Egyptian Informatics Journal 18 (2017) 141–149

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Full length article

Parametric comparison of EMDS algorithm with some symmetric cryptosystems

Mani Arora [a,*], Sandeep Sharma [b], Derick Engles [b]

[a] Khalsa College, Amritsar, Punjab, India
[b] Guru Nanak Dev University, Amritsar, Punjab, India

ARTICLE INFO

Article history:
Received 22 April 2016
Revised 20 October 2016
Accepted 10 November 2016
Available online 24 November 2016

Keywords:
EMDS
Algorithm
Data security
Encryption
Decryption
Reduce cipher
Throughput

ABSTRACT

Over the last decades owing to the incredible boost in the electronics industry and wireless technology, there has been an extraordinary outburst in the extent of digital data transmitted via the internet by means of handheld chic devices. The hefty amount of transmitted data requires data to be safe and sound in addition to transmission speed should be swift. In this document we have prepared qualitatively crypt-analysis of our proposed technique 'EMDS' and evaluated against it with further preferred symmetric algorithms. We have analyzed the diverse variety of symmetric algorithms by following the tangible approach and examined dissimilar parameters implicated. This document endows with estimation of ten of the majority of frequent algorithms. A contrast has been carried out connecting those algorithms and EMDS based on diverse parameters.

1. Introduction

Cryptography is considered necessary to solve tribulations concerning secrecy, authentication, integrity and fraudulent community. Owing to incredible boost in transmission of data over the networks by the side of security there is requirement to ensure on dimension of transmitted data so as to condense the utilization of bandwidth and memory space required to store data. If these factors will be restricted it will automatically enhance the pace of transmission [1]. There have been a number of suggestions to cut down the size of cipher text while providing security for asymmetric encryption however there have been extremely inadequate proposals of symmetric encryption cipher technique. EMDS algorithm anticipated by us gratifies the problem of huge storage space, amplified bandwidth, energy consumption, transmission speed and security. In this document preferred block cipher algorithms DES, triple DES, IDEA, Blowfish, CAST 128 as well as stream cipher algorithm RC2 and RC5 are examined simultaneously with EMDS algorithm. The performance of various symmetric algorithms is usually determined by using simulation or mathematical methods. In this paper we present the simulation results of various metrics used to measure the performance of proposed and existing symmetric cryptography algorithms. The performance of EMDS is evaluated by implementing the algorithm in Java

The purpose of this analysis is to

1. Compare EMDS with other symmetric algorithms.
2. Calculate the average cutback in cipher text size as compared to plain text size with reverence to EMDS algorithm

2. Various techniques for cryptography

In the present time, most symmetric encryption schemes are based on Fiestel network. It consist of number of rounds where each round includes bit shuffling, non linear substitutions (S-boxes) and exclusive OR operations. A short description of diverse Fiestel Network symmetric cryptographic algorithms is as follows.

2.1. Simplified Data Encryption Standard (SDES)

Simplified DES (S-DES) is an educational but not a secure block cipher algorithm. It was simplified version of DES formulated by

* Corresponding author.
E-mail addresses: mani_mcain@yahoo.com (M. Arora), Sandeep_gndu@yahoo.com (S. Sharma), derickengeles@yahoo.com (D. Engles).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.eij.2016.11.004

Professor Edward Schaefer of Santa Clara University. Encryption algorithm acquires an 8-bit block of plaintext along with a 10-bit key as input and generates an 8-bit block of cipher text as output vice versa decryption algorithm acquires 8 bit cipher text and a 10 bit key as input to create 8 bit original plain text [2].

Major operations of encryption algorithm

1. An initial permutation (IP).
2. An intricate function marked as fK, which engages both permutation as well as substitution operations furthermore it depends on a key input.
3. An effortless permutation function that switches (SW) the two halves of the data
4. The function fK yet again.
5. To end with a permutation function that is the contrary to the initial permutation (IP^{-1}).

S-boxes

S-boxes offer non linearity to algorithm. It makes use of fixed type (Non Key dependent) of S-boxes.

Number of S-boxes used: 02 (S0 and S1).

Number of operators

It employs merely a particular operator i.e. exclusive OR while computing function fK.

Number of Keys.

It employs merely 01 key of size 10 bits.

2.2. Data Encryption Standard (DES)

Frequently and extensively used of all encryption algorithms is data encryption standard. It was designed by IBM in the 1970s and implemented by the National Bureau of standards (NBS) [now the National Institute for Standards and Technology] in 1977 for commercial and unspecified government applications .It encrypts and decrypts 64 bit data at a time. The 56 bit cipher key is utilized for both encryption and decryption.[3] It is most generally used for conservative algorithm.

Major operations of encryption algorithm

1. An initial permutation IP.
2. Permutations and substitution functions are applied using XOR operator and S-boxes in 16 rounds.
3. An uncomplicated permutation function is implemented that substitute two halves i.e.32 bit data.
4. Contrary to initial permutation (IP^{-1})

S-boxes

Number of S-boxes used: 08(S0, S1–S8).

Number of operators

It uses only single operator i.e. exclusive OR.

Number of keys

It makes use of merely 01 key of size 56 bits.

2.3. Triple DES (3DES)

Formulated by Tuchman to preserve privacy and amalgamation of information illustrated by data during transmission or while in storage. Triple data encryption algorithm was chosen for applying in financial applications in ANSI standard X9.17.It is a three level put up using DES at each level to defend in opposition to savage drive attacks, exclusive of devising an utterly latest block cipher algorithm. In addition to it, Triple DES is put into operation with diverse number of keys i.e.

1. 3DES (2):-DES used thrice with two different keys.
2. 3DES (3): DES used thrice with three different keys.

2.3.1. 3DES(2)

It acquires 64 bit block of plain text and key of 112 bits as input and generates cipher text of 64 bits. Permutation and substitution functions are applied in 48 rounds.

Major operations of encryption algorithm

1. Encrypt using key K1.
2. Decrypt using key K2.
3. Resulted data is again encrypted using key K1.

S-boxes

Number of S-boxes used: 24.

Number of operators

It employs merely particular operator i.e. exclusive OR.

Number of keys

It exercises 02 keys.

2.3.2. 3DES(3)

Triple DES with two keys undergo meet in the middle attack so triple DES with three keys move towards subsistence. Despite the fact that it is more safe and sound but time-consuming than DES. It is applied by Federal organizations to protect receptive data. It captures 64 bit plaintext in addition to 168 bit key to produce cipher text of 64 bits in 48 rounds. Encryption course is an effort of Encryption and decryption process of single DES cipher.

Major operations of encryption algorithm

1. Encrypt data using DES with key K1 characterized as E_{K1}
2. Decrypt E_{K1} with key K2 by means of DES symbolized as D_{K2}
3. Yet again DES encryption is executing of D_{K2} with key K3.

Where K1, K2, K3 are independent keys.

S-boxes

Number of S-boxes used: 24.

Number of operators

It utilizes only single operator i.e. exclusive OR.

Number of keys

It uses 03 keys.

2.4. International Data Encryption Algorithm (IDEA)

The global Data Encryption Algorithm (IDEA) is a symmetric block cipher projected by Xuejia Lai and James Massey of ETH Zurich in 1991. It was anticipated to be a substitution for the Data Encryption Standard. Encryption algorithm acquires a 64-bits block of plaintext in addition to a 128-bit key as input along with it brings into being a 64-bits block of cipher text as output [4]. The strength of IDEA lies with utilization of XOR, binary addition and binary multiplication of 16-bit integers.

Major operations of encryption algorithm

1. Key Scheduling: Sub key generation algorithm is used to create 52 16bit sub keys as output.
2. Plaintext is disintegrated into four 16 bit sub blocks.
3. Three dissimilar categories of operations are executed on splitted data in 8 rounds.
   (i) Addition (+):-Addition of integers modulo 2^{16} with inputs and outputs treated as unsigned 16 bit integers.
   (ii) Multiplication Θ:-Multiplication of integers modulo $2^{16}+1$, with inputs and outputs taken care of as unsigned 16-bit integers excluding a block of all zeros is considered as representing 2^{16}.
   (iii) Bitwise exclusive OR ($\oplus$).

4. Output transformation: Again operators are used on interchanged data and sub keys to produce final output.

S-boxes
It use no S-box for substitution.
Number of operators
It utilizes three operators.

(i) Addition(+).
(ii) Multiplication Θ.
(iii) Bitwise exclusive OR(⊕).

Number of keys
Its utilization of 01 key.52 subkeys are originated from the primary128 bit key.

2.5. Blowfish

Blowfish Algorithm was formulated peculiarly by Bruce Schneier. Encryption algorithm acquires a 64-bits block of plaintext and an inconsistent length key as input and produces a 64-bits block of cipher text as output [5]. It is merely appropriate for applications where the key fails to modify frequently, similar to an automatic file encryption. This algorithm is one of secure traditional encryption algorithm for execution because both the sub keys and S boxes are formed by course of constant functioning of algorithm itself. Its key extent can be as long as 448 bits.

Major operations of encryption algorithm

1. Key Scheduling: Sub keys and S-boxes are produced .Key ranges from 32 bits to 448 bits is used to create 18 32bit sub keys and four S-boxes.
2. Plaintext divides into 32 bit halves.
3. Permutations and substitutions are performed using two primitive operations in 16 rounds.
   (i) Addition (+):-Addition of words is performed modulo 2^{32}.
   (ii) Bitwise exclusive OR(⊕)

S-boxes
Four S-boxes are used which are generated from key itself.
Number of operators
It uses two operators.

(i) Addition(+).
(ii) Bitwise exclusive OR(⊕).

Number of keys
It use 01 key.18 sub keys are crafting from the original key.

2.6. CAST-128

CAST 128 Encryption algorithm is pioneered by Carliste, Adams and Stafford Tavared of Entrust Technology in 1996 [6]. It has a fine obstruction to distinctive cryptanalysis, linear cryptanalysis and interrelated key cryptanalysis. It formulates the utilization of key size differing from 40 bits to 128 bits in augmentation of 8 bits each. It functions on 64-bits block of plaintext to generate 64-bits block of cipher text in 16 rounds .It makes use of two sub keys in each round: a 32 Ki_m and 56 bit Kr_{is} and the function F depends on the round.

Major operations of encryption algorithm

1. Key Scheduling: From key K, 16 pairs of sub keys are figured out.
2. Plaintext is disintegrated in two halves.
3. Permutations as well as substitution functions are applied by means of four operators in 16 rounds.

Four operators used are

(i) Addition (+):-Addition of words is performed modulo 2^{32}
(ii) Subtraction (−):-Inverse operation denoted by -.is performed modulo2^{32}
(iii) Bitwise exclusive OR(⊕)
(iv) Left circular rotation:-The cyclic rotation of word x left by y bits is indicated by x⋘y.

S-boxes
It uses 8 S-boxes, out of which 4 S-boxes are used in encryption and decryption process and 4 are used in sub key generation.

Number of operators
It uses four operators.

(i) Addition (+).
(ii) Subtraction (−).
(iii) Bitwise exclusive OR (⊕).
(iv) Left circular rotation (⋘)

Number of keys
It uses 01 key.16 pairs of sub keys are produced from the original key.

2.7. RC2

Developed by Ron Rivest. It is a conventional encryption algorithm. It is simple to employ on 16-bit microprocessor. It acquires input of 64 bit stored in the 16 bit words and fabricates an output of similar size i.e. of 64 bits. The variable key size is taken one byte up to 128 bytes.

Major operations of encryption algorithm

1. Sub keys are generated using sub key generation algorithm taking key as input.
2. Amalgamation and decocting of data is carried out in 18 rounds. It makes use of five primeval operations for amalgamation and decocting.

Six primitive operations are

(i) Addition (+):-Addition of words is performed modulo 2^{32}.
(ii) Subtraction (−):-Inverse operation denoted by - is performed modulo2^{w}.
(iii) Bitwise exclusive OR (⊕).
(iv) Bitwise complement (~).
(v) Bitwise AND (&).
(vi) Left circular rotation:-The cyclic rotation of word x left by y bits is denoted by x⋘y.

S-boxes
It use no S-box for substitutions.
Number of operators
It uses 6 operators.

(i) Addition(+).
(ii) Subtraction (−).
(iii) Bitwise exclusive OR (⊕).
(iv) Bitwise complement (~).
(v) Bitwise AND (&)
(vi) Left circular rotation (⋘).

Number of keys
It applies 01 key where sub keys are created from the original key.

2.8. RC5

Formulated independently by Rivest [7].The algorithm is very swift and compliant to the conductors of diverse word lengths. RC5 is word oriented. The fundamental functions work on full words of data at a time. The number of bits in a word is a parameter of RC5. The Key length being inconsistent thus provides a tradeoff flanked by security and speed. It is apt for devices of small memory. Owing to its trouble-free structure several cryptographers were concerned in cryptanalysis of it. RC5 encrypts blocks of plaintext of length 32, 64, or 128 bits into blocks of cipher text of the same length. The key length varies from 0 to 2040 bits. It acquires three parameters as input. (1) w (word size) (2) r (number of rounds) (3) b (number of bytes in encryption key K)

Major operations of encryption algorithm

1. Sub keys are produced by means of sub key generation algorithm taking key and number of rounds as input.
2. Plaintext disjunctioned in two halves using two sub keys and operation of addition.
3. Permutations and substitution function are employs by means of three primitive operations (and their inverses).

Three primitive operations (and their inverses) used are:

i) Addition:-Addition of words denoted by +, is executed modulo 2^w .The inverse operation indicated by -,is subtraction modulo 2^w.
ii) Bitwise exclusive-OR: This operation is represented by $\oplus$.
iii) Left circular rotation: The cyclic rotation of word x left by y bits is symbolized by $x \lll y$. The inverse is the right circular rotation of word x by y bits, denoted by $x \ggg y$.

S-boxes
It uses no S-box for substitutions.
Number of operators
It uses 5 operators

(i) Addition (+)
(ii) Subtraction (−)
(iii) Bitwise exclusive OR ($\oplus$)
(iv) Left circular rotation ($\lll$)
(v) Right circular rotation ($\ggg$)

Number of keys
It use 01 key where sub keys are spawn from the original key.

3. EMDS technique for cryptography

EMDS is a dictionary technique for encryption and decryption [8] .It is a modified version of MDS algorithm of encryption [9]. It centralizes on security in addition to a size of data. Its purpose is to condense the cipher text along with security constraints so that bandwidth consumption can be abridged. Furthermore, the less memory space will be requisite to store encrypted message at destination machine. EMDS algorithm is implemented in JAVA using the Net Beans IDE. In this algorithm two keys of diverse sizes are utilized to encrypt as well as decrypt the data to preserve a firm security. To a certain extent their sizes are minute however intruder can't decode the data till he is able to identify both keys. In this algorithm two dictionaries are used one static and one dynamic. Static dictionary will be accessible in encrypted mode to both the sender and destination machines .Barely authorized user can decrypt it and still made changes in it. Dynamic dictionary is being formed at runtime and is not transmitted from source to destination machine so that bandwidth utilization can be condensed.

Major operations of encryption algorithm

1. Read primary dictionary in memory.
2. Split data into blocks of 2048 words.
3. Substitutions are carried out using dictionaries.
4. Split data into blocks of 12 bits.
5. Operation by means of one operator is executed .The operator used is Bitwise exclusive-OR denoted by $\oplus$.

S-boxes
Instead of S-boxes, dictionaries are used for substitutions.
Number of operators
It uses 01 operator.

(i) Bitwise exclusive OR ($\oplus$).

Number of keys
It uses 02 keys while dictionary codes perform like sub keys.

3.1. Implementation

The above algorithm is implemented via java on core i5 processor M480 @ 2.67 Ghz, 4 GB RAM, 64 bit operating system. The observations are recorded in Table 1. The program acknowledges the plaintext as an input. After a thriving implementation the data is created in encrypted form which can also be supplementary decrypted. It also calculates the size of plaintext given by user and the size of cipher text generated in bytes. It also indicates the transmission speed of data and total encryption and decryption time taken. The snapshots of sender and receiver user interface are given in Figs. 1 and 2.

4. Result analysis and comparison

EMDS has been cautiously estimated for the following factors and similarity has been made with other encryption algorithms:

4.1. Cipher text size

From the experiment average 48% reduction is recorded by means of EMDS algorithm .In other cryptography techniques it is considered that chiefly the cipher text generated has size same that of plain text size or larger than it. To condense the size of cipher text a little hybrid algorithms are anticipated in which data compression method is used to lessen the size of plain text and then encrypt the data via encryption algorithm. On the other hand lot of overheads are concerned in it. Due to this we tried to reduce cipher text as well as to provide adequate security in one technique only named 'EMDS'. Table 2 and Fig. 3 signify plain text size and cipher text size of other cryptography algorithms and EMDS

4.2. Encryption key length

It means number of characters in the encryption key. All the block ciphers, which we have studied, are using bits in the key . So, when we talk about length of the key, it means number of bits in the key. E.g.: If Key K = 1010; Length = 4 bits.

Key lengths of these block ciphers have been divided into two categories.

Table 1
Analysis of EMDS technique.

Sr no	Ps	Cs	Tt	Et	Dt	Rd	Per (%)
1	22	15	10	14	33	7	32
2.	54	36	16	15	31	18	33
3.	106	64	21	64	49	42	40
4.	212	101	42	94	47	111	52
5.	338	169	49	112	82	169	50
6.	392	175	65	139	101	217	55
7.	664	305	90	125	94	359	54
8.	800	381	117	205	149	419	52
9.	932	406	120	156	95	526	56
10.	1430	636	149	220	158	794	56

Ps: Plain text size in bytes.
Cs: Cipher text size in bytes.
Tt: Transmission time in ms.
Et: Encryption time in ms.
Dt: Decryption time in ms.
Rd: Amount of data reduced in bytes.
Per: Percentage of data reduced.
So average reduction is $\sum$ per/10 = 48%.

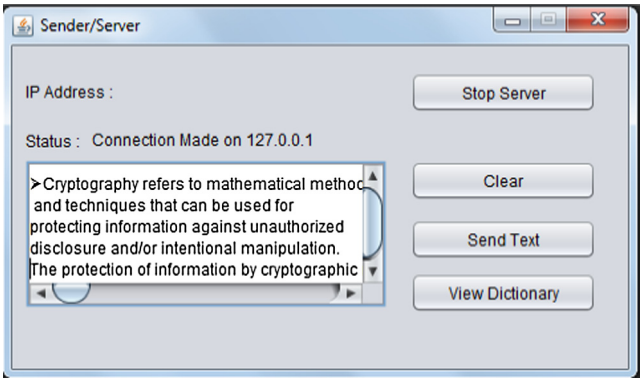

Figure 1. Sender/server.

1. Cipher with fixed key length.
2. Cipher with variable key length.

Following Table 3 will give Key lengths of most commonly used block ciphers and EMDS

If we contrast EMDS with above ciphers in terms of key length then there are two key features about it

1. EMDS is using two keys for encryption and decryption.
2. Length of two keys are permanent, one key is of 12 bits (depends on code length used in dictionary) and other key is of 7 bits.

4.3. Block size

Cryptographic strength (Security) $\propto$ Block Size.i.e. Larger the block size, secure the cipher.

While Encryption time is inversely proportional to Block sizei.e. Larger the block size more time it will take to encrypt data.

Block lengths of block ciphers have been separated into two categories.

1. Cipher with preset Block length (1–8, See table below).
2. Cipher with inconsistent block length (RC5 and EMDS).

Following Table 4 will give block size of most commonly used block ciphers and EMDS

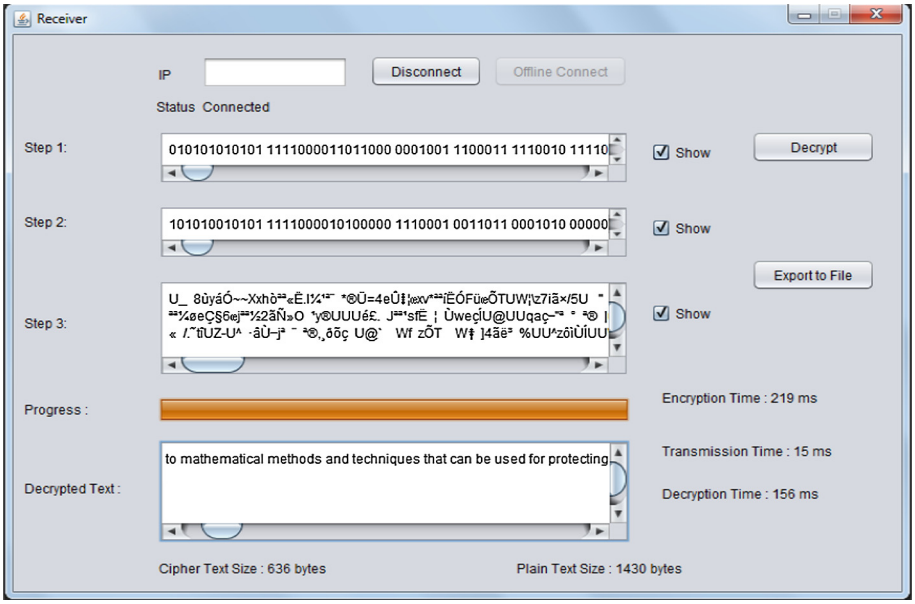

Figure 2. Receiver.

Table 2
Comparison of cipher text size produced by EMDS and other symmetric algorithms.

S. No.	Algorithm	Plain text size (in bits)	Cipher text size (in bits)
1.	SDES	8	8
2.	DES	64	64
3.	3DES(2)	64	64
4.	3DES(3)	64	64
5.	IDEA	64	64
6.	BLOWFISH	64	64
7.	RC5	64	64
8.	RC2	64	64
9.	CAST- 128	64	64
10.	EMDS	64	31

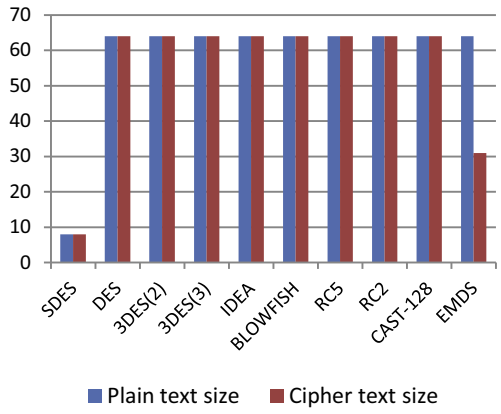

Figure 3. Analysis of cipher text size produced by EMDS and other symmetric algorithms.

Table 3
Analysis of Key length.

S. No.	Cipher (Block)	Key Length (in bits)
1.	SDES	10
2.	DES	56
3.	3DES(2)	112
4.	3DES(3)	168
5.	IDEA	128
6.	BLOWFISH	32–448
7.	RC5	40–2040
8.	RC2	8–1024
9.	CAST- 128	40–128
10.	EMDS	7and12

Table 4
Block size analysis.

S.No.	Cipher (block)	Block size (in bits)
1.	SDES	8
2.	DES	64
3.	3DES(2)	64
4.	3DES(3)	64
5.	CAST-128	64
6.	IDEA	64
7.	BLOWFISH	64
8.	RC5	32, 64, 128
9.	RC2	64
10.	EMDS	7, 12

In EMDS Algorithm data is divided into blocks of 7 bits and 12 bits prior to pertain mathematical function on it for final result. At the same time as cryptography strength is unswervingly proportional to block size so owing to this basis we have separated data primarily into huge blocks of 2048 words and afterwards for making algorithm further secure we divided a block into sub blocks of 12 bits and 7 bits.

4.4. S-boxes

S-boxes provides non-linearity (it means number of input bits ≠ number of output bits or output bits are not a linear combinations of input bits) to the encryption algorithm, In addition to it the S-boxes are used to obscure the relationship between cipher text and key. For example the S-box from DES (S_5) mapping 6 bit input to 4 bit output is shown in Table 5. So, we can say S-boxes offer security (Cryptographic strength) to the cipher.

Two types of S-boxes are there.

1. Fixed S-boxes (Non–Key dependent S-boxes) e.g. SDES, DES, 2DES, 3DES(2), 3DES(3), CAST-128.
2. Variable S-boxes (Key dependent S-boxes) e.g. BLOWFISH.

Note:-

1. Variable S-boxes provides more security.
2. S-boxes are the mainly complicated phase of any cipher.

Table 5 shows the structure of S-box used in DES

We have studied 09 block ciphers, out of which 6 ciphers have S-boxes.

Table 6 will show information on S-boxes of block ciphers

RC5, RC2 and IDEA have no S-boxes as fundamental objective of S-boxes is to generate confusion property, This property will be produced in these cipher by primeval operations. Likewise, in EMDS algorithm there are no S-boxes and the confusion property is attained by using two dictionaries The output given by dictionary is in non linear form in contrast to input and afterwards exclusive OR operation is carried out to attain the confusion property.

4.5. Number of operators involved

Cryptographic strength (Security) $\propto$ Mixed Operators.Security of cipher depend on number of operators and their strength, more the number of operators more secure the cipher. The use of more than one arithmetic and/or Boolean operator complicates cryptanalysis, especially if these operators do not satisfy distributive and associative laws.

Following ciphers use mixed operators.

1. IDEA
2. BLOWFISH
3. RC2
4. RC5
5. CAST-128

Following don't use mixed operators.

1. SDES
2. DES
3. 3DES(2)
4. 3DES(3)

Following Table 7 will give information regarding operators.

EMDS algorithm doesn't employ varied operators. It use only one operator i.e. exclusive OR. In EMDS if we will use mixed operators, it will obscure the technique so it will influence the turnaround time of the algorithm.

4.6. Number of keys

It means number of encryption keys used to secure data.

Table 5
The structure of S-box.

S6		Middle 4 bits of input														
		0000	**0001**	**0010**	**0011**	**0100**	**0101**	**0110**	**0111**	**1000**	**1001**	**1010**	**1011**	**1100**	**1101**	**1110**	**1111**
Outer bits	**00**	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

Table 6
Analysis of S-boxes.

S.No.	Cipher (Block)	Number of S-boxes
1.	SDES	2
2.	DES	8
3.	3DES(2)	24
4.	3DES(3)	24
5.	IDEA	No S-box
6.	BLOWFISH	4
7.	RC5	No S-box
8.	RC2	No S-box
9.	CAST-128	8
10.	EMDS	No S-box

Table 7
Analysis of number of operators.

S.No.	Cipher (Block)	Number Of operators
1.	SDES	1
2.	DES	1
3.	3DES(2)	1
4.	3DES(3)	1
5.	IDEA	3
6.	BLOWFISH	2
7.	RC5	5
8.	RC2	5
9.	CAST-128	4
10.	EMDS	1

Table 8
Analysis of number of keys.

S. No.	Cipher (Block)	Number of keys
1.	SDES	01
2.	DES	01
3.	3DES(2)	02
4.	3DES(3)	03
5.	IDEA	01
6.	BLOWFISH	01
7.	RC5	01
8.	RC2	01
9.	CAST- 128	01
10.	EMDS	02

Cryptographic strength (Security) $\propto$ Number of keys i.e. more number of keys more secure the data. All the block ciphers, which we have studied, are using 01 key.

Following Table 8 will give number of Keys used by most commonly used block ciphers and EMDS

4.7. Encryption time

It's the time taken to encrypt the plaintext. The encryption time has major role in the efficiency of any cryptography technique. Cryptography efficiency is inversely proportional to encryption time i.e. less the encryption time taken by algorithm to encrypt data, more efficient will be the algorithm. On comparing results of encryption time taken by EMDS and some prominent algorithms, it has been observed that EMDS outperformed other algorithms. Results are obtained by observing encryption time taken by algorithms for same size of plain text as shown in Table 9 Also

Table 9
Analysis of Encryption time.

Input plain text size in (K bytes)	DES	3DES	AES	RC5	RC2	Blowfish	EMDS
51	31	56	58	45	61	38	15
61	36	58	35	27	63	39	17
100	50	82	92	60	92	41	40
254	52	115	116	81	126	86	80
330	92	178	156	110	176	90	91
696	365	224	215	126	266	127	125
902	238	229	261	166	272	166	156
976	260	302	211	128	306	168	127
5456	1298	1566	1246	697	1581	220	219
7315	1735	1787	1466	758	1916	238	235
Average time	415.7	459.7	385.6	219.8	485.9	121.3	110.5
Throughput (Kbytes/ms)	3.882	3.511	4.185	7.343	3.321	13.307	14.607

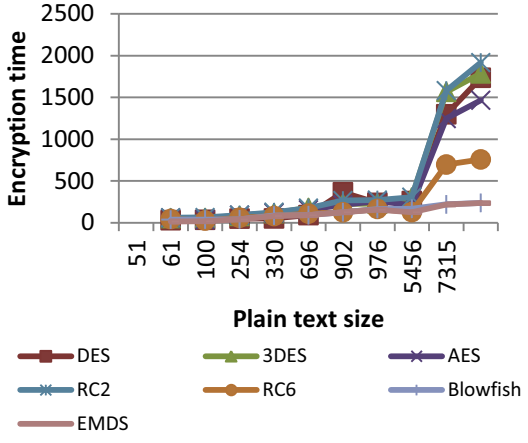

Figure 4. Comparison of encryption time taken by EMDS and other symmetric algorithms.

Table 10
Analysis of Decryption time.

Input size in (K bytes)	DES	3DES	AES	RC5	RC2	Blowfish	EMDS
51	48	56	65	36	66	40	30
61	36	58	35	27	63	39	41
100	50	82	92	60	92	41	40
254	52	115	116	81	126	86	51
330	92	178	156	110	176	90	79
696	365	224	215	126	266	127	96
902	238	229	261	166	272	166	110
976	260	302	211	128	306	168	112
5456	1298	1566	1246	697	1581	220	210
7315	1735	1787	1466	758	1916	238	235
Average time (ms)	417.4	459.7	386.3	218.9	486.4	121.5	100.4
Throughput (K bytes/ms)	3.867	3.511	4.178	7.373	3.318	13.284	16.076

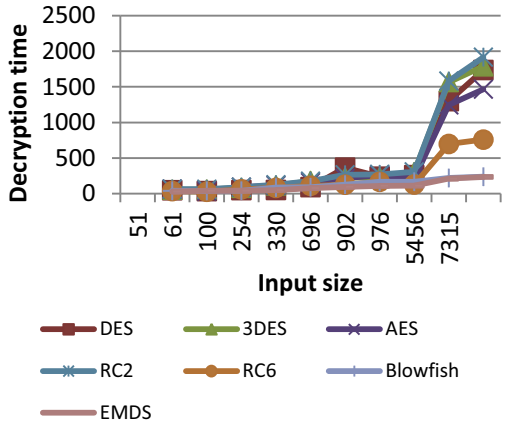

Figure 5. Comparison of decryption time taken by EMDS and other symmetric algorithms.

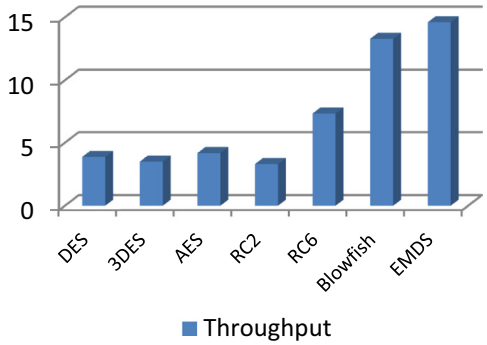

Figure 6. Comparison of throughput of encryption algorithm of EMDS and other symmetric algorithms.

from Fig. 4 it can be depicted that EMDS has taken less encryption time as compared to other algorithms so it is more efficient.

4.8. Decryption time

Decryption time is the time taken by algorithm to decrypt the cipher text at receiver side. The efficiency of decryption algorithm is inversely proportional to decryption time taken by an algorithm i.e. less the decryption time taken to decrypt the cipher text, more efficient will be the algorithm. The decryption time is determined by executing decryption algorithms of some prominently used symmetric algorithms and comparison among them and EMDS is done as shown in Table 10. From Fig. 5 we can conclude that EMDS takes less decryption time as compared to other symmetric algorithms so it is more efficient than other symmetric decryption algorithms.

Table 11
Analysis of throughput of encryption.

Algorithm	Throughput (kb/ms)
DES	3.882
3DES	3.511
AES	4.185
RC5	7.343
RC2	3.321
Blowfish	13.307
EMDS	14.607

4.9. Throughput of encryption algorithm

Performance of the algorithm can be determined through its throughput. Performance $\propto$ Throughput i.e. more will be the throughput of the algorithm, higher will be its performance. Throughput of encryption algorithm is calculated as follows:

$$Throughout = \frac{Size\,of\,plain\,text}{Encryption\,time}$$

Throughput is calculated from the analysis done for encryption time. It can be observed from Table 11 and Fig. 6 that throughput of EMDS encryption algorithm is more than the others.

5. Conclusion

The performance evaluation consequences show that

- The existing symmetric algorithms principally divide data into blocks of 64 bit size prior to performing computation on it. In EMDS algorithm data is divided into small blocks of 7 bits and 12 bits to recover the speed of encryption as well as decryption. This will also augment its security as data is divided into two diverse types of blocks.
- In EMDS algorithm no S-box is used for substitution as a replacement for dictionaries are used to accomplish confusion property. The output specified by two dictionaries in technique is nonlinear. This endows with more cryptographic strength to cipher.
- Analysis of existing symmetric methods signifies that cipher text formed by them is of same size as that of plain text. The size of cipher text is formed by projected encryption technique is abridged by using dictionary technique. The estimated results

show that the proposed EMDS algorithm is proficient to lessen text by 50% which accommodates to the tributary of immense storage space, augmented bandwidth, energy utilization as well as protection.

- The security of EMDS has been enhanced by using two private keys for encryption as well as decryption. The analysis of previously active algorithms show that generally the cipher use single key however to attain additional security DES like algorithms were modified as triple DES which utilize 02 or 03 keys .The key length of keys used in EMDS is set aside minute as encryption speed of any algorithm is contrariwise proportional to key size so encryption speed of EMDS is extra owing to this factor.
- The estimated result shows that EMDS take a lesser amount of time to encrypt and decrypt data as measured up to other algorithms. It has been initiated to be enhanced among existing algorithms in terms of encryption as well as decryption. It is also shown that the proposed algorithms give additional throughputs in contrast to other algorithms.

References

[1] Lv Chuanfeng, Zhao Qiangfu. Integration of data compression and cryptography: another way to increase the information security. AINA workshops 2007, vol. 2. p. 543–7.
[2] Garg Poonam. A comparison between Memetic algorithm and Genetic algorithm for the cryptanalysis of Simplified Data Encryption Standard Algorithm. Int J Network Secur Appl (IJNSA) 2009;1(1):34–42.
[3] Singh Sambir, Maakar Sunil K, Kumar Sudesh. Enhancing the security of DES algorithm using transposition cryptography techniques. Int J Adv Res Comput Sci Softw Eng 2013;3(6):464–71.
[4] Khovratovich D, Leurent G, Rechberger C. Narrow-bicliques: cryptanalysis of full IDEA. Advances in cryptology – EUROCRYPT 2012 lecture notes in computer science, vol. 7237. Springer-Verlag; 2012.
[5] Schneier B. The blowfish encryption algorithm. Dr. Dobb's J 1994;19(4):38–40.
[6] Adams Carlisle M. Constructing symmetric ciphers using the CAST design procedure. Des Codes Crypt 1997(12):283–316.
[7] Rivest RL. The RC5 encryption algorithm. In: Proceedings of the second international workshop on fast software encryption (FSE); 1994. p. 86–96.
[8] Arora Mani, Sharma Sandeep, Engles Derick. Efficient key mechanism and reduced cipher text technique for secured data communication. International journal of systems, control and communications, vol. 7. no. 2. Inderscience publisher; 2016. p. 186–96.
[9] Arora Mani, Engles Derick, Sharma Sandeep. MDS algorithm for encryption. J Comput Sci 2015;11(3):479–83.