Design and Analysis of a Robust and Efficient Block Cipher using Cellular Automata

Pallavi Joshi[1], Debdeep Mukhopadhyay[2], and Dipanwita RoyChowdhury[3]

[1] B.Tech Student, Department of Computer Science and Engg.
[2] PhD Student, Department of Computer Science and Engg.
[3] Associate Professor, Department of Computer Science and Engg.
Indian Institute of Technology, Kharagpur, India

Abstract. *Cellular Automaton (CA) has been shown to be capable of generating complex and random patterns out of simple rules. There has been constant efforts of applying CA to develop ciphers, but the attempts have not been successful. This paper describes how repeated application of simple CA transforms may be used to achieve confusion and diffusion, needed in block ciphers. The components have been evaluated for their robustness against conventional cryptanalysis and the results have been found to be comparable to standards. Finally, the parts are assembled in an unconventional way to construct a self-invertibe CA based round, which is resistant against linear and differential cryptanalysis and yet can be efficiently implemented.*

Key Words: Block Cipher, Cellular Automata, Cryptanalysis, Cycles Self-invertibility, Simple Rules

1 Introduction

Almost all cryptographic applications depend on the underlying strength of their primitives. Certain basic blocks are repeated in order to build cryptographic algorithms. The requirements on these sub-components or functions are varied and depends on the applications which are built out of them. But there are certain cryptographic properties, like non-linearity, avalanche effect which the primitives must satisfy. For the functions to be used for block ciphers, invertible mappings are necessary. With the advent of electronic commerce and portable devices for communications, cryptographic implementations have become exceedingly important. Hence, it is also imperative for the designers of crypto-algorithms that the designs are amenable to both hardware and software implementations.

Cellular Automaton (CA) was first introduced by von Neumann and later by Wolfram [1] as simple models for physical, biological and computational systems. The fact that the simple underlying rules of the CA can be very efficiently implemented and repeated applications of these simple rules can demonstrate complex behaviors, have lured researchers to develop CA based ciphers. In [2], non-homogenous Cellular Automata, which have different rules for different cells, were proposed for public-key cryptography. But the paper lacks specifications like key-size, key generation procedures and also real life examples. This makes it difficult to perform cryptanalysis of the cipher, thus leaving its security untested. The block ciphers and stream ciphers proposed in [3] were broken in [4] due to the affine property of the used CAs. In [5] another block cipher was proposed but it was unable to get rid of the affine property and thus could not achieve the claimed security. In [6] an extended Cellular Automaton ($GF(2^8)$) was used to develop a cryptosystem which used the Galois Field multiplication as the non-linear step. However the paper also lacked detailed cryptanalysis and the key generation algorithm. The recent CA based block cipher, proposed in [7] mixes an affine CA with non-affine mappings. However the block cipher has been successfully cryptanalyzed in [8].

The reasons behind the failure should not be attributed to the Cellular Automaton, which is on the contrary a wonderful machine conducive for cipher design. In [9] the CA has been revisited and a generalised block cipher round has been composed using a special technique. The elegance of the composition was the fact that the combination of the linear and non-linear part did not disturb the cyclic structure of the linear part. However in order to develop a complete block cipher many details like block sizes (of both the data and the key), number of rounds required and a detailed security analysis have to be performed, which were missing in the previous attempts of designing CA based cryptosystems [8]. In the present work we thus adopt a "tame" approach [10] in building the block cipher. In this approach first cryptographic primitives, with well defined properties are built from simple rules. Then the components are composed using the technique of [9] along with some other new methodologies to build the complete cipher. In this work it is shown for the first time in the literature of Cellular Automata based cipher design how the features of a cipher based on Substitution and Permutation can be derived using CA rules. The paper discusses the construction of blocks imparting adequate non-linearity (through S-Box) and avalanche effect through Diffusion Box (D-Box) to the cipher. The paper explains that varying the number of "cycles" of the non-linear CA, one can obtain S-Boxes of very high resistance against Differential Cryptanalysis. The linear part providing diffusion is implemented using three linear CA such that the Avalanche criterion is satisfied. Further instead of key mixing using the traditional exclusive-or (xor), we perform key mixing using addition modulo 2^n and subtraction modulo 2^n, where n is the block size. Such a key xor helps to foil Linear Cryptanalysis as the bias reduces exponentially fast with the bit position [15]. The CA based round thus constructed has the properties of self-invertiblility and fast-forwardness, thus leading to efficient implementations. Finally, a technique is presented through which the CA based rounds can be composed to build the complete cipher, with the property of self-invertibility retained. Computation of the number of rounds required is currently underway.

The paper is organised as follows: *Section 2* describes some of the preliminaries required in the work. *Section 3* constructs the CA based round and describes the internal blocks. The security analysis of the round of the block cipher is presented in *section 4*. *Section 5* describes the composition of rounds and future scope of work. Finally the work is concluded in *section 6*.

2 Preliminaries

In the current section some of the preliminary concepts used in the work have been stated.

2.1 Cellular Automata

A Cellular Automaton (CA) consists of a number of cells arranged in a regular manner, where the state transitions of each cell depends on the states of its neighbors. The next state of a particular cell is assumed to depend only on itself and on its two neighbors (3-neighborhood dependency). The state q of the i^{th} cell at time $(t+1)$ is denoted as $q_i^{t+1} = g(q_{i-1}^t, q_i^t, q_{i+1}^t)$, where q_i^t denotes the state of the i^{th} cell at time t and g is the next state function called the rule of the automaton[1]. Since g is a function of 3 variables, there are 2^{2^3} or 256 possible next state functions. The decimal equivalent of the output column in the truth table of the function is denoted as the rule number. The next state function for Rule 90 and Rule 150 are as below :

Rule 90 : $q_i^{t+1} = q_{i-1}^t \oplus q_{i+1}^t$

Rule 150 : $q_i^{t+1} = q_{i-1}^t \oplus q_i^t \oplus q_{i+1}^t$

The CA preliminaries where the CA is in GF(2) are described in the book [11]. An outline of it is as follows.

For an n-cell one dimensional CA, it can be shown that the linear operator is an $n \times n$ matrix whose i^{th} row corresponds to the neighborhood relation of the i^{th} cell. The next state of the CA is generated by applying this linear operator on the present state. The operation is simple matrix multiplication, but the addition involved is modulo-2 sum. The matrix is termed as the characteristic matrix of the CA and is denoted by T.

If X_t represents the state of the automaton at the t^{th} instant of time, then the next state, i.e., the state at the $(t+1)^{th}$ time, is given by

$$X_{t+1} = T * X_t$$

that is, $X_{t+p} = T^p * X_t$.

If for a CA all states in the state transition graph lie in some cycle, it is called a group CA; otherwise it is called a non group CA. It has been shown in [11] that for a group CA its T matrix is non-singular, i.e., $det[T] = 1$.

If the characteristic matrix T of a CA is singular , i.e. $det[T] = 0$, then the CA is a non group CA. In a non group CA , all the states are not cyclic.

An n cell $GF(2)$ CA can be characterized by a $n \times n$ characteristic matrix T as follows:

$$T = \begin{cases} 1 \text{ if the next state of the } i^{th} \text{ cell depends on the} \\ \quad \text{present state of the } j^{th} \text{ cell} \\ 0 \text{ otherwise} \end{cases}$$

Fundamental Transformations

The Cellular Automata having certain rules 195, 153 and 51 are known as fundamental transformations. They are also referred to as the complemented CA. They have the following definitions

$x_i(t) = xnor(x_{i-1}(t-1), x_i(t-1))$,

$x_i(t) = xnor(x_i(t-1), x_{i+1}(t-1))$,

$x_i(t) = xnor(x_i(t-1))$

Each of these rules is a group rule and thus the corresponding CA exhibits group-rule properties. It can be observed that such a CA forms equal cycles of even lengths.

2.2 Definitions related to the Security Analysis of a Block Cipher

Definition 1. *Balancedness: The vector space of n tuples of elements from GF(2) is denoted by V_n. Let f be a (Boolean) function from V_n to $GF(2)$. The truth table of f is defined as ($f(\alpha_0)$, $f(\alpha_1)$,...,$f(\alpha_{2^n-1})$), where α_i, $i = 0, 1, ..., 2^n - 1$, denote vectors in V_n. f is said to be balanced if its truth table has an equal number of zeroes and ones.*

Definition 2. *Affine Function: We call $h(x)$=$a_1x_1 \oplus \ldots \oplus a_nx_n \oplus c$ an affine function where $x = (x_1, \ldots, x_n)$ and a_j,$c \in GF(2)$. In particular, h will be called a linear function if $c = 0$.*

Definition 3. *Hamming Weight: The Hamming weight of a vector x, denoted by $W(x)$, is the number of ones in x.*

Definition 4. *Hamming Distance: Let f and g be functions on V_n. Then $d(f,g)$ $=\sum_{f(x)\neq g(x)} 1$, where the addition is over the reals, is called the Hamming distance between f and g.*

Definition 5. *Non-linearity: Let $\psi_0, \ldots \psi_{2^{n+1}-1}$ be the affine functions on V_n. Then $N_f = min_{i=0,\ldots,2^{n+1}-1}\ d(f,\psi_i)$ is called the non-linearity of f. It is well-known that the non-linearity of f on V_n satisfies $N_f \leq 2^{n-1}$-$2^{n/2-1}$, when n is even.*

Definition 6. *Bias of linear approximation: Let the linear approximation be of the form:*

$< X_{i1} \oplus X_{i2} \oplus \ldots X_{iu} > \oplus < Y_{j1} \oplus Y_{j2} \ldots \oplus Y_{jv} >= 0$

where X_i represents the i-th bit of the input $X = [X_1, X_2, \ldots]$ and Y_j represents the j-th bit of the output $Y = [Y_1, Y_2, \ldots]$. This equation is representing the exclusive OR of u input bits and v output bits.

If the bits are chosen randomly then the above approximated linear expression will hold with probability $1/2$. If p_l is the probability with which the expression holds then the bias is defined as $|p_l - 1/2|$.

Definition 7. *Robustness of S-Box [12]: Let F=(f_1,...,f_s) be an $n \times s$ S-box, where f_i is a function on V_n, $i = 1, \ldots, s$, and $n \geq s$. We denote by L the largest value in the difference distribution table of F, and by R the number of non-zero entries in the first column of the table. In either case the value 2^n in the first row is not counted. Then we say that F is ϵ-robust against differential cryptanalysis, where ϵ is defined by*

$$\epsilon = (1 - R/2^n)(1 - L/2^n) \qquad (1)$$

3 Construction of a round using CA Transforms

In this section a round of the CA based block cipher is presented. As depicted in figure 1 an input to the round, denoted by X is transformed by the round to result in the output Y. The round key is denoted by K and the size of the block of data and key by n. It is suggested that the value of n is 128 for adequate security margin, though the algorithm is scalable. The composition is based on the technique proposed in [9].

The round of the block cipher, denoted by r, has essentially three parts:

- A non-linear, invertible transform f
- A linear part denoted by c (Diffusion Box)
- The inverse non-linear transform f^{-1}

The composition of the round of the cipher is expressed as $r = f^{-1} \circ c^i \circ f$. One of the interesting properties of the construction is that the structure can be programmed easily to perform both encryption and decryption. Encryption is achieved when $i = n-1$, while the round performs decryption when $i = n+1$. When the round performs encryption we denote the round by E_K and when it performs decryption by E_K^{-1}. Apart from having self-invertibility, the non-linear components do not disturb the cyclic nature of the linear CA. The result is easily derived from the following theorem.

Theorem 1. *[9] The cycle structure of any transformation T_1 is the same as that of $T_2 = f^{-1} \circ T_1 \circ f$.*

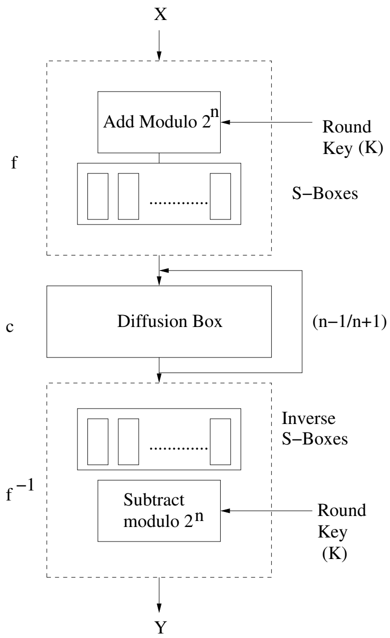

Fig. 1. The CA based round

Thus, the cyclic structure of the transformation $T_1 = c^i$ is the same as that of $T_2 = r = f^{-1} \circ c^i \circ f$.

Further, the round has a fast forwardness property [9], resulting in high speed ciphers. Finally the non-linear part f has two sub-components:first, a key mixing step performed through addition modulo 2^n with the round key K and then a non-linear S-Box. Likewise the part f^{-1} has two components:first the inverse S-Box and then the inverse key mixing step obtained through subtraction modulo 2^n with the round key K.

Next, we describe the individual steps of the round and explain how confusion and diffusion are achieved through the CA based transforms.

3.1 Construction of the Diffusion Box

The Diffusion Box (D-Box) comprises of linear transformations, aimed at providing diffusion to the cipher. The crux of the D-Box is a complemented Cellular Automata characterized in [13] and generalised in [14]. For example when a Cellular Automaton with rule 153 [11] is fed with an initial seed of X, it produces an output $\overline{T}(X) = T(X) + IF$. Here T is the transformation matrix which has the ith and the $(i+1)$th elements of the ith row as 1 with the exception of the last row where it has only its last element as 1. I is the identity matrix and F is the vector with all 1's. Also, the length of a cycle of an n-cell CA implemented with rule 153 can be characterized in the following theorem :

Theorem 2. *The length of cycle for an n-cell CA, having rule $\overline{T}$, is*

$$l = 2^{\lfloor logn \rfloor + 1} \text{ , } n \geq 2 \tag{2}$$

The cyclic property of the complemented CA leads to the fact that the same transformation can be used to both encrypt and decrypt. For, example if the length of the cycle is l, then we have the identity $I = \overline{T}^l$. Thus, if the D-Box is characterized by the transformation $\overline{T}^{l/2}$ (i,e $D = \overline{T}^{l/2}$) the same transformation performs both encryption and decryption.

Inspite of its cyclic property, the CA cannot be as it is applied to contsruct the D-Box. A serious weakness in the linear transformation is that at each step of the application of rule 153, the last bit of the input simply gets toggled. Indeed, as we have $l/2$ even, as is evident from Theorem 2, the last bit of the output will be the same as the last bit of the input to the linear transformation.

Thus, the transformation is augmented with the transformation $D = (A\overline{T}A^{-1})^{l/2}$, where A is an invertible linear function. A is chosen so as to thwart the above weakness. The overall transformation is still cyclic according to theorem 1. Also, the overhead is almost same, due to the fast forwardness property [9]. According to the property, $D = (A\overline{T}A^{-1})^{l/2} = A(\overline{T})^{l/2}A^{-1}$.

That is when $l/2$ number of clock cycles are applied to the transformation $A\overline{T}A^{-1}$ is the same as applying A^{-1}, followed by $l/2$ clock cycles of $\overline{T}$ and then finally A.

An important property of the Diffusion Box is that it should satisfy the avalanche criterion. According to the criterion, if one bit of the input is changed then at least half of the output bits should be affected. From the Cellular Automata theory it may be observed that if the transformation $\overline{T}$ is applied for $n/2$ clock cycles, then the overall avalanche is poor. To observe this, let two inputs to the complemented CA ($\overline{T}$) X_1 and X_2 differ at a single bit position. Then the output xor can be represented by $\overline{T}(X_1) \oplus \overline{T}_(X_2) = \overline{T}(X_1) \oplus \overline{T}(X_2) = T(X)$ where X is $X_1 \oplus X_2$. The number of ones in the xor of the outputs reflects the number of output bits which change when a single bit in the input flips.

Thus, after the $l/2th$ application of $\overline{T}$ the xor of the outputs would be $T^{l/2}(X)$. Since, the block size n is usually chosen to be a power of 2 (in our case 128) $l/2$ is equal to n (from theorem 2).

Thus, the output xor is reduced to $T^n(X)$ which can be shown to be X itself (as T^n is the identity matrix I [11]). Thus a change in a single position in the input to the complemented CA transformation confines itself to that particular position in the output and does not get diffused. Inorder to obtain a healthy diffusion thus $\overline{T}$ is applied $(n-1)$ times while encrypting and $(n+1)$ times while decrypting. This leads to a stronger avalanche effect.

To further improve on the Avalanche effect the matrix A and its inverse are applied. We give our example with a 8 bit structure. However, the matrix is extendible to any block size. The matrices A and A^{-1} are as follows:

$$\mathbf{A} = \begin{pmatrix} 1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&0 \\ 0&1&1&1&1&1&1&0 \\ 0&0&1&1&1&1&1&0 \\ 0&0&1&1&1&1&0&0 \\ 0&0&1&1&1&0&0&0 \\ 0&0&0&1&1&0&0&0 \\ 0&0&0&0&1&0&0&0 \end{pmatrix}$$

and its inverse:

$$\mathbf{A^{-1}} = \begin{pmatrix} 0&1&1&0&0&0&0&0 \\ 0&0&1&1&0&0&0&0 \\ 0&0&0&0&0&1&1&0 \\ 0&0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&1 \\ 0&0&0&0&1&1&0&0 \\ 0&0&0&1&1&0&0&0 \\ 1&1&0&0&0&0&0&0 \end{pmatrix}$$

The regularity in the structure of A helps us to extend it to any dimension and the matrices can also be synthesized using Cellular Automata.

3.2 Construction of the S-Box

In order to construct the S-Box using Cellular Automata, we first require a non-linear Cellular Automata rule which is reversible. Also, the Cellular Automata based S-Box should satisfy the various properties required for a robust S-Box design [12]. A skewed version of rule 30 is:
$x^i_{t+1} = x^i_t \oplus (x^{i+1}_t + x^{i+2}_t)$.

The inverse rule for the non-linear CA may be obtained, as the above rule is invertible, unlike rule 30 CA. However, a serious flaw with it is that the last bit in it passes unchanged. To avoid this, we propose to use "cycles" of this simple rule. After each cycle the output is completely reversed, that is the last bit becomes the first, the second last the second and so on. It is interesting to observe that the robustness of the thus developed S-Box is a function of the number of clock cycles. Indeed in a later section we fix the number of cycles of the simple non-linear rule required to obtain a cryptographically strong S-Box. Indeed, we show that the robustness of the S-box is comparable to some of the strongest S-Boxes obtained in the paper of [12], with the advantage that the implementation is very simple. The idea of repeating simple rules to obtain the behavior is also in conformation with the concept envisaged by [1].

3.3 Performing Key Mixing Using Addition Modulo 2^n

The classical technique to perform key mixing in block ciphers is through exclusive-or (exor). In the CA based block cipher we perform the key mixing using addition modulo 2^n, when the size of the data and the key block is n-bits. It can be shown both theoretically and experimentally that in such a case the bias of the linear approximations falls exponentially fast and helps in foiling Linear Cryptanalysis [15].

The result may be summarised with the following theorems proved in [15].

Theorem 3. *For given n-bit inputs x and k the output is denoted by another n-bit number $y = (x+k) \bmod 2^n$. The probability that each output bit $y[i]$ can be denoted by the linear function $x[i] \oplus k[i]$ is denoted by p_i, $0 \le i < n$. Then $p_i = 1/2 + (1/2)^{i+1}$ and $1/2 < p_i \le 1$.*

Theorem 4. *For given n-bit inputs x and k the output is denoted by another n bit number $y = (x + k) \bmod 2^n$. The largest bias of a linear approximation of $y[i]$ is $(1/2)^{i+1}$.*

From the above results it is evident that:

Corollary 1. *The best linear approximation for $s[i]$ is $a[i] \oplus k[i]$, where the probability of match is $1/2 + 2^{-(i+1)}$ and hence the bias is $2^{-(i+1)}$.*

So, if the key-mixing step in the block cipher is an addition modulo 2^n step, the probability of any linear expression relating to the key elements may be estimated using the above result and the Piling-Up lemma [16]. If the resulting linear expression involves any particular bit position, say the i^{th} bit of the key, the bias of the resulting equation is lesser than $(1/2)^{i+1}$ and as the following table suggests the biases become negligible very fast.

The biases of the linear expression relating the key bits have been computed using the above expression and tabulated in table 1.

We see that the bias of the linear approximations involving the key bits falls very fast. With a key size of 128 the bias of the linear approximations is almost zero (negligible) beyond a bit position of six (marked in table 1). This fact makes the finding of linear approximations in the cipher with a large bias a

Table 1. Biases of Linear Approximations Involving Key Bits

Key Bit Position	0	1	2	3	4	5	↓ 6	7	8	9	10
Bias	0.5	0.25	0.125	0.0625	0.0313	0.0156	0.0079	0.0039	0.0020	0.0010	0.0004

more difficult task. Discovering the key through Linear Cryptanalysis becomes improbable. The result have also been experimentally verified in [15].

Thus, the overall round of the CA based block cipher is elaborated in figure 2. The parameters of the block in the figure have been explained in the next section.

4 Security analysis of the block cipher

In this section the round of the block is evaluated for its security against various security parameters.

4.1 Avalanche Criterion

We first consider the linear transform (D-Box) of the cipher, that is the $D = A\overline{T}^{n-1}A^{-1}$ unit. While discussing the motivation for choosing A, we have indicated that the block is expected to have a strong avalanche effect. Though the discussion was for 8 bits, the matrix can be easily extended to larger block sizes. We have experimentally performed the test on a block size of 128 bits and found out that on an average the outputs differ in 80 bits when the inputs differ in only one bit. The frequency distribution table for the number of bits affected in the output is provided in table 2. The results demonstrate that the block cipher satisfies avalanche criterion.

Table 2. Avalanche Effect of the proposed Cipher

Number of bits where the outputs differ	Frequency
0-10	4
10-20	5
20-30	5
30-40	5
40-50	5
50-60	5
60-70	11
70-80	15
80-90	15
90-100	15
100-110	15
110-120	15
120-130	15

4.2 Evaluation of the S-Box

The S-Box is a crucial component of Substitution Permutation Networks. The S-boxes are supposed to be defiant against Linear Cryptanalysis (LC) and Differential Cryptanalysis (DC). For this they should satisfy various properties like

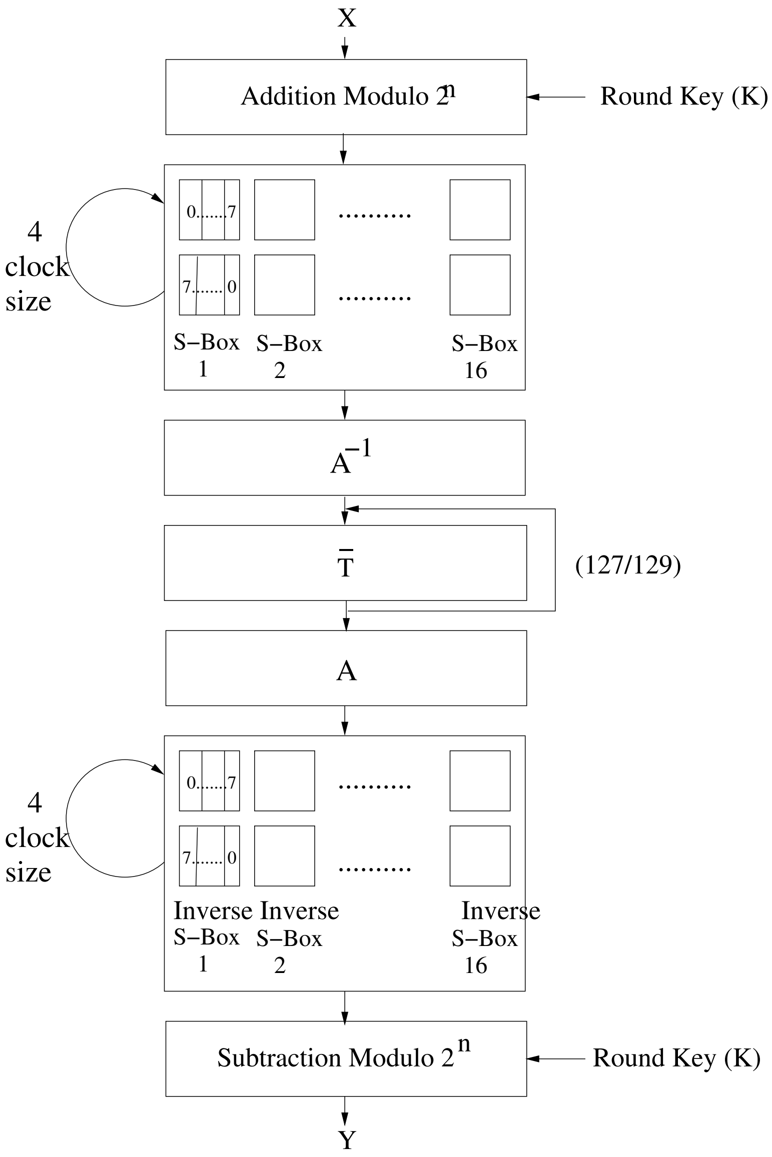

Fig. 2. The CA based round (elaborated)

high non-linearity, balancedness, robustness against differential cryptanalysis and small biases of linear approximations.

Non-linearity and Balancedness:

First, the non-linearity of the S-Box output bits should be high and also they should represent a balaned boolean function. If we consider a m bit S-Box, the non-linear CA (mentioned earlier) gives us a permutation on V_m, that is, we have an invertibe mapping. So the enumeration of the truth tables of the output bits show that all the possible elements of V_m appear. Thus the truth table of each output bit has an equal number of zeros and ones, resulting in the balancedness of the output function. Now, a balanced boolean function cannot attain the highest non-linearity, but it should be close to the maximum non-linearity value. As already mentioned in the preliminaries the maximum non-linearity value for a function on V_m is $2^{m-1} - 2^{m/2-1}$. As described before we iterate the simple non-linear rule over some "cycles". Our observation is that as the number of cycles increase the non-linearity of the S-Box varies and gradually approaches its maximum value. The non-linearity of the cipher for different number of bits in a block and cycles is tabulated in table 3.

Table 3. Non-linearity for different bits and rounds for the cipher

No of bits	Number of cycles	Non-linearity (N_f)	Maximum non-linearity (N_f^{max})	Ratio (N_f/N_f^{max})
4	4	4	6	0.67
4	8	4	6	0.67
8	4	64	120	0.53
8	8	82	120	0.68

Based on the above result we choose to construct a S-Box on 8 bits. Thus each bit is a boolean function of 8 bits and has a high non-linearity. We, next perform a Linear Cryptanalysis and show that the biases obtained are less. In order to facilitate the respresentation we tabulate the result for a 4 bit S-Box, although the 8 bit Linear Approximation Table is even better.

Linear Cryptanalysis:

Linear Cryptanalysis essentially deals with the probability of approximating the input and output of non-linear functions, used in the block cipher with linear expressions [17, 18]. The approach in linear cryptanalysis is to determine expressions of the form below which have a high or low probability of occurrence [17, 18]

Let us consider an expression of the form:

$< X_{i1} \oplus X_{i2} \oplus \ldots X_{i_u} > \oplus < Y_{j_1} \oplus Y_{j_2} \ldots \oplus Y_{j_v} >= 0$

where X_i represents the i-th bit of the input $X = [X_1, X_2, \ldots]$ and Y_j represents the j-th bit of the output $Y = [Y_1, Y_2, \ldots]$. This equation represents the exclusive OR of u input bits and v output bits.

If the bits are chosen randomly then the above approximated linear expression will hold with probability 1/2. It is the deviation from the probability of 1/2 (bias) for an expression to hold that is exploited in linear cryptanalysis: the further away a linear expression is from holding with a probability of 1/2, the better the cryptanalyst is able to apply linear cryptanalysis. We thus prepare a linear approximation table 4 for the non-linear S-Box of the cipher round. Each element in the table represents the number of matches between the linear equation represented in hexadecimal as "Input Sum" and the sum of the output bits represented in hexadecimal as "Output Sum" minus 8. The hexadecimal value representing a sum, when viewed as a binary value indicates the variables involved in the sum. As we can see from table 4 the biases are small and thus the corresponding equations will offer no considerable help for the cryptanalysis. This table is comparable to the table that we obtain while doing a similar analysis for the S-boxes of DES.

Table 4. Linear Approximation Table for 4 bit S-Box

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	-2	2	1	0	2	2	4	0	-2	-2	4	0	2	-2	0
2	0	2	2	0	2	0	-4	-2	0	2	-2	4	2	0	0	2
3	0	-4	0	0	2	-2	2	2	0	4	0	0	2	-2	2	2
4	0	0	2	2	2	2	0	0	2	-2	4	0	0	-4	-2	2
5	0	2	0	2	2	0	-2	4	2	0	-2	-4	0	2	0	2
6	0	2	-4	2	0	-2	0	2	2	0	2	4	-2	0	2	0
7	0	0	-2	2	0	0	2	-2	2	-2	0	0	6	2	0	0
8	0	0	0	0	-2	2	2	-2	4	4	0	0	-2	2	-2	2
9	0	-2	2	0	2	0	0	-2	0	-2	2	0	-2	4	4	2
10	0	2	2	0	4	-2	2	0	0	2	2	0	0	2	-2	4
11	0	4	0	0	0	0	4	0	-4	0	0	0	0	0	0	4
12	0	0	2	2	-4	0	-2	2	-2	2	4	0	2	2	0	0
13	0	2	0	-6	0	2	0	2	2	0	2	0	2	0	2	0
14	0	2	4	2	-2	0	2	0	2	0	-2	0	0	-2	4	-2
15	0	0	-2	2	2	6	0	0	-2	2	0	0	0	0	2	-2

Also, to be noted is that the effect of the non-linear key mixing (described previously) does not give linear approximations of the cipher with high bias. The results of Theorem 3, 4 and Corollary 1 show that the bias of linear approximations through the key mixing step goes down exponentially fast with the bit size. Thus, we do not get linear approximations of the cipher rounds with large bias.

Differential Cryptanalysis:

Next, we evaluate the robustness of the S-Box against Differential Cryptanalysis (DC). Differential cryptanalysis takes the advantage of entries with high values in the difference distribution tables of S-boxes employed by block ciphers. The difference distribution table for a $n \times s$ S-box is a $2^n \times 2^s$ matrix. The rows of the matrix, indexed by the vectors in V_n, represent the change in the input, while the columns, also indexed by the vectors in V_n represent the change in the output of the S-box.

An entry in the table indexed by $(\Delta X, \Delta Y)$ indicates the number of input vectors which when changed by ΔX (bitwise XOR), result in a change in the output by ΔY (bitwise XOR).

From the definition of robustness against DC, it is evident that the values of L and R have to be less. That is, in addition to the requirement of having no large values, the difference distribution table of an S-box should also contain as less non-zero entries as possible in its first column [12].

We again observe that the robustness against Differential Cryptanalysis varies with the number of "cycles" of the S-Box. The robustness of the non-linear transformation of our cipher for different values of the number of bits in a block and the number of cycles are shown in table 5.

Table 5. ϵ-robustness against Differential Cryptanalysis

Number of bits	Number of cycles	ϵ	L	R	Maximum value of $\epsilon(\epsilon_m)$	Ratio(ϵ/ϵ_m)
4	4	0.6250	6	0	0.875	0.71
4	8	0.5000	8	0	0.875	0.57
8	4	0.8125	48	0	0.992	0.82
8	8	0.9297	18	0	0.992	0.94

It follows from the above tables that among all the values, the ratios for non-linearity and ϵ-robustness against differential cryptanalysis are the highest when the block size is 8 bits and the number of cycles is 8.

The performance of the S-Box, with respect to the powerful Differential Cryptanalysis may be compared with that of the some of the standard S-Boxes in literature table 6. From the results we see that the S-Box constructed out of the simple non-linear rule of a CA results in a S-Box which is comparable to that of the standard ciphers. The elegance of the CA based S-Box is that it is simple and extremely easy to implement.

Table 6. Comparision of robustness against Differential Cryptanalysis

S-Box	Robustness (ϵ)
S_1 (DES)	0.316
S_2 (DES)	0.363
S_3 (DES)	0.316
S_4 (DES)	0.469
S_5 (DES)	0.387
S_6 (DES)	0.367
S_7 (DES)	0.340
S_8 (DES)	0.328
[12]	0.875
[12]	0.96875
[12]	0.992
CA based S-Box	0.9297

4.3 Parameters of a Round of the Cipher

Based on the above discussion and the available computational resources of a modern day adversary we decide upon the following structure of the self-invertible CA based round of the block cipher. The block size is 128 bits both for the data and the key. The input to the round and the key are mixed with addition modulo 2^{128}. Next, the data is divided into 16 smaller groups of 8 bits. Each block is operated with the non-linear S-Boxes independently. The S-Box is thus a 8×8 S-box, where the number of cycles is 8. Then the Diffusion Box comes to play and is applied to the 128 bit output of the 16 S-Boxes. The number of iterations of the linear step is 127 for encryption andd 129 for decryption. Next, the data is then divided into 16 blocks and the inverse non-linear step is also applied for 8 cycles. Finally, the 128 bit data is mixed with the round key, this time with a subtraction modulo 2^{128}.

5 Composition of Rounds

Key Mixing is a necessary part of cipher design. According to the basic principles of cryptography the security of an algorithm lies in the key. We compose the unconventional CA based block ciphers (as opposed to a SPN cipher) developed recursively using the same composition principle that we have used (Theorem 1). We thus always have the self-invertible property retained. Figure 3 shows the composition of three CA based rounds. There are two encryption rounds (E_{K1} and E_{K2}) and one decryption round ${E_{K1}}^{-1}$. The encryption rounds have two round keys K_1 and K_2 and the linear part is cycled 127 times, while the decryption block also has the round key K_1 and number of cycles of the linear part is 129. We are currently working on computing the number of such recursions required and in devising a key scheduling algorithm based on the CA Transforms developed in this work.

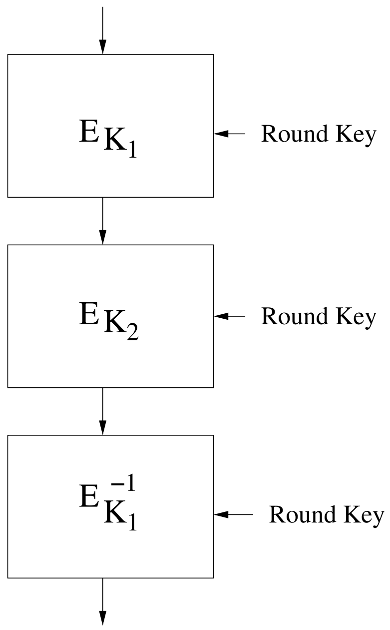

Fig. 3. Composition of the Rounds

6 Conclusions

For the first time, the paper discusses how confusion and diffusion can be achieved in block ciphers using CA based transforms. The self-invertible round proposed is a combination of linear and non-linear CA. The paper shows how repeated applications of simple non-linear rules can help to devise an S-Box with security features comparable to that of the strongest ciphers. Finally, the rounds are also composed recursively to develop a complete block cipher. An interesting future scope of work will be to compute the number of rounds required to achieve the security margin provided by the best block ciphers.

References

1. S. Wolfram, "Statistical mechanics of cellular automata," *Rev. Mod. Phys.*, vol. 55, no. 3, pp. 601–644, July 1983.

2. P. Guan, "Cellular automata public key cryptosystem," *Complex Systems*, vol. 1, pp. 51–57, 1987.
3. B. Kar S. Nandi and P. Pal.Chaudhury, "Theory and applications of cellular automata in cryptography," *IEEE Transactions on Computers*, vol. 43, no. 12, pp. 1346–1357, Dec. 1994.
4. S. Merphy S. Blackburn and K. Paterson, "Comments on theory and applications of cellular automata in cryptography," *IEEE Transactions on Computers*, vol. 46, no. 5, pp. 637–638, May 1997.
5. B. Sikdar N. Ganguly, A. Das and P. Pal.Chaudhury, "Cellular automata model for cryptosystem," in *Cellular Automata 2001*, Yokohama University, Japan, 2001, pp. 120–125.
6. Debdeep Mukhopadhyay and D. Roy Chowdhury, "Cellular automata based cryptosystem employing galois field algebra," Yokohama, Japan, 2001, International Symposium on Cellular Automata.
7. S. Sen, C. Shaw, D. R. Chowdhury, N. Ganguly, and P. Pal Chowdhury, "Cellular automata based cryptosystem (cac)," in 4^{th} *International Conference on Informations and Computer Security (ICICS 2002)*. Dec 2002, pp. 303–314, LNCS 2513.
8. Feng Bao, "Cryptanalysis of a new cellular automata cryptosystem," in *Information Security and Privacy, 8th Australasian Conference, ACISP 2003*, Wollongong, Australia, July 9-11 2003, vol. 2727 of *Lecture Notes in Computer Science*, Springer.
9. Debdeep Mukhopadhyay and Dipanwita RoyChowdhury, "Cellular automata: An ideal candidate for a block cipher," in *In the Proceedings of First International Conference on Distributed Computing and Internet Technology ICDCIT 2004, LNCS 3347*, December 2004.
10. Alexander Klimov, *Applications of T-functions in Cryptography*, Ph.D. thesis, The Weizmann Institute of Science, 2005.
11. P. Pal Chaudhuri, D.Roy Chowdhury, Sukumar Nandi, and Santanu Chattopadhyay, *Additive Cellular Automata Theory and its Application*, vol. 1, chapter 4, IEEE Computer Society Press, 1997.
12. Jennifer Seberry and Xian-Mo Zhang and Yuliang Zheng, "Systematic Generation of Cryptographically Robust S-boxes," 1^{st} Conference Computer and Communication Security, VA, USA, 1993, pp. 171–182.
13. Debdeep Mukhopadhyay and Dipanwita RoyChowdhury, "Characterization of a class of complemented group cellular automata," in *In the Proceedings of ACRI 2004, LNCS 3305*. University of Amsterdam, Science Park Amsterdam, The Netherlands, October 2004.
14. D. Mukhopadhyay and D. RoyChowdhury, "Cellular Automata Based Key Agreement," 2^{nd} International Conference on E-business and Telecommunication Networks, Microsoft Convention Centre at Reading, UK, October, 3-7 2005.
15. D. Mukhopadhyay and D. RoyChowdhury, "Key Mixing in Block Ciphers through Addition modulo 2^n," Cryptology ePrint Archive, 2005.
16. Mitsuru Matsui, "Linear Cryptanalysis method for DES cipher," in *Advances in Cryptology-Eurocrypt 1993*. 1993, pp. 386–397, Springer, volume 765 of LNCS.
17. Douglas R. Stinson, *Cryptography : Theory and Practice*, chapter 3, pp. 79–88, 2002.
18. Howard M. Keys, "A Tutorial on Linear and Differential Cryptanalysis," www.engr.mun.ca/ howard/PAPERS/ldc_tutorial.ps.