Code Equivalence is Hard for Shor-like Quantum Algorithms

Hang Dinh

Indiana University South Bend

Workshop on Code-Based Cryptography
(CBC2012)

Code Equivalence (CE)

- The CE Problem:
  - Given two linear codes C and C’
  - Decide if C is equivalent to C’ up to a permutation of the codeword coordinates
- Petrank and Roth, 1997 proved
  - Code Equivalence is unlikely NP-complete,
  - but is at least as hard as Graph Isomorphism
    - There’s an efficient reduction from Graph Isomorphism to CE

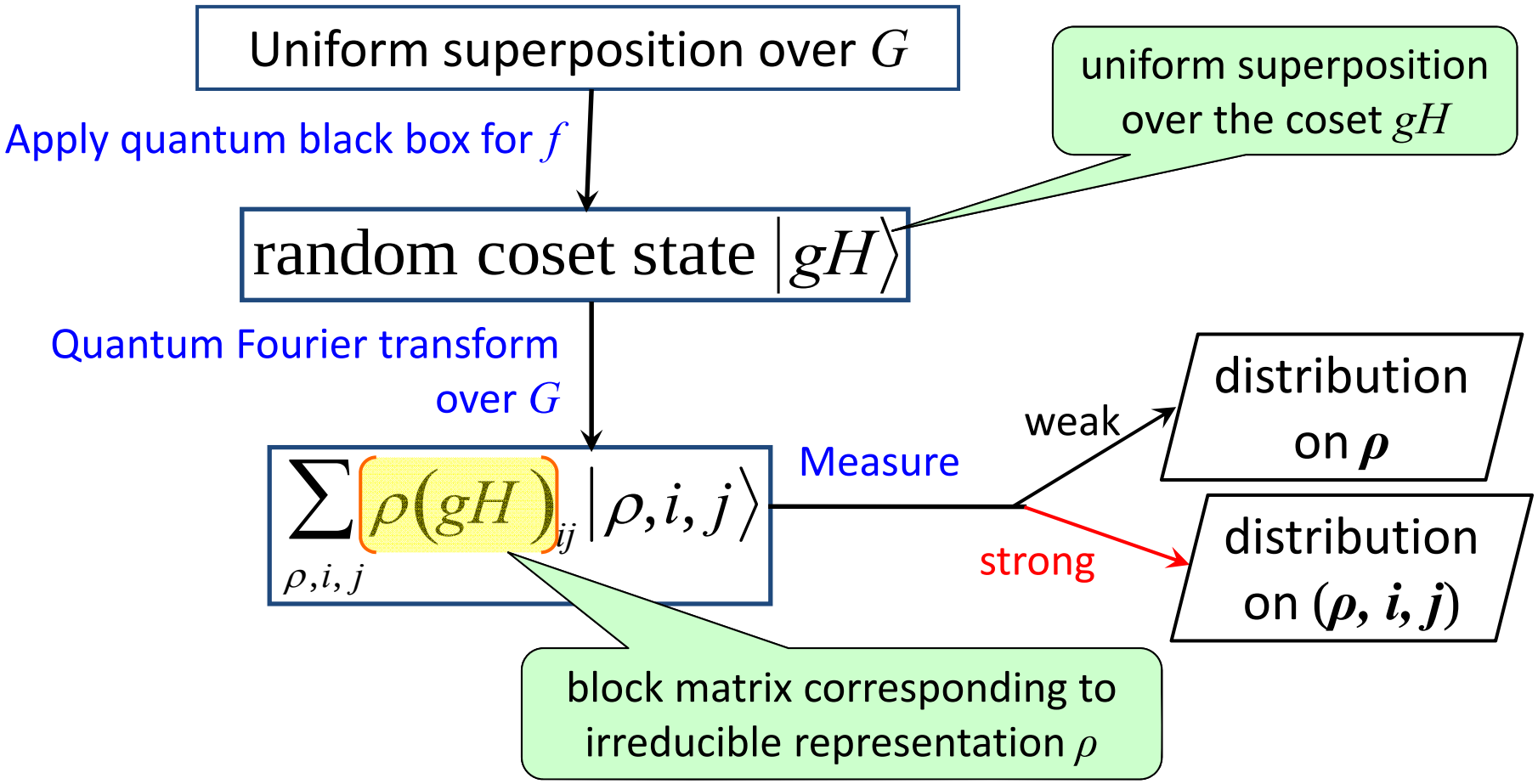
Code Equivalence (CE)

- A search version of CE:
  - Given two permutation-equivalent linear codes C and C'
  - Find a permutation between C and C'
- Related to security of McEliece-type cryptosystems
  - In the case where the secret code is known
- Support Splitting Algorithm [Sendrier 1999]
  - Efficient for codes with small hull dimension, including Goppa codes and many binary codes
  - **Inefficient** for other codes, such as Reed-Muller codes.

Hidden Subgroup Problem (HSP)

- HSP is a generalization of problems possibly solved by Shor-like quantum algorithms.
- HSP over a finite group G:
  - Input: a black-box function f on G that *distinguishes* the left cosets of an unknown subgroup $H < G$, i.e.,
    $$f(x) = f(y) \Leftrightarrow xH = yH$$
  - Output: a generating set for H.
- There is a natural reduction from CE to HSP
  - where the group G is non-abelian (a *rich* wreath product)
  - So, can CE be solved efficiently by Shor-like algorithms?

Quantum Fourier Sampling (QFS)- Quantum part of Shor-like algorithms



Efficiency of Shor-like Algorithms

- Shor's quantum algorithms efficiently solve
  - HSP over cyclic groups Z_N → factorization
  - HSP over $Z_N \times Z_N$ → discrete logarithm
- Quantum Fourier Sampling
  - Efficient for HSP over abelian groups
  - There are efficient quantum Fourier transforms for certain non-abelian cases [See Lomont 2004 for a survey].
  - But inefficient (or not known to be efficient) for interesting non-abelian cases, including symmetric and dihedral groups.

Our Results

- We show that in many cases of interest,
  - Solving the case of HSP reduced from CE by QFS requires rich, entangled measurements.
- Our results apply to many codes, including
  - Classical Goppa codes, rational Goppa codes
    [Dinh, Moore, Russell, CRYPTO 2011]
  - Reed-Muller codes (used in the Sidelnikov cryptosystem)
    [Dinh, Moore, Russell, Preprint 2011 , arXiv:1111.4382]

→Shor-like algorithms are unlikely to help break code-based cryptosystems in these cases.

HSP-hard Codes

- What codes make CE hard for Shor-like algorithms?
  - A linear code C is called *HSP-hard* if strong QFS reveals negligible information about the permutation between C and any code equivalent to C.
- Theorem[Dinh, Moore, Russell, CRYPTO 2011]: Let C be a q-ary $[n, k]$-code s.t. $k^2 \leq 0.2n \log_q n$. Then C is HSP-hard if
  1) The automorphism group $Aut(C)$ has size $\leq e^{o(n)}$
  2) The *minimal degree* of $Aut(C)$ is $\geq \Omega(n)$.

the minimal number of coordinates moved by a non-identity permutation in $Aut(C)$

Reed-Muller Codes are HSP-hard

- Binary Reed-Muller code $\mathrm{RM}(r, m)$
  - has length $n = 2^m$ and dimension $k = \sum_{j=0}^{r} \binom{m}{j}$.
  - If $r < 0.1m$, then $k^2 \leq 0.2nm$ for sufficiently large m.
- If C is a binary Reed-Muller code of length $n = 2^m$, then
  1. $|Aut(C)| \leq 2^{m^2+m} \leq 2^{O(\log^2 n)} \leq e^{o(n)}$
  2. The minimal degree of $Aut(C)$ is exactly $2^{m-1} = n/2$.

  Proof: Use the fact that

  $Aut(C)$ = general affine group of space $\mathbf{F}_2{}^m$

Open Question

- Are there other HSP-hard codes that are of cryptographic interest?