Easy and hard functions for the Boolean hidden shift problem

Maris Ozols  
(IBM)

Andrew Childs, Robin Kothari  
(University of Waterloo & IQC)

Martin Roetteler  
(NEC Labs)

arXiv:1304.4642

May 21, 2013
Outline

1. Motivation and problem
2. Hard instances
3. Easy instances
   - bent functions
   - random functions
4. Conclusions
Motivation

Hidden subgroup problem

- Factoring [Sho97]
- Discrete logarithm [Sho97]
- Pell’s equation [Hal07]
- Lattice problems [Reg04], [Kup05]
- Graph isomorphism [AMRR11]

Attacks on cryptosystems?
New algorithms [ORR12]?

Legendre symbol [vDHI06]
Motivation

Hidden shift problem

Legendre symbol [vDHI06]

Attacks on cryptosystems

New algorithms [ORR12]

Hidden subgroup problem

Factoring [Sho97]

Discrete logarithm [Sho97]

Pell’s equation [Hal07]

Lattice problems [Reg04] [Kup05]

Graph isomorphism [AMRR11]
Problem

Boolean hidden shift problem

- **Given:** complete description of $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ and access to a black-box oracle for $f_s(x) := f(x + s)$
Problem

Boolean hidden shift problem

- **Given:** complete description of $f: \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2$ and access to a black-box oracle for $f_s(x) := f(x + s)$
- **Determine:** hidden shift $s \in \mathbb{Z}_2^n$
Problem

Boolean hidden shift problem

- **Given:** complete description of \( f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2 \) and access to a black-box oracle for \( f_s(x) := f(x + s) \)
- **Determine:** hidden shift \( s \in \mathbb{Z}_2^n \)

Quantum query complexity

- **Oracle:** \( O_{f_s} \): \( |x\rangle \mapsto (-1)^{f(x+s)}|x\rangle \)
Problem

Boolean hidden shift problem

- **Given:** complete description of $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ and access to a black-box oracle for $f_s(x) := f(x + s)$
- **Determine:** hidden shift $s \in \mathbb{Z}_2^n$

Quantum query complexity

- **Oracle:** $O_{f_s} : |x\rangle \mapsto (-1)^{f(x+s)}|x\rangle$
- $Q(\text{BHSP}_f) :=$ bounded error quantum query complexity of the Boolean hidden shift problem for function $f$
Hard instances

Delta functions

- \( f(x) := \delta_{x,x_0} \) for some \( x_0 \in \mathbb{Z}_2^n \)
Hard instances

Delta functions

- $f(x) := \delta_{x,x_0}$ for some $x_0 \in \mathbb{Z}_2^n$
Hard instances

Delta functions

- $f(x) := \delta_{x,x_0}$ for some $x_0 \in \mathbb{Z}_2^n$
- Equivalent to Grover’s search: $\Theta(\sqrt{2^n})$

Brute force approach

Completely extract the truth table of $f_s$

Oracle identification problem [AIK+04]

$Q(BHSP_f) = \mathcal{O}(\sqrt{2^n})$
Hard instances

Delta functions

- \( f(x) := \delta_{x,x_0} \) for some \( x_0 \in \mathbb{Z}_2^n \)
- Equivalent to Grover’s search: \( \Theta(\sqrt{2^n}) \)

Brute force approach

- Completely extract the truth table of \( f_s \)
Hard instances

Delta functions

- \( f(x) := \delta_{x,x_0} \) for some \( x_0 \in \mathbb{Z}_2^n \)
- Equivalent to Grover’s search: \( \Theta(\sqrt{2^n}) \)

Brute force approach

- Completely extract the truth table of \( f_s \)
- Oracle identification problem \([\text{AIK}^+04]\)
Hard instances

Delta functions

- \( f(x) := \delta_{x,x_0} \) for some \( x_0 \in \mathbb{Z}_2^n \)
- Equivalent to Grover’s search: \( \Theta(\sqrt{2^n}) \)

Brute force approach

- Completely extract the truth table of \( f_s \)
- Oracle identification problem [AIK+04]
- \( Q(BHSP_f) = O(\sqrt{2^n}) \)
Hard instances

Algorithm

1. Use Grover’s algorithm to find some $x_0$ with $f_s(x_0) = 1$
2. Brute force through all $s$ that give $f_s(x_0) = 1$
Hard instances

Algorithm

1. Use Grover’s algorithm to find some $x_0$ with $f_s(x_0) = 1$
2. Brute force through all $s$ that give $f_s(x_0) = 1$

Complexity

- $|f| :=$ the Hamming weight of the truth table of $f$
Hard instances

Algorithm

1. Use Grover’s algorithm to find some $x_0$ with $f_s(x_0) = 1$
2. Brute force through all $s$ that give $f_s(x_0) = 1$

Complexity

- $|f| :=$ the Hamming weight of the truth table of $f$
- $Q(BHSP_f) = \frac{\pi}{4} \sqrt{2^n / |f|} + O(\sqrt{|f|})$
Hard instances

Algorithm

1. Use Grover’s algorithm to find some $x_0$ with $f_s(x_0) = 1$
2. Brute force through all $s$ that give $f_s(x_0) = 1$

Complexity

- $|f|$ := the Hamming weight of the truth table of $f$
- $Q(BHSP_f) = \frac{\pi}{4} \sqrt{2^n/|f|} + O(\sqrt{|f|})$
- $Q(BHSP_f) = \Omega\left(\sqrt{2^n/|f|}\right)$ via adversary method
Hard instances

Algorithm

1. Use Grover’s algorithm to find some $x_0$ with $f_s(x_0) = 1$
2. Brute force through all $s$ that give $f_s(x_0) = 1$

Complexity

- $|f| :=$ the Hamming weight of the truth table of $f$
- $Q(BHSP_f) = \frac{\pi}{4} \sqrt{2^n/|f|} + O(\sqrt{|f|})$
- $Q(BHSP_f) = \Omega(\sqrt{2^n/|f|})$ via adversary method

Punchline

- For $f$ to be hard, it is necessary that $|f|$ is $O(1)$ or $\Theta(2^n)$
- Delta functions are the hardest instances
- Hamming weight alone does not determine hardness
Easy instances

Algorithm [Röt10]

\[
|0\rangle^{\otimes n} \xrightarrow{H^{\otimes n}} O_f s \xrightarrow{H^{\otimes n}} D^{-1} \xrightarrow{H^{\otimes n}} |s\rangle
\]
Easy instances

Algorithm [Röt10]

\[ |\Phi(s)\rangle \]

\[ |0\rangle^\otimes n \rightarrow H^\otimes n \circlearrowright O_{fs} \circlearrowright H^\otimes n \rightarrow D^{-1} \circlearrowright H^\otimes n \rightarrow |s\rangle \]

- \[ |\Phi(s)\rangle := \sum_{w \in \mathbb{Z}_2^n} (-1)^{s \cdot w} \hat{F}(w) |w\rangle \]
Easy instances

Algorithm [Röt10]

\[ |\Phi(s)\rangle = \sum_{w \in \mathbb{Z}_2^n} (-1)^{s \cdot w} \hat{F}(w) |w\rangle \]

\[ D := \text{diag}(\sqrt{2^n} \hat{F}(w)), \text{ may not be unitary in general} \]
Easy instances

Algorithm [Röt10]

\[ |\Phi(s)\rangle = \sum_{w \in \mathbb{Z}_2^n} (-1)^{s \cdot w} \hat{F}(w) |w\rangle \]

- \(D := \text{diag}(\sqrt{2^n} \hat{F}(w))\), may not be unitary in general

Bent functions

- \(|\hat{F}(w)| = 1/\sqrt{2^n}\) for all \(w \in \mathbb{Z}_2^n\)
- \(D\) is unitary
- Exact algorithm with one query!
Easy instances

Algorithm [Röt10]

\[ |\Phi(s)\rangle \]

\[ |0\rangle^\otimes n \xrightarrow{H^\otimes n \ O_f \ H^\otimes n \ D^{-1} \ H^\otimes n} |s\rangle \]

- \[ |\Phi(s)\rangle := \sum_{w\in\mathbb{Z}_2^n} (-1)^{s\cdot w} \hat{F}(w) |w\rangle \]
- \[ D := \text{diag}(\sqrt{2^n} \hat{F}(w)) \], may not be unitary in general

Bent functions

- \[ |\hat{F}(w)| = 1/\sqrt{2^n} \text{ for all } w \in \mathbb{Z}_2^n \]
- \[ D \text{ is unitary} \]
- Exact algorithm with one query!

Converse

If an exact one-query algorithm exists for BHSP\(_f\) then \(f\) is bent
Easy instances

PGM algorithm

1. Prepare $|\Phi^t(s)\rangle := \left( O_{f_s} |+\rangle \otimes n \right) \otimes t$

2. Perform Pretty Good Measurement for $\{ |\Phi^t(s)\rangle : s \in \mathbb{Z}_2^n \}$
Easy instances

PGM algorithm

1. Prepare $|\Phi^t(s)\rangle := \left( O_{fs} |+ \rangle \otimes n \right) \otimes^n$

2. Perform Pretty Good Measurement for $\{ |\Phi^t(s)\rangle : s \in \mathbb{Z}_2^n \}$

For $t = 1$ this agrees with [Röt10]
Easy instances

PGM algorithm

1. Prepare $|\Phi^t(s)\rangle := \left( O_f \left| + \right\rangle \otimes n \right) \otimes t$

2. Perform Pretty Good Measurement for $\{ |\Phi^t(s)\rangle : s \in \mathbb{Z}_2^n \}$

For $t = 1$ this agrees with [Röt10]

Random functions are easy

- $f$ is chosen uniformly at random
- $s$ is chosen adversarially

PGM solves BHSP$_f$ with two queries and expected success probability exponentially close to 1
Easy instances

PGM algorithm

1. Prepare $|\Phi^t(s)\rangle := \left(O_f|+\rangle\otimes^n\right)^\otimes t$

2. Perform Pretty Good Measurement for \(\{\Phi^t(s) : s \in \mathbb{Z}_2^n\}\)

For \(t = 1\) this agrees with [Röt10]

Random functions are easy

- \(f\) is chosen uniformly at random
- \(s\) is chosen adversarially

PGM solves BHSP\(_f\) with two queries and expected success probability exponentially close to 1

Proof involves: second moment method, a \(t\)-fold generalization of the Fourier transform, combinatorics of pairings
## Comparison

<table>
<thead>
<tr>
<th>Approach</th>
<th>Functions</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>delta</td>
<td>bent</td>
<td>random</td>
<td></td>
</tr>
<tr>
<td>PGM</td>
<td>$O(2^n)$</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>[ORR12]</td>
<td>$O(\sqrt{2^n})$</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[GRR11]</td>
<td>$O(n\sqrt{2^n})$</td>
<td>$O(n)$</td>
<td>$O(n)$</td>
<td></td>
</tr>
<tr>
<td>[AS05]</td>
<td>$O(n \log n \sqrt{2^n})$</td>
<td>$O(n \log n)$</td>
<td>$O(n \log n)$</td>
<td></td>
</tr>
<tr>
<td>Lower bounds:</td>
<td>$\Omega(\sqrt{2^n})$</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
Conclusions

Summary

- $O(\sqrt{2^n})$ queries for any $f$
- $\Theta(\sqrt{2^n/|f|})$ queries when $|f|$ is small
- Exact one-query algorithm $\iff f$ is bent
- Two queries suffice for random $f$
Conclusions

Summary

- $O(\sqrt{2^n})$ queries for any $f$
- $\Theta(\sqrt{2^n/|f|})$ queries when $|f|$ is small
- Exact one-query algorithm $\iff f$ is bent
- Two queries suffice for random $f$

Open questions

- Query-optimal quantum algorithm for all $f$
- Time-efficient algorithm for some $f$
- Applications in cryptography

Thank you!
Conclusions

Summary

- \(O(\sqrt{2^n})\) queries for any \(f\)
- \(\Theta(\sqrt{2^n/|f|})\) queries when \(|f|\) is small
- Exact one-query algorithm \(\iff f\) is bent
- Two queries suffice for random \(f\)

Open questions

- Query-optimal quantum algorithm for all \(f\)
- Time-efficient algorithm for some \(f\)
- Applications in cryptography

Thank you!


Bibliography II


