Code Equivalence is Hard for Shor-like Quantum Algorithms

Hang Dinh – Indiana University South Bend

joint work with

Cristopher Moore – University of New Mexico
Alexander Russell – University of Connecticut

Webinar on Symbolic Computations and Post-Quantum Cryptography
(SCPQ Webinar May 17, 2012)
Outline

• Overview/Motivation
  – Code Equivalence
  – Why care?
• Shor-like algorithms
  – Quantum Fourier Sampling (QFS)
  – Hidden Subgroup Problems (HSP)
• Reduction from Code Equivalence to HSP
• Our results
  – General results
  – Codes that make Code Equivalence hard for QFS
Code Equivalence (CE)

- **Code Equivalence** [Petrank and Roth, 1997]
  - Given the generator matrices of two linear codes $C$ and $C'$
  - Decide if $C$ is equivalent to $C'$ up to a permutation of the codeword coordinates

- A search variant of CE:
  - Find a permutation between two given equivalent codes

- **Hardness** [Petrank and Roth, 1997]
  - Code Equivalence is unlikely NP-complete,
  - but at least as hard as Graph Isomorphism
    - There’s an efficient reduction from Graph Isomorphism to CE
CE and Code-based Cryptosystems

<table>
<thead>
<tr>
<th></th>
<th>McEliece systems</th>
<th>Neiderreiter systems</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Secret code C</strong></td>
<td>$q$-ary $[n, k]$-code</td>
<td>$q$-ary $[n, n - lk]$-code</td>
</tr>
<tr>
<td><strong>Secret key</strong></td>
<td>$M: k \times n$ generator matrix of $C$</td>
<td>$M: k \times n$ parity check matrix over $\mathbb{F}_{q^l}$ of $C$</td>
</tr>
<tr>
<td></td>
<td>$S: k \times k$ invertible matrix over $\mathbb{F}_q$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$P: n \times n$ permutation matrix</td>
<td></td>
</tr>
<tr>
<td><strong>Public key</strong></td>
<td></td>
<td>$M' = SMP$</td>
</tr>
</tbody>
</table>

- If the secret code is known to the adversary
  - recover secret key $S$ and $P \rightarrow$ solve CE for the secret code
CE and Code-based Cryptosystems

• The secret code can be known to the adversary
  – if the space of all codes of the same parameters \((q, n, k)\) and same family as the secret code is small.

• **Example**: Reed-Muller codes \((q=2)\)
  – used in the Sildelnikov cryptosystem [Sidelnikov, 1994]
  – there’s a *single* Reed-Muller code of given length and dimension.

• **Example**: *special* binary Goppa codes
  – those generated by polynomials of binary coefficients
  – can exhaustively search [Loidreau and Sendrier, 2001]
Best Known Algorithm for CE

• Support Splitting Algorithm [Sendrier, 1999]
  – Classical, deterministic
  – Efficient for **binary** codes with small hull dimension, including binary Goppa codes.
  – *Likely* to be efficient for non-binary codes with small hull dimension
  – **Inefficient** for other codes, such as Reed-Muller codes.
Can Quantum Algorithms Do Better?

• The most popular paradigm of quantum algorithms
  – generalize Shor’s algorithms
  – reply on quantum Fourier transform
  – solve the class of hidden subgroup problems (HSP).
  – Nearly all known quantum algorithms that provide exponential speedup are designed in this paradigm.

• There’s a natural reduction from CE to HSP
  – So, can CE be solved efficiently by Shor-like algorithms?
Outline

• Overview/Motivation
  – Code Equivalence
  – Why care?

• Shor-like algorithms
  – Quantum Fourier Sampling (QFS)
  – Hidden Subgroup Problems (HSP)

• Reduction from Code Equivalence to HSP

• Our results
  – General results
  – Codes that make Code Equivalence hard for QFS
Hidden Subgroup Problem (HSP)

• HSP over a finite group $G$:
  – **Input**: a black-box function $f$ on $G$ that *separates* the left (or right) cosets of an unknown subgroup $H < G$, i.e.,
    $$f(x) = f(y) \text{ iff } xH = yH$$
  – **Output**: a generating set for $H$.

• Well-known interesting cases
  – HSP over cyclic groups $\mathbb{Z}_N$ \(\rightarrow\) factorization
  – HSP over $\mathbb{Z}_N \times \mathbb{Z}_N$ \(\rightarrow\) discrete logarithm
  – HSP over symmetric groups $S_n$ \(\rightarrow\) Graph Isomorphism
  – HSP over dihedral groups $D_n$ \(\rightarrow\) unique-Shortest-vector
Shor-like Algorithms

• To solve the HSP over $G$ with hidden subgroup $H$

Quantum Fourier Sampling (QFS) over $G$ using back box $f$ that separates cosets of $H$

a probability distribution, denoted $\text{QFS}_G(H)$

Classically recover $H$ using information from the distribution $\text{QFS}_G(H)$
Quantum Fourier Sampling (QFS)

1. Uniform superposition over $G$
2. Random coset state $gH$
3. Apply quantum black box for $f$
4. Quantum Fourier transform over $G$
5. Measure distribution on $\rho$
6. Block matrix corresponding to irreducible representation $\rho$
7. Uniform superposition over the coset $gH$
8. Weak distribution on $(\rho, j)$
9. Strong distribution on $(\rho, j)$

$QFS_G(H)$
Efficiency of Shor-like Algorithms

• QFS is efficient for HSP over abelian groups.

• Some nonabelian HSPs *may* be efficiently solvable
  – They have efficient quantum Fourier transforms.
  – Subexponential time for dihedral HSP [Kuperberg, 2003]

• Strong QFS doesn’t work for $S_n$ if $|H| = 2$
  – it can’t distinguish among conjugates of $H$ and the trivial one
  – i.e., $\text{QFS}_G(gHg^{-1})$ is close to $\text{QFS}_G(\{1\})$, for most $g \in G$.
  – [Moore, Russell, Schulman, 2008].
Outline

• Overview/Motivation
  – Code Equivalence
  – Why care?

• Shor-like algorithms
  – Quantum Fourier Sampling (QFS)
  – Hidden Subgroup Problems (HSP)

• Reduction from Code Equivalence to HSP

• Our results
  – General results
  – Codes that make Code Equivalence hard for QFS
Reduce CE to HSP

- Search variant of Code Equivalence
- Scrambler-Permutation Problem
- Hidden Shift Problem
- Hidden Subgroup Problem
CE to Scrambler-Permutation

**Scrambler-Permutation Problem**

- **Input:** $k \times n$ matrices $M$ and $M'$ over a field $F_{q^l} \supseteq F_q$ such that $M' = SMP$ for some $(S, P) \in \text{GL}_k(F_q) \times S_n$
- **Output:** $(S, P)$

**Special case:** attacking McEliece systems

- $l = 1$ ($F_{q^l} = F_q$)
- $M$ is a generator matrix of a $q$-ary $[n, k]$-code.

**Special case:** attacking Neiderreiter systems

- $M$ is parity check matrix of a $q$-ary $[n, n - lk]$-code.
Scrambler-Permutation to Hidden Shift

• **Hidden Shift Problem** over a finite group $G$:
  - **Input**: two functions $f_1, f_2$ on $G$ s.t. $\exists s \in G$ satisfying
    $$f_1(sg) = f_2(g) \text{ for all } g \in G$$
  - **Output**: a hidden shift $s$

Input: $M$ and $M' = SMP$. Output: $(S, P) \in \text{GL}_k(F_q) \times S_n$

Hidden Shift Problem over $\text{GL}_k(F_q) \times S_n$
  - **Input**: $f_1(X, Y) = X^{-1}MY$ and $f_2(X, Y) = X^{-1}M'Y$
  - **Output**: a hidden shift $(S^{-1}, P)$
Hidden Shift to Hidden Subgroup

**Hidden Shift Problem** over a finite group $G$:

- **Input**: two functions $f_1, f_2$ on $G$ s.t. $\exists s \in G$ satisfying
  \[ f_1(sg) = f_2(g) \text{ for all } g \in G \]

- **Output**: a hidden shift $s$

**HSP** over wreath product $G \wr \mathbb{Z}_2$ (semidirect product of $G^2$ and $\mathbb{Z}_2$)

- **Input**: function $f$ defined as:
  \[ f((g_1, g_2), 0) = (f_1(g_1), f_2(g_2)) \]
  \[ f((g_1, g_2), 1) = (f_2(g_2), f_1(g_1)) \]
Hidden Shift to Hidden Subgroup

**Hidden Shift Problem** over a finite group $G$:

- **Input**: two functions $f_1, f_2$ on $G$ s.t. $\exists s \in G$ satisfying
  $$f_1(sg) = f_2(g) \text{ for all } g \in G$$
- **Output**: a hidden shift $s$

**HSP** over wreath product $G \wr \mathbb{Z}_2$ (semidirect product of $G^2$ and $\mathbb{Z}_2$)

- **Output**: subgroup $H = ((H_0, s^{-1}H_0s), 0) \cup ((H_0s, s^{-1}H_0), 1)$
  where
  $$H_0 = \{ g \in G \mid f_1(g) = f_1(1) \} < G$$
  $$H_0s = \text{The set of all hidden shifts}$$

\[ f_1 \text{ must separate right cosets of } H_0 \]
Scrambler-Permutation to HSP

**Scrambler-Permutation Problem**

- **Input:** \( M \) and \( M' = SMP \) for some \((S, P) \in \text{GL}_k(\mathbb{F}_q) \times S_n\)
- **Output:** \((S, P)\)

**HSP over the wreath product** \((\text{GL}_k(\mathbb{F}_q) \times S_n) \rtimes \mathbb{Z}_2\)

- **hidden subgroup:** \( H = ((H_0, s^{-1}H_0s), 0) \cup ((H_0s, s^{-1}H_0), 1) \)

where

\[
H_0 = \{(S, P) | S^{-1}MP = M\} < \text{GL}_k(\mathbb{F}_q) \times S_n
\]

\[
s = (S^{-1}, P)
\]

Can this HSP be solved efficiently by strong QFS?
Can QFS distinguish conjugates \( gHg^{-1} \) and \( \{1\} \)?

Hang Dinh - IU South Bend
Outline

• Overview/Motivation
  – Code Equivalence
  – Why care?
• Shor-like algorithms
  – Quantum Fourier Sampling (QFS)
  – Hidden Subgroup Problems (HSP)
• Reduction from Code Equivalence to HSP
• Our results
  – General results
  – Codes that make Code Equivalence hard for QFS
Our Results

• We show that in many cases of interest,
  – $QFS_G(gHg^{-1})$ is exponentially close to $QFS_G(\{1\})$, for most $g \in G$.
  – In such a case, $H$ is called *indistinguishable* by strong QFS.

• Apply to $G = S_n$ with $|H| \geq 2$

• Apply to the CE for many codes, including
  – Goppa codes, generalized Reed-Solomon codes
    [Dinh, Moore, Russell, CRYPTO 2011]
  – Reed-Muller codes
Hidden Symmetries

• Recall: the hidden subgroup reduced from matrix $M$ is determined by the subgroup

$$H_0 = \{(S, P) | S^{-1}MP = M\} < \text{GL}_k(F_q) \times S_n$$

• Projection of $H_0$ onto $S_n$ is the automorphism group

$$\text{Aut}(M) := \{P \in S_n | \exists S \in \text{GL}_k(F_q), SMP = M\}$$

  − Each $P \in \text{Aut}(M)$ has the same number $N$ of preimages $S \in \text{GL}_k(F_q)$ in this projection.

  − **Fact:** Let $r$ be the column rank of $M$. Then $N \leq q^{lk(k-r)}$.

  − Hence, $|H_0| \leq |\text{Aut}(M)| q^{lk(k-r)}$. 

Hang Dinh - Indiana University South Bend
General Results for CE

• **Theorem** [Dinh, Moore, Russell, CRYPTO 2011]:
  – Assume $k^2 \leq 0.2n \log_q n$.
  – The hidden subgroup reduced from matrix $M$ is indistinguishable by strong QFS if
    1) $|\text{Aut}(M)| \leq e^{o(n)}$
    2) The *minimal degree* of $\text{Aut}(M)$ is $\geq \Omega(n)$.
    3) The column rank of $M$ is $\geq k - o(\sqrt{n})/l$.

The *minimal degree* of $\text{Aut}(M)$ is the minimal number of points *moved* by a non-identity permutation in $\text{Aut}(M)$.
HSP-hard Codes

• What codes make CE hard for Shor-like algorithms?
  – A linear code is called **HSP-hard** if it has a generator matrix or parity check matrix $M$ s.t. the hidden subgroup reduced from $M$ is indistinguishable by strong QFS.

• Observe: If $M$ is a generator matrix of a code $C$
  – Then $\text{Aut}(M) = \text{Aut}(C)$, and $M$ has full rank.

• **Corollary:** Let $C$ be a $q$-ary $[n, k]$-code such that $k^2 \leq 0.2n \log_q n$. Then $C$ is HSP-hard if
  1) $|\text{Aut}(C)| \leq e^{o(n)}$
  2) The minimal degree of $\text{Aut}(C)$ is $\geq \Omega(n)$. 
Reed-Muller Codes are HSP-hard

- Reed-Muller code RM\((r, m)\)
  \[\{(f(\alpha_1), \ldots, f(\alpha_n)) \mid f \in \mathbb{F}_2[X_1, \ldots, X_m], \deg(f) \leq r\},\]
  where \((\alpha_1, \ldots, \alpha_n)\) is a fixed ordered list of all vectors in \(\mathbb{F}_2^m\)
  - has length \(n = 2^m\) and dimension \(k = \sum_{j=0}^{r} \binom{m}{j}\).
  - If \(r < 0.1m\), then \(k < r \binom{m}{0.1m} < r2^{0.47m}\), and \(k^2 \leq 0.2nm\) for sufficiently large \(m\).

- **Theorem**: Reed-Muller codes RM\((r, m)\) with \(r < 0.1m\) and \(m\) sufficiently large are HSP-hard.
Automorphism Group of Reed-Muller Codes

**Fact:**

\[ \text{Aut}(\text{RM}(r, m)) = \text{general affine group of space } \mathbb{F}_2^m \]

\[ = \{ \sigma_{A,b} : \mathbb{F}_2^m \rightarrow \mathbb{F}_2^m, \sigma_{A,b}(x) = Ax + b | A \in \text{GL}_m(\mathbb{F}_2), b \in \mathbb{F}_2^m \} \]

**Propositions:**

1. \[ |\text{Aut}(\text{RM}(r, m))| = |\text{GL}_m(\mathbb{F}_2)| \times |\mathbb{F}_2^m| \leq 2^{m^2+m} \]

\[ \leq 2^0(\log^2 n) \leq e^o(n) \], where \( n = 2^m \)

2. The minimal degree of Aut(RM(r, m)) is exactly \( 2^{m-1} \).
Automorphism Group of
Reed-Muller Codes

2a. The minimal degree of $\text{Aut}(\text{RM}(r, m))$ is $\leq 2^{m-1}$.

*Recall:* $\text{min \ deg. of } \text{Aut}(C) := \min \{ \text{supp}(\pi) | \pi \in \text{Aut}(C), \pi \neq \text{Id} \}$, where $\text{supp}(\pi) := \#\{i: \pi(i) \neq i\}$.

*Proof:*

- An affine transformation $\sigma_{A,0}: \mathbb{F}_2^m \rightarrow \mathbb{F}_2^m$ with support $2^{m-1}$

$$\sigma_{A,0}(x) = Ax = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ 1 & 1 & \ldots & 1 \end{pmatrix} x$$

- This $\sigma_{A,0}$ fixes all vectors $x \in \mathbb{F}_2^m$ with $x_m = 0$.

- There are $2^m - 2^{m-1} = 2^{m-1}$ vectors not fixed by $\sigma_{A,0}$
Automorphism Group of Reed-Muller Codes

2b. The minimal degree of $\text{Aut}(\text{RM}(r, m))$ is $\geq 2^{m-1}$.

- **Claim 1**: If $\sigma_{A,b}$ fixes a set $S$ that spans $\mathbb{F}_2^m$, then $\sigma_{A,b} = \text{Id}$.

- **Claim 2**: Any set $S \subseteq \mathbb{F}_2^m$ with size $> 2^{m-1}$ spans $\mathbb{F}_2^m$.

$\rightarrow$ No none-identity affine transformation can fix $> 2^{m-1}$ vectors.
Automorphism Group of Reed-Muller Codes

2b. The minimal degree of $Aut(RM(r, m))$ is $\geq 2^{m-1}$.

- **Claim 1**: If $\sigma_{A,b}$ fixes a set $S$ that spans $\mathbb{F}_2^m$, then $\sigma_{A,b} = Id$.

  *Proof*: Let $s \in S$ and $S' = S - s$. Then $S'$ also spans $\mathbb{F}_2^m$, and $A$ fixes $S'$, in which case $A = 1$. Then $b = 0$. Note $\sigma_{1,0} = Id$.

- **Claim 2**: Any set $S \subseteq \mathbb{F}_2^m$ with size $> 2^{m-1}$ spans $\mathbb{F}_2^m$.

$\rightarrow$ No none-identity affine transformation can fix $>2^{m-1}$ vectors.
Automorphism Group of Reed-Muller Codes

2b. The minimal degree of $Aut(RM(r, m))$ is $\geq 2^{m-1}$.

- Claim 1: If $\sigma_{A,b}$ fixes a set $S$ that spans $\mathbb{F}_2^m$, then $\sigma_{A,b} = Id$.

Proof: Let $s \in S$ and $S' = S - s$. Then $S'$ also spans $\mathbb{F}_2^m$, and $A$ fixes $S'$, in which case $A = 1$. Then $b = 0$. Note $\sigma_{1,0} = Id$.

- Claim 2: Any set $S \subseteq \mathbb{F}_2^m$ with size $> 2^{m-1}$ spans $\mathbb{F}_2^m$.

Proof: Let $B \subseteq S$ be a maximal set that consists of linearly independent vectors. Since $B$ spans $S$, $2^{|B|} \geq |S| > 2^{m-1}$. Then $|B| = m$. So $B$, and therefore $S$, spans $\mathbb{F}_2^m$.

$\rightarrow$ No none-identity affine transformation can fix $>2^{m-1}$ vectors.
Open Question and Notes

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

• Cautionary notes
  – Shor-like algorithms are unlikely to help break code-based cryptosystems using HSP-hard codes.
  – **But** we have not shown that other quantum algorithms, or even classical ones, cannot break code-based cryptosystems.
  – Nor have we shown that such an algorithm would violate a natural hardness assumption (such as lattice-based cryptosystems and Learning With Errors).