Quantum Random Oracles

(References)

Dominique Unruh

Insufficiency of classical RO. The fact that the classical RO is not a good model in the quantum case was already observed in [BDF$^+$11], using the fact that the quadratic speedup in inverting a hash function is only captured by the QRO. In [YZ20], an example protocol is given that is secure in the RO and completely insecure in the QRO (not just a quadratic gap in attack complexity).

One-wayness. Hardness of preimage-finding / one-wayness of the QRO can be shown elementarily (slight adaptation of the optimality of Grover in [NC10], for example), is shown in different variations in a number of papers, and can also be shown easily using the O2H theorem. The specific bound given in the talk follows from [HRS16, Theorem 1 in the eprint].

Collision resistance. Collision resistance of the QRO is shown in [Zha15], together with other useful properties such as the indistinguishability of a random function and a random permutation.

Replacing the oracle. The "history-free reductions" from [BDF$^+$11] essentially do what I called "replacing the oracle". [BDF$^+$11] proves several special cases of full-domain hash using this method. Oracle-indistinguishability shows that two oracles are indistinguishishable if the distributions of the individual outputs are indistinguishishable [Zha12a, Section 7 of the eprint].

One-way to hiding. The original one-way to hiding theorem was presented in [Unr15]. More advanced O2H theorem, e.g., in [AHU19].

Compressed oracles. Compressed oracles were introduced in [Zha19]. The presentation in my talk is based on the introduction from [Unr21, Section 3.1].

Further techniques. A few useful techniques that I didn't cover: Small-range distributions [Zha12a], allowing us to see the QRO as a function with small range. 2q-wise independent functions [Zha12b, Thm. 6.1 of the eprint], allowing us simulate the QRO efficiently without using computational assumptions. The "polynomial-method" and the "adversary method" are useful tools for query complexity related questions (I am not very familiar with them, one example of the polynomial method is in [Zha15]).

References

[AHU19] Andris Ambainis, Mike Hamburg, and Dominique Unruh. Quantum security proofs using semi-classical oracles. In *CRYPTO 2019*, pages 269–295. Springer, 2019. eprint `https://eprint.iacr.org/2018/904.pdf`.

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In *Asiacrypt 2011*, pages 41–69, Berlin, Heidelberg, 2011. Springer. eprint `https://eprint.iacr.org/2010/428.pdf`.

[HRS16] Andreas Hülsing, Joost Rijneveld, and Fang Song. Mitigating multi-target attacks in hash-based signatures. In *PKC 2016, Proceedings, Part I*, volume 9614 of *LNCS*, pages 387–416. Springer, 2016. eprint is `https://eprint.iacr.org/2015/1256.pdf`.

[NC10] M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, 10th anniversary edition, 2010.

[Unr15] Dominique Unruh. Revocable quantum timed-release encryption. *Journal of the ACM*, 62(6):49:1–49:76, 2015. eprint `https://eprint.iacr.org/2013/606.pdf`.

[Unr21] Dominique Unruh. Compressed permutation oracles (and the collision-resistance of sponge/sha3). `https://eprint.iacr.org/2021/062.pdf`, 2021.

[YZ20] Takashi Yamakawa and Mark Zhandry. A note on separating classical and quantum random oracles. `https://eprint.iacr.org/2020/787.pdf`, 2020.

[Zha12a] Mark Zhandry. How to construct quantum random functions. In *FOCS 2013*, pages 679–687, Los Alamitos, CA, USA, 2012. IEEE Computer Society. eprint is IACR ePrint 2012/182.

[Zha12b] Mark Zhandry. Secure identity-based encryption in the quantum random oracle model. In *Crypto 2012*, volume 7417 of *LNCS*, pages 758–775. Springer, 2012. eprint is `https://eprint.iacr.org/2012/076.pdf`.

[Zha15] Mark Zhandry. A note on the quantum collision and set equality problems. *Quantum Information & Computation*, 15(7&8):557–567, 2015. eprint arXiv:1312.1027v3 [cs.CC].

[Zha19] Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In *Crypto 2019*, pages 239–268. Springer, 2019. Eprint is IACR ePrint 2018/276.

Recap: Random Oracles

- Idealization of hash function
- Some hash-based protocols hard/imposs. to prove:
  FDH, Fiat-Shamir, Fujisaki-Okamoto
- Solution:
  - Replace hash fun by random fun
  - Prove sec.
  - "Conclude" sec. for orig proto

<u>Pro</u>

- Easier proofs
- Get around imposs.
- Efficient protos

<u>Con</u>

- Unsound in general

How it works:

- Take existing sec. def.
- Add $H \xleftarrow{\$} \mathrm{Fun}(X \to Y)$ in the def. game
- Give H to everyone as oracle
- Replace honest hash-calls by H

Why so easy?

Lazy sampling

- Replace rnd. H by "lazy" H
- Initially empty
- For any x: On first $H(x)$-query pick result rnd on demand
- Upon further queries: use cached result

Rnd fun RO $\equiv$ lazy RO

$\Rightarrow$ Can reason about indep. of values more easily

<u>Also</u>: can "program" RO

Quantum RO

Problem 1:

- Classical RO can only be evaluated classically (no superpos.)
- Real-life hash can be eval'd in superpos:

$$\sum_x 2^{-n/2} |x\rangle \longmapsto \sum_x 2^{-n/2} |x\rangle |H(x)\rangle$$

$\Rightarrow$ Allow superpos. queries in QROM!

- $H \xleftarrow{\$} \mathrm{Fun}(X \to Y)$
- Give $|H\rangle$ to everyone as oracle:

$$U_H : |x, y\rangle \longmapsto |x, y \oplus H(x)\rangle$$

Example 1: Preimage-finding

$\forall t$-time A: $\Pr[\text{win}] \le \varepsilon$

$$\boxed{\begin{array}{l} y \xleftarrow{\$} Y \\ x \leftarrow A(y) \\ \text{win} := [\, f(x) = y \,] \end{array}}$$

In QRO:

$\forall t$-time q-query A: $\Pr[\text{win}] \le \varepsilon$

$$\boxed{\begin{array}{l} H \xleftarrow{\$} \mathrm{Fun}(X \to Y) \\ y \xleftarrow{\$} Y \\ x \leftarrow A^{|H\rangle}(y) \\ \text{win} := [\, H(x) = y \,] \end{array}}$$

Fact: $\Pr[\text{win}] \le O(q^2/2^{-m})$

Example 2: Collision resistance

$$H \overset{\$}{\leftarrow} \mathrm{Fun}(X \to Y)$$

$$x, x' \leftarrow A^{|H\rangle}$$

$$\mathrm{win} := [x \neq x', \quad H(x) = H(x')]$$

$$\Pr[\mathrm{win}] \leq O(q^3/2^{-m})$$

QROM

Pro	Con
- Allows to overcome imposs. - More eff. protos	- Unsound - Proofs harder

Why are proofs harder?

Lazy sampling does not work anymore.

Example: $A^{|H\rangle}$ does:

Queries $\sum 2^{-n/2} |x\rangle |0\rangle$

$\rightsquigarrow \sum 2^{-n/2} |x\rangle | H(x)\rangle$

$\Rightarrow$ all of H involved

$\Rightarrow$ cannot argue about "unqueried" values.

Rest of talk: QROM proof techniques

Technique 1: Replacing the oracle

- Replace H by diff. function chosen with same (or close) distrib.

E.g.: for perm. π,

$$H \rightsquigarrow \pi \circ H$$

Consider:

G_1

$$\boxed{\begin{array}{l} H \xleftarrow{\$} \mathrm{Fun}(X \to X) \\ x, x' \leftarrow A^H \\ \mathrm{win} := [\, H(x) \oplus H(x') = x \oplus x', \\ \qquad\qquad x \neq x' \,] \end{array}}$$

TS: $\Pr[\mathrm{win}]$ small

G_2

$H \xleftarrow{\$} \mathrm{Fun}(X \to Y)$

$G := (x \mapsto H(x) \oplus x)$

$x, x' \leftarrow A^G$

$\mathrm{win} := [\, G(x) \oplus G(x') = x \oplus x',\ x \neq x' \,]$

$\Pr[\mathrm{win} : G_2] = \Pr[\mathrm{win} : G_1]$

G_3

$H \xleftarrow{\$} \mathrm{Fun}(X \to Y)$

$x, x' \leftarrow \tilde{A}^H$

$\mathrm{win} := [\, \underbrace{H(x) \oplus x \oplus H(x') \oplus x' = x \oplus x'}_{H(x) = H(x')},\ x \neq x' \,]$

$\underbrace{\Pr[\mathrm{win} : G_3]}_{\leq O(q^3/2^m)} = \Pr[\mathrm{win} : G_2]$

- Works for some special cases of FDH.
- Sometimes nice to repl. by indist. G

  ⇝ Useful: "Oracle indist"

<u>Technique 2:</u> Oneway to hiding (O2H)

"Replacing the RO" technique: Change in very beginning, 100% consistency

<u>But:</u> Sometimes we need inconsistent replacement (change RO somewhere, still use orig $H(x)$ somewhere else)

→ Hope adv does not notice!

Classically: Adv cannot notice unless adv queries changed value:

$$\left|\Pr[\text{win} : \text{orig-game}] - \Pr[\text{win} : \text{newgame}]\right|$$
$$\leq \Pr[\text{query } H(x) : \text{new game}]$$

Can we do this quantumly?

meaning?

Example:

owp

$$\mathrm{Enc}(m) := \left(f(r),\ m \oplus H(r)\right)$$

Claim: IND-CPA sec.

(G1)

$H \xleftarrow{\$} \mathrm{Fun}(X \to Y)$ $\quad b \xleftarrow{\$} \{0,1\}$

$m_0, m_1 \leftarrow A^H$

$c \leftarrow \mathrm{Enc}(m_b)$

$b' \leftarrow A^H(c)$

$\mathrm{win} := [b' = b]$

TS:

$\Pr[\mathrm{win}] \approx 1/2$

G_2
$$
\begin{array}{l}
H \xleftarrow{\$} \mathrm{Fun}(X \to Y) \quad r \xleftarrow{\$} X \\
b \xleftarrow{\$} \{0,1\} \\
m_0, m_1 \leftarrow A^H \\
b' \leftarrow A^H(f(r),\ m_b \oplus H(r))
\end{array}
$$

$$\Pr[\mathrm{win} : G_2] = \Pr[\mathrm{win} : G_1]$$

G_3
$$
\begin{array}{l}
H \xleftarrow{\$} \mathrm{Fun}(X \to Y) \quad r \xleftarrow{\$} X \quad y \xleftarrow{\$} Y \\
b \xleftarrow{\$} \{0,1\} \\
m_0, m_1 \leftarrow A^H \\
b' \leftarrow A^H(f(r),\ m_b \oplus y) \\
\mathrm{win} := [b' = b]
\end{array}
$$

$$\Pr[\mathrm{win} : G_3] = 1/2$$

Classically:

$$|\Pr[\mathrm{win} : G_2] - \Pr[\mathrm{win} : G_3]|$$
$$\leq \Pr[A^H \text{ queries } r : G_3] \approx 0$$

How to do this quantumly?

Problem: "A^H queries r" not well-def.

Trick: We "def" $\Pr[A^H \text{ queries } r]$ as $\Pr$[we see r if we stop A at random query and measure query reg.]

Thm (orig O2H)

Fix adv C (q-queries)

Let $B^H(x,y)$ run $C^H(x,y)$ till i-th query ($i \xleftarrow{\$} \{1..q\}$), and measure + output query-reg.

Then:

$$\left|\Pr[C^H(x,H(x))=1]-\Pr[C^H(x,y)=1]\right|$$
$$\leq q\sqrt{\Pr[B^H(x,y)=x]}$$

$\underline{C^H(r, H(r))}$

$H \xleftarrow{\$} Fun(X \to Y) \qquad r \xleftarrow{\$} X$

$b \xleftarrow{\$} \{0,1\}$

$m_0, m_1 \leftarrow A^H$

$b' \leftarrow A^H(f(r), m_b \oplus H(r))$

$\underline{C^H(r, y)} = G_3$

$$\xRightarrow{O2H} \left| Pr[win : G_2] - Pr[win : G_3] \right|$$

$$\le O\left(q\sqrt{Pr[win : G_{2\frac{1}{2}}]}\right)$$

$G_{2\frac{1}{2}}$
- Runs G_3 till i-th query
- Measure $\rightsquigarrow \tilde{r}$
- win := $[r = \tilde{r}]$

$Pr[win : G_{2\frac{1}{2}}] \approx 0$ (by f OWP)

Orig O2H limited

- Only one pos reprogrammable
- Only for uniformly rnd oracles
- x, y uniform

$\Downarrow$

~~Then~~ Future work solves this

Technique 3: Compressed oracles

Lazy sampling

→ Keep track of adv-queries and answers

→ Efficient rep of RO

I said: cannot have "log" because $\sum |x\rangle|H(x)\rangle$ would put everything in the log.

But we could have entangled log:

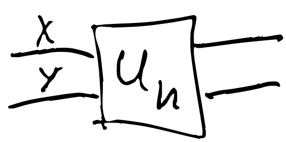
$$\sum_x \underbrace{|x\rangle |H(x)\rangle}_{\text{adv state}} \quad \underbrace{|x\rangle}_{\text{log}}$$

Compressed Oracles

Step 1: RO as superpos of funcs

Normal QRO

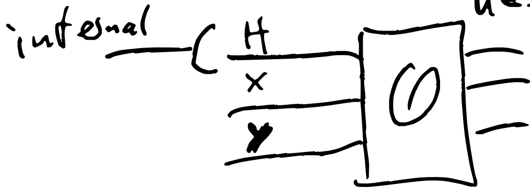
$h \xleftarrow{\$} \mathrm{Fun}(X \to Y)$



$|x, y\rangle \mapsto |x, y \oplus H(x)\rangle$

Diff view

$$H \leftarrow \sum_{h \in \mathrm{Fun}(X \to Y)} |h\rangle \qquad (\text{"std oracle"})$$



$|h, x, y\rangle$

$\mapsto |h, x, y \oplus h(x)\rangle$

Fact: U_H, O perf. indist.

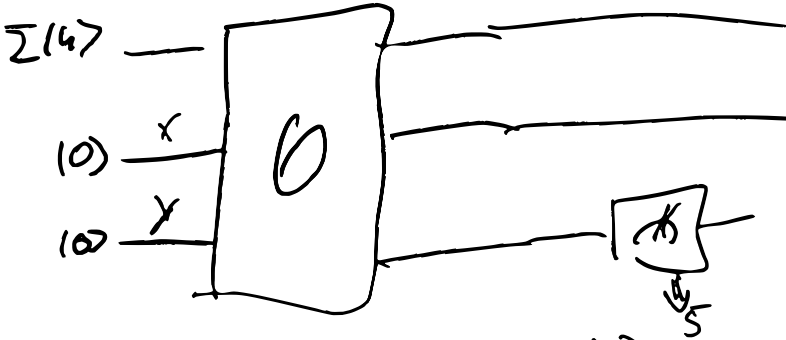
Pro: state of H tells us something about how much/what is def'd in the RO

Eg: if "$H = \sum |k\rangle$" the RO is completely unknown

E.g: if "$H = \sum_{\substack{h \\ h(0)=0}} |h\rangle$" then $h(0)$ has been sampled

$\Rightarrow$ Kind of "lazy sampling"

(But in a very hard to use form.)



$$\sum_h |h, 0, h(0)\rangle$$

$$\downarrow$$

$$\sum_{\substack{h \\ h(0)=5}} |h, 0, 5\rangle$$

Representing H (the oracle-state-reg)

Easiest to work with

$$H = H_1 H_2 \dots H_N$$

(H_x contains the $h(x)$ output)

Eg: $H_1 = |0\rangle + |1\rangle$, $H_x = |0\rangle$ $(x \neq 1)$

means $H = |f_0\rangle + |f_1\rangle$

$f_0 = 0$, $f_1(0) = 1$, $= 0_{\text{else}}$

In particular: Init state:

$H_1 \leftarrow \sum |y\rangle =: |*\rangle$, $H_2 \leftarrow |*\rangle, \dots$

Also allow $|\bot\rangle$ in H_x

<u>Step 2</u> Identifying unqueried inputs

$H_x = |*\rangle$ means $h(x)$ is unqueried

To "mark" those, apply unitary like this to every H_x:

$$\text{Compress}_1 : |*\rangle \to |\bot\rangle$$
$$|y\rangle \to |y\rangle$$

If we apply Compress_1 to all H_x in init state, we get:

$$H = |\bot\rangle \dots |\bot\rangle = |\emptyset\rangle$$

If, e.g. $h(0)=5$ was queried

$H = |5\rangle|*\rangle \dots |*\rangle$ (before compr.)

$H = |5\rangle|\bot\rangle \dots |\bot\rangle$ (after compr.)

$= |0 \mapsto 5\rangle$

Compressed oracle:

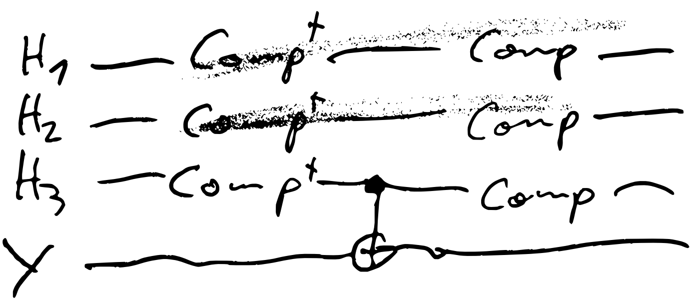
Init. state : $H \leftarrow |\emptyset\rangle = |\bot\rangle \dots |\bot\rangle$

Upon query:

- $\mathrm{Compress}_1^{\dagger}$ on each H_x
- $\mathcal{O}$ (std. oracle)
- $\mathrm{Compress}_1$

CO perf. ind. from std oracle $\mathcal{O}$

If $X = |3\rangle$



Conseqs

- H_x is modified only if we query x
- Each query can make ≤ 1 $H_x \notin |\perp\rangle$

$\Longrightarrow H = \sum \alpha_h |h\rangle$ with all h having $\leq q$ entries

$\Longrightarrow$ Comp. oracle

Problem: Compress_1 does not exist.

$\text{Comp}|y\rangle = |y\rangle$

$\Rightarrow \text{Comp}|*\rangle = \sum_y \text{Comp}|y\rangle$

$= \sum_y |y\rangle = |*\rangle = |\perp\rangle$

Instead:

$$\text{Compress}_1 |*\rangle = |\perp\rangle$$

$$\text{Compress}_1 |y\rangle = |y\rangle + \text{small error}$$

$$\left(\text{Compress}_1 := Q U_\perp Q^\dagger\right.$$

$$Q|0\rangle = |*\rangle$$

$$Q|\perp\rangle = |\perp\rangle$$

$$\left. U_\perp |\perp\rangle = |0\rangle,\ U_\perp|0\rangle = |\perp\rangle,\ U_\perp = id_{\text{else}}\right)$$

$\Longrightarrow$ Can change QRO in CO

→ perf. indist.

→ Compact / efficient

→ State of H is a readable log of queries

Example

zero-preimage-finding

$\boxed{G_0}$

$H \xleftarrow{\$} \mathrm{Fun}(X \to Y)$

$x \leftarrow A^H$

$\mathrm{win} := [H(x) = 0]$

Step 1 Replace RO by CO

G_1

$H \leftarrow |\emptyset\rangle = |\bot\rangle \dots |\bot\rangle$

$x \leftarrow A^{CO}$

$y \leftarrow CO(x)$

$\mathrm{win} := [y = 0]$

Invariant: $I := \mathrm{span}\{|h\rangle : 0 \notin \mathrm{image}(h)\}$

$I \otimes \mathcal{H}_{rest}$

Initial state: H satisfies I

In each invocation of CO,
if state sat's I, (before)
then state $O(1/\sqrt{M})$-close to satisfying I

<u>Conseq:</u>

In the end:

- state is $O\left(\frac{q}{\sqrt{M}}\right)$-close to I
- H is superpos of $|h\rangle$ with $h(x)=y$

$$\Pr[y=0] \leq O\left(\frac{q}{\sqrt{M}}\right)^2 = O\left(\frac{q^2}{M}\right)$$

$\blacksquare$