A Synchronous Stream Cipher Generator Based on Quadratic Fields (SSCQF)

Younes ASIMI
LabSiv, Equipe SCAM
Faculty of sciences, Ibn Zohr
University B.P 8106, City Dakhla, Agadir, Morocco

Ahmed ASIMI
LabSiv, Equipe SCAM
Faculty of sciences, Ibn Zohr
University B.P 8106, City Dakhla, Agadir, Morocco

***Abstract*—In this paper, we propose a new synchronous stream cipher called SSCQF whose secret-key is $K_S = (\mathrm{z}_1,...\mathrm{z}_\mathrm{N})$ where z_i is a positive integer. Let $d_1, d_2,..., d_N$ be N positive integers in $\{0,1,...,2^m - 1\}$ such that $d_i \equiv z_i \bmod 2^m$ with $m \in \mathbb{N}$ and $m \geq 8$. Our purpose is to combine a linear feedback shift registers LFSRs, the arithmetic of quadratic fields: more precisely the unit group of quadratic fields, and Boolean functions [14]. Encryption and decryption are done by XRO'ing the output pseudorandom number generator with the plaintext and ciphertext respectively. The basic ingredients of this proposal stream generator SSCQF rely on the three following processes:**

In process I, we constructed the initial vectors $IV = \{\mathrm{X}_1,...,\mathrm{X}_\mathrm{N}\}$ from the secret-key $K_S = (\mathrm{z}_1,...\mathrm{z}_\mathrm{N})$ by using the fundamental unit of $\mathbb{Q}\left(\sqrt{d_i}\right)$ if d_i is a square free integer otherwise by splitting d_i, and in process II, we regenerate, from the vectors X_i, the vectors Y_i having the same length L, that is divisible by 8 (equations (2) and (3)). In process III, for each Y_i, we assign $L/8$ linear feedback shift registers, each of length eight. We then obtain $N \times L/8$ linear feedback shift registers that are initialized by the binary sequence regenerated by process II, filtered by primitive polynomials, and the combine the binary sequence output with $L/8$ Boolean functions. The keystream generator, denoted K, is a concatenation of the output binary sequences of all Boolean functions.

Keywords*—Synchronous stream cipher SSCQF; linear feedback shift registers LFSRs; arithmetic of quadratic fields; Boolean functions; pseudorandom number generator and keystream generator

I. Introduction

The proposed stream cipher SSCQF is a binary addition stream cipher [14]. In a binary addition stream cipher, the plaintext is given as a string $m_1, m_2,...$ of elements of the finite field $\mathbf{k}_2 = \{0,1\}$. The keystream $z_1, z_2,...$ is a binary pseudorandom sequence [13]. The sender encrypts the plaintext message according to the rule $c_t = m_1 \oplus z_t$ for all $t \geq 0$. The ciphertext $c_1, c_2,...$ is decrypted by the receiver by adding bitwise the keystream $z_1, z_2,...$ to the received ciphertext sequence $c_1, c_2,...$. Sender and receiver produce the keystream $z_1, z_2,...$ via identical copies of the stream generator.

Let $z_1, z_2,..., z_N$ be N positive integers, $d_1, d_2,..., d_N$ be N positive integers in $\{0,1,...,2^m - 1\}$ such that $d_i \equiv z_i \bmod 2^m$ with $m \in \mathbb{N}$ and $m \geq 8$, and ε_i be a fundamental unit of a quadratic field $\mathbb{Q}\left(\sqrt{d_i}\right)$, if d_i is a square free integer.

In this paper, we propose a new synchronous stream cipher called SSCQF whose secret-key is $K_S = (z_1,..., z_N)$ where z_i are positive integers, based upon the combination of a linear feedback shift registers LFSRs [14], the congruence modulo 2^m with $m \in \mathbb{N}$ and $m \geq 8$, the arithmetic of quadratic fields: more precisely the unit group of quadratic fields, and the $L/8$ combining functions. The basic ingredients of this proposal stream cipher generator SSCQF rely on the following three processes:

In process I, we construct the initial vectors $IV = \{X_1,..., X_N\}$ from the secret-key K_S by using the fundamental unit of $\mathbb{Q}\left(\sqrt{d_i}\right)$ if d_i is a square free integer otherwise by splitting d_i, and in process II, we regenerate, from the vectors X_i, the vectors Y_i having the same length L, more precisely the length L must be divisible by eight (Equations (2) and (3)). In process III, for each Y_i, we assign $L/8$ linear feedback shift registers of length eight filtered by primitive polynomials of degree eight. They are $\frac{\varphi(2^8 - 1)}{8} = 25$ primitive polynomials [12]. We then obtain $N \times L/8$ linear feedback shift registers that are initialized by the binary sequence regenerated by process II. And we combine the output binary sequence of all linear feedback shift registers, namely, LFSR_{ij} with $L/8$ Booleans functions $\mathrm{R}_1, ..., \mathrm{R}_{L/8}$. The Boolean function R_j combines the output bits of LFSR_{ij} for all $i \in \{1,..., N\}$. The keystream generator denoted K, is a concatenation of the output binary sequences of all Boolean functions R_j.

The output function of our stream cipher is parameterized only by the secret-key K_S. As the keystream bits are produced independently of the plaintext, the proposed stream cipher SSCQF belongs to the category of synchronous stream ciphers.

In this section, we introduce the notations that will be used throughout this paper in TABLE 1.

TABLE I. NOTATIONS

K_s	: Input secret-key.
keystream	: Output secret-key.
$\oplus$	: XOR operation.
$\|\|$	: Concatenation.
LFSR_{ij}	: Linear feedback shift registers.
R_j	: Boolean functions.
F	: Feedback function.
$\overset{-2}{x}$	: Binary sequence of any integer x.
IV	: Initial Vector.
$\mathbf{k}_2$	: Binary finite field of characteristic two.
$\mathbf{k}_2^m$	: $\mathbf{k}_2$-vector space of dimension m.
$Lmc(k,k')$	: Lowest common multiple of positive integers $k;\ k'$.
Γ	: Set of periodic binary functions not necessarily the same period.
$\mathbb{N}$	: Set of natural numbers.
$\sqrt{\ }$	: Square root.
L_{Bi}	: Length of ith binary sequence.
$\mathrm{L}_{1/2\mathrm{Bi}}$	: Half-length of ith binary sequence.

II. PRELIMINARY

Stream cipher [14] is a secret-key cryptosystem constructed for improve secrecy of transmitted data. It is a lightweight and efficient cryptographic primitive for ensure confidentiality of transmitted data between two communicated pairs. It proves its robustness by its ability to resist against attacks [3][4] [7][14]. It has a wide application area especially in mobile devices and embedded systems. In this section we introduce the notation and terminology that will be used throughout the proposal. We use the symbol $\mathbf{k}_2 = \{0,1\}$ to denote the binary finite field of characteristic two, $\oplus$ to denote logical XOR (OR exclusive), $\mathbf{k}_2^m = \{0,1\}^m$ to denote the $\mathbf{k}_2$-vector space of dimension m, $\overset{-2}{n}$ to denote the binary sequence of any integer $n \in \mathbb{N}^*$ and $\|$ denotes concatenation of two bits sequences. Bit sequence means a sequence built from 0 and 1.

Definition 2.1: Let $X = (x_1,..., x_n)$ and $Y = (y_1,..., y_n)$ be two vectors of $\mathbf{k}_2^n = \{0,1\}^n$.

1) $X = Y$ if only if $x_i = y_i$ for all $i \in \{1,..., n\}$.
2) $X \oplus Y = (x_1 \oplus y_1,..., x_n \oplus y_n)$.

Theorem 2.1: Let X, Y and Z be three vectors of $\mathbf{k}_2{}^{\mathrm{n}} = \{0,1\}^n$.
$X = Y$ if and only if $X \oplus Z = Y \oplus Z$.

Proof : Let $X = (x_1,..., x_n)$, $Y = (y_1,..., y_n)$ and $Z = (z_1,..., z_n)$ be three vectors of $\mathbf{k}_2^n = \{0,1\}^n$. $X \oplus Z = Y \oplus Z$ if and only if $x_i \oplus z_i = y_i \oplus z_i$ for all $i \in \{1,..., n\}$

(**Definition 2.1**), if and only if $(x_i \oplus z_i) \oplus z_i = (y_i \oplus z_i) \oplus z_i$ if and only if $x_i \oplus (z_i \oplus z_i) = y_i \oplus (z_i \oplus z_i)$ if and only if $x_i \oplus 0 = y_i \oplus 0$ if and only if $x_i = y_i$ if and only if $X = Y$.

Let m be a positive integer. A binary feedback shift register (FSR) of length m is uniquely determined by its feedback function $F : \{0,1\}^m \to \{0,1\}$.

Definition 2.2 (see [20]): A feedback function $F : \{0,1\}^m \to \{0,1\}$ is nonsingular if and only if the algebraic normal form of F has the form $F(x_0,..., x_{m-1}) = x_0 + G(x_1,..., x_{m-1})$, where $G : \{0,1\}^{m-1} \to \{0,1\}$ is a polynomial in the variables $x_1,..., x_{m-1}$.

If the feedback function F of an m-stage feedback shift register is linear, one speaks of a linear feedback shift registers (LFSR). Otherwise one speaks of a nonlinear feedback shift register (NLFSR). All feedback shift registers used in this paper are nonsingular and linear. In this case, $F(x_0,..., x_{m-1}) = x_0 + a_1 x_1 + ... + a_{m-1} x_{m-1}$ modulo 2 where the a_i's are either 0 or 1 for all $i \in \{1,..., m-1\}$ and its linear recursion is of the form: $x_{n+m} = x_n + \sum_{i=1}^{i=m-1} a_i x_{n+i}$

modulo 2 for all $n \geq 0$ [6][11][17]. An alternative way to describe this recursion is to specify the m^{th} degree binary characteristic polynomial [16]: $f(x) = x^m + \sum_{i=1}^{i=m-1} a_i x^i + 1.$

To obtain the maximal period of $2^m - 1$, a sufficient condition is that $f(x)$ be a primitive m^{th} degree polynomial modulo two.

Definition 2.3 (see [12])**:** Let $f(x) \in \mathbf{k}_2[x]$ be a polynomial of degree at least l. Then $f(x)$ is said to be irreducible over $\mathbf{k}_2$ if it cannot be written as a product of two polynomials in $\mathbf{k}_2[x]$, each of positive degree.

Definition 2.4 (see [12])**:** Let $f(x) \in \mathbf{k}_2[x]$ be an irreducible polynomial of degree N. Then $\mathbf{k}_2[x]/(f(x))$; the set of polynomials in $\mathbf{k}_2[x]$ of degree less than N, is a field of order 2^N. Addition and multiplication are performed modulo $f(x)$. Therefore $\mathbf{k}_{2^N} = \mathbf{k}_2[x]/(f(x))$. In this case, $\mathbf{k}_{2^N}$ is called the splitting field of $f(x)$.

Definition 2.5 (see [12])**:** A polynomial $f(x) \in \mathbf{k}_2[x]$ of degree N is called a primitive polynomial over $\mathbf{k}_2$ if it is the minimal polynomial over $\mathbf{k}_2$ of a primitive element of $\mathbf{k}_{2^N}$.

Definition 2.6 : We call a Boolean function upon $\{0,1\}^N$, all function defined from $\{0,1\}^N$ into $\{0,1\}$. They are 2^{2^N} Boolean functions upon $\{0,1\}^N$.

III. A Brief Description of SSCQF Algorithm

Stream cipher encrypts the plaintext by using a key stream generator. The latter can be a synchronous or an asynchronous stream cipher. This property is related to regenerate a nature of secret-key. A generator is qualified as a synchronous stream cipher if the regeneration of the secret-keys carries out independently of the plaintext and ciphertext messages. By contrast, an asynchronous stream cipher products the keystreams as a function of the input secret-key and previous ciphertexts [14]. Our synchronous algorithm SSCQF can briefly be described as follows:

It takes a secret-key constructed by a sequence of positive integers $z_1,...,z_N$ and let $d_i \equiv z_i \bmod 2^m$ with $m \in \mathbb{N}$ and $m \geq 8$.

For each d_i we assign them only two positive integers n_i and m_i as follows:

- If $d_i = s_i^{\,2} r_i$ where $r_i = 1$ or r_i is a square free integer, then $n_i = r_i$ and $m_i = s_i^2$.
- If d_i is a square free integer, then we assign only one fundamental unit ε_i of the quadratic field

$\mathbb{Q}\left(\sqrt{d_i}\right)$ [2] [5] where

$$\varepsilon_i = \begin{cases} n_i + m_i\sqrt{d_i} & \text{if } d \equiv 2 \text{ or } 3 \mod 4 \\ \dfrac{n_i + m_i\sqrt{d_i}}{2} & \text{if } d \equiv 1 \mod 4 \end{cases} \qquad (1)$$

We then construct the initial vectors $IV = \{X_1,..., X_N\}$ where $X_i = \overline{n_i}^2 \,\|\, \overline{d_i}^2 \,\|\, \overline{m_i}^2$ for all $i \in \{1,..., N\}$. Since the vectors X_i do not have the same length, then we regenerate the vectors Y_i, from the vectors X_i, having the same length L. The number L is divisible by eight via equations 2 and 3. Each binary standard sequence is subdivided into $L/8$ binary sequences of length eight, each of them initializes one linear feedback shift register of length eight. We then obtain $L/8$ LFSRs for each Y_i, namely, $\mathrm{LFSR}_{i1},..., \mathrm{LFSR}_{\mathrm{iL/8}}$ filtering by primitive polynomials of degree eight. And we combine the output binary sequence of all LFSR_{ij} with $L/8$ Boolean functions $\mathrm{R}_1, ..., \mathrm{R}_{L/8} : \{0,1\}^N \rightarrow \{0,1\}$ defined as follows: For each $j \in \{1,..., L/8\}$, the Boolean function R_j combines the output bits of LFSR_{ij} for all $i \in \{1,..., N\}$. The keystream digit is obtained by concatenation of the output binary sequences of all Boolean functions R_j.

IV. Detailed Description of SSCQF Algorithm

The overall structure of the keystream generator SSCQF is depicted in the following figure.

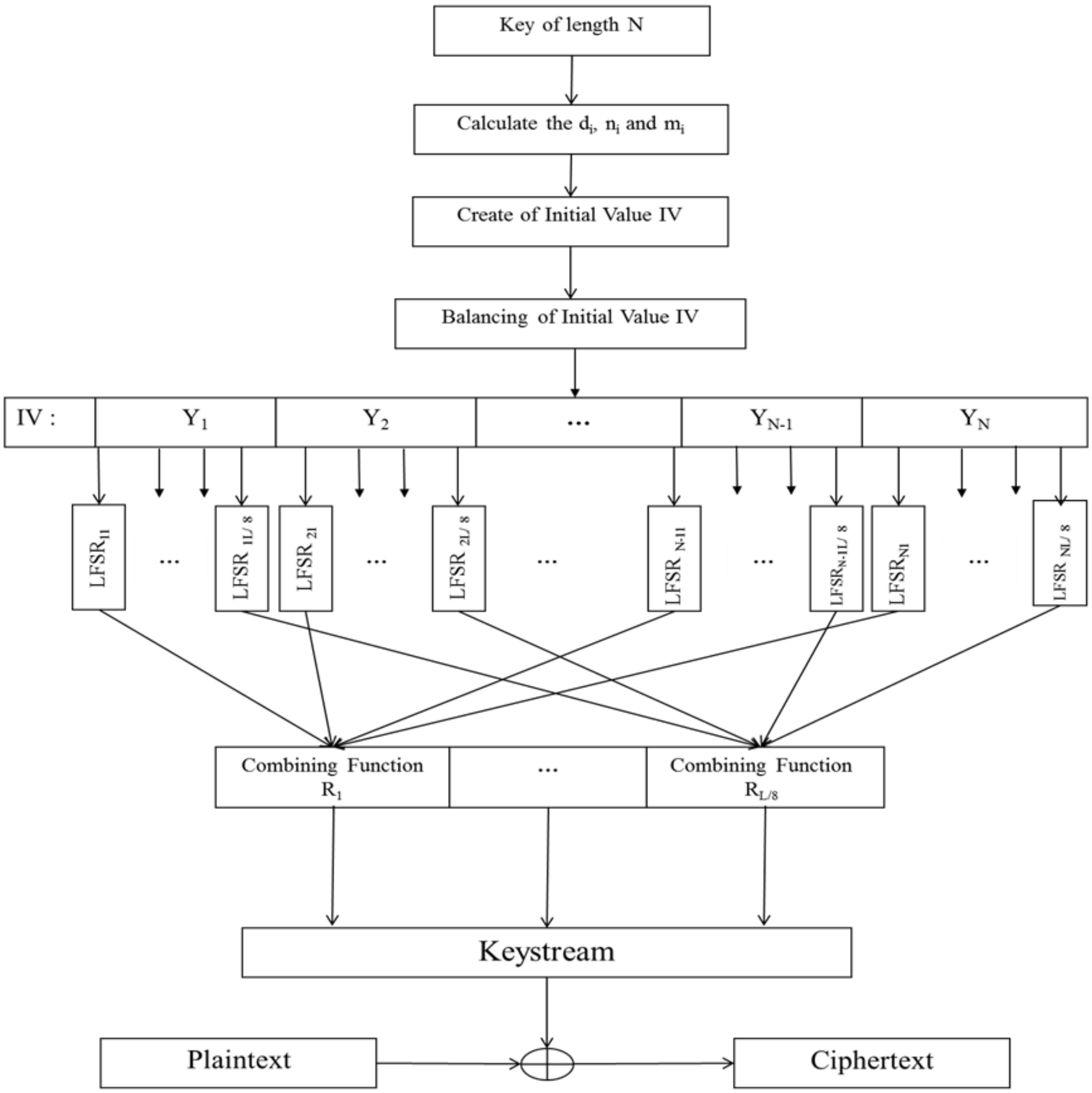


Fig. 1. Detailed description of ASCGQF algorithm

The basic ingredients of the keystream generator SSCQF rely on the following three processes:

A. Process I

The main goal of this process is to generate the initial vectors $IV=\{X_1,...,X_N\}$ from a secret-key $K_S=(z_i,...,z_N)$ where z_i are positive integers for all $i\in\{1,...,N\}$. We then proceed as follows:

- We compute the positive integers d_i such that $d_i = z_i \bmod 2^m$ with $m\in\mathbb{N}$ and $m\geq 8$ for all $i\in\{1,...,N\}$.
- For each d_i we assign only two positive integers n_i and m_i:
  - Assume that $d_i = s_i^2 r_i$ where $r_i = 1$ or r_i is a square free integer, we then get $n_i = r_i$ and $m_i = s_i^2$.
  - Assume that d_i is a square free integer, we assign only one fundamental unit ε_i of the quadratic field $\mathbb{Q}\left(\sqrt{d_i}\right)$ [2] [5] together with

$$\varepsilon_i = \begin{cases} n_i + m_i\sqrt{d_i} & \text{if } d \equiv 2 \text{ or } 3 \bmod 4 \\ \dfrac{n_i + m_i\sqrt{d_i}}{2} & \text{if } d \equiv 1 \bmod 4 \end{cases}$$

- For all $i\in\{1,...,N\}$, $X_i = \overline{n_i}^2 \,\|\, \overline{d_i}^2 \,\|\, \overline{m_i}^2$.

B. Process II

The vectors X_i for all $i \in \{1,...,N\}$ are not necessarily of the same length. The goal of this process is to balancing those vectors. For that, we then choose a vector of a maximal length, for example X_k of length $l_k = L'$, and we proceed as follows :

For each vector $X_i = (x_{i1},...,x_{il_i})$ one assigns the only vector $Y_i = (y_{i1},...,y_{iL})$ defined as follows:

If $L \equiv 0 \mod 8$, $L = L'$, we get:

$$\begin{cases} y_{ij} = x_{ij} & \text{for all} \quad 0 \le j \le \mathrm{l_i} \\ y_{\mathrm{i}(\mathrm{l_i}+t)} = \mathrm{x}_{\mathrm{i(t\,mod\,li)}} \oplus \mathrm{x_{kt}} & \text{for all} \ 0 \le \mathrm{t} \le \mathrm{L - l_i} \end{cases} \tag{2}$$

Otherwise, $L = L' + (8 - L' \mod 8)$, we get:

$$\begin{cases} y_{ij} = x_{ij} & \text{for all} \quad 0 \le j \le \mathrm{l_i} \\ y_{\mathrm{i}(\mathrm{l_i}+t)} = \mathrm{x}_{\mathrm{i}(\mathrm{t\ mod\ l}i)} \oplus \mathrm{x_{kt}} & \text{for all} \ 0 \le \mathrm{t} \le \mathrm{L'\text{-}l_i} \\ y_{i(L'+s)} = \sum_{t=0}^{s} \mathrm{x_{it}} \oplus \mathrm{x_{ks}} & \text{for all} \ 0 \le \mathrm{s} \le 8 - (\mathrm{L'} \mod 8) \end{cases} \tag{3}$$

C. Process III

The vectors Y_i for all $i \in \{1,...,N\}$ generated in the process II, are of the same length L divisible by eight. We subdivide it into $L/8$ binary sequences of length eight; each initializes a linear feedback shift register filtered by the primitive polynomial of degree eight. We then obtain, for each Y_i , $L/8$ linear feedback shift registers, namely, $\mathrm{LFSR}_{i1},...,\mathrm{LFSR}_{\mathrm{iL/8}}$. And we combine the output binary sequence of all LFSR_{ij} with $L/8$ Boolean functions $\mathrm{R}_1, ..., \mathrm{R}_{L/8} : \{0,1\}^N \to \{0,1\}$ defined as follows: For each $j \in \{1,...,L/8\}$, the Boolean function R_j combines the output bits of LFSR_{ij} for all $i \in \{1,...,N\}$, together with $R_j(x_1,...,x_N) = R(x_1,...,x_{j-1},1,x_{j+1},...,x_N)$ and $R(x_1,...,x_N) = \sum_{i=1}^{i=N} x_i + \sum_{i<j=1}^{N} x_i x_j \mod 2$. The keystream is obtained by concatenation of the output binary sequences of all Boolean functions.

V. Behavioral Study

After presenting and explaining the principle components of our SSCQF algorithm, in this section, we focus a behavioral study for all elements constituting our regenerator in order to highlight its internal characteristics. We begin by studying the complexity of the output binary sequences of all Boolean functions R_j related to their lengths for a given password. Effectively, our goal, in this subsection, is to appear the cryptographic nature of the internal states of our regenerator of binary sequences. Then, we pass to analysis the keystream regenerated by our system after the minimal perturbations on the initial condition. Finally, we present an analytical study simulating the human system.

A. Correlation and normalized distance of periodic binary strings

For the binary sequences, we must exploit the Hamming principle to make sure their nature distribution. It aids in estimating the complexity of binary strings that have the same period. However, the testing of the keystreams regenerated by our regenerator show that not necessarily of the same period. Hence, we should use an extension of a Hamming distance as we defined in [1] [21]:

Let S and S' be two elements of Γ of periods k and k' respectively and $K = Lmc(k,k')$.

The function $D' : \Gamma \times \Gamma \to [0,1]$ defined by:

$$D'(S,S') = \frac{\sum_{i=0}^{K-1} \left(\left(S(i) + S'(i) \right) \% 2 \right)}{K} \tag{4}$$

is a normalized distance of Γ.

Also in [21], we defined another interesting property allowing to more ensure the nature of binary sequences: uncorrelation of the binary strings. Thus, for all S and S' in Γ, we say that two binary strings are weakly correlated if:

$$D'(S,S') \simeq 0.5 \tag{5}$$

This property allows us to prove the complexity of the binary sequences not necessarily of the same period. More precisely, the obtained values of a normalized distance are used to make sure about the uncorrelation or the correlation of the sets of periodic binary strings.

B. Impact of the lengths on the output binary sequences of all Boolean functions

Firstly, we propose an analysis study of each output binary sequences of all Boolean functions R_j related to their lengths for a given password. In this case, we change the length of output binary sequences of all Boolean functions R_j in order to ensure the internal nature of our regenerator. For this object, we propose a fixed secret-key $K_S = (z_i,...,z_N)$ where z_i are positive integers and N equal to 50 as follows:

K_s={12, 3, 6, 77, 80, 81, 90, 95, 44, 54, 56, 47, 2, 8, 10, 15, 18, 16, 28, 99, 29, 55, 60, 67, 86, 84, 26, 37, 35, 34, 311, 57, 41, 5, 13, 11, 512, 73, 92, 40, 42, 47, 19, 388, 39, 71, 73, 79, 188, 115}

For each case, for same secret-key K_s, we adapt our program to regenerate the primitive signals not have the same length. Then, we obtain:

- In first case (Fig.2), the length of a binary sequence is: L_{B1}=2005 bits.
- In second case (Fig. 3), the length of a binary sequence is: L_{B2}=4005 bits.
- In third case (Fig.4), the length of a binary sequence is: L_{B3}=6005 bits.

From [14], we say the binary sequences $X_1,...,X_N$ of same lengths are independent if each taking on the values 0 or 1 with probability $1/2$. Then, we talk about the unpredictable and uncorrelated primitive signals if the distribution of hamming distance accumulates near to half-length ($L_{1/2Bi}$) of this binary sequence. This means that almost half the bits in same position of two set of the binary sequence are different.

- $L_{1/2B1}$≈1002 bits.
- $L_{1/2B2}$≈2002 bits.
- $L_{1/2B3}$≈3002 bits.

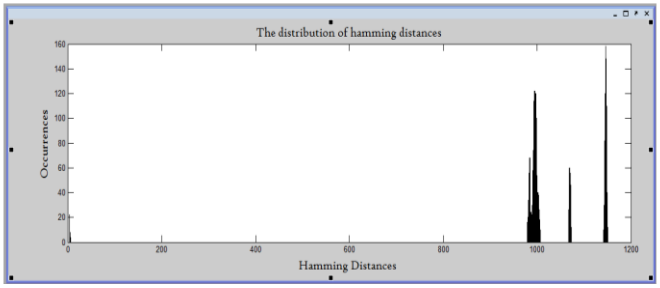


Fig. 2. The distribution of hamming distances for L_{B1}=2005 bits

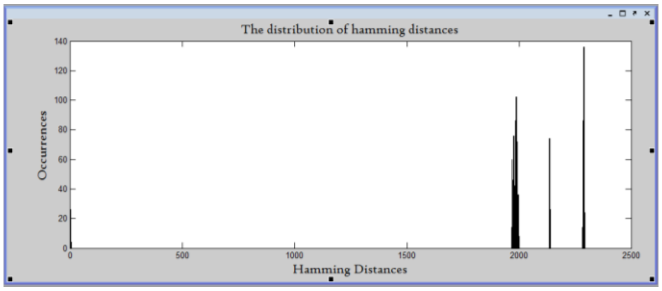


Fig. 3. The distribution of hamming distances for L_{B2}= 4005 bits

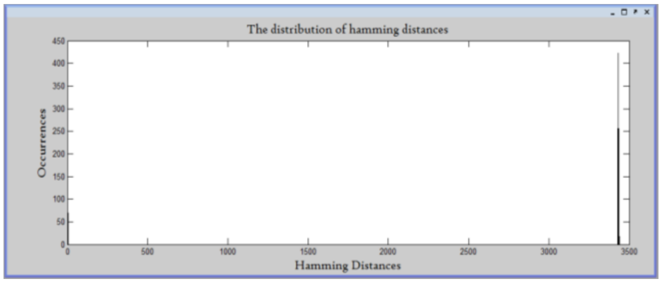


Fig. 4. The distribution of hamming distances for L_{B3}= 6005 bits

From these histograms, we notice, for a same secret-key, the distribution of hamming distances in these three cases accumulates in the vicinity of half-length of each output binary sequences of all Boolean functions. In addition, the obtain results are almost identical in all three histograms. In two first cases, we have three accumulations regions nearest to half-length. But, in third case, we have only a peak nearest to half-length. Accordingly, the cryptographic nature of each primitive signal in any internal state is not only related to the length of the regenerated a binary sequence. Effectively, these results are strongly linked to Boolean Functions and linear feedback shift registers filtered by the primitive polynomials of degree eight integrated in our system. Hence, our purpose has unpredictable internal characteristics [1][21], which is recommended in order to resist against attack periodic sequences [5][10]. This enables us to ensure the cryptographic nature of SSCQF algorithm. Finally, for each internal state, we can summarize these features as follows:

- The length of each block regenerated has a positive effect on the cryptographic quality of the regenerated primitive signals.
- The distribution of lengths and periods are random.
- The primitive signals are unpredictable or cryptographically strong.
- When we increase the period length of the internal states, their regenerated the primitive signals became more uncorrelated. Then, long period has a positive impact on the cryptographic nature of internal primitive signs. This property is more desirable for an efficient stream cipher generator.
- The cryptographic quality of each regenerated primitive signals is strongly related to Boolean Functions and linear feedback shift registers filtered by the primitive polynomials of degree eight integrated in our system.

C. Impact of Minimal Perturbations

After introducing an analytical study of the internal states of our system, in this subsection, we concentrate to the behavioral study of external states Keystream of our system. The benefit is to interpret the responses of our proposed system in the minimal conditions. Objectively, for each iterations, we choose the secret-keys the same length $K_S = (z_i,...,z_N)$ where z_i are a positive integer in an interval [2,…,50], N equal to 6, the first secret-key is $K_S = (2,2,2,2,2,2)$ and the last secret-key is $K_S = (50,50,50,50,50,50)$. Also, we perform the minimal perturbations on the input secret-key in order to examine their impact on the lengths and the nature of primitive signals of the associated keystreams. We increment, in each iteration, an integer number z_i of input secret-key in a given position progressively. The importance is to show if the linearity of input secret keys has an effect on the cryptographic quality of output secret-keys.

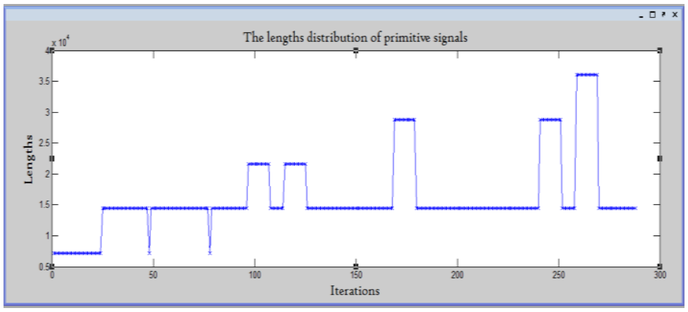


Fig. 5. The lengths distribution of primitive signals

From this histogram (Fig.5), we observe, for the minimal perturbations, that the lengths distribution of primitive signals does not admit a probabilistic law. That means, it hard to an attack to infer the input length according to the lengths of output secret-keys. Its period represents an important benefit to distinguish a good stream cipher regenerator. This dynamite confirms another robustness factor of our regenerator of binary sequences.

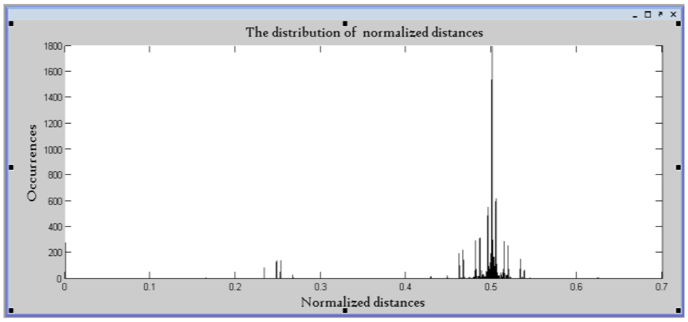


Fig. 6. The distribution of normalized distances

In this histogram (Fig.6), it appears clearly the accumulation of normalized distances nearest to 0.5 followed by small peaks and a large peak exactly in 0.5. This result of normalized distances reassures another significant property filled by our proposed system: unpredictable of each binary sequence. Therefore, we confirm the uncorrelation of generated primitive signals able to withstand the collision and correlation attacks [5][8] [9][10][14][18] [19] [21].

D. Simulating a human system

In reality, Man has a chaotic mind. It is hard to control an user during the choice its input secret-key K_s. But, we can - simulate a human system for regenerate the inputs secret-keys the same length (N=6). For this work, we adapt a Rand function in order to product the integer numbers z_i in interval [1,…, 200] randomly. The aim, in this emulation, is to study the dynamic nature and the cryptographic quality of regenerated primitive signals in the real situations.

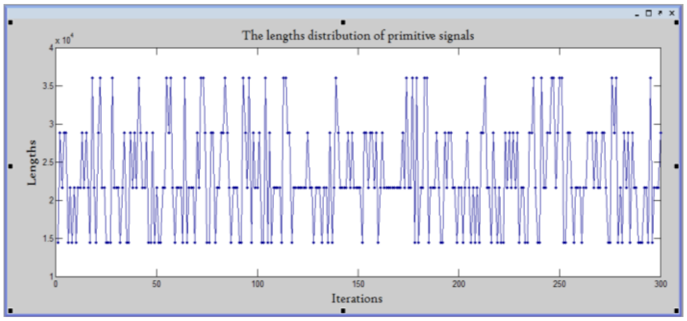


Fig. 7. The lengths distribution of primitive signals

This dynamite (Fig.7) reconfirms the random nature of the lengths distribution of regenerated primitive signals for the inputs secret-keys of same length. It is random and unpredictable over time. This result is highly dependent on calculated positive integers d_i such that $d_i = z_i \bmod 2^m$ with $m \in \mathbb{N}$ and $m \geq 8$. More specifically, it depends on the quadratic structure (square-free integer or integer with square factor) of the calculated positive integers d_i. Because, the binary representations of positive integers d_i, n_i and m_i, have an impact on the balancing results. Wherefore, our system inspires its robustness.

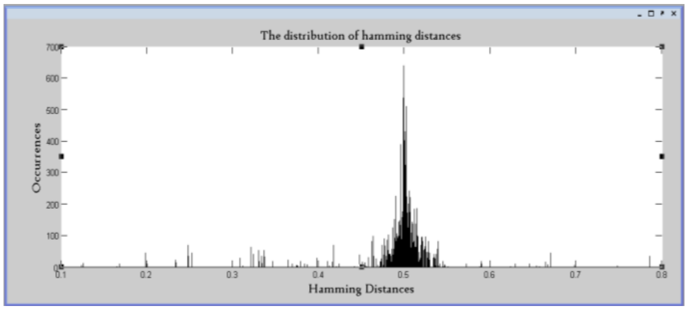


Fig. 8. The distribution of normalized distances

This outcome (Fig.8) is identical to the result obtained in figure 6. It proves, in the minimal conditions, the cryptographic nature of SSCQF algorithm [21]. In effect, our algorithm is efficient and able to resist against attack periodic sequences [5][10]. Likewise, the keystreams are cryptographically strong. This stream ciphers design generate the keystream digits pseudo-randomly from smaller inputs secret-keys without lessening security. They are also able to withstand against to correlation, collision and exhaustive search attacks on stream ciphers [3][4][7][8][9][14][15][18][19][21]. We aim, by this work, to evolve and improve at the symmetric-key encryption scheme.

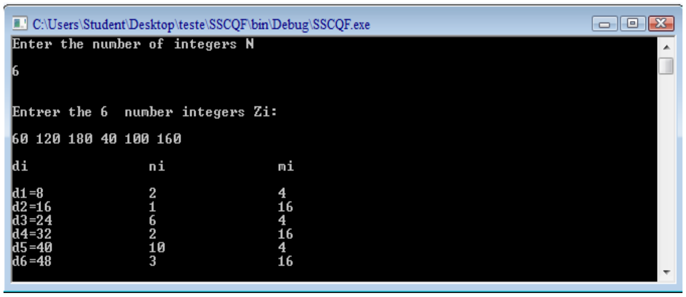
VI. Implementation

This SSCQF regenerator of binary sequences can be executed in different types of symmetric cryptosystem. We aim, in this work, to evolve the cryptographic quality secret-keys against various types of attacks [3][4][7][9][10][14][18] [19]. Thus, according to behavioral study, this property of the primitive signals regenerated is assured. In this section, we itemize practically different execution stages of our proposed system.

A. Implementation of process I

The first aim of this process is to generate the integer numbers d_i, n_i and m_i for each element z_i of a secret-key

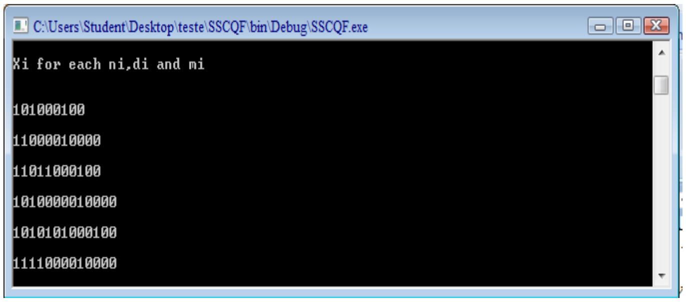
K_s, then, their binary representations. In each iteration, the binary representations of d_i, n_i and m_i will be combined in order to create an initial vector as follows $X_i = \overline{n_i}^2 \| \overline{d_i}^2 \| \overline{m_i}^2$.



```
Enter the number of integers N

6


Entrer the 6  number integers Zi:

60 120 180 40 100 160

di                ni               mi

d1=8              2                4
d2=16             1                16
d3=24             6                4
d4=32             2                16
d5=40             10               4
d6=48             3                16
```

Fig. 9. Regeneration of the d_i, n_i and m_i for a secret-key

From this figure (Fig.9), we show that the values of n_i and m_i don't depend on the values of d_i, but, these are strongly related to its quadratic structure. In reality, it gives more complexity and dynamite of our proposed system. It suffices to behold here that any added bit has an impact on the balancing results of initial binary vectors X_i.



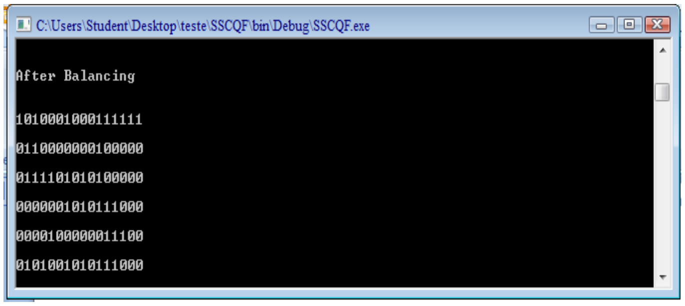
```
Xi for each ni,di and mi


101000100

11000010000

11011000100

1010000010000

1010101000100

1111000010000
```

Fig. 10. Binary representation of each initial vector X_i

From this outcome (Fig.10), the binary representations of each initial vector X_i don't have the same length. But, in our proposal, we want to get the binary sequences which have the same length L divisible by eight. This is the object of the following process.

B. *Implementation of process II*

As we have previously explained, we dedicate this process to balancing the binary sequences generated in previous process. The aim is to obtain initial binary vectors X_i that have a length multiple to eight. Because, in these situation, we use a linear feedback shift register filtered by the primitive polynomial of degree eight. So, if we change the degree of primitive polynomial, in this case, we should adapt this process to regenerate the initial vectors that have a length of its degree. The results of this process are presented in following figure (Fig.11).

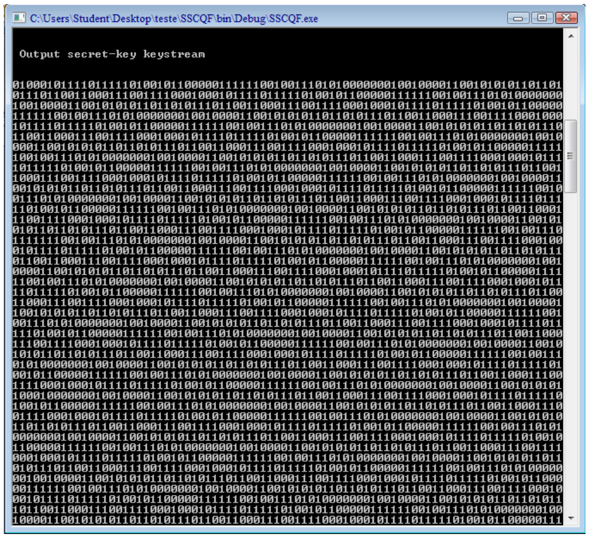


```

After Balancing


1010001000111111

0110000000100000

0111101010100000

0000001010111000

0000100000011100

0101001010111000
```

Fig. 11. Balancing of each X_i

C. *Implementation of process III*

After balancing each initial vectors comes this important process. We implement this process for create an output Keystream digit specific to each input secret-key K_s. In first time, for each Y_i, we construct $L/8$ linear feedback shift registers, namely, $\mathrm{LFSR}_{i1}, \ldots, \mathrm{LFSR}_{iL/8}$. Then, we exercise the Boolean functions R_j on all LFSR_{ij} as defined in process III. The output keystream is obtained by concatenation of the output binary sequences of all Boolean functions. This following figure presents an embodiment of this process (Fig.12).

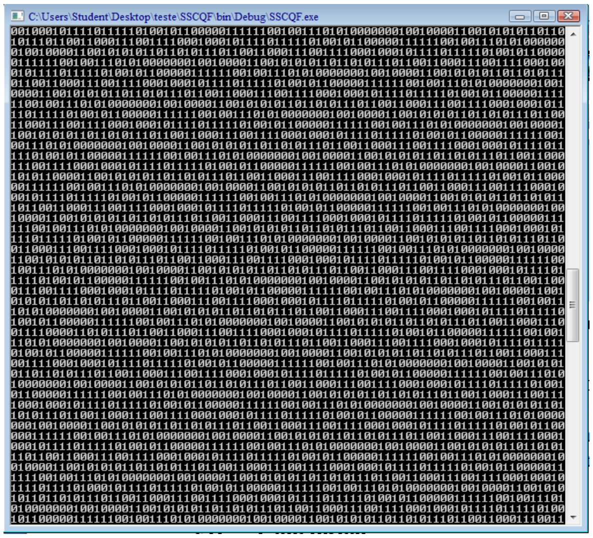


```
C:\Users\Student\Desktop\teste\SSCQF\bin\Debug\SSCQF.exe

 Output secret-key keystream

[illegible]
```

Binary sequence (1)

```
C:\Users\Student\Desktop\teste\SSCQF\bin\Debug\SSCQF.exe
[illegible]
```

Binary sequence (2)



```
C:\Users\Student\Desktop\teste\SSCQF\bin\Debug\SSCQF.exe
[illegible]
```

Binary sequence (3)

```
C:\Users\Student\Desktop\teste\SSCQF\bin\Debug\SSCQF.exe
[illegible]
```

Binary sequence (4)

Fig. 12. Binary sequence of Keystream digit

Note: The Keystream is obtained by concatenation of all binary sequences (1, 2, 3, 4).

In this work, we innovate a quick, dynamic and complex generator of the binary sequences. We are combined a large theory concept for product a pseudorandom stream cipher. It will be used as a symmetric key cipher for avoid the serious security problems. This synchronous generator products primitive signals uncorrelated, unpredictable and independents of the same input secret-keys lengths. Moreover, it ensures the cryptographic quality of internals states in order to avoid correlation attacks [9][14][18] [19].

VII. Conclusion

We introduced, in this paper, a new synchronous stream generator cipher named SSCQF. Our proposed symmetric key system is founded on quadratic fields. We aim by this work to improve the confidentiality of transmitted data between two communicated pairs. A behavioral study, in the minimal conditions, appears the cryptographic nature of our construction. It also confirms the concrete security of the internal and external states, more, its ability to conserve the unpredictable nature of each regenerated primitive signals. In addition, the output secret-key length is not related to the input secret-key length, but, is strongly linked to quadratic nature of each element constructing an input secret-key. Idem, these dynamite and robustness are clearly proved in implementation section.

Reference

[1] A. Sabour, A. Asimi, and A. Lbekkouri, “The three states functions: Theoretical foundations and estimated complexity”, in The 3rd International Conference on Information Technology, pp. 1{9, 2007}.

[2] Cassels. J.W.S and Frohlich. A, “Algebraic number theory”, Academic Press, 1967.

[3] Courtois N. and Meier W.: “Algebraic attacks on stream ciphers with linear feedback”, Advances in Cryptology Eurocrypt 2003, LNCS 2656, Springer-Verlag, pp. 345-359, 2003.

[4] Courtois N.: “Algebraic Attacks on Combiners with Memory and Several Out-puts”,ICISC 2004, LNCS 3506, pp. 320, 2005.

[5] E.R. Berlekamp. “Algebraic coding theory”. McGraw-Hill, 1967.

[6] Golomb. S.W, “Shift register sequences”, revised edition, Aegean park press, laguna hills, California, 1982.

[7] Hawkes P. and Rose G.: “Rewriting variables: the complexity of fast algebraic attacks on stream ciphers”, Advances in Cryptology Crypto 2004, LNCS 3152, SpringerVerlag, pp.390-406, 2004.

[8] J. D. Golic. “Cryptanalysis of Alleged A5 Stream Cipher”. In Advances in Cryptology { Eurocrypt'97, LNCS 1233, pp. 239-255, Springer-Verlag, 1997.

[9] J. D. Golic. \Towards Fast Correlation Attacks on Irregularly Clocked Shift Registers." In Advances in Cryptography { Eurocrypt'95, pp. 248-262, Springer-Verlag,1995.

[10] J.L. Massey. “Shift-register synthesis and BCH decoding”. IEEE Transactions on Information Theory, vol. 15, pp. 122–127, 1969.

[11] Lewis. T.G and Payne, W,H, “Generalized feedback shift register pseudo-random number algorithms”, Journal of the ACM, 20: 456-468, 1973.

[12] Lidl. R and Neiderreiter. H, Introduction to finite fields and their applications, Cambridge University Press: Cambridge, London, New York, 1968.

[13] Menezes A.J., Oorschot P.C., Vanstone S.A.: “Handbook of Applied Cryptography”, Chapter 5: Pseudorandom Bits and Sequences, CRC Press,1996.

[14] Menezes A.J., Oorschot P.C., Vanstone S.A: Handbook of Applied Cryptography, Chapter 6: Stream Ciphers, CRC Press, 1996.

[15] S. Babbage. “A Space/Time Tradeoff in Exhaustive Search Attacks on Stream Ciphers”. European Convention on Security and Detection, IEE Conference publication, No. 408, May 1995.

[16] Samuel. P, “Théorie algébrique des nombres”, Hermann, Paris 1971.

[17] Tausworthe. R. C, Random numbers generated by linear recurrence modulo two, Mathematics of Computation, 19: 201-209, 1965.

[18] V.V. Chepyzhov, T. Johansson and B. Smeets. “A Simple Algorithm for Fast Correlation Attacks on Stream Ciphers”. In Fast Software Encryption { FSE 2000, LNCS 1978, pp. 181-195, Springer-Verlag, 2000.

[19] W. Meier and O. Staffelbach. “Fast Correlation Attacks on Certain Stream Ciphers”. Journal of Cryptography, 1(3):159-176, 1989.

[20] Walker. E. A, “Nonlinear recursive sequences can”. J. Math 11, 370-378, 1959.

[21] Younes Asimi, Abdallah Amghar, Ahmed Asimi, and Yassine Sadqi, “New Random Generator of a Safe Cryptographic Salt Per Session”, International Journal of Network Security, Vol.18, No.3, PP.445-453, May 2016.