Lecture 18: Quantum Key Distribution

March 28, 2006

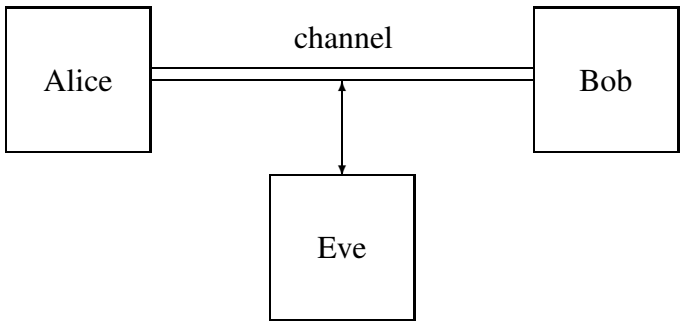
In this lecture we will begin discussing quantum cryptography. There are indeed many aspects to cryptography, and many different cryptographic tasks that one may consider. Likely the first cryptographic task that comes to mind is *private communication*: Alice and Bob wish to communicate, and they do not want an eavesdropper (Eve) to learn any information about their communication. There are many other cryptographic tasks or protocols one may consider, such as *message authentication*, *digital signatures*, and *voting schemes*. Often these tasks are broken down into more primitive operations, such as *bit-commitment* and *oblivious transfer*.

In this lecture we will just consider private communication, and will discuss bit commitment in the next lecture.

A perfectly secure cryptosystem

Let us first consider a purely classical situation. Suppose that Alice and Bob can communicate through a classical channel, and a third party Eve is able to monitor the communications on that channel. Alice wishes to transmit a particular message to Bob, and wants Eve to learn as little as possible about it. Eve wants to learn as much as possible about the message.

In cryptography we always need to be precise about our assumptions and model. In this example, we will assume that the channel Alice and Bob are communicate through is an *authenticated* channel: when Bob receives a message, he can verify that it came from Alice (and vice versa). We will also assume that Eve cannot jam the channel or modify messages sent over it, meaning that Alice and Bob always receive one another's messages without error.



There exists a provably secure classical scheme, known as the *one-time pad*, for accomplishing the task at hand. It requires that Alice and Bob share a sequence of random bits, called a *private key*, that is secret from Eve. This private key could have been created a long time ago, and is completely independent of the message Alice wishes to send. The key must be of the same length as the message. The scheme works as follows.

One-time Pad

Alice and Bob share a private key $K \in \{0,1\}^n$.

Alice wishes to send Bob a message $M \in \{0,1\}^n$. She computes $E = K \oplus M$ (bitwise exclusive OR) and sends E (the encrypted message) to Bob.

Bob receives E, and computes $E \oplus K$. The result is $E \oplus K = M \oplus K \oplus K = M$.

Eve can learn nothing (aside from the length of the message) by looking at E. If K is uniformly distributed, then so too is E.

One very important note is that the key K can only be used once (hence the name "one-time pad"). If they were to use it twice with two different messages, the encrypted texts would be

$$E_1 = M_1 \oplus K$$
$$E_2 = M_2 \oplus K.$$

Eve could then compute

$$E_1 \oplus E_2 = M_1 \oplus M_2,$$

which could contain a lot of information about M_1 and M_2. This could also allow Eve to learn something about K that could be exploited if Alice and Bob were to use the key a third time.

The one-time pad is unbreakable (if used properly), but it is very inefficient or impossible in many situations. For instance, a single entity such as a bank or online business may wish to communicate securely with many different individuals, and may not have met with them previously to exchange a key. It is impossible for two parties to classically generate a secure private key over a public channel.

Quantum key distribution

The aim of quantum key distribution is to allow Alice and Bob to generate a secure private key that can be used for the one-time pad without having to meet privately. They will be able to accomplish this task by using quantum information. There are a few different schemes for doing this, such as:

1. BB84. Easy to implement, relatively hard to prove security.
2. B92. Similar to BB84.
3. Lo-Chau. Hard to implement, easy to prove security.

We will only look at BB84, so named because its inventors Charles Bennett and Gilles Brassard proposed it in 1984. Proofs of security actually took a long time; Mayers gave a (complicated) proof in 1998, and Shor and Preskill gave a fairly simple proof based on the theory of quantum error correcting codes in 2000.

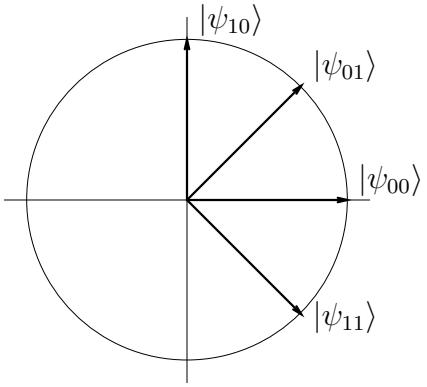
Once again, we need to be precise about our assumptions. We will assume that as before, Alice and Bob can communicate over a classical channel that is authenticated and cannot be tampered

with, but that is readable by Eve.[1] We will call this the *public* channel. They can also send qubits over a quantum channel. Eve has full access to this channel, meaning that she can intercept messages sent by Alice or Bob, perform measurements or quantum computations on these messages, and transmit the modified qubits to whichever party was to receive them in the first place.

The BB84 Key Exchange Protocol (first stage)

Alice randomly generates two strings of bits $x, y \in \{0,1\}^m$.

Define $|\psi_{00}\rangle = |0\rangle$, $|\psi_{10}\rangle = |1\rangle$, $|\psi_{01}\rangle = |+\rangle$, and $|\psi_{11}\rangle = |-\rangle$. These four states may be pictured as follows:



Alice prepares m qubits in the state

$$|\psi_{x,y}\rangle = |\psi_{x_1 y_1}\rangle \, |\psi_{x_2 y_2}\rangle \cdots |\psi_{x_m y_m}\rangle$$

and sends these m qubits over a quantum channel to Bob.

Bob receives m qubits, although they may not longer be in the state $|\psi_{x,y}\rangle$ because Eve may have tampered with them, or possibly the channel is noisy. (To understand how the protocol works, it is helpful to imagine first that Eve is not present and the channel is not noisy, so Bob receives precisely the state $|\psi_{x,y}\rangle$.)

Bob randomly chooses $y' \in \{0,1\}^m$, and measures each qubit received from Alice as follows:

- If $y'_i = 0$, Bob measures qubit i (with respect to the standard basis).
- If $y'_i = 1$, Bob performs a Hadamard transform on qubit i and then measures it with respect to the standard basis. (Equivalently, Bob measures qubit i with respect to the *diagonal basis* $\{|+\rangle, |-\rangle\}$, interpreting the result as 0 or 1, respectively.)

Let $x' \in \{0,1\}^m$ be the string corresponding to the results of Bob's measurements. The important thing to note at this point is that if $y_i = y'_i$ for some i and there was no noise or eavesdropping, then it is certain that $x_i = x'_i$.

[1] It is not a trivial assumption that the classical channel is authenticated, and this assumption represents an inevitable weakness of quantum key distribution. In the lecture I have briefly discussed how authenticated channels can be implemented using very short private keys (that only need to remain private for the duration of the protocol).

Finally, Alice and Bob publicly compare y and y'. They discard all bits x_i and x'_i for which $y_i \neq y'_i$. The remaining bits of x and x' represent a "semi-private" (and possibly noisy) key that will go into the next stage of the protocol.

End of protocol (stage 1).

Example 1. Suppose $m = 8$. Alice randomly chooses

$$x = 01110100$$
$$y = 11010001.$$

She sends $|\psi_{x,y}\rangle$ to Bob.

Suppose there is no tampering of the message, so Bob receives precisely $|\psi_{x,y}\rangle$. He randomly chooses

$$y' = 01110110$$

and measures. The first qubit is in state $|\psi_{01}\rangle = |+\rangle$, and because $y'_1 = 0$ Bob measures in the standard basis. He will get a uniform random bit; say $x'_1 = 1$. The second qubit is in state $|\psi_{11}\rangle = |-\rangle$, and because $y'_2 = 1$ Bob measures in the diagonal basis. He gets $x'_2 = 1$ with certainty this time. Continuing in this way, we might obtain the following table:

x	y	x'	y'
0	1	1	0
1	1	1	1
1	0	0	1
1	1	1	1
0	0	0	0
1	0	1	1
0	0	1	1
0	1	1	0

Alice and Bob publicly compare y and y'. They agree at positions 2, 4, and 5, so they keep the bits of x and x' at these positions and discard the rest:

x	y	x'	y'
~~0~~	~~1~~	~~1~~	~~0~~
1	1	1	1
~~1~~	~~0~~	~~0~~	~~1~~
1	1	1	1
0	0	0	0
~~1~~	~~0~~	~~1~~	~~1~~
~~0~~	~~0~~	~~1~~	~~1~~
~~0~~	~~1~~	~~1~~	~~0~~

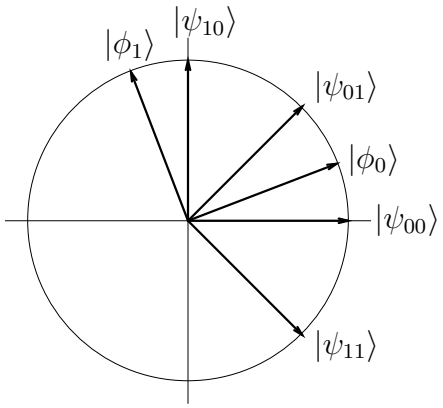
The three remaining bits of x and x' agree.

Of course the difficulty arises when Eve tries to extract information about x. The general principle working for Alice and Bob is the **uncertainty principle**—Eve cannot learn something about $|\psi_{x,y}\rangle$ without disturbing it. In particular, Eve does not know y, so she cannot hope to learn the bits of x without making some mistakes and causing some disturbance to $|\psi\rangle$.

For example, suppose that $y_i = y'_i$ for some index i, but Eve decided to measure qubit number i. Perhaps she measures the qubit in the *Breidbart basis*

$$|\phi_0\rangle = \cos(\pi/8)\,|0\rangle + \sin(\pi/8)\,|1\rangle\,, \quad |\phi_1\rangle = -\sin(\pi/8)\,|0\rangle + \cos(\pi/8)\,|1\rangle\,,$$

which is pictured as follows:



This is a good basis for Eve to choose—she will learn x_i with probability $\cos^2(\pi/8) \approx 0.85$. However, this measurement will cause the i-th qubit to be in state $|\phi_0\rangle$ or $|\phi_1\rangle$ after Eve's measurement. When Bob then measures, he will then get $x'_i \neq x_i$ with probability $\sin^2(\pi/8) \approx .15$, even though he should get $x_i = x'_i$ with certainty.

Informal and unsubstantiated claim: the more Alice learns about x, the more positions in which x and x' will disagree.

The difficulty in proving security in general is that Eve may use any strategy allowed by quantum mechanics—it is not sufficient to consider just a single attack such as measuring each qubit in the Breidbart basis.

Second stage of protocol.

Alice and Bob now need to estimate how much Eve might know about x and x'. They do this by sacrificing some of the remaining bits of x and x', say one-half of them (selected randomly). By comparing these bits publicly, they can estimate the error rate with high accuracy, and if it is too large they abort (Eve potentially has too much information for them to succeed). The maximum error rate that can be tolerated is about 11%.

If they have an acceptable error rate, Alice and Bob will have two strings x and x' (which now include only the bits that they did not sacrifice to estimate the error) that agree in a high percentage of positions with high probability. They have some bound on the amount of information Eve possesses about these strings.

Classical cryptography can now take over. They perform:

1. Information reconciliation.
2. Privacy amplification.

Information reconciliation is essentially distributed error correction that corrects x and x' so that they agree in all positions with very high probability. This may leak some information about these strings, but not too much. Privacy amplification transforms a shared string about which Eve has some bounded amount of information and compresses it to a shared string about which Eve has almost no information. Basically, Alice and Bob apply some suitable hash function to their shared strings.

End of protocol (stage 2).

There has been a lot of work done on information reconciliation and privacy amplification, but we will not have time to discuss these things in detail. In the lecture I have discussed some simplified versions of these problems to give you the feel for how they might work, but I will not include these examples in the notes.