Quantum Error Correcting Codes and Quantum Cryptography

Peter Shor

M.I.T.

Cambridge, MA 02139

We start out with two processes which are fundamentally quantum: superdense coding and teleportation.

Superdense coding begins with the question: How much information can one qubit convey? Holevo's bound (we discuss this later) says that one qubit can only convey one bit. However, if a sender and receiver share entanglement, i.e., one EPR pair, they can use this entanglement and one qubit to send two classical bits.

The four states (the Bell basis) are mutually orthogonal

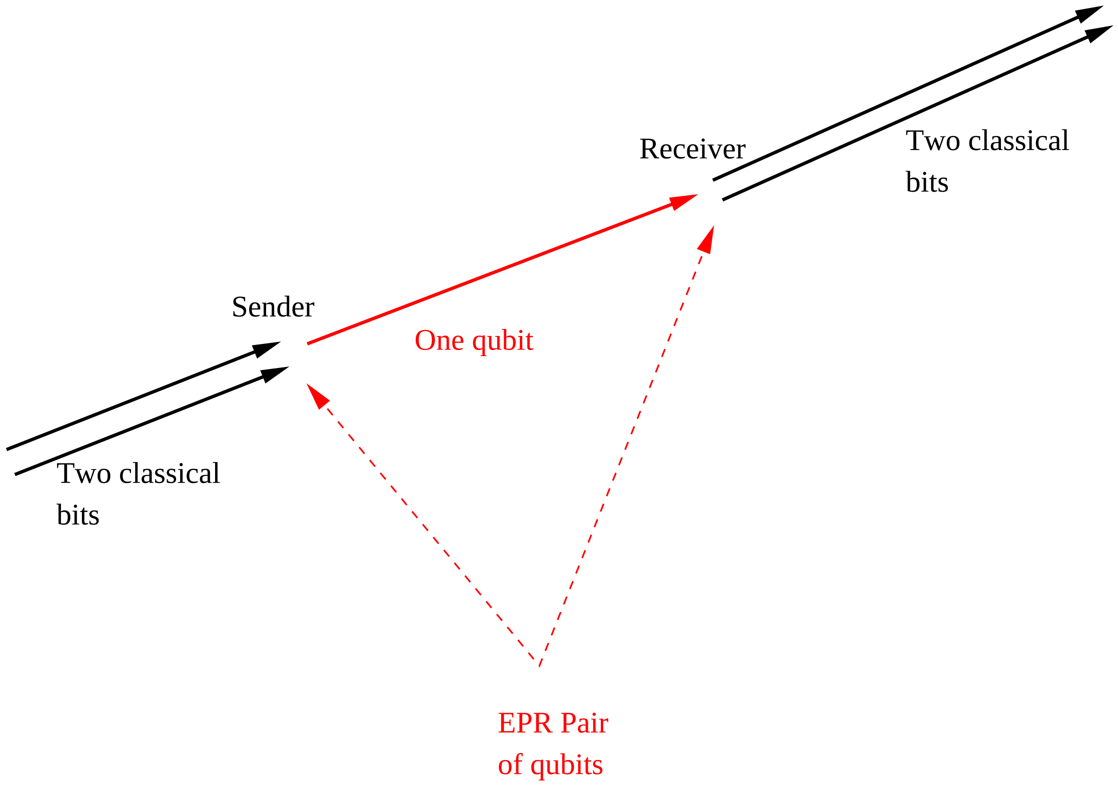
$$\tfrac{1}{\sqrt{2}}(|\,00\rangle + |\,11\rangle) \quad \tfrac{1}{\sqrt{2}}(|\,00\rangle - |\,11\rangle) \quad \tfrac{1}{\sqrt{2}}(|\,01\rangle + |\,10\rangle) \quad \tfrac{1}{\sqrt{2}}(|\,01\rangle - |\,10\rangle)$$

Any one can be turned into any other (up to a phase factor) by applying the appropriate Pauli matrix to either of the two qubits.

$$\mathrm{id} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Superdense Coding:

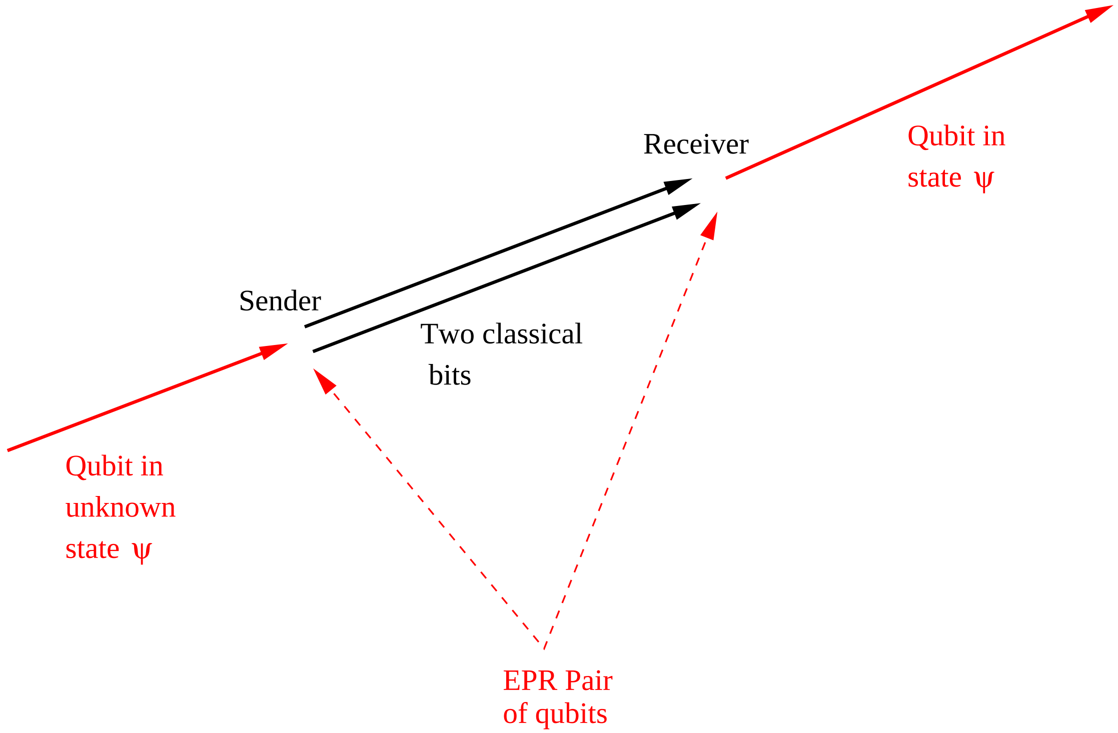


Superdense Coding

Alice and Bob start with the state $\frac{1}{\sqrt{2}}(|\,00\rangle_{AB} + |\,11\rangle_{AB})$, where Alice holds the first qubit and Bob the second. Alice applies one of the four Pauli matrices to her qubit, and sends her qubit to Bob. He can then tell all four Bell states apart, so he knows which Pauli matrix Alice applied. This sends four possibilities (two classical bits) using one qubit and one EPR pair (ebit).

Teleportation:

This is a converse proces to superdense coding.



Teleportation:

Alice and Bob share an EPR pair in the state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$.
Alice has a qubit in the unknown state $\alpha|0\rangle + \beta|1\rangle$. The joint state is

$$\frac{1}{\sqrt{2}}(\alpha|0\rangle + \beta|1\rangle)(|00\rangle + |11\rangle)$$

This can be rewritten (using distributive law, etc.) as

$$\begin{aligned} \frac{1}{\sqrt{8}}\Big[& \ (|00\rangle + |11\rangle) \quad (\alpha|0\rangle + \beta|1\rangle) \\ & +(|00\rangle - |11\rangle) \quad (\alpha|0\rangle - \beta|1\rangle) \\ & +(|10\rangle + |01\rangle) \quad (\beta|0\rangle + \alpha|1\rangle) \\ & +(|10\rangle - |01\rangle) \quad (\beta|0\rangle - \alpha|1\rangle)\Big] \end{aligned}$$

Teleportation:

Recall the joint state was

$$\frac{1}{\sqrt{8}}\Big[\quad (|\,00\rangle + |\,11\rangle) \quad (\alpha\,|\,0\rangle + \beta\,|\,1\rangle)$$
$$+(|\,00\rangle - |\,11\rangle) \quad (\alpha\,|\,0\rangle - \beta\,|\,1\rangle)$$
$$+(|\,10\rangle + |\,01\rangle) \quad (\beta\,|\,0\rangle + \alpha\,|\,1\rangle)$$
$$+(|\,10\rangle - |\,01\rangle) \quad (\beta\,|\,0\rangle - \alpha\,|\,1\rangle)\Big]$$

Alice's four states on the above four lines are all orthogonal. This means Alice can measure which of the four Bell states she has. By the rules of quantum measurement, Bob then has his corresponding state. Alice sends her results to Bob, who can then transform his state into $\alpha\,|\,0\rangle + \beta\,|\,1\rangle$ using a unitary operator.

Let's generalize to higher dimensional quantum states, and to more arbitrary generalizations of the Pauli matrices. Suppose Alice and Bob share the state

$$|\,\psi\rangle = \frac{1}{\sqrt{d}} \sum_{i=1}^{d} |\,i\rangle_A \,|\,i\rangle_B$$

and suppose they have a set of d^2 unitary transformations U_x generalizing the Pauli matrices.

What it means in this case to generalize the Pauli matrices is that all d^2 states

$$(U_x \otimes I)\,|\,\psi\rangle$$

form an orthonormal basis. This is clearly what you need for superdense coding. It suffices for teleportation as well.

Lemma:

$$(U \otimes I)\,|\,\psi\rangle = (I \otimes U^\dagger)\,|\,\psi\rangle$$

Proof: (dropping the normalization $1/\sqrt{d}$)

$$\begin{aligned}
{}_A\langle j\,|_B\langle k\,|\,U \otimes I\left(\sum_i |\,i\rangle_A\,|\,i\rangle_B\right) &= \sum_i \langle j\,|\,U\,|\,i\rangle\,\langle k|i\rangle \\
&= \langle j\,|\,U\,|\,k\rangle = \langle k\,|\,U^\dagger\,|\,j\rangle \\
&= \sum_i \langle j|i\rangle \sum_i \langle k\,|\,U^\dagger\,|\,i\rangle \\
&= {}_A\langle j\,|_B\langle k\,|\,I \otimes U^\dagger\left(\sum_i |\,i\rangle_A\,|\,i\rangle_B\right)
\end{aligned}$$

Now, Alice starts with $|\,\phi\rangle_A\,|\,\psi\rangle_{AB}$. She measures this state in the basis ${}_{AA}\langle\psi\,|\,(I\otimes U_x^\dagger)$. Thus, she will observe one of these states. What state does Bob now have?

$$ {}_{AA}\langle\psi\,|\,(I\otimes U_x^\dagger\otimes I)\,|\,\phi\rangle_A\,|\,\psi\rangle_{AB} = {}_{AA}\langle\psi\,|\,(I\otimes I\otimes U_x)\,|\,\phi\rangle_A\,|\,\psi\rangle_{AB} $$

Now, Alice can send the measured x to Bob, who can then apply $U_x^\dagger$ to his state, obtaining

$$ \begin{aligned} ({}_{AA}\langle\psi\,|)\,|\,\phi\rangle_A\,|\,\psi\rangle_{AB} &= \frac{1}{d}\sum_j {}_A\langle j\,|\,{}_A\langle j\,|\sum_i \phi_i\,|\,i\rangle_A \sum_k |\,k\rangle_A\,|\,k\rangle_B \\ &= \frac{1}{d}\sum_k \phi_k\,|\,k\rangle_B\,, \end{aligned} $$

as desired. The $\frac{1}{d}$ reflects the probability of $1/d^2$ of obtaining each U_x.

Teleporting through a gate

Suppose that we have a quantum gate G such that $GU_x = U_{x'}G$ for some x', for all of the U_x above. Then, if Alice and Bob start with the state

$$G\psi = \frac{1}{\sqrt{d}} G \sum_{i=1}^{d} |\, i\rangle_A \,|\, i\rangle_B\,,$$

and Alice also has the state $|\,\phi\rangle$. then Alice and Bob can apply the teleportation procedure, and Bob can correct his state using $U_{x'}$ instead of U_x. This gives Bob the state $G\,|\,\phi\rangle$.

This can be useful in certain situations. For instance, if the U_i are the Pauli matrices, and G is the controlled not gate,

$$\mathrm{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Then Alice and Bob can do the two-qubit CNOT gate using joint Bell state measurements and the one-qubit Pauli gates.

One of the nicest sets of generalizations of the Pauli matrices in d-dimensions is obtained by using R^aT^b, where $0 \le a, b \le d-1$.

$$R = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & e^{2i\pi/d} & 0 & \dots & 0 \\ 0 & 0 & e^{4i\pi/d} & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & e^{2(d-1)i\pi/d} \end{pmatrix} \qquad T = \begin{pmatrix} 0 & 1 & 0 & 0 & \dots \\ 0 & 0 & 1 & 0 & \dots \\ 0 & 0 & 0 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \\ 1 & 0 & 0 & 0 & \dots \end{pmatrix}$$

These have nice communtation relations, so $TR = e^{2\pi i/d}RT$.

When the quantum prime factorization algorithm was announced in 1994, one reaction was that this would never work because errors would inevitably disrupt the computation.

How bad is the situation?

To do 10^9 steps on a quantum computer, you need to do each step with inaccuracy less than 10^{-9}. This seems virtually impossible to experimental physicists.

The same objection was raised to scaling up classical computers in the 1950's.

Von Neumann showed that you could build reliable classical computers out of unreliable classical components.

Currently, we don't use many of these techniques because we have *extremely* reliable chips, so we don't need them.

Classical fault-tolerance techniques.

1) Consistency checks

2) Checkpointing

3) Error-correcting codes

4) Massive redundancy

No-cloning theorem:

You cannot duplicate an unknown quantum state.

Heisenberg uncertainty principle:

You cannot completely measure a quantum state.

Classical fault-tolerance techniques.

1) Consistency checks

    Doesn't get you very far (quantum or classical)

2) Checkpointing

    Doesn't work in quantum case

3) Error-correcting codes

    These work!

4) Massive redundancy

    Doesn't seem to get you very far in quantum case.

Best current results

If the quantum gates are accurate to reduce noise to 1 part in 30, you can make fault-tolerant quantum circuits (Knill).

This requires a ridiculous amount of overhead. Noise of 1 part in 1000 might lead to more reasonable overheads.

(These estimates may use some heuristic arguments; the best rigorous result I know of is 1 part in 10^6.)

The best upper bound is around 1 part in 5.

Open question: what is the right threshold?

Outline:

- Quantum error correcting codes
  - a simple example
  - CSS codes
  - quantum cryptography

The simplest classical error correcting code is the repetition code.

$$\begin{array}{ccc} 0 & \rightarrow & 000 \\ 1 & \rightarrow & 111 \end{array}$$

What about the quantum version?

$$\begin{array}{ccc} |\,0\rangle & \rightarrow & |\,000\rangle \\ |\,1\rangle & \rightarrow & |\,111\rangle \end{array}$$

This works against bit flips

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\sigma_x(2) : \begin{array}{ccc} |\,000\rangle & \rightarrow & |\,010\rangle \\ |\,111\rangle & \rightarrow & |\,101\rangle \end{array}$$

Can measure “which bit is different?”

Possible answers: none, bit 1, bit 2, bit 3.

Applying σ_x to incorrect bit corrects error.

This works for superpositions of encoded $|\,0\rangle$ and $|\,1\rangle$.

$$\sigma_x(2) : \alpha\,|\,000\rangle + \beta\,|\,111\rangle \rightarrow \alpha\,|\,010\rangle + \beta\,|\,101\rangle$$

When this is measured, the result is “bit 2 is flipped,” and since the measurement gives the same answer for both elements of the superposition, the superposition is not destroyed.

Thus, bit 2 can now be corrected by applying $\sigma_x(2)$.

Three-bit code

$$\begin{aligned} |\,0\rangle &\rightarrow |\,000\rangle \\ |\,1\rangle &\rightarrow |\,111\rangle \end{aligned}$$

What about a phase flip error

$$\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} ?$$

$$\begin{aligned} |\,E_0\rangle = |\,000\rangle &\rightarrow \quad |\,000\rangle = |\,E_0\rangle \\ |\,E_1\rangle = |\,111\rangle &\rightarrow -|\,111\rangle = |\,E_1\rangle \end{aligned}$$

A phase flip on any qubit gives a phase flip on the encoded qubit, so phase flips are three times as likely.

Our story so far.

The three-bit code corrects bit flips, but makes phase flips three times as more likely

The same thing happens for a general phase error

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}.$$

Another 3-qubit code

The unitary transformation $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ takes phase flips to bit flips and vice versa:

$$H \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} H = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Suppose we apply H to the 3 encoding qubits and the encoded qubit. What does this do to our code?

Get new code

$$\begin{aligned} |\,0\rangle &\rightarrow \frac{1}{2}\left(|\,000\rangle + |\,011\rangle + |\,101\rangle + |\,110\rangle\right) \\ |\,1\rangle &\rightarrow \frac{1}{2}\left(|\,100\rangle + |\,010\rangle + |\,001\rangle + |\,111\rangle\right) \end{aligned}$$

$$
\begin{aligned}
|\,0\rangle &\rightarrow \frac{1}{2}\left(|\,000\rangle + |\,011\rangle + |\,101\rangle + |\,110\rangle\right) \\
|\,1\rangle &\rightarrow \frac{1}{2}\left(|\,100\rangle + |\,010\rangle + |\,001\rangle + |\,111\rangle\right)
\end{aligned}
$$

Encoded $|\,0\rangle$: superposition of states with an even number of 1's.
Encoded $|\,1\rangle$: superposition of states with an odd number of 1's.

A bit flip on any qubit exchanges 0 and 1.

A phase flip on any qubit is correctable. E.g., $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ on bit 3.

$$
\sigma_z(3)\,|\,E_0\rangle = \frac{1}{2}\left(|\,000\rangle - |\,011\rangle - |\,101\rangle + |\,110\rangle\right)
$$

This is orthogonal to $\sigma_z(a)\,|\,E_b\rangle$ unless $a = 3$, $b = 0$.

So we can measure "which qubit has a phase flip?" and then correct this qubit.

The 9-qubit code

First quantum error correcting code discovered:

$$|\,0\rangle \rightarrow \frac{1}{2}(|\,000000000\rangle + |\,000111111\rangle + |\,111000111\rangle + |\,111111000\rangle)$$

$$|\,1\rangle \rightarrow \frac{1}{2}(|\,111000000\rangle + |\,000111000\rangle + |\,000000111\rangle + |\,111111111\rangle)$$

This code will correct any error in one of the nine qubits.

Two concatenated codes: the outer one corrects phase errors, and the inner one corrects bit errors.

If you have a bit flip: $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, it is corrected by comparison with the other two qubits in its group of three.

The 9-qubit code

$$|\,0\rangle \rightarrow \frac{1}{2}(|\,000000000\rangle + |\,000111111\rangle + |\,111000111\rangle + |\,111111000\rangle)$$

$$|\,1\rangle \rightarrow \frac{1}{2}(|\,111000000\rangle + |\,000111000\rangle + |\,000000111\rangle + |\,111111111\rangle)$$

If you have a phase flip on a single qubit: $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$, it gives the same result as a phase flip on any of the other qubits in the same group of three.

The correction works exactly as it does in the three-qubit phase-correcting code.

Objection: We've shown that our code can correct any one phase flip, or any one bit-flip. (And in fact, it can correct one phase flip and a separate bit flip, as these processes are independent.)

But there are an infinite number of kinds of error. How can we correct those?

Theorem: If you can correct a tensor product of t of any of the following three types of error

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

then you can fix any error restricted to t qubits.

Proof Sketch:
The identity matrix and σ_x, σ_y and σ_z for a basis for 2×2 matrices. One can thus decompose any error matrix into a sum of these four matrices. If the error only affects t qubits, it applies the identity matrix to the other qubits, so the decomposition never has more than t terms in the tensor product not equal to the identity.

Example in 3-qubit phase code

$$\begin{aligned} |\,0\rangle &\rightarrow \frac{1}{2}\,(|\,000\rangle + |\,011\rangle + |\,101\rangle + |\,110\rangle) \\ |\,1\rangle &\rightarrow \frac{1}{2}\,(|\,100\rangle + |\,010\rangle + |\,001\rangle + |\,111\rangle) \end{aligned}$$

Suppose we apply a general phase error $\begin{pmatrix} 1 & 0 \\ 0 & e^{2i\theta} \end{pmatrix}$ to qubit 1, say. Can we correct this?

Rewrite error as $\begin{pmatrix} e^{-i\theta} & 0 \\ 0 & e^{i\theta} \end{pmatrix}$

We can do this, since global phase changes are immaterial.

$$\begin{aligned} |\,E_0\rangle \rightarrow \quad & e^{-i\theta}\big(\,|\,000\rangle + |\,011\rangle\,\big) + e^{i\theta}\big(\,|\,101\rangle + |\,110\rangle\,\big) \\ = \quad & \cos\theta\big(\,|\,000\rangle + |\,011\rangle + |\,101\rangle + |\,110\rangle\,\big) \\ & -i\sin\theta\,\big(\,|\,000\rangle + |\,011\rangle\ -\ |\,101\rangle\ -\ |\,110\rangle\,\big) \end{aligned}$$

We applied an arbitrary phase error to qubit 1, on an encoded 0.

$$
\begin{aligned}
|\,E_0\rangle \rightarrow \qquad & e^{-i\theta}\big(\,|\,000\rangle + |\,011\rangle\,\big) + e^{i\theta}\big(\,|\,101\rangle + |\,110\rangle\,\big) \\
= \quad & \cos\theta\big(\,|\,000\rangle + |\,011\rangle + |\,101\rangle + |\,110\rangle\,\big) \\
& -i\sin\theta\big(\,|\,000\rangle + |\,011\rangle \;-\; |\,101\rangle \;-\; |\,110\rangle\,\big) \\
= \quad & \cos\theta\,|\,E_0\rangle - i\sin\theta\,\sigma_z^{(1)}\,|\,E_0\rangle
\end{aligned}
$$

When we measure "which bit has a phase flip," we get "bit 1" with probability $|\sin^2\theta|$. The state has "collapsed," so our measurement is now correct.

The same thing will happen to any superposition $\alpha\,|\,E_0\rangle + \beta\,|\,E_1\rangle$.

We have a 9-qubit code that can correct any error in 1 qubit. How can we make more general quantum codes?

Better classical codes exist than repetition codes.

The $[7, 4, 3]$ Hamming code, for example.

The codewords are the binary row space of

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

This code maps 4 bits to 7 bits. The minimum distance between two codewords is 3, so it can correct one error.

Generator matrix for Hamming code:

$$G = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

To correct errors, we use the parity check matrix H. This is a generator matrix of $\mathbf{C}^{\perp} = \{v : v \cdot w = 0 \text{ for all } w \in \mathbf{C}\}$.

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

$H \cdot v = 0$ if $v \in \mathbf{C}$. $H \cdot v$ is called the *syndrome*. This syndrome gives the location of the incorrect bits.

Quantum Hamming code

$$|\,0\rangle \rightarrow \frac{1}{\sqrt{8}}\begin{pmatrix} & |\,0000000\rangle & + & |\,1110100\rangle \\ + & |\,0111010\rangle & + & |\,0011101\rangle \\ + & |\,1001110\rangle & + & |\,0100111\rangle \\ + & |\,1010011\rangle & + & |\,1101001\rangle \end{pmatrix} \qquad (H)$$

$$|\,1\rangle \rightarrow \frac{1}{\sqrt{8}}\begin{pmatrix} & |\,1100010\rangle & + & |\,0110001\rangle \\ + & |\,1011000\rangle & + & |\,0101100\rangle \\ + & |\,0010110\rangle & + & |\,0001011\rangle \\ + & |\,1000101\rangle & + & |\,1111111\rangle \end{pmatrix} \qquad (H+\bar{1})$$

The bit flip errors are correctable, since all elements in these superpositions are in the Hamming code (if you measure the syndrome, you know which bit to correct).

The phase flip errors are correctable since applying $H = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ to all 7 encoding qubits and to the encoded qubit takes the code to itself.

You can thus correct bit flip errors by correcting phase flips in the Hadamard dual space.

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.

Suppose you have a set of errors $\mathcal{E}_1$, $\mathcal{E}_2$, ..., $\mathcal{E}_k$, and you want the code to correct them exactly. What properties do your codewords need to have?

Knill-Laflamme criterion.
Suppose we have a set of codewords $|\, w_1\rangle$, $|\, w_2\rangle$, ..., $|\, w_d\rangle$. Then we need

$$\langle w_k \,|\, \mathcal{E}_i^\dagger \mathcal{E}_j \,|\, w_l\rangle = \gamma_{i,j}\delta_{k,l}$$

How does this work?

Recall in the 9-qubit code $\sigma_z^{(1)^\dagger} \sigma_z^2 \,|\, w_k\rangle = |\, w_k\rangle$ for all codewords $|\, w_k\rangle$, so a phase error on the first qubit cannot be distinguished from a phase error on the second. However, this is O.K.; since they both have the same effect, they can be corrected using same protocol.

This is the worst that can happen on CSS codes or stabilizer codes (errors are either orthogonal, or have identical effects).

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.

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.

⤡	⤢	⤢	↔	⤡	↕	↕	⤢	⤡	↔	↔	↕
×	+	×	×	×	+	×	+	×	+	+	×
⤡	↔	⤢	⤡	⤡	↕	⤡	↕	⤡	↔	↔	⤢
•		•		•	•			•	•	•	
1		0		1	0			1	1	1	

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

$$\frac{1}{2}\,|\updownarrow\rangle\,\langle\updownarrow| + \frac{1}{2}\,|\leftrightarrow\rangle\,\langle\leftrightarrow| = \frac{1}{2}\,|\nearrow\!\!\!\!\swarrow\rangle\,\langle\nearrow\!\!\!\!\swarrow| + \frac{1}{2}\,|\nwarrow\!\!\!\!\searrow\rangle\,\langle\nwarrow\!\!\!\!\searrow|$$

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.

Alice and Bob take an n-bit key $\mathbf{k}$, apply a 0-1 $n \times m$ matrix to it, 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 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 $w \in \mathbf{F}_2^n/\mathbf{C}_2^{\perp}$, $z \in \mathbf{F}_2^n/\mathbf{C}_1$ 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, 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

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

$w \in \mathbf{F}_2^n / \mathbf{C}_2^\perp$ and $z \in \mathbf{F}_2^n / \mathbf{C}_1$ 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 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

$$\frac{|\mathbf{C}_1|}{2^{2n}} \sum_{\substack{w \in \mathbf{F}_2^n / \mathbf{C}_2^{\perp} \\ z \in \mathbf{F}_2^n / \mathbf{C}_1}} \sum_{x_1, x_2 \in \mathbf{C}_2} (-1)^{w \cdot (x_1 + x_2)} \, | \, v + x_1 + z \rangle \langle v + x_2 + z \, |$$

$$= \frac{|\mathbf{C}_1|}{2^n |\mathbf{C}_2|} \sum_{z \in \mathbf{F}_2^n / \mathbf{C}_1} \sum_{x \in \mathbf{C}_2} | \, v + x + z \rangle \langle v + x + z \, |$$

Since Alice chooses a random $v \in \mathbf{C}_1 / \mathbf{C}_2$, this state is indistinguishable from 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.

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.

Equivalence to BB84

Alice sends a random shift of the quantum code

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

Alice never needs to reveal w; since Bob wants to get the bits Alice sends and doesn't care about the phases, he doesn't need to correct the phase errors. Thus, the state Bob receives, for a given key v and a given shift z, is the average over w of the quantum states encoded above, or the density matrix

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

Bob receives

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

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

This average destroys all the phase information, and thus is the same as the average over $x \in \mathbf{C}_2$ of $|\, v + x + z \rangle$. Now, we see the equivalence to BB84. The code $\mathbf{C}_1$ is the error correcting code used in BB84 to fix hash errors. The code $\mathbf{C}_2$ corresponds to the linear hash function in BB84, as adding any vector in $\mathbf{C}_2$ gives the same key, and is thus equivalent to multiplying by a generator matrix for $\mathbf{C}_2^\perp$.

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.