Quantum Cryptography : On the Security of the BB84 Key-Exchange Protocol

Thomas Baignères

EPFL - LASEC
thomas.baigneres@epfl.ch

Abstract. In 1984, C.H. Bennet and G. Brassard proposed a new protocol aimed to solve the problem of symmetric cryptographic key exchange. This protocol was called BB84 after the name of its authors. While a traditional method would rely on public key cryptography (like RSA), the BB84 protocol takes benefit of the laws of quantum mechanics, like for example the fact that any quantum measurement can perturb the system. Traditional public key algorithms security often rely on a typical hard mathematical problem. It is well known for example that the ability to factorize easily any number would make the usage of RSA completely insecure. Quantum Key Exchange (QKE) protocols security cannot be proved in a similar way. In this work, we will try to give an overview of security proofs of quantum key exchange protocols, focusing on the BB84 protocol.

1 Introduction

Since the late 70's, several public key cryptographic algorithms have been proposed. Diffie and Hellman [9] first came with this concept in 1976. Since that time, several other public key cryptosystems were invented, such as the well known RSA [21], ElGamal [10] or Rabin [20] cryptosystems. Roughly, the scope of these algorithms is to allow the secure exchange of a secret key that will later on be used to encrypt a larger amount of data. All these algorithms share one particularity, namely that their security rely on some mathematical problem which is *supposed* to be hard (computationally speaking) to solve. For example, it is well known that the ability to factorize easily the product of two large primes without any indication about the primes, would lead to break the RSA cryptosystem. With current technology, the best known factoring algorithm is the Number Field Sieve [12][1]. Currently RSA cryptosystems using a large enough public key are still secure against it. Pessimistic minds argue that, as the computational power increases over time, a cryptosystem which is secure today may be insecure tomorrow. Also it is possible (under the assumption that no better

[1] In 1994, P.W. Shor [22] came up if a factoring algorithm way more powerful than the Number Field Sieve as it runs in polynomial time, but has the drawback of requiring a quantum computer which practical conception with today technology is (to the best of our knowledge) still unrealistic.

factoring algorithm is discovered) to consider that an RSA cryptosystem using a large enough key will remain unbreakable long enough to ensure that the protected information will be deprecated by the time the cryptosystem is broken, the threat seems to be strong enough to lead current research in the field towards new solutions to improve the security of cryptographic public key exchange.

In 1984, C.H. Bennet and G. Brassard [2] proposed a new key exchange protocol, called BB84 after their name, which takes advantage of the physical properties of quantum channels. At that time they could only prove its security against practical attacks, i.e. that could be implemented with existing technologies. Quantum Key Exchange (QKE) security against the most general type of attacks, i.e. those where the enemy has access to an unlimited computational power[2], has been widely studied during the past few years and several proofs have been proposed [4, 13, 14].

In the next section we will describe the BB84 Quantum Key Exchange protocol over noiseless channels and give the reason why it is secure. As realistic channels are inevitably submitted to noise, the protocol is obviously unrealistic. Although other proofs had already been proposed, P.W. Shor and J. Preskill [24] came up with the first simple proof of the security of the BB84 protocol over noisy channels. This proof relies on the security of another QKE protocol which relies itself on a fundamental topic of Quantum Information Theory, namely Quantum Error Correcting Codes. This topic will be explored in Sect. 3. Sections 4 and 5 will successively prove the security of the QKE protocol using Quantum Error Correcting Codes and show why this implies the security of the BB84 protocol over noisy channels.

2 The BB84 Protocol over Noiseless Channels

We introduce three main characters: Alice and Bob, who look forward to share a secret key, and Eve, whose objective is to obtain some information about this secret key. Alice and Bob have access to a quantum channel (which we consider to be noiseless in this section) and to a classical authenticated channel. Eve can act freely on the quantum channel, but can only listen to what happens on the classic channel. Therefore it is impossible for Eve to modify the information sent through the classical channel. On the contrary, we consider that she has total access to the quantum channel, keeping in mind that

- according to the No-Cloning Theorem of Quantum Mechanics, she won't be able to duplicate the quantum information, and that
- according to the Heisenberg Uncertainty Principle, she won't be able to completely measure a quantum state.

[2] Including quantum computers.

The goal of the protocol is to make sure that the knowledge of Eve about a secret key shared between Alice and Bob is very small. So, with high probability, either Alice and Bob will agree on a key about which Eve knowledge is very small, either Alice and Bob will decide to abort the protocol.

Alice chooses at random a basis among

$$\mathcal{B}_0 = \{|0\rangle, |1\rangle\} \quad \text{and} \quad \mathcal{B}_1 = \left\{ \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right\} .$$

Then she chooses a bit at random. If the bit is 0, she sends the first state of her basis to Bob, if it is 1 she sends the second state of her basis. She iterates the procedure (that is, choosing a random basis and a random bit) N times and sends the N resulting states to Bob.

When receiving the N states, Bob measures them randomly in either $\mathcal{B}_0$ or $\mathcal{B}_1$ and obtain an N bits string (called the *raw key*). Next, Alice and Bob reveal the sending and receiving basis (but not the result they obtained). When both basis coincide, they keep the corresponding bit of their string. When they differ, they discard the corresponding bit. Therefore they obtain a string of n bits ($n \leq N$) that they agree on (called the *sifted key*). Notice that whenever Eve introduces errors, Alice and Bob can notice it easily as their respective sifted key would differ, so that any subsequent communication making use of it to crypt and decrypt would fail[3]. But in the worst possible case, Eve would have succeeded in grabbing say t bits of information on the sifted key without perturbing the quantum transmission[4]. In order to bound Eve's knowledge about their secret key, Alice and Bob can apply a Privacy Amplification scheme [3, 8].

Privacy Amplification will allow Alice and Bob to agree on a secret key, on which Eve will have bounded information. One possibility makes use of universal hashing [7].

Definition 1. *A class $\mathcal{H}$ of functions $\{0,1\}^n \to \{0,1\}^r$ is universal if, for any distinct x_1 and x_2 in $\{0,1\}^n$, the probability that $h(x_1) = h(x_2)$ is at most 2^{-r} when h is chosen uniformly at random from $\mathcal{H}$.*

If we denote x the n bits sifted key Alice and Bob agreed on, and if we consider that Eve has at most t information bits about it, the following result is proven [3]: let $s < n - t$ be a positive safety parameter, and let $r = n - t - s$. If Alice and Bob choose $h(x)$ as their secret key, where h is chosen at random from

[3] As this can seem very crude, we can also consider another scenario where Alice intersperse the key bits with check bits that will later be used to detect errors. If any error is detected, the entire sifted key can be discarded. More about check bits in Sect. 5.

[4] Imagine for example that she chose t times the right measurement basis to measure t different states of the raw key, although the probability of this happening is exponentially low.

a universal class of hash functions from $\{0,1\}^n$ to $\{0,1\}^r$, then Eve's expected information about the secret key $h(x)$, given h and her t bits of information, is less than $2^{-s}/\ln 2$. Alice and Bob therefore compute the hashed value of x in order to obtain a secret key on which Eve has bounded information.

We can formalize the reason why Eve cannot gain information about the states Alice sends on the quantum channel, without taking the risk to perturb the signal. Suppose Alice chooses $\mathcal{B}_0$ to encode the uniformly distributed random bit. The density matrix describing the state Eve has access to can be computed:

$$\rho^{(\mathcal{B}_0)} = \frac{1}{2}\left|0\right\rangle\left\langle 0\right| + \frac{1}{2}\left|1\right\rangle\left\langle 1\right| = \frac{1}{2}\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix} . \tag{1}$$

In the case Alice chooses $\mathcal{B}_1$, we obtain

$$\rho^{(\mathcal{B}_1)} = \frac{1}{4}\begin{pmatrix}1 & 1\\ 1 & 1\end{pmatrix} + \frac{1}{4}\begin{pmatrix}1 & -1\\ -1 & 1\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix} . \tag{2}$$

Thus, because the density matrices for the two basis are equal, Eve cannot measure which basis Alice has chosen to encode her bit. As a consequence she can only choose at random a basis for her own measurement and therefore take the risk to disturb the system. The problem is: what happens on a realistic channel where noise will disturb the states anyway ?

3 Quantum Error Correcting Codes: CSS codes

Quantum error-correcting codes are subspaces of the Hilbert space $\mathbb{C}^{2^n}$ which are protected from errors in a small number of these qubits, so that any such error can be measured and subsequently corrected without disturbing the encoded state[5]. In this section we will make use of the theory of classical linear codes in order to study a large class of quantum error correcting codes known as the Calderbank-Shor-Stean (CSS) codes [6].

The CSS codes will exploit the concept of dual codes. Let $\mathcal{C}_1$ be a $[n, k_1]$ classical linear code, i.e. which uses 2^{k_1} so-called codewords of n bits which can be generated by a $n \times k_1$ generator matrix G_1. Let $\mathcal{C}_2$ be a $[n, k_2]$ subcode of $\mathcal{C}_1$, with $n \times k_2$ generator matrix G_2, such that $k_2 < k_1$ and such that $\mathcal{C}_2 \subset \mathcal{C}_1$. We denote by H_1 and H_2 their respective parity check matrices. We will consider that $\mathcal{C}_1$ and $\mathcal{C}_2^{\perp}$ (the dual code of $\mathcal{C}_2$) can correct up to t errors. Note that as $\mathcal{C}_2$ is a subcode of $\mathcal{C}_1$, the rows of H_1 are spanned by the rows of H_2. The subcode $\mathcal{C}_2$ defines an equivalence relation over $\mathcal{C}_1$. We will consider two codewords $x, y \in \mathcal{C}_1$ to be equivalent whenever there exist some $w \in \mathcal{C}_2$ such that $x = y \oplus w$, where $\oplus$ denotes the bitwise addition modulo 2. There are exactly $|\mathcal{C}_1| / |\mathcal{C}_2| = 2^{k_1 - k_2}$ equivalence classes. The set of all possible equivalences classes (or *cosets*) will

[5] Definition taken from [24].

be denoted $\mathcal{C}_1/\mathcal{C}_2$.

We now define the class of CSS($\mathcal{C}_1$,$\mathcal{C}_2$) quantum codes. Consider the state $|x\rangle$ where $x \in \mathcal{C}_1/\mathcal{C}_2$. The CSS codeword coding $|x\rangle$ is the quantum state

$$|x\rangle \rightarrow \frac{1}{\sqrt{|\mathcal{C}_2|}} \sum_{w \in \mathcal{C}_2} |x \oplus w\rangle \ . \tag{3}$$

One can see that two states $|x\rangle$ and $|y\rangle$ such that x and y are in the same equivalence class are coded by the same CSS codeword and thus, we just defined a $[n, k_1 - k_2]$ quantum correcting code. We will now see that it is able to correct up to t bit-flip errors and t phase-flip errors simultaneously.

Let e_1 (resp. e_2) be the n-bit vector with 1s where bit-flip (resp. phase-flip) errors occurred and 0s elsewhere. We only consider the case where the Hamming weights of e_1 and e_2 are less than t, i.e. less than t errors occurred in both cases. The corrupted state can be represented by

$$\frac{1}{\sqrt{|\mathcal{C}_2|}} \sum_{w \in \mathcal{C}_2} (-1)^{(x \oplus w) \cdot e_2} |x \oplus w \oplus e_1\rangle \ , \tag{4}$$

where $\cdot$ is the inner-dot product. We first try to correct bit-flip errors. We add to our system $n - k_1$ ancillary qubits and compute (using a reversible transformation) the state

$$\frac{1}{\sqrt{|\mathcal{C}_2|}} \sum_{w \in \mathcal{C}_2} (-1)^{(x \oplus w) \cdot e_2} |x \oplus w \oplus e_1\rangle |\mathsf{H}_1(x \oplus w \oplus e_1)\rangle \ . \tag{5}$$

As x and w are two codewords of $\mathcal{C}_1$ (remember that $\mathcal{C}_2 \subset \mathcal{C}_1$),

$$\mathsf{H}_1(x \oplus w \oplus e_1) = \mathsf{H}_1 e_1 \ . \tag{6}$$

The ancillary qubits are just in the state $|\mathsf{H}_1 e_1\rangle$ which is called the bit-flip syndrome. As any two different error syndromes are orthogonal, it is possible to measure the ancillary qubits in order to obtain the value $\mathsf{H}_1 e_1$. Since $\mathcal{C}_1$ is able to correct up to t errors, this knowledge allows us to deduce where qubit flips occurs and by mean of quantum circuit composed entirely of controlled-NOT gates, it is possible to recover the state

$$\frac{1}{\sqrt{|\mathcal{C}_2|}} \sum_{w \in \mathcal{C}_2} (-1)^{(x \oplus w) \cdot e_2} |x \oplus w\rangle \ . \tag{7}$$

Now that we got rid of the bit-flip errors we have to see how one can manage to correct phase flip errors in a similar way. Recalling that Hadamard transform acting on each qubit of an n-qubit state $|x\rangle$ is

$$H^{\otimes n} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{z=0}^{2^n - 1} (-1)^{x \cdot z} |z\rangle \ , \tag{8}$$

applying the Hadamard transform on each qubit of the state (7) gives

$$\frac{1}{\sqrt{2^n\,|\mathcal{C}_2|}}\sum_{w\in\mathcal{C}_2}\sum_{z=0}^{2^n-1}(-1)^{(x\oplus w)\cdot(e_2\oplus z)}\,|z\rangle\ . \tag{9}$$

With $z' = e_2 \oplus z$ we obtain

$$\frac{1}{\sqrt{2^n\,|\mathcal{C}_2|}}\sum_{z'=0}^{2^n-1}\sum_{w\in\mathcal{C}_2}(-1)^{(x\oplus w)\cdot z'}\,|z'\oplus e_2\rangle\ . \tag{10}$$

We can notice that if $z' \in \mathcal{C}_2^{\perp}$ it is, by definition, orthogonal to every $w \in \mathcal{C}_2$ so that in that case $\sum_{w\in\mathcal{C}_2}(-1)^{w\cdot z'} = |\mathcal{C}_2|$. Conversely, if $z' \notin \mathcal{C}_2^{\perp}$, half of the $w\cdot z'$ will be zero so that in that case $\sum_{w\in\mathcal{C}_2}(-1)^{w\cdot z'} = 0$. From this consideration we can reduce (10) to

$$\frac{1}{\sqrt{2^n/\,|\mathcal{C}_2|}}\sum_{z'\in\mathcal{C}_2^{\perp}}(-1)^{x\cdot z'}\,|z'\oplus e_2\rangle\ . \tag{11}$$

Our phase-flip correction problem is now reduced to a bit-flip correction which can be solved using the parity check matrix G_2 of $\mathcal{C}_2^{\perp}$. Once the error e_2 has been removed from (11), applying once again the Hadamard transform to each qubit of the state gives back the original error free state (3).

The CSS code we have just seen is a $[n, k_1 - k_2]$ quantum error correcting code that can correct up to t bit-flip and phase-flip error provided that the underlying linear codes can correct up to t errors.

4 Quantum Key Exchange with CSS codes

We will now take into account the fact that a real channel will inevitably introduce errors into the transmitted states, so that Alice and Bob could mix up the naive action of noise with a detrimental action of Eve. The natural idea would be to find a way for Alice and Bob to correct errors, and thus make use of Quantum Error Correcting Codes. But, as we noticed at the end of Sect. 2, we have to make sure that the state going through the channel cannot be differentiated from the maximally random density matrix $2^{-N}I^{\otimes N}$. To collect information about the key bits, Eve would thus have to entangle her qubits with the encoded state. If Alice and Bob state is protected against errors, i.e. against the entanglement with an outside system, they will prevent Eve from acquiring information about the key.

A solution is to use a set of shifted CSS codes. Alice chooses uniformly at random some $\alpha \in F_2^n/\mathcal{C}_2^{\perp}$ and $\beta \in F_2^n/\mathcal{C}_1$ (using the notation introduced in Sect. 3) and encode a state $|k\rangle$ such that $k \in \mathcal{C}_1/\mathcal{C}_2$ using the parameterized state

$$|k\rangle \rightarrow \frac{1}{\sqrt{|\mathcal{C}_2|}} \sum_{w \in \mathcal{C}_2} (-1)^{\alpha \cdot w} |k \oplus w \oplus \beta\rangle \ . \tag{12}$$

We will denote such a codeword with parameters α and β by $Q_{\alpha,\beta}$. Alice is thus going to encode the random chosen key $|k\rangle$ using the CSS code $Q_{\alpha,\beta}$ which she then sends to Bob (notice that the key is chosen among a set of $2^{k_1-k_2}$ possible keys). As the parameters α and β are chosen uniformly at random, the density matrix ρ of the state available to Eve is the average of all possible encoding of $|k\rangle$, i.e.

$$\rho = \frac{\left|\mathcal{C}_2^{\perp}\right|}{2^n} \frac{|\mathcal{C}_1|}{2^n} \sum_{\substack{\alpha \in F_2^n/\mathcal{C}_2^{\perp} \\ \beta \in F_2^n/\mathcal{C}_1}} \frac{1}{|\mathcal{C}_2|} \sum_{w_1, w_2 \in \mathcal{C}_2} (-1)^{\alpha \cdot (w_1 \oplus w_2)} |k \oplus w_1 \oplus \beta\rangle \langle k \oplus w_2 \oplus \beta| \ .$$

The same arguments we used in Sect. 3 to reduce (10) give

$$\sum_{\alpha \in F_2^n/\mathcal{C}_2^{\perp}} (-1)^{\alpha \cdot (w_1 \oplus w_2)} = \begin{cases} \frac{2^n}{|\mathcal{C}_2^{\perp}|} & \text{if } w_1 = w_2 \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

From the last two equations, we deduce:

$$\rho = \frac{|\mathcal{C}_1|}{2^n |\mathcal{C}_2|} \sum_{\beta \in F_2^n/\mathcal{C}_1} \sum_{w \in \mathcal{C}_2} |k \oplus w \oplus \beta\rangle \langle k \oplus w \oplus \beta| \ . \tag{14}$$

As Eve doesn't have any a priori on $k \in \mathcal{C}_1/\mathcal{C}_2$ as it is chosen at random by Alice, ρ finally is the average over all possible keys that is

$$\rho = \frac{1}{2^n} \sum_{k \in \mathcal{C}_1/\mathcal{C}_2} \sum_{\beta \in F_2^n/\mathcal{C}_1} \sum_{w \in \mathcal{C}_2} |k \oplus w \oplus \beta\rangle \langle k \oplus w \oplus \beta| \tag{15}$$

$$= \frac{1}{2^n} I^{\otimes n} \ . \tag{16}$$

As wanted, the state available to Eve is indistinguishable from the maximally random density matrix. The protocol then succeeds if Bob can recover the key. Once he has received the codeword $Q_{\alpha,\beta}$ he announces it to Alice who the reveals the parameters α and β. With this knowledge, Bob can reduce $Q_{\alpha,\beta}$ to the classical CSS code that encodes the key k. As k was encoded by a CSS code, up to t errors can be corrected, so that the encoded state $|k\rangle$ is delivered to Bob with high fidelity. Any external state (including any state in Eve possession) is thus disentangled from Bob state. Therefore the key is secure.

We can note that this protocol works, provided that the error rate (including errors caused by Eve) is low enough, so that CSS codes can perform their function. A solution to detect a too high error rate would be the following. Just

before sending $Q_{\alpha,\beta}$ on the quantum channel, Alice can interspersing it with check bits at random positions. After Bob has received the state, she reveals the positions and values of these check bits to him so that he can compute the error rate. If it is too high, the whole state is discarded and they abort the protocol. If it is low enough, Bob knows that the CSS codes will fulfill their mission, and only the check bits are discarded. Note that as the check bit are random and at random positions, the density matrix of the state available to Eve is still indistinguishable from the maximally random density matrix.

The drawback of this secure protocol is that it is impractical. When Bob receives the quantum state $Q_{\alpha,\beta}$ he has to store it until Alice reveals the values of α and β. To achieve this, Bob must have access to a quantum computer (or a quantum memory) which is not feasible with actual technology. The advantage of the BB84 protocol is that Bob can measure the state individually as soon as they reach him, so that he does not need to have access to a quantum memory. We shall see why the security of the QKE protocol using CSS codes implies the security of the BB84 protocol over noisy channels in the next section.

5 Security Proof of BB84 over Noisy Channels

We describe the BB84 protocol over noisy channels [24] (which slightly differ from the BB84 protocol we saw in Sect. 2) on Figure 1, using the notations of Sect. 2.

1. Alice creates $(4+\delta)n$ random bits.
2. Alice chooses a random $(4+\delta)n$-bit string b. For each bit, she creates a state in the $\mathcal{B}_0$ basis (when the corresponding bit of b is 0) or in the $\mathcal{B}_1$ basis (when the corresponding bit of b is 1).
3. Alice sends the resulting qubits to Bob.
4. Bob receives the $(4+\delta)n$ qubits, measuring each in $\mathcal{B}_0$ or $\mathcal{B}_1$ at random.
5. Alice announces b.
6. Bob discard any result where his basis doesn't coincide with Alice's one. With high probability, there are at least $2n$ bits left (if not, abort the protocol). Alice decides randomly on a set of $2n$ bits to use for the protocol, and chooses at random n of these to be check bits.
7. Alice and Bob announce the values of their check bits. If too few of these value agree (high error rate), they abort the protocol.
8. Alice announces $u \oplus v$, where v is the string consisting of the remaining non-check bits, and u is a random codeword in $\mathcal{C}_1$.
9. Bob subtracts $u \oplus v$ from his own remaining non-check bits $v \oplus \epsilon$ (where ϵ represents errors), and corrects the result $u \oplus \epsilon$ in order to obtain u, a codeword in $\mathcal{C}_1$.
10. Alice and Bob use the coset of u in $\mathcal{C}_1/\mathcal{C}_2$ as the secret key.

Fig. 1. BB84 protocol over noisy channels

Looking back at the QKE protocol with CSS codes, we can see that the only things Bob cares about are the encoded bits of the key value. Consider the case where Alice never sends the value of α. Can Bob still decode the states he receives and deduce the key bits ? If Bob doesn't receive the value of α, we can consider that the density matrix describing the state he receives from Alice is the average of $Q_{\alpha,\beta}$ over all possible α, i.e.

$$\begin{aligned}\rho_{\text{Bob}} &= \frac{|\mathcal{C}_2^\perp|}{2^n} \sum_{\alpha \in F_2^n/\mathcal{C}_2^\perp} \frac{1}{|\mathcal{C}_2|} \sum_{w_1,w_2 \in \mathcal{C}_2} (-1)^{\alpha\cdot(w_1 \oplus w_2)} \left|k \oplus w_1 \oplus \beta\right\rangle \left\langle k \oplus w_2 \oplus \beta\right| \\ &= \frac{1}{|\mathcal{C}_2|} \sum_{w \in \mathcal{C}_2} \left|k \oplus w \oplus \beta\right\rangle \left\langle k \oplus w \oplus \beta\right| \ , \end{aligned} \tag{17}$$

using (13). We see that all the information Bob cares about (namely the bits k) is available in the density matrix ρ_{Bob} so that Bob does not need the value of α. The state ρ_{Bob} can be equivalently seen as a mixture of states $|k \oplus w \oplus \beta\rangle$ where $w \in \mathcal{C}_2$ would have been chosen uniformly at random. We can thus consider that Alice has sent the state $|k \oplus w \oplus \beta\rangle$ with w chosen randomly in $\mathcal{C}_2$. After a measurement, Bob recovers the corrupted bits $k \oplus w \oplus \beta \oplus \epsilon$. Alice reveals the value of β, which Bob subtracts from his string of bits in order to obtain $k \oplus w \oplus \epsilon$. Bob knows that $k \oplus w \in \mathcal{C}_1$, so that he can correct (with high probability) the result in order to recover the value $k \oplus w$. The equivalence class of $k \oplus w$ corresponds to the secret key.

The reasoning of the last paragraph is just another way of looking a the QKE protocol with CSS codes. But we shall see why this new point of view has the advantage to prove the security of the BB84 protocol presented on Figure 1. In the BB84 protocol, after having discarded the check bits, Bob remains with the bits $v \oplus \epsilon$. In the CSS protocol Alice would reveal β, in the BB84 protocol she reveals $u \oplus v$, which is uniformly distributed over F_2^n (because v is uniformly distributed over F_2^n). Bob can deduce the value of $u \oplus \epsilon$ (just as he would have deduced the value of $k \oplus w \oplus \epsilon$ in the CSS protocol) and, as $u \in \mathcal{C}_1$, he can recover u. Alice and Bob finally use the equivalence class of u as a key (just as they would have used the equivalence class of $k \oplus w$ in the CSS protocol).

We conclude that the QKE with CSS codes and the BB84 protocol over noisy channels are completely equivalent, so that they are both secure.

6 Conclusion

In this paper, we have gone through the proof of the security of the BB84 protocol over noisy channels [24] proposed by P.W. Shor and J. Preskill. As they already notice in the original paper, the proof has a few loose ends like for example the fact that it does not meet completely the reality of experiment. In particular, the proof does not take into account the existence of imperfect sources (although this issue wasn't taken into account in earlier proofs [4, 14] either).

Acknowledgments

I would like to emphasize the fact that I made extensive use of several existing articles [24, 8, 13], presentation slides [23, 19], course notes [18, 15] and books [17, 5]. I hope I didn't introduce too much noise in the great information they provide. Finally, I would like to thank Prof. Erdal Arikan for having proposed this work to me.

References

[1] M. Ardehali, H.F. Chau, and H. Lo. Efficient Quantum key distribution. Available on http://arxiv.org/abs/quant-ph/9803007.

[2] 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*, IEEE Press, pages 175–179, Bangalore, India, 1984.

[3] C.H. Bennett, G. Brassard, C. Crépeau, and U.M. Maurer. Generalized privacy amplification. In *IEEE Trans. Information th.*, volume 41, pages 1915–1923, 1995.

[4] E. Biham, M. Boyer, P.O. Boykin, T. Mor, and V. Roychowdhury. A Proof of the Security of Quantum Key Distribution. In *Proceedings of the thirty-second annual ACM Symposium on Theory of Computing*, 1999. Available on http://arxiv.org/abs/quant-ph/9912053.

[5] D. Bouwmeester, A. Ekert, and A. Zeilinger, editors. *The Physics of Quantum Information*. Springer-Verlag, 2000.

[6] A.R. Calderbank and P.W. Shor. Good quantum error-correcting codes exist. 1995. Available on http://xxx.lanl.gov/abs/quant-ph/9512032.

[7] J.L. Carter and M.N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18:143–154, 1979.

[8] D. Deutsch, A. Ekert, R. Jozsa, C. Macchiavello, S. Popescu, and A. Sanpera. Quantum privacy amplification and the security of quantum cryptography over noisy channels. *Phys. Rev. Lett.*, 77:2818–2821, 1996. Available on http://arxiv.org/abs/quant-ph/9604039.

[9] W. Diffie and M.E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22:644–654, 1976.

[10] T. ElGamal. A public key cryptosystem and signature scheme based on discrete logarithms. In *Advances in Cryptology - Proceedings of CRYPTO'84*, volume 196, pages 10–18, 1985.

[11] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden. Quantum cryptography. 2003. Submitted to Reviews of Modern Physics.

[12] A.K. Lenstra, H.W. Lenstra Jr., M.S. Manasse, and J.M. Pollard. The number field sieve. In A.K. Lenstra and Lenstra Jr., editors, *The Development of the Number Field Sieve*, volume 1554 of Lectures Notes in Mathematics, pages 11–42. Springer-Verlag, 1993.

[13] H. Lo and H.F. Chau. Unconditional Security of Quantum Key Distribution over Arbitrarily Long Distances. *Science*, 283:2050–2056, 1999. Available on http://arxiv.org/abs/quant-ph/9803006.

[14] D. Mayers. Unconditional security in Quantum Cryptography. In *J. Assoc. Computing Machinery*, 1998. Available on http://arxiv.org/abs/quant-ph/9802025.

[15] Z. Meglicki. *Introduction to Quantum Computing*, 2002. Available on http://beige.ucs.indiana.edu/B679/.

[16] A.J. Menezes, P.C. Van Oorschot, and S.A. Vanstone. *Handbook of applied cryptography*. The CRC Press series on discrete mathematics and its applications. CRC-Press, 1997.

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

[18] J. Preskill. *Lecture Notes of the Quantum Computation course*. Caltech (California Institute of Technology), 2000-2001. Available on http://www.theory.caltech.edu/people/preskill/ph229/.

[19] J. Preskill. *Presentation Slides : Quantum error-correcting codes and quantum cryptography*. Caltech (California Institute of Technology), June 2003. Available on http://www.quantum-ucalgary.org/sschool/qis-school/encodings.html.

[20] M.O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. MIT/LCS/TR-212,MIT Laboratory for Computer Science, 1979.

[21] R.L. Rivest, A. Shamir, and L.M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21:120–126, 1978.

[22] P.W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In S. Goldwasser, editor, *Proceedings of the 35th Symposium on Fundations of Computer Science*, IEEE Computer Society Press, pages 124–134. Los Alamitos, 1994.

[23] P.W. Shor. *Presentation Slides : Quantum Error Correcting Codes and Quantum Cryptography*. AT&T Labs, Florham Park, NJ, 2001. Available on http://cm.bell-labs.com/cm/ms/events/DIMACSBL/viewgrap.html.

[24] P.W. Shor and J. Preskill. Simple Proof of Security of the BB84 Quantum Key Distribution Protocol. *Phys. Rev. Lett.*, 85:441–444, 2000. Available on http://arxiv.org/abs/quant-ph/0003004.