Notes 8.370/18.435 Fall 2021

Lecture 31 Prof. Peter Shor

Today we will explain the BB84 key distribution protocol, and give the proof of its security based on quantum error-correcting codes.

First, I'm going to say a few things about key distribution protocols and cryptography in general. Then, I'll explain the BB84 key distribution protocol. Next, I'll explain a key distribution protocol based on quantum error correcting codes, which has a fairly straightforward proof that it is secure. Finally, I'll explain why the two protocols are essentially equivalent, so if the QECC-based protocol is secure, then BB84 is.

BB84 is a key distribution protocol; it's named after its inventors, Charlie Bennett and Gilles Brassard, and the year it was invented. It was based on some ideas that Stephen Wiesner had in 1969. Wiesner wrote the paper up and sent it to a journal, at which point it was rejected, and he gave up. The paper was eventually published in 1983 when Charlie Bennett sent it to a theoretical computer science newsletter whose editor he knew.

What is a key distribution protocol? We've seen one before in this class—the Diffie-Hellman key distribution protocol, which is based on the hardness of the discrete log problem. The basic idea is that two participants, Alice and Bob, want to agree on a secret key that an eavesdropper, Eve, does not know. We assume that Alice and Bob start the protocol without any secret information in common, so they have to agree on a secret key using a public channel. Classically, the only way to do this is to base the protocol on a hard problem, which we assume that the eavesdropper cannot solve. Quantum mechanically, however, we don't need to make any hardness assumptions—BB84 is secure as long as the laws of quantum mechanics hold.

We will be making some assumptions about the communication channels we use in the protocol, so first we're going to give the reasons for these assumptions. One thing that you have to guard against is the man-in-the-middle attack. Suppose that Eve cuts the channels, and inserts herself in the middle

$$\text{Alice} \longleftrightarrow \text{Eve} \longleftrightarrow \text{Bob}$$

Now, when Eve talks to Alice she pretends to be Bob, and when she talks to Bob, she pretends to be Alice. Since Alice and Bob don't know any secret identifying information about each other, they cannot uncover her impersonation, and she learns everything they say to each other.

Our assumptions will be that Alice and Bob have a classical channel that is not spoofable—that is, Eve cannot take over the channel and pretend to be Bob, and a quantum channel on which Eve is allowed to do anything that is consistent with the laws of physics. Since Eve has complete control of the quantum channel, she can cut it and prevent Alice and Bob from communicating at all. So what we will require from the protocol is that it is very unlikely that Alice and Bob think they've successfully shared a secret key while Eve has more than an exponentially small amount of information about that secret key.

The advantage that quantum key distribution has over classical key distribution protocols is that classical key distribution protocols must rely on some hard problem, and we have no guarantee that a very clever mathematician won't come up with a way to solve (say) the discrete log problem in polynomial time tomorrow. The disadvantage of quantum key distribution is that you need a quantum channel between the two participants. This is possible over an optical fiber. There are lots of these already in the ground, but there are extra requirements for them to be used for key distribution—you can't have an amplifier on the optical fiber between the two people who want to use the BB84 protocol, because amplifiers destroy quantum coherence. With current technology, this limits the distance that quantum key distribution can be used on optical fiber to a few hundred kilometers (and good luck finding an optical fiber that long that's already in place which doesn't have amplifiers on it; if you want to use quantum key distribution over existing optical fibers, your distance limitations will be substantially shorter).

1 BB84

So what is the BB84 protocol? We will give the original BB84 protocol; there have been many different quantum key distribution protocols proposed since them, and some of them have substantial practical advantages over BB84. We first explain how it works when the state preparations and the measurements are perfect, and the channel between Alice and Bob is noiseless.

1. Alice sends Bob a sequence of qubits that she has prepared in one of the four states $|\,0\rangle$, $|\,1\rangle$, $|+\rangle$, $|-\rangle$:

Alice prepares	$\|\,0\rangle$	$\|\,1\rangle$	$\|+\rangle$	$\|+\rangle$	$\|\,0\rangle$	$\|-\rangle$	$\|-\rangle$	$\|\,1\rangle$	$\|\,1\rangle$	$\|\,0\rangle$	$\|\,0\rangle$

2. Bob measures the qubits he gets in a random basis:

Bob measures	$0/1$	$0/1$	$0/1$	$+/-$	$+/-$	$0/1$	$+/-$	$0/1$	$+/-$	$0/1$	$0/1$
and gets	$\|\,0\rangle$	$\|\,1\rangle$	$\|\,1\rangle$	$\|+\rangle$	$\|+\rangle$	$\|\,1\rangle$	$\|-\rangle$	$\|\,1\rangle$	$\|+\rangle$	$\|\,0\rangle$	$\|\,0\rangle$

3. At this point, Alice announces her basis, and Bob tells Alice which ones agree. They discard the measurement results for bases that disagree.

Alice′s basis	$0/1$	$0/1$	$+/-$	$+/-$	$0/1$	$+/-$	$+/-$	$0/1$	$0/1$	$0/1$	$0/1$
places they agree	$\|\,0\rangle$	$\|\,1\rangle$		$\|+\rangle$			$\|-\rangle$	$\|\,1\rangle$		$\|\,0\rangle$	$\|\,0\rangle$

4. Now, Alice (or Bob) announces a random sample of the qubits to use to check whether they agree. If they do, they know that Eve couldn't have been measuring many of the qubits. They turn the remaining qubits into a secret key, using (say) the mapping of $|\,0\rangle$ and $|+\rangle$ to 0 and $|\,1\rangle$ and $|-\rangle$ to 1:

check qubits		?		?						?	
Alice		1		+						0	
Bob		1		+						0	
secret key	0						1	1			0

The reason that this protocol works is that if Eve tries to measure a qubit, she doesn't know whether to measure in the $0/1$ basis or the $+/-$ basis. If she chooses the wrong basis, then she will disturb the quantum state that Alice sent, and Alice and Bob will notice that some of their check bits disagree.

But what if their channel is noisy? Some of Alice's and Bob's string of bits will disagree anyway, so how can they tell whether Eve is eavesdropping? How can they get a string of bits that they agree on after that? And even if they do, how can they ensure that Eve doesn't have any information about this secret key.

The first problem is solved by using error correcting codes. Suppose Alice and Bob have strings a and b of length m. Because they tested their check bits, they know that they expect around ϵm of their bits to differ, where ϵ is relatively small. Now, Alice chooses an error correcting code C of length m that will correct $\epsilon' m$ bits, where $\epsilon' > \epsilon$, so that even accounting for random fluctuations in the noise, the number of places where Alice and Bob's bits differ is less than $\epsilon' m$ with high probability. Alice then chooses a random codeword $c \in C$, and sends

$$a + c$$

to Bob. Bob takes this message and subtracts b from it to get $a - b + c$. This is a string that differs from the codeword c in fewer than $\epsilon' m$ positions, so Bob can apply error correction and get c. Alice and Bob then share c, and Eve does not know what c is. Why not? Because a was essentially random, $a + c$ is also random. Since this is the only information Eve sees about a and c, she should not have any information on what c is. (You should note that this is not a rigorous proof; it took a decade and a half after BB84 was proposed to get a rigorous proof that it was secure.)

Finally, it's possible that after this protocol, Eve has some information about c. To fix this, Alice and Bob choose a hash function f that maps m bits into ℓ bits where $\ell < m$, If this hash function is sufficiently random, and ℓ is sufficiently shorter than m, then a theorem from classical cryptography says that the information that Eve has about $f(c)$ is much less than the information Eve has about c. In this protocol, to make the proof work, we will assume that f is a linear function, so $f(c) = Mc$ for some binary matrix M.

2 The adapted Lo-Chao protocol

We now give a protocol for which the security proof is relatively simple. The difference between this and the original BB84 protocol is that for this one, Bob needs quantum memory to store all the qubits he receives in, and thus it is not currently practical, and it's not going to be as cheap to implement as BB84 in the foreseeable future. However, we will then show that these protocols are equivalent in that if somebody can break BB84, they can also break this protocol. The protocol is based on one published by Lo and Chau.

The idea behind this protocol is that if Alice and Bob share perfect EPR pairs, then measuring them in the $0/1$ basis will give them a secret key. Eve can never determine the outcomes of their measurements; since Alice and Bob have perfect entanglement, then Eve cannot be entangled with their state, and Eve's state will remain uncorrelated with Alice and Bob's secret key. There is a theorem called "monogamy of entanglement" which says that the more you are entangled with one system, the less entanglement you can have with any other system. This theorem gives intuition for why this QKD protocol works.

So what is the Lo-Chau protocol?

1. Alice prepares n EPR pairs.

2. Alice chooses a CSS code CSS($C_1 : C_2$) and a translate of it by s in bit space and t in phase space.

3. Alice encodes half of each EPR pair with this code, randomly intersperses test bits which are equally likely to be in one of the four bases $|\,0\rangle$, $|\,1\rangle$, $|+\rangle$, $|-\rangle$, and sends this string of qubits to Bob.

4. Bob puts everything into his quantum memory.

5. Alice announces the code and the strings s and t it was translated by, which bits were test bits, which bits were code bits, and the values of the test bits.

6. Bob checks the test bits to determine the error rate. He then decodes the EPR pairs, and Alice and Bob measure each EPR pair in the $0/1$ basis to obtains a secret key.

The first thing to note is that because Alice sends a random translate of CSS(C_1, C_2), and because the test qubits are equally likely to be in any of the four states, the density matrix that Eve sees is completely random; i.e., is the identity matrix. Thus, Eve cannot tell which of the qubits are code qubits and which are test qubits, so the noise rate she induces on the test qubits will also with high probability be induced on the code qubits.

Now, because the rate of noise on the test bits is sufficiently low, the probability that the CSS code does not transmit the correct state is ϵ, where ϵ can be made exponentially small. The state that Alice and Bob share after the transmission is then

$$\sqrt{1-\epsilon}\,|\,\phi\rangle^{\otimes n} + \sqrt{\epsilon}\,|\,E\rangle$$

where $|\,\phi\rangle$ is an EPR pair and $|\,E\rangle$ is an arbitrary error state.

So how much information can Eve find about the EPR pairs in this scenario? Suppose that Alice and Bob got together and measured their state. Then with probability $1-\epsilon$, they would have $|\,\phi\rangle^{\otimes n}$ and with probability ϵ, they would have something else. Eve can have information about their secret key only with probability ϵ. If ϵ is exponentially small, Eve has an exponentially small amount of information about the key (To state this formally and prove it would require explaining more information theory than I want to do right now.) Thus, the Lo-Chau protocol works.

3 The equivalence of the protocols

Now, let's show that these protocols are equivalent. In the second protocol, we can assume that Alice measures her half of the EPR pairs before she encodes the quantum state, because the operation of measuring commutes with the operation of encoding and sending the other half of the EPR pairs. Thus, we can assume that the quantum state that she sends is a random string of n classical bits which is encoded in the CSS code CSS$(C_1 : C_2)$.

So what happens when Alice encodes a random string of bits to encode. What she is essentially doing is choosing a random coset $x + C_2$ and encoding it. But choosing a random coset x is exactly the same as choosing a random bit string y and taking the coset $y + C_2$. When it's encoded by the shifted CSS code, it will look like

$$\frac{1}{|C_2|^{1/2}} \sum_{c_2 \in C_2} (-1)^{t\cdot(y+c_2)} \, |\, s + y + c_2 \rangle$$

For the secret key, Bob needs to find the coset of C_2 that this belongs to. He can find the coset by measuring this string in the $|\,0\rangle\,, |\,1\rangle$ basis, and subtracting s to get $y + c_2$. Now, note that Bob doesn't actually need t to find this coset, so we can assume that Alice never sends him t. If Alice doesn't send him t, the density matrix of her message when you take the average over t is

$$\frac{1}{|C_2|} \left(\sum_{c_2 \in C_2} (-1)^{t\cdot(y+c_2)} \, |\, s + y + c_2 \rangle \right) \left(\sum_{c_2 \in C_2} (-1)^{t\cdot(y+c_2)} \, \langle s + y + c_2 \,| \right)$$

$$= \frac{1}{C_2} \sum_{c_2 \in C_2} |\, s + y + c_2 \rangle\langle s + y + c_2 \,|\,,$$

which is the same as Alice taking a random c_2 and adding it to $s + y$.

Because Bob and Alice are communicating over a noisy channel, Bob actually gets $s + y + c_2 + e$, where e is some error. He then needs to subtract s and apply the classical decoding procedure to get $y + c_2$.

So let's compare the protocols. In both cases, test qubits are interspersed with the code qubits. In BB84, Alice sends Bob a on the quantum channel. Bob receives $b = a + e$. Alice then sends Bob $a + c_1$ on the classical channel. Bob subtracts these two quantities to get $c_1 + e$ and decodes it to the codeword $c_1 \in c_1$.

In the second protocol, Alice sends Bob $s + y + c_2$, where $y + c_2 \in C_1$. Bob receives $s + y + c_2 + e$. Alice then sends Bob s. Bob now subtracts these two quantities to obtain $y + c_2 + e$ and decodes it to get a codeword of C_1.

These two protocols look a lot alike. In both cases, Alice sends Bob a random string on the quantum channel. Then she sends to him over the classical channel a string that differs from the first sting by a random codeword of the code C_1. The secret message (before privacy amplification) is the codeword of C_1. Thus, by equating $a = s + y + c_2$ and $a + c_1 = s$, we can show that the two protocols are completely equivalent.

Now, let's deal with the privacy amplification. In the BB84 protocol, Alice and Bob get the codeword $c \in C_1$ that they will use to derive their the secret key. They then apply a linear hash function to the codeword to amplify the privacy. If you think about it, the only thing that matters about the linear hash functions is which codewords in c_1 get mapped to the same string $f(c_1)$. Because f is a linear hash function, i.e., $f(x) = Mx$ for some matrix M, the codewords that get mapped to 0 (call these codewords

S) are a linear subspace of C_1, and the codewords that get mapped to any other value are a coset of S in C_1. Thus, we can take S to be C_2 — a linear code is just a linear subspace, and random codes (like those generated by random matrices M) are highly likely to be good error correcting codes.

We have thus shown that BB84 is equivalent to our other key distribution protocol based on Lo and Chau's ideas, so BB84 is secure.