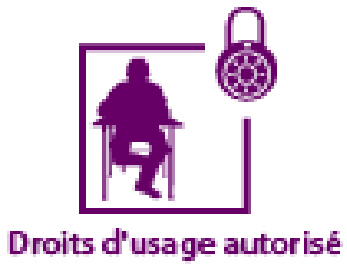
Quantum Cryptography

Patrick Bellot

Licence des droits d'usage



Par le téléchargement ou la consultation de ce document, l'utilisateur accepte la licence d'utilisation qui y est attachée, telle que détaillée dans les dispositions suivantes, et s'engage à la respecter intégralement.

La licence des droits d'usage de ce document confère à l'utilisateur un droit d'usage sur le document consulté ou téléchargé, totalement ou en partie, dans les conditions définies ci-après, et à l'exclusion de toute utilisation commerciale.

Le droit d'usage défini par la licence autorise un usage dans un cadre académique, par un utilisateur donnant des cours dans un établissement d'enseignement secondaire ou supérieur et à l'exclusion expresse des formations commerciales et notamment de formation continue. Ce droit comprend :

- le droit de reproduire tout ou partie du document sur support informatique ou papier,
- le droit de diffuser tout ou partie du document à destination des élèves ou étudiants.

Aucune modification du document dans son contenu, sa forme ou sa présentation n'est autorisé.

Les mentions relatives à la source du document et/ou à son auteur doivent être conservées dans leur intégralité.

Le droit d'usage défini par la licence est personnel, non exclusif et non transmissible.

Tout autre usage que ceux prévus par la licence est soumis à autorisation préalable et expresse de l'auteur : bellot@enst.fr.

Nota Bene

Ce cours a été écrit à partir d'articles classiques de recherche accessibles sur Internet.

Le cours n'a été expérimenté qu'une seule fois. Il est donc susceptible de contenir des erreurs typographiques et même des erreurs plus graves.

Si vous découvrez une erreur n'hésitez pas à le signaler à l'auteur : `bellot@telecom-paristech.fr`.

Introduction

This course is about the algorithmic part of the Quantum Key Distribution protocol named BB84. The BB84 protocol has three main parts:

- The algorithmic part which assumes the quantum properties of the world and the existence of experimental devices handling these properties.
- The Quantum Physics and the conception and development of experimental devices which achieve the functions required by the BB84 protocol.
- The proof of security of the BB84 properties that relies on Quantum Physics and Quantum Information theory.

There is no attempt to teach Cryptography, Information Theory or Quantum Physics. This is not the subject of the course. Only the necessary bases are quickly reminded.

Introduction

After the necessary bases are reminded, we give a light description of the BB84 protocol. That is to say that we do not deeply enter in the different phases of the protocol. We only give simple and inefficient realizations of these phases and we do not give the proofs.

Then we go back to each of these phases and give a detailed and proven implementation of the phase with a proof. This part uses Information Theory.

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).

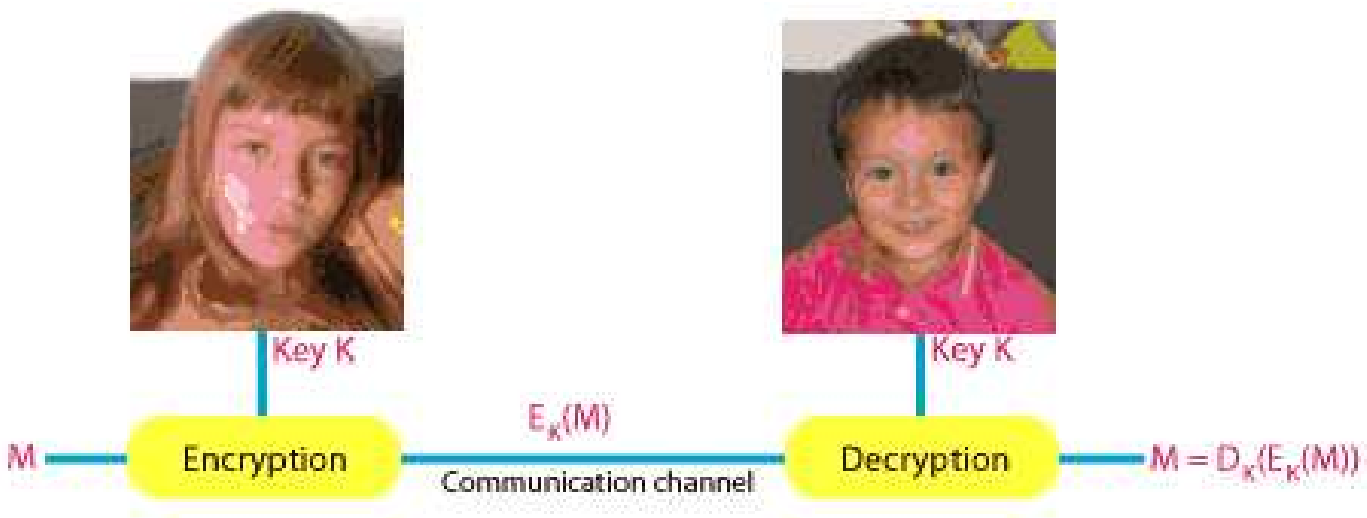
Eve can be passive if she only listens to the communication links.
Eve can be active if she drops messages or modify messages or introduce messages.

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.



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* are symmetric. They are based on chars transposition and substitution. Keys size is 64 or 128 bits or even more. There is no proof of unconditional security.

Symmetric Key Distribution

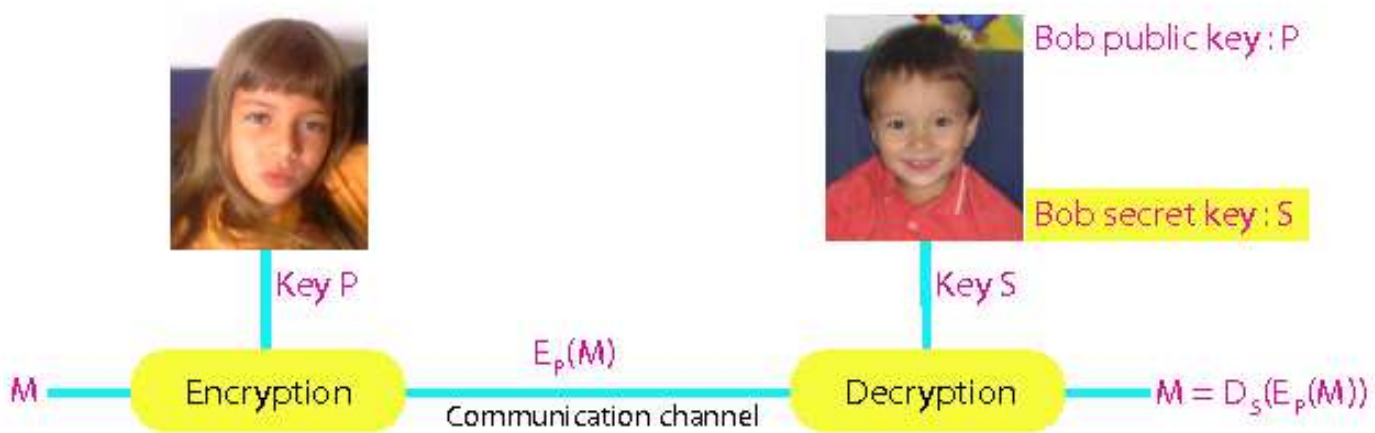
Diffie-Hellman

- The Diffie-Hellman allows two users to share a key to be used in symmetric cryptosystems. It is 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.: $\forall b \in [1, p-1], \exists g$ s.t. $a^g = b \mod p$
  - 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 $K = y_B^{x_A} \ mod \ p = y_A^{x_B} \ mod \ p$.
- The security is based on the intractability of computing discrete logarithms in a finite in $\mathbb{Z}/p\mathbb{Z}$: given $u, v \in [1, p-1]$, find w such that $u = v^w \mod p$.

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:



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 ?

The Quantum Threat

- Quantum Computers are theoretical computers processing quantum bits (qubits).
- There exists an algorithm, 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 communications currently considered as secret could be broken in a few months or 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.
- $E_K(M) = M \oplus K$
- $D_K(C) = C \oplus K$
- $D_K(E_K(M)) = E_K(M) \oplus K = M \oplus K \oplus K = M$.

Vernam Ciphering
Washington talks to Kremlin...

Properties:

- As far as the key is used only once, the Vernam ciphering is unconditionally secure. *Proof later.* Vernam ciphering needs a new key for each message. Thus the main problem is the key distribution. Vernam ciphering is also called one-time pad.
- Let us assume that the key is used two times with two different messages. Eve's listening to the channel learns $E_K(M_1) = M_1 \oplus K$ and $E_K(M_2) = M_2 \oplus K$.
  She computes:
  $E_K(M_1) \oplus E_K(M_2) = (M_1 \oplus K) \oplus (M_2 \oplus K) = M_1 \oplus M_2$
  And, despite of appearances, she got a lot of information. If the k^{th} bit of $M_1 \oplus M_2$ is zero, that means that the k^{th} bits of M_1 and M_2 are equals, otherwise they are opposed.

Vernam Ciphering
Washington talks to Kremlin...

Properties:

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.

In other words, for every pair (M, C), one can find a key K such that $M \oplus K = C$, i.e. : $K = C \oplus M$.

Even with infinite computing power and infinite time, one cannot decode the message.

Example:

The string 'OUI' in binary is:

$$M = 01001111 \quad 01010101 \quad 01001001$$

Let 'KEY' be the key to encode it:

$$K = 01001011 \quad 01000101 \quad 01011001$$

The cryptogram is:

$$E_K(M) = 00000100 \quad 00010000 \quad 00010000$$

If we use the following key to decode:

$$K' = 01001010 \quad 01011111 \quad 01011110$$

We get the following clear text:

$$D_{K'}(E_K(M)) = 01001110 \quad 01001111 \quad 01001110$$

That is the string 'NON' !

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.

- If the coin is regular, 50% head and 50% tail, then the uncertainty is maximal.
- If the coin is not regular, 25% head and 75% tail for instance, then the uncertainty is less than above.
- 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.

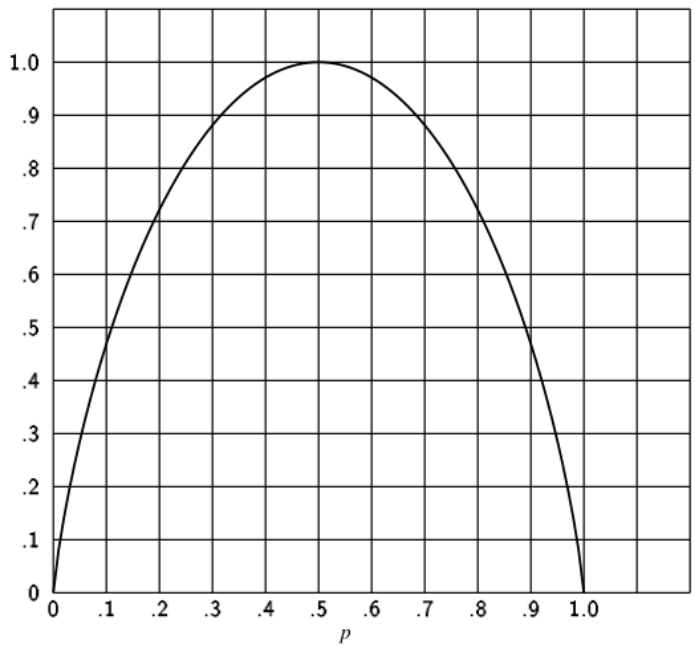
Let $n > 1$ and X be a random variable with possible values $x_1, \ldots, x_n$ occuring with respective probabilities $p_1, \ldots, p_n$, the entropy of X is defined by:

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

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 75% 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:
  $H(X) = -1 \times \log(1) = 0$

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 describes the variation of the entropy in function of p:



This function is usually named the binary entropy h_2:

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

In the general case, X is a variable with n possible values $x_1, \ldots, x_n$ with respective 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 2^k values have equal probabilities and is $H(X) = \log(2^k) = k$.

Let:

- X be a n-valued variable and
- Y be a m-valued variable,

then :

- $H(X,Y)$ is the joint entropy, i.e. the entropy of $Z=(X,Y)$ considered as a single variable.
- $H(X/Y)$ is the entropy of X assumed that Y is known. $H(X/Y)=\sum_{i=1}^{m} p(Y=y_i)\times H(X/Y=y_i)$.
- $I(X;Y)$ is the mutual information. It measures the statistical dependance between X and Y.

We have:

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

Let X be a n-valued variable and Y be a m-valued variable, let x_i be a possible value for X and let y_j be a possible value for Y.

Depending on the context, we may write :

- $p(x_i)$ instead of $p(X = x_i)$;
- $p(y_j)$ instead of $p(Y = y_j)$;
- $p(x_i, y_j)$ instead of $p(X = x_i, Y = y_j)$;
- $p(x_i/y_j)$ instead of $p(X = x_i/Y = y_j)$.

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

Let us show: $H(X,Y) = H(X/Y) + H(Y)$

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

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:

$$H(X,Y) = -\sum_{x_i,y_j} p(x_i,y_j) \times \log(p(x_i/y_j)) + H(Y)$$

.

Then:

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

Thus:

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

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

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.

N.B. Shannon proved that a necessary condition for unconditional secrecy is that $H(K) \geq H(M)$,

a rather unrealistic condition !

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.
- $K = C \oplus M$.

Given any k, since the key is randomly chosen and is independant of the message M, we have:

$$p(K = k_0) = \frac{1}{2^k} \quad \text{and} \quad P(K = k_0 / M = m_0) = \frac{1}{2^k}$$

And we have:

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

Unconditional secrecy of Vernam cipher
Continued...

We have:

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

And:

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

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.

Key Distribution

Key distribution or establishment can be done:

1. Using physical means. When Kremlin and White House exchange keys using military planes. Trusting the whole process is necesary.
2. Using algorithmic means. For instance RSA and Diffie-Hellman. With no proof of security.
3. Using quantum means. That is the object of Quantum Cryptography.

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.

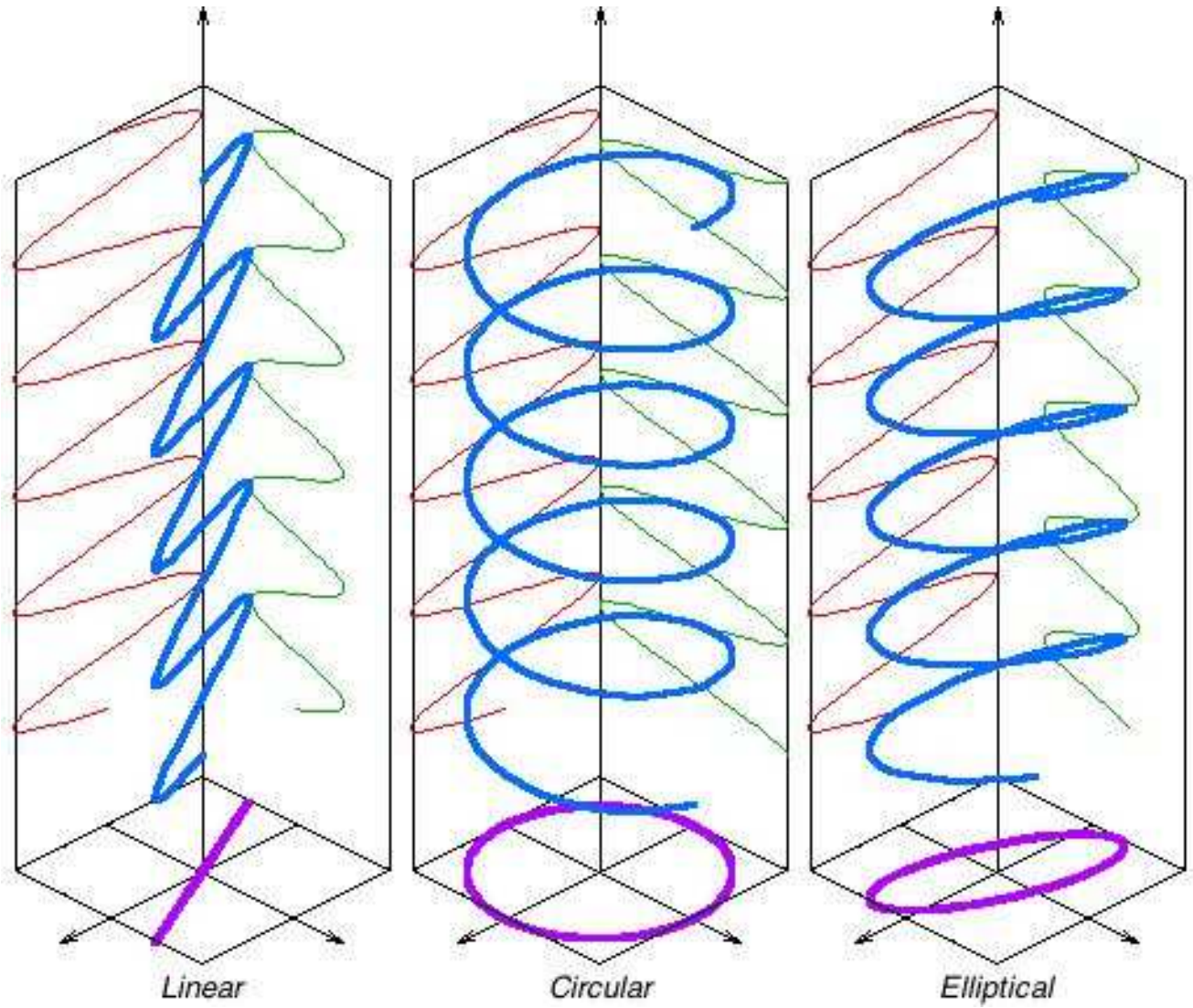
The evolution of a closed quantum system is described by a unitary transformation, i.e. $|\psi'\rangle = U|\psi\rangle$. U is a unitary matrix.

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.

Quantum Basics
Example : a polarized photon



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).

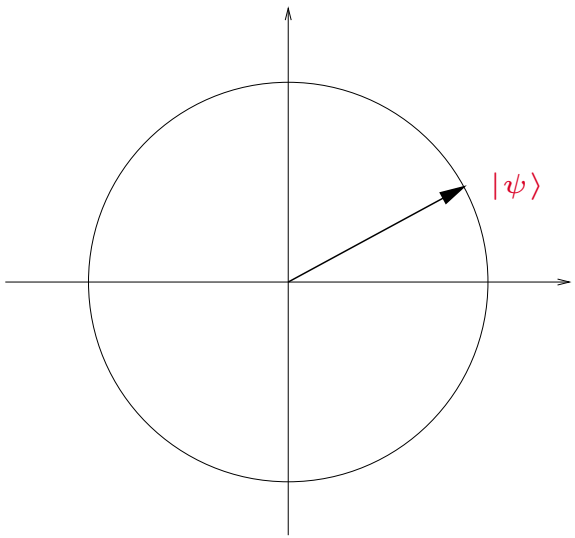
Full information is also provided by the amplitude and phase of oscillations .

In our case, the azimuth angle is sufficient.

Quantum Basics
Example : a polarized photon

The polarization of a photon is represented by the azimuth angle.

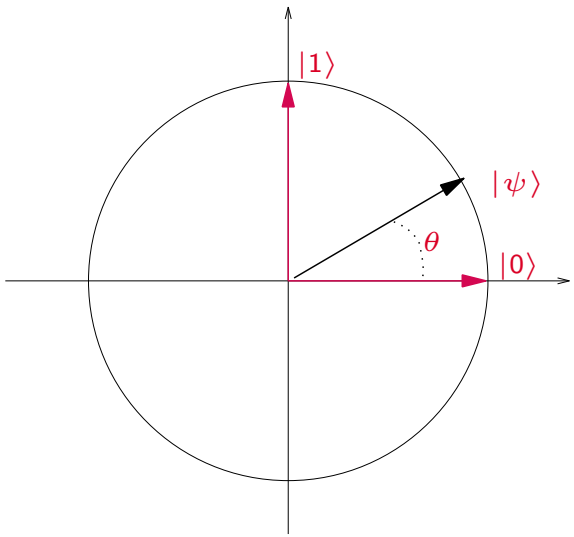
This can be in turn represented by a unitary vector in a 2-dimensional space:



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
$\|0\rangle$	$\|0\rangle$ in 100% of the cases
$\|\pi/4\rangle$	$\|0\rangle$ in 50% of the cases $\|1\rangle$ 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.

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.

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.

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 reads an unknown quantum information, i.e. 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 behaviours 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 the 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.

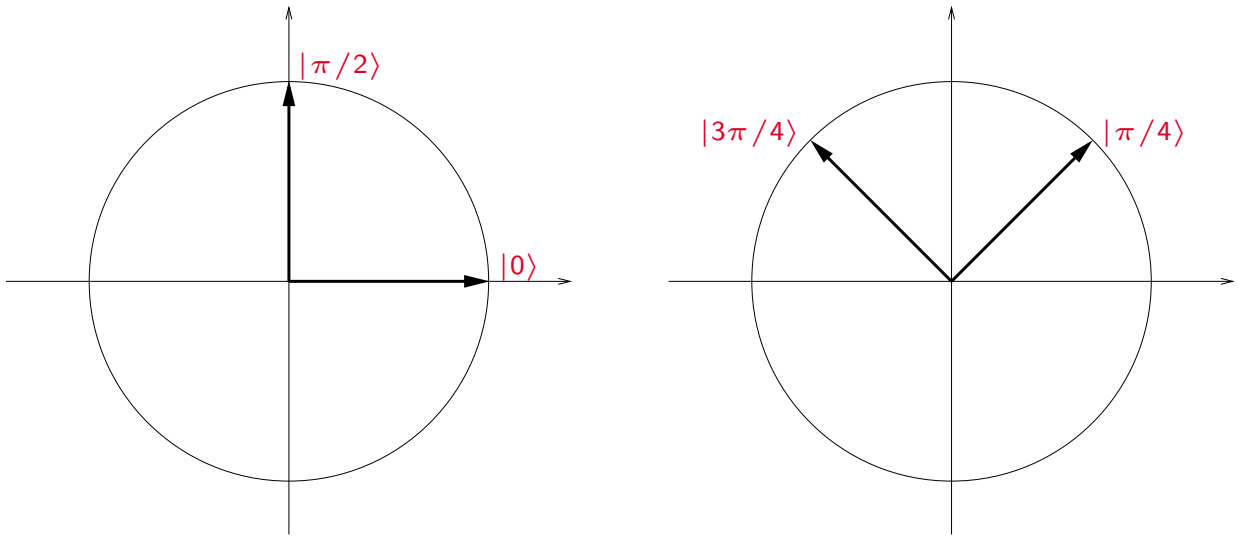
BB84 uses quantum states in a quantum system of dimension 2.

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 $\otimes = \{|\overline{0}\rangle, |\overline{1}\rangle\}$.

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 $|\pi/4\rangle$ 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 with a very high probability.

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

> *When elementary quantum systems are used to transmit digital information, the uncertainty principle gives rise to novel cryptographic phenomena unachievable with traditional transmission media.*

Quantum Cryptography (QC) is unproperly named.

The proper name should be:

Quantum Key Distribution (QKD)

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.

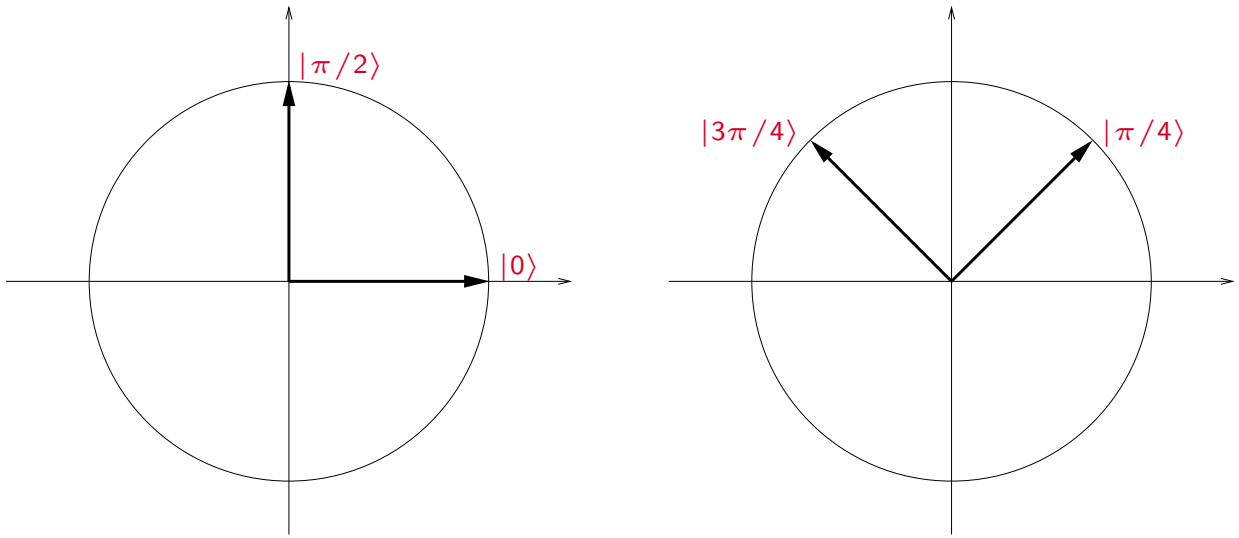
BB84
Information encoding

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

As we have seen, polarization of photons is a 2-dimensional system.

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

Two maximally conjugate bases:



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

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.

Remark. 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.

BB84
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 uses quantum communication in order to establish a raw key.
- Eavesdropper detection. This very simple step detects if someone was spying the communications.
- Bit reconciliation. This step allows to fix the quantum apparatus errors.
- Privacy amplification. This step allows to reduce the eavesdropper's information to a vanishing part if the eavesdropper was not detected.
- Authentication. The two parties authenticate themselves.

Sifting (Alice's part). During the phase, Alice chooses a big number 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:	$\|\bar{1}\rangle$	$\|1\rangle$	$\|0\rangle$	$\|\bar{1}\rangle$	$\|1\rangle$	$\|0\rangle$	$\|\bar{0}\rangle$	$\|\bar{0}\rangle$

BB84
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:	$\|\overline{1}\rangle$	$\|1\rangle$	$\|0\rangle$	$\|\overline{1}\rangle$	$\|1\rangle$	$\|0\rangle$	$\|\overline{0}\rangle$	$\|\overline{0}\rangle$
Bob's random bases:	$\otimes$	$\oplus$	$\otimes$	$\otimes$	$\otimes$	$\otimes$	$\oplus$	$\otimes$
Bob's results:	1	1	?	1	?	?	?	0

BB84
Sifting

Sifting (Bases agreement). Alice and Bob publicly agree 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 key. The other quantum states and their results are discarded.

Note that the bases agreement communications are done on 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:	$\|\overline{1}\rangle$	$\|1\rangle$	$\|0\rangle$	$\|\overline{1}\rangle$	$\|1\rangle$	$\|0\rangle$	$\|\overline{0}\rangle$	$\|\overline{0}\rangle$
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

In the ideal scenario, no quantum apparatus errors occurs.

The only errors come from Eve's measurements.

If Eve does not measure anything, she introduces no error.

If Eve measures all or part of the states, she introduces errors.

The BB84 Protocol
Eavesdropper Detection

If Eve measures all states, she has to guess bases as Bob does.

If N is large, Eve chooses the right bases in 50% of the cases.

When Eve chooses a wrong basis, she introduces no error in 50% of the cases because the perturbation may randomly provide a good result.

Thus, Eve introduces an error rate of 25% if she measures all the states.

The BB84 Protocol
Eavesdropper Detection

Thus, we are facing a simple problem:

- A real quantum system introduces intrinsic errors, typically a few percents.
- Eve introduces errors varying from 0% to 25% following the percentage of states she tries to measure.

For instance, if Eve measures only 10% of the states, she will introduce a supplementary error rate of 2.5% meanwhile she will get about 5% of the sifted key.

The BB84 Protocol
Eavesdropper Detection

The strategy adopted by BB84 is to limit the knowledge of Eve about the sifted key, say 5% and then, to make this knowledge disappear with the privacy amplification algorithm.

Thus, if e is the intrinsic error rate of the quantum apparatus, we must limit the total error rate on the sifted key to $e + 2.5\%$.

The error rate is evaluated by sacrifying a percentage of randomly chosen bits of the sifted key, for instance 20%. These bits are publicly compared by Alice and Bob. This gives a good estimation of the error rate.

If the estimated error rate is more than the threshold $e + 2.5\%$, then the protocol has detected Eve and the session is aborted.

The BB84 Protocol
Bit Reconciliation

An error correction or bit reconciliation algorithm is an algorithm that allows Alice and Bob to eliminate the errors that have occured during quantum operations: producing quantum states, transporting quantum states, measuring quantum states.

Of course, such an algorithm will reduce the size of the key.

In this part, we assume that quantum apparatus is perfect. Thus, we do not need error correction.

We will go back on error correction 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 or Bob. 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.

Let us assume that Eve knows a part $e \in [0, 1]$ of the key shared by Alice and Bob. Because Eve has not been detected, we may assume that e is small, e.g. $e < 0.10$.

N.B.: if $H(b_1) = 1$ or $H(b_2) = 1$, then $H(b_1 \oplus b_2) = 1$.

Assuming that Eve fully knows some bits, entropy is 0, and fully does not know the other bits, entropy is 1. When replacing b_1 and b_2 by $b_1 \oplus b_2$:

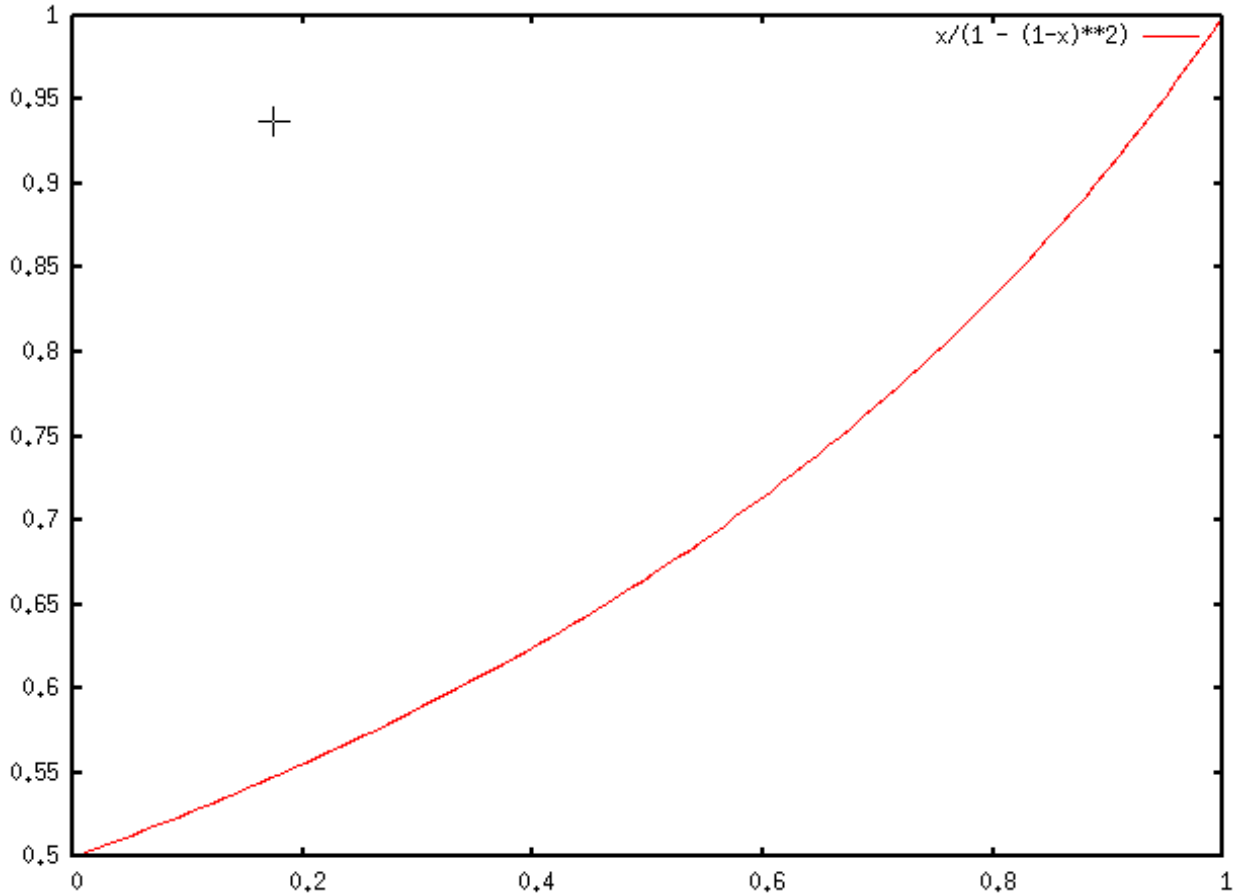
- If Eve knows b_1 and b_2. She, of course, knows $b_1 \oplus b_2$. This occurs with a probability of e^2.
- However, if Eve knows nothing about b_1 but knows b_2. Then she knows nothing about $b_1 \oplus b_2$. This occurs with a probability of $e(1-e)$.
- However, if Eve knows nothing about b_2 but knows b_1. Then she knows nothing about $b_1 \oplus b_2$. This occurs with a probability of $(1-e)e$.
- However, if Eve knows nothing about b_1 and b_2, then she knows nothing about $b_1 \oplus b_2$. This occurs with a probability of $(1-e)^2$.

With an approximate reasonning :

- At each run of the loop, we reduce the length of the key by 1. Thus if we run the loop K times, we loose K bits. Of course, we want $K < N$ where N is the length of the key.
- In the three first cases, Eve looses the knowledge of 1 bit of information. This occurs with a probablility of $1-(1-e)^2$.
- In the fourth case, Eve does not looses the knowledge of a bit. This occurs with a propability of $(1-e)^2$.
- If we run the loop K times, Eve statistically looses $K(1-(1-e)^2)$ bits and we want $K(1-(1-e)^2) > eN$.

Thus we want to run the algorithm K times where: $eN < K(1-(1-e)^2)$ and $K < N$.

Curve of % of lost bits (K) in function of e:



What we have presented yet is a simplified view of privacy amplification. The computation of the number K of runs of the loop is an approximation because e is changed at each run.

Moreover, quantum measurements allows Eve to partially know the value of the bits. Eve could measure bits in the basis $|\pi/6\rangle, |4\pi/6\rangle$.

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.

Example. 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 a privacy amplification algorithm uses information theory.

It is rather technical and mathematically complex.

A generic form for privacy amplification algorithms and a formal sketch of the proof will be given later.

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 delivering 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 that is used and dropped by the authentication process.

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, then 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

Detailed BB84 Algorithms
Introduction

In the previous section, we did not give the details of the algorithms used in BB84.

We gave only trivial version of these algorithms and we did not truly prove them.

In this section, we give the detailed algorithms and their proofs when it is necessary.

Advantage distillation
Introduction

In BB84, before doing the bit reconciliation, in other word the error correction part, we have assumed that Alice and Bob share more information about the sifted key than Alice and Eve do and than Bob and Eve do.

This condition holds because we check that Eve has not measured too much quantum states by checking the error rate. However, we must mention the technique of advantage distillation introduced in:

U.M. Maurer,
Secret Key Agreement by Public Discussion from Common Information
IEEE Trans. on Information Theory, Vol. 39, N. 3; may 1993.

Suprisingly ! Secret key agreement does not require the correlation between Alice and Bob's information to be stronger than between Alice and Eve's information.

Advantage Distillation
Introduction

Advantage distillation comes from thoughts about Shannon's theorem which states that an encryption scheme can be perfectly secret only if $H(K) \geq H(M)$, the entropy on the keys is bigger than the entropy on the messages. A rather restrictive condition...

An underlying condition of this theorem is that Eve gets the same information as Bob.

$\rightarrow$ Thus, a way to circumvent this theorem and its desesperating result is to do things in such a way that Bob and Eve do not receive the same information.

Advantage Distillation

Main result

How can Bob and Eve do not reveive the same information ?

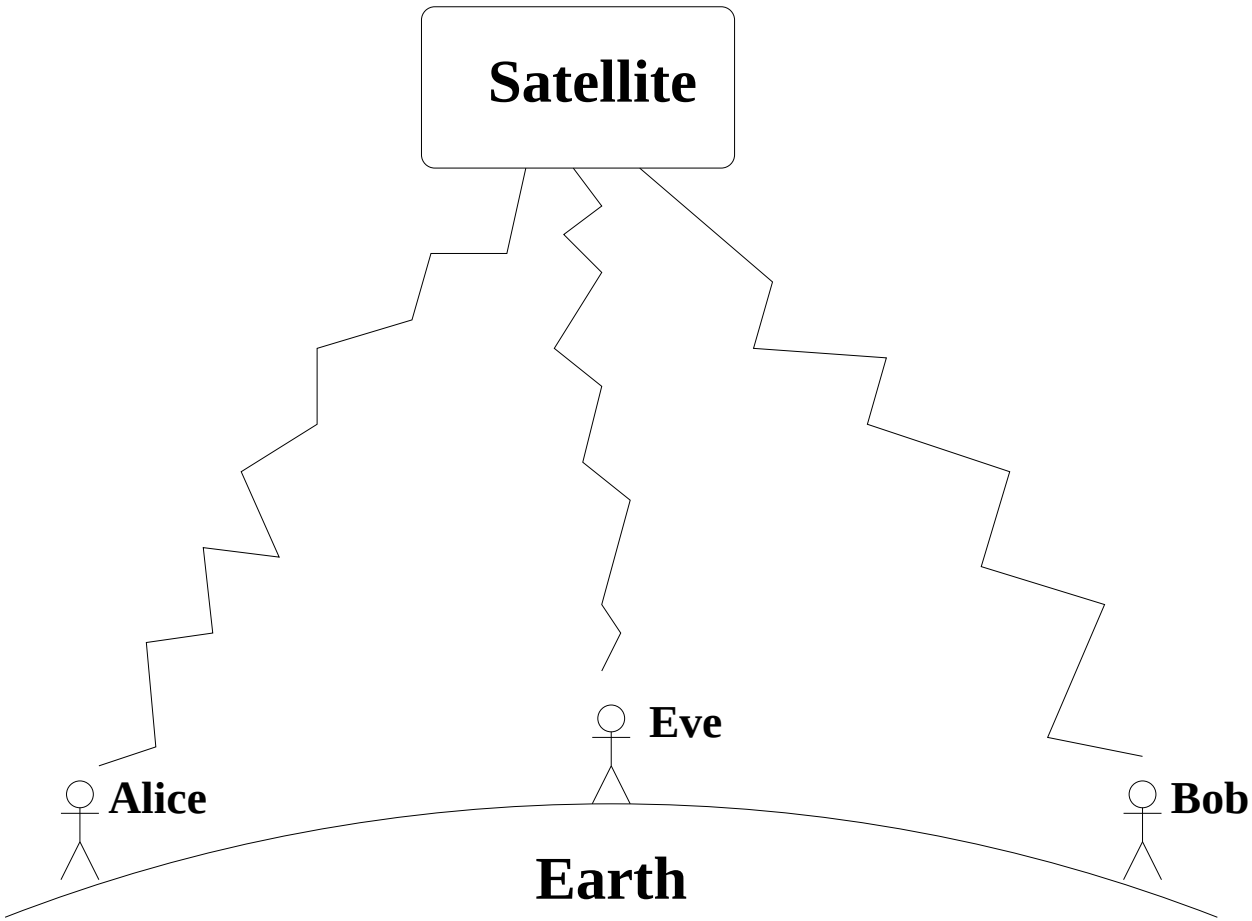
$\rightarrow$ By using noisy channel... without error-correction.

The main result of advantage distillation can be stated as follows. Let us assume that a satellite broadcasts random bits such that Alice, Eve and Bob can receive these bits over independent binary channels with respective error probabilities ϵ_A, ϵ_E and ϵ_B where $\epsilon_E < \epsilon_A$ and $\epsilon_E < \epsilon_B$.
$\rightarrow$ Then it is possible that Alice and Bob establish a secret key.

The independance of the channel can be shown unnecessary.

Advantage Distillation
Example

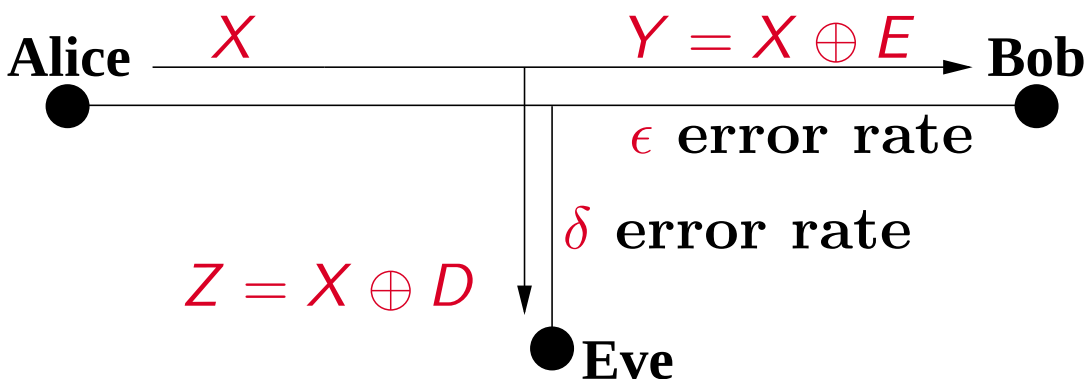
Example: distributing random bits using a low power satellite emission.



Advantage Distillation

A simple example from Wyner

Let us assume that Eve listen to the communication between Alice and Bob:



In order to establish a secret key between Alice and Bob, we assume:

- The noisy channel with error rates ϵ and δ.
- A classical public authentified channel with no errors.

We do not assume that the communication to Eve is degraded compared to the communication to Bob.

Advantage Distillation
A simple example from Wyner

Definition. The secrecy capacity $C_s(P_{Y,Z/X})$ of such a broadcast channel, specified by the probability distribution $P_{Y,Z/X}$, is the maximal rate $R \in [0,1]$ for which:

- for every $\gamma \in [0,1]$
- and every sufficiently large N

there exists:

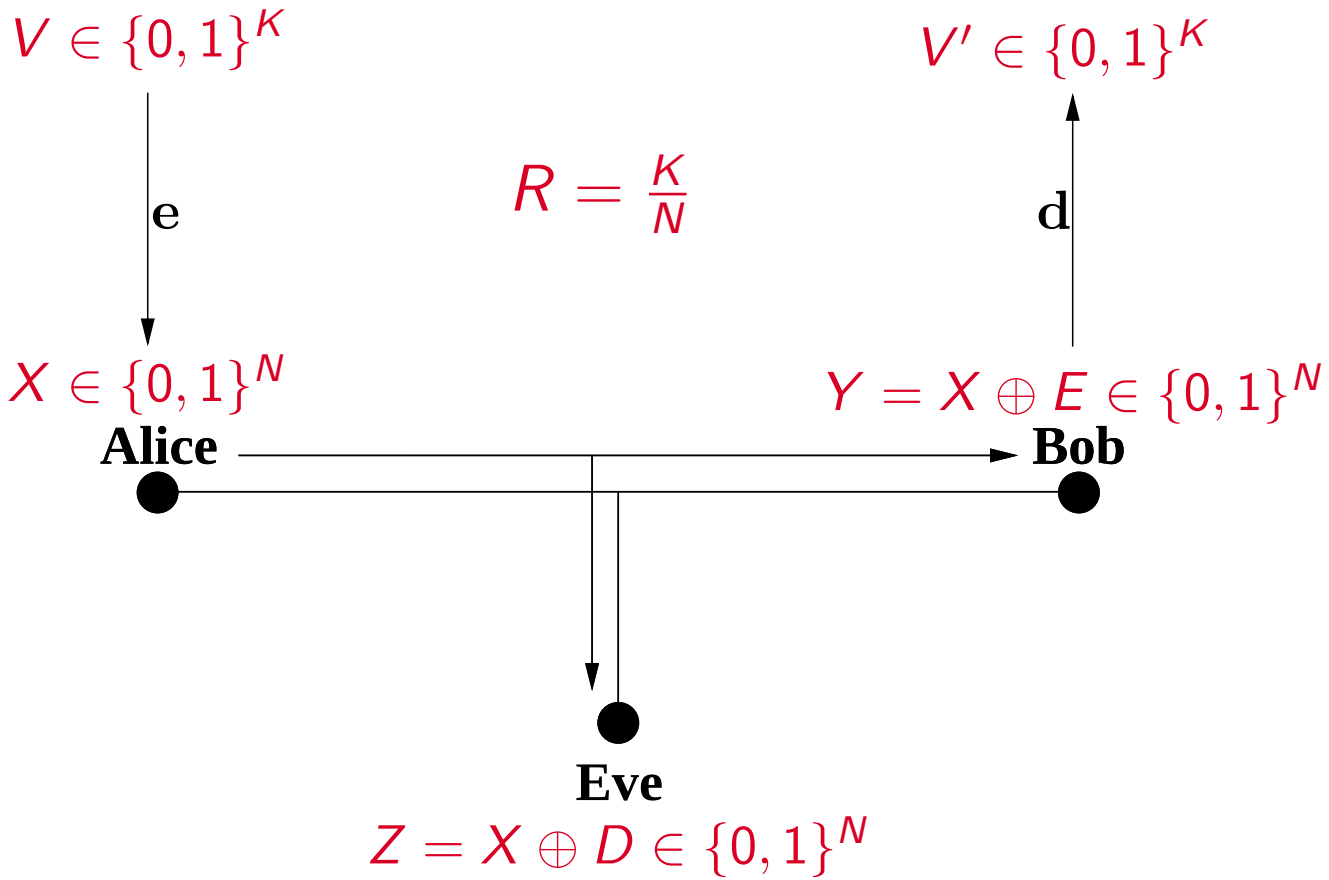
- an encoding function $e : \{0,1\}^K \to \{0,1\}^N$ where $K = \lfloor R \times N \rfloor \in [0,N]$
- and the decoding function $d : \{0,1\}^N \to \{0,1\}^K$

such that for any random variable $V \in \{0,1\}^K$, we have:

- Let $X = e(V)$ sent by Alice, let Y be the message received by Bob et Z be the message received by Eve.
- $p[d(Y) \neq V] < \gamma$
- $H(V/Z)/K > 1 - \gamma$

Advantage Distillation

A simple example from Wyner

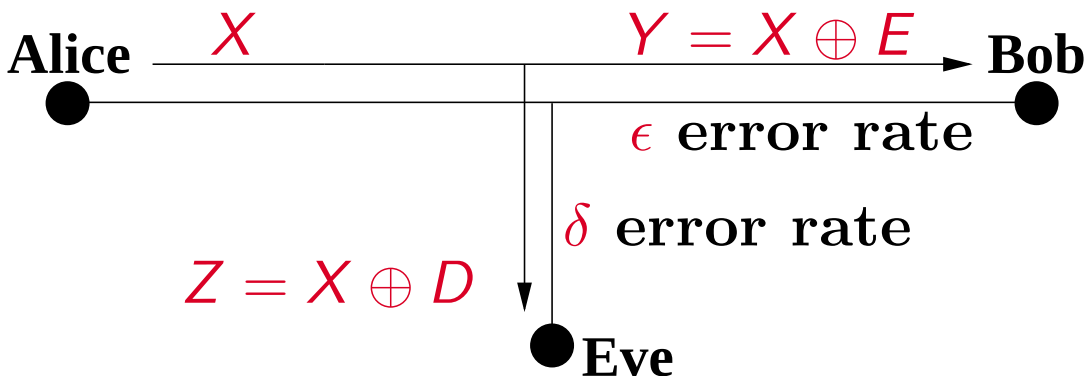


We must have $p[V' \neq V] < \gamma$ and $H(V/Z)/K > 1 - \gamma$.
Then the secrecy capacity is: $C_s(P_{Y,Z/X}) = K/N$.

It depends on the security parameter γ.

Advantage Distillation

A simple example from Wyner



We have:

$$p_{Y/X}(y/x) = p(Y = y/X = x) = \begin{cases} 1-\epsilon & \text{if } y = x \\ \epsilon & \text{if } y \neq x \end{cases}$$

and:

$$p_{Z/X}(z/x) = p(Z = z/X = x) = \begin{cases} 1-\delta & \text{if } z = x \\ \delta & \text{if } z \neq x \end{cases}$$

If the channels are independant: $P_{Y,Z/X} = P_{Y/X} \times P_{Z/X}$

Advantage Distillation
A simple example from Wyner

We reasonably assume that $\epsilon \leq 1/2$ and $\delta \leq 1/2$.

Let us write $D(\epsilon, \delta)$ for $P_{Y,Z/X}$.

Theorem 1. The secrecy capacity of the described broadcast binary channel is given by:

$$C_s(D(\epsilon,\delta)) = \begin{cases} h_2(\delta) - h_2(\epsilon) & \text{if } \delta > \epsilon \\ 0 & \text{otherwise} \end{cases}$$

Proof.

I. Csiszar, J. Korner,
Broadcast Channels with Confidential Messages
IEEE Trans. on Information Theory, Vol. IT-24, pp. 339-348, may 1978.

U.M. Maurer,
Secret Key Agreement by Public Discussion from Common Information
IEEE Trans. on Information Theory, Vol. 39, N. 3; may 1993.

Advantage Distillation

A simple example from Wyner

According to theorem 1, the secrecy capacity vanishes when $\epsilon \geq \delta$.

We now, show that by allowing error-free feedback from Bob to Alice, then Alice and Bob are able to establish a secret key.

Theorem 2. The secrecy capacity with public discusssion of the described binary broadcast channel is given by:

$$C_s(D(\epsilon, \delta)) = h_2(\epsilon + \delta - 2\epsilon\delta) - h_2(\epsilon)$$

We have that $C_s(D(\epsilon, \delta)) > 0$ unless $\epsilon = 1/2$ and $\delta = 0$ or $\delta = 1$.

Proof

Alice sends a random bit string X over the broadcast channel. We have: $p_X(0) = p(X = 0) = 1/2$ and $p_X(1) = p(X = 1) = 0.5$.

Let Y and Z be the bit strings received by *Bob* and *Eve*. We have: $Y = X \oplus E$ with $p_E(1) = P(E = 1) = \epsilon$ and $Z = X \oplus D$ with $p_D(1) = p(D = 1) = \delta$.

Let Bob chooses a random string V. Bob sends $W = Y \oplus V$ on the error free channel.

Alice computes:
$W \oplus X = Y \oplus V \oplus X = X \oplus E \oplus V \oplus X = V \oplus E$.

Eve computes:
$W \oplus Z = Y \oplus V \oplus Z = X \oplus E \oplus V \oplus X \oplus D = V \oplus E \oplus D$.

Advantage Distillation

A simple example from Wyner

For Eve:

$$
\begin{aligned}
& H(V/Z,W) \\
= \;& H(V/Z \oplus W, W) \\
& \quad \text{because } (Z \oplus W, W) \text{ determines } (Z,W) \\
= \;& H(V,W/Z \oplus W) - H(W/Z \oplus W) \\
& \quad \text{because } p(U/W) = P(U,W)/p(W) \\
= \;& H(V/Z \oplus W) + H(W/V, Z \oplus W) - H(W/Z \oplus W) \\
& \quad \text{because } p(V,U) = p(V) \times p(U/V)
\end{aligned}
$$

Then:

$$
\begin{aligned}
& H(W/V, Z \oplus W) \\
= \;& H(X \oplus V \oplus E/V, V \oplus E \oplus D) \\
= \;& 1 \\
& \quad \text{because } X \text{ is totally random and} \\
& \quad \text{statistically independent from } V, E, D
\end{aligned}
$$

And: $H(W/Z \oplus W) = 1$ for the same reasons.

Advantage Distillation

A simple example from Wyner

Thus:

$$\begin{aligned} & H(V/Z,W) \\ = \; & H(V/Z \oplus W) \\ = \; & H(V/X \oplus D \oplus X \oplus E \oplus V) \\ = \; & H(V/V \oplus D \oplus E) \\ = \; & H(D \oplus E) \quad \text{(easy proof)} \end{aligned}$$

- Bob's error on V is 0 because Bob choosed V.
- Alice's error on V is E with $p(E=1)=\epsilon$.
- Eve's error on V is $D \oplus E$ with $p(D \oplus E = 1) = \epsilon + \delta - 2\epsilon\delta$.

According to theorem 1, the secret capacity is

$$h_2(\epsilon + \delta - 2\epsilon\delta) - h_2(\epsilon)$$

We have that $h_2(\epsilon + \delta - 2\epsilon\delta) - h_2(\epsilon) > 0$ unless $\epsilon = 1/2$ and $\delta = 0$ or $\delta = 1$.

Advantage Distillation
Conclusion

- This is only a short introduction to advantage distillation.
- All details, proofs and theoretical limits, in the mentionned articles.
- Advantage distillation shows that it is possible to circumvent Shannon's theorem about unconditional secrecy.

Bit Reconciliation
Introduction

Based on:

G. Brassard, L. Salvail,
Secret-Key Reconciliation by Public Discussion
in Advances in Cryptology (Eurocrypt 93), LNCS 765,
pp. 410-423, 1993.

Bit Reconciliation
Introduction

At this stage of the QKD protocol, Alice and Bob have done the sifting and they share a sequence of bits.

However, the channel is considered as noisy because:

- Eve may have measured some qubits without being detected. She may have introduced errors.
- The equipment is not reliable: dark counts in the receptor, bad measurements, bad transmission of the quantum states in the quantum channel.

Bit Reconciliation
Introduction

Reconciliation is the process of correcting error's between Alice and Bob using public discussion. Because the discussions are public, some information is leaked to Eve. One interesting problem is to build a reconciliation protocol that leaks a minimum of information to Eve.

Main results:

- There exists a protocol that reveals the minimum of information but it cannot be implemented efficiently.
- If Alice and Bob are willing to reveal an arbitrary small amount of information beyond the minimum, there exists polynomial protocols.
- We present a protocol named Cascade which leaks an amount of information acceptably close to the minimum.

Bit Reconciliation

Stating the problem

Let us state the problem in a general setting:

We consider a Binary Symmetric Channel, a BSC, allowing the transmission of sequences of bits A from Alice to Bob with a probability p of error. Bob receives B.

If the string sent by Alice has length n and is totally random, then $H(A) = |A| = n$ and we have:

$$H(A/B) = n \times h_2(p) = H(A \oplus B)$$

Proof: next slide.

We denote such a Binary Symmetric Channel by $BSC(p)$, p is the probability of an erroneous transmission of a bit.

Bit Reconciliation
Stating the problem

Proof: Let $A = (A_1, .., A_n)$, if the A_i are independant, then $H(A) = \sum_{i=1}^{n} H(A_i)$:

$$
\begin{aligned}
& H(A) \\
= \quad & \textstyle\sum_{a \in \{0,1\}^n} -p(A = a) \times \log(p(A = a)) \\
& \text{we write: } a = (a_1, \ldots, a_n) \\
= \quad & \textstyle\sum_{a_1=0}^{1} \cdots \sum_{a_n=0}^{1} -p(A_1 = a_1, .., A_n = a_n) \times \log(p(A_1 = a_1, .., A_n = a_n)) \\
= \quad & \textstyle\sum_{a_1=0}^{1} \cdots \sum_{a_n=0}^{1} -p(A_1 = a_1, .., A_n = a_n) \times \log(\prod_{i=1}^{n}(p(A_i = a_i))) \\
= \quad & \textstyle\sum_{a_1=0}^{1} \cdots \sum_{a_n=0}^{1} -p(A_1 = a_1, .., A_n = a_n) \times \sum_{i=1}^{n} \log(p(A_i = a_i)) \\
= \quad & \textstyle\sum_{i=1}^{n} \sum_{a_1=0}^{1} \cdots \sum_{a_n=0}^{1} -p(A_1 = a_1, .., A_n = a_n) \times \log(p(A_i = a_i)) \\
= \quad & \textstyle\sum_{i=1}^{n} \\
& \quad \textstyle\sum_{a_i=0}^{1} \\
& \quad\quad \textstyle\sum_{a_1=0}^{1} \cdots \sharp i \cdots \sum_{a_n=0}^{1} \\
& \quad\quad\quad -p(A_1 = a_1, .., A_n = a_n) \times \log(p(A_i = a_i)) \\
= \quad & \textstyle\sum_{i=1}^{n} \\
& \quad \textstyle\sum_{a_i=0}^{1} \\
& \quad\quad \textstyle\sum_{a_1=0}^{1} \cdots \sharp i \cdots \sum_{a_n=0}^{1} \\
& \quad\quad\quad -p(A_1 = a_1, .. \sharp i .., A_n = a_n) \times p(A_i = a_i) \times \log(p(A_i = a_i)) \\
= \quad & \textstyle\sum_{i=1}^{n} \\
& \quad \textstyle\sum_{a_i=0}^{1} \\
& \quad\quad -p(A_i = a_i) \times \log(p(A_i = a_i)) \\
= \quad & \textstyle\sum_{i=1}^{n} - \sum_{a_i=0}^{1} p(A_i = a_i) \times \log(p(A_i = a_i)) \\
= \quad & \textstyle\sum_{i=1}^{n} H(A_i)
\end{aligned}
$$

Bit Reconciliation
Stating the problem

Proof (continued):

Thus:
$H(A \oplus B) = \sum_{i=1}^{n} H(A_i \oplus B_i) = n \times h_2(p)$
because:

- $A_i \oplus B_i = 1$ with a probability of p (error probability) and
- $A_i \oplus B_i = 0$ with a probability of $1 - p$

i.e. $H(A_i \oplus B_i) = h_2(p)$.

Then:

$$\begin{array}{rl} & H(A/B) \\ = & \sum_{b \in \{0,1\}^n} p(B = b) \times H(A/B = b) \\ = & \sum_{b \in \{0,1\}^n} p(B = b) \times \sum_{i=1}^{n} H(A_i/B = b) \\ = & \sum_{b \in \{0,1\}^n} p(B = b) \times \sum_{i=1}^{n} h_2(p) \\ = & \sum_{b \in \{0,1\}^n} p(B = b) \times n \times h_2(p) \\ = & n \times h_2(p) \end{array}$$

because:

- $A_i = b_i$ with a probability of $1 - p$ (no error probability) and
- $A_i = \neg b_i$ with a probability of p

i.e. $H(A_i/B = b) = h_2(p)$

Bit Reconciliation
Stating the problem

We model the quantum channel as $BSC(p)$ for some p.

A bit reconciliation protocol R^p is an algorithm runned on Alice and Bob strings A and B to produce a shared string S by exchanging some information Q on the public channel. This is denoted by $R^p(A, B) = [S, Q]$.

If the protocol fails to produce a shared string S, then we write $S = \perp$.

The amount of Shannon information leaked to Eve is written as $I_E(S/Q)$.

Bit Reconciliation

Stating the problem

Let $0 \leq \epsilon \leq 1$ be a security parameter.

Definition 1. A reconciliation protocol R^p is ϵ-robust if

$$\exists N_0(\epsilon) \quad \text{such that} \quad \forall n \geq N_0(\epsilon),$$

$$\sum_{\alpha,\beta \in \{0,1\}^n} p(A = \alpha, B = \beta) \times p(R^p(\alpha, \beta) = [\perp, \cdot]) \leq \epsilon$$

Bit Reconciliation
Stating the problem

Let $0 \leq p \leq 1/2$ be the error probability of a $BSC(p)$.

Let $0 \leq \epsilon \leq 1$ be a security parameter.

Let $R^p = [S, Q]$ be an ϵ-robust reconciliation protocol.

Theorem 2.

$$\lim_{n \to \infty} \frac{I_E(S/Q)}{n \times h_2(p)} \geq 1$$

This theorem is direct consequence of the noiseless coding theorem. It states a lower bound which is used to justify the definition of optimality.

Bit Reconciliation
Stating the problem

Let us name $0 \leq \epsilon \leq 1$ the security parameter.

Definition 3. A reconciliation protocol R^p is optimal if

1. $\forall\ 0 \leq \epsilon \leq 1$, the protocol R^p is ϵ-robust
2. and:

$$\lim_{n \to \infty} \frac{I_E(S/Q)}{n \times h_2(p)} = 1$$

Protocol 1. Let $m \leq n$:

1. Alice randomly chooses a random function f from $\{0,1\}^n$ into $\{0,1\}^m$. Alice publicly communicates this function to Bob.
2. Alice computes $f(A)$ and sends the result to Bob.
3. Decoding. Bob chooses B' in the set $\{D \in \{0,1\}^n \ / \ f(D) = f(A)\}$ s.t. $d(B, B')$ is minimal.

The Hamming distance $d(A, B)$ is the number of places where A and B differs. It could be defined as:

$$d(A,B) = \sum_{i=1}^{n} A_i \oplus B_i = w(A \oplus B)$$

where $w(A) = \sum_{i=1}^{n} A_i$ is the weight of A.

Theorem 4. Protocol 1 is optimal for an adequate choice of the parameter m.

Proof of theorem 4.

Let $C = A \oplus B$ recording the difference between A and B.

Let Err be the event of a decoding error, i.e. $B' \neq A$.

We have $E[w(C)] = n \times p$.
Proof later.

And $V[w(C)] = E[w^2(C) - (np)^2] = n \times p \times (1-p)$.
Proof later.

Proof of $E[w(C)] = n \times p$.

$$\begin{aligned} & E[w(C)] \\ = \; & E[C_1 + \cdots + C_n] \\ = \; & E[C_1] + \cdots + E[C_n] \end{aligned}$$

because the C_i are independant

And:

$$\begin{aligned} & E[C_i] \\ = \; & 0 \times p(C_i = 0) + 1 \times p(C_i = 1) \\ = \; & 0 \times (1 - p) + 1 \times p \\ = \; & p \end{aligned}$$

Proof of $V[w(C)] = n \times p \times (1-p)$
or equivalently $E[w^2(C)] = n \times p - n \times p^2 + n^2 \times p^2$.

The result can be verified for $n = 1$. Trivial.

We write C_n just to mean that C has length n.

$$
\begin{aligned}
& E[w^2(C_n)] \\
= \; & \sum_{c=0}^{n} c^2 \times p(w(C_n) = c) \\
= \; & E[(w(X) + w(C_{n-1}))^2] \\
& \quad \text{where: } C_n = (X, \cdots C_{n-1} \cdots) \\
= \; & \sum_{x=0}^{1} \sum_{c=0}^{n-1} (x+c)^2 \times p(X = x, w(C_{n-1}) = c) \\
= \; & \sum_{x=0}^{1} \sum_{c=0}^{n-1} (x+c)^2 \times p(X = x) \times p(w(C_{n-1}) = c) \\
& \quad \text{because } X \text{ and } C_{n-1} \text{ are independant} \\
= \; & \sum_{c=0}^{n-1} c^2 \times (1-p) \times p(C_{n-1} = c) \\
& + \sum_{c=0}^{n-1} (1+c)^2 \times p \times p(C_{n-1} = c) \\
= \; & (1-p) \times \sum_{c=0}^{n-1} c^2 \times p(C_{n-1} = c) \\
& + p \times \sum_{c=0}^{n-1} (1 + 2 \times c + c^2) \times p(C_{n-1} = c)
\end{aligned}
$$

$$
\begin{aligned}
&= (1-p) \times \sum_{c=0}^{n-1} c^2 \times p(w(C_{n-1}) = c) \\
&\quad +p \times \sum_{c=0}^{n-1} (1 + 2 \times c + c^2) \times p(w(C_{n-1}) = c) \\
&\quad \text{from previous slide} \\
&= (1-p) E[w^2(C_{n-1})] \\
&\quad +p \times \sum_{c=0}^{n-1} p(w(C_{n-1} = c)) \\
&\quad +p \times \sum_{c=0}^{n-1} 2 \times c \times p(w(C_{n-1}) = c) \\
&\quad +p \times \sum_{c=0}^{n-1} c^2 \times p(w(C_{n-1}) = c) \\
&= (1-p) E[w^2(C_{n-1})] \\
&\quad +p \times 1 \\
&\quad +2 \times p \times E[w(C_{n-1})] \\
&\quad +p \times E[w^2(C_{n-1})] \\
&= E[w^2(C_{n-1})] + 2 \times p \times E[w(C_{n-1})] + p
\end{aligned}
$$

Then $E[w^2(C_n)] = E[w^2(C_{n-1})] + 2 \times p \times E[w(C_{n-1})] + p$ can be used for reasoning by recurrence.

Bit Reconciliation

Proof of theorem 4.

Err is the event of a decoding error, $C = A \oplus B$ records the transmission errors, let us prove

$$p(\neg Err / w(C) \leq r) \geq (1 - 1/2^m)^{\sum_{j=1}^{r} \binom{n}{j}}$$

For each C / $w(C) \in [0, r]$, a correct decoding surely occurs if there exists only one B' such that $d(B', B) \leq d(A, B)$ and $f(B') = f(A)$.

We have $p(\neg Err / C = C_i) = p(\neg Err / A, B$ fixed but $\neq)$.

Let us name $\mathcal{E} = \{B'$ s.t. $d(B', B) \leq d(A, B)\}$:

$$\sharp\{f : \mathcal{E} \to \{0,1\}^m \text{ s.t. } f(A) \text{ fixed}\} = (\sharp\mathcal{E} - 1) \times 2^m$$

$$\sharp\{f : \mathcal{E} \to \{0,1\}^m \text{ s.t. } f(A) \text{ fixed } \wedge f(X) \neq f(A) \text{ if } X \neq A\} = (\sharp\mathcal{E} - 1) \times (2^m - 1)$$

Then $p(\neg Err / C = C_i) \geq \frac{(\sharp\mathcal{E}-1)\times(2^m-1)}{(\sharp\mathcal{E}-1)\times 2^m} = \frac{2^m-1}{2^m}$

Bit Reconciliation

Proof of theorem 4.

We have:

$$
\begin{aligned}
& p(\neg Err / w(C) \leq r) \\
= \;& p(\neg Err / \bigvee_{j=0}^{r} w(C) = j) \\
\geq \;& p(\neg Err / \bigvee_{j=1}^{r} w(C) = j) \\
& \quad \text{case } j = 0 \text{ means no error} \\
= \;& \prod_{j=1}^{r} p(\neg Err / w(C) = j) \\
= \;& \prod_{j=1}^{r} \prod_{i=1}^{\binom{n}{j}} p(\neg Err / C = C_i) \\
& \quad \text{when } w(C) = j, \text{ there exists } \binom{n}{j} \text{ cases for } C \\
\geq \;& \prod_{j=1}^{r} \prod_{i=1}^{\binom{n}{j}} \frac{2^m - 1}{2^m} \\
& \quad \text{because } p(\neg Err / C = C_i) \geq (2^m - 1)/2^m \\
= \;& \prod_{j=1}^{r} \prod_{i=1}^{\binom{n}{j}} (1 - 2^{-m}) \\
= \;& \prod_{j=1}^{r} (1 - 2^{-m})^{\binom{n}{j}} \\
= \;& (1 - 2^{-m})^{\sum_{j=1}^{r} \binom{n}{j}}
\end{aligned}
$$

Bit Reconciliation

Proof of theorem 4.

$$\begin{aligned} p(Err) &= p(w(C) \leq r) \times p(Err/w(C) \leq r) \\ &\quad + p(w(C) > r) \times p(Err/w(C) > r) \\ &\leq p(Err/w(C) \leq r) + p(w(C) > r) \end{aligned}$$

Let $r = \lfloor n \times p + n \times \epsilon_n \rfloor$ with $\epsilon_n = 1/\log n$:

$$\begin{aligned} & p(w(C) > r) \\ = \; & p(w(C) > \lfloor n \times p + n \times \epsilon_n \rfloor) \\ \leq \; & p(|w(C) - n \times p| \geq n \times \epsilon_n) \end{aligned}$$

because:

$$|w(C) - n \times p| \geq n \times \epsilon_n \Rightarrow w(C) > \lfloor n \times p + n \times \epsilon_n \rfloor$$

$$\begin{aligned} = \; & p(|w(C) - E[w(C)]| \geq n \times \epsilon_n) \\ \leq \; & V[w(C)]/(n \times \epsilon_n)^2 \end{aligned}$$

because: $p(|X - E(X)| \geq a) \leq V(x)/a^2$ (Chebyshev)

$$\begin{aligned} = \; & n \times p \times (1-p)/(n \times \epsilon_n)^2 \\ = \; & p \times (1-p)/(n \times \epsilon_n^2) \\ = \; & \log^2 n \times p \times (1-p)/n \end{aligned}$$

Bit Reconciliation

Proof of theorem 4.

We have:

$$\lim_{n\to\infty} p(w(C) > r) \leq \lim_{n\to\infty} \log^2 n \times p \times (1-p)/n = 0$$

We also have:

$$\begin{aligned} & p(Err/w(C) \leq r) \\ = \; & 1 - p(\neg Err/w(C) \leq R) \\ \leq \; & 1 - (1-2^{-m})^{\sum_{j=1}^{r} \binom{n}{j}} \\ \leq \; & 1 - \left(\frac{1}{e}\right)^{2^{-m} \times \sum_{j=0}^{r} \binom{n}{j}} \\ \leq \; & 1 - \left(\frac{1}{e}\right)^{2^{-m} \times 2^{n \times h_2(p+\epsilon_n)}} \end{aligned}$$

Tail inequality.
For $0 \leq \lambda \leq 0.5$, we have $\sum_{j=0}^{\lfloor \lambda \times n \rfloor} \binom{n}{j} \leq 2^{n \times h_2(\lambda)}$.
In our case: $r = \lfloor n \times p + n \times \epsilon_n \rfloor$ with $\epsilon_n = 1/\log n$. Thus $\lambda = p + \epsilon_n$.

Bit Reconciliation

Proof of theorem 4.

Thus:

$$p(Err/w(C) \leq r) \leq 1 - (\frac{1}{e})^{2^{n \times h_2(p+\epsilon_n) - m}}$$

With $m = \lceil \log \lceil \log n \rceil + n \times h_2(p + \epsilon_n) \rceil$, we have:

$$p(Err/w(C) \leq r) \leq 1 - (\frac{1}{e})^{2^{-\log \lceil \log n \rceil}} \leq 1 - (\frac{1}{e})^{1/\lceil \log n \rceil}$$

Thus: $\lim_{n \to \infty} p(Err/w(C) \leq r) = 0$

And finally: $\lim_{n \to \infty} p(Err) = 0$.

And the protocol can be easily proved to be ϵ-robust. Moreover, it is easy to see that the amount m of leaked information is asymptotically equal to $n \times h_2(p)$.

Bit Reconciliation

Proof of theorem 4.

We have proved that the algorithm is optimal. That is:

$$lim_{n\to\infty} \frac{I_E(S|Q)}{n \times h_2(p)} = 1$$

where S is the secret key established by Alice and Bob, Q is the information exchanged on the clear channel and $I_E(S|Q)$ is the amount of information that Eve has on S knowing Q.

In the algorithm, Alice has to randomly chose a function from $\{0,1\}^n$ to $\{0,1\}^m$.

We restrict this choice to a universal class of hash functions from $\{0,1\}^n$ to $\{0,1\}^m$.

Bit Reconciliation

Proof of theorem 4.

Definition. Let H be a class of functions from $\{0,1\}^n$ to $\{0,1\}^m$, we say that H is universal if for all different $x, y \in \{0,1\}^n$, we have:

$$\sharp\{f \in H \text{ s.t. } f(x) = f(y)\} \leq \frac{\sharp H}{2^m}$$

Wegman and Carter have described a universal class of hash function for which randomly choosing and evaluating functions can be efficiently achieved.

J-L. Carter and M.N. Wegman,,
Universal Classes of Hash Functions
Journ. of Computer and Systems Sciences, vol. **18**, pp. 143-154, 1979.

Another, less efficient, universal class of hashing functions could be H_3, the class of all linear functions from $\{0,1\}^n$ to $\{0,1\}^m$.

Bit Reconciliation
Proof of theorem 4.

Using a universal class of hash functions and the settings for ϵ_n, r and m already seen, we can prove the optimality of the algorithm.

Thus, we have a way to automatically generate an optimal reconciliation protocol by specifying a universal class of hash functions in a short and efficient way.

The problem is that there is no known efficient algorithms for Bob's part, i.e. to compute the decoded string B'.

And there is strong beliefs that there does not exist such an algorithm.

Bit Reconciliation
Efficiency

An ideal protocol for bit reconciliation is a protocol that is ϵ-robust, optimal and efficient, i.e. it can be executed in polynomial time.

The previous slide told us that such a protocol looks impossible. For instance, concerning H_3, it can be proved that the protocol is ideal if and only if $NP \subseteq BPP$ where BPP is the class of probabilistics Turing machine.

That is why we introduce the notion of almost-ideal protocol.

Bit Reconciliation
Efficiency

An amost-ideal protocol has an error probability approaching 0 as the length of the string increases. But the amount of leaked information is allowed to be slighty greater then the theoretical limit.This allowance is materialized by a security parameter γ.

Let us note $\tau(R_\gamma^p, n)$ the expected running time with an input data of length n.

Definition. A bit reconciliation protocol R_γ^p is almost-ideal if:

1. $\forall \epsilon \in [0,1]$, R_γ^p is ϵ-robust.
2. $lim_{n\to\infty} \frac{I_E(S/Q)}{n \times h_2(p)} \leq 1 + \gamma$
3. $\exists P(n)$ a polynomial s.t. $\exists N$ s.t. $\forall n \geq N, \tau(R_\gamma^p, n) \leq P(n)$

Bit Reconciliation
The SHELL almost-ideal protocol

The SHELL protocol uses interaction on the public channel to efficiently correct the secret strings by dividing them into fixed length blocks.

If the sub-blocks of length k have been corrected, using a subprotocol, with a decoding error probability δ_k, then the SHELL protocol will correct them by producing an amount of leaked information proportional to δ_k and $1/k$.

The SHELL protocol is constructed using the primitives:

- BINARY : correcting one error,
- CONFIRM : testing strings equality, and
- BICONF : correcting several errors.

Bit Reconciliation
The SHELL almost-ideal algorithm

BINARY : Correcting a single error.

Assume that A and B, of length n, have an odd number of errors, i.e. parities are different. Then Alice and Bob can perform a binary search to find an error by exchanging no more than $\lceil \log(n) \rceil$ bits:

1. Alice sends Bob the parity of the first half of the string.
2. Bob tells Alice if the error is in the first half or the second one.
3. The process is repeatidly applied to the half determined in step 1.
4. One erroneous bit will eventually be found and fixed.

CONFIRM : testing strings equality

If the strings of Alice and Bob are different, the algorithm will tell it with a probability of $1/2$. If they are the same, the algorithm will tell it with a probability of 1. The test:

- Alice and Bob commonly and randomly choose an identical subset of their bit strings.
- Alice and Bob compare the parities of their subsets.

Bit Reconciliation
The SHELL almost-ideal algorithm

CONFIRM(k). The test is applied k times.

- If a difference is found, then the strings are different.
- If no difference is found, then the string are considered as identical.

If strings are different, A test will not detect it with a probability of $1/2$ (an even number of erroneous bits).

If the k tests conclude to identity, the error probability is $1/2^k$.

Bit Reconciliation
The SHELL almost-ideal algorithm

Let $s > 0$, BICONF(s) : correcting several errors

1. Run CONFIRM(s), i.e. CONFIRM s times.
2. Each time an error is detected, run BINARY.

Let $\Delta^s(l|e)$ be the probability that BICONF(s) corrects l errors providing there is e errors ($e \geq l$). We have:

$$\Delta^s(l|e) = \begin{cases} \binom{s}{l} 2^{-s} & \text{if } l < e \\ \sum_{j=e}^{s} \frac{1}{2} \binom{j-1}{e-1} 2^{-(j-1)} & \text{if } l = e \end{cases}$$

Proof: counting argument.

Bit Reconciliation
The SHELL almost-ideal algorithm

Let ρ^k be any basic protocol for correction of blocks of length k.

First, Alice and Bob divide their strings into k-bits long *primary blocks*. Then:

- ρ^k is applied to each to the blocks
- For $s = 1$ to $\lceil \log(n/k) \rceil$:
  1. Alice and Bob join pair of adjacent blocks
  2. On each blocks, run BICONF(s)
     - If an error is detected, Bob's block parity is made equal to Alice's block parity

It can be shown that if the probability of error of the protocol ρ^k is strictly less than 0.5, then SHELL is almost-ideal. Thus:

If SHELL is used with previous Protocol 1, SHELL is almost-ideal (but not efficient).

Bit Reconciliation
CASCADE

CASCADE is the bit reconciliation protocol that must be implemented for QKD:

- It can be efficiently implemented.
- It is ok for $BSC(p)$ with $p < 0.15$.
- Information leaking is close to optimal.

Let $A = A_1, ..., A_n$ be the string of Alice.
Let $B = B_1, ..., B_n$ be the string of Bob.

CASCADE proceeds in several passes.

The number of passes is determined by Alice and Bob before execution. It depends on the error probability p.

Bit Reconciliation
CASCADE

CASCADE Pass 1.

1. Alice and Bob choose k_1 and divide their strings into consecutive blocks of k_1 bits.
2. Alice sends the parities of all her blocks to Bob.
3. If two parities are not equal, Alice and Bob use BINARY to fix one error.

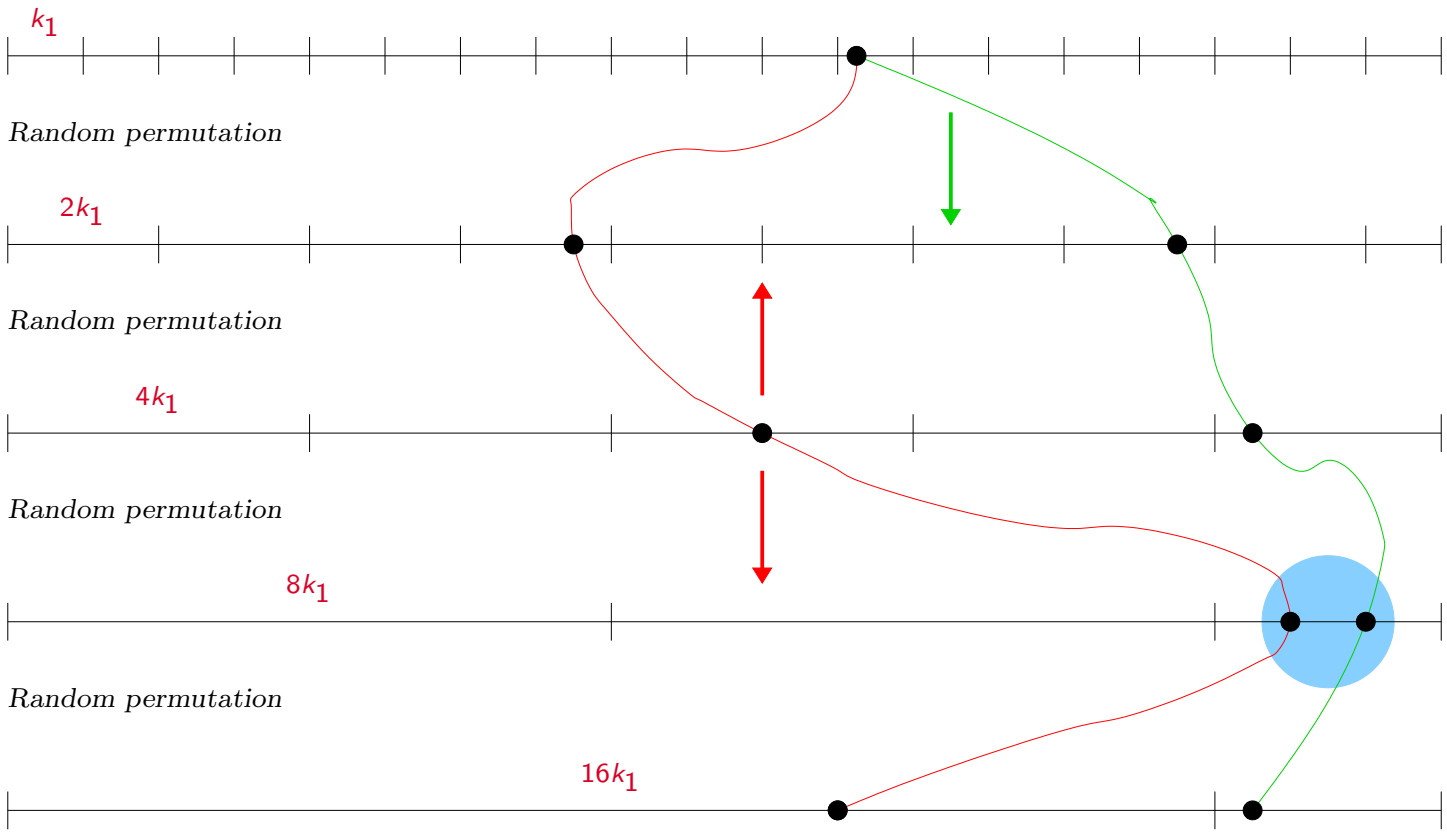
At this point, all blocks of Bob have an even number of errors, possibly 0.

Bit Reconciliation
CASCADE

CASCADE Pass i, $i > 1$.

1. Alice and Bob choose k_i and $f_i : [1, n] \to [1, \lceil n/k_i \rceil]$
2. The bits which position are in $K_j^i = \{l / f_i(l) = j\}$ form the block j
3. For all j, Alice sends $a_j = \oplus_{l \in K_j^i} A_i$, parity of its block j
4. For all j, Bob sends $b_j = \oplus_{l \in K_j^i} B_i$, parity of its block j
5. For all j, if $a_j \neq b_j$, Alice and Bob run BINARY on blocks j:
   a) BINARY fixed a bit $l \in K_j^i$
   b) Let $\mathcal{K} = \{K_u^v, 1 \leq u < i \text{ s.t. } l \in K_u^v\}$
   c) All elements of $\mathcal{K}$ have *now* distinct parities
   d) Choose one smallest K_u^v in $\mathcal{K}$ and run BINARY
   e) BINARY fixed bit $l' \in K_u^v$
   f) Let $\mathcal{K}' = \{K_u^v, 1 \leq u \leq i \text{ s.t. } l' \in K_u^v\}$
   g) $\mathcal{K} := (\mathcal{K} \cup \mathcal{K}') \backslash (\mathcal{K} \cap \mathcal{K}')$
   h) If $\mathcal{K} \neq \emptyset$ then goto c)

Bit Reconciliation
CASCADE



Bit Reconciliation
CASCADE

THEN:

- To completely eliminate the errors, run SHELL on *big* blocks obtained by the concatenation of a certain number of the *small* Pass 1 blocks.

Analysis. The algorithm eventually terminates because at most n bits have to be fixed.

The only latitude we have is choosing the block sizes. Ideally, we would like a block size such that the probability that a block K_v^1 has one or more error exponentially decreases with the number of passes.

Complex computations and simulations give the following values which depend on the error probability:

Bit Reconciliation
CASCADE

p error proba- bility	k ideal block size	$l(4)$ computed leakage 4 passes	$k \times h_2(p)$ minimum leakage 4 passes	(4) measured leakage 4 passes
0.15	5	4.12	3.05	3.80
0.10	7	3.99	3.28	3.81
0.05	15	4.64	4.01	4.60
0.01	73	6.81	5.89	6.47

A privacy amplification algorithm is used at the time when Alice and Bob share a secret random key but Eve may know a limited part of the key.

Because Alice and Bob have checked the error rate, they are able to put an upper limit on the percentage of the key which is known by Eve.

Privacy amplification is the art of distilling highly secret shared information from a larger body of shared information that is only partially secret.

C.H. Bennett, G. Brassard, C. Crepeau, U.M. Maurer, *Generalized Privacy Amplification*
IEEE Trans. on Information Theory, Vol. 41, N. 6; nov. 1995.

Let us assume that Alice and Bob share a n-bits string given by a random variable W.

Let us assume that Eve knows a correlated random variable V providing at most $t < n$ bits of information about W, i.e. $H(W/V) \geq n - t$.

Principle. Alice and Bob publicly and randomly choose a compression (hash) function $g : \{0,1\}^n \to \{0,1\}^r$ for a computed r and choose as secret the value $K = g(W)$.

And the resulting K must be uniformly distributed given Eve's information.

The length r of K may depend on n and t as well on the kind of information available to Eve. $H(W/V) \geq n - t$ could mean:

- Eve may know t arbitrary bits simply by measuring them and not being detected by BB84.
- Eve may know t parity checks of W simply by listening to the previous stages of a protocol.
- Eve may know the result of an arbitrary function mapping n-bits strings to t-bits strings.
- Eve can listen to the channel but its access has an error probability ϵ s.t. the binary entropy $h_2(\epsilon) = 1 - t/n$, thus she receives t bits from n.

The main result can be described as follow:

Condition. Eve is allowed to choose the distribution P_{VW}. The constraint is $R(W/V = v) \geq n - t$ with high probability over v, where R denotes the second order conditional (Rényi) entropy. R is described later.

Result. For any $s < n - t$, Alice and Bob can distill $r = n - t - s$ bits of secret key $K = g(W)$ such while keeping Eve's knowledge exponentially small in s.

More specifically, Alice and Bob randomly and publicly chooses a function $g : \{0,1\}^n \to \{0,1\}^{n-t-s}$ in a suitable class of functions such that:

$$H(K/G, V = v) \geq r - 2^{-s}/\log(2)$$

N.B. We assume Eve cannot tamper the channel.

Privacy Amplification
Analysing the problem

$W \in \{0,1\}^n$ is known to Alive and Bob.

$V \in \{0,1\}^n$ is known to Eve.

V is Eve's information about W.

Eve may be able to choose which partial information about W she would like to see. I.e. P_{VW} could partially be under the control of Eve and may be chosen by Eve in a set of admissible distributions.

The choice of P_{VW} may be influenced by P_W, in which case it is a conditional distribution $P_{V/W}$.

Alice and Bob usually don't know P_{VW} and may even not know P_W.

Alice and Bob publicly agree on a function $g : \{0,1\}^n \to \{0,1\}^r$ for a suitable r and they compute the key $k = g(W)$.

g can be randomly selected in order to avoid that Eve knows g before she chooses her strategy, i.e. P_{VW}.

In other words, the compression function is a random variable G taking values in a subset of functions from $\{0,1\}^n$ to $\{0,1\}^r$.

Privacy Amplification
Analysing the problem

We are interested in upper bounds of the form:

$$I(K\ ;\ G, V) < \epsilon \quad \text{for arbitrarily small } \epsilon$$

provided that P_{VW} satisfies given constraints.

Or bounds of the form:

$$H(K\ /\ G, V = v) \geq r - \epsilon$$

provided that $P_{W/V=v}$ satisfies certain constraints.

If the previous inequality is verified for a set of v with total probability at least $1 - \delta$ for a small δ then:

$$(1 - \delta) \times (r - \epsilon) \leq H(K\ /\ G, V) \leq r$$

Privacy Amplification
Analysis of the problem

The length of the distilled key r depends on P_{VW} and on the constraints that P_{VW} satisfies.

Two cases are not interesting:

- $H(W/V) = 0$, i.e. Eve knows W ;
- $H(W/V) = n$, i.e. Eve cannot obtain information about W.

Thus we assume $H(W/V) = n - t$ with $0 < t < n$ meaning that Eve knows "t bits". And we assume that Eve can choose secretly the positions of these "t bits", otherwise the problem is not very interesting...

Privacy Amplification
Existing Results

Provided $t < n$ and Eve really knows t bits, it is always possible to distill a key K on which Eve has no information except its length.

The compression (hash) function may be choosen in full view of Eve *before* she decides P_{VW}.

However, K must be much shorter than W, i.e. $r \ll n - t$. For instance, when $t = 2$, the length of K is at most $2/3$ of the length of W.

It is known that the best result are obtained if the set of functions among which g is choosen is a set of non-linear functions.

Less restrictive constraints are:

- either Eve knows t parity bits ;
- or Eve is allowed to specify a function $e : \{0,1\}^n \to \{0,1\}^t$ and to know $e(W)$.

i.e. Eve is allowed to learn t bits of information.

In the case of Quantum Cryptography, Eve can receive the bits of W through a Binary Symmetric Channel (BSC) with an error probability she can control *but* subject to global constraints over bit error probability for all the n bits.

The result we will present applies to QC.

Universal Hashing · · ·

Privacy Amplification
Universal Hashing

Définition. Let H be a class of functions from $\{0,1\}^n$ into $\{0,1\}^r$. H is universal if for any $x, y \in \{0,1\}^n$ and for $h \in H$ chosen at random with a uniform distribution, the probability that $h(x) = h(y)$ is at most $1/2^r$.

i.e. if no pair of distinct x, y collides under more that $1/2^r$ functions or, equivalently, the number of functions h for which $h(x) = h(y)$ is less than $|H|/2^r$.

When $n = r$, the class consisting only of the identity function is universal.

When $n \neq r$, the class of all functions is universal.

But these are not very interesting sets...

For a set of functions from $\{0,1\}^n$ into $\{0,1\}^r$ to be universal, a randomly chosen function must, with equal probability, map any 2 distinct points of $\{0,1\}^n$ to any values in $\{0,1\}^r$.

In other words, any 2 distinct points of $\{0,1\}^n$ must be randomly distributed throughout $\{0,1\}^r$ by the functions of the class.

J. L. Carter, M. N. Wegman,
Universal Classes of Hash Functions,
Journal of Computer and System Sciences,
n. **18**, pp. 143—154 (1979).

proposed two more interesting universal set of functions according to the following criteria:

- a function must be easily specified ;
- a function must be easily computable.

They are known as Wegman-Carter functions.

The class H_1.

Let p be a prime number greter that 2^n.
Let $a, b \in [0, p-1]$, we define $h_{a,b}$ as the function from $\{0,1\}^n$ into $\{0,1\}^r$ such that $h_{a,b}(x)$ computes the r last bits of $(a \times x + b) \mod p$:

$$\begin{array}{rccl} h_{a,b} & : & \{0,1\}^n & \rightarrow & \{0,1\}^r \\ & & x & \mapsto & ((a \times x + b) \mod p \mod 2^r \end{array}$$

The class $H_1 = \{h_{a,b},\ a, b \in [0, p-1]\}$ is a class of universal functions from $\{0,1\}^n$ into $\{0,1\}^r$.

A function of the class $H_1 = \{h_{a,b},\ a, b \in [0, p-1]\}$ is specified by a and b, that is $2 \times \log p$ bits.

The computation of $h_{a,b}(x)$ requires:

- a product in $[0, p-1]$;
- an addition in $[0, p-1]$.

The class H_3.

Let M be a $n \times r$ matrix with coefficients in $\{0,1\}$, we define the function g_M from $\{0,1\}^n$ into $\{0,1\}^r$ such that $g_M(x) = M\ x$ where x is processed as a vector.

$H_3 = \{g_M,\ M$ a $n \times r$ matrix with coefficients in $\{0,1\}\}$ is a class of universal function from $\{0,1\}^n$ into $\{0,1\}^r$.

The class H_3 is the class of all linear functions from $\{0,1\}^n$ into $\{0,1\}^r$.

A function of H_3 is specified by $n \times r$ bits.

Computing $h_M(x)$ is computing the product of a matrix and a vector.

Privacy Amplification
Universal Hashing

$\cdots$ *UniversalHashing*

Definition. The collision probability $P_c(X)$ of a random variable X taking values in $\mathcal{X}$ is defined as the probability that X takes on the same value twice on two independant experiments:

$$\begin{aligned} P_c(X) &= \sum_{x\in\mathcal{X}} p(X=x, X=x) \\ &= \sum_{x\in\mathcal{X}} p(X=x) \times p(X=x) \\ &= \sum_{x\in\mathcal{X}} p(X=x)^2 \end{aligned}$$

Privacy Amplification
Collision probability and Rényi entropy

Example. Let us assume that X is coin tossing with a regular coin, i.e. $p(X = \text{tail}) = 0.5$ and $p(X = \text{head}) = 0.5$.

If we assume two independant experiments:

$$\begin{aligned} P_c(X) &= \sum_{x \in \{\text{head},\text{tail}\}} p(X = x)^2 \\ &= 0.5^2 + 0.5^2 \\ &= 0.5 \end{aligned}$$

The Rényi entropy of order 2, or simply the Rényi entropy is defined as:

$$R(X) = -\log(P_c(X))$$

We remark:

$$\begin{aligned} H(X) &= -\sum_{x\in\mathcal{X}} p(X=x) \times \log(p(X=x)) \\ &= -E(\log(p(X)) \end{aligned}$$

$$\begin{aligned} R(X) &= -\log(\sum_{x\in\mathcal{X}} p(X=x) \times p(X=x)) \\ &= -\ log(E(p(X)) \end{aligned}$$

Example (continued). Let us assume that X is coin tossing with a regular coin, i.e. $p(X = \text{tail}) = 0.5$ and $p(X = \text{head}) = 0.5$.

$$R(X) = -log(P_c(X)) = -log(0.5) = 1$$

In this case, we have $R(X) = H(X)$ but is not always true.

The Rényi entropy conditionned on a random variable, i.e. $R(X, Y)$ is defined as for Shannon entropy:

$$R(X/Y) = \sum_{y \in \mathcal{Y}} P(Y = y) \times R(X/Y = y)$$

However, the notion of mutual information cannot be defined in the same way.

E.g. $R(X) - R(X/Y) \neq R(Y) - R(Y/X)$.

Lemma. For every discrete random variable X, we have:

$$R(X) \leq H(X)$$

Proof. Jensen's inequality: let f be concave on $[a, b]$, let $(\lambda_i)_{i=1}^n$ be positive s.t. $\sum_{i=0}^n \lambda_i = 1$, let $(x_i)_{i=1}^n$ in $[a, b]$, then $f(\sum_{i=1}^n \lambda_i x_i) \geq \sum_{i=1}^n \lambda_i f(x_i)$.
If X takes values in $\mathcal{X}$, let us choose $[a, b] = [0, 1]$, $f = \log$ is concave, let $(\lambda_i)_{i=1}^n = (p(X = x))_{x \in \mathcal{X}}$ and $(x_i)_{i=1}^n = (p(X = x))_{x \in \mathcal{X}}$ too, this gives us:

$$log(\sum_{x \in \mathcal{X}} p(X = x) \times p(X = x)) \geq \sum_{x \in \mathcal{X}} p(X = x) \times \log(p(X = x))$$

i.e. $-R(X) \geq -H(X)$, thus $R(X) \leq H(X)$.

If $\mathcal{X}$ contains n elements and X is uniformly distributed, then each $p(X = x)$ is equal to $1/n$. Then:

$$R(X) = H(X) = \log(n)$$

We can easily prove that we also have:

$$R(X/Y) \leq H(X/Y)$$

Using the following theorem, we will show that Rényi entropy can play the role of the general information measure that we are looking for.

Theorem. Let X a random variable over $\{0,1\}^n$, let G be a random variable corresponding to the random choice with uniform distribution of a member of a universal class of hash functions from $\{0,1\}^n$ to $\{0,1\}^r$, and let $Q = G(X)$, then:

$$H(Q/G) \geq R(Q/G) \geq r - \log(1 + 2^{2-R(X)}) \geq r - \frac{2^{2-R(X)}}{ln(2)}$$

Proof.

$$\begin{aligned} & R(Q/G) \\ = \; & R(G(X)/G) \\ = \; & \sum_g P(G=g) \times R(G(X)/G=g) \\ = \; & \sum_g P(G=g) \times -\log(P_c(G(X)/G=g)) \\ \geq \; & -\log(\sum_g P(G=g) \times P_c(G(X)/G=g)) \end{aligned}$$

The last step comes from Jensen's inequality.

Let us look at $\sum_g P(G=g) \times P_c(G(X)/G=g)$.

Proof. $P_c(G(X)/G = g) = P_c(g(X)) = p(g(X_1) = g(X_2))$ where X_1 and X_2 are independant random variable on $\{0,1\}^n$ with the same distribution as X. Thus:

$$\begin{aligned}
& \sum_g P(G=g) \times P_c(G(X)/G=g) \\
= \; & \sum_g P(G=g) \times p(g(X_1)=g(X_2)) \\
= \; & p(G(X_1)=G(X_2)) \\
= \; & p(X_1=X_2) \\
& + p(X_1 \neq X_2) \times p(G(X_1)=G(X_2) \;/\; X_1 \neq X_2) \\
= \; & P_c(X) + (1-P_c(X)) \times p(G(X_1)=G(X_2) \;/\; X_1 \neq X_2) \\
\leq \; & P_c(X) + \frac{1-P_c(X)}{2^r} \quad \text{(universal class)} \\
\leq \; & 2^{-R(X)} + 2^{-r} \\
= \; & 2^{-r}(1+2^{r-R(X)})
\end{aligned}$$

Privacy Amplification
Collision probability and Rényi entropy

Proof. Thus:

$$\begin{aligned} & R(Q/G) \\ \geq\ & -\log(\textstyle\sum_g P(G=g) \times P_c(G(X)/G=g)) \\ \geq\ & -\log(2^{-r}(1+2^{r-R(X)})) \\ \geq\ & r - \log(1+2^{r-R(X)}) \\ \geq\ & r - \frac{2^{r-R(X)}}{ln(2)} \end{aligned}$$

because $\log(1+y) \leq \frac{y}{ln(2)}$.

$$\begin{aligned} & ln(1+x) = x - x^2/2 + x^3/3 - x^4/4 + \cdots \\ & ln(1+x) \leq x \\ & 1 + x \leq e^x \\ & \log(1+x) \leq \log(e^x) = x \times \log(e) = x/ln(2) \end{aligned}$$

Note. This theorem also applies to conditional probabilities

Let W the n-bits string shared by Alice and Bob.

Let $V = v$ a particular value observed by Eve.

Let us assume $R(W/V = v) \geq c$.

Alice and Bob chooses $K = g(W)$ where $G = g$ is chosen at random from a universal class of hash functions from $\{0,1\}^n$ to $\{0,1\}^r$.

Then:

$$H(G(W)/G, V = v) \geq r - \log(1 + 2^{r-c}) \geq r - \frac{2^{r-c}}{ln(2)}$$

Privacy Amplification
Collision probability and Rényi entropy

We proved:

$$H(G(W)/G, V = v) \geq r - \frac{2^{r-c}}{ln(2)}$$

Thus, when $r < c$, Eve's entropy about the final key $K = G(W)$ is close to the maximal entropy, i.e. r.

Eve's information about K is $H(K) - H(K/G, V = v)$. It decreases exponentially in $c - r$.

If the probability is at least $1 - \delta$ that V takes a value v satisfying $R(W/V = v) \geq c$, then

$$\begin{aligned} H(K/G, V) &= \sum_{v \in \{0,1\}^n} p(V = v) \times H(K/G, V = v) \\ &\geq (1 - \delta)(r - \tfrac{2^{r-c}}{ln(2)}) \end{aligned}$$

and

$$I(K; G, V) \leq \delta r + (1 - \delta)\tfrac{2^{r-c}}{ln(2)} \leq \delta r + \tfrac{2^{r-c}}{ln(2)}$$

The string W shared by Alice and Bob is a n bits string with uniform distribution avec $\{0,1\}^n$.

Let $V = e(W)$ be the string observed by Eve (for an arbitrary eavesdropping function e from $\{0,1\}^n$ to $\{0,1\}^t$.

Let $s < n - t$ and let $r = n - t - s$.

If Alice and Bob choose $K = G(W)$ as this secret key where G is randomly chosen in a universal class of hash functions from $\{0,1\}^n$ to $\{0,1\}^r$, the:

$$I(K; G, V) \leq \frac{2^{-s}}{ln(2)}$$

N.B. This result is stated on V, not a particular value of V.

N.B. Alice and Bob do not care about e, they only care about t.

Proof. For $v \in \{0,1\}^t$, let c_v be the number of $w \in \{0,1\}^n$ consistent with v, i.e. such that $e(w) = v$.

Given $V = v$, all consistent w are equally candidates for W since W is uniformly distributed. Hence:

$$p(W = w/V = v) = \begin{cases} 1/c_v & \text{if } e(w) = v \\ 0 & \text{otherwise} \end{cases}$$

Then:

$$\begin{aligned} P_c(W/V = v) &= \sum_{w \in \{0,1\}^n} p(W = w/V = v)^2 \\ &= \sum_{w \in \{0,1\}^n \wedge e(w) = v} p(W = w/V = v)^2 \\ &= c_v \times \frac{1}{c_v^2} \\ &= \frac{1}{c_v} \end{aligned}$$

Proof. We have $P_c(W/V = v) = \frac{1}{c_v}$, thus:

$$R(W/V = v) = -\log(P_c(W/V = v)) = \log(c_v)$$

Then, according to previous theorem:

$$H(K/G, V = v) \geq r - \frac{2^{r-\log(c_v)}}{ln(2)} = r - \frac{2^r}{c_v \cdot ln(2)}$$

Proof. For each $v \in \{0,1\}^t$, we have:

$$\begin{aligned} p(V=v) &= \sum_{w\in\{0,1\}^n} p(V=v/W=w) \times p(W=w) \\ &= \frac{1}{2^n} \times \sum_{w\in\{0,1\}^n} p(V=v/W=w) \end{aligned}$$

And:

$$p(V=v/W=w) = \begin{cases} 1 & \text{if } w \text{ if one of the } c_v \\ & \text{consistent values} \\ 0 & \text{otherwise} \end{cases}$$

Thus:

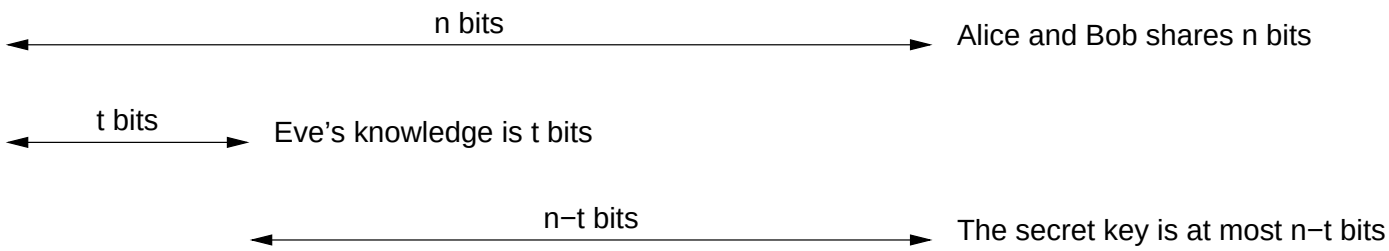
$$p(V=v) = \frac{c_v}{2^n}$$

Proof. And we conclude:

$$
\begin{aligned}
& I(K;G,V) \\
= \;& H(K) - H(K/G,V) \\
= \;& r - \sum_{v\in\{0,1\}^t} P(V=v) \times H(K/G,V=v) \\
\leq \;& r - \sum_{v\in\{0,1\}^t} P(V=v) \times (r - \frac{2^r}{c_v \times ln(2)}) \\
= \;& r - \sum_{v\in\{0,1\}^t} P(V=v) \times r \\
& \quad + \sum_{v\in\{0,1\}^t} P(V=v) \times \frac{2^r}{c_v \times ln(2)} \\
= \;& r - r + \sum_{v\in\{0,1\}^t} P(V=v) \times \frac{2^r}{c_v \times ln(2)} \\
= \;& \sum_{v\in\{0,1\}^t} P(V=v) \times \frac{2^r}{c_v \times ln(2)} \\
= \;& \sum_{v\in\{0,1\}^t} \frac{c_v}{2^n} \times \frac{2^r}{c_v \times ln(2)} \\
= \;& \sum_{v\in\{0,1\}^t} \frac{2^{r-n}}{ln(2)} \\
= \;& 2^t \times \frac{2^{r-n}}{ln(2)} \\
= \;& \frac{2^{t+r-n}}{ln(2)} \\
= \;& \frac{2^{-s}}{ln(2)}
\end{aligned}
$$

Conclusions. At the beginning, Alice and Bob shares a key of n bits from which Eve knows t bits.

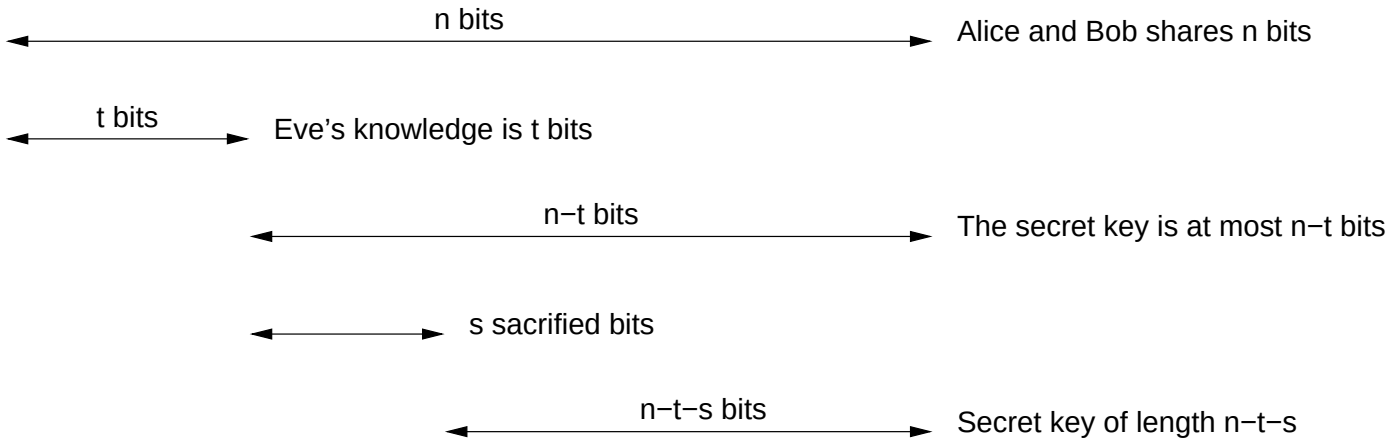
The maximum length for the final key is $n - t$.



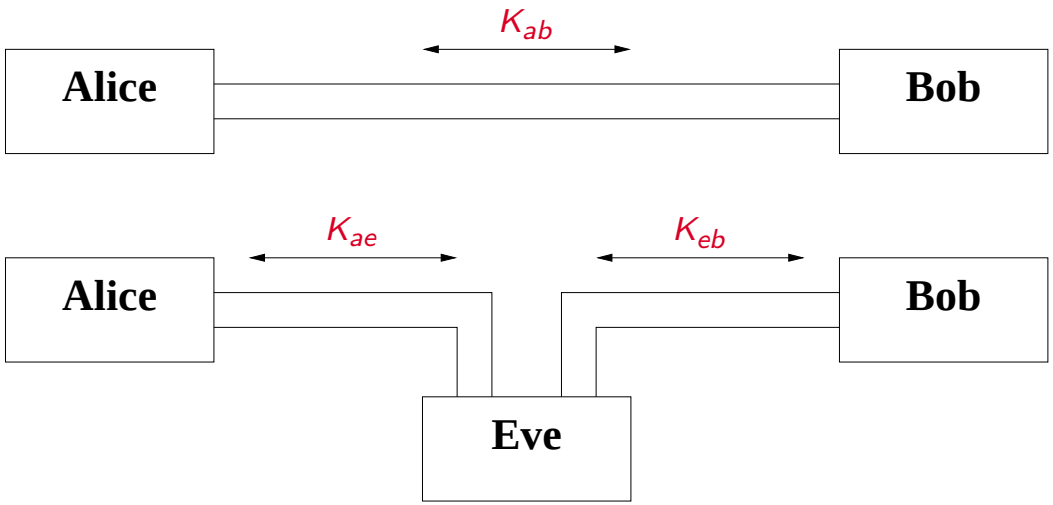
Privacy Amplification

Main theorem

Conclusions. The previous result states that we can reduve Eve's knowledge about the final key to any $\epsilon > \frac{2^{-(n-t)}}{ln(2)}$ by reducing the length of the final key in the following manner: we choose s such that $\epsilon > \frac{2^{-s}}{ln(2)}$ and $r = n - t - s$ in the previous algorithm:



One problem we have is that the protocol we described is subject to Man-in-the-middle attacks:



Authentication
Stating the problem

If Eve is in position of doing a Man-in-the-middle attack, then:

- Alice may think she is establishing a key with Bob meanwhile she is establishing a key with with Eve.
- Bob may think he is establishing a key with Alice meanwhile he is establishing a key with with Eve.
- Eve could decode the message from Alice to Bob using Alice-Eve key and re-encode them using the Eve-Bob key before resending them to Bob.

Thus, there is a need for Alice and Bob to be sure that each one has established a key with the other one and not another person.

Authentication
Stating the problem

Another type of attack is the Denial of Service (DoS) attack.

In this type of attack, Eve tries to use a service so many times that the service becomes denied:

- For instance, she can continuously try to log on a system so that no other user can try to log on. The login service can be considered as dead. The same can be done with a HTTP server.
- Another more perverse attack can occur when each tentative of using the service uses a resource that is consumed. As the resource is usually not infinite, it may be exhausted by Eve and the service is no more available.

Authentication
Stating the problem

Unfortunately, classical (numerical) authentification is a process subject to Denial of Service attack.

Classical authentication uses an authentication key. Each time the authentication process is executed, a part of the key is consumed and has to be renewed.

That means that the entropy of the authentication key is decreased each time the authentication process is executed.

Reducing the information leakage about the key or having a very large bank of authentication keys does not solve the problem. This only delays the apparition of the denial of the service.

Authentication
Stating the problem

How authentification is done ?

Before the transaction, Alice and Bob shares an authentication key k.

When Alice sends a message x to Bob, she also sends an authenticator, that is a value $y = f(x, k)$ for a given f.

It is nearly the equivalent of handwritten signature on classical letters sent over the unsecure post office channel. However, Bob can forge a false Alice signature.

The authenticator is sometimes called an authentication tag. And f is called the authentification function.

How Eve attacks ?

When Eve intercepts the communications, she has the message x, the authentication function f and the authentication tag $y = f(x, k)$.

Eve may want to replace (x, y) by (x', y') where $y' = f(x', k)$. She has to guess k.

Eve can compute $\mathcal{K} = \{k'/f(x, k') = y\}$, the set of all keys compatible with the message (x, y). Of course, $k \in \mathcal{K}$.

- If $\mathcal{K}$ contains only one element, Eve knows k and is able to replace (x, y) by a forged message (x', y').
- Otherwise, she has to randomly choose a k in $\mathcal{K}$ and she has $1/|\mathcal{K}|$ probability of succeeding.

VIIVEKE FAK,
Repeated Use of Codes which Detect Deception
IEEE Trans. on Information Theory, Vol. 25, N. 2, march 1979.

shows that if the authentification keys are r bits strings, i.e. $k \in \{0,1\}^r$, the best security we can obtain is that the probability of success of Eve can be reduced to $\frac{1}{2^r}$.

Of course, the authentication key must be used only once. If the authentication key is used N times, then the probability of success of Eve is increased to $\frac{1}{2^{\frac{r}{N}}}$

And authentication is subject to DoS attack.

Authentication
Stating the problem

The result of the article can be summarized as follows:

- If the length of the authentication keys is r, then the best security we can obtain is $\frac{1}{2^r}$. This is the probability of success of a cheating Eve.
- This security level can be reached only if the key is used only once. And r bits of information, the length of the key, are lost.

Authentication
Stating the problem

In this formulation, we considered an authentication key k and an authentication function $f(\cdot,\cdot)$ used to compute the authentication tag $y = f(x,k)$ of the message x.

Another formulation is to say that Alice and Bob shares an authentication function $a(\cdot)$ chosen from a set of functions. And that the authentication tag is $y = a(x)$.

These two formulations are equivalent:

- From first to second: the set of functions is defined as the set of functions $f(\cdot,k)$ for all the keys k.
- From second to first: use the computability theory out of scope of this course.

Authentication
Stating the problem

We have several choices:

- To try to use quantum tools to authenticate the quantum transaction at the beginning of the protocol, sometimes called quantum authentication.
- To authenticate all the classical transactions between Alice and Bob, could be called protocol authentication.
- To authenticate the keys at the end of the transaction, could be called key authentication.

Quantum Authentication. A possible way of doing this is to authenticate the quantum states communications. The situation is as follows:

- It requires an authentication key previously positionned between Alice and Bob.
- Optimal Quantum Authentication exists when the quantum equipment is perfect. However, it is not the case.
- It does not seem that optimal Quantum Authentication exists when the quantum equipment are not perfect.

Protocol Authentication. One way to proceed to authentication is to authenticate all the messages on the classical channel.

However, if we want a security of $\frac{1}{2^{128}}$, for instance, then we have to sacrifice 128 bits of prepositionned key for each message.

And there is a lot of messages in the protocol, especially bit reconciliation and privacy amplification.

Key Authentication. An finally, a way to proceed to authentication is to authenticate the final secret key.

As follows:

- We assume that Alice and Bob share two prepositionned authentication keys k_1 and k_2.
- Alice and Bob choose a security level, for instance $\frac{1}{2^{128}}$ corresponding to the success probability of a cheating Eve.
- The final secret key is split into K_A and K_B.
- Alice authenticates K_A with k_1.
- Bob authenticates K_B with k_2.
- In both cases, 128 bits (at least) are leaked to Eve.
- These 256 bits are renewed using bits of the secret key (secret growing).

Authentication
Using Wegman-Carter functions

Thus, we look for an authentication algorithm that reaches the optimal limit in term of security and information leakage.

And once again, we go back the universal class of hash functions given by Wegman and Carter.

M. N. Wegman, J. L. Carter
New Hash Functions and their Use in Authentication and Set Equality
Journ. of Computer and System Sciences, Vol. 22,
pp. 265-279, 1981.

This article also gives a new proof of FAK optimality result.

Authentication
Using Wegman-Carter functions

Main results of Wegman-Carter article:

- They define a almost-universal class of hash functions from $\{0,1\}^n$ into $\{0,1\}^r$ where a function can be specified in $\mathcal{O}((r + \log\log(n)) \times \log(n))$ instead of $\mathcal{O}(n)$ in the case of universal class of hash functions.
- An application of this class is a provably secure authentication techniques for sending messages over insecure lines.
- An enemy, even one with infinite computer resources, cannot forge or modify a message.

Definition

For $k \geq 2$, a class H of functions from $\{0,1\}^n$ into $\{0,1\}^r$ is strongly universal$_k$ if for all $a_1, \ldots, a_k \in \{0,1\}^n$ and all $b_1, \ldots, b_k \in \{0,1\}^r$, there exist $|H|/2^{rk}$ functions that maps a_1 to b_1,...,a_k to b_k.

The class H is strongly universal$_\omega$ if it is strongly universal$_n$ for all $n \geq 2$.

Remarks

We have already seen Wegman-Carter strongly universal$_2$ classes of functions.

Strongly universal$_n$ class of functions can be built using polynomials of degree less than n over finite fields: there exists exactly one polynomial of degree less than n which interpolates through the n designated pairs.

Authentication
Definitions (Wegman-Carter)

Let $M = \{0,1\}^n$ be the set of message, let $T = \{0,1\}^r$ be the set of tags, let F be a set of functions mapping M into T, we define an authentication scheme as follows: Alice and Bob agree on a fucntion $f \in F$; when sending a message m, Alice also sends $f(m)$, the tag ; Bob checks the identity of Alice by controlling the tag.

An authentication scheme is unbreakable with certainty p if given any message m, its tag $f(m)$ and a message $m' \neq m$, Eve can guess $f(m')$ with a probability at most p. In other word, the probability of successful cheating is at most p.

Authentication
A simple scheme

To create an authentication system which is unbreakable with certainty p, we choose $T = \{0,1\}^r$ to have at least $1/p$ elements. And we let F be a strongly universal$_2$ class of hash functions from $M = \{0,1\}^n$ to T.

Given a message m and its tag $f(m)$, we define F' to be the subset of F of the functions which maps m to $f(m)$. Eve may be able, with enough time and computation power, to compute F' and she knows that $f \in F'$.

Following the definition of strongly universal$_2$, for $m' \neq m$, the proportion of functions of F' which maps m' to a particular tag t' is $1/|T| = 1/2^r$. Since $|T| = 2^r > 1/p$, any choice the forger makes has no more than a probability of p of being correct.

However...

Authentication
A simple scheme

However, no one knows how to build an admissible strongly universal$_2$ class of hash functions for that purpose.

The difficulty with all known strongly universal$_2$ class of hash functions for that purpose is that the set of functions is so large that specifying a function requires a key as long as the messages...

A second problem is that the key may be used only once to get the optimal security...

Authentication
Stating the problem

The goal of the Wegman-Carter article is:

- To define a set of functions that can be specified with a key of reasonnable length. This will be a set of almost-strongly universal functions instead of a set of universal functions.
- To be able to do authentication of multiple messages with a reasonable cost. Depending on the desired security level, a reasonnable number of bits will be sacrified for each message authentication.

Authentication
Almost stongly universal class of hash functions

We wish to construct a set of hash functions from $\{0,1\}^n$ into $\{0,1\}^r$.

Let $s = r + \log\log(n)$.

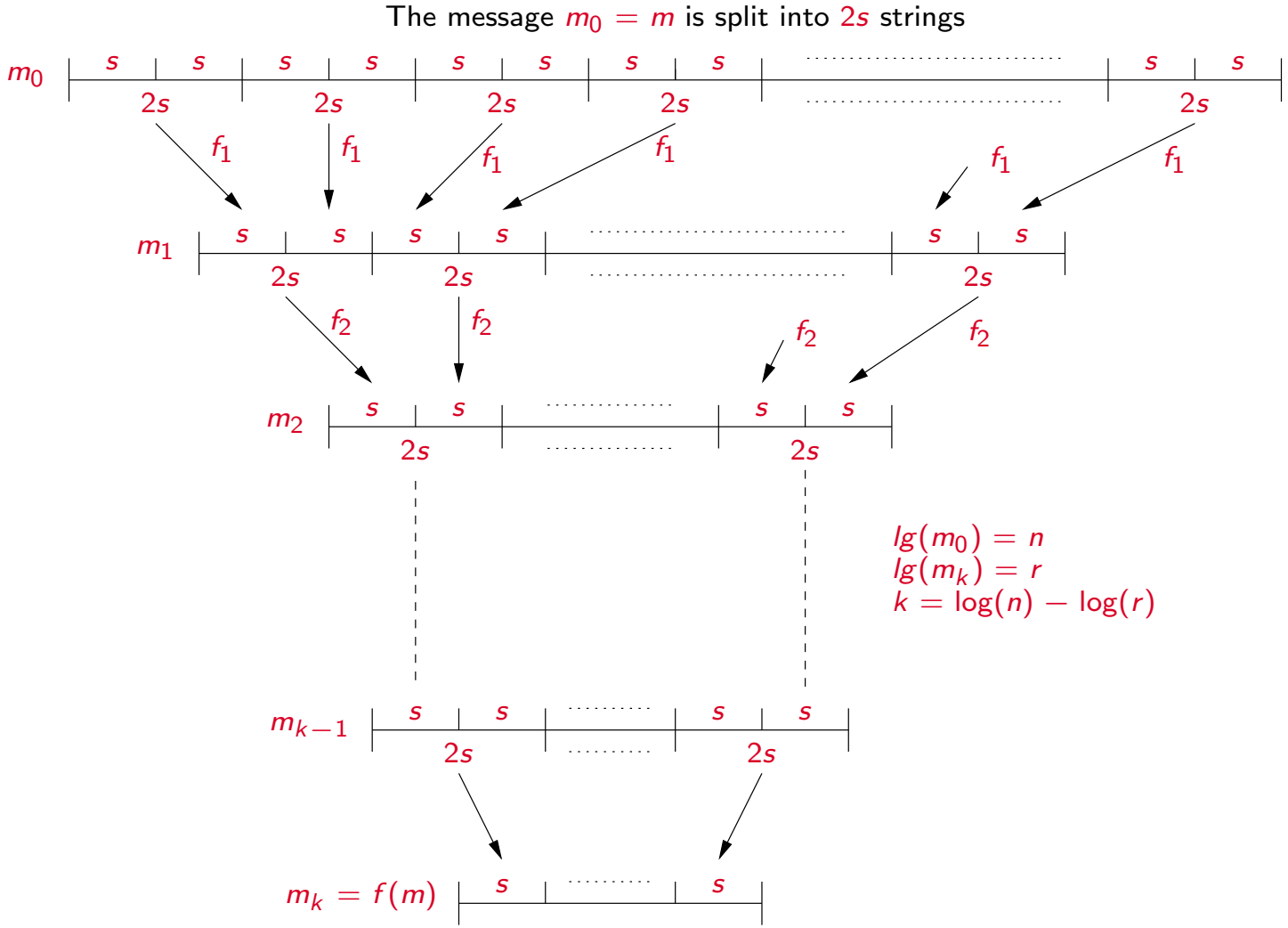
Let H be a strongly universal$_2$ class of hash functions from $\{0,1\}^{2s}$ into $\{0,1\}^s$. For instance H_1 or H_3 already seen.

We define H' a set of H functions from $\{0,1\}^n$ into $\{0,1\}^r$ as follows. Let $k = \log(n) - \log(r)$. Each element of H' is a sequence $f = (f_1, f_2, \ldots, f_k)$ of elements of H.

Let $f = (f_1, f_2, \ldots, f_k)$ where each f_i is a hash function from $\{0,1\}^{2s}$ into $\{0,1\}^s$. For $m \in \{0,1\}^n$, we define $f(m)$:

Authentication

Almost stongly universal class of hash functions



For $i = 1, \ldots, k$, we write $m_i = f_i^*(m_{i-1})$.

Authentication
Almost stongly universal class of hash functions

Let us take $H = H_1$ as the base strongly universal$_2$ class of hash functions.

$$H_1 = \left\{ \begin{array}{lccl} f: & \{0,1\}^{2s} & \to & \{0,1\}^s \\ & x & \mapsto & (a \times x + b) \mod 2^s \end{array} \text{ with } a, b \in \{0,1\}^{2s} \right\}$$

The specification of a function $f \in H$ is a, b.
Its length is $4s$.

Thus, the specification of $f = (f_1, \ldots, f_k)$, with $k = \log(n) - \log(r)$ has length $4s \times (\log(n) - \log(r))$, i.e. $4(r + \log\log(n))(\log(n) - \log(r))$ which is a $\mathcal{O}((r + \log\log(n)) \times \log(n))$

This is a big improvment compared to previous schemes H_1 and H_3. However $\cdots$

However $\cdots$ one cannot prove that the class H' is a strongly universal$_2$ class of hash functions.

One can prove that H' is almost universal$_2$ in the following sense:

Definition

Given two distinct messages $a, b \in \{0,1\}^n$, given two tags $t_a, t_b \in \{0,1\}^r$, the probability that a function takes a to t_a is $1/2^r$. And among the functions that takes a to t_a, the probability that a function also takes b to t_b is $2/2^r$.

Authentication

Almost stongly universal class of hash functions

Let us compare. Given two distinct messages $a, b \in \{0,1\}^n$, given two tags $t_a, t_b \in \{0,1\}^r$:

Almost strongly universal$_2$

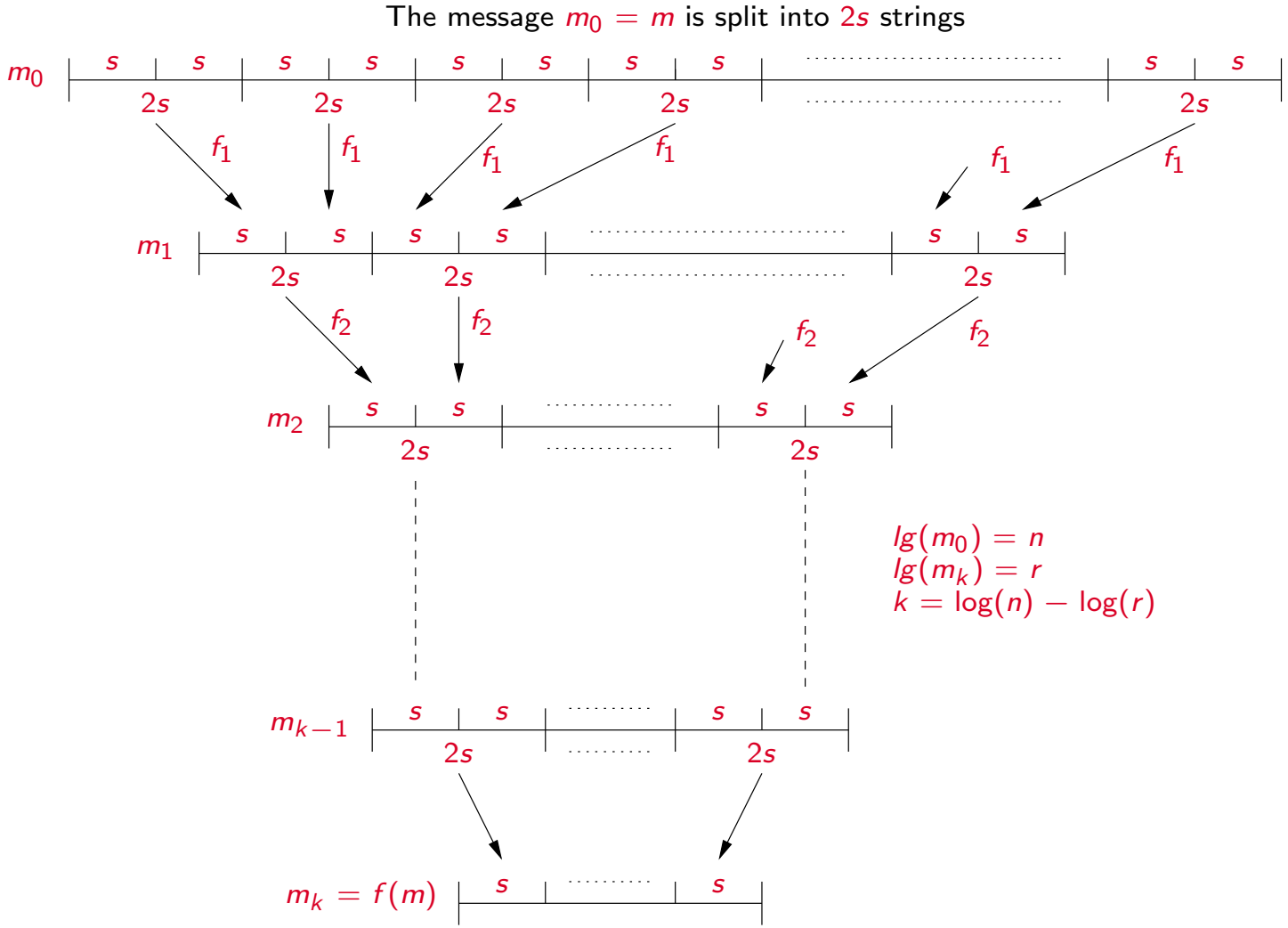
- $p(f(a) = t_a \ / \ f \in H') = 1/2^r$
- $p(f(b) = t_b \ / \ f \in H' \wedge f(a) = t_a) = 2/2^r$
- After looking at (a, t_a), the probability of sucessful cheating is $2/2^r$.

Strongly universal$_2$

- $p(f(a) = t_a \ / \ f \in H') = 1/2^r$
- $p(f(b) = t_b \ / \ f \in H' \wedge f(a) = t_a) = 1/2^r$
- After looking at (a, t_a), the probability of sucessful cheating is $1/2^r$.

Authentication

Almost stongly universal class of hash functions



For $i = 1, \ldots, k$, we write $m_i = f_i^*(m_{i-1})$.

Authentication
Almost stongly universal class of hash functions

Let us prove that H' is almost-universal$_2$:

- When computing $f(a)$ and $f(b)$, we have $a_i = f_i^*(a_{i-1})$ and $b_i = f_i^*(b_{i-1})$ for $i = 1, \ldots, k$ with $k = \log(n) - \log(r)$.
- At each stage we halve the length of a and b, the probability that $a_i = b_i$ provided that $a_{i-1} \neq b_{i-1}$ is less than $1/2^s$, the worst case that occurs when all the $2s$ parts of a are identical and all the $2s$ parts of b are identical.
- Thus, the probability that $a_{k-1} = b_{k-1}$ is less than $(k-1)/2^s$ which is less than $\log(n)/2^s$ and $\frac{\log(n)}{2^s} = \frac{\log(n)}{2^{r+\log\log(n)}} = \frac{1}{2^r}$.
- The fact that f_k is chosen from a strongly universal$_2$ class of hash functions can be used to show that $f = (f_1, \ldots, f_k)$ maps a to any tag t_a with equal probability over the f. And this probability is $1/2^r$ since there is 2^r tags.
  - Then, if $a_{k-1} \neq b_{k-1}$, $1/2^r$ if these functions will map b to t_b. Thus, if $t_a \neq t_b$, this implies that $a_{k-1} \neq b_{k-1}$, less than $1/2^r$ of these functions will map b to t_b.
  - If $t_a = t_b$, as above: less $1/2^r$ of these functions will map b to t_b if $a_{k-1} \neq b_{k-1}$. But less than $1/2^r$ of all the functions are such that $a_{k-1} = b_{k-1}$.
  - The total is less than $2/2^r$.

Authentication
Almost stongly universal class of hash functions

We have proven that we can build a class of hash functions from $\{0,1\}^n$ into $\{0,1\}^r$. Given:

- $s = r + \log\log(n)$ and H be a strongly universal$_2$ class of hash functions from $\{0,1\}^{2s}$ into $\{0,1\}^s$.
- H' be the set of sequences $f = (f_i)_{1 \leq i \leq k}$ with $k = \log(n) - \log(r)$, these functions being applied as described in the previous slides.

Then:

- The specification of a function $f \in H'$ requires $\mathcal{O}((r + \log\log(n))\log(n))$ bits.
  That means that if H' is used for authentication, the length of the key is $\mathcal{O}((r + \log\log(n))\log(n))$ bits.
- Given a message a and a tag t_a, the probability that a $f \in H'$ maps a to t_a is $1/2^r$.
- Given a distinct message b and a tag t_b, the probability that such a $f \in \{g \in H' / g(a) = t_a\}$ maps b to t_b is $2/2^r$.
  That means that if the authentication keys are used only once, then the probability of a successful cheating is at most $2/2^r$ which can be made as small as required.

Using the almost strongly universal functions of Wegman-Carter, we can authenticate n-bits messages using $\mathcal{O}((r + \log\log(n))\log(n))$ authentication keys and r-bits tags for a security of $1/2^{r-1}$.

Let us examine a realistic case where we want to authenticate messages of length 2^{16}, that is 65536 bits. Let us decide we want a security of $1/2^{63}$, that is approximatively 10^{-19} which could be qualified as absolute security. Then the tags are 64 bits, that is very reasonnable. However, the authentication key length is $4(64 + \log\log(2^{16}))(\log(2^{16}) - \log(64)) = 3264$ bits.

Thus, each time we authenticate a 64 Kb messages, we loose a little more than 3 Kb key. Is it possible to do better without decreasing the security ?

How to do better ?

Let us imagine that we pre-position a randomly chosen function $f \in H'$ and a r-bits randomly chosen mask b between Alice and Bob. When sending a message m, let us choose $t = f(m) \oplus b$ as the authentification tag. And the mask b is lost. Next time, another mask b' will be used.

For Eve, $H(b) = r$ since the mask b was randomly choosen. Eve knows m and t. Eve has a message m' and would like to forge an authentification tag t' such that $t' = f(m') \oplus b'$. But she does not know anything about b'. Because $H(b') = r$ for Eve, we also have $H(f(m') \oplus b') = r$ and Eve has a probability $1/2^r$ of guessing the good tag t'.

Thus, if the mask is discarded after use, there is no need for changing the function f.

The method is described as follows:

- Let us assume that Alice wants to send and authenticate $n \geq 1$ messages $m_1, \ldots, m_n$ to Bob with a security of $1/2^r$.
- Alice and Bob preposition a function $f \in H'$.
- Alice and Bob preposition n randomly chosen masks $b_1, \ldots, b_n$ of bit-lengh r.

Then for $i = 1, \ldots, n$, Alice sends the message m_i with its tag $t_i = f(m_i) \oplus b_i$.

Even after reading (m_i, t_i) for $i = 1, .., n-1$, if Eve wants to forge a tag $t' = f(m') \oplus b_n$, she can only randomly choose t' with a probability of success of $1/2^r$ because $H(b_n) = r \Rightarrow H(t') = r$ for Eve.

When authenticating k multiple messages in the above condition of security, we use a pre-positionned function and k masks.

The specification of the function is $4(r - \log\log(n))(\log(n) - \log(r))$ bits. The length of a mask is r bits. Therefore we use:

$$\sigma(n, r, k) = 4(r - \log\log(n))(\log(n) - \log(r)) + k \times r \text{ bits}$$

And:

$$\frac{\lim_{k\to\infty} \sigma(n, r, k)}{k} = r$$

Conclusions

Getting a security of $1/2^r$ with tags of length r is the maximum that one could get.

A security of $1/2^r$ with tags of length r means that even with infinite computer power and infinite time, Eve can do nothing more that randomly choosing a tag for her cheating message.

Wegman-Carter strongly universal functions allows to asymptotically reach the optimal number of bits required; i.e. r, by each authentication as the total number of authentications increases.

And if we use almost strongly universal functions, we have the same results but functions can be specified in $\mathcal{O}((r + \log\log(n)) \times \log(n))$ bits.

Bibliography

J. Preskill, cours avec exercices,
`http://www.theory.caltech.edu/~preskill/`.

M. Nielsen et I. Chuang,
Quantum Computation and Quantum Information,
Cambridge University Press, Cambridge (2000).

C. Crepeau, cours en ligne,
`http://www.cs.mcgill.ca/~crepeau/COURSES/teaching.html`.

C.H. Bennett, G. Brassard, C. Crepeau, U.M. Maurer,
Generalized Privacy Amplification
IEEE Trans. on Information Theory, Vol. 41, N. 6; nov. 1995.

U.M. Maurer,
Secret Key Agreement by Public Discussion from Common Information
IEEE Trans. on Information Theory, Vol. 39, N. 3; may 1993.