Quantum Cryptography & Computing
defined on
Chapter 19 of Trappe & Washington
"A Physics-Free Introduction to the Quantum Computation Model"
by Stephen Fenner

CIS 428/628  Intro. to Cryptography

December 4, 2009

A Little Optics

- photon = a quanta of light
- polarization = orientation of a photon’s EM wave
- when a photon hits a polarized filter, either
  - it is absorbed, or
  - it is transmitted with the polarization of the filter.
- A photon has “no memory” of a previous polarization.

see pictures on the board

A Little Optics, Continued

Recall:

\[(\sin \theta)^2 + (\cos \theta)^2 = 1.\]

Now, in the y – z plane: See pictures on board

- if \( \alpha \) = the angle of the photon’s polarization
- if \( \beta \) = the angle of the filter’s polarization then

\[
\text{Prob[the photon is absorbed]} = (\sin(\alpha - \beta))^2.
\]
\[
\text{Prob[the photon is transmitted]} = (\cos(\alpha - \beta))^2.
\]

- For \( \alpha = \beta + \pi/2 \), certain absorption.
- For \( \alpha = \beta \), certain transmission.
- For \( \alpha = \beta + \pi/4 \), 50%-50%.

Polarized Photons, continued

FACT: “Cloning” a photon is inconsistent with QM.

\[\therefore\] Encoding info with polarized photons means you have an uncopiable, read-once medium.

- Formalizing all this involves just some linear algebra, geometry, and probability.
Quantum Key Distribution: BB84

**The QC Phase**

- **Alice**
  - Chooses: a seq. of random bits $x_1, \ldots, x_k$
  - Chooses a seq. of random pol. bases $B_1, \ldots, B_k$
  - Transmits: $B_1(x_1), B_2(x_2), \ldots, B_k(x_k)$ over the QC.

- **Bob**
  - Randomly chooses bases $B_1', \ldots, B_k'$
  - For each photon received: $y_i$
  - Bob measures it on basis $B_i'$
  - If $B_i \neq B_i'$, Bob gets a random answer: $\frac{1}{2} : 0 \& \frac{1}{2} : 1$
  - If $B_i = B_i'$, Bob gets the bit sent.

However, there is noise!

- Photons can get lost or go strange in transmission.
- Eve's Problem
  - Observation destroys data.
  - Hence, Eve must supply junk info.
- But with high probability, this is detected.

Testing for Easedropping

- **Bob**
  - Randomly chooses, say, $\frac{1}{3}$, of the bits received and makes them public.
- **Alice**
  - Checks that these are correct and tells Bob.

Eve's Problem

- Observation destroys data.
- Hence, Eve must supply junk info.
- But with high probability, this is detected.

Man-in-the-middle

- A&B can use a PKC to defend against this attack.

**More on Quantum Crypto**

- There is an alternative to BB84 based on entangled photons (Artur Ekert, 1991).
- The noise in the key exchange makes pure BB84 problematic.
- However, there is a crypto-version of error-correction, privacy amplification, that can deal with these problems.
- Quantum crypto is edging towards commercial uses.
Quantum Computing: Classical Boolean Circuits

We view them as naming maps \( \{0, 1\}^n \rightarrow \{0, 1\}^n \)

\[
\begin{align*}
    a & \quad \text{control} \\
    b & \quad a \land b \quad \text{target}
\end{align*}
\]

Consider

\[
\begin{align*}
    a & \quad \neg a \\
    b & \quad (a \land b) \lor c \\
    c & \quad c
\end{align*}
\]

We can describe this by either of:

\[
\begin{align*}
    b & \leftarrow a \land b; \\
    a & \leftarrow \neg a; \\
    b & \leftarrow b \lor c
\end{align*}
\]

\[
|a, b, c\rangle \mapsto |a, a \land b, c\rangle \mapsto |\neg a, a \land b, c\rangle \mapsto |\neg a, (a \land b) \lor c, c\rangle
\]

Classical Boolean Circuits, Continued

Input/Output Conventions

- The first \( k \) registers are input \( 0 \leq k \leq n \)
- The first \( \ell \) registers are output \( 0 \leq \ell \leq n \)
- Each non-input register is assigned 0 or 1

\[
\begin{align*}
    a & \quad \text{a} \\
    0 & \quad \text{a}
\end{align*}
\]

\[
|a, (a, a)\rangle \mapsto |a, a\rangle
\]

Uniform Computation

- A circuit family, \( C \), is a sequence of circuits \( C_0, C_1, C_2, \ldots \) \( \exists \) for each \( i \), \( C_i \) has \( i \)-inputs and 1-output.
- \( L(C) = \text{def} \{ w : |w| = n \land C_n(w) = 1 \} \), the language defined by \( C \).
- A circuit family is ptime uniform iff \( \exists \) a poly-time alg \( D \) \( \exists \) for all \( i \), \( D(1^{i \text{ many}}) = \) a description of \( C_i \).

FACT: \( P = \) the languages accepted by ptime uniform circuit families.
Reversible Circuits

Reversible circuits have inverses.

Toffoli Gate (T) where \( \odot(x, y, z) = z \oplus (x \land y) \)

\[
\begin{array}{ccc}
& a & \\
\hline
b & \quad & b \\
\hline
c & & c \oplus (a \land b)
\end{array}
\]

The controlled not gate (CNOT)

\[
\begin{array}{ccc}
& a & \\
\hline
b & \quad & a \oplus b \\
\hline
\end{array}
\]

Reversible circuits do not collapse states. (Why?)

Probabilistic Circuits

The Biased Coin-Flip Gate — \( p, q \) —

\[
\begin{array}{c|c}
\text{input} & \text{output} \\
\hline
0 & 0:p & 1:(1 - p) \\
1 & 0:q & 1:(1 - q)
\end{array}
\]

\( \overline{\vec{v}} \): \( 2^n \) basis vectors

\[ H : a \text{ 2}^n \text{-dim. real vector space} \]

\[
\begin{array}{c|c}
\vec{x}_1 & \\
\hline
\vec{x}_i & p:q \\
\hline
\vec{x}_n & \\
\end{array}
\]

We can also represent the \( p, q \) gate by the matrix

\[
\begin{bmatrix}
p & q \\1 - p & 1 - q
\end{bmatrix}
\]

This is a stochastic matrix: all entries \( \geq 0 \), all cols sum to 1.

Probabilistic Circuits, Continued

Consider the subspace spanned by \( |0\rangle \) and \( |1\rangle \).

The gate \( p, q \) always maps the line segment from (1,0) to (0,1) to itself.

We can also represent the \( p, q \) gate by the matrix

\[
\begin{bmatrix}
p & q \\1 - p & 1 - q
\end{bmatrix}
\]

“Majority coin flips” circuit: on the board.

Prob. Circuits: Gates as linear maps

Definition

A probabilistic circuit is a circuit built from Boolean \( p, q \) gates, where

- The input state is a basis state.
- The output state is of the form: \( \sum_{x \in \{0,1\}} p_x |x\rangle \) \( \exists \)
  - (i) each \( p_x \geq 0 \) and (ii) \( \sum |p_x| = 1 \).
- \( p_x \) = the probability that the output will be \( |x\rangle \).

“Majority coin flips” circuit: on the board.
Quantum Circuits (Á La Fenner)

- states = vectors in $\mathcal{H}$  
- gates = matrices
- Now allow nonnegative entries in matrices. (But all real numbers)
- Now require: $\|Mv\|_2 = \|v\|_2$ for all $v$.
- Note: $\|\vec{a}\|_2 = \sqrt{a_1^2 + \cdots + a_n^2}$
- This forces the matrices to be orthonormal, i.e., its cols form an orthogonal basis of $\mathcal{H}$.
- Registers are now called qubits, quantum bits.
- The Hadamard gate, $[H]$, has the matrix:

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Note: $H^2 = I$.

Fact: The H and T gates form a universal collection of quantum gates.

The $[p, q]$ gates now correspond to measurements.

“Traditional” Quantum Circuits

- In place of vector spaces over $\mathbb{R}$, we use v.s.’s over $\mathbb{C}$.
- In place of orthonormal matrices, we use unitary matrices.
- Etc., etc. See §6 of Fenner for details.

- Past this point, we shall be even sketchier than before. ...SO, we won’t digress in to complex linear algebra

Quantum Circuits (Á La Fenner), II

QCF (Quantum Coin Flip)

This is a variation on Hadamard gate.

$$\text{QCF} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

Note that $(\text{QCF})^2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ = the not gate.

So, QCF = $\sqrt{\text{NOT}}$.

Quantum I/O

Input: basis states
Output: $\sum_{x \in \{0,1\}^n} a_x |x\rangle$  

$a_x^2 = \text{the probability associated with} |x\rangle$  

$a_x = \text{the probability amplitude for} |x\rangle$

Shor’s Factoring Algorithm, I

The exponent factorization method (T&W, §6.4.2)

An algorithm that given $a$ and $r \in \mathbb{Z}$

$$a^r \equiv 1 \pmod{n}$$

produces a factorization of $n$ with reasonable probability.

Peter Shor’s clever idea

Use QC to find $a$ and $r$. That is:
- Choose $a$ at random
- Consider $1, a^1, a^2, a^3, \ldots \pmod{n}$
- If $a^r \equiv 1 \pmod{n}$, then the seq. repeats every $r$ times.
  - Finding the period of the sequence, finds $r$.
- In signal processing, Fourier Transforms are used to find periods.
Quantum Fourier Transform

\[ \text{QFT}(|x\rangle) \overset{\text{def}}{=} \frac{1}{\sqrt{2^n}} \sum_{c \in \{0,1\}^n} e^{2\pi i cx} |c\rangle \]

- This can be realized as a quantum circuit.
- We’ll come back to the properties of this thing shortly.

Shor’s Factoring Algorithm, II

\[
\left|0\ldots0,0\ldots0\right\rangle_{m+n \text{ long}} \downarrow \\
\frac{1}{\sqrt{2}} \left( \left|0\ldots0,0\ldots0\right\rangle + \left|1\ldots0,0\ldots0\right\rangle \right) \downarrow \\
\vdots \downarrow \\
\frac{1}{\sqrt{2^n}} \sum_{c \in \{0,1\}^n} |c,\tilde{0}\rangle \quad \text{superimposition of } 2^n \text{ states} \\
\downarrow \\
\frac{1}{\sqrt{2^n}} \sum_{c \in \{0,1\}^n} |c,a^c \mod n\ldots\rangle \\
\downarrow \\
\text{QFT (—)} \quad \text{Now what???}
\]

Shor’s Factoring Algorithm, III

- When you measure \(\sum_i a_i |x_i\rangle\) you get state \(|x_i\rangle\) with probability \(a_i^2\).
- Thanks to QFT, states near the period have pretty high probability.
- . measure, test, and refine. (See text.)
- A similar trick (using QFT) can compute discrete logs.

Grover’s Algorithm

- Suppose that \(C : \{0,1\}^n \rightarrow \{0,1\}\) is such that \(C(s) = 1\) for only one \(s \in \{0,1\}^n\).
- Classically, finding this \(s\) takes \(\Theta(2^n)\) time.
- Using QFT trickery, one can do this in \(\Theta(\sqrt{2^n})\) time.