PERFORMANCE OF QUANTUM BASED IMAGE SCRAMBLING CRYPTOGRAPHY USING QPQ-CD ON 5G NETWORK AUTHENTIFICATION KEY AGREEMENT

RAKOTONDRAMANANANA Radiarisainana Sitraka[1],
RANDRIAMITANTSOA Paul Auguste[2],

[1] *PhD student, TASI, ED-STIII, Antananarivo, Madagascar*
[2] *Thesis director, TASI, ED-STII, Antananarivo, Madagascar*

ABSTRACT

The QPQ-CD (Quantum and PostQuantum CipherKey Dynamic uses a dynamic key after the registration of the mobile at the 5G Network. The algorithm could be divided in two parts : The Quantum Cryptography and Post Quantum Cryptography. The Quantum Cryptography used Quantum Image Scrambling methods with the Red, Green, and Blue Component which are treated separately by the two algorithms QAT and QHT. They change only the position of the image pixel without changing the value of this. QAT and QHT use 2 method different : one is based on the permutation of the pixel position and the others is based on one the space-filling curveIn this article, the performance of the Quantum Cryptography using at the QPQ-CD is analyzed by the PSNR(Peak Signal to Noise Ratio), SSIM (Structural SIMilarity), NPCR(Number of Pixel exchange rate), UACI (Unified Average Changing Intensity), Correlation and the binary entropy. PSNR for the two methods is near than the value 8 which is a good visual quality visualThe SSIM and Correlation also offers a best parameter which is near zero. The image transformed by scrambling methods and originals are no similitude and no correlation. The NPCR give also a good performance because 99% of pixels changed in the image. But the UACI is not more than 20%. So, the intensity light of the new image doesn't change with the maximum. The algorithm of the scrambling is not resistant to the attack physic using difference power analysis. The binary entropy is less than 50%. There are quite no disorder between the bits one and bit zeros. The QPQ-CD with evaluation of the Quantum Cryptography is simulated to Matlab and all curve for this is represented in this article. The QAT and QHT have quite a similar performance.

Keyword *QHT, QAT, PSNR, SSIM, NPCR*

1. Introduction

In the network 5G, the UE and operator use mutual authentication based on the master key K. A static key K is so a vulnerability for the user. The algorithm QPQ-CD uses a dynamic key with optimized selectors with high probability of extremity, probability of proximity, probability of a bit changed and probability in case of entropy.

2.1 Implementation of simplified QPQ-CD

QPQ_CD uses the master key Dynamicity K. This function has as input the previous key K and an activation signal A and a parameter r defining the complexity rule of QPQ-CD.
The steps of the algorithm are:

- The initialization phase: the goal is to initialize K, r, and i an activation counter and to generate from the Expansion towards the Matrix (E.M) a matrix of 16r × 16r of 8 bits
- The insertion phase: It consists of periodically inserting while scanning the line of the matrix 16r × 16r a key obtain through the Expansion towards the Linearity (E.L). The insertion is executed only at each

activation signal. Since the QPQ-CD algorithm uses 3 16r × 16r matrices, a key-generating function denoted by G makes it possible to generate 3 parts of key of initializations for each matrix.

- The phase of quantum cryptography: The phase of quantum cryptography uses the method of confusion either by the Hilbert method or by Arnold's method.
- The PQ Cryptography phase: it uses several hash algorithm samples to summarize the matrix after confusion in order to have a 256-bit key.
- Phase selectors : it selects the next key K+ appropriate.

The output of the QPQP-CD algorithm is another key generated K+, for other applications especially in the authentication. QPQ-CD is also a family of KDF algorithm. The simplified schema of the QPQ-CD algorithm is represented in Figure 1.

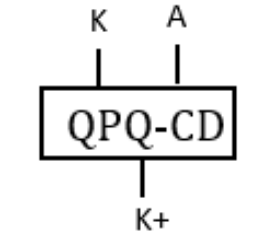

Fig -1 QPQ-CD

2.2 Evaluation of QPQ-CD

For the performance study of the QPQ-CD algorithm, the activation counter will be traversed until the end of the insertion line. Thus, i vary from 1 ... 16r.

The QPQ-CD algorithm will be characterized by the initialization phase that generates r, i, K then JR, JG and JB. The insertion of the keys generated by the Expansion towards the Linearity (E.L) and the key generator G will be repeated at each blur up to 16r. The QC algorithm followed by PQC will be finalized by the optimization selector to obtain the key next K +.

The selection algorithm uses several criteria to identify the best key using the probability of not detecting the key from the previous key by focusing on how opponents think and other relevant criteria. Since the insertion of the matrix is done at each line from 1 to 16r, the authentication sample will be limited to this value 16r.

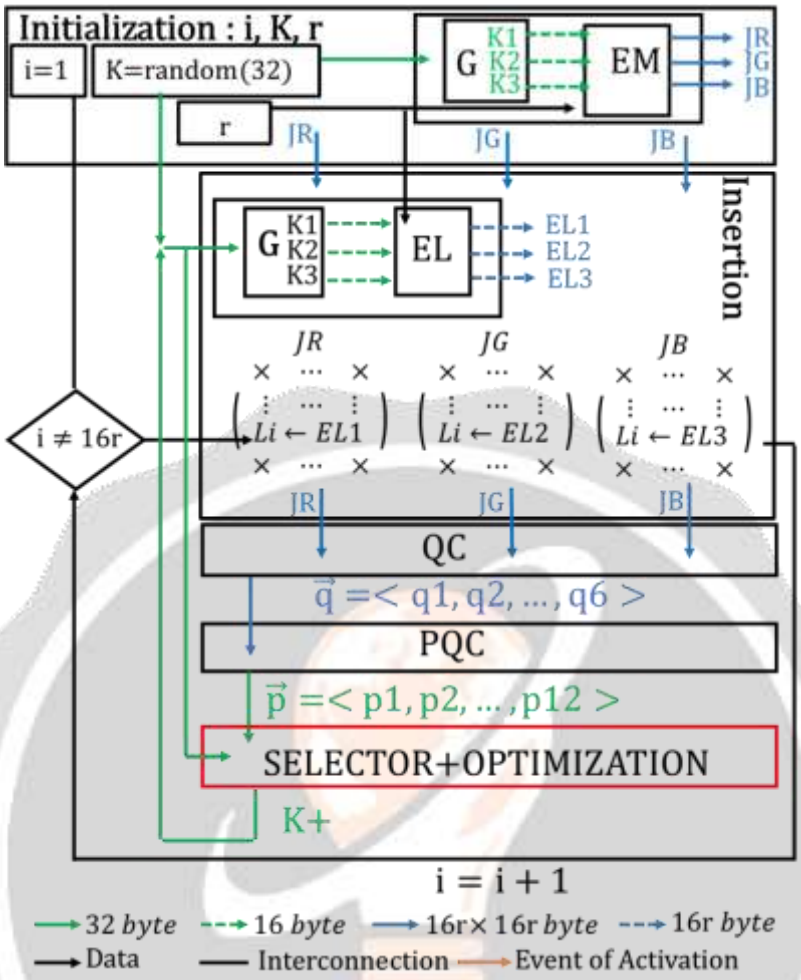

Fig -2 Evaluation of QPQ-CD

2.3 Quantum Circuit Gate

The different quantum circuit gates for quantum computing are presented like in Figure 1. Each quantum gate has its own transfer function [1- 6]

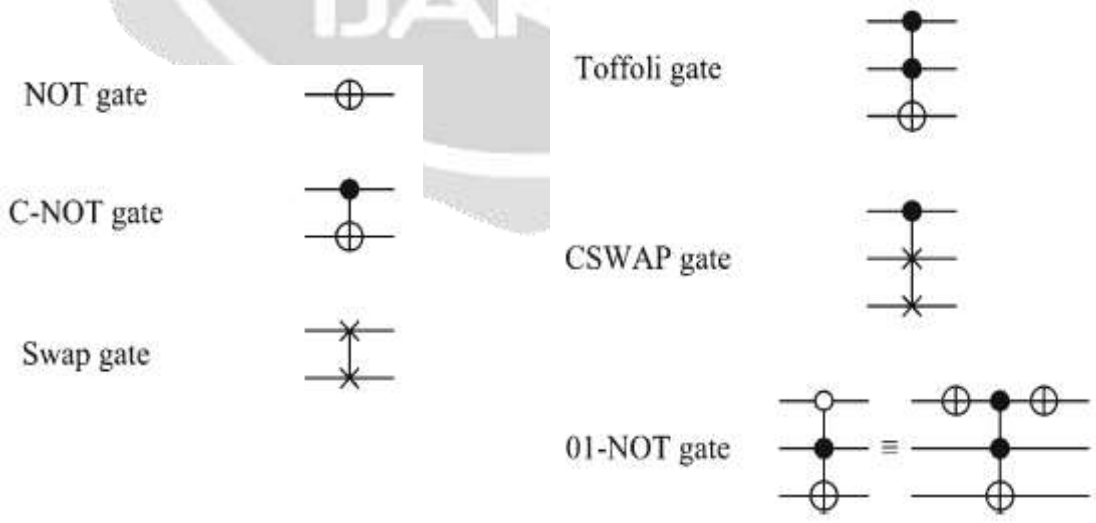

Fig-3 Quantum Circuit

2.4 FRQI representation

All matrixes could be represented like FRQI [7-12] defined by the formula (1) and (2)

$$|I(\theta)\rangle = \frac{1}{2^n}\sum_{i=0}^{2^{2n}-1}|C_i\rangle\otimes|i\rangle \quad (1)$$

$$|C_i\rangle = \cos(\theta_i) + j\sin(\theta_i), \theta_i = \left[0;\frac{2}{\pi}\right], i = 0,1,\ldots,2^{2n}-1 \quad (2)$$

$|C_i\rangle$ The corposant of the matrix

$|i\rangle$ The Linearized index of the matrix

$|I(\theta)\rangle$ The Representation of the matrix using FRQI

2.5 Hilbert Transform

The Hilbert transform H_n is a permutation recursive represented like on Figure 4.

n = 2 n =3 n = 4 n = 5

Fig-4 Graphics representation of Hilbert Transform

2.6 Quantum Hilbert Transform

The Quantum Hilbert Transform could be implemented using matrix conversion and an algorithm either even or odd [7-12]

2.6.1 Transformation of Matrix

To realize the Quantum Hilbert Transform, all the matrix transform are needed: transpose, up-down, left- and central rotation expressed respectively by $A^T, A^{ud}, A^{lr}, A^{pp}$:

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \ldots & a_{1,m} \\ a_{2,1} & a_{2,2} & \ldots & a_{2,m} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m,1} & a_{m,2} & \ldots & a_{m,m} \end{bmatrix}$$

So,

$$A^T = \begin{bmatrix} a_{1,1} & a_{2,1} & \ldots & a_{m,1} \\ a_{1,2} & a_{2,2} & \ldots & a_{m,2} \\ \vdots & \vdots & \vdots & \vdots \\ a_{1,m} & a_{2,m} & \ldots & a_{m,m} \end{bmatrix}; \; A^{lr} = \begin{bmatrix} a_{1,m} & \ldots & a_{1,2} & a_{1,1} \\ a_{2,m} & \ldots & a_{2,2} & a_{2,1} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m,m} & \ldots & a_{m,2} & a_{m,1} \end{bmatrix};$$

$$A^{ud} = \begin{bmatrix} a_{m,1} & a_{m,2} & \dots & a_{m,m} \\ \vdots & \vdots & \vdots & \vdots \\ a_{2,1} & a_{2,2} & \dots & a_{2,m} \\ a_{1,1} & a_{1,2} & \dots & a_{1,m} \end{bmatrix} ; \; A^{pp} = \begin{bmatrix} a_{m,m} & \dots & a_{m,2} & a_{m,1} \\ \vdots & \vdots & \vdots & \vdots \\ a_{2,m} & \dots & a_{2,2} & a_{2,1} \\ a_{1,m} & \dots & a_{1,2} & a_{1,1} \end{bmatrix}$$

The transform recursive path of Hilbert is defined by the Formula:

$$H_{n+1} = \begin{cases} \begin{pmatrix} H_n & 4^n E_n + H_n^T \\ (4^{n+1}+1)E_n - H_n^{ud} & (3.4^n+1)E_n - (H_n^{lr})^T \end{pmatrix} \; if\ n\ even \\ \begin{pmatrix} H_n & (4^{n+1}+1)E_n - H_n^{lr} \\ 4^n E_n + H_n^T & (3.4^n+1)E_n - (H_n^T)^{lr} \end{pmatrix} \; if\ n\ odd \end{cases} \quad (3)$$

With n is a positive integer and $H_1 = \begin{pmatrix} 1 & 2 \\ 4 & 3 \end{pmatrix}$ and $E_n = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \dots & 1 \end{pmatrix}$

For the realization of the Quantum Hilbert Transform, all these properties are considered:

If A, B, C, D, four matrixes $m \times m$, so,

1. $(A+B)^{pp} = A^{pp} + B^{pp}$ (4)
2. $(A+B)^{lr} = A^{lr} + B^{lr}$ (5)
3. $(A+B)^{ud} = A^{ud} + B^{ud}$ (6)
4. $\begin{pmatrix} A & B \\ C & D \end{pmatrix}^{lr} = \begin{pmatrix} B^{lr} & A^{lr} \\ D^{lr} & C^{lr} \end{pmatrix}$ (7)
5. $\begin{pmatrix} A & B \\ C & D \end{pmatrix}^{ud} = \begin{pmatrix} C^{ud} & D^{ud} \\ A^{ud} & B^{ud} \end{pmatrix}$ (8)
6. $\begin{pmatrix} A & B \\ C & D \end{pmatrix}^{pp} = \begin{pmatrix} D^{pp} & C^{pp} \\ B^{pp} & A^{pp} \end{pmatrix}$ (9)
7. $(A^T)^{ud} = (A^{lr})^T$ (10)
8. $A^{ud} = ((A^T)^{lr})^T$ (11)
9. $(A^{ud})^{pp} = (A^{pp})^{ud} = A^{lr}$ (12)
10. $(A^{lr})^{pp} = (A^{pp})^{lr} = A^{ud}$ (13)
11. Using the recursive form, we could express the Hilbert Transform like this:

$$H_{n+1} = \begin{cases} \begin{pmatrix} H_n & (H_n + 4^n E_n)^T \\ (H_n + 3.4^n E_n)^{pp} & (H_n + 2.4^n)^T \end{pmatrix} \; if\ \ n\ even \\ \begin{pmatrix} H_n & (H_n + 3.4^n E_n)^{pp} \\ (H_n + 4^n E_n)^T & (H_n + 2.4^n E_n)^T \end{pmatrix} \; if\ n\ odd \end{cases}$$

If n is a positive integer, the initial matrix is $H_1 = \begin{pmatrix} 1 & 2 \\ 4 & 3 \end{pmatrix}$ and $E_n = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \dots & 1 \end{pmatrix}$

12. In the case of, n even, the 4 expressions should be similar :

(i) $H_n = H_n$

(ii) $4^n E_n + H_n^T = (H_n + 4^n E_n)^T$

(iii) $(4^{n+1}+1)E_n - H_n^{ud} = (H_n + 4^n E_n)^T$

(iv) $(3.4^n + 1)E_n - (H_n^T)^{lr} = (H_n + 2.4^n E_n)^T$

The expression (i) is an evident identity, so no demonstration is needed,

For the expression (ii),

$$(H_n + 4^n E_n)^T = H_n^T + (4^n E_n)^T = 4^n E_n + H_n^T$$

For the expression (iii),

$H_n + H_n^{lr} = (4^n + 1)E_n$

$\left(H_n + H_n^{lr}\right)^{pp} = H_n^{pp} + H_n^{ud} = [(4^n + 1)E_n]^{pp} = (4^n + 1)E_n$

$(4^n + 1)E_n - H_n^{ud} = H_n^{pp}$

$(4^n + 1)E_n - H_n^{ud} + 3.4^n E_n = H_n^{pp} + 3.4^n E_n$

$(4^{n+1} + 1)E_n - H_n^{ud} = H_n^{pp} + 3.4^n E_n$

$(4^{n+1} + 1)E_n - H_n^{ud} = H_n^{pp} + (3.4^n E_n)^{pp}$

$(4^{n+1} + 1)E_n - H_n^{ud} = (H_n + 3.4^n E_n)^{pp}$

For the expression (iv),

$H_n + H_n^{lr} = (4^n + 1)E_n$

$\left(H_n + H_n^{lr}\right)^T = H_n^T + \left(H_n^{lr}\right)^T = [(4^n + 1)E_n]^T = (4^n + 1)E_n$

$(4^n + 1)E_n - \left(H_n^{lr}\right)^T = H_n^T$

$(4^n + 1)E_n - \left(H_n^{lr}\right)^T + 2.4^n E_n = H_n^T + 2.4^n E_n$

$(3.4^n + 1)E_n - \left(H_n^{lr}\right)^T = H_n^T + 2.4^n E_n = H_n^T + (2.4^n E_n)^T$

$(3.4^n + 1)E_n - \left(H_n^{lr}\right)^T = (H_n + 2.4^n E_n)^T$

If n even,

$$H_{n+1} = \begin{pmatrix} H_n & (H_n + 4^n E_n)^T \\ (H_n + 3.4^n E_n)^{pp} & (H_n + 2.4^n)^T \end{pmatrix}$$

In the case, n odd, all 4 expressions should be similar:

(i) $H_n = H_n$

(ii) $(4^{n+1} + 1)E_n - H_n^{lr} = (H_n + 3.4^n E_n)^{pp}$

(iii) $4^n E_n + H_n^T = (H_n + 4^n E_n)^T$

(iv) $(3.4^n + 1)E_n - \left(H_n^{lr}\right)^T = (H_n + 2.4^n)^T$

For the expression (i), $H_n = H_n$, it's a general truth, no need of demonstration

For the expression (ii),

$H_n + H_n^{ud} = (4^n + 1)E_n$

$\left(H_n + H_n^{ud}\right)^{pp} = H_n^{pp} + H_n^{lr} = \left((4^n + 1)E_n\right)^{pp} = (4^n + 1)E_n$

$(4^n + 1)E_n - H_n^{lr} = H_n^{pp}$

$(4^n + 1)E_n - H_n^{lr} = H_n^{pp}$

$(4^n + 1)E_n - H_n^{lr} + 3.4^n E_n = H_n^{pp} + 3.4^n E_n$

$(4^n + 1)E_n - H_n^{lr} + 3.4^n E_n = H_n^{pp} + (3.4^n E_n)^{pp}$

$(4^{n+1} + 1)E_n - H_n^{lr} = H_n^{pp} + (3.4^n E_n)^{pp}$

$(4^{n+1} + 1)E_n - H_n^{lr} = (H_n + 3.4^n E_n)^{pp}$

For the expression (iii), $(H_n + 4^n E_n)^T = H_n^T + (4^n E_n)^T = H_n^T + 4^n E_n = 4^n E_n + H_n^T$

For the expression (iv), $H_n + H_n^{ud} = (4^n + 1)E_n$

$\left(H_n + H_n^{ud}\right)^T = H_n^T + \left(H_n^{ud}\right)^T = H_n^T + (H_n^T)^{lr} = [(4^n + 1)E_n]^T = (4^n + 1)E_n$

$(4^n + 1)E_n - (H_n^T)^{lr} = H_n^T$

$(4^n + 1)E_n - (H_n^T)^{lr} + 2.4^n E_n = H_n^T + 2.4^n E_n$

$(4^n + 1)E_n - (H_n^T)^{lr} + 2.4^n E_n = H_n^T + (2.4^n E_n)^T$

$(3.4^n + 1)E_n - (H_n^T)^{lr} = H_n^T + (2.4^n E_n)^T$

$(3.4^n + 1)E_n - (H_n^T)^{lr} = (H_n + 2.4^n E_n)^T$

$$H_{n+1} = \begin{pmatrix} H_n & (H_n + 3.4^n E_n)^{pp} \\ (H_n + 4^n E_n)^T & (H_n + 2.4^n E_n)^T \end{pmatrix}$$ When n odd,

2.6.2 Quantum Circuit of Hilbert Transform

Based on the recursive theorem, the Quantum Hilbert Transform is dived into 3 parts: initialization, the odd part and the even part. Two methods could be proposed:

1. The result of the recursive path of Hilbert on H_n needs to compute the matrix H_n, and directly swaps the pixel on $(\lfloor H_n(i,j) - 1 \rfloor / 2^n, H_n(i,j) - 1 \bmod 2^n)$ with (i,j)

2. Using the partition of H_n, consists to make partition of each size 2*2, 4*4, 8*8, 16*16 … The size of the output of the image is always $2^n * 2^n$. This method uses the partition of each initialization steps: the odd part and the even part.

The method partition(k) consists to subdivide $2^n * 2^n$ size in input of $2^{n-k-1} * 2^{n-k-1}$ block sized $2^{k+1} * 2^{k+1}$.

- For the first step k =0, matrix is partitioned to $2^{n-1} * 2^{n-1}$ block sized $2 * 2$.
- For the second step k=1, matrix is partitioned to $2^{n-2} * 2^{n-2}$ block sized $4 * 4$.
- For the third step k=2, matrix is partitioned to $2^{n-2} * 2^{n-2}$ block sized $8 * 8$.…
- For the n-th step k=n-1, matrix is partitioned to 1 block sized $2^n * 2^n$.

2.6.3. Partition(k)

Partition(k) is dived into two steps :

(1) Swap x_{k+1} and x_{k+2} ; x_{k+2} andx_{k+3},…. x_{n-2} and x_{n-1}

(2) Swap x_{n-1} and y_k

The quantum circuit swap gate is needed to swap 2 quantum state for having the quantum circuit on Figure 5.

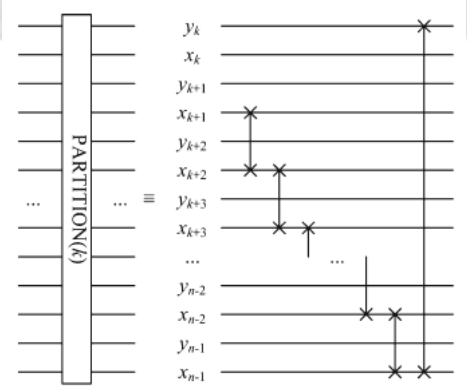

Fig-5 Partition (k)

For each dimension $|y\rangle \; and \; |x\rangle$, the matrix $|i\rangle$ could be expressed by: $|i\rangle = |y\rangle|x\rangle$

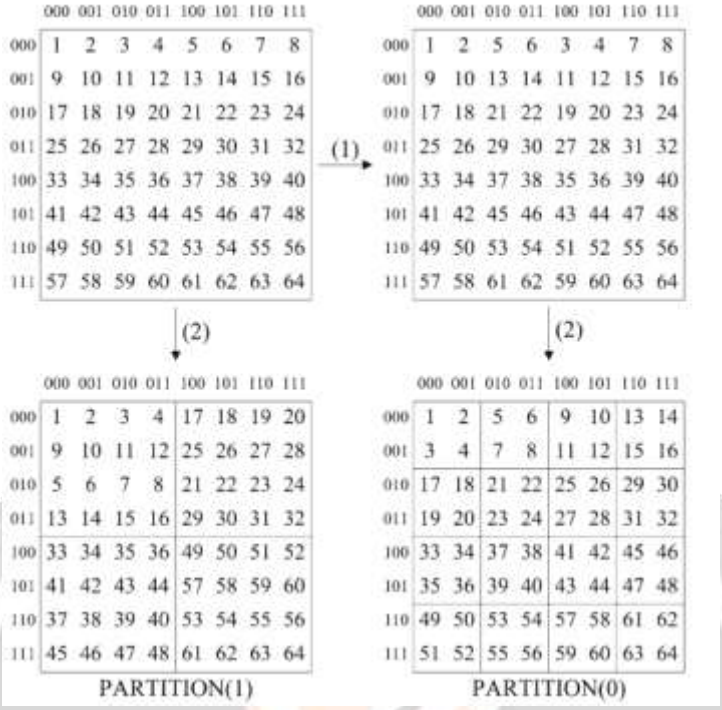

Fig-6 Figure explained partition (1) and partition (0)

2.6.4 Le module O (k)

Having 4 matrixes A, B, C, D sized $2^{k-1} * 2^{k-1}$; The function O(k) Transforms the matrix $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$ to $\begin{pmatrix} A & D^{pp} \\ B^T & C^T \end{pmatrix}$ in the same way as the recursive form of Hilbert Path H_n when n is odd. So, O(k) is used when k is odd and could be obtained by dividing into three steps :

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \xrightarrow{step(1)} \begin{pmatrix} A & D \\ B & C \end{pmatrix} \xrightarrow{step(2)} \begin{pmatrix} A & D \\ B^T & C^T \end{pmatrix} \xrightarrow{step(3)} \begin{pmatrix} A & D^{pp} \\ B^T & C^T \end{pmatrix}$$

Step 1: Using C-NOT gate and CSWAP gate
Step 2: Using CSWAP gate
Step 3: Using 01-NOT and 0-Control gate

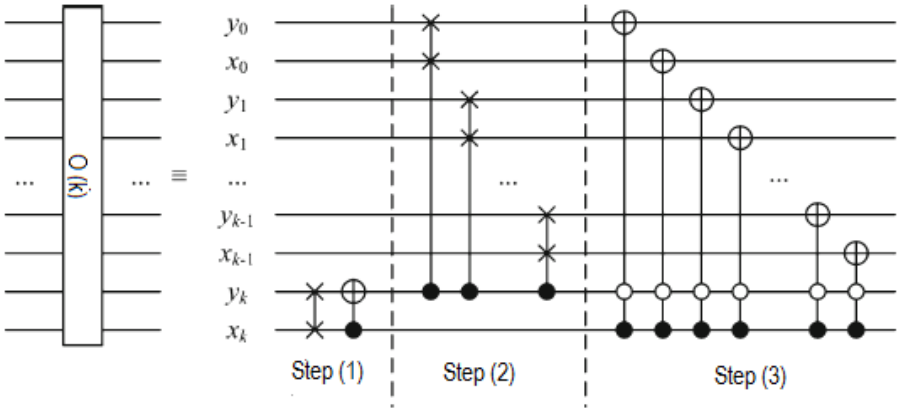

Fig 7- Odd Quantum Circuit module O(k)

2.6.5. The module E(k)

Having 4 matrixes A, B, C, D sized $2^{k-1} * 2^{k-1}$; The function E(k) Transforms the matrix $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$ to $\begin{pmatrix} A & B^T \\ D^{pp} & C^T \end{pmatrix}$ in the same way as the recursif form of Hilbert Path H_n when n is even. so, E(k) is used when k is even and could be obtained by dividing into three steps :

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \xrightarrow{Step(1)} \begin{pmatrix} A & B \\ D & C \end{pmatrix} \xrightarrow{Step(2)} \begin{pmatrix} A & B^T \\ D & C^T \end{pmatrix} \xrightarrow{Step(3)} \begin{pmatrix} A & B^T \\ D^{pp} & C^T \end{pmatrix}$$

Step 1 : Using C-NOT gate
Step 2 : Using CSWAP gate
Step 3 : Using 01-NOT and 0-Control gate

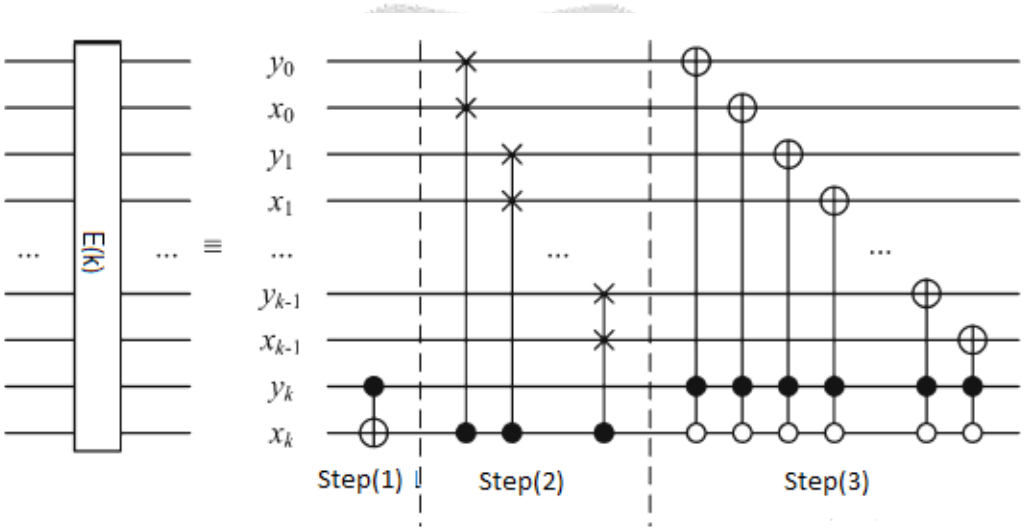

Fig 8- Even Quantum Circuit module Module E(k)

2.6.6. Module Initialization

The module initialization is by definition the partition (0). The Classic Hilbert Transform, uses the matrix H_1 initialization. The partition(0) will divide the matrix input $2^n * 2^n$ to $2^{n-1} * 2^{n-1}$ size 2*2 which has the same size of H_1.

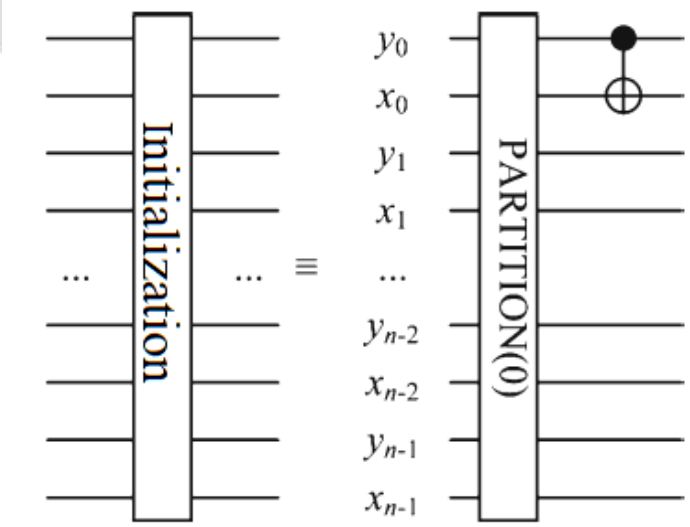

Fig 9- Quantum circuit for initialization

2.6.7. Module part even and Part Odd

For having all different pixels localization, the module partition is before the E(k) giving the module part even and the module partition is before the O(k) giving the module part odd.

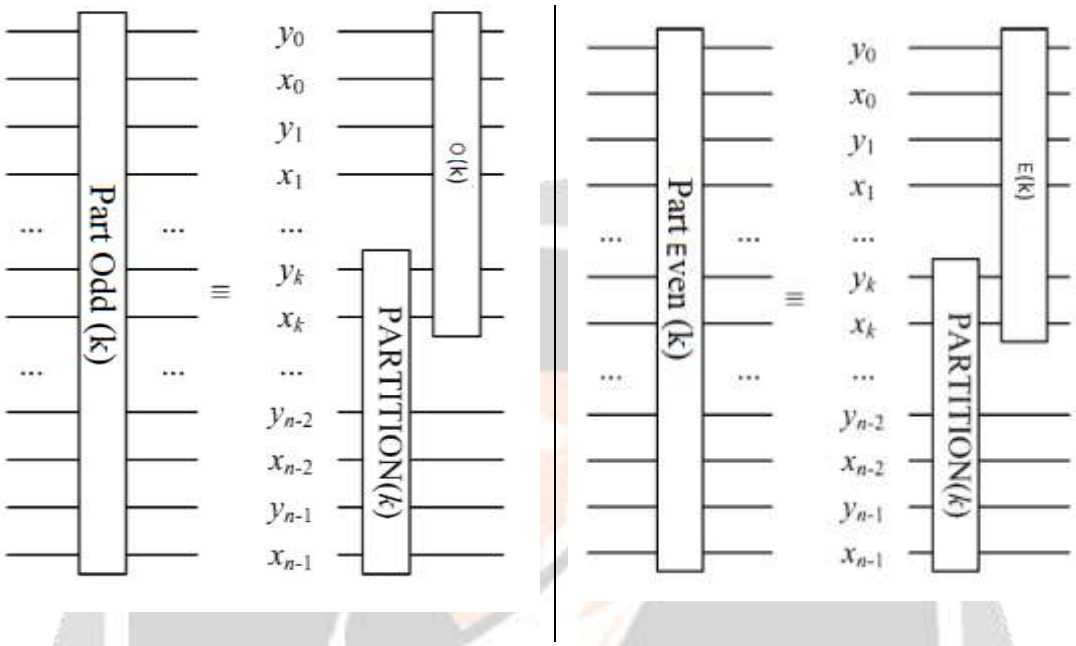

Fig 10- Quantum Circuit for part even and part odd

2.6.8. Quantum circuit of QHT and inverse circuit

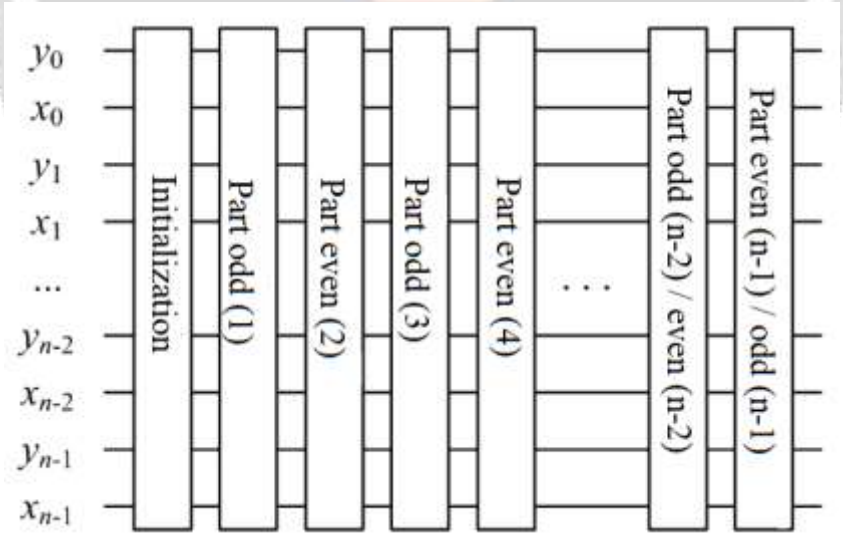

Fig 11- Quantum QHT

To realize the quantum circuit, the module of initialization part-odd, part-even will be alternate. To recover the matrix of the output, the order is reversed.

For n = 3, we could have :

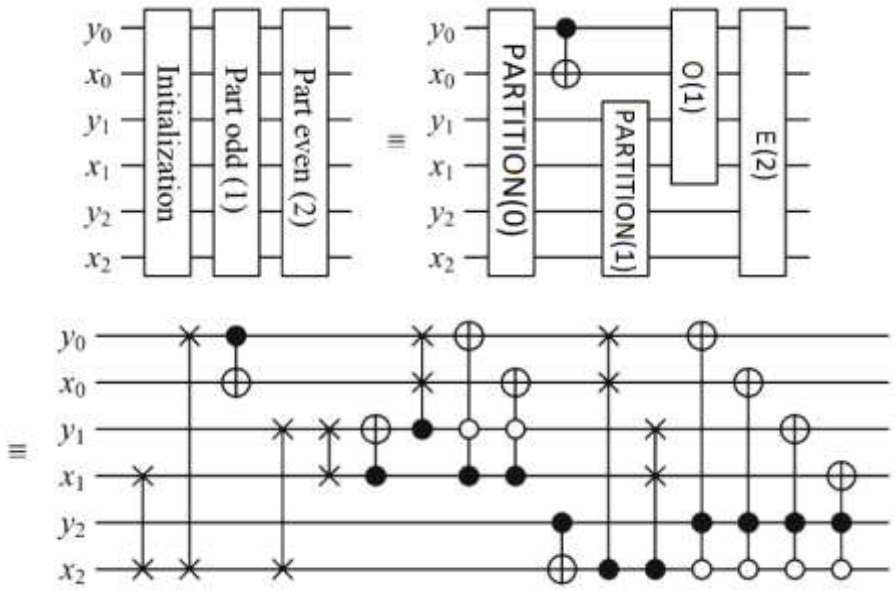

Fig 12- Quantum Hilbert Transform for n = 3

2.7. QAT

The general form of Arnold Transform is represented like :

$$\begin{cases} x' = (x + ty) mod N \\ y' = (mx + (tm + 1)y) mod N \end{cases}$$

Using quantum images, the representation of the same equation represented like :

$$\begin{cases} |x'\rangle = |(x + ty) mod 2^n\rangle \\ |y'\rangle = |(mx + (tm + 1)y) mod 2^n\rangle \end{cases} \quad (14)$$

$|x'\rangle, |y'\rangle$ are calculated on the quantum circuit using the general transform of Arnold

The realization of the quantum circuit of Arnold is like to study an adder and adder modulo N, multiplication modulo N, exponential modulo N with a quantum computing. The adder permit of the sum of two quantic input and give a quantum number result.

Définition

Using a quantum computer, the classic register will be replaced by a quantum register. The adders of the two quantum bit could be expressed by:

$$|a, b\rangle \rightarrow |a, a + b\rangle \quad (15)$$

a, b are the quantum register and the output function permit to calculate the sum of the two entries. The addition modulo N of two Quantum bits could by defined by:

$$|a, b\rangle \rightarrow |a, (a + b) mod N\rangle \quad (16)$$

Figure 13 represents the schema of the complete adder quantic and adder quantic modulo.

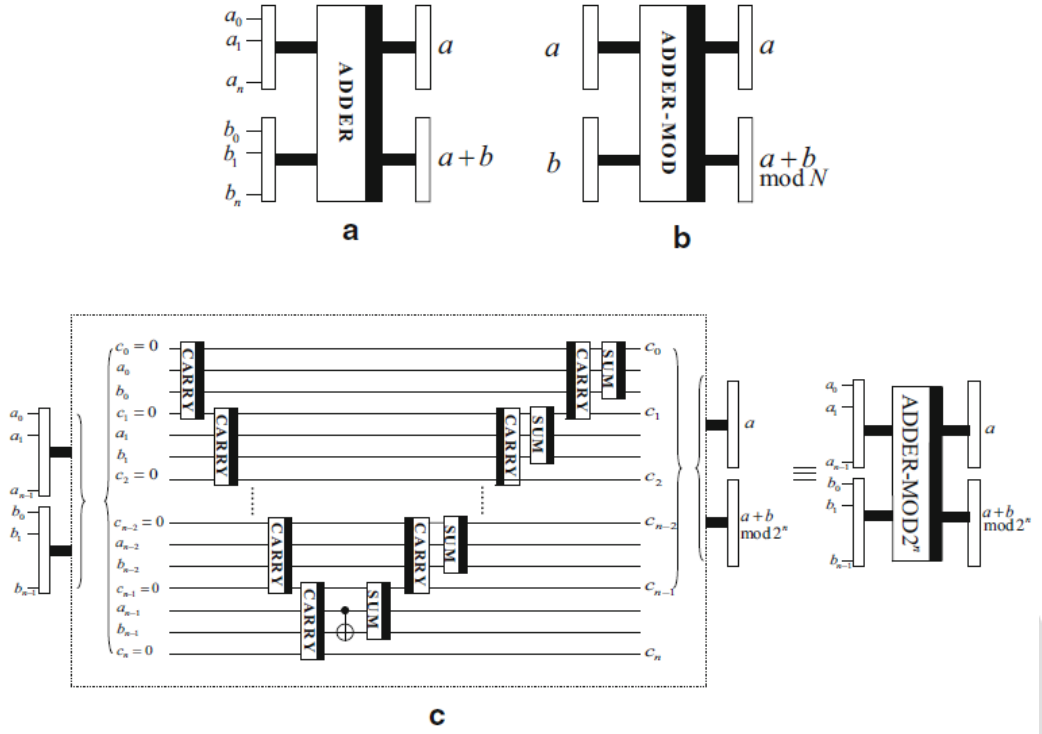

Fig 13- the schema of the complete adder quantic and adder quantic modulo

Like on the adder classic, the quantum adder use also two quantum modules of « carry » and « sum » and the quantum circuit is presented in the figure 14.

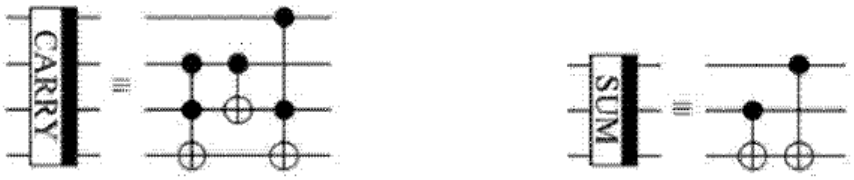

Fig 14- Quantum circuit of the adder

2.7.1 The Abscissa Circuit of $|x'\rangle$

For the cas of the Quantum Arnold Transform, the state quantic $|x'\rangle$ and $|y'\rangle$ are independent between them.

By using the quantum adder modulo, it's possible to realize the full schema of Quantum Arnold Transform.

Let x,y have n-quantum bit $x = x_{n-1}x_{n-2}\dots x_0$ and $y = y_{n-1}y_{n-2}\dots y_0$ with $x_i, y_i \in \{0,1\}$;

$i = n-1, n-2, \dots, 0$

$$x' = y, (ty + x) mod 2^n$$

So,

$$|y,x\rangle \rightarrow |y,(y+x)mod2^n\rangle \rightarrow |y,(2y+x)mod2^n\rangle \rightarrow \cdots \rightarrow |y,\big((t-1).y+x\big)mod2^n\rangle \rightarrow |y,(ty+x)mod2^n\rangle \rightarrow |x'\rangle$$

(17)

Demonstration

Using the propriety of the modulo,

$$(x+y)mod2^n = (xmod2^n + ymod2^n)mod2^n = (x+ymod2^n)mod2^n$$

So $(x+2y)mod2^n = (x+2ymod2^n)mod2^n$

For having an adder modulo 2^n, it's possible to use recurrence until t-th steps by :

$$|y,x\rangle \rightarrow |y,(y+x)mod2^n\rangle \rightarrow |y,(2y+x)mod2^n\rangle \rightarrow \cdots \rightarrow |y,\big((t-1).y+x\big)mod2^n\rangle \rightarrow |y,(ty+x)mod2^n\rangle$$

The quantum circuit for having the abscissa transformation could be represented by the Figure 15.

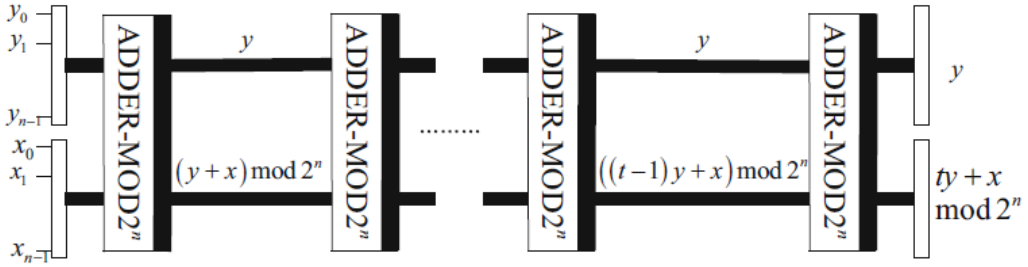

Fig 15- Quantum Circuit of the network $|x'\rangle$

2.7.2 Ordinate circuit $|y'\rangle$

$$y' = (mx + (tm+1)y)mod2^n$$

So,

$$|x,x\rangle \rightarrow |x,(2.x)mod2^n\rangle \rightarrow \cdots \rightarrow |x,mx\ mod2^n\rangle \rightarrow |y,mx\ mod2^n\rangle \rightarrow \cdots |y,y + mx\ mod2^n\rangle \rightarrow \cdots \rightarrow |y,\big((t-1).y+x\big)mod2^n\rangle \rightarrow |y,(tmy+x)mod2^n\rangle \rightarrow |y,(tmy+x)mod2^n\rangle \rightarrow |y,\big((tm+1)y+x\big)mod2^n\rangle \rightarrow |y'\rangle$$

(18)

Demonstration

The network $|y'\rangle$ gives input $|x\rangle$ and $|y\rangle$. The realization could be divised by $tm+m+1$ steps

The first (m-1)-th steps permit to calculate mx by using adder modulo.

In the m-th step, one of the input x is replaced by y for having $y+mx$.

To repeat this last step (m+1)-th step, we could have $\big((tm+1)y+x\big)mod2^n$

$$|y'\rangle = |(mx + (tm + 1)y) mod 2^n\rangle$$

So,

$$|x, x\rangle \to |x, (2.x) mod 2^n\rangle \to \cdots \to |x, mx\ mod 2^n\rangle \to |y, mx\ mod 2^n\rangle \to \cdots |y, y + mx\ mod 2^n\rangle \to \cdots \to |y, \big((t-1).y + x\big) mod 2^n\big\rangle \to |y, (tmy + x) mod 2^n\rangle \to |y, (tmy + x) mod 2^n\rangle \to |y, \big((tm+1)y + x\big) mod 2^n\big\rangle \to |y'\rangle$$

The quantum circuit of the network $|y'\rangle$ will so represented by the Figure 16

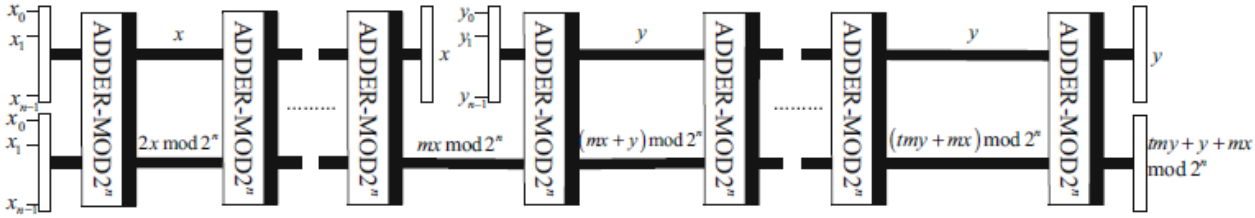

Fig 16- Quantum Circuit of the Network $|y'\rangle$

2.8 Quantum Cryptography on QPQ-CD

The Q-Crypto [13-14] algorithm implements a technique of scrambling on the digital image processed by quantum processors using QIS (Quantum Image Scrambling). The input of the algorithm is a classical image of size 16r × 16r. The matrix will be transformed into a quantum FRQI model then processed in QAT (Quantum Arnold Transform) and QHT (Quantum Hilbert Transform). Since the matrixes of the red, green, blue have their own component separately, the corresponding FRQI images are processed separately according to the inputs JR, JG, JB. To be able to process by conventional computers, the PQ-Crypto module after Q-Crypto, a measurement module FRQI makes it possible to determine the digital matrix result of qht_R, qat_R, qht_G, qat_G, qht_B, qat_B.

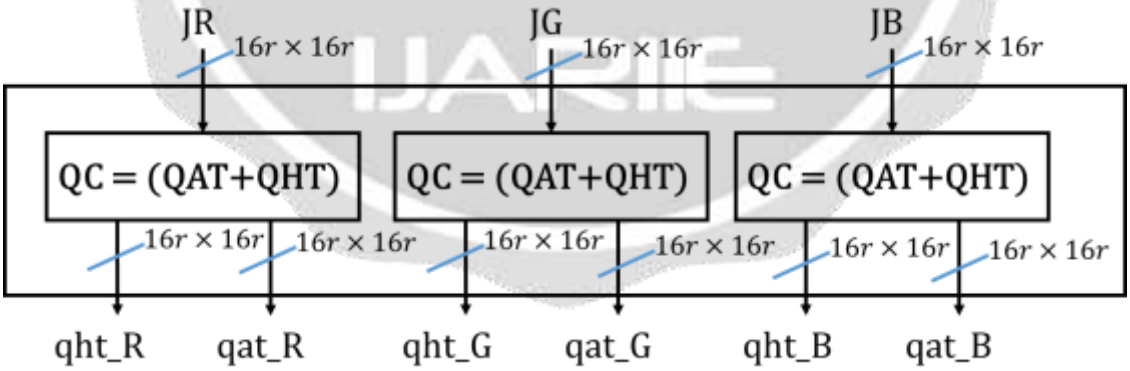

Fig -17 QC algorithm

In the general application qht_R, qat_R, qht_G, qat_G, qht_B, qat_B will be simplified by q formed by the components q1 ... .q6.

3. Interpretation

The following results have been obtained through Matlab simulation. The parameters used for the evaluation of the selected algorithm are: the PSNR (Peak Signal to Noise Ratio),the SSIM (Structural SIMilarity), the

NPCR(Number of Pixel Change Rate), the UACI (Unified Average Changing Intensity), the rxy (coefficient of correlation), the entropy and the response time of the algorithm.

- The PSNR is a unit of distortion used for measuring in matter of digital images. The PSNR is defined by the following formula [15-19] :

$$PSNR = 10.\log_{10}(\frac{d^2}{EQM}) \tag{19}$$

d being the possible maximum value for a pixel. In general d=255 and EQM is the average quadratic error defined by:

$$EQM = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(I_0(i,j) - I_r(i,j))^2 \tag{20}$$

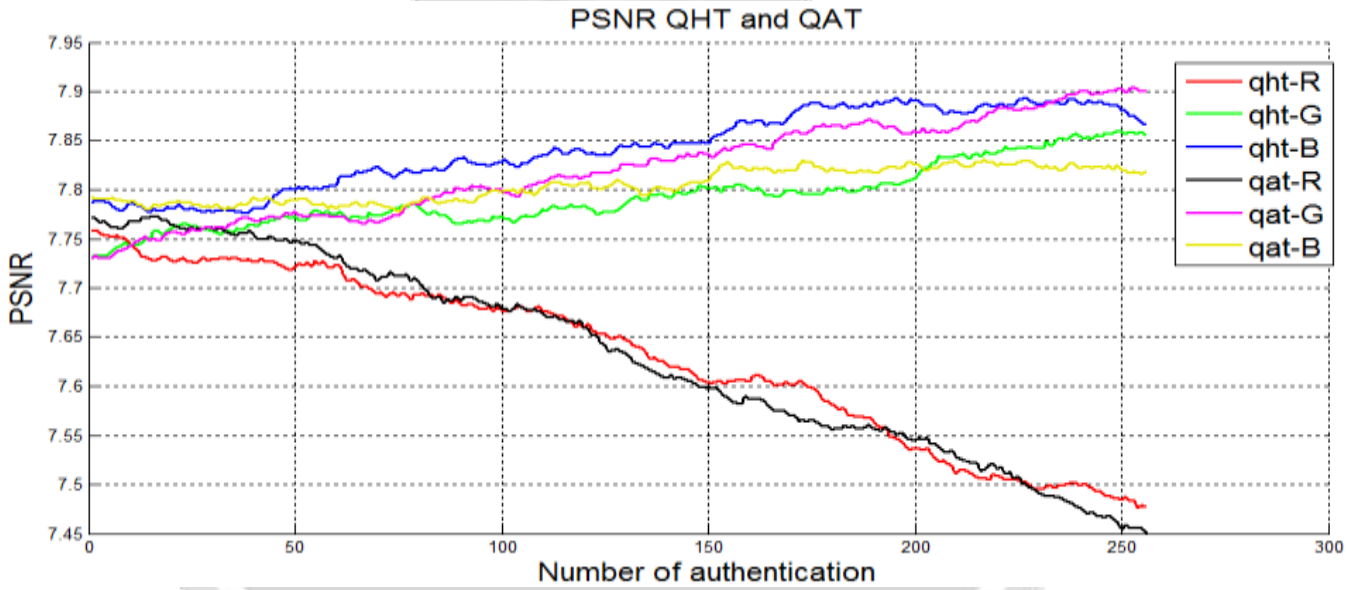

Fig -18 PSNR for QHT and QAT in QPQ-CD with r=16

Interpreting :

After QHT and QAT scrambling methods, the curve at the output gives the PSNR like in the Figure 18. Even for The Quantum, Image representing by FRQI, after computing PSNR, the value should be near 8. The part red is most quiet better than the two other parts Green and Blue. The PSN varies between 7.45 and 7.9. So, after the QHT and QAT, the transformed image still the properties of image's PSNR. The PSNR is used for measuring the proximity between the original image and the compressed image but it doesn't consider the visual quality of the reconstruction, thus, is a simple objective measure for visual quality only. So the PSNR could affirm that the image transformed is a good quality visual.

- The Structural Similarity ou SSIM is a reliable unit measurement for the similarity between two digital images [15-19].

$$SSIM(X,Y) = \frac{(2\mu_X\mu_Y + c_1)(2\sigma_X\sigma_Y + c_2)(2COV(X,Y) + c_3)}{(\mu_X^2 + \mu_Y^2 + c_1)(\sigma_X^2 + \sigma_Y^2 + c_2)(\sigma_X\sigma_Y + c_3)} \tag{21}$$

μ_X , μ_Y being the average of X, Y; σ_X^2, σ_Y^2 being the variance of X, Y; the covariance between X and Y; c_1, c_2, c_3 the three values used to stabilize the division in case the value is too low.

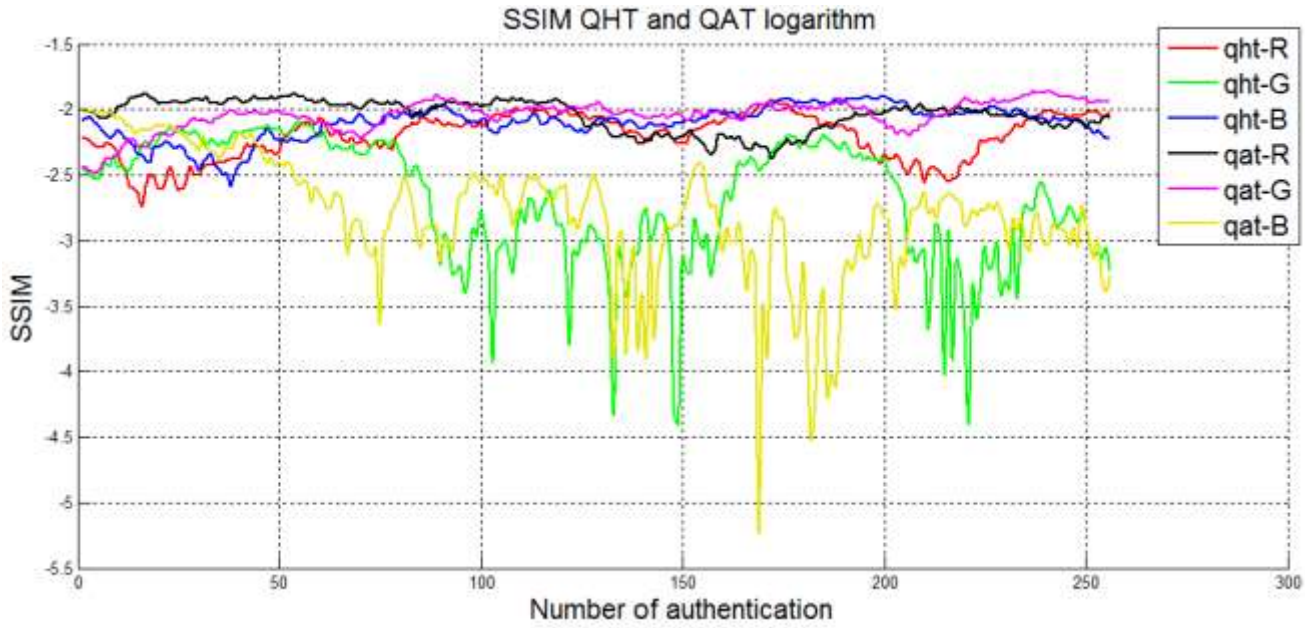

Fig -19 SSIM for QHT and QAT in QPQ-CD with r=16

Interpreting :

The SSIM is a similarity measure of the image. The similarity of the image on the Figure 19 is represented by logarithms using basis 10. .It shows that it less than 10^{-2} which is less than 1% of the similarity of the original image. QAT with the blue component could even achieve until 10^{-5} wich is near the 0.001%. So; the QAT and QHT give a good confusion image which is near zero. So, there is no similarity between the image transformed and original image.

- The NPCR is used to measure the percentage of pixels differentiating two given images.

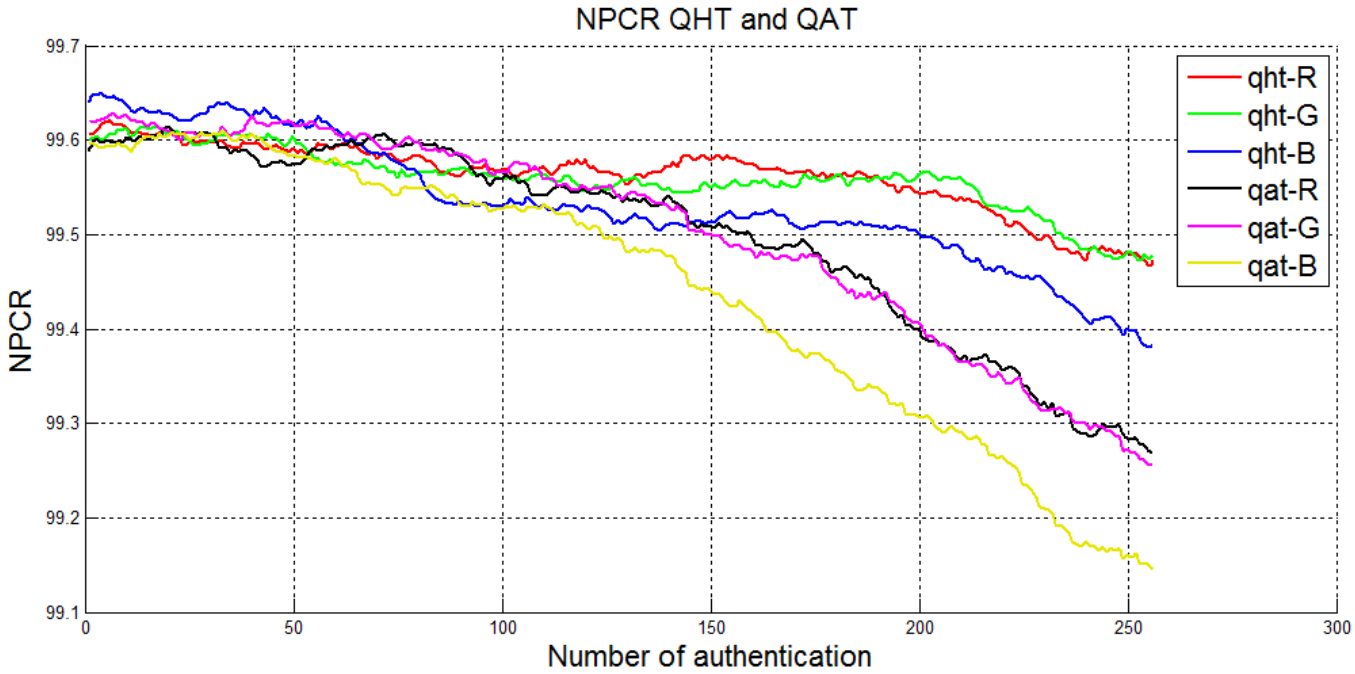

Fig -20 NPCR for QHT and QAT in QPQ-CD with r=16

Interpreting :

The NPCR is defined by [15-19] :

$$NPCR^{R/G/B} = \frac{\sum_{i=1}^{H}\sum_{j=1}^{W} D^{R/G/B}{}_{i,j}}{W \times H} 100\% \quad (22)$$

with

$$D_{i,j}^{R/G/B} = \begin{cases} 0 & si \quad C_{i,j}^{R,G,B} = \overline{C}_{i,j}^{R,G,B} \\ 1 & si \quad C_{i,j}^{R,G,B} \neq \overline{C}_{i,j}^{R,G,B} \end{cases} \quad (23)$$

$C_{i,j}^{R,G,B}$ and $\overline{C}_{i,j}^{R,G,B}$ represent the Red, Green and Blue channel colors of both images

$$L^{R/G/B} = 8$$

W and *H* represent the width and the length of the image.

Looking the difference pixel by pixel images, the scrambled image using QAT and QHT could achieve between 99.1 and 99.6 difference. So, there are many pixesl which change between the result and original image.

- UACI is the average value of two image light intensities [15-19].

$$UACI^{R/G/B} = \frac{1}{W \times H} \sum_{i=1}^{H}\sum_{j=1}^{W} \frac{C_{i,j}^{R/G/B} - \overline{C}_{i,j}^{R/G/B}}{2^{L^{R/G/B}} - 1} \times 100\% \quad (24)$$

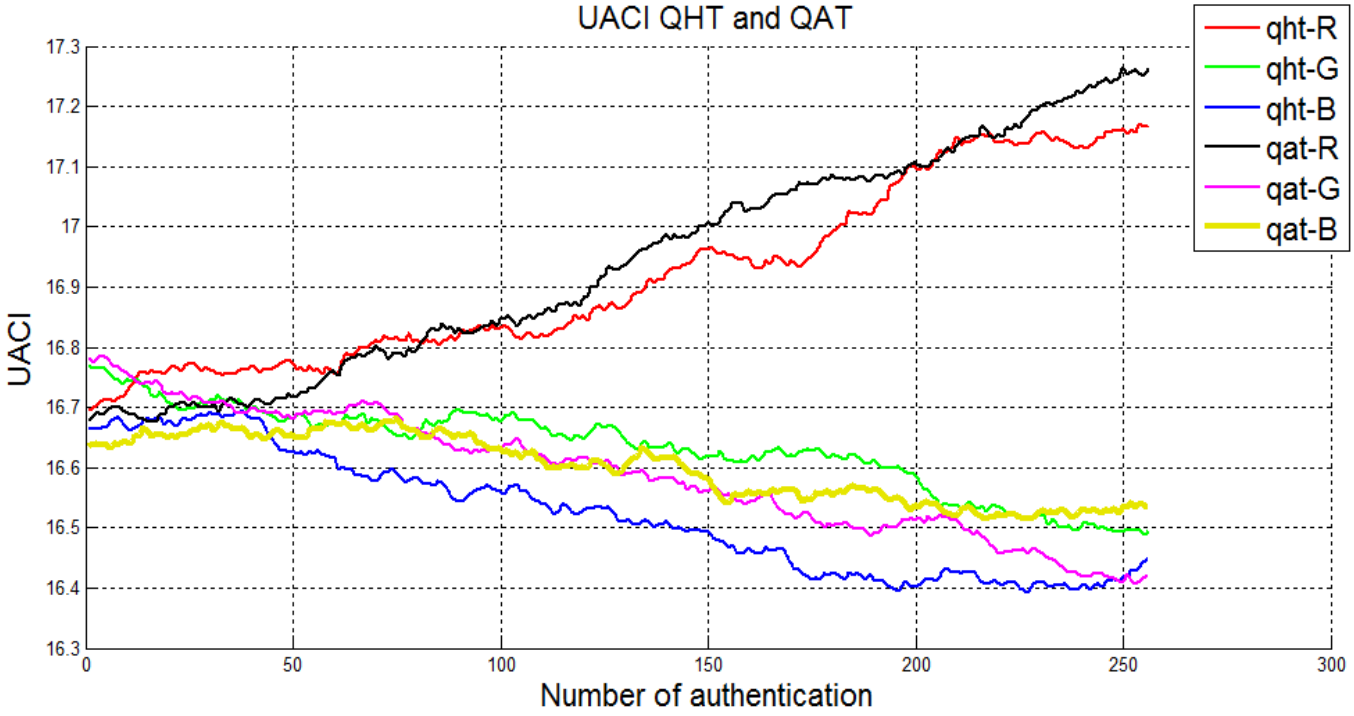

Fig -21 UACI for QHT and QAT in QPQ-CD with r=16

Interpreting :

UACI using QAT and QHT is not so good due to it's between 16.3 and 17.3%. The light intensity of the original image and scrambled image is also more similar and it makes easy for the attacker like DPA (Differential Power Analysis) method by analyzing the light intensity to recognize the image.

- Coefficient of correlation [15-19] is defined by :

$$r_{X,Y} = \frac{COV(X,Y)}{\sqrt{V(X).V(Y)}} = \frac{COV(X,Y)}{\sigma_X \sigma_Y} \tag{25}$$

COV(X, Y) being the covariance between the random variables X and Y ; $V(X), V(Y)$ being the variance of X and Y ; σ_X, σ_Y the classical gap between X and Y.

- The covariance is equal to the expectation of the product of the targeted variables. The covariance is defined by the following formula :

$$COV(X,Y) = E[(X - E[X])(Y - E[Y])] \tag{26}$$

E being the mathematical expectation; X, Y being any random variables.

- The variance is defined by the following formula :

$$V(X) = E[(X - E[X])^2] = COV(X, X) \tag{27}$$

E being the mathematical expectation; COV being the covariance.

The purpose of the covariance is to quantize the liaison between two random variables X et Y, so as to emphasize the aim of the liaison and its intensity. The coefficient of simple linear correlation of Bravais-Pearson (or of Pearson), as it is called, is a standardization of the covariance by the product of the classical variable gaps. The correlation varies between -1 and +1. The nearer the extreme values they are, the more likely and the stronger the similarity between the variables is. The expression « strongly correlated » means that both variables are quite similar and that their correlation move towards 1. The expression « linearly independent » or « total absence of correlation » means that there is no correlation at all, thus no similarity between the two random variables. The expression « thorough correlation » means that the value of r is ± 1.

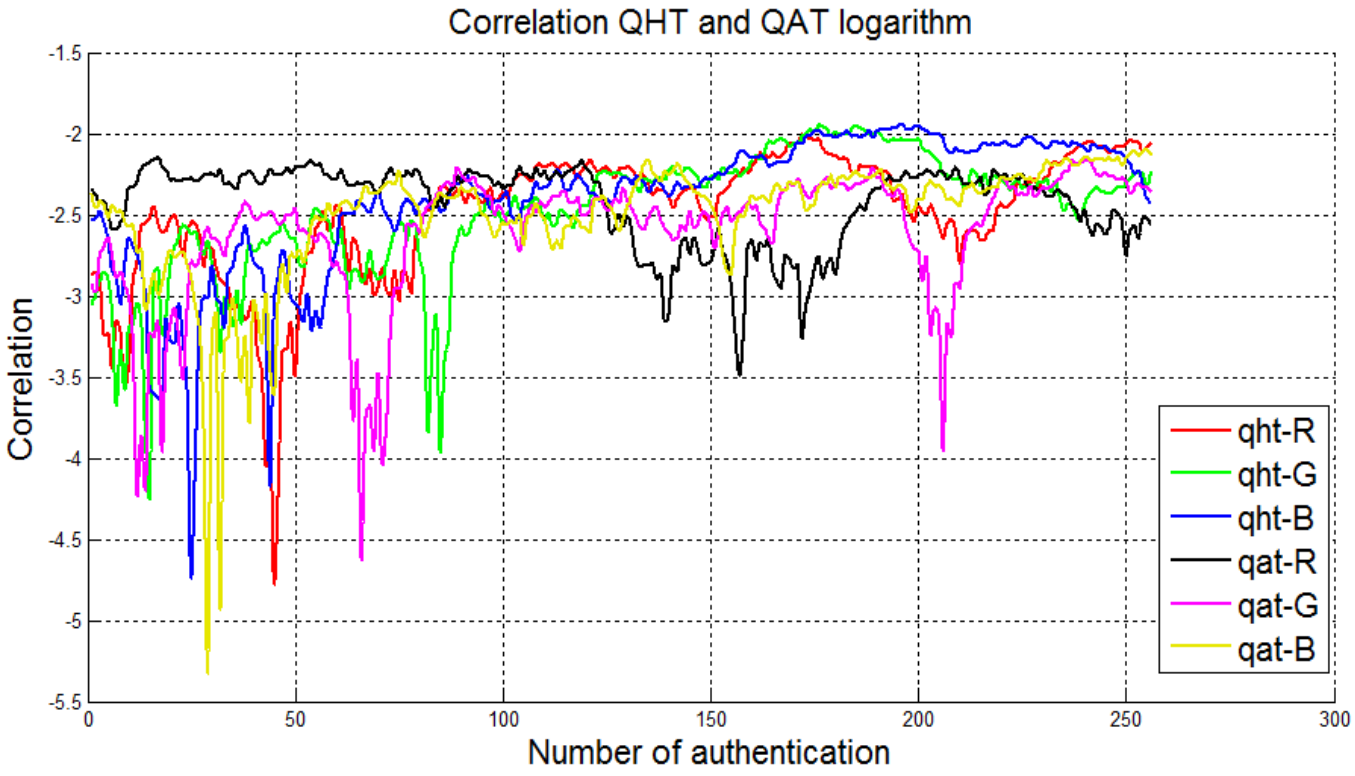

Fig -22 Correlation for QHT and QAT in QPQ-CD with r=16

Interpreting :

Like using with the SSIM, The Figure 22 also repesent the absolute of the logarithm of the correlation. It shows that the correlation is less than 10^{-2} which is also near the 1%. So there is no correlation between the scrambled image and the original image.

- The entropy binary is defined by the following formula [15-19]:

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x) \quad (28)$$

X being a random variable composed by one and zero; the entropy binary measures the uncertainty relating to the result of the new key after QPQ-CD.

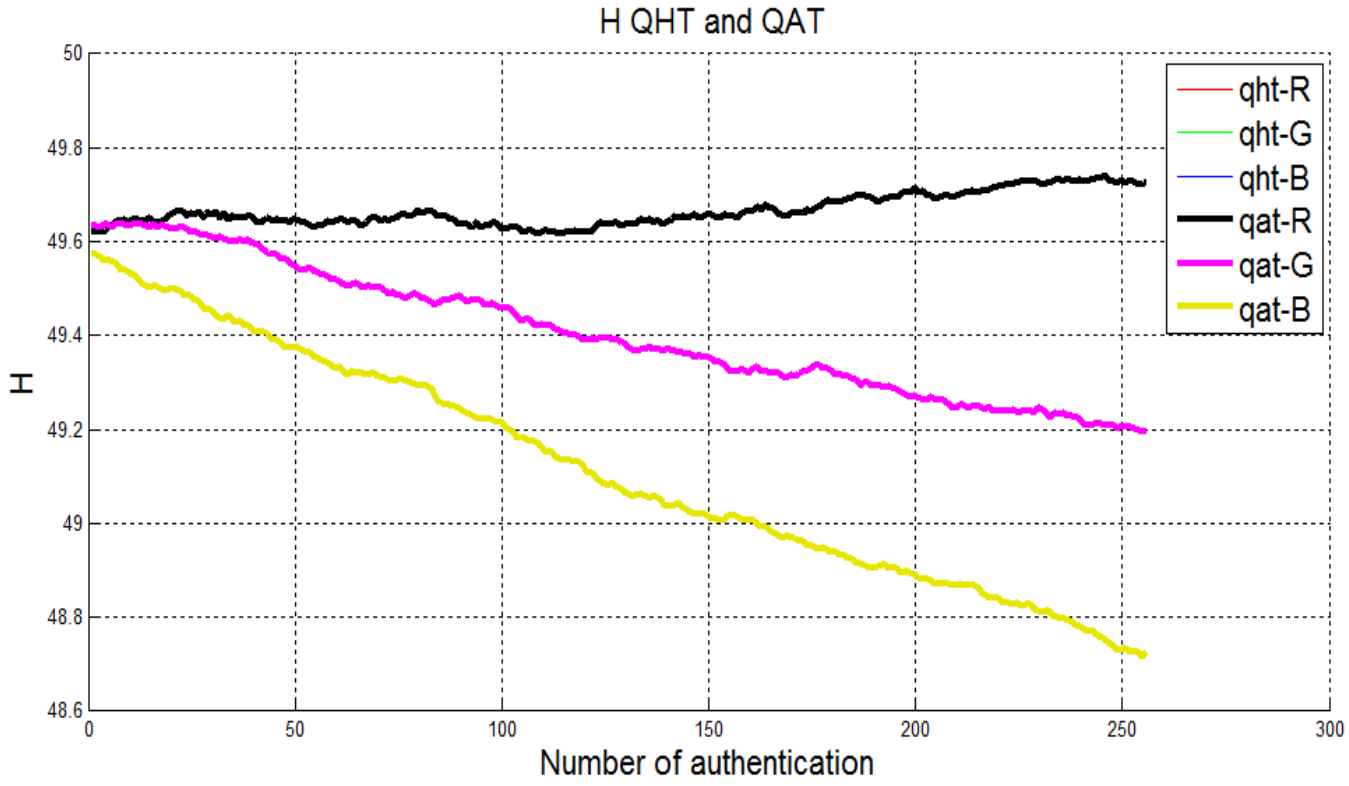

Fig -23 Binary entropy for QHT and QAT in QPQ-CD with r=16

Interpreting :

The binary entropy measure the disorder of the bit zeros and of the bit ones in the image code at binary. In Figure 23, the QAT and QHT image has always the same binary entropy, it's due the scrambling methods QAT and QHT modify only the position of the image not the value of this. The value of binary entropy is also not so good with value between 48.6% and less than 50%. So QAT and QHT give only a quiet good disorder scrambling methods. The entropy binary is maximal when the probability to have bit zeros and bit ones is identic and equal to 50%. In this, entropy will be 100%.

4. Conclusion

The QPQ-CD is an algorithm used when the mobile want to authenticate on the network. After registration, the mobile and the operator change dynamically the key of this. Before changing the key, the QPQ-CD is performed with the scrambling methods using the Quantic algorithm indeed QAT and QHT.

The scrambled image will be hashed with multiple algorithms and the QPQ-CD selects the best key of them. The Quantum Image Scrambling methods are implemented using Quantum Logic Gate. After this, it is analyzed following the PSNR to make sure that the image has a good quality which is near 8 ; SSIM to make sure that image not gives a high similarity between them which is near than 1% of similitude ; the NPCR to make sure that the pixel really change ; the correlation which is really good near than 1%. The other parameter doesn't give a good quality like the binary entropy which is near the 50% only and the UACI is also near 16% only. Even over parameters is not good, the Post Quantum Cryptography after the Quantum Cryptography will increase the result of key selection.

5. Bibliographie

[1] Y. Wu, H. Huang, C. Wang, Y. Pan, *« 5G Enabled Internet of Thing »*, CRC Press, 2019

[2] V. C. M. Leung, H. Zhang, X. Hu, Q. Liu, Z. Liu, *« 5G for Future Wireless Networks »*, ICST Institute for Computer Sciences, 2019

[3] V. C. M. Leung, H. Zhang, X. Hu, Q. Liu, Z. Liu, *« 5G for Future Wireless Networks »*, ICST Institute for Computer Sciences, 2019

[4] W. Lei, Anthony C.K. Soong, L. Jianghua ,W. Yong , B. Classon, W. Xiao, D. Mazzarese, Z.Yang, T. Saboorian, *«5G System Design An End to End Perspective»*, Springer, 2020

[5] H. Fattah, *« 5G LTE Narrowband Internet of Things »*, CRC Press, 2019

[6] S. M. A. Kazmi, L. U. Khan, N. H. Tran, C. S. Hong, *« Network Slicing for 5G and Beyond Networks »*, Springer, 2019

[7] Z. Ri-Gui, W. Qian, Z. Man-Qun and S. Chen-Yi, "A Quantum Image Encryption Algorithm Based on Quantum Image Geometric Transformations", Article in International Journal of Theoretical Physics, June 2012

[8] L. Shen-Yi, C. Chih-Shen, L. Li and H. Chua-Huang, "Tensor Product Formulation for Hilbert Space-Filling Curves", National Science Council, Taiwan, R.O.C. under grant NSC 91-2213-E-035-015, 2015

[9] C. A. Kuo-Liang, H. A Yi-Luen and L. Yau-Wen, "Efficient algorithms for coding Hilbert curve of arbitrary-sized image and application to window query", Information Sciences 177 2130–2151, 2007

[10] https://github.com/Alcinoos/QHT_RHT_THT , 2019

[11] Z. Ri-Gui, S. Ya-Juan and F. Ping, "Quantum image Gray-code and bit-plane scrambling, in Quantum Information Processing", Shanghai Maritime University, May 2015

[12] R. Z. Nan, X. H. Tian, H. G. Li, J. P. Dong and H. L. Qing, "Quantum image encryption based on Generalized Arnold Transform and double random-phase encoding" in Quantum Inf Process Nanchang University; Shanghai Jiao Tong University, 28 January 2015

[13] H. Liu, B. Zhao, L. Huang, *«Quantum Image Encryption Scheme Using Arnold Transform and S-box Scrambling»*, Journal of MDPI, Mar. 2019

[14] S. Heidari, M. Houshmand, N. T. Mashadi, *«A dual quantum image scrambling method»*, Quantum Information Processing, Jan. 2019

[15] Mamy Alain Rakotomalala, Tahina E. Rakotondraina et Sitraka R. Rakotondramanana, *« Transmission sécurisée d'image utilisant un chiffrement par bloc combine avec la transformée d'Arnold »*, Afrique SCIENCE

[16] Mamy Alain Rakotomalala, Falimanana Randimbindrainibe,Sitraka R. Rakotondramanana, Roméo T. Rajaonarison, *«Performances Ofimage Encryption Based on Chaotic Artificial Neuronal Networks Combined With The Fibonacci Transform», IOSR, Vol.20 Jul-Aug 2018*

[17] Mamy Alain Rakotomalala, Tahina E. Rakotondraina, Sitraka R. Rakotondramanana, *«Contribution for Improvement of Image Scrambling Technique Based on Zigzag Matrix Reodering»*, IJCTT, Vol. 61, Jul 2018

[18] Mamy Alain Rakotomalala1, Roméo T. Rajaonarison,Falimanana Randimbindrainibe, Sitraka R. Rakotondramanana, *«Image Ciphering Based On chaotic ANN and Fibonacci Transform Improved by using the Wavelet Transform»*, IJCTT, Vol. 61, Jul 2018

[19] Mamy Alain Rakotomalala, Falimanana Randimbindrainibe, Sitraka R. Rakotondramanana, *«Symmetric Image Encryption using Scrambling Technique Based on Matrix Reodering Coding»*, ***IJCTT***, Vol. 62, Aug. 2018