ISSN: 0067-2904

Construct a New System as a Combining Function for the LFSR in the Stream Cipher Systems Using Multiplicative Cyclic Group

Mithaq Abdulkareem Abdulwahed[1*], Ayad G. Naser Al-Shammari [2]
[1]Department of Mathematics, Collage of Science, University of Baghdad, Baghdad, Iraq
[2]Directorate General for Vocational Education Ministry of Education, Baghdad, Iraq

Abstract

In this paper, we construct a new mathematical system as Multiplicative Cyclic Group (MCG), called a New Digital Algebraic Generator (NDAG) Unit, which would generate digital sequences with good statistical properties. This new Unit can be considered as a new basic unit of stream ciphers.

A (NDAG) system can be constructed from collection of (NDAG) units using a Boolean function as a combining function of the system. This system could be used in cryptography as like as Linear Feedback Shift Register (LFSR) unit. This unit is basic component of a stream cipher system.

Keywords: Cryptography, LFSR, Stream Cipher, Multiplicative Cyclic Group.

تصميم نظام جديد كدالة مركبة للمسجلات الزاحفة في أنظمة التشفير الانسيابي باستخدام الزمرة الضربية الدوارة

ميثاق عبد الكريم عبد الواحد*[1]، اياد غازي ناصر الشمري[2]
[1]قسم الرياضيات، كلية العلوم، جامعة بغداد، بغداد، العراق
[2]وزارة التربية، المديرية العامة للتعليم المهني، بغداد، العراق

الخلاصة:

في هذا البحث، تم تقديم بناء نظام رياضي جديد باستخدام الزمرة الضربية الدوارة Multiplicative Cyclic Group (MCG) سمي هذه النظام بوحدة المولد الرقمي الجبري الجديد (NDAG) التي من شأنها توليد متتابعات رقمية لها خواص احصائية جيدة. يمكن اعتبار هذا النظام الجديد كوحدة أساسية في نظم التشفير الانسيابي.

تم إنشاء نظام (NDAG) من مجموعة من وحدات (NDAG) باستخدام الدالة المنطقية كدالة مركبة للنظام. يمكن استخدام هذا النظام في التشفير كما في المسجل الزاحف ذات التغذية المرجعية (LFSR). والذي يعتبر الجزء الاساس في التشفير الانسيابي.

1. Introduction

Cryptography is the study of mathematical techniques which are related to the aspects of information security such as confidentiality, data integrity, entity authentication, and data origin authentication[1].

Number theory, in mathematics, is primarily the theory of the properties of integers (whole numbers) such as parity, primarily, and multiplicatively, etc.

*Email: methaq90alheety@gmail.com

There are many different types of secret key cryptosystems, like monographic (character) ciphers, block ciphers, exponentiation ciphers and stream ciphers in which we shall focus[2].

Jennings Generator scheme uses a multiplexer to combine two LFSR's [3]. The multiplexer, controlled by LFSR-1, selects 1 bit of LFSR-2 for each output bit. There is also a function that maps the output of LFSR-2 to the input of the multiplexer. The key is the initial state of the two LFSR's and the mapping function. Although this generator has great statistical properties, it fell to Ross Anderson's meet-in-the middle consistency attack[4].

Mitchell presents a random number (0..9) generator, for use in generating cryptographic keystreams, which based on the division algorithm. This generator permits choice of seed values giving a long cycle length that is known a priori and costlessly decimal keystream periodicity. The keystream can be used in stream cipher system [5].

The stream cipher Grain was developed by M. Hell, T. Johansson, and was especially designed for being very small and fast in hardware implementation. It uses the key of length 80 bits and the IV is 64 bits, its internal state is of size 160 bits [6].

At FSE 2004, a new stream cipher called VMPC was proposed by Bartosz Zoltak, which appeared to be a modification of the RC4 stream cipher. In cryptanalysis, a linear distinguishing attack is one of the most common attacks on stream ciphers. In the paper it was claimed that VMPC is designed especially to resist distinguishing attacks[7].

Recently, a new European project eSTREAM has started, and at the first stage of the project 35 new proposals were received by May 2005. Although many previous stream ciphers were broken, collected cryptanalysis experience allowed strengthening new proposals significantly, and there are many of them that are strong against different kinds of attacks. One such good proposal was the new stream cipher Grain [8].

Dragon is a word oriented stream cipher submitted to the eSTREAM project, designed by Ed Dawson et al. It is a word oriented stream cipher that operates on key sizes of 128 and 256 bits [9].

Perez Ramirez et al. presents an algorithm to reduce the Multiplicative Computational Complexity (MCC) in the creation of digital holograms, where an object is considered as a set of point sources using mathematical symmetry properties of both the core in the Fresnel integral and the image. The image is modeled using group theory [10].

Theorem (1) [11]: (the fundamental theorem of arithmetic)

Any positive integer n>1 can be written uniquely in the following prime factorization form:

$$n= p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k} = \prod_{i=1}^{k} p_i^{\alpha_i} \tag{1}$$

where $p_1<p_2<\ldots<p_k$ are primes, and $\alpha_1, \alpha_2, \ldots, \alpha_k$ are Nonnegative integers.

Remark(1):

1. The function *f* has the property of being "one-to-one" (or "injective") if no two elements in Domain are mapped into the same element in Range.
2. The function *f* has the property of being "onto" (or "surjective") if the range R of *f* is all of B (R=B).

Definition (1)[12]: Let (R,+,•) be a ring with identity element, if the Idempotency law be satisfied $a^2=a$, $\forall a\in R$, then the ring is called Boolean ring.

Example (1): Let P(X) represents the set of all the subsets of the universal set X, then the ring (P(X),⊕,•) is Boolean ring.

Definition (2): The Boolean algebra is the mathematical system $(B,\vee,\wedge)$ where $B\neq\varphi$, and the binary operations $\vee$ and $\wedge$ are defined on B as follows:

1. The operations $\vee$ and $\wedge$ are commutative.
2. The operations $\vee$ and $\wedge$ satisfy the distribution law for each other.
3. $\exists$ two identity distinct elements 0 and 1 of the operations $\vee$ and $\wedge$ respectively s.t. $a\vee 0=a$ and $a\wedge 1=a$, $\forall a\in B$.

Definition(3): In Boolean ring (B,⊕,•), we defined:

1. Complement: $a=a\oplus\bar{1}$, $\forall a\in B$.
2. Sum (OR): $a+b=a\oplus b\oplus a.b$ $\forall a,b\in B$.

Example (2): The system (P(X),⋃,⋂) is boolean algebra, X≠φ, we use φ=0 and X=1. If B be a set of subsets of X including φ and X which is closed on ⋃ and complement then (B,⋃,⋂) is boolean algebra too.

Definition (4)[13]: Any non-empty set G with two operations (+) addition and multiplication (*) called Field if satisfying the following conditions:

1. (G,+) is Abelian group.
2. (G,*) is Abelian group.
3. (G,+,*) is ring.

Remark (2): A field of order q with q prime power is called Galois field and is denoted by GF(q) or just F_q.

Example (3): The finite field F_4 has elements {0, 1, 2, 3} and is described by the Table-1 addition and multiplication table.

Table 1-The addition and multiplication for F_4.

⊕	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

⊗	1	2	3
1	1	2	3
2	2	0	2
3	3	2	1

Remark(3): for Galois field if n=1 and P=2 we obtain the binary system which contain only two elements zero and one that mean each value can be represent with zero or one.

Example (4): $17 = 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$; So the binary representation of 17 is (1 0 0 0 1).

Definition (5)[14]: **Linear Feedback Shift Register**

The basic component of key generator for stream cipher is feedback shift registers (FSRs) because they are appropriate to hardware implementation and they produce sequence with good statistical properties.

The feedback shift registers Figure-1 these are small circuits containing a number of memory cells each of which contains one bit the set of such cells forma a register in each cycle a certain predefined set of cells are tapped and their value passed through function called the feedback function .the register is then shifted down by one bit .with the output bit of the feedback shift register being the bit that is shifted out the register .the combination of the tapped bits is then fed in to the empty cell at the top of register and if feedback function is linear then it's called Linear Feedback Shift Register and denoted by LFSR .

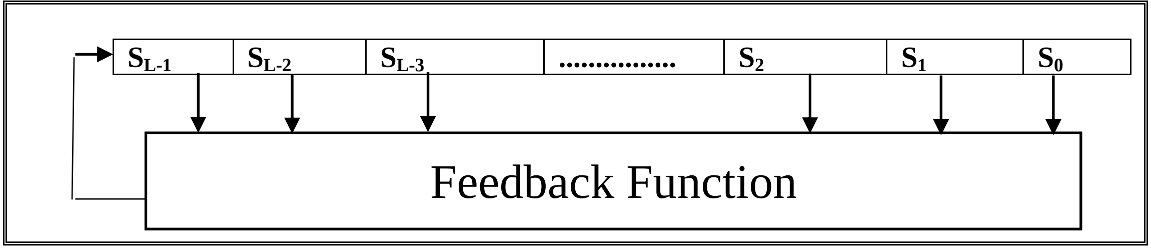

Figure 1-Shift Register Diagram.

Theorem (2): Let ⟨G,*⟩ be a mathematical system, with |G|=q-1, ⟨G,*⟩ is a Multiplicative Cyclic Group (MCG) if and only if q is a prime number.

Definition (6): The element α∈G called a primitive (generator) element iff $\alpha^i=\beta$, where i=1,2,…,q-1, ∀β∈G.

Theorem (٣): If α a primitive element of G on the MCG ⟨G,*⟩, then ∀$\beta=\alpha^i$, s.t. gcd(i,q-1)=1, β is another generator of G.

Remark (٤)[14]:

1. Let us denote the set of primitive elements of the MCG ⟨G,*⟩ by **P(G).**

2. Depending on theorem (٣), then there are $\phi(q-1)$ primitive elements of the MCG $\langle G,*\rangle$. This means $|\mathbf{P}(G)|=\phi(q-1)$.

Algorithm(1)

By using theorem (٣), we can introduce Primitive Elements Generation Algorithm (PEGA) to find all other primitive elements of the MCG $\langle G,*\rangle$ for prime number q from one primitive element α. The steps of this algorithm are as follows:

```
Primitive Element Generation Algorithm (PEGA)
INPUT  : q , α
PROCESS      : i = 2
               REPEAT
                      i = i +1
                      β := f( α , i , q )
                  IF  gcd(i,q-1) = 1 THEN β  is another generator
               UNTIL  i = q-2
OUTPUT       : P (G)
END.
```

Table 2-Number of primitive elements of different MCG $\langle G,*\rangle$

Primes	**| P *(G)*|**	**P *(G)***
7	2	3, 5
11	4	2, 6, 7, 8
13	4	2, 6, 7, 11

2. Generating a New Digital Sequences From MCG Elements

2.1 Generating Digital Sequences From MCG Elements

Let $m\in Z^+$ where $2\le m\le (q-1)/2$. The set $G=\{\beta_1,\beta_2,\ldots,\beta_{q-1}\}$ partitioned into m subsets name N_i, $0\le i\le m-1$which is consists of some ordered elements $\beta_j\in G$, $1\le j\le q-1$. The subsets N_i is the set of all integers β_j s.t. the index j lies between the two real numbers $\frac{q.i}{m}$ and $\frac{q.(i+1)}{m}$

$$N_i=\{\forall\beta_j: \frac{q.i}{m} < \beta_j < \frac{q.(i+1)}{m}\} \qquad (2)$$

Example(6): Let q=13, choose m=3, then i=0,1,2, by using equation (2) then:
For i=0, j=1,2,3,4, then $N_0=\{1,2,3,4\}$. In the same way we get:
$N_1=\{5,6,7,8\}$ and $N_2=\{9,10,11,12\}$.

Theorem (4): The subsets defined in equation (2) are disjoint sets, where i=0,1,2,…,m-1.

Proof: We have to prove that $\bigcap_{i=0}^{m-1} N_i=\Phi$ and $\bigcup_{i=0}^{m-1} N_i=G$.

Let us assume that $\bigcap_{i=0}^{m-1} N_i\ne\Phi$, then $\exists$an integer $x_j\in\bigcap_{i=0}^{m-1} N_i$, this means:

$x_j\in N_0$ where $0<x_j<(1/m).q$,
$x_j\in N_1$ where $(1/m).q<x_j<(2/m).q$, and so on, we have
$x_j\in N_k$ where $(k/m).q<x_j<((k+1)/m).q$ for $0\le k\le m-1$.
Finally,
$x_j\in N_{m-1}$ where $((m-1)/m).q<x_j<q$.
x_j has the same lowerbound and upperbound $\forall N_i$, for $0\le i\le m-1$, then
$\frac{m-1}{m}.q<j<\frac{m-1}{m}.q$, then q<j<q that's C! since $1\le j\le q-1$.

$\therefore$must $\bigcap_{i=0}^{m-1} N_i$ $=\Phi$.

let us assume that $\bigcup_{i=0}^{m-1} N_i \neq G$, since $G \not\subset \bigcup_{i=0}^{m-1} N_i$, then $\bigcup_{i=0}^{m-1} N_i \subset G$.

$\exists x \in G$ but $x \notin \bigcup_{i=0}^{m-1} N_i$, then $x \notin N_0$ and $x \notin N_1$and…and $x \notin N_{m-1}$, then

$x \leq 0$ or $x \geq q$, because of the definition of N_i, then 0 or $q \in G$ C!.

$\therefore$must $\bigcup_{i=0}^{m-1} N_i$ = G. ■

It's clear that q.$\frac{i}{m}$, and q.$\frac{i+1}{m}$ are real numbers.

From equation (2) we get i< $\frac{m}{q}$.β_j<i+1, then

$i \leq s_{ij} \leq i+1$, where $s_{ij}=(m.\ \beta_j)$ div q (3)

The term "**div**" gives the integer part of $\frac{m \cdot \beta_j}{q}$.

if letting $j=\beta_j$, then:
[(i.q) div m]+1≤ j≤((i+1).q) div m.
s_{ij} is the element j of the sequence S_i which is corresponding to the subset N_i depending on equation (3), where

$S_i=\left\{s_{ij}\right\}_{j=[(i.q)\ div\ m]+1}^{((i+1).q)\ div\ m}$.

The finally sequence we want to generate is:

$$S=\bigcup_{i=0}^{m-1} S_i \qquad (4)$$

This sequence is corresponding to $\bigcup_{i=0}^{m-1} N_i$ depending on equation (3).

Remark (5): It's clear that the period P(S)=q-1.
Example (7): Using example (6), when

$\bigcup_{i=0}^{2} N_i$ ={1,2,3,4,5,6,7,8,9,10,11,12}, then S={0,0,0,0,1,1,1,1,2,2,2,2}.

Remark (6): Notice that in equations (3) and (4), and examples (6) and (7) that in general, $\beta_j = j$, for this reason:

G=$\bigcup_{i=0}^{m-1} N_i$ ={1,2,…,q-1} and S={0,0,…,0,1,1,…,1,…,m-1,m-1,…,m-1}.

So the sequence S in example (11) has been generated without using primitive element.
In the next subsections we try to use $\beta_j=f(\alpha,i,q)$, where α is primitive element of the MCG $\langle G,*\rangle$.
Algorithm (2):

Depending on equations (3) and (4), No Primitive Element algorithm (NPEA) can be introduced to find the sequence S without using primitive element.

```
No-Primitive Element Algorithm (NPEA)
INPUT  : q , m
PROCESS       :  j = 0
                  REPEAT
                         j = j +1
                      s_j = (m.j) div q
                 UNTIL  j = q-1
OUTPUT        : the sequence S
END.
```

2.2 Generation of digital sequences (DS) from Single Primitive Elements

In this subsection we want to generate the sequence S by using one primitive element α of the MCG ⟨G,*⟩.

Let $\beta_j=f(\alpha,j,q)$ and $s_j=(m.\beta_j)$ div q then the subsets N_i, $0\le i\le m-1$ has no ordered elements of G and still disjoint subsets.

Notice that $\bigcup_{i=0}^{m-1} N_i$ =G' where |G'|=|G| but has another permutation of elements.

Example (8): Let q=13, choose m=3 and α=2, then
N_0={2,4,8,3}, N_1={6,12,11,9} and N_2={5,10,7,1} then
G' = {2,4,8,3,6,12,11,9,5,10,7,1}
∴ S={0,0,1,0,1,2,2,2,1,2,1,0}.
Table-3 shows five sequences generated from q=13, and α=2, for m=2,3,…,6.

Table 3-five sequences using q=13, α=2 with m=2,…,6.

j	β	m_1=2 S_1	m_2=3 S_2	m_3=4 S_3	m_4=5 S_4	m_5=6 S_5
1	2	0	0	0	0	0
2	4	0	0	1	1	1
3	8	1	1	2	3	3
4	3	0	0	0	1	1
5	6	0	1	1	2	2
6	12	1	2	3	4	5
7	11	1	2	3	4	5
8	9	1	2	2	3	4
9	5	0	1	1	1	2
10	10	1	2	3	3	4
11	7	1	1	2	2	3
12	1	0	0	0	0	0

Remark (7): From the above table we notice that:
1. Each sequence has period q-1=12.
2. All generated sequences have balance frequencies that mean we expect good statistical random properties.

Algorithm (3)

Now we can introduce Single-Primitive Element algorithm (SPEA) to generate sequence with good randomness properties, generated from one primitive element α of the MCG ⟨G,*⟩.

```
Single-Primitive Element Algorithm (SPEA)
INPUT        : q , α , m
PROCESS      :  j = 0
                 REPEAT
                       j = j +1
                       β = f( α , j , q )
                    sj = (m.β) div q
                UNTIL  j = q-1
OUTPUT       : the sequence S
END.
```

The DS generated from the SPE algorithm can be called Single Primitive Element Sequence (SPES).

Remark (8): From table (3), we can notice that the period of S, theoretically, is q-1, but analytically, its (q-1)/2. The reason is described and proved in the next theorem.

Theorem (5): Let S be a SPES, if $\beta_j+\beta_k=q$, where j=1,…,(q-1)/2 and k=j+(q-1)/2 then $s_j+s_k=m-1$.

Proof: since $s_j=(m.\beta_j)$ div q, then

$s_j+s_k= (m.\beta_j)$ div q + $(m.\beta_k)$ div q

$= (m.\beta_j - r_j)/q + (m.\beta_k - r_k)/q = [m.(\beta_j+\beta_k) - (r_j+r_k)]/q$

Since both β_j and $\beta_k<q$, then β_j mod $q=r_j$ so $\beta_j = r_j$, for the same reason $\beta_k = r_k$, then $(r_j+r_k)/q=1$.

$s_j+s_k= [m.(\beta_j+\beta_k)]$ div q – 1 = (m.q) div q – 1 = m – 1. ■

From the above theorem we can deduce the second half of the sequence S from the first half.

Example (9): let i=1 and j=7, so $\beta_1=2$ and $\beta_7=11$, for m=4, then $s_1=0$ and $s_7=3$.

We notice the following disadvantages:

1. The periodicity decreased from q-1 to (q-1)/2.
2. The general complexity of S decreases to (q-1)/2.

In this manner we want to construct a method to maximize the complexity and the periodicity to be q-1. This maximization must not effect the good randomness of S.

2.3 Generation of DS from Double Primitive Elements

In this subsection we want to generate the sequence S by using double primitive elements of the MCG ⟨G,*⟩.

Definition (7): Let α_1 and α_2 be two primitive elements of the MCG ⟨G,*⟩, if $\alpha=f(\alpha_1,j,q)$ and $\beta=f(\alpha_2,\alpha,q)$ s.t. $1\leq\alpha,j\leq q-1$, then

$$\beta=f(\alpha_2,\alpha,q) = f(\alpha_2, f(\alpha_1,j,q),q) = g(\alpha_1,\alpha_2,j,q) = \alpha_2^{\alpha_1^j}$$

where $g=f\circ f$ and g:G→G.

Remark (9): It is clear that $\beta\in G$, since f is 1-1 function, and α_1 and α_2 are primitive elements of the MCG ⟨G,*⟩. To guarantee that all elements of G can be generated we have to prove that the function g is 1-1 function.

Theorem (6): If the function f is 1-1 then the function g is 1-1.

Proof: we have to prove that $j\neq j'$ if and only if $\beta\neq\beta'$.

Let $j\neq j'$ then $\alpha\neq\alpha'$ since f is 1-1 function, then $\beta\neq\beta'$.

Let $\beta\neq\beta'$ then $\alpha\neq\alpha'$ since f is 1-1 function, then $j\neq j'$. ■

Example (10): Let q=13, choose m=3, $\alpha_1 = 2$ and $\alpha_2=6$ are two primitive elements, then:

N_0={10,9,3,8}, N_1={12,1,11,5} and N_2={2,4,7,6} so:

G' = {10,9,3,8,12,1,11,5,2,4,7,6} then S={2,2,0,2,1,0,2,1,0,0,1,1}

Table-4 shows five sequences generated from q=13, $\alpha_1=2$ and $\alpha_2=6$ for m=2, 3,…,6.

Table 4-MCG sequences from q=13, α_1 =2 and α_2=6 with m=2,…,6.

J	β	$m_1=2$ S_1	$m_2=3$ S_2	$m_3=4$ S_3	$m_4=5$ S_4	$m_5=6$ S_5
1	10	1	2	3	3	4
2	9	1	2	2	3	4
3	3	0	0	0	1	1
4	8	1	2	3	4	3
5	12	1	1	2	3	5
6	1	0	0	0	0	0
7	11	1	2	3	4	5
8	5	0	1	1	1	2
9	2	0	0	0	0	0
10	4	0	0	1	1	1
11	7	1	1	2	2	3
12	6	0	1	1	2	2

Frequency	**0**	**6**	**4**	**3**	**2**	**2**
	1	6	4	3	3	2
	2	--	4	3	2	2
	3	--	--	3	3	2
	4	--	--	--	2	2
	5	--	--	--	--	2

We noticed that the generated sequences have balance frequencies for different digits, this happened since G divided into m subsets N_i with approximate equal orders. This means we expect that S has good statistical properties.

Algorithm (4)

The Double-Primitive Elements Algorithm (PDEA) can be introduced to generate a new sequence has good randomness properties generated from two primitive elements α_1 and α_2 of the MCG $\langle G,*\rangle$.

Double-Primitive Element Algorithm (DPE)

```
INPUT       : $q, \alpha_1, \alpha_2, m$
PROCESS     : $j = 0$
                REPEAT
                    $j = j + 1$
                    $\beta = g(\alpha_1, \alpha_2, j, q)$
                  $s_j = (m.\beta)$ div $q$
               UNTIL $j = q-1$
OUTPUT      : the sequence S
END.
```

4. (4.5) Encoding System

Before we deal with our proposed generator, we have to know that the output sequence which is the encryption key (K), must be combined with encoding plaintext (P) by using Encryption (E) function (or process) to gain the Ciphertext (C).

$C = E(K,P)$ (5)

It is important to mention that, the plaintext characters must be encoded by some suitable number system.

If E is linear (additive) operation, then relation (5) can be as follows:

$C = K + P \pmod m$ (6)

where $E:\{0,..,m-1\}\rightarrow\{0,..,m-1\}$.

So, as we see in relation (6), the encoding plaintext and the key must be from the same number system (m).

The inverse of function E is the Decryption (D) function (or process), where $D=E^{-1}$ and $D:\{0,..,m-1\}\rightarrow\{0,..,m-1\}$, so:

$P = D(K,C)$ (7)

As E defined in relation (6), the function D in equation (7) will be:

$P = C - K \pmod m$ (8)

Example (11): Let m=4, then E and D are Linear functions, then we can construct the following encryption and decryption tables:

Encryption Table

		P			
		0	**1**	**2**	**3**
K	**0**	0	1	2	3
	1	1	2	3	0
	2	2	3	0	1
	3	3	0	1	2

Decryption Table

		C			
		0	**1**	**2**	**3**
K	**0**	0	1	2	3
	1	3	0	1	2
	2	2	3	0	1
	3	1	2	3	0

In this manner we can introduce three examples of encoding systems that can be used with the suggested generator.

First: m=27, if we use the English language (26 letters with "space" character), then #"A"=0, #"B"=1,…,#"Z" =25 and #"space"=26. We can use relation (6) and (8) as encipher and decipher operations respectively.

Second: As alternative encoding system, we can use the binary number system (m=2), first we have to translate the plaintext characters to the corresponding binary using 5 per character (or 5..8 bit per character, this is related to the size of language alphabetic), so relation (6) and (8) will be: C = K XOR P and P = K XOR C respectively.

Third: Here we can use the ASCII code which is used in computer, as en example, this means m=256 [15].

Remark(10):

1. When we use some encoding system, we must take care of choosing the prime q, the condition of choosing that the lower bound is $q \geq 2m+1$.
2. The encoding system (specified the variable m) can be treated as fixed secret or public variable. In our proposed generator we can use m as an optional key variable specified by the user or as a part from the initial key.

3. Design PRG for DS Using MCG

The function $g(\alpha1,\alpha2,i,q)$ and PDEA can represent the smallest Pseudo Random Generator (PRG) to generate a digital sequence has good statistical properties.

In order to get the sequence S we have fed the generator by which we called seed or initial key. The length, size and the variables of the initial key depend on how we combined the generator. In the next subsection we will introduce the New Digital Algebraic Generator (NDAG) unit and NDAGsystem.

3.1 Designing of NDAGUnit

The mentioned simple PRG can be treated as a single unit; we called it a New Digital Algebraic Generator Unit (NDAGU), where the function $g(\alpha_1,\alpha_2,i,q)$ represents the heart of this generator. Now we have to show how the secret initial key can be specified?

1. Choose the prime number q as the 1st variable of the initial key, and it does prefer to be as large as possible.

2. We can choose any two primitive elements α_1 and $\alpha_2 \in$ **P** (G), these two elements can be considered as the 2nd and 3rd initial key variables.

3. We noticed in Tables-(3) and (4), that the index j always starts from j=1 and ends with q-1, so we can choose another start value, name k, where $1 \leq k \leq q-1$ then k can be treated as the 4th variable (initial key) s.t. we suggest using another index say i which start from k and it will be cyclic value.

4. As we mentioned before, the variable m can be treated as one of the initial key variables, so it considered the 5th variable.

If we consider that NDAGU as a function of five variables (seeds), then S=NDAGU (q, α_1, α_2, k, m), this function can be viewed as the NDAGU algorithm.

Algorithm(5)

The NDAGU algorithm can be introduced to generate a good statistical random properties sequence, with initial key variables q , α_1 , α_2 , k and m with length $L \leq q-1$.

```
NDAGU Algorithm
INPUT  : q , $\alpha_1$ , $\alpha_2$ , k , m , L
PROCESS        :  i = k-1 ,  j = 0
                  REPEAT
                        i = i (mod(q-1))+1
                        j = j +1
                        $\beta = g(\alpha_1 ,\alpha_2 , i , q )$
                     $s_j$ = (m.$\beta$) div q
                  UNTIL  j = L
OUTPUT         : the sequence S
END.
```

Example (12): Let us encrypt the message "MAN" using the binary representation (m=2), so #"MAN"= 12 0 13 =1100 0000 1101. To generate encryption key from MCGU we used the initial keys q=13, α_1=2, α_2=6, k=5 , m=2 , L=12.

Plaintext	1	1	0	0	0	0	0	0	1	1	0	1
Key	1	0	1	0	1	0	0	1	0	1	0	1
Cipher	0	1	1	0	1	0	0	1	1	0	0	0

3.2 Designing of NDAG System

A NDAGU can be considered as a basic construction unit in NDAGSystem (NDAGS) with combining boolean function (CF). If S is the sequence that generates from NDAGS, and the system has F_n=CF as combining function with n-NDAGunits, then $S=F_n(S_1,S_2,\ldots,S_n)$ s.t. $S_i=NDAGU_i(q_i,\alpha_{1i},\alpha_{2i},k_i,m)$, where $1\leq i\leq n$.

S_i represents the sequence i generate from the MCG unit i.

We defined the addition (+) and the multiplication (*) operations of the system, as follows:

$s_j = s_{ij} + s_{kj}$ (mod m)

$s_j = s_{ij} * s_{kj}$ (mod m)

where $s_j\in S$, j=1,2,…, and $s_{ij}\in S_i$ and $s_{kj}\in S_k$, $1\leq i,k\leq n$, $i\neq k$.

Let us represent the NDAGU number i by NDAGU(i), where $1\leq i\leq n$.

Algorithm(6)

Now we can introduce the New Digital Algebraic Generator System algorithm (NDAGS) to generate a digital sequence with length L, after feeding the generator of the system which is combined from NDAGU by the initial keys q_i , α_{1i} , α_{2i} ,k_i , m.

```
New Digital Algebraic Generator System algorithm NDAGS
INPUT  : READ n , m , L
                 For i = 1 to n
                        READ $q_i$ , $\alpha_{1i}$ , $\alpha_{2i}$ , $k_i$
                 ENDFOR {i}
PROCESS        :  j = 0 ;
                  REPEAT
                        j = j + 1
                        FOR i = 1 TO n  CALL FDAGU(i)
                        $s_j = F_n(s_{1j},s_{2j},\ldots,s_{nj})$
                 UNTIL j = L
OUTPUT         : the sequence S
END.
```

5. Conclusions and future work

This paper concludes the following aspects and Some of future works are presented to be implemented in the future:

1. The LFSR is considered as the main unit which can be used in stream cipher systems designing, but in this research we introduce a new unit which can be used, side by side to LFSR, as an another basic unit of stream cipher systems designing.
2. This suggested NDAG unit has good randomness statistic properties, high linear complexity, accepted periodicity and high correlation immunity. These efficiency criteria are enough to use the NDAG as cryptosystem.
3. The NDAG candidate to be used in many forms for digital cryptosystems like voice encryption systems.
4. As a future work we have to generalize the other basic criteria of efficiency it is known that be applied on digital sequences.
5. We suggest constructing a hybrid pseudo random generator with good properties by mixing the LFSR unit and NDAG unit.

References

1. Stinson, D. R. **1995.** "*Cryptography: Theory and Practice*" CRC Press.

2. Ayad Ghazi Naser and Fatin Majeed**.** A.H. **2017.** "Constructing of Analysis Mathematical Model for Stream Cipher Cryptosystems". *Iraqi Journal of Science*, **58**(2A): 707-715

3. Jennings, S. M. **1984.** "Autocorrelation Function of the Multiplexed Sequence" *IKE Proceedings*, **131**(2): 169-172.

4. Anderson, R.J. **1990.** "Solving a Class of Stream Ciphers", *Cryptologia*, **14**(3): 285-288.

5. Mitchell, D. W. **1993.** "*A Nonlinear Random Number Generator with Known, Long Cycle Length*", Dept. of Economics, West Virginia University, Morgantown WV 26506-602 USA.

6. Johnson, D. W. and Johnson, F. P. **2002**. "*Joining Together: Group Theory and Group Skills*", Allyn & Bacon, July.

7. Zoltak, B**. 2004.** "*VMPC One-Way Function and Stream Cipher", In B. Roy and W. Meier, editors, Fast Software Encryption* 2004, volume 3017 of Lecture Notes in Computer Science, pages 210–225. Springer-Verlag.

8. eSTREAM: ECRYPT Stream Cipher Project, IST-2002-507932. Available at http://www.ecrypt.eu.org/stream/ (accessed September 29, 2005), 2005.

9. Chen, K. and et al, **2005.** "Dragon: A Fast Word Based Stream Cipher", eSTREAM, ECRYPT Stream Cipher Project, Report 2005/006 (2005-04-29), 2005.

10. Agustín Pérez-Ramírez et al., **2017**. "*Application of Mathematical Symmetrical Group Theory in the Creation Process of Digital Holograms* ", Hindawi, Mathematical Problems in Engineering, Volume 2017, Article ID 5612743, 7 pages

11. Apostol, T. M. **1998.** "*Introduction to Analytic Number Theory*", Corrected 5th Printing, Undergraduate Texts in Mathematics, Springer-Verlag.

12. Whitesitt, J. E**. 1995.** "*Boolean Algebra and its Application*", Dover Publications, April 1995.

13. Ayad G. Naser Al-Shammari, Rusol M. Shaker Alzewary, **2016.** "Design of High Efficiency Non-linear Keys Generator Based on Shift Registers", *Iraqi Journal of Science*, 2016, Special Issue, Part B, pp: 693 – 406

14. Gilbert, W. J. **2002.** "*Modern Algebra with Applications*", Wiley-Interscince, March.

15. Gustafson, H., Dawson, E. Nielsen, L. and Caelli, W. **1998.** "A Computer Package for Measuring the Strength of Encryption Algorithm", Information Security Research Centre at Queensland University of Technology, 1998.