Quantum Cryptography

Pros & Cons

Andrea Pasquinucci

UCCI.IT

0. Abstract

In this short document, a definition of what Quantum Cryptography is will be given, and it will be compared to the traditional (or Classical) approaches. The aim is to explain what Quantum Cryptography can do without entering in any details.

Table of Contents

0. Abstract.....1

1. What is Quantum Cryptography ?.....3

2. The Quantum and the Classical way.....3

3. QKD vs. Public/Private Key protocols.....5

4. Is QKD (QC) good for me ?.....6

5. Bibliography.....7

1. What is Quantum Cryptography ?

First of all, what is Quantum Cryptography ? A more precise name is

Quantum Key Distribution (QKD)

that is a set of protocols, systems and procedures by which is possible to create and distribute *Secret Keys*. In other words

1. QKD is NOT used to encrypt and protect information
2. QKD is NOT used to transfer encrypted information
3. QKD is NOT used to store in a secure way important information

instead

- QKD can be used to generate and distribute secret keys which can then be used together with classical algorithms and protocols to encrypt and transfer in a secure way information (data) between two distant correspondents.

The typical application for QKD is the following:

- two distant correspondents (Alice and Bob) need to exchange some information electronically in a secure way
- they choose classical[1] algorithms, protocols, systems and transport technologies to exchange the data in an encrypted form
- they use a Quantum Cryptographical channel (QKD) to generate and exchange the secret keys needed by the classical cryptographical algorithms
- they use the secret keys generated with QKD and the classical algorithms to encrypt the data
- they exchange the encrypted data using the chosen classical protocols and transfer technologies.

2. The Quantum and the Classical way

Having understood what QKD can do, let us try to compare it with what can be done in a classical way. How can we generate and exchange secret keys classically ?

1 The term *Classical* is here used for algorithms, protocols etc. which are not based on *Quantum Mechanics*.

In practice a secret key is a random number of the appropriate length. So to generate a key in a classical way on a computer we need a good **PSEUDO RANDOM NUMBER GENERATOR**, that is a program which generates a sequence of numbers which are *enough random* to be acceptable as secret keys for the chosen cryptographical algorithm. On a computer it is impossible to generate pure random numbers since we have to use programs which are deterministic, the opposite of random, and the best we can do is to generate pseudo random number.

To exchange the secret keys classically, there are in practice two alternatives (and variations thereof):

1. the keys are written in two identical little books (pen and paper !) that Alice and Bob exchange directly (in person) or through a secure courier
2. Alice and Bob adopt a **Public/Private Key protocol/algorithm** (like RSA, Diffie-Hellmann etc.) to exchange electronically the secret keys.

If the first option to exchange the keys is obvious, the second should be very briefly explained. The Public/Private Key protocols use some mathematical algorithms which have the property that some computations are easy in one direction (as for example computing the product of two special numbers) but *very difficult* in the opposite direction (in the example just mentioned, given the product of the two numbers find the two original special numbers). Notice that we wrote *very difficult,* not impossible! Moreover it is not known[2] if an *easy* solution to these mathematical problems exists, it could or it could not. If one day an easy solution to these mathematical problems will be found, these algorithms will become instantaneously useless from a security point of view.

The Public/Private Key protocols allow to generate and exchange keys between Alice and Bob, or to encrypt safely small data, like the secret keys that Alice can use to encrypt her message. For example, Alice could choose a secret key to encrypt her message to Bob, and send to Bob her encrypted message through a normal channel, and the secret key using a Public/Private Key protocol. (If you wonder why Alice and Bob do not use the Public/Private Key algorithm/protocol to encrypt all data, the answer is that these algorithms are very slow and resource hungry.)

Thus the alternative to QKD is to use a Public/Private key protocol.

But how would QKD do to generate and exchange the secret keys ?

As of today QKD works only with photons (the smallest particles of which is composed light) in an optical fiber.[3] Thus we need an **OPTICAL FIBER**, currently up to 150km long, which connects Alice

2 At least publicly: some Secret Service Agencies could know more than what is currently on the public domain.

3 Other ways have been proposed, such as even in plain air, but are yet at a very early research stage.

and Bob. At the two ends of this optical fiber are connected two devices which generate and exchange single photons. Using some basics law of physics *(Quantum Mechanics)* and exchanging photons along the optical fiber, Alice and Bob can create a secret key as a sequence of 0 and 1 which is sufficiently random. At the same time Quantum Mechanics guarantees that nobody has intercepted or copied the key.

3. QKD vs. Public/Private Key protocols

A tentative comparison between QKD and Public/Private Key protocols is described in the next table:

	QDK	***Public/Private Key***	
CON	Requires dedicated HW and communication lines	Can be implemented in software, very portable	PRO
PRO	Mathematically proven secure based on basic physics laws	Mathematically undecided, based on mathematical problems for which is not known (but could be/have been discovered) an easy solution	CON
PRO	Security is based on basic principles, does not requires changes in future	Requires using longer P/P keys as computer power increases	CON
PRO	Will still be as secure even if Quantum Computer will be built	If a Quantum Computer will be built, it will be able to break it instantaneously	CON
CON	As of today very expensive	Affordable by anyone	PRO
CON	Still young and in strong development	Quite tested and deployed	PRO
CON	As of today works only at limited distances and only with (direct) optical fibers	Works at any distance and with any kind of network connection	PRO
?	Bit rate for key creation still low for some kind of applications, but it will improve soon (technical problem)	Requires considerable amount of computing power, not a problem with little data like normal secret keys, but not practical with larger data	?
PRO	Can be used with One-Time-Pad, the only mathematically proven secure cryptographical algorithm	Cannot be used with One-Time-Pad	CON

From this table of comparison, it seems that QKD is "*more secure*" but also more expensive and more difficult to implement.

4. Is QKD (QC) good for me ?

This is the most difficult question to answer since a real answer depends on what is to be secured and how.

One would need to seriously consider the adoption of QKD if one of the following applies:

- one aims at military grade security
- network communication is the weakest point in the security of the computer (digital) system
- one wants to participate in the development of a new and interesting technology
- one wants to adopt a technology which guarantees the security of network communications from now to anytime in the future.

Thus QKD (QC) is not for all. Moreover it must be remembered that QKD is only a small, albeit essential, piece of an overall security project for computer and in general digital information communication channels.

5. Bibliography

[1] *A first Glimpse at Quantum Cryptography*, Andrea Pasquinucci, to appear on http://www.ucci.it/

[2] *The Physics and Technology behind Quantum Cryptography*, Andrea Pasquinucci, to appear on http://www.ucci.it/

[3] *Quantum Cryptography*, N.Gisin, G.Ribordy, W.Tittel, H.Zbinden, *Reviews of Modern Physics,* Vol. 74, p. 145 (2002), http://arXiv.org/abs/quant-ph/0101098 (this is a very technical introduction to Quantum Cryptography)

[4] Some other references can be found for example at

http://www.csa.com/hottopics/crypt/overview.html and http://jfi.uchicago.edu/~pelton/reading.html