G53NSC and G54NSC
Non Standard Computation
Research Presentations

March the 23rd and 30th, 2010

Tuesday the 23rd of March, 2010

- 11:00 - James Barratt
  Quantum error correction
- 11:30 - Adam Christopher Dunkley and Domanic Nathan Curtis Smith-Jones
  One-Way quantum computation and the Measurement calculus
- 12:00 - Jack Ewing and Dean Bowler
  Physical realisations of quantum computers

Tuesday the 30th of March, 2010

- 11:00 - Jiri Kremser and Ondrej Bozek Quantum cellular automaton
- 11:30 - Andrew Paul Sharkey and Richard Stokes Entropy and Information
- 12:00 - Daniel Nicholas Kiss Quantum cryptography

QUANTUM ERROR CORRECTION

JAMES BARRATT

Abstract. Quantum error correction is currently considered to be an extremely important area of quantum computing as any physically realisable quantum computer will need to contend with the issues of decoherence and other quantum noise. A number of techniques have been developed that provide some protection against these problems, which will be discussed.

1. Introduction

It has been realised that the quantum mechanical behaviour of matter at the atomic and subatomic scale may be used to speed up certain computations. This is mainly due to the fact that according to the laws of quantum mechanics particles can exist in a superposition of classical states. A single bit of information can be modelled in a number of ways by particles at this scale. This leads to the notion of a qubit (quantum bit), which is the quantum analogue of a classical bit, that can exist in the states 0, 1 or a superposition of the two. A number of quantum algorithms have been invented that provide considerable improvement on their best known classical counterparts, providing the impetus to build a quantum computer.

Although theoretically possible, it is not clear whether building a sizeable quantum computer is feasible. One difficulty is the issue of decoherence, which occurs when a quantum system interacts with its environment, causing its wave function to collapse into one of its base states. It is unreasonable to believe that a quantum system can be kept totally isolated from its surroundings and therefore decoherence should be considered unavoidable. However, that is not to say that its effects on the system cannot be reduced.

Quantum error correction is concerned with the development of techniques that allow a quantum computer to recover from errors that are introduced by decoherence and other quantum noise. Some of the techniques that have so far been developed are inspired by techniques taken from coding theory that are already used within classical computing. The basic idea is to encode information in such a way that the existence of errors can be detected and, hopefully, corrected.

Date: 19th March 2010.

2. Classical error correction

Some techniques of classical error correction are briefly described as quantum error correction builds upon such ideas.

2.1. **Parity check.** A parity check involves adding an extra bit to some data to signify whether the number of 1s, or conversely the number of 0s, in the data is odd or even. For example, given the data 010 a parity bit would be added to signify that the data contains an odd number of 1s. In this case the parity bit is set to 1 if odd and 0 if even.

$010 \rightarrow \mathbf{1}010$

This technique will only detect an odd number of errors, since if and even number of bits were to flip the parity would be the same. It also fails to identify where errors have occurred, providing no way of correcting them. If this data had been send over a communication channel it would have to be resent.

2.2. **The repetition code.** The repetition code makes use of redundancy by taking each bit of data and repeating it a number of times. For example,

$010 \rightarrow 000111000$

Upon receiving this data each group of three bits is decoded. It is assumed that errors are independent, so the likelihood of one bit flip occurring is higher than that of two or more bit flips occurring. Therefore, if a group of three bits is found to disagree a majority vote is taken since it is more likely that the two that agree hold the correct value.

From the example above, if a bit flip occurs on the first bit then the first group of three bits will be 100. By taking a majority vote we deduce that the original group was 000 and flip the first bit back to 0. It is possible that the second and third qubits had flipped, in which case we would be decoding to the wrong value. However, as long as the probability of this occurring is lower than the probability of receiving incorrect data when error correction is not employed there is an improvement.

2.3. **Hamming codes.** Hamming codes are a class of linear error correction codes that make use of several overlapping parity bits [7]. A $[n, k, d]$ code is defined by a $n \times k$ generator matrix G or equivalently by a $(n-k) \times n$ parity check matrix H, where n denotes the length of the codewords generated, k is the length of the data encoded and d is the minimum hamming distance between any two codewords.

The set of all possible codewords corresponds to the linear combinations of the column vectors of G. So that data is uniquely encoded to a particular codeword the columns should be linearly independent. Alternatively, the code is defined as the set of n-element vectors x where $Hx = 0$. It may be noted that the columns of this vector contain all binary values of length $n - k$, excluding the zero vector, which will indicate the absence of any errors.

A code can only identify and correct $< d/2$ errors [3]. It is most likely that a corrupt word was reached from the codeword requiring the least number of flips. By insisting on this limit there will always be a closest codeword to recover to.

The encoded form c of some data a is obtained by applying the generator matrix to the column vector containing a.

$c = Ga$

From the definition of the parity check matrix, if the encoding is received unaffected by errors then $Hc = 0$ as it will be a member of the set of valid codewords. However, if some error e has affected the codeword then $c' = c \oplus e$ will be received, where e is a vector containing 1s to signify where a bit will be flipped. It follows that $Hc' = He$, where He will be found as a column in the parity check matrix. This column will identify which bit has been flipped, for example, if it is found in the fifth column, then it is the fifth bit that has been flipped. For a worked example see appendix.

3. Quantum error correction

Error correction techniques specific to quantum computing are now discussed. The basic process follows that of classical error correction in that data is encoded so that the presence of certain errors may be detected and corrected.

3.1. **The bit flip code.**

The bit flip code is based on the repetition code described above, which measured and repeated each bit in order to add redundancy. But this copying of data is not possible when dealing with qubits due to the no-cloning theorem [6]. As soon as a qubit is measured it's wave function collapses into one of its base states. However, the state of a qubit can be shared among a number of other qubits. For example, given a qubit in the state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ the following entanglement can be created with two CNOT gates.

$|\psi\rangle = \alpha|000\rangle + \beta|111\rangle$

These qubits are said to be entangled as it is no longer possible to descirbe the state of one individual qubit without also describing the state of the others. In the three qubit bit flip code an individual qubit is encoded in just this way. The code is similar to the repetition code in instances where there has been more than one bit flip. As a majority vote is taken it is possible for the recovery to obtain the wrong value.

If, while travelling through a communication channel, the first bit of the encoding were to become flipped the state received would be $|\psi'\rangle = \alpha|100\rangle + \beta|011\rangle$. In order to detect whether an error is present and which bit has been flipped a process known as syndrome diagnosis is performed. For each case there is a projection operator, which will allow us to determine whether that case applies to the state we have received. These projection operators are defined as:

$P_0 \equiv |000\rangle\langle 000| + |111\rangle\langle 111|$
$P_1 \equiv |100\rangle\langle 100| + |100\rangle\langle 100|$
$P_2 \equiv |010\rangle\langle 010| + |010\rangle\langle 010|$
$P_3 \equiv |001\rangle\langle 001| + |110\rangle\langle 110|$

The subscripts of the operators indicate which qubit is identified as being corrupt if its diagnosis is positive. The first corresponds to the case where no error is present as all qubits agree. This notation of a *ket* followed by a *bra*, such as $|x\rangle\langle y|$, refers to the outer product, giving a matrix, and the projection operator is applied by matrix multiplication.

The relevant case is identified by taking the expectation value (the statistical mean value) of each projection operator, which for our corrupted state $|\psi'\rangle$ will be:

$\langle\psi'|P_0\psi'\rangle = 0$
$\langle\psi'|P_1\psi'\rangle = 1$
$\langle\psi'|P_2\psi'\rangle = 0$
$\langle\psi'|P_3\psi'\rangle = 0$

The above notation of a *bra* followed by a *ket*, such as $\langle x|y\rangle$, denotes the inner product. From the above expectation values it is taken that an error has affected the first qubit. All that remains is to correct the error by applying a Pauli X gate to the appropriate qubit, hopefully recovering the original state. Notice that syndrome diagnosis contains no information about the state of the qubits, specifically, about the amplitudes of the states in superposition. No measurement has been made.

See appendix for an example of how these projection operators are applied and the expectation values obtained.

3.2. **The phase flip code.** Whereas in classical computing the only errors on bits are bit flips, in quantum computing qubits may also be corrupted by a flip of their relative phase. For example, a qubit in the state $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ may have it's relative phase flipped to give the state $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.

This turns out to be equivalent to the bit flip code from the perspective of a new computational basis [4]. If the base states are taken to be $|+\rangle$ and $|-\rangle$ then a phase flip acts just like a bit flip with respect to this basis. The operators of the bit flip code can in fact be reused here by simply applying Hadamard gates at various points to switch between computational basses.

A qubit in the state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ will be encoded as $|\psi'\rangle = \alpha|+++\rangle + \beta|---\rangle$ by using two CNOT gates and then applying a Hadamard gate to each qubit.

3.3. **Shor's code.** An obvious problem with the previous codes is the fact that each can only correct for one kind of error, whereas in reality all sorts of errors may be encountered at any time. The Shor code combines these codes into one that can check for the presence of both kinds of error [1]. In fact, the Shor code can correct arbitrary single qubit errors

[4]. It uses a total of nine qubits to encode the state of a single qubit as it first encodes using one code, then encodes this encoding again with the other.

Firstly, the qubit is encoded using the phase flip code.

$|0\rangle \rightarrow |+++\rangle$
$|1\rangle \rightarrow |---\rangle$

Each of these three qubits is then encoded using the bit flip code.

$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \rightarrow \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$
$|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \rightarrow \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle)$

So the overall encoding procedure results in the following:

$|0\rangle \rightarrow \frac{(|000\rangle+|111\rangle)(|000\rangle+|111\rangle)(|000\rangle+|111\rangle)}{2\sqrt{2}}$
$|1\rangle \rightarrow \frac{(|000\rangle-|111\rangle)(|000\rangle-|111\rangle)(|000\rangle-|111\rangle)}{2\sqrt{2}}$

As an example the state $|\psi\rangle = |0\rangle$ may be encoded and transmitted but subject to both a bit flip and a phase flip giving

$|\psi'\rangle = \frac{(|100\rangle-|011\rangle)(|000\rangle+|111\rangle)(|000\rangle+|111\rangle)}{2\sqrt{2}}$

In order to determine whether a bit flip has occurred the syndrome diagnosis is performed on each group of three qubits. The fact that a group may have had its phase flipped does not affect this process. If a bit flip is detected it is corrected in the usual manner.

When the phase of a qubit is flipped the phase of the group is flipped as can be seen above. To detect this the syndrome diagnosis is performed on the three groups of qubits as opposed to individual qubits. If the sign of a group is found to disagree with the other two then it's sign is flipped to recover the original state.

3.4. **Steane's code.** The Steane code is an example of a wider class of codes known as CSS codes and is based on the [7,3,4] simplex code, referred to as C from here onwards, and its dual $C^{\perp}$, the [7,4,3] Hamming code, an example of which is found in the appendices.

Given a code C with a generation matrix G and a parity check matrix H, the dual of C is defined as the code $C^{\perp}$ with generation matrix H^T and parity check matrix G^T. It turns out that $C^{\perp}$ contains all codewords that are orthogonal to all codewords in C [4].

It should be noted that *if the state of a system forms a linear code C in superposition with equal coefficients within the first computational basis, then the words of the superposition from within the second basis will be those of the dual code $C^{\perp}$* [2]. The first computational basis uses the base states $|0\rangle$ and $|1\rangle$ whereas the second basis consists of $|+\rangle$ and $|-\rangle$. So we can check for the presence of both bit flip and phase flip errors using these linear codes.

The encoding is based on the generation matrix of the simplex code [3]. A qubit in the state $|0\rangle$ is encoded as an equal superposition of all the codewords contained in C. A qubit in the state $|1\rangle$ is encoded as an equal superposition of all the codewords of C with a Pauli X gate applied to each qubit.

Again, the set of codewords in the simplex code is obtained by taking the linear combination of each column in its generation matrix, so the encoded state of $|0\rangle$ and $|1\rangle$ will be as follows.

$|0\rangle \rightarrow |C\rangle = \frac{1}{\sqrt{8}}(|0000000\rangle + |1010101\rangle + |1100110\rangle + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle)$

$|1\rangle \rightarrow |C^{\perp}\rangle = \frac{1}{\sqrt{8}}(|1111111\rangle + |0101010\rangle + |0011001\rangle + |1110000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle)$

An arbitrary qubit with state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ will be encoded as $|\psi\rangle = \alpha|C\rangle + \beta|C^{\perp}\rangle$.

In order to determine if a bit flip has affected a qubit the same approach is taken as in classical Hamming codes. The parity check matrix H of C is used to perform a diagnosis, and if the result is anything other than the zero vector then a NOT gate is applied to the appropriate qubit. However, a phase flip can also be detected by applying the parity check matrix of $C^{\perp}$. Recall that when a system is in equal superposition of the codewords of one code, in the other basis the codewords in superposition are of the dual code.

4. Further reading

Further reading on this subject may include the five qubit error correction code found by Raymond Laflamme. Although it requires less qubits it is thought that the Steane code is more practical [5]. There is also the topic of stabilizer codes, which are a general class of codes analogous to classical linear codes, of which CSS codes are a subclass.

5. References

(1) P. Shor, Phys. Rev. A 52, 4, Scheme for reducing decoherence in quanum computer memory (1995)

(2) A. Steane, Phys. Rev 77, 5, Error Correcting Codes in Quantum Theory (1996)

(3) A. Steane, Multiple-Particle Interference and Quantum Error Correction (1995)

(4) M. Nielsen, I. Chuang, Quantum Computation and Quantum Information, Cambridge (2002)

(5) N. D. Mermin, Quantum Computer Science, Cambridge (2007)

(6) Quantum error correction, www.wikipedia.org [March 2010]

(7) Hamming code, www.wikipedia.org [March 2010]

6. Appendices

6.1. **Hamming code example.** As an example the Hamming [7,4,3] code is taken.

The matrices are defined as follows:

$$G \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

$$H \equiv \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

The encoded form c of some data $a = 1011$ is obtained by applying the generator matrix.

$$c = Ga = G \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

In the case where no errors have occurred $r = c$.

$$Hr = H \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Whereas some error e may affect c so that $r = c \oplus e$.

$$r = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

As a result the parity check matrix will give a non zero vector.

$$Hr = H \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

By comparing this vector to the columns vectors of the parity check matrix, the corrupt bit can be identified. As Hr is equal to the second column, it was the second bit that was subject to the flip.

6.2. **Bit flip code example.** Having received two qubits, the first of which has been flipped in communication, in the state $|\psi\rangle = \alpha|10\rangle + \beta|01\rangle$, syndrome diagnosis must be performed to detect any errors. The first two projection operators are defined as:

$P_0 \equiv |00\rangle\langle 00| + |11\rangle\langle 11|$
$P_1 \equiv |10\rangle\langle 10| + |01\rangle\langle 01|$

The outer product is obtained by ab^T, where a and b are column vectors. So,

$$P_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Once the projection operator is applied the expectation value must be taken. The inner product is defined as $|a\rangle^\dagger|a\rangle$, where † denotes the adjoint (transverse and conjugate).

$$|\psi'\rangle = \begin{bmatrix} 0 \\ 0 \\ \alpha \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \beta \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \beta \\ \alpha \\ 0 \end{bmatrix}$$

The first projection operator tells us that an error has occurred.

$$\langle\psi'|P_0\psi'\rangle = \begin{bmatrix} 0 & \bar{\beta} & \bar{\alpha} & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = 0$$

The second tells us that a bit flip occurred on the first bit.

$$\langle\psi'|P_0\psi'\rangle = \begin{bmatrix} 0 & \bar{\beta} & \bar{\alpha} & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = |\alpha|^2 + |\beta|^2 = 1$$

The remaining projection operators have been omitted.

One-Way Quantum Computation and the Measurement Calculus

Adam Christopher Dunkley ⟨acd07u@cs.nott.ac.uk⟩
Domanic Nathan Curtis Smith-Jones ⟨dns07u@cs.nott.ac.uk⟩

March 19, 2010

Contents

1 Introduction

Quantum circuits form a powerful and familiar framework for developing and reasoning about quantum computation (§2). Such circuits are unitary, and thus are not subject to decoherence . Recently a method has been described [RB01], utilizing a grid of entangled states and the inherent properties of measurement, to model computation that looks to improve the complexity of quantum computation[DKP07]. In this measurement based computation all basic operations are non-unitary. Striking at the core of our understanding of quantum mechanics, these measurement based computations can simulate any arbitrary quantum operation – even those that are unitary.

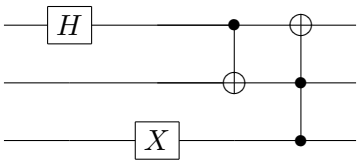
In this report we will describe quantum circuits, and how their unitary nature strives to avoid decoherence, before introducing cluster states as a method for quantum computation, in which one-way measurements form computation.

2 Quantum Circuits Model

A quantum circuit is a model of quantum computation within which a set of quantum gates are applied which translate to reversible transformations of qubits which are the unit of data in this system.

A qubit is the unit of quantum information and is analogous to the classical bit. The key difference is that rather than just 2 states 0 or 1, a qubit is able to have a state which is a superposition of the two base states of $|0\rangle$ or $|1\rangle$. This means it has the possibility to have an unlimited number of different states. This is symbolised by $|\psi\rangle$ where α and β are the probabilities of observing either $|0\rangle$ or $|1\rangle$, leading to the definition that $|\alpha|^2 + |\beta|^2 = 1$

The quantum circuit model is able to depict the quantum computer as a circuit representing various controls and calculations that can be completed.



The horizontal left to right lines show the progression of a qubit through time. The boxes containing letters signify different computations and the vertical lines headed by a black dot show a controlled gate. Each qubit is initialised at the left to ket 1 or 0. The calculations are then performed, and the qubits measured at the conclusion of the program.

Quantum circuits are reversible and assume a perfectly insulated system. This allows no room for decoherence and thus (in theory) should enable more efficient computations to be made. In order for this to take place, you must be able to keep track of all qubits that are included in the initial register. As any that are lost mean that you will be unable to reverse the computation. For example, the classical OR gate producing an output of 1, would not lend itself to reproducing the inputs, as combinations of 01, 10 and 11 would all produce this output.

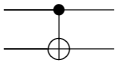
Processing in these circuits is performed using quantum gates that can take various numbers of inputs.

An example of a one-bit gate is the Hadamard Gate:

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

This takes one qubit and performs the defined transformation.

An example of a 2-bit gate that flips the target qubit if the control (top) qubit is $|1\rangle$:



2.1 The Physical Circuit

The problem with the quantum circuit model is its perfection and simplicity. The very fact that it is totally reversible assume a totally isolates system in which the circuit exists. In simulations it is possible to do this, but when looking to build a physical quantum computer, this entirely isolated coherent system is impractical to attempt to construct. Although in a theoretical circuit reversible gates can be assumed, in a physical implementation it is difficult to stop qubits interacting with the environment.

2.2 Decoherence

This is a loss of information and is what quantum circuits strive to avoid, given their unitary nature. Quantum circuits in contrast to this are coherent and rely on this to complete computations. The constant phase of these systems is destroyed when a proper measurement of a qubit is made.

3 Cluster State Model

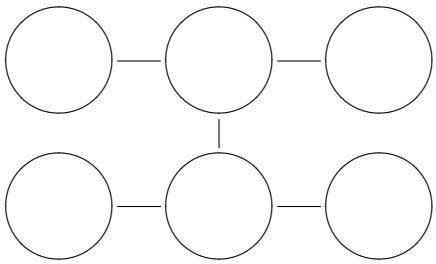
Where the quantum circuits model represents quantum computations as a set of unitaries, the cluster state model is a set of entangled states that, when measured, act as such a computation.

3.1 Functioning

A cluster state measurement comprises of entangling a set of many quantum states (the cluster state) followed by a series of single-qubit measurements (the processing), all remaining states (i.e. those not collapsed by measurements), when read out, comprise the result of the computation.

3.1.1 Initialisation

In detail, a cluster state is a set of n quantum states – whose underlying structure can be represented as a matrix of n vertices



The recipe for constructing a cluster state is first to prepare all non-input qubits as state $|+\rangle \equiv (|0\rangle + |1\rangle)/\sqrt{2}$, which can be derived from state $|0\rangle$ by applying the

Hadamard gate, before connecting qubits along vertices and corresponding points across vertices by applying the controlled-phase gate (controlled Pauli-Z).

We can model the above 6 qubit cluster in QIO[1] as follows; we introduce our cluster's type as a matrix of qubits:

```
type Cluster = [[Qbit]]
```

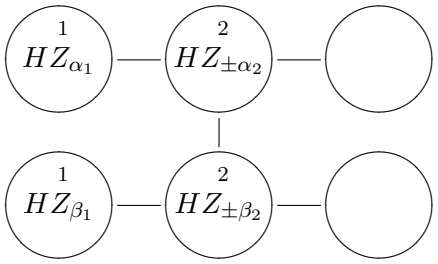
and define our cluster:

```
cluster :: QIO Cluster
cluster = do
  p0 ← plus
  p1 ← plus
  p2 ← plus
  p3 ← plus
  p4 ← plus
  p5 ← plus
  applyU $ controlledZ p0 p1
  applyU $ controlledZ p1 p2
  applyU $ controlledZ p3 p4
  applyU $ controlledZ p4 p5
  applyU $ controlledZ p1 p4
  return $ (p0 : p1 : p2 : []) : (p3 : p4 : p5 : []) : []
```

It can be noted that the controlled-phase gate is commutative – it is merely stylistic that we've adopted to fill in along the vertices before filling in vertex spanning connections.

3.1.2 Computation

To illustrate, we could have a (somewhat contrived) cluster-state computation such that:



To compute this, we perform a rotation (given by a unitary HZ_α) setting the basis for the proceeding measurement. We then feed forward the information of this measurement to compute the next state in the cluster. The order of computation is dictated by the number at the top of each node, where numbers that are the same can be computed

[1] QIO is a library for simulating quantum computations in Haskell

in any order. In detail, we apply a Hadamard followed by the exponentiated Pauli-Z operations (with some α or β, which are real numbers) before measuring the state and carrying forward the classical bit. The $\pm$ indicates the sign based on the calculation fed forward, where $0 \rightarrow |+\alpha\rangle$ and $1 \rightarrow |-\alpha\rangle$.

We can represent the measurement of such a cluster in QIO:

```
measureCluster :: Float → Float → Cluster → QIO [Qbit]
measureCluster a b ((p0 : p1 : p2 : []) :
  (p3 : p4 : p5 : []) : []) = do
    applyU $ uExpZ p0 a
    b0 ← measQbit p0
    applyU $ uExpZ p3 b
    b2 ← measQbit p3
    applyU $ uExpZ p1 (applySign b0 a)
    b1 ← measQbit p1
    applyU $ uExpZ p4 (applySign b2 b)
    b3 ← measQbit p4
    return $ p2 : p5 : []
```

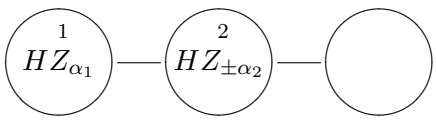
As you can see, as we go through the computation we measure the qubits, having the effect of collapsing them into their base states, these measurements are then expressed as signs for θ in the Z_θ operation on the subsequent measurement. Above, the result of our computation are the two initially "blank" qubits. Another result that may be useful is the set of base states computed from the measurements ($b0$ to $b3$).

3.2 Example

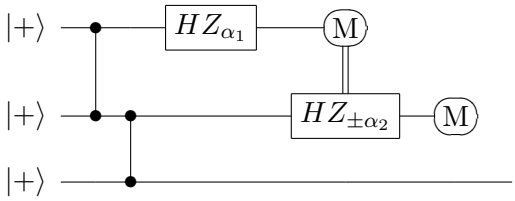
We can simulate quantum circuits in the cluster-state model. Here, using examples from The Measurement Calculus[DKP07], which provides a calculus for describing and reasoning about cluster-states, and continuing with examples from Cluster-State Quantum Computation[Nie06] we introduce some circuits, giving their equivalent quantum circuits and their representation in QIO.

3.2.1 Teleportation

Teleportation is a simple 3 qubit cluster, which closely resembles its quantum circuit counterpart.



We can represent this as the following quantum circuit:



It should be intuitive that we can model this in QIO:

```
teleport :: Float → Qbit → QIO Qbit
teleport a q = do
  p0 ← plus
  p1 ← plus
  applyU $ controlledZ q p0
  applyU $ controlledZ p0 p1
  b0 ← opM q a
  b1 ← cOpM b0 a p0
  applyU $ cControlledX b0 q
  applyU $ cControlledZ b1 p0
  return $ p1
```

Here, as before, we set up our non-input states as plus states before entangling the vertices. We then run our measurements. These two measurements correspond closely to that of the measurement operations given in the measurement calculus, where *opM* moves the qubit in to the computational before performing the measurement and *cOpM* applies a sign to alpha based on the previous computation.

References

[DKP07] Vincent Danos, E. Kashefi, and P. Panangaden. The measurement calculus. *Journal of the ACM (JACM)*, 54(2):8, 2007.

[Nie06] M.A. Nielsen. Cluster-state quantum computation. *Reports on Mathematical Physics*, 57(1):147–161, 2006.

[RB01] Robert Raussendorf and Hans J. Briegel. A one-way quantum computer. *Phys. Rev. Lett.*, 86(22):5188–5191, May 2001.

Physical Realisations of Quantum Computers

While quantum computers have been a major topic for scientists and engineers around the world for a number of years now, there appears to have been no move whatsoever towards quantum computers being commercially or even widely available. The power offered by a quantum computer could easily surpass that of a classical computer if there were any simple way to produce quantum computers capable of manipulating anything more than a small number of Qubits. It is believed that if we could produce a computer with as little as one hundred and fifty qubits then it would be more powerful than every supercomputer in the world combined.

The advances made of late in quantum computing are significant, but when comparing its progress to the time line of classical computers it is relatively clear that quantum computing is still in the very early stages. There are a number of viable options for a method of implementing a quantum computer, all based around a set of rules given for what a quantum computer should be.

The requirements for a practical quantum computer, as listed by staff at IBM are:

- qubits can be initialized to arbitrary values
- quantum gates faster than decoherence time
- universal gate set
- qubits can be read easily
- scalable physically to increase the number of qubits.

It is worth noting that without factors of quantum computing such as entanglement, an increase in the efficiency of a quantum computer over a classical computer is much less likely to be achieved.

The main difficulty, and most varied approach, in implementing a quantum computer lies in finding a way to create and store a qubit while still preserving the laws of quantum mechanics, yet avoiding issues such as quantum decoherence. Quantum decoherence is the physical property of a qubit whereby it can be distorted by interaction with the world around it. Ideally a qubit should be isolated completely from its surroundings but this raises issues when trying to read it.

Dealing with decoherence puts another constraint on the quantum system, in that it must be fast enough to perform operations on individual qubits before the qubit in question is subject to decoherence. The amount of time that a system can maintain quantum-mechanical coherence, in conjunction with the speed of operations, is a major factor in evaluating the feasibility of a quantum computer.

Once this issue has been overcome, we then need to deal with the problem of manipulating the qubits we have created, it is implicitly specified that we should be able to manipulate individual qubits, which is not a simple task. Quantum mechanics evolved from the idea that light is both a wave and a particle simultaneously. As with waves, qubits are subject to interference when they interact with each other. A closed quantum system evolves in a unitary fashion based on the energy of the system.

Quantum computation requires us to control the system's energy to allow us to affect an arbitrary portion of the system using a unitary transformation composed from single spin operations and controlled-NOT gates. The fact that any unitary transformation can be made using these single two operations makes them a primary goal in the creation of quantum computers. The operations performed need to be as close to perfect as possible as imperfections can lead to quantum decoherence.

Another major issue with quantum computers is input, there is no point in creating a perfect quantum computer if it takes us longer to input the specifics to be calculated than the advantage over a classical computer. With the state of quantum computing at the moment generalising input is difficult as the input method depends on the implementation of the system as a whole, but it is only necessary to be able to input a single quantum state accurately to provide any input, as the single state can be transformed to any other using a unitary transformation. The possible accuracy of input states is another factor used in measuring the feasibility of a quantum system.

Once a quantum system has been implemented and input given, the next step is to read the output. The output from a quantum algorithm is a superposition of all the possible answers to a given problem. We know from quantum mechanics that when measured the system collapses to the output state, which is an easily readable output. When measured there is a high probability that the superposition will collapse to a useful answer.

A good example of this is Shor's algorithm, which collapses a superposition of all possible integers into a random integer that has a high probability of allowing us to calculate a value $r \epsilon R$ where R is the set of all factors of the number in question.

The problem with output in a quantum system is measuring a qubit with accuracy. For accurate measurement of qubits they are required to be coupled strongly, this is something which we try to avoid generally because it can lead to quantum decoherence, therefore most quantum systems are forced to compromise between output fidelity and level of decoherence.

Issues of fidelity and decoherence can be offset by using weak measurements, performed constantly, by completing the calculation in a short time and using multiple quantum computers together to give an aggregate signal that can be read more easily. This in itself can raise some issues, some algorithms, for example shor's algorithm, require an integer output that cannot be obtained by combining multiple different outputs, as the average may not be an integer, but with some adaptation these can also be made to work in this way.

With so many different problems and solutions to the issues of quantum computing it is unsurprising that there are a large number of different approaches to the issue. Some of the more major candidates for feasible quantum computers are:

- Superconductor-based quantum computers (including SQUID-based quantum computers)
- Trapped ion quantum computer
- Optical lattices
- Topological quantum computer
- Quantum dot on surface
- Nuclear magnetic resonance on molecules in solution (liquid NMR)
- Solid state NMR Kane quantum computers
- Electrons on helium quantum computers
- Cavity quantum electrodynamics (CQED)
- Molecular magnet
- Fullerene-based ESR quantum computer
- Optic-based quantum computers (Quantum optics)
- Diamond-based quantum computer
- Bose–Einstein condensate-based quantum computer
- Transistor-based quantum computer - string quantum computers with entrainment of positive holes using an electrostatic trap
- Spin-based quantum computer
- Adiabatic quantum computation
- Rare-earth-metal-ion-doped inorganic crystal based quantum computers

Although there is an extremely large number of available approaches to the issue of quantum computation there are three main categories into which most implementations can be placed, these are:

- Spin based
- Charge based
- Photon based

Most of the progress in quantum computing has come out of spin based methods but there are notable possibilities in the other implementations.

Spin Based Quantum Computers

Within this area of quantum computing there are two main implementation methods, nuclear spin and electron spin.

Nuclear spin, known as Nuclear Magnetic Resonance (NMR) uses the spin state of molecules (usually alanine) as qubits. It differs from other quantum computers in that it uses a group of systems (molecules in this case). The molecules are initialised as the thermal equilibrium state, given by the density matrix:

$$\rho = \frac{e^{-\beta H}}{\mathrm{Tr}(e^{-\beta H})},$$

Where H is the Hamiltonian matrix (measurement of the system's energy) of a molecule and:

$$\beta = (\text{Boltzmann constant} \cdot \text{temperature})^{-1} = \frac{1}{kT}$$

The Boltzmann constant is the physical constant relating to a particle's energy at the temperature of the system as a whole. Operations are performed on the group of systems as a whole, applying magnetic pulses perpendicular to a strong static field from a very large magnet.

This type of system has already proven itself a viable solution. In 2001 researchers from IBM managed to implement Shor's algorithm factorising 15 on a 7-qubit NMR computer, however some people do not believe that the experiment demonstrated quantum computation as there was no observation of entanglement.

Electron spin quantum computing has become much more prevalent over the past few years. The field of electron spin work is based around controlling the spin of an electron around an atom. In this sense it can model a Bloch sphere very well, as the position of the electron relative to the nucleus can be directly related to its position on the sphere relative to the origin.

There are two methods to represent a qubit within this field, the electron spin and the charge. Representing the qubit by a real spin-1/2 is well defined as the two-dimensional Hilbert space is the whole space, there are no extra dimensions that could possibly cause decoherence. Real spins also have a relatively long decoherence time due to the fact that their spin is not affected by environmental fluctuations of surrounding electrons' electric potential, this is a very useful effect. For quantum computation to be performed there is also a need for controlled entanglement, which in this case is achieved by temporarily coupling the spin of two electrons (via a Heisenberg Hamiltonian). Subject to certain conditions we can apply unitary operations to an electron and alter its rotation about the nucleus.

A major issue with this implementation was that until very recently, the most commonly used approach: electronic spin resonance (ESR), would indiscriminately spray a sample with electromagnetic radiation, causing all the electrons within the sample to spin with the same orientation. This approach was successful but defied the overall goal of having individual electrons represent distinct pieces of data.

In recent weeks there have been significant advances made in Electron Spin Control, researchers from two universities in America were able to control the spin of a single electron using tiny electrodes without affecting neighbouring electrons. The control is over the orientation of the spin as opposed to exact three dimensional coordinates but gives us a small scale way of representing a qubit, pushing Electron Spin Control into the limelight in quantum computing. The method adopted also has the additional benefit of being able to control an electron's spin in roughly one billionth of a second, a speed up of nearly one hundred orders of magnitude over previous electron spin methods. While this is a major step forward the researchers still need to overcome the problem of having qubits interact, research into the creation of a two-qubit gate is currently being undertaken.

Charge Based Quantum Computing

This type of quantum computing involves adding a single excess electron to a system made up of two quantum dots and using the location of the negative charge to represent a location on a Bloch sphere. This system is popular as it is possible to control the location of the electron relatively easily using electrodes that are biased to affect its relative position between them. A quantum dot is a semiconductor that is subject to strict confinement of the bound state of its electrons.

The problem with this sort of method is that it requires that the excess electron normally be stored in one of the two lowest orbitals of the atoms in the system. For this to occur it usually requires the temperature to be lower than the excitation level of the third orbital, this leads to the requirement that manipulation of the system occurs without changing the entropy of the system so as to not excite the electrons into an incorrect layer and cause decoherence. The electrodes used to control the system have two different functions, a set of electrodes control the potentials (voltages) of the two dots and a single electrode controls the electron tunnelling rate between the dots.

To measure this type of qubit we use a quantum point contact (QPC) or single electron transistor (SET) to perform projective measurements, both these methods essentially extract the electron and base the output on which of the two dots provides it.

Two qubit systems can be created using the properties of Coulomb potential. The asymmetrical formation of the dots allows us to use the slight charge at different points of the dots to create a coupling between two systems. Using supporting dots in this way we can create a two qubit interaction.

Optic Based Quantum Computing

The most prevalent field of optic based quantum computing is that of optic lattices. An optical lattice is a set of wave lasers; they create a spatially periodic polarisation pattern formed by interference of two or more of the laser beams. The electric field they create can interact with atoms, the atoms react to the potential and will generally congregate in the potential minima. A way to visualise this is to think of an egg carton where the atoms are represented by marbles trapped in each hollow of the carton.

The depth and periodicity of the lattice can be controlled by changing the power of the laser and the wavelength respectively. To get two atoms to entangle they must be moved into the same minima, which can be done by changing the lasers' respective phases; by rotating the phase of the lattice you move the location of the potential minima taking the trapped states along with it. If you only rotate the phase of one polarisation then you only move one of the types of states leaving the other where it is.

If the system uses ions trapped inside the lattice, their charge means they can interact with any stray electric field in the environment and so are extremely susceptible to decoherence. For this reason it is more viable to use neutral atoms as they will not interact with the electric fields and should barely influence each other; this gives the potential to trap millions of atoms at one time. Often alkali atoms are chosen as candidates due to their relatively simple structure, being that there is only a single outermost electron.

Another more recent development in the field of optic based quantum computing is the use of protons to represent qubits, encoding the state by the polarisation of a photon. Using photons is of high interest because they are relatively free of decoherence and one-qubit gates can easily be performed with polarisation rotation using wave-guides made of birefringent (double refraction) materials.

In 2001 the KLM (Knill-Laflamme-Milburn) scheme was introduced, which showed that scalable quantum computing was possible using a single photon source with linear optical circuits. Efforts have been moved to focus on high-efficiency of single photon detectors and sources; Current systems operate at room temperature at ten megahertz, with only seventy percent efficiency, with photon loss posing similar issues as decoherence. Solving this issue while maintaining the speed of a system is a major issue for the field. One possible solution to high-efficiency is to multiplex the optical sources to emit photon-pairs instead.

In 2009 the University of Bristol developed a silicon chip that uses a single photon source. This system has implemented Shor’s algorithm successfully factoring 15; four photons where coupled and used for input, directed in and out of the chip using optical fibres. On the chip photons travel through silica waveguides to form a sequence of logic gates with output being of a high-efficiency.

While there is much research still ongoing in the field of quantum computing there appears to be no indication of a general consensus towards one method of implementing a quantum computer. It is clear that research will continue to improve quantum computers and one day they may even become viable as a commercial enterprise. Yet in predicting how the field will evolve one opens themselves up to as much ridicule as Thomas J Watson received for his comments on the world's need for only five computers; ironically quantum computers may yet prove his conjecture correct.

G53NSC Non-Standard Computation

Coursework

QCA – Quantum Cellular Automata

Bc. Jiří Kremser *jxk19u*
Bc. Ondřej Božek *oxb09u*

Academic Year 2009-2010

Abstract

Quantum Cellular Automata (QCA) is computational model based on classical cellular automata (CA), reversible computation and exploiting quantum phenomena. The aim of this paper is to give a reader insight to a general concepts of QCA, as well as to show some models, especially Quantum dot QCA, however it does not exploit quantum phenomena. Since QCA is generalization of classical CA, the history and fundamentals of CA will be described. Some examples of emergent complex behaviour on the basis of a set of simple rules will be discussed on John Conway's Game of Life.

1.Introduction

The concept of Cellular Automata comes originally from John von Neumann and is spread across many fields. It can model any complex closed systems if we accept the constraint that the time and space are discretized to time steps and cells. These models are intuitive in contrast with the system of differential equations, however they preserve the continuity of the nature. Since these models can model physical laws and on the subatomic level we can observe quantum phenomena with all its weirdness, it is obvious to extend this concept to quantum world. Moore's law implies, amongst other, the computer circuits will get smaller and smaller. Rather than avoiding the quantum phenomena it is better to exploit them and make quantum exponential speed-up possible. QCA is theoretical computation model and this paper is summary of this problematic.

2.Classical Cellular Automata

Formally CA is a 4-tuple (C, Σ, N , *f*), where C denotes an d-dimensional array of cells or lattice (cells are indexed by vectors from Z^N), Σ denotes the alphabet, giving the possible states each cell may take, N denotes the neighbourhood (i.e. $N \subset Z^d$) and *f* denotes the transition function of type $\Sigma^N \rightarrow \Sigma$. The state of all cells in time is called configuration. Significant configuration is the starting configuration, since it has to be provided with the CA.

CA is discrete computational model which is capable to provide the same computational power as Turing Machine (TM) therefore it is Turing Complete. CA were probably firstly used by famous scientist Jon von Neumann in late 1940s when he was trying to describe self reproducing automaton. He succeeded by introducing two dimensional Von Neumann's cellular automata with rules (*f* function from definition) and starting configuration such that after certain amount of time steps there were two copies of the pattern from starting configuration and so on. Later on in 1980s Stephen Wolfram in his famous book *New Kind of Science*[14] defined four classes of cellular automata depending on complexity and predictability of their behaviour.

2.1.Wolfram Classes

He showed his classes on one dimensional CA, where neighbourhood of particular cell were the cell on the left hand side and the cell on the right hand side. There were only two states of cells – zero/one. There were only eight possible combination of states of a cell and its neighbour cells. Generally it is a^b where a denotes the cardinality of Σ and b the size of the neighbourhood. By application transition function *f* on each of these eight triplets it could make this cell 0 or 1 in the next time step. This set of rules can be rewritten as an 8 bit string, where each bit denotes the return value for arguments 111, 110, ..., 000 as shown at *figure 1*.

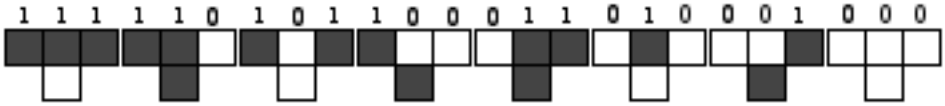


Figure 1: Rule 90 (90dec = 01011010bin)

Running CA with Rule 90 with starting configuration ...000000010000000... (one zero in the center) will "draw" a Sierpiński triangle as shown at *figure 2*.
Rows represent the development of configurations in time. The starting configuration is on the top.

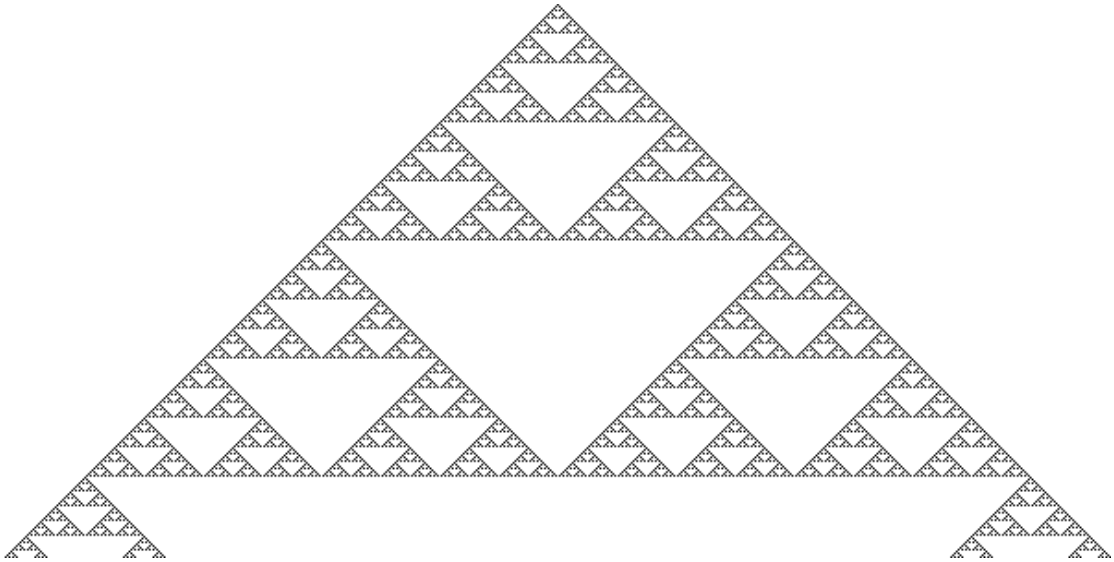


Figure 2: Progress after ~300 iterations.

a) Class 1

Evolution in time leads to some stable configuration. If there is no change between two configurations, this configuration will never change, because the transitional function f can depend only on cells and its neighbourhoods (not on time).

b) Class 2

Evolution in time leads to some periodic patterns. It is not stable, but still very well predictable.

c) Class 3

Chaos behaviour. Without simulating it is impossible to predict the future development.

d) Class 4

On the edge of chaos. Characteristics are between class 2 and 3, some regions behave randomly, some are relatively stable or predictable.

Research assistant of Wolfram's Matthew Cook showed and proved[1][9] that one dimensional CA with "Rule 110" is universal computational model. The same result was achieved with John Conway's Game of Life[10]. Providing the Curch-Turing thesis holds, these models are equivalent and can simulate every computable function.

2.2.Game of Life

Introduced by British mathematician John Conway in 1970. It is probably the most known example of a CA. The Game of Life (GoL) is a 2-dimensional CA with neighbourhood made of eight adjacent (horizontally, vertically or diagonally) cells. The state of cell can be either life (1) or dead (0). The evolution of each cell in time depends only on the states of its neighbours. It is member of Wolfram's *class 4*.

Schema for transition function:

- A dead cell comes to life if it has exactly three living cells in its neighbourhood.
- A cell remains living if it has two or three living cells in its neighbourhood.
- A cell dies otherwise. (on overpopulation or loneliness)

Conway discovered some interesting patterns and shapes in his game and divided them into these following categories.

a) stable or periodic repeating – stable patterns survive from generation to generation without changing, whilst periodical patterns (*oscillators*) repeats after certain number of time steps. **Block** and **Blinker** are shown at *figure 3.*

b) spaceships – they repeat its shape after certain number of time steps, but they on the position next to original. The most famous spaceship and key pattern in *GoL* is the **glider** shown at *figure 3*.

c) guns – "fires" spaceships

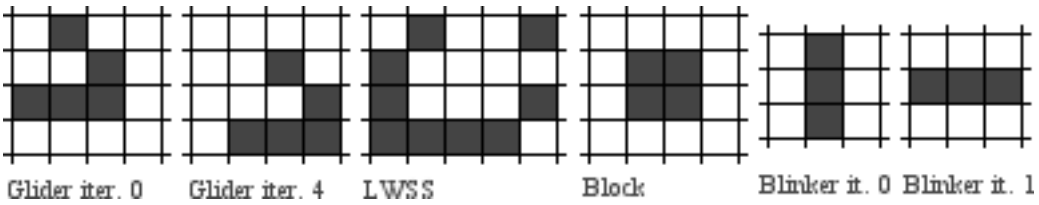


Figure 3: Some basic GoL patterns.

1 He proved it by emulating another universal model – Tag System [8] on this 1-dim. CA.

Gliders are able to move diagonally across the 2D grid. LWSS (Lightweight spaceship) moves vertically or horizontally according to its rotation. Using gliders and other patterns it is possible to implement Minsky Machine or Turing Machine so it is universal as well. Furthermore, a generation can contain a collection of guns that fire gliders in such a way as to construct new objects, including copies of the original pattern. A "universal constructor"[11] can be built which contains a Turing complete computer, and which can build many types of complex objects, including more copies of itself, like Von Neumann's self reproducing CA.

3.Reversible Cellular Automata

Global transition function F maps an arbitrary configuration to another one. A CA is said to be reversible cellular automaton (RCA), if and only if his global transition function F is bijective. In other words each configuration has exactly one preimage according to F. Configuration which has no preimage[2], informally called Garden of Eden, appears when F is not surjective. John Myhill and Edward Forrest Moore proved[3], that if CA is injective then it is also bijective (surjectivity implies injectivity) thus reversible. There are 16 RCA out of 256 1-dimensional CA with neighbourhood size of one. For dimension grater or equals to two it is undecidable whether an arbitrary CA is reversible. Luckily Tommaso Toffoli proved[12], that any n-dimensional CA can be simulated by n+1-dimensional RCA in 1977. In 1990 Toffoli and Norman H. Margolous introduced Partitioning Cellular Automata which can be build up from classical CA by partitioning technique[13]. The reversibility is important in the world of quantum mechanics, since Landauer’s principle can predict the energy consumption within closed system.

4.QCA

It seem obvious that cellular automaton are themselves physics-like models of computations. These models are not continuous but represents space as simple lattice. In CA Everything is discrete including time, which jumps discontinuously. In the late 70s, Fredkin proposed that the world we live in is a huge cellular automaton. Fredkin s thought that all physical quantities can be seen as packets of information in a cellular automaton. Therefore it seems natural to study quantum extensions of CA as world from the most detailed view seems quantum-based at the moment. There is a big advantage in QCA models in comparison with other models of quantum computation. Principle of computation is in quantum cells interaction. Need for for environment interaction is reduced, and so is possibility of decoherence, the main obstacle for realization of a quantum computer. Cells are not required to be able to distinguish one neighbour from another. In cellular automaton uniform rules are applied in parallel across a whole lattice, therefore it is not needed to address each cell (qubit) separately.

2 They could not been created by application of F on any configuration.
3 Garden of Eden theorem.

This helps to eliminate errors resulting from cross talk on neighbouring cells (qubits) known from other models of quantum computational machines. Another benefit of CA framework is that many fabrication techniques naturally produce equally spaced units suitable as base for lattice used in cellular automata computation. Physical systems proposed as framework for QCA include quantum dot arrays and endohedral fullerenes.

4.1.History

- 1965 the first example of a Quantum Cellular Automata was Feyman's "quantum checkerboard" model of spinors in 2d space-time. Feynman invented the model in the 1940s while developing his space time approach to quantum mechanics. He did not publish the result until it appeared in a text on path-integrals co-authored by Albert Hibbs.

- 1988 Grossing and Zeilinger attempted to introduce the concept of quantum cellular automata, however their model has little in common with the models currently in use.

- 1990 Norm Marglous wrote Parallel Quantum Computation. Feynman and others have shown that the quantum formalism permits a closed, microscopic, and locally interacting system to perform deterministic serial computation. In this paper Marglous show that this formalism can also describe deterministic parallel computation

- 1995 the first successful model of one dimensional quantum cellular automata was due to Watrous.

4.2.What is QCA in fact?

There is no only generally accepted QCA model. There are many different definitions of QCA. Many of them seem to be computationally equal. But one unique, computationally powerful definition is missing. There is no such axiomatic definition, unlike its classical counterpart, that can immediately bring up way how to construct or enumerate all the instances of this model. Each set of authors defines QCA in their own particular way. The main common signs of various QCA definitions consists of a d-dimensional lattice of identical finite-dimensional quantum systems, a finite set of states, a finite neighbourhood scheme of single cell, and a set of local unitary transition rules. The states $s \in \Sigma$ are basis states spanning a finite-dimensional Hilbert space. At each point in time a cell represents a finite-dimensional quantum system in a superposition of basis states. The global evolution function represents the discrete-time evolution of strictly finite cell lattice. There is demand on this evolution function to be unitary.

5.QCA types

5.1.Grössing-Zeilinger QCA

Grőssing and Zeilinger introduced the term "quantum cellular automata" and they were also first which attempted to create appropriate model. This pioneering definition of QCA, however, has not been studied much further, mostly because the "non-local" behaviour makes the Grőssing-Zeilinger definition non-physical. In addition, it has little in common with the concepts developed in quantum computation later on. The Grőssing-Zeilinger definition really concerns what is called today a quantum random walk.

5.2.LQCA - Watrous QCA

The first model of QCA researched in depth was that introduced by Watrous. A Watrous-QCA is defined over an infinite 1-dimensional lattice, a finite set of states including a quiescent state. The transition function maps a neighbourhood of cells to a single quantum state simultaneously over whole lattice. This model of Quantum Cellular Automaton is in literature also referred to as 1d-QCA or Linear Quantum Cellular Automaton (LQCA). Following definition is adapted from [3].

Definition (LQCA). A linear quantum cellular automaton is a 4-tuple $A = (\Sigma, q, N, \delta)$, where (with $q\Sigma = \{q\} \cup \Sigma$):

- Σ is a finite set of symbols (i.e. "the alphabet", giving the possible basic states each cell may take);
- q is a symbol such that $q \notin \Sigma$ (i.e. "the quiescent symbol", which may be thought as a special state for empty cells);
- N is a set of n successive signed integers (i.e. "the neighbourhood", telling which cell is next to whom);
- $\delta : Ή_{(q\Sigma)^n} \rightarrow Ή_{q\Sigma}$ is a function from super-positions of n symbols words to super-positions of one symbol words (i.e. "the local transition function", describing the way a cell interacts with its neighbours).

Moreover δ must verify:

- the quiescent stability condition: $[\delta|q^n\rangle) = |q\rangle]$;
- the no-nullity condition: $\forall w \in (q\Sigma)^n$, $[\delta|w\rangle \neq 0]$.

In this definition $\acute{H}_{\Sigma}$ denotes the Hilbert space filled with the cell states Σ. Set of possible configurations of the CA is extended to include all linear super-positions of the classical cell configurations. Similarly to their classical counterparts Watrous Quantum Cellular Automata, or LQCA in other words, consist of a row of identical, finite dimensional, quantum systems. One cell is labeled "accept" cell. The quiescent state of allows only a finite number of cells to be active and thus makes the lattice finite. Finite lattice is essential to avoid an infinite product of global evolution (more thereinafter) and, to obtain a well-defined QCA. These rows evolve in discrete time steps by means of local transition function. Transition function maps the cell configurations of a given neighbourhood to a quantum state. Homogenous and synchronous application of transition function gives rise to global evolution Δ.

In order to make LQCA physically acceptable model of computation, it must be confident that the global evolution Δ is physically acceptable in a quantum theoretical setting, it must be certain that Δ is unitary. It is possible to define transition functions that do not represent unitary evolution of the configuration. There are two properties that transition function must not have in order to construct unitary evolution. Firstly it must not produce super-positions of configurations which do not preserve the norm, global evolution preserves the sum of probabilities squared to 1. Secondly they must not include a global transition function which is not unitary. This leads to non-physical properties such as super-luminal signalling. Unfortunately this requirement for global evolution function to be unitary is really non-trivially related to the definition of the transition function δ. (The set of LQCA is not closed under composition and inverse.)

It would help us a lot if there would exist a function which can efficiently decide if given transition function δ constructs global evolution function which is unitary. Such a function could be than applied to whichever local transition function. This would help us to construct particular LQCA, suitable for simulation of physics, much easier. Number of local transition functions which do induce a unitary global evolution is likely to be rather scarce.

There is tendency in computer science to decide if program is valid, by means of syntactical correctness. Its obvious that once programing language is universal, adding more expressiveness does not mean adding more computational power, but it only allows us to express something easier and in more than one way. There is a catch, if we enrich syntax of the language so much that it allows us to create non-valid (unrealistic, non-physical) programs. So the user needs to performs non-trivial (non-syntactic) decisions to recognize and exclude those non-realistic variants and the process of program creation is significantly more complicated. Such a programing language can be considered too lose. This is analogous to current state of LQCA formalism.

Therefore there is a necessity to tighten the definition of linear quantum cellular automata. We need to find more restrictive definition whose unitarity may be checked algebraically or syntactically, but computational power must remain same. Pablo Arrighi managed to algebraically characterize unitary LQCA by adding constraints into the model which do not change the quantum cellular automata computational power.

This is step towards possibility of algebraical verification of local transition function. But there is still a long way ahead.

5.3.Partitioned Quantum Cellular Automata (PQCA)

As mentioned above in order to have physically acceptable model of computation we need to fairly uneasily decide the right local transition function. Transition function has to construct unitary global evolution function. If we want to avoid this non-trivial decision process, some restrictions on LQCA need to be introduced. Partitioned Quantum Cellular Automata is such kind of restricted LQCA that allows us to construct well-formed QCA easily.

One-dimensional Partitioned Quantum Cellular Automata is 1d-QCA where each cell in whole lattice is partitioned in to three sub-cells. The next state of any cell now depends only on the states of the left sub-cell of the right neighbour, the middle sub-cell of the cell itself, and the right sub-cell of the left neighbour.

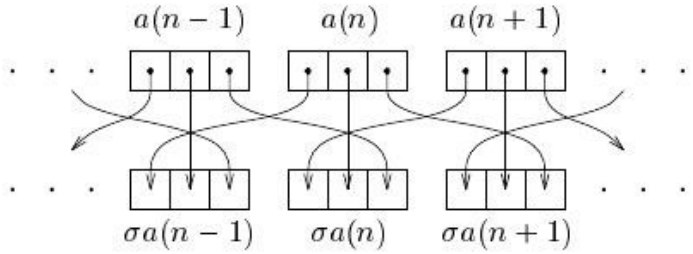


Figure 4: evolution of cells in 1d-PQCA [2]

It was shown that any Quantum Turing Machine can be simulated by PQCA with constant slowdown and every PQCA can be simulated by QTM with linear slowdown.[2] This is also proof of computational universality of PQCA. Thence this restriction does not reduce computational power of plain LQCA more than acceptable.

5.4.Quantum-Dot Cellular Automata

Quantum-Dot Cellular Automata (QdCA) often called only QCA are classical CA implemented in quantum mechanical structures. They do not exploit quantum effects for the actual computation. QdCA is more a hybrid of a quantum circuit with individual qubit control and a QCA with constant nearest-neighbour interaction. The cell in this automaton consists of four quantum dots forming a square. Electrons can tunnel between dots, but cannot leave the cell. If two excess electrons are in the same cell, Coulomb repulsion will force them to dots on opposite corners. As a corollary, there are two ground states representing zero and one as shown at figure 5.

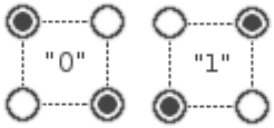
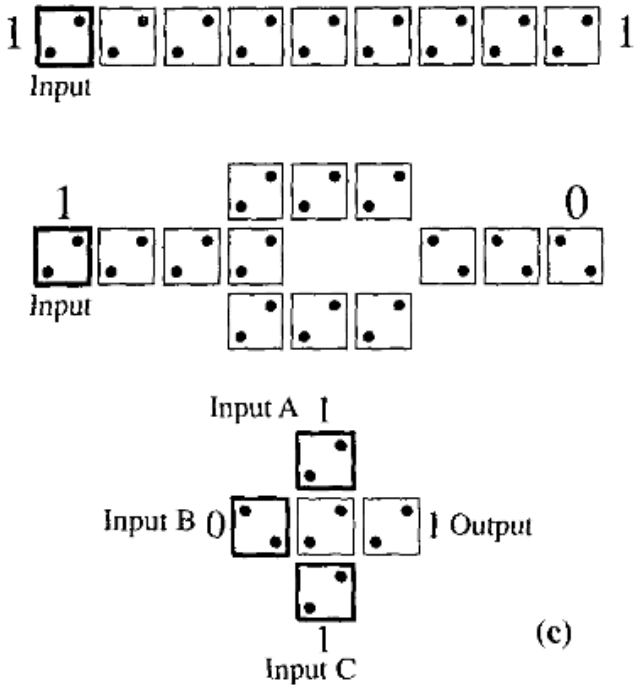


Figure 5: Q-dot ground states.

Coulombic interaction between electrons cause the cells of QdCA to take the same polarization, if two cells are close enough together. Using this principle, it is easy to build circuits and logic gates from these uniform cells.

At the figure bellow there are shown a) line of cells representing a wire, b) inverter representing operation NOT and c) majority gate.



a) When some ground state is set to the input of the line (wire) of the left hand side, the other cells tend to take the same polarization to decrease energy. In the state, when all cells have the same polarization, the electrons are as widely separated as possible.

b) The input is split into two lines and then other line is connected in the angle of 45∞. Again since the cells are capacitively coupled to their neighbour cells, the output line will take opposite polarization than the input one. This circuit is thus realizing logical function **NOT**.

c) The majority gate sets output to one, if most of inputs are set to one, zero otherwise. If input C is set to one, then output is equal to A **OR** B. If the C is set to zero, then output is equal to A **AND** B.

This set of logic operators (even without OR or AND) is universal set. Each other logic function can be build up from this basic set.

6.Conclusion

The concept of Quantum Cellular Automata is quiet young. It is currently in its early phase. Many of important definitions appeared most recently. As obvious from this paper many accurate definitions are also still missing. Many branches of this young discipline are waiting for exploration. For example multi-dimensional QCA were researched very slightly, although they for sure hide many interesting and beneficial findings. Now we can see less than tip of an iceberg. Many interesting discoveries can be expected in near future.

QCA formalism seems as promising framework for building robust and scalable computers suitable for quantum computations. They overcome some technical obstacles known from other approaches. Difficulties with manipulation of individual quantum registers and related decoherence of adjacent qubits induced by cross-talks are surpassed by global evolution of QCA without need for addressing of individual cells.

CMOS technology is approved industry standard for VLSI devices for past decades. In years to come CMOS will impose its fundamental limits. Quantum Dot Cellular Automata is one of the many proposed replacements. QCA resolves all problems of current CMOS technology, but also brings its own.

Only feasible implementation method for mass production of QCA devices is molecular QCA with inter-dot distance of 2 nm and an inter-cell distance of 6 nm. However this technology is in the presence limited by availability of its practical fabrication methods.

7.Resources

[1] *Simulating Physics with Computers*, Richard P. Feynman, 1981

[2] *On One-Dimensional Quantum Cellular Automata*, John Watrous, 1995?

[3] *Algebraic Characterizations of Unitary Linear Quantum Cellular Automata*, Pablo Arrighi, 2006

[4] *Parallel Quantum Computation*, Norman Margolus, 1994

[5] *Quantum Cellular Automata*, Bassam Aoun, Mohamad Tarifi

[6] *Quantum Mechanics and Path Integrals*, Feynman and Hibbs, New York: McGraw-Hill, Problem 2-6, pp. 34-36, 1965

[7] *Quantum Cellular Automata*, Karoline Wiesner, 2008

[8] *Universality of Tag Systems with P=2***,** Cocke, J. and Minsky, M. J. Assoc. Comput. Mach. 11, 15-20, 1964.

[9] *Universal Cellular Automata and Class 4*, A. Dhar, P. Lakdawala, G. Mandal, S. R. Wadia, 1994.

[10] *The fantastic combinations of John Conway's new solitaire game "life"* , Martin Gardner , Scientific American 223 (October 1970): 120-123.

[11] *Construction Theory, Self-Replication, and the Halting Problem*, Hiroki Sayama, Journal-ref: Complexity 13(5):16-22, 2008.

[12] *Computation and construction universality of reversible cellular automata,* Tommaso Toffoli, J. Comp. Syst. Sci. 15 (1977), 213-231.

[13] *Invertible cellular automata: A review.* Tommaso Toffoli and Norman H. Margolus. Physica D: Nonlinear Phenomena, 45:229–253, September 1990.

[14] *A New Kind of Science*, Wolfram, Stephen, 1st edn.. (Wolfram Media, 2002).

Spring 10

Entropy and Information

Andrew Sharkey (aps07u) and Richard Stokes (rxs27u)

Entropy is the measure of the uncertainty within a variable whether it being the uncertainty of correctness or within a quantum system where it is the uncertainty of what state that system is in. This paper explores in more detail what entropy is, how it is measured and how the entropy of other variables affects it. It first looks at the entropy within a classical system where it is commonly known as Shannon entropy and then expands onto quantum system where it looks at the Von Neumann entropy and how known properties of the entropy can be used to provide clues as to what state the system is in.

Shannon Entropy

When the term entropy is used within the context of classical information theory, majority of the time people are referring to the Shannon entropy. The Shannon entropy is a way of measuring how much information we gain, on average, from finding out the value of a random variable. Another way of looking at it is that the entropy is a measurement of how much uncertainty there is about the value of a random variable before its value is revealed. Both of these views fully complement each other, so entropy can be viewed either as a measurement of uncertainty on the value of a random variable, or the amount of information we gain from discovering the value of a random variable. This concept was first introduced by Claude E. Shannon in his 1948 paper "A Mathematical Theory of Communication".

The labels attached to the possible values of a variable have no effect at all on the amount of information we gain, or the variables initial uncertainty. It is the probability of each of these values occurring that tells us how much information we are going to gain from finding out the value of the variable. For example if we had jar of sweets and we knew that picking a sweet at random has a probability of 1/3 that the sweet is apple and a probability of 2/3 that the sweet is strawberry. Then we gain the exact same amount of information from finding out the value of a random variable which has the possible value of 'a' and 'b' with probabilities of 1/3 and 2/3 respectively. So for this reason the entropy of a random variable is a function of the probabilities of each of the possible values of the variable occurring. So for a random variable X which can take the possible values $x_1, x_2, \dots, x_n$ with the respective probabilities of $p_1, p_2, \dots, p_n$ the Shannon entropy is defined by

$$H(X) = -\sum_x p_x \log p_x \tag{1}$$

With entropy there is no need to worry about a case where log 0 will occur because if $p_x = 0$ there is no possibility of that value being obtained therefore there is no need to include it as it will have no effect on the entropy of the variable. The maximum value of entropy for a random variable where all possible values have equal probabilities is log n, where n is the number of possible values.

Entropy can be used to calculate the absolute best possible lossless compression of any communication. It has been proven that the average length of a message which has been compressed as small as possible is equal the entropy of the message. So for a concrete example, there are two people communicating in Morse code, but the only messages they are sending to each other are 1, 2, 3 and 4. So with Morse code only having a possible alphabet of 0 and 1, with out compression it would take 2 bits to send each symbol. But some symbols are sent more often than others, 1 is sent 1/2 of the time 2 is sent 1/4 of the time and 3 and 4 both have a probability of 1/8. These probabilities can be used to compress the message so that more common symbols take fewer bits to store than the more rare ones. So the average length before compression would be

$\frac{1}{2}\cdot 2+\frac{1}{4}\cdot 2+\frac{1}{8}\cdot 2+\frac{1}{8}\cdot 2=2$ and the entropy of the message would be $-\frac{1}{2}\log\frac{1}{2}-\frac{1}{4}\log\frac{1}{4}-\frac{1}{8}\log\frac{1}{8}-\frac{1}{8}\log\frac{1}{8}=\frac{7}{4}$. So one way of encoding these values would be to encode 1 as a bit string 1, 2 as 01, 3 as 001 and 4 as 000. So now the average length of the message would be $\frac{1}{2}\cdot 1+\frac{1}{4}\cdot 2+\frac{1}{8}\cdot 3+\frac{1}{8}\cdot 3=\frac{7}{4}$ which means the average length is now equal to the entropy therefore this is the highest level of compression that can be achieved without loss of data.

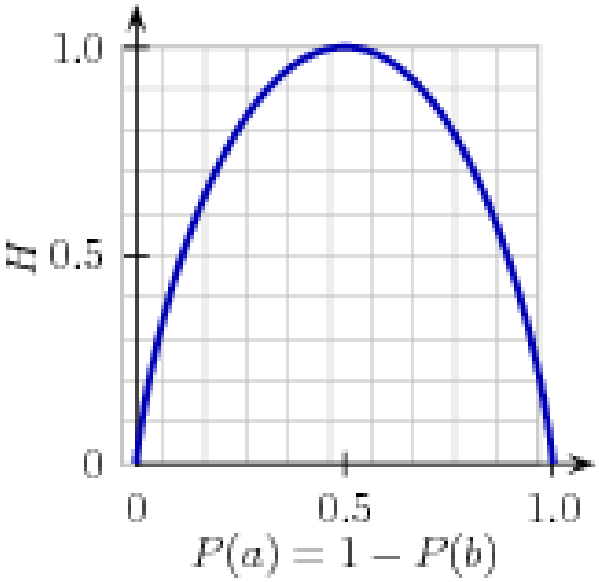
Basic properties

Binary entropy

Binary entropy is the entropy of a variable with only two possible outcomes; it is very useful in classical and quantum entropy and is defined by

$$H_{Bin}(p) = -p\log p-(1-p)\log(1-p) \tag{2}$$

With p and 1-p being the probabilities of the two possible outcomes. The binary entropy can be plotted like so



Notice how $P(a)=1-P(b)$ and binary entropy reaches its maximum value of 1 when $P(a)=\frac{1}{2}$ and as can easily be seen by the graph the binary entropy is a concave function since the function is above any line that can cut the graph.

Relative entropy

Relative entropy, or the Kullback-Leibler divergence, is a non-symmetric measurement of how close two different probability distributions, p_x and q_x, are over the same index set, x. Relative entropy measures the likely amount of extra bits that would be required to encode samples from p_x when

compressed using a method based on q_x, rather than using a method based on p_x. We can define the relative entropy by

$$H(p_x \,||\, q_x) \equiv \sum_x p_x \log \frac{p_x}{q_x} \tag{3}$$

On its own relative entropy is not greatly useful, it is mainly useful when regarding other entropic quantities as special cases of relative entropy. For example it has been proven relative entropy is non-negative, $H(p_x \,||\, q_x) \geq 0$, with equality if and only if $p_x = q_x$ for all x. It is this property that allows us to prove one of the fundamental facts about entropies. For p_x, the probability distribution for X, over d number of outputs, and let $q_x \equiv \frac{1}{d}$ be the uniform probability distribution over the outputs, then we can define this as

$$H(p_x \,||\, q_x) = H\left(p_x \,||\, \frac{1}{d}\right) = -H(X) - \sum_x p_x \log \frac{1}{d} = \log d - H(X) \tag{4}$$

Then from the non-negativity of relative entropy we can deduce that $H(X) \leq \log d$ with equality if and only if X is uniformly distributed. The technique of finding expressions for entropic quantities in terms of relative entropy is used in both classical and quantum entropy.

Conditional entropy and mutual information

The first thing that's needs to be defined is joint entropy; joint entropy is the measurement of the entropy in a joint system of two random variables and can be defined as

$$H(X,Y) \equiv -\sum_{x,y} p_{x,y} \log p_{x,y} \tag{5}$$

So an example of joint entropy would be picking two coins, X and Y, at random out of a jar that only contains 1ps, and 2ps, and then the entropy would be calculated using the probability of every possible combination of the two coins, $p_{x,y}$. This equation can easily be extended to more than just two variables by changing the probabilities to $p_{x,y,z}$ as long as you check every possible combination of the variables.

One of the properties of joint entropy is that the joint entropy is always greater then or equal to the entropy of each of the original systems $H(X,Y) \geq H(X)$, though this is untrue when Y is a deterministic function of X. Also in joint entropy the sum of the entropy of the original systems will always be greater than or equal to the joint entropy of the systems $H(X) + H(Y) \geq H(X,Y)$ this is known as subadditivity, though it does not hold true if X and Y are completely independent variables.

If we were to discover the value of one of our systems in a joint entropy say Y, then we now know H(Y) bits of data about the joint system, so we would have to define a new entropy which would be the entropy of X conditional on knowing Y, this is known as conditional entropy, and that would be defined as

$$H(X|Y) \equiv H(X,Y) - H(Y) \tag{6}$$

The mutual information of a joint system is the measurement of how much information the original systems have in common. We would define the mutual information of X and Y as

$$H(X:Y) \equiv H(X) + H(Y) - H(X,Y) \tag{7}$$

It is also worth noting that the mutual information can easily be calculated with the conditional entropy like so $H(X:Y) \equiv H(X) - H(X|Y)$
Some additional properties of Shannon entropy are:

- $H(X,Y) = H(Y,X)$, $H(X:Y) = H(Y:X)$ *but* $H(X|Y) \neq H(Y|X)$ because $H(X|Y)$ means that you know the value of Y and $H(Y|X)$ means that you know the value of X.

- Since entropy cannot be negative $H(X|Y) \geq 0$, and therefore $H(X:Y) \leq H(X)$ since it is impossible for the shared information to be bigger than the entire information of a system, unless X is a deterministic function of Y.

- $H(Y|X) \leq H(Y)$ because since you now know X you also know the mutual information that X held, and therefore $H(X:Y) \geq 0$ also has to be true. Though these properties do not hold true when X and Y are completely independent variables.

- Strong subadditivity also holds true for Shannon entropy so $H(X,Y,Z) + H(Y) \leq H(X,Y) + H(Y,Z)$ this is because on the right hand side of the equation the mutual information between X and Z is not exploited. Though this property does not hold true if Z -> Y -> X forms a Markov chain, Markov chains are explained in the next section.

- Conditioning reduces entropy so $H(X|Y,Z) \leq H(X|Y)$. This is true because if we know both Y and Z we therefore know the mutual information shared by Y and X and the mutual information shared Z and X, as opposed to just knowing the mutual information between Y and X.

- There is also a rule for chaining conditional entropies which is for a set of random variables $X_1 , \dots , X_n$ and Y then

$$H(X_1 , \dots , X_n \mid Y) \sum_{i=1}^{n} H(X_i \mid Y, X_1, \dots , X_{i-1}) \qquad (8)$$

So an example would be for n = 3,

$$H(X_1, X_2 , X_3 \mid Y) = H(X_3 \mid Y, X_1, X_2) + H(X_2 \mid Y, X_1) + H(X_1 \mid Y,)$$

The data processing inequality

Most of time when doing computations we are doing them on information that we have available to us, but that information can be flawed as information is subject to noise before we are able to obtain it. The data processing inequality, which is a basic inequality of information theory, states that the information about the output of a source can only decrease with time, and once that information has been lost it is impossible to retrieve it.

A Markov chain of random variables captures the idea of the data processing inequality. A Markov chain is a sequence of random variables such that they all independent of the other variables except the variable that came right before. So for example in the Markov chain X->Y->Z, X and Y are not independent because Y follows after X but Z and X are completely independent because they have Y in-between them. The formal definition of a Markov chain is

$$p(X_{n+1} = x_{n+1} \mid X_n = x_n, \dots, X_1 = x_1) = p(X_{n+1} = x_{n+1} \mid X_n = x_n) \quad (9)$$

So an example of a data processing inequality would be, if we had a Markov chain, X->Y->Z, then

$$H(X) \geq H(X:Y) \geq H(X:Z) \quad (10)$$

The inequality is saturated if and only if, when given a value for Y, it is possible to reconstruct the value of X.
This result is intuitively plausible because it tells us that if we have a random variable, X, and X is subject to noise, this producing the variable Y, then it is impossible for us to take any action that would result in increasing the amount of mutual information between Y and X.
We are able to prove (10) using some of the previously defined properties of Shannon entropy. With the simple logical step that if X->Y->Z is a Markov chain then so must Z->Y->X, combined with the fact that $H(X:Z) \leq H(X:Y) \equiv H(X|Y) \leq H(X|Z)$ then we know that $H(X|Y) = H(X|\,Y,Z)$. Then from here the problem can be reduced down to $H(X,Y,Z) - H(Y,Z) = H(X|Y,Z) \leq H(X|Z) = H(X,Z) - H(Z)$ which has already been proved by the strong subadditivity inequality.
If $H(X:Y) < H(X)$, then it is impossible to reconstruct X from Y since we do not have all the data of X within Y, and if Z is created only from the information of Y then X->Y->Z must be a Markov chain and therefore $H(X) > H(X:Z)$ according to the data processing inequality(10). But if $H(X:Y) = H(X)$ then we know that Y contains all of the information of X and therefore $H(X|Y) = 0$ and thus whenever $p(X = x, Y = y) > 0$ it is equivalent to $p(X = x, Y = y) = 1$. This means that if we know that $Y = y$ we can reconstruct X since we will also know that $X = x$.
Now that we know that if X->Y->Z is a Markov chain then Z->Y->X is also one then we can witness the data pipelining inequality which is defined for the Markov chain X->Y->Z as

$$H(Z:Y) \geq H(X:Z) \quad (11)$$

This inequality is basically saying that any information that Z shares with X must have been obtained from Y, and therefore part of the mutual information of Z and Y. The information is said to be pipelined from X through Y to Z.

Von Neumann Entropy

The Von Neumann Entropy is the quantum equivalent of Shannon's entropy where it measures uncertainty with regard to quantum states but using density matrices as opposed to probability distributions. A density matrix describes the probability distribution of a state.

Von Neumann described the entropy of a quantum state ρ by:

$$S(\rho) = -tr(\rho \log_2 \rho) \quad (12)$$

The tr function is the trace function defined as the sum of the diagonal values within the matrix: $tr(A) = \sum_i A_{ii}$. The above formula can also be represent as (if λ_x are eigenvalues of ρ, eigenvalues being values that only change the magnitude and not its direction of a vector within the vector space) where $0 \log 0 = 0$ (as you would expect where something has a probability of 0 then it shouldn't contribute to the entropy):

$$S(\rho) = -\sum_x \lambda_x \log_2 \lambda_x \quad (13)$$

Properties of Von Neumann Entropy

- Entropy cannot be negative.
- Entropy can only ever be 0 if and only if the state is pure.
- In a d-dimensional Hilbert space the entropy is lesser or equal to log d.
- Within a composite system if AB is pure then S(A) = S(B).

With regard to all the equations, inequalities and properties that are talked about in this section their uses are to provide clues to what state a quantum system is currently in. In comparison to a classical system where the state is already known, in a quantum system if the state was wanted a measurement of some kind would need to be performed which isn't always desired. Thus these properties are used to remove some of the uncertainty (reduce the entropy) to attempt to gain the state without performing a measurement.

Quantum Relative Entropy

The quantum relative entropy is yet again another extension of the relative entropy that Shannon defined classically. Where ρ and σ are density matrices the relative entropy of ρ to σ is:

$$S(\rho \| \sigma) = tr(\rho \log_2 \rho) - tr(\rho \log_2 \sigma) \quad (14)$$

The quantum relative entropy is non-negative if ρ = σ. (Klein's inequality) but sometimes the quantum relative entropy can be infinite where the kernel of σ is orthogonal (meaning that the two vectors never meet within the vector space) therefore $\lim_{\sigma \to 0} \rho \log \sigma = \infty$ -> $S(\rho \| \sigma) = \infty$.

Continuity of the Entropy

To show that the entropy is continuous (meaning any small change in the input results in a small change in the output. We can acquire the bound of how much the change could be by using Fannes' inequality:

$$|S(\rho) - S(\sigma)| \leq T(\rho,\sigma)\log_2 d + \frac{1}{e} \qquad (15)$$

where d is the dimension of the Hilbert space, T( ) is the trace distance. The trace distance is the average success probability when distinguishes two states by using $\delta(\rho,\sigma) = \frac{1}{2}tr(|\rho - \sigma|)$.

Measurements and Entropy

Depending on what sorts of measurements are carried out on the system the entropy of the quantum system can change. For example if projective measurement was used with projectors P_i then $\rho' = \sum_i P_i \rho P_i$ where ρ is the state before measurement and ρ' is the state after measurement. You can see that it can never decrease after each measurement giving the inequality:

$$S(\rho') \geq S(\rho) \qquad (16)$$

Subadditivity

Subadditivity is the property of a function where evaluating the sum of two elements returns something that is less or equal to the sum of the results of the function individually on each element. For quantum systems A and B where ρ^{AB} is a joint state then these subadditivity inequalities apply:

$$S(A,B) \leq S(A) + S(B) \qquad (17)$$

$$S(A,B) \geq |S(A) - S(B)| \qquad (18)$$

The first inequality is known as the subadditivity inequality for Von Neumann entropy. It only holds if and only if $\rho^{AB} = \rho^A \otimes \rho^B$ where A and B have no correlation. The second inequality is known as the triangle inequality (or Araki-Lieb inequality), which is similar to the $H(X,Y) \geq H(X)$ for Shannon entropy.

Concavity of Entropy

As seen earlier classically entropy is a concave function. Here the function is given by:

$$S(\sum_i p_i \rho_i) \geq \sum_i p_i S(\rho_i) \qquad (19)$$

Here you can see that we are taking the sum of the probabilities of states the system could be in. This should be higher than the average entropy of the states.

Entropy of a Mixture of Quantum States

$$\sum_i p_i S(\rho_i) \leq S(\sum_i p_i \rho_i) \leq \sum_i p_i S(\rho_i) + H(p_i) \qquad (20)$$

This formula up above gives the upper bound of the entropy of a mixture of quantum states using the inequality from the concavity of entropy. Basically the average entropy is lesser or equal to the sum of probabilities of states the system could be in which is lesser or equal to the average entropy plus the maximum amount of uncertainty about the current possible state contributes to the system.

Strong Subadditivity

With subadditivity where it concerns the subadditivity and triangle inequalities for two quantum systems it can be extended for three quantum systems.

$$S(A,B,C) + S(B) \leq S(A,B) + S(B,C) \qquad (21)$$

This property is used within quantum information theory. Even though there are import theorems that note:

- Conditioning reduces entropy: $S(A \mid B,C) \leq S(A \mid B)$
- Discarding quantum systems never increases mutual information: $S(A:B) \leq S(A:B,C)$
- Quantum operations never increase mutual information.

Subadditivity of the conditional entropy

Where A,B,C and D composite of four quantum systems then:

$$S(A,B \mid C,D) \leq S(A \mid C) + S(B \mid D) \qquad (22)$$

$$S(A,B \mid C) \leq S(A \mid C) + S(B \mid C) \qquad (23)$$

$$S(A \mid B,C) \leq S(A \mid B) + S(A \mid C) \qquad (24)$$

Monotonicity of the relative entropy

Where ρ^{AB} and σ^{AB} are two density matrices of a composite system AB:

$$S(\rho^{A} \| \sigma^{A}) \leq S(\rho^{AB} \| \sigma^{AB}) \qquad (25)$$

You would expect as if one section of the system was forgotten about then it makes it harder to distinguish between the possible states thus making the relative entropy less (decreasing the distance between them).

References

Quantum Computation and Quantum Information – Micheal A.Nielsen and Isaac L.Chung – Chapter 11

Vedral V., 2002, Rep. Math. Phys. 74, 197, eprint quant-ph/0102094

György Pólya and Gábor Szegő. "Problems and theorems in analysis, volume 1". Springer-Verlag, New York (1976). ISBN 0-387-05672-6.

On Information and Sufficiency, By S. Kullback and R. A. Leibler, Annals of Mathematical Statistics, Volume 22, Number 1 (1951), 79-86.

A Mathematical Theory of Communication, By C. E. SHANNON, The Bell System Technical Journal, Vol. 27, pp. 379–423, 623–656, July, October, 1948.

G53NSC

Non-Standard Computation

Quantum Cryptography

Initial Report

15 March 2010

by

Daniel Nicholas Kiss (dnk07u)

School of Computer Science and Information Technology

University of Nottingham

ABSTRACT

To secretly communicate information is almost ancient as writing itself.[12] Cryptology is the art of creating methods in which way an information could be coded, so that it can only be retrieved by the party to who it was intended.[15] In the era of information technology, many encryption process has been automated and used in many aspects of our lives, including business and personal use. However, the most common methods – at the time of writing this report – are based on preconditions, that some mathematical formula is hard to compute or that no one has such large amount of computational power. Further, by using quantum mechanics, these preconditions can be broken. Hence, a main aim of Quantum Cryptography is to provide encrypting methods, which are not based on such preconditions, but on the laws of physics – which should make them unbreakable.

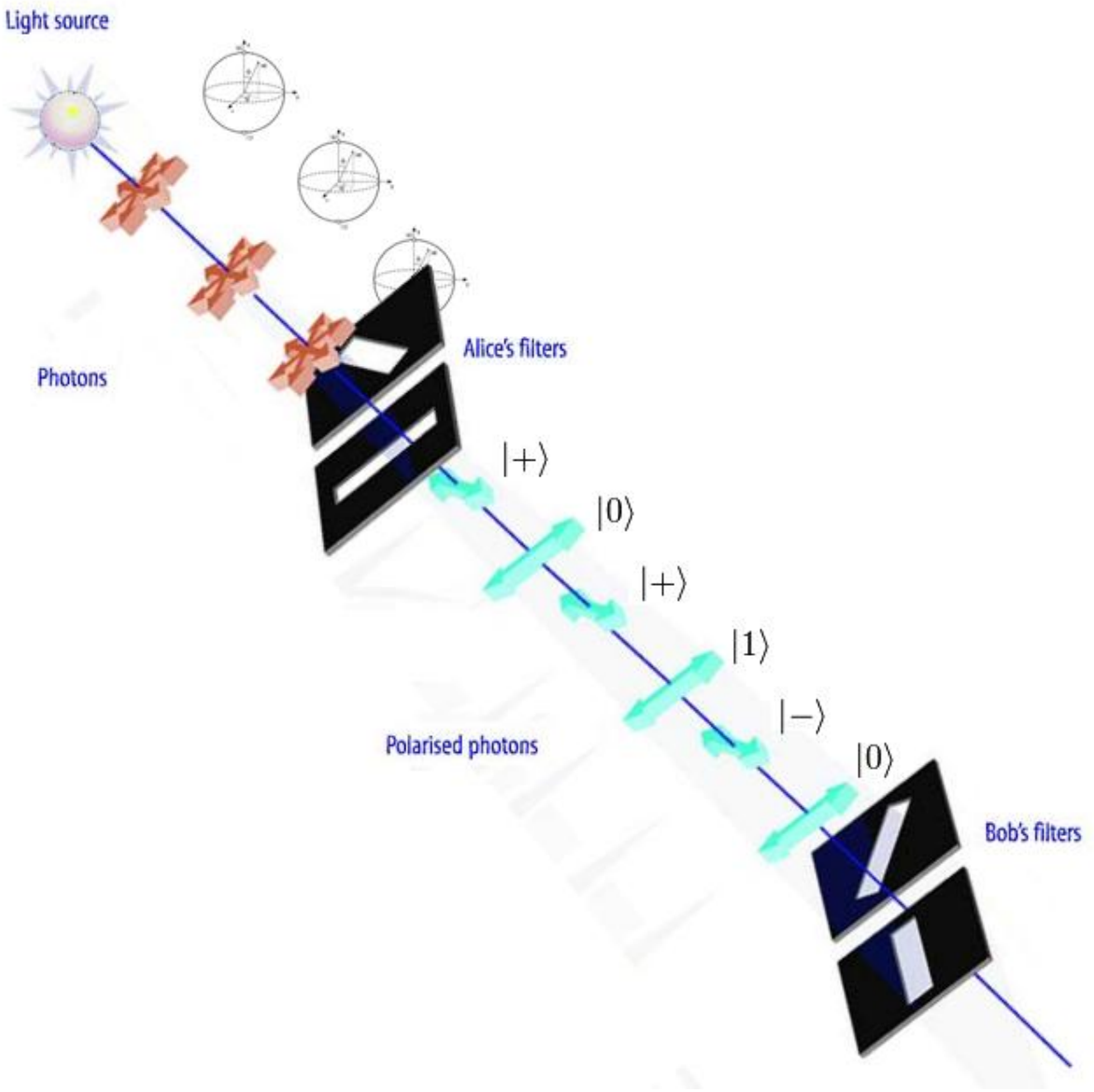


Figure 1 – Sources: New Scientist, Course's webpage

TABLE OF CONTENTS

INTRODUCTION

The aim of this report is to explain the basic concepts of Quantum Cryptography from a computer scientific perspective, with some physics involved – mainly at the parts describing implementations.

There are 3 main categories of cryptological uses of quantum mechanics: 1) which involves cryptanalysis, hence it is used to break cryptographic protocols (e.g.: quantum factoring using Shor's algorithm), and it is not the aim of this report to describe these; 2) which involves protecting classical information by using quantum bits, the main focus of this report is on these applications; and 3) which involves protecting quantum information (e.g.: quantum authentication, secret sharing), and some basic information can be found about this topic at the end of this report.[18]

Cryptography

Cryptography is more than 2400 years old, hence it had enough time to evolve and became verified by mathematics and information theory, thus to actually analyse and construct it from a scientific perspective.

The aim of cryptography is to construct a method, which would allow a message to be transmitted in a public channel, in a way that only its intended recipient would know what the actual content of the message is, even if someone else is able to read the transmission. Its history contains the very ancient scytale, which was basically just a strip of parchment around a narrowing baton; the Caesar cipher, which was a simple constant letter substitution; the polyalphabetic ciphers using for example an Alberti Disk; and the Enigma machine in which the encrypting sequence consisted from a very long period of substitutions. [12],[33]

However, the problem with these mechanisms is the same problem what the currently most common mechanisms have: they are breakable. The ones describe above were actually broken by different ways, and those that we use nowadays are theoretically breakable as well; because they have such preconditions, which might not hold true in the future.

For example, public/private-key cryptosystems are very common, from which the most popular is the RSA (Rivest-Shamin-Adleman). And RSA has a precondition that factorizing large numbers is a difficult / time consuming task, and hence to theoretically broke the system it would require many millions / billions of years depending on the length of the key.[32] However, if someone has unlimited computer power, or just simply comes up with a particularly clever algorithm which would solve factorization in polynomial time; then the private-key could be generated from the public-key, and hence the system could be broken.

At the time of writing this report, we also know that by using Shor's algorithm on a large-enough quantum computer, factorization of large numbers can be done in polynomial time, hence the RSA cryptosystem could be broken.

One-time Pad

Thus, the remaining question is: is there a truly unbreakable crypto-system? And the answer is yes. By using Shannon's theorem it can be proven that following system is truly unbreakable[2]: if we would like to transmit a message represented as a binary string of *n* bits, then using another binary string of *n* bits as the key and applying a bitwise XOR on the two strings we can create an encrypted message, which can only be decrypted by someone who actually knows the key. (The get the plaintext from the encrypted message with the key is simple: apply a bitwise XOR on the two strings of bits, and you would get the original message.)

This method is known for many years, and it is called One-time Pad, however it isn't as popular, because it has the two drastic drawbacks: 1) is that the key have to be very long (at least as long as the message) otherwise the system would be breakable, and 2) is that the key cannot be reused (hence the name), because then by comparing two messages encrypted with the same key some information could be gained.[3]

Quantum Key Distribution (QKD)

To do not compromise security, one must not use public/private-key cryptosystems like RSA, but it is required to use a system which is theoretically proven to be unconditionally secure, like the One-time Pad system.

There are some options how a One-time Pad system could be used: for example the parties could meet in person and exchange a very long string of bits, which then could be used to generate keys from them, whenever they would like to make a secure communication. And when they run out of keys, they would meet again. Alternatively, they could use a trusted secure carrier to do that for them, when the number of unused bits is getting low.

However, it is not just simply inconvenient, but also compromises security (in which case the whole procedure is meaningless), because a third party could get the key while it is at the carrier, or because they both need to store the future keys for a relatively long time, a third party have many chances to get those keys which then could be used to decrypt the messages.

The aim of quantum key distribution is to provide a solution to this problem: by using quantum mechanics and a quantum channel a key could be transmitted (using qubits) between two parties on demand, in a way, that if there is an eavesdropper on the channel then the parties would know about it, and could cancel the transmission. But if they know that the key is securely transmitted – without anyone else being able to look at it – they could use it to encrypt a message, and transmit it on a classical channel, in the same way it has been described for One-time Pad systems.

Hence, in the following protocols described, the quantum mechanics and the quantum channel only used at the key distribution process, but not at the actual message transmission process; thus and eavesdropper is allowed to get the key as long as the parties will know about it, since in that case they can just simply cancel the transmission / retry to distribute the key securely.

SINGLE PARTICLE BASED PROTOCOLS

By tradition, the following sections are going to use the notion of Alice, Bob and Eve; where Alice denotes the sender, Bob denotes the receiver and Eve denotes possible eavesdroppers who have unlimited computational power.

The QKD protocols described in this report exploit the fact, that measuring a qubit in a quantum system causes to disturb the system, hence Eve can be detected. However there is an additional fact, which is exploited, and a categorization of protocols is based on what is this second fact. There are two categories: Single Particle Based and Entanglement Based Protocols. In the followings, this section will describe the first one, and the next section describes the latter one.

Single Particle Based Protocols (also known as Prepare and Measure Protocols[24]) – as their name suggests – using only a single qubit to transmit a bit of the key, however there are different bases, which allows a bit to be encoded differently in a qubit. And this basis is only known by Alice, who encodes the bits; hence when someone would like to get the actually encoded bits, either would need to ask Alice about the bases in which they were encoded or would need to guess.[1] This is the second fact of quantum mechanics that this kind of protocols exploit.

BB84 Protocol

To explain how these protocols actually work, I am going to describe (using examples) some of the most known ones of this category, starting with the first-ever created Quantum Cryptographic protocol by Charles H. Bennett and Gilles Brassard in 1984.[4],[26]

Alice and Bob is connected by both a quantum channel and a public classical channel (e.g.: the Internet). Alice has an arbitrary string of bits – which is going to be the basis of the key – that she would like to transmit to Bob by using qubits on the quantum channel. Alice has two bases which she can use to encode her bits: a normal base: |0> for 0 and |1> for 1, and a Hadamard base: |+> for 0 and |–> for 1 (technically any two orthogonal bases would do, but for simplicity this report will only uses this two bases). Alice then transmits these encoded qubits to Bob, and waits for his acknowledgment on the classical channel.

Alice's Classical Bits	1	0	0	1	0	1	1	0	0
Alice's Base	0/1	+/–	0/1	0/1	+/–	+/–	0/1	+/–	0/1
Alice's Qubit	|1>	|+>	|0>	|1>	|+>	|–>	|1>	|+>	|0>
Bob's Base	0/1	0/1	+/–	0/1	0/1	+/–	+/–	+/–	0/1
Bob's Classical Bit	1	0	1	1	0	1	0	0	0
Matching Bases?	Y	–	–	Y	–	Y	–	Y	Y
Raw Key	1			1		1		0	0

Figure 2.

Whenever Bob receives a qubit, he guesses a base in which it was encoded, measures the qubit according to it, and stores the measured classical bit and also which base it was measured in. (He does that, because it is easier to store a classical bit than a qubit, and it is required to store a relatively large amount of bits.) When Bob receives all the qubits, he sends the acknowledgment to Alice, who then knows that either Bob has her qubits and no one else (as it is proven by the No Cloning), thus Eve lost her chance to measure it, or Bob does not have Alice's qubit (which is then going to be detected); and hence she can safely release – on the public channel – the states in which she encoded the bits; and Bob does the same: announces the bases in which he measured the qubits. For example 0 for a normal base, and 1 for a Hadamard base. However neither of them releases the actual bits they sent/measured, just their bases. Then knowing which bases were the bits encoded/measured, Alice and Bob should be able to construct two identical keys (Raw Key) by ignoring those bits where their bases were different, *see Figure 2*.

Security Proof

The previous description works fine so far, however it does not take into account a possibility of an eavesdropper, Eve. In the following section, it is described what happens if a passive eavesdropper is there, and how the protocol handles it to remain secure.

Eve's aim is to be able to eavesdrop without being detected, hence she has three possibilities that she can try to do: 1) measure the qubit in some way, then pass it to Bob (she might change the base of the qubit or do anything else with it), or 2) store the qubit in a quantum memory (which can be arbitrary large) in order to be able to measure them correctly once the bases are announced, then create and pass some other qubit to Bob, imitating like if it was sent by Alice; and 3) do neither of the previous possibilities in which case she can't have any information about the key. However, in the first two cases, she has to guess the original base of the qubit in order to be able to send it to Bob.

Clearly, once a qubit is created in an arbitrary base, it is impossible to tell which base was used (hence whether a Hadamard gate was used or not); thus if the wrong base is guessed (by a 50% chance) when it is measured the result of the measurement is going to be 0 or 1 in 50%-50% of the cases, hence she will produce a wrong result in 25% of the cases, *see Figure 3*. In the second case, she even has to guess the value of the qubit, hence she will produce an even higher error rate of 50%.

Alice's Classical Bits	1	0	0	1	0	1	1	0	0
Alice's Base	0/1	+/–	0/1	0/1	+/–	+/–	0/1	+/–	0/1
Alice's Qubit	|1>	|+>	|0>	|1>	|+>	|–>	|1>	|+>	|0>
Eve's Base	**0/1**	**+/–**	**+/–**	**+/–**	**0/1**	**+/–**	**0/1**	**0/1**	**+/–**
Eve's Classical Bit	**1**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**0**
Eve's Qubit	**|1>**	**|+>**	**|+>**	**|+>**	**|0>**	**|–>**	**|1>**	**|0>**	**|–>**
Bob's Base	0/1	0/1	+/–	0/1	0/1	+/–	+/–	+/–	0/1
Bob's Classical Bit	1	0	**0**	**0**	0	1	0	0	**1**
Matching Bases?	Y	–	–	Y	–	Y	–	Y	Y
Raw Key	1			**0**		1		0	**1**

Figure 3 – Underline means that two different bases are used.

Because Alice and Bob would like to detect whether Eve is doing something with their qubits or not, they publicly announce and compare a random subpart of their raw key, because they can start with an arbitrary large key anyway (e.g.: half of the bits randomly selected). If too many of the bits do not mach, then they cancel the transmission and re-try to establish a safe secret key (possibly on a different channel). However if the error-rate is below a preset threshold, then they discard the bits compared and continue the protocol with the remaining raw key as described below. From this point on, every communication between Alice and Bob is done on the classical channel, because the theories / techniques they use only effect their classical bits.

Information Reconciliation

Please note, that the original method described in 1992[16] has been improved and modified in some ways, and this report describes this improved version, and hence not the original one.

Because Alice and Bob is required to share the exact same keys, they need to make sure that there are no errors in their keys generated either by noise or by Eve. But in order for the key to remain secure, they need to reveal as little information about it as possible, but it still has to be guaranteed that they have ended up with identical keys.

Hence, Alice and Bob publicly select a random subset of their keys and compare its parity. If an error is found, it means there are odd number of errors in that subset, hence they perform a binary search for the error(s) by selecting smaller-and-smaller random subsets in the part which had the error. Once they have found the error, they discard that bit (or even the last 2 bits, so Eve would have no knowledge of it); and continue checking the parities of random subsets in the whole key again. If there is a disagreement in two random subsets, the probability of finding it by parity checks is 50% (as long as they select a different subset each time); thus they repeat the procedure described above until they know that the total number of possible errors is low enough.[17]

Privacy Amplification

Because Alice and Bob have revealed some parity information about the key, and because Eve might have known some (but not all) bits anyway by measuring them herself, their raw key is not secure enough yet, because Eve could have too much knowledge about it.

Based on the amount of errors found in the compared bits before the Information Reconciliation, on the number of errors found during the parity checks and on the actual number of parity checks made, Alice and Bob can calculate the maximum amount of deterministic or probabilistic (see below) information that Eve can have about the key. Based on this number *k*, Alice and Bob has to choose a compression function *g,* randomly from a predefined set of such functions, in which each function has the characteristic described in the followings, and would have the effect of 'amplifying privacy'.

Hence, each function in this set would produce from *n* number of bits of the original key, *r* number of bits ($\{0,1\}^n \rightarrow \{0,1\}^r$), where $r = n - k - s$, and *k* is the original knowledge of Eve, and *s* is a safety parameter such that Eve's knowledge of the generated hash key would be $(2^{-s}/\ln 2)$ bits.[17] Hence by creating a reasonably smaller hash key from the original raw key, the information of Eve about this new key can be negligence, if *s* is set appropriately.

Probabilistic Knowledge

If instead of Eve guessing the base of Alice's qubit, she first rotates the qubit by -45° degrees around the Y axis, and then she measures the qubit she would have the right value of Alice's classical bit with a ~85.4% chance independently of the base.[17] Because if Alice's classical bit is 0, then her qubit is either $|0\rangle$ (0° from $|0\rangle$) or $|+\rangle$ (90° from $|0\rangle$), and then she rotates the qubit by -45°, then her qubit is going to be either -45° or 45° away from $|0\rangle$, in either case, the probability of measuring 0, is going to be $\cos^2(45°/2)$ = ~85.4%. The case is similar if Alice's classical bit is 1.

Hence, now Eve's overall knowledge of a pair of bits (originally in average she would have guessed 1 base right and 1 wrong) is going to be ~85.4% * ~85.4% = ~72.9% instead of the original 100% * 50% = 50%. However she still has to guess for Alice's original base, to be able to send the qubit to Bob, hence she is still going to be detected. Further, knowing that this is the maximum knowledge of Eve, it is still possible to make an appropriate Privacy Amplification.[17]

Prerequisites / Limitations

The protocol described above (and the ones in the followings) does not indeed have prerequisites about the computational power of Eve, however it has the laws of physics as its prerequisites. Hence they are secure as long as quantum theory holds true.

However, there are some other requirements as well, which mainly is a constraint of the implementation and not the protocol itself[24]:

- Alice's and Bob's encoding / decoding devices must be secure in a sense the Eve must not be able to access it (e.g.: see Trojan horse attack described below).
- Similarly Alice's and Bob's classical memory (and technically their whole inner system should be) must be inaccessible and protected from Eve, even after the message has been sent, thus they must securely destroy the keys they have used, once they no longer need it.
- The selected bases of Alice and Bob most be truly random and trusted (e.g.: a Quantum random number generator), otherwise Eve could just simply calculate their bases.
- They must authenticate each other on the public classical channel, using a protocol known to be unconditionally secure (e.g.: see Man in the middle attack below).

Known Attacks

As it has been described above, the following attacks mainly exploit the imperfection of the implementation, but not the protocol itself; and only a few of them is described. In many implementations a photon is used as a qubit, and two different polarisers are used to handle them.

Man in the middle attack

If the classical channel is not securely authenticated, then it is easy to do a man in the middle attack, where Eve imitates for each communicating party that if she is the other party, and play that role (hence Alice for Bob, and Bob for Alice), this way every communication would go through her, hence she would clearly known their message, and it would no longer be secret. Alternatively, theoretically she could just store the qubits in a large quantum memory, and ask Alice to announce her bases, and once they are measured, she could pass the qubits to Bob.

Because authentication itself usually needs a predefined secret key, it is a good strategy to save and use some part of the newly generated key as the secret key of the next authentication procedure at the next time, when key distribution is needed[23]. However, of course, at the very first time they would need to agree on a secret key manually.

Photon number splitting attack

Because it is difficult to create a photon-source, which would generate exactly 1 photon on demand, in many implementations they usually use laser pulses with an average photon number below than 1 (e.g.: 0.1). It means that in many cases no photon is sent (no pulse created), in some cases 1 photon is generated (which is desired) and in few cases 2 or even more photon is generated. And Eve can use this extra photon to catch and measure it (or if more than 1 photons, then measure in both bases) without being detected, because an original photon from Alice is still passed to Bob.

One defence could be to actually track then number of photons generated, however it is hard without disturbing the system, and by hence collapsing the photon to a normal base. Another approach is done by the redesign of the protocol (see SARG04)[10].

Trojan horse attack

In this attack, once Eve received the qubit of Alice (and before the next one is sent), she sends a large pulse of light to Alice, and the reflected light from Alice's device reveals the state of her polarizer, so Eve can safely measure and pass the qubit to Bob without the need to guess the base.

Denial of service attack

It is relatively easy to cause a denial of service, by just measuring all the qubits or simply blocking the quantum channel; however please not that (usually) the main aim of Eve is not to prevent communication, but to get a hold of some secret information without being detected[2].

Other Limitations (at the time of writing the report)

- Because there is no known method to 'boost' qubits in a quantum channel, the distance between Alice and Bob is limited[2].
- Each connection made must be 1 to 1, hence to connect a whole network of parties it would need a drastic amount of connections (see SECOQC in the Implementations section).
- Quantum Systems are relative expensive and the qubits used (photons / ions) are very 'delicate', hence they must be handled accordingly.

B92 Protocol

According to the B92 Protocol (developed by Charles H. Bennett in 1992) any two non-orthogonal states can be used for QKD, and it is based on that no two non-orthogonal states can be distinguished.[6] In the followings I am going to describe the protocol using an example of the states $|0\rangle$ and its 135° rotation around the Y axis. Please note, that this is my interpretation / example and not part of the original B92 Protocol, and it approximates chances for simplicity.

Alice encodes a classical bit of 0 to the qubit state of |0>, and a classical bit of 1 to the qubit state of (0.08|0> + 0.92|1>) = |135°>. Then transmits this qubit to Bob, who then randomly guesses which of the state the qubit sent is in, and applies a rotation accordingly. Then he measure the qubit: if he measures 0, it means that he has a 87% chance that he guessed the right state and 13% chance that he was wrong, but since he does not know which is the case, he will just discard the qubit. However if he measures 1, it means that there is a 100% chance, that he has was wrong, hence he will write down the opposite bit of the one which was guessed by him, *see Figure 4*.

Alice's Classical Bits	0			1		
Alice's Qubit	|0>			|135°>		
Bob's Guess	0	1		0		1
Bob's Qubit	|0>	|-135°>		|135°>		|0>
Bob's Result	0	0	1	0	1	0
Bob's Classical Bit	-	-	0	-	1	-
Chance	100%/4	15%/4	85%/4	15%/4	85%/4	100%/4

Figure 4 – Case Analysis of B92's example.

At the end of the transmission, Bob announces the bits which he discarded, so that Alice can discard them as well. Bob will measure 0 in 57.5% of the cases, hence the length of their raw key is going to be only 42.5%. Further, similarly to BB84, they can detect whether Eve was measuring their qubits or not, because Eve does not have the opportunity to discard bits (she has to pass every single qubit to Bob in order to do not get caught), hence if she measures 0, but her guess was wrong (in 7.5% of the cases) she will produce an error in Bob's and Alice's raw key; hence by comparing a subpart of their raw keys Alice and Bob can detect if Eve was eavesdropping, and continue the protocol with Information Reconciliation (because of noise and Eve's possible disturbance) and Privacy Amplification as described at the BB84 protocol.

By changing the notion of 1 to another state, it is possible to control how large disturbance will Eve generate, however the larger Eve's disturbance, the larger the chance that Bob will have to discard a bit (measures 0), hence Alice would need to transmit more qubits.

The advantage of B92 over BB84 is that it only uses 2 states instead of the original 4, hence it is easier to implement, however it has the drawback of harder detection of Eve, hence it is harder to implement it in a way that it remains secure.[8]

SARG04 Protocol

The SARG04 Protocol was developed from BB84 so that it would provide a defence against photon number splitting attacks.[11] It had the requirement that it must use the exact same hardware as BB84, hence any changes made, must be in the protocol.

SARG04 indeed handles photon number splitting attacks better, because in SARG04 things work differently in the case when 2 or more photon is generated; however experiments showed that it is more vulnerable in single-photon implementations, hence it is not widely used.[24]

ENTANGLEMENT BASED PROTOCOLS

Entanglement Based Protocols (also known as EPR State Based Protocols[9]) exploits the fact of quantum mechanics, that it is possible to create two maximally entangled particles to which locality does not hold, hence the particles depend on each other no matter how large is the distance between them. Thus these protocols exploit this characteristic of quantum mechanics in addition to the fact that it is impossible to tell which entangled states a pair of qubits is in, without having both the qubits (e.g.: having only one of them); and if one of the qubits of the pair is measured, then the entanglement is broken.

Please note, that this section highlights only the main differences compared to the Single Particle Based Protocols, since for example the same Information Reconciliation and Privacy Amplification procedures are done in these protocols as well, however this is not explained here.

E91 Protocol

In the original E91 protocol (developed by Artur K. Ekert in 1991)[5] Bob starts with an entangled pair of qubits, for simplicity Bob will use the Bell state which gives (0, 0) when using the Bell measurement = Bell(0,0) = $\frac{1}{\sqrt{2}}|00> + \frac{1}{\sqrt{2}}|11>$. Then he keeps one of the qubits, but sends the other one to Alice, who then have four choices depending on the states of two of her classical bits of her key she would like to transmit[13]:

- for the bits of 00, she does nothing → Bell(0,0)
- for the bits of 01, she applies an X rotation → Bell(0,1)
- for the bits of 10, she applies a Z rotation → Bell(1,0)
- and for the bits of 11, she applies both, which is equivalent to applying a Y rotation → Bell(1,1).

And then she send back the qubit to Bob, who will do a Bell measurement on the two qubits: the one that he kept and the one that he received from Alice, and he knows that the result of the this Bell measurement is going to be the 2 classical bits of Alice's original key she would like to transmit[14].

Security Proof

Please note that this section describes my views of the E91 protocol using an example, hence the followings are not strictly part of the original protocol.

If Eve tries to measure the qubit transmitted she will gain no information, because she will get 0 and 1 by 50%-50% chance, hence to actually be able to gain information she would need both qubits, however Bob has the second qubit and he will not release it. They repeat the procedure for every 2 bits of Alice's key; and once their mutual key is long enough they continue the protocol similarly as described for BB84, hence they check the ratio of disturbance (which is caused by either noise or Eve), do Information Reconciliation and finally Privacy Amplification.

In this case Eve's actions will be detected when they check for the ratio of disturbance, however she will have no knowledge of the key at all, but Privacy Amplification is still needed because of the parity bits announced during the Information Reconciliation phase.

Alternatively, Alice and Bob could sacrifice one of their carrying bit capacity (hence the speed of transmission would be its half) in order to detect Eve / be able to produce lower disturbance rate if Eve only measures the qubits. In this situation Alice has only 2 options: for 0 she does nothing, and for 1 she applies an X rotation. In this case the state of the pair is going to be either Bell(0,0) or Bell(0,1); and clearly the second bit shows Alice's original classical bit. However, if there is some disturbance in the system – hence the entanglement is broken (either because Eve has measured the qubit or just simply caused by noise) – the result of the bell measurement will give 1 for the first bit with a chance of 50%. Hence if the first bit is 1, it means that the entanglement is broken and they both will discard that bit, this way they can filter half of the disturbed states. Moreover, if no rotation is made on Alice's qubit (e.g.: it has only been measured), then the second bit of the Bell measurement will still give the right answer, lowering the overall error-ratio, *see Figure 5*.

Alice's Classical Bits	0			1		
The pair's state	Bell (0,0)			Bell (0,1)		
Eve Measures?	–	Y		–	Y	
Bob Measures	(0,0)	(0,0)	(1,0)	(0,1)	(0,1)	(1,1)
Chance	100%	50%	50%	100%	50%	50%
Bob's Classical Bit	0			1		

Figure 4 – Case Analysis of E91's example, when Eve only measures.

However please note that the previous procedure still not provide a defence again denial of service attacks, because Eve could just simply randomly apply an X rotation on Alice's qubit causing them to have the maximal error ratio of 50% (if their error-ratio is higher than 50% and they detect it before the information reconciliation, they can just simply flip all their bits giving them an error-ratio less than 50% = 100% - original error-ratio).

Advantages

The advantage of this protocol over the BB84 protocol is that if Eve measures the transmitting qubit then her disturbance will be detected by 50% chance instead of the BB84's 25% chance.

Drawbacks

For every bit transferred, a qubit has to travel twice the distance between Alice and Bob, because Bob initiates the qubit and then he is the one who receives it eventually; and the maximum distance a qubit can travel is limited by the implementation, hence the maximum distance between Alice and Bob is going to be the half of the maximum distance in the case of BB84.

Ekert Scheme

The original E91 protocol has been modified inspired by the BB84 protocol and a new protocol is created combining techniques used in both protocols.[7] There is a trusted source of maximally entangled pair of qubits between Alice and Bob, which generates such a pair on demand. It is equally probable that measuring a qubit the result is going to be 0 or 1, hence the overall state of the pair should be: $\frac{1}{\sqrt{2}}|00> \pm \frac{1}{\sqrt{2}}|11>$. Then the source sends one of the qubits from this pair to Alice and sends the other one to Bob.

Whenever Alice and Bob receives such a qubit from the trusted source, they randomly chose a base (similarly to BB84), which is either a normal base – do nothing – or a Hadamard base – applying a Hadamard rotation –, and they measure the qubit.[12] They both register the base they chose and the result of the measurement. When the length of the key is long enough the source stops sending the qubits. Because the two qubits are entangled, their result should be the same if they have chosen the same base (but it can be both 0 and 1 by a 50%-50% chance), however if they chose a different base their results are completely independent – hence the qubits act if they would not be entangled. Therefore Alice and Bob publicly announces the base they used to measure the qubits, and they discard the bits which were measured in different bases. The raw key of Alice and Bob is going to be the string of remaining bits, and hence now both Alice's and Bob's string of bits should be identical.

Security Proof

If Eve measures one of the qubits (she guesses the right base by 50% chance), then she will broke the entanglement of the pair, hence even if Alice and Bob chose the same base their result is not guaranteed to be the same (50% chance that it is going to be different), hence by comparing – and then discarding – a subset of their raw keys, they can determine whether Eve was listening. Then they finish the protocol with Information Reconciliation and Privacy Amplification, *see Figure 5*.

Alice's Base	0/1		0/1		+/-		+/-	
Bob's Base	0/1		+/-		0/1		+/-	
Matching Bases / Raw Key?	Y		-		-		Y	
Eve Measures?	-	Y	-	Y	-	Y	-	Y
Chance of Matching results	100%	75%	50%	50%	50%	50%	100%	75%
Chance of Error in Raw Key	0%	25%	-	-	-	-	0%	25%

Figure 5.

Advantages

- The maximum distance between Alice and Bob can be twice as large as it was in BB84, because now they have a trusted source in the middle, and transmission on the quantum channel is only required between the Alice and the source, and between the source and Bob; but it is not required that Alice and Bob should be directly connected to each other by a quantum channel.
- If Alice and Bob don't need the key instantly, they could store the qubits in a quantum memory and they only need to reveal the state of the qubits, when they actually need a secret key. It is more secure, because if the qubits are measured after they arrived by someone else, then it will be detected by Alice and Bob, however if they would need to store the key as classical bits (like in BB84), then those bits could be easily measured without being detected, between the time they have arrived and before they are used and destroyed.[9]

Drawbacks

- It is relatively more difficult to implement this protocol, than the BB84 protocol; because it is harder to create a (trusted) source which can generate maximally entangled pair of bits.

OTHER CRYPTOGRAPHIC APPLICATIONS

There are other uses of quantum mechanics and quantum information theory in the field of cryptography, and this section will introduce the reader to the basic concept / aim of these. They are not explained in details here, however the reader can get more information about these topics by using the references given.

Quantum Authentication

Quantum authentication tackles the problem of authenticating quantum information over a quantum channel, hence to transmit an unchangeable string of qubits in a way that Bob can tell its source is genuinely Alice. Its basic idea is to use a secret key and an error-detecting code selected – based on the key – from a predefined set of codes ,which would detect different kind of errors. Because Eve does not know the key, she does not know which error detecting code will Alice and Bob use, hence if she tries to temper with the key she will get caught. Moreover it is also necessary to encrypt the message with the key in order so that Eve cannot read it, because if she can, she could change signals in the messages without being detected. [20]

Uncloneable Encryption

If Eve is able to read and store an encrypted message, then she will be able to decrypt the message if she finds out the key later on (e.g.: Alice or Bob forgot to destroy the key). Uncloneable Encryption tackles this problem by exploiting the No Cloning Theorem, hence an encrypted strings of qubits sent over a quantum channel similarly as described for Quantum Authentication; and if Bob can verify that the source is genuinely Alice, then they also know that Eve cannot have a copy of the encrypted string, hence she will not be able to decrypt it even if she finds out later what was the key.[21]

Quantum Digital Signatures

A quantum digital signature is used to authenticate a piece of quantum information, which is then sent to many parties (hence there are more than one Bobs), thus the aim of Alice is to produce a digital object (message, document, etc.), where her authority can be verified by the recipients.[24] In order to do that, Alice creates a qubits of strings, where every qubit has a predefined state, and the sequence of these states is going to be her private key. Because once the qubits are generated it is impossible to tell what states the qubits are in, hence she is the only one who can generate this sequence of qubits, which is going to be her public key. Alice then needs to create as many public keys as many recipients she intended to have – because the public key cannot be copied –, but this number has to be limited otherwise Eve could figure out the sates of qubits by testing on many public keys. In order to verify her identity, Alice must release its private key (when all the objects are delivered); and then the recipients (Bobs) can check whether, the digital object they have received is indeed from Alice; but because of that, clearly a key can be used only once.[22]

Quantum Bit Commitment

Quantum Bit Commitment tries to solve the problem of interchanging messages between mutually untrusting parties: Both Alice and Bob would like to send a message to each other, however neither of them would like to release their message first, because then the other party could just simply change its message. Some theories were created which seemed to solve this problem by sending a string of qubits to Bob but each state of the qubit is only known by Alice, hence Bob cannot measure them. However, it turned out, that if Bob cannot read the values of Alice's qubits, it allows Alice to safely change her bits without Bob finding it out (e.g.: by sending qubits of entangled pairs).[19],[29]

Quantum Secret Sharing

Quantum Secret Sharing tackles the problem of trying to distribute a quantum secret information over multiple parties. Hence distributing a secret key among n people in a way that k number of people can reconstruct the secret key, which is a secret string of qubits, however k-1 or less people have no information about the key (where k, $n \in \mathbb{N}$ and $k \leq n$). And it is shown that it possible to solve the problem as long as $n/2 < k$, and the cause of this constraint is the No Cloning Theorem.[18]

IMPLEMENTATIONS

Quantum Cryptography is one of the most advanced field of quantum computation in the sense that there are many physical implementations, and there are some commercial systems as well implementing Quantum Cryptography[28], however these are mainly designed for large institutes (e.g.: governments, banks), and they have some limitations.

Even though implementing such a system is a real challenge – because of the technical difficulties, for example producing a single qubit (e.g.: photon) and then handling it according to its delicate nature[3] –, there are successful implementations, including the first working demonstration of the BB84 protocol done by its authors: it was done in 1992 over a 30 cm distance in open air.[16] And similarly, the first experimental implementation of the E91 protocol was done in 1991 by its author.[12]

Since then it has improved a lot, it is now – at the time of writing this report – possible to reach tens of kilometres of distances with both protocols[27], and the highest rate achieved so far is 1 Mb/s over 20km and 10 kb/s over 100km.[35]

The longest distance achieved using optic fibre is 148.7 km using the BB84 protocol[36]; and the maximum distance achieved through free space is 144 km using the Ekert scheme[37]; which suggests that it is possible to transmit to satellites, because of the much lower atmospheric density at high distances[30]. There are even existing quantum cryptographic networks, for example the 10-noeded DARPA Quantum Network which has been running since 2004[38]. And it has commercial applications as well, for example the first bank transfer using quantum cryptography was made in 2004 in Vienna.[39]

SECOQC

SECOQC is a European project started in 2004, and the computer network created by the project is protected by quantum cryptography. Hence the main aim of the project is to provide an unconditionally secure network and to tackle the problem that the current quantum cryptographic protocols can provide only 1:1 connections. The network implemented by the project was launched in 2008 in Vienna using over 200km of optic fibre.

It supports the research of Quantum Cryptography[23] similarly to the Quantum Cryptography Roadmap[31], and it had its first conference[25] at the time of launching its network in 2008.

SUMMARY

Hence, quantum cryptography provides a theoretically unconditional security[34] – leaving some constraints to the implementations –, and its many implementation shows its feasibility and the real need for such protocols. Its most important use is in Key Distribution where a secret key is need to be shared securely by two parties automatically without the need for a meeting. Since the first publications in 1984, this field has evolved and improved a lot and many research has been done to extend quantum information theory to many other cryptological areas; and it shows that the future of secure communication has its foundation in quantum mechanics, and quantum cryptography will completely replace classical cryptography – even if that will not happen in the near future.

BIBLIOGRAPHY AND REFERENCES

1. Nielsen, Michael A. and Chuang, Isaac L. – *Quantum Computation and Quantum Information* – ISBN0-521-63503-9.
2. Ekert, Artur K.; Zeilinger, Anton and Bouwmeester, Dirk – *The physics of quantum information: quantum cryptography, quantum teleportation, quantum computation* – ISBN3-540-66778-4 .
3. Van Assche, Gilles – *Quantum cryptography and secret-key distillation* – ISBN0-521-86485-2.
4. Bennett, C. H.; Brassard, G.; Breidbart, S. and Wiesner, S. – *Eavesdrop-detecting quantum communications channel* – IBM Technical Disclosure Bulletin, vol. 26, no. 8, January 1984, pp. 4363 - 4366.
5. Ekert, Artur K. – *Quantum cryptography based on Bell's theorem* – Physical Review Letters, vol. 67, no. 6, 5 August 1991, pp. 661 - 663.
6. Bennett, C. H. – *Quantum cryptography using any two nonorthogonal states* – Physical Review Letters, vol. 68, no. 21, 25 May 1992, pp. 3121 - 2124.
7. Bennett, C. H., Brassard, G. and Ekert, A. K. – *Quantum cryptography* – Scientific American, October 1992, pp. 50 - 57.
8. CKI – *The B92 Quantum Coding Scheme* – Quantum Informatics at the University of Aarhus: http://www.cki.au.dk/experiment/qrypto/doc/QuCrypt/b92coding.html.
9. Bennett, C. H. – *Quantum cryptography: Uncertainty in the service of privacy* – Science, vol. 257, 7 August 1992, pp. 752 - 753.
10. Branciard, Cyril; Gisin , Nicolas; Kraus, Barbara and Scarani, Valerio – *Security of two quantum cryptography protocols using the same four qubit states* – Phys. Rev. A 72, 032301 (2005).
11. Fung , Chi-Hang Fred; Tamaki, Kiyoshi and Lo, Hoi-Kwong – *On the performance of two protocols: SARG04 and BB84* – Phys. Rev. A 73, 012337 (2006).
12. Ekert, Artur K. – *Cracking codes* – Plus Magazine, Issue 34, 35; Faculty of Mathematics, University of Cambrdige.
13. BestNeo – *Quantum Cryptography* – Engineering Seminar Topics, June 2008: http://bestneo.com/wp-content/uploads/2008/06/quantum_cryptography.pdf.
14. Ilic, Nikolina – *The Ekert Protocol* – Journal of Phy. 334, 2007.
15. Vittorio, Salvatore – *Quantum Cryptography: Privacy Through Uncertainty* – CSA's Technology Research, October 2002: http://www.csa.com/discoveryguides/crypt/overview.php.
16. Bennett, C. H.; Bessette, F.; Brassard, G.; Salvail, L. and Smolin, J. – *Experimental quantum cryptography* – Journal of Cryptology, vol. 5, no. 1, 1992, pp. 3 - 28.
17. Goldwater, Sharon – *Quantum Cryptography and Privacy Amplification* – SRI International's Artificial Intelligence Center, 12 October 1996: http://www.ai.sri.com/~goldwate/quantum.html
18. Gottesman, Daniel – *Quantum Cryptographic Protocols* – Perimeter Institute, 5 September 2003: http://www.perimeterinstitute.ca/personal/dgottesman/index.html.
19. Gottesman, Daniel and Lo, Hoi-Kwong – *From Quantum Cheating to Quantum Security* – Physics Today, vol. 53, No. 11, p. 22, Nov. 2000.
20. Barnum, Howard; Crepeau, Claude; Gottesman, Daniel; Smith, Adam and Tapp, Alain – *Authentication of Quantum Messages* – Proc. 43rd Annual IEEE Symposium on the Foundations of Computer Science (FOCS '02), pp. 449-458. IEEE Press, 2002.
21. Gottesman, Daniel – *Uncloneable Encryption* – Quantum Information and Computation, vol. 3, pp. 581-602 (2003).

22. Gottesman, Daniel and Chuang, Isaac L. – *Quantum Digital Signatures* – arXiv:quant-ph/0105032 v2, 2001.
23. SECOQC – *White Paper on Quantum Key Distribution and Cryptography* – arXiv:quant-ph/0701168 v1, 2007.
24. Wikipedia – *Quantum Cryptography* – http://wikipedia.org.
25. SECOQC – *Demonstration and International Conference* – 08-10 October 2008: http://www.secoqc.net/html/conference/schedule.html.
26. Bennett, C. H. and Brassard, G. – *Quantum cryptography: Public-key distribution and coin tossing* – Proceedings of IEEE International Conference on Computers, Systems and Signal Processing, Bangalore, India, December 1984, pp. 175 - 179.
27. Gazeau, Jean-Pierre; Nešetřil, Jaroslav and Rovan, Branislav – *Physics and theoretical computer science: from numbers and languages to (quantum) cryptography security* – ISBN1-586-03706-4.
28. Patch, Kimberly and Smalley, Eric – *Quantum Cryptography: Potentially Perfect Security* – Technology Research News, Making the Future report no. 1., 1 December 2002.
29. Salvail, Louis – *The Search for the Holy Grail in Quantum Cryptography* – Lecture notes in computer science, 1999, vol. 1561, pp. 183-216.
30. Grabowski, Tomasz – *The future of cryptography* – 12 May 2003: http://obfusc.at/ed/cryptography_eng.html.
31. Quantum Cryptography Roadmap – *The Theory Component of the QKD and Quantum Cryptography* – Quantum Information Science and Technology Roadmapping Project, Version 1.0, 19 July 2004.
32. International Islamic University Malaysia (IIUM) – *Quantum Security: A New Standard in Cryptography?* – 13 July 2009: http://blogs.iium.edu.my/jaiz/2009/07/13/quantum-security-a-new-standard-in-crytopgraphy.
33. Ekert, Artur K. – *Codes and the Quantum Computer* – Mathematically based videoconferences for schools, 19 May 2005: http://motivate.maths.org/conferences/conference.php?conf_id=61.
34. Stix, Gary – *Best-Kept Secrets* – Scientific American Magazine, January 2005.
35. Dixon, A. R.; Yuan, Z. L.; Dynes, J. F.; Sharpe, A. W. and Shields, A. J.– *Gigahertz decoy quantum key distribution with 1 Mbit/s secure key rate* – Optics Express, Vol. 16, Issue 23, pp. 18790-18979.
36. Hiskett, P. A.; Rosenberg, D.; Peterson, C. G.; Hughes, R. J.; Nam, S.; Lita, A. E.; Miller, A. J. and Nordholt, J. E. – *Long-distance quantum key distribution in optical fibre* – New Journal of Physics 8 (2006) 193.
37. Ursin, R.; Tiefenbacher, F.; Schmitt-Manderbach, T.; Weier, H.; Scheidl, T.; Lindenthal, M.; Blauensteiner, B.; Jennewein, T.; Perdigues, J.; Trojek, P.; Oemer, B.; Fuerst, M.; Meyenburg, M.; Rarity, J.; Sodnik, Z.; Barbieri, C.; Weinfurter, H. and Zeilinger, A. – *Free-Space distribution of entanglement and single photons over 144 km*– Nature Physics 3, 481 - 486 (2007).
38. Knight, Will – *Quantum cryptography network gets wireless link* – New Scientist, Info-Tech, 7 June 2005.
39. Zeilinger, Anton; Vienna University; ARC Seibersdorf research GmbH; City of Vienna; Wien Kanal Abwassertechnologien GmbH and Bank Austria – *World Premiere: Bank Transfer via Quantum Cryptography Basedon Entangled Photons* – 21 April 2004, 11:30, Vienna City Hall - Steinsaal.