Design of a New Software Based Block Encryption Scheme (BES)

Meena Kumari, SAG, Delhi

ABSTRACT

BES, a new software based block encryption scheme is proposed, which is based on a novel way of defining invertible permutations by functions defined over Galois Fields $GF(2^4)$ and $GF(2^5)$ and also using algebraic operations like addition modulo 2^8 and multiplication modulo fermat prime $2^{16}+1$. In this system, 64 bits of plaintext is converted to 64 bits of ciphertext using user defined secret key of 128 bits. The system has 8 rounds, each round operating on 9 sub-blocks of unequal length.

Key words: Confusion, Diffusion, Galois fields, Imprimitive group, S-boxes

1. Introduction

In today's world of fast computation and secret key cryptography the requirement was to design good block cipher scheme. The first approach was to investigate the soundness of some of design rules used in earlier systems like DES, IDEA, RC5, CAST[1]& AES[13][14]. Based on the design rules construct a pseudo random function which generates pseudo random invertible permutations and use cryptographic composition of these, to increase security because if one composes, permutation generators which are slightly secure the result is more secure than either one alone [2]. The concept of confusion and diffusion are the principles behind design of block cipher system. Where confusion means that the ciphertext depends on plaintext and key, in a complicated and involved way and diffusion means that each plaintext bit should influence every ciphertext bit and each key bit should influence every ciphertext bit[3]. As we see in DES[1], in each round, transposition and substitution of plaintext bits results in ciphertext bits following these twin requirements of confusion and diffusion suggested by Shannon[4]. A new secret key block cipher IDEA proposed by Lai & Massey[5] is based on the design concept of mixing three different group operation to achieve the required confusion and diffusion. Confusion is achieved by arranging the operation in a way that no pair of successive operations are of the same type and by the fact that operations of different type are incompatible. The aim is always to choose such that diffusion can be achieved by using a small number of operations. The operations multiplication modulo $2^{16}+1$ and addition modulo 2^{16} are applied in 8 rounds on 16 bit sub-block using 16 bit key. In this system plaintext and ciphertext are 64 bit block while the secret key is 128 bit long. Willi Meier[6] proved that arithmetic properties of the basic operations can be exploited in cryptanalysis of IDEA only upto 2 rounds. IDEA is fully secure if the number of rounds are 8. Another secret key block cipher RC5 proposed by Ron Rivest [7], in 1994 has variable word size, variable number of rounds and a variable length of the key. The novel feature of the algorithm is the use of data dependent rotations and mixed use of XOR and ADD operations. RC5 works on 32 bit word, has 12 rounds and 128 bit key. Knudsen & Meier[8] has shown that RC5 has many weak keys with respect to differential attack. These weaknesses can be exploited by linear & differential attacks if number of rounds is less. Recently, it was shown that 64 bit word version of RC5 is also much weaker than it was expected[9].

The basic ingredients of modern fast software block encryption schemes are the computer instructions like SHIFT, ADD, XOR, arithmetic operations like exponentiation and multiplication. Instead of using these by trial and error the aim was to analyse each of these and subsets of these to use in new design. Also the properties such as avalanche i.e. small change in key should result

in complete change in ciphertext, nonlinearity, resistance against several known cryptanalytic attacks are most essential to be fulfilled by any cipher system. Besides these, another important criterion that the given encryption algorithm realizes a large variety of different permutations among its binary input vectors. The important question in terms of resistance against cryptanalytic attacks is whether the generated permutation group is primitive or imprimitive group[10]. Suppose P is the plaintext, C is the ciphertext obtained by applying the encryption algorithm E_k based on key K. Assume that encryption algorithm E generates an imprimitive group G. Let B_i and B_j be the blocks for G with $P \in B_i$ and $C \in B_j$. Then according to the definition of imprimitivity, every other plaintext P' which is contained in B_i will automatically be mapped to B_j. This will certainly reduce the search amount. This property would undermine the security of encryption scheme. Thus it is essential that the encryption scheme should be designed such that it generate a primitive group. Also to make the block cipher scheme resistant against linear and differential attacks, many number of rounds of designed scheme should be applied with different keys.

Thilo Zieschang[11] has studied the combinatorial properties of basic encryption operations. The importance of these combinatorial properties on the security of block ciphers had been pointed out by several authors. In section 2 , we will analyze the combinatorial structure of those basic cryptographic functions which are basic ingredients of any encryption scheme. Different subsets and combinations of basic operations result in completely different levels of security. In order to introduce non linearity in the block cipher design, and also to form an encryption function that generates a primitive group, a new function is defined over higher field which leads to invertible permutations of the given input vector. This is defined in section 3.

In section 4, encryption and decryption algorithms of proposed block cipher design, BES are explained.Also key schedule algorithm is explained. A block cipher first divides the plaintext in blocks of fixed size and then encipher each of these blocks with the same key. Disadvantages of block ciphers is that for any chosen key inputting a given block of data always results in the same block of cipher. Another disadvantage is that for short messages, message extension is necessary. Also a single error transmission is likely to cause a complete block of data to be in error. But Block Ciphers can be used in block chaining mode, stream cipher mode, cipher feedback mode and output feedback mode for file encryption, mail encryption, satellite data protection and other applications.

2. Basic Encryption Operations

The basic operations are bijective functions on the set of n-bit binary vectors. All basic operations like modulo-2 addition i.e. XOR; ADD, addition modulo 2^n, MUL multiplication modulo 2^n and MULT multiplication modulo 2^n+1 when applied on n-bit vectors can be regarded as permutations. Also permutations can be realized as product of these basic operations[10]. We will analyse the cycle structure and element order of these basic operations .

XOR-C This function operating on binary vector of length n works as follows. The input value is XORed with some arbitrary, but constant value c. If c is not equal to all zero string then we get an involution.

Example 2.1 Let n=4 and suppose C=[1010]. Identify the set of 4 bit strings with the corresponding decimal number 0,1,.....15. This yield the permutation

XOR_C ≈
(0,10)(1,11)(2,8)(3,9)(4,14)(5,15)(6,12)(7,13)

For 2^n-1 different values of C, XOR-C yields 2^n-1 different fixed point free involutions. Also XOR-C is always an even permutation since the number of transpositions in its cycle decomposition is equal to 2^{n-1}.

SHIFT-K This function is a logical shift (rotation) to the right by K-positions. The order of this permutation is n/gcd(k,n). The permutation is not fixed point free and contains cycles of different length dividing n. SHIFT-K has $2^{gcd(k,n)}$ fixed points of length gcd(k,n).

Example 2.2 Let n=4

SHIFT-1≈
(0)(1,2,4,8)(3,6,12,9)(5,10)(7,14,13,11)(15)

ADD-C This function represent addition modulo 2^n of the constant value C, ignoring overflow. Add-C, $0 < C < 2^n$ is always fixed point free regular permutation i.e. it consists of k cycles of length m, $m \neq 1$ such that $km = n$. Cycle length $m = 2^n/\gcd(C,2^n)$. Thus ADD-C is always an even permutation. ADD-C is odd permutation if $\gcd(C,2^n) = 1$. ADD-C is even if C is even.

Example 2.3 ADD-1 ≡ $(0,1,2,\ldots,2^n-1)$ is a 2^n-cycle. For $n = 4$

ADD-5 ≈ (0,5,10,15,4,9,14,3,8,13,2,7,12,1,6,11) is a 2^4- cycle

MUL-C This operation means multiplication modulo 2^n (ignoring overflow) by same constant C, where C is odd and $0 < C < 2^n$.

MUL-C is a bijective mapping iff $\gcd(C,2^n) = 1$. Zero & 2^n-1 are always fixed points, odd numbers are always mapped to odd and even number are always mapped on even numbers. The length of every cycle in MUL-C divides 2^n-2. So MUL-C produces odd permutations as well as even permutations depending on C, more exactly MUL-C is an even permutation even iff $C \equiv 1 \bmod 4$. Example 2.4 let $n = 4, C = 5$

MUL-5≈
(0)(1,5,9,13)(2,10)(3,15,11,7)(4)(6,14)(8)(12)

MULT-C For values of C such that $(C,2^n+1) = 1$. MULT-C is a bijective function which multiplies the input variable by the constant C modulo 2^n+1. Instead of numbers $\{0,1,2,\ldots,2^n-1\}$ we can consider the set $\{1,2,\ldots,2^n\}$ replacing zero by 2^n. If 2^n+1 is prime, MULT-C is a fixed point free regular permutation. If C is a quadratic residue modulo 2^n+1, then MULT-C must be an even permutation.

Example 2.5 Let $n = 2$, then $2^n+1 = 5$ is prime,

MULT-2 = (1,2,4,3)

MULT-3 = (1,3,4,2)

MULT-4 = (1,4)(2,3)

Result 1:

$\langle \{XOR\text{-}c \mid c \in \{0,1\}^n\} \rangle = \langle \{XOR\text{-}c_j \mid 1 \leq j < n\} \rangle$ is an elementary abelian group of order 2^n and is isomorphic to $(Z_2)^n$

Result 2:

$\langle \{SHIFT\text{-}k \mid 0 \leq k < n\} \rangle = \langle SHIFT\text{-}1 \rangle$, The generated group is isomorphic to cyclic group of order n

Result 3:

$\langle \{ADD\text{-}C \mid 0 \leq c < 2^n\} \rangle = \langle ADD\text{-}1 \rangle$, The generated group is isomorphic to cyclic group of order 2^n

Result 4:

$\langle \{MULT\text{-}c \mid 0 \leq c < 2^n\} \rangle$ where $p = 2^n+1$ is prime. The generated group is isomorphic to cyclic group of order 2^n. If 2^n+1 is not prime then structure of the multiplicative group $Z/(2^n+1)Z$ varies in accordance with n.

Result 5 :

$\langle \{MUL\text{-}c \mid c \text{ odd}, 0 \leq c < 2^n\} \rangle$ The generated group is a multiplicative group of $Z/(2^nZ)$ and is an abelian group of order 2^{n-1}.

Also the subsets of these basic operations can be combined together to achieve a large number of encryption function i.e. permutation of binary n-bit vectors with presumably high structural complexity[10].

Theorem 2.1: The group G generated by the set $S := \{SHIFT\text{-}d, XOR\text{-}c \mid 0 \leq d < n, c \in \{0,1\}^n\}$ of all SHIFT and XOR operations is isomorphic to a semi direct product $G = NU$ of an elementary abelian 2-group N of order 2^n by a cyclic group U of order n. The order of G is equal to $n2^n$ and G can be generated by $n+1$ elements. If $n = 2^k$ is a power of 2, then G is non abelian 2-group of order 2^{k+n}

Theorem 2.2 : The group G of degree 2^n, $n \in N$ generated by the set

$\langle \{ADD\text{-}c, SHIFT\text{-}d \mid 0 \leq c < 2^n, 0 \leq d < n\} \rangle$ of all ADD and SHIFT operations is isomorphic to the symmetric group of degree 2^n.

Theorem 2.3 : The group $G = \langle MUL\text{-}c, SHIFT\text{-}d \mid 0 < c < 2^n, c \text{ odd}, 0 \leq d < n \rangle$

of degree 2^n-1, $n \in N$ generated by all MUL and SHIFT operations acting on the set

$\Omega' = \Omega - \{0\}$ of order 2^n-1, is isomorphic to symmetric group of degree 2^n-1

Based on above results and theorems, we came to the conclusion of incorporating the following points in designing

1. XOR and SHIFT operations are not to be incorporated singly because Result 1 and Result 2 confirm that the generated groups are of order n and is less than 2^n.
2. Addition modulo 2^n should be incorporated because of Result 3, the generated group is of order 2^n.
3. Multiplication modulo 2^n+1 should always be incorporated when 2^n+1 is prime because of Result 4. This is only possible in the case when n=8 and n=16 which justifies that in earlier block cipher designing generally blocks are divided into length 8 or 16.
4. Multiplication modulo 2^n should not be applied because the generated group is of order 2^{n-1} (see Result 5).
5. SHIFT and XOR operations should not be used together because on binary vector of length n. this scheme produces at most $n2^n$ different encryptions(see Theorem 3.1).
6. SHIFT and ADD operations should be applied together because on binary vector of length n, this scheme generates all possible encryptions i.e. $2^n!$
7. MUL and SHIFT can be incorporated because this combination generates all possible encryptions of order $2^n-1!$ on set $\Omega-\{0\}=\{1,2,\ldots,2^n-1\}$
8. If a pair {A,B} of basic operations already generate the symmetric group of degree 2^n then obviously larger set {A,B,C....} of basic operations also forms a generating set for this group.

3. Genration of Pseudo Random Invertible Permutations from a Function Defined Over Higher Field.

As mentioned in the introduction, the security of the system can only be enhanced if the encryption scheme generates the primitive group of permutations. The problem reduces to developing a function which generate a pseudo random invertible permutations and finding which cryptographic compositions of these pseudo random invertible permutations should be used to make the system secure. In case of DES, the main security of the system is enhanced by the nonlinearity introduced by the S-Boxes, which consists of 32 permutations of numbers 0 to 15. These permutatins appear random although it was felt that some function must be used to generate these. Nonlinearity; can be introduced by multiplying the elements of the field. The elements of higher field say $GF(2^m)$, $m>1$ can be generated by considering the mth degree primitive polynomial over GF(2). The elements of $GF(2^m)$ are m-length vectors over GF(2) or can also be expressed as powers of the primitive element α of $GF(2^m)$[12].

Example 3.1: The elements of $GF(2^4)$ = GF(16) fields when α is a primitive root of $1+x+x^4$ are listed in Table 3.1

Table 3.1 Elements of Galois Field GF(16)

Power of α	Vector	Decimal No.	Power of α	Vector	Decimal No.
$\alpha^{-\infty}$	0000	0	α^7	1011	11
α^0	0001	1	α^8	0101	5
α^1	0010	2	α^9	1010	10
α^2	0100	4	α^{10}	0111	7
α^3	1000	8	α^{11}	1110	14
α^4	0011	3	α^{12}	1111	15
α^5	0110	6	α^{13}	1101	13
α^6	1100	12	α^{14}	1001	9

Permutation Group S_{16} is the set of 1-1 onto mappings of $\Omega=\{0,1,\ldots..,15\}$. Then permutations over $\Omega=\{0,1,\ldots..,15\}$ can be defined by the function $f(x) = \alpha^p x+ \alpha^q$, $x = \alpha^I$, $I= -\infty, 0, 1\ldots,14$ where $0 \le p \le 2^4-2$, $0 \le q \le 2^4-2$, and α is a primitive root of 4th degree primitive polynomial over GF(2).

Let α is a primitive root of polynomial $1+x+x^4$ over GF(2). When p = 1, q = 6, f(x) for different values of x is given by

When $x = 0 = \alpha^{-\infty}$ $\alpha^1\alpha^{-\infty}+ \alpha^6= \alpha^6$ $= 12$
$x = 1 = \alpha^0$ $\alpha^1\alpha^0 + ^6= \alpha^{11}$ $= 14$
$x = 2 = \alpha^1$ $\alpha^1\alpha^1 + ^6= \alpha^3$ $= 8$
$x = 3 = \alpha^4$ $\alpha^1\alpha^4 + ^6= \alpha^9$ $= 10$
$x = 4 = \alpha^2$ $\alpha^1\alpha^2 + ^6= \alpha^2$ $= 4$
$x = 5 = \alpha^8$ $\alpha^1\alpha^8 + ^6= \alpha^5$ $= 6$
$x = 6 = \alpha^5$ $\alpha^1\alpha^5 + ^6= \alpha^{-\infty}$ $= 0$
$x = 7 = \alpha^{10}$ $\alpha^1\alpha^{10} + ^6= \alpha^1$ $= 2$
$x = 8 = \alpha^3$ $\alpha^1\alpha^3 + ^6= \alpha^{12}$ $= 15$
$x = 9 = \alpha^{14}$ $\alpha^1\alpha^{14} + ^6= \alpha^{13}$ $= 13$
$x =10 = \alpha^9$ $\alpha^1\alpha^9 + \alpha^6= \alpha^7$ $= 11$
$x =11 = \alpha^7$ $\alpha^1\alpha^7 + \alpha^6= \alpha^{14}$ $= 9$
$x = 12 = \alpha^6$ $\alpha^1\alpha^6 + \alpha^6= \alpha^{10}$ $= 7$
$x = 13 =\alpha^{13}$ $\alpha^1\alpha^{13} + \alpha^6= \alpha^8$ $= 5$
$x = 14 = \alpha^{11}$ $\alpha^1\alpha^{11} + \alpha^6= \alpha^4$ $= 3$
$x = 15 = \alpha^{12}$ $\alpha^1\alpha^{12} + \alpha^6= \alpha^0$ $= 1$

$$\alpha^1 x + \alpha^6 = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 12 & 14 & 8 & 10 & 4 & 6 & 0 & 2 & 15 & 13 & 11 & 9 & 7 & 5 & 3 & 1 \end{bmatrix}$$

Inverse permutation generated by function $f(x) = \alpha^p x+ \alpha^q, \forall x = \alpha^I$, $I= -\infty, 0, 1\ldots,15$ is given by

$\alpha^{15-p} x+ \alpha^{q-p}$ if $q \ge p$ and
$\alpha^{15-p} x+ \alpha^{q-p+15}$ if $q < p$

For the above example, inverse permutation of permutation generated by $\alpha^1 x + \alpha^6$ is given by $\alpha^{15-1} x + \alpha^{6-1} = \alpha^{14} x + \alpha^5$

$$\alpha^{14} x + \alpha^5 = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 6 & 15 & 7 & 14 & 4 & 13 & 5 & 12 & 2 & 11 & 3 & 10 & 0 & 9 & 1 & 8 \end{bmatrix}$$

The set of permutations generated by the set $<\{\alpha^p x+ \alpha^q, 0 \le p \le 2^4-2, 0 \le q \le 2^4-2\}>$ is a group because product of two permutations is a permutation generated by this set and there exists a inverse of each permutation .

Similarly permutations over $\Omega = \{0, 1, 2, \ldots\ldots..,31\}$ can be defined by the function

$F(x)=\beta^p x+\beta^Q$, $\forall x=\beta^I, I=-\infty, 0, 1, 2, \ldots..,30$

Where β is a primitive root of the primitive polynomial of degree 5 over GF(2)

Example 3.2 : The elements of Galois Field $GF(2^5)$ = GF(32), when β is a primitive root of $1+x^2+x^5$ are listed in Table 3.2

The set of permutations generated by $<\{\beta^P x+ \beta^Q, 0 \le P \le 2^5, 0 \le Q \le 2^5\}>$ is a group because product of two permutations belongs to this set

Table 3.2 Elements of Galois Field GF(32)

Powers of β	Vector	Decimal No.	Powers of β	Vector	Decimal No.
$\beta^{-\infty}$	00000	0	b^{15}	11111	31
β^0	00001	1	b^{16}	11011	27
β^1	00010	2	b^{17}	10011	19
β^2	00100	4	b^{18}	00011	3
β^3	01000	8	b^{19}	00110	6
β^4	10000	16	b^{20}	01100	12
β^5	00101	5	b^{21}	11000	24
β^6	01010	10	b^{22}	10101	21
β^7	10100	20	b^{23}	01111	15
β^8	01101	13	b^{24}	11110	30
β^9	11010	26	b^{25}	11001	25
β^{10}	10001	17	b^{26}	10111	23
β^{11}	00111	7	b^{27}	01011	11
β^{12}	01110	14	b^{28}	10110	22
β^{13}	11100	28	b^{29}	01001	9
β^{14}	11101	29	b^{30}	10010	18

and inverse of each permutation exists and is given by function

$\beta^{31-P}\, x + \beta^{Q-P}$ if $Q \geq P$ and

$\beta^{31-P}\, x + \beta^{Q-P+31}$ if $Q < P$

For example the inverse of permutation generated by

$\beta^{11}\, x + \beta^{6} \approx$

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
10	13	4	3	22	17	24	31	23	16	25	30	11	12	5	2
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
21	18	27	28	9	14	7	0	8	15	6	1	20	19	26	29

is given by $\beta^{-11+31}\, x + \beta^{6-11+31} = \beta^{20}\, x + \beta^{26} \approx$

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
23	27	15	3	2	14	26	22	24	20	0	12	13	1	21	25
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
9	5	17	29	28	16	4	8	6	10	30	18	19	31	11	7

4. Description of BES

In the block cipher BES (A new software based Block Encryption Scheme) plaintext and ciphertext are 64 bit block and key is 128 bit long. The encryption algorithm is based on a novel design of using MULT and ADD on 16 bit and 8 bit number respectively and using nonlinear functions f and F defined in GF(2^4) and GF(2^5) respectively to generate S-box type permutations. It also implements the Feistel Network structure as used in DES,CAST and others on 32 bit blocks. The use of this was necessary because it helps to satisfy the basic block cipher requirements and simplify decryption algorithm.

The BES is an iterated cipher consisting of 8 computationally identical rounds. The complete one round encryption process is depicted in the diagram in figure 4.1and described in section 4.1.

4.1 Encryption

In the encryption process, three different arithmetic group operations and two functions f and F on higher fields GF(2^4), GF(2^5) are used Instead of dividing the 64 bits plaintext block into equal length sub blocks to apply arithmetic operations, the 64 bits block is divided into 5 sub blocks X_1, X_2, X_3, X_4, X_5 of unequal length 16, 8, 16, 8, 16 respectively. Also out of 82 bit key, first 64 bit key is divided into 5 sub blocks Z_1, Z_2, Z_3, Z_4, Z_5 of length 16,8,16,8,16. Out of the remaining 18 bits, 8 bits Z_6 and Z_7, each of length 4 each are used to define the function f over GF(2^4) and 10 bits Z_8 and Z_9, each of length 5 are used to define the function F over GF(2^5).

- Each of 16 bit sub block of plaintext bits X_1 and of key Z_1 is treated as usual radix-2 representation of an integer except that the all zero sub-block is treated as representing 2^{16}, then multiplication of the two integers formed is done modulo $2^{16}+1$ (this is denoted by ©) and the resulting number is represented as X_{11} in binary form. Similarly X_3 ©Z_3 leads to X_{33} and X_5 © Z_5 leads to X_{55}.
- Each of 8 bit sub-blocks of plaintext bits i.e. X_2 & X_4 are treated as radix-2 representation of the integers and also of the key i.e. Z_2 & Z_4 and then addition of integers modulo 2^{16} is applied this is denoted by [+]; i.e. X_2[+]Z_2 leads to X_{22} and X_4[+]Z_4 leads to X_{33}, where X_{22} and X_{44} are the resulting numbers in binary form.
- Bit by bit exclusive OR of two 32 bit sub-blocks is denoted as ⊕.
- * is used for concatenation of the bits.
- The key bits Z_6 & Z_7 are treated as radix-2 representation of integer p & q and function f defined as $f(x) = \alpha^p x + \alpha^q$ (see section 3) is applied to the 16 bits of X_{11} by dividing it into 4 sub-blocks of 4 bits and applying function f by representing 4 bits as some power of α, where α is a primitive root of primitive polynomial $1+x+x^4$ over GF(2) (see Table 3.1) and representing the resulting power of alpha i.e. as a vector of 4 bits(from Table 3.1). Similarly applying the same function to other subblocks of 4 bits each and then concatenate the four 4 bits outputs to form 16 bits output and denote it by $f(X_{11})$.

Example 4.1

if Z6 : 1101

Z7 : 0101 and

X11 : 1101 1001 0001 0000

p = 13

q = 5

$f(x) = \alpha^{13}x + \alpha^5 \quad \forall\ \alpha^I$,

$I = -\infty, 0, 1, 2, \ldots, 14$

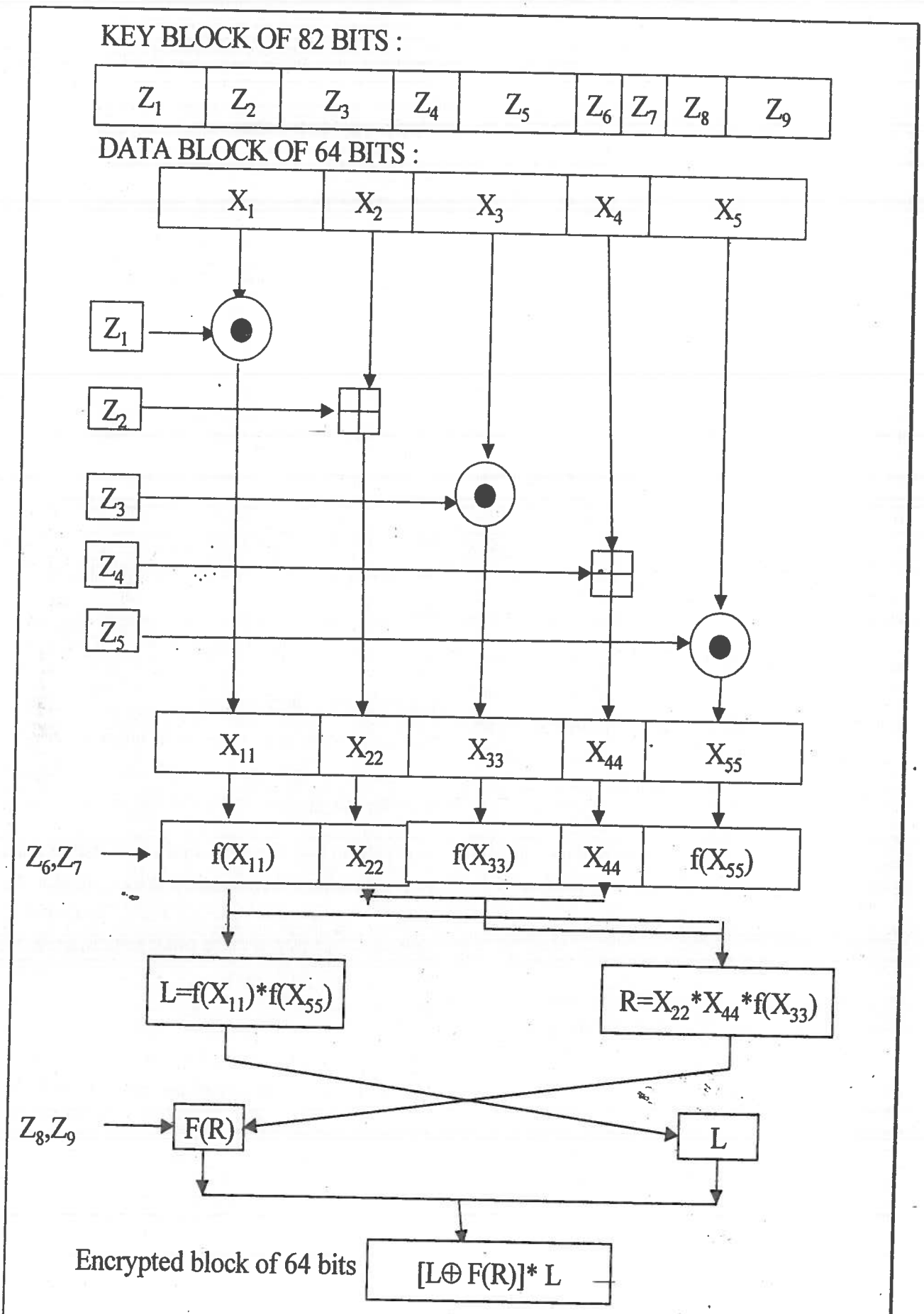

Fig. 4.1 Encryption Process of one round

From Table 3.1 ,first 4 bits of X_{11} that is x=1101 represents α^{13}

$f(\alpha^{13}) = \alpha^{13}.^{13}+\alpha^5 = \alpha^{26-15}+\alpha^5 = \alpha^{11}+\alpha^5 = \alpha^3 = 1000$

The next 4 bits of X_{11} are 1001, when x = 1001 , it is α^{14}(see Table 3.1)

$f(\alpha^{14}) = \alpha^{13}.\alpha^{14}+\alpha^5 = \alpha^{27-15}+\alpha^5 = \alpha^{12}+\alpha^5 = \alpha^{14} = 1001$

when, x = 0001, it repesents α^0

$f(\alpha^0) = \alpha^{13}.\alpha^0+\alpha^5 = \alpha^{13}+\alpha^5 = \alpha^7 = 1011$

when, x = 0000, it represents $\alpha^{-\infty}$

$f(\alpha^{-\infty}) = \alpha^5 = 0110$

$f(X_{11}) = 1000\ 1001\ 1011\ 0110$

Similarly we can find $f(X_{33})$, $f(X_{55})$.

Now form a block L of 32 bits by concatenation of $f(X_{11})$ & $f(X_{55})$

i.e. $L = f(X_{11}) * f(X_{55})$ and also a block R of 32 bits by concatenation of X_{22} ,X_{44} and $f(X_{33})$ i.e. $R = X_{22} * X_{44} * f(X_{33})$

- The key bits Z_8 & Z_9 are treated as radix-2 representation of integer P & Q and function $F(x) = \beta^P x+\beta^Q$ generates a permutation of 32 numbers 0 to 31 when β is a primitive root of primitive polynomial $1+x^2+x^5$over GF(2).
- 32 bits of R are permuted to form F(R) by permutation generated by $F(x) = \beta^P x+\beta^Q$.
- Then apply XOR operation to L & F(R) i.e. L⊕F(R) . It is a 32-bit vector.
- Then finally concatenate L⊕F(R) and the 32 bits of L to form the output bits i.e. encrypted block

$Y = L \oplus F(R) * L$ of block X .

Thus 64 bits plaintext block X is transformed to the 64 bits ciphertext block Y under the control of 82 key bits selected from 128 bit secret key(selection procedure is described in the key schedule. For r=1,2,..8, the 9 subblocks of key used in the rth round are denoted as $Z1^{(r)}$, $Z2^{(r)}$,....,$Z9^{(r)}$

4.2 The Decryption Process

The computation graph of the decryption process is shown in the figure 4.2. The decryption key sub-blocks $K_i^{(r)}$ are computed from encryption key subblocks $Z_i^{(r)}$ as follows

$(K_1^{(r)}, K_2^{(r)}, \ldots\ldots, K_5^{(r)}) = (Z_1^{(9-r)}, -Z2^{(9-r)}, Z3^{(9-r)}, -Z4^{(9-r)}, Z5^{(9-r)})$

Where Z^{-1} denote multiplicative inverse (modulo $2^{16}+1$) of Z i.e. $Z \odot Z^{-1}=1$ and

-Z denote the additive inverse modulo 2^{16} of Z i.e. $-Z [+] Z = 0$

Also if $f(x) = \alpha^P x+\alpha^q$

Then $f^{-1}(x) = \alpha^{15-p} x+ \alpha^{q-p}$ if $q \geq p$

and

$= \alpha^{15-p} x+ \alpha^{q-p+15}$ if $q < p$

Also if $F(x) = \beta^P x+\beta^Q$

Then $F^{-1}(x) = \beta^{31-P} x+ \beta^{Q-P}$ if $Q \geq P$

and

$= \beta^{31-P} x+ \beta^{Q-P+31}$ if $Q < P$

4.3 The Key Schedule

The key bits used in encryption process in eight rounds are generated from 128 user – selected key bits as follows:

The 128 bit key is partitioned into 9 sub-blocks which are directly used as the first 9 key sub-blocks (leftmost bit = most significant bit). The ordering of the key sub-blocks is defined as follows: $Z_1^{(1)}$, $Z_2^{(1)}$, ..., $Z_9^{(1)}$, $Z_1^{(2)}$,....., $Z_9^{(2)}$,....., $Z_1^{(8)}$, $Z_2^{(8)}$,..., $Z_1^{(8)}$

The 128 bit user selected key is then cyclic shifted to the right by 31 positions, after which the resulting 128 bit block is again partitioned into 9 sub-blocks that are taken as the next nine key blocks. The obtained 128 bit block is again cyclic shifted to the right by 31 positions to produce the next nine key sub-blocks, and this procedure is repeated until all 72 key sub-blocks have been formed.

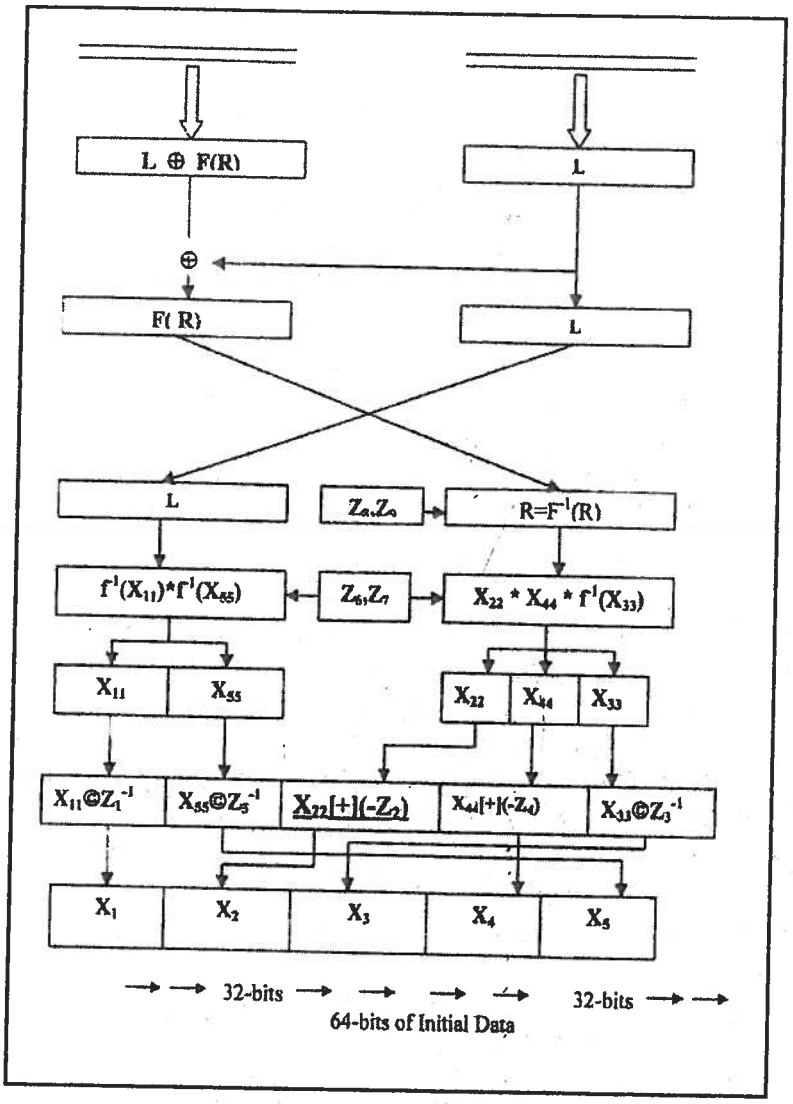

Fig. 4.2 Decryption Procedure of one round

5. Conclusion

A highly secure (order of magnitudes higher than DES) has been designed satisfying the requirements of confusion, diffusion, nonlinearity, resistance against cryptanalytic attacks. In this system, complicated S-Boxes have been replaced, by a novel way of generating S-boxes type permutations. This is done by defining a function over higher fields GF(2^n), which leads to permutations of order 2^n. The function which defines the inverse permutation is also formed, which is used in decryption. This helps in reducing the requirement of memory. As we see in other systems like DES, CAST[1], the initial permutations and expansion tables, as well as S-boxes require a lot of memory to implement. Another important point is that the S-boxes defined in these systems were either random or not key dependent. In this system the S-box type permutations of order 2^4, 2^5 are generated by a function dependent on user defined key. The special features like unequal lengths block lengths, confusion and diffusion implemented through basic operations, less memory requirement because permutation tables are not used and S-boxes type permutations depending on the user defined key, make the system unconventional and highly secure.

Acknowledgement

The authors wish to thank Director, SAG for giving permission to pursue this work and permission to send this paper. The authors would like to thank Dr. S. S. Bedi, Sc. 'F', for many valuable suggestions and encouragement.

References :

1. Schneier Bruce, "Applied Cryptography, Protocols, Algorithms and Source code in C", John Wiley and Sons, Inc.,New York, 1996
2. Luby Michael and Rackoff Charles, "How to Construct Pseudo Random Permutations from Pseudo Random Functions", SIAM J. Comp., vol 117,NO. 2 1988, pp 373-386
3. Kumar I. J.; "Cryptology, System Identification and Key Clustering", Aegean Park Press , USA
4. Shannon C.E., "Communication Theory of Secrecy Systems", Bell System Technical Journal,28,July,1949
5. Lai X. and Massey J.L., "A proposal for a new Block Encryption Standard, advances in Cryptology", Eurocrypt 90 Proceedings, Lecture Notes in Computer Science, Springer Verlag, Berlin1991, pp 389-404
6. Meier Willi, "On the Security of the IDEA Block Cipher, advances in Cryptology", Eurocrypt 93, Proceedings, Lecture Notes in Computer Science 765, Springer Verlag, Berlin, 1993, pp 371-385
7. Rivest R. L., "The RC5 Encryption on Algorithm", Lecture Notes in Computer Science 1008, Fast Software Encryption, Springer Verlag, Berlin, 1994, pp 86-95
8. Knudsen Lars R. and Meier Willi, "Improved Differential Attacks on RC5", Proceedings Crypto 96, Lecture Notes in Computer Science 1109, Springer Verlag, Berlin, 1996, pp 216-228
9. Birgukov Alex and Kushilevitz Eyal, "Improved cryptanalysis of RC5, advances in Cryptology", Eurocrypt 98, Proceedings, Lecture Notes in Computer Science 1403,1999, Springer Verlag, Berlin, pp85-96
10. Zieschang Thilo, "Combinatorial Properties of Basic Encryption Operations, advances in Cryptology", Eurocrypt 97, Proceedings, Lecture Notes in Comp. Science 1233,1997, Springer Verlag, Berlin, pp14-26
11. Wielandt H., "Finite Permutation Group", Academic,1964
12. Lidl Rudolf & Niederreiter H., "Introduction to Finite Fields And Their Applications", Cambridge university Press,1986
13. Daemen J. & Rijmen V. , "AES Proposal", Rijndael
14. Fuller J. and Millan W., "Linear redundancy in the AES S-box", Aug 2002, manuscript 2002/111 on IACR E-print Archive