Implementation of Stream Cipher System Based on Representation of Integers In Residue Number System

Ganesh Aithal
Research Scholar
Department Of Electronics and Communication
National Institute of Technology
Karnataka, Srinivasanagr, Suratkal,
Mangalore, India.
ganeshaithal@gmail.com

K.N.Hari Bhat
Department of Electronics and Communication
Nagarjuna College of Engineering,Venkatagiri Kote post,
Bangalore 562110,Karnataka,
India

U.Sripathi
Department Of Electronics and Communication
National Institute of Technology
Karnataka, Srinivasanagr, Suratkal,
Mangalore 575 025,Karnataka,
India

Abstract—**Residue Number Systems (RNS) based on Chinese Remainder Theorem (CRT) permits the representation of large integers in terms of combinations of smaller ones. The set of all integers from 0 to M-1 with RNS representation and component wise modular addition and multiplication constitutes direct sum of smaller commutative rings. Encryption and decryption algorithm based on the properties of direct sum of smaller rings offers distinct advantages over decimal or fixed radix arithmetic. In this paper representation of integer using RNS, is successfully utilized in additive, multiplicative and affine stream cipher systems.**

The property of the cipher system based on RNS number system allow speeding up the encryption / decryption algorithm, reduce the time complexity and provides immunity to side channel, algebraic, and known plain text attacks. In this paper, the characteristics of additive, multiplicative and affine stream cipher systems, the key generation, and encryption and decryption based on RNS number system representation are discussed.

Keywords- Chinese Remainder Theorem(CRT), Residue Number System(RNS), Ring structure, Stream Ciphert, Parall processing, Key Sequence.

INTRODUCTION

The properties like carry free, parallelism and modularity in arithmetic, of the RNS based on CRT, is more appropriate in some applications, than the standard number systems. They are, used in very high speed low power VLSI implementation [1], coding theory [2], signal processing [3] and cryptographic applications [4].

The RNS based on CRT number system has received considerable interest in arithmetic computation and digital signal processing. The main reasons for the wide spread use of RNS system is its inherent properties of parallelism resulting in fast addition and multiplication process, which is used in speech processing [3]. A new architecture for modular adder is presented in [5], which offers smaller area and high speed. Application of CRT in designing quasi-cyclic code is discussed in [2]. In [6] new residue comparator based on CRT, are developed for three-moduli set $\{2^{n-1}, 2^n, 2^{n+1}\}$, and the FPGA implementation results there show that the proposed modulo comparators are about 20% faster and smaller than one of the previous best designs. CRT decomposition is also used in reducing chip area of edge sampler circuit by employing the method of delay technique [7]. RNS systolic approach offers an impressive amount of fault tolerance at low cost in hardware, which is discussed in [8]. The residue number system based on CRT is also used in cryptographic applications. CRT RSA is used in wide verity of application. A smart card application is presented in [9], with fault tolerance attack immunity. In [10] a new class of product-sum type public key cryptosystem is defined, based on CRT and it is invulnerable to low density attack. Application of CRT in decomposition of block encrypted cipher text for transmission over multi channel is discussed in [11]. This scheme hides the cipher text in order to increase the difficulty of attacking the cipher. In this paper we propose a stream cipher system, which makes use of the advantages of representation of integer using RNS.

Most of the existing stream cipher systems [12] [13] [14] deal with binary plain text and key sequences. Encryption operation is generally bit by bit mod 2 addition. In case of non binary stream cipher system, in addition to additive encryption operation it is also possible to have multiplication or both addition and multiplication operations. Incorporation of two different algorithms successively leads to higher security level. There is no analogous scheme in case of binary stream cipher system.

The level of security provided by any stream cipher system generally depends on three factors they are

- Number of choices of the key (order of key space)
- Period of the key
- Computational complexity of the encryption and decryption algorithms.

A basic a stream cipher scheme is shown in figure 1.1, where the message (plain text) is a sequence of source alphabets, which is mapped to corresponding sequence $\{r_j\}$ integers. Each

978-1-4244-4791-6/10/$25.00 ©2010 IEEE

integer representing the message alphabet is subjected to key dependent reversible mapping called encryption. The key sequence is also a sequence of random integers generated using any existing method [15] [16]. A stream cipher based on RNS representation handles smaller integers by decomposing the plain text integers into smaller ones. These smaller components can be handled independently, and facilitates us to processes the operation in parallel. This leads to reduction in time for encryption / decryption without sacrificing the security.

Further, security systems are generally classified in to system of strategic and tactical application. In strategic application the security systems have to withstand attack for a very long duration, typically several decades. As a result size of the key is large; the required security can be achieved even with simpler encryption algorithm like additive cipher, multiplicative cipher or both. The computational complexity mainly depends on encryption and decryption algorithms. Generally larger the complexity, better the security is. Thus with large key size it is possible to have required security even with simpler encryption algorithm.

In stream cipher system, the plain text sequence $\{r_j\}$ is sequence of integers and is encrypted integer by integer using random sequence of integers as key sequence $\{K_j\}$, to get sequence of cipher text $\{c_j\}$, where r_j, K_j and c_j are from Z_m, ring of residue class integer modulo m.

Generally the plain text sequence element and key sequence elements are subjected to addition or multiplication or combinations of both addition and multiplication, modulo an integer m. The resulting systems are respectively called as additive, multiplicative or affine systems. Here encryption algorithm is modular addition or multiplication or both carried out using key elements. In case of additive cipher a single key sequence is used for encryption. In case of multiplicative cipher a single key sequence of integers having multiplicative inverse in Z_m, is used for encryption. However in case of affine system two different key sequences, one for addition and another for multiplication are used. The scheme shown in figure 1.1, corresponds to additive or multiplicative system.

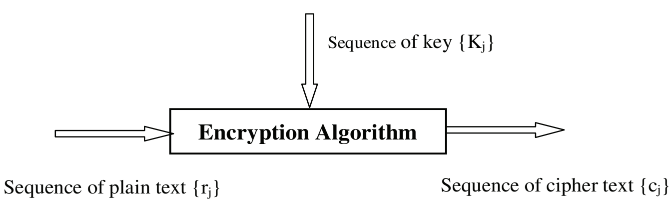


Figure 1.1 Basic Stream Cipher System

The rest of the paper is organized as follows. In section II we introduce the necessary mathematical background. Algorithm and figure of merits are discussed in section III. Section IV makes the security analysis. An Example is discussed in section V and section VI is conclusion.

MATHEMATICAL PRELIMINARIES

Two number systems, binary and binary coded decimal are commonly used in computers. Both are weighted number systems. Even though weighted number system has got many advantages, the major disadvantage is speed with which arithmetic operations are performed, because of carry propagation. One of the ways to increase the speed of arithmetic is to have the carry free operation in parallel. This carry free and concurrent operation can be accomplished using the residue number system based on CRT.

A. *Residue Number System Based On CRT*

Let M be any composite integer, which can be expressed in terms of product of its k factors, which are pair wise relatively prime,

$$M=\prod_{i=0}^{k-1} m_i \qquad (1)$$

That is, g.c.d. $(m_i, m_j) = 1$ for all $i \neq j$. Let r be any integer such that, $0 \leq r \leq M-1$, Then the system of congruence, $r \equiv r_0 \bmod m_0$, $r \equiv r_1 \bmod m_1$, $r \equiv r_2 \bmod m_2 \ldots.$ $r \equiv r_{k-1} \bmod m_{k-1}$, has a unique solution r mod M, satisfying all the congruences

$$\{r_{k-1}, r_{k-2}, \ldots\ldots.r_1, r_0\} \qquad (2)$$

This is called Chinese Remainder Theorem (CRT).

The digits r_i $i = 1,2,3,\ldots.k-1$, are such that $0 \leq r_i \leq m_i-1$. Thus 'r' can be expressed in terms of unique 'k' tuple using CRT. The representation of r in terms of k digits is called RNS representation based on CRT.

B. *Conversion of RNS Representation to Fixed Radix*

Suppose a positive integer r, $0 \leq r \leq M-1$, has RNS representation $r \rightarrow (r_{k-1}\ r_{k-2}\ r_{k-3} \ldots\ldots r_2\ r_1\ r_0)$ then r is given by, $(r_{k-1}\times w_{k-1}+r_{k-2}\times w_{k-2}+\ldots r_{k-3}\times w_{k-3}\ldots r_2\times w_2+r_1\times w_1+r_0\times w_0) \bmod M$, w_j's are obtained as follows. We define M_j equal to M/m_j, which is relatively prime to m_j, $0 \leq j \leq k-1$. Now w_j is defined as [1]

$$w_j= [\{M_j^{-1} \bmod m_j\}M_j] \bmod M \qquad (3)$$

and

$$r = [\sum_0^{k-1} r_i\, w_i] \bmod M \qquad (4)$$

C. *Algebraic Structure of Set of all k Tuple* $\{r_{k-1},r_{k-2},\ldots\ldots.r_1, r_0\}$

Let M be as defined in equation (1). Consider set S of integer {0, 1, 2, 3, M-1}. Each integer in the set can be uniquely represented in RNS system. The set of all RNS representation of integer in set S constitute a commutative ring, where the operation addition and multiplication are defined as bellow:

Let the CRT representation of any integer $0 \leq a \leq M-1$ be $(a_{k-1}\ \ a_{k-2}\ \ldots..\ a_2,a_1,a_0)$ And another integer $0 \leq b \leq M-1$ be $(b_{k-1},\ b_{k-2}\ \ldots..\ b_2, b_1, b_0)$ Then addition of two representations is defined as

$$(a_{k-1}\oplus b_{k-1}) \bmod m_{k-1}\ldots(a_i\oplus b_i) \bmod m_i\ldots(a_0\oplus b_0) \bmod m_0 \qquad (5)$$

Multiplication is defined as

$$a_{k-1} \otimes b_{k1}) \bmod m_{k1}\ldots(a_i \otimes b_i) \bmod m_i\ \ldots..(a_0 \otimes b_0) \bmod m_0 \qquad (6)$$

Further additive identity is (0,0,0,0,0……,0,0)

Additive inverse of $a_{k-1}\ \ a_{k-2}\ \ldots..\ a_2\ \ a_1\ \ a_0$ is

$$(m_{k-1}\oplus(-a_{k-1})) \bmod m_{k-1} \ldots \ldots \ldots \ldots(m_i\oplus(-a_i)) \bmod m_i \ldots(m_0\oplus(-a_0))) \bmod m_0 \qquad (7)$$

Multiplicative identity is (1,1,1,1, 1 ……1, 1)

Further it is verified that under these operations, the set of all k tuple satisfies the axioms of a finite commutative ring [17], called direct sum of rings $Z_{m(k-1)}\ Z_{m(k-2)} \ldots Z_{mi} \ldots Z_{m0}$ and denoted as $Z_{m(k-1)}\oplus Z_{m(k-2)}\oplus\ldots Z_{mi}\oplus\ldots\oplus Z_{m0}$. Further it is shown that this ring is isomorphic to Z_M, ring of residues modulo M.

Stream Cipher Algorithm Based On RNS Representation

First we consider basic cipher system defined over Z_m. Consider a sequence of plain text where each symbol is assigned a numerical value greater than or equal to zero and

less than or equal to m. Referring to fig 1.1 the encryption and decryption algorithm for the stream cipher is defined as below. In each case there is single stream of plain text $\{r_j\}$, key sequence$\{K_j\}$ or $\{a_j\}$ and cipher text $\{c_j\}$.

A. ADDITIVE CIPHER

Encryption algorithm: $\{c_j\} = \{r_j\} \oplus \{K_j\} \bmod m$ (8)

Decryption algorithm: $\{r_j\}=\{c_j\} \oplus \{-K_j\} \bmod m$

$=\{c_j\} \Theta \{K_j\} \bmod m$ (9)

Where $-K_j$ is additive inverse of K_j mod m The scheme of generating random binary sequence using feed back shift register are discussed in [18]. The same concept can be extended to generate non-binary key sequence. In general sequence $\{K_j\}$ is non-binary. Many methods of generating sequence of random integers with desirable properties are reported in literature [15] [16].

B. MULTIPLICATIVE CIPHER

The key sequence $\{a_j\}$ should be a sequence of integers, which are relatively prime to m. In which case, multiplicative inverse of a_j mod m exists.

Encryption algorithm: $\{c_j\} = \{r_j\} \otimes \{a_j\} \bmod m$ (10)

Decryption algorithm: $\{r_j\} = \{c_j\} \otimes \{a_j^{-1}\} \bmod m$ (11)

Where a_j^{-1} is multiplicative inverse of a_j mod m. The key sequence $\{a_j\}$ is generated using feedback shift register, where feedback operation is multiplication modulo m of contents of selected stages to get a periodic sequence. The initial content should not all be 1's and zero or zero divisor as initial value. That is the initial content should be only integers, which are relatively prime to m. The number of stages and initial value may be appropriately chosen, so that the resulting sequence has desired period. A scheme of generating sequence $\{a_j\}$ using shift registers is discussed in [19][20]

C. AFFINE CIPHER

This is combinations of additive and multiplicative cipher.

Encryption algorithm: Let$\{c_j'\}= \{r_j\} \oplus \{K_j\} \bmod m$,

The cipher text $\{c_j\}=\{c_j'\} \otimes \{a_j\} \bmod m$ (12)

Decryption algorithm:$\{c_j'\}=\{c_j\} \otimes \{a_j^{-1}\} \bmod m$

Plain text $\{r_j\}=\{c_j'\} \oplus \{-K_j\} \bmod m$

or $\{r_j\}=\{c_j'\} \Theta \{K_j\} \bmod m$ (13)

where the key sequence $\{a_j\}$ satisfies the conditions given in section III-B.

D. ENCRYPTION BASED ON RNS REPRASENTATION

Because of carry free arithmetic RNS representation, in general is well suited for parallel arithmetic computations. In case of cipher systems, the plain text is subjected to a key dependent reversible mapping. Any plain text data is first mapped in to sequence of integers less than M. Every integer representation of plain text, $0 \le r \le M-1$, can be uniquely represented by k tuple in RNS. In the encryption operation, depending on the key, the plain text k tuple is mapped to cipher text k tuple. During decryption, the inverse mapping, maps the cipher text k tuple to the corresponding original k tuple. Any integer $0 \le r \le M-1$ can be expressed as k tuple of smaller integers. Thus it is possible, using RNS representation, to decompose a large integer less than M, in to k components, which can be processed independently.

The plain text sequence $\{r_j\}$ is thus decomposed in to k component sequences $\{r_{ij}\}$. The k random key sequences $\{K_{ij}\}$ for additive cipher or k random sequences$\{a_{ij}\}$ for multiplicative cipher i=0,1,2,3….k-1 and j=0,1,2,3,…….. are generated such that, $0 \le K_{ij} < m_i$, and $1 \le a_{ij} < m_i$ using feedback shift register as discussed in section III-A and III-B respectively. As shown in figure 3.1 each component sequence $\{r_{ij}\}$ i=0,1,2,3,4,5,…..k-1 and j= 0,1,2…….. is encrypted using the corresponding k-key sequences $\{K_{ij}\}$ and /or k-key sequences $\{a_{ij}\}$ to get sequence of cipher text $\{c_{ij}\}$ i = 0,1,2,3….k-1 and j = 0,1,2,3,…….. . As mentioned earlier $\{a_{ij}\}$ should satisfy condition discussed in section III-B. The encryption and decryption algorithms are identical to given in equation (8) to (13), except that the plain text sequence, key sequence and cipher text sequences are all streams of k tuples.

The encryption algorithms are defined as follows considering ith channel.

a ADDITIVE ENCRYPTION

Cipher text $\{c_{ij}\}=[\{r_{ij}\} \oplus \{K_{ij}\}] \bmod m_i$ (14)

i = 0,1,2,3….k-1 and j=0,1,2,3,……

b MULTIPLICATIVE ENCRYPTION

Cipher text$\{c_{ij}\}=[\{r_{ij}\} \otimes \{a_{ij}\}]$ (15)

mo),1,2,3. k-1 an 2,3 …

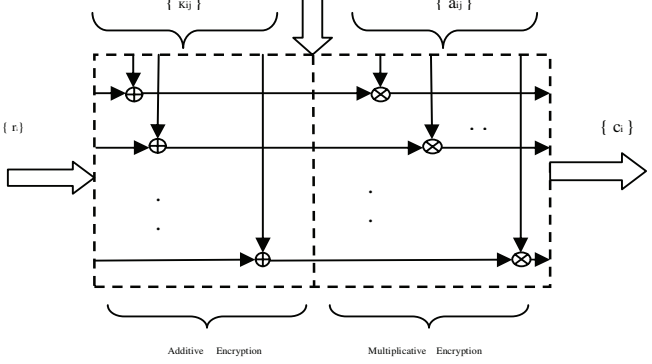


Figure 3.1 Affine scheme of Encryption Algorithm

c AFFINE ENCRYPTION

This is the combinations of additive and multiplicative scheme $\{c_{ij}'\}=\{r_{ij}\} \oplus \{K_{ij}\} \bmod m_i$ and

Cipher text $\{c_{ij}\}=\{c_{ij}'\} \otimes \{a_{ij}\} \bmod m_i$ (16)

i=0,1,2,3….k-1 and j = 0,1,2,3,…

The scheme is shown in figure 3.1.

Where the key sequences $\{K_{ij}\}$ and $\{a_{ij}\}$ satisfies the conditions discussed in III-A and III-B. The cipher text elements are stored, transmitted serially or in parallel. The k encryption can be carried out in parallel.

E. DECRYPTION ALGORITHM

Having received component of $\{c_{ij}\}$ of cipher text blocks of k components, the decryption can also be done in parallel as shown in figure 3.2. The plain text sequence $\{r_{ij}\}$ is recovered

from the received cipher text and locally generated key sequences $\{-K_{ij}\}$ or $\{a_{ij}^{-1}\}$ at the receiver. The $\{a_{ij}^{-1}\}$ are generated as discussed in section III-B, with initial content being multiplicative inverse of initial content used for encryption.

a Additive Decryption

Plain text $\{r_{ij}\}=\{c_{ij}\}\oplus\{-K_{ij}\}\bmod m_i$

or $=\{c_{ij}\}\Theta\{K_{ij}\}\bmod m_i$ (17)

i = 0,1,2,3….k-1 and j=0,1,2,3,…….

b Multiplicative Decryption

$$\text{Plain text } \{r_{ij}\}=\{c_{ij}\}\otimes\{a_{ij}^{-1}\}\bmod m_i \quad (18)$$

i = 0,1,2,3…k-1 and j =0,1,2…

c *AFFINE DECRYPTION*

$\{c_{ij}'\}=\{c_{ij}\}\otimes\{a_{ij}^{-1}\}\bmod mi$ and

Plain text $\{r_{ij}\}=\{c_{ij}'\}\oplus\{-K_{ij}\}\bmod m_i$

$=\{c_{ij}'\}\Theta\{K_{ij}\}\bmod m_i$ (19)

i=0,1,2,3….k-1and j=0,1,2,3,…

Figure 3.2 depicts affine decryption scheme.

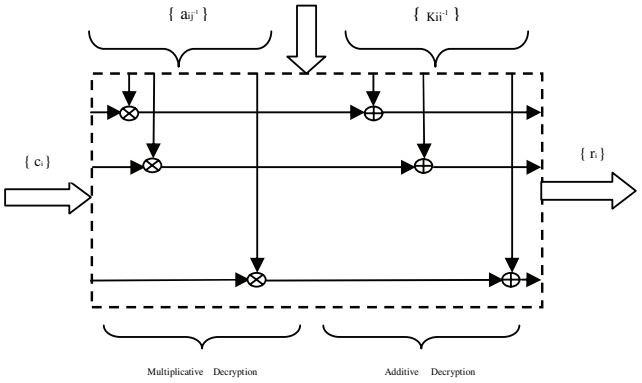


Figure 3.2 Affine scheme of Decryption Algorithm

Even though figure 3.1 and 3.2 shows the encryption and decryption scheme of affine cipher system of the proposed cipher system respectively, for the additive scheme multiplicative block and for multiplicative scheme additive blocks are omitted.

From the algorithm it is observed that the encryption of plain text integer is component wise addition or multiplication or both. This denotes that the encryption of the plain text is evaluated component wise in parallel. In parallel operation the time taken for encryption of plain text is time taken for its largest CRT component. The time estimation of the arithmetic operation can be expressed in terms of bit operation as given by Koblitz [21]. It is shown that, two decimal numbers 'x' and 'y' with x > y, the number of bit operation for addition is given by $2\log_2(x)$ and for multiplication is $(\log_2(x)+1)^2$ respectively. The bit operation is also expressed in terms of '*O*' notation [21]. Since the operations are done in parallel the time taken for encryption and decryption are predominantly decided by the larger factor of M in the expression (1). It can be easily shown that the time taken for additive and multiplicative encryption is respectively $O \log (m_j)$ and $O (\log m_j)^2$ where
$m_j = \max(m_0, m_{1...}..... m_{k-1})$.

If the arithmetic operations are carried out using lookup table then the time taken for encryption / decryption algorithm are independent of factor (m_0, $m_{1...}$….. m_{k-1}) of M.

As discussed earlier the plain text elements are mapped to corresponding integers. To evaluate the performance of the schemes the cipher text elements are also converted into corresponding decimal integers using RNS to decimal conversion. The performance of the encryption algorithms are investigated on the basis of following parameters.

- the standard deviation of the number of occurrence of the cipher text elements,
- the entropy of the set of all cipher text elements,
- the histograms of plain text and cipher text for the three algorithms,
- the mean value of absolute difference between integer corresponding to cipher text and plain text.

Avalanche effects of the algorithms are also observed by encrypting the same plain text with two different key sequences generated using initial values differing in one bit position. The standard deviation of number of occurrence of cipher text element is compared as follows:

Let n_i be the number of occurrence of integer 'i' in cipher text $0 \le i < M$. Let N be the total number of integers in the sequence. Then mean value of number of occurrence

$$\bar{n} = \lim_{N\to\infty} \frac{1}{M}\sum_{i=0}^{M-1} n_i \quad (20)$$

Where $N = \sum_{i=0}^{M-1} n_i$ Then standard deviation of number of occurrences of cipher text σ is given by

$$\sigma = \sqrt{\frac{1}{M}\sum_{i=0}^{M-1}\left(n_i - \bar{n}\right)^2} \quad (21)$$

A small value of σ indicates that, all the cipher text integers occur with almost equal number of times in the sequence of length N. This provides security against statistical attack.

Entropy is a measure of uncertainty. Entropy of a source is maximum and hence the uncertainty, when all the elements in the source occurs with equal probability. Let p_i be the probability of occurrence of integer 'i' in the cipher text sequence i = 0,1, 2 ….M-1

$$pi = \lim_{N\to\infty} \frac{n_i}{N} \text{ and}$$

$$\text{Entropy} = \lim_{N\to\infty} \sum_{i=0}^{M-1} p_i \log_2 \frac{1}{p_i}$$

average bits/cipher text element (22)

Higher the value of entropy better the level of security.

Histogram is a plot of n_i verses i, i = 0, 1….M-1. The integer i may correspond to plain text or cipher texts. A flat histogram of cipher text indicates that all cipher texts occurs equal number of times.

$$\text{Mean absolute difference} = \frac{1}{N}\sum_{i=1}^{N} \left|c_j \sim r_j\right| \quad (23)$$

where c_j is cipher text integer and r_j is plain text integer. A large value of absolute mean difference indicates very small residual information in the cipher text.

SECURITY ANALYSIS

Shannon [22] stated that by making the statistical structure of plain text is dissipated in to long range statistics of the cipher text and relationship between the statistics of the cipher text and value of encryption key as complex as possible, cryptanalysts task becomes hard. The first technique is called diffusion and the second one is confusion. Further in a stream cipher system if the period of the key sequence is larger than the length of plain text the cipher system approaches one time pad, which is theoretically secure [23]. It is seen that the proposed scheme exhibits confusion. By properly choosing the number of stages of shift register it is possible to have the period of key sequence larger than plain text sequence length and make the scheme approach one time pad.

The proposed scheme has immunity against attacks like exhaustive key search, algebraic attack and side channel attacks which are discussed bellow.

A. *EXHAUSTIVE KEY SEARCH*

Given a segment of key amount of known key sequence stream from the key stream generator, the most obvious way to determine the state of generator is to search through all possible states, checking for the match between the resulting and observed key stream. This search is conducted by guessing all the values of the inner state. For each guess, the attacker generates all values of the generator and compares the result with the known key stream; if they differ the guess is discarded as being wrong. Otherwise guess is added to the set of key candidate. The method of encryption and key generation discussed is based on Feedback Shift Register (FSR). The total number of states for a binary FSR is 2^L, for non binary case of arithmetic modulo m the number of state will be m^L, where L is the number of stages of FSR. Again there are k FSRs which generate k different key sequences. The number of stages of FSR used and the number of factors of the CRT system used decides the key space. Increase in number of factors of M or number of stages of key generation increases the size of the key space..

In cryptographic algorithm key space refers to the set of all possible keys. In the proposed system the key space depends on i) n_1; number of possible stages of feedback shift register ii) n_2 ; number of possible feedback coefficients of the feedback shift register used in key generation iii) n_3 ; number of possible seed values iv) s; number of choices for choosing factors of M to represent RNS v) d; the orders in which the k factors are chosen for representing M in RNS. Some of the properties of i, ii, and iii using feedback shift register, of binary values are discussed in [18]. The fourth, fifth and sixth properties depend on the characteristics of RNS system representation and these properties form part of the key.

For brute force attack, the number A of trails cryptanalyst has to perform is given by the product of number shown

$$A=n_1\times n_2\times n_3\times k\times s\times d \qquad (24)$$

By proper choice of values of n1,n2,n3, k, d and s whenever possible the value of A can be made sufficiently large depending on the application.

B. *ALGEBRAIC ATTACK*

Algebraic attack on stream cipher is to find and solve appropriate system of multivariate linear equations in order to break the target cryptosystem [24][25]. Algebraic attack can be foiled by using nonlinear feedback in the generation of key sequences. Generation of key sequence $\{K_{ij}\}$, using shift register employs addition modulo m_i and generation of key sequence $\{a_{ij}\}$ using shift register employs multiplication modulo m_i. Thus whether it is addition or multiplication modulo m_i, the feedback is nonlinear. Hence the proposed scheme is immune against algebraic attack.

C. *SIDE CHANNEL ATTACK*

All cipher systems, which need security not only on the channel but also to the device, which processes the encryption and decryption. That is, one can find secret key or plain text or both even by analyzing the processing system. These cipher systems are generally implemented on a physical device which interact with and influenced by their environment. The electronic devices like computers mobiles smart cards pagers etc. consume power and emits radiations as they operate; they also react to temperature changes and electromagnetic field. The performance of a certain specific operation in takes specified duration. All the above can be used for cryptanalysis. This type of analysis is called as side channel attack. If computations are processed in parallel with a parallel architecture for computation; there is immunity agents side channel attack [26] [27].The proposed system supports the processes the operation in parallel for encryption and decryption by using RNS system, in which case the system becomes immune to side channel attack.

The application of the proposed scheme is illustrated by an example. The plain text chosen is 3000 characters from first page of "*Contemporary Cryptology a Foreword*" (Cryptology (from the Greek…)) [28].The proposed scheme is compared with the standard binary stream cipher A5/1 [29] recommended for GSM. The comparison is made from the point of view of entropy of set of cipher text, standard deviation of number of occurrences of cipher text characters, mean absolute difference between cipher text and plain text integers and histogram of plain text and cipher text.

EXAMPLE

The example of RNS based cipher system, considered here are of three category they are i) Additive ii) Multiplicative and iii) Affine cipher systems. In these examples, as mentioned above a paragraph of text of 3000 seven bit ASCII character from a text file is considered as finite discrete information source. These 3000 characters are first converted to eight bit ASCII with parity. The plain text sequence $\{r_j\}$ is the sequence of decimal value of 8 bit ASCII characters. The sequences of integers $\{r_j\}$ is converted in to three sequences of RNS digits with factors m_0=3, m_1=5 and m_2=17, as $\{r_{0j}\}$, $\{r_{1j}\}$ and $\{r_{2j}\}$ respectively, which are prime integers, hence pair wise relatively prime. All these digit sequences are encrypted in three separate channels with three separate key sequences $\{K_{ij}\}$ for additive and $\{a_{ij}\}$ for multiplicative cipher, six key sequences are used for affine cipher. The sequence is generated using 15 stages feedback shift register, with feedback and initial conditions chosen randomly. The initial value chosen are given in table 5.2 for additive, table 5.3 for multiplicative and table 5.4 for affine ciphers. The key sequence satisfies condition given in section III-A and III-B. After encryption the cipher text sequence of each channel $\{c_{0j}\}$, $\{c_{1j}\}$ and $\{c_{2j}\}$ are

processed for decryption, separately in three channels with a locally generated key sequence, using feedback shift registers. As discussed in section III-E, and conditions given in section III-A and III-B, for additive cipher the initial values of feedback shift registers are same, where as for multiplicative cipher initial values are multiplicative inverse modulo m_i for values given in table 5.3 and 5.4. the initial values for affine is shown in table 5.4

To evaluate the performance of the system, quantitatively and graphically, i) Standard deviation ii) Entropy iii) Mean absolute differences are computed as discussed in section III-E. Histogram of plain and cipher text are plotted. The avalanche effect is observed by plotting cipher text sequence, with two different key sequences differing with one bit position in initial value, for a given plain text sequence.

All the above calculations are done as follows. Let binary 8 bit (pattern) be defined by P_i, where index i is decimal equivalent of binary pattern. For example P_0 = 00000000, P_1= 00000001, ….. P_{254}= 11111110. For the sake of obtaining good mathematical structure, only 255 are considered by omitting 255^{th} level, by merging the highest two levels. Let n_i denote number of occurrence of P_i i = 0, 1, 2 ….254 and N be the total number of bytes. Then standard deviation and entropy are computed as follows.

Standard Deviation

Here the number of symbols N=3000, Standard deviation σ of number of occurrence of any P_i, is computed using equation 31 and 32 where M is 255. it is calculated for three encryption schemes and tabulated in table 5.1. Very small value of σ indicates all the bytes occur with almost equal number of times or equiprobabl. 'σ ' should be small .

Entropy of a cipher text is calculated as follows:

$\sum_{i=0}^{255} p_i \log_2(\frac{1}{p_i})$, where p_i is probability of occurrence of the pattern P_i. In the example considered the maximum entropy is 8 bit / character, which occur when all the 8 bit characters are equally likely.

To compare the efficiency of the proposed scheme based on RNS representation, the standard binary stream cipher A5/1 [20] recommended for GSM is considered. In this a binary plain text sequence is encrypted using arbitrary binary seed key of 64 bits. Received cipher text bit streams are divided in to blocks of 8 bits, converted in to decimal and entropy of set of cipher texts, standard deviation of number of occurrences of cipher text, mean absolute difference between plain text and cipher text are determined and histogram is plotted. The values of both proposed and A5/1are tabulated in table 5.1.

Table 5.1 shows the value of Entropy, Standard deviation and Mean absolute difference, for three different CRT cipher systems with additive multiplicative, affine and A5/1 system

S. No	Scheme of Encryption	Standard Deviation	Entropy	Absolute Mean Difference
1	Additive Cipher with three digit of residue number system based on CRT with m0=3, m1=5 and m2 =17	1.8538	7.93151	83.34
2	Multiplicative Cipher with three digit of residue number system based on CRT with m0=3, m1=5 and m2 =17	2.84181	7.65330	80.63
3	Affine Cipher with three digit of residue number system based on CRT with m0=3, m1=5 and m2 =17	1.91246	7.92437	79.59
4	A5/1 encryption system with random seed key elements	4.4084	7.89677	84.89

Entropy is a measure of uncertainty. Higher the value of entropy of set of cipher texts implies that larger effort is needed by the cryptanalyst. In case of source, with 8 bit plain text elements (or cipher text elements) the largest possible entropy is 8 bit per plain text element. Higher value of entropy indicates security against statistical attack. It is shown in table 5.1 that the entropy in additive and affine scheme are greater than 7.92 and hence it is immune to statistical attack. This value for A5/1 scheme is 7.89677. Small value of standard deviation of number of occurrences of cipher text implies that all cipher texts occur almost equal number of times. It is found that standard deviation of number of occurrences of cipher text of the proposed scheme is small compared to the standard system A5/1.

The histogram of the plain text is given in figure 5.1 and of cipher text elements for three schemes in figure 5.2, 5.3 and 5.4 respectively for additive, multiplicative and affine cipher system of the proposed scheme. The histogram of cipher text of A5/1, is shown in figure 5.5. These cipher text histograms indicates that the histogram is distributed over all integers compared to plain text.

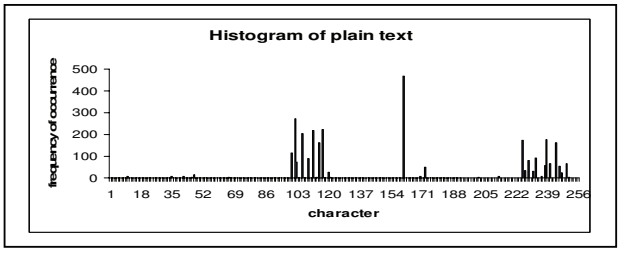


Figure 5.1 Histogram of Plain Text of 3000 character chosen for encryption

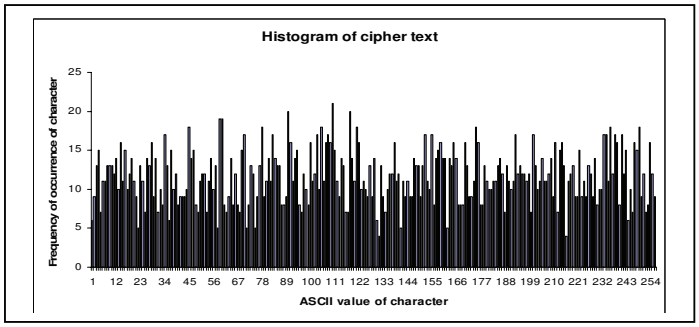


Figure 5.2 Histogram of the additive cipher system with three tuple CRT number with m_0=3, m_1=5, and m_2=17 and initial values of feedback shift registers are as shown in table 5.2 For all three shift registers content of stage 14 and 15 are added mod m_i and fed back, i = 0,1 and 3

Table 5.3, Initial conditions of feedback shift registers SR_1, SR2, and SR3 for additive cipher system

shift register Stage		*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*
Initial values	SR 1	*1*	*2*	*1*	*2*	*1*	*2*	*1*	*2*	*1*	*2*	*1*	*2*	*1*	*2*	*1*
	SR 2	*1*	*2*	*3*	*4*	*1*	*2*	*3*	*4*	*0*	*1*	*2*	*3*	*4*	*0*	*1*
	SR 3	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*

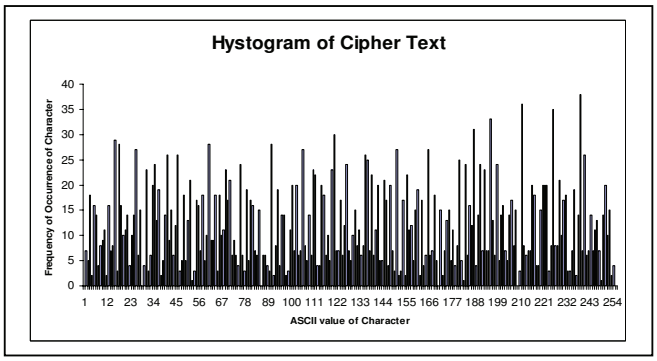


Figure 5.3 Histogram of the multiplicative cipher system with three tuple CRT number with m_0=3, m_1=5, and m_2=17 and initial values of feedback shift registers are as shown in table 5.3 for channel 3. For all three shift registers content of stage 14 and 15 are added modulo m_i and fed back, i = 0, 1 and 3.

Table 5.2, Initial conditions of feedback shift registers SR_1, SR2, and SR3 for multiplicative cipher system

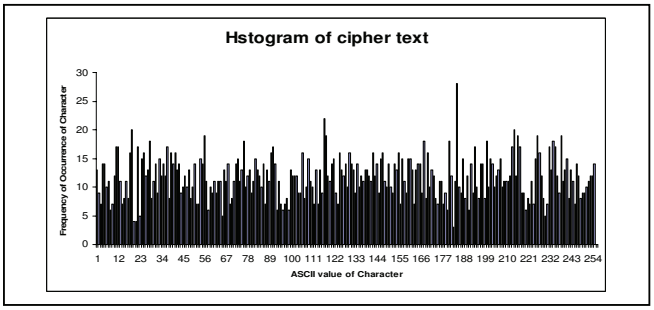
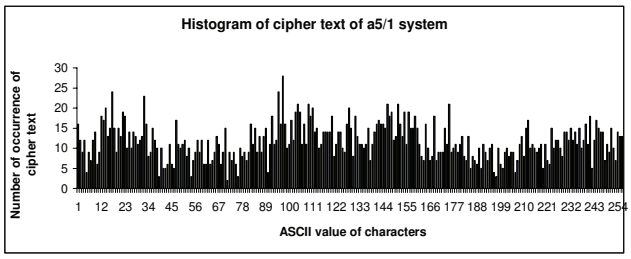


Figure 5.4. Histogram of the affine cipher system with three tuple CRT number with m_0=3, m_1=5, and m_2=17 with initial values of feedback shift registers for additive and multiplicative keys are given in table 5.4 For all three shift registers, of additive and multiplicative content of stage 14, 13, 11 and 9 are added, and multiplied modulo m_i respectively and fed back, i=0,1 and 3.

Table 5.4, Initial conditions of feedback shift registers SR_1, SR2, and SR3 for both multiplicative and additive cipher system

Shift register Stage			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Additive	*Initial values*	SR 1	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1
		SR 2	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1
		SR 3	2	2	4	4	6	7	8	9	10	11	12	13	14	15	16
Multiplicative	*Initial values*	*SR 1*	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
		SR 2	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1
		SR 3	2	2	4	5	6	7	8	9	10	11	12	13	14	15	16



5.5 Histogram of cipher text of A5/1 cipher system, with initial values of feedback shift registers are arbitrarily chosen

The avalanche effect is observed by considering the encryption of first 100 plain text integers $\{r_{ij}\}$ out of 3000 integers of the example. For each plain text, r_j two cipher texts c_j and c_j' with two different key sequences, differing in one digit are computed. The result of each scheme is plotted in figures 5.6, 5.7 and 5.8 for additive, multiplicative and affine ciphers respectively as j vs r_j,c_j,c_j'. The horizontal axis is the plain text element number and vertical axis gives the integer value of plain texts and the corresponding cipher texts. It is observed that in all the three cases the two cipher text values differ appreciably indicating the presence of avalanche effect.

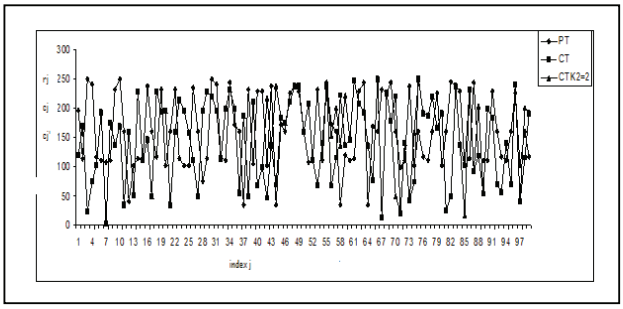
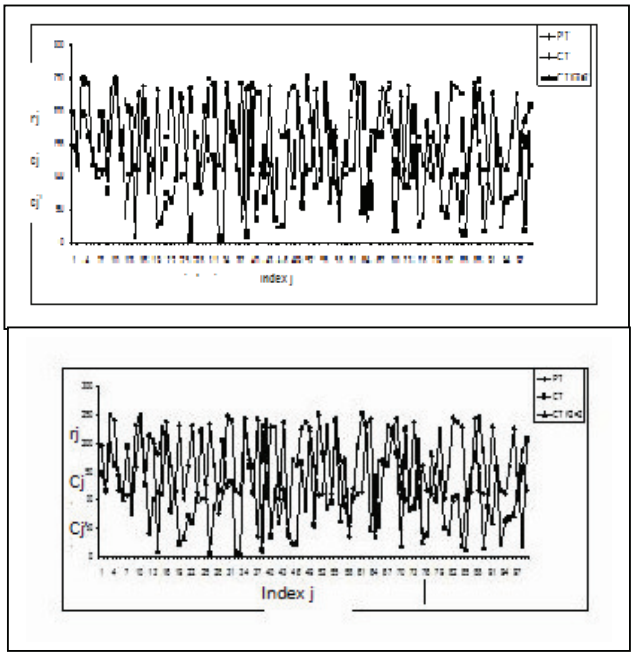


Figure 5.6 Plot of 100 plain text r_j, c_j ,c_j' for additive cipher



CONCLUSIONS

It is shown in this paper that RNS representation of integers based on CRT can be advantageously used in stream cipher

system. RNS representation of integers aids parallel processing in both encryption and decryption algorithms thus, reducing computational and time complexity. It is also seen that confusion and diffusion in encryption algorithm is inherently present. The modular arithmetic employed in the system, makes the system immune to algebraic attack. Immunity to side channel attack is obtained by parallel processing in the system. It is also shown by means of example that all the cipher texts occurs with almost equal probability, which provides immunity against cipher text only attack. The scheme also exhibits avalanche effect. By proper choice of number of stages of shift register and hence the period of key sequence, it is possible to make the scheme theoretically secure. The proposed scheme is compared with standard binary stream cipher A5/1 recommended for GSM. It is observed that the performance of proposed scheme and A5/1 are comparable. .

REFERENCES :

[1] Nicholas S. Szabo' and Richard I Tanaka " Residue Arithmetic and its Applications to Computer technology " Mc Graw Hill Publication 1967.

[2] Xueqin Jiang and Moon Ho Lee, "Large Girth Quasi-Cyclic LDPC Codes Based on the Chinese" IEEE Communication Letters, Vol. 13, No. 5, May 2009

[3] Markus Houenstein " A Computationally Efficient Algorithm For Calculating Loudness Patterns of Narrowband Speech.". IEEE Transactions On on Acoustics, Speech, and Signal Processing, 1997. ICASSP-97, 1997 , vol.2 page(s): 1311-1314

[4] Mathieu Ciet, Michael Neve, Eric Peeters and Jean Jacques "Parallel FPGA Implementation of RSA with Reduced number system – Can side channel threat be avoided ?" UCL Crypto group du Levant, 3 1348 Louvain-La-Neuva, Belgium

[5] Ahmad A. Hiasat, " High-Speed and Reduced Area Modular Adder Structure for RNS'' IEEE Transaction on Computers, Vol. 51, No.1 January 2008, pp 84 – 89.

[6] Shaoqiang Bi and Warren J.Gross "The Mixed Radix and Its Applications to Residue Comparison" IEEE Transaction on Computers, Vol. 57, No.12 December 2008, pp 1624 – 1632.

[7] William A. Chren, Jr."Low-Area Edge Sampler Using The Chinese Remainder Theorem" IEEE Transaction on Instrument and Measurement, Vol. 48, No. 4, August 1999, pp. 793-797.

[8] Ronald J. Cosentino " Fault Tolerance in a Systolic Residue Arithmetic Processor Array" IEEE Transaction on Computers,Vol.37, No.7 July 1988, pp 886-890.

[9] Sining Liu, Brian King and Wei Wang " A CRT-RSA Algorithm Secure Against Hardware Fault Attack" Proceeding of the 2nd IEEE Dependable, Autonomic and Secure Computing (DASC'06), 2006, pp.1-7.

[10] Yasuyuki MURAKAMI, Kiyoko Katayanagi and Masao KASAHARA "A new Class of Cryptosystems Based on The Chinese Remainder Theorem" International Symposium on Information Theory and Applications, ISITA 2008 Auckland, New Zealand, 7-10, December, 2008,

[11] Abdelhamid S. Abdelhamid and Ahmed A. Belel " Secure Transmission of Sensitive Data Using Multiple Channels" IEEE international Conference on Computer System Application 3rd ACS 6th January 2005.

[12] S. Halevi, D. Coppersmith and C. S. Jutla. Scream "A Software-E cient Stream Cipher. In Fast Software Encryption" - FSE 2002, pp 195 - 209, Lecture Notes in Computer Science, Springer-Verlag, 2002.

[13] G. G. Rose and P. Hawkes. Turing "A Fast Stream Cipher" In Fast Software Encryption FSE 2003, pages 290-306. Springer-Verlag, 2003.

[14] D. Watanabe, S. Furuya, H. Yoshida, K. Takaragi and B. Preneel. "A New Keystream Generator MUGI." In Fast Software Encryption, FSE 2002, pages 179-194. Springer-Verlag, 2002.

[15] Donald D Kunth "Art of Computer Programming Semi numerical Algorithm " Vol 2 Third Edition Pearson Education Inc. Publication. 2006

[16] Jerry Banks, John S. Carson II and David M. Nicol "Discrete-Event System Simulation" Prentice Hall, 2002.

[17] Thomas W.Hungerford " Algebra" Springer – Verlag, New York Inc. 2004

[18] Rueppl. A. " Stream Cipher" In Contemporary Cryptography N.Y. IEEE Press 1992.

[19] K.N. Hari Bhat and B.B. Amberker " Sequences of unit over ring Zm of integer modulo m" National Communication Conference (NCC), I.I.T.Kanpur,1995.

[20] B.B. Amberker " Modified Vigenere Cipher System" M.Tech. Thesis of Electronics and Communication K.R.E.C. now called as National Institute of Technology -1991.

[21] NEAL KOBLITZ "A Course in Number theory and Cryptography" Springer Verlog new York 1987 page 11

[22] Shannon C. " Communication Theory and Secrecy System" Bell System Technical Journal, no. 4 1949.

[23] Gilbert S. Vernam, "Cipher Printing Telegraph Systems For Secret Wire and Radio Telegraphic Communications", Journal of the IEEE, Vol 55, pp109-115 (1926)

[24] W T Penzhorn "Algebraic Attack on Cipher System" IEEE AFRICON 2004 pp 969 - 973

[25] Ju Young KIM and Hong Yeop SONG " A Nonlinear Boolean Function With Good Algebraic Immunity " IEEE Proceeding Of IWSDA '07, 2007, pp. 94-98.

[26] Mathieu Ciet, Michael Neve, Eric Peeters and Jean Jacques "Parallel FPGA Implementation of RSA with Reduced number system – Can side channel threat be avoided?" UCL Crypto group du Levant, 3 1348 Louvain-La-Neuva, Belgium

[27] Jean- Claude Bajard, and Laurent Imbert " A Full RNS Implementation of RSA " IEEE Transactions On Computers, Vol. 53. No. 6, June 2004, pp. 769-774.

[28] Gustavus J. Simmons "Contemporary Cryptology The Science of Information Integrity " IEEE Press 1992.

[29] Recommendation of GSM 02.09 European Telecommunication Standards institute (ETSI), Security aspects