Quantum Cryptography

Peter Shor
M.I.T.
Cambridge, MA 02139

CSS codes (Calderbank & Shor, Steane)

Start with two binary codes such that

$$\{0\} \subseteq \mathbf{C}_2 \subseteq \mathbf{C}_1 \subseteq \mathbf{F}_2^n$$

(So $\{0\} \subseteq \mathbf{C}_1^\perp \subseteq \mathbf{C}_2^\perp \subseteq \mathbf{F}_2^n$)

The quantum code has basis elements corresponding to $v \in \mathbf{C}_1/\mathbf{C}_2$.

$$v \to \frac{1}{2^{k/2}} \sum_{x \in \mathbf{C}_2} |\, v + x \rangle$$

$k = \dim \mathbf{C_1} - \dim \mathbf{C_2}$

This will correct t errors, where

$$2t + 1 \leq \min(\mathrm{wt}\mathbf{C}_1, \mathrm{wt}\mathbf{C}_2^\perp)$$

$\mathbf{C}_1$ corrects bit errors

$\mathbf{C}_2^\perp$ corrects phase errors

rate: $\frac{\dim \mathbf{C_1} - \dim \mathbf{C_2}}{n}$

Suppose we have a CSS code with

$$\{0\} \subseteq \mathbf{C}_2 \subseteq \mathbf{C}_1 \subseteq \mathbf{F}_2^n.$$

What happens when we apply $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ to each qubit of it?

We start with the encoding of $v \in \mathbf{C}_1/\mathbf{C}_2$:

$$|\, E_v\rangle = |\mathbf{C}_2|^{-1/2} \sum_{x \in \mathbf{C}_2} |\, v + x\rangle\,.$$

When we apply $H^{\otimes n}$ to a state $|\, s\rangle$, we get a (-1) factor for each qubit where $|\, 1\rangle$ goes to $|\, 1\rangle$. Thus,

$$H^{\otimes n}\,|\, s\rangle = \frac{1}{2^{n/2}} \sum_{t \in \mathbf{F}_2^n} (-1)^{s \cdot t}\,|\, t\rangle$$

Applying this to the encoded state $|\, E_v\rangle$, we get

$$H^{\otimes n}\,|\, E_v\rangle = \frac{|\mathbf{C}_2|^{-1/2}}{2^{n/2}} \sum_{t \in \mathbf{F}_2^n} \sum_{x \in \mathbf{C}_2} (-1)^{(v+x) \cdot t}\,|\, t\rangle$$

$$
\begin{aligned}
H^{\otimes n} \,|\, E_v \rangle &= \frac{|\mathbf{C}_2|^{-1/2}}{2^{n/2}} \sum_{t \in \mathbf{F}_2^n} \sum_{x \in \mathbf{C}_2} (-1)^{(v+x)\cdot t} \,|\, t \rangle \\
&= \frac{|\mathbf{C}_2|^{1/2}}{2^{n/2}} \sum_{t \in \mathbf{C}_2^\perp} (-1)^{v \cdot t} \,|\, t \rangle \\
&= \frac{|\mathbf{C}_2|^{1/2}}{2^{n/2}} \sum_{t \in \mathbf{C}_2^\perp / \mathbf{C}_1^\perp} (-1)^{v \cdot t} \sum_{y \in \mathbf{C}_1^\perp} |\, t + y \rangle \\
&= \frac{1}{2^{k/2}} \sum_{t \in \mathbf{C}_2^\perp / \mathbf{C}_1^\perp} (-1)^{v \cdot t} \left| \hat{E}_t \right\rangle .
\end{aligned}
$$

where $\left| \hat{E}_t \right\rangle$ is t encoded in the dual quantum code,

$$\{0\} \subseteq \mathbf{C}_1^\perp \subseteq \mathbf{C}_2^\perp \subseteq \mathbf{F}_2^n.$$

Thus, $H^{\otimes n} \,|\, v \rangle$ is the Fourier transform of the vector v encoded in the dual quantum code.

Quantum Cryptography

First published in 1982, this was one of the first applications of quantum weirdness to computer science tasks.

The BB84 (Bennett and Brassard, 1984) protocol is a key distribution protocol. Two parties, Alice and Bob, are trying to agree on a key which any eavesdropper (Eve) will not be able to ascertain by listening to their conversation. The idea is to use the fact that any attempt to measure a quantum state must inescapably disturb it. Alice sends Bob photons. Bob chooses some of these photons at random and checks for disturbance, while others yield the secret key.

The model we discuss: Alice and Bob have a classical channel which Eve can eavesdrop, and a quantum channel which Eve can do anything to (you can't eavesdrop on a quantum channel without disturbing it). Since Eve can potentially cut the quantum channel, Alice and Bob don't have any guarantee that they will be able to agree on a key. The goal of the protocol is to make the chance of Eve knowing a key that Alice and Bob have agreed on very small.

So with very high probability, Alice and Bob will either agree on a key that Eve doesn't know, or decide that the quantum channel is too noisy for secure communication.

Suppose that we don't like the assumption that Eve can't spoof the classical channel. Then there seems to be no defense against the "man in the middle" attack, where Eve cuts all the channels in the middle, and pretends to be Alice when talking to Bob, and pretends to be Bob when talking to Alice.

What can we do in this case? Suppose Alice and Bob start with a small amount of shared information. They can use this shared information to verify that they are talking to each other, and then use quantum key distribution to generate more shared secret key. In fact, they can do this in a way that Eve can't disrupt. This is secret key amplification.

Secret key amplification cannot be done securely classically without assuming the computational difficulty of some problem.

A) Alice sends random qubits in one of the four states.

$$|0\rangle, \quad |1\rangle, \quad \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

B) Bob measures them randomly in either the $\{|0\rangle, |1\rangle\}$ basis or the $\frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$ basis

C) Alice and Bob reveal the sending and receiving bases, and obtain a string of bits that they agree on.

D) Some of these bits are used to test for errors; the rest form the key.

Example protocol when things work perfectly (i.e., no noise).

Alice sends random states	↖↘	↗↙	↗↙	↔	↖↘	↕	↕	↗↙	↖↘
Bob measures random basis	×	+	×	×	×	+	×	+	×
Bob gets these results	↖↘	↔	↗↙	↖↘	↖↘	↕	↖↘	↕	↖↘
They keep when bases agree	•		•		•	•			•
Alice and Bob get secret bits	1		0		1	0			1

Because the density matrices for the two complementary bases are equal, i.e.,

$$\frac{1}{2}\left|\updownarrow\right\rangle\left\langle\updownarrow\right| + \frac{1}{2}\left|\leftrightarrow\right\rangle\left\langle\leftrightarrow\right| = \frac{1}{2}\left|\nearrow\right\rangle\left\langle\nearrow\right| + \frac{1}{2}\left|\nwarrow\right\rangle\left\langle\nwarrow\right|$$

Eve cannot measure which basis Alice sent her bits in. If Eve gains information in about one of the two possible complementary bases, she disturbs states sent in the other. Thus, if Eve gets any information, she incurs a probability to perturb the signals, and thus be detected.

But Alice and Bob's channel won't be perfect, so there will be some disturbance anyway. How can they do quantum cryptography given a noisy channel?

Quantum Cryptography over Noisy Channels

Alice and Bob use an error correcting code to fix any errors caused by noise.

They then apply a hash function to the resulting key to gain privacy.
We prove security in the case where they use a *linear* hash function.

Linear hash function: Alice and Bob take an n-bit key $\mathbf{k}$, multiply a 0-1 $n \times m$ matrix to it (in binary), and obtain an m-bit key $\mathbf{k}'$.

It took till 1996 to obtain a proof of security for quantum cryptography over noisy channels. We give a much simpler proof (discovered jointly with John Preskill) by using quantum error correcting codes.

The proof first shows a different key distribution protocol, based on error correcting codes, is secure, and then shows that the two protocols are equivalent in terms of what an eavesdropper can learn.

CSS key distribution protocol

Idea: Alice encodes the key using a CSS code. She sends the CSS code to Bob, interspersing it with test bits. Bob uses the test bits to find the error rate, and if the error rate is low enough, he decodes the key and uses it.

If the error rate is low, the CSS code delivers the encoded state with high fidelity. Thus, by the no-cloning theorem of quantum information, the code cannot leak very much information, and the key is thus secure.

This code assumes that Bob has quantum memory, as he has to store the qubits until Alice tells him which are test bits and which are code bits.

So it is currently impractical.

CSS key distribution protocol

Problem: We have to make sure that Eve cannot detect which bits are being used for the code and which bits are being used for the test bits. (If she could, she could leave the test bits alone, and just measure the code bits).

Solution: We use one of a set of shifts of the CSS code. Instead of encoding

$$v \to |\mathbf{C}_2|^{-1/2} \sum_{x \in \mathbf{C}_2} |v + x\rangle$$

we pick a random $w \in \mathbf{F}_2^n$, $z \in \mathbf{F}_2^n$ and encode

$$v \to |\mathbf{C}_2|^{-1/2} \sum_{x \in \mathbf{C}_2} (-1)^{w \cdot x} |v + x + z\rangle$$

Adding z gives a random shift of the code in bit space, and changing the phase using w gives a random shift of the code in the Fourier transform space.

Alice sends a random shift of the CSS quantum code

$$v \to |\mathbf{C}_2|^{-1/2} \sum_{x \in \mathbf{C}_2} (-1)^{w \cdot x} \, | \, v + x + z \rangle$$

where $w \in \mathbf{F}_2^n$ and $z \in \mathbf{F}_2^n$ are chosen at random.

Here Alice has chosen a random $v \in \mathbf{C}_1/\mathbf{C}_2$. The random choices of Alice have the effect of making the quantum state Alice sends perfectly random (without knowledge of z, w, which Alice does not reveal until Bob has received the data). Thus Eve cannot distinguish between the code bits and the random test bits.

Calculations for previous slide:

The density matrix averaging over all encodings of v is

$$\frac{1}{2^{2n}|\mathbf{C}_2|} \sum_{\substack{w \in \mathbf{F}_2^n \\ z \in \mathbf{F}_2^n}} \sum_{x_1, x_2 \in \mathbf{C}_2} (-1)^{w \cdot (x_1 + x_2)} \, |\, v + x_1 + z \rangle \langle v + x_2 + z \,|$$

Summing over w in $(-1)^{w \cdot (x_1 + x_2)}$ gets rid of all $x_1 \neq x_2$.

$$= \frac{1}{2^n |\mathbf{C}_2|} \sum_{z \in \mathbf{F}_2^n} \sum_{x \in \mathbf{C}_2} |\, v + x + z \rangle \langle v + x + z \,|$$

With a random $v \in \mathbf{C}_1/\mathbf{C}_2$, a random z, and a sum over x, this state is the maximally random density matrix $2^{-n} I$.

Equivalence to BB84

Alice never needs to reveal w (the amount the phase encodings are shifted) because Bob only cares about the bits Alice sends for his key, and not the phases of these bits; thus, he doesn't need to correct phase errors.

Thus, the density matrix Eve sees is (from the previous slide)

$$\frac{1}{2^n|\mathbf{C}_2|}\sum_{z\in\mathbf{F}_2^n}\sum_{x\in\mathbf{C}_2}|\,v+x+z\rangle\,\langle v+x+z\,|$$

This is the same as Alice sending the random state $|\,v+x+z\rangle$.

When averaging over all values of w, for a given key, Eve sees the exact same density matrix for the good bits of the BB84 key distribution protocol (those where Alice and Bob use the same basis for sending and receiving), and the CSS code key distribution protocol. Thus, if one is secure, the other must also be secure.

Details of Equivalence

CSS codes are composed of two linear codes, one to correct the bit values, and the other to correct the phases. BB84 has a linear code to correct errors, and a linear hash function used for privacy amplification. The bit correcting code of CSS is exactly the error correcting code for BB84. The phase correcting code of CSS corresponds to the linear hash function for BB84 (The code consists of everything mapped to 0 by the hash function).

For further details, one needs to write down the equations.

What Eve sees.

CSS protocol:
Alice sends $|v + x + z\rangle$, with $v \in \mathbf{C}_1/\mathbf{C}_2$, $z \in \mathbf{F}_2^n$, and $x \in \mathbf{C}_2$.
She then reveals z, and Bob figures out $v \in \mathbf{C}_1/\mathbf{C}_2$.

BB84 protocol:
Alice sends random state $|r\rangle$ ($\equiv |v + x + z\rangle$).
She then sends Bob $r + t$, for a random $t \in \mathbf{C}_2$, and Bob uses error correction to figure out t ($r + t \equiv z$).
Finally, Alice and Bob use hash functions to reduce t to (effectively) a coset of $t \in \mathbf{C}_1/\mathbf{C}_2$ ($t \equiv v + x$).

Eve sees the same density matrix arising from the good bits of BB84 (those where Alice and Bob use the same basis for sending and receiving) and the CSS key distribution protocol. Thus, if one is secure, the other is as well.