Untrusted quantum devices

Yaoyun Shi
University of Michigan
updated Feb. 1, 2014
Quantum power is incredible!
Quantum power is incredible!
Quantum power is incredible!

Spooky action at a distance
Quantum power is incredible!

Simulating quantum physics

Spooky action at a distance
Quantum power is incredible!

- simulating quantum physics
- super-fast factoring
- Spooky action at a distance
Quantum power is incredible!

- Super-fast factoring
- Simulating quantum physics
- Spooky action at a distance
- Unbreakable codes
- Super-fast factoring
Quantum power is incredible!

simulating quantum physics

super-fast factoring

Spooky action at a distance

Unbreakable codes

Bla...bla...bla...
BEING SIMULTANEOUSLY DEAD AND ALIVE IN THE BOX GAVE ME AN INCREDIBLE PERSPECTIVE OVER THE "LIFE, THE UNIVERSE AND EVERYTHING". AND I AM HERE TO TELL IT TO THE WORLD!
Do you believe in half-dead half-alive cats?

BEING SIMULTANEOUSLY DEAD AND ALIVE IN THE BOX GAVE ME AN INCREDIBLE PERSPECTIVE OVER THE "LIFE, THE UNIVERSE AND EVERYTHING". AND I AM HERE TO TELL IT TO THE WORLD!
Do you believe in half-dead half-alive cats?

Quantum Poems, No. 4
by Smita Krishnaswamy

A group of physicist perform a hoax
about a cat half-alive half-dead in a box
The equations so well worked out
And observations to tout
Only mathematicians fall for such jokes!
Quantum power is incredible!
Quantum power is incredible!

- We are classical beings...
Quantum power is incredible!

- We are classical beings...
Quantum power is incredible!

- We are classical beings...

It's THE CAT!
Quantum power is incredible!

- We are classical beings...
- Technological challenges
Quantum power is incredible!

- We are classical beings...
- Technological challenges
Quantum power is incredible!

- We are classical beings...
- Technological challenges
Quantum power is incredible!

- We are classical beings...
- Technological challenges
- Entanglement experiments still have "Loopholes"


Quantum Factorization of 143 on a Dipolar-Coupling Nuclear Magnetic Resonance System

Nanyang Xu¹, Jing Zhu¹,², Dawei Lu¹, Xianyi Zhou¹, Xinhua Peng¹,², and Jianguo Du¹,²
¹Hefei National Laboratory for Physical Sciences at Microscale and Department of Modern Physics, University of Science and Technology of China, Hefei, Anhui 230026, China
²National Laboratory of Quantum Information, University of Science and Technology of China, Hefei, Anhui 230026, China
Quantum power is incredible!

- We are classical beings...
- Technological challenges
  - Entanglement experiments still have "Loopholes"
- Security concerns
Quantum power is incredible!

- We are classical beings...
- Technological challenges
- Entanglement experiments still have "Loopholes"
- Security concerns
Quantum power is incredible!

- We are classical beings...
- Technological challenges
  - Entanglement experiments still have "Loopholes"
- Security concerns

Certified by ???


Quantum Factorization of 143 on a Dipolar-Coupling Nuclear Magnetic Resonance System
Can we still reap the quantum benefits without trusting the device?
Can we still reap the quantum benefits without trusting the device?
Can we still reap the quantum benefits without trusting the device?

- Untrusted-device quantum computation
Can we still reap the quantum benefits without trusting the device?

- Untrusted-device quantum computation
- Untrusted-device randomness extraction
Can we still reap the quantum benefits without trusting the device?

- Untrusted-device quantum computation
- Untrusted-device randomness extraction
- Untrusted-device key distribution
Foundation: Classical Test of a Quantum Duck
Foundation: Classical Test of a Quantum Duck
Interrogate the device classically, check if behaving like the ideal q. device
Foundation: Classical Test of a Quantum Duck

- Interrogate the device classically, check if behaving like the ideal q. device

- Self-testing: if behaving exactly as ideal, then must be ideal
Foundation: Classical Test of a Quantum Duck

- Interrogate the device classically, check if behaving like the ideal q. device

- Self-testing: if behaving exactly as ideal, then must be ideal

- Robust Self-testing: if behaving close to ideal, then must be close to the ideal
Strategy: mixing work and play (test)
Strategy: mixing work and play (test)
Strategy: mixing work and play (test)

- The device can't tell if it's work or test
Strategy: mixing work and play (test)

- The device can’t tell if it’s work or test
- If trying to avoid work, will fail the test
Strategy: mixing work and play (test)

- The device can’t tell if it’s work or test
  - If trying to avoid work, will fail the test
- Two guarantees
Strategy: mixing work and play (test)

- The device can't tell if it's work or test
  - If trying to avoid work, will fail the test
- Two guarantees
  - Completeness. On honest device (thus job well-done), accept w.h.p.
Strategy: mixing work and play (test)

- The device can’t tell if it’s work or test
  - If trying to avoid work, will fail the test
- Two guarantees
  - Completeness. On honest device (thus job well-done), accept w.h.p.
  - Soundness. On any device, almost always reject or accept a well-done job
Strategy: mixing work and play (test)

• The device can't tell if it's work or test
  • If trying to avoid work, will fail the test
• Two guarantees
  • Completeness. On honest device (thus job well-done), accept w.h.p.
  • Soundness. On any device, almost always reject or accept a well-done job
  • Reject a malicious device w.h.p.
How could self-testing be possible?

Do you love me, daughters?

Dear father, we do!

Impossible if no additional constraints on the device
Additional constraint:
Spatial separation

Communication impossible
Additional constraint: Spatial separation

- Device has multiple-parts

Communication impossible
Additional constraint: Spatial separation

- Device has multiple-parts
- Spatially separated so that no communications allowed among parts within the testing time
The CHSH game
The CHSH game

- Each part receives a bit, outputs a bit
The CHSH game

- Each part receives a bit, outputs a bit
- Input bits are uniformly random
The CHSH game

- Each part receives a bit, outputs a bit
- Input bits are uniformly random
- Device wins if \( a \oplus b = x \land y \)
Classical strategies

A

B

x

y

a

b
Classical strategies

A

B

x

y

a

b
Classical strategies

- Each applies a deterministic function on his/her input
Classical strategies

• Each applies a deterministic function on his/her input
• Best classical winning prob: $\leq \frac{3}{4}$
Classical strategies

- Each applies a deterministic function on his/her input
- Best classical winning prob: $\leq \frac{3}{4}$

Bell-type inequality
Classical strategies

- Each applies a deterministic function on his/her input
- Best classical winning prob: \( \leq \frac{3}{4} \)
  - Both output 0 on all inputs achieves \( \frac{3}{4} \)

Bell-type inequality
Quantum strategies
Quantum strategies
Quantum strategies

- Share entanglement before game starts
Quantum strategies

- Share entanglement before game starts
- Each applies a local measurement chosen by the input
Quantum strategies

- Share entanglement before game starts
- Each applies a local measurement chosen by the input
- Best quantum winning prob: $\leq \cos^2\frac{\pi}{8}$
Quantum strategies

- Share entanglement before game starts
- Each applies a local measurement chosen by the input
- Best quantum winning prob: \( \leq \cos^2 \frac{\pi}{8} \)
- Achievable
Quantum strategies

- Share entanglement before game starts
- Each applies a local measurement chosen by the input
- Best quantum winning prob: $\leq \cos^2 \frac{\pi}{8}$
- Achievable

Tsirelson-type inequality
CHSH is a strong self-test
CHSH is a strong self-test
CHSH is a strong self-test

• Self-test [Popescu, Rohrlick 1999]: Any q. strategy achieving $\cos^2 \frac{\pi}{8}$ must be the ideal protocol (up to local isometry)
CHSH is a strong self-test

- **Self-test** [Popescu, Rohlich 1999]: Any q. strategy achieving $\cos^2\frac{\pi}{8}$ must be the ideal protocol (up to local isometry)
- **Robust self-test** [McKague et al., Miller-S, Reichardt et al. 2012]: Any q. strategy achieving close to $\cos^2\frac{\pi}{8}$ must be close to ideal
CHSH is a strong self-test

- **Self-test** [Popescu, Rohlick 1999]: Any q. strategy achieving $\cos^2 \frac{\pi}{8}$ must be the ideal protocol (up to local isometry)
- **Robust self-test** [McKague et al., Miller-S, Reichardt et al. 2012]: Any q. strategy achieving close to $\cos^2 \frac{\pi}{8}$ must be close to ideal
- **Strong (robust) self-test** [Miller-S, Reichardt et al. 2012]: $\epsilon$-close in prob implies $O(\sqrt{\epsilon})$-close to ideal (strongest possible)
Many other robust self-tests
Many other robust self-tests
Many other robust self-tests

- GHZ-game: classical $\frac{3}{4}$ vs quantum 1
Many other robust self-tests

- GHZ-game: classical $\frac{3}{4}$ vs quantum 1
- Binary XOR games: input/output binary, XOR of output bits determines game outcome
Many other robust self-tests

• GHZ-game: classical $\frac{3}{4}$ vs quantum 1
• Binary XOR games: input/output binary, XOR of output bits determines game outcome
• All strong delft-testing binary XOR games characterized in [Miller, Shi 2012]
Theorem [RUV12]
If an average game wins with very close to quantum optimal, a random block of significant length is close to ideal tensor strategy.
Self-testing graph states

[McKague 2013]

Figure 1: A triangular lattice graph

Let us use the shorthand notation $M = n_j = 1^M_x j$, where $x$ is an $n$-bit string, and $M_j$ operates on the $j$-th subsystem. Then the stabilizer group for $|G_i$ is generated by the operators $S_v = X_v Z A_{1v}$.

That is, $S_v$ has $X$ on vertex $v$ and $Z$ on each of the neighbours of $v$. The stabilizers are products of the generators, and have the form $(1^1 2^t) A_t X_t Z A_t$ for some $n$-bit string $t$. In particular, if $T$ is a triangle in $G$ with characteristic vector $\nu$ then the operator $X_\nu Z A_\nu$ is a stabilizer for $|G_i$.

The graph states that we will be considering are triangular cluster states which are graph states where the underlying graph is a triangular lattice such as in figure 1.

Triangular cluster state model - We will make use of the following theorem, due to Mahalla and Perdrix [MP12], which allows us to transform any quantum circuit into a measurement-based quantum computation in a form which we can test.

Theorem 2 (Mahalla and Perdrix [MP12]). Triangular cluster states are universal resources for measurement-based computation based on measurements $X, Z, X \pm Z, p^2$, and the number of vertices in the cluster state is polynomial in the size of the original circuit we wish to perform.

The algorithm for making a measurement-based computation in this model is given in Algorithm 3. The measurement settings for each qubit $M(L, x, a_1, \ldots, a_v)$ $\{X, Z, X \pm Z\}$ and the output $\text{RESULT}(L, x, a_1 \ldots a_n)$ can be computed in polynomial time. An important aspect of this algorithm is that the measurement settings are adaptive, meaning that the measurement setting for vertex $v$ can depend on the outcome of previously applied measurements.

Algorithm 3 $\text{CALCULATE}(L, x)$

Prepare a triangular cluster state of size $n = \text{poly}(|x|)$

for $v = 1$ to $n$
do
Measure vertex $v$ in basis $M(L, x, a_1 \ldots a_v)$ to obtain outcome $a_v$
end for

Output $\text{RESULT}(L, x, a_1 \ldots a_n)$.

In order to put $\text{CALCULATE}(L, x)$ into a form that we can test, we distribute the $n$-qubit cluster state to $n$ provers, one vertex per prover. Honest provers will accept a measurement setting in $\{X, Z, X \pm Z\}$, measure their qubit in the specified basis, and return the result as $\pm 1$. For the dishonest case, we will assume that the $n$ provers are non-communicating and are limited to quantum operations (no "post-quantum" operations allowed.) We put no further restrictions on the provers, and in particular we make no assumptions about the dimension of the Hilbert space of their states.

6 Submission number 133 to STOC 2014: DO NOT DISTRIBUTE!
Self-testing graph states

[McKague 2013]

- Graph states

Figure 1: A triangular lattice graph

Let us use the shorthand notation $M^x_j = N_{n\sum_{j=1}^M x^j}$, where $x$ is an $n$-bit string, and $M_j$ operates on the $j$-th subsystem. Then the stabilizer group for $|G_i\rangle$ is generated by the operators $S_v = X^v Z A^1 v \cdot A^t X^t Z A^t$.

That is, $S_v$ has $X$ on vertex $v$ and $Z$ on each of the neighbours of $v$. The stabilizers are products of the generators, and have the form $(1)^1 2^t \cdot A^t X^t Z A^t$ for some $n$-bit string $t$. In particular, if $T$ is a triangle in $G$ with characteristic vector $\tau$ then the operator $X^\tau Z A^\tau$ is a stabilizer for $|G_i\rangle$.

The graph states that we will be considering are triangular cluster states which are graph states where the underlying graph is a triangular lattice such as in figure 1.

Triangular cluster state model - We will make use of the following theorem, due to Mahalla and Perdrix [MP12], which allows us to transform any quantum circuit into a measurement-based quantum computation in a form which we can test.

Theorem 2 (Mahalla and Perdrix [MP12]). Triangular cluster states are universal resources for measurement-based computation based on measurements $X$, $Z$, $X \pm Z \sqrt{2}$, and the number of vertices in the cluster state is polynomial in the size of the original circuit we wish to perform.

The algorithm for making a measurement-based computation in this model is given in Algorithm 3. The measurement settings for each qubit $M(L, x, a_1, \ldots, a_v)$ and the output $RESULT(L, x, a_1, \ldots, a_n)$ can be computed in polynomial time. An important aspect of this algorithm is that the measurement settings are adaptive, meaning that the measurement setting for vertex $v$ can depend on the outcome of previously applied measurements.

Algorithm 3 CALCULATE($L, x$)

1. Prepare a triangular cluster state of size $n = \text{poly}(|x|)$
2. For $v = 1$ to $n$
   1. Measure vertex $v$ in basis $M(L, x, a_1, \ldots, a_v)$ to obtain outcome $a_v$
3. Output $RESULT(L, x, a_1, \ldots, a_n)$.

In order to put CALCULATE($L, x$) into a form that we can test, we distribute the $n$-qubit cluster state to $n$ provers, one vertex per prover. Honest provers will accept a measurement setting in $\{X, Z, X \pm Z \sqrt{2}\}$, measure their qubit in the specified basis, and return the result as $\pm 1$. For the dishonest case, we will assume that the $n$ provers are non-communicating and are limited to quantum operations (no "post-quantum" operations allowed.) We put no further restrictions on the provers, and in particular we make no assumptions about the dimension of the Hilbert space of their states.
Self-testing graph states

[McKague 2013]

- Graph states
- Each vertex corresponds to a qubit
Self-testing graph states

[McKague 2013]

- Graph states
  - Each vertex corresponds to a qubit
  - May be highly entangled

Figure 1: A triangular lattice graph

Let us use the shorthand notation $M^x_j = \prod_{i=1}^n M^x_j$, where $x$ is an $n$-bit string, and $M^x_j$ operates on the $j$-th subsystem. Then the stabilizer group for $|G_i\rangle$ is generated by the operators $S^v = X^v Z^{A_1^v} \ldots$. (4)

That is, $S^v$ has $X$ on vertex $v$ and $Z$ on each of the neighbours of $v$. The stabilizers are products of the generators, and have the form $(1)^{1,2\ldots t} \cdot A_t X^t Z^{A_t}$ for some $n$-bit string $t$. In particular, if $T$ is a triangle in $G$ with characteristic vector $\tau$ then the operator $X^{\tau} Z^{A_{\tau}}$ is a stabilizer for $|G_i\rangle$.

The graph states that we will be considering are triangular cluster states which are graph states where the underlying graph is a triangular lattice such as in figure 1.

Triangular cluster state model - We will make use of the following theorem, due to Mahalla and Perdrix [MP12], which allows us to transform any quantum circuit into a measurement-based quantum computation in a form which we can test.

**Theorem 2** (Mahalla and Perdrix [MP12]).

Triangular cluster states are universal resources for measurement-based computation based on measurements $X, Z, X \pm Z/p^2$, and the number of vertices in the cluster state is polynomial in the size of the original circuit we wish to perform.

The algorithm for making a measurement-based computation in this model is given in Algorithm 3. The measurement settings for each qubit $M(L, x, a_1, \ldots a_n)$ are chosen from $\{X, Z, X \pm Z/p^2\}$ and the output $\text{RESULT}(L, x, a_1, \ldots a_n)$ can be computed in polynomial time. An important aspect of this algorithm is that the measurement settings are adaptive, meaning that the measurement setting for vertex $v$ can depend on the outcome of previously applied measurements.

**Algorithm 3**

```
CALCULATE(L, x)

Prepare a triangular cluster state of size $n = \text{poly}(|x|)$ for $v = 1$ to $n$ do
  Measure vertex $v$ in basis $M(L, x, a_1, \ldots a_v)$ to obtain outcome $a_v$
end for
Output $\text{RESULT}(L, x, a_1, \ldots a_n)$.
```

In order to put CALCULATE($L, x$) into a form that we can test, we distribute the $n$-qubit cluster state to $n$ provers, one vertex per prover. Honest provers will accept a measurement setting in $\{X, Z, X \pm Z/p^2\}$, measure their qubit in the specified basis, and return the result as $\pm 1$. For the dishonest case, we will assume that the $n$ provers are non-communicating and are limited to quantum operations (no "post-quantum" operations allowed.) We put no further restrictions on the provers, and in particular we make no assumptions about the dimension of the Hilbert space of their states.
Self-testing graph states

[McKague 2013]

- Graph states
  - Each vertex corresponds to a qubit
  - May be highly entangled
- Robust self-testing

Figure 1: A triangular lattice graph

Let us use the shorthand notation

\[ M^x_j = n \sum_{j=1}^M x_j \cdot w, \quad \text{where } x \text{ is an } n \text{-bit string, and } M_j \text{ operates on the } j\text{-th subsystem.} \]

Then the stabilizer group for \(|G_i\rangle\) is generated by the operators

\[ S_v = X_v Z A_1 v. \]

That is, \(S_v\) has \(X\) on vertex \(v\) and \(Z\) on each of the neighbours of \(v\).

The stabilizers are products of the generators, and have the form

\[ (1)^{1 \cdot 2^t} \cdot A_t X_t Z A_t \]

for some \(n\)-bit string \(t\). In particular, if \(T\) is a triangle in \(G\) with characteristic vector \(\tau\) then the operator \(X \tau Z A \tau\) is a stabilizer for \(|G_i\rangle\).

The graph states that we will be considering are triangular cluster states which are graph states where the underlying graph is a triangular lattice such as in figure 1.

Triangular cluster state model - We will make use of the following theorem, due to Mahalla and Perdrix [MP12], which allows us to transform any quantum circuit into a measurement-based quantum computation in a form which we can test.

**Theorem 2** (Mahalla and Perdrix [MP12]). Triangular cluster states are universal resources for measurement-based computation based on measurements \(X, Z, X \pm Z \sqrt{2}\), and the number of vertices in the cluster state is polynomial in the size of the original circuit we wish to perform.

The algorithm for making a measurement-based computation in this model is given in Algorithm 3. The measurement settings for each qubit \(M(L, x, a_1, \ldots, a_v)\) \(\{X, Z, X \pm Z \sqrt{2}\}\) and the output \(\text{RESULT}(L, x, a_1, \ldots, a_n)\) can be computed in polynomial time. An important aspect of this algorithm is that the measurement settings are adaptive, meaning that the measurement setting for vertex \(v\) can depend on the outcome of previously applied measurements.

**Algorithm 3**

\[
\text{CALCULATE}(L, x) \\
\begin{align*}
\text{Prepare a triangular cluster state of size } n = \text{poly}(|x|) & \\
\text{for } v = 1 \text{ to } n & \\
\text{Measure vertex } v \text{ in basis } M(L, x, a_1, \ldots, a_v) & \\
\text{to obtain outcome } a_v & \\
\end{align*}
\]

Output \(\text{RESULT}(L, x, a_1, \ldots, a_n)\).

In order to put \(\text{CALCULATE}(L, x)\) into a form that we can test, we distribute the \(n\)-qubit cluster state to \(n\) provers, one vertex per prover. Honest provers will accept a measurement setting in \(\{X, Z, X \pm Z \sqrt{2}\}\), measure their qubit in the specified basis, and return the result as \(\pm 1\).

For the dishonest case, we will assume that the \(n\) provers are non-communicating and are limited to quantum operations (no "post-quantum" operations allowed.) We put no further restrictions on the provers, and in particular we make no assumptions about the dimension of the Hilbert space of their states.

6 Submission number 133 to STOC 2014: DO NOT DISTRIBUTE!
Self-testing graph states

[McKague 2013]

- Graph states
  - Each vertex corresponds to a qubit
  - May be highly entangled
- Robust self-testing
  - Translated error depends on vertex number linearly.
Toward untrusted-device quantum computation

random bits → deterministic

Randomized algorithm

Quantum resource → restricted q. operations

Quantum algorithm
Toward untrusted-device quantum computation

- Randomized algorithm = deterministic algorithm + random bits

Randomized algorithm → deterministic

Quantum resource → restricted q. operations

Quantum algorithm
Toward untrusted-device quantum computation

- Randomized algorithm = deterministic algorithm + random bits
- Quantum algorithm = quantum resources + restricted operations
Computation by Teleportation

[Gottesman and Chuang 1999]

EPR pairs → Bell-measurements + single-qubit operations → Quantum algorithm
UD quantum computation by Self-testing sequential CHSH
[Reichardt, Unger, Vazirani 2012]
UD quantum computation by Self-testing sequential CHSH
[Reichardt, Unger, Vazirani 2012]

- Mix sequential CHSH test with work for teleportation-based computation

Randomized classical Verifier
UD quantum computation by Self-testing sequential CHSH
[Reichardt, Unger, Vazirani 2012]

- Mix sequential CHSH test with work for teleportation-based computation
- Not following instruction risks failing the test

Randomized classical Verifier
UD quantum computation by Self-testing sequential CHSH
[Reichardt, Unger, Vazirani 2012]

- Mix sequential CHSH test with work for teleportation-based computation
- Not following instruction risks failing the test
- Honest device starts with EPR pairs, applies instructed measurements and single-qubit operations

Randomized classical Verifier
UD quantum computation by
Self-testing graph state
[McKague 2013]

Let us use the shorthand notation
\[ M^{x_j} = n_{j=1}^M x_j, \]
where \( x \) is an \( n \)-bit string, and \( M^j \) operates
on the \( j \)-th subsystem. Then the stabilizer group for
\( |G_i \) is generated by the operators
\[ S^v = X^v Z^{A_1^v} \ldots \]
(4)
That is, \( S^v \) has \( X \) on vertex \( v \) and \( Z \) on each of the neighbours of \( v \).
The stabilizers are products of the generators, and have the form
\[ \prod_{t=1}^t A_t^{X^t Z^{A_t^t}} \]
for some \( n \)-bit string \( t \). In particular, if \( T \) is
a triangle in \( G \) with characteristic vector \( \xi \) then the operator
\[ X^{\xi} Z^{A^{\xi}} \]
is a stabilizer for \( |G_i \).

The graph states that we will be considering are
triangular cluster states
which are graph states
where the underlying graph is a triangular lattice such as in figure 1.

Triangular cluster state model -
We will make use of the following theorem, due to Mahalla
and Perdrix [MP12], which allows us to transform any quantum circuit into a measurement-based
quantum computation in a form which we can test.

Theorem 2 (Mahalla and Perdrix [MP12]).

Triangular cluster states are universal resources for
measurement-based computation based on measurements
\( X, Z, X \pm Z \sqrt{2} \), and the number of vertices in
the cluster state is polynomial in the size of the original circuit we wish to perform.

The algorithm for making a measurement-based computation in this model is given in Algorithm 3.
The measurement settings for each qubit
\[ M(L, x, a_1, \ldots, a_v) \]
\{ \( X, Z, X \pm Z \sqrt{2} \} \]
and the
output \( \text{RESULT}(L, x, a_1, \ldots, a_n) \) can be computed in polynomial time. An important aspect of this
algorithm is that the measurement settings are adaptive, meaning that the measurement setting
for vertex \( v \) can depend on the outcome of previously applied measurements.

Algorithm 3
\[ \text{CALCULATE}(L, x) \]
Prepare a triangular cluster state of size
\[ n = \text{poly}(|x|) \]
for \( v = 1 \) to \( n \) do
Measure vertex \( v \) in basis
\[ M(L, x, a_1, \ldots, a_v) \]
to obtain outcome \( a_v \)
end for
Output \( \text{RESULT}(L, x, a_1, \ldots, a_n) \).

In order to put \( \text{CALCULATE}(L, x) \) into a form that we can test, we distribute the
\( n \)-qubit
cluster state to
\( n \) provers, one vertex per prover. Honest provers will accept a measurement setting
in
\{ \( X, Z, X \pm Z \sqrt{2} \} \]
measure their qubit in the specified basis, and return the result as
\( \pm 1 \). For
the dishonest case, we will assume that the
\( n \) provers are non-communicating and are limited to
quantum operations (no "post-quantum" operations allowed.) We put no further restrictions on
the provers, and in particular we make no assumptions about the dimension of the Hilbert space
of their states.
UD quantum computation by
Self-testing graph state
[McKague 2013]

Each vertex is a device component

Randomized classical Verifier
UD quantum computation by Self-testing graph state

[McKague 2013]

- Each vertex is a device component
- No communications among them

Figure 1: A triangular lattice graph

Let us use the shorthand notation $M^{x} = N_{j=1}^{n} M^{x_{j}}$, where $x$ is an $n$-bit string, and $M^{j}$ operates on the $j$-th subsystem. Then the stabilizer group for $|G_i$ is generated by the operators $S^{v} = X^{v} Z^{A_{1}^{v}}$. (4)

That is, $S^{v}$ has $X$ on vertex $v$ and $Z$ on each of the neighbours of $v$. The stabilizers are products of the generators, and have the form $1 \cdot A^{t}_{t} \cdot X^{t} Z^{A_{t}}$ for some $n$-bit string $t$. In particular, if $T$ is a triangle in $G$ with characteristic vector $\tau$ then the operator $X^{\tau} Z^{A_{\tau}}$ is a stabilizer for $|G_i$.

The graph states that we will be considering are triangular cluster states which are graph states where the underlying graph is a triangular lattice such as in figure 1.

Triangular cluster state model - We will make use of the following theorem, due to Mahalla and Perdrix [MP12], which allows us to transform any quantum circuit into a measurement-based quantum computation in a form which we can test.

Theorem 2 (Mahalla and Perdrix [MP12]).

Triangular cluster states are universal resources for measurement-based computation based on measurements $X, Z, X \pm Z \sqrt{2}$, and the number of vertices in the cluster state is polynomial in the size of the original circuit we wish to perform.

The algorithm for making a measurement-based computation in this model is given in Algorithm 3. The measurement settings for each qubit $M(L, x, a_{1}, \ldots, a_{v_{1}})$ $\in \{X, Z, X \pm Z \sqrt{2}\}$ and the output $RESULT(L, x, a_{1}, \ldots, a_{n})$ can be computed in polynomial time. An important aspect of this algorithm is that the measurement settings are adaptive, meaning that the measurement setting for vertex $v$ can depend on the outcome of previously applied measurements.

Algorithm 3

CALCULATE($L, x$)

Prepare a triangular cluster state of size $n = poly(|x|)$

for $v = 1$ to $n$
do

Measure vertex $v$ in basis $M(L, x, a_{1}, \ldots, a_{v_{1}})$ to obtain outcome $a_{v}$
end for

Output $RESULT(L, x, a_{1}, \ldots, a_{n})$.

In order to put $\text{CALCULATE}(L, x)$ into a form that we can test, we distribute the $n$-qubit cluster state to $n$ provers, one vertex per prover. Honest provers will accept a measurement setting in $\{X, Z, X \pm Z \sqrt{2}\}$, measure their qubit in the specified basis, and return the result as $\pm 1$. For the dishonest case, we will assume that the $n$ provers are non-communicating and are limited to quantum operations (no "post-quantum" operations allowed.) We put no further restrictions on the provers, and in particular we make no assumptions about the dimension of the Hilbert space of their states.
UD quantum computation by Self-testing graph state
[McKague 2013]

- Each vertex is a device component
- No communications among them
- A single message to and from each component

Randomized classical Verifier

Figure 1: A triangular lattice graph

Let us use the shorthand notation $M_x = \sum_{j=1}^{n} M_{x,j} w$, where $x$ is an $n$-bit string, and $M_{j}$ operates on the $j$-th subsystem. Then the stabilizer group for $|G\rangle$ is generated by the operators $S_v = X_v Z_{A_1 v}$.

That is, $S_v$ has $X$ on vertex $v$ and $Z$ on each of its neighbors. The stabilizers are products of the generators, and have the form $(1)^{1} \cdot (A_{t} X_{t} Z_{A_t})$ for some $n$-bit string $t$. In particular, if $T$ is a triangle in $G$ with characteristic vector $\rho$ then the operator $X_{\rho} Z_{A_{\rho}}$ is a stabilizer for $|G\rangle$.

The graph states that we will be considering are triangular cluster states which are graph states where the underlying graph is a triangular lattice such as in figure 1.

Triangular cluster state model - We will make use of the following theorem, due to Mahalla and Perdrix [MP12], which allows us to transform any quantum circuit into a measurement-based quantum computation in a form which we can test.

Theorem 2 (Mahalla and Perdrix [MP12]). Triangular cluster states are universal resources for measurement-based computation based on measurements $X$, $Z$, $X \pm Z \sqrt{2}$, and the number of vertices in the cluster state is polynomial in the size of the original circuit we wish to perform.

The algorithm for making a measurement-based computation in this model is given in Algorithm 3. The measurement settings for each qubit $M(L, x, a_1,...,a_n)$ $\in \{X, Z, X \pm Z \sqrt{2}\}$ and the output $RESULT(L, x, a_1...a_n)$ can be computed in polynomial time. An important aspect of this algorithm is that the measurement settings are adaptive, meaning that the measurement setting for vertex $v$ can depend on the outcome of previously applied measurements.

Algorithm 3 CALCULATE($L, x$)

1. Prepare a triangular cluster state of size $n = \text{poly}(|x|)$ for $v = 1$ to $n$ do
2. Measure vertex $v$ in basis $M(L, x, a_1...a_v)$ to obtain outcome $a_v$
3. end for
4. Output $RESULT(L, x, a_1...a_n)$.
UD quantum computation by Self-testing graph state [McKague 2013]

- Each vertex is a device component
- No communications among them
- A single message to and from each component
- Mix test with work

Randomized classical Verifier

Let us use the shorthand notation $M_x = N_{j=1}^n M_x^j$, where $x$ is an $n$-bit string, and $M_j$ operates on the $j$-th subsystem. Then the stabilizer group for $|G_i$ is generated by the operators $S_v = X_v Z_A$. (4)

That is, $S_v$ has $X$ on vertex $v$ and $Z$ on each of the neighbours of $v$. The stabilizers are products of the generators, and have the form $(1)^t_1 (2)^t_2 \cdots (n)^t_n$ of the form for some $n$-bit string $t$. In particular, if $T$ is a triangle in $G$ with characteristic vector $\tau$ then the operator $X_\tau Z_A^\tau$ is a stabilizer for $|G_i$.

The graph states that we will be considering are triangular cluster states which are graph states where the underlying graph is a triangular lattice such as in figure 1.

Triangular cluster state model -

We will make use of the following theorem, due to Mahalla and Perdrix [MP12], which allows us to transform any quantum circuit into a measurement-based quantum computation in a form which we can test.

**Theorem 2** (Mahalla and Perdrix [MP12]). Triangular cluster states are universal resources for measurement-based computation based on measurements $X, Z, X \pm Z \sqrt{2}$, and the number of vertices in the cluster state is polynomial in the size of the original circuit we wish to perform.

The algorithm for making a measurement-based computation in this model is given in Algorithm 3. The measurement settings for each qubit $M(L, x, a_1, \ldots, a_n)$ is in the form $\{X, Z, X \pm Z \sqrt{2}\}$ and the output $\text{RESULT}(L, x, a_1, \ldots, a_n)$ can be computed in polynomial time. An important aspect of this algorithm is that the measurement settings are adaptive, meaning that the measurement setting for vertex $v$ can depend on the outcome of previously applied measurements.

**Algorithm 3** CALCULATE($L, x$)

1. Prepare a triangular cluster state of size $n = \text{poly}(|x|)$
2. for $v = 1$ to $n$
   - Measure vertex $v$ in basis $M(L, x, a_1, \ldots, a_v)$ to obtain outcome $a_v$
3. Output $\text{RESULT}(L, x, a_1, \ldots, a_n)$.

In order to put CALCULATE($L, x$) into a form that we can test, we distribute the $n$-qubit cluster state to $n$ provers, one vertex per prover. Honest provers will accept a measurement setting in $\{X, Z, X \pm Z \sqrt{2}\}$, measure their qubit in the specified basis, and return the result as $\pm 1$. For the dishonest case, we will assume that the $n$ provers are non-communicating and are limited to quantum operations (no "post-quantum" operations allowed.) We put no further restrictions on the provers, and in particular we make no assumptions about the dimension of the Hilbert space of their states.
UD quantum computation by Self-testing graph state

[McKague 2013]

Each vertex is a device component
- No communications among them
- A single message to and from each component
- Mix test with work
- Honest device stores the graph state, applies instructed measurements

Randomized classical Verifier

Let us use the shorthand notation $M^x = \sum_{j=1}^{N} M^x_j$, where $x$ is an $n$-bit string, and $M^j$ operates on the $j$-th subsystem. Then the stabilizer group for $|G_i\rangle$ is generated by the operators

$$S^v = X^v Z^A_1^v.$$

That is, $S^v$ has $X$ on vertex $v$ and $Z$ on each of the neighbours of $v$. The stabilizers are products of the generators, and have the form $\prod_{t} A_t X^t Z^A_t$ for some $n$-bit string $t$. In particular, if $T$ is a triangle in $G$ with characteristic vector $\xi$ then the operator $X^\xi Z^A_\xi$ is a stabilizer for $|G_i\rangle$.

The graph states that we will be considering are triangular cluster states which are graph states where the underlying graph is a triangular lattice such as in figure 1.

Triangular cluster state model - We will make use of the following theorem, due to Mahalla and Perdrix [MP12], which allows us to transform any quantum circuit into a measurement-based quantum computation in a form which we can test.

Theorem 2 (Mahalla and Perdrix [MP12]). Triangular cluster states are universal resources for measurement-based computation based on measurements $X$, $Z$, $X \pm Z \sqrt{2}$, and the number of vertices in the cluster state is polynomial in the size of the original circuit we wish to perform.

The algorithm for making a measurement-based computation in this model is given in Algorithm 3. The measurement settings for each qubit $M(L, x, a^1, \ldots, a^n)$ and the output $RESULT(L, x, a^1, \ldots, a^n)$ can be computed in polynomial time. An important aspect of this algorithm is that the measurement settings are adaptive, meaning that the measurement setting for vertex $v$ can depend on the outcome of previously applied measurements.

Algorithm 3

1. Prepare a triangular cluster state of size $n = \text{poly}(|x|)$.
2. For $v = 1$ to $n$:
   - Measure vertex $v$ in basis $M(L, x, a^1, \ldots, a^v)$ to obtain outcome $a^v$.
3. Output $RESULT(L, x, a^1, \ldots, a^n)$.

In order to put $\text{CALCULATE}(L, x)$ into a form that we can test, we distribute the $n$-qubit cluster state to $n$ provers, one vertex per prover. Honest provers will accept a measurement setting in $\{X, Z, X \pm Z \sqrt{2}\}$, measure their qubit in the specified basis, and return the result as $\pm 1$. For the dishonest case, we will assume that the $n$ provers are non-communicating and are limited to quantum operations (no "post-quantum" operations allowed.) We put no further restrictions on the provers, and in particular we make no assumptions about the dimension of the Hilbert space of their states.
Randomness is vital
Randomness is vital

- Digital security
Randomness is vital

- Digital security
Randomness is vital

- Digital security
- Randomized algorithms
Randomness is vital

- Digital security
- Randomized algorithms
- Scientific simulations
Randomness is vital

• Digital security
• Randomized algorithms
• Scientific simulations
• Gambling
Wish list for randomness
Wish list for randomness

- High quality
Wish list for randomness

- High quality
Wish list for randomness

- High quality
- Close to uniform
Wish list for randomness

- High quality
- close to uniform
- secure
Wish list for randomness

- High quality
- close to uniform
- secure
- Crypto secure
Wish list for randomness

- High quality
- close to uniform
- secure
- Crypto secure
- Large quantity
Wish list for randomness

- High quality
  - close to uniform
  - secure
  - Crypto secure
- Large quantity
  - 1 trillion bits/day?
Wish list for randomness

- High quality
  - close to uniform
  - secure
  - Crypto secure
- Large quantity
  - 1 trillion bits/day?
- Minimum assumptions
Wish list for randomness

- High quality
- close to uniform
- secure
- Crypto secure
- Large quantity
- 1 trillion bits/day?
- Minimum assumptions
- least amount of trust
Widely used: Linux's random number generator

- system boot time
- current process id
- Deterministic
- Deterministic stretch
- user input
- On-chip circuit

On-chip circuit
The perils of the lack of randomness and blinded trust
The perils of the lack of randomness and blinded trust

- The lack of initial randomness has led to a large number of broken keys
The perils of the lack of randomness and blinded trust

- The lack of initial randomness has led to a large number of broken keys
- Backdoors built-in chips and standards
The perils of the lack of randomness and blinded trust

- The lack of initial randomness has led to a large number of broken keys
- Backdoors built-in chips and standards
- Security based on unproved computational assumptions
The perils of the lack of randomness and blinded trust

- The lack of initial randomness has led to a large number of broken keys
- Backdoors built-in chips and standards
- Security based on unproved computational assumptions
- Vulnerable to advances in algorithms and technology
How can we be sure it’s random?
How can we be sure it's random?

How could fundamentally unpredictable events possible?
We can't be sure
We can’t be sure
... without believing
We can't be sure
... without believing
first of all its existence
How can we generate a large number of provably unconditional secure random bits with minimal assumptions?
Classical theory is inadequate:
Independence requirement for min-entropy sources

Min-entropy source

Must be independent!

deterministic

Min-entropy source
Classical theory is inadequate:
Independence requirement for min-entropy sources

- Classical theory of Randomness Extraction
Classical theory is inadequate: Independence requirement for min-entropy sources

- Min-entropy source
- Must be independent!
- Deterministic
- Min-entropy source

- Classical theory of Randomness Extraction
- Models weak source as min-entropy string
Classical theory is inadequate: Independence requirement for min-entropy sources

- Classical theory of Randomness Extraction
  - Models weak source as min-entropy string
  - Min-entropy $= k$: adversary can’t predict with $> 2^{-k}$ prob

Min-entropy source

Must be independent!

Min-entropy source
Classical theory is inadequate: Independence requirement for min-entropy sources

- Classical theory of Randomness Extraction
  - Models weak source as min-entropy string
  - Min-entropy $= k$: adversary can't predict with $> 2^{-k}$ prob
  - Possible only when having 2 or more independent sources!
Classical theory is inadequate: Independence requirement for min-entropy sources

- Classical theory of Randomness Extraction
  - Models weak source as min-entropy string
  - Min-entropy=k: adversary can't predict with > $2^{-k}$ prob
  - Possible only when having 2 or more independent sources!
  - How to be sure of independence? Impossible!
Can we get around the independence requirement?
Trusting quantum device solves almost all the problems

FEATURES

- True quantum randomness (passes all tests)
- High bit rate of 4Mbits/sec
- Affordable, compact and reliable
- Continuous status check

QUANTIS IS OFFICIALLY CERTIFIED
Trusting quantum device solves almost all the problems

- Postulates of quantum mechanics promise true randomness

**FEATURES**
- True quantum randomness (passes all statistical tests)
- High bit rate of 4Mbits/sec
- Affordable, compact and reliable
- Continuous status check

QUANTIS IS OFFICIALLY CERTIFIED
Trusting quantum device solves almost all the problems

- Postulates of quantum mechanics promise true randomness
- Measuring the $|+\rangle$ state gives 0/1 with equal prob
Trusting quantum device solves almost all the problems

- Postulates of quantum mechanics promise true randomness
- Measuring the $|+\rangle$ state gives 0/1 with equal prob
- Commercial products available

**FEATURES**

- True quantum randomness (passes a)
- High bit rate of 4Mbits/sec
- Affordable, compact and reliable
- Continuous status check

**QUANTIS IS OFFICIALLY CERTIFIED**
Trusting quantum device solves almost all the problems

- Postulates of quantum mechanics promise true randomness
- Measuring the $|+\rangle$ state gives 0/1 with equal prob
- Commercial products available
- Problem: Should we trust

**FEATURES**

- True quantum randomness (passes a
- High bit rate of 4Mbits/sec
- Affordable, compact and reliable
- Continuous status check

QUANTIS IS OFFICIALLY CERTIFIED
Trusting quantum device solves almost all the problems

- Postulates of quantum mechanics promise true randomness
- Measuring the $|+\rangle$ state gives 0/1 with equal prob
- Commercial products available
- Problem: Should we trust the device not malicious?

FEATURES
- True quantum randomness (passes a)
- High bit rate of 4Mbits/sec
- Affordable, compact and reliable
- Continuous status check

QUANTIS IS OFFICIALLY CERTIFIED
Trusting quantum device solves almost all the problems

- Postulates of quantum mechanics promise true randomness
- Measuring the $|+\rangle$ state gives 0/1 with equal prob
- Commercial products available
- **Problem**: Should we trust
  - the device not malicious?
  - the device working properly?
Trusting quantum device solves almost all the problems

• Postulates of quantum mechanics promise true randomness
• Measuring the |+> state gives 0/1 with equal prob
• Commercial products available
• **Problem:** Should we trust
  • the device not malicious?
  • the device working properly?
  • trust the device maker and the certifying agency?
Randomness from untrusted quantum device
Randomness from untrusted quantum device

Randomness Amplification
[Colbeck and Renner 2012]
Randomness from untrusted quantum device

Randomness Amplification
[Colbeck and Renner 2012]

SV-source

untrusted q. devices

one near perfect random bit
Randomness from untrusted quantum device

Randomness Amplification
[Colbeck and Renner 2012]

- SV-source
- untrusted q. devices
- one near perfect random bit

SV source: each bit of constant bias conditioned on previous bits
Randomness from untrusted quantum device

Randomness Amplification
[Colbeck and Renner 2012]

Randomness Expansion
[Colbeck 2006; Colbeck and Kent 2011]

- SV-source
  - untrusted q. devices
  - one near perfect random bit

SV source: each bit of constant bias conditioned on previous bits
Randomness from untrusted quantum device

Randomness Amplification
[Colbeck and Renner 2012]

- SV source: each bit of constant bias conditioned on previous bits
- SV-source
- untrusted q. devices
- one near perfect random bit

Randomness Expansion
[Colbeck 2006; Colbeck and Kent 2011]

- Perfect randomness
- untrusted q. devices
- more near perfect randomness

SV source: each bit of constant bias conditioned on previous bits
Framework for extracting untrusted quantum device
[Chung, Shi, Wu 2014]

Adversary
deterministic
min-entropy source
almost perfect randomness
deterministic
Device A
Device B
Framework for extracting untrusted quantum device

[Chung, Shi, Wu 2014]

- Adversary: all powerful

Adversary

Device A

Device B

Deterministic

Min-entropy source

Almost perfect randomness
Framework for extracting untrusted quantum device
[Chung, Shi, Wu 2014]

- Adversary: all powerful
- prepares/may entangle with device

Adversary

Device A

Device B

deterministic

min-entropy source

almost perfect randomness
Framework for extracting untrusted quantum device
[Chung, Shi, Wu 2014]

- Adversary: all powerful
- prepares/may entangle with device
- can’t talk to/change device after protocol starts

Adversary
deterministic
min-entropy source
almost perfect randomness

Device A

Device B
Framework for extracting untrusted quantum device
[Chung, Shi, Wu 2014]

- Adversary: all powerful
  - prepares/may entangle with device
  - can't talk to/change device after protocol starts

Device: multi-component

Adversary

Deterministic

Min-entropy source

Almost perfect randomness
Framework for extracting untrusted quantum device
[Chung, Shi, Wu 2014]

- Adversary: all powerful
- prepares/may entangle with device
- can’t talk to/change device after protocol starts
- Device: multi-component
- User: deterministic

min-entropy source
almost perfect randomness
deterministic
Framework for extracting untrusted quantum device

[Chung, Shi, Wu 2014]

- Adversary: all powerful
  - prepares/may entangle with device
  - can’t talk to/change device after protocol starts
- Device: multi-component
- User: deterministic
  - can restrict communication among device components

Adversary
deterministic

Device A

Device B

min-entropy source
almost perfect randomness
deterministic
Framework for extracting untrusted quantum device

[Chung, Shi, Wu 2014]

- Adversary: all powerful
  - prepares/may entangle with device
  - can't talk to/change device after protocol starts
- Device: multi-component
- User: deterministic
  - can restrict communication among device components
- Min-entropy source

Adversary

Device A

Device B

Deterministic

Min-entropy source

Almost perfect randomness
Framework for extracting untrusted quantum device [Chung, Shi, Wu 2014]

- **Adversary**: all powerful
  - prepares/may entangle with device
  - can’t talk to/change device after protocol starts

- **Device**: multi-component

- **User**: deterministic
  - can restrict communication among device components

- **Min-entropy source**: necessary; otherwise Adversary can cheat
Amplification and expansion seen in the framework

Adversary

Device A

Device B

deterministic

min-entropy source

almost perfect randomness
Amplification and expansion seen in the framework

- Randomness Amplification: seedless extraction with an SV source

Device A → deterministic → Device B

- min-entropy source
- almost perfect randomness
Amplification and expansion seen in the framework

- Randomness Amplification: seedless extraction with an SV source
- min-entropy source
- almost perfect randomness
- deterministic
- Untrusted SV
- One bit

Device A

Device B

Adversary
Amplification and expansion seen in the framework

- Randomness Amplification: seedless extraction with an SV source
- Randomness Expansion: seeded extraction (classical source perfectly random)

Device A

Adversary

Device B

Deterministic

Min-entropy source

Almost perfect randomness

SV

Untrusted one bit
Amplification and expansion seen in the framework

- Randomness Amplification: seedless extraction with an SV source
- Randomness Expansion: seeded extraction (classical source perfectly random)

Device A

Adversary

Device B

Deterministic

Min-entropy source

Almost perfect randomness

Perfect untrusted

More

SV untrusted

One bit
Amplification and expansion seen in the framework

- Randomness Amplification: seedless extraction with an SV source
- Randomness Expansion: seeded extraction (classical source perfectly random)

Randomness to be extracted is in the device

Device A

Deterministic

Almost perfect randomness

Device B

Min-entropy source

Perfect untrusted

One bit

SV

More

Untrusted

Untrusted
Amplification and expansion seen in the framework

- Randomness Amplification: seedless extraction with an SV source
- Randomness Expansion: seeded extraction (classical source perfectly random)

Randomness to be extracted is almost perfect randomness

Used to unlock the randomness

Deterministic

Min-entropy source

Almost perfect randomness

Perfect untrusted

More

Untrusted

One bit

SV

Adversary
Goals: basic list

Adversary

Device A

Device B

deterministic

min-entropy source

almost perfect randomness
1. Security: quantum

Adversary

Device A

Device B

min-entropy source

deterministic

almost perfect randomness
Goals: basic list

1. Security: quantum
2. Quality: negligible errors

Adversary

Device A

Device B

deterministic

min-entropy source

almost perfect randomness
Goals: basic list

1. Security: quantum
2. Quality: negligible errors
   • Completeness error
Goals: basic list

1. Security: quantum
2. Quality: negligible errors
   • Completeness error
   • Soundness error

Adversary

Device A

Device B

deterministic

min-entropy source

almost perfect randomness
Goals: basic list

1. Security: quantum
2. Quality: negligible errors
   - Completeness error
   - Soundness error
   - Cryptographic security

Adversary

Device A

Device B

deterministic

min-entropy source

almost perfect randomness
Goals: basic list

1. Security: quantum
2. Quality: negligible errors
   • Completeness error
   • Soundness error,
   • Cryptographic security
3. Output length: extract full device capacity

Adversary

Device A

Device B

deterministic

min-entropy source

almost perfect randomness
Goals: basic list

1. Security: quantum
2. Quality: negligible errors
   - Completeness error
   - Soundness error
   - Cryptographic security
3. Output length: extract full device capacity
4. Classical source: arbitrary min-entropy source
Goals: Premium list critical for realization

Adversary

Device A

Device B

deterministic

min-entropy source

almost perfect randomness
Goals: Premium list critical for realization

5. Robustness: tolerate a constant-level device imprecision

- Adversary
- Device A
- Device B
- Deterministic
- Min-entropy source
- Almost perfect randomness
5. Robustness: tolerate a constant-level device imprecision

- Model noise by deviation from ideal
Goals: Premium list critical for realization

5. Robustness: tolerate a constant-level device imprecision
   • Model noise by deviation from ideal

6. Efficiency

Adversary

Device A

Device B

deterministic

min-entropy source

almost perfect randomness
5. Robustness: tolerate a constant-level device imprecision
   • Model noise by deviation from ideal

6. Efficiency
   • Minimize device number

Adversary
deterministic
min-entropy source
almost perfect randomness

Goals: Premium list critical for realization
Goals: Premium list critical for realization

5. Robustness: tolerate a constant-level device imprecision
   - Model noise by deviation from ideal

6. Efficiency
   - Minimize device number
   - Computationally efficient

Adversary
deterministic

Device A

Device B

min-entropy source

almost perfect randomness
5. Robustness: tolerate a constant-level device imprecision
   - Model noise by deviation from ideal

6. Efficiency
   - Minimize device number
   - Computationally efficient

7. Quantum memory: the smaller needed the better

Seeded extraction of Vazirani-Vidick [2012] and subsequent works

\[ N \text{ rounds of CHSH} \]

Deterministic

\[ k \text{ uniform bits} \]

\[ \exp(k^e) \text{-bits} \]

\[ \exp(-k^e) \text{ error} \]
Seeded extraction of Vazirani-Vidick [2012] and subsequent works

- 1 device (2 components), play \( N \) sequential CHSH

\[ \text{deterministic} \]

\( k \) uniform bits \( \exp(k^c) \)-bits \( \exp(-k^c) \) error
Seeded extraction of Vazirani-Vidick [2012] and subsequent works

- 1 device (2 components), play \( N \) sequential CHSH
- choose a small number of games for testing; others for generating

\[ \exp(k^e) \text{-bits error} \]

\[ \exp(-k^e) \text{-bits} \]
Seeded extraction of Vazirani-Vidick [2012] and subsequent works

- 1 device (2 components), play $N$ sequential CHSH
- Choose a small number of games for testing; others for generating
- Test round: use random input
- Generating round: use constant

$N$ rounds of CHSH

$\begin{align*}
&\text{k uniform bits} \\
&\exp(k^c)\text{-bits} \\
&\exp(-k^c)\text{ error}
\end{align*}$
Seeded extraction of Vazirani-Vidick [2012] and subsequent works

- 1 device (2 components), play \( N \) sequential CHSH
- Choose a small number of games for testing; others for generating
- Test round: use random input
- Generating round: use constant
- Abort when losing too much in test rounds

\[ k \text{ uniform bits} \quad \exp(k^c) \text{-bits} \quad \exp(-k^c) \text{ error} \]
Self-testing and generating random numbers

$A \xrightarrow{a} x \xleftarrow{a} A$

$B \xrightarrow{b} y \xleftarrow{b} B$
Self-testing and generating random numbers

- Optimal quantum strategy: \( a \) is uniform (to adversary)
Self-testing and generating random numbers

- Optimal quantum strategy: \( a \) is uniform (to adversary)
- by strong self-testing: close to optimal implies close to uniform
Intuition for why it should work
Intuition for why it should work

Testing

Generating
Intuition for why it should work

- Mix work and test: cheating on input 0 risks failing test.
Intuition for why it should work

- Mix work and test: cheating on input 0 risks failing test
- Winning ratio of testing rounds good estimate for that of generating rounds
Intuition for why it should work

- Mix work and test: cheating on input 0 risks failing test
- Winning ratio of testing rounds good estimate for that of generating rounds
- Strong self-testing ensures randomness if many rounds’ strategy close to optimal
Intuition for why it should work

- Mix work and test: cheating on input 0 risks failing test
- Winning ratio of testing rounds good estimate for that of generating rounds
- Strong self-testing ensures randomness if many rounds' strategy close to optimal
- Actual proof is very difficult
Intuition for why it should work

- Mix work and test: cheating on input 0 risks failing test
- Winning ratio of testing rounds good estimate for that of generating rounds
- Strong self-testing ensures randomness if many rounds’ strategy close to optimal
- Actual proof is very difficult
Seeded extraction of Miller-Shi [2014]

\[ \text{N rounds of CHSH} \]

\[ k \text{ uniform bits} \quad \exp(k^{c})\text{-bits} \quad \exp(-k^{c}) \text{ error} \]

deterministic
Seeded extraction of Miller-Shi [2014]

- Cryptographic security: errors = negligible in running time

N rounds of CHSH

deterministic

k uniform bits  \exp(k^{c^{}})-bits  \exp(-k^c) \text{ error}
Seeded extraction of Miller-Shi [2014]

- Cryptographic security: errors = negligible in running time
- Robustness: OPT - constant winning prob. suffices

\[ N \text{ rounds of CHSH} \]

\[ k \text{ uniform bits} \rightarrow \text{deterministic} \rightarrow \exp(k^c)\text{-bits} \]

\[ \exp(-k^c) \text{ error} \]
Seeded extraction of Miller-Shi [2014]

- Cryptographic security: errors = negligible in running time
- Robustness: OPT - constant winning prob. suffices
- Constant size q. memory: allow in-between-rounds communications

\[ N \text{ rounds of CHSH} \]

\[ k \text{ uniform bits} \quad \exp(k^c)-\text{bits} \quad \exp(-k^c)\text{ error} \]

\[ \text{deterministic} \]
Seeded extraction of Miller-Shi [2014]

- Cryptographic security: errors = negligible in running time
- Robustness: OPT - constant winning prob. suffices
- Constant size q. memory: allow in-between-rounds communications
- Any strong self-test can be used (including CHSH and GHZ)

$N$ rounds of CHSH

Deterministic

$k$ uniform bits

$\exp(k^c)$-bits

$\exp(-k^c)$ error
Seeded extraction of Miller-Shi [2014]

- Cryptographic security: errors negligible in running time
- Robustness: OPT - constant winning prob. suffices
- Constant size q. memory: allow in-between-rounds communications
- Any strong self-test can be used (including CHSH and GHZ)
- Translates to QKD: exponential expansion at the same time

\[ \text{Deterministic} \]

\[ k \text{ uniform bits} \quad \exp(k^c)-\text{bits} \quad \exp(-k^c) \text{ error} \]

\[ N \text{ rounds of CHSH} \]
Seeded extraction of Miller-Shi [2014]

- Cryptographic security: errors = negligible in running time
- Robustness: OPT - constant winning prob. suffices
- Constant size q. memory: allow in-between-rounds communications
- Any strong self-test can be used (including CHSH and GHZ)
- Translates to QKD: exponential expansion at the same time
- Composed for unbounded expansion

\[ N \text{ rounds of CHSH} \]

\[ k \text{ uniform bits} \rightarrow \exp(k^c)\text{-bits} \rightarrow \exp(-k^c) \text{ error} \]

\[ \text{deterministic} \]
Seeded extraction of Miller-Shi [2014]

- Cryptographic security: errors = negligible in running time
- Robustness: OPT - constant winning prob. suffices
- Constant size q. memory: allow in-between-rounds communications
- Any strong self-test can be used (including CHSH and GHZ)
- Translates to QKD: exponential expansion at the same time
- Composed for unbounded expansion
- Proof quite involved

N rounds of CHSH

deterministic

k uniform bits
exp(ke)-bits
exp(-ke) error
Quantum-proof seeded extractors for min-entropy source
[De et al. 2012]

min-entropy source

-----

deterministic

-----

short seed

-----

near uniform
Quantum-proof seeded extractors for min-entropy source
[De et al. 2012]

min-entropy source

---

deterministic

---

short seed

---

near uniform

---

Trevisan’s Extractors still work against quantum adversaries
Quantum Somewhere Randomness
[Chung, Shi, Wu 2014]

Min-entropy Source $X$

$\text{Ext}\,\text{seed}=1\cdots0$

$\text{Ext}\,\text{seed}=0\cdots0$

$\text{Ext}\,\text{seed}=1\cdots1$

Figure 1: A physical extractor extracting from untrusted quantum devices and a min-entropy source. The input from the min-entropy source $X$ is duplicated. Each copy is used as the input for an instance of a seeded quantum-proof randomness extractor Ext. For each possible value of the extractor seed there is one instance of the extractor with the seed fixed to that value. The output of each extractor instance is used as the input to a "randomness decoupling" (RD) protocol $\uparrow$. A RD protocol makes use of a untrusted quantum device and transforms an input random to the device to an output (almost perfectly) random to all systems other than the device. It also outputs a bit indicating Accept/Abort. We show several existing untrusted-device quantum protocols are randomness decoupling. The output strings of the $\uparrow$ protocols are XOR'ed to form the final output string. The protocol accepts if all the RD protocol accepts.
Quantum Somewhere Randomness

[Chung, Shi, Wu 2014]

Min-entropy Source $X$

$\text{Ext}\text{seed}=1\cdots0$

$\text{Ext}\text{seed}=0\cdots0$

$\text{Ext}\text{seed}=1\cdots1$

$\uparrow$

Output $Y$

Accept/Abort $G$

$X$

$X$

$X$

$\uparrow$

$X$

$\uparrow$

$X$

$\uparrow$

$\uparrow$

$\uparrow$

Figure 1: A physical extractor extracting from untrusted quantum devices and a min-entropy source. The input from the min-entropy source $X$ is duplicated. Each copy is used as the input for an instance of a seeded quantum-proof randomness extractor $\text{Ext}$. For each possible value of the extractor seed there is one instance of the extractor with the seed fixed to that value. The output of each extractor instance is used as the input to a “randomness decoupling” ($\uparrow$) protocol. A RD protocol makes use of a untrusted quantum device and transforms an input random to the device to an output (almost perfectly) random to all systems other than the device. It also outputs a bit indicating Accept/Abort. We show several existing untrusted-device quantum protocols are randomness decoupling. The output strings of the $\uparrow$ protocols are XOR’ed to form the final output string. The protocol accepts if all the RD protocol accepts.

- The average of $y$'s is close to uniform to Adversary

Adversary
Quantum Somewhere Randomness

[Chung, Shi, Wu 2014]

Min-entropy Source $X$

$\text{Ext}_{\text{seed}=0 \cdots 0}$

$\text{Ext}_{\text{seed}=1 \cdots 0}$

$\text{Ext}_{\text{seed}=1 \cdots 1}$

$Y_{0 \cdots 0}$

$Y_{1 \cdots 0}$

$Y_{1 \cdots 1}$

The average of $y$'s is close to uniform to Adversary

A $y$ in an unknown location is close to uniform to Adversary
Randomness Decoupling
[Chung, Shi, Wu 2014]

Adversary

X → D → Y
Randomness Decoupling
[Chung, Shi, Wu 2014]

- If: X uniformly random to Device

![Diagram showing adversary and device interactions]
Randomness Decoupling

[Chung, Shi, Wu 2014]

- If: X uniformly random to Device
- Could be known completely to Adversary
Randomness Decoupling
[Chung, Shi, Wu 2014]

- If: X uniformly random to Device
  - could be known completely to Adversary
- Then: Y is close to uniform to all except Device
Randomness Decoupling
[Chung, Shi, Wu 2014]

- If: X uniformly random to Device
- could be known completely to Adversary
- Then: Y is close to uniform to all except Device
- In particular Y is random to Adversary, even conditioned on X
Randomness Decoupling
[Chung, Shi, Wu 2014]

- If: X uniformly random to Device
- could be known completely to Adversary
- Then: Y is close to uniform to all except Device
- In particular Y is random to Adversary, even conditioned on X
- Turn local randomness to global randomness
Seedless extraction of Chung-Shi-Wu [2014]

Figure 1: A physical extractor extracting from untrusted quantum devices and a min-entropy source. The input from the min-entropy source $X$ is duplicated. Each copy is used as the input for an instance of a seeded quantum-proof randomness extractor $\text{Ext}$. For each possible value of the extractor seed there is one instance of the extractor with the seed fixed to that value. The output of each extractor instance is used as the input to a “randomness decoupling” (RD) protocol $\uparrow$. A RD protocol makes use of a untrusted quantum device and transforms an input random to the device to an output (almost perfectly) random to all systems other than the device. It also outputs a bit indicating Accept/Abort. We show several existing untrusted-device quantum protocols are randomness decoupling. The output strings of the $\uparrow$ protocols are XOR’ed to form the final output string. The protocol accepts if all the RD protocol accepts.
Seedless extraction of Chung-Shi-Wu [2014]

- Input: arbitrary min-entropy source

Min-entropy Source $X$

- $X$

- $seed=0\cdots0$
- $seed=1\cdots0$
- $seed=1\cdots1$

- $\text{Ext}$
- $\Pi$
- $G_{0\cdots0}$
- $Y_{0\cdots0}$
- $G_{1\cdots0}$
- $Y_{1\cdots0}$
- $G_{1\cdots1}$
- $Y_{1\cdots1}$

- $\land$
- $\oplus$

Accept/Abort $G$

Output $Y$

Figure 1: A physical extractor extracting from untrusted quantum devices and a min-entropy source. The input from the min-entropy source $X$ is duplicated. Each copy is used as the input for an instance of a seeded quantum-proof randomness extractor $\text{Ext}$. For each possible value of the extractor seed, there is one instance of the extractor with the seed fixed to that value. The output of each extractor instance is used as the input to a “randomness decoupling” (RD) protocol $\land$. A RD protocol makes use of a untrusted quantum device and transforms an input random to the device to an output (almost perfectly) random to all systems other than the device. It also outputs a bit indicating Accept/Abort. We show several existing untrusted-device quantum protocols are randomness decoupling. The output strings of the $\land$ protocols are XOR’ed to form the final output string. The protocol accepts if all the RD protocol accepts.
Seedless extraction of Chung-Shi-Wu [2014]

- Input: arbitrary min-entropy source
- Delicate composition of quantum-proof extractors for min-entropy source and Randomness Decoupling

![Diagram of the seedless extraction process]

Min-entropy Source $X$

- $Ext$ with seed $= 0 \ldots 0$
- $Ext$ with seed $= 1 \ldots 0$
- $Ext$ with seed $= 1 \ldots 1$

$\uparrow$

$G_{0\ldots 0}$ $Y_{0\ldots 0}$ $G_{1\ldots 0}$ $Y_{1\ldots 0}$ $G_{1\ldots 1}$ $Y_{1\ldots 1}$

Accept/Abort $G$  Output $Y$
Seedless extraction of Chung-Shi-Wu [2014]

- **Input:** arbitrary min-entropy source
- **Delicate composition of quantum-proof extractors for min-entropy source and Randomness Decoupling**
- **First create “quantum somewhere randomness”, then transformed to global randomness through Randomness Decoupling**

![Diagram](image)

- Min-entropy Source $X$
- $\text{Ext}^\text{seed}=1\cdots1$
- $\text{Ext}^\text{seed}=0\cdots0$
- $\text{Ext}^\text{seed}=1\cdots0$
- $\cap$
- $\oplus$
- $G_0\cdots0$
- $G_1\cdots1$
- $Y_0\cdots0$
- $Y_1\cdots1$
- Output $Y$
- Accept/Abort $G$
Seedless extraction of Chung-Shi-Wu [2014]

- **Input**: arbitrary min-entropy source
- **Delicate composition of quantum-proof extractors** for min-entropy source and Randomness Decoupling
- **First create** "quantum somewhere randomness", then transformed to global randomness through Randomness Decoupling
Seedless extraction of Chung-Shi-Wu [2014]

- Input: arbitrary min-entropy source
- Delicate composition of quantum-proof extractors for min-entropy source and Randomness Decoupling
- First create “quantum somewhere randomness”, then transformed to global randomness through Randomness Decoupling

Min-entropy Source $X$

![Diagram showing the process]

- $\text{Ext}$ is used to extract randomness from $X$.
- $\text{Ext}$ is used to extract randomness from $X$.
- The output strings of the protocols are XOR'ed to form the final output string.
- The protocol accepts if all the RD protocol accepts.

Accept/Abort $G$

Output $Y$
Seedless extraction of Chung-Shi-Wu [2014]

- Input: arbitrary min-entropy source
- Delicate composition of quantum-proof extractors for min-entropy source and Randomness Decoupling
- First create "quantum somewhere randomness", then transformed to global randomness through Randomness Decoupling
- XORed with others still globally random

Min-entropy Source $X$

![Diagram of the extraction process]

---

**Key Points:**
- Input: arbitrary min-entropy source
- Delicate composition of quantum-proof extractors for min-entropy source and Randomness Decoupling
- First create "quantum somewhere randomness", then transformed to global randomness through Randomness Decoupling
- XORed with others still globally random
Equivalence Lemma

[Chung, Shi, Wu 2014]

P works for global-uniform input if and only if P works for uniform-to-device input
The Equivalence Lemma [Chung, Shi, Wu 2014]

- Copying $X$ to Adversary commutes with $P$: b/c no interaction between Adversary and Device

$P$ works for global-uniform input if and only if $P$ works for uniform-to-device input.

**Global-uniform**

**Uniform to device**
Equivalence Lemma

[Chung, Shi, Wu 2014]

- Copying $X$ to Adversary commutes with $P$: b/c no interaction between Adversary and Device
- Uniform-to-device = Apply commuting operation on global-uniform

$P$ works for global-uniform input if and only if $P$ works for uniform-to-device input.
Implication of E.L: Instantiation of seedless extraction

Min-entropy Source $X$

Ext

seed=0···0

······

Ext

seed=1···0

······

Ext

seed=1···1

⇧

Output $Y$

Accept/Abort $G$

Figure 1: A physical extractor extracting from untrusted quantum devices and a min-entropy source. The input from the min-entropy source $X$ is duplicated. Each copy is used as the input for an instance of a seeded quantum-proof randomness extractor $Ext$. For each possible value of the extractor seed there is one instance of the extractor with the seed fixed to that value. The output of each extractor instance is used as the input to a "randomness decoupling" (RD) protocol $⇧$. A RD protocol makes use of a untrusted quantum device and transforms an input random to the device to an output (almost perfectly) random to all systems other than the device. It also outputs a bit indicating Accept/Abort. We show several existing untrusted-device quantum protocols are randomness decoupling. The output strings of the $⇧$ protocols are XOR'ed to form the final output string. The protocol accepts if all the RD protocol accepts.
Implication of E.L: Instantiation of seedless extraction

- $X$ needs only to have min-entropy against Devices

Min-entropy Source $X$

$\text{Ext}_{\text{seed}=0\cdots0} \quad \text{Ext}_{\text{seed}=1\cdots0} \quad \text{Ext}_{\text{seed}=1\cdots1}$

$\Pi \quad \Pi \quad \Pi$

$G_{0\cdots0} \quad Y_{0\cdots0} \quad G_{1\cdots0} \quad Y_{1\cdots0} \quad G_{1\cdots1} \quad Y_{1\cdots1}$

Accept/Abort $G$  Output $Y$
Implication of E.L: Instantiation of seedless extraction

- \( X \) needs only to have min-entropy against Devices
- \( \Pi \) can be any untrusted device expansion or KD protocol

Figure 1: A physical extractor extracting from untrusted quantum devices and a min-entropy source. The input from the min-entropy source \( X \) is duplicated. Each copy is used as the input for an instance of a seeded quantum-proof randomness extractor \( \text{Ext} \). For each possible value of the extractor seed there is one instance of the extractor with the seed fixed to that value. The output of each extractor instance is used as the input to a “randomness decoupling” (RD) protocol \( \uparrow \). A RD protocol makes use of a untrusted quantum device and transforms an input random to the device to an output (almost perfectly) random to all systems other than the device. It also outputs a bit indicating Accept/Abort. We show several existing untrusted-device quantum protocols are randomness decoupling. The output strings of the \( \uparrow \) protocols are XOR’ed to form the final output string. The protocol accepts if all the RD protocol accepts.
Implication of E.L: Instantiation of seedless extraction

- $X$ needs only to have min-entropy against Devices
- $\Pi$ can be any untrusted device expansion or KD protocol
- Vazirani-Vidick for expansion (not robust)
Implication of E.L: Instantiation of seedless extraction

- $X$ needs only to have min-entropy against Devices
- $\Pi$ can be any untrusted device expansion or KD protocol
- Vazirani-Vidick for expansion (not robust)
- Miller-Shi (robust)
Implication of E.L: Instantiation of seedless extraction

- $X$ needs only to have min-entropy against Devices
- $\Pi$ can be any untrusted device expansion or KD protocol
  - Vazirani-Vidick for expansion (not robust)
  - Miller-Shi (robust)
  - Vazirani-Vidick for KD (robust)
Unbounded Expansion
Unbounded Expansion

- Straightforward by sequential composition
  - To get N bits need only \( O(\log^* N) \) applications of an exp expanding protocol
- Seed length independent of N
- Classically seed length \( \geq \log N \)
- Multiple-Device extraction
  - Not surprising in our view since randomness comes from device
Unbounded Expansion

- Straightforward by sequential composition
- To get $N$ bits need only $O(\log^* N)$ applications of an exp expanding protocol
- Seed length independent of $N$
- Classically seed length $\geq \log N$
- Multiple-Device extraction
- Not surprising in our view since randomness comes from device
Unbounded Expansion

- Straightforward by sequential composition
- To get $N$ bits need only $O(\log^* N)$ applications of an exp expanding protocol
- Seed length independent of $N$
- Classically seed length $\geq \log N$

- Multiple-Device extraction
- Not surprising in our view since randomness comes from device

- Reduce $\log^* N$ to a constant? Cross-feeding composition problem: the output of $D_0$ when used by $D_1$ is not global-uniform [Fehr-Gelles-Schaffner'13]
- Constant-number-device Unbounded (non-robust) expansion first claimed by Coudron-Yuen [2013]
- Use sequential rigidity of CHSH [RUV] to transform device-uniform input to adversary-uniform output
Implication of E.L.: Unbounded Expansion using 2 devices and any sub-protocol
Implication of E.L.: Unbounded Expansion using 2 devices and any sub-protocol

$X_0, X_2, X_4, \ldots, X_{2N}$

$Y_0, Y_2, Y_4, \ldots, Y_{2N}$

$X_1, X_3, X_5, \ldots, X_{2N+1}$

$Y_1, Y_3, Y_5, \ldots, Y_{2N-1}$

$D_0$

$D_1$
Implication of E.L.: Unbounded Expansion using 2 devices and any sub-protocol

- Each output is random to the other device, ensuring next output is globally random by E.L.
- Any expansion protocol can be used:
  - Vazirani-Vidick (not robust)
  - Miller-Shi (robust)
Put together MS and CSW

Adversary

many devices

2 devices

CSW

$\Theta(k)$-bit local randomness

$K$ min-entropy

MS for unbounded expansion

$N$-bit $\exp(-k^c)$-close to uniform
Put together MS and CSW

- Robust unbounded expansion from an arbitrary min-entropy source

Adversary

k min-entropy

CSW

\(\Theta(k)\)-bit local randomness

MS for unbounded expansion

2 devices

\(\exp(-k^c)\)-close to uniform

many devices

N-bit

Put together MS and CSW

- Robust unbounded expansion from an arbitrary min-entropy source
- Based on an arbitrary strong self-test

Adversary

k min-entropy

CSW

\( \Theta(k) \)-bit local randomness

MS for unbounded expansion

exp\((-k^c)\)-close to uniform

N-bit

many devices

2 devices
Untrusted Device Key Distribution
[Vazirani, Vidick 2013], [Miller, Shi 2014]
Untrusted Device Key Distribution
[Vazirani, Vidick 2013], [Miller, Shi 2014]

Adversary

public channel

Coordinate through public channel

Alice

X

Bob
Untrusted Device Key Distribution
[Vazirani, Vidick 2013], [Miller, Shi 2014]

- Coordinate through public channel
- Not a problem by Equivalence Lemma
Untrusted Device Key Distribution
[Vazirani, Vidick 2013], [Miller, Shi 2014]

Adversary

- Coordinate through public channel
- Not a problem by Equivalence Lemma
- Leakage due to classical post-processing
Adversary

Coordinate through public channel

Not a problem by Equivalence Lemma

Leakage due to classical post-processing

Miller-Shi

Untrusted Device Key Distribution
[Vazirani, Vidick 2013], [Miller, Shi 2014]
Untrusted Device Key Distribution

[Vazirani, Vidick 2013], [Miller, Shi 2014]

Adversary

Coordinate through public channel

- Not a problem by Equivalence Lemma
- Leakage due to classical post-processing
- Miller-Shi
  - expansion at the same time
Untrusted Device Key Distribution

[Vazirani, Vidick 2013], [Miller, Shi 2014]

Adversary

Coordinate through public channel

Not a problem by Equivalence Lemma

Leakage due to classical post-processing

Miller-Shi

expansion at the same time

Any strong self-test
Conclusions
Conclusions
Conclusions

- We may not want to trust quantum devices
Conclusions

• We may not want to trust quantum devices

• Yet we can still make them work their magic
Conclusions

• We may not want to trust quantum devices

• Yet we can still make them work their magic

• Many fundamental questions remain open
Open Problems on robust self-testing
Open Problems on robust self-testing

• Characterize all (robust) self-testing non-local games
Open Problems on robust self-testing

- Characterize all (robust) self-testing non-local games
- Going beyond binary XOR games and graph states
Open Problems on robust self-testing

- Characterize all (robust) self-testing non-local games
  - going beyond binary XOR games and graph states
- Characterize all sequentially self-testing games
Open Problems on untrusted device q. computation

Randomized classical Verifier
Open Problems on untrusted device q. computation

- Robustness

Randomized classical Verifier
Open Problems on untrusted device q. computation

- Robustness
- Quantum memory

Randomized classical Verifier
Open Problems on untrusted device q. computation

- Robustness
- Quantum memory
- Broader class of games

Randomized classical Verifier
Perfect Untrusted-Device Extractor?
Perfect Untrusted-Device Extractor?

Is there a untrusted device protocol achieving the following simultaneously?
Perfect Untrusted-Device Extractor?

Is there a untrusted device protocol achieving the following simultaneously?

- Classical source: arbitrary min-entropy source
Is there a untrusted device protocol achieving the following simultaneously?

- Classical source: arbitrary min-entropy source
- Device (honest):
Is there a untrusted device protocol achieving the following simultaneously?

- Classical source: arbitrary min-entropy source
- Device (honest):
  - Single (multi-part) device
Perfect Untrusted-Device Extractor?

Is there a untrusted device protocol achieving the following simultaneously?

- Classical source: arbitrary min-entropy source
- Device (honest):
  - Single (multi-part) device
  - Robust
Is there a untrusted device protocol achieving the following simultaneously?

- Classical source: arbitrary min-entropy source
- Device (honest):
  - Single (multi-part) device
  - Robust
  - Constant quantum memory
Is there a untrusted device protocol achieving the following simultaneously?

- Classical source: arbitrary min-entropy source
- Device (honest):
  - Single (multi-part) device
  - Robust
  - Constant quantum memory

Output:
Perfect Untrusted-Device Extractor?

Is there a untrusted device protocol achieving the following simultaneously?

• Classical source: arbitrary min-entropy source
• Device (honest):
  • Single (multi-part) device
  • Robust
• Constant quantum memory
• Output:
  • Exponential expanding (or even more?)
Is there a untrusted device protocol achieving the following simultaneously?

- Classical source: arbitrary min-entropy source
- Device (honest):
  - Single (multi-part) device
  - Robust
  - Constant quantum memory
- Output:
  - Exponential expanding (or even more?)
  - Quantum crypto security
Other open problems in untrusted-device extractors
Other open problems in untrusted-device extractors

- Parameter limits and tradeoffs in seeded extraction
Other open problems in untrusted-device extractors

- Parameter limits and tradeoffs in seeded extraction
- Output length in terms of entanglement
Other open problems in untrusted-device extractors

- Parameter limits and tradeoffs in seeded extraction
  - output length in terms of entanglement
  - maximum output length for a fixed number of devices
Other open problems in untrusted-device extractors

- Parameter limits and tradeoffs in seeded extraction
  - output length in terms of entanglement
  - maximum output length for a fixed number of devices
- Security proof based only on non-signaling (local changes will not affect other systems' local behavior)
Physical Extractors: a new paradigm for randomness extraction

- Allow “physical sources”
- Base security of the validity of physical theory
- Any different systems?