Quantum cryptography

Instructions for laboratory assignment 2 in TSIT03 Cryptology
Jan-Åke Larsson, ISY, LiU

October 10, 2012

Before you start the assignment, make sure that you have read and understood the basics about quantum cryptography that are mentioned in this document. Also, look at the slides from the lecture. You need to know what it is meant to do (and what it is not supposed to do). The five steps mentioned in the lectures are:

- Raw key generation
- Sifting
- Reconciliation
- Privacy amplification
- Authentication

The raw key generation is done on the quantum channel, and the other four steps on the classical channel. A full system is quite complicated, so in this assignment we will concentrate on a simpler setup:

- Raw key generation
- Sifting
- Bit error rate estimation

The assignment is divided into two parts. One part is concerned with the raw key generation, and that takes place in the information coding group's research lab, close to the computer labs at ISY. Since we only have one QKD setup, this will be done in small groups. You will use the OKD setup to generate raw key data, and store the data on your lab account.

In the other part of the lab, you will analyze the stored raw key data. This is done in the computer labs at ISY, using standard (classical) computers, and in this part you should put together a system that performs the three steps above, using the generated raw key data.

1 The physical QKD system

The system uses polarization coding for the key bits. There are two choices of encoding, horizontal/vertical and plus/minus 45°. In quantum-mechanical notation, we either use

$$|\,0_{\mathrm{hv}}\rangle = |\leftrightarrow\rangle, \quad |\,1_{\mathrm{hv}}\rangle = |\updownarrow\rangle, \tag{1}$$

or

$$|\,0_{\text{pm}}\rangle = |\text{⤢}\rangle. \quad |\,1_{\text{pm}}\rangle = |\text{⤡}\rangle, \tag{2}$$

Do not worry about the quantum-mechanical formulation, it is only used for completeness here. There is also circular polarization (clockwise/counterclockwise),

$$|\,0_{\text{cc}}\rangle = |\circlearrowright\rangle, \quad |\,1_{\text{cc}}\rangle = |\circlearrowleft\rangle, \tag{3}$$

but we do not use that further in these notes.

In the protocol proposed by Bennett and Brassard in 1984 (BB84), Alice and Bob randomly selects between the two alternatives "hv" and "pm". Alice also selects random bit values, encodes this with her random encoding, and sends the photon off to Bob. When Bob receives a photon, he decodes using his random setting, and produces an output that is called "raw key". Alice and Bob repeat this until they reach an agreed length of transferred bits, and then proceed to perform the next step of the protocol, "sifting".

The QKD system used in this laboration is slightly different. Alice and Bob still chooses random settings between the two alternatives "hv" and "pm", but Alice does not need to encode random bit values into photons using her settings. Instead, she has a source that produces *two* photons in a pair that are in a special state (the "Bell" state), that has the property that if both photons are measured in the same setting (both "hv" or both "pm"), then the result are opposite. This means that if the settings are the same, Alice and Bob receive opposite bits (the analyzer program you will use below inverts Alices bits so that this is not a problem). In quantum-mechanical language,

$$|\,\psi_{\text{Bell}}\rangle = \frac{1}{\sqrt{2}}\Big(\,|\,0_{\text{hv}}1_{\text{hv}}\rangle - |\,1_{\text{hv}}0_{\text{hv}}\rangle\Big) = \Big(\,|\,0_{\text{pm}}1_{\text{pm}}\rangle - |\,1_{\text{pm}}0_{\text{pm}}\rangle\Big). \tag{4}$$

When the users have different settings (for example, Alice chooses "hv" and Bob "pm"), all combinations are equally probable, because

$$|\,\psi_{\text{Bell}}\rangle = \frac{1}{2}\Big(\,|\,0_{\text{hv}}0_{\text{pm}}\rangle + |\,0_{\text{hv}}1_{\text{pm}}\rangle - |\,1_{\text{hv}}0_{\text{pm}}\rangle + |\,1_{\text{hv}}1_{\text{pm}}\rangle\Big). \tag{5}$$

Again, do not worry about the quantum-mechanical notation, but do remember the properties of the state that is output from the source: opposite outputs if the encodings are the same, random combinations if they are different.

At each detector station, there is a "polarizing beamsplitter" (PBS, the little cube of glass), that separates the quantum state

$$|\,0_{\text{hv}}\rangle = |\leftrightarrow\rangle, \tag{6}$$

from the state

$$|\,1_{\text{hv}}\rangle = |\updownarrow\rangle, \tag{7}$$

After this PBS, the photons are coupled into optical fibers, and eventually counted in the detector unit.

The question is now how to make the PBS separate the states in the "pm" encoding without rotating (and re-aligning) the cube and fiber-coupler setup. This is done with the "half-wave-plate" (HWP, the round filter) that is just before the PBS. The HWP changes the polarization, by mirroring it along its "optical axis". This mirroring does not change polarization if it is along the optical axis or perpendicular to it. So, if the optical axis is vertical, the mirroring does not change horizontal and vertical polarization ("hv") at all. But this mirroring switches the state between

$$|\,0_{\text{pm}}\rangle = |\text{⤢}\rangle, \quad \text{and} \quad |\,1_{\text{pm}}\rangle = |\text{⤡}\rangle. \tag{8}$$

On the other hand, if the optical axis is at 22.5°, the effect is that it switches

$$|\, 0_{\mathrm{hv}} \rangle = |\leftrightarrow\rangle \quad \text{and} \quad |\, 0_{\mathrm{pm}} \rangle = |\, \swarrow\!\!\!\nearrow \,\rangle, \tag{9}$$

and also

$$|\, 1_{\mathrm{hv}} \rangle = |\, \updownarrow \,\rangle, \quad \text{and} \quad |\, 1_{\mathrm{pm}} \rangle = |\, \nwarrow\!\!\!\searrow \,\rangle. \tag{10}$$

The effect is that it changes from “pm” to “hv” encoding, and vice versa, so that rotating the HWP 22.5° will change the choice of encoding between “hv” and “pm”, without rotating the PBS. (Try to picture how this is possible to do by mirroring along an axis at 22.5°.)

The source is more complicated, but a brief description is in order. The “pump” laser emits light at 405nm (bright purple) at 65mW, which is around 10^{17} photons/s. They hit a nonlinear chrystal (made of Beta-Barium-Borate), which in the strong field produces pairs of photons in a correlated state. The photons have half the energy of the photons in the pump beam, and the wavelength is 810 nm, in the infrared. A couple of million pairs is produced every second. The photons in a pair emerge 6° from the pump, one of them bounces against a mirror, and then both are put through some filters to produce the mentioned quantum state. They are collected in couplers into optical fibres and sent to the analyzing stations of Alice and Bob. The analyzing stations detect around 17-20 thousand photons per second, but this is relatively few of the millions produced. This means that very few pairs will have both photons detected, only 60-70 pairs per second on a good day.

In this part of the assignment you should log on to the computer in the laboratory, open a terminal, and execute the commands

```
module add TSIT03
qkd
```

This will start the data-collecting program. There are buttons to capture and save data from the setup, and you should save data for the four combinations of settings “hv”+“hv”, “hv”+“pm”, “pm”+“hv”, and “pm”+“pm”. Appropriate filenames are

```
data-1-hvhv.txt
data-1-hvpm.txt
data-1-pmhv.txt
data-1-pmpm.txt
```

The analyzing program that you are to use in the other part of the assignment needs the “-hvhv.txt” parts to find the data to analyze. You can, if there is time, insert the “Eve” optical element, and capture data using that too, of course using a different first part of the filename. The assistant will help you with this.

2 The QKD-setup computer system

In this part of the assignment you should put together a simple QKD system using building blocks available to you in a drag-and-drop environment. You will have a number of different building blocks that are capable of performing different tasks, as described below. The blocks all have inputs (inward triangles) and outputs (outward triangles), some have only inputs, some have only outputs, and some have several. Different colors give hints of what can be done with the data. Black (gray) are general classical data streams, green are key streams, and red is the quantum channel. Don’t connect the quantum channel to a classical connector, we haven’t tested what happens. Remember, if the program breaks, you get to keep the pieces. Seriously, if it should hang, restart the thing.

You start the program by logging in, opening a terminal, and executing

```
module add TSIT03
qkdsystem
```

A first attempt would be to encrypt something with the one-time pad, the procedure is as follows.

- Take one "Manual input" box and put it in Alice's area, right-click on that and enter a love letter to Bob.
- Now, take a PRNG box, a OTP box, and a Onscreen box and put them in Alice's area.
- Connect the output from the love letter to the message input on the OTP box, and connect the PRNG output to the key input on the OTP box.
- Now connect the output of the OTP to the Onscreen box.
- Go to the File menu, and select "Run". A dialog will pop up and show you the ciphertext.
- Having received a message from Alice, Bob would like to decrypt it. Put one OTP box and one Onscreen box in Bob's area.
- You can connect this to the cryptotext output of Alice's OTP box without removing the existing connection, and this is how: connect *from* Bob's OTP message input *to* Alice's OTP output. The connection to Alice's Onscreen box should remain.
- Bob also needs the key. We'll assume that Bob has received it via courier, so connect the key input of Bob's OTP to the PRNG output using the same method again, Alice's OTP should still be connected.
- Connect another Onscreen to the output of Bob's OTP, and run the system. The cryptotext should show up first unless you've disconnected Alice's Onscreen box. Click "OK" so that Bob can read the love letter.

Some more details to use the QKD part: you need four files with data from the QKD system, and the visit to the physical system should supply these. There is test data that you can use to try things out, and these can be found in

```
/site/edu/icg/krypto/data/
```

To use these or your own files in the drag-and-drop environment you need to load them, and this can be done under the "File" menu.

Once you've loaded the data and given Alice and Bob each an end of the quantum channel, there are a number of things you still need to do. First, you need to get rid of the non-detections (the "x"es in the raw key stream). Second, you need to do sifting, to get rid of the bits where Alice and Bob have used different settings. Finally, you need to estimate the error rate. All of this can be accomplished with the available tools. Note: *no key material should be transferred between Alice and Bob, except for when they use the quantum channel.* Show the assistant your system when you are done. If you have time, try to use the generated to OTP Alice's love letter to Bob, but don't count on it to be readable, the the error rate may be too high.

A Component reference

The boxes are only active when they are in Alice's or Bob's area. You can connect several inputs to one output by connecting *from* the input *to* the output. Deleting boxes can be done by dragging them off-window or onto the borders. Deleting a line is done by grabbing a line end, disconnecting it and dropping it some distance away from any connection point. Some bugs may still remain, so be patient.

Note that right-click the boxes that have input gates to see a description of what the inputs are.

FromFile.txt: This box lets you load data from a file, either text or bitvalues (in text form). Right-click the box to bring up a dialog that lets you choose which file to load.

Manual input: This box lets you enter text or bitvalues (in text form) into the system. Right-click the box to bring up a dialog that lets you enter the text.

PRNG: This box outputs a bitstream from the internal Pseudo-Random Number Generator. Default is to produce 10000 bits, if you need more, right-click the box to bring up a dialog that lets you increase the number.

Onscreen: This box shows output in a dialog, on screen. The order you create these determines the order they appear.

ToFile.txt: This box lets you save data to a file, either text or bitvalues (in text form). Right-click the box to bring up a dialog that lets you choose which filename to save as.

OTP: This box implements the One-Time-Pad. The left (**green**) input is the key input and needs to be a bitstream, and the right input is the message input and can be either text or bitvalues (in text form).

QKD Alice: This box contains Alice's end of the quantum channel. The settings input takes a bitstream that chooses between "hv" (0) and "pm" (1), and the output is the key stream, including nondetections "x". The boxes for the quantum channel will not work if the quantum channel (in red) is not connected to **QKD Bob** box, and will not work if the data file has not been loaded in "File→Open quantum data", where you should load the file ending in "-hvhv.txt". Here, connect the output from the local **PRNG** box to the input.

QKD Bob: This box contains Bob's end of the quantum channel, see Alice's end above. Remember to connect the output of the local **PRNG** box to the input of this box.

Detections: This box takes the settings "1" and "0" and replaces them with "x" if there is a nondetection in the raw key data. The left (**green**) input to this box is raw key data from the local **QKD** box, while the right input is the settings used. The output is a combination of nondetections and settings, to be used in sifting.

Sifting: This box does the sifting. Input key bits will be dropped if the setting inputs (the gray inputs) are different, or if one of them indicates that there is a nondetection. Here, the center input is the raw key input, and the outer inputs are settings+detections, from the "Detections" box, from Alice and Bob, respectively.

Rand. Sel.: This box removes a random subset from the key stream, and outputs this on the subset output (the gray output). The green output is the remaining sifted key. Input to this box is the output from the local sifting.

Err. Est.: This box does error estimation. It removes the subset specified on the subset input (the gray input) from the key stream, and uses it for the estimate, which is displayed onscreen, in a dialog. Here, the **gray** input is the remote gray output from the **Rand. Sel.** box. And the **green** input is the local sifting output.