

Title: 13/14 PSI - Quantum Information Review - Lecture 14

Date: Mar 07, 2014 11:30 AM

URL: <http://pirsa.org/14030028>

Abstract:

- Fault tolerance
- Quantum key distribution

- Fault tolerance
- Quantum key distribution

Strategy for fault-tolerant QC:

- suppose probability of error per gate is p
- construct encoding, scheme for performing logical gates

- Fault tolerance
- Quantum key distribution

Strategy for fault-tolerant QC:

- suppose probability of error per gate is p
- construct encoding, scheme for performing logical gates so that error prob. per logical gate is $< C \cdot p^2$

- Fault tolerance
- Quantum key distribution

Strategy for fault-tolerant QC:

- suppose probability of error per gate is p
- construct encoding, scheme for performing logical gates so that error prob. per logical gate is $< C \cdot p^2$
- for $p < \frac{1}{C}$, this is an improvement

ant QC:

error per gate is p

scheme for performing
error prob. per

p^2

is an improvement

Strategy for arb. small error: concatenation

ant QC:

error per gate is p

scheme for performing
error prob. per

p^2

is an improvement

Strategy for arb. small error: concatenation

- # of qubits goes up exponentially with # levels of concatenation

ant QC:

error per gate is p

scheme for performing
error prob. per

p^2

is an improvement

Strategy for arb. small error: concatenation

- # of qubits goes up exponentially with # levels of concatenation

- but, error rate goes down doubly exponentially

ant QC:

error per gate is p

scheme for performing

error prob. per

p^2

is an improvement

Strategy for arb. small error: concatenation

- # of qubits goes up exponentially with # levels of concatenation

- but, error rate goes down doubly exponentially

$\Rightarrow$ overhead is $\text{poly}(\log \frac{1}{\epsilon})$

ant QC:

error per gate is p

scheme for performing

error prob. per

p^2

is an improvement

Strategy for arb. small error: concatenation

- # of qubits goes up exponentially with # levels of concatenation

- but, error rate goes down doubly exponentially

$\Rightarrow$ overhead is $\text{poly}(\log \frac{1}{\epsilon}, \log(\# \text{ original gates}))$

ant QC:

error per gate is p

scheme for performing

error prob. per

p^2

is an improvement

Strategy for arb. small error: concatenation

- # of qubits goes up exponentially with # levels of concatenation

- but, error rate goes down doubly exponentially

$\Rightarrow$ overhead is $\text{poly}(\log \frac{1}{\epsilon}, \log(\# \text{ original gates}))$

Threshold Theorem: This works if the physical error rate is below some threshold.

Quantum key distribution

Public-key crypto relies on (unproven)
hardness assumptions

Instead, can we have information-theoretic security?

proven)

Alternative: Private-key cryptography
(One-time pad)

Theoretic security?

Alternative: Private-key cryptography
(One-time pad)



proven)

theoretic security?

Alternative: Private-key cryptography
(One-time pad)



A wants to send $m = m_1, \dots, m_n \in \{0, 1\}^n$ to B

Alternative: Private-key cryptography
(One-time pad)



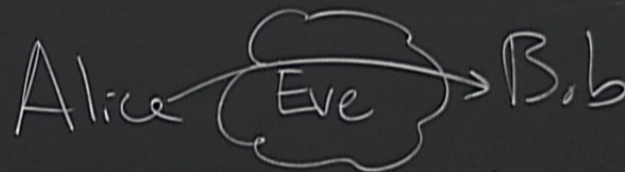
A wants to send $m = m_1 \dots m_n \in \{0, 1\}^n$ to B
they share a secret key $k = k_1 \dots k_n \in \{0, 1\}^n$

Alternative: Private-key cryptography
(One-time pad)



A wants to send $m = m_1, \dots, m_n \in \{0, 1\}^n$ to B
they share a secret key $k = k_1, \dots, k_n \in \{0, 1\}^n$
with each k_i uniformly random, known only to A & B.

Alternative: Private-key cryptography
(One-time pad)



A wants to send $m = m_1, \dots, m_n \in \{0, 1\}^n$ to B
they share a secret key $k = k_1, \dots, k_n \in \{0, 1\}^n$
with each k_i uniformly random, known only to A & B.

Alternative: Private-key cryptography
(One-time pad)



A wants to send $m = m_1, \dots, m_n \in \{0, 1\}^n$ to B
they share a secret key $k = k_1, \dots, k_n \in \{0, 1\}^n$
with each k_i uniformly random, known only to A & B.
(have to meet to establish key)

Alternative: Private-key cryptography
(One-time pad)

$$\text{Ciphertext: } c_i = m_i \oplus k_i$$

Alice  Bob

A wants to send $m = m_1, \dots, m_n \in \{0, 1\}^n$ to B
they share a secret key $k = k_1, \dots, k_n \in \{0, 1\}^n$
with each k_i uniformly random, known only to A & B.
(have to meet to establish key)

Alternative: Private-key cryptography
(One-time pad)



A wants to send $m = m_1, \dots, m_n \in \{0, 1\}^n$ to B
they share a secret key $k = k_1, \dots, k_n \in \{0, 1\}^n$
with each k_i uniformly random, known only to A & B.
(have to meet to establish key)

Ciphertext: $c_i = m_i \oplus k_i$
Eve can't learn anything about m_i from c_i .

Alternative: Private-key cryptography
(One-time pad)



A wants to send $m = m_1, \dots, m_n \in \{0, 1\}^n$ to B
they share a secret key $k = k_1, \dots, k_n \in \{0, 1\}^n$
with each k_i uniformly random, known only to A & B.
(have to meet to establish key)

Ciphertext: $c_i = m_i \oplus k_i$

Eve can't learn anything about m_i from c_i .

But $c_i \oplus k_i = m_i$, so B can recover m .

QKD avoids the need to meet
to establish a key

need:

- ability to send q. states

QKD avoids the need to meet
to establish a key

need:

- ability to send q . states
- authenticated classical channel

BB84 protocol

1. A chooses random states from
2. A sends state to B

(have to meet to establish key)

to meet

BB84 protocol

1. A chooses random states from $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$
2. A sends state to B
- 3.

bates
sical

(have to meet to establish key)

to meet

BB84 protocol

1. A chooses random states from $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$
2. A sends state to B
3. B measures in basis $\{|0\rangle, |1\rangle\}$ or $\{|+\rangle, |-\rangle\}$,
chosen unif. at random

states
classical

(have to meet to establish key)

to meet

BB84 protocol

1. A chooses random states from $\{\underline{10}, \underline{11}, \underline{1+}, \underline{1-}\}$
2. A sends state to B
3. B measures in basis $\{10, 11\}$ or $\{1+, 1-\}$,
chosen unif. at random
4. A & B publicly compare their bases

states
classical

(have to meet to establish key)

to meet

BB84 protocol

1. A chooses random states from $\{\underline{|0\rangle}, \underline{|1\rangle}, \underline{|+\rangle}, \underline{|-\rangle}\}$

2. A sends state to B

3. B measures in basis $\{|0\rangle, |1\rangle\}$ or $\{|+\rangle, |-\rangle\}$,
chosen unif. at random

4. A & B publicly compare their bases & only keep
the cases where B measured in the basis A prepared
in state in.

Simple attack: intercept-resend

Simple attack: intercept-resend

E could measure in $\{|0\rangle, |1\rangle\}$ basis
if basis is correct (A prepared such a
state & B measured in this basis),

Simple attack: intercept-resend

E could measure in $\{|0\rangle, |1\rangle\}$ basis
if basis is correct (A prepared such a
state & B measured in this basis),

E will learn a bit of key
if basis is wrong, A & B's answers will
differ with probability $\frac{1}{2}$, and they
can detect this

Full BB84 protocol (works even with sufficiently small noise)

- Repeat basic unit n times to get $\approx \frac{n}{2}$ "key bits"
- A chooses a subset of $\frac{n}{4}$ bits and announces their locations & values
-

Full BB84 protocol (works even with sufficiently small noise)

- Repeat basic unit n times to get $\approx \frac{n}{2}$ "key bits"
- A chooses a subset of $\frac{n}{4}$ bits and announces their locations & values
- B compares to his outcomes and estimates an error rate; if the rate is above some threshold ($\approx 20\%$)

Full BB84 protocol (works even with sufficiently small noise)

- Repeat basic unit n times to get $\approx \frac{n}{2}$ "key bits"
- A chooses a subset of $\frac{n}{4}$ bits and announces their locations & values
- B compares to his outcomes and estimates an error rate; if the rate is above some threshold ($\approx 20\%$), they abort
- Information reconciliation

Full BB84 protocol (works even with sufficiently small noise)

- Repeat basic unit n times to get $\approx \frac{n}{2}$ "key bits"
- A chooses a subset of $\frac{n}{4}$ bits and announces their locations & values
- B compares to his outcomes and estimates an error rate; if the rate is above some threshold ($\approx 20\%$), they abort
- Information reconciliation: A & B compare parities of subsets of their remaining bits, and use this info to correct bit flips (ECCs); discard one bit of key for every parity they reveal

• Privacy amplification: A & B compute parities of random subsets of their bits (hashing) to produce final key bits (about which Eve has arbitrarily little information, even about one bit)

This protocol can be proven secure
(within some model of devices, channel, etc.)

This protocol can be proven secure
(within some model of devices, channel, etc.)
Have to worry about side channel attacks

This protocol can be proven secure
(within some model of devices, channel, etc.)
Have to worry about side channel attacks
Even better: device-independent QKD.

This protocol can be proven secure
(within some model of devices, channel, etc.)

Have to worry about side channel attacks

Even better: device-independent QKD.

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

This protocol can be proven secure
(within some model of devices, channel, etc.)

Have to worry about side channel attacks

Even better: device-independent QKD

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}} \quad \text{Ekert 9?}$$