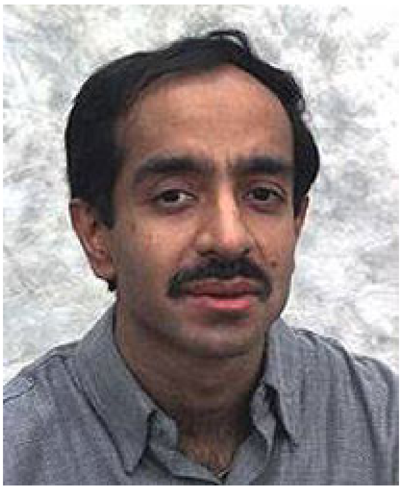 

Grover vs. McEliece

D. J. Bernstein
University of Illinois at Chicago

Thanks to:
NSF ITR–0716498
and
Cisco University Research Program

Conventional wisdom:
Grover's algorithm
forces 2× key size.

e.g. Want security 2^{128}?
128-bit AES key seems safe
before quantum computers;
but need 256-bit AES key
to resist Grover's algorithm.

e.g. Want security 2^{256}?
256-bit AES key seems safe
(despite "related-key" silliness)
before quantum computers;
but need new 512-bit cipher
to resist Grover's algorithm.

Breaking a good b-bit cipher
takes 2^b bit operations
but $2^{b/2}$ qubit operations.

Breaking a good b-bit cipher takes 2^b bit operations but $2^{b/2}$ qubit operations.

Correction: Have to multiply 2^b by cost of cipher evaluation; and have to multiply $2^{b/2}$ by more, namely cost of *quantum* cipher evaluation.

Plausible scaling hypotheses $\Rightarrow$ Correction changes comparison by various constant factors and logarithmic factors.
Key-size ratio < 2;
but ratio $\to 2$ as $b \to \infty$.

Many problems analogous to finding b-bit cipher key.

For 2^b security of finding a hash preimage:
Before quantum computers, need a good $(1+o(1))b$-bit hash.
After quantum computers, need a good $(2+o(1))b$-bit hash.

Ratio: $\dfrac{(2+o(1))b}{(1+o(1))b} = 2+o(1)$.

Many problems analogous to finding b-bit cipher key.

For 2^b security of finding a hash preimage:
Before quantum computers, need a good $(1 + o(1))b$-bit hash.
After quantum computers, need a good $(2 + o(1))b$-bit hash.

Ratio: $$\frac{(2 + o(1))b}{(1 + o(1))b} = 2 + o(1).$$

But there are many problems where the conventional wisdom seems to be wrong!
Ratio $c + o(1)$ with $c < 2$.

For 2^b security of
finding a hash collision:

Before quantum computers,
need a good $(2+o(1))b$-bit hash.

Common belief, based on
1998 Brassard–Høyer–Tapp:
After quantum computers,
need a good $(3+o(1))b$-bit hash.
Size ratio $1.5+o(1)$.

For 2^b security of
finding a hash collision:

Before quantum computers,
need a good $(2 + o(1))b$-bit hash.

Common belief, based on
1998 Brassard–Høyer–Tapp:
After quantum computers,
need a good $(3 + o(1))b$-bit hash.
Size ratio $1.5 + o(1)$.

2009 Bernstein: Actually,
a good $(2 + o(1))b$-bit hash
stops all known attacks,
including Brassard–Høyer–Tapp.
Size ratio $1 + o(1)$.

Size ratio $1 + o(1)$
can often be *proven*:
e.g., Grover's algorithm
obviously has no effect
on the key size needed for the
1974 Gilbert–MacWilliams–Sloane
authentication system.

Can also find cases where
$1 + o(1)$ is *conjectured*.

2009 Overbeck–Sendrier:
"Grover's algorithm is not able
[to] give a significant speed-up
for the existing attacks" against
the McEliece cryptosystem.

Information-set decoding

McEliece public key:
linear map $G : \mathbf{F}_2^k \hookrightarrow \mathbf{F}_2^n$.

McEliece plaintext:
$m \in \mathbf{F}_2^k$; and $e \in \mathbf{F}_2^n$ of weight t.

McEliece ciphertext:
$Gm + e \in \mathbf{F}_2^n$.

Typical parameter choices:
$k = Rn$ with $R = 0.8$;
$t = (n - k)/\lceil \lg n \rceil$
$\approx (1 - R)n/\lg n$.

Basic information-set decoding, given G and $y \in \mathbf{F}_2^n$:

Choose uniform random size-k subset $S \subseteq \{1, 2, \ldots, n\}$.

Hope that the composition $\mathbf{F}_2^k \xrightarrow{G} \mathbf{F}_2^n \to \mathbf{F}_2^S$ is invertible (S is an "information set").
If not invertible, try new S.

Project y from $\mathbf{F}_2^n$ to $\mathbf{F}_2^S$.
Apply inverse, obtaining m.
Compute $e = y - Gm$.
If weight of e is not t, try new S.

For typical G and $y = Gm + e$:
Pr[S finds m and e]
$\approx 0.29\binom{n-t}{k}/\binom{n}{k}$
$\in 1/c^{(1+o(1))n/\lg n}$.

Here $c = 1/(1-R)^{1-R} \approx 1.38$;
$o(1) \to 0$ as $n \to \infty$.

Total time $c^{(1+o(1))n/\lg n}$.

Advanced information-set decoding has many speedups. 2009 Bernstein–Lange–Peters–van Tilborg: these save $n^{>\text{const}}$, but still total time $c^{(1+o(1))n/\lg n}$.

Previous Grover decoding

1998 Barg–Zhou: Grover's algorithm can decode any length-n code C, linear or not, "on a quantum computer of circuit size $O(n|C|^{1/2})$ in time $O(n|C|^{1/2})$, which is essentially optimal following a result in [1997 Bennett et al.]."

Previous Grover decoding

1998 Barg–Zhou: Grover's algorithm can decode any length-n code C, linear or not, "on a quantum computer of circuit size $O(n|C|^{1/2})$ in time $O(n|C|^{1/2})$, which is essentially optimal following a result in [1997 Bennett et al.]."

Much slower than information-set decoding.

Previous Grover decoding

1998 Barg–Zhou: Grover's algorithm can decode any length-n code C, linear or not, "on a quantum computer of circuit size $O(n|C|^{1/2})$ in time $O(n|C|^{1/2})$, which is essentially optimal following a result in [1997 Bennett et al.]."

Much slower than information-set decoding.

2009 Overbeck–Sendrier: Begin with "the simplifying assumption that by Grover's algorithm we are able to search a set of size N in $O(\sqrt{N})$ operations on a quantum computer with at least $\log_2(N)$ QuBits."

Cannot search for sets S: "this would either require an iterative application of Grover's algorithm (which is not possible) or a memory of size of the whole search space, as the search function in the second step depends on the first step.
This would clearly ruin the 'divide-and-conquer' strategy and is thus not possible either."

Grover's root-finding method

1996 Grover "A fast quantum mechanical algorithm for database search" is not actually a database-search algorithm.

Input to Grover's transformation: circuit that computes a function $f : \mathbf{F}_2^b \to \mathbf{F}_2$.

Output: quantum circuit that (if possible) computes $x \in \mathbf{F}_2^b$ such that $f(x) = 0$.

The transformation is explicit and efficient.

Simplest version—adequate
when f has small, fast circuit:

circuit for f
$\Rightarrow$ combinatorial circuit for f
$\Rightarrow$ reversible circuit for f
$\Rightarrow$ quantum circuit for f
$\Rightarrow$ quantum circuit for f
plus quantum rotation etc.
$\Rightarrow$ root-finding quantum circuit.

Root-finding circuit is small
and uses $\approx \sqrt{2^b/r}$ fast iterations
if f has r roots.

(1996 Grover for $r = 1$; 1996
Boyer–Brassard–Høyer–Tapp)

Quantum information-set decoding

Choose big b and $\mathbf{F}_2^b \twoheadrightarrow \{S\}$,
close to uniformly distributed.

Define $f : \mathbf{F}_2^b \to \mathbf{F}_2$ as follows:

Compute corresponding S.
Return 1 if the composition
$\mathbf{F}_2^k \xrightarrow{G} \mathbf{F}_2^n \to \mathbf{F}_2^S$ is not invertible.
Project y from $\mathbf{F}_2^n$ to $\mathbf{F}_2^S$.
Apply inverse, obtaining m.
Compute $e = y - Gm$.
Return 1 if weight is not t.
Return 0.

Compute this function f
using a combinatorial circuit
containing $n^{O(1)}$ bit operations.

Basic information-set decoding
searches randomly for a root of f.
$c^{(1+o(1))n/\lg n}$ evaluations of f,
each taking time $n^{O(1)}$.

Basic quantum information-set
decoding: Apply Grover.
Root-finding circuit
uses $c^{(1/2+o(1))n/\lg n}$
quantum evaluations of f,
each taking time $n^{O(1)}$;
and has size $n^{O(1)}$.

Consequence for McEliece users:

Before quantum computers, need $n \in (1 + o(1))(b/\lg c)\lg b$ for security 2^b. Key size

$$\left(\frac{R(1-R)}{(\lg c)^2} + o(1)\right) b^2 (\lg b)^2.$$

After quantum computers, need $n \in (2 + o(1))(b/\lg c)\lg b$ for security 2^b. Key size

$$\left(\frac{4R(1-R)}{(\lg c)^2} + o(1)\right) b^2 (\lg b)^2.$$

Ratio $4 + o(1)$.