1. **LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| I | The relationship between no. of Rounds and Key Size in AES | 9 |
| II | S- Box Table for Substitute Bytes Round in AES | 14 |
| III | Inverse S- Box Table for Substitute Bytes Round in AES | 15 |
| IV | Photon outputs for different input and filter scenarios in BB84 Protocol | 23 |
| V | Table for BB84 without Eve | 24 |
| VI | Table for BB84 with Eve | 29 |

1. **LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
|  | Structure of a Cryptosystem | 6 |
|  | Symmetric Cryptographic system | 7 |
|  | Asymmetric Cryptographic system | 8 |
|  | AES structure schematics | 9 |
|  | AES Key Expansion Algorithm | 11 |
|  | Substitute bytes stage of the AES algorithm | 14 |
|  | AES Encryption round | 17 |
|  | Structure of a Quantum Cryptosystem | 20 |
|  | Basic idea if Quantum Key distribution | 21 |
|  | Overview of BB84 Protocol | 22 |
|  | Filter scenarios in BB84 Protocol | 23 |
|  | Pictorial representation of BB84 | 25 |
|  | Final Key after Sifting in BB84 | 28 |
|  | Encryption of data by AES using key obtained from QKD | 34 |
|  | Micius distributing keys | 36 |

1. **CRYPTOGRAPHY**

**1. INTRODUCTION**

**Cryptography:** “An Art of coding and decoding the messages.”

The word ‘cryptography’ was coined by combining two Greek words, ‘Krypto’ meaning hidden and ‘graphene’ meaning writing.

The basic idea is **to modify a message so as** **to make it unintelligible or Enigmatic to anyone other than the intended recipient.**

Cryptographers create ciphers - **Cryptography**

Cryptanalyst break ciphers – **Cryptanalysis**

Cryptography prior to the modern age was effectively synonymous with [***encryption***](https://en.wikipedia.org/wiki/Encryption)**,** the conversion of information from a readable state to apparent [**nonsense**](https://en.wikipedia.org/wiki/Nonsense). The originator of an encrypted message shared the decoding technique needed to recover the original information only with intended recipients, thereby precluding unwanted persons from doing the same. The cryptography literature often uses the name ***Alice*** ("A") for the sender, ***Bob*** ("B") for the intended recipient, and ***Eve*** ("[eavesdropper](https://en.wikipedia.org/wiki/Eavesdropper)") for the adversary.

Modern cryptography is heavily based on mathematical theory and computer science practice; cryptographic algorithms are designed around [computational hardness assumptions](https://en.wikipedia.org/wiki/Computational_hardness_assumption), making such algorithms hard to break in practice by any adversary. It is theoretically possible to break such a system, but it is infeasible to do so by any known practical means.

Cryptographers create ciphers - **Cryptography**

Cryptanalyst break ciphers – **Cryptanalysis**

[Cryptanalysis](https://en.wikipedia.org/wiki/Cryptanalysis) is the term used for the study of methods for obtaining the meaning of encrypted information without access to the key normally required to do so; i.e., it is the study of how to crack encryption algorithms or their implementations.

**2. HISTORICAL BACKGROUND**

The roots of cryptography are found in Roman and Egyptian civilizations.

**Hieroglyph** – The Oldest Cryptographic Technique The first known evidence of cryptography can be traced to the use of ‘hieroglyph’. Some 4000 years ago, the Egyptians used to communicate by messages written in hieroglyph. This code was the secret known only to the scribes who used to transmit messages on behalf of the kings.

Later, the scholars moved on to using simple mono-alphabetic substitution ciphers during 500 to 600 BC. This involved replacing alphabets of the message with other alphabets with some secret rule. This rule became a key to retrieve the message back from the garbled message.

The earlier Roman method of cryptography, popularly known as the **Caesar Shift Cipher**, relies on shifting the letters of a message by an agreed number (three was a common choice), the recipient of this message would then shift the letters back by the same number and obtain the original message.

Improved coding techniques such as **Vigenere** Coding came into existence in the 15th century, which offered moving letters in the message with a number of variable places instead of moving them the same number of places.

In the early 20th century, the invention of mechanical and electromechanical machines, such as the **Enigma** rotor machine, provided more advanced and efficient means of coding the information.

**3. SECURITY SERVICES OF CRYPTOGRAPHY:**

The primary objective of using cryptography is to provide the following four fundamental information security services.

**Confidentiality** - Confidentiality is the fundamental security service provided by cryptography. It is a security service that keeps the information from an unauthorized person. It is sometimes referred to as privacy or secrecy. Confidentiality can be achieved through numerous means starting from physical securing to the use of mathematical algorithms for data encryption.

**Data Integrity -** It is security service that deals with identifying any alteration to the data. The data may get modified by an unauthorized entity intentionally or accidentally. Integrity service confirms that whether data is intact or not since it was last created, transmitted, or stored by an authorized user. Data integrity cannot prevent the alteration of data but provides a means for detecting whether data has been manipulated in an unauthorized manner.

**Authentication -** Authentication provides the identification of the originator. It confirms to the receiver that the data received has been sent only by an identified and verified sender. Authentication service has two variants:

∙ Message authentication identifies the originator of the message without any regard router or system that has sent the message.

∙ Entity authentication is assurance that data has been received from a specific entity, say a particular website.

Apart from the originator, authentication may also provide assurance about other parameters related to data such as the date and time of creation/transmission.

**Non-repudiation** - It is a security service that ensures that an entity cannot refuse the ownership of a previous commitment or an action. It is an assurance that the original creator of the data cannot deny the creation or transmission of the said data to a recipient or third party.

Non-repudiation is a property that is most desirable in situations where there are chances of a dispute over the exchange of data. For example, once an order is placed electronically, a purchaser cannot deny the purchase order, if non-repudiation service was enabled in this transaction.

**4. CRYPTOGRAPHY PRIMITIVES:**

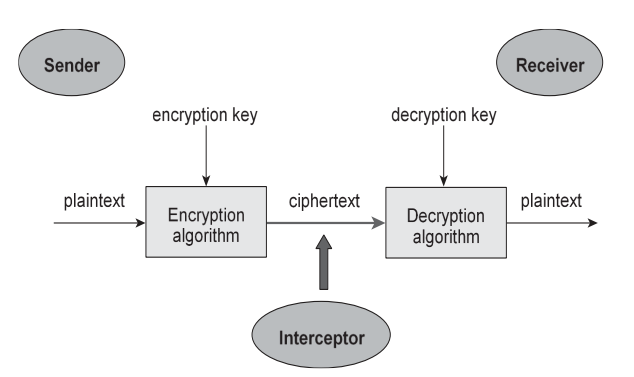
Cryptography primitives are nothing but the tools and techniques in Cryptography that can be selectively used to provide a set of desired security services:

∙ Encryption

∙ Hash functions

∙ Message Authentication Codes (MAC)

∙ Digital Signatures

**5. STRUCTURE OF A CRYPTOSYSTEM:** 

**6. TYPES OF CRYPTOGRAPHY:**

Fundamentally, there are two types of cryptosystems based on the manner in which encryption-decryption is carried out in the system:

∙ Symmetric Key Cryptography

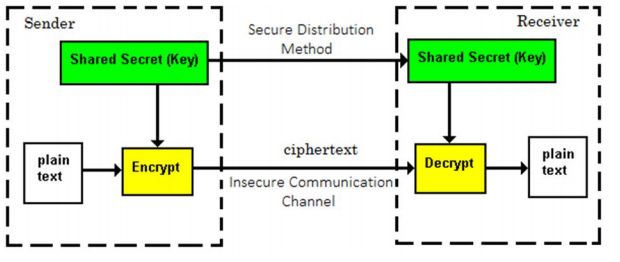
∙ Asymmetric Key Cryptography

The main difference between these cryptosystems is the relationship between the encryption and the decryption key. Logically, in any cryptosystem, both the keys are closely associated. It is practically impossible to decrypt the ciphertext with the key that is unrelated to the encryption key

**6.1 SYMMETRIC CRYPTOGRAPHY:**

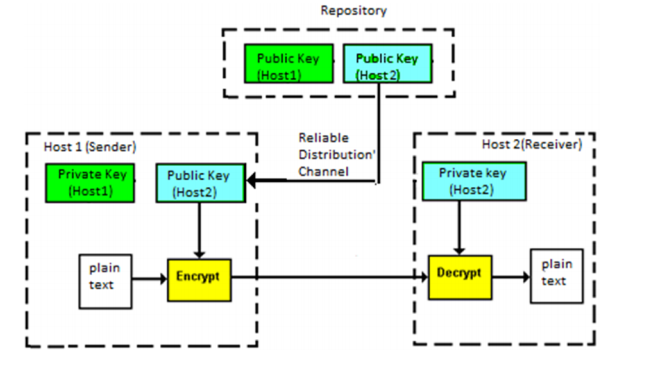
The encryption process where same keys are used for encrypting and decrypting the information is known as Symmetric Key Encryption.

The study of symmetric cryptosystems is referred to as symmetric cryptography. Symmetric cryptosystems are also sometimes referred to as secret key cryptosystems. A few well-known examples of symmetric key encryption methods are: Digital Encryption Standard (DES), Triple-DES (3DES), IDEA, and BLOWFISH



**6.2 ASYMMETRIC CRYPTOGRAPHY:**

The encryption process where different keys are used for encrypting and decrypting the information is known as Asymmetric Key Encryption. Though the keys are different, they are mathematically related and hence, retrieving the plaintext by decrypting ciphertext is feasible. The process is depicted in the following illustration:



**7. ADVANCED ENCRYPTION STANDARD (AES):**

The more popular and widely adopted symmetric encryption algorithm likely to be encountered nowadays is the Advanced Encryption Standard (AES).

Like DES, AES is a symmetric block Algorithm. This means that it uses the same key for both Encryption and Decryption.

The [National Institute of Standards and Technology (NIST)](https://searchsoftwarequality.techtarget.com/definition/NIST) started development of AES in 1997 when it announced the need for a successor algorithm for the [Data Encryption Standard (DES)](https://searchsecurity.techtarget.com/definition/Data-Encryption-Standard), which was starting to become vulnerable to [brute-force attacks](https://searchsecurity.techtarget.com/definition/brute-force-cracking).

However, AES is quite different from DES in a number of ways. The algorithm Rijndael allows for a variety of block and key sizes and not just the 64 and 56 bits of DES’ block and key size. It is found at least six time faster than triple DES.

It is not a Feistel algorithm. Recall that in a Feistel structure, half of the data block is used to modify the other half of the data block and then the halves are swapped.

NIST specified the new advanced encryption standard algorithm must be a block cipher capable of handling 128-bit blocks, using keys sized at 128, 192, and 256 bits. The block and key can, in fact, be chosen independently from 128, 160, 192, 224, 256 bits and need not be the same.

The features of AES are as follows:

1) A symmetric key symmetric block cipher

2) Encrypts 128-bit data, 128/192/256-bit keys

3) Stronger and faster than Triple-DES

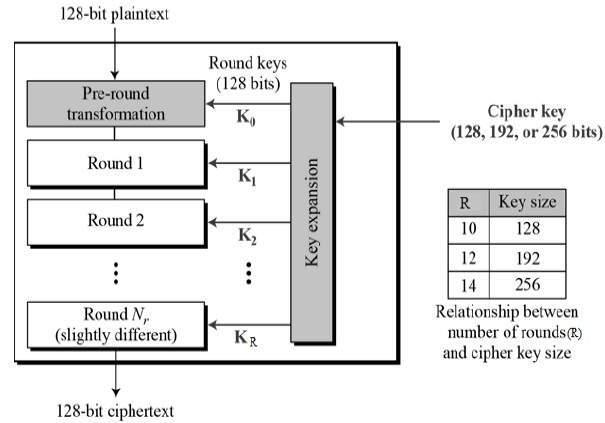
4) Hardware and Software implementable as well (in C, Python and Java)

5) AES is an iterative rather than Feistel cipher

Depending on which version is used, the name of the standard is modified to AES-128, AES-192 or AES256 respectively.

A number of AES parameters depend on the key length. For example, if the key size used is 128 then the number of rounds is 10 whereas it is 12 and 14 for 192 and 256 bits respectively. At present, the most common key size likely to be used is the 128-bit key.

The schematic of AES structure is given in the following illustration



The input is a single 128-bit block both for decryption and encryption and is known as the **in** the matrix. This block is copied into a **state** array which is modified at each stage of the algorithm and then copied to a **State** matrix.

Both the plaintext and key are depicted as a 128-bit square matrix of bytes. This key is then expanded into an array of key schedule words (the w matrix).

It must be noted that the ordering of bytes within the matrix is by column. The same applies to the w matrix.

The algorithm begins with an Add round key stage followed by 9 rounds of four stages and the tenth round of three stages. This applies for both encryption and decryption with the exception that each stage of a round the decryption algorithm is the inverse of its counterpart in the encryption algorithm.

**7.1 PREREQUISITE INFORMATION:**

AES encrypts input Blocks of Size 128 bits or 16 bytes or 4 words. Here, we are using AES -128 Algorithm. So, the key of size 128 bit or 16 bytes or 4 words.

The AES key expansion algorithm takes as input a 4-word key and produces a linear array 44 words. Each Subkey is of size 1 word or 32 bits. Each Round uses 4 subkeys or 4 words(128 bits).

Before First Round, Add round key Transformation is Performed which uses 4 subkeys. So, in total 44 subkeys(4x10=40 +4=44) are used.

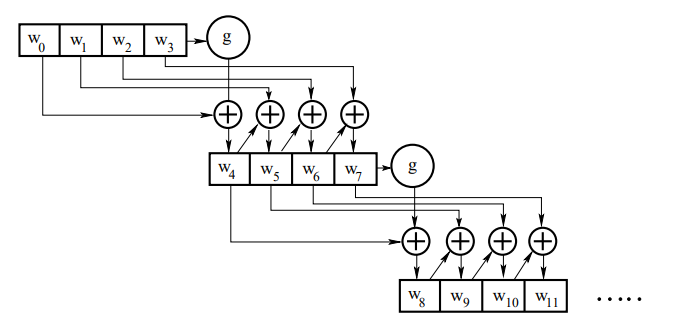
**7.2 AES KEY EXPANSION ALGORITHM:**

The AES key expansion algorithm takes as input a 4-word key and produces a linear array of 44 words. Each round uses 4 of these words as shown in figure 7.2. Each word contains 32 bytes which mean that each subkey is 128 bits long.

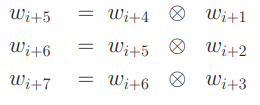
Let’s say that we have the four words of the round key for i th round: wi wi+1 wi+2 wi+3 For these to serve as the round key for i th round, I must be a multiple of 4. These will obviously serve as the round key for the (i/4)th round. For example, w4, w5, w6, w7 is the round key for round 1, the sequence of words w8, w9, w10, w11 the round key for round 2, and so on.

Now we need to determine the words wi+4 wi+5 wi+6 wi+7

from the words wi wi+1 wi+2 wi+3.



From Figure above, we write



Note that except for the first word in a new 4-word grouping, each word is an XOR of the previous word and the corresponding word in the previous 4-word grouping.

So now we only need to figure out wi+4. This is the beginning word of each 4-word grouping in the key expansion. The beginning word of each round key is obtained by:



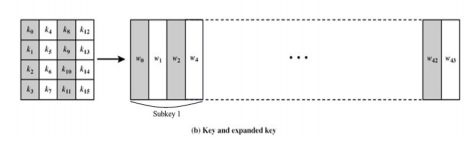
That is, the first word of the new 4-word grouping is to be obtained by XOR’ing the first word of the last grouping with what is returned by applying a function g() to the last word of the previous 4-word grouping.

The function g() consists of the following three steps: –

1. Perform a one-byte left circular rotation on the argument 4- byte word.
2. Perform a byte substitution for each byte of the word returned by the previous step by using the same 16 × 16 lookup table as used in the SubBytes step of the encryption rounds.
3. XOR the bytes obtained from the previous step with what is known as a round constant. The round constant is a word whose three rightmost bytes are always zero. Therefore, XOR’ing with the round constant amounts to XOR’ing with just its leftmost byte.

The round constant for i th round is denoted Rcon[i]. Since, by the specification, the three rightmost bytes of the round constant is zero, we can write it as shown below. The left-hand side of the equation below stands for the round constant to be used in i th round. The right-hand side of the equation says that the rightmost three bytes of the round constant are zero.

The round constant is different for each round and is defined as Rcon[j] = (RC[J], 0,0,0), with RC[1]= 1, RC[j]= 2• RC[j − 1] and with multiplication defined over the field GF(2^8)



**7.3 STAGES IN EACH ROUND:**

The first nine rounds of the Encryption algorithm consist of the following:

1. Substitute bytes
2. Shift rows

3. Mix Columns

4. Add Round Key

The tenth round simply leaves out the Mix Columns stage.

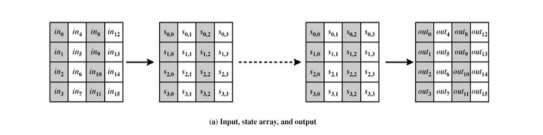
The first nine rounds of the Decryption algorithm consist of the following:

1. Inverse Shift rows
2. Inverse Substitute bytes
3. Inverse Add Round Key
4. Inverse Mix Columns

Again, the tenth round here also leaves out the Inverse Mix Columns stage.

**a) SUBSTITUTE BYTES:**

The 128-bit input is converted to 4x4 Matrix with Each Element is a Byte(8x16=128).



This stage (known as SubBytes) is simply a table lookup using a 16×16 matrix of byte values called an s-box. This matrix consists of all the possible combinations of an 8-bit sequence (2^8 = 16 × 16 = 256). However, the s-box is not just a random permutation of these values and there is a well-defined method for creating the s-box tables.

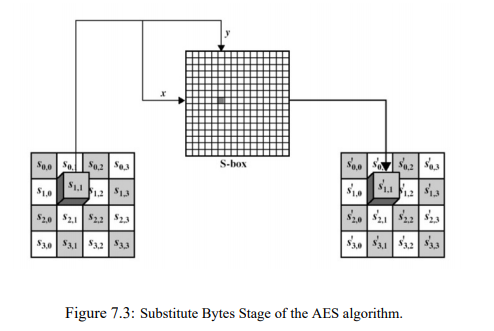
Each Element of in matrix is of 8 bit. First four bits correspond to Rows(0 to 15) and Remaining 4 bits corresponds to Columns(0 to 15) of the table.

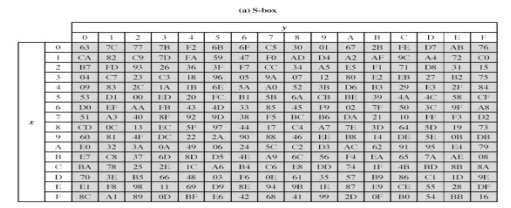
Based on this, we will get a unique value for Each element of in matrix.

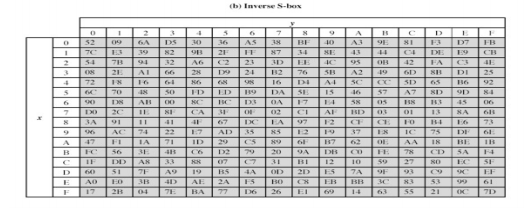
Then, these Values are stored in Intermediate Matrix called **State** Matrix.

The Inverse substitute byte transformation (known as InvSubBytes) makes use of an **inverse s-box.**

The s-boxes are designed to be resistant to known cryptanalytic attacks.







**b) SHIFT ROW TRANSFORMATION:**

This stage (known as Shift Rows) is shown in the figure below. This is a simple permutation nothing more.

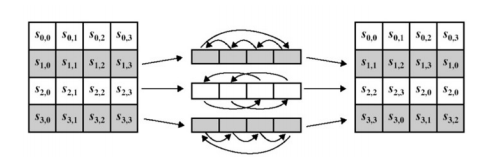
It works as follows:

1. The first row of the state is not altered.

2. The second row is shifted 1 bytes to the left in a circular manner.

3. The third row is shifted 2 bytes to the left in a circular manner.

4. The fourth row is shifted 3 bytes to the left in a circular manner

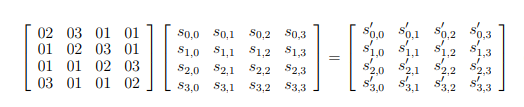
****

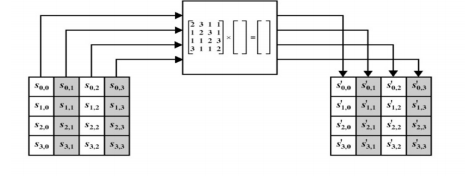
The Inverse Shift Rows transformation (known as InvShiftRows) performs these circular shifts in the opposite direction for each of the last three rows (the first row was unaltered to begin with).

**c) MIX COLUMN TRANSFORMATION:**

This stage (known as MixColumn) is Shown in the figure below. Each column is operated on individually.

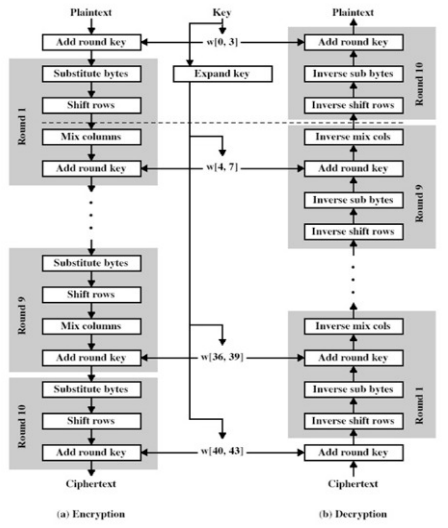
Here, State matrix is multiplied with a 4x4 Matrix to get a matrix which is stored in a State Matrix.

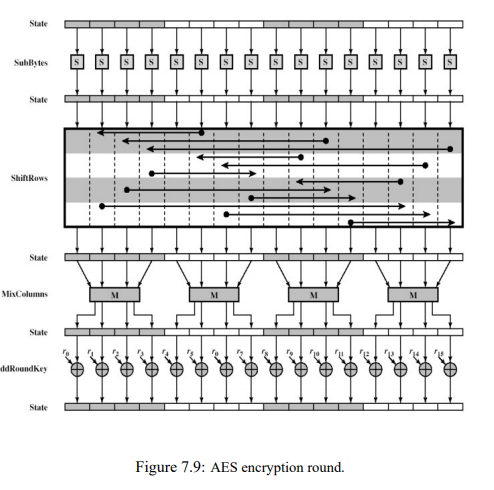
****

****

**d) ADD ROUND KEY TRANSFORMATION:**

In this stage (known as AddRoundKey) the 128 bits of state are bitwise XORed with the 128 bits of the round key. The operation is viewed as a column-wise operation between the 4 bytes of a state column and one word of the round key.

****

****

**4. QUANTUM CRYPTOGRAPHY**

**1. INTRODUCTION**

Quantum cryptography is a new method for secret communications offering the ultimate security assurance of the inviolability of a Law of Nature. Quantum cryptography draws its strength from the weirdness of reality at small scales. One of the principal problems of cryptography is the so-called “**Key distribution problem**.” How do the sender and intended recipient come into possession of secret key material while being sure that third parties (“eavesdroppers”) cannot acquire even partial information about it? It is probably impossible to establish a secret key with conventional communications, and so key distribution has relied on the establishment of a physically secure channel (“trusted couriers”) or the conditional security of “difficult” mathematical problems in public key cryptography. However, provably secure key distribution becomes possible with **Quantum communications**. It is this procedure of key distribution that is accomplished by quantum cryptography, and not the transmission of an encrypted message itself. Hence, a more accurate name is **Quantum key distribution (QKD).**

**Quantum cryptography** is a technique which uses Quantum Mechanics to provide secure communication between two parties.

**Quantum Key Distribution (QKD)** is a part of Quantum Cryptography in which key will be distributed among two parties securely by the application of **Quantum Mechanics**

The advantage of quantum cryptography lies in the fact that it allows the completion of various cryptographic tasks that are proven or conjectured to be impossible using only classical (i.e. non-quantum) communication. For example, it is [**impossible to copy**](https://en.wikipedia.org/wiki/No-cloning_theorem) data encoded in a quantum state without changing its state and the very act of reading data encoded in a quantum state [**changes the state**](https://en.wikipedia.org/wiki/Observer_effect_(physics)#Quantum_mechanics). This could be used to detect eavesdropping in quantum key distribution.

**2. HISTORICAL BACKGROUND**

The origins of quantum cryptography can be traced to the work of Wiesner, who proposed that if single-quantum states could be stored for long periods of time they could be used as counterfeit-proof money.

* In the early 1970, Stephen Wiesner firstly introduced the concept of quantum conjugate coding in New York.
* In 1982 Richard Feynman suggested individual quantum systems could be used for computation.
* In 1984, Charles H. Bennett, of the IBM Thomas J. Watson Research Center, and Gilles Brassard, of the Université de Montréal, proposed a method for secure communication based on Wiesner’s “conjugate observables”.
* In 1990, Arthur Ekert, then a Ph.D. student of Oxford University, developed a different approach to quantum cryptography known as quantum entanglement.
* Then in 1998, Nicholas Gisin at the University at Geneva was able to demonstrate the polarization of photons over many cables

**3. WHY IS QUANTUM CRYPTOGRAPHY SECURE?**

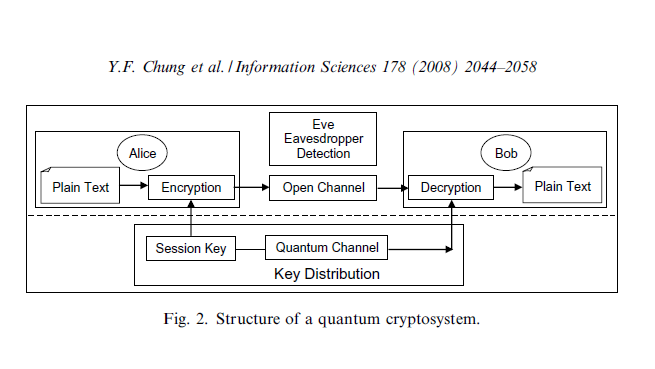
The most obvious security feature of QKD is that it is impossible to “tap” single quantum signals in the conventional sense. At a deeper level, QKD resists 3 interceptions and retransmission by an eavesdropper because in quantum mechanics, in contrast to the classical world, the result of a measurement cannot be thought of as revealing a “possessed value” of a quantum state. Moreover, Heisenberg’s uncertainty principle ensures that the

**It is impossible to determine Momentum and Position of Quantum State at the same time.**

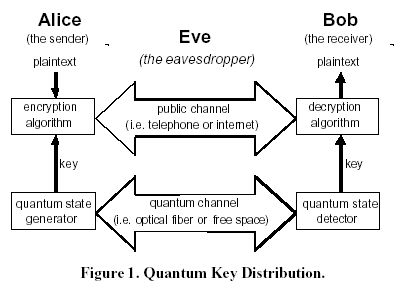
In case of Quantum Theory, this principle implies that it is impossible to detect and extract information from Photon at the same time that means the state of Photon will change if one copies its state.

These changes will introduce an anomalously high error rate in the transmissions between the sender and intended recipient, allowing them to detect the attempted eavesdropping. Thus, the two important security features of QKD are that eavesdroppers cannot reliably acquire key material, and any attempt to do so will be detectable.

**3. STRUCTURE OF A QUANTUM CRYPTOSYSTEM:**

****

**4. BASIC IDEA:**

****

**5. LIST OF QKD PROTOCOLS:**

* [**BB84**](https://en.wikipedia.org/wiki/BB84)**:** Proposed by Bennett and Brassard in 1984.
* [**Decoy state protocol**](https://en.wikipedia.org/wiki/Decoy_state_protocol)**:** A practical QKD scheme using imperfect single photon sources, such as weak [coherent state](https://en.wikipedia.org/wiki/Coherent_state) sources
* [**E91 protocol**](https://en.wikipedia.org/wiki/E91_protocol)**:** entanglement protocol
* [**COW protocol**](https://en.wikipedia.org/w/index.php?title=COW_protocol&action=edit&redlink=1)**:** coherent one-way protocol by Gisin
* [**DPS protocol**](https://en.wikipedia.org/w/index.php?title=DPS_protocol&action=edit&redlink=1)**:** differential phase shift by Yamamoto
* [**KMB09 protocol**](https://en.wikipedia.org/wiki/KMB09_protocol)**:** High Error-rate QKD protocol by Khan et al.
* [**S09 protocol**](https://en.wikipedia.org/w/index.php?title=S09_protocol&action=edit&redlink=1)**:** Protocol with Private-Public Key
* [**S13 protocol**](https://en.wikipedia.org/w/index.php?title=S13_protocol&action=edit&redlink=1)**:** Quantum Key Distribution From A Random Seed

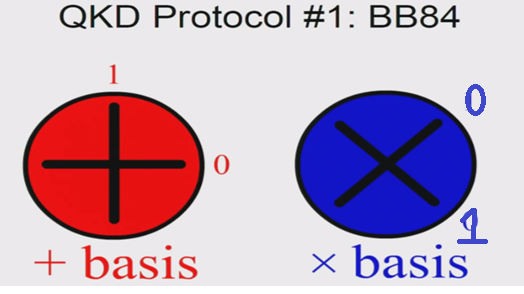
**6. BB84 PROTOCOL:**

**6.1 OVERVIEW OF BB84 PROTOCOL:**

BB84 is a [quantum key distribution](https://en.wikipedia.org/wiki/Quantum_key_distribution) scheme developed by [Charles Bennett](https://en.wikipedia.org/wiki/Charles_H._Bennett_(computer_scientist)) and [Gilles Brassard](https://en.wikipedia.org/wiki/Gilles_Brassard) in 1984. It is the first [quantum cryptography](https://en.wikipedia.org/wiki/Quantum_cryptography) [protocol](https://en.wikipedia.org/wiki/Quantum_cryptography_protocol). The protocol is [provably secure](https://en.wikipedia.org/wiki/Provable_security), relying on the quantum property that information gain is only possible at the expense of disturbing the signal if the two states one is trying to distinguish are not orthogonal.

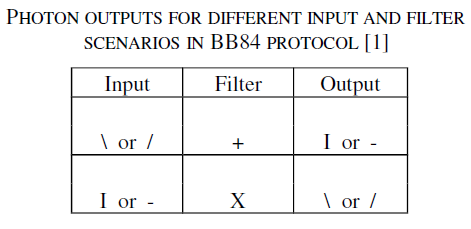
Its first experimental demonstration was performed in 1991. This protocol makes use of photon polarization states. The data sent in the channel is encoded by means of linear Polarisation states. These states cannot be measured without disturbing the original state and such quantum characteristic ensures the security of the whole system. This characteristic is often referred to as quantum indeterminacy.

To simplify the whole procedure it is assumed that a photon might be polarized in one of four possible directions, i.e. 0,45,90,135 degrees. Furthermore, some convention of a bit representation for photon orientations is essential. From the table below it can be observed that if a photon is vertical or 45 degrees tilted then its corresponding binary representation will be 1. Simultaneously all horizontal or 135 degrees tilted photons will be represented by 0.

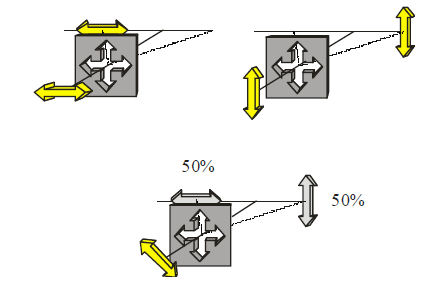


The whole system must be equipped with two polarization filters. Two pairs of states are used in BB84 protocol and they are always conjugate to each other. States within a single pair are orthogonal to each other and known as a basis. The commonly used orientations of basis are the rectilinear basis (vertical – 0 deg and horizontal – 90 deg) and diagonal basis (tilt of 45 deg and 135 deg).

For clarity, the rectilinear filter is denoted by + and diagonal one by X. A rectilinear filter detects correctly rectilinearly-oriented photons, whereas a diagonal filter diagonally-oriented photons. In other words, whatever the orientation of the photon hitting the filter it will always be detected but in statistically half cases the result will be wrong.

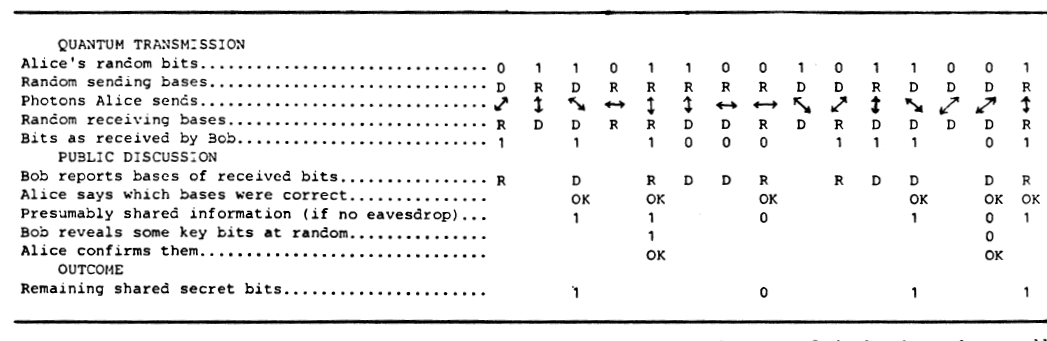


When the orientation of the photon is not same as that of receiving polariser it will be detected with a Probability=0.5 i.e.statistically in half of the cases the result will be wrong as shown

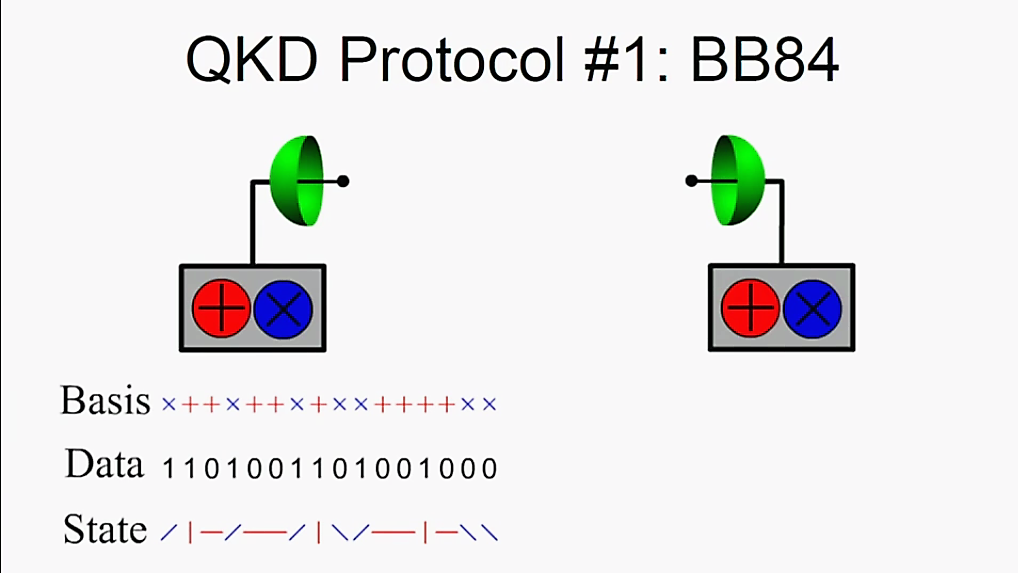


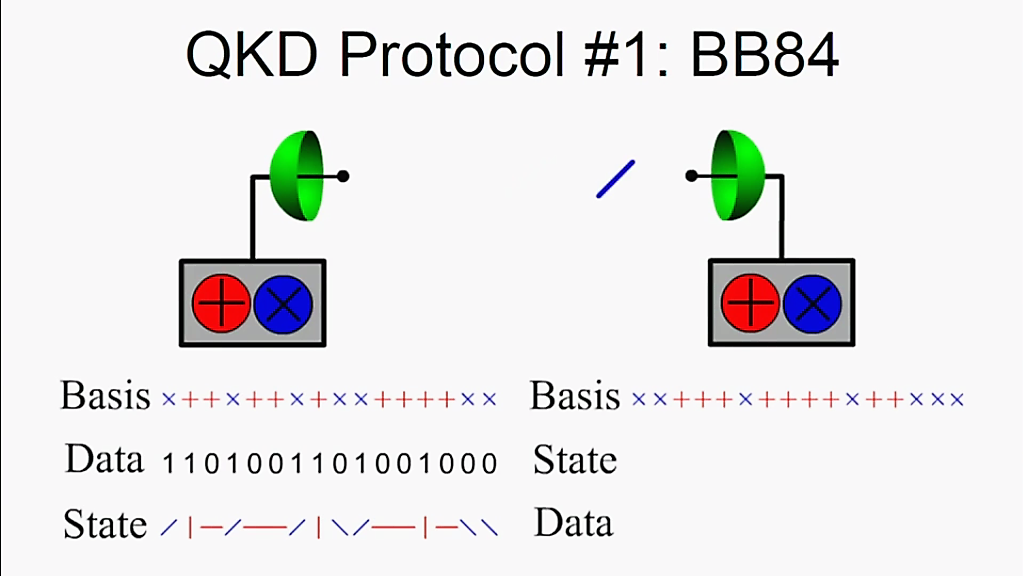
**6.2 STEPS OF BB84 PROTOCOL:**

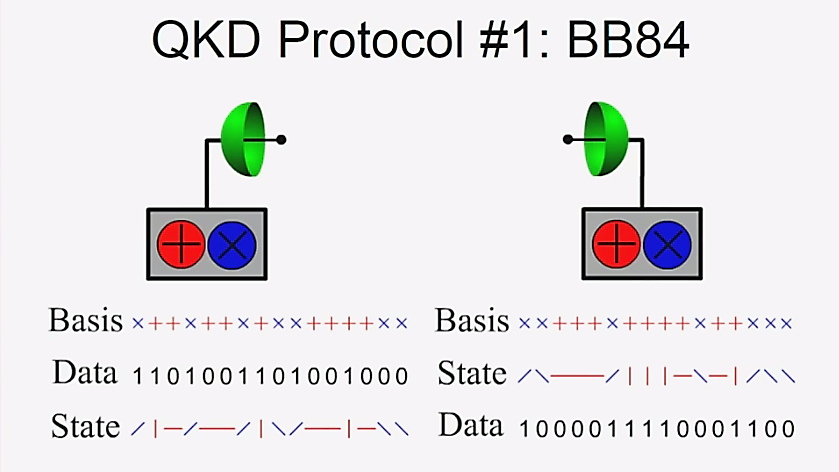
1. Sender Alice creates a random sequence of bits of length n, switching between rectilinear and diagonal basis and then sends it to recipient Bob, taking notes of state, basis and time of each photon sent.
2. As Bob does not know of which basis the incoming photons are polarized, he must switch randomly between two types of Bases. For every single photon, he takes notes of which Basis he used and the binary value that he obtained for the given Basis.
3. After the transmission process is completed, Bob needs to inform Alice which Basis was used for every single photon.
4. Then, Alice must also share which Basis he used to polarise a given photon at his end.
5. All bits for which the Basis used are discarded at both ends whereas the remaining bits constitute the key. The maximum length of key obtained will be n/2.
6. Initially, Alice and Bob use Hash Function on some of their bits and Exchange their Hash values. When the incoming Hash Value Matches with their own Hash value then they are sure that there is no attacks on transmission medium. If Hash values don’t match then, they will come to know that Some Disturbance or noise caused these errors and then they will discard the key and they will restart the whole process.
7. Above steps were shown pictorially below :

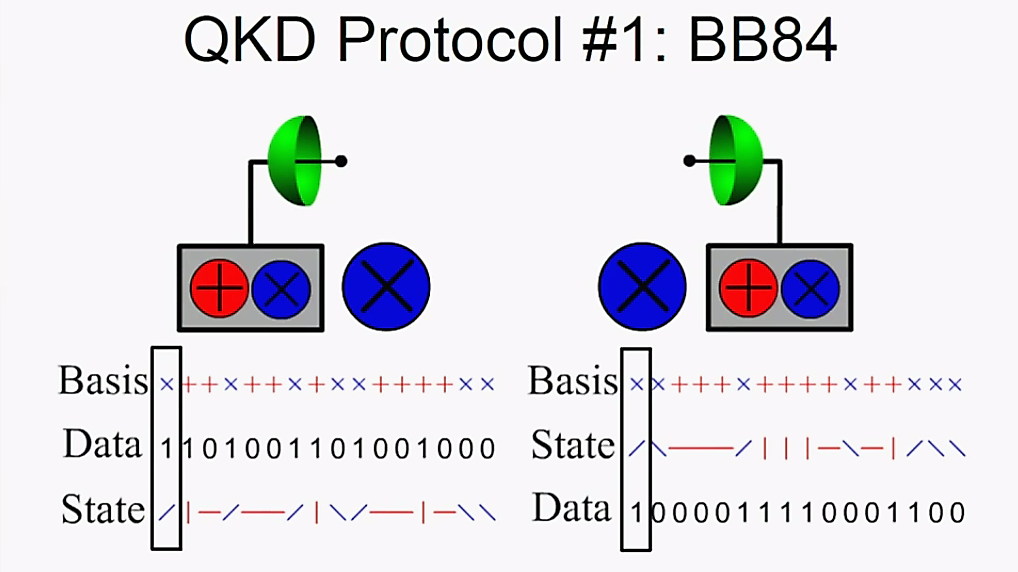
****

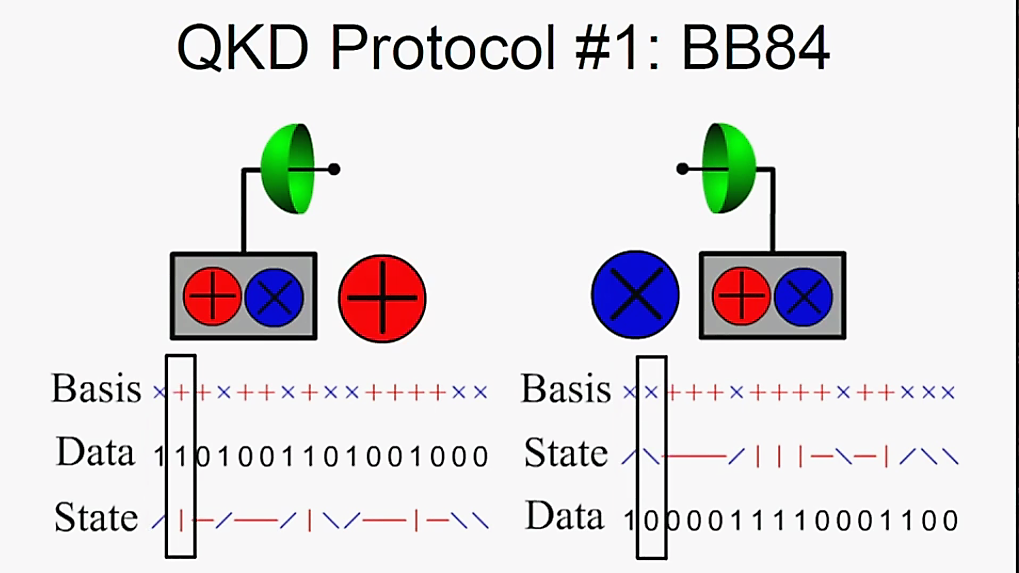
**6.3 PICTORIAL REPRESENTATION OF BB84:**

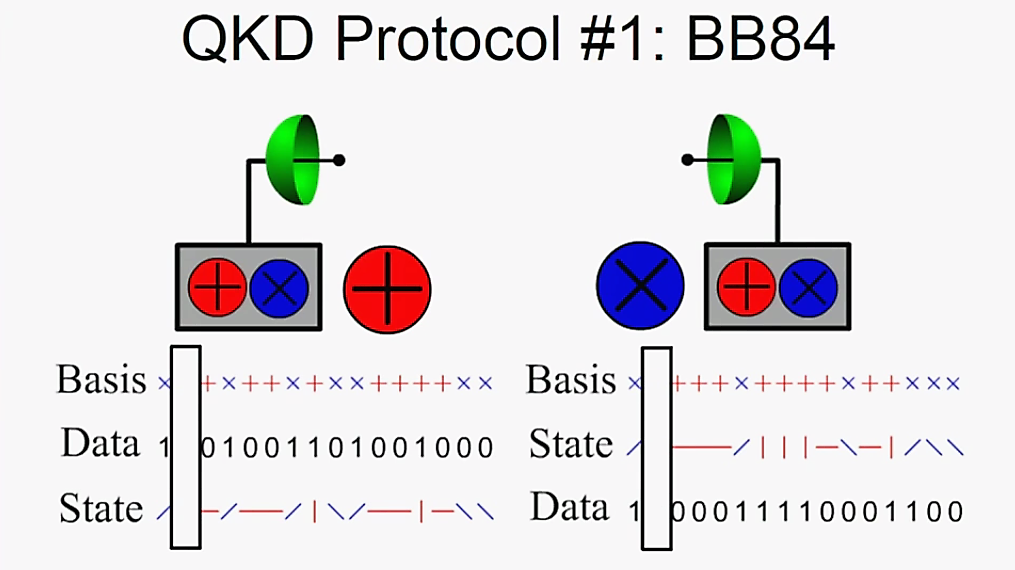
****

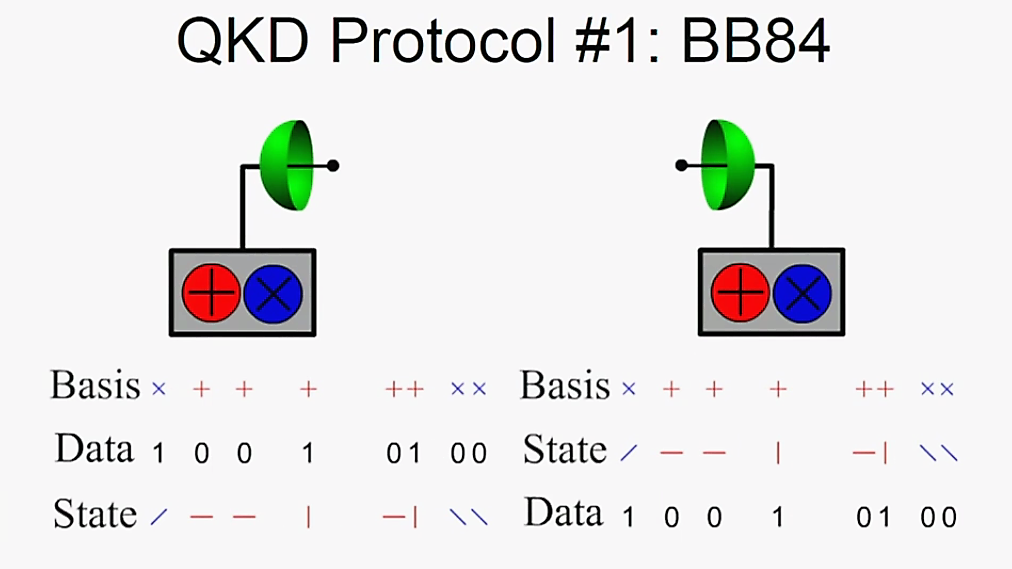
****

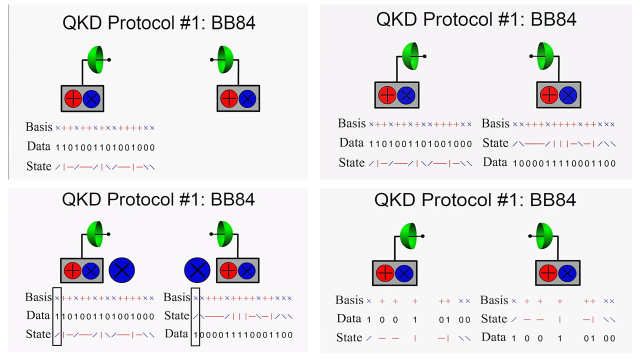
****

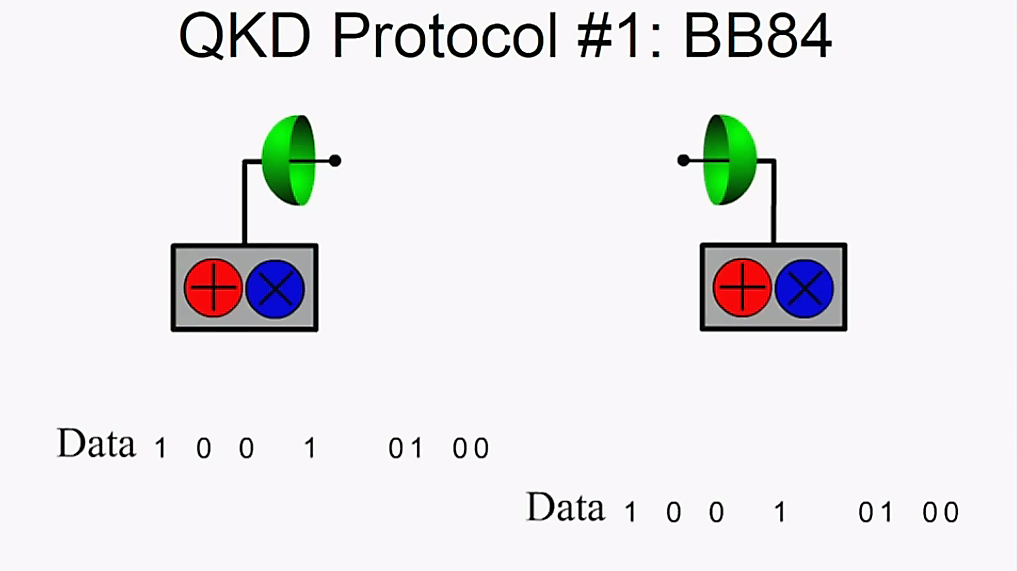
****

****

****

****

****

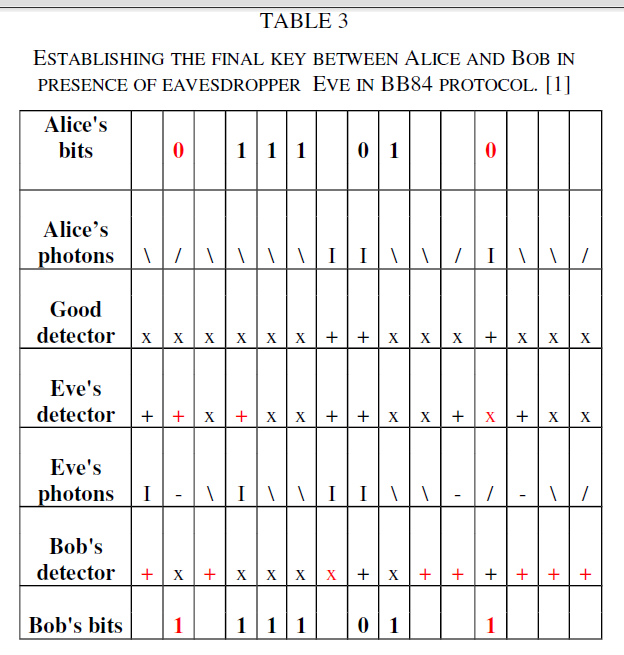
****

**Final Key after Sifting**

**6.4 BB84 WITH EVESDROPPER:**

As it is already known the act of measuring the polarization of a photon may alter the polarization itself. Eavesdropper, proverbial Eve, take notes of polarization of photons but simultaneously changes Polarization of some of them. Therefore the string of photons that Bob receives may be considerably different from the one sent by Alice. Eve is in the exact same position as Bob, which means that she is forced to choose detectors randomly. That, in turn, must result in statistically 50% wrong choice of detector. Still, even having the improper detector chosen, she has got a 50% chance of sending a photon polarized in the way that will yield Bob the bit representation equal to one sent by Alice.

When Eve tries to extract information from the photon, photon’s information (Polarized State) will be flipped according to Heisenberg Uncertainty Principle. After discarding the bits for which randomly chosen Bases are not matched with those of Alice, the sequences of Alice and Bob will be different as shown below



In this example, all wrong detectors chosen by Bob and Eve as well as the final bits that became changed because of the act of eavesdropping are denoted in red. As it can be observed, in this sequence of 15 bits 8 of them for which Bob used a wrong detector have been thrown away at once. From the remaining

7 bits a quantum distribution key should be created. However, it turns out that Alice’s key differs from Bob’s key due to the action of an eavesdropper, who changed 2 of 7 bits.

Alice sends first bit equal to 0 using a diagonal filter, which according to the convention assumed becomes 45 deg oriented photon. Now, Eve sets the randomly chosen detector for that particular photon, which in this case is a rectilinear one. As quantum indeterminacy implies no possible measurement can distinguish four different polarization states that are not all orthogonal. The only possible measurement is between any two orthogonal states- a basis. That means that when Eve measures in the rectilinear basis it will give her a rectilinearly oriented photon. If this photon had been horizontally or vertically polarized before going through the polarizer the measurement would have been absolutely correct. Eve is unfortunate as the photon is 45 deg tilted and thus the rectilinear measurement yields either horizontally or vertically polarized photon with the same probability. Furthermore, all information about the initial polarization of the photon is lost after Eve’s measurement. In the case considered the photon becomes horizontally oriented and as such sent to Bob, which uses the detector oriented in the exact same manner as Alice- diagonally. The horizontally oriented photon passes this detector and must turn into a diagonal orientation, either 45 deg or 135 deg. In this case, Bob receives bit 1, which means that the photon turned into 135 deg orientation. Summing up Eve by the act of eavesdropping changed the final bit on Bob’s side.

For the seventh bit in Alice’s final sequence, the situation is similar. Eve uses an incorrect detector and it results in incorrect bit received by Bob. For the second bit of Alice’s final key, the detector used by Eve is again wrongly oriented but the photon passing Bob’s detector become polarized in the way that results in the correct bit representation being 1. For the bits from third to the sixth one of Alice’s key Eve luckily uses correctly oriented detectors so the polarization of photons will not be altered and Bob will receive correct bits. The final conclusion follows that sender and recipient unable to communicate will have the invaluable information about the potential eavesdropper on the line. Therefore the whole process of key exchange will have to be initialized once again.

1. **WHY IS PYTHON USED FOR BB84?**

As there are no Sufficient Modules and Components in most of Optical Simulation software for Quantum Key Distribution, it is not possible to implement using them. So, we chose Python and Tkinter (For GUI) to implement BB84. And the best thing with python is that its syntax is very closely related to English. Any Commoner can understand. Python also has many Libraries and Modules for Random Bit Generation and Encryption like Crypto.cipher for Encryption.

1. **IMPLEMENTATIONS**

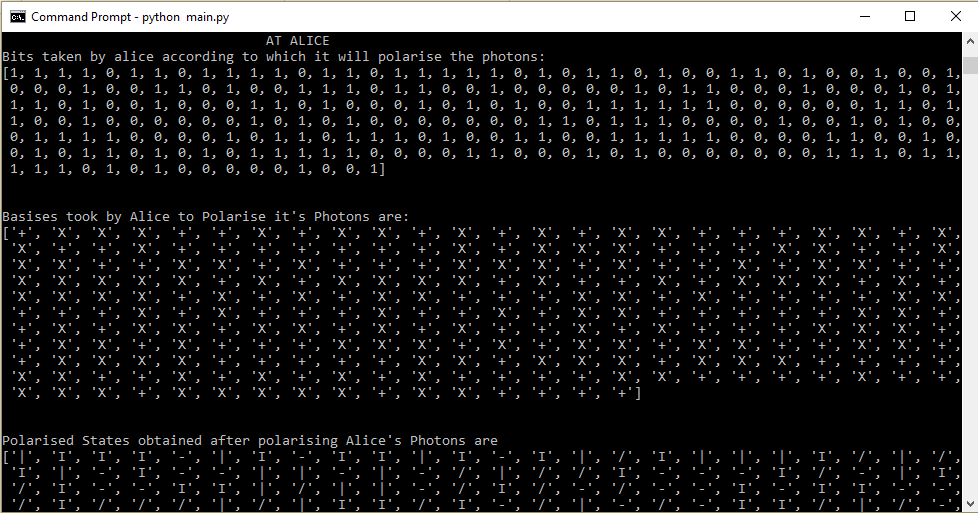
**1. QUANTUM KEY GENERATION USING PYTHON:**

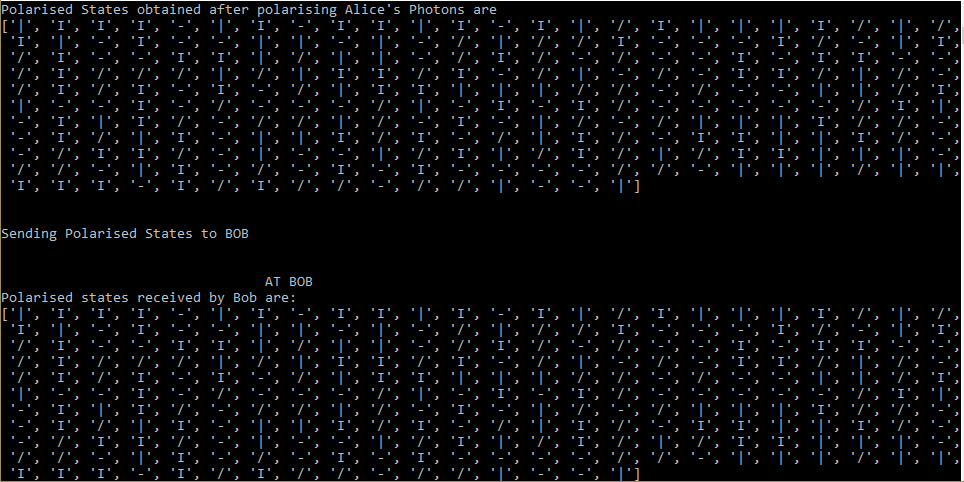
First, Alice will take some bits obtained Randomly using Secrets.choice.([0,1]). Then, Alice polarizes its photons with Bases chosen Randomly from Secrets.choice([+, X]) and jots down which basis has been used for a photon. Now, it sends the Polarised photons to Bob via Quantum Channel.

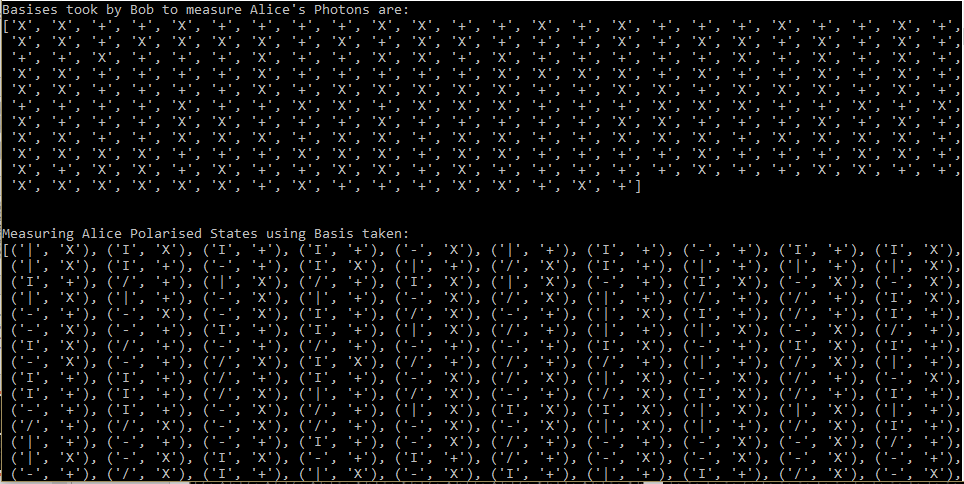
Bob receives the Polarised Photons from Alice and as it doesn’t know about the polarisers Alice used, it measures photons with basis randomly chosen using Secrets.choice([+, X]) and if photon is Rectilinearly oriented and Basis is rectilinear basis then, bob measures photon perfectly or else photon polarized state changes according to Basis used and bob has 0.5 probability of measuring photon correctly.

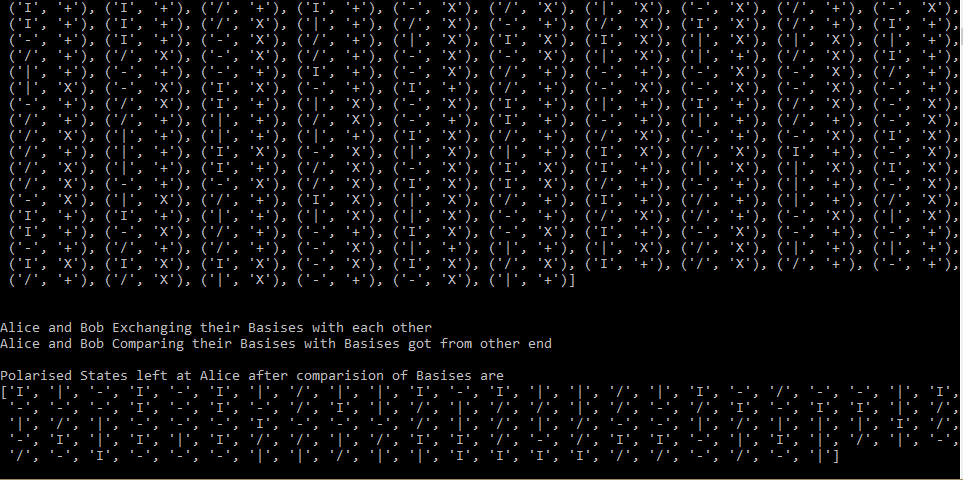
Now, both Alice and Bob exchange their Sequences of Basis used and Compare their own Basis with the incoming basis and Discard the bits for which different Basis are used.

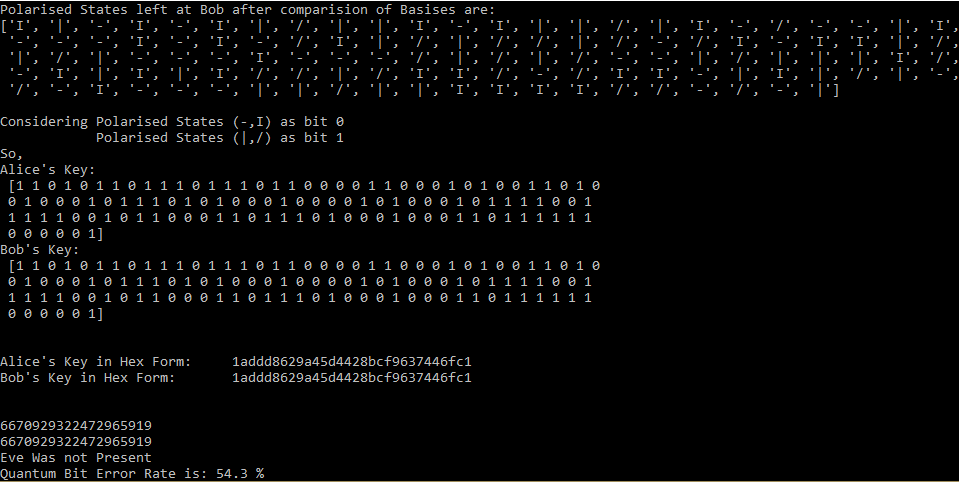
At last, they use Hash Function on some of its bits to verify whether keys obtained are same or not. They exchange the Hash Values and they check their Hash values with incoming hash values and if they are same then, they will be sure that the keys are same and if they aren’t they will find that an eve or noise intercepting their transmission and they will start Key Distribution again.

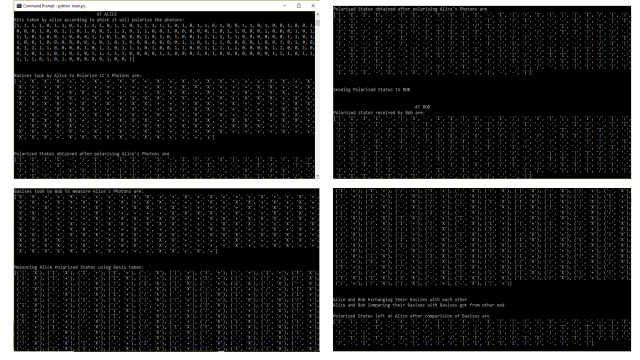












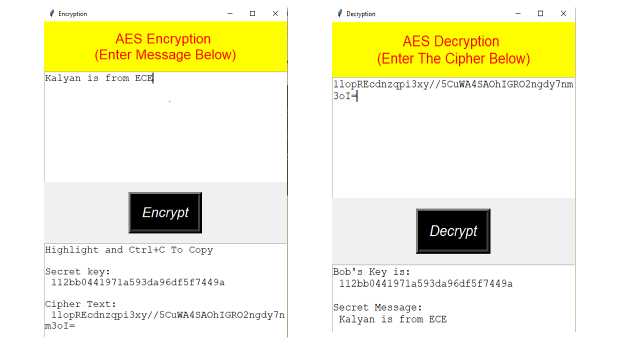
2**. ENCRYPTION OF DATA BY AES USING KEY OBTAINED FRO QKD:**

First, the Key obtained from Quantum Key Distribution is Converted into Hexadecimal Form.

In Python, there is a Library called Crypto.cipher. In this Library, there is a module called AES in which AES will be performed which performs Encryption and decryption as well.

The Hexadecimal key will be given to AES Encryption Function which Generates Cipher for the Message given.

Cipher will be given to the AES Decryption Function which Decrypts Cipher and Gives the Message.

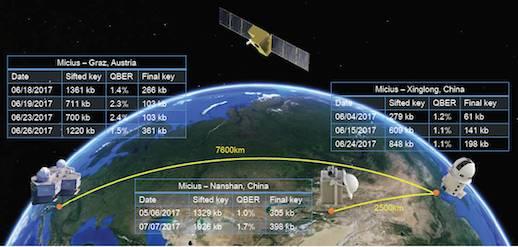


1. **PRACTICAL IMPLEMENTATIONS OF QKD**

BB84 has been experimentally demonstrated to act correctly with a bit rate of 1Mbit/s over 20 km and 10 kbit/s over 100 km of fiber optic cable. On 21st October 2007, the Geneva government implemented for the first time IDQ's hybrid encryption solution, using state of the art Layer 2 encryption combined with Quantum Key Distribution (QKD). The Cerberis solution secures a point-to-point Gigabit Ethernet link used to send ballot information for the federal and cantonal elections from the central ballot counting station to the Geneva government data center. IDQ's Cerberis solution combines leading Layer 2 encryption techniques, based on 256-bit AES cipher (Advanced Encryption Standard), with the extra protection of Quantum Key Distribution.

In 2016, China developed a sophisticated satellite, named after Micius, dedicated for quantum science experiments (for the project timeline and its design details, see Methods), which was successfully launched on 16th August 2016, from Jiuquan, China, orbiting at an altitude of ~500 km (Fig. 1a). Using one of the satellite payloads—a decoy-state QKD transmitter at 850 nm wavelength—and cooperating with Xinglong ground observatory station (near Beijing, 40°23’45.12’’N, 117°34’38.85’’E, altitude 890m), we establish the decoy-state QKD with polarization encoding from the satellite to the ground with ~kHz rate over a distance up to 1200 km.

In June 2017, Chinese researchers [demonstrated](https://www.theregister.co.uk/2017/06/16/secure_quantum_communication_tech/) they could maintain space-to-ground entanglement, and at the same time, also maintained entanglement over a record distance, 1,200 km, between ground stations.



Micius distributing keys.

In the latest research, the researchers added an Austrian node into the mix, achieving kilohertz-rate key distribution between stations that were 7,600 km apart.

China doesn't have the quantum space race all to itself. Last year, boffins from the University of Padua in Italy conducted their own [earth-to-space experiment](https://www.theregister.co.uk/2016/06/08/quantum_encrypted_messages_sent_space/). However, unlike Micius, which carries its own QKD transmitter, the Italian experiment created an entangled state on the ground and bounced it off a satellite, demonstrating that they were able to maintain the photons' quantum state on the round trip.

Japan also has a [Quantum satellite experiment](https://www.theregister.co.uk/2017/07/11/japan_joins_quantum_space_race_with_microsatellite_demo/).

1. **CONCLUSION**

Unlike asymmetric methods of cryptography, Quantum cryptography is heavily dependent on Hardware used. This is the most crucial factor that limits its practical application.

The proper transmission and detection of photons must be satisfied so a precise method of emitting and detecting single photons is indispensable. Photons as very small particles of energy are difficult to be sent separately. By supplying the photon generator with only slightly too much energy several photons might be emitted at once, which is undesirable. Among the techniques proposed for generating single photon states the following are: faint laser pulses, parametric down conversion, single electrons in mesoscopic p-n junctions, photon emission of electron-hole pairs in a semiconductor quantum dot.

Along with precise emission equipment, detection equipment must be precise. A few possible solutions enabling photon detection to exist and those are photomultipliers, avalanche photo-diodes, multichannel plates and superconducting Josephson junctions. Detectors should have a high efficiency over a large spectral range and a short recovery time. Based on those criteria Single Photon avalanche photo diodes are most advantageous. They operate beyond breakdown voltage of the diode, in a state called **Geiger mode**. In this mode, the energy from a single absorbed photon is enough to cause an electron avalanche, which manifests itself in a detectable flood of current. To detect another photon, the diode needs to be reset, which is a time-consuming process and results in detection rate that remains unsatisfactory. Depending on the wavelengths at which detection takes place different semiconductors (silicon, germanium and indium gallium arsenide) may be used. Unfortunately, silicon has too large band gap so its sensitivity is not sufficient. Best detection wavelength of silicon is 800 nm, whereas at 1100 nm it becomes insensitive, which is still less than standards for telecommunications applications (1300 and 1550 nm). Therefore Germanium or Indium-Gallium-Arsenide detectors must be used at telecommunications wavelengths, even though they are far less efficient and must be cooled considerably below room temperature.

Among other factors influencing the wider use of quantum cryptography is the distance of transmission, and dedicated network of fiber lines can be listed.

As a medium of transmission, fiber optic cables are used most often. Unfortunately, their distance of transmission is limited whereas amplifiers cannot be used to send data on the longer distances as they may change the polarization of photons and facilitate the process of eavesdropping. Next trouble encountered concerning fiber lines is their integration with existing optical networks. The cost of building additional optical infrastructure still remains relatively too high to use quantum cryptography more widely.

Furthermore, the maintenance of fiber lines is also expensive and if they are not properly protected then a cutting or blocking some part of the network may lead to denial of service, which is unacceptable.

To avoid a use of fiber network an alternative technology in which Quantum keys are exchanged in this method by means of free space with the aid of satellites. Such transmission is very fluctuating and has got high impedance in comparison with less noisy optical fiber transmission. The communication takes place between a terrestrial station and a low orbit satellite. The absorption of photons in the atmosphere can be minimized using an adequate wavelength. The atmosphere has a high transmission window at a wavelength of about 770 nm, where photons can be easily detected using efficient photon counting modules. At these wavelengths, the atmosphere would not change the polarization of photons, which is a great advantage. The type of weather obviously influences the transmission as well. Phase shifts and polarization dependent losses would also have to be taken care of. A satellite obtains the key from the station on the ground, moves with respect to the earth surface and detecting a receiving station sends the key to it.

**9.** **REFERENCES**

1. BB84 analysis of the operation and practical considerations and implementations of quantum key distribution systems by Patryk Winiarczyk, Wojciech Zabierowski.
2. [BB84] Bennett, C. H. and Brassard, G., "Quantum Cryptography: Public key distribution and coin tossing.", International Conference on Computers, Systems & Signal Processing, Bangalore, India, 10-12 December 1984,
3. [BB84] Quantum Cryptography Lecture by Peter McMahon, Postdoctoral Researcher, Department of Applied Physics, Stanford University.
4. Unconditionally secure cryptosystems based on quantum cryptography by Yu-Fang Chung, Zhen Yu Wub, Tzer Shyong Chen, Taiwan.
5. A Survey of the Prominent Quantum Key Distribution Protocols by Mart Haitjema, [mart.haitjema@wustl.edu](mailto:mart.haitjema@wustl.edu)
6. Quantum Cryptography Richard J. Hughes D. M. Alde, P. Dyer, G. G. Luther, G. L. Morgan and M. Schauer University of California Physics Division Los Alamos National Laboratory Los Alamos, NM 87545
7. C. H. Bennett et al., J. Crypt. 5, 3 (1992).
8. C. H. Bennett, Phys. Rev. Lett. 68, 3121 (1992); A. K. Ekert, Nature 358, 14 (1992).
9. G. Brassard, “Lecture Notes in Computer Science 325: Modern Cryptology,” (Springer, New York, 1988)
10. C.H. Bennett, Quantum cryptography using any two non-orthogonal states, Physical Review Letters 68 (21) (1992) 3121–3124.
11. C. Marand, P.D. Townsend, Quantum key distribution over distances as long as 30 km, Optics Letters 20 (August) (1995) 1695–1697.
12. http://en.wikipedia.org/wiki/BB84
13. http://en.wikipedia.org/wiki/Quantum\_cryptography
14. <http://arxiv.org/ftp/quant-ph/papers/9905/9905009.pdf>
15. https://www.idquantique.com/wordpress/wp-content/uploads/user-case-gva-gov.pdf
16. <https://www.theregister.co.uk/2018/01/22/china_flaunts_its_qkdinspaaace_by_securing_videoconference/>
17. http://www.xinhuanet.com/english/2018-01/22/c\_136915204.htm
18. AES Proposal: Rijndael by Joan Daeme(Proton World Int.l ) and Vincent Rijmen(Katholieke Universiteit Leuven, ESAT-COSIC), Belgium.