RIVF 2006 Tutorial "Quantum Communications"

Prof. Patrick Bellot
PhD. Dang Minh Dung

ENST (Paris) & IFI (Hanoi)

Outline

1. Classical Cryptography
2. Quantum Basics
3. Ideal BB84 Protocol
4. OT and BC Protocols
   - Foundations of Cryptographic Protocols
   - Quantum Primitives
5. Bibliography

The Four Main Goals of Cryptography

- Confidentiality
  - The information should be readable only by the intended receiver. i.e. to protect the information from being eavesdropped.
- Integrity
  - The receiver is able to confirm that a message has not been altered during transmission, i.e. to protect the information from tampering.
- Authentication
  - Any party can check that the other party is who he or she claims to be, i.e. to validate the identity of the other party.
- Non repudiation
  - The sender or the receiver cannot deny what he/she has done.

Modern Cryptography

- We are interested in hiding communications over a public channel.
- Communications have to be encrypted.
- Modern encryption algorithms use encryption keys.
- The algorithms are publicly available.
- Two types of encryption algorithms:
  - Asymmetric algorithms: different keys are used for encryption and decryption.
  - Symmetric algorithm: the same key is used for encryption and decryption.

The characters

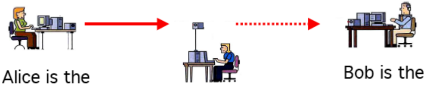
Three main characters:

- Alice: the sender.
- Bob: the receiver.
- Eve: the eavesdropper (spy).

Eve can be passive if she only listens to the communication links.
Eve can be active if she drops messages or modify messages or introduce messages.

The characters

Three main characters, second version:



Alice is the message sender

Eve is the eavesdropper

Bob is the receiver

Alice Cooper the Rock Music Star

Eve according Cranach' the Elder (Germain Painting)

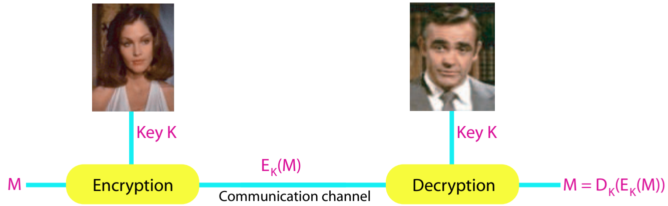
Bob Marley the Reagea Music Star

Alphabet

- M : a message ;
- K : a key ;
- $E_K(\cdot)$: encryption algorithm with key K ;
- $D_K(\cdot)$: decryption algorithm with key K ;
- $E_K(M)$: the cryptogram.
- $M = D_{K'}(E_K(M))$: the decrypted message.

Symmetric Cryptosystems

A cryptosystem is said symetric if the same key is used for encryption and decryption.



Key K

Key K

M

Encryption

$E_K(M)$

Communication channel

Decryption

$M = D_K(E_K(M))$

Symmetric Cryptosystems

- The encryption key must be randomly chosen in the set of allowed keys.
- The key must remain secret to any third party.
- The main question is "How do Alice and Bob share a chosen key ?"
- In a network, another question is "Does every pair of users of the network requires a shared key ?". In case of n users, this makes $\frac{n\times(n-1)}{2}$ keys. For 100 users, it is about 5000 keys, for 1000 users it is about 500 000 keys.

Standard algorithms such as AES, *Advanced Encryption Standard*, or AES are symmetric. They are based on chars transposition and substitution. Keys size is 64 or 128 bits. There is no proof of unconditional security.

Symmetric Key Distribution
Diffie-Hellman

- The Diffie-Hellman allows two users to share a key to be used in symmetric cryptosystems.
- Diffie-Hellman based algorithms are used in most of the commercial products: SSH, HTTPS, etc.
- Diffie-Hellman:
  - Alice and Bob publicly agree on a prime number p and a primitive a of p, i.e.:

    $$\forall b \in [1, p-1], \exists g \text{ s.t. } a^g \equiv \ b \ mod \ p$$

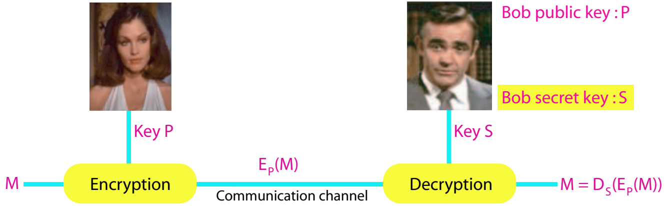
    .
  - Alice randomly chooses $x_A \in [1, p-1]$ and publishes $y_A = a^{x_A} \ mod \ p$.
  - Bob randomly chooses $x_B \in [1, p-1]$ and publishes $y_B = a^{x_B} \ mod \ p$..
  - The key is $K = y_B^{x_A} \ mod \ p = x_A^{y_B} \ mod \ p$.
- The security is based on the intractability of computing discrete logarithms in $\mathbb{Z}/p\mathbb{Z}$.

Asymmetric Cryptosystems

Also called Public Key Cryptosystems

- Each user u has two keys:
  - A public key P_u which is publicly available.
  - A private key S_u which is only known by the user.
- Messages are encrypted using P_u.
- Messages are decrypted using S_u.
- $M = D_{S_u}(E_{P_u}(M))$
- Knowing P_u must not allow to deduce S_u.

Alice sends a message to Bob:



Bob public key : P

Bob secret key : S

Key P

Key S

M

Encryption

$E_P(M)$

Communication channel

Decryption

$M = D_S(E_P(M))$

Asymmetric Cryptosystems

Also called Public Key Cryptosystems

- RSA, *Rivest, Shamir, Adleman* is an example of widely used asymmetric cryptosystem.
- Each user u generates its pair of keys (P_u, S_u) according to an algorithm based on large prime numbers, for instance.
- Keys are 512, 1024 or more bits.
- Asymmetric encryption is about 1000 times slower that symmetric encryption.
- Because:
  - There is no proof of unconditional security.
  - Keys have to be renewed on a regular basis.
  - Eve can publish a key under the name of Bob.

  There is a need for a PKI, *Public Key Infrastructure*, which role is to certify keys and to maintain the list of revocated keys.

Pros and Cons

- Classical cryptography tools are widely used to provide confidentiality of communication, authentication of originator, integrity of messages and so on.
- The security of classical cryptography is based on the assumption that some mathematical problems are intractable: factoring large integers, computing discrete logarithms, etc.
- There is no formal proof of security.
- If we can reasonnably assume that a single user cannot break classical cryptography, what about governmental organizations ?

Pros and Cons

- Every advance in code-making (cryptography) has been defeated by advances in code-breaking (cryptanalysis).
    - German Enigma Machine with 10 million billion possible combinations (keys) looked unbreakable.
    - Allied breaking machine "Bombe" broke it.
- Factoring large integers (RSA) or computing discrete logarithms (Diffie-Hellman) look like hard problems. What about heuristics ?
- If you discover a way of solving one of these problems, what would you do ? Will you publish it ?
- Unanticipated advances in hardware:
    - 1946: Eniac, first electronic computer.
    - 1978: Apple II, 64 Kb, 5MHz.
    - 2005: PowerBook G4, 2 Gb, 1.7 GHz.
    - 2050: ?
    - 2100: Quantum Computers ?

Pros and Cons

Cryptography is used in:

- E-commerce and e-business.
    - No one with a Cray computer would be interested in your €300 transaction :-)
- Military and diplomatic applications.
    - Current encryption may be breakable in 2030 :-(
- Worlwide companies applications.
    - Secret services may help national companies :-(
    - Use of other methods ?

How do you trust classical cryptography ?

The Quantum Threat

- Quantum Computers are theoretical computers processing quantum bits (qubits).
- There exists an algorithm, Peter Shor's algorithm, for factoring large integers.
- Running Shor's algorithm on a quantum computer would allow to break classical cryptography.
- Fortunately, nobody knows if quantum computers can be built. The more optimistic views require at least 30 years.
- Moreover, it is not clear that Shor's algorithm can be executed on quantum computers...

Unconditional Security

Unconditional security, also called perfect secrecy, means security against the eavesdropper Eve, no matter what computing power she has, even Quantum Computers, and no matter how much time she has.

- The appropriate definition of unconditionnal security uses the notion of entropy of the theory of information.
- Vernam ciphering is an example of unconditionally secure cryptosystem.
- Except Vernam ciphering, no classical cryptosystem is proven unconditionnally secure. That means that a document currently considered as secret could be broken in a few years.

Vernam Ciphering

Washington talks to Kremlin...

- Modern Vernam ciphering uses the XOR operation:

$$\begin{cases} 0 & \oplus & 0 & = & 0 \\ 0 & \oplus & 1 & = & 1 \\ 1 & \oplus & 0 & = & 1 \\ 1 & \oplus & 1 & = & 0 \end{cases}$$

  which has the property : $(x \oplus y) \oplus y = x$.
- Let $A \equiv a_1 \cdots a_n$ and $B \equiv b_1 \cdots b_n$ be two strings of n bits, we generalize the XOR operation: $C = A \oplus B$ with $C \equiv c_1 \cdots c_n$ and $c_i = a_i \oplus b_i$ for $i \in [1, n]$.
- We have $(A \oplus B) \oplus B = A$.

Vernam Ciphering

Washington talks to Kremlin...

The ciphering:

- Let M be the message, a string of n bits.
- Let K be the key, a string of n bits.
- $E_K(M) = M \oplus K$, $D_K(E_K(M)) = E_K(M) \oplus K = M$.

Properties:

- As far as the key is used only once, the Vernam ciphering is unconditionally secure. *Proof later.* Vernam ciphering is also called one-time pad.
- Vernam ciphering needs a new key for each message. Thus the main problem is the key distribution.

Knowing an encrypted message only does not give any information on the clear message or, equivalently, on the key. Every key applied to the encrypted message gives a possible clear message. Even with infinite computing power and infinite time, one cannot decode the message.

Information theory

Entropy

Information theory was introduced by Shannon (1948). One of the most interesting (for us) notion is that of entropy to measure the uncertainty on the output of a random variable.

Example. The random variable is the output of a coin flipping.

1. If the coin is regular, 50% head and 50% tail, then the uncertainty is maximal.
2. If the coin is not regular, 25% head and 75% tail for instance, then the uncertainty is less then in case 1.
3. If the coin belongs to David Copperfield, 0% head and 100% tail for instance, then the uncertainty is null.

The entropy allows to measure this uncertainty.

Information theory

Entropy

Let $n > 1$ and X be a random variable with possible values $x_1, \ldots, x_n$ occuring with respective probabilities $p_1, \ldots, p_n$, the entropy of X is defined by:

$$H(X) = \sum_{i=1}^{n} -p_i \log(p_i)$$

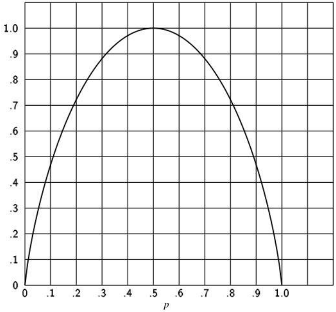
Example:

- A coin, 50% head and 50% tail:
  $H(X) = -0.5 \times \log(0.50) - 0.5 \times \log(0.50) = 1$
- A coin, 25% head and 50% tail:
  $H(X) = -0.25 \times \log(0.25) - 0.75 \times \log(0.75) \simeq 0.8$
- A coin, 0% head and 100% tail:
  $H(X) = -1 \times \log(1) = 0$

Information theory

Entropy of a binary variable

A binary variable is a variable that can take two values, 0 and 1 for instance, with probability p and $1-p$. The following curve describe the variation of the entropy in function of p:



This function is usually named the binary entropy h_2:

$$h_2(p) = -p \times \log(p) - (1-p) \times \log(1-p)$$

Information theory
Entropy of a n-valued variable

In the general case, X is a variable with n possible values $x_1, \ldots, x_n$ with repective probabilities $p_1, \ldots, p_n$. We have that:

- $H(X)$ is always positive.
- $H(X) = 0$, the minimal value, if and only if, all p_i but one are zero, this one being equal to 1.
- $H(X)$ is maximal if and only if all p_i are equal to $1/n$. In this case:

$$H(X) = \sum_{i=1}^{n} -\frac{1}{n} \log\left(\frac{1}{n}\right) = \sum_{i=1}^{n} \frac{1}{n} \log(n) = \log(n)$$

.

Example. Let X be a variable which output are binary strings of $k \geq 0$ bits. There is 2^k values. The maximum entropy is reached when all values have equal probabilities and is $H(X) = \log(2^k) = k$.

Information theory

Entropy with two variables

Let X be a n-valued variable and Y be a m-valued variable:

- $H(X|Y)$ is the entropy of X assumed that Y is known. $H(X|Y) = H(X,Y) - H(X)$.
- $H(X,Y)$ is the joint entropy, i.e. the entropy of $Z = (X,Y)$ considered as a single variable.
- $I(X;Y)$ is the mutual information. It measures the statistical dependance between X and Y.

We have:

$$\begin{aligned} I(X;Y) &= H(X) + H(Y) - H(X,Y) \\ &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \end{aligned}$$

Information theory

Entropy with two variables

Bayes formula: $p(U,V) = p(V|U) \times p(U)$. Thus:

$$\begin{aligned} & H(X,Y) \\ = \; & -\sum_{x_i,y_j} p(x_i,y_j) \times \log(p(x_i,y_j)) \\ = \; & -\sum_{x_i,y_j} p(x_i,y_j) \times \log(p(x_i|y_j) \times p(y_j)) \\ = \; & -\sum_{x_i,y_j} p(x_i,y_j) \times (\log(p(x_i|y_j) + \log(p(y_j))) \\ = \; & -\sum_{x_i,y_j} p(x_i,y_j) \times \log(p(x_i|y_j) - \sum_{x_i,y_j} p(x_i,y_j) \times \log(p(y_j) \end{aligned}$$

but:

$$\begin{aligned} & -\sum_{x_i,y_j} p(x_i,y_j) \times \log(p(y_j) \\ = \; & -\sum_{y_j} \left(\sum_{x_i} p(x_i,y_j) \times \log(p(y_j))\right) \\ = \; & -\sum_{y_j} \left(\sum_{x_i} p(x_i,y_j)\right) \log(p(y_j) \\ = \; & -\sum_{y_j} p(y_j) \log(p(y_j)) \\ = \; & H(Y) \end{aligned}$$

thus:

$$H(X,Y) = H(X|Y) + H(Y) = H(Y|X) + H(X)$$

.

If X and Y are independant, $p(X,Y) = p(X) \times p(Y)$ and $H(X,Y) = H(X) + H(Y)$.

Unconditional security

Back to Vernam cipher

We have:

- A message M.
- A key K.
- A cryptogram $C = E_K(M)$.

Before all, M and K have entropy $H(M)$ and $H(K)$. Eve, the eavesdropper, has access to C. Unconditional security means that knowing C does not give any information about K:

$$H(K|C) = H(K) \text{ or equivalently } H(M|C) = H(M)$$

Thus, whatever computation power and time she has, she will not be able to discover anything because there is nothing to discover.

A necessary condition for unconditional secrecy is that $H(K) \geq H(M)$.

Unconditional secrecy of Vernam cipher

We have:

- M, the message, is a random string of k bits.
- K, the key, is a random string of k bits.
- $C = K \oplus M$ is a string of k bits.
- $K = C \ominus M$.

Given any k, since the key is randomly chosen:

$$p(K = k) = \frac{1}{2^k}$$

Then:

$$p(M = m | C = c) = \frac{p(M = m, C = c)}{p(C = c)} \quad \text{[Bayes formula]}$$

Unconditional secrecy of Vernam cipher

Continued...

We have:

$$\begin{aligned} p(M=m, C=c) &= p(M=m, K=c \ominus m) \\ &\qquad [M \text{ and } K \text{ are independant}] \\ &= p(M=m) \times p(K=c \ominus m) \\ &= p(M=m)/2^k \end{aligned}$$

And:

$$\begin{aligned} p(C=c) &= \textstyle\sum_m p(C=c, M=m) \times p(M=m) \\ &= \textstyle\sum_m p(K=m-c)p(M=m) \\ &= \textstyle\sum_m \frac{p(M=m)}{2^k} \\ &= \frac{1}{2^k} \end{aligned}$$

Unconditional secrecy of Vernam cipher

Continued...

Thus:

$$\begin{aligned} p(M=m|C=c) &= \frac{p(M=m,C=c)}{p(C=c)} \\ &= \frac{\frac{p(M=m)}{2^k}}{\frac{1}{2^k}} \\ &= p(M=m) \end{aligned}$$

And finally:

$$H(M|C) = H(M)$$

Knowing the cryptogram gives no information on the message.

Summary

1. Classical cryptography is widely used but not proven unconditionally secure.
2. Classical cryptography is threatened by Quantum Computers.
3. Information theory allows to formalize secrecy.
4. Vernam cipher is proved unconditionnally secure.
5. The main problem is key distribution.

Quantum Basics

Not a course...

Associated to any isolated physical system is a complex vector space with inner product, a Hilbert space, known as the state space of the system.

The state of the system is completely described by its state vector, a unit vector in the state space. Such vectors are usually written $|\psi\rangle$, $|\phi\rangle$, etc.

The evolution of a closed quantum system is described by a unitary transformation, i.e. $|\psi'\rangle = U|\psi\rangle$. U is a unitary matrix.

Quantum Basics

Example : a polarized photon

From Wikipedia.org.

The simplest manifestation of polarization to visualize is that of a plane wave where the direction of the magnetic and electric fields are confined to a plane perpendicular to the propagation direction. Simply because the plane is two-dimensional, the electric vector in the plane at a point in space can be decomposed into two orthogonal components. For a simple harmonic wave, where the amplitude of the electric vector varies in a sinusoidal manner, the two components have exactly the same frequency. However, these components have two other defining characteristics. First, the two components may not have the same amplitude. Second, the two components may not have the same phase, that is they may not reach their maxima and minima at the same time in the fixed plane.

RIVF2006

Quantum Basics

Example : a polarized photon

RIVF 2006 Tutorial "Quantum Communications"

Prof. Patrick Bellot
PhD. Dang Minh Dung

Classical Cryptography

Quantum Basics

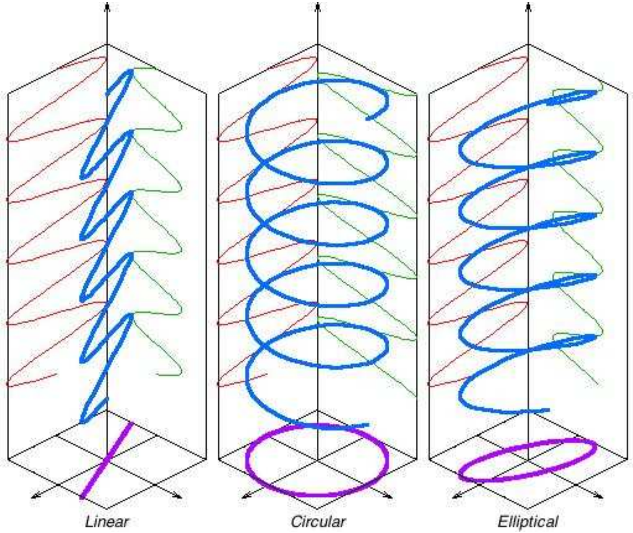
Ideal BB84 Protocol

OT and BC Protocols

Foundations of Cryptographic Protocols

Quantum Primitives

Bibliography

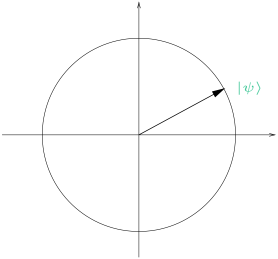


Quantum Basics

Example : a polarized photon

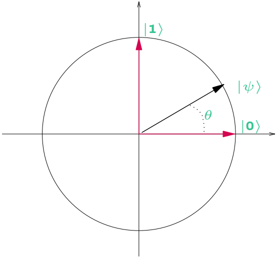
Polarization states are often specified in terms of the polarization ellipse. A common parameterization uses the azimuth angle, ψ (the angle between the major semi-axis of the ellipse and the x-axis) and the ellipticity, ϵ (the ratio of the two semi-axes). Full information is also provided by the amplitude and phase of oscillations .

The polarization of a photon is represented by the azimuth angle. This can be in turn represented by a unitary vector in a 2-dimensional Hilbert space:



Quantum Basics
Example : a polarized photon

At this level, the measurement of the polarization of a photon can be viewed according to an orthogonal measurement basis. For instance the basis $\{|0\rangle, |1\rangle\}$ where $|0\rangle$ is the unit vector with angle 0 and $|1\rangle$ is the unit vector with angle $\pi/2$.



The result of the measurement will be $|0\rangle$ with probability $cos^2\theta$ and $|1\rangle$ with probability $sin^2\theta$. And the photon is modified according to the result of the measurement.

Quantum Basics

Example : a polarized photon

Distinguishing two orthogonal polarizations is possible. For instance, let us assume that photon is polarized either with angle θ or with angle $\theta + \pi/2$. It suffices to measure the polarization according to the basis $\{|\theta\rangle, |\theta + \pi/2\rangle\}$.

Conversely, distinguishing two non orthogonal polarizations is impossible. For instance, let us assume that we want to distinguish $|0\rangle$ and $|\pi/4\rangle$ and that we do a measurement with the basis $\{|0\rangle, |1\rangle\}$. The following table describes the result:

Photon polarization	Result of the measure
$\lvert 0\rangle$	$\lvert 0\rangle$ in 100% of the cases
$\lvert \pi/4\rangle$	$\lvert 0\rangle$ in 50% of the cases $\lvert 1\rangle$ in 50% of the cases

If the result of the measurement is $|1\rangle$, we are sure that the polarization is $\pi/4$. Otherwise, we are not sure.

Quantum Basics

Non-cloning theorem

One of the main result used to guaranty the secrecy of communications is the Non-cloning theorem :

Non-cloning theorem (1982): It is impossible to duplicate an unknown quantum state.

Let us assume that we can clone quantum state:
$|\psi\rangle \longrightarrow |\psi\rangle\ |\psi\rangle$ (tensor product)
$|\phi\rangle \longrightarrow |\phi\rangle\ |\phi\rangle$
And:
$|\psi\rangle + |\phi\rangle \longrightarrow (|\psi\rangle + |\phi\rangle)\ (|\psi\rangle + |\phi\rangle)$ (superposition)
Thus:
$|\psi\rangle + |\phi\rangle \longrightarrow |\psi\rangle\ |\psi\rangle + |\phi\rangle\ |\phi\rangle$
Therefore:
$|\psi\rangle\ |\phi\rangle + |\phi\rangle\ |\psi\rangle = 0$ for all $|\psi\rangle$, $|\phi\rangle$
That's impossible.

Quantum Basics

Non-cloning theorem

Now, let us assume that an information is encoded using quantum states.

For the eavesdropper, Eve, the quantum state is unknown.

Thus : Eve cannot duplicate the information.

Quantum Basics

Measurements

Another important principle of Quantum Physics is:

One cannot take a measurement without perturbing the system.

More precisely, as mentionned before, the quantum state is modified according to the result of the measurement.

The consequence is that if the eavesdropper, Eve, try to intercept the communications and to measure them, then she pertubs the information.

The BB84 Protocol

Introduction

The idea of Quantum Cryptography (QC) was first proposed in the 1970s by Stephen Wiesner and then by Charles H. Bennett and Gilles Brassard in 1984 at the university of Montréal, hence the name BB84.

> *"... and it may well be doubted whether human ingenuity can construct an enigma of the kind which human ingenuity may not, by proper application resolve."*

Edgar Allan Poe
The Gold Bug, Tales of Mystery and Ratiocination, 1943

Can we, today, invalidate the Poe's prediction ?

The BB84 Protocol

Introduction

The main Quantum Physics principles used in BB84 are:

- If one read an unknown quantum information (measures it), then there is a great probability that the information is modified, for instance if the basis used for the measurement is different from the basis used to produce the information.
- An unknown quantum information cannot be duplicated, that is to say that if you don't know the encoding basis for an information, then you are unable to duplicate this information.

The BB84 Protocol

Introduction

- Most of the results of Quantum physics are true with a very high probability. For instance, if the photon is polarized in the $\oplus$-basis and you measure it with the $\oplus$-basis, then you get the correct result with a very high probability, something very close to 1 but not 1.
- Most of the Quantum apparatus, for producing or measuring quantum quantum states or for transporting them are not perfect.

Describing the BB84 protocol in the ideal case where all "very high probabilities" are 1 and all apparatus are perfect, is very simple. However, it does not correspond to any reality.

The BB84 Protocol
Introduction

Quantum Cryptography (QC) is unproperly named.

The proper name should be:

Quantum Key Distribution (QKD)

or "Quantum Key Establishment" because the goal of BB84 is to establish a random secret key between Alice and Bob.

Then, this key can be used for many purposes. For instance, it can be used with Vernam encoding to guaranty an unconditionally secure transmission of information. Hence the name Quantum Cryptography.

The BB84 Protocol

Information encoding

BB84 uses quantum states in a quantum system of dimension 2.

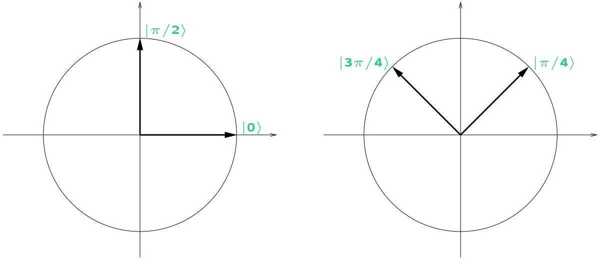
As we have seen, polarization of photons is a 2-dimensional system. And that is the quantum system that we will use for the description of BB84.

In this space, we choose two orthogonal bases which are maximally conjugate. That is two bases oriented so that a measurement in one randomizes the measurement in the other. Maximaly conjugate means that randomness is maximum.

The BB84 Protocol

Information encoding

Two maximally conjugate bases:



- The first basis is $\oplus = \{|0\rangle, |\pi/2\rangle\}$, also written $\oplus = \{|0\rangle, |1\rangle\}$.
- The second basis is $\otimes = \{|\pi/4\rangle, |3\pi/4\rangle\}$, also written $\otimes = \{|\overline{0}\rangle, |\overline{1}\rangle\}$.

The BB84 Protocol

Information encoding

These two bases are maximally conjugate because:

- If you measure the polarization of a photon in the base $\otimes = \{|\pi/4\rangle, |3\pi/4\rangle\}$, you get a photon that is polarized either $|\pi/4\rangle\rangle$ or $|3\pi/4\rangle$.
- Then, if you measure this photon in the base $\oplus = \{|0\rangle, |\pi/2\rangle\}$, you get either $|0\rangle$ or $|\pi/2\rangle$, each with a probability of 50%. That is to say that the result is totally random.
- If you interchange the roles of the bases $\oplus$ and $\otimes$, the same reasonning applies.
- That is the definition of two maximally conjugate bases.

The BB84 Protocol

Information encoding

Encoding of bits: bits, 0 or 1, are encoded using quantum states. For each bit, there is two possible quantum states:

- A bit 0 is encoded either by $|0\rangle$ or $|\overline{0}\rangle$.
- A bit 1 is encoded either by $|1\rangle$ or $|\overline{1}\rangle$.

If a bit is encoded using $|0\rangle$ or $|1\rangle$, we say that it is encoded in the $\oplus$-basis.

If a bit is encoded using $|\overline{0}\rangle$ or $|\overline{1}\rangle$, we say that it is encoded in the $\otimes$-basis.

The BB84 Protocol

Qubits

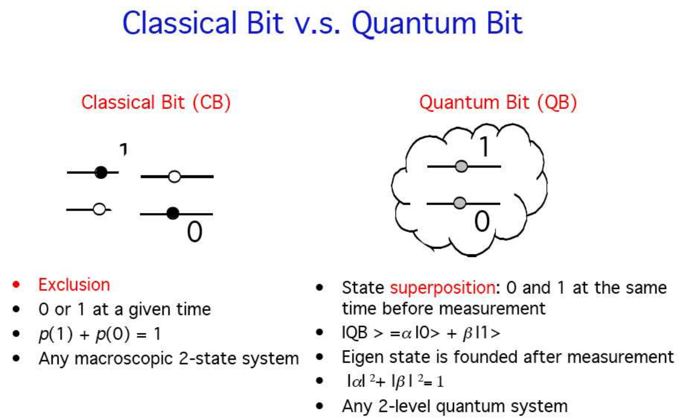
If a bit $b \in \{0, 1\}$ is encoded in a basis $\beta \in \{\oplus, \otimes\}$ giving a quantum state $q = |b_\beta\rangle$ with $b_\beta \in \{b, \overline{b}\}$, then knowing β and q allows to recover the value of the bit b.

The encoding of a bit as a quantum state is simply called a qubit or quantum bit.

We will see some other means for bit encoding using quantum states. All these bit encodings can also be used by BB84.

The BB84 Protocol

Qubits



Classical Bit v.s. Quantum Bit

Classical Bit (CB)

- Exclusion
- 0 or 1 at a given time
- $p(1) + p(0) = 1$
- Any macroscopic 2-state system

Quantum Bit (QB)

- State superposition: 0 and 1 at the same time before measurement
- IQB > =α I0> + β I1>
- Eigen state is founded after measurement
- $|\alpha|^2 + |\beta|^2 = 1$
- Any 2-level quantum system

The BB84 Protocol
Two channels

The BB84 protocol uses two communication channels:

- The quantum channel for transporting the quantum states, i.e. the qubits. This could be an optic fiber or a free-space laser beam. This is a one-way channel from Alice to Bob.
- The clasical channel for transporting classical bits. This can simply be Internet or any radio communation. This is a two-way channel between Alice and Bob.

The BB84 Protocol

Phases of the protocol

The protocol BB84 allows Alice and Bob to share a secret random key. It has five main steps:

- Sifting. This step use quantum communications in order to establish a raw or sifted key.
- Eavesdropper detection. This very simple step detects if someone was spying the communications.
- Bit reconciliation. This step allows to correct the quantum apparatus errors.
- Privacy amplification. To allow to reduce the Eve's information to a vanishing part if she was not detected.
- Authentication. The two parties authentify themselves.
- Secret communication of data.

The BB84 Protocol
Sifting

Sifting (Alice's part). During the phase, Alice chooses a big number of bits N. Then:

- She randomly chooses N bases $(\beta_i)_{1 \leq i \leq N} \in \{\oplus, \otimes\}$.
- She randomly chooses N bits $(b^i)_{1 \leq i \leq N} \in \{0, 1\}$.
- She sends the N quantum states $|b^i_{\beta_i}\rangle$, $1 \leq i \leq N$ to Bob.

Example: $N = 8$

Random bases:	$\otimes$	$\oplus$	$\oplus$	$\otimes$	$\oplus$	$\oplus$	$\otimes$	$\otimes$
Random bits:	1	1	0	1	1	0	0	0
Q states sent:	$\vert\overline{1}\rangle$	$\vert 1\rangle$	$\vert 0\rangle$	$\vert\overline{1}\rangle$	$\vert 1\rangle$	$\vert 0\rangle$	$\vert\overline{0}\rangle$	$\vert\overline{0}\rangle$

The BB84 Protocol

Sifting

Sifting (Bob's part). Bob receives the quantum states $(b^i_{\beta_i})_{1 \leq i \leq N}$ and he has to measure them. But for each quantum state, he does not know the basis used for the bit encoding.

Thus, Bob has to randomly guess the N bases. For N very large, he will be wrong 50% of the cases:

- If he measures a quantum state with the wrong basis, he gets a random result.
- If he measures a quantum state with the good basis, he gets the good value with a very high probability.

Example continued:

Alice's random bases:	$\otimes$	$\oplus$	$\oplus$	$\otimes$	$\oplus$	$\oplus$	$\otimes$	$\otimes$
Alice's random bits:	1	1	0	1	1	0	0	0
Quantum states sent:	$\lvert\overline{1}\rangle$	$\lvert 1\rangle$	$\lvert 0\rangle$	$\lvert\overline{1}\rangle$	$\lvert 1\rangle$	$\lvert 0\rangle$	$\lvert\overline{0}\rangle$	$\lvert\overline{0}\rangle$
Bob's random bases:	$\otimes$	$\oplus$	$\otimes$	$\otimes$	$\otimes$	$\otimes$	$\oplus$	$\otimes$
Bob's results:	1	1	?	1	?	?	?	0

The BB84 Protocol

Sifting

Sifting (Bases agreement). Alice and Bob agrees on the bases they used. Each one knows on which quantum states the bases chosen by Alice and Bob were the same. On these quantum states, the bit obtained by Bob was the bit encoded by Alice. The sequence of these bits is called the raw or sifted key. The other quantum states and their results are discarded.

Note that the bases agreement communications are done in clear and eavesdropable channel such as ordinary Internet.

Example continued:

Alice's random bases:	$\otimes$	$\oplus$	$\oplus$	$\otimes$	$\oplus$	$\oplus$	$\otimes$	$\otimes$
Alice's random bits:	1	1	0	1	1	0	0	0
Quantum states sent:	$\vert\bar{1}\rangle$	$\vert 1\rangle$	$\vert 0\rangle$	$\vert\bar{1}\rangle$	$\vert 1\rangle$	$\vert 0\rangle$	$\vert\bar{0}\rangle$	$\vert\bar{0}\rangle$
Bob's random bases:	$\otimes$	$\oplus$	$\otimes$	$\otimes$	$\otimes$	$\otimes$	$\oplus$	$\otimes$
Bob's results:	1	1	?	1	?	?	?	0
Shared raw key:	1	1		1				0

The BB84 Protocol

Eavesdropper detection

The eavesdropper Eve may use an intercept-resend strategy.

That means that:

- Eve measures the photon between Alice and Bob.
- Eve resends a photon corresponding to its measure.

However, when the photon reaches Eve, she is not aware of the encoding basis, either $\oplus$ or $\otimes$.

The BB84 Protocol

Eavesdropper detection

Thus, Eve has to guess the measurement basis for each photon.

Eve randomly chooses the bases for measuring the photon.

She will be right 50% of the cases and she will be wrong 50% of the cases.

The BB84 Protocol

Eavesdropper detection

When Eve is right, for instance, she chooses to measure according to the $\oplus$-basis while the bit was encoding using the $\oplus$-basis, then:

- she gets the correct answer with a very high probability and
- the quantum state is not pertubated.

The BB84 Protocol

Eavesdropper detection

When Eve is wrong, for instance, she chooses to measure according to the $\oplus$-basis while the bit was encoding using the $\otimes$-basis, then:

- she gets a random answer and
- the quantum state is pertubated.

The photon resent by Eve is not the same as the photon sent by Alice.

The BB84 Protocol

Eavesdropper detection

- Let us assume that the quantum world is perfect, i.e. there is no errors in measurements...
- If Eve is not present and Alice sent N qubits, the length of the sifted key is $N/2$.
- If Eve is present, she got 50% of the sifted key, i.e. $N/4$ bits. And she perturbed the other $N/4$ bits.
- When Bob measured these $N/4$ perturbed bits, he got random results. 50% of these random results, i.e. $N/8$ were correct, the other $N/8$ were erroneous.
- That means that if Eve is present, there is an error rate of 25% in the sifted key.

The BB84 Protocol
Eavesdropper detection

- Thus, in a perfect word, if Eve measures all photons, this introduces an error rate of 25% in the sifted key.
- If Eve measures only 10% of the qubits, the error rate will be only 2.5%.
- To detect Eve, Alice and Bob publicly checks a subset of the sifted key.
- These bits then, have to be discarded. They are lost.

The BB84 Protocol

Eavesdropper detection

- Alice and Bob have to choose a detection threshold, for instance 5%:
  - If the error rate is greater than the threshold, then Eve is detected and the key exchange session is aborted.
  - If the error rate is less than the threshold, then the session continue. Eve may have got some of the bits. The privacy amplification will handle this.
- Practically, the world is not perfect. The measures are not perfect. The quantum channel is not perfect.
- Consequently, there exists an intrinsic error rate which has to be evaluated by running the system and measuring this average error rate when no eavesdropper is present.

The BB84 Protocol

Error correction

At this point of the protocol:

- Alice and Bob shares the sifted key but there is errors due to technical imperfections (bad measures on Alice or Bob side, bad transmission on the quantum channel, dark count, etc.).
- Typical error rate is a few percent. It is named the QBER, Quantum Bit Error Rate. To be compared to the 10^{-9} of classical transmission (BER).
- Eve may know a few percent of the sifted key without being detected.

The BB84 Protocol

Error correction

- The purpose of error correction algorithms is to fix transmission errors.
- We have the additional constraint that the communications are public but we do not want to reveal information to Eve.
- Consequently, we will loose information, that is bits of the sifted key.
- An error correction algorithm is considered as efficient if it guarantees error correction while minimizing the percentage of lost bits.
- The result of the error correction phase is a shortened key which is shared by Alice and Bob without errors with a very high probability. Eve may know some of the bits.

The BB84 Protocol
Error correction

A very simple and not efficient error-correction:

- Repeat:
    - Alice randomly chooses two bits b_1 and b_2 and sends position of b_1, position of b_2 and the value of $b_1 \oplus b_2$ to Bob.
    - Bob checks that he has the same value for $b_1 \oplus b_2$:
        - If equal, Alice and Bob keep b_1 and discard b_2.
        - If no equal, they discard both b_1 and b_2.
- This algorithm is not efficient. More sophisticated algorithms are issued from classical theory.

The BB84 Protocol

Error correction

- Let us assume that the error rate as measured at the Eve detection phase is $e \in [0, 1]$.
- For each bit b_1 or b_2 chosen by Alice, the bit is correct with a probability $1 - e$ and wrong with a probability e.
- The different cases:

Bit b_1	Bit b_2	Good bit discarded	Bad bit discarded	Probability
good	good	b_2		$(1 - e) \times (1 - e)$
good	bad	b_1	b_2	$(1 - e) \times e$
bad	good	b_2	b_1	$e \times (1 - e)$
bad	bad		b_2	$e \times (1 - e)$

- Probability of discarding a bad bit: $2e - e^2$.
- Probability of discarding a good bit: $1 - e + e^2$.

The BB84 Protocol
Error correction

For each run, the probability of discarding a bad bit is $2e - e^2$ and the probability of discarding a good bit is $1 - e + e^2$.

If we do K runs, we have statistically discarded $K(2e - e^2)$ bad bits and $K(1 - e + e^2)$ good bits,a total of $K(1 + e)$.

If the length of the sifted key is N, we have to discard eN bits. Thus, we choose K such that:

$$K(2e - e^2) > eN \quad \text{that is} \quad K = \frac{N}{2 - e}$$

The total number of discarded bits, either good or bad, is:

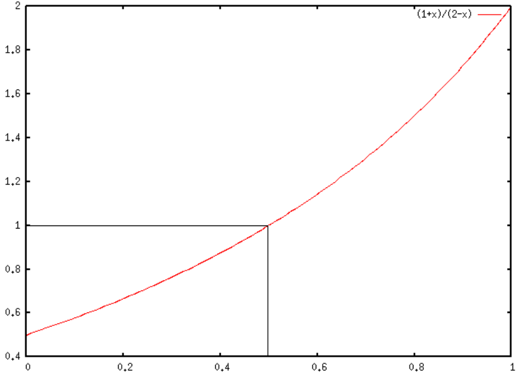
$$N\frac{1 + e}{2 - e}$$

For $e = 0,05$, i.e. an error rate of 5%, we loose 53.8% of the bits of the initial sifted key.

The BB84 Protocol

Error correction

The following curve describes the percentage of lost bits in function of the initial error rate:



For $e = 0.5$, we have $(1 + e)/(2 - e) = 1$ and all bits are lost. Thus, this algorithm is unable to correct an error rate of more than 50%.

The BB84 Protocol

Error correction

Usually, error correction algorithms proceed by adding information to the transmitted message. For instance, one may add parity bits or one may duplicate the information.

In our case, the added informations are publicly sent afterward. Another property is that we do not want to recover the initial message but only to agree on a message which is a part of the initial message. That is why this stage of the protocol is usually called bit reconciliation instead of error correction.

After error correction, Alice and Bob share a corrected key and there is no error. However, Eve may know some bits of this key. The privacy amplification phase will fix this.

In the BB84 Analysis part, we will more formally describe bit reconciliation algorithms.

The BB84 Protocol

Privacy amplification

A privacy amplification algorithm is an algorithm that allows Alice and Bob to reduce the knowledge of Eve to a vanishing part.

Of course, such an algorithm assumes that Bob shares more information with Alice than Eve shares information with Alice. In term of mutual information, this is written:

$$I(Alice; Bob) > I(Alice; Eve)$$

and

$$I(Alice; Bob) > I(Bob; Eve)$$

Here, it is the case.

Of course again, a privacy amplification algorithm will loose a part of the corrected key.

The BB84 Protocol

Privacy amplification

A simple and not efficient privacy amplification algorithm:

- Repeat:
  - Alice randomly chooses two bits b_1 and b_2.
  - Alice sends to Bob, the position of b_1 and the position of b_2.
  - Alice and Bob both discard b_2 and replace b_1 by $b_1 \oplus b_2$.

At each run, the key is shortened and no error is introduced. And Eve is likely to loose information.

The BB84 Protocol

Privacy amplification

When replacing b_1 and b_2 by $b_1 \oplus b_2$:

- If Eve knows b_1 and b_2, she, of course, knows $b_1 \oplus b_2$ and we discarded a known bit.
- However, if Eve knows nothing about b_1, i.e. $H_E(b_1) = 1$, then she knows nothing about $b_1 \oplus b_2$.
- Quantum measurements allows Eve to partially know the value of the bits. For instance, according to her measurements, she may know that $b_1 = 0$ with a probability of 0.75. Assuming, that Eve knows the two bits with a probability of 0.75, then the knowledge of Eve is reduced by the $\oplus$ operation.

If Eve knows that $b_1 = 0$ with a probability of 0.75 and knows that $b_2 = 0$ with a probability of 0.75, then she knows that the result $b_1 \oplus b_2 = 0$ with a probability of 0.625 and we have $h_2(0.625) > h_2(0.75)$, i.e. the entropy is increased.

The BB84 Protocol

Privacy amplification

The complete proof of this privacy amplification algorithm uses information theory.

It is rather technical and mathematically complex.

A formal sketch of the proof is given in the BB84 Analysis section.

The BB84 Protocol

Authentication

BB84 does not address the authentification of the two parties.

Eve could use the man-in-the-middle attack. Eve cuts the links between Alice and Bob. When Alice thinks she is talking with Bob, she is in fact talking with Eve delevering her all the secrets.

The only solution available today is to use classical cryptography tools.

The BB84 Protocol

Authentication

Alice and Bob can authenticate their classical communications provided they already share a small secret key.

Then, the QKD process provides them with the exchanged QKD key which is longer than the initial authentication key.

A part of the QKD key is used to renew the small authentication key.

This is sometimes called secret growing protocol.

The BB84 Protocol
Secret communications

As presented, BB84 allows Alice and Bob to establish a secret key. If Eve is spying the quantum communication, then she will be detected, the QKD process will be aborted, no key will be established and no secret commucation will take place.

However, if the QKD process properly ends, the the secrecy of the key is unconditionnal. That means that whatever computational power Eve can use (even quantum computers), Eve will never be able to discover the key, even after thousands of years.

Thus, if the key is used with unconditionnally secure encryption algorithm to send message, the secrecy of the message is absolute. The key will be used with Vernam ciphering.

BB84 Movie

For the fun...

BB84 with polarization

Secure Distributed Computations

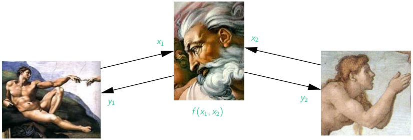
Computation over distributed private inputs

- Function $f : X^n \mapsto Y^n$
- The data are distributed: party A_1 holds x_1; ...; party A_n holds x_n
- Computing of $(y_1, ..., y_n) = f(x_1, ..., x_n)$: A_1 gains y_1; ...; A_n gains y_n
- Security: A_i gains no more information than what can be deduced from x_i and y_i
- Most cases where all of the parties have the same result $y = g(x_1, ..., x_n)$: $f(...) = (y, ..., y)$

Secure Computations

Ideal model

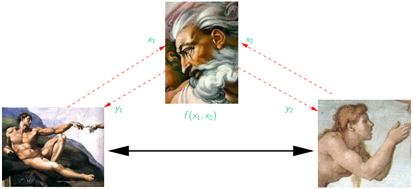
- Ideal model:
  - There is a trusted party T
  - All A_i sends x_i to T who computes $f(x_1, ..., x_n)$ and returns y_i to each A_i



Secure Computations

Real model

- Real model:
  - There is no trusted party
  - Each party A_i communicates with the others to compute y_i
  - In need of rigorous protocols → "emulates" the Ideal model



Ad-hoc and Heuristic approaches

- **Ad-hoc**
  1. Build a protocol
  2. Try to break it
  3. Fix the found bugs
- **Heuristic**
  1. Design a protocol
  2. Provide a list of attacks
  3. Reason that the list is complete

The Dangers

Intuition can mislead

- Ad-hoc: → "Tests can only help to find bugs, cannot find that they do not exist!"
- Heuristic: → The list is often not complete
- These are based on intuition
  - Cannot be evaluated: infinite set of potential strategies
  - Dangerous: Security is tricky and anti-intuitive

Security Reduction

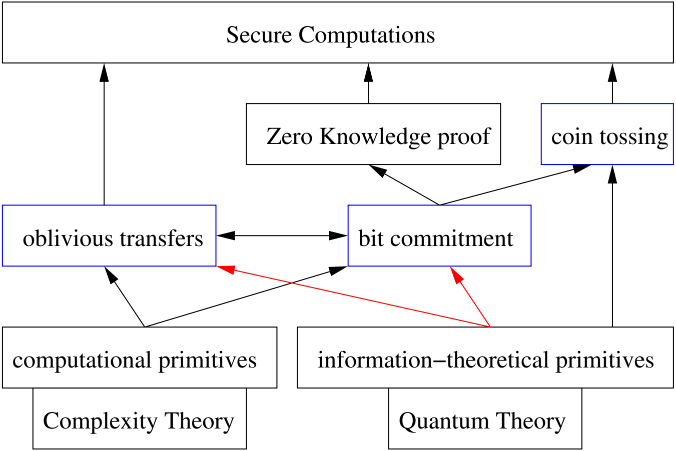
- Cryptographers want to build unbreakable systems
- Rigorous Treatment: based on firm foundations
  1. Provide the complete definition of the problem
  2. Reduce to more basic problems:
     - Computational ones: largely adopted assumptions
     - Information-theoretical ones
  3. Prove the reduction
- Example: “Breaking RSA systems is as hard as factoring large integers”

Abstracting cryptographic assumptions

Cryptographic Engineering

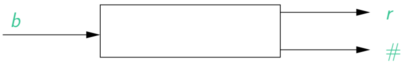
- Modern Cryptography is based on various concrete primitives: factoring large integers, discrete logarithms, ...
- One could base it on more abstract assumptions
  1. Make intermediate layers
  2. These intermediate primitives can then be implemented from concrete primitives
- The system is more flexible
- Example: "One can ... if trapdoor functions exist" instead of "... if factoring large integers is hard"

Foundations of Cryptographic Protocols



Probabilistic Oblivious Transfer (Rabin)

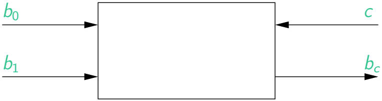
- Rabin supposes a asymmetrical channel
  - Half of times, the channel becomes noisy
  - The receiver is aware of the channel status
  - Sender is not aware of the channel status



- Protocol definition
  - The sender sends a bit b
  - The receiver receives the output r
  - If, the receiver receives the channel status $\#$: $r = b$, otherwise, r is a random bit

Chosen one-out-of-two OT

- One supposes a two-sides computing black box
  - At the sender side, there are two inputs
  - At the receiver side, there are one input and one output



- Protocol definition
  1. The sender has two bits b_0, b_1 as inputs of the first side
  2. The receiver has a bit choice c as the input of the second side
  3. The receiver receives then b_c
  4. The sender does not know c, the receiver does not receive b_{1-c}

Equivalence between the two OTs

Rabin OT built from 1-of-2 OT

1. Sender generates a random bit a
2. He randomly couple a with his message b to make an input pair of 1-of-2 OT: (a, b) or (b, a)
3. Receiver randomly chooses to receive one of the inputs
4. Receiver got b with probability $1/2$
5. After the ending of 1-of-2 OT, Sender tells in which order the inputs are
6. Receiver knows if he has got the bit.

Equivalence between the two OTs

Crepeau: 1-of-2 OT built from Rabin OT

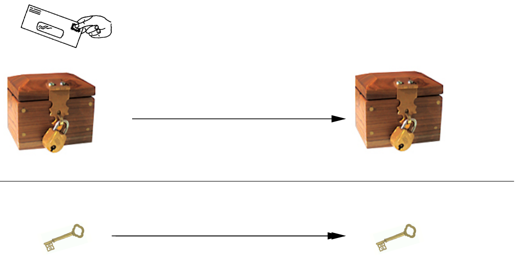
- Protocol
  1. Sender makes a vector of random bits $m = (m_1, .., m_n)$ end sends them via Rabin OT
  2. Receiver constructs two disjoint index sets $I_0, I_1 \subset \{1, .., n\}$ with $|I_0| = |I_1| = n/3$ such that $\forall i \in I_0, m_i$ is received
  3. Receiver settles (I_c, I_{1_c}) according to choice bit c and sends them to Sender
  4. Senders generates $\hat{b}_0 = \bigoplus_{i \in I_c} m_i \oplus b_0$, $\hat{b}_1 = \bigoplus_{i \in I_{1-c}} m_i \oplus b_1$ and sends $(\hat{b}_0, \hat{b}_1)$
  5. Receiver decrypts $b_c = \bigoplus_{i \in I_0} m_i \oplus \hat{b}_c$
- By the Law of Large Numbers, we can choose n large enough
  - Receiver receives more than $n/3$ m_i to construct I_0
  - Receiver does not receive more than $2.n/3$ to construct I_1 as I_0

Public-key based implementation of 1-of-2 OT

1. Party 1 uses two public keys k_0, k_1 and sends them to party 2
2. Party 2 generates a secret key K, encrypts it using party 1's key k_c and sends to party 1
3. Party 1 decrypts with both private keys and has K_0, K_1: $K_c = K$ and K_{1-c} is random
4. Party 1 sends $b_0 \oplus K_0, b_1 \oplus K_1$ to party 2 who can only get b_c

Definition

1. Commit phase
    - Alice has a bit b in her mind
    - Alice send commitment information c_b to Bob
    - Bob cannot determine b from c_b
2. Opening phase
    - Alice unveils b to Bob
    - Bob can always detect if Alice unveils $1 - b$, with help of c_b



Reduction between OT and BC

- BC from 1-of-2 OT
  - Commit phase
    1. Alice prepares n random pairs $(x_1, y_2), ..., (x_n, y_n)$ such as $x_i \oplus y_i = b$, i.e.
       $(x_i, y_i) \leftarrow \{(0,0),(1,1)\}$ for $b = 0$,
       $(x_i, y_i) \leftarrow \{(0,1),(1,0)\}$ for $b = 1$
    2. Alice sends each pair via 1-of-2 OT to Bob who randomly chooses to receive one of its member.
  - Opening phase
    3. Alice reveal b and opens all of the pairs (x_i, y_i)
    4. Bob verifies if his recorded results match the chosen members in all of the pairs.
- OT from BC: quantum implementation

Coin Tossing

- Coin Tossing: Parties 1 and 2 obtain the same random bit r that is not be controlled by any party
- Given Bit Commitment
  1. Party 1 generates a random bit a and commits it to party 2
  2. Party 2 generates a random bit b and sends to party 1
  3. Party 1 opens a and they compute $a \oplus b$

Zero-Knowledge

- Zero-Knowledge proofs: a prover wishes to prove a statement to the verifier so that
    - Zero knowledge: the verifier will learn nothing beyond the fact that the statement is correct
    - Soundness: the prover will not be able to convince the verifier of a wrong statement
- Theorem: Given Bit Commitment, Zero-Knowledge proofs exist for all languages in $\mathcal{NP}$

Two-party protocol
Security definition

- Malicious model: in real computation, adversaries are malicious
  - unlimited behavior
  - substituting local input
  - aborting the protocol (before sending the last message)
- Semi-honest model: the parties are honest but curious
  - The parties respect the protocol
  - They record all inputs, outputs → to learn something more
- We can build computation for semi-honest model, and then force adversarial behavior

Implementation in Semi-honest models

Random shares Paradigm

- Terminology:
  - Let a be some value
  - Party 1 holds a random a_1
  - Party 2 holds $a_2 = a \oplus a_1$
  - We say that the parties hold random shares of a: Each cannot discover a without help of the other
- Solution: let f be the function to be compute
  1. f is decomposed into XOR and AND gates
  2. Each party creates a secret key k_i and the random shares $k_i \oplus x_i$ of the input x_i and sends to the other
  3. All inputs and outputs of intermediate gates are random shares
  4. At the end, the parties combine shares of the output wires to obtain actual output

Random-shared XOR gate

- Inputs: shares of a, b
  - Party 1 holds shares a_1, b_1
  - Party 2 holds shares a_2, b_2
  - $a_1 \oplus a_2 = a$, $b_1 \oplus b_2 = b$
- Computation: shares of $c = a \oplus b$
  - Party 1 computes $c_1 = a_1 \oplus b_1$
  - Party 1 computes $c_2 = a_2 \oplus b_2$
  - $c_1 \oplus c_2 = a \oplus b = c$

Random-shared AND gate

- Inputs: shares of a, b
  - Party 1 holds shares a_1, b_1
  - Party 2 holds shares a_2, b_2
  - $a_1 \oplus a_2 = a$, $b_1 \oplus b_2 = b$
- Computation: shares of $a.b = (a_1 \oplus a_2)(b_1 \oplus b_2)$
  1. Party 1 generates a random bit r and prepares a table $A[4]$: $A[i, j] = (a_1 \oplus i).(b_1 \oplus j) \oplus r$ for $i, j \in \{0, 1\}$
  2. Party 1 sends A to party 2 via 1-of-4 OT: party 2 enters the inputs (a_2, b_2) to receive $A[a_2, b_2] = (a_1 \oplus a_2).(b_1 \oplus b_2) = a.b \oplus r$
  3. Finally: party 1 holds r, party 2 holds $a.b \oplus r$

1-of-4 Oblivious Transfer from 1-of-2 OT

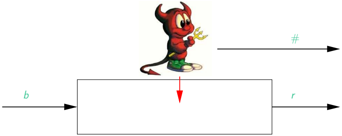
1. Party 1 generates 2 pairs of random bits $(r_1, r_2), (r_3, r_4)$, and 4 keys $k_1 = r_1 \oplus r_3, k_2 = r_1 \oplus r_4, k_3 = r_2 \oplus r_3, k_4 = r_2 \oplus r_4$
2. Party 1 sends each pair to party 2 via 1-of-2 OT: party 2 can only gain one of k_i
3. Party 1 sends $(m_1 \oplus k_1, m_2 \oplus k_2, m_3 \oplus k_3, m_4 \oplus k_4)$ to party 2

Conclusion and results

- Given Oblivious Transfer or Bit Commitment
- Any function can be securely computed in the malicious model
  1. Build secure computation in the semi-honest model
  2. Forcing adversarial behaviors with Coin Tossing and Zero-Knowledge proofs: no more information is revealed

Information-theoretical OT

- One would like information-theoretical implementation to gain unconditional security
- Hypothesis of noisy channels
  - There's a Demon sitting on the channel
  - The Demon flips two coins
  - if the first coin turns "head", he does nothing
  - if the first coin turns "tail", he flips the second coin and replace the transmitted bit on the channel with 0 if "head" and 1 if "tail"



- Quantum Physicists expect a quantum Demon with the Uncertainty Principle

Quantum Oblivious Transfer

First code of Wiesner

- “Conjugate Coding” was used in BB84 for QKD protocol, here for OT
  1. Alice picks a random basis $\theta \leftarrow \{\bigoplus, \bigotimes\}$
  2. Alice sends the encoding qubit $|b\rangle_\theta$ to Bob
  3. Bob randomly chooses a basis $\theta' \leftarrow \{\bigoplus, \bigotimes\}$ and measures the *qubit*
  4. Bob used the right basis with probability $1/2$ and got b, otherwise he got a random bit
  5. Alice announces the basis θ (in BB84 QKD protocol, Bob tells his basis to Alice, and they know both if Bob has received the bit)
- Flaw:
  - Bob can use an intermediate basis $\{\pi/8, 5\pi/8\}$ to guess the message, even more do any general POVM measurement
  - Bob can stock the *qubit* and wait until Alice announces θ to get b

Quantum Oblivious Transfer

Privacy amplification

- One can modify the protocol to make Bob's intermediate basis useless
  1. Alice generates k random bits r_i and a random basis $\theta \leftarrow \{\bigoplus, \bigotimes\}$
  2. Alice sends k encoding qubits $|r_i\rangle_\theta$
  3. Bob randomly chooses θ' to measure all qubits, outputs r'_i
  4. Alice announces θ and $\left(\bigoplus_{i=1}^{k} r_i\right) \oplus b$
  5. If $\theta = \theta'$ then Bob can decode b
- For any measurement of Bob, $\bigoplus_{i=1}^{k} r_i$ is secure with large value of k

Quantum Oblivious Transfer

Prevent photon storing attack

- Bit Commitment can help
  1. Alice inserts k encoding qubits into a sequence of $n-1$ blocks random of qubits $|r_{i,j}\rangle_{\theta_i}, i = 1..n-1, j = 1..k$
  2. Alice sends all n blocks of qubits to Bob
  3. Bob measures each block i with a random basis θ'_i and gains $r'_{i,j}$
  4. Bob makes commitment of all $(\theta'_i, r'_{i,j})$ to Alice
  5. Alice announces the position i_0 of the main block in the sequence
  6. Bob opens the commitments on all $(\theta'_i, r'_{i,j}), i \neq i_0$: Alice can check Bob's honesty
  7. Alice announces θ_{i_0} to complete the OT protocol
- Bob has to measure the qubits before committing

Quantum Oblivious Transfer

Crepeau's Quantum 1-of-2 OT

1. Alice randomly picks $2n$ bits r_i and $2n$ bases θ_i
2. Alice sends $|r_i\rangle_{\theta_i}$ to Bob
3. Bob use a random basis θ'_i to measure $\longrightarrow r'_i$
4. Bob does the commitment of all θ'_i, r'_i
5. Alice randomly sends a set I of n positions $1..2n$
6. Bob does the decommitment of $\theta'_j, r'_j, j \in I$
7. Alice announces $\theta_i, i \in \{1, .., 2n\} \setminus I$
8. Bob randomly builds two disjoint subsets $I_0, I_1, |I_0| = |I_1| = n/3$ where $\forall i \in I_0, \theta'_i = \theta_i$ and sends (I_c, I_{1-c}) to Alice
9. Alice sends $$\left(\hat{b}_0 = \bigoplus_{i \in I_c} r_i \oplus b_0, \hat{b}_1 = \bigoplus_{i \in I_{1-c}} r_i \oplus b_1\right)$$
10. Bob decrypts $b_c = \bigoplus_{i \in I_0} r'_i \oplus \hat{b}_c$

Quantum Bit Commitment

BB84 Coin tossing

- Presented in the same article of BB84 QKD protocol
- BB84-commit(b)
  1. Alice generates n random bits $(c_1, ..., c_n)$
  2. Alice encodes c_i by a $|c_i\rangle\ b$
  3. Alice sends all *qubits* to Bob
  4. Bob measures each *qubit* with a random basis b_i and records the result (c'_i)
- BB84-decommit
  5. Alice unveils b and sends all c_i to Bob
  6. Bob accepts if $\forall i, b_i = b \Rightarrow c'_i = c_i$

EPR attack on BB84 Bit Commitment

- EPR Commit
  1. Alice prepares n EPR states $(qa, qb)_i = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ and sends qb_i to Bob
  2. Bob measures qb_i with basis b_i and records c'_i
- EPR Decommit
  3. Alice chooses b as she wishes and measure qa_i in the basis b to gain c_i
  4. Alice sends b and $(c_i)_n$ to Bob
- EPR trick: $b_i = b \Rightarrow c_i = c'_i$
- $\Rightarrow$ Alice can successfully cheat in BB84 BC protocol

BC91 Bit Commitment

Brassard and Crepeau

- BC91-commit
  1. Alice builds a random matrix $C \in \{0,1\}^{n \times n}$ such as $\forall i, \bigoplus_{j=1}^{n} c_{i,j} = b$
  2. Alice chooses a random basis matrix $B \in \{\bigoplus, \bigotimes\}^{n \times n}$
  3. Alice sends $|c_{i,j}\rangle_{b_{i,j}}$ to Bob
  4. Bob chooses a random basis matrix $B' \in \{\bigoplus, \bigotimes\}^{n \times n}$ to measure, and has the result matrix $C' = (c'_{i,j})_{n \times n}$
- BC91-decommit
  1. Alice announces b, C, B to Bob
  2. Bob verifies if $b'_{i,j} = b_{i,j} \Rightarrow c'_{i,j} = c_{i,j}$ and $\forall i, \bigoplus_{j=1}^{n} ci, j = b$

Arguments failed

- EPR attack on individual qubit, as with BB84 scheme, is not possible in BC91
  - Alice prepares EPR pairs $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ and send half of each pair to Bob
  - At the opening, Alice can choose $b_{i,j}$ to measure
  - The output $c_{i,j}$ is random
  - Alice has to assume $\forall i, \bigoplus_{j=1}^{n} c_{i,j} = b$, if not for a i, she would flip one of $c_{i,j}$
  - Bob can detect this if $b'_{i,j} = b_{i,j}$
- BC91 scheme was developed, and claimed to be secure against EPR attack in BCJL93 (Brassard, Crepeau, Jozsa, and Langlois)
- But a general attack was given by Mayers and Lo & Chau
- Unconditionally secure Quantum Bit Commitment is impossible

Attack on Quantum Bit Commitment

General Bit Commitment

- Bit Commitment is simply an interrupted computation
    - function $f : inputs \mapsto \{0, 1\}$
    - Alice helps Bob to compute f
    - After the commit phase, the computation is interrupted
    - Till this moment, Bob cannot determine the output yet
    - The computation goes on with the opening phase
    - After the commit phase, it is too late for Alice to deflect the computation

Attack on Quantum Bit Commitment

Classical Determinist Bit Commitment is impossible

- The computation is deterministic with public description
- At a moment, the image of the computation is the set of all variables and constants at the both sides
- The computation process is then a determinist sequence of images $I_1 \rightarrow \ldots \rightarrow I_n$
- At the interrupted moment, the image is I_b whose partial image at Bob side is the commitment $part_B(I_b)$, and at Alice side is $part_A(I_b)$
- For the security of b, the partial images must be the same for all b: $part_B(I_0) = part_B(I_1)$
- Alice can therefore always deflect the computation by replacing $part_A(I_b)$ as she want

Attack on Quantum Bit Commitment

Quantum Bit Commitment is determinist !!!

- We consider all quantum systems that participate to the computation
- Any probabilistic variable can be purified by extending the Hilbert space: coupling the variable with another system. For example, if we choose a variable b from $|0\rangle, |1\rangle$ with equal probability, then we prepare two variables $ab = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$
- Any controlled with measurement can be replaced with quantum controlled gate
- The computation becomes a determinist unitary process, acting on a pure state in the whole space - quantum church
- The computation image at each moment is the state at that moment

Attack on Quantum Bit Commitment

Replacement of quantum images

- At the moment just before the opening phase, the image is ρ_b depending on b
- These states are located in a joint space $\mathcal{H}_A \otimes \mathcal{H}_B$ where $\mathcal{H}_A, \mathcal{H}_B$ are the spaces describing all quantum systems at Alice side and Bob side respectively, at the considered moment
- The partial images at Bob's side must be the same: $part_B(I_0) = tr_A(\rho_0) = part_B(I_1) = tr_A(\rho_1)$
- Theorem: if ρ_0, ρ_1 have $tr_A(\rho_0) = tr_A(\rho_1)$ then there exists a local unitary transformation U at Alice side: $U(\rho_0) = \rho_1, U^{-1}(\rho_1) = \rho_0$
- Therefore, Alice can always deflect the computation as she wishes, using U, if we want to assume the security at Bob side

Attack on Quantum Bit Commitment

Quantum Bit Commitment is impossible

- In conclusion, any QBC has to emulate the mentioned model, and is insecure
  - If Bob has no information about the committed bit, than Alice can change her mind
  - Stronger: the less Bob gains information, the more Alice can successfully change her mind
- With the same arguments, H. K. Lo shown also that quantum secure two-party computation is impossible
- These imply the impossibility of QOT

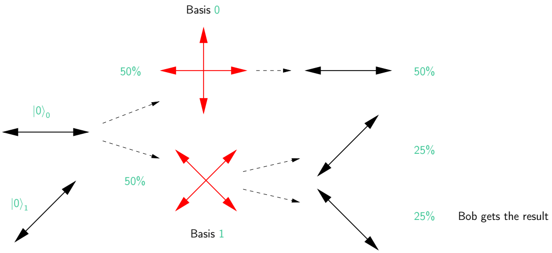
Imperfect protocols

- One can however build imperfect quantum protocols
  - Mixed with computational protocol: quantum OT is secure by temporal public-key bitcommitment
  - In restricted models: bounded quantum memory, without quantum computer, ...
  - Weak protocols: the security is not perfect at both sides
  - Cheat-sensitive protocols: Each party has a non-zero probability to be detected while cheating

Some Variations of QOT

A weak OT

- Alice and Bob agree on a bit r
- Alice encodes b by the qubit $|r\rangle_b$ and sends to Bob
- Bob chooses a random basis b' to measure and gains r'
  - if $r' = 1 - r$ then Bob knows that $b' = 1 - b$
  - if $r' = r$ then Bob does not know whether $b' = b$ or $b' = 1 - b$



Some Variations of QOT

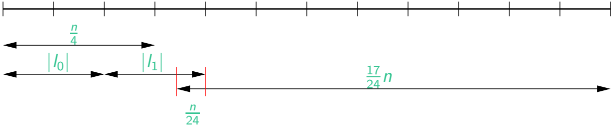
Weak OT parameters

- If Bob use one of the two defined basis
  - Bob gets Alice's message with probability $1/4$
  - If Bob does not get it, he has an uncertainty about it: $H(b) = 0.9183$
- If Bob use any intermediate basis, he has a probability $17/24$ to received an uncertainty $H(b) \geq \alpha > 0$
- Alice can control the probability of "receiving" at Bob side by sending any state that is not one of the two defined states
- Bob can use any general POVM to measure
  - The parameters are modified
  - But the security analysis rests the same

Some Variations of QOT

Bob side secure 1-of-2 OT from Weak OT

- We use Crepeau's reduction to get 1-of-2 OT



- The protocol can be parameterized to be secure at Bob side
- Alice can control the probability distribution and distinguish I_0, I_1

Some Variations of QOT

- Modified protocol
  1. Alice sends to Bob $m > n$ qubits
  2. Bob randomly chooses $m - n$ positions $1..m$ and announces to Alice
  3. Alice announces the corresponding bits sent via WOT: Bob measures and verifies
  4. Alice and Bob continue with n remaining qubits
- When m is sufficiently large, Alice cannot cheat
- Bob can measure first and announces the positions at which he has not got Alice's messages while keeping good result to build I_1
- When $m > \frac{4}{3}n$ Bob can successfully cheat
- We can calibrate m to have some weak 1-of-2 OT

Conclusions

- The theorem of Mayers and Lo & Chau has denied a class of unconditionally secure applications
- We can however build these with
  - weaker security requirements
  - weaker assumptions on computing power, memory capacity
- With current technologies, we are able to build the mentioned quantum protocols which are defeated by future technologies

Bibliography

- John Preskill, cours avec exercices, `http://www.theory.caltech.edu/  preskill/`.
- Michael Nielsen et Isaac Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge (2000).
- Claude Crépeau, cours en ligne, `http://www.cs.mcgill.ca/ crepeau/COURSES/teachi`