$ is of weight w.

Dumer's algorithm

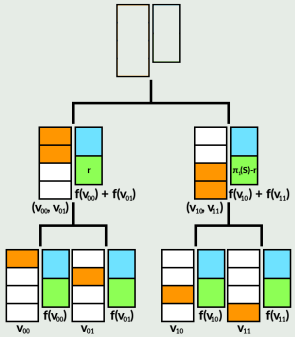
$$\begin{aligned} \mathcal{G} &= \mathbb{F}_2^{\ell}, \ \ \mathcal{E} = \mathbb{F}_2^{k+\ell}, \ \ f(v) = H'v^T \\ \mathcal{V}_0 &= \{(e_0, 0_{(k+\ell)/2}) \in \mathbb{F}_2^{k+\ell} : e_0 \in \mathbb{F}_2^{(k+\ell)/2}, \ |e_0| = p/2\} \\ \mathcal{V}_1 &= \{(0_{(k+\ell)/2}, e_1) \in \mathbb{F}_2^{k+\ell} : e_1 \in \mathbb{F}_2^{(k+\ell)/2}, \ |e_1| = p/2\} \end{aligned}$$

$g(v_0, v_1) = 0$ if and only if the e resulting from $e' = v_0 + v_1$ is of weight w.

Dumer's algorithm solves the 2-sum problem using collision search in expected time $|\mathcal{V}_0| + |\mathcal{V}_1| + \frac{|\mathcal{V}_0| \cdot |\mathcal{V}_1|}{|\mathcal{G}|}$.

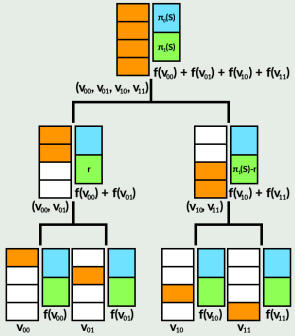
Suppose $\mathcal{G} = \mathcal{G}_0 \times \mathcal{G}_1$ where $|\mathcal{G}_0| = \Theta(|\mathcal{G}_1|) = \Theta(|\mathcal{G}|^{1/2})$, and let $\pi_i : g = (g_0, g_1) \mapsto g_i$.

Shamir-Schroeppel Algorithm



Suppose $\mathcal{G} = \mathcal{G}_0 \times \mathcal{G}_1$ where $|\mathcal{G}_0| = \Theta(|\mathcal{G}_1|) = \Theta(|\mathcal{G}|^{1/2})$, and let $\pi_i : g = (g_0, g_1) \mapsto g_i$.

Shamir-Schroeppel Algorithm



Need to do this for every $r \in \mathcal{G}_1$.

[Bernstein, Jeffery, Lange & Meurer 2013] : Quantum Shamir-Schroeppel algorithm for the subset sum problem

- First idea: use Grover Search to find r in time $O\left(\sqrt{|\mathcal{G}_1|}\right)$.
- Second idea: use a Quantum Walk algorithm to look for e.

Johnson graphs

$J(V, U)$

- Nodes: subsets $\mathcal{U}$ of size U of a set $\mathcal{V}$ of size V
- Edges: $(\mathcal{U}, \mathcal{U}')$ is an edge iff $|\mathcal{U} \cap \mathcal{U}'| = U - 1$
- Spectral gap: $\delta = \frac{V}{U(V-R)} = \Omega\left(\frac{1}{U}\right)$

Quantum walk on $J(V,U) \times J(V,U) \times J(V,U) \times J(V,U)$ where $V = |\mathcal{V}_{ij}|$.
Cost: $T_s + \frac{1}{\sqrt{\varepsilon}} \left(T_c + \frac{1}{\sqrt{\delta}} T_u \right)$

Cost of the quantum walk

- δ: $\Omega\left(\frac{1}{U}\right)$.
- ε: $\left(\frac{U}{V}\right)^4$.
- Setup time T_s: $O(U)$.
- Check time T_c: $O(1)$.
- **Update time T_u:** $O(\log U)$ under the hypotheses $|\mathcal{G}_1| = \Theta(U), |\mathcal{G}| = \Theta(U^2)$

Cost : $O\left(U + \left(\frac{V}{U}\right)^2 \left(1 + \sqrt{U} \log U\right)\right)$
This is optimal and equal to $\tilde{O}(U)$ for $U = V^{4/5}$.

Cost of the algorithm

$$O\left(\frac{\sqrt{\mathcal{G}_1}V^{4/5}}{\sqrt{P_{\ell,p}}}\right) \text{ with } |\mathcal{G}_1| = \Theta(V^{4/5})$$

$$\alpha_{\text{SSQW}}(R,\omega) \stackrel{\triangle}{=} \min_{(\pi,\lambda)\in\mathcal{R}} \left(\frac{H_2(\omega) - (1-R-\lambda)H_2\left(\frac{\omega-\pi}{1-R-\lambda}\right) - \frac{2}{5}(R+\lambda)H_2\left(\frac{\pi}{R+\lambda}\right)}{2} \right)$$

$$\mathcal{R} \stackrel{\triangle}{=} \left\{ (\pi,\lambda)\in[0,\omega]\times[0,1) : \lambda = \frac{2}{5}(R+\lambda)H_2\left(\frac{\pi}{R+\lambda}\right), \pi \leq R+\lambda, \lambda \leq 1-R-\omega+\pi \right\}$$

Representation technique

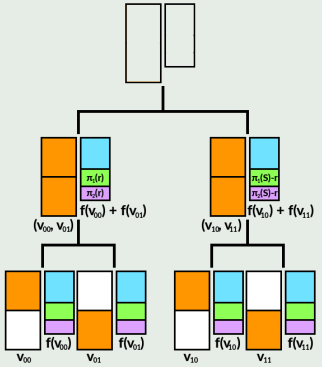
$\binom{p}{p/2}$ possible representations

MMT 4-sum problem

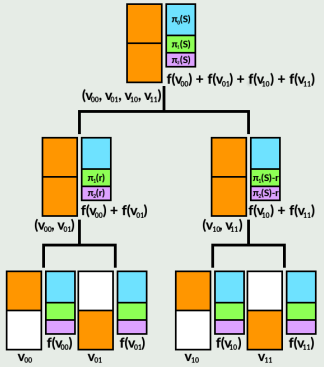
$$\mathcal{V}_{00} = \mathcal{V}_{10} \stackrel{\triangle}{=} \{(e_{00}, 0_{(k+\ell)/2}) \in \mathbb{F}_2^{k+\ell} : e_{00} \in \mathbb{F}_2^{(k+\ell)/2},\ |e_{00}| = \tfrac{p}{4} + \tfrac{\Delta p}{2}\}$$
$$\mathcal{V}_{01} = \mathcal{V}_{11} \stackrel{\triangle}{=} \{(0_{(k+\ell)/2}, e_{01}) \in \mathbb{F}_2^{k+\ell} : e_{01} \in \mathbb{F}_2^{(k+\ell)/2},\ |e_{01}| = \tfrac{p}{4} + \tfrac{\Delta p}{2}\}$$

The $\mathcal{V}_{ij}$ are bigger.
But we can now write $\mathcal{G} = \mathcal{G}_0 \times \mathcal{G}_1 \times \mathcal{G}_2$.

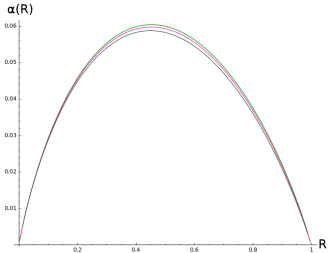
Quantum May-Meurer-Thomae Algorithm



Quantum May-Meurer-Thomae Algorithm



Need $|\mathcal{G}_1| \cdot |\mathcal{G}_2| = \Omega\left(V^{4/5}\right)$ and $|\mathcal{G}| = \Omega\left(V^{8/5}\right)$ **but** only need to do this for every $\pi_1(r) \in \mathcal{G}_1$.



$\alpha_{\text{Bernstein}}$ in green, α_{SSQW} in pink, α_{MMTQW} in grey.

Author(s)	$\max_{0 \leq R \leq 1} \alpha(R, \omega_{\text{GV}})$
Prange (1962)	0.1207
Bernstein (2009)	0.06035
Our first algorithm (SSQW)	**0.05970**
Our second algorithm (MMTQW)	**0.05869**

ω_{GV} : Gilbert-Varshamov bound

- **Why has it been difficult to do better?**

  - Complexity is given by $O\left(\frac{T_{\text{algorithm}}}{\sqrt{P_{\ell,p}}}\right)$.

  - Space complexity seems to be a lower bound on the time complexity of the algorithm.
- **What about BJMM's algorithm?** Has worse complexity (uses more space than MMT).
- **What about May and Ozerov's algorithm?** Open question, but it has high space complexity.

Thank you for your attention