# Quantum Cryptography: a new hope

Christian Paquin*

Université de Montréal
paquin@iro.umontreal.ca

August 2, 1999

**Abstract**

Men always had the desire to communicate in secrecy. With the advent of computers, this desire became a necessity (for Internet communications, bank transactions, etc.) Over time, methods have been developed to achieve this goal (many of which were poor attempts), and some were even unconditionally secure (i.e. secure against a spy who has unlimited time and computer power), but they required a private key exchange before the actual communication could take place. It has been recently discovered that there is a way, using properties of quantum mechanics, for two strangers to communicate in perfect secrecy. This method led to the study of quantum cryptography. In this paper, we will first present some classical cryptographic schemes (one-time pad, DES, RSA) and their strengths and weaknesses. Then, we will study the aforementioned quantum protocol.

## 1 Introduction

The study of systems to encode messages dates back to 1900 BC. Back then, it was considered an art rather than a science, due to the lack of mathematical knowledge to study them more accurately. One of the most well-known system of the time was the cypher where a message is encoded by shifting (modulo 26) every letters by a fixed number called the key (Caesar used this system with a key value of 3.) For example, the message "julius" would be encoded as "rctqca" with a key of 8 (because j+8 = r, u+8 = c, etc.) To decode an encryption, simply subtract 8 from each letter. But such a code can easily be broken by systematically trying every 26 keys on an encrypted message and seeing which one makes sense. In another system called the substitution cypher, the key is a table describing a permutation of the alphabet letters (for example, a→k, b→m, c→y, etc.) The text would then be encrypted by replacing each letter

*supported by NSERC

by the corresponding permuted letter (replace 'a' with 'k', 'b' with 'm', etc.) The substitution cypher is more difficult to break than the shift cypher, but its security is still weak. If we know that the message is written in English, statistical analysis tells us that some letters are more frequent than others. So, by comparing frequencies of the letters in the encrypted text, we could guess relatively well which letter is substituted for which one. This may be a good puzzle for a human, but it's a piece of cake for a computer.

In World War I, a more mathematical study of cryptosystems turned the field into a science, yielding more secure systems. In what we now call classical cryptography, the participants (that we call Alice and Bob) share a secret *key* $k$. When Alice wants to send a secret *message* $m$ to Bob, she uses an encoding function $E$ and the key to produce an encrypted *cyphertext* $c = E_k(m)$. The cyphertext is sent on a *public* (insecure) channel. Nobody intercepting the cyphertext should be able to understand its meaning. When Bob receives the cyphertext, he uses his *decoding* function $D$ along with the key to recover the message $m = D_k(c)$. Some classical systems have perfect secrecy. In this context, the users must exchange a key, in advance, on a secure channel. This restrains the flexibility of such systems, particularly in today's world where a lot of private communications must take place between strangers.

Fortunately, the invention of public key cryptosystems changed the face of cryptography. In such a scheme, two different keys are used: one for the encoding and one for the decoding. Everybody have access to the encoding key of a particular user, so anybody can send messages to that person. On the other hand, the decoding key is known only by that person, ensuring the privacy of the messages he receives. Public key cryptosystems solve the key distribution problem, but such systems are by definition only *computationally* secure. This means that an eavesdropper intercepting the cyphertext who has enough time and computer power could retrieve the message.

If our goal is to communicate with unconditional security, without having to exchange keys in advance, then there is a new hope: quantum cryptography. We will see a protocol which uses properties of photons to exchange a key over a public channel, that can detect the presence of an eavesdropper if any. The key can then be used with an absolutely secure classical cryptosystem.

## 2 Classical cryptosystems

Many cryptosystems were developed over time. We present in this section three systems which played an important role in cryptography.

## 2.1 One-time pad

The one-time pad cryptosystem was created by Gilbert Vernam in 1917. It is very simple and yet, very effective. Indeed, this system has perfect secrecy, meaning that no matter the time and computer power available, an adversary could not break it (as long as the key remains secret). The system works as follows:

1. Prior to the communication, Alice and Bob secretly exchange a random key $k = k_1 \dots k_n$ of $n$ bits.
2. When Alice wants to send the message $m = m_1 \dots m_n$ to Bob, she encodes it by computing the bitwise exclusive-or (XOR) between the message and the key producing the cyphertext $c = (m_1 \oplus k_1) \dots (m_n \oplus k_n)$. Recall that $0 \oplus 0 = 1 \oplus 1 = 0$ and $0 \oplus 1 = 1 \oplus 0 = 1$.
3. Upon reception, Bob decodes the cyphertext by applying the inverse operation $m = (c_1 \oplus k_1) \dots (c_n \oplus k_n)$.

For example, if the participants share the secret key 011100 and that Alice wants to send the message $m = 101110$ to Bob, she encodes it with the key obtaining

$$c = m \oplus k = 101110 \oplus 011100 = 110010$$

and sends the cyphertext $c = 110010$ over the public channel. To recover the message, Bob XORs the key to the cyphertext yielding

$$m = c \oplus k = 110010 \oplus 011100 = 101110$$

Despite its security, the one-time pad is very impractical. For every message encoded with the system, the participants need to exchange a secret key that has at least the same length. Also, one must not use the same key twice (hence the name of the system). If you use the same key $k$ to encode the two messages $m$ and $m'$, then if a eavesdropper intercepts the two cyphertexts $c$ and $c'$, he only has to compute

$$c \oplus c' = m \oplus k \oplus m' \oplus k = m \oplus m' \oplus k \oplus k = m \oplus m'$$

to obtain the XOR of both messages, which is a lot of information! The one-time pad is used mainly for highly confidential communications (in the government or in the military).

Many other systems were developed to allow long messages to be encoded with short keys that can be reused. One of the must popular is the DES.

## 2.2 DES

The Data Encryption Standard (DES) is currently the most used cryptosystem in the world. We find it in bank transactions, Unix password systems and many

other places. It became a standard on January 15th, 1977. With DES, Alice and Bob share a 56-bit secret key that they use to encode and decode 64-bit pieces of messages. The encoding and decoding functions are too complex to present here, let's just say that the 64-bit message passes through a series of permutations and transformations depending on the key value [1].

Because the key length (56) is smaller than the message being encoded (64), information theory [2] tells us that the system can't be unconditionally secure. DES is rather old and nowadays, many hardware chips are designed to break it. With a reasonable effort, a computer network can break a DES cyphertext within a day of work [3]. This is the price to pay to use small keys. In fact, the only absolutely secure system is the one-time pad (or variations). But in both cases, the users must agree in advance on a secret key.

## 2.3 Problem of key exchange and RSA

The systems presented until now share a major disadvantage: for two people to communicate secretly, they must exchange in advance a secret key in person or over a secure channel. This is known as the key distribution problem. This implies that two strangers can't communicate with those systems (assuming that they don't have a secure channel; if they did they would not need encryption in the first place.)

In 1976, Whitfield Diffie and Martin Hellman [4] revolutionized the cryptographical world by introducing the theoretical notion of *public key* cryptography. In such a system, every person would have a public encoding key and a secret decoding key, such that encryption and decryption using these keys would be inverse functions. The encoding key would be published in some kind of index (similar to a phone book) and the decoding key would be kept secret by every user. If Alice wants to send a secret message to Bob (that she doesn't necessary know), she would look up in the index to find Bob's encoding key, she would then encrypt her message with this key and send it through an insecure channel. Bob, receiving the message, would then decode it using his secret key. In short, anyone can write a secret message to Bob (using his public key) but only Bob can decode them with his private key (even the sender can't decode his own message once it has been encrypted).

The first realization of a public key cryptosystem is due to Ron Rivest, Adi Shamir and Leonard Adleman [5] who invented the popular RSA cryptosystem in 1977. The system is now widely spread, particularly over the Internet. The security of RSA is based on the factoring problem, i.e. the difficulty, given a large number $n$, to find its prime factors. If Bob wants to have a public key, he secretly and randomly chooses two large prime numbers $p$ and $q$ (about the same length) that he multiplies to get $n = pq$. He then randomly chooses an $e$ such that $e$ and $(p-1)(q-1)$ are relatively prime and publishes it, along with $n$, in the index book (those two numbers constitute the public key). His decoding

private key will be $d = e^{-1} \bmod ((p-1)(q-1))$, easily computable with the Euclidean algorithm. If Alice wants to send Bob the message $m$ ($0 \leq m < n$, number converted from a binary string), she computes the encoding function $c = m^e \bmod n$ and transmits $c$ to Bob. To recover the message, Bob applies the decoding function $m = c^d \bmod n$. We can check that both functions are inverse operations. If $m$ and $n$ are relatively prime, it's easy to see that, since $ed \equiv 1 \,(\text{mod } (p-1)(q-1))$ and that $(p-1)(q-1) = \phi(n)$ (the Euler function),

$$c^d \equiv m^{ed} \equiv m^{k(p-1)(q-1)+1} \equiv m m^{k\phi(n)} \equiv m \,(\text{mod } n)$$

(because $m^{\phi(n)} \equiv 1 \,(\text{mod } n)$ when $m \in \mathcal{Z}_n^*$). With more number theory, we can generalize the verification to all $m$. All the operations required in the system can be efficiently calculated, so the encryption and decryption are fast (which is a desirable property in a cryptosystem).

Here is a small example. If the random choices of Bob are $p = 173$ and $q = 149$, he obtains $n = pq = 25777$ and $(p-1)(q-1) = 172 \times 148 = 25456$. Bob randomly chooses $e = 3417$, checks that $gcd(n, e) = gcd(25777, 3417) = 1$ and computes $d = e^{-1} \pmod{25456} = 19593$. He publicly announces the numbers $e = 3417$ and $n = 25777$. Now, if Alice wants to send him the message $m = 9273$, she computes $c = 9273^{3417} \bmod 25777 = 6878$ and sends it through the public channel. After reception, Bob recovers the text $m = 6878^{19593} \bmod 25777$.

The two prime numbers $p$ and $q$ are never used in the communication. In fact, they should be discarded after Bob computes $n$ and $(p-1)(q-1)$. If a spy could learn those numbers, he could easily break the system by computing $d$ the same way Bob did. If an efficient algorithm to factor large numbers was known, RSA would be useless because anybody could factor $n$ into $p$ and $q$. This task is believed to be difficult, since the best algorithm known to factor large numbers takes superpolynomial time. The longer $n$ is, the more secure the system is. The RSA company constantly challenges people to factor some numbers that they provide [3]. The last reported factored number had 140 digits. The task required 8.9 CPU years distributed on many computers (one month in real time). They estimate that a 1024-bit number would be 40 million times longer to factor.

It's interesting to know that there exists an efficient algorithm for factoring. Only, it requires a quantum computer [8, 9], a computer model based on quantum mechanics. For more information on classical cryptography, consult [7, 6].

# 3 The quantum protocol

We will now see how we can use quantum mechanics to develop a scheme that allows Alice and Bob to exchange a key right under the nose of an adversary.

This scheme was invented by Charles Bennett and Gilles Brassard [10, 11] in 1982. The protocol uses photons, so let's first study some of their properties.

## 3.1 Properties of photon polarization

Without getting lost in details, we know that photons have a polarization angle, corresponding to the angle of the plane in which they oscillate on their propagation axis. The polarization angle is a number $\theta$ such that $0° \leq \theta < 180°$ since there are no differences between a photon polarized at $\theta$ and another at $\theta + 180°$.

Photons emerging from a light source often have an unknown polarization angle. To induce a particular polarization to a photon, we use a light filter that has the property of letting only photons polarized in this angle pass through. If we use a filter that lets photons polarized at degree $\theta$ pass through (we will call it a $\theta$-filter), any photon polarized at this angle will pass through undisturbed, and photons polarized at any other angle will either be stopped by the filter or will emerge with a polarization $\theta$. The two possibilities are dictated by the laws of probability. Quantum mechanics tells us that a photon polarized at angle $\phi$ passing through a $\theta$-filter has probability $\cos^2(\phi - \theta)$ of emerging with polarization $\theta$ and a probability $\sin^2(\phi - \theta)$ of being stopped by the filter.

Physical explanations of this phenomenon are beyond the scope of this paper (if you are interested, see [12].) All we need to know is that the event is a real probabilistic choice, i.e. it's a *true* random event (as opposed to a technologically unpredictable one.) The choice is made by *nature*, at the moment where the photon hits the filter, depending only on the difference of the angles.

We will now study some specific cases. Suppose we are only interested in photons polarized at 0° (represented by ↔), 45° (↙↗), 90° (↕) and 135° (↖↘). Let's see the behavior of these photons when they go through 0° and 45°-filters. The table 1 gives the probability (given by the above law) for each photon to pass through the filters undisturbed, depending on the type of the filter.

| | 0° | 45° |
|---|---|---|
| ↔ | 1 | 1/2 |
| ↕ | 0 | 1/2 |
| ↙↗ | 1/2 | 1 |
| ↖↘ | 1/2 | 0 |

Table 1: Probability that a photon passes through a $\theta$-filter

Suppose we want to use the polarization angle to encode bits to be transmitted. We need to choose a basis in which we can distinguish the two values (0 and 1) without ambiguity. One choice is the *rectilinear* basis (photons polarized at angle 0° or 90°). We can arbitrarily decide that the bit 0 is encoded by a

horizontally polarized photon ($\leftrightarrow$) and that the bit 1 is encoded with a vertically one ($\updownarrow$). Here is a simple protocol allowing Alice and Bob to communicate with photons prepared at angle 0° or 90°. If Alice wants to send a 0 to Bob, she prepares a $\leftrightarrow$ and sends if through optical fiber. Bob, at the other end, measures it with a 0°-filter. If the photon emerges, then he knows that the bit was 0 (probability = 1). If the photon is stopped, then he concludes that the bit was 1.

If instead Alice and Bob only had at their disposal photons polarized in the *diagonal* basis (45° or 135°), they could use this alternate protocol to communicate. They agree that $\swarrow\nearrow$ means 0 and that $\nwarrow\searrow$ means 1. For Alice to send, for example, a 1 to Bob, she prepares $\nwarrow\searrow$ and sends it through the channel. Bob then uses a 45°-filter to recover the bit (0 if it passes, 1 if it's stopped).

What happens if we encode a bit in one basis and measure in the other one? For example, suppose Alice sends $\updownarrow$ (meaning 1) to Bob who measures in the diagonal basis (using a 45°-filter). Table 1 tells us that the photon will emerge (interpreted as 0) with probability 1/2 and will be stopped (interpreted as 1) also with probability 1/2 . This means that the value obtained by Bob will be a random bit. Another way of saying it is that if Alice sends a photon in one basis and Bob measures in the other, he can gain no information about the original bit. This fact is at the core of the quantum key distribution described next.

## 3.2 Quantum Key Distribution

We are now ready to see how Alice and Bob can obtain a common and random key using quantum mechanics. The protocol is a key distribution, i.e. it allows Alice and Bob to agree on a secret bit string to be used as a key in a cryptosystem.

The protocol starts with Alice who sends a sequence of photons randomly polarized ($\leftrightarrow, \updownarrow, \swarrow\nearrow, \nwarrow\searrow$) (step 1). Bob receives the photons and for each one, he independently chooses to measure them in the rectilinear (+) or the diagonal ($\times$) basis (2) and he notes down the measuring results (3). Note that some photons may not be received due to imperfection of the transmitting and measuring devices (the ? in the table). Bob then sends, over the classical channel, the basis in which he measured each photon he received (4). Alice transmits back the positions where her encoding basis matches Bob's measuring one (5). Bob only keeps those results (6) and interprets them as bits (7). Alice also interprets those photons as bits, yielding a common and random bit sequence. Here is an example of the protocol.

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1) | $\leftrightarrow$ | $\nearrow$ | $\searrow$ | $\nearrow$ | $\searrow$ | $\updownarrow$ | $\leftrightarrow$ | $\searrow$ | $\updownarrow$ | $\leftrightarrow$ | $\updownarrow$ | $\nearrow$ |
| 2) | $\times$ | $\times$ | $+$ | $+$ | $\times$ | $+$ | $\times$ | $+$ | $+$ | $+$ | $\times$ | $\times$ |
| 3) | $\nearrow$ | $\nearrow$ | $\leftrightarrow$ | ? | $\searrow$ | $\updownarrow$ | $\searrow$ | $\updownarrow$ | ? | $\leftrightarrow$ | ? | $\nearrow$ |
| 4) | $\times$ | $\times$ | $+$ | | $\times$ | $+$ | $\times$ | $+$ | | $+$ | | $\times$ |
| 5) | | $\surd$ | | | $\surd$ | $\surd$ | | | | $\surd$ | | $\surd$ |
| 6) | | $\nearrow$ | | | $\searrow$ | $\updownarrow$ | | | | $\leftrightarrow$ | | $\nearrow$ |
| 7) | | 0 | | | 1 | 1 | | | | 0 | | 0 |

After the protocol, Alice and Bob share a random key (here 01100) which they can use to communicate secretly over a classical channel. If they use the one-time pad cryptosystem, their communication will be unconditionally secure. Note that the protocol could not be used to directly send a message because nearly half the bits are (randomly) discarded in the process.

### 3.3 The protocol vs Eve

The protocol works fine when Alice and Bob communicate over secure classical and quantum lines. But if we want to use it as a key distribution, we must study what happens when an eavesdropper (called Eve) is watching and possibly playing a role in the communication.

If we want the protocol to be really secure, we must allow Eve to do whatever she wants. We suppose that Eve can spy on both the classical and the quantum channels. When the photons pass right by her, Eve is free to measure them in any basis she likes, to intercept the photon and send another one. We will only consider the case where she measures the passing photons in either the rectilinear or the diagonal basis (the same as Bob's) and retransmits the measured photon (the protocol has been proven secure against any attack; we use this example for simplicity.) The more Eve measures (spies), the more she will cause noticeable disturbances. Indeed, if Alice sends a photon polarized in one basis and Bob measures in that same basis, they will keep the bit at this position. If Eve measures in between with the same basis, she will not change the polarization of the photon and she too will know the corresponding bit (i.e. she will know one bit of the key.) But if, instead, Eve had measures in the other basis, the photon will be repolarized in this basis and with probability 1/2, Bob will obtain a different bit than Alice. For example, if Alice sends $\leftrightarrow$ (meaning 0) and Eve measures with a 45°-filter getting $\nearrow$, and if Bob measures with a 0°-filter, with probability 1/2 he will get $\updownarrow$ (interpreted as 1). Alice and Bob would agree on this bit, even though it's different. We are only interested in the effect of Eve's measurement when Alice and Bob use the same basis because if they don't, the corresponding bits will be discarded. The next table gives the probability that Alice and Bob get the same bit depending on the basis they and Eve used.

| Alice & Bob | Eve | Probability |
|---|---|---|
| + | + | 1 |
| + | × | 1/2 |
| × | + | 1/2 |
| × | × | 1 |

For every bit that Alice and Bob keep and that Eve measured in between, there is a probability of 1/4 that they will be different. So every time Eve measures a photon that they will keep, one time out of four, she will introduce an error in their key. To ensure that their communication was secret, Alice and Bob will compare some information about their respective key strings. If Eve spied too many times, the odds that she will be discovered will be high. One way to test for eavesdropping is to compare the parity of random subsets of the key chain. For example, suppose Alice chain is 01001011 and that Bob's is 01101001 (the two errors were caused by Eve's measurement.) Alice would choose a random subset of positions (1,3,5,8) and publicly tell them to Bob along with the parity of the bits at those positions, which is 0 ($0 \oplus 0 \oplus 1 \oplus 1$). Bob would reply that the parity of his bits is 1 ($0 \oplus 1 \oplus 1 \oplus 1$). Because they differ, they would conclude that there was a spy on the line. They would then stop the communication and restart the protocol over. It can be shown that this parity comparison can detect, with probability 1/2, that Alice and Bob have a difference in their key, regardless of the number and positions of errors. They only have to repeat this process twenty times to reduce the probability that Eve spied unnoticed to less than one in a million. Note that the protocol is probabilistically secure, meaning that with a high probability (as close to 1 as we want), the protocol is unconditionally secure.

# 4 Experimental realizations

The quantum key distribution is a nice result, but is it technologically feasible? For a long time, it was considered science-fiction but in 1989, at the IBM Thomas J. Watson Research Center, the first working prototype was built by Charles Bennett, Gilles Brassard and some of their students [13]. It allowed Alice and Bob (two computers) to communicate with perfect secrecy at a rate of 10 bits/second over a distance of 30cm! A small step for Alice and Bob but a large step for quantum cryptography. The protocol had to be slightly modified to work in practice. For instance, it's difficult to send only one photon at a time, some dim flashes of light are sent instead. This complicates the proof of security. Also, some errors occur in the key even when Eve is not present. So the test for eavesdropping must be modified. A technique known as privacy amplification is used, in which Alice and Bob distill (through public discussion) from their partly secret key a smaller but highly secure key. The probability that Eve know even one bit of this new key will be very low.

Later, some other implementations were realized at a larger scale. At the Los Alamos National Laboratory, they realized an experiment where a quantum

key distribution takes place over 48 km of optic fiber. They even succeed in transmitting the photons in free-space over 1 km. A group at the University of Geneva established the protocol under the lake of Geneva on a distance of 23 km. This means that quantum cryptography will be part of our lives in a near future.

## Acknowlegment

I thank Gilles Brassard for his encouragement and support. I am grateful to Sébastien Paquet, Frédéric Légaré and Anton Stiglic for their useful comments on this paper.

## References

[1] http://www.itl.nist.gov/div897/pubs/fip46-2.htm

[2] C. E. Shannon, "Communication theory of secrecy systems", Bell Systems Technical Journal, 28 (1949), pg 656–715.

[3] http://www.rsa.com/rsalabs/html/challenges.html

[4] Whitfield Diffie and Martin Hellman, "New Directions in Cryptography", IEEE Transactions in Information Theory, v. IT-22, (Nov. 1976), pp. 644–654.

[5] Ron Rivest, Adi Shamir and Leonard Adleman, "A method for obtaining digital signatures and public key cryptosystems", Communications of the ACM, 21 (1978), pp 120–126.

[6] Bruce Schneier, "Applied Cryptography", John Wiley & Sons, New York (1996).

[7] Douglas R. Stinson, "Cryptography, Theory and Practice", CRC Press, New York (1995)

[8] Gilles Brassard, Isaac Chuang, Seth Lloyd and Christopher Monroe, "Quantum Computing", Proceedings of the National Academy of Sciences USA, Vol. 95, (Sept. 1998), pp. 11032–11033.

[9] Christian Paquin, "Computing in the quantum world", Proceedings of the Canadian Undergraduate Mathematical Conference 1998, (1998).

[10] Charles H. Bennett and Gilles Brassard "Quantum cryptography and it's application to provably secure key expansion, public key distribution, and coin-tossing", Proceedings of IEEE International Symposium on Information Theory, (Sept. 1983), p. 91.

[11] Charles H. Bennett, Gilles Brassard and Artur K. Ekert, "Quantum Cryptography", Scientific American, (October 1992), pp. 50–57.

[12] Asher Peres, "Quantum Theory: Concepts and Methods", Kluwer Academic Publishers, Dordrecht (1993).

[13] Charles H. Bennett, François Bessette, Gilles Brassard, Louis Salvail and John Smolin, "Experimental Quantum Cryptography", Journal of Cryptology, Vol. 5, No 1 (1992), pp. 3–28.