Cryptographic Primitives in Quantum Idealized Models

Samuel Bouaziz--Ermann

PhD Defense

Supervised by Alex Bredariol Grilo and Damien Vergnaud

September 15, 2025

Outline

Introduction

Context

Cryptography

Cryptography is the science of designing mathematical methods to protect information and ensure security properties such as confidentiality, integrity, and authenticity.

Context

Cryptography

Cryptography is the science of designing mathematical methods to protect information and ensure security properties such as confidentiality, integrity, and authenticity.

Context

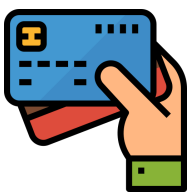
Cryptography

Cryptography is the science of designing mathematical methods to protect information and ensure security properties such as confidentiality, integrity, and authenticity.

Context

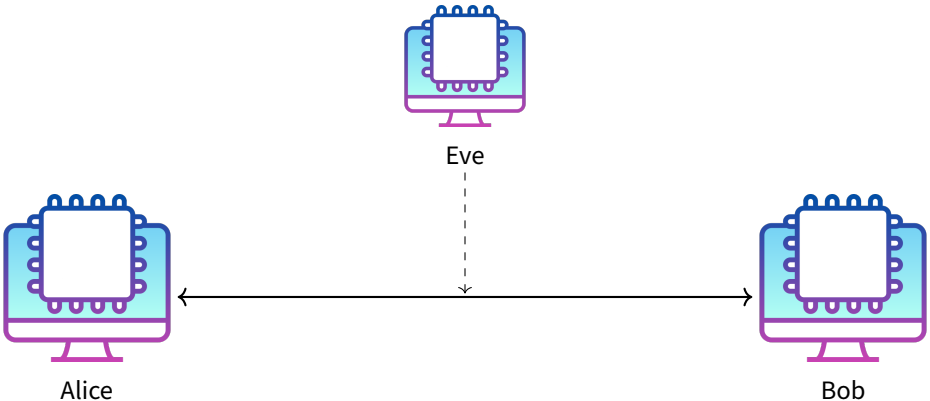
Cryptography

Cryptography is the science of designing mathematical methods to protect information and ensure security properties such as confidentiality, integrity, and authenticity.

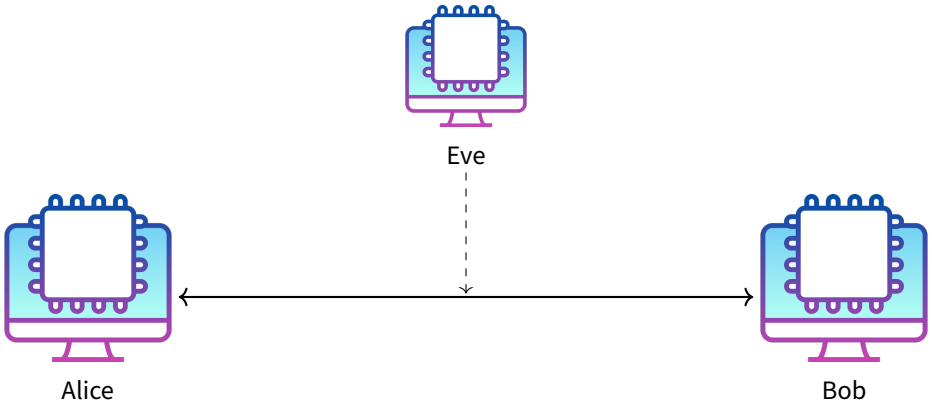


All images in this presentation are from Flaticon.com

Classical Cryptography

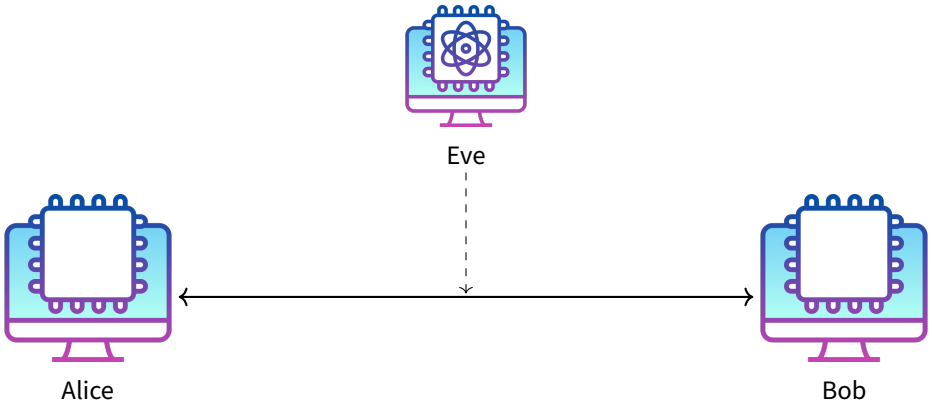


Classical Cryptography

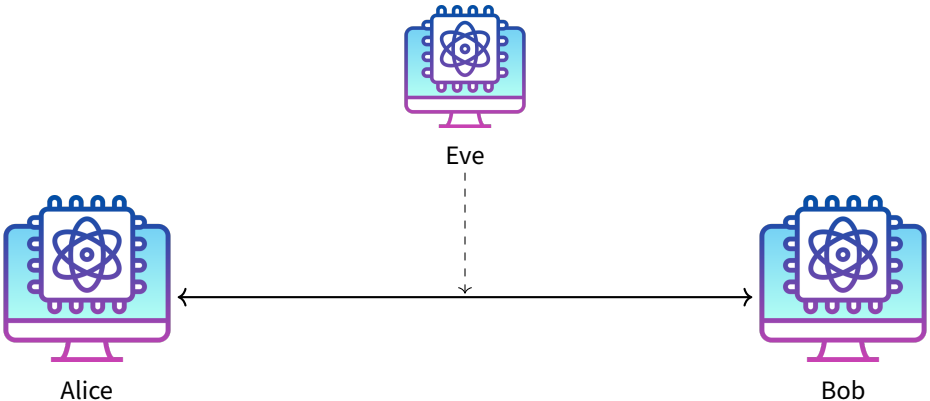


Quantum computers can be used to break classical cryptography [Sho94]!

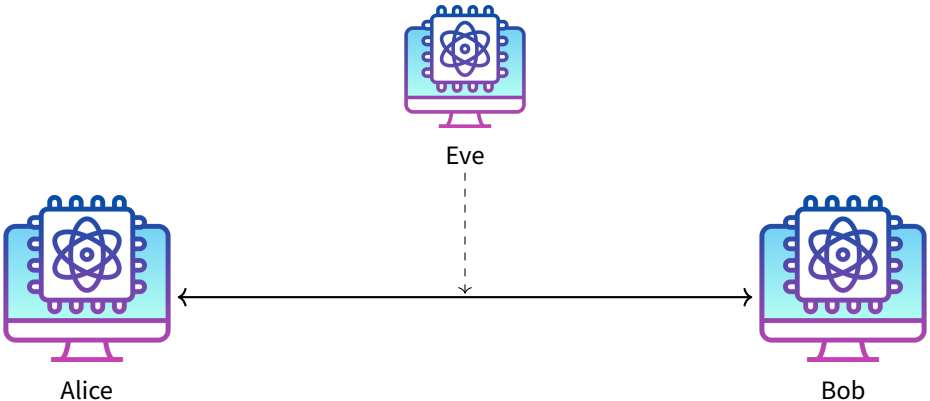
Post-Quantum Cryptography



Quantum Cryptography

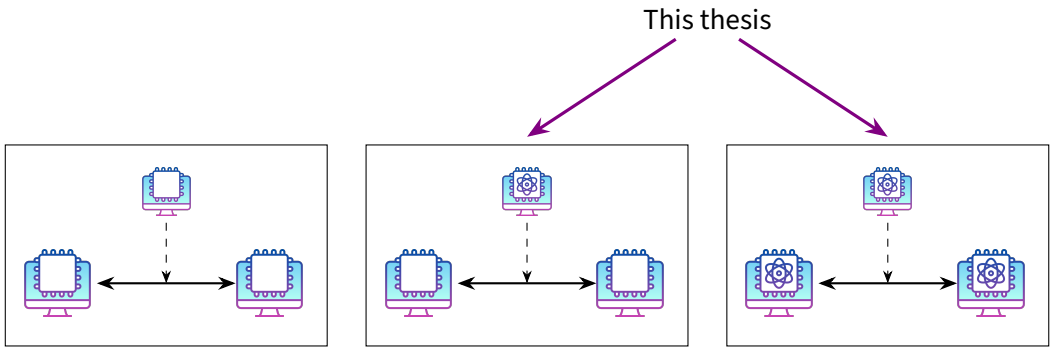


Quantum Cryptography



Example: Quantum Key Distribution (QKD) [BB84].

Context


This thesis
Classical Cryptography
Post-Quantum Cryptography
Quantum Cryptography

This thesis

- This thesis study idealized models, i.e. where primitives are perfect.

This thesis

- This thesis study idealized models, i.e. where primitives are perfect.
- Foundation work

This thesis

- This thesis study idealized models, i.e. where primitives are perfect.
- Foundation work
- The goal is to understand the strength of assumptions and primitives

Results

This thesis is separated in three parts:

- Quantum security of Subset Cover Problems [BGV23] (Part I)
- Key Agreements and Public Key Encryption from One-Way Functions [BGVV24]
- On Limits on the Provable Consequences of Quantum Pseudorandomness [BM24, BHMV25] (Part II)

Quantum Security of the Subset Cover Problems

Signatures

Signatures

- Alice wants to send a message to Bob
- Bob wants to make sure that the message comes from Alice
- Bob wants to verify the integrity of the message

Alice Bob

Signatures

Signatures

- Alice wants to send a message to Bob
- Bob wants to make sure that the message comes from Alice
- Bob wants to verify the integrity of the message

Signatures

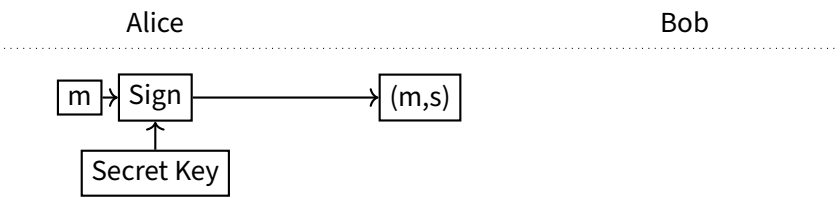
Signatures

- Alice wants to send a message to Bob
- Bob wants to make sure that the message comes from Alice
- Bob wants to verify the integrity of the message

Signatures

Signatures

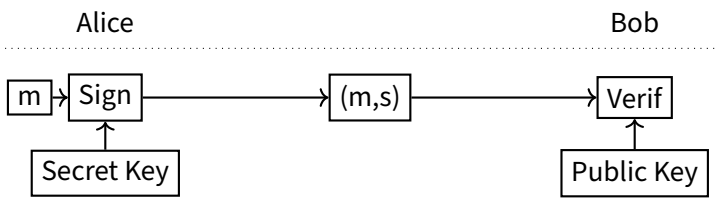
- Alice wants to send a message to Bob
- Bob wants to make sure that the message comes from Alice
- Bob wants to verify the integrity of the message



Signatures

Signatures

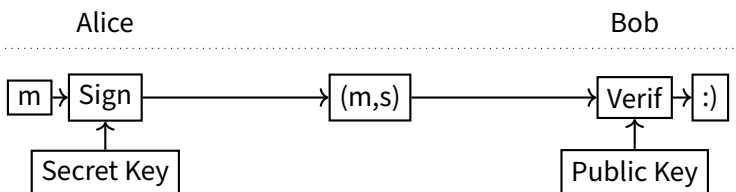
- Alice wants to send a message to Bob
- Bob wants to make sure that the message comes from Alice
- Bob wants to verify the integrity of the message



Signatures

Signatures

- Alice wants to send a message to Bob
- Bob wants to make sure that the message comes from Alice
- Bob wants to verify the integrity of the message



NIST Competition

- 2017: NIST opens a call for post-quantum signatures.

SPHINCS+

NIST Competition

- 2017: NIST opens a call for post-quantum signatures.
- 2024: three signature schemes are selected, including SPHINCS+.

SPHINCS+

NIST Competition

- 2017: NIST opens a call for post-quantum signatures.
- 2024: three signature schemes are selected, including SPHINCS+.

Hash functions

- To build SPHINCS(+), we rely on *hash functions*.

SPHINCS+

NIST Competition

- 2017: NIST opens a call for post-quantum signatures.
- 2024: three signature schemes are selected, including SPHINCS+.

Hash functions

- To build SPHINCS(+), we rely on *hash functions*.
- A hash function maps data of arbitrary length to fixed-sized values.

SPHINCS+

NIST Competition

- 2017: NIST opens a call for post-quantum signatures.
- 2024: three signature schemes are selected, including SPHINCS+.

Hash functions

- To build SPHINCS(+), we rely on *hash functions*.
- A hash function maps data of arbitrary length to fixed-sized values.
- We usually require collision resistance and pre-image resistance, but here we need subset-cover resistance.

Subset Cover

Relation between SPHINCS and the SC problem

- A lower bound for the SC problem is necessary for SPHINCS's security.

Subset Cover

Relation between SPHINCS and the SC problem

- A lower bound for the SC problem is necessary for SPHINCS's security.
- An algorithm that finds a subset cover **is not** an attack on SPHINCS.

Subset Cover

Relation between SPHINCS and the SC problem

- A lower bound for the SC problem is necessary for SPHINCS's security.
- An algorithm that finds a subset cover **is not** an attack on SPHINCS.
- The SC variations (RSC, TSC, and ITSC) are also linked to the security of SPHINCS(+).

Subset Cover

Relation between SPHINCS and the SC problem

- A lower bound for the SC problem is necessary for SPHINCS's security.
- An algorithm that finds a subset cover **is not** an attack on SPHINCS.
- The SC variations (RSC, TSC, and ITSC) are also linked to the security of SPHINCS(+).

Our model

We have quantum access to $h_1, \cdots, h_k$ such that $h_i : \mathcal{X} \to \mathcal{Y}$, and $|\mathcal{Y}| = N$.

Subset Cover

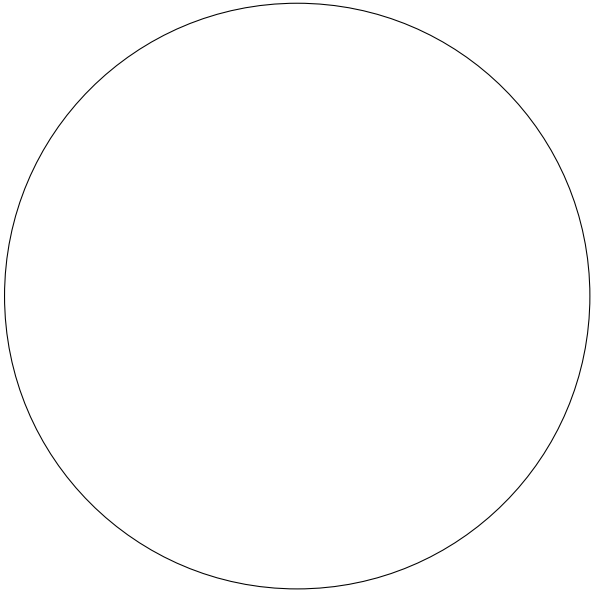
Definition

A (r, k)-subset cover ((r, k)-SC) for $h_1, \ldots, h_k$ consist of $r + 1$ elements $x_0, x_1, \ldots, x_r$ in $\mathcal{X}$ such that:

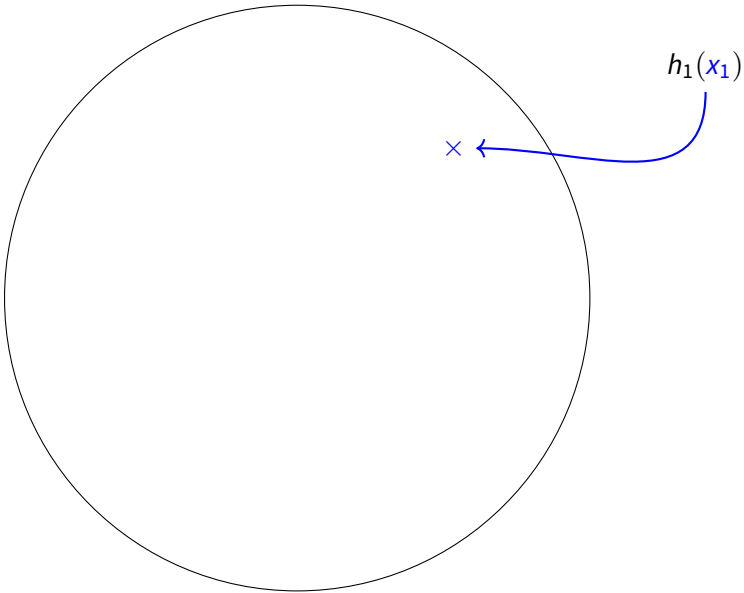
$$x_0 \notin \{x_1, \ldots, x_r\}, \text{ and}$$

$$\{h_i(x_0) | 1 \leq i \leq k\} \subseteq \bigcup_{j=1}^{r} \{h_i(x_j) | 1 \leq i \leq k\}.$$

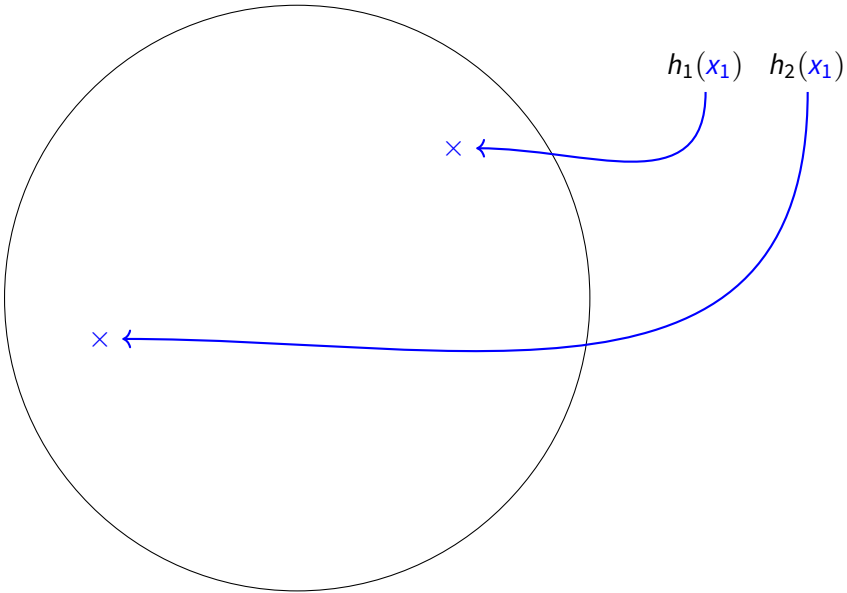
Subset Cover



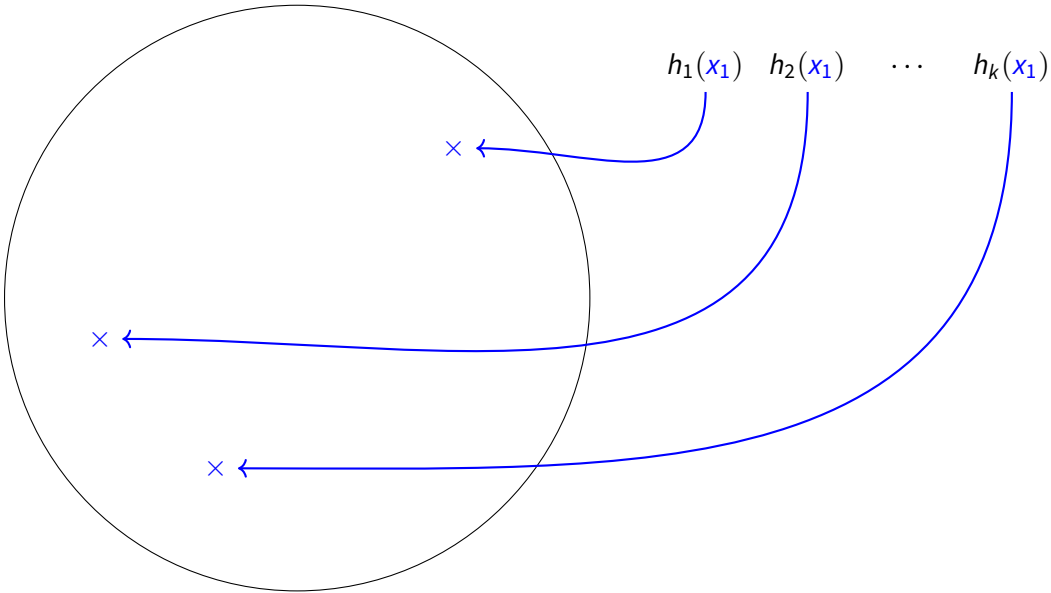
Subset Cover



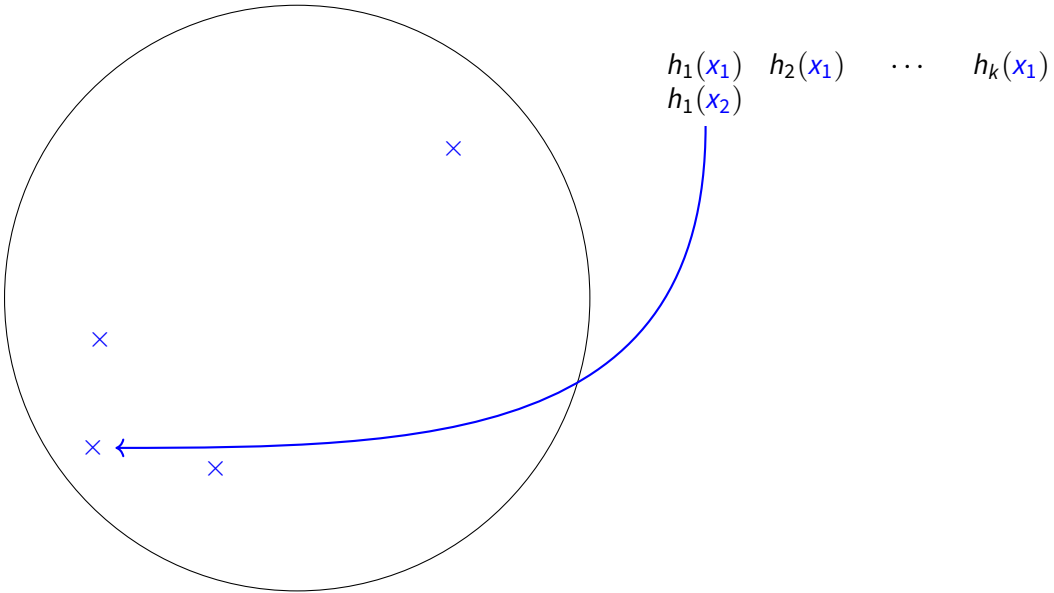
Subset Cover



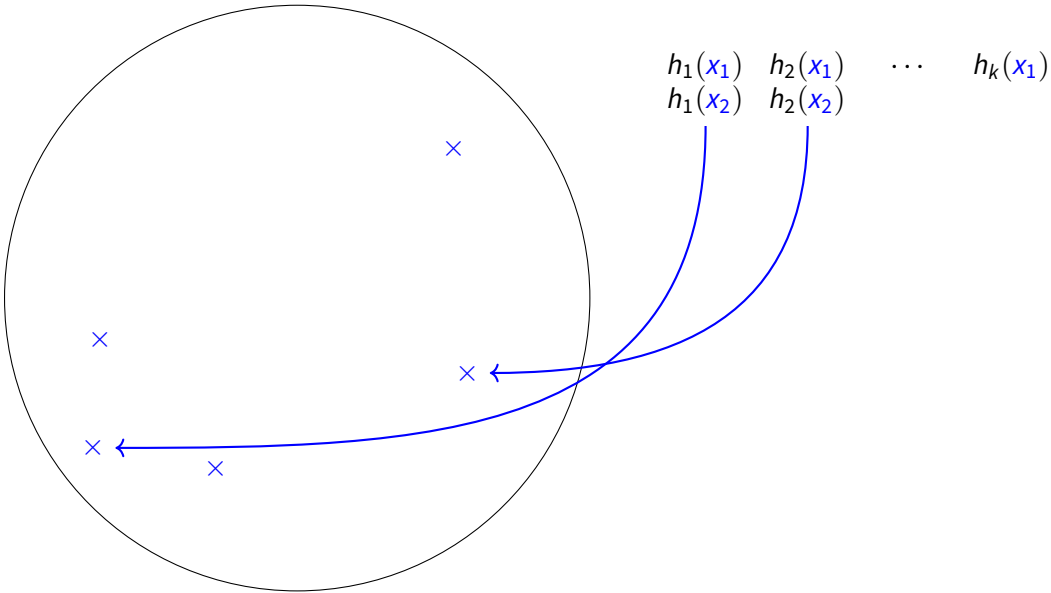
Subset Cover



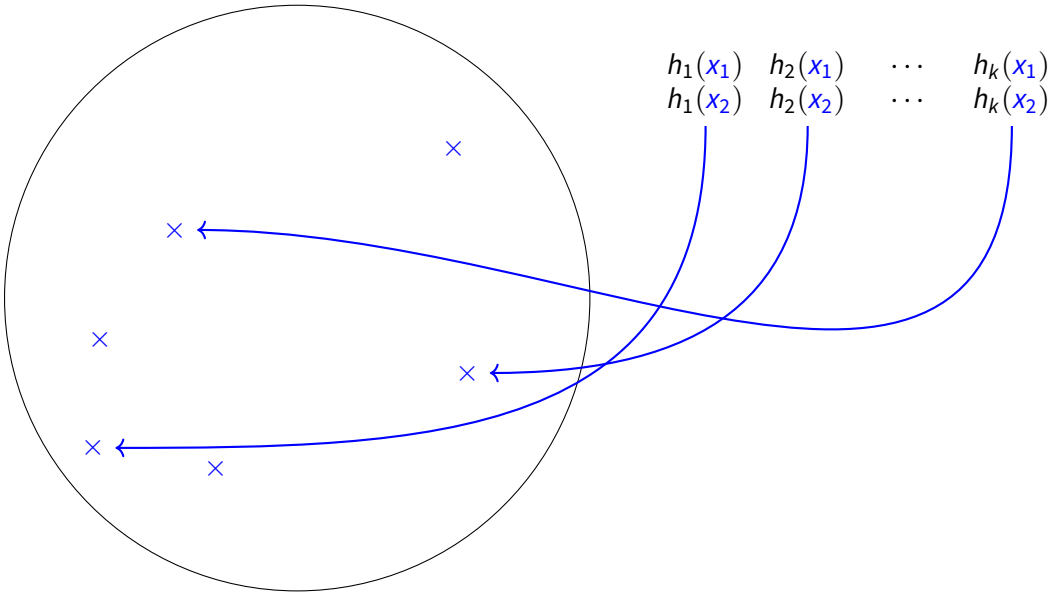
Subset Cover



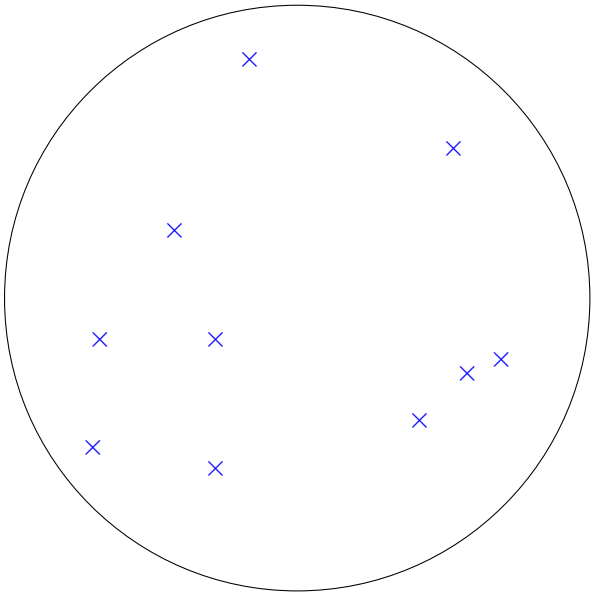
Subset Cover



Subset Cover

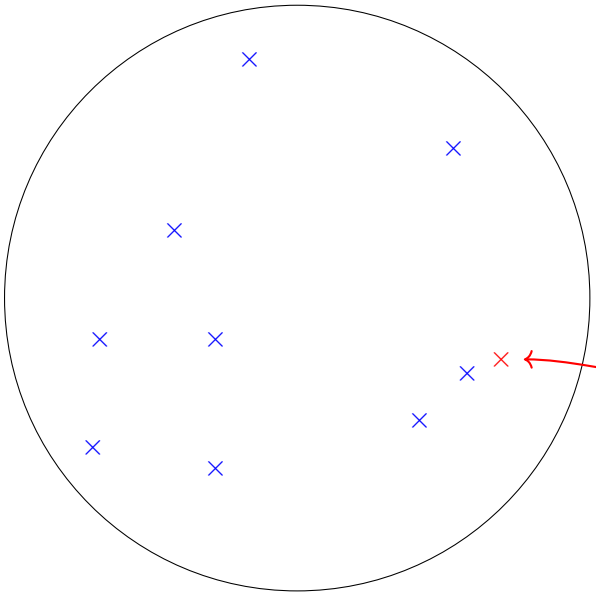


Subset Cover



$$\begin{matrix} h_1(x_1) & h_2(x_1) & \cdots & h_k(x_1) \\ h_1(x_2) & h_2(x_2) & \cdots & h_k(x_2) \\ & \vdots & & \\ h_1(x_r) & h_2(x_r) & \cdots & h_k(x_r) \end{matrix}$$

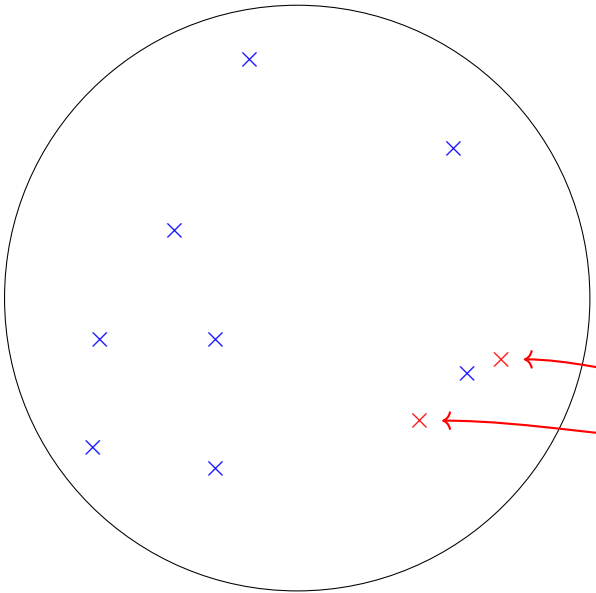
Subset Cover



$$
\begin{array}{cccc}
h_1(x_1) & h_2(x_1) & \cdots & h_k(x_1) \\
h_1(x_2) & h_2(x_2) & \cdots & h_k(x_2) \\
 & \vdots & & \\
h_1(x_r) & h_2(x_r) & \cdots & h_k(x_r)
\end{array}
$$

$h_1(x_0)$

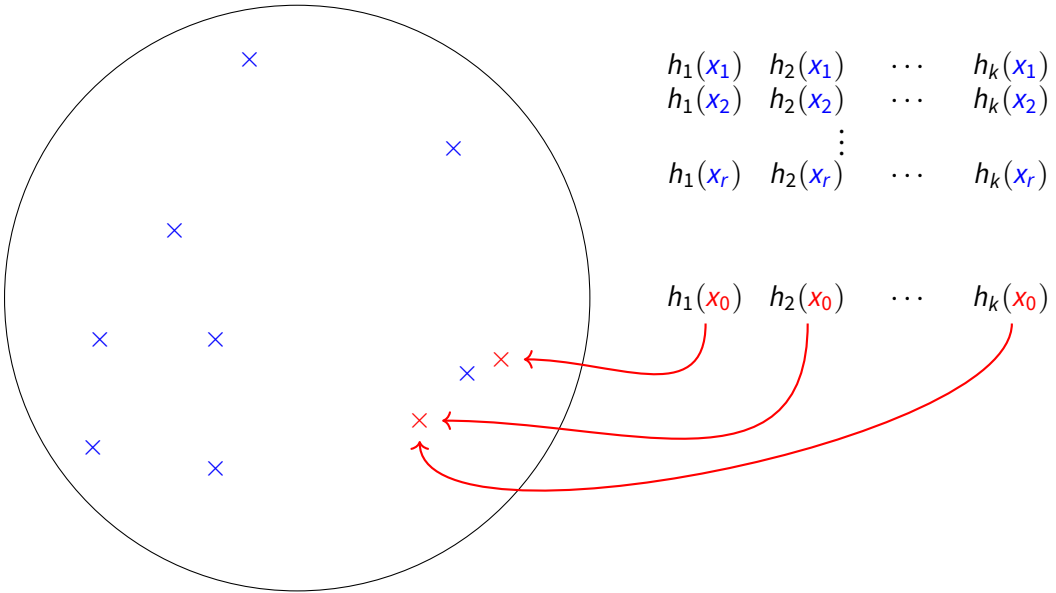
Subset Cover



$$
\begin{array}{cccc}
h_1(x_1) & h_2(x_1) & \cdots & h_k(x_1) \\
h_1(x_2) & h_2(x_2) & \cdots & h_k(x_2) \\
 & \vdots & & \\
h_1(x_r) & h_2(x_r) & \cdots & h_k(x_r)
\end{array}
$$

$$
h_1(x_0) \quad h_2(x_0)
$$

Subset Cover



Subset Cover

Definition

A (r, k)-subset cover ((r, k)-SC) for $h_1, \ldots, h_k$ consist of $r + 1$ elements $x_0, x_1, \ldots, x_r$ in domain $\mathcal{X}$ such that:

$$x_0 \notin \{x_1, \ldots, x_r\}, \text{ and}$$

$$\{h_i(x_0) | 1 \leq i \leq k\} \subseteq \bigcup_{j=1}^{r} \{h_i(x_j) | 1 \leq i \leq k\}.$$

Restricted Subset Cover

Definition

A k-restricted subset cover (k–RSC) for $h_1, \ldots, h_k$ consist of $k+1$ elements $x_0, x_1, \ldots, x_k$ in $\mathcal{X}$ such that:

$$x_0 \notin \{x_1, \ldots, x_k\}, \text{ and}$$

$$\begin{aligned} h_1(x_0) &= h_1(x_1), \\ h_2(x_0) &= h_2(x_2), \\ &\vdots \\ h_k(x_0) &= h_k(x_k), \end{aligned}$$

State of the art

Theorem 1 ([YTA22])

There exists an algorithm that find a k–RSC by making $O\left(k \cdot N^{\frac{2^k-1}{2^{k+1}-1}}\right)$ *queries to* $h_1, \ldots, h_k$.

State of the art

Theorem 1 ([YTA22])

There exists an algorithm that find a k–RSC by making $O\left(k \cdot N^{\frac{2^k-1}{2^{k+1}-1}}\right)$ *queries to* $h_1, \ldots, h_k$.

No other work! (For quantum cryptanalysis)

Our results on k–RSC

Our goal is to start the quantum cryptanalysis of these problems.

Our results on k–RSC

Our goal is to start the quantum cryptanalysis of these problems.

Lower bound on k–RSC

We prove that $\Omega\left(k^{-\frac{2^k}{2^{k+1}-1}} \cdot N^{\frac{2^k-1}{2^{k+1}-1}}\right)$ quantum queries to the idealized hash functions are needed to find a k–RSC with constant probability.

Our results on k–RSC

Our goal is to start the quantum cryptanalysis of these problems.

Lower bound on k–RSC

We prove that $\Omega\left(k^{-\frac{2^k}{2^{k+1}-1}} \cdot N^{\frac{2^k-1}{2^{k+1}-1}}\right)$ quantum queries to the idealized hash functions are needed to find a k–RSC with constant probability.

Lower bound on 2–RSC

Given two random functions $h_1, h_2 : \mathcal{X} \to \mathcal{Y}$ where $|N| = \mathcal{Y}$, a quantum algorithm needs to make $\Omega(N^{3/7})$ queries to h_1 and h_2 to find a 2–RSC with a constant probability.

Quantum Random Oracle

Definition

- In the **Random Oracle Model** (ROM), a hash function is modelized as a *random* function $\mathcal{H} : \mathcal{X} \to \mathcal{Y}$.

Quantum Random Oracle

Definition

- In the **Random Oracle Model** (ROM), a hash function is modelized as a *random* function $\mathcal{H} : \mathcal{X} \to \mathcal{Y}$.
- In the **Quantum Random Oracle Model** (QROM), the random function $\mathcal{H}$ can be queried in superposition. The function $\mathcal{H}$ is fixed before the computation.

$$\sum_{x,y} \alpha_{x,y} |x\rangle |y\rangle \xmapsto{\text{StO}} \sum_{x,y} \alpha_{x,y} |x\rangle |y \oplus \mathcal{H}(x)\rangle$$

Our Model

The Compressed Oracle Model [Zha19]

- We add a new register, for the database: $|d\rangle$. Let $\mathcal{D}$ the set of all possible database.

Our Model

The Compressed Oracle Model [Zha19]

- We add a new register, for the database: $|d\rangle$. Let $\mathcal{D}$ the set of all possible database.
- $|d\rangle$ is initially equal the uniform superposition:

$$\sum_{D\in\mathcal{D}} \frac{1}{|\mathcal{D}|} |D\rangle = \bigotimes_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} \frac{1}{|\mathcal{Y}|} |y\rangle_x$$

Our Model

The Compressed Oracle Model [Zha19]

- We add a new register, for the database: $|d\rangle$. Let $\mathcal{D}$ the set of all possible database.
- $|d\rangle$ is initially equal the uniform superposition:

$$\sum_{D\in\mathcal{D}} \frac{1}{|\mathcal{D}|} |D\rangle = \bigotimes_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} \frac{1}{|\mathcal{Y}|} |y\rangle_x$$

- When we make a query to $\mathcal{O}$:

$$\sum_{x,y,D} \alpha_{x,y,D} |x\rangle |y\rangle |D\rangle \xmapsto{\mathcal{O}} \sum_{x,y,D} \alpha_{x,y,D} |x\rangle |y\rangle |D \oplus (x,y)\rangle$$

Our Model

The Compressed Oracle Model [Zha19]

- We add a new register, for the database: $|d\rangle$. Let $\mathcal{D}$ the set of all possible database.
- $|d\rangle$ is initially equal the uniform superposition:

$$\sum_{D\in\mathcal{D}} \frac{1}{|\mathcal{D}|} |D\rangle = \bigotimes_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} \frac{1}{|\mathcal{Y}|} |y\rangle_x$$

- When we make a query to $\mathcal{O}$:

$$\sum_{x,y,D} \alpha_{x,y,D} |x\rangle |y\rangle |D\rangle \xmapsto{\mathcal{O}} \sum_{x,y,D} \alpha_{x,y,D} |x\rangle |y\rangle |D \oplus (x,y)\rangle$$

There is also a compression part that is omitted here for simplicity.

Our Model

The Compressed Oracle Model [Zha19]

- We add a new register, for the database: $|d\rangle$. Let $\mathcal{D}$ the set of all possible database.
- $|d\rangle$ is initially equal the uniform superposition:

$$\sum_{D\in\mathcal{D}} \frac{1}{|\mathcal{D}|} |D\rangle = \bigotimes_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} \frac{1}{|\mathcal{Y}|} |y\rangle_x$$

- When we make a query to $\mathcal{O}$:

$$\sum_{x,y,D} \alpha_{x,y,D} |x\rangle |y\rangle |D\rangle \xmapsto{\mathcal{O}} \sum_{x,y,D} \alpha_{x,y,D} |x\rangle |y\rangle |D \oplus (x,y)\rangle$$

There is also a compression part that is omitted here for simplicity.
This model is equivalent to the QROM.

Using CROM to compute lower bounds

Database properties [CFHL21]

- A database property P is a subset of $\mathcal{D}$.
- For example, $P_{pre-image} = \{D | \exists x \in \mathcal{X}, (x, 0) \in D\}$.
- Any database property P can be seen as a projector on $\mathcal{D}$, as follows:

$$\sum_{D \in P} |D\rangle \langle D|$$

Using CROM to compute lower bounds

Database properties [CFHL21]

- A database property P is a subset of $\mathcal{D}$.
- For example, $P_{pre-image} = \{D | \exists x \in \mathcal{X}, (x, 0) \in D\}$.
- Any database property P can be seen as a projector on $\mathcal{D}$, as follows:

$$\sum_{D \in P} |D\rangle \langle D|$$

Computing lower bounds

If we write $|\psi_i\rangle = \mathcal{O} \cdot \mathrm{U_i} \cdot \mathcal{O} \cdot \mathrm{U_{i-1}} \cdot \cdots \cdot \mathcal{O} \cdot \mathrm{U_1} |0\rangle^{\otimes n}$ the state of an algorithm after i queries to the oracle, we want to bound:

$$|P |\psi_i\rangle| \leq f(i)$$

Such that $f(i)$ does not depend of the algorithm.

Sketch of the proof

Lower bound on 2–RSC

Given two random functions $h_1, h_2 : \mathcal{X} \to \mathcal{Y}$ where $|N| = \mathcal{Y}$, a quantum algorithm needs to make $\Omega(N^{3/7})$ queries to h_1 and h_2 to find a 2–RSC with a constant probability.

Sketch of the proof

Lower bound on 2–RSC

Given two random functions $h_1, h_2 : \mathcal{X} \to \mathcal{Y}$ where $|N| = \mathcal{Y}$, a quantum algorithm needs to make $\Omega(N^{3/7})$ queries to h_1 and h_2 to find a 2–RSC with a constant probability.

Database properties

- P_2 as the set of databases that contain a 2-RSC.

Sketch of the proof

Lower bound on 2–RSC

Given two random functions $h_1, h_2 : \mathcal{X} \to \mathcal{Y}$ where $|N| = \mathcal{Y}$, a quantum algorithm needs to make $\Omega(N^{3/7})$ queries to h_1 and h_2 to find a 2–RSC with a constant probability.

Database properties

- P_2 as the set of databases that contain a 2-RSC.
- P'_ℓ as the set of databases that contain *at least* ℓ *distinct* collisions on h_1.

Sketch of the proof

Lower bound on 2–RSC

Given two random functions $h_1, h_2 : \mathcal{X} \to \mathcal{Y}$ where $|N| = \mathcal{Y}$, a quantum algorithm needs to make $\Omega(N^{3/7})$ queries to h_1 and h_2 to find a 2–RSC with a constant probability.

Database properties

- P_2 as the set of databases that contain a 2-RSC.
- P'_ℓ as the set of databases that contain *at least* ℓ *distinct* collisions on h_1.

Writing $|\psi_i\rangle$ the state after the i^{th} query to $H = (h_1, h_2)$, our goal is to bound $|P_2 |\psi_i\rangle|$.

Sketch of the proof

Claim 1

We have that:

$$\textit{Find with } i \textit{ queries} \leq \textit{Find with } i-1 \textit{ query} + \textit{Find with the } i^{th} \textit{ query.}$$

Sketch of the proof

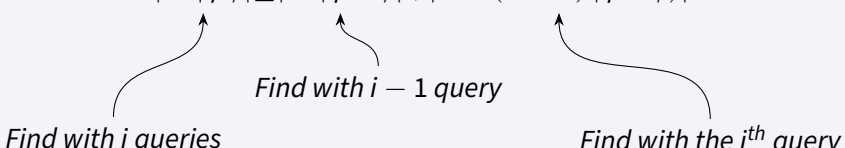
Claim 1

We have that:

$$\textit{Find with i queries} \leq \textit{Find with } i-1 \textit{ query} + \textit{Find with the } i^{th} \textit{ query.}$$

This translates to:

$$|P_2 |\psi_i\rangle| \leq |P_2 |\psi_{i-1}\rangle| + |P_2 \mathcal{O}(I - P_2) |\psi_{i-1}\rangle)|$$



Sketch of the proof

$$|P_2\mathcal{O}(I-P_2)\,|\psi_{i-1}\rangle| = \left|P_2\mathcal{O}\sum_{\substack{x,y\\ D\text{: no 2–RSC}}}\alpha_{x,y,D}\,|x,y,D\rangle\right|$$

Sketch of the proof

$$|P_2\mathcal{O}(I-P_2)\,|\psi_{i-1}\rangle| = \left|P_2\mathcal{O}\sum_{\substack{x,y\\ D\text{: no 2–RSC}}}\alpha_{x,y,D}\,|x,y,D\rangle\right|$$

$$= \left|P_2\sum_{\substack{x,y\\ D\text{: no 2–RSC}}}\alpha_{x,y,D}\,|x,y,D\oplus(x,y)\rangle\right|$$

Sketch of the proof

$$|P_2\mathcal{O}(I-P_2)\,|\psi_{i-1}\rangle| = \left|P_2\mathcal{O}\sum_{\substack{x,y\\ D\text{: no 2–RSC}}}\alpha_{x,y,D}\,|x,y,D\rangle\right|$$

$$= \left|P_2\sum_{\substack{x,y\\ D\text{: no 2–RSC}}}\alpha_{x,y,D}\,|x,y,D\oplus(x,y)\rangle\right|$$

$$= \left|P_2\sum_{\substack{x,y\\ D\text{: no 2–RSC}}}\frac{1}{\sqrt{N^2}}\sum_{y'}\omega_n^{yy'}\alpha_{x,y,D}\,|x,y,D\cup(x,y')\rangle\right|$$

Sketch of the proof

$$
\begin{aligned}
\left|P_2\mathcal{O}(I-P_2)\left|\psi_{i-1}\right\rangle\right| &= \left|P_2\mathcal{O}\sum_{\substack{x,y\\ D\text{: no 2–RSC}}}\alpha_{x,y,D}\left|x,y,D\right\rangle\right| \\
&= \left|P_2\sum_{\substack{x,y\\ D\text{: no 2–RSC}}}\alpha_{x,y,D}\left|x,y,D\oplus(x,y)\right\rangle\right| \\
&= \left|P_2\sum_{\substack{x,y\\ D\text{: no 2–RSC}}}\frac{1}{\sqrt{N^2}}\sum_{y'}\omega_n^{yy'}\alpha_{x,y,D}\left|x,y,D\cup(x,y')\right\rangle\right| \\
&= \left|\sum_{y'}\sum_{\substack{x,y\\ D\text{: no 2–RSC}}}\frac{\omega_n^{yy'}\alpha_{x,y,D}}{\sqrt{N^2}}\cdot P_2\left|x,y,D\cup(x,y')\right\rangle\right|.
\end{aligned}
$$

Sketch of the proof

2-RSC

Recall that a 2–RSC consist of x_0, x_1, x_2 such that:

- $h_1(x_0) = h_1(x_1)$
- $h_2(x_0) = h_2(x_2)$

Sketch of the proof

2-RSC

Recall that a 2–RSC consist of x_0, x_1, x_2 such that:

- $h_1(x_0) = h_1(x_1)$
- $h_2(x_0) = h_2(x_2)$

We have three possible ways to get from D that does not have a 2–RSC to $D \cup (x, y')$ that has a 2–RSC:

- $x = x_2$ or $x = x_1$
- $x = x_0$

Sketch of the proof

2-RSC

Recall that a 2–RSC consist of x_0, x_1, x_2 such that:

- $h_1(x_0) = h_1(x_1) \leftarrow$ There must be a collision in the database
- $h_2(x_0) = h_2(x_2) \leftarrow$ We need a new collision there

We have three possible ways to get from D that does not have a 2–RSC to $D \cup (x, y')$ that has a 2–RSC:

- $x = x_2$ or $x = x_1$
- $x = x_0$

Sketch of the proof

2-RSC

Recall that a 2–RSC consist of x_0, x_1, x_2 such that:

- $h_1(x_0) = h_1(x_1)$ ← There must be a collision in the database
- $h_2(x_0) = h_2(x_2)$ ← We need a new collision there

We have three possible ways to get from D that does not have a 2–RSC to $D \cup (x, y')$ that has a 2–RSC:

- $x = x_2$ or $x = x_1$
- $x = x_0$

If there are ℓ collisions on h_1, there are ℓ values of y' such that $D \cup (x, y')$ has a 2–RSC.

$$2 \cdot \sum_{\ell \geq 0} \frac{\ell}{N^2} \left| P'_\ell \left| \psi_{i-1} \right\rangle \right|$$

Sketch of the proof

2-RSC

Recall that a 2–RSC consist of x_0, x_1, x_2 such that:

- $h_1(x_0) = h_1(x_1)$ ← We need a new collision there
- $h_2(x_0) = h_2(x_2)$ ← We need a new collision there

We have three possible ways to get from D that does not have a 2–RSC to $D \cup (x, y')$ that has a 2–RSC:

- $x = x_2$ or $x = x_1$
- $x = x_0$

Sketch of the proof

2-RSC

Recall that a 2–RSC consist of x_0, x_1, x_2 such that:

- $h_1(x_0) = h_1(x_1)$ ← We need a new collision there
- $h_2(x_0) = h_2(x_2)$ ← We need a new collision there

We have three possible ways to get from D that does not have a 2–RSC to $D \cup (x, y')$ that has a 2–RSC:

- $x = x_2$ or $x = x_1$
- $x = x_0$

There are $(i-1)^2$ values of y' such that $D \cup (x, y')$ has a 2–RSC.

$$\frac{(i-1)^2}{N^2}$$

Sketch of the proof

Thus,

$$\left|P_2\mathcal{O}(I-P_2)\left|\psi_{i-1}\right\rangle\right| \leq \sqrt{2\cdot\sum_{\ell\geq 0}\frac{\ell}{N}\left|P'_\ell\left|\psi_{i-1}\right\rangle\right|^2} + \frac{i-1}{N}$$

Sketch of the proof

Thus,

$$\left|P_2\mathcal{O}(I-P_2)\left|\psi_{i-1}\right\rangle\right| \leq \sqrt{2\cdot\sum_{\ell\geq 0}\frac{\ell}{N}\left|P'_\ell\left|\psi_{i-1}\right\rangle\right|^2}+\frac{i-1}{N}$$

Giving:

$$\left|P_2\left|\psi_i\right\rangle\right| \leq \left|P_2\left|\psi_{i-1}\right\rangle\right| + \sqrt{2\sum_{\ell\geq 0}\frac{\ell}{N}\left|P'_\ell\left|\psi_{i-1}\right\rangle\right|^2}+\frac{i-1}{N}$$

Sketch of the proof

Thus,

$$\left|P_2\mathcal{O}(I-P_2)\left|\psi_{i-1}\right\rangle\right| \leq \sqrt{2\cdot\sum_{\ell\geq 0}\frac{\ell}{N}\left|P'_\ell\left|\psi_{i-1}\right\rangle\right|^2}+\frac{i-1}{N}$$

Giving:

$$\begin{aligned}\left|P_2\left|\psi_i\right\rangle\right| &\leq \left|P_2\left|\psi_{i-1}\right\rangle\right| + \sqrt{2\sum_{\ell\geq 0}\frac{\ell}{N}\left|P'_\ell\left|\psi_{i-1}\right\rangle\right|^2}+\frac{i-1}{N}\\ &\leq \dots\\ &\leq \sum_{s=0}^{i-1}\left(\sqrt{2\sum_{\ell\geq 0}\frac{\ell}{N}\left|P'_\ell\left|\psi_s\right\rangle\right|^2}+\frac{s}{N}\right).\end{aligned}$$

Sketch of the proof

Lemma 2

For every $i \in \mathbb{N}$, we have that:

$$\left|P_2 \left|\psi_i\right\rangle\right| \leq 4\sqrt{e}\frac{i^{7/4}}{N^{3/4}} + 4\frac{i^2}{N} + O\left(N^{-1/48}\right).$$

Sketch of the proof

Lemma 2

For every $i \in \mathbb{N}$, we have that:

$$|P_2 |\psi_i\rangle| \leq 4\sqrt{e}\frac{i^{7/4}}{N^{3/4}} + 4\frac{i^2}{N} + O(N^{-1/48}).$$

So when $i \ll N^{3/7}$, the right terms goes to 0. Hence if we want $|P_2 |\psi_i\rangle|$ to be constant, we must have $i \gg N^{3/7}$.

Sketch of the proof

Lemma 2

For every $i \in \mathbb{N}$, we have that:

$$|P_2 |\psi_i\rangle| \leq 4\sqrt{e}\frac{i^{7/4}}{N^{3/4}} + 4\frac{i^2}{N} + O(N^{-1/48}).$$

So when $i \ll N^{3/7}$, the right terms goes to 0. Hence if we want $|P_2 |\psi_i\rangle|$ to be constant, we must have $i \gg N^{3/7}$.

Lower bound on 2–RSC

Given two random functions $h_1, h_2 : \mathcal{X} \to \mathcal{Y}$ where $|N| = \mathcal{Y}$, a quantum algorithm needs to make $\Omega(N^{3/7})$ queries to h_1 and h_2 to find a 2–RSC with a constant probability.

Results on (r, k)–SC

Lower bound on $(1, k)$–SC

We prove that $\Omega\left((k!)^{-1/5} \cdot N^{k/5}\right)$ quantum queries to the idealized hash functions are needed to find a $(1, k)$–SC with constant probability.

Results on (r, k)–SC

Lower bound on $(1, k)$–SC

We prove that $\Omega\left((k!)^{-1/5} \cdot N^{k/5}\right)$ quantum queries to the idealized hash functions are needed to find a $(1, k)$–SC with constant probability.

Upper bound on (r, k)–SC

We design a quantum algorithm that finds a (r, k)–SC with constant probability by making $O\left(N^{k/(2+2r)}\right)$ queries to the hash functions.

Results on (r, k)–SC

Lower bound on $(1, k)$–SC

We prove that $\Omega\left((k!)^{-1/5} \cdot N^{k/5}\right)$ quantum queries to the idealized hash functions are needed to find a $(1, k)$–SC with constant probability.

Upper bound on (r, k)–SC

We design a quantum algorithm that finds a (r, k)–SC with constant probability by making $O\left(N^{k/(2+2r)}\right)$ queries to the hash functions.

Quantum algorithm for finding a (r, k)–SC

Input: $t \in \mathbb{N}, k' \in \mathbb{N}$

Results on (r, k)–SC

Lower bound on $(1, k)$–SC

We prove that $\Omega\left((k!)^{-1/5} \cdot N^{k/5}\right)$ quantum queries to the idealized hash functions are needed to find a $(1, k)$–SC with constant probability.

Upper bound on (r, k)–SC

We design a quantum algorithm that finds a (r, k)–SC with constant probability by making $O\left(N^{k/(2+2r)}\right)$ queries to the hash functions.

Quantum algorithm for finding a (r, k)–SC

Input: $t \in \mathbb{N}, k' \in \mathbb{N}$

1. Apply the algorithm recursively to find t distinct $(r-1, k')$–SC.

Results on (r, k)–SC

Lower bound on $(1, k)$–SC

We prove that $\Omega\left((k!)^{-1/5} \cdot N^{k/5}\right)$ quantum queries to the idealized hash functions are needed to find a $(1, k)$–SC with constant probability.

Upper bound on (r, k)–SC

We design a quantum algorithm that finds a (r, k)–SC with constant probability by making $O\left(N^{k/(2+2r)}\right)$ queries to the hash functions.

Quantum algorithm for finding a (r, k)–SC

Input: $t \in \mathbb{N}, k' \in \mathbb{N}$

1. Apply the algorithm recursively to find t distinct $(r-1, k')$–SC.
2. Apply Grover's algorithm to find a (r, k)–SC.

Summary

Problem	Lower bound	Upper bound	Tight?
k-RSC	$\Omega\left(N^{\frac{2^k-1}{2^{k+1}-1}}\right)$	$O\left(N^{\frac{2^k-1}{2^{k+1}-1}}\right)$	Yes
(r,k)-SC	$\Omega\left(N^{k/(2+2r)}\right)$	$O\left(N^{k/(2+2r)}\right)$	Yes

- State of the art [YTA22]
- Our contribution
- Concurrent work [YTA23]

On Limits on the Provable Consequences of Quantum Pseudorandomness

Minimal assumption for classical cryptography

One-Way Functions

A function $f : \{0,1\}^n \to \{0,1\}^n$ is a One-Way Function (OWF) if:

1. $f(x)$ can be computed efficiently.
2. Given $y = f(x)$, it is hard to compute x.

Minimal assumption for classical cryptography

One-Way Functions

A function $f : \{0,1\}^n \to \{0,1\}^n$ is a One-Way Function (OWF) if:

1. $f(x)$ can be computed efficiently.
2. Given $y = f(x)$, it is hard to compute x.

One-Way Functions are minimal for cryptography

- Most advanced cryptographic schemes require one-way functions.
- For example, a hash function has to be a one-way function.
- There is nothing (interesting) weaker.

Pseudorandomness

Pseudorandom Number Generator

A function $g : \{0,1\}^n \to \{0,1\}^\ell$ is a Pseudorandom Number Generator (PRNG) if:

1. $g(x)$ can be computed efficiently.
2. $g(x) \approx \mathcal{U}_\ell$, when $x \leftarrow \mathcal{U}_n$.
3. $\ell > n$.

Pseudorandomness

Pseudorandom Number Generator

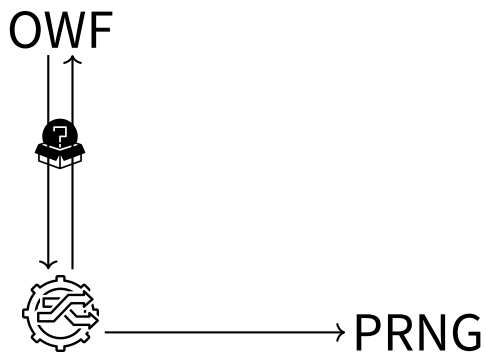
A function $g : \{0,1\}^n \to \{0,1\}^\ell$ is a Pseudorandom Number Generator (PRNG) if:

1. $g(x)$ can be computed efficiently.
2. $g(x) \approx \mathcal{U}_\ell$, when $x \leftarrow \mathcal{U}_n$.
3. $\ell > n$.

Theorem 3 ([HILL99])

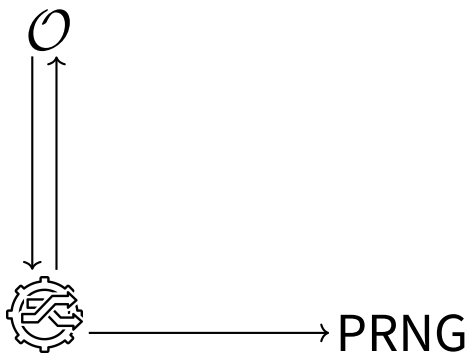
PRNG and OWF are equivalent in a black-box way.

Black-box constructions



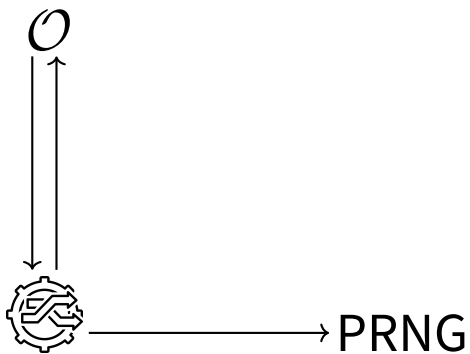
Black-box constructions

$\mathcal{O}$ such that OWF exist



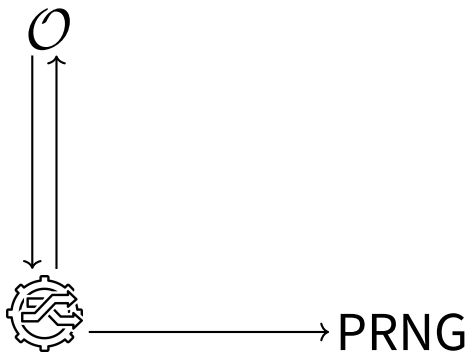
Black-box constructions

$\forall \mathcal{O}$ such that OWF exist



Black-box constructions

$\forall \mathcal{O}$ such that OWF exist



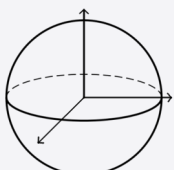
A black-box impossibility construction of $\mathcal{A}$ from $\mathcal{B}$ is an oracle $\mathcal{O}$ such that, relative to $\mathcal{O}$, $\mathcal{B}$ exists but not $\mathcal{A}$.

Quantum Pseudorandomness

Quantum Randomness

We can also consider quantum randomness.

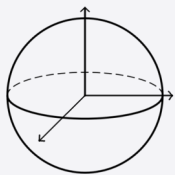
The equivalent to the uniform distribution is the *Haar measure* μ_{2^n}.



Quantum Pseudorandomness

Quantum Randomness

We can also consider quantum randomness.
The equivalent to the uniform distribution is the *Haar measure* μ_{2^n}.



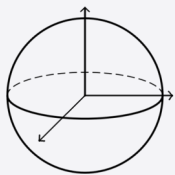
Pseudorandom Quantum States Generators [JLS18]

A function $F : \{0,1\}^\lambda \to \left(\mathbb{C}^2\right)^{\otimes n}$ is a Pseudorandom Quantum State generator (PRS) if:

Quantum Pseudorandomness

Quantum Randomness

We can also consider quantum randomness.
The equivalent to the uniform distribution is the *Haar measure* μ_{2^n}.



Pseudorandom Quantum States Generators [JLS18]

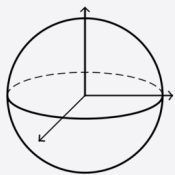
A function $F : \{0,1\}^\lambda \to \left(\mathbb{C}^2\right)^{\otimes n}$ is a Pseudorandom Quantum State generator (PRS) if:

1. $F(k)$ can be computed efficiently.

Quantum Pseudorandomness

Quantum Randomness

We can also consider quantum randomness.
The equivalent to the uniform distribution is the *Haar measure* μ_{2^n}.



Pseudorandom Quantum States Generators [JLS18]

A function $F : \{0,1\}^\lambda \to \left(\mathbb{C}^2\right)^{\otimes n}$ is a Pseudorandom Quantum State generator (PRS) if:

1. $F(k)$ can be computed efficiently.
2. $F(k) \approx \mu_{2^n}$, when $k \leftarrow \mathcal{U}_\lambda$.

Quantum Primitives

Claim 2

We can build Quantum Cryptography from PRSs.

oblivious transfer, multi party computation, public key encryption with quantum keys, quantum one-time digital signatures, pseudo one-time pad encryption schemes, statistically binding and computationally hiding commitments and quantum computational zero knowledge proofs, bit commitments...

Quantum Primitives

Claim 2

We can build Quantum Cryptography from PRSs.

oblivious transfer, multi party computation, public key encryption with quantum keys, quantum one-time digital signatures, pseudo one-time pad encryption schemes, statistically binding and computationally hiding commitments and quantum computational zero knowledge proofs, bit commitments...

Theorem 4 ([JLS18])

$$PRNG \Rightarrow PRS$$

Quantum Primitives

Claim 2

We can build Quantum Cryptography from PRSs.

oblivious transfer, multi party computation, public key encryption with quantum keys, quantum one-time digital signatures, pseudo one-time pad encryption schemes, statistically binding and computationally hiding commitments and quantum computational zero knowledge proofs, bit commitments...

Theorem 4 ([JLS18])

$$PRNG \Rightarrow PRS$$

Theorem 5 ([Kre21])

$$PRS \not\Rightarrow PRNG$$

Quantum Primitives

Claim 2

We can build Quantum Cryptography from PRSs.

oblivious transfer, multi party computation, public key encryption with quantum keys, quantum one-time digital signatures, pseudo one-time pad encryption schemes, statistically binding and computationally hiding commitments and quantum computational zero knowledge proofs, bit commitments...

Theorem 4 ([JLS18])

$$PRNG \Rightarrow PRS$$

Theorem 5 ([Kre21])

$$PRS \not\Rightarrow PRNG$$

There can be quantum pseudorandomness even if

“P = NP”

Different type of Quantum Pseudorandomness

PRU, PRFS, PRS, 1PRS, EFI pairs, OWSG...

Different type of Quantum Pseudorandomness

PRU, PRFS, PRS, 1PRS, EFI pairs, OWSG...

Claim 3

Classically, all these primitives are equivalent.

Different type of Quantum Pseudorandomness

PRU, PRFS, PRS, 1PRS, EFI pairs, OWSG...

Claim 3

Classically, all these primitives are equivalent.

In the quantum setting however...

The Landscape of Quantum Assumptions

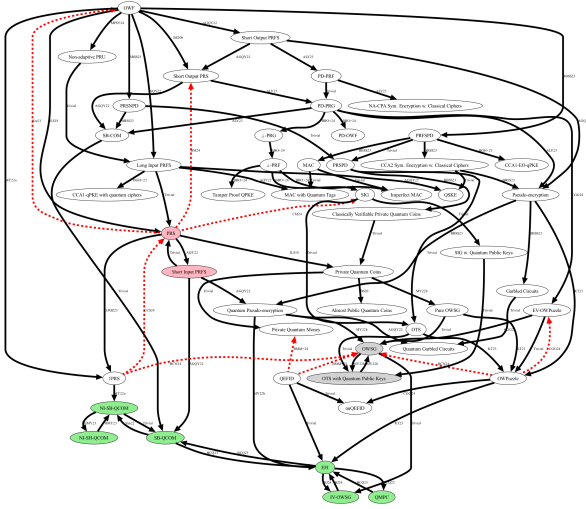
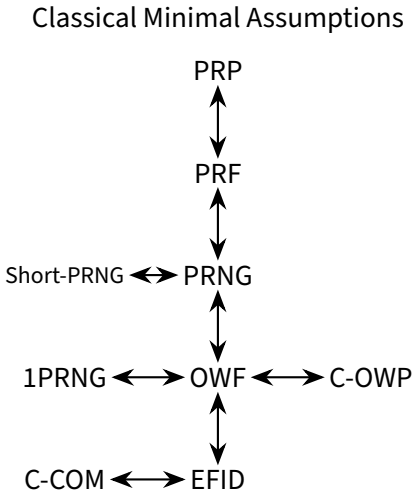
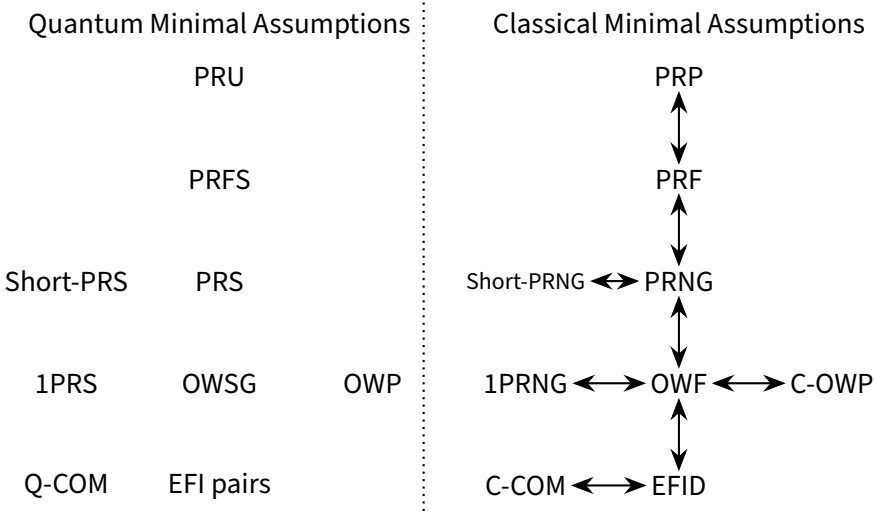


Figure: https://sattath.github.io/microcrypt-zoo/

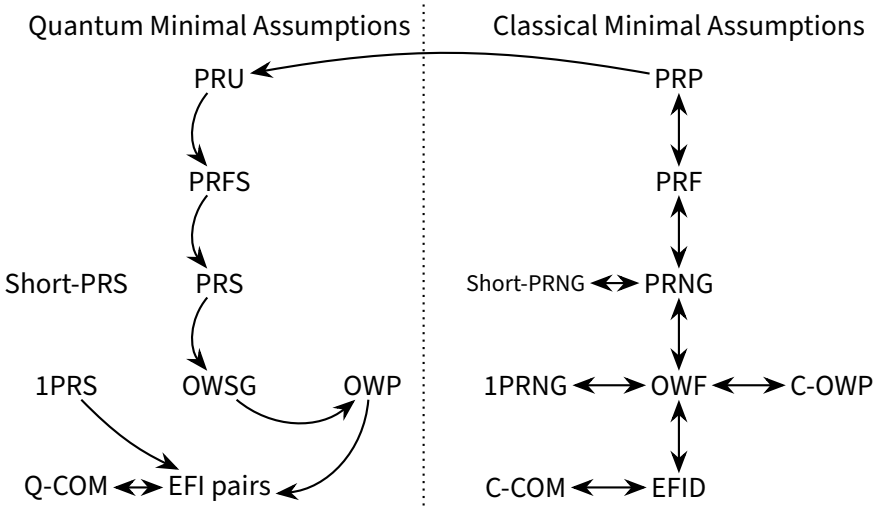
The Landscape of Quantum Assumptions



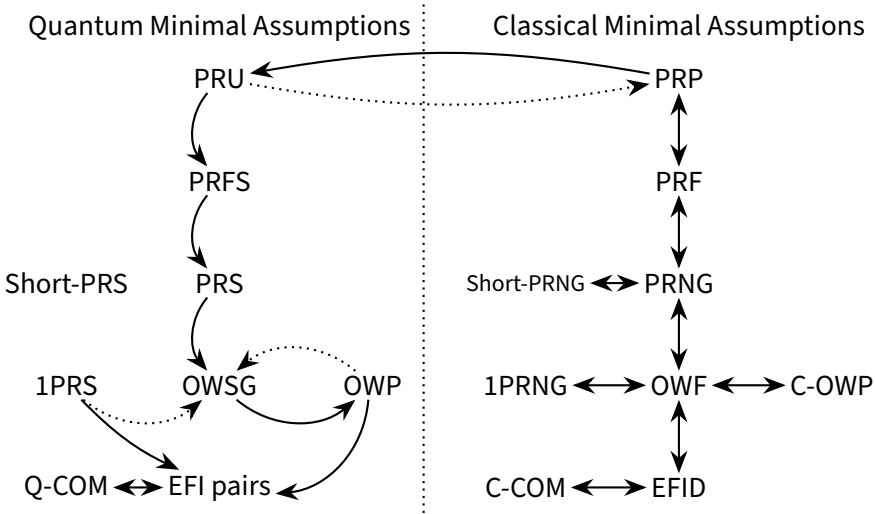
The Landscape of Quantum Assumptions



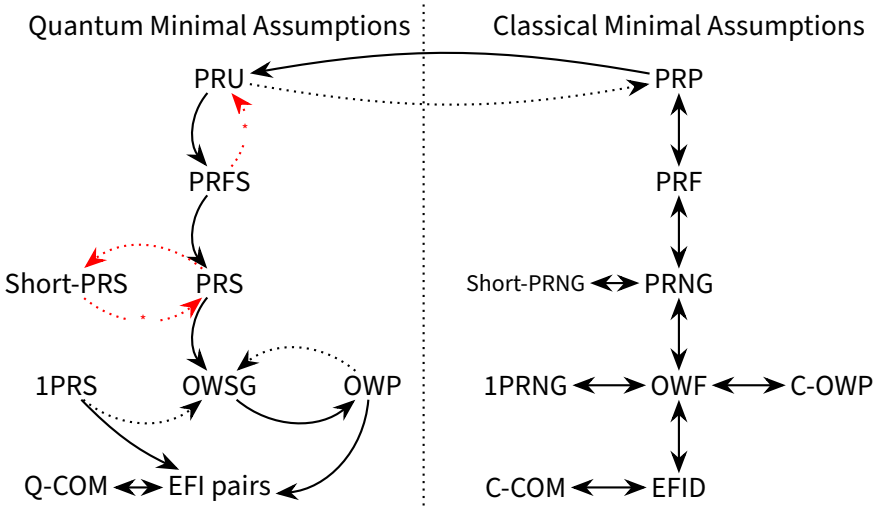
The Landscape of Quantum Assumptions



The Landscape of Quantum Assumptions



The Landscape of Quantum Assumptions



Formal definition of PRSs

Definition 6 (Pseudorandom quantum states [JLS18])

A keyed family of n-qubit quantum states $\{|\varphi_k\rangle\}_{k\in\{0,1\}^\lambda}$ is *pseudorandom* if the following two conditions hold:

Formal definition of PRSs

Definition 6 (Pseudorandom quantum states [JLS18])

A keyed family of n-qubit quantum states $\{|\varphi_k\rangle\}_{k\in\{0,1\}^\lambda}$ is *pseudorandom* if the following two conditions hold:

1. **Efficient generation**. There is a QPT algorithm G such that:

$$G_\lambda(k) = |\varphi_k\rangle\,.$$

Formal definition of PRSs

Definition 6 (Pseudorandom quantum states [JLS18])

A keyed family of n-qubit quantum states $\{|\varphi_k\rangle\}_{k\in\{0,1\}^\lambda}$ is *pseudorandom* if the following two conditions hold:

1. **Efficient generation**. There is a QPT algorithm G such that:

$$G_\lambda(k) = |\varphi_k\rangle\,.$$

2. **Pseudorandomness**. For any QPT adversary $\mathcal{A}$ and all polynomials $t(\cdot)$, we have:

$$\left|\Pr_{k\leftarrow\{0,1\}^\lambda}\left[\mathcal{A}\left(|\varphi_k\rangle^{\otimes t(\lambda)}\right)=1\right]-\Pr_{|\nu\rangle\leftarrow\mu_{2^n}}\left[\mathcal{A}\left(|\nu\rangle^{\otimes t(\lambda)}\right)=1\right]\right|\leq \mathsf{negl}(\lambda).$$

Formal definition of PRSs

Definition 6 (Pseudorandom quantum states [JLS18])

A keyed family of n-qubit quantum states $\{|\varphi_k\rangle\}_{k\in\{0,1\}^\lambda}$ is *pseudorandom* if the following two conditions hold:

1. **Efficient generation**. There is a QPT algorithm G such that:

$$G_\lambda(k) = |\varphi_k\rangle\,.$$

2. **Pseudorandomness**. For any QPT adversary $\mathcal{A}$ and all polynomials $t(\cdot)$, we have:

$$\left|\Pr_{k\leftarrow\{0,1\}^\lambda}\left[\mathcal{A}\left(|\varphi_k\rangle^{\otimes t(\lambda)}\right)=1\right]-\Pr_{|\nu\rangle\leftarrow\mu_{2^n}}\left[\mathcal{A}\left(|\nu\rangle^{\otimes t(\lambda)}\right)=1\right]\right|\leq \mathsf{negl}(\lambda).$$

If $n\approx\lambda$, it is a **long-PRS**, or **PRS**.

Formal definition of PRSs

Definition 6 (Pseudorandom quantum states [JLS18])

A keyed family of n-qubit quantum states $\{|\varphi_k\rangle\}_{k\in\{0,1\}^\lambda}$ is *pseudorandom* if the following two conditions hold:

1. **Efficient generation**. There is a QPT algorithm G such that:

$$G_\lambda(k) = |\varphi_k\rangle\,.$$

2. **Pseudorandomness**. For any QPT adversary $\mathcal{A}$ and all polynomials $t(\cdot)$, we have:

$$\left|\Pr_{k\leftarrow\{0,1\}^\lambda}\left[\mathcal{A}\left(|\varphi_k\rangle^{\otimes t(\lambda)}\right)=1\right]-\Pr_{|\nu\rangle\leftarrow\mu_{2^n}}\left[\mathcal{A}\left(|\nu\rangle^{\otimes t(\lambda)}\right)=1\right]\right|\leq \mathsf{negl}(\lambda).$$

If $n\approx\lambda$, it is a **long-PRS**, or **PRS**.
If $n\approx\log\lambda$, it is a **short-PRS**.

Main result

Theorem 7

Assuming Conjecture 1, there exists an oracle $\mathcal{O}$ relative to which short-PRSs exist, but long-PRSs (with reversible generation algorithm) do not.

Main result

Theorem 7

Assuming Conjecture 1, there exists an oracle $\mathcal{O}$ relative to which short-PRSs exist, but long-PRSs (with reversible generation algorithm) do not.

$$\mathcal{O} = (\quad \mathcal{O}_1 \quad , \quad \mathcal{O}_2 \quad)$$

$\exists$ short-PRS $\qquad$ $\nexists$ long-PRS

Main result

Theorem 7

Assuming Conjecture 1, there exists an oracle $\mathcal{O}$ relative to which short-PRSs exist, but long-PRSs (with reversible generation algorithm) do not.

$$\mathcal{O} = (\ \mathcal{O}_1\ ,\ \mathcal{O}_2\ )$$

$\exists$ short-PRS $\quad$ $\nexists$ long-PRS

Remember: A black-box impossibility construction of $\mathcal{A}$ from $\mathcal{B}$ is an oracle $\mathcal{O}$ such that, relative to $\mathcal{O}$, $\mathcal{B}$ exists but not $\mathcal{A}$.

Common Haar Function-like State Oracle

The Common Haar State Oracle (CHS oracle) with length ℓ is a family of isometries $\{S_x\}_{x\in\{0,1\}^*}$ such that:

Common Haar Function-like State Oracle

The Common Haar State Oracle (CHS oracle) with length ℓ is a family of isometries $\{S_x\}_{x\in\{0,1\}^*}$ such that:

$$S_x : |\alpha\rangle \mapsto |\alpha\rangle\,|\phi_x\rangle$$

where $|\phi_x\rangle$ is a predetermined Haar-random state of length $\ell(|x|)$.

Common Haar Function-like State Oracle

The Common Haar State Oracle (CHS oracle) with length ℓ is a family of isometries $\{S_x\}_{x\in\{0,1\}^*}$ such that:

$$S_x : |\alpha\rangle \mapsto |\alpha\rangle\,|\phi_x\rangle$$

where $|\phi_x\rangle$ is a predetermined Haar-random state of length $\ell(|x|)$.

Claim 4

If $\ell = \log(x)$, then short-PRSs exist relative to $\mathcal{O}_1 =$ the CHS oracles.

Common Haar Function-like State Oracle

The Common Haar State Oracle (CHS oracle) with length ℓ is a family of isometries $\{S_x\}_{x\in\{0,1\}^*}$ such that:

$$S_x : |\alpha\rangle \mapsto |\alpha\rangle\,|\phi_x\rangle$$

where $|\phi_x\rangle$ is a predetermined Haar-random state of length $\ell(|x|)$.

Claim 4

If $\ell = \log(x)$, then short-PRSs exist relative to $\mathcal{O}_1 =$ the CHS oracles.

Now let's rule out long-PRSs!

Breaking PRSs

By contradiction, consider the long-PRS algorithm $\{G_k\}_{k\in\{0,1\}^*}$:

$$G_k : |0\rangle^n \mapsto |\psi_k\rangle = U_T^{(k)} \; S_{x_T^{(k)}} \; \ldots \; U_1^{(k)} \; S_{x_1^{(k)}} \; U_0^{(k)} \; |0\rangle^n$$

Unitary Query Unitary Query Unitary

Breaking PRSs

By contradiction, consider the long-PRS algorithm $\{G_k\}_{k\in\{0,1\}^*}$:

$$G_k : |0\rangle^n \mapsto |\psi_k\rangle = U_T^{(k)} \; S_{x_T^{(k)}} \; \ldots \; U_1^{(k)} \; S_{x_1^{(k)}} \; U_0^{(k)} \; |0\rangle^n$$

Unitary Query Unitary Query Unitary

We assume there is no trace out.

Attack idea

The algorithm

Input ρ: either one of $\{|\psi_k\rangle\}$ or a truly Haar random state.

Attack idea

The algorithm

Input ρ: either one of $\{|\psi_k\rangle\}$ or a truly Haar random state.
For all k:

- We compare ρ with $|\psi_k\rangle$, using swap tests.
- If they are close, output 1.

Output 0.

Attack idea

The algorithm

Input ρ: either one of $\{|\psi_k\rangle\}$ or a truly Haar random state.
For all k:

- We compare ρ with $|\psi_k\rangle$, using swap tests.
- If they are close, output 1.

Output 0.

- $O(2^n)$ operations, but we can chose $\mathcal{O}_2$ to make it possible.

Attack idea

The algorithm

Input ρ: either one of $\{|\psi_k\rangle\}$ or a truly Haar random state.
For all k:

- We compare ρ with $|\psi_k\rangle$, using swap tests.
- If they are close, output 1.

Output 0.

- $O(2^n)$ operations, but we can chose $\mathcal{O}_2$ to make it possible.
- However, we want $\mathcal{O}_2$ not to break short-PRS! To do so, we make it independent of $\mathcal{O}_1$.

Attack idea

The algorithm

Input ρ: either one of $\{|\psi_k\rangle\}$ or a truly Haar random state.
For all k:

- We compare ρ with $|\psi_k\rangle$, using swap tests.
- If they are close, output 1.

Output 0.

- $O(2^n)$ operations, but we can chose $\mathcal{O}_2$ to make it possible.
- However, we want $\mathcal{O}_2$ not to break short-PRS! To do so, we make it independent of $\mathcal{O}_1$.
- We will approximate G_k without querying $\mathcal{O}_1$, and $\mathcal{O}_2$ will be independent of $\mathcal{O}_1$.

Uncomputing the Generation

Claim 5

Because the output is pure, we can write G_k as:

$$G_k : |0\rangle^n \mapsto |\psi_k\rangle = U_T \; S_{x_T} \; \cdots \; U_1 \; S_{x_1} \; U_0 \; |0\rangle^n$$

$$= U_T \; U_{T-1} \; \cdots \; U_1 \; U_0 \; |0\rangle^n \; |\phi_{x_1}\rangle \; \cdots \; |\phi_{x_T}\rangle$$

Uncomputing the Generation

Claim 5

Because the output is pure, we can write G_k as:

$$G_k : |0\rangle^n \mapsto |\psi_k\rangle = U_T \; S_{x_T} \; \cdots \; U_1 \; S_{x_1} \; U_0 \; |0\rangle^n$$

$$= U_T \; U_{T-1} \; \cdots \; U_1 \; U_0 \; |0\rangle^n \; |\phi_{x_1}\rangle \; \cdots \; |\phi_{x_T}\rangle$$

- We can exploit the *structure* of the input state, by uncomputing and applying a product state test.

Uncomputing the Generation

Claim 5

Because the output is pure, we can write G_k as:

$$G_k : |0\rangle^n \mapsto |\psi_k\rangle = U_T \; S_{x_T} \; \cdots \; U_1 \; S_{x_1} \; U_0 \; |0\rangle^n$$

$$= U_T \; U_{T-1} \; \cdots \; U_1 \; U_0 \; |0\rangle^n \; |\phi_{x_1}\rangle \; \cdots \; |\phi_{x_T}\rangle$$

- We can exploit the *structure* of the input state, by uncomputing and applying a product state test.
- Problem: how to learn the x_i?

The Concentration Inequality

Concentration inequality (informal)

Let $\mathcal{X}$ be the product space of pure quantum states with the corresponding product Haar measure σ_N. Suppose that $f : \mathcal{X} \to \mathbb{R}$ is L-Lipschitz in the Frobenius norm. For every $t > 0$, it holds that

$$\Pr_{U \leftarrow \sigma_N} \left[f(U) \geq \mathbb{E}_{V \leftarrow \sigma_N} [f(V)] + t \right] \leq \exp\left(-\frac{(N-2)t^2}{24L^2} \right).$$

Consequence

We cannot extract information from long-PRS.

Our Conjecture

Conjecture 1 (informal)

Let X be the product space of pure quantum states with the corresponding product Haar measure σ. If S_0, S_1 are two measurable subsets of X such that $\sigma(S_0), \sigma(S_1) \geq A$, and if $d(S_0, S_1) \geq B$ for some distance d on X, then $\sigma(X \setminus (S_0 \cup S_1)) \geq \text{poly}(A, B)$.

Consequence

We cannot extract information from short-PRS with negligible probability.

Our Conjecture

Conjecture 1 (informal)

Let X be the product space of pure quantum states with the corresponding product Haar measure σ. If S_0, S_1 are two measurable subsets of X such that $\sigma(S_0), \sigma(S_1) \geq A$, and if $d(S_0, S_1) \geq B$ for some distance d on X, then $\sigma(X \setminus (S_0 \cup S_1)) \geq \text{poly}(A, B)$.

Consequence

We cannot extract information from short-PRS with negligible probability.

Note: it as been shown that we can extract information with $\frac{1}{poly(\cdot)}$ error.

Our Conjecture

Conjecture 1 (informal)

Let X be the product space of pure quantum states with the corresponding product Haar measure σ. If S_0, S_1 are two measurable subsets of X such that $\sigma(S_0), \sigma(S_1) \geq A$, and if $d(S_0, S_1) \geq B$ for some distance d on X, then $\sigma(X \setminus (S_0 \cup S_1)) \geq \text{poly}(A, B)$.

Consequence

We cannot extract information from short-PRS with negligible probability.

Note: it as been shown that we can extract information with $\frac{1}{poly(\cdot)}$ error.
This allows us to learn the x_i without querying $\mathcal{O}_1$.

Breaking long-PRS

Input ρ: either one of $\{|\psi_k\rangle\}$ or a truly Haar random state.

Breaking long-PRS

Input ρ: either one of $\{|\psi_k\rangle\}$ or a truly Haar random state.
Informally: we apply the uncomputation $G_k^\dagger$ and test if it is a product state, for all k.

Breaking long-PRS

Input ρ: either one of $\{|\psi_k\rangle\}$ or a truly Haar random state.
Informally: we apply the uncomputation $G_k^\dagger$ and test if it is a product state, for all k.

The attack

Prepares $\Phi = (|\rho\rangle \otimes |\rho\rangle)^{\otimes M}$ for some large M and do for all k:

P_k: Apply $(G_k^\dagger \otimes G_k^\dagger)^{\otimes M}$, apply M product-tests on each copy; if they pass, return 1. Otherwise, return 0.

Breaking long-PRS

Input ρ: either one of $\{|\psi_k\rangle\}$ or a truly Haar random state.
Informally: we apply the uncomputation $G_k^\dagger$ and test if it is a product state, for all k.

The attack

Prepares $\Phi = (|\rho\rangle \otimes |\rho\rangle)^{\otimes M}$ for some large M and do for all k:

P_k: Apply $(G_k^\dagger \otimes G_k^\dagger)^{\otimes M}$, apply M product-tests on each copy; if they pass, return 1. Otherwise, return 0.

Analysis

- if $\rho = |\psi_k\rangle$, P_k returns 1 w.h.p.,

Breaking long-PRS

Input ρ: either one of $\{|\psi_k\rangle\}$ or a truly Haar random state.
Informally: we apply the uncomputation $G_k^\dagger$ and test if it is a product state, for all k.

The attack

Prepares $\Phi = (|\rho\rangle \otimes |\rho\rangle)^{\otimes M}$ for some large M and do for all k:

P_k: Apply $(G_k^\dagger \otimes G_k^\dagger)^{\otimes M}$, apply M product-tests on each copy; if they pass, return 1. Otherwise, return 0.

Analysis

- if $\rho = |\psi_k\rangle$, P_k returns 1 w.h.p.,
- if ρ is a Haar random state, P_k returns 0 w.h.p..

Breaking long-PRS

Input ρ: either one of $\{|\psi_k\rangle\}$ or a truly Haar random state.
Informally: we apply the uncomputation $G_k^\dagger$ and test if it is a product state, for all k.

The attack

Prepares $\Phi = (|\rho\rangle \otimes |\rho\rangle)^{\otimes M}$ for some large M and do for all k:

P_k: Apply $(G_k^\dagger \otimes G_k^\dagger)^{\otimes M}$, apply M product-tests on each copy; if they pass, return 1. Otherwise, return 0.

Analysis

- if $\rho = |\psi_k\rangle$, P_k returns 1 w.h.p.,
- if ρ is a Haar random state, P_k returns 0 w.h.p..

Inefficient, but we can chose $\mathcal{O}_2 =$ QPSPACE! (Quantum OR Lemma)

Breaking long-PRS

Input ρ: either one of $\{|\psi_k\rangle\}$ or a truly Haar random state.
Informally: we apply the uncomputation $G_k^\dagger$ and test if it is a product state, for all k.

The attack

Prepares $\Phi = (|\rho\rangle \otimes |\rho\rangle)^{\otimes M}$ for some large M and do for all k:

P_k: Apply $(G_k^\dagger \otimes G_k^\dagger)^{\otimes M}$, apply M product-tests on each copy; if they pass, return 1. Otherwise, return 0.

Analysis

- if $\rho = |\psi_k\rangle$, P_k returns 1 w.h.p.,
- if ρ is a Haar random state, P_k returns 0 w.h.p..

Inefficient, but we can chose $\mathcal{O}_2 =$ QPSPACE! (Quantum OR Lemma)
Relative to $\mathcal{O} = (\mathcal{O}_1, \mathcal{O}_2)$ =(CHS,QPSPACE), we have short-PRSs but not long-PRSs!

Summary

Theorem 8 ([BHMV25])

Assuming Conjecture 1,

$$\exists \textit{short-PRS} \not\Rightarrow \exists\ \textit{long-PRS}$$

(reversible generation)

Summary

Theorem 8 ([BHMV25])

Assuming Conjecture 1,

$$\exists \textit{short-PRS} \not\Rightarrow \exists\ \textit{long-PRS}$$

(reversible generation)

Theorem 9 ([BM24])

$$\exists\ \textit{long-PRS} \not\Rightarrow \exists\ \textit{short-PRS}$$

Summary

Theorem 8 ([BHMV25])

Assuming Conjecture 1,

$$\exists \textit{short-PRS} \not\Rightarrow \exists \textit{ long-PRS}$$

(reversible generation)

Theorem 9 ([BM24])

$$\exists \textit{ long-PRS} \not\Rightarrow \exists \textit{ short-PRS}$$

Theorem 10

$$\exists \textit{PRU} \not\Rightarrow \exists \textit{ (some) qPKE}$$

Conclusion

On subset cover

Results

- We study problems in the Quantum Random Oracle Model that are linked to post-quantum schemes, namely SPHINCS and SPHINCS+.

On subset cover

Results

- We study problems in the Quantum Random Oracle Model that are linked to post-quantum schemes, namely SPHINCS and SPHINCS+.

Perspectives

- However, it remains to prove the security of these schemes against quantum adversaries.

On black-box proofs

Results

- We study the relationship between quantum cryptography primitives, namely Pseudorandom Quantum States (PRSs) and its variants.

On black-box proofs

Results

- We study the relationship between quantum cryptography primitives, namely Pseudorandom Quantum States (PRSs) and its variants.
- We also study the construction of quantum public key encryption from weaker primitives, such as One-Way Functions and PRSs.

On black-box proofs

Results

- We study the relationship between quantum cryptography primitives, namely Pseudorandom Quantum States (PRSs) and its variants.
- We also study the construction of quantum public key encryption from weaker primitives, such as One-Way Functions and PRSs.

Perspectives

- Which primitives will be relevant?
- What will the full picture look like?

On black-box proofs

Results

- We study the relationship between quantum cryptography primitives, namely Pseudorandom Quantum States (PRSs) and its variants.
- We also study the construction of quantum public key encryption from weaker primitives, such as One-Way Functions and PRSs.

Perspectives

- Which primitives will be relevant?
- What will the full picture look like?

Thank you for your attention.

Bibliography


Charles Bennett and Gilles Brassard.
Quantum cryptography: Public key distribution and coin tossing.
volume 560, pages 175–179, 01 1984.


Samuel Bouaziz-Ermann, Alex B. Grilo, and Damien Vergnaud.
Quantum security of subset cover problems.
In Kai-Min Chung, editor, *ITC 2023*, volume 267 of *LIPIcs*, pages 9:1–9:17. Schloss Dagstuhl, June 2023.


Samuel Bouaziz-Ermann, Alex B. Grilo, Damien Vergnaud, and Quoc-Huy Vu.
Towards the impossibility of quantum public key encryption with classical keys from one-way functions.
CiC, 1(1):32, 2024.


Samuel Bouaziz–Ermann, Minki Hhan, Garazi Muguruza, and Quoc-Huy Vu.
On limits on the provable consequences of quantum pseudorandomness.
2025.


Samuel Bouaziz–Ermann and Garazi Muguruza.
Quantum pseudorandomness cannot be shrunk in a black-box way.
arXiv preprint arXiv:2402.13324, 2024.


Kai-Min Chung, Serge Fehr, Yu-Hsuan Huang, and Tai-Ning Liao.

The tools

Lemma 11 (Quantum OR lemma)

Let $\{\Pi_i\}_{i\in[N]}$ be POVMs. Let $0 < \varepsilon < 1/2$ and $\delta > 0$. Let Ψ be a quantum state such that either

1. *there exists $i \in [N]$ such that $\mathrm{Tr}[\Pi_i\Psi] \geq 1-\varepsilon$, or*
2. *for all $i \in [N]$, $\mathrm{Tr}[\Pi_i\Psi] \leq \delta$.*

Then, there is a quantum circuit C, such that in case i)

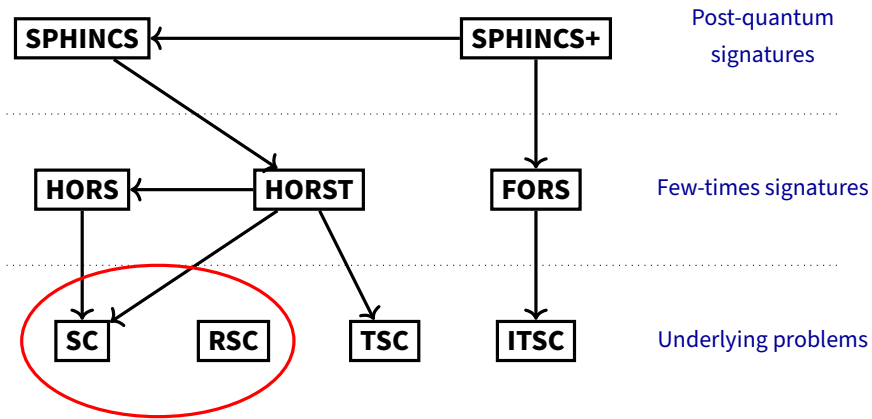
$$\Pr(1 \leftarrow C(\Psi)) \geq \frac{(1-\varepsilon)^2}{7},$$

and in case ii),

$$\Pr(1 \leftarrow C(\Psi)) \leq 4N\delta.$$

The circuit C can be implemented in QPSPACE.

Motivation


SPHINCS
SPHINCS+
Post-quantum signatures
HORS
HORST
FORS
Few-times signatures
SC
RSC
TSC
ITSC
Underlying problems

Our algorithm

Algorithm for finding a (r, k)–SC

Input: $t \in \mathbb{N}, k' \in \mathbb{N}$.

1. Execute the $(r-1, k')$–SC algorithm t times to find t distinct $(r-1, k')$–SC in $h_1, \ldots, h_{k'}$.
   Let T be the set of these $(r-1, k')$–SC.
2. Define $F : \mathcal{X} \to \{0, 1\}$ as follows:

$$F(x) = \begin{cases} 1, & \text{if there exists } (x_0, x_1, \ldots, x_{r-1}) \in T \text{ such that} \\ & \forall 1 \leq m \leq k - k', h_m(x) = h_{k'+m}(x_0), \\ 0, & \text{otherwise.} \end{cases}$$

3. Execute Grover's algorithm to find an x such that $F(x) = 1$
4. Find $(x_0, x_1, \ldots, x_{r-1})$ in T and output $(x_0, x_1, \ldots, x_{r-1}, x)$.