Example of Quantum Key Distribution

I. Case with no eavesdropper

Alice sends a sequence of qubits (e.g., single photon polarizations) to Bob. For each qubit, she chooses a random bit (0 or 1) to transmit, and randomly selects either the X basis ($|\pm\rangle = \sqrt{1/2}(|0\rangle \pm |1\rangle)$) or Z basis ($|0\rangle$ or $|1\rangle$) in which to encode it. Bob randomly chooses to measure in either the X or Z basis. If there is no eavesdropper, and no noise in the channel, then Bob should always receive the same bit that Alice sent in the cases when they used the same basis. When they used different bases, Bob receives a random bit.

The table below gives an example of what such an error-free transmission should look like. Alice sent 32 qubits to Bob. The numbers marked in blue indicate those qubits where Bob chose the same basis as Alice. (The bases were chosen randomly, but it turned out that Bob guessed correctly 16 times, exactly half.)

Alice	X	X	Z	X	X	X	X	Z	X	Z	Z	Z	X	X	Z	Z	X	X	Z	Z	Z	X	Z	Z	X	Z	X	X	Z	X	Z	Z
	1	1	0	0	0	1	0	0	1	0	0	1	1	1	1	0	1	0	1	1	0	0	0	0	0	0	0	1	0	1	0	1
Bob	Z	Z	X	X	X	X	X	X	X	Z	Z	X	X	Z	X	Z	X	Z	Z	X	X	X	X	Z	Z	Z	Z	Z	X	Z	Z	Z
	0	1	0	0	0	1	0	1	1	0	0	1	1	0	1	0	1	0	1	0	1	0	0	0	1	0	1	1	1	0	0	1

II. Case with an eavesdropper

Now suppose that Eve intercepts each qubit before Bob receives it. She doesn't know what basis Alice used, so she randomly selects either X or Z to measure each qubit, then prepares a new qubit in the measured state and sends it on to Bob. If Eve guessed the same basis as Alice, she will learn the value of the bit that Alice sent, and if Bob also guesses that basis, she will know one bit of their shared key. If Eve guessed the wrong basis, she will get a random bit value, and so will Bob.

The table below now shows what would happen with the same stream of 32 qubits sent from Alice to Bob if Eve intercepts all of the qubits. The 10 numbers marked in blue are qubits where Bob chose the same basis as Alice and received the same bit that she sent; the 6 numbers marked in red are those qubits where Bob chose the same basis as Alice but received an incorrect bit value, due to Eve's measurement. The 8 numbers that are underlined are the bit values from the shared key that Eve learned correctly.

Alice	X	X	Z	X	X	X	X	Z	X	Z	Z	Z	X	X	Z	Z	X	X	Z	Z	Z	X	Z	Z	X	Z	X	X	Z	X	Z	Z
	1	1	0	0	0	1	0	0	1	0	0	1	1	1	1	0	1	0	1	1	0	0	0	0	0	0	0	1	0	1	0	1
Eve	X	Z	Z	X	Z	Z	X	X	Z	X	Z	X	Z	Z	X	X	X	X	Z	X	Z	X	X	X	Z	Z	X	X	Z	X	Z	X
	1	0	0	0	0	1	0	0	1	0	0	1	1	0	0	1	1	0	1	1	0	0	1	1	0	0	0	1	0	1	0	0
Bob	Z	Z	X	X	X	X	X	X	X	Z	Z	X	X	Z	X	Z	X	Z	Z	X	X	X	X	Z	Z	Z	Z	Z	X	Z	Z	Z
	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	1	1	1	0	1	1	0	0	1	1	1	0	0	0

III. Detecting Eve's presence.

Eve's measurements introduced errors into Alice and Bob's shared key: qubits where they measured in the same basis but got different bit values. In the example above, this happened in 6 out of the 16 bits, which is a bit higher than average. (The average would be 4 erroneous bits out of 16.)

Alice and Bob can detect the eavesdropper by choosing a random sample of their bits and comparing them publicly. (Since these bit values have been announced publicly, they cannot be used for secret key, and must be discarded.)

For this example suppose Alice and Bob randomly select the 4 circled numbers to compare:

Alice	X	X	Z	X	X	X	X	Z	X	Z	Z	Z	X	X	Z	Z	X	X	Z	Z	Z	X	Z	Z	X	Z	X	X	Z	X	Z	Z
	1	1	0	0	0	1	0	0	1	0	0	1	1	1	1	0	1	0	1	1	0	0	0	0	0	0	0	1	0	1	0	1
Eve	X	Z	Z	X	Z	Z	X	X	Z	X	Z	X	Z	Z	X	X	X	X	Z	X	Z	X	X	X	Z	Z	X	X	Z	X	Z	X
	1	0	0	0	0	1	0	0	1	0	0	1	1	0	0	1	1	0	1	1	0	0	1	1	0	0	0	1	0	1	0	0
Bob	Z	Z	X	X	X	X	X	X	X	Z	Z	X	X	Z	X	Z	X	Z	Z	X	X	X	X	Z	Z	Z	Z	Z	X	Z	Z	Z
	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	1	1	1	0	1	1	0	0	1	1	1	0	0	0

Two of these bits contain errors--they do not match between Alice and Bob. This allows Alice and Bob to detect the presence of the eavesdropper.

IV. Example of privacy amplification

If Eve intercepts every qubit, there is nothing Alice and Bob can do to share a secret key, but they will be aware of her presence and will not use an insecure key to send messages. If, however, Eve intercepts only a fraction p of the qubits, then Alice and Bob can try to extract a smaller, secure key by *privacy amplification*. Let's see an example of how this works.

For each qubit that Eve intercepts, there is a probability 1/2 that she will guess the correct basis to measure in. In this case, she will learn the bit value that Alice sent, and not introduce any errors in what Bob measures. However, if she guesses *in*correctly, she will get a random bit value, and if Bob measures in Alice's basis he receives a random bit. So if Eve intercepts a fraction p of the qubits, then she will know a fraction $p/2$ of the key bits shared by Alice and Bob, and a fraction $p/4$ of the bits will have errors (i.e., will not match between Alice and Bob). By comparing (and discarding) a sample of their shared bits and counting the number of mismatches, Alice and Bob can estimate the fraction p.

Suppose that Alice has sent a stream of qubits to Bob; they have compared the bases that they used, and dropped all the results where the bases didn't match. They have compared a random sample, and estimate that Eve intercepted a fraction $p = 1/4$ of the qubits sent.

Here are the 32 key bits possessed by Alice and Bob after carrying out the steps above:

Alice	0	1	0	0	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	1	1	0	0	1	1	1	1	0	1
Bob	0	1	1	0	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	1	1	0	0	1	0	1	1	0	1

The 2 bits marked in red contain errors. The 4 underlined bits are known to Eve. (Of course, Alice and Bob have no idea which bits have errors, or which ones are known to Eve, but they can estimate the likely fraction of both from the number of errors they observed in the sample they compared.)

Alice and Bob now want to *distill* a shorter key, with much lower probability of error, and much lower probability that any bits are known to Eve. They do this in two steps: *error correction*, and *privacy amplification*. For this example, we will illustrate the simplest versions of these procedures.

A. Error correction

Alice and Bob agree on a random grouping of their 32 bits into pairs: {{1, 6}, {2, 3}, {5, 29}, {7, 21}, {9, 10}, {11, 13}, {12, 15}, {14, 20}, {17, 19}, {18, 28}, {8, 23}, {22, 30}, {24, 25}, {26, 27}, {4, 31}, {16, 32}}. For each pair, Alice and Bob each calculate the parity of their two bits and announce the result publicly. If the two parities don't match, there must be an error, and they discard the error. If the two parities do match, either both bits are correct, or both bits are wrong (with probability $p^2/16$). In this case, they randomly choose one of the bits of the pair to keep, and discard the other one.

Let's see how this works in a few cases:

	1	6	Parity	Keep bit?
Alice	0	1	1	1
Bob	0	1	1	

	2	3	Parity	Keep bit?
Alice	1	0	1	Discard both
Bob	1	1	0	

	5	29	Parity	Keep bit?
Alice	1	1	1	29
Bob	1	1	1	

	7	21	Parity	Keep bit?
Alice	0	1	1	21
Bob	0	1	1	

Etc., for all 16 pairs. After this procedure, Alice and Bob will be left with 14 bits:

Alice	0	1	1	1	0	1	0	0	0	0	0	1	0	0
Bob	0	$\underline{1}$	1	$\underline{1}$	0	1	0	$\underline{0}$	0	0	$\underline{0}$	1	0	0

Notice that all errors have been eliminated. The rate of errors after one step of error correction is $p^2/16 = 1/256 = 0.00390625$, so we would expect there to be much less than one erroneous bit on average. For a larger (more realistic) key, more than one step might be necessary.

B. Privacy Amplification

However, Eve still knows 4 of the remaining 14 bits. (In fact, the *fraction* of bits that Eve knows actually goes up during error correction, to roughly p.) To eliminate this, Alice and Bob next do *privacy amplification*.

Alice and Bob again agree on a random pairing of their 14 bits: {{1, 3}, {2, 7}, {5, 13}, {4, 10}, {9, 11}, {8, 12}, {6, 14}}. For each pair of bits, they separately calculate the parity and retain that value in place of the pair. Let's see how this works:

	1	3	Parity
Alice	0	1	1
Bob	0	1	1

	2	7	Parity
Alice	1	0	1
Bob	$\underline{1}$	0	1

	5	13	Parity
Alice	0	0	0
Bob	0	0	0

	4	10	Parity
Alice	1	0	1
Bob	$\underline{1}$	0	1

	9	11	Parity
Alice	0	0	0
Bob	0	0	0

	8	12	Parity
Alice	0	1	1
Bob	0	1	1

	6	14	Parity
Alice	1	0	1
Bob	1	0	1

Notice that in four of these cases, Eve knows 1 of the 2 bits in the parity. However, to know the parity, she would need to know *both* of the bits. So after this step of privacy amplification, Eve actually has no knowledge of the shared key. In general, the fraction of bits that Eve knows will go from approximately p to approximately p^2. (If Alice and Bob want to be more confident that Eve does not know any bits, they could do another round of privacy amplification, at which point the fraction of known bits would go down to approximately p^4.) Stopping at this point, Alice and Bob have gone from a "raw key" of 32 bits (with 4 known to Eve and 2 containing errors) to a "distilled key" of 7 bits (with no errors and none known to Eve):

Alice	1	1	0	1	0	1	1
Bob	1	1	0	1	0	1	1

There are more efficient schemes for error correction and privacy amplification, based on error-correcting codes, but the scheme outlined above was the first one that was proposed, and is straightforward to understand.