Applications of Quantum Cryptography

by

Naya Nagy

A thesis submitted to the
School of Computing
in conformity with the requirements for
the degree of Doctor of Philosophy

Queen's University
Kingston, Ontario, Canada
March 2010

An meine Mutter,

in Dankbarkeit

Abstract

This thesis extends the applicability of quantum cryptography.

First, we prove that quantum cryptography at least equals classical cryptography in an important area, namely authentication. The quantum key distribution protocols presented here show that, contrary to previous belief, authentication can be done with quantum methods only.

In addition, we have designed quantum security systems in unconventional settings. The security of sensor networks poses specific challenges, as the sensor nodes in particular can be physically picked up by the intruder. Our scheme protects both the integrity of the communication messages and it also protects the identity of the nodes, such that a reading intrusion of a node is detectable.

The problem of access control in a hierarchy refers to a large number of users, organized in a hierarchy, having selective access rights to a database. Our quantum solution introduces quantum keys to the effect that the cryptographic scheme is dynamically adaptable to changes in the user structure, and it exhibits increased security levels.

To the best of our knowledge, this thesis is the first to introduce quantum keys, that is secret keys defined by an array of qubits. We show that quantum keys make it possible for two parties to communicate with one-time pads without having to meet in

advance. Also, opposite to previous cryptographic "common sense", the security level of a quantum cryptosystem with quantum keys and quantum messages is maintained while being used.

Acknowledgments

The words and ideas in this thesis could take form only after I made friends with them. The words needed my trust and the ideas my love. This whole thesis was mainly a journey of love and trust.

I am most indebted to my father, Ilie Hodor, for the existence of this work. Without his strong, vertical character, I would never have imagined that pursuing a PhD could be a woman's career, much less my own. I owe him every step, the appreciation of reason.

Many thanks go to my supervisor Selim Akl for his inspirational freedom of thought: Walking past the edge is the immediate consequence of putting on your shoes.

My husband, Marius Nagy, has had an important influence on the development of this research. He hurtfully challenged every single, newly developed idea, thus improving many of them. Thank you.

This thesis is dedicated to my mother, the first to show me trust and love.

Statement of Originality

I, Naya Nagy, certify that the work presented in this thesis is original unless otherwise noted.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Domains of computer science tend to develop somewhat independently from each other, as each specialty develops its own theoretical background, strategy, applications and even professional language and group of researchers. The purpose of this thesis is to find meeting points, areas of common interest, between apparently disjoint subareas of computer science.

The main problem addressed in this thesis is to extend the applicability of quantum cryptography to new and unconventional domains. Therefore, we identify domains and applications that have not been connected to quantum computation in the past, but actually benefit quite naturally from the presence of quantum approaches. The affinity of large computer science domains to quantum cryptographic solutions can be clearly seen from the strict improvement that quantum cryptographic schemes consistently show over their classical counterparts. Additionally, we will design schemes that exhibit security properties that exceed the previously accepted capabilities of even general purpose quantum cryptography.

Quantum Cryptography applies methods of quantum computation to cryptographic applications. Cryptography has been improved by the addition of quantum methodology. Mainly, quantum cryptography has been shown to improve the security of key distribution protocols.

Quantum computation exploits the quantum properties of atomic or subatomic particles such as atoms, photons, or electrons. Such particles encode qubits. Quantum computation computes with qubits rather than bits. Qubits add to the possible states of binary bits 0 and 1, any fractious combination of 0 and 1 at the same time. As such, a qubit can be in a superposition of 0 and 1.

For the purpose of this thesis, we will call *classical cryptography*, cryptographic algorithms and solutions that use classical bits and classical computation only. In particular, classical cryptography does not use any qubits, or quantum properties.

Most quantum cryptography algorithms to date perform the distribution of a secret key between two cryptographic persons. These persons will use the secret key further in their private communication. The major advantage of all previous quantum key distribution protocols is that the secret key is effectively unbreakable. Also, the intrusion of a malevolent party is detectable. These protocols rely on classical cryptography in that they generally use a classical channel and the ability to authenticate such a channel.

We show in this thesis that quantum cryptography can fully function without any help from classical cryptography results. Specifically, quantum cryptography can perform authentication of the communication partners. Quantum authentication had been thought to be impossible previously, as it was believed that authentication can be done exclusively by classical means. With this result in mind, several specific

applications ensue.

Sensor networks form over a geographical area where a set of sensors are deployed at random. Depending on their usage, privacy of message contents might be desired. Also, the security of the sensors themselves might be an issue. Security in sensor networks poses specific settings and environments. We have designed schemes that address security issues in sensor networks that surpass in quality any classical scheme. Our scheme protects the network from eavesdropping, masquerading and identity theft. Its natural environment makes a sensor network prone to eavesdropping, as an intruder can readily listen to the radio signals in the environment. Also, a sensor node can be picked up by the intruder in the attempt to pervert its behavior. It can be clearly seen from here that sensor networks are generally vulnerable to attacks. Quantum approaches to security protect the privacy and identity of a node to a level where any intrusion can be detected.

Quantum cryptography is the more desirable option for managing access in a large hierarchical structure using quantum security means. The problem is to assign secret keys to users in a hierarchy. The hierarchy may be dynamically changing, as new users arrive or others leave. Also, users may change their position in the hierarchy (through promotion for example). We show that the use of quantum keys greatly simplifies the process of key management.

The contributions of this thesis are shown below.

1. Quantum key distribution has been considered to need a classical authenticated channel. We have shown that the classical channel does not need to be authenticated. The two parties authenticate each other through the quantum protocol itself [31].

2. A protocol has been developed that distributes a secret key among two parties using a quantum channel only. This is unique, as all previous quantum distribution protocols have used both a quantum and a classical channel [30].

3. We have shown that authentication can be done using quantum means only. This has been thought to be impossible previously. In fact we show that secret information can be produced using public information only. This is unique for quantum computation and is not possible to be achieved classically [32].

4. We have designed a security scheme that protects the communication in a sensor network. The scheme relies on quantum cryptographic methods [33]. It comes with all the advantages of quantum cryptography, namely effectively unbreakable keys. This is the first time that sensor networks have been linked to quantum computation.

5. Further, the identity of a sensor node in a network can be protected using quantum security means [34]. The scheme designed by us, is the first to unequivocally protect the identity of a communication partner. Note that the identity of a sensor node is generally vulnerable to identity theft. Our scheme can detect any violation of the node's identity, even an attempt to only read the memory of a node.

6. In every security scheme to date, the secret key has always been a classical key, meaning a string of bits. Even in quantum protocols, the quantum properties are used exclusively to obtain a classical binary key. We design the first scheme that uses both a classical key and a quantum key. The keys provide access to a database. This database is supposed to serve a very large number of users.

Moreover, the users are organized in a hierarchy. The advantage of our scheme, besides increased security, is easy adaptability of the key structure to changes in the users' hierarchy. Users can easily leave the organization as well as new users can easily join the organization. This means that the access keys of remaining users are not affected by the change of the user hierarchy or structure.

7. Quantum messages are messages consisting of quantum bits. We designed the first scheme to use quantum messages and with surprising results. Each message is encrypted with a one-time pad, a key that is used only once and will never be used again. Moreover, after a first meeting of the communicating partners, or a first deployment, the partners never have to meet again: all one-time pads are generated as needed, without any other meeting prior to the communication. Needless to say, any communication between the two parties does not reveal any information about the one-time pads.

The rest of the thesis is organized as follows: Chapter 2 defines the basic tools of Quantum Computation. Chapter 3 proves that quantum authentication is possible and gives two protocols for authentication with quantum communication. Chapter 4 describes sensor networks enhanced with qubit memories and their security. Chapter 5 describes a scheme in which one-time pads are created and used without a prior meeting of the communication partners. This is possible with quantum messages consisting of a string of qubits. Chapter 6 developes a quantum security scheme for access control in a hierarchy. Chapter 7 concludes the thesis. Previous work will be presented for each problem in the corresponding chapter.

Chapter 2

Basics of Quantum Computation

It is today apparent that computer components (circuits) are getting ever smaller to the level of micro and even nano dimensions. As such it becomes impossible to describe them by macroscopic laws of classical physics. Indeed, it is a reality that gates and bits are reaching atomic and subatomic levels, as demonstrated by the (new) field of nanotechnology [21]. At this level, the laws that govern the behavior of computing components fall necessarily under the authority of quantum mechanics. In classical physics all quantities can be predicted and measured with arbitrarily small error. Not so in quantum mechanics. The best one can hope for in the quantum mechanical world is to predict the probability of a value or phenomenon to be one way or another at some chosen time in the future, or conversely at some fixed point in the past. As Brian Greene [23] puts it "the universe, according to quantum mechanics, is *not* etched into the present; the universe, according to quantum mechanics, participates in a game of chance." This simply means that computing devices of subatomic sizes partake of this type of universe and consequently will be players in the game of chance.

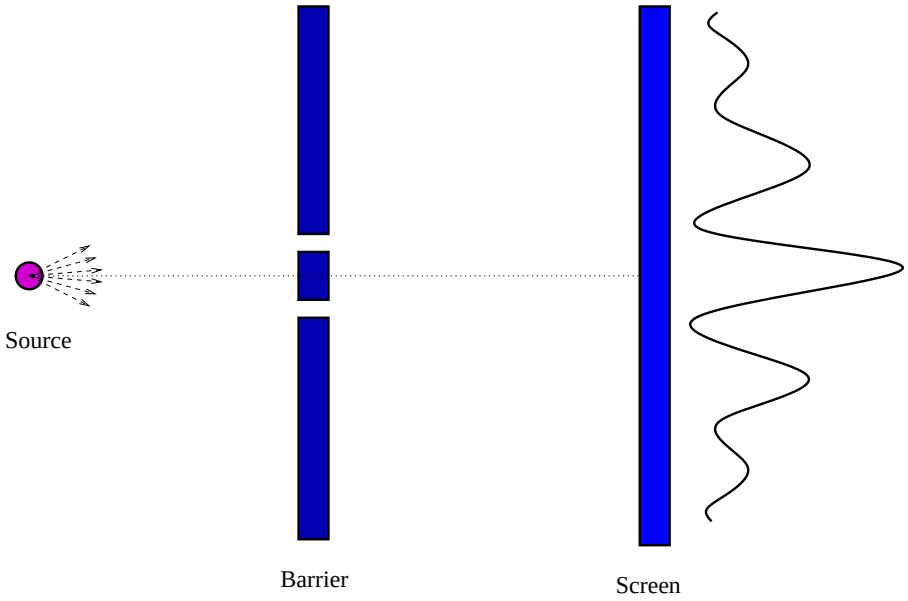

Figure 2.1: Young's double-slit experiment. Light projected onto a screen through a barrier pierced with two closely spaced slits creates an *interference pattern.*

Atomic and subatomic particles have a dual nature. Depending on the experimental context, they may behave either as particles or as waves. Examples of particles with such a dual nature are photons, electrons, neutrons, or entire atoms.

Thomas Young, in 1801, devised an experiment [22] that proved that light behaves as a wave. A source of light hits a barrier that allows light to pass only through two thin, closely spaced slits (see Fig. 2.1). The light rays that manage to pass the barrier through the slits go on to impress a photosensitive screen. The image projected on the screen is an interference pattern, thus showing that a light-wave has passed both slits simultaneously. With the two slits considered synchronous sources of light-waves, the interference pattern is expected to show on the screen.

With today's technology, the source of light can be much better controlled than the source used by Young, which was a simple candle. In fact the sources that are available now can emit very dim light. The source of light can be made so low as

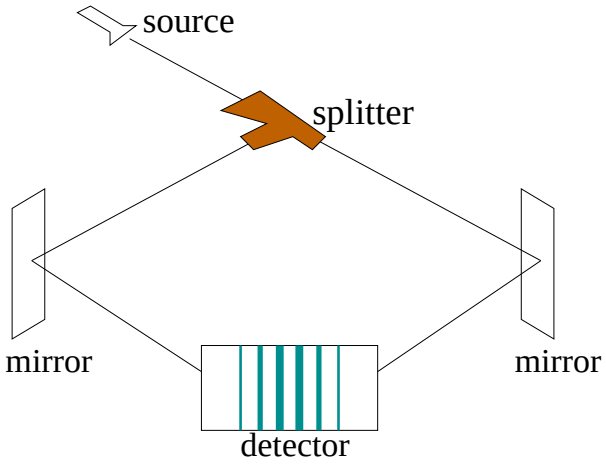

Figure 2.2: The beam splitter experiment. Alternative photon traveling paths are far apart.

to emit exactly one photon at a time. The beautiful property of light's behavior is that the overall outcome of Young's experiment using such a light source is exactly the same, after an ensemble average of measurements. At a closer look, each photon generates a small white dot on the screen. In order to reach the screen, the photon has two alternative paths: through the left or the right slit of the barrier. Over time more and more dots accumulate on the screen with each captured photon. If the experiment is run for a sufficiently long time, it will be clear that photons hit the screen preferentially along the interference lines obtained previously, such that the interference is now constructed dot by dot.

The effect of the experiment can be enlarged in the sense that the separation of the two alternative paths, the two slits in Young's experiment, can be made arbitrarily large. The beam splitter experiment exhibits just this property [23] (see Fig. 2.2). The experiment again exploits the properties of light. A light beam generated by a laser source is divided in two halves by a beam splitter and sent further along orthogonal directions. The beam splitter is literally a half clarity mirror. Light that

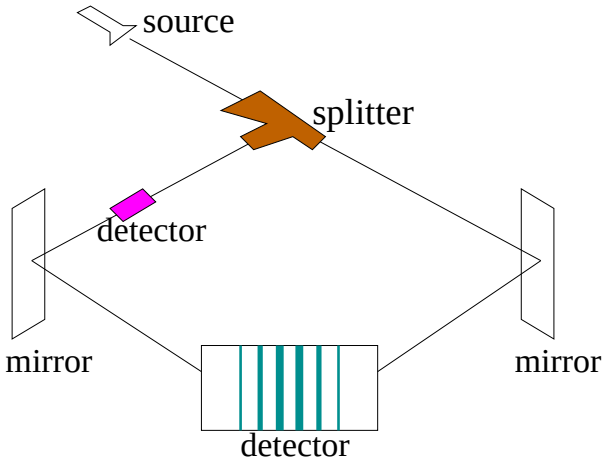

Figure 2.3: The beam splitter experiment with a detector. Alternative photon traveling paths can be detected.

hits this mirror has a 50% chance to pass through the mirror and continue towards the right or to be reflected and continue its way towards the left. The two orthogonal paths are then redirected by two regular mirrors towards a screen. The screen will again show an interference pattern of two waves. If photons are emitted sequentially, the interference pattern will again be constructed over time. This shows that even with the alternative paths being far apart, one single photon is still behaving like a wave. The photon has passed both paths at once, so that it "interfered" with itself. Differently put, the photon, as a particle, was in both places at the same time.

A detector can be put on one of the paths, for example the left path (see Fig. 2.3). The detector gives a signal if a photon passes that way. A photon emitted by the laser has a 50% chance to be reflected by the mirror. Indeed, the detector will signal exactly one half of the photons. Moreover, if a photon is emitted and the detector does not signal, then a detector on the alternative right path would signal the presence of the photon. Using detectors, we determine which path the photon has "chosen". In such a setting, it is known by which path the photons have traveled. Following

this knowledge, all photons behave like particles. There will be no interference, just one bright fringe.

2.1 Superposition and Measurement

In order to form an interference, the photon is supposed to have a probability amplitude to travel on both paths at once. The two paths are equally likely. The photon has a 50% chance to choose left or right. In principle, the distribution of probabilities may take any values that add up to 100%. It is said that the photon is in a *superposition* of both possibilities. If the photon is observed, it can be found either on the right path or on the left path but not on both. However, once the photon is observed, the superposition collapses and interference is destroyed. The photon's choice of paths collapses to only one of the possibilities and the other path ceases to be an option of choice.

This is valid for any quantum particle that is allowed to be in multiple states. For example, an electron may have its spin *up* or *down*. The electron is directed specifically up or down with a certain probability amplitude, for example the electron is 30% up and 70% down. As the probability is distributed over both directions, the electron is in a superposition of both states. When measured, the electron collapses to either one or the other direction, according to the probability amplitude squared. The above electron will be rather observed to be *down*. The state of the electron after measurement remains down, conforming to the direction that has been observed.

2.1.1 The Qubit

The description of a pure quantum state is well formalized by means of linear algebra.

For an electron's spin, the two directions are assigned formal values. By convention, we denote *up* with 1, and *down* with 0. In this simple case, the spin direction is called an observable. For a system with several observables, the state of a system is described as a vector: the vector of state, or the state vector. This is accurate for discrete states. Dirac [17] first introduced the *bra/ket* notation for state vectors. State vectors are *kets* $|x\rangle$, simple column vectors. The conjugate transpose of a ket is a *bra* $\langle x|$. For a simple system then, of only one electron with a spin, there are two possible vector states $|0\rangle$ and $|1\rangle$.

The general form of a qubit is

$$q = \alpha|0\rangle + \beta|1\rangle,$$

where α and β are complex numbers, and $|\alpha|^2 + |\beta|^2 = 1$. Thus, q is in a superpostion of 0 and 1, with a $|\alpha|^2$ probability to be 0, and a $|\beta|^2$ probability to be 1.

2.2 Quantum Gates

Quantum gates are the equivalent of classical gates. They transform a qubit according to the gate's logical meaning. Quantum gates always have the number of output qubits equal to the number of input qubits. This requirement ensures the reversibility of quantum computations. In the following, only single qubit gates will be discussed, as the quantum basis of this thesis does not require higher order quantum gates.

The effect of a quantum gate can be seen from its effect on the base vectors $|0\rangle$ and $|1\rangle$. Suppose, some arbitrary gate G transforms $|0\rangle$ into a superposition $\delta_1|0\rangle + \delta_2|1\rangle$,

with $|\delta_1|^2 + |\delta_2|^2 = 1$, and applied to $|1\rangle$ generates the output $\gamma_1|0\rangle + \gamma_2|1\rangle$, with $|\gamma_1|^2 + |\gamma_2|^2 = 1$. The description of the gate's effect can be conveniently written as a matrix with two rows and two columns.

$$G = \begin{bmatrix} \delta_1 & \gamma_1 \\ \delta_2 & \gamma_2 \end{bmatrix}$$

The effect of G on the arbitrary qubit $q = \alpha|0\rangle + \beta|1\rangle$ is the result of the dot product

$$G \cdot q = \begin{bmatrix} \delta_1 & \gamma_1 \\ \delta_2 & \gamma_2 \end{bmatrix} \cdot \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \delta_1 \cdot \alpha + \gamma_1 \cdot \beta \\ \delta_2 \cdot \alpha + \gamma_2 \cdot \beta \end{bmatrix}$$

The description of the most common unary quantum gates is given below. They have intuitive logical meaning. The transformation performed on the qubit can be easily seen, at least for particular values of the initial qubit.

2.2.1 The NOT Gate

The NOT gate reverses the roles of the $|0\rangle$ and $|1\rangle$. The coefficients of the base vectors are flipped. The matrix describing the NOT operation is denoted with X.

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

The equation describing the NOT gate in action is

$$X \cdot \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix}$$

2.2.2 The Phase-Shift Gate

The phase-shift gate, or Z gate, rotates the phase of the $|1\rangle$ by changing the sign of its coefficient. The transformation matrix of the Z gate is

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

If Z is applied on an arbitrary qubit, the following transformation happens

$$Z \cdot \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha \\ -\beta \end{bmatrix}$$

2.2.3 The Hadamard Gate

The Hadamard gate is a very common gate that transforms a simple state into a balanced superposition. The gate is defined by the transformation matrix

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

The Hadamard transformation is its own inverse. This can be seen from the fact that the Hadamard transformation matrix multiplied by itself results in the identity transformation.

$$H \cdot H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

If the Hadamard gate is applied twice to some qubit, that qubit is obtained again after the second application of the gate.

Applying the Hadamard gate to an arbitrary qubit, we have $H(\alpha|0\rangle + \beta|1\rangle) = \alpha\frac{|0\rangle+|1\rangle}{\sqrt{2}} + \beta\frac{|0\rangle-|1\rangle}{\sqrt{2}}$.

In particular, for a qubit in a simple state $q_{zero} = |0\rangle$, the Hadamard gate transforms q_{zero} into a balanced superposition:

$$H(q_{zero}) = H(|0\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle),$$

with its inverse transformation

$$H(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)) = |0\rangle = q_{zero}.$$

For the simple state $q_{one} = |1\rangle$, similar transformations exist:

$$H(q_{one}) = H(|1\rangle) = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle),$$

with the inverse transformation

$$H(\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)) = |1\rangle = q_{one}.$$

2.2.4 Sequence of Gates

A sequence of gates can be applied to a qubit. In this case the dot operator is simply applied in sequence from right to left. The sequence of a phase shift and NOT gate will be used in chapter 4.

$$Z \cdot X \cdot \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = Z \cdot \begin{bmatrix} \beta \\ \alpha \end{bmatrix} = \begin{bmatrix} \beta \\ -\alpha \end{bmatrix}$$

2.3 Entangled Qubits

A collection of several qubits can be viewed together, forming one single unity with a perspective on all qubits at once. Qubits considered together form an ensemble.

We consider q_A and q_B pertaining to two computational entities: Alice and Bob. These names are commonly used in cryptography, where Alice and Bob are two communication partners. This means that Alice "has" q_A and Bob "has" q_B. A general pair of these two qubits q_A and q_B is described by a superposition state of the form

$$q_A q_B = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle. \tag{2.1}$$

There is the additional condition on the state vector to be unitary, meaning of length one: $\sqrt{|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2} = 1$. The pair is a superposition of four states $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$. If these four states are equally likely, the superposition becomes, for example

$$q_A q_B = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle). \tag{2.2}$$

If we want to inspect the two qubits independently, their state should be clearly distinguishable. For the pair above, there exists the following equivalent rewriting as a tensor product:

$$q_A q_B = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle). \tag{2.3}$$

The two qubits have independent states. Each qubit is in a balanced superposition of $|0\rangle$ and $|1\rangle$. If one qubit of the pair is measured, it does not affect the state of the other qubit. The unmeasured qubit will still remain in a balanced superposition of $|0\rangle$ and $|1\rangle$.

If in equation 2.2 the two middle terms are missing

$$q_A q_B = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{2.4}$$

the state of the pair is still well defined. Yet, the behavior of the system is very different. The two qubits cannot be written as a tensor product. This fact can be

proven by contradiction. Suppose the pair *can* be written as a tensor product

$$q_A q_B = (a|0\rangle + b|1\rangle) \otimes (c|0\rangle + d|1\rangle) \quad (2.5)$$

$$= ac|00\rangle + ad|01\rangle + bc|10\rangle + bd|11\rangle. \quad (2.6)$$

Then the outer terms must be $ac = bd = \frac{1}{\sqrt{2}}$, and the middle terms must be zero: $ad = bc = 0$. If $ad = 0$, then either $a = 0$, or $d = 0$. If $a = 0$, then $ac = 0$. This is a contradiction. Else, if $d = 0$, then $bd = 0$, which is again a contradiction. Therefore, the pair cannot be written as a product of two independent qubits.

This conclusion entails profound consequences. The states of the qubits are dependent. From the point of view of their state, these qubits are inseparable, they are described by a single state [2]. In fact, they behave as one. Erwin Schrödinger called this phenomenon entanglement [41].

Let us look at the behavior of the entangled pair at measurement. Alice has the first qubit and Bob has the second. If Alice measures her qubit and sees a 0, then the pair has collapsed to $q_A q_B = |00\rangle$. Bob's qubit has collapsed to $|0\rangle$ as well. Bob will measure a 0 with certainty, that is, with probability 1. This is very interesting, as Alice's measurement value, "forces" Bob's qubit to take on a value compatible with Alice's measurement. Again, if Alice measures a 1, the pair has collapsed to $q_A q_B = |11\rangle$, and Bob will measure a 1 as well, with probability 1. Exactly the same scenario happens if Bob is the first to measure his qubit. Alice's qubit will collapse to the same binary value.

Note that any measurement on one qubit of this entanglement collapses the other qubit to a *classical* state. If Alice measures her qubit, she *knows* what value Bob will measure. Conversely, if Bob measures his qubit, he *knows* what value Alice will measure.

The entanglement pair defined above is one of the four Bell states:

$$\Phi^+ = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{2.7}$$

$$\Phi^- = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle). \tag{2.8}$$

$$\Psi^+ = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle). \tag{2.9}$$

$$\Psi^- = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle). \tag{2.10}$$

The Bell states are the most common form of entanglement. They also form a measurement basis. The primary choice to measure a pair of two qubits is the computational basis $\{|00\rangle, |01\rangle, |10\rangle$, and $|11\rangle\}$. Nevertheless, as the four Bell states are orthogonal, a pair of two qubits may well be measured in the Bell basis $\{\Phi^+, \Phi^-, \Psi^+$, and $\Psi^-\}$.

2.3.1 Entanglement Caused by Phase Incompatibility

When considering the algebraic description of an entangled qubit pair, we tend to imagine a sum with missing terms. The feeling is that if the expression has less than four terms, it may become impossible to rewrite the expression as a tensor product of independent qubits. Yet, the condition of missing terms is not necessary. Let us look at an unusual form of entanglement. Consider the following pair of two qubits:

$$\phi = q_A q_B = \frac{1}{2}(-|00\rangle + |01\rangle + |10\rangle + |11\rangle)$$

The pair has all four components, $|00\rangle, |01\rangle, |10\rangle$, and $|11\rangle$, in its expression. And yet, this pair is entangled.

The following is a proof. Suppose the pair ϕ is not entangled. This means ϕ can be written as a tensor product of two independent qubits:

$$\phi = q_A q_B = \frac{1}{2}(\alpha_1|0\rangle + \beta_1|1\rangle) \otimes (\alpha_2|0\rangle + \beta_2|1\rangle)$$

Matching the coefficients from each base vector, we have the following conditions:

1. $\alpha_1\alpha_2 = -1$
2. $\alpha_1\beta_2 = 1$
3. $\alpha_2\beta_1 = 1$
4. $\beta_1\beta_2 = 1$

The multiplication of conditions 1 and 4 have the result: $\alpha_1\alpha_2\beta_1\beta_2 = -1$. From conditions 2 and 3, we have: $\alpha_1\alpha_2\beta_1\beta_2 = 1$. This is a contradiction. The product $\alpha_1\alpha_2\beta_1\beta_2$ cannot have two values, both $+1$ and -1. It follows that ϕ cannot be decomposed and thus the two qubits are entangled.

The entanglement of the pair is caused by the *signs* in front of the four base vector components. Thus, it is not that some vector is missing in the expression of the pair, but the phases of the base vectors keep the two qubits entangled.

Note also that the measurement of the qubits is asymmetric. Alice and Bob do not perform the same operation. This feature will be exploited in the algorithms developed in the next chapter when checking for eavesdropping.

Measurement

Let us investigate what happens to the pair ϕ, when the entanglement is disrupted through measurement.

If the first qubit q_A is measured and yields $q_A = |0\rangle$ (a classical 0) then the second qubit collapses to $q_B = \frac{1}{\sqrt{2}}(-|0\rangle + |1\rangle)$. This is not a classical state, but a simple Hadamard gate transforms q_B into a classical state. We have $H(q_B) = H(\frac{1}{\sqrt{2}}(-|0\rangle + |1\rangle)) = -|1\rangle$. This is a classical 1.

The converse happens when qubit q_A yields 1 through measurement. In this case q_B collapses to $q_B = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Applying the Hadamard gate transforms q_B to $H(q_B) = H(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)) = |0\rangle$. Again this is a classical state 0.

It follows that by using the Hadamard gate, there is a clear correlation between the measured values of the first and second qubit. In particular, they always have opposite values.

A similar scenario can be developed, when the second qubit q_B is measured first. In this case, the first qubit q_A, transformed by a Hadamard gate, yields the opposite value of q_B.

2.3.2 Entanglement Verification

In quantum cryptography many protocols [20], [6], [44] rely on the availability of entangled qubit pairs. Each pair is shared by Alice and Bob. Alice has an array of qubits of a certain length, and Bob has an array of qubits of the same length. Every qubit of Alice's array is entangled with Bob's correspondent qubit.

Cryptography protocols face the threats of a third malevolent party, an eavesdropper, conventionally called Eve. In the case of entanglement, Eve may aim to destroy the entanglement between Alice and Bob. Thus, it is imperative that Alice and Bob have a way to ensure that their qubits are still entangled and untouched by the intruder Eve.

For example, when Eve reads one qubit q_B of an entangled qubit pair $q_A q_B = \Phi^+ = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, she destroys the entanglement. Henceforth, q_A and q_B become independent qubits. If she sees a 0 then the pair has collapsed to $q_A q_B = |00\rangle = |0\rangle \otimes |0\rangle$. These are independent qubits, each with a classical value of 0. If Eve sees a 1 then the pair has collapsed to $q_A q_B = |11\rangle = |1\rangle \otimes |1\rangle$. These qubits are again independent and carrying each the classical value 1.

Alice and Bob can test whether q_A and q_B are still entangled or the entanglement has been disrupted by a third party. The entanglement test is statistical, there is a probability (less than 100 %) that Eve's intervention is revealed. Also, after the test, the state of (q_A, q_B) is collapsed to the measured values and can therefore no longer be used in the secret communication protocol. Qubits that are tested for entanglement have to be discarded from any further use.

The first experiment to test for entanglement was based on Bell's inequality [20] and is not used here. A test for entanglement that does not use Bell's inequality is presented in [6]. The method proposed here is an abstract version of [6] as given in [29]. Consider the same qubits q_A and q_B, generated by a source of entangled qubits and then sent to Alice and Bob. Assume that $(q_A, q_B) = \Phi^+ = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ are entangled in the first Bell state. Without prior agreement, Alice randomly decides to measure her qubit q_A either directly (in the computational basis) or after applying a Hadamard gate to it. Bob, without knowing Alice's decision, also randomly measures q_B either directly or after applying a Hadamard gate to q_B. This would mean a measurement in the computational basis (direct measurement) or in a Hadamard basis (measurement after a Hadamard gate).

If Alice and Bob measure in the same measurement basis, there is a correlation

between the measured bit values. This happens when Alice and Bob both measure in the computational basis or in the Hadamard basis:

$$(I \otimes I)|\Phi^+\rangle = |\Phi^+\rangle. \tag{2.11}$$

$$\begin{aligned}(H \otimes H)|\Phi^+\rangle &= \frac{1}{\sqrt{2}}(H|0\rangle \otimes H|0\rangle + H|1\rangle \otimes H|1\rangle) \\ &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = |\Phi^+\rangle. \end{aligned} \tag{2.12}$$

In both cases, Alice and Bob measure the same bit value: both measure 0 or both measure 1.

If Alice and Bob have measured in different bases, there is no correlation between the measured values:

$$\begin{aligned}(H \otimes I)|\Phi^+\rangle &= (I \otimes H)|\Phi^+\rangle \\ &= \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle - |11\rangle). \end{aligned} \tag{2.13}$$

This state is still entangled, but as it is spread over all four states, Alice and Bob may measure either equal or different qubit values with the same probability. Therefore, whenever Alice and Bob have measured in the same basis, q_A and q_B have to yield the same binary value, otherwise q_A and q_B were not entangled to begin with.

It was said that when Eve measures the pair, she disrupts the entanglement. Suppose Eve has measured Bob's qubit. Eve's measurement was in the computational basis, without applying any transformation before measurement. The pair has now collapsed to a nonentangled state, say $q_A q_B = |00\rangle$. The other option $|11\rangle$ behaves

identically. If Alice and Bob, following their verification protocol, choose to measure in the same computational basis, the value of their measurement will agree: $q_A = |0\rangle$, $q_B = |0\rangle$. Eve's intervention remains hidden. Yet, if Alice and Bob both measure their qubit in the Hadamard basis, they will first apply a Hadamard transformation. Alice's qubit becomes $q_A = H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and yields a measurement of 0 or 1 with equal probability. Bob's qubit will behave similarly $q_B = H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, which again is measured as 0 or 1 with equal probability. Because the entanglement has been interrupted, the measured values are not correlated. Any combination of the binary values is possible. There is a 50% probability that the measured values will be different. In this case, Alice and Bob conclude that their qubits were not entangled to begin with. Eve's intervention is exposed.

If Eve knew in advance that Alice and Bob will check their qubits by measuring in the Hadamard basis, she also can behave so that she remains hidden. Recall that Eve has access to Bob's qubit. Eve first applies a Hadamard gate to Bob's qubit:

$$\begin{aligned} q_A(Hq_B) &= \frac{1}{\sqrt{2}}(|0\rangle(H|0\rangle) + |1\rangle(H|1\rangle)) \\ &= \frac{1}{\sqrt{2}}(|0\rangle\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) + |1\rangle\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)) \\ &= \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle - |11\rangle). \end{aligned} \tag{2.14}$$

Then Eve measures Bob's qubit and disrupts the entanglement. If Eve sees a 0, then the pair has collapsed to $q_Aq_B = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle = (H|0\rangle) \otimes |0\rangle$. If Eve sees a 1, then the pair has collapsed to $q_Aq_B = \frac{1}{\sqrt{2}}(|01\rangle - |11\rangle) = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \otimes |1\rangle = (H|1\rangle) \otimes |1\rangle$.

Eve now finally applies a Hadamard transformation to Bob's qubit and gives it back to Bob. The result is either $q_AHq_B = (H|0\rangle) \otimes (H|0\rangle)$ or $q_AHq_B = (H|1\rangle) \otimes$

($H|1\rangle$). In both cases, when Alice and Bob measure in the Hadamard basis, their binary values coincide. They will wrongly conclude that Eve has not meddled with the entanglement. Fortunately, Alice and Bob have a chance to catch Eve on this procedure too, if they measure their qubits in the computational basis. In this case, Alice and Bob each measure a binary 0 or 1 with equal probability. The values are not correlated. Therefore, there is a 50 % chance that the measured values will be opposite. Thus, Eve will be discovered with a 50 % probability.

The conclusion appears to be that Eve could remain hidden if she knew in advance what basis Alice and Bob will be measuring in. Nevertheless, this is not possible. Alice and Bob each choose their measurement basis randomly without prior agreement. It is only after measurement that Alice and Bob discuss the measurement method and the outcome. On each qubit there is a 50 % chance that Alice and Bob have measured in the same basis. On each such qubit there is again a 50 % chance to catch Eve's intervention, because of opposite binary values. Overall, there is a $50\% \times 50\% = 25\%$ chance to catch Eve on each qubit. Additionally, it is worth mentioning that there is no possible non-entangled pair (q_A, q_B) that consistently yields the same bit value when measured in both the computational basis and the Hadamard basis. This means Eve cannot remain hidden in both bases at once.

If Alice and Bob share an entire array of entangled qubits, they have to sacrifice a number of qubits for this checking. The more qubits they sacrifice, the higher the probability to catch the intruder, Eve. In fact, this probability can be made arbitrarily high. Consider for example that Alice and Bob check ten qubits. On each qubit checked, Eve has a probability of $75\% = \frac{3}{4}$ to escape being caught. Over ten qubits, Eve escapes with probability $(\frac{3}{4})^{10} = 0.0563$. This means that Eve is caught

with probability $p = 1 - (\frac{3}{4})^{10} = 1 - 0.0563 = 0.943 = 94.3\%$.

Chapter 3

Quantum Authentication in Cryptography

Cryptosystems aim to offer the means of exchanging secret messages securely and reliably. Suppose that two entities, Alice and Bob, want to exchange secret messages; specifically, Bob prepares a secret message to be sent to Alice. Unfortunately, all they have available is some *insecure* communication channel. This means that a malevolent third party, Eve, makes every effort to ruin the secrecy or content of Bob's message. Eve can listen to the communication channel to find out the content of Bob's message. And also, Eve can tamper with Bob's message, adding, deleting or editing parts of the message.

Privacy of a message and its authenticity is well satisfied in a private key cryptosystem setting. Alice and Bob share one and the same secret key, k_s. Bob uses the secret key for encryption and Alice consequently decrypts the message with the same key. As long as k_s is unknown to anybody else, the secrecy of the communication is satisfied. There exist various encryption/decryption functions using k_s, such that

the encrypted message reveals no information whatsoever about the content of the message, provided the key k_s is unavailable.

Under normal, practical conditions, for Alice and Bob to simply share a secret key is an unachievable ideal. It is *how* that secret key gets into Alice's and Bob's hand, which makes a simple private key cryptosystem totally unrealistic. The problem is for Alice and Bob to reach a consensus on the value of a secret key or on the encryption/decryption method, using insecure communication methods *only*.

Classical cryptographic results permeate commercial cryptosystems to date. Most of these cryptosystems rely on the principles of public key cryptography. Public key cryptosystems offer commercially satisfactory security levels. The security of the public key cryptosystem relies on the difficulty of inverting particular algebraic functions, also called "one-way" functions.

Secure communication in public key cryptosystems is achieved using two types of keys: a public key and a private key. If Bob wants to send a secret message to Alice, he uses the public key to encrypt the message. Alice then reads the message after using her private key for decoding. There are a few very important characteristics of the two keys implied in this communication. Alice's private key is secret, and not shared with anybody else. In particular, Bob does not need to know Alice's private key. This is a major advantage, as the private key is never seen on any communication channel and therefore, its secrecy is ensured. By contrast, Alice's public key is available to anybody. Bob needs to know it, and also an eavesdropper, Eve, has access to it. In order for the protocol to work, the public key is guaranteed to be protected. This means that there is a consensus about the public key value. Both Bob and Alice are sure that they use the correct, same public key. Eve cannot masquerade as Alice and

change the value of the public key, making Bob use a false public key to encrypt his message. This feature is crucial for a public key cryptosystem to work. The public key cryptosystems need the public key to be protected, and accept it as given that such a protection of the public key is practically possible. Our quantum authentication protocols make use of this property of the protected public key. It is crucial in both the classical sense of authentication protocols, as well as in our protocol, that such a public key can be published with the guarantee that the key *is and remains* protected from masquerading.

The security of the public key distribution protocol relies on the theoretically unproven assumption that factoring large numbers is intractable on classical computers. As described in [35], quantum computers can break some of the best public key cryptosystems. Current public key algorithms, such as the RSA [38], need to continuously increase the length of the protected public key in order to maintain acceptable security levels.

Quantum cryptography aims to design mechanisms for secret communication with higher security than protocols based on the public key approach. Quantum cryptosystems take their inspiration from the private key system. A quantum protocol is designed to *distribute* a secret key to Alice and Bob. As such, key distribution is the process whereby two parties reach an agreement on the value of a secret key.

Some authors refer to the process as *key enhancement*, under the justification that a true secret key can be distributed only if a *small* secret key is already shared by Alice and Bob and used for authentication. Indeed, all quantum key distribution algorithms to date are in fact key enhancement algorithms. We prove in this thesis that such a small secret key is not necessary, and that true key distribution lies within

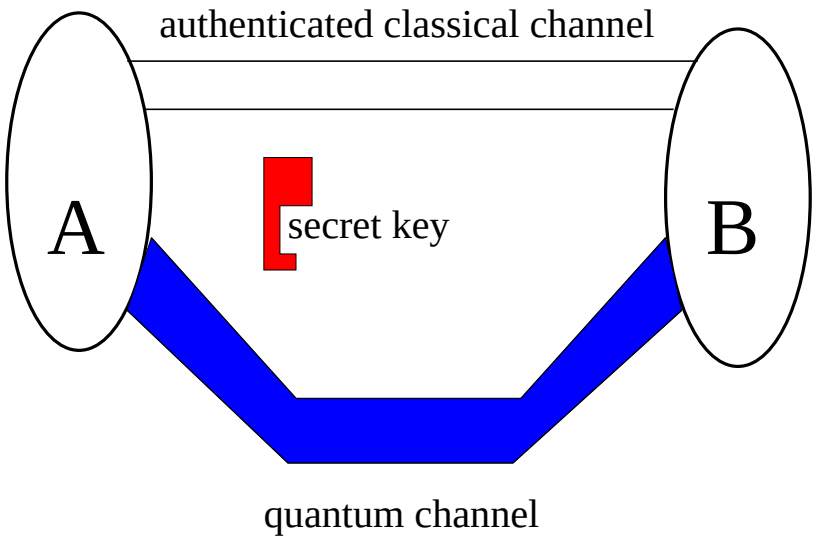

Figure 3.1: Historically, quantum key distribution protocols use two communication channels: one authenticated classical channel, and one quantum channel.

the capabilities of quantum cryptography. We present two algorithms that perform a genuine key distribution to Alice and Bob. This is done in different communication settings: with and without a classical communication channel. Therefore, we will continue to use the term of key distribution throughout the following exposition.

Quantum key distribution protocols establish secret keys via insecure quantum and classical channels. Existing quantum key distribution algorithms generally use two communication channels (see Fig. 3.1) between Alice and Bob: a quantum channel which transmits qubits and a classical channel for classical binary information. The classical channel is used to communicate the measurement strategy, or the basis for measurement, and to check for eavesdropping.

3.1 The History of Quantum Key Distribution Protocols

Quantum key distribution protocols may derive their efficiency from different quantum properties. The idea of applying quantum methods in cryptography dates back to Stephen Wiesner's concept of quantum unforgeable money [48]. This event, in the 1960's marks the beginning of quantum cryptography.

The first actual protocol was developed by Charles Bennett and Gilles Brassard, and is known as the BB84 protocol [5]. It is a key enhancement protocol and relies on measuring qubits in two different orthonormal bases. Alice and Bob have two communication channels, as in Fig. 3.1, a classical channel and a quantum channel. The classical channel needs to be authenticated using a small secret key previously known to only Alice and Bob. This is why the protocol is called a key enhancement protocol. On the quantum channel Alice can send quantum bits to Bob. As most quantum key distribution protocols, the protocol is a sequence of two phases. The first phase implies communication on the quantum channel. The second phase performs a checking of the events of the first phase. During the second phase, Alice and Bob communicate over the classical channel.

Alice possesses an array of qubits. She manipulates each qubit to align to one of two orthogonal bases. She then sends the qubit to Bob, who randomly measures it again in one of the two orthonormal bases.

As an example for a photon's polarization, the following convention can be made [27]. One measurement basis is vertical/horizontal, the other is an oblique basis, with the two directions rotated at 45°. Then the binary values are

Alice	0	1	1	0	0	1	0	0	0	1	0	0	1	1	0
	×	+	×	+	+	+	×	×	+	+	×	+	×	×	×
	↗	↑	↖	→	→	↑	↗	↗	→	↑	↗	→	↖	↖	↗
Bob	+	+	×	+	×	×	+	×	×	+	+	+	×	+	×
	1	1	1	0	0	1	1	0	1	1	0	0	1	1	0
key		1	1	0				0		1		0	1		0

Table 3.1: BB84 protocol for quantum key distribution in the absence of eavesdropping.

$$\begin{cases} \text{“0”} & = & |\rightarrow\rangle \\ \text{“1”} & = & |\uparrow\rangle \end{cases}$$

in the case of the vertical/horizontal basis, and

$$\begin{cases} \text{“0”} & = & |\nearrow\rangle \\ \text{“1”} & = & |\nwarrow\rangle \end{cases}$$

for the oblique basis.

After all the array of qubits is measured by Bob, Alice and Bob start the second phase of communicating on the classical channel. On the classical channel, they reveal their respective measurement bases and retain only the values of the qubits treated in the same base. As there are two bases to be chosen from randomly, there is a 50% chance for Alice and Bob to hit the same basis. When measured in the same basis, the measured binary values will coincide, giving Alice and Bob consensus over the array of binary digits: the raw secret key. Table 3.1 shows an example of 15 qubits and how the raw secret key is formed.

In order to check whether the intruder Eve has meddled with the communication, Alice and Bob have to check the values of some of the qubits of the raw key. These qubits sacrificed for checking will be discarded from the final key. Eve will be detected

if she has measured some qubits or has replaced some qubits with qubits of her choice. On each qubit, there is a 50% chance to catch Eve. Therefore, overall, on each initial qubit that Eve has tampered with, there is a 25% chance that Eve will be caught. The confidence of catching Eve is increased exponentially with the number of qubits used for checking.

The idea of BB84 applies to any two nonorthogonal bases [9]. In [28] the quantum key distribution algorithm is derived from the quantum Fourier transform. Based on the property of entanglement, Artur Ekert [20] gave a quantum key distribution solution using entangled qubits to be shared by Alice and Bob. A simpler version with qubits entangled in the same way, namely in the Bell states, is described in [6].

Shi et. al [44] describe in their paper a quantum key distribution algorithm that does not use a classical channel at all. Authentication is done by a trusted authority, that provides the entangled qubits to Alice and Bob. This is a strong assumption. In our algorithm, such a trusted authority is not needed. The entangled qubits may come from an insecure source.

3.1.1 Classical Information Is Public

There is an implicit property of the classical channel in *all* quantum key enhancement protocols. Information on the **classical channel is essentially public**. Eve is allowed to listen to the classical channel. The information exchanged on the classical channel does not reveal any information about the value of the secret key. This is specific for quantum key enhancement protocols. Again, this idea has never been formulated explicitly before. It has been said that the classical channel is insecure, but fundamentally the classical channel is public. We will use this idea when we show

that quantum protocols do not need classical authentication methods. Quantum authentication has been thought to be impossible.

The protocols described in section 3.1 require the classical channel to be authenticated. If the channel were not authenticated, Eve could masquerade on both channels (classical and quantum) such that Alice *never* speaks to Bob, but only to Eve. In the same way Bob is only connected to Eve and *never* speaks to Alice. In this case both Alice and Bob have no way to detect the masquerader Eve. Therefore, the classical channel, if authenticated, prevents this situation from happening.

Now here is the interesting characteristic of the BB84 algorithm and all similar ones. **The classical authentication step authenticates *public* information and public information only**. In this, the quantum key enhancement protocols are unique.

It follows that public information does not need a communication channel. Public information does not need to be authenticated by authenticating the communication channel. The problem of a certain public information reliably belonging to a certain source (say Alice) is no longer a problem of authentication. It is reduced to *protecting* the public information published by Alice. Normally, under commercially viable circumstances, this is accepted to be possible. As an example, publishing a telephone number in a telephone book, is accepted to provide accurate information, for which the telephone company is responsible. "Eve" cannot masquerade as someone else in a telephone book. It is therefore reasonable to consider that there are means to publish *protected* public information, and these means are available to Alice and Bob. Note that all cryptosystems that use public keys presuppose and crucially rely on the fact that a public key *can* be posted in a protected manner.

3.2 Catch 22

The drawback in all existing quantum cryptography algorithms is that for the algorithm to work, the classical channel needs to be authenticated. Authentication is supposed to be done by classical means. It can be done using a small secret key. The authenticated classical channel prevents Eve from masquerading as someone else and tampering with the communication. This means that in order to distribute a (larger) secret key between Alice and Bob, a small secret key needs to be used to authenticate the classical channel. This small secret key has to be shared between Alice and Bob prior to the quantum key distribution protocol. Therefore, these protocols are in fact quantum key *enhancement* protocols. It has been claimed by Lomonaco [39] that authentication is altogether not possible in quantum computation, that for any secure quantum communication a classical authentication scheme needs to be used.

We show that quantum cryptography has more to offer than key enhancement: authentication can be done by quantum means only. In fact, Alice and Bob can reach a consensus about the value of a secret key without sharing *any* secret information prior to the quantum algorithm that distributes this key. Moreover, the secret key can be arbitrarily large and consequently can be used as a one-time pad. All classical information exchanged between Alice and Bob is intrinsically public. This means that it is accessible to any eavesdropper or masquerader.

In our first algorithm the classical channel is not authenticated. Yet Alice and Bob do have an authentication step at the end of the protocol, with the help of protected public keys. Authentication is derived from the quantum algorithm itself and can catch any masquerading over the classical channel. Further, our second algorithm shows that a classical channel is not needed at all. The drawback of not using a

classical channel is that the amount of qubits used is larger for the same length of the final secret key and the same security level.

Lomonaco in his "Talk on Quantum Cryptography or How Alice Outwits Eve" [39] describes in a humorous way the basics of classical cryptography and then the qualitative addition of quantum cryptography. As a weakness of all cryptographic schemes to date, Lomonaco talks of a circuitous argument that ultimately cannot be circumvented. It is known as the *Catch 22* of classical cryptography, namely:

> **Catch 22.** *There are perfectly good ways to communicate in* secret, *provided we can communicate in* secret *...*

Classical cryptography is subject to this catch and according to the literature to date, quantum cryptography falls in the same category. Our two algorithms prove however, that quantum cryptography steps out of these limits. Indeed, secret communication using quantum methods does not need any prior secret or secure private communication. It only needs *public* communication. This is the strongest requirement, namely, that "some limited" public information is protected. This means that this public information is guaranteed to come from the expected source (e.g., Alice) and that the information is truthful: An eavesdropper Eve could not tamper with the contents of this information and could not masquerade as Alice. These ideas of protected public information are well known in public key cryptosystems and are referred to as the public keys. Alice publishes her public key in a secure, protected place, such as the yellow pages of a telephone book. The key is available to everybody. Eve can see the key but cannot tamper with it. Bob can see/read Alice's public key and is *absolutely certain* that he now possesses Alice's correct public key. Note that classical cryptographic protocols, the public key cryptosystems for instance, rely on

the fact that Alice *is able* to publish her key in this secure way.

The protocols presented in this chapter rely on public communication only, therefore weakening the requirements for secret communication. In conclusion, for quantum cryptography, Catch 22 has to be reformulated as:

> **Quantum Catch 22.** *There are perfectly good ways to communicate* secretly*, provided we can communicate* publicly *in a protected way ...*

In our quantum key distribution protocols, Alice and Bob will have one public key each. The keys will be developed during the algorithm and posted at the end. These public keys are regular binary numbers, but differ in meaning from the conventional public key, such as the RSA key [38]. We will call them *quantum generated public keys*. Alice has a protected quantum generated public key and Bob has another protected quantum generated public key. In fact these two public keys are the only *protected information* exchange between Alice and Bob. Exactly as in the case of classical public key cryptosystems, our algorithm requires that such public keys can be published in a protected way, with the guarantee that the keys' values *are and remain* protected from masquerading. As will be seen from the algorithm itself, besides having public keys, Alice and Bob share only insecure information: classical and/or quantum.

The following two algorithms make use of these public keys. They are used for authentication.

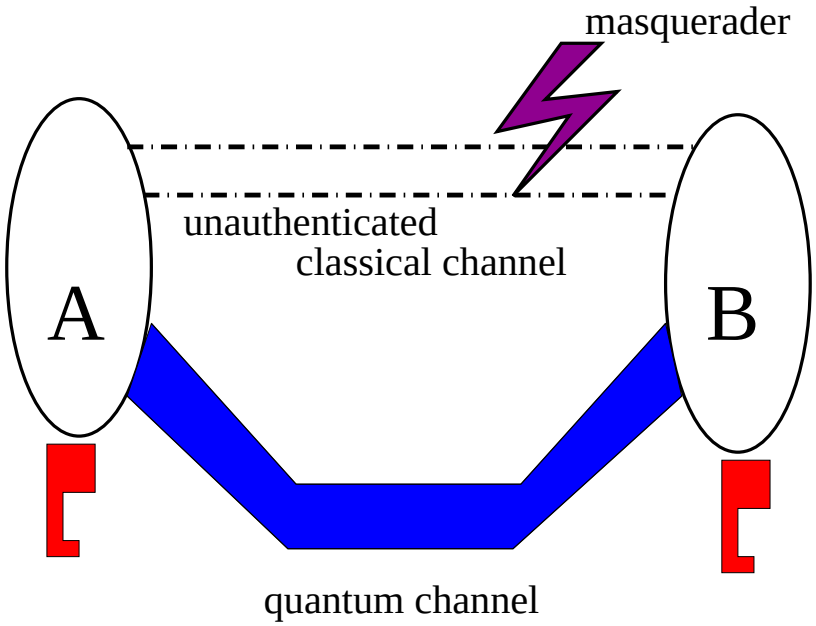

Figure 3.2: This quantum key distribution algorithm uses one insecure classical channel and one insecure quantum channel.

3.3 Quantum Key Distribution Algorithm with Unautheticated Classical Channel

The first algorithm works in the setting shown in Fig. 3.2. Alice and Bob are connected by a quantum channel and an unauthenticated classical channel. The quantum channel consists of an array of entangled qubit pairs shared by Alice and Bob. In addition, Alice and Bob will generate two protected public keys.

The goal of the protocol is for Alice and Bob to establish a secret key, to be used henceforth to encrypt/decrypt messages. One session is required to establish a classical secret key, called *secret*, such that Alice and Bob are in consensus about the value of the secret key. The secret key *secret* consists of n bits, $secret = b_1 b_2 ... b_n$.

On the classical channel, classical binary information can be exchanged. The

channel is *unprotected* and *not authenticated*. Thus, being unprotected, it is sensitive to attacks of eavesdropping: Eve may attempt and successfully read information from the channel. Also, the channel, not being authenticated, is sensitive to masquerading: Eve may disconnect the channel and then talk to Alice pretending she is Bob, and talk to Bob pretending she is Alice.

The array of the entangled qubits has length l, it consists of l qubit pairs denoted $(q_{1A}, q_{1B}), (q_{2A}, q_{2B}), ..., (q_{lA}, q_{lB})$. The array is split between Alice and Bob. Alice receives the first qubit of each entangled qubit pair, namely q_{1A}, q_{2A}, ..., q_{lA}, and Bob receives the second half of the qubit pairs, q_{1B}, q_{2B}, ..., q_{lB}. The entanglement of the qubit pair is of the type described in the previous chapter, namely, phase incompatibility:

$$q_A q_B = \frac{1}{2}(-|00\rangle + |01\rangle + |10\rangle + |11\rangle).$$

The array of qubits is unprotected. There is no guarantee that the qubits of a pair are indeed entangled, as Eve may have disrupted the entanglement. Also, Eve may have masqueraded as either Alice or Bob, modifying the entangled qubits, such that Alice's qubit is actually entangled with a qubit in Eve's possession rather than Bob's, and the same holds for Bob.

Two public keys are needed by the algorithm. Alice has a public key key_A and Bob has a public key key_B. The two public keys key_A and key_B are independent. These keys are necessary for authentication. They have some characteristics that are different from the classical public keys. The keys are established *during* the computation. They are not known prior to the key distribution algorithm and are defined in value during the computation according to the measured values of some of the qubits. This means that the keys are available *after* the key distribution protocol.

Consequently, the keys have to be posted after the algorithm, which is unlike the classical case, where a public key is known in advance. Also, the two public keys key_A and key_B are valid for one session, for one application of the key distribution algorithm. If Alice and Bob want to distribute a second secret key using the same algorithm, they will have to create new public keys, which are different in value from the public keys of the previous session.

The key distribution algorithm, like all quantum key distribution algorithms, develops the value of the secret key during the computation. Implicitly, the values of the public keys as well are developed *during* the computation. There exists no knowledge whatsoever about the values of the keys (secret and public) prior to running the algorithm.

The algorithm follows the steps below:

- **Step 1 - Establish the value of the secret key**

  For each entangled qubit pair (q_{iA}, q_{iB}) in the array, the following actions are taken. On the classical channel, Alice and Bob decide randomly who is going to perform the first measurement.

  Suppose it is Alice. Therefore, Alice measures her qubit q_{iA} thereby collapsing Bob's qubit q_{iB} to the state consistent with Alice's measurement. If Alice has measured a 0, $q_{iA} = 0$ then Bob's qubit has collapsed to $q_{iB} = \frac{1}{\sqrt{2}}(-|0\rangle + |1\rangle)$. If Alice's measurement resulted in $q_{iA} = 1$ then Bob's qubit collapsed to $q_{iB} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Now, Bob transforms his qubit via a Hadamard gate. For Alice's $q_{iA} = 0$, Bob has a $Hq_{iB} = 1$, and conversely for Alice's $q_{iA} = 1$, Bob has a $Hq_{iB} = 0$. Bob now measures and his value will consistently be the complement of Alice's.

If Bob is the one who measures first his qubit q_{iB}, the procedure is simply mirrored. Alice now has to apply a Hadamard gate on her qubit, thus obtaining Hq_{iA}. Again Alice and Bob will have measured complementary binary digits.

Ideally, with no interference from Eve, be it through eavesdropping or masquerading, applying this measurement and Hadamard measurement on each qubit is enough to establish the secret key. After going through all the qubit pairs, Alice and Bob will have complements of the same binary number. This, for example Alice's binary number, is the established secret key. Then Bob will simply invert each bit that he measured to obtain the same key as Alice.

- **Step 2 - Authentication and Eavesdropping Checking**

  Some $2m$ qubits will be sacrificed for security checking, where $2m < l$. The secret key will be formed by the remaining $n = l - 2m$ qubits. Alice and Bob decide via the classical channel, the set of qubit indices to be sacrificed. Alice looks at the first m qubits, and forms a binary number with the values she reads. This is Alice's *public key*. Alice now publishes her public key, which can be seen by Bob. As the public key is protected, Bob is certain that the public key is safe from masquerading. Note that this is the only step, where Bob is certain to have contact with Alice with no masquerading. Bob now compares Alice's public key with his own measured qubits. If these two binary numbers are complementary, then Bob concludes that he has been talking with Alice all the while in the previous step, and also that no eavesdropper has changed some of the qubit values. The same procedure applies to Bob's *public key* formed by the second m sacrificed qubits. If Bob or Alice have encountered a mismatch in the values checked, they discard the secret key and try again.

Alice's qubits: q_{1A}, q_{2A}, ..., q_{lA}. Bob's qubits: q_{1B}, q_{2B}, ..., q_{lB}.

Establish the secret key
for every entangled pair (q_{iA}, q_{iB}) do
on the **classical channel**:
Decide randomly who is to measure first.
on the **quantum channel**:
if Alice measures first then
Alice measures q_{iA}.
Bob applies a Hadamard gate and measures q_{iB}.
elseif Bob measures first then
Bob measures q_{iB}.
Alice applies a Hadamard gate and measures q_{iA}.

Check for eavesdropping on the **classical channel**: Alice and Bob choose $2m < l$ qubit indices to be sacrificed for checking. **public protected publishing**: Alice publishes the value of the first m qubits. This is Alice's public key. Bob publishes the value of the last m qubits. This is Bob's public key. if the public key of the other person is a complement of what I have then Eve has not meddled with the communication. else Eve has meddled with the communication. Retry the whole protocol.
Form the secret key from the remaining qubit measurements. Bob complements his key.

Table 3.2: Key distribution with unauthenticated classical channel.

Table 3.2 summarizes the key distribution algorithm described above.

Let us analyze what Eve can do to get some knowledge about the secret key without being caught.

Note that Eve can have access to the qubits before they reach Alice and Bob, prior to the actual protocol. For example, Alice generates entangled qubit pairs. From each pair, she sends one qubit to Bob. The qubits are sent to Bob insecurely: this

is the moment where Eve can act upon the qubits through measuring or exchanging qubits with her own. The initial distribution of entangled qubits is therefore insecure. Eve is allowed any intervention of her choice on the qubits. Alice and Bob have no guarantee that the qubits in their possession are indeed entangled with each other as expected. *But*, and this is essential for the security of the algorithm, once Alice and Bob, respectively, have their qubits in their possession, Eve has no more access to them. This is a perfectly reasonable assumption, that Alice and Bob can guard the qubits in their possession from outside attacks. Eve cannot measure or exchange a qubit that has already reached Alice and is in her possession. As such, *if* Alice and Bob have received a correctly entangled qubit pair before they have started the protocol, Eve has no chance to disrupt the entanglement afterwards. This means that Eve's attack on the entangled qubits can happen only prior to any decision concerning measurement. This property is even stronger in the next algorithm with no classical communication, as the decision on how to measure a qubit (directly or with a Hadamard gate) is not made on a classical communication channel, but in private by Alice and Bob independently. Now, the advantage of an asymmetric measurement scheme (one party measures directly, the other party applies a Hadamard gate) can be clearly seen. Eve has no gain in measuring a qubit of the entanglement before the protocol is started. For example, if she does measure Alice's qubit directly, the later random measurement decision during the protocol may result in a Hadamard gate measurement for Alice, thus exposing Eve's intervention.

To evaluate a specific attack, let us consider that Alice's and Bob's qubits are not really entangled, but Eve has sent qubits of her own choice to both of them. Eve also can listen to the classical channel. The best she can do is send a classical 0 to Alice

and a *Hadamard* 1 to Bob. Actually, all other combinations are equivalent to, or less advantageous than, this one. Alice and Bob decide who is to measure first once they already have the qubits. With 50% probability, Alice is measuring, in which case the readings are consistent. Bob measures first, again with probability 50%. Bob will measure a 1 or 0 with equal probability. Then Alice transforms the classical 0 with a Hadamard gate, and also reads a 0 or 1 with equal probability. This means that if Bob measures first, Alice and Bob will read the same value with 50% probability. As such, Eve is caught with 25% probability. If the number of qubits to be tested is large, this probability can be made arbitrarily large.

Note that the classical channel does not need to be authenticated. Eve can masquerade, such that she completely severs all connection between Alice and Bob. In this case, Eve will establish one secret key with Alice, and another secret key with Bob. Unfortunately for Eve, she has no control over the value of the secret keys, as their values are determined probabilistically, through quantum measurement. Therefore, the two keys will necessarily differ. As Alice and Bob publish part of their secret keys as protected public keys, they will notice the difference and consequently discard the keys.

This algorithm features a few notable characteristics. Checking for eavesdropping and authentication happens in one and the same step. Until this checking phase, there is no certainty whatsoever about either the validity of the key, the validity of the classical connection or the quantum qubit entanglement. But then, the key is not *useful* or *used* before the checking step.

Another interesting feature is the way in which the two public keys are used in our case. In classical settings, the public key is established and known by both

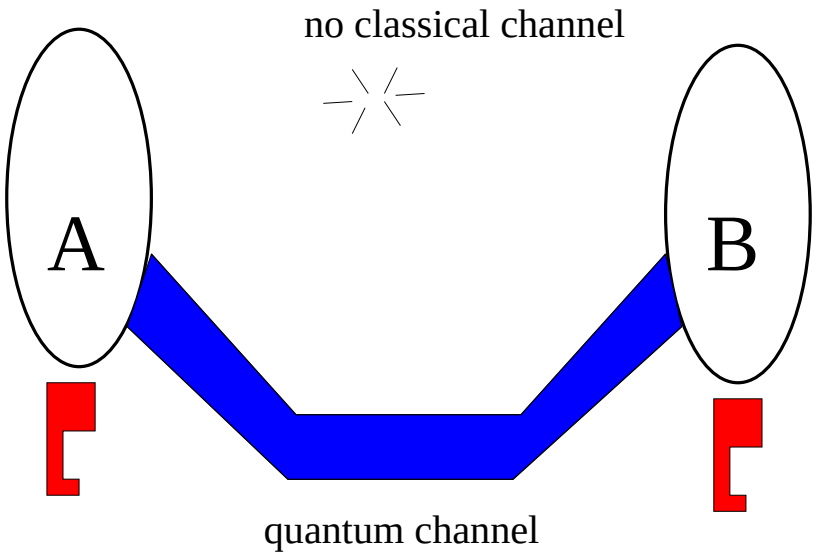

Figure 3.3: This quantum key distribution algorithm uses only one insecure quantum channel.

parties, before the algorithm or communication begins. By contrast, in this quantum distribution algorithm, the public keys are determined *during* the computation and they are available only after the secret key is established. The publishing of the public keys is the very last step of Alice and Bob's communication, whereas in previous algorithms this is the first step. The usage of the public keys is in reverse order by comparison with classical secret communications, such as those in the public key cryptosystems.

3.4 Quantum Key Distribution Algorithm without Classical Channel

As will be clear from the algorithm below, authentication of a quantum communication protocol is not only possible by quantum means only, but in fact a classical

channel is superfluous. In the present improved version of the protocol, the classical channel is removed completely (see Fig. 3.3). The robustness of the algorithm comes also from the simplicity of the communication support available. Alice and Bob share an insecure quantum channel and two quantum generated public keys. They have an authentication step at the end of the protocol, with the help of the quantum generated public keys. Note that authentication in this algorithm, as in the previous algorithm, is done at the end of the protocol and is derived from the quantum algorithm itself.

The goal of the key distribution algorithm described below is to establish a secret key, known only to Alice and Bob. Subsequently, when Alice and Bob exchange messages, they will use this key to encrypt/decrypt their messages. One session is required to establish a binary secret key, called *secret*, such that Alice and Bob are in consensus about the value of the secret key. The secret key *secret* consists of n bits, $secret = b_1 b_2 ... b_n$. Technically, to perform the algorithm, Alice and Bob need an array of entangled qubit pairs, and two protected public keys. Note that Alice and Bob do not communicate on *any* classical channel.

The quantum communication channel is identical to the one in the previous algorithm. It consists of an array of entangled qubits. The array of entangled qubits has length l, it consists of l qubit pairs denoted $(q_{1A}, q_{1B}), (q_{2A}, q_{2B}), ..., (q_{lA}, q_{lB})$. The array is split between Alice and Bob. Alice receives the first qubit of each entangled qubit pair, namely $q_{1A}, q_{2A}, ..., q_{lA}$, and Bob receives the second half of the qubit pairs, $q_{1B}, q_{2B}, ..., q_{lB}$. The entanglement of a qubit pair is of the type caused by phase incompatibility:

$$q_A q_B = \frac{1}{2}(-|00\rangle + |01\rangle + |10\rangle + |11\rangle).$$

The array of qubits is unprotected. There is no guarantee that the qubits of a pair

are indeed entangled; indeed, Eve may have disrupted the entanglement or may have masqueraded. In case Eve has disrupted the entanglement or has masqueraded, any result of the algorithm is discarded and the key distribution is attempted all over again, from the beginning.

The size n of the secret key is less than half of the length l of the initial qubit array, $n < \frac{l}{2}$. Indeed, $\frac{l}{2}$ qubits, that is half of the qubits, are discarded because the bases in which Alice and Bob measure are inconsistent 50% of the time. From the remaining half of qubits a further arbitrary number is sacrificed for security checking. The number of qubits thus sacrificed depends on the desired degree of security.

Two public keys are needed by the algorithm. Alice has a public key key_A and Bob has a public key key_B. The two public keys key_A and key_B are independent. Alice and Bob use these public keys to exchange classical binary information and also, very importantly, for authentication. The keys, unlike classical public keys, are established *during* the computation. They are not known prior to the key distribution algorithm and are defined in value during the computation according to the measured values of some of the qubits. This means that the keys are available *after* the key distribution protocol. Consequently, the keys have to be posted after the algorithm, which is unlike the classical case, where a public key is known in advance.

Also, the two public keys key_A and key_B are valid for one session, that is, for one application of the key distribution algorithm. If Alice and Bob want to distribute a second secret key using the same algorithm, they will have to create new public keys, which are different in value from the public keys of the previous session.

This algorithm, like all quantum key distribution algorithms, develops the value of the secret key during the computation. Implicitly, the values of the public keys as

well are developed *during* the computation.

Both Alice and Bob follow the same steps briefly denoted below:

1. Measure your entangled qubits.

2. Compute your own public key and post it.

3. Read your partner's key and check for eavesdropping.

4. Construct the value of the secret key.

A detailed description of the algorithm follows.

Step 1

Alice works with the array of qubits q_{1A}, q_{2A}, ..., q_{lA}. Binary information is rendered by the results of measuring. Alice has two options for processing her qubits. She either measures a qubit directly in the computational basis, or she transforms the qubit by a Hadamard gate and measures afterwards. For each qubit, q_{iA}, Alice decides randomly on one of the two processing options. Notably, there is no communication with Bob at this stage. To look at a concrete example, suppose Alice has 10 qubits q_{1A}, q_{2A}, ..., q_{10A}. Qubits q_{iA} transformed by the Hadamard gate are denoted Hq_{iA}; for those measured directly the notation is unchanged. Suppose that by random choice, Alice has processed her qubits as follows:

$$q_{1A}, Hq_{2A}, Hq_{3A}, q_{4A}, q_{5A}, q_{6A}, Hq_{7A}, Hq_{8A}, q_{9A}, q_{10A},$$

and suppose again, she has measured the following binary values:

$$1, 1, 1, 0, 0, 0, 0, 1, 1, 1$$

In the meantime, Bob processes his qubits q_{1B}, q_{2B}, ..., q_{10B} following the same policy. He too, has a random choice on each qubit: to measure directly or to measure after a Hadamard transformation. Suppose again, that by random choice, Bob has obtained the following array:

$$Hq_{1B}, Hq_{2B}, q_{3B}, Hq_{4B}, q_{5B}, q_{6B}, q_{7B}, Hq_{8B}, Hq_{9B}, q_{10B},$$

with the values

$$0, 1, 0, 1, 1, 0, 1, 0, 0, 1$$

We have seen in the previous chapter that two entangled qubits $q_{iA}q_{iB} = \frac{1}{2}(-|00\rangle + |01\rangle + |10\rangle + |11\rangle)$, consistently render opposite classical bit measurements, if and only if exactly one qubit is measured directly and the other is measured after a Hadamard transformation. It is of no consequence whether the first qubit is Hadamard tranformed or the second. The order of the qubits is irrelevant, the important issue is that exactly one of the qubits is passing a Hadamard gate. Thus, there are two "valid" measurement options:

1. q_{iA}, Hq_{iB} and

2. Hq_{iA}, q_{iB}

These measurement scenarios are valid in the sense that they, and only they, yield consistently opposite classical bits after measurement. Each of Alice and Bob knows with certainty the value the other person has measured. Such qubits are considered valid by Alice and Bob and will be used to form the secret key and to check for eavesdropping.

Measurements of the form

3. q_{iA}, q_{iB} and

4. Hq_{iA}, Hq_{iB}

cannot be used by Alice and Bob. For any value measured by Alice, the value measured by Bob is still determined probabilistically. Qubits measured according to these scenarios, will unfortunately have to be discarded. As the scenarios 1, 2, 3, 4 are equally likely, 50% of the initial qubits will be discarded because of probabilistically inconsistent measurements.

As mentioned, half of the l qubits are discarded because of incompatible measurement bases. The size n of the secret key is therefore $n < \frac{l}{2}$. From the remaining qubits, depending on the desired security level, some other qubits are sacrificed for checking.

For the example of the 10 qubits given above, there are five valid qubit-pairs:

$$(q_{1A}, Hq_{1B}), (Hq_{3A}, q_{3B}), (q_{4A}, Hq_{4B}), (Hq_{7A}, q_{7B}), (q_{9A}, Hq_{9B}),$$

carrying the values

$$(1,0), (1,0), (0,1), (0,1), (1,0)$$

Step 2

At this point Alice has no idea what measuring option Bob has employed on his qubits. She does not know that qubits 1, 3, 4, 7, and 9 are valid. Bob is in the same situation.

Therefore, Alice will publish her measuring strategy in her public key. Alice has measured $l = 10$ qubits. As such, the first l bits of Alice's public key explain which qubits have been Hadamard transformed and which were measured directly. If Alice

has applied the Hadamard gate on qubit q_{iA} then the i-th bit of the public key is set to 1, $key_A(i) = 1$. Otherwise, if q_{iA} has been measured directly, then the i-th bit is 0, $key_A(i) = 0$. For the example of 10 qubits, the first ten bits of Alice's public key are

$$key_A(1..10) = 0110001100$$

The second part of Alice's public key is used for security checking. A certain fraction f, for example $f = 40\%$, of the original qubits are made public for Bob to check for eavesdropping. Alice chooses randomly 40% of her l qubits. For each chosen qubit, Alice publishes the index of the qubit and the binary value she has measured. To continue our example, Alice chooses randomly the indices 1, 2, 9, 10. She will publish index 1 with value 1, index 2 with value 1, index 9 with value 1 and index 10 with value 1. Translated in binary this is

$$(0001)1(0010)1(1001)1(1010)1$$

Alice's final public key is the concatenation of the measuring (Hadamard / no Hadamard) information and the qubit checking information:

$$key_A = 0110001100 \;\; 0001\,1 \;\; 0010\,1 \;\; 1001\,1 \;\; 1010\,1$$

The length of the public key depends on the length l of the qubit array and also on the desired security level given by the fraction f. The following formula computes the length of the key:

$$length(key_A) = l + f(1 + \log l)$$

Here, l, the first term in the sum, refers to the measuring strategy; the second term, $f(1 + \log l)$, represents the part that publishes the qubits for eavesdropping checking.

Bob creates his public key following exactly the same steps. Bob's measuring strategy is encoded at the beginning of his public key. For our example, this means

$$key_B(1..10) = 1101000110$$

Suppose Bob sacrifices qubits 1, 5, 7, 8 for checking. In his public key he will publish (0001)0(0101)1(0111)1(1000)0. Thus, Bob's final key, the one that Alice and indeed everybody can see, is:

$$key_B = 1101000110 \quad 0001 \quad 0 \quad 0101 \quad 1 \quad 0111 \quad 1 \quad 1000 \quad 0$$

Both Alice's and Bob's keys, key_A and key_B are made public and are available to everybody, including Eve.

Step 3

At this stage, Alice and Bob, in full knowledge and consensus of each other's publicly published keys, will proceed to check for eavesdropping. Alice is looking at Bob's public key key_B and learns the values Bob has measured on the randomly sacrificed $f = 40\%$ of his qubits, here qubits 1, 5, 7, 8. Because of the various measuring options, only half of the $f = 40\%$ qubits will be useful. In our example, qubits 1 and 7 are measured with correct options, namely exactly one Hadamard gate applied to an entangled pair. Alice can find out the valid qubits by XOR-ing the measuring strategy of Bob with her own:

$$(0110001100)XOR(1101000110) = (1011001010)$$

which means qubits 1, 3, 4, 7, 9 have been measured well. Alice is left only to compare the values of qubits 1 and 7 she has measured with the values posted by

Bob. With no malevolent interference, the binary values are opposite. Thus, if these values are opposite, Alice concludes that the protocol was not influenced by Eve. Otherwise, Alice discards all information and starts all over again. Bob performs the same checking. He will find the valid qubits posted by Alice, 1 and 9, and will compare Alice's binary measured values with his own. Thus Bob makes his own independent decision concerning eavesdropping. For reasonably large qubit arrays and a resonably large number of qubits checked, Alice and Bob will reach the same conclusion concerning the validity of the measured binary data. This conclusion effectively implies the absence of eavesdropping/masquerading (assuming, of course, that the qubits were initially entangled).

Step 4

At this stage, the possibility of eavesdropping has already been eliminated. The qubits that have not been published by Alice or Bob in their public keys continue to be unknown to everybody else. These unpublished qubits form the secret key *secret*, that is, *secret* will be formed from Alice's recorded values, and Bob's complementary values. In our ten qubit example, valid unpublished qubits are qubits 3 and 4. Therefore, the secret key will be Alice's qubits 3 and 4:

$$secret = 10$$

Bob has to complement his qubits to reach the same value as Alice.

The size (length) n of the secret key depends on the initial length of the qubit array l, as well as the fraction of discarded qubits f. Alice and Bob have decided randomly which qubits to publish. In the worst case, the set of qubits published by Alice is disjoint from the set published by Bob. Thus, the fraction of unpublished qubits is

$1-2f$. From these unpublished qubits, only half (50%) are measured correctly. The length of the secret key is given by the formula

$$n = (1-2f)\frac{1}{2}l$$

For our example

$$n = (1-2\frac{40}{100})\frac{1}{2}10 = 1$$

The length of the array of qubits that may form the secret key varies according to which qubits Alice and Bob have chosen to post. In the best case Alice and Bob have chosen the same set of qubits, leaving a fraction $(1-f)$ of secret qubits. In the worst case, Alice and Bob post disjunct sets of qubits, thus leaving a fraction of $(1-2f)$ secret qubits. For our example, the length of the secret key is 1 in the worst case. For Alice's and Bob's particular choices, we may use 2 bits.

3.4.1 Security Evaluation or Catching the Evil Eavesdropper

Let us consider now the algorithm from the point of view of the eavesdropper Eve. Eve wants to ideally gather knowledge about the value of the secret key without being noticed by either Alice or Bob. It is well known that an entangled qubit pair reveals no information whatsoever unless the qubits are measured and the entangled state collapses. Even so, the algorithm presented in this chapter supposes that the entanglement is not protected, only the public keys are protected. This means that the qubits are not guaranteed to be entangled. Eve may masquerade and distribute qubit arrays of her own choice. It is of no advantage to Eve to distribute entangled qubits, as she gains no knowledge about the future secret key from unmeasured entangled qubits. The best choice for Eve is to distribute classical bits, or independent qubits in a known state.

The best Eve can do is to give Alice an array of classical 0s:

$$q_{1A}q_{2A}...q_{lA} = 00...0$$

and to Bob an array of $H1$:

$$q_{1B}q_{2B}...q_{lB} = H1\ H1...H1$$

All other possible arrays Eve could send to Alice and Bob are equivalent or less advantageous than the arrays above. In particular, Eve will want to send any pair (q_{iA}, q_{iB}) that *can* be measured correctly : $(0, H1)$, $(H0, 1)$, $(1, H0)$, or $(H1, 0)$. Any such pair is equally advantageous. For simplicity we will discuss the arrays of 0s and $H1$s, respectively. For a pair $(0, H1)$, Alice and Bob apply randomly one of the four measurement options. The first correct measurement option (q_{iA}, Hq_{iB}) consistently yields complementary correct results, namely $(0, 1)$. The second correct measurement option (Hq_{iA}, q_{iB}) yields all four possible classical bit combinations $(0, 0)$, $(0, 1)$, $(1, 0)$, and $(1, 1)$. Moreover, these combinations are equally likely. In one-half of the cases, measurements will be $(0, 0)$ or $(1, 1)$. This cannot happen, if the qubits are entangled and untouched. This situation reveals the intervention of Eve. Thus, on any qubit checked for eavesdropping, there is a $\frac{1}{4} \times \frac{1}{2} = \frac{1}{8}$ chance of detecting Eve.

As Alice and Bob respectively check a fraction f of the original array, the expected number of times Eve is detected, that is, the *expected detection rate*, is

$$expected\_detection\_rate = 1 - (\frac{7}{8})^{f \times l}$$

For our example, the expected detection rate is

$$expected\_detection\_rate = 1 - (\frac{7}{8})^{\frac{40}{100} \times 10} = 1 - (\frac{7}{8})^4 \approx 0.41 = 41\%$$

Eve is caught 41% of the time. This expected detection rate is rather low given the toy example we have considered, but of course it can be increased arbitrarily by increasing f and/or l.

Suppose we have an array of 1024 qubits and work with the same fraction $f = \frac{40}{100}$. In this case, the length of the final key in the worst case is

$$n = (1 - 2\frac{40}{100})\frac{1}{2}1024 \approx 100$$

This is a length that can be used in practice.

The number of qubits checked by Alice (and also by Bob) is

$$checked\_qubits = \frac{1}{2} \times \frac{40}{100} \times 1024 = 204.8$$

On each qubit, Eve can escape being caught with probability $\frac{3}{4}$. Thus Eve can escape with probability $\frac{3}{4}^{204.8} = 3.25 \times 10^{-26}$. This probability is infinitesimal for any practical purposes.

For a succint presentation of the algorithm see table 3.3.

3.5 Conclusion

The algorithms presented in this chapter show clearly that quantum computation has the means of producing secret information (a secret key) using public information only. This is a major difference compared to existing quantum protocols and also to classical cryptography. In our algorithms a true secret key is developed such that the eavesdropper has no knowledge whatsoever of the value of the key. For Eve, every single binary bit of the final secret key is totally unknown.

In a more general sense, in cryptography, Alice and Bob want to share a secret key to subsequently encode/decode messages. If the key is indeed secret, then messages

Alice's qubits: q_{1A}, q_{2A}, ..., q_{lA}.
Bob's qubits: q_{1B}, q_{2B}, ..., q_{lB}.

Measure the qubits
for every entangled pair (q_{iA}, q_{iB}) do
    Alice randomly measures q_{iA}
        either directly
        or with a Hadamard gate.
    Bob randomly measures q_{iB}
        either directly
        or with a Hadamard gate.

Publish public protected keys
for every entangled pair (q_a, q_b) do
    Alice publishes the measured q_a and the measuring basis.
    Bob publishes the measured q_b and the measuring basis.
Alice randomly chooses a fraction f of qubits.
    for every q_{iA} publish
        the index iA and the measured q_{iA} and
        the measuring basis: simple or Hadamard.
Bob randomly chooses a fraction f of qubits.
    for every q_{iB} publish
        the index iB and the measured q_{iB} and
        the measuring basis: simple or Hadamard.

Check for eavesdropping
Alice looks at each qubit value q_{iB} published by Bob.
    if the measuring bases for q_{iB} and q_{iA} are different
        then check whether the values are opposite.
        if the values of q_{iB} and q_{iA} are opposite
            then continue with next qubit. (No intervention from Eve.)
            else report intruder. (Eve has touched those qubits.)
Bob looks at each qubit value q_{iA} published by Alice.
    if the measuring bases for q_{iA} and q_{iB} are different
        then check whether the values are opposite.
        if the values of q_{iA} and q_{iB} are opposite
            then continue with next qubit. (No intervention from Eve.)
            else report intruder. (Eve has touched those qubits.)

Form the secret key from the remaining qubit measurements.
Bob complements his key.

Table 3.3: Key distribution without classical channel.

can be indeed exchanged secretly. Secrecy of the message is ensured as long as the secret key remains totally secret and unbreakable. If Alice and Bob meet in advance to exchange a secret key this subsequent secret communication is easily achieved. If they want to communicate in secret without a prior meeting, the secrecy is much more difficult to achieve. Classical solutions with good practical results are offered by public key cryptosystems. Alice has both a private and a public key. The public key is used by Bob to encrypt a message that can be decoded only by Alice's secret key. The encryption function is a one-way function, for which it is not feasible to compute the inverse, and hence the secret key. It is accepted though that with enough computational power such an inverse can be obtained. This means that the public key *reveals* some information about the decoding method. The secret key becomes potentially breakable.

In quantum cryptography, to date, key enhancement assures that the secret key obtained through enhancement protocols is unbreakable. Communication between Alice and Bob does not reveal any information about the secret key. But, as stated in the beginning, quantum key enhancement only obtains a longer key from a shorter one.

This chapter presents, for the first time, two algorithms that develop a secret key and overcome both disadvantages of classical cryptography and previous quantum cryptography. The following are the properties of the secret key produced by our algorithm.

1. The secret key is obtained without using a shorter secret key. This is a major improvement over the previous quantum key enhancement protocols.

2. The secret key is effectively unbreakable. This is common to all previous quantum protocols. The public postings of Alice and Bob do not reveal anything about the value of the key. For Eve, any bit of the secret key still has a 50% chance of being 0 or 1.

Specifically, the algorithms presented perform quantum key distribution based on entangled qubit pairs. The entanglement type is not of the generally used Bell states, but an unusual entanglement based on phase incompatibility. The advantage of this type of entanglement is that Alice and Bob perform *different* measurement steps: one is measuring the qubit directly, and the other is measuring after applying a Hadamard gate. Therefore, the measurement is not symmetric. This property, combined with random choice on the measurement steps leaves Eve with no knowledge of how to measure a qubit in advance. How other protocols and algorithms may benefit from asymmetric measurement is an open problem.

The algorithms use an insecure quantum channel. In addition, the first algorithm uses a classical channel, but unlike all previous quantum key distribution algorithms, the classical channel is not authenticated. The second algorithm does not use any classical channel at all. Authentication and security checking are done at the same time, *after* the algorithm, with the help of two public keys. The two algorithms are equivalent in terms of security levels. They differ though in the resources they use, and thus depending on the practical situation, one may be preferred over the other. The second algorithm, because of no classical communication, has to use twice as many entangled qubit pairs. Also, in the case of the second algorithm, the public postings are longer and have a more complex structure.

Specific to quantum key distribution algorithms, is the fact that the value of the

secret key is not known prior to performing the distribution. The key is developed *during* the execution. Likewise, in our algorithms, the value of the secret key is known after the distribution protocol. Interestingly, we have the same behavior for the public keys. They are not known prior to the execution of the algorithm and are developed during the execution. The consequence is that the public keys are session specific, rather than permanent for one person. The public keys are distinct for each application of the quantum key distribution algorithm.

The parallel of our scheme with the classical authentication scheme is simple. In classical authentication, Alice and Bob have

1. An insecure classical channel.
2. One or two standard protected public keys, posted before any communication on the channel. These keys are specific to the cryptographic entity: Alice or Bob.
3. Any secret key is known prior to any communication between Alice and Bob.

In the quantum authentication scheme presented in this paper, Alice and Bob equivalently have the following items:

1. An insecure quantum communication channel and possibly an insecure classical channel.
2. Two quantum generated protected public keys. These keys are posted towards the end of the protocol. These keys are specific for the session and for the cryptographic entity: Alice or Bob. Keys will differ from one session to another.
3. The secret key is developed during the computation and is *known* only at the end of the protocol.

The main new idea of the protocols presented here was to use public keys or public postings to communicate public information. In fact, as seen in the second algorithm, a classical communication channel is not needed at all as the public keys provide all the information needed. This idea of using public keys to communicate binary information in quantum protocols has a more general applicability. In fact *all* quantum key enhancement protocols to date can be reformulated to work with public postings rather than classical communication channels. And this applies to quantum protocols using entanglement as well as protocols without entanglement. This is important, as it shows the general capability of quantum cryptography to generate secret information from public information. Protocols reformulated to use public postings instead of classical channels, would not need the small secret key for authentication and thus would become true quantum key distribution protocols similar to the one presented here.

If entangled qubits are easily available, the secret key established by the protocol can be arbitrarily long. Our algorithm thus allows Alice and Bob to share a one-time pad [47] without prior meeting. To use one time pads, traditionally, Alice and Bob meet in secret and exchange a long list of keys, each as long as the message it is supposed to encrypt, and each to be used exactly once. To share one-time pads has never been thought possible without the explicit meeting of the two parties.

As shown in this chapter, quantum authentication can be done with the help of a variation of protected public keys. This might not be the only solution. It is an open problem what other structures can support authentication of quantum channels.

The principle of checking and authenticating at the end of the protocol with public keys, is not restricted to the algorithm described here. The same type of public keys,

namely per session keys, posted after the execution of the main body of the algorithm, can be successfully used for authentication in other types of algorithms. This is also a direction worth investigating.

Chapter 4

Quantum Sensor Networks

Wireless sensor networks are becoming increasingly more feasible in monitoring or evaluating various data fields. Their domain of applicability is steadily increasing, ranging from civil objective surveillance to strategic surveillance, from environmental forest condition monitoring to urban information gathering. Given the large variety of working environments, the question of protecting the privacy of the gathered data is almost overdue. The problem of the network privacy, its security, refers more generally to the whole process of acquiring the data and querying the network. This problem has been studied sensibly less in comparison to the network's reliability. The purpose of this chapter is to address the problem of security in sensor networks. The solution given here aspires to be comprehensive, in the sense that it addresses a large variety of possible attacks. Our security scheme relies on quantum cryptography and quantum properties. Therefore, the scheme comes with all the advantages of quantum cryptography, such as effectively unbreakable keys. In addition, we solve the problem of identity theft. The identity of a node is protected by quantum means, in a way that is not only impossible classically, but has never been achieved before.

In general, a sensor network is a collection of sensor nodes arbitrarily spread over a geographic field [51]. The purpose of the network is to collect or monitor data from the field. From an abstract point of view, each point of the field is defined by a small set of significant parameters. Each node in its turn is able to measure (sense) the field parameters of its geographical location.

Sensor nodes can communicate with each other via radio signals, which means that they are not hardwired to one another. Each node has a certain transmission power, and it can send messages to any of the nodes within its transmission range. Also, a sensor node can receive messages sent by another node. Note that, the energy consumed to receive a message is independent of the distance between the source and the destination and thus, a node can receive a message from arbitrarily large distances (provided that it falls within the transmission range of the sender). As the nodes are deployed at random across the field, they self organize in a network, restricted only by their transmission range.

Each sensor node has a local limited computational capacity and is therefore able to perform modest sized computations locally.

4.1 Protecting the Sensor Network

The reliability of sensor networks [1] has been studied extensively and refers to the correct functioning of the network in the face of adverse events and failure of some of the nodes. Indeed, sensor nodes function in more challenging and unpredictable circumstances than regular computers and therefore can fail for multiple reasons. For example, sensor nodes are battery operated and battery failure implicitly causes the failure of the node. Again, sensor nodes are deployed in real natural environments,

where natural events may destroy the node. Thus, the network as a whole needs to be operational, even though a fraction of the nodes are not operational. Algorithms to deal with node failure are basic to sensor network management and ensure that sensor networks work reliably.

Note that all the challenges of the network considered up to now are natural, read *unintentional*. In this chapter, by contrast, we explore some aspects of a *malevolent* intervention in the network. The issue of *security* in a sensor network unduly claims modest research effort to date. Security treats the situation where an unwanted intruder purposefully inserts itself in the sensor network in order to impede its correct functioning or simply listens to the environment to gather information that is not rightfully accessible to the intruder.

The cryptographic entities for the sensor network deviate slightly from the common definitions. The cryptographic problem is customarily viewed as two equivalent parties Alice and Bob, that want to communicate secretly. In the sensor network setting described in this chapter, the communication entities are not equivalent. One entity is considerably more vulnerable to attacks than the other.

In cryptography in general, Alice and Bob protect the contents of their messages through encryption and decryption with secret keys. The method used for encryption and decryption as well as how the secret key is established define a specific security scheme. All security measures have the aim to protect the communication content from a third malevolent party Eve.

Eve attempts any possible attack to gather information about the keys and the messages:

1. Eve may listen to the communication channel and read the encrypted messages

transmitted among sensor nodes. This is eavesdropping.

2. Eve may tamper with the content of a message. For example, consider a message, as a string of bits, sent from Bob to Alice. Eve deletes a substring from the message and/or inserts a substring of her own into the message string.

3. Insert false messages in the network. Eve may try to break the keys to be able to both encrypt and decrypt messages any time.

4. Insert itself on a privileged communication line and then drop a message.

5. Eve may masquerade as Bob and send messages to Alice, pretending she were Bob. This is in the realm of identity theft.

The aim of a security system is to provide an environment in which these attacks from Eve cannot succeed, or at least that Eve's actions are detected and exposed. The security system has to provide the means for Alice and Bob to communicate *secretly* and *trustworthily*.

Depending on the particular security setting, the three main characters of the cryptographic play, Alice, Bob, and Eve, personify different entities: humans, computers, some other device, or maybe even an abstract person. The straightforward simple definitions given for Alice, Bob, and Eve are satisfactory for most practical settings. As will be seen in the next sections, the above definitions are positively insufficient for wireless sensor networks.

Our approach to protecting the privacy of the data field and the integrity of the sensor nodes relies on quantum means. We envision sensor nodes that have both a memory of qubits and a set of entangled qubits. Quantum cryptography methods will be used to establish secret keys.

The secret key that is obtained from a quantum key distribution protocol has several desirable and important properties:

1. The secret key is effectively unbreakable [39]. This means that the protocol that establishes the key, does not reveal any information about the value of the key. There is no advantage for an intruder, Eve, to listen to the quantum key distribution protocol. Any particular bit in the secret key still has a 50% chance of being either 0 or 1.

2. Intrusion detection is possible with high probability[39]. If Eve tampers with the messages and the quantum bits during the protocol, her presence is detected.

3. Information exchanged during the protocol is public [32]. There is no need for classical authentication of messages between Alice and Bob. This authentication would typically require a small secret key known to Alice and Bob prior to the protocol, yet the quantum protocol described in [32] provides authentication based on protected public information only.

All these properties are implicitly inherited by our security scheme designed for sensor networks.

Experiments with quantum bits are very impressive [18] [19] [25] [45]. Although mostly in the experimental stage, the age of commercially used quantum devices may be nearer than we expect. Already, practical implementations of the BB84 [5] protocol are commercially available.

Our security scheme has a requirement that is not yet practically feasible. Quantum bits, as used in our protocol, have to be *entangled*. Additionally, our quantum bits have to persist in time. That is, these quantum bits have to retain their state

for a reasonable amount of time and be able to be moved and deployed with the deployment of the sensor nodes. Trapping and transporting entangled quantum bits has not yet been done. Nevertheless, once entangled quantum bits can be stored and transported, applications of the kind described in this chapter become very attractive indeed.

4.2 The Identification versus the Identity of Bob

Bob will be associated with a sensor node. Nevertheless, his identity is not trivial both to define and to protect. The following sections show a clear distinction between the capabilities of classical versus quantum cryptography. The identity of Bob in classical cryptography will inherently remain fuzzy from the point of view of the cryptographic system. Classically, the identity of Bob can be fully defined only *outside* the system. *A judge* who seeks to decide unequivocally whether some entity is Bob or not (maybe Eve) has to step out of the system and make a decision that is not based on cryptographic protocols. On the other hand, the contribution of quantum memories, as described in this security scheme, is to make the full identification of Bob possible from inside the system. With quantum security Eve *cannot* steal Bob's identity without being caught. Our security scheme is the first to achieve full protection from identity theft. Protection from identity theft is unprecedented in cryptography in general. Our scheme totally eliminates the problem of identity theft, which threatens security and safety in many areas of society.

Full and unequivocal identification of Bob is offered through the cryptographic protocol itself. The link between Bob's identity and Bob's identifying signature is *ensured* through the protocol itself. When Eve attempts to steal Bob's signature, she

is implicitly destroying Bob's identity as well. Bob ceases to exist. And Alice can detect the disappearance of Bob. This entire process is possible *only* with quantum cryptography, as classical cryptography does not have any means, even theoretically, to offer this feature. Establishing a link between the signature and the identity of Bob within the cryptographic system is beyond the scope of classical cryptography. However, the identity of Bob can be fully defined and protected by resorting to the laws of quantum mechanics. Thus, endowing Bob with a quantum memory allows a level of protection which has not been attained, nor will ever be possible to attain through classical information processing alone.

Let us elaborate on the aim of a security scheme, which is to provide *secret* and *trustworthy* communication.

1. **Secret Communication**. The communication is secret, if Eve cannot read and understand the message. Suppose Bob sends a message to Alice. The message is encrypted and only Alice has the decryption key. She is able to read the message. As long as Eve cannot get the decryption key or cannot compute and break the decryption key, the message remains secret.

2. **Trustworthy Communication**. When Alice gets a message, she wants to be sure that the contents of the message represents the intention of the sender, and the sender is known. Here, the 'sender' means the person that Alice knows has sent the message, namely Bob. Trustworthiness refers to both the *content of the message* and also very importantly to *the identity of the sender*. This means the content of the message has not been touched by Eve, but also the sender is indeed Bob. We will see that only quantum cryptographic means provide unequivocal identification of Bob.

Suppose now, that Bob sends a message to Alice and signs it with his name

"Bob", so that Alice knows who wrote the message. The message is encrypted by some encryption key and Bob's name is also encrypted.

If the content of the message is able to consistently reach Alice without any changes, this means the message is trustworthy. Eve has not tampered with the content of the message. Eve has not added or deleted parts of the message. As long as Eve does not know the encryption key, she cannot add to the message information encrypted properly.

If Alice can consistently be sure that the messages signed with the name of "Bob" are coming from Bob, this means that the messages are *authenticated*. The signature of Bob must be highly discriminant, meaning that it is very unlikely to be produced by chance. The signature must also be impossible to copy or to falsify. Usually Bob's name is encrypted using Bob's secret encryption key. As long as Eve does not know Bob's encryption key, Eve cannot sign messages with Bob's name. This protection prevents Eve from masquerading, that is, sending messages to Alice while pretending she is Bob.

Another application of Bob's signature is to *certify* a message coming from Bob. The practical application of this is electronic signatures. A document signed with Bob's electronic signature is binding for Bob. Bob cannot legally retract the validity of the document signed by him.

We see that Bob's authenticated signature has a double role: it assures Alice that the message comes from Bob, and it binds Bob to his word. From the point of view of the cryptographic system, Bob's signature identifies Bob. In fact, this is the only way to identify Bob, namely, through his signature. His signature, by convention, is usually the encrypted version of his name. Behind the identifying

signature of Bob stands "Bob himself". Bob's identity and identification are closely linked. Yet, there is a clear distinction between the identification signature and the identity behind the signature. If Eve steals the secret key, she steals Bob's identity. The identity behind the secret key is now Eve. To anticipate our result in sensor networks: classical cryptography is not able to "ensure" the unique correspondence between signature and identity, whereas our quantum security scheme does guarantee this correspondence. Eve cannot steal Bob's identity, she cannot become Bob without being caught.

Classically, the unique identification of Bob by his signature comes from the secrecy of the encryption key. Bob's encryption key is known only to Bob and should be again highly discriminant. The latter means that it is unlikely to find a copy of his encryption key by chance. Therefore, Bob's identification, from the point of view of the cryptographic system, is done by his secret encryption key. Actually, Bob is *cryptographically identical* to his secret key.

From the point of view of the cryptographic system, there is no distinction between Bob and his secret key. They are one and the same cryptographic entity. Note here, the important distinction between Bob as a "legal person", or the "actual person", and Bob as a "cryptographic entity". Bob as a "legal person" is a human and different from his secret key, whereas Bob as a cryptographic entity *is* his secret key. Moreover in our real world, Bob, the "legal person", is directly responsible to protect his secret key. Furthermore, Bob has to perform this protection outside of the cryptographic system. Let us illustrate this concept by a story-example:

Alice and Bob are two real-life people. Their professional task is to communicate secretly on some arbitrary subject. Bob has to authenticate his messages using his personal private key. This key is secret and is known only to Bob. Yet, Bob is only human ... he does not fully trust his own memory. To say nothing of the fact that keys and passwords get longer by the day. Bob has written his private key in his personal notebook. Eve can do two things:

1. *Eve can use her huge computational power and compute/break Bob's private key, or*

2. *Eve can cunningly enter Bob's office when he is not there and copy his private key from his notebook.*

The first attack is an attack **inside** the cryptographic system. If Eve manages to break the key using an inside attack, then the cryptographic system has **failed**. A good cryptographic system should be able to withstand an attack of this kind.

The second attack is **outside** the cryptographic system. If Eve manages to get Bob's private key by this method, it is not the fault of the cryptographic system. The cryptographic system **was not broken**. Yet, the cryptographic system has **failed** in that it did not cover this situation. "Bob's" cryptographic identity is now carried by two legal persons: Bob and Eve, both humans. This responsibility of protecting Bob's key pertained to "Bob" the legal person. The cryptographic system was not intended to offer protection from this second type of attack. By contrast, the cryptographic scheme described in this chapter will prove to be superior to the scheme described in our story in that attacks outside of the system are not possible.

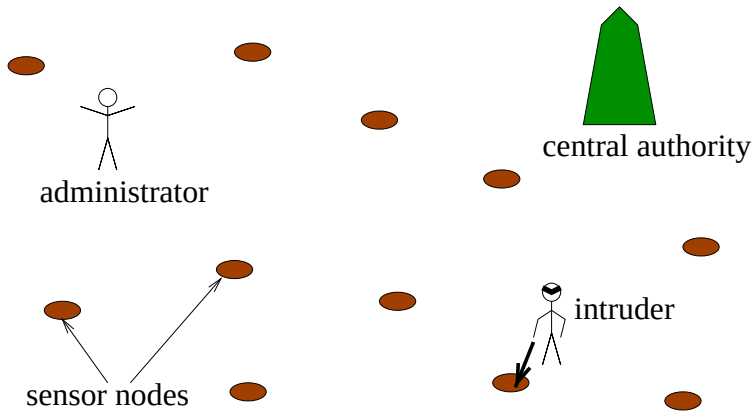

Figure 4.1: A network of sensor nodes with the administrator walking in the field.

In both attacks described above, if the attack is successful, then Eve has achieved her goal: the cryptographic identity of Bob, namely, his secret key, has been stolen. There is no longer an unequivocal link between Bob and his secret key. To repeat, in the scheme described in this chapter, both types of attacks, inside and outside of the system, are impossible. Eve is caught in both cases. Eve cannot break Bob's private key and she cannot attack "outside" of the cryptosystem. There is no identity for Bob outside the cryptographic system.

4.3 The Sensor Network and Its Security

A sensor network performs a monitoring task over a geographic area. For definiteness, consider the sensor nodes to be monitoring a set of environmental parameters. The administrator of the network is a mobile administrator (possibly a person) moving among the sensor nodes in the field (see Fig. 4.1). The administrator should be able to take decisions based on the information gathered from the field. Consider the following toy example.

> *The administrator is a fox hunting rabbits. The sensor nodes are able to detect the presence of a rabbit and also the size of the rabbit. The fox wants to be able to* know *where the rabbits are, without walking through the whole field, indeed it wants to get this information without moving from its present location. Once the fox knows about the position and sizes of the rabbits, it will decide to go catch the largest rabbit. The security question translates for our game to the following scenario. Besides the fox, there is also a large cat walking in the field. Formally, we will call the cat the intruder, or adversary. The cat also wants to catch rabbits. The problem of the entire network is to prevent the cat from gathering any knowledge about the rabbits in the field. The cat is able to listen to the environment and record the messages transmitted among the sensor nodes. The protocol presented below will make the messages unintelligible to the cat. Also the cat should not be able to disguise into a sensor node, or steal a sensor node.*

Thus, the administrator is interested in gathering information about any point in the field directly through the network, without having to move to that point. For example, the administrator wants information about a possible danger in some place, *before* moving there. The collected information will then affect its decision and movement in the field. From the point of view of the network, the administrator will be querying arbitrary sensor nodes. Also, if an unexpected event is sensed in the environment of some node, that node should signal to the administrator the unusual situation.

The environment in which the sensor network operates is considered to be hostile. Wireless communication is not secure. The intruder harbors malevolent plans and may take any of the following actions: listen to the environment to intercept messages; send spurious messages in the environment; capture a node (read its content and reprogram its behavior). Sensor nodes, given their size and working environment, are particularly vulnerable to attacks.

The administrator is associated with Alice. Alice is in a position of authority over the network and implicitly over Bob. This is different from standard settings, where Alice and Bob are equivalent cryptographic entities. Alice decides when communication with a sensor node is going to take place. Also, Alice is the ultimate authority to decide whether the intruder, Eve, has meddled in the communication. The administrator's communication partner is any arbitrary sensor node. Thus, Bob is associated with the sensor node.

To defy the intruder's schemes, the network will be equipped with quantum bits and quantum computation gates. The quantum security scheme will offer effectively unbreakable secret keys, intrusion detection, and protection from identity theft. These three qualities of the quantum scheme are absent in classical existing schemes.

To be able to effectively use the quantum bits, we will require the existence of a base station or central authority (see fig. 4.1). The base station is situated anywhere outside the field. It does not need to be in the communication range of any sensor node. It can be far from the sensor field, and is not directly connected to the sensor nodes. The administrator is able to communicate with the base station on an authenticated telephone line. This telephone line can be made available prior to any interaction between the administrator and the field.

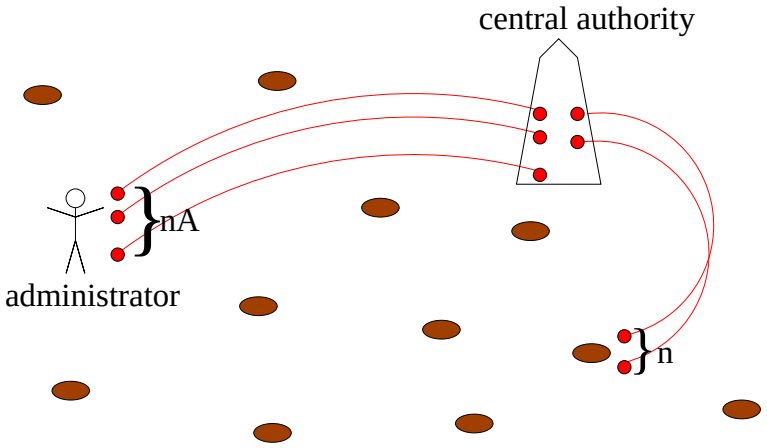

Figure 4.2: The base station possesses qubits entangled with the administrator and the nodes.

The reason for the base station is that it makes the connection between the administrator and the sensor nodes in terms of quantum bits. In short, both the sensor nodes and the administrator are entangled via multiple quantum bits with the base station and the main purpose of the base station is to manage these quantum bits (see fig. 4.2). All entangled qubit pairs are entangled according to the first Bell state entanglement:

$$q_A q_B = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{4.1}$$

This entanglement has been described in detail in chapter 2.

Following a quantum teleportation protocol, described in the next section, the base station will be able to entangle qubits of the administrator with qubits of some chosen sensor node. The result is that the administrator now is directly entangled with a sensor node of its choice and can establish a secure secret key.

It is important now to mention that in this security scheme, several objects are trusted, namely:

1. The base station is trusted. This is a reasonable assumption, as the base station is not part of the field and can be located in a secure place.

2. The administrator is trusted. The administrator is the basic decision making component and thus is given authority and trust.

3. The sensor nodes are not trusted. They are vulnerable to reading, writing and their identity may be attempted to be stolen.

4. The environment is not trusted. Messages among sensor nodes can be freely intercepted. Also the telephone line between the administrator and the base station is not secure, though authenticated. The adversary can listen to the telephone conversations.

The quantum characteristics of the network will affect all entities of the security scheme: the administrator, the sensor nodes and the base station.

4.4 Quantum Characteristics of the Sensor Network

All the quantum properties of the network will be used to provide security/secrecy of the monitoring and querying. Qubits are not inherently necessary to the monitoring task, classical bits would work just as well.

Every **node** has a local memory made of qubits (see Fig. 4.3). It is a quantum memory. In fact, the node does not use *any* classical bits. It is well known, that qubits subsume classical bits, meaning that everything that classical bits can compute, qubits can compute identically [8].

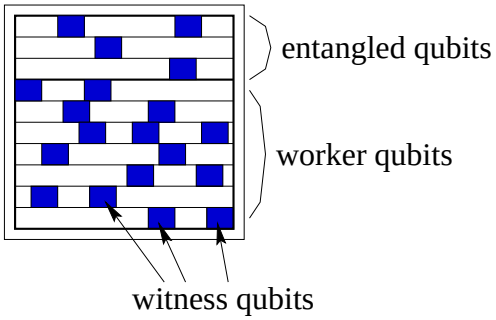

Figure 4.3: The structure of a node's memory. The whole memory consists of qubits.

Thus nodes can work in the expected classical way, having the additional advantage of quantum based security protocols.

Each node has qubits allocated to three different tasks:

1. **Entangled qubits** are used for establishing secret keys. The key generation qubits, $qKey_1$, $qKey_2$, ..., $qKey_n$, are destined to help the key distribution protocol. As a result of the protocol, the administrator and the node will share a secret key. Qubits of this kind are entangled pairwise with qubits from the base station.

2. **Witness qubits** signal the presence of an intruder. The witness qubits, $qWit_1$, $qWit_2$, ..., $qWit_m$, are spread out over the whole memory of the node. Their role is to show the existence of an intruder. They can be hidden as part of the node's program or may be dedicated witnesses. The state of the witness qubits is different depending on whether an intruder has touched the node through reading or the node has remained untouched from the outset. Just reading of the qubits is enough to change their state, the intruder does not need to attempt writing the memory of the node. Reading a witness qubit leaves an

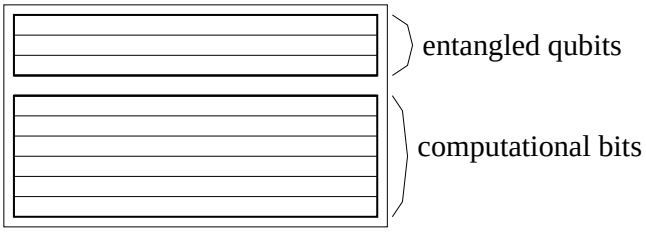

Figure 4.4: The structure of the administrator's memory. The two parts are structurally different: the entangled qubits are a quantum memory whereas the computational bits form an ordinary binary memory.

unmistakable mark of the act on the state of the qubit.

The witness qubits are spread out over the memory in a random way, such that an intruder will have to read witness qubits with high probability, even if the intruder reads only a fraction of the memory.

3. **Worker qubits** perform the regular job of the node. The worker qubits, $qWork_1$, $qWork_2$, ..., $qWork_p$, contain programs to be used by the node and data collected from the environment. Transit packets are also stored in this part of the memory.

An arbitrary qubit of the memory can belong to exactly one of the three categories. Every qubit maintains its category from the deployment of the sensor node, throughout the node's lifetime.

The **administrator** does not need protection from the intruder. Therefore, most of its memory may, for simplicity, consist of classical bits (see Fig. 4.4). Nevertheless, qubits are necessary to establish a secret key with a node. Thus, the administrator's memory consists of two types of bits/qubits.

1. **Entangled qubits** will provide secret keys for all communications with nodes of the network. The qubits $qAKey_1$, $qAKey_2$, ..., $qAKey_{nA}$, are entangled

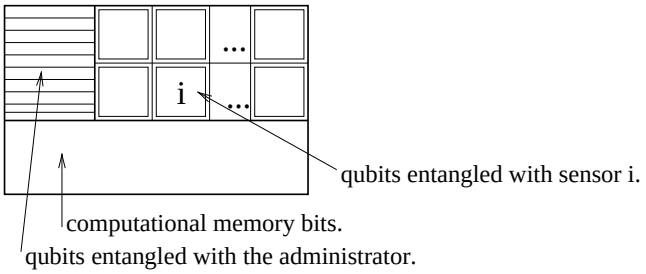

Figure 4.5: The structure of base station's memory.

pairwise with qubits from the base station. The number of entangled qubits nA that the administrator has is considerably larger than the number of entangled qubits n of any sensor node, $nA >> n$.

2. **Worker bits** form a regular memory. The administrator uses this memory for regular computing and storage of data. This memory is of considerable size and can be viewed as the memory of a computer.

The two types of memory are used for their specific tasks only. This means that entangled qubits, for example, will not be used for regular computations.

The **base station** has the largest memory. From the point of view of the security scheme, the base station possesses partners of *all* entangled qubits from the field nodes and from the administrator (see Fig. 4.5). The set corresponding to the administrator is $qAKey'_1$, $qAKey'_2$, ..., $qAKey'_{nA}$. For each node, the base station has a set of corresponding qubits $qKey'_1$, $qKey'_2$, ..., $qKey'_n$. The qubits are entangled in the expected way:

$$qAKey_1 \; with \; qAKey'_1; \quad qAKey_2 \; with \; qAKey'_2; \quad ...; \\ qAKey_{nA} \; with \; qAKey'_{nA}. \tag{4.2}$$

and

$$qKey_1 \; with \; qKey'_1; \quad qKey_2 \; with \; qKey'_2; \quad ...; \quad qKey_n \; with \; qKey'_n. \tag{4.3}$$

4.5 Entanglement Swapping in the Sensor Network

Quantum teleportation was defined in [7], [46]. It refers to the transfer of an *unknown* quantum state from one geographical source location to another destination location. This state transfer does not involve any transfer of matter from the source to the destination. It needs an entangled qubit pair, with the first qubit located at the source and the second qubit located at the destination. The second qubit will receive the desired unknown state. In transferring the state to the destination, it disappears from the source, in agreement with the "no cloning" theorem [49].

To obtain the desired teleported state at the destination, two bits of classical information need to be sent from the source to the destination. Depending on this information, the destination qubit needs to be transformed by a simple gate. This property complies with the principle that information cannot be transmitted at a speed greater than the speed of light.

A variant of quantum teleportation is entanglement swapping. Note that, in teleportation, the quantum state of the source qubit q_{source} disappears from the source location and reappears in the destination qubit $q_{destination}$ as exactly the same state. If the original state q_{source} was entangled with some other qubit q_{other}, this entanglement will be transferred to the destination qubit $q_{destination}$, causing the latter to be entangled with q_{other}. This scenario is called entanglement swapping and has been demonstrated in practice [24].

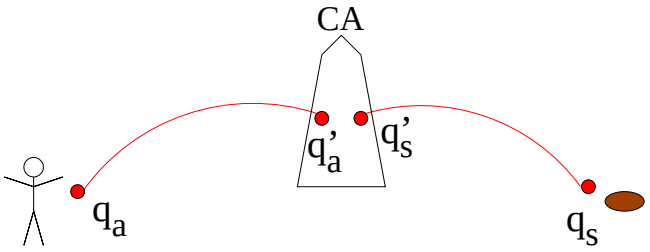

Figure 4.6: Entangled qubit pairs before swapping.

Entanglement swapping is the basic step towards private communication between the administrator and some sensor node.

Consider some qubit of the administrator q_{ai} entangled with its base station companion qubit q'_{ai}. The administrator intends to communicate secretly with node s. The node's qubit offered for this entanglement swapping may be q_{sj} entangled with the base station's qubit q'_{sj} (see Fig. 4.6). These four qubits form a pair

$$pair = q_{ai}q'_{ai}q'_{sj}q_{sj}. \tag{4.4}$$

Note that the first qubit of the pair belongs to the administrator. The second and third qubits belong to the base station and the fourth qubit belongs to the sensor node. This order has been chosen so that the transformations applied by the base station are easier to see. As both the administrator's qubit pair (q_{ai}, q'_{ai}) and the sensor node's qubit pair (q_{sj}, q'_{sj}) are entangled in the Φ^+ Bell state, the pair can be rewritten as

$$\begin{aligned} pair &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \otimes \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\ &= \frac{1}{2}(|0000\rangle + |0011\rangle + |1100\rangle + |1111\rangle). \end{aligned} \tag{4.5}$$

The following formula rewrites the base station's two qubits (namely, q'_{ai} and q'_{sj})

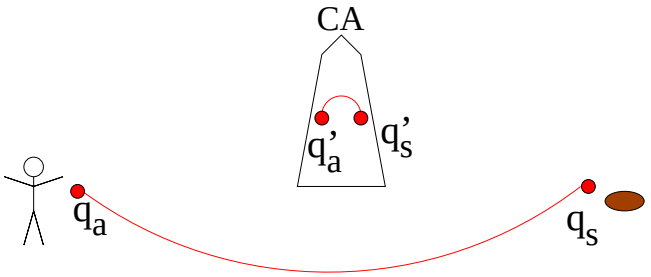

Figure 4.7: Entangled qubit pairs after swapping.

highlighting the Bell basis

$$
\begin{aligned}
pair &= \frac{1}{2}(|0\rangle \otimes \frac{1}{\sqrt{2}}(|\Phi^+\rangle + |\Phi^-\rangle) \otimes |0\rangle + \\
&\quad |0\rangle \otimes \frac{1}{\sqrt{2}}(|\Psi^+\rangle + |\Psi^-\rangle) \otimes |1\rangle + \\
&\quad |1\rangle \otimes \frac{1}{\sqrt{2}}(|\Psi^+\rangle - |\Psi^-\rangle) \otimes |0\rangle + \\
&\quad |1\rangle \otimes \frac{1}{\sqrt{2}}(|\Phi^+\rangle - |\Phi^-\rangle) \otimes |1\rangle) \\
&= \frac{1}{2\sqrt{2}}(|0\rangle \otimes |\Phi^+\rangle \otimes |0\rangle + |1\rangle \otimes |\Phi^+\rangle \otimes |1\rangle + \\
&\quad |0\rangle \otimes |\Phi^-\rangle \otimes |0\rangle - |1\rangle \otimes |\Phi^-\rangle \otimes |1\rangle + \\
&\quad |0\rangle \otimes |\Psi^+\rangle \otimes |1\rangle + |1\rangle \otimes |\Psi^+\rangle \otimes |0\rangle + \\
&\quad |0\rangle \otimes |\Psi^-\rangle \otimes |1\rangle - |1\rangle \otimes |\Psi^-\rangle \otimes |0\rangle). \qquad (4.6)
\end{aligned}
$$

The base station now measures qubits two and three (namely, q'_{ai} and q'_{sj}, located at the station) in the Bell basis (Φ^+, Φ^-, Ψ^+, Ψ^-).

It is interesting to see what happens to the state of the other two qubits after this measurement (see Fig. 4.7). The base station will have to communicate the result of the measurement to the administrator. This is done via the insecure classical channel. If the station's measurement was:

1. Φ^+. The remaining qubits have collapsed to

$$pair_{1,4} = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{4.7}$$

This is a Bell Φ^+ entanglement. The administrator and the field node are now entangled. The administrator knows that the measured value of its qubit q_{ai} will coincide with the measured value of the node's qubit q_{sj}.

2. Φ^-. The remaining qubits have collapsed to

$$pair_{1,4} = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle). \tag{4.8}$$

This is not quite a Φ^+ entanglement, as the phase is rotated. Still, the values measured for the qubits coincide, and that is sufficient to have a consensus on the measured values of the two qubits.

3. Ψ^+. The remaining qubits have collapsed to

$$pair_{1,4} = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle). \tag{4.9}$$

In this case, the administrator has a qubit in which the bit values ($|0\rangle$ and $|1\rangle$) compared to the field node are reversed. After measuring its qubit, the administrator has to take the complement of the resulting bit.

4. Ψ^-. The remaining qubits have collapsed to

$$pair_{1,4} = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle). \tag{4.10}$$

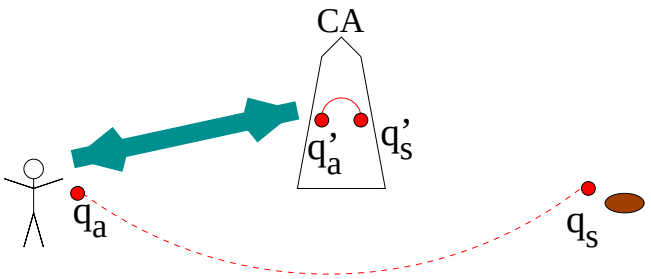

Figure 4.8: Classical information between the base station and administrator has to be exchanged on the telephone line.

> Now the administrator's qubit has both the bit values reversed and the phase is also rotated. After measuring its qubit, the administrator has to take the complement of the resulting bit.

The administrator has to communicate with the base station by telephone line (see Fig. 4.8) in order to know the value measured by the base station: Φ^+, Φ^-, Ψ^+, or Ψ^-. If the base station has measured Φ^+ or Φ^- then the administrator simply measures its qubit and has the same binary value as the node. If the base station has measured Ψ^+ or Ψ^- then the administrator has to measure its qubit and then complement the resulting binary value in order to obtain the value measured by the node. Thus, there are two possible options for the administrator. The base station has to send just one bit of information to discriminate between the two options.

Note that if the administrator wants to have a known entanglement with the field qubit, the administrator will have to discriminate among all four of the base station's measurement outcomes. In this case two bits of information would need to be sent by the base station to the administrator.

After the communication step, the administrator and the field node will be able to have a consensus on the value of a bit without having ever met.

4.6 A New Problem in Cryptography: Who Is Bob?

You look like an angel, walk like an angel, talk like an angel

But ... you're the devil in disguise

Elvis Presley, Devil in Disguise

The cryptographic problem as defined above has some inherent assumptions. They are rather obvious so that often they are not even stated: *Alice and Bob are trusted.* All cryptographic schemes aim to safely *transfer* a message from Alice to Bob or vice-versa. Alice and Bob, the end-points of the message transfer, are supposed to exist from the beginning to the end and their intention is consistently the same and trustworthy: to communicate with each other truthfully and secretly. It is against "common cryptographic sense" to consider that Bob *changes his mind* and becomes more like evil Eve. A more palatable way to describe this scenario is that Bob dies and out of "his ashes" appears a new communication partner with the personality of Eve. As outlandish as these descriptions may seem, secure wireless sensor networks *have* to deal with scenarios of exactly this kind. The intruder can physically capture a node, then read it and change its contents. The node is thus reprogrammed to work according to the intruder's intention. This is equivalent to Bob's death and then Eve's appearance in exactly the same place. The cryptographic problem here becomes for Alice to be able to detect the change of persons from Bob to Eve.

This problem is totally new and is impossible to even address classically except with certain assumptions:

1. It takes time to reprogram a node [36].

2. Attacks follow certain geographical distributions: a node close to a corrupted node is likely to be corrupted in the future [14].

3. Eve misbehaves and does not stay hidden. For example, she floods the network with spurious virus messages.

But in principle, classical cryptography has absolutely no means to *sign* one entity with the name of Bob such that Eve cannot fully copy the signature without any difference. This is also the contribution of using quantum computation.

Note here that in the original cryptographic setting, Eve may attempt to masquerade. She pretends to be Bob and sends a message to Alice signed Bob. There is a similarity between masquerading and Bob's replacement with Eve. In both cases, *full* messages come from the wrong person. Nevertheless, the similarity stops here. The difference remains fundamental. When Eve masquerades as Bob, Alice receives messages from the wrong source. Alice's job is to discern the authenticity of the message. Alice's verdict refers to the message only, and every message has to be analyzed for itself. Communication with Bob may still be attempted. In the case of wireless sensor networks, when a node is physically captured, Bob no longer exists. Eve sends messages using all the physical resources of Bob. Communication with Bob is no longer possible. Alice can no longer perform a handshake with Bob. The problem of Alice is now to clearly establish the *identity* of her communication partner: *Is Bob still the person with the intended characteristics?*

4.6.1 Previous Solutions to Node Capturing

As previously defined, a node is captured when Eve physically seizes the node and takes full control of its functioning. Such a node is then called *compromised*. Node

capturing means that Eve first reads the contents of the node and then reprograms the node to a behavior in accordance with her own plans.

In the literature, there are several approaches to dealing with this situation.

From the early days of algorithms that deal with network reliability, various designs exist to avoid using a faulty node. The idea is to locate a node that is behaving erratically or is non-responsive and then logically eliminate the node from the network. This also implies rerouting paths for packets that were previously using the node. As such, any rerouting algorithms may be effective in the task of eliminating a node that has been malevolently captured by Eve and thus rendered unusable for the network.

How to determine that a node is compromised comes down to judging its behavior. This doesn't seem to be directly a problem of cryptography and depends heavily on the particular application. If a node behaves erratically or is not trustworthy, it is concluded that the node is compromised. Let us discuss now the characteristics of Eve. If Eve behaves exactly like Bob, the node will behave correctly and will not be eliminated from the network. If Eve behaves largely differently from Bob, it will be easy for Alice to detect the compromised node. Here, Eve has the option to remain undetected as long as she behaves like Bob. In fact, classically, Eve will be indistinguishable from Bob. Of course, Eve prefers to remain undetected and she has good chances to remain so, while behaving even only approximately like Bob, or behaving like Bob most of the time. After all, a good lie is one that contains much truth in it. Eve may surely try to outwit Alice and lie cleverly and only sometimes. But then again, Alice will try to outwit Eve and design traps to catch her even if Eve is lying sparingly or intelligently. Alice and Eve thus start a race over who is able to

outwit the other. Basically, Alice's scheme is successful if she is able to outwit Eve, or in more technical terms, if Alice's computational power is sufficiently superior to that of Eve. If Eve has unbounded computational power, she is not likely to be caught. The success of Alice's scheme relies on a weakness of Eve. By contrast, our quantum solution does not rely on a weakness of Eve. Eve is caught in spite of unbounded computational power. Eve's detection is made possible because of the inherent nature of quantum bits.

The quantum cryptographic solution given in [13] describes a scheme that ensures the identification of a cryptographic entity, such as Bob, to within a given (non-zero) probabilistic error. The scheme presented here has the same property. In addition, in our scheme the cryptographic entity is identical to the factual entity, the communication partner. That is, Bob's identity is the same as a communicating agent, considered outside the cryptosystem, and as a cryptographic entity, as defined within the cryptographic system.

4.6.2 Bob's Quantum Signature

We are ready to describe how the identity of Bob is uniquely protected in our security scheme. Bob is associated with a sensor node. The administrator wants to be able to detect whether an intruder has "touched" the sensor node. The action of the intruder reading the memory of a node will leave an unmistakable mark on the node. The unavoidable mark or change of the node will then be detectable by the administrator, Alice. Bob's witness qubits, as defined in section 4.4, serve this purpose.

Suppose a witness qubit $qWit_i$ is in the state $qWit_i = H(|0\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.

When Eve reads this qubit, $qWit_i$ will collapse to either $|0\rangle$ or $|1\rangle$ with equal probability. After Eve has read $qWit_i$, the state of the qubit is changed. Alice can check the state of $qWit_i$. Measuring $qWit_i$ directly will not offer any information, as Alice would just measure the same value as Eve. So Alice first applies a Hadamard gate to $qWit_i$.

If Eve has not touched the qubit then

$$H\, qWit_i = H\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |0\rangle. \tag{4.11}$$

If Eve has previously read $qWit_i$ then one of the following two possibilities will happen:

$$H\, qWit_i = H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle). \tag{4.12}$$

$$H\, qWit_i = H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \tag{4.13}$$

If Eve has not read $qWit_i$ then Alice, after applying the Hadamard gate, will consistently measure a $|0\rangle$. Otherwise, Alice will measure a $|0\rangle$ or $|1\rangle$ with equal probability. Thus, Alice catches Eve with 50 % probability.

A similar scenario happens for a witness qubit $qWit_j$ in the state $qWit_j = H(|1\rangle) = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. In this case, Alice would expect to read $H\, qWit_j = H\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |1\rangle$, which is consistently a binary 1. If Eve has read the qubit before, then Alice will read again a $|0\rangle$ or $|1\rangle$ with equal probability.

The witness qubits of each node are set to $H|0\rangle$, $H|1\rangle$, $|0\rangle$, or $|1\rangle$ before the node's deployment. The plain $|0\rangle$ and $|1\rangle$ are added so that Eve cannot consistently apply a Hadamard gate to obtain correct results. Each node's witness qubits are

Node Id	Location	Signature	Reading Mask
...			
i	(x, y)	0010	$H**H$
...			

Table 4.1: The administrator's table with information about the field nodes.

set to a different sequence of $H|0\rangle$, $H|1\rangle$, $|0\rangle$, and $|1\rangle$. The particular sequence for some node k is unique and determines the node. Suppose the witness qubits are set to $qWit_1\ qWit_2\ qWit_3\ qWit_4... = H|0\rangle\ |0\rangle\ |1\rangle\ H|0\rangle...$. This sequence is the node's signature. Interesting is that the node, although carrying its signature, does not know its value, nor does the node need to know where its witness qubits are in the memory. It is enough if the internal programs of the node do not touch any witness qubits.

The signature of every node is known before deployment and is communicated to the administrator. In order to keep track of the nodes in the network with their geographic locations, the administrator keeps a table with four columns (see Table 4.1). Each line describes exactly one node. The first column represents the identifier of the node. The identifier does not give any information about the node's location. The second column gives the location of the node, such as its coordinates x and y. The third column is a binary number, the node's signature after reading. The fourth column describes the reading strategy of the quantum signature. It contains an H on the positions of qubits in superposition and $*$ otherwise.

When the administrator wants to check the node for intrusion, it has to measure some witness qubits. Depending on the capabilities of the sensor network, the administrator might need to move geographically beside the node to obtain the quantum

state of the witness qubits to be measured. If the network allows the transfer of quantum states, for example optically, then the node just needs to send the witness qubits to the administrator via the network. The administrator will be able to determine the intrusion, but the qubits used for checking will be destroyed. Witness qubits, as any qubits, cannot be cloned. This means, there exists exactly one instance of the node's signature. The signature cannot be copied. When the administrator checks the qubits, this is done through measurement. The qubits collapse to a classical state. Thus, the initial quantum signature is destroyed in the checking process. This may be a disadvantage. On the other hand, it may also be seen as the signature of a node being a one-time pad, used exactly once. Nevertheless, the usage of a quantum signature may be extended. First, the administrator may check only part of the signature, thus allowing for several checkings. Secondly, the administrator may be allowed to write the signature back to the node. The administrator, after reading the signature may reverse the quantum circuit and recreate the original quantum signature. Then, if the administrator is geographically at the node's location, the signature gets written back into the node. If the administrator communicates with the node via a quantum channel, the signature is sent back to the node, possibly in an encrypted form.

Whenever Eve reads the memory of a node, she destroys the node's signature. Eve does not know the position of the witness qubits, nor does she know the signature. Note that, it is only Alice, the administrator, that knows this information. Not even Bob, that is, the node, is informed about the position and value of the witness qubits. This is a clear advantage, as Bob is most vulnerable to the intruder.

With this setting, the identity of Bob is very easy to define. It is the node's quantum signature. Whoever carries this particular quantum signature *is* Bob. From

the point of view of the security scheme Bob's quantum signature is his identity. The really beautiful part of this definition is that even *outside* of the security scheme, Bob can be identified as his quantum signature. When Eve corrupts a node, she destroys the quantum signature, thus destroying the identity of Bob. Bob no longer exists, the node has a new identity, namely, the identity of Eve. Note that, even if Eve is smart and tries to behave like Bob, her presence is detected from the signature, that is, from the identity change. Eve cannot hide by behaving like Bob.

Also, because of the no-cloning theorem, Eve cannot make a copy of Bob's signature. If Eve could make a copy of Bob's signature, she would gather information about it from the copy, while preserving the original. This cannot be done with quantum bits.

4.7 Secret Key Distribution

The protocol that distributes the secret key between the administrator (Alice) and a sensor node (Bob) follows the steps below. Consider the node to be numbered i and positioned at (x, y).

1. Communication Request. The communication request is initiated by the administrator that has the intention to communicate with some sensor node i. The administrator informs the base station via the telephone line about its intention together with the node's number. The base station knows now that the administrator will communicate with node i.

The telephone line is authenticated, but public. The intruder is able to learn that the node i will be queried. Nevertheless, the intruder does not know who this node is. The number i does not reveal the position of the node (x, y).

2. Entanglement Swapping. The base station performs an entanglement swapping between a set of k qubits of the administrator and of the node. This has been described in section 4.5. The base station combines pairwise the administrator's qubits with the node's qubits and measures each pair in the Bell basis. The result of the measurement is sent to the administrator via the telephone line.

3. Entanglement Verification. The administrator verifies the entanglement directly with the node. This is the standard procedure described in chapter 2. Communication with the node is done via the sensor network. The verified entangled qubits are discarded.

4. Secret Key Measurement. Both the administrator and the node measure the remaining entangled qubits. This is **the secret key**.

5. Identity Checking of Bob. This is optional. Once the key is established, the administrator checks the node's identity to protect the communication from a masquerader. As described in section 4.6.2, the administrator checks the node's quantum signature. This checking may be done also at the end of all communication. This validates the entire communication in terms of the genuineness of the node.

4.8 Query Protocols

The following two query scenarios follow simply from the structure of the sensor network:

1. **Administrator query** (see Fig. 4.9). The administrator has a map of the field and wishes to obtain information from a selected location (x, y) regarding a possible event e. The location (x, y) to be queried will be visible by the

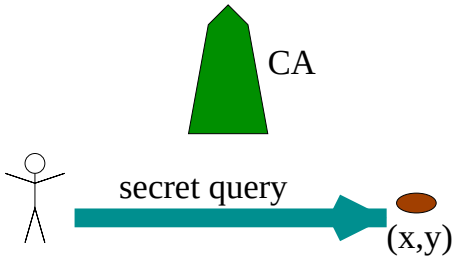

Figure 4.9: The administrator queries the field.

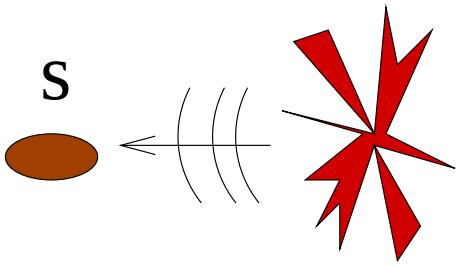

Figure 4.10: A sensor node signals an event.

intruder. Yet, the nature of the event and the parameters of the event will be private.

2. **Sensor node event signaling** (see Fig. 4.10). A sensor node located at (x, y) detects an event of importance. It sends a signal to the administrator. The administrator then queries the node as to the nature and parameters of the event. Again, the intruder will know the location of the event but will not have any information about the nature of the event and its parameters.

We are ready now to describe the steps that allow the administrator to query the field in some specific location:

1. The administrator a sends the location (x, y) of the query to the base station.

2. The base station locates a sensor node s that is closest to the location (x, y) and performs an entanglement swapping for the qubit pairs necessary for the

secret key.

3. The administrator and the node s establish a secret key k.

4. The administrator uses this secret key to encrypt a message containing the nature of the event of interest. Then it broadcasts the message in the network. The message will be unintelligible to all nodes except s which shared the secret key k.

5. When s receives the encrypted message, it reads the parameters of the requested event. These parameters are then encrypted using the same key k. The new message is broadcasted in the field again and the administrator eventually receives the desired information.

6. The administrator performs an identity check of the node. If the node has proven its identity, it means that all the previous communication with the node was genuine.

In the second scenario, in which the sensor node is signaling the event, the procedure is very similar to the previous one. One step is performed ahead of the previous algorithm.

1. The sensor node that has detected an event broadcasts its location on the network. The administrator will read this message with the position of the sensor node and start a query procedure with this location.

The important feature of both algorithms is that the wireless environment does not reveal the measured parameters, nor the nature of the event. The only information which is not encrypted in the network is the location of the event or query.

4.9 Conclusion

Our security scheme for sensor networks using quantum means connects two domains thus far considered unrelated, namely, quantum computing and wireless sensor networks. By adding quantum memories to each sensor node, the security of the network reaches levels never before attained in any other general purpose cryptographic scheme. Our scheme completely solves the problem of identity theft. Identity theft has never been solved satisfactorily in any cryptographic setting, be it general purpose or for wireless sensor networks. This stands in sharp contrast to the need for identity protection, as today's weak identity management poses a real threat to many security sensitive transactions in our society. Our result clearly shows that quantum memories is one possible answer to the issue of identity theft.

Normally, Alice and Bob are cryptographic entities with equal power. Classical cryptographic schemes are usually symmetric. In sensor networks with an administrator, this is not the case. Bob, the sensor node, is a weak entity. Bob is more susceptible to attacks from Eve. Every imaginable attack is easily effected on Bob, including the pervasive identity theft. Alice, the administrator of the network, is a powerful computational entity; the administrator has authority over the sensor node. Cryptographically, Alice "knows" Bob's identification code, his quantum signature. She is the only one to know Bob's identification code. Not even Bob knows his own quantum signature. This is a new paradigm: Alice trusts Bob, but trusts him "weakly". Bob is very susceptible to being corrupted and destroyed.

The cryptographic scheme described in this chapter has some unique features. Bob's quantum signature always refers to the intended cryptographic identity, that is the functional sensor node. Eve cannot steal Bob's signature and masquerade as

Bob. This property results directly from quantum features. If Eve reads the quantum signature, the superposition collapses and the signature is destroyed. Thus, together with the signature, Bob's identity is destroyed also. The destruction of the signature is easily detectable by Alice. Therefore, Bob's quantum signature and his identity are totally protected, due to the weird properties of quantum laws. This level of identity protection has never been achieved previously. By representing and manipulating information at the quantum level, we are able to effectively solve the identity theft problem, a feat impossible to achieve through only classical means, regardless of how much computational power we have at our disposal.

In addition, our quantum scheme inherits all advantages of quantum cryptography, in particular the secret keys are effectively unbreakable. Any messages traveling in the network during the secret key distribution protocol reveal nothing about the value of the secret key. For the eavesdropper, any bit of the secret key still has a 50% chance of being either 0 or 1. Also, any secret key is used exactly once and is afterwards discarded. It is a one time pad.

Thus, the system designed in this chapter protects the network from a huge variety of attacks. The eavesdropper does not profit from listening to the environment. Eve cannot corrupt a sensor node without being caught. Just reading the contents of a sensor node leaves an unmistakable mark on the node, which can be detected by the administrator. Eve cannot masquerade, meaning she cannot simply enter the network as a new node. The security level of our scheme does not depend on Eve's computational power, nor on Eve's smart behavior. Even if Eve has an unbounded computational power, the system is equally safe.

The scheme exploits directly the specific properties of quantum mechanics. We

believe that this is a beginning, and that quantum properties have much more to say in terms of cryptographic protocols. Here, secret messages are still encrypted with a classical binary secret key, albeit quantum generated. Whether a secret communication can be conceived to be fully quantum, with quantum keys and quantum messages, requires an unconventional approach to cryptography in general, still waiting to be initiated.

Some research on closed timelike curves [10] [11] [12] [16] claims that by allowing qubits to travel in time, even quantum protocols can be easily broken. As this research is not supported by current experiments, it does not seem to be an immediate danger to the security of quantum protocols in general.

Chapter 5

One-Time Pads Without Prior Communication

Suppose that similarly to all previous chapters, Alice and Bob want to ensure the secrecy of the messages they exchange. If Alice and Bob *share* a set of long secret keys, the secrecy of their communication is guaranteed. They use each key for exactly one communication and the key is discarded afterwards. Needless to say, the secret keys are independent, randomly generated and thus, one key does not reveal any information about any other key. In the cryptographic world, these keys are usually referred to as one-time pads [42]. Any encryption/decryption function will be good enough to ensure secrecy, such as for example, a binary XOR of the message text with the one-time pad (assuming the text and key are binary strings). The only condition on the key is to be at least as long as the message itself, so that there are no repetitions of the key's usage.

The obvious drawback of any scheme with one-time pads is that Alice and Bob need a prior reliable agreement on the value of the secret keys. In practice, to date,

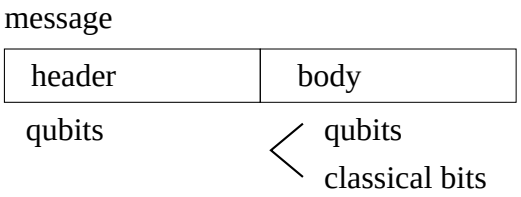

Figure 5.1: A message consists of two concatenated parts. The header is an array of qubits. The body may be an array of classical bits or an array of qubits as well.

the only viable solution to reaching a consensus and keeping the secrecy of the keys is for Alice and Bob to meet in advance. They need to have a secure, private meeting in which they agree on the value of *all* secret keys to be used henceforth. If, after communicating for some time, they run out of keys, Alice and Bob need another secret meeting. The basic idea is that for *any* one-time pad that Alice and Bob use, there existed a prior secret meeting of Alice and Bob and in this meeting the value of the one-time pad was defined.

By endowing messages with quantum properties, we show that encryption and decryption can be done with one-time pads and the value of the one-time pads are generated without Alice and Bob having to meet. Thus, messages are quantum messages or at least partially quantum. A message (see Fig. 5.1) consists of two concatenated arrays. The message header is the first part and is an array of qubits. The header renders the value of the one-time pad. The message body may be an array of classical bits or again an array of qubits. We will discuss both options. The body contains the information to be transmitted in an encrypted form.

5.1 Reading Masks

As mentioned above, the value of the one-time pad is carried by the array of qubits in the header. Some of these qubits are in a basic state, either 0 or 1, while the rest of the qubits are in a balanced superposition of 0 and 1. We say that the (binary) value of the one-time pad is quantum encrypted. For each classical bit b_i of the pad, there exist four possible encryption options, resulting into the qubit q_i. These options are:

1. $q_i = b_i$. The qubit carries the exact value of the classical bit.
   - if $b_i = 0$ then $q_i = 0$.
   - if $b_i = 1$ then $q_i = 1$.
2. $q_i = NOT\ b_i$. The qubit carries the value of the complement of the classical bit.
   - if $b_i = 0$ then $q_i = 1$.
   - if $b_i = 1$ then $q_i = 0$.
3. $q_i = Hb_i$. The qubit is a superposition, obtained by applying the Hadamard gate on b_i.
   - if $b_i = 0$ then $q_i = H0 = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.
   - if $b_i = 1$ then $q_i = H1 = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.
4. $q_i = H\ NOT\ b_i$. The qubit is a superposition obtained by applying the Hadamard gate to the complement of b_i.
   - if $b_i = 0$ then $q_i = H\ NOT\ 0 = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.

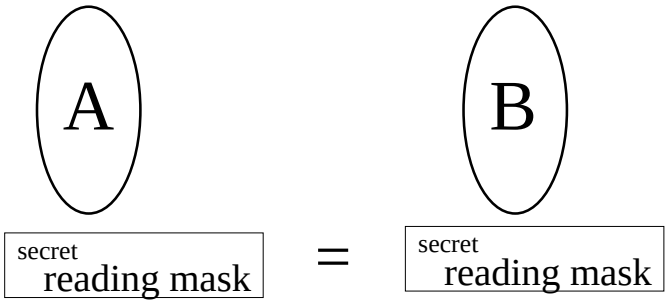

Figure 5.2: Alice and Bob share a secret reading mask.

- if $b_i = 1$ then $q_i = H\ \ NOT\ \ 1 = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.

An array that explains how to encrypt each b_i, or conversely, explains how to read each q_i is called an *encryption/decryption mask* or simply a *reading mask*. Let us see how the reading mask works on an example.

Suppose the secret key is

$$secret = 00111001$$

It is eight bits long. The reading mask has to have the same length. The value of the reading mask is independent of the value of the secret key. Consider the mask to be

$$mask = *\ \ H\ \ *\ \ (H\ \ NOT)\ \ NOT\ \ H\ \ H\ \ *,$$

where $*$ means read directly and the other notations are self explanatory.

To obtain the quantum encrypted version of the secret key, each bit needs to be transformed as defined by that position in the mask. For our particular example:

$$secret\_encrypt = 0\ \ H0\ \ 1\ \ H0\ \ 0\ \ H0\ \ H0\ \ 1$$

5.2 One-Time Pad Communication with Classical Message

It is clear by now that Alice and Bob can exchange messages containing quantum encrypted keys provided that they share the reading mask. We consider that Alice and Bob do have a "meeting point" before they start an indefinite exchange of messages. This means that they meet and agree on a reading mask (see Fig. 5.2), or they may develop a secret reading mask using any quantum key distribution algorithm such as the ones presented in chapter 3. Another way of viewing the sharing of the reading mask is to consider Alice and Bob as two devices, rather than two people. In this case the secret reading mask is given to both Alice and Bob before their deployment.

In addition to a secret reading mask, Alice and Bob have to share some secret information to be used for witnesses to catch the intruder. In particular, for every witness qubit two informations are needed: an index describing its position in the header and the binary value it carries.

Once Alice and Bob share the reading mask, they can exchange messages. Either Alice or Bob can initiate such a message. Suppose Alice wants to send a message to Bob. She generates a random key k to be used only once. Then she encrypts the information with k and places it in the body of the message. Subsequently, Alice quantum-encrypts k using the reading mask, places it in the header, and sends the message. Fig. 5.3 shows the steps performed by Alice.

Bob on the other end receives both the header and the body of the message. He first decrypts the header using the secret reading mask and retrieves the one-time pad k. Then Bob decrypts the message body with the one-time pad k. Fig. 5.4 shows

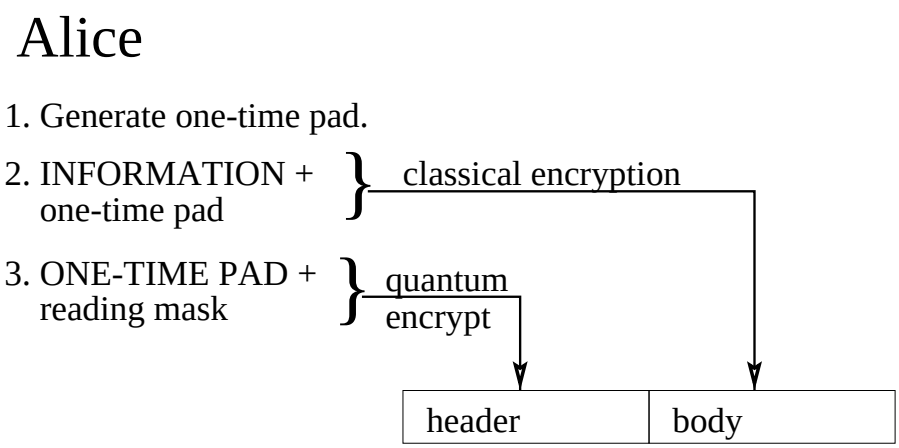

Figure 5.3: Alice takes four steps to send a message to Bob.

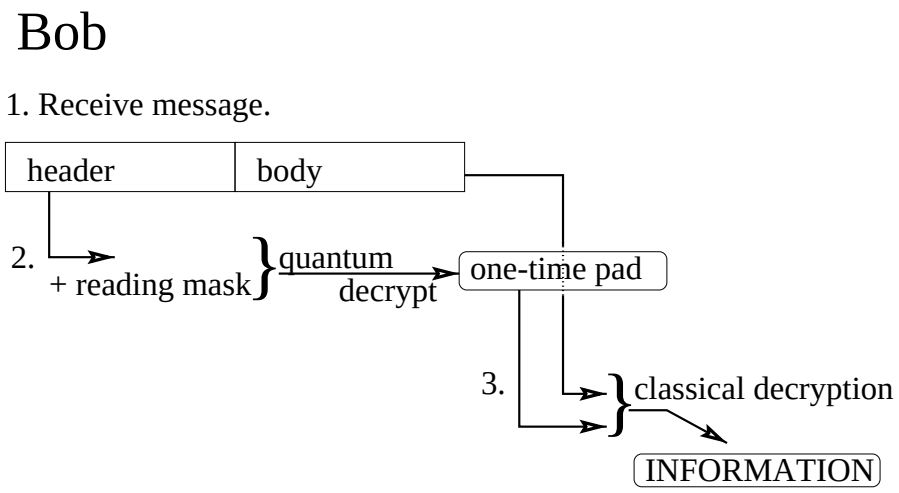

Figure 5.4: Bob takes three steps when receiving a message from Alice.

the steps performed by Bob.

5.3 What Eve Can Do

We consider the standard setting, in which Alice and Bob are trusted. The communication channel is vulnerable to attacks from Eve. Eve may intercept a message, try to read it, change it, or send spurious messages. Suppose Eve intercepts a message. She is interested to decrypt the message body that contains the information being transmitted. Looking at the encrypted message body does not reveal anything about the decryption key. In order to get the secret key, Eve has to look at the header of the message. Because the header of the message is quantum encrypted, Eve has no way of knowing how to read the header.

The best Eve can do to retrieve the one-time pad through measurement of the head, is to guess the reading strategy, meaning that she guesses the value of the reading mask. Suppose Eve has chosen to read all qubits simply in the computational basis. This means the reading mask would be

$$mask = * \ \ * \ \ *...$$

Actually all other guesses are equivalent in terms of the performance of the guess. Note that on average 25% of the header qubits are indeed encoded with $*$, 25% of the qubits are encoded with NOT, 25% with H, and 25% with $H \ \ NOT$. Therefore Eve's guessed key will have the following performance:

1. On the 25% $*$ qubits, all guesses are correct.

2. On the 25% NOT qubits, all guesses are wrong.

3. On the 25% H qubits, half of the guesses are correct.

4. On the 25% H NOT qubits, half of the guesses are correct.

Overall, 50% of the secret key bits are guessed correctly and the rest are wrong. Yet this is exactly as if Eve guesses the secret key directly, by tossing a coin, without bothering to read the header at all. It means that without any information about the reading mask, Eve cannot get any information about the secret key.

Eve's intervention, on the other hand, is detectable by Bob. If Eve has touched a witness qubit of the header of some message through reading, the state of the qubit will be detectably changed. This situation has been described in detail in the previous chapter. Remember that Eve does not know whether a qubit is in a simple state or in a superposition.

Suppose the qubit is in a simple state: $|0\rangle$ or $|1\rangle$. If Eve chooses to read the qubit in the computational basis, her intervention remains hidden. Yet, if Eve chooses to rotate the qubit with a Hadamard gate first and then measures, she will have rotated the qubit into a superposition: $H0$ or $H1$. Measurement collapses the superposition to any binary value with 50% chance. Thus, there is a 50% chance that the qubit has collapsed to the wrong value. As Eve has a 50% chance to choose applying a H gate, the chance of altering the value of the qubit is 25%. A similar scenario happens if the qubit was initially in a superposition $H0$ or $H1$. If Eve decides to measure in the computational basis, the result of the measurement has a 50% chance of being wrong. If Eve applies a H gate, measures and then rotates the qubit back with a H gate, her intervention remains hidden. Overall, Eve has altered the value of the qubit with a chance of 25%.

For Bob to notice the change in the value of the qubit, he has to expect the qubit

to have a certain value. For this, Alice has to set the witness qubits to the known values. Bob checks the witnesses and if they match the expected values, he concludes that Eve has not touched the header.

As such, any reading intervention of Eve will be of no advantage to her and will be probabilistically detectable by Bob. The probability to detect Eve's intervention grows exponentially with the number of qubits touched by Eve.

The only "information" Eve can see is the encoded message body. While the encoded version does not reveal anything about the information content, it is still a sequence of encoded information that Eve can read and copy without leaving any trace of her intervention. We will see in the next section that fully quantum messages will restrict Eve even in this action.

5.4 Fully Quantum Messages

Consider now that qubits are easily available and there are no technical impediments to use fully quantum messages. That is, both the header and the body of the message are arrays of qubits.

In this case, we may consider the body of the message to be quantum encrypted. As such

- the header contains the one-time pad, quantum encrypted with the secret encryption mask, and
- the body contains the intended information, quantum encrypted with the one-time pad.

When Alice wants to send a message to Bob, she goes through the same steps as before. Alice randomly generates a one-time pad that gets quantum encrypted with the secret reading mask and represents the header of the message. Then Alice quantum encrypts the information with the one-time pad. This is done by interpreting the one-time pad as a reading mask. The one-time pad is divided into groups of two bits. Every group of two bits encodes the reading strategy ($*$, NOT, H, $H\ NOT$) of one qubit. Then Alice places the encrypted message in the message body. Bob, on receiving the message performs the same steps reversed. The only difference to the partially quantum communication is that the body is quantum encrypted.

5.4.1 Discussion

Fully quantum message exchange offers at least all advantages of the partially quantum message exchange. In addition, whenever Eve attempts to read the message body, she changes the states of the superpositions of the qubits she reads. If the message body contains witness qubits as described in section 5.3, Bob can detect her intervention. It means that Eve is detectable now whenever she reads a part of the message.

When Eve reads the message body, she has absolutely no benefit from the classical binary array she obtains. This array is not even some classical encryption of the information transmitted. The array Eve gets is actually not even uniquely determined, as it depends on how the superpositions collapse.

Therefore, what Eve reads is nothing but garbage and has no clear connection to the binary information to be transmitted. It is not even fully predictable and its predictability depends on Eve's reading strategy for each qubit, whether Eve measures

directly or applies a Hadamard gate.

5.5 Conclusion

We have seen that one-time pad communication is made readily available by quantum or partially quantum messages. The one-time pads are generated as needed by the initiating party of the communication. The two parties do not need to meet in order to agree on the value of the one-time pad. The only secret information that Alice and Bob share a priori is a reading mask. This reading mask defines the quantum encryption of the one-time pad.

Because the one time pads are randomly generated, even if Eve would possess a series of quantum-encrypted one-time pads, this would not reveal anything about the reading mask. In addition, as Eve's intervention is detectable, Alice and Bob will know how many message headers or fully quantum messages have been captured by Eve. Therefore, Alice and Bob will know whether Eve has in her possession a series of header readings and may act accordingly.

The unprecedented advantage of using fully quantum messages is that Eve is clearly detectable in all her actions. If Alice and Bob have communicated undisturbed for some time, the amount of shared secret information increases. They may agree on a longer or a new reading mask, using their previous message exchanges only. As such, the confidence and security of the communication increases over time.

The idea of encrypting the encryption key and sending it along with the encrypted message is part of the cryptographic folklore: for example a DES (data encryption standard) key is used to encrypt a message, but is itself encrypted using a RSA key.

RSA stands for Rivest, Shamir and Adleman the authors of an algorithm for public-key cryptography. Our quantum version of the idea adds to the benefits of the classic version the detectability of the intruder and the impossibility to copy the qubits. Thus, the intruder is detected when only reading the message header or body. Also, for fully quantum messages, both the encrypted key and the message body cannot be copied by Eve for later use.

Chapter 6

Quantum Access Control in a Hierarchy

This chapter revisits the problem of access control in a hierarchy [3]. A collection of data, such as a database, is accessed by a very large number of users. Users have different access rights to the data items. Regular users are organized in groups and may access group specific data and data of general interest. Managers and directors are able to access data belonging to a whole category or group of users and may also access data to remain secret from regular users. Groups of users are organized as a

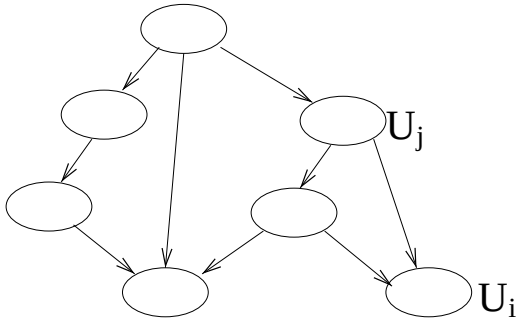

Figure 6.1: Formal sets in a poset.

partially ordered set (poset), where each node, or group of users, represents a category with identical access rights (see Fig. 6.1). A node that is in a parent position in the hierarchy shares all access rights of its children.

To formalize, consider two sets U_i and U_j (see Fig. 6.1), that are members of the poset. The partial order $U_i \leq U_j$ means that U_i is on a lower level than U_j in the poset. Additionally, there is a line in the diagram connecting U_i with U_j. U_i, on the lower level, has less access rights than U_j, and conversely, U_j can do anything that U_i can do.

6.1 Access Control in a Hierarchy Using Classical Cryptography

Classical cryptographic solutions to the problem of access control build on a system of secret keys. Each user has a secret key that is used by the encryption/decryption function to transform encoded information into readable format. A manager or director, who is the root of a subtree, has a key that subsumes all keys in the subtree [26].

The mechanism that uses secret keys, generally works with a secret encryption key k^e and a secret decryption key k^d. The original text v is encrypted with an encryption function E to obtain the encrypted text u:

$$u = E_{k^e}(v).$$

The original text can be retrieved by decrypting u with the key k^d:

$$v = E_{k^d}(u).$$

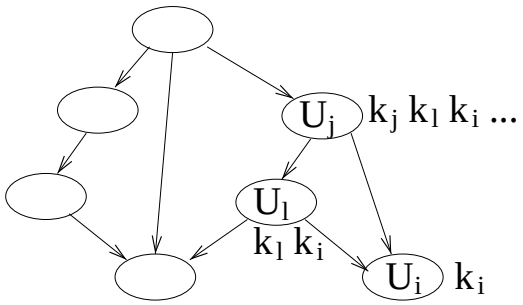

Figure 6.2: Straightforward cryptographic solution.

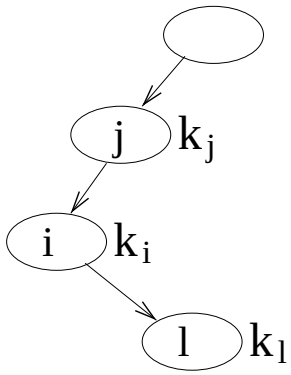

Figure 6.3: Solution for a totally ordered set.

Depending on the encryption method, the two keys, the encryption key and the decryption key, may coincide.

Several cryptographic solutions have been proposed. They all come with a specific range of successful applicability as well as their weakness or disadvantages:

1. First straightforward solution. The first solution (see Fig. 6.2) begins by assigning separate keys to all members of the poset. These keys are independent, meaning one key cannot be obtained from another. The users of authority, higher up in the poset, inherit all keys from children groups below. Though this solves the problem of access to the database, the direct disadvantage is that groups high in the poset own too many keys.

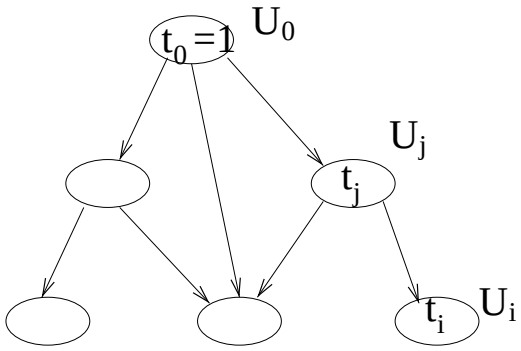

Figure 6.4: Solution to a poset that computes keys in an up-down fashion. First, assign public integers t to each group in a partially ordered set, then compute the keys.

2. Solution for a totally ordered set. If instead of a poset, the users are organized in a totally ordered set (see Fig. 6.3), there is a simple solution that assigns exactly one key per user. We define a one-way function f and also initialize the key of the root. Consequently, the key of a child is computed as the function f of the key of the parent: $k_i = f(k_j)$ and $k_l = f(k_i)$. The child is given its key only. Because of the noninversability of f, the child cannot compute the keys of its ancestors.

3. Up-down computable keys for a partially ordered set. The method we describe now, combines the advantages of the previous two methods. It assigns exactly one secret key to each group of users, while preserving the order of a partially ordered set. By choice, the root is the first to be assigned a secret key K_0. This key is known only to the root, thus hidden from anybody else in the poset. Again, by choice, a number M is defined as the product of two large primes p and q: $M = p \times q$. The number M will be used for modulo operations. Additionally, a structure of integers $t_i, t_j, ...$, is assigned to the poset (see Fig. 6.4). These integers are public.

The condition on the integers is that the integer of a parent divides the integers

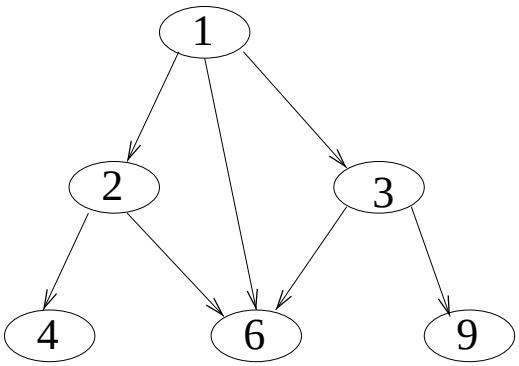

Figure 6.5: The integer assigned to a child node is a common multiple of the integers of its parents.

of its children (see Fig. 6.5). Formally, if $U_i \leq U_j$ then $t_j | t_i$. For a node with several parents, the integer assigned to it may be the least common multiple of the integers of all parents, or simply *some* common multiple.

Now all secret keys can be computed. For group U_i, with its integer t_i, the secret key is a power of the initial K_0:

$$K_i = {K_0}^{t_i} \bmod M.$$

Each user of some group U_i gets only the key of its group K_i.

This ingenious scheme now allows a user to compute all keys that are below in the hierarchy. For $U_i \leq U_j$, U_j, using K_j, can compute K_i, namely

$$K_i = {K_0}^{t_i} = [{K_0}^{t_j}]^{\frac{t_i}{t_j}} = [K_j]^{\frac{t_i}{t_j}} \bmod M.$$

By intention, U_i cannot compute K_j, as it is computationally intractable to extract roots modulo a large number.

This method gives simplicity to the access procedure. The major disadvantage is that it can be broken by collusion attacks. This means that users on lower levels can collaborate to compute a higher level key. The following is an example of a collusion

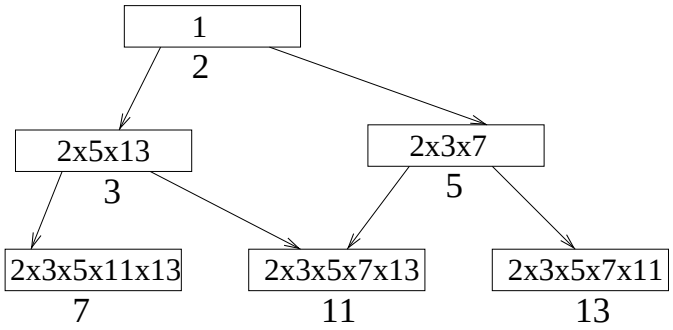

Figure 6.6: Robust up-down computable keys using a structure of primes.

attack. Suppose U_l, with $t_l = 4$, collaborates with U_i, with $t_i = 9$. Their secret keys are $K_l = {K_0}^4$ and $K_i = {K_0}^9$, respectively. The operation $(K_l)^{-2}K_i = {K_0}^{-8}{K_0}^9 = K_0 \bmod M$ computes the secret key of the root.

4. Robust up-down computable keys for a partially ordered set. The shortcomings of the previous solution can be solved, by choosing the integers t_i appropriately. First, the poset is assigned a structure of primes p_i (see Fig. 6.6). The integer t_i, associated with class U_i, will be defined as the product of all primes associated with nodes not below U_i in the poset.

$$t_i = \prod_{U_j \not\preceq U_i} p_j$$

The secret keys are computed as before $K_i = {K_0}^{t_i} \bmod M$. The advantage of this scheme is that it eliminates collusion attacks. A parent can still easily compute the keys of its children.

This scheme satisfies an organization where the users are stable, and few people join or leave the organization over a considerable time. This may be unrealistic in real life. If a user leaves the organization, the user's secret key has to be invalidated. This means right away that the whole group that the user belonged to has to receive a new secret key. Because keys in the hierarchy are interdependent, invalidating one

key may affect a whole area of the poset. In the worst case, the whole poset needs to receive new keys. The same problem arises when a user gets a promotion and becomes a member of a different group. Again, the same difficulty may appear when a new group of users are to be added. It may be easier to add a group at the bottom of the poset and more difficult to insert a group at some arbitrary level of the poset. In the literature [4] [15] [26] [37] [40] [43] [50], numerous schemes based on cryptography have been proposed to address these problems, but none of them truly succeeds.

We will see in the next section that a quantum solution addresses many of these issues.

6.2 Quantum Setting for Access Control

The quantum scheme designed here takes full advantage of quantum cryptography [39] [46]. It achieves the following improvements. Any local change to a user does not affect the other users. In particular, any user may join or leave the system without affecting the other members of the user community. Also, if a user changes its position in the hierarchy, such as being promoted to a manager position, it is only this user's key that will have to be changed.

6.2.1 Quantum Card and Classical Key

The access to the database is managed by two keys. Every user has **two keys**: a classical key and a quantum key (see Fig. 6.7). The purpose of the user's keys is to provide the information necessary to generate a database access key, *dbAccess*. The database access key is the one that defines the access rights of the user to the

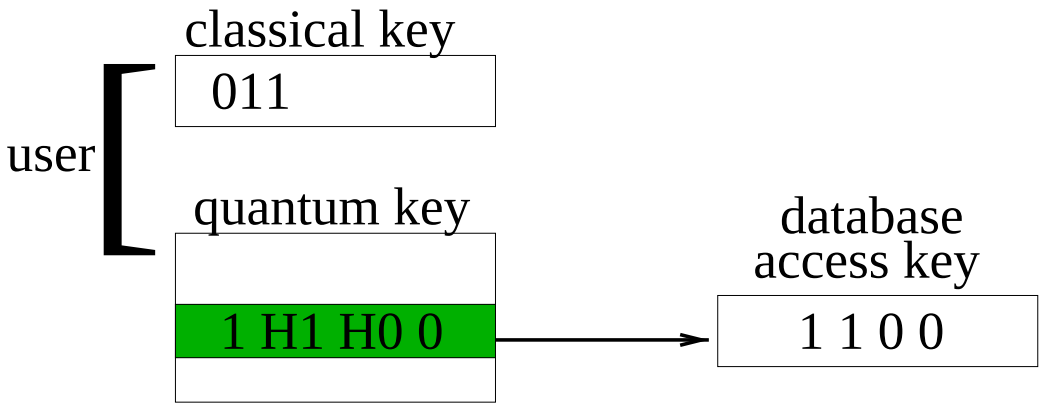

Figure 6.7: Each user has two keys: a classical key and a quantum key.

database. *dbAccess* is not in the direct possession of the user. And again the two keys that *are* in the possession of the user serve the sole purpose to retrieve *dbAccess* and are meant to *hide* the value of *dbAccess* from the user.

The user's *classical key* is a binary number. This number is unique for each user and secret, expected to be known to that user only. It is equivalent of a password and thus it is the user's responsibility to keep it secret.

The user's *quantum key* is an array of qubits and is registered on a card. The quantum key is a quantum encrypted version of the database access key *dbAccess*. This key may be unique to the user or even unique for each session. This means that each time a user connects to the database, the quantum card may have another quantum key written on it.

The quantum key is not known to the user. Although, the key is written on the card, the user does not need to know its qubit values. The qubits written on the card are in different quantum states. Some states represent classical values, such as $|0\rangle$ and $|1\rangle$. Other qubits are in a balanced superposition of $|0\rangle$ and $|1\rangle$, namely $\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle) = H|0\rangle$ and $\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle) = H|1\rangle$. As the user does not know which qubits

Access Key	Quantum Key	Decryption Mask
1100	0 1 $H0$ 0	$NOT * H *$
	$H0$ $H1$ 0 1	$(H\ \ NOT)\ H * NOT$
	$H1$ 1 $H0$ $H0$	$H * H\ H$
	$H0$ 1 $H0$ $H1$	$(H\ \ NOT) * H\ (H\ \ NOT)$

Table 6.1: There are 4^n quantum keys that encrypt the same access key. The decryption mask yields the reading strategy to obtain the access key.

are simple states and which are in a superposition, the user cannot retrieve his/her *dbAccess* key by illicitly reading the card. Moreover, reading the card destroys the quantum states of the qubits, as they collapse to some classical value. Thus, a card that has been illicitly read, cannot be used afterwards to connect to the database.

Quantum encryption was defined in the previous chapter, chapter 5. The quantum encryption is not unique. Each bit of *dbAccess* may be quantum encrypted in four different ways:

1. *****: Copied directly with no change.

2. **NOT**: The bit is negated.

3. **H**: The bit is transformed with a Hadamard gate.

4. **H NOT**: The bit is negated and then transformed with a Hadamard gate.

If the access key is n bits long, there are 4^n possible quantum keys that encrypt the same *dbAccess* key. In table 6.1 the second column shows possible encryptions of a short example-key $dbAcces = 1100$.

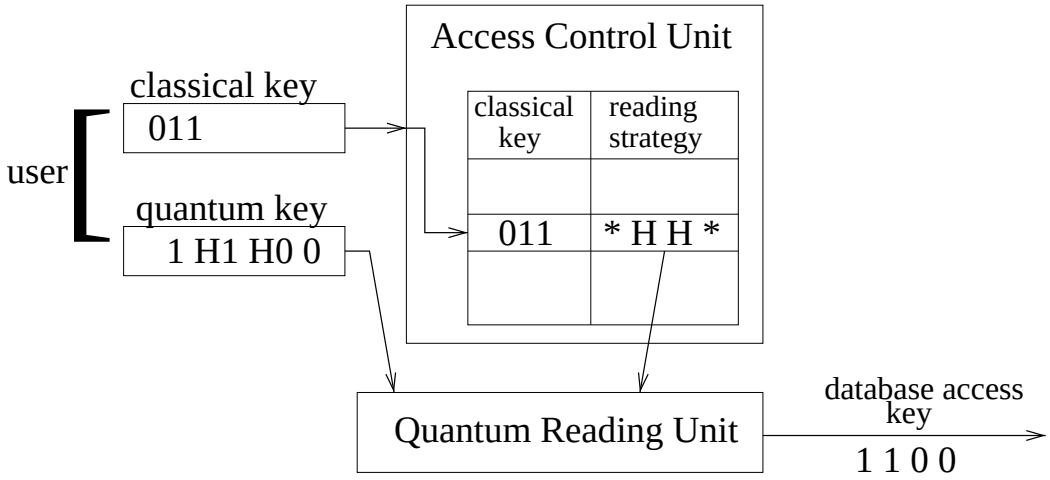

Figure 6.8: How the database access key is obtained.

6.2.2 The Access Control Unit

To know how to retrieve the access key from the array of qubits of the quantum key, we need to have a decryption mask (see chapter 5). The decryption mask simply says how to read the qubits of the quantum key in order to obtain the intended binary value. It shows the positions in the qubit array of the quantum key that are in superposition and/or negated. The third column of table 6.1 defines the decryption masks for the quantum keys of the second column.

In order to manage the decryption of the quantum keys, there is a unit attached to the database, called the access control unit, ACU (see Fig. 6.8). The ACU translates the two user keys into the final database access key. The ACU has a table that has entries for each user's classical key. A decryption mask, or reading strategy mask, corresponds to each classical key value. This reading strategy mask is then submitted to the quantum reading unit. This unit is now able to correctly read the quantum key. Simple qubits are read directly and qubits in superposition are first transformed by a Hadamard gate. If necessary, the bits are then negated. Thus, the output of the

quantum reading unit is the final database access key.

Note that the ACU is attached to the database. It is not visible to the user. Once deployed, the ACU does not need to be managed by a human.

When a user accesses the database, the user has to type in the classical key and also provide the quantum card for reading. Whenever the card is used, the quantum key is destroyed by reading. Therefore, at the end of each session, the card needs to be restored, meaning that the quantum key is written back to the card. It is interesting to note, that the quantum key need not be the same. At the end of the session, the ACU may generate a new, random reading mask, and then write a new quantum key on the card. Thus, the user has a quantum key per session.

6.2.3 Changes in the User Structure of the Organization

The clear advantage of this scheme is that the user is disconnected from the database access key. The user has no knowledge of its value and no way of retrieving information about it.

Now the *dbAccess* key defines the access rights of the user as a member of the poset. As *dbAccess* is hidden from the user, a hierarchical structure, as the ones described at the beginning of this chapter will serve the purpose. For example, the *dbAccess* can be obtained by the up-down computable key method.

When a user joins an existing group, this means that the node of the poset exists and is working. The particular access key of the group will be assigned to the new user, using some arbitrary quantum encryption. A line is added to the ACU's table to represent the new user. In addition, a classical key will be given to the user. This key is independent of the poset structure and is an index in the ACU's table. When a

user leaves the organization, its line in the ACU table is invalidated. Therefore, there is no entry in the ACU for this particular classical key. The user can no longer access the ACU with the classical key. Normally, the user would be required to return the quantum card, but this does not affect the security of the system, as will be seen in the next subsection.

When a user changes its position in the hierarchy of the poset, its quantum card needs to be updated to a quantum encryption of the new database access key. The quantum card will reflect the change in the access rights. The classical key may remain the same. Also, the ACU table needs to be updated with a new decryption mask.

Note that all changes described above affect exactly one user, namely the user whose status is changed. This is remarkable, compared to all previous classical solutions existing in the literature, as described in section 6.1. In our scheme, a change in the status of an arbitrary user leaves all the rest of the users undisturbed. This is an important advantage, when considering large organizations with millions of users and presumably a very dynamic structure.

The scheme designed is less adaptable to changes in the poset structure itself. Adding another leaf to the poset, that is, creating a new group, should pose no problems, as a new *dbAccess* can be created to define the node. This key would be some common multiple of its parents. Yet, it might be difficult to insert a node in some arbitrary position of the poset, as an appropriate *dbAccess* key might not be available. Deleting a group of users is easy again, as it simply means to stop using a certain database access key.

6.2.4 What the Intruder Can/Cannot Do

Let us consider first that the intruder has access to the property of the users only, but cannot access the ACU, as it is stored in a secure place.

If the intruder, Eve, steals the classical key, she will have absolutely no access to the system without the quantum card. In the same way, if Eve steals the quantum card, but does not know the classical key, she cannot access the ACU.

Because of the nonclonability theorem, Eve cannot copy the quantum card. If Eve tries to read the quantum card, she destroys the quantum key, and the card will be unusable to the legitimate user as well. As Eve does not know the decryption mask, there is no way of reading the quantum card and gaining some knowledge about *dbAccess*. In fact, as shown in chapter 5, if Eve guesses a reading strategy, the probability on each qubit to be measured correctly is still 50%.

Therefore, Eve's only option is to steal both the classical key and the quantum card. Note that this is a complete identity theft. The legitimate user has lost its quantum card. Yet, this theft is *detectable*, the legitimate user will *know* that his/her identity has been stolen: *the user cannot find his/her card*. In this case, the user's identity has to be invalidated from the cryptographic system, and a new identity has to be given to the user. Again, an aspect specific to this quantum system is that an identity cannot be copied. It is not possible that two persons carry the same cryptographic identity.

Let us consider now that Eve may gain access to the ACU. The ACU has no *dbAccess* key stored into it. Just looking at the ACU's table does not reveal anything about the access key, as the access key is solely written on the quantum card. Therefore, Eve has absolutely no gain from looking at the ACU, unless she also has both a

classical key and a quantum key. This means that Eve would need to steal both the identity of a user and access the ACU in a very short interval, which is practically difficult.

Also, two or more users cannot collaborate to break the system. They cannot even gain knowledge about their own *dbAccess*. This is because their quantum keys are different and have no meaningful connection to the value of *dbAccess*, except through the decryption mask.

6.3 Conclusion

Our scheme shows that adding quantum keys to the access mechanism of a database has advantages both in terms of security of the system and of adaptability to changes in the underlying user structure.

The system cannot be broken easily, as the quantum key cannot be copied, and in fact may be unique for the session. The identity of a user cannot be stolen without the user noticing the theft.

Also the system is designed to support a large variety of changes in the user structure. Users may join and leave the organization without affecting the security system at large.

The idea of using two keys, a classical key and a quantum key, is not necessarily connected to this specific application, namely access control in a hierarchy. In fact, the access system behind the two-key front end, may have any structure. The idea may be successfully applied, whenever the user is to be distanced from the actual security access of the sensitive data. In our scheme, the user is distanced from the hierarchical structure of the users' security rights. It is the specific value of *dbAccess*

for each user, which reflects the security rights. The *dbAccess* keys of all users form the poset structure and are therefore considered to be defined according to the up-down computable method (see section 6.1).

The vulnerability of the database itself, or the ACU has not been treated in this scheme. As a future work, we envision to store both the database and the table of the ACU using quantum memories. This would allow the definition of a security scheme for these data based on a quantum cryptographic approach. For example, the database could be quantum encrypted.

Chapter 7

Conclusion

We have seen by various examples that quantum computation recommends itself to improve cryptographic schemes. This applies to both standard schemes with two equivalent partners (Alice and Bob), as well as nonstandard schemes, such as wireless sensor networks and the secure access to a database system. Improvements seem to be available at every turn.

The quality of a security scheme is described by security parameters, such as the computational effort to break a scheme, the detectability of an intruder, the amount of secret information that leaks out of the system, and so on.

One contribution of this thesis is to show that quantum cryptography at least replicates *all* security parameters obtainable using classical cryptography.

Thus, consider that quantum cryptography was thought to be unable to replicate all the results of classical cryptography. In particular, it is known that authentication can be done by classical means, but authentication was thought to be impossible with quantum cryptographic methods only. Yet, in this thesis, we have proven that quantum authentication lies intrinsically within the capabilities of quantum cryptography.

Chapter 3 presented two elegant algorithms that develop a secret key and authenticate the communication partners without stepping into classical cryptography. In addition, the algorithms are direct and simple.

A second contribution of the thesis is to show that certain security characteristics that are partially solved by classical cryptography are effectively solved, or at least considerably improved, with quantum approaches.

Quantum cryptography has been shown to bring two important improved properties over classical cryptography. First, secret keys generated by quantum algorithms are effectively unbreakable, and secondly, the intruder is easily detectable. All previous quantum key enhancement algorithms exhibit these properties. This thesis extends the cryptographic settings in which secret keys can be quantum generated. In particular, we have designed a cryptographic solution for wireless sensor networks. The secret keys used for communication in the network are quantum generated and thus inherit the two properties of effectively unbreakable keys and the detectability of the intruder.

Some characteristics of the schemes presented in this thesis come specifically from the quantum properties of the cryptographic systems and have no equal in classical cryptography. Maybe the most attractive quantum specific result has been given in chapter 4 and refers to the protection of the identity of the sensor nodes. It was said there that the intruder is allowed to pick up a sensor node and read and write its content. The quantum memory of the sensor node unequivocally preserves its identity. This is to the point that even looking at the node, that is, reading its memory, reveals the intrusion. Intrusion of a node destroys its identity. This result is uniquely attainable by relying on the quantum noncloning theorem and cannot be

replicated by classical definitions of identity. The presence of qubits renders a direct definition of a node's identity as its quantum signature. Therefore, as long as the signature exists, and this can be tested, the node exists in its original, pristine form, the absence of any evil intervention is ensured. This results from using quantum "hardware".

Another specifically quantum result presented in this thesis comes from using quantum secret keys. We have been accustomed to think of a secret key as an array of bits or as a piece of text in a human language. In the case of quantum keys, as presented in chapters 5 and 6, the secret key is connected to its physical support. That is, the secret key cannot be disassociated from its expression as a string of physical qubits. Because qubits of an unknown state cannot be copied, the secret key exists for as long as its physical support exists. In particular, an intruder may destroy a quantum secret key to the point where it is irretrievable, or unusable by the communicating parties. This happens with the intruder gaining *no* secret information. The explanation is that the intruder destroys the key by changing its physical support. In chapter 5, this property facilitates the creation of one-time pads on demand, without the parties having to bother with a secret meeting. Chapter 6 exploits the same property to generate uncopyable access keys.

Stepping now a few steps back for an overview of the whole work, we can see that the use of quantum cryptography brings about a fundamental change in the way we think about secret communication. This refers to a change in the philosophy of cryptography, of some core beliefs taken for granted. For example, one such belief was stated as follows:

The more Alice and Bob communicate secretly in a certain cryptographic setting, the less secure that setting becomes and it needs to be updated or reinitialized from outside of the system.

From this principle comes the advice to change passwords regularly. If a password has been used for a long time, it has been exposed to attacks all along and *some* of its content might have leaked out. Also, security systems tend to be checked after a while, in order to determine whether they still work reliably. This check is done from *outside* of the security system.

A fully quantum security system may work in a fundamentally different way. Consider the result from chapter 3. Authentication can be done with quantum cryptography. An effectively secret key can be distributed using protected *public* information only. Using this secret key as a reading mask, Alice and Bob may now communicate secretly with one-time pad encrypted messages (see chapter 5). Moreover, after communicating for some time, Alice and Bob will know whether Eve was present in the slightest way during their communication, as Eve's intervention, such as simple reading is detectable. This means that the more Alice and Bob use the quantum communication system, the more their confidence in the secrecy of their communication increases. Implicitly, the amount of effectively secret information shared by Alice and Bob increases over time. This may include the change or lengthening of the reading mask, or the change of the encryption/decryption strategy. That is, Alice may communicate secretly to Bob the intention of changing the reading mask, and after Bob agrees by an equally secret message, Alice may encrypt and send a new, longer encryption mask. Note that these revisions of the security scheme are now made from the *inside* of the system. A check from the *outside* of the system is no longer needed.

The following short story summarizes the quantum philosphy of cryptography:

> *Alice and Bob, two cryptographic entities unknown to each other, meet in a public place. The public place may be a crowded cafeteria. Here, a formal introduction takes place. Alice and Bob present themselves with some public names, called public keys. Interesting to note is that these public names are interdependent. Alice would introduce herself in a personalized way to Bob, different from the presentation to Charles, for example. Once the introduction is over, Alice and Bob leave the cafeteria to go each on a different life-path. From then on, Alice and Bob can communicate secretly "forever". As time passes, their "acquaintance" develops into a "friendship" and as such the trust in the privacy of their messages increases.*

As this heart-warming story suggests, the quantum cryptographic system is self-sustainable and moreover builds itself up over time.

With the attention turned towards future work, we may see that cryptographic systems subject to the philosophy above are just at initiating stages. We believe that a large variety of cryptographic needs may be answered with quantum schemes. The present thesis is a modest beginning.

The present research invites to many extensions:

1. Security in sensor networks has been defined as an unconventional cryptographic setting that benefits from quantum cryptography. It is an open question what other unconventional cryptographic settings and needs can be successfully addressed with quantum cryptography.

2. In chapter 6 a quantum scheme has been developed to solve access control in a

hierarchy. The method is not restriced to a hierarchy, but can be applied to any cryptographic structure. It would be useful to identify domains of cryptographic structures to which our two keys access method is relevant. Note that each structure might need a specific adaptation.

3. For the same problem of access control in a hierarchy, the problem of encrypting the database and the access control unit themselves has been left open. The idea of a quantum database with a quantum access control unit is worth pursuing, as it would allow a quantum encryption of the data and would permit the detection of an intruder.

4. As it was mentioned above in this chapter, the philosophy of quantum cryptography is different from its classical counterpart. Quantum cryptography allows a cryptosystem to become securer over time. None of the schemes presented in this thesis actually exploits this capacity. It is our intention for future work to design a full scheme with this capacity, namely to start from public information and develop a secret communication scheme that is self-sustainable.

Bibliography

[1] H.M.F. AboElFotoh, E.S. ElMallah, and H.S. Hassanein. On the reliability of wireless sensor networks. In *IEEE International Conference on Communications (ICC)*, pages 3455–3460, June 2006.

[2] A. D. Aczel. *Entanglement.* Raincoast Books, Vancouver, 2002.

[3] Selim G. Akl and Peter D. Taylor. Cryptographic solution to a problem of access control in a hierarchy. *ACM Transactions on Computer Systems*, 1(3):239–248, 1983.

[4] G. Ateniese, A. De Santis, A. L. Ferrara, and B. Masucci. Provably-secure time-bound hierarchical key assignment schemes. In *Proceedings of 13th ACM Conference on Computer and Communication Security (CCS'06)*, pages 288–297, 2006.

[5] C. H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*, pages 175–179, IEEE, New York, 1984. Bangalore, India, December 1984.

[6] C. H. Bennett, G. Brassard, and D. N. Mermin. Quantum cryptography without Bell's theorem. *Physical Review Letters*, 68(5):557–559, February 1992.

[7] C.H. Bennett, G. Brassard, C. Crepeau, R. Jozsa, A. Peres, and W.K. Wootters. Teleporting an unknown quantum state via dual classical Einstein-Podolsky-Rosen channels. *Physical Review Letters*, 70:1895–1899, 1993.

[8] Charles H. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 17(6):525–532, 1973.

[9] Charles H. Bennett. Quantum cryptography using any two nonorthogonal states. *Physical Review Letters*, 68(21):3121–3124, May 1992.

[10] Charles H. Bennett, Debbie Leung, Graeme Smith, and John A. Smolin. Can closed timelike curves or nonlinear quantum mechanics improve quantum state discrimination or help solve hard problems? *Physical Review Letters*, 103(17):170502, 2009.

[11] Todd A. Brun. Computers with closed timelike curves can solve hard problems efficiently. *Journal Foundations of Physics Letters*, 116(3):245–253, 2003.

[12] Todd A. Brun, Jim Harrington, and Mark M. Wilde. Localized closed timelike curves can perfectly distinguish quantum states. *Physical Review Letters*, 102(21):210402, 2009.

[13] J. Cederlof and J. A. Larsson. Security aspects of the authentication used in quantum cryptography. *IEEE Transactions on Information Theory*, 54:1735–1741, 2008.

[14] X. Chen, K. Makki, K. Yen, and N. Pissinou. Attack distribution modeling and its applications in sensor network security. *EURASIP Journal on Wireless Communications and Networking*, 2008:11, 2008.

[15] J. Crampton. Cryptographically-enforced hierarchical access control with multiple keys. In *Proceedings of 12th Nordic Workshop on Secure IT Systems (NordSec 2007)*, pages 49–60, 2007.

[16] D. Deutsch. Quantum mechanics near closed timelike lines. *Physical Review D*, 44(10):3197–3218, 1991.

[17] P. Dirac. *The Principles of Quantum Mechanics*. Oxford University Press, 4th edition, 1958.

[18] A. R. Dixon, Z. L. Yuan, J. F. Dynes, A. W. Sharpe, and A. J. Shields. Gigahertz decoy quantum key distribution with 1 mbit/s secure key rate. *Optics Express*, 16(23):18790–18979, 2008.

[19] J. F. Dynes, Z. L. Yuan, A. W. Sharpe, and A. J. Shields. Practical quantum key distribution over 60 hours at an optical fiber distance of 20km using weak and vacuum decoy pulses for enhanced security. *Optics Express*, 15(13):8465–8471, 2007.

[20] A. Ekert. Quantum cryptography based on Bell's theorem. *Physical Review Letters*, 67:661–663, 1991.

[21] Richard Feynman. There's plenty of room at the bottom, December 1959.

[22] Richard Feynman, R. B. Leighton, and M. Sands. *The Feynman Lectures on Physics*, volume III. Addison-Wesley, Reading, Mass., 1965.

[23] Brian Greene. *The Fabric of the Cosmos, Space, Time, and the Texture of Reality.* Alfred A. Knopf, New York, 2004.

[24] M. Halder, A. Beveratos, N. Gisin, V. Scarani, C. Simon, and H. Zbinden. Entangling independent photons by time measurement. *Nature Physics*, 3:659–692, 2007.

[25] P. A. Hiskett, D. Rosenberg, C. G. Peterson, R. J. Hughes, S. Nam, A. E. Lita, A. J. Miller, and J. E. Nordholt. Long-distance quantum key distribution in optical fibre. *New Journal of Physics*, 8(193), 2006.

[26] Stephen J. MacKinnon, Peter D. Taylor, Henk Meyer, and Selim G. Akl. An optimal algorithm for assigning cryptographic keys to control access in a hierarchy. *ACM Transactions on Computers*, c-34(9):797–802, 1985.

[27] Marius Nagy. *Using Quantum Mechanics to Enhance Information Processing, PhD Thesis.* Queen's University, Kingston, Ontario, June 2007.

[28] Marius Nagy and Selim G. Akl. Quantum key distribution revisited. Technical Report 2006-516, School of Computing, Queen's University, Kingston, Ontario, June 2006.

[29] Marius Nagy and Selim G. Akl. Entanglement verification with application to key distribution protocols. In *Proceedings of the 2008 International Conference on Information Theory and Statistical Learning (ITSL'08, part of WORLDCOMP'08)*, Las Vegas, Nevada, July 2008.

[30] Naya Nagy and Selim G. Akl. Authenticated quantum key distribution without classical communication. *Parallel Processing Letters*, 17:323–335, 2007.

[31] Naya Nagy and Selim G. Akl. Quantum authenticated key distribution. In *Proceedings of International Conference on Unconventional Computation. Lecture Notes in Computer Science 4618*, pages 127–136. Springer-Verlag, Heidelberg, 2007.

[32] Naya Nagy, Marius Nagy, and Selim G. Akl. Key distribution versus key enhancement in quantum cryptography. Technical Report 2007-542, School of Computing, Queen's University, Kingston, Ontario, 2007.

[33] Naya Nagy, Marius Nagy, and Selim G. Akl. Quantum wireless sensor networks. In *Proceedings of the Seventh International Conference on Unconventional Computation, Vienna, Austria*, pages 177–188, Berlin, August 2008. Springer-Verlag. in: Calude C.S., et al, Eds., Lecture Notes in Computer Science 5204.

[34] Naya Nagy, Marius Nagy, and Selim G. Akl. Sensor networks with quantum memories. Technical Report 2008-551, School of Computing, Queen's University, Kingston, Ontario, October 2008.

[35] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, 2000.

[36] A. Perrig, R. Szewczyk, V. Wen, D. E. Culler, and J. D. Tygar. SPINS: security protocols for sensor networks. In *Mobile Computing and Networking*, pages 189–199, 2001.

[37] I. Ray and N. Narasimhamurthi. A cryptographic solution to implement access control in a hierarchy and more. In *Proceedings of the 7th ACM Symposium on Access Control Models and Technologies*, pages 65–73, Monterey, CA, 2002.

[38] Ronald L. Rivest, Adi Shamir, and Len M. Adleman. A method of obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[39] Jr. S. J. Lomonaco. A Talk on Quantum Cryptography or How Alice Outwits Eve. In *Proceedings of Symposia in Applied Mathematics*, volume 58, pages 237–264, Washington, DC, January 2002.

[40] R. Sandhu. Cryptographic implementation of tree hierarchy for access control. *Information Processing Letters*, 27:1–100, January 1988.

[41] Erwin Schrödinger. Discussion of probability relations between separated systems. In *Proceedings of the Cambridge Philosophical Society*, number 31, pages 555–563, 1935.

[42] Claude Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28(4):656–715, 1949.

[43] V. Shen and T. Chen. A novel key management scheme based on discrete logarithms and polynomial interpolations. *Computers and Security*, 21(2):164–171, 2002.

[44] B.-S. Shi, J. Li, J.-M. Liu, X.-F. Fan, and G.-C. Guo. Quantum key distribution and quantum authentication based on entangled states. *Physics Letters A*, 281(2-3):83–87, 2001.

[45] R. Ursin, F. Tiefenbacher, T. Schmitt-Manderbach, H. Weier, T. Scheidl, M. Lindenthal, B. Blauensteiner, T. Jennewein, J. Perdigues, P. Trojek, B. mer, M. Frst, M. Meyenburg, J. Rarity, Z. Sodnik, C. Barbieri, H. Weinfurter, and A. Zeilinger.

Entanglement-based quantum communication over 144 km. *Nature Physics*, 3:481–486, 2007.

[46] L. Vaidman. Teleportation of quantum states. *Phys. Rev. A*, 49(2):1473–1476, Feb 1994.

[47] Serge Vaudenay. *A Classical Introduction to Cryptography: Applications for Communications Security.* Springer, 2006.

[48] Stephen Wiesner. Conjugate coding. *SIGACT News*, 15:78–88, 1983.

[49] W. K. Wootters and W. H. Zurek. A single quantum cannot be cloned. *Nature*, 299:802–803, October 1982.

[50] C. Yang and C. Li. Access control in a hierarchy using one-way functions. *Elsevier: Computers and Security*, 23:659–644, 2004.

[51] F. Zhao and L. Guibas. *Wireless Sensor Networks - An Information Processing Approach.* Elsevier, 2004.