HOW TO CONSTRUCT QUANTUM RANDOM FUNCTIONS

Mark Zhandry – Stanford University
(Classical) Pseudorandom Functions

[GGM’84]

Adversary

Choose random bit $b$

$F(x)$

$q$ queries

$F$

$F(x)$

$\gamma^x$

PRF

$F$

$b = 0$

$b = 1$

PRF is secure if $\left| \Pr[b = b'] - \frac{1}{2} \right| < \text{negl}$
(Classical) Pseudorandom Functions

[A•GGM’84]

Adversary

Choose random bit $b$

Choose $F(x)$

$q$ queries

PRF

$F(x)$

$\forall x$

PRF is secure if $\left| \Pr [b = b'] - \frac{1}{2} \right| < \text{negl}$
Quantum Pseudorandom Functions

Choose random bit $b$

$F$

$\sum_x \alpha_x |x, F(x)\rangle$

$q$ queries

Single query evaluates $F$ on exponentially-many inputs
Quantum Pseudorandom Functions

PRFs: building block for most of symmetric crypto
Quantum PRFs: may be needed when end-users are quantum

Specific applications:

• Proofs in the Quantum Random Oracle Model [BDFLSZ’11]
• Needed for MACs secure against quantum chosen message attacks [BZ’12]
• Step towards quantum PRP (e.g. Luby-Rackoff)
Theorem: If PRFs exist, then there are PRFs that are not quantum PRFs

- Construct a PRF that is periodic with large, secret period
- Cannot find period with classical queries
- Easy with quantum queries
How to Construct Quantum PRFs

We prove security for some classical PRF constructions:

- From quantum-secure pseudorandom generators [GGM'84]
- From quantum-secure pseudorandom synthesizers [NR'95]
- Directly from lattices [BPR’11]

Classical proofs do not carry over into the quantum setting

⇒ Need new proof techniques

Example: GGM
Pseudorandom Generators

\[ S \rightarrow G \rightarrow G_0(s) \quad \approx_{QP} \quad G_1(s) \rightarrow y \]

Indistinguishable for Quantum Machines
The GGM Construction
Original Security Proof

Step 1: Hybridize over levels of tree
Original Security Proof: Step 1

Hybrid 0
Original Security Proof: Step 1

Hybrid 1
Original Security Proof: Step 1

Hybrid 2
Original Security Proof: Step 1

Hybrid 3
Original Security Proof: Step 1

Hybrid n
Original Security Proof: Step 1

PRF distinguisher will distinguish two adjacent hybrids
Original Security Proof: Step 1

PRF distinguisher will distinguish two adjacent hybrids
Original Security Proof

Step 1: Hybridize over levels of tree

Step 2: Simulate hybrids using q samples
Original Security Proof: Step 2

Simulate
Original Security Proof: Step 2

Simulate

Put samples here

yyy yyy yyy yyy yyy yyy yyy

SSSSSSSSSSSSSSSSSSSSSSSSSSS
Original Security Proof: Step 2

Rows are exponentially wide

Problem?
Active node: value used to answer query

Only need to fill active nodes

Adversary only queries polynomial number of points
Original Security Proof

Step 1: Hybridize over levels of tree ✓

Step 2: Simulate hybrids using q samples ✓

Step 3: Pseudorandomness of one PRG sample implies pseudorandomness of q samples
Original Security Proof: Step 3

$S \approx_{QP} \gamma$

$\approx_{QP}$
Original Security Proof

Step 1: Hybridize over levels of tree ✓

Step 2: Simulate hybrids using q samples ✓

Step 3: Pseudorandomness of one PRG sample implies pseudorandomness of q samples ✓
Quantum Security Proof Attempt

Step 1: Hybridize over levels of tree
Quantum Security Proof Attempt

Step 1: Hybridize over levels of tree

Step 3: Quantum pseudorandomness of one PRG sample implies quantum pseudorandomness of q samples
Quantum Security Proof Attempt

Step 1: Hybridize over levels of tree ✓

Step 2: Simulate hybrids using q samples X

Step 3: Quantum pseudorandomness of one PRG sample implies quantum pseudorandomness of q samples ✓
Difficulty Simulating Hybrids

Adversary can query on all exponentially-many inputs
Difficulty Simulating Hybrids

All nodes are active!

Exact simulation requires exponentially-many samples

Need new simulation technique
A Distribution to Simulate

Any distribution $D$ on values induces a distribution on functions

For all $x \in \mathcal{X}$

$$y_x \leftarrow D$$

$$H(x) = y_x$$

$D$ $D$ $D$ $D$ $D$ $D$ $D$ $D$ $D$ $D$ $D$ $D$ $D$ $D$ $D$

$H$:  

$$D^\mathcal{X}$$
Main Tool: Small Range Distributions

\[ (y_1, \ldots, y_r) \leftarrow D^r \]

For all \( x \in \mathcal{X} \)

\[ i_x \leftarrow [1, r] \]

\[ H(x) = y_{i_x} \]

\[ \text{SR}_{r}^\mathcal{X}(D) \]
Main Technical Lemma

Lemma: $\text{SR}_r^x(D)$ is indistinguishable from $D^x$ by any $q$-query quantum algorithm, except with probability $O(q^3/r)$.
Lemma: $\text{SR}_r^X(D)$ is indistinguishable from $D^X$ by any $q$-query quantum algorithm, except with probability $O(q^3/r)$.
Fixing the GGM Proof

PRF distinguisher will distinguish two adjacent hybrids.
Fixing the GGM Proof

PRF distinguisher will distinguish two adjacent hybrids
Fixing the GGM Proof

PRF distinguisher will distinguish two adjacent hybrids.
Quantum Security Proof

Step 1: Hybridize over levels of tree

Step 2: Simulate hybrids \textit{approximately} using \textit{polynomially-many} samples

Step 3: Quantum pseudorandomness of one sample implies quantum pseudorandomness of \textit{polynomially-many} samples
Summary

Separation: PRFs $\neq$ QPRFs

We prove security for some classical PRF constructions:

- From quantum-secure pseudorandom generators [GGM’84]
- From quantum-secure pseudorandom synthesizers [NR’95]
- Directly from lattices [BPR’11]
Future Work

Quantum secure PRPs

Other crypto primitives:
- Signatures and MACs under quantum chosen message attacks
- Encryption secure under quantum chosen ciphertext attacks
Future Work

Quantum secure PRPs

Other crypto primitives:

- Signatures and MACs under quantum chosen message attacks
- Encryption secure under quantum chosen ciphertext attacks

Thank you!