MSEA: Modified Symmetric Encryption Algorithm

Rajul Kumar[1], K.K. Mishra[2], Ashish Tripathi[3], Abhinav Tomar[4], Surendra Singh[5]

{raj35mit[1], ashish.mnnit44[3], profession.abhinav[4], rathorsurendra.jec[5]}@gmail.com, kkm@mnnit.ac.in[2]

Motilal Nehru National Institute of Technology Allahabad
Allahabad, India

Abstract: In this article, a new symmetric block cipher named MSEA is proposed. MSEA is based on ARX cryptographic design technique. MSEA is simple in nature due to the use of combinations of elementary operations like modular addition, bit-wise rotation and bit-wise XOR. In MSEA, plain text block, secret key, and number of encryption rounds are variable in size, while the size of cipher text is double of size of plain text. Data-dependant rotation is the most vital feature of MSEA through which the unpredictability of encrypted text is increasing. Key formation and encryption/decryption schemes of MSEA are significantly fast.

Index Terms: Symmetric Key Cryptography, Data dependant rotations, Block Cipher, ARX Design Technique, MSEA

1. Introduction

In this Internet era, whenever any confidential and sensitive information transmitted over a public channel, there is possibility that an eavesdropper could intercept this message and steal this sensitive information. To protect these modern-day transmissions, the principles of cryptography are used. Cryptography is the study of transmitting secure messages and the art of secret writing by which the sensitive information may be prevented from any adversary. The general concept behind the cryptography is that the sender selects a message that he would like to transmit, applies some sort of encryption process, and transmits this encrypted message across the channel. The receiver obtains the encrypted message, also referred to as a cipher text, uses a known decryption process to recover the sender's original message. If

an adversary intercepts an encrypted message, he will be unable to recover the original message without knowledge of the secret decryption process.

There are various types of cryptographic techniques available for securing the sensitive information based on symmetric key cryptography and public key cryptography. In symmetric key cryptography, sender and receiver use a shared key for encryption and decryption, known as secret key. DES, 3-DES, AES, IDEA, RC4 and RC5 are some of the most famous symmetric key algorithms. In public key cryptography, sender uses public key of receiver, known to everyone, to encrypt the message and receiver uses his private key, known only to him, to decrypt the message. RSA is the one of the most famous public key algorithm which is based on Diffie-Hellman Key Exchange [9].

In this article, MSEA is proposed to provide flexibility to user according to his needs of security level with more secure and fast implementation. To achieve these goals, MSEA have following characteristics:

- ÷ It is a symmetric block cipher.
- ÷ It is simple in nature due to the use of combinations of elementary operations like modular addition, bit-wise rotation and bit-wise XOR.
- ÷ It is iterative in nature, having variable number of rounds for encryption. User can use number of rounds as per security level needed.
- ÷ It uses variable size plain text message block for encryption. User can choose block size according to his need and desired level of security.
- ÷ It uses variable size key for encryption process which directly depends on the size of plain text message block.
- ÷ Data dependent rotations are used heavily in encryption, decryption, and key generation process of MSEA by which the cryptographic strength is increased.

This article is organized as follows. Literature review is covered in section 2. Proposed algorithm, which includes key formation and the encryption/decryption schemes, is given in section 3. Analysis is given in section 5. Finally, the conclusion is drawn in section 5.

2. Literature Review

Lai and Massey [10] introduced a proposal for a new block encryption standard named IDEA which is based on ARX design with three different group operations modular addition, bitwise XOR and modular multiplication to achieve strong confusion and diffusion.

IDEA is vulnerable due the problem of weak keys. Biryukov et al. [1] introduced new weak key classes of IDEA. In 2007, Biham et al. [4] presented a new attack against 6-round (reduce) IDEA. Khovratovich et al. [3] introduced narrow-bicliques cryptanalysis of full IDEA which is a type of meet in the middle attack.

Gonzalo et al. [5] introduced a new block cipher algorithm named Akelarre which is also based on ARX design and its overall structure is same as of IDEA instead of 16 bit sub-block, fixed length key and fixed number of rounds as in IDEA, it uses 32 bit sub-block, variable key length and variable number of rounds. It used data dependent rotations instead of modular multiplication.

In 1997, Ferguson and Schneier [7] presented the cryptanalysis of Akelarre in which they conclude that Akelarre is disappointingly weak. They have shown that the round function of Akelarre preserve the parity of input and insecure against chosen plain text attack. They also conclude that the 31- bits information about master key can be recover using trivial methods from improved key schedule of a new version of Akelarre [6].

Hong et al. [2] introduced a 128-bit block cipher for fast encryption on common processor named LEA which is based on ARX design technique. Its key schedule uses several constants with modular addition and rotations to generate round keys. LEA is iterative in nature have different number of encryption rounds for different block size. It is faster than DES, AES and various existing algorithms.

3. Proposed Algorithm

MSEA is based on ARX cryptographic design technique. Only elementary operations like rotation, modular addition and bit-wise XOR are used to develop the proposed algorithm. MSEA provides flexibility to user to choose plain text block size and number of encryption rounds. Data dependent rotations are used heavily in encryption, decryption, and key generation process of MSEA. The concept of data dependent rotations is taken from RC5 algorithm [8]. MSEA key formation, encryption, and decryption are describes in this section.

In this article following parameters are used as input for algorithm:

r: Number of encryption rounds,

s: Size of the plain text message block,

k: Key for two-phase swap function,

$\mathbf{K_M}$: Master Key, used to generate round keys,

$\mathbf{K_1 \ldots . K_r}$: Round Keys for encryption and decryption process

In this proposed algorithm following symbols are used:

R: Data dependent rotations

⊕: Bit-wise XOR operation

<<<: Bit-wise Left Rotation

>>>: Bit-wise Right Rotation

+: Modular Addition

~: One's Complement

3.1 Key Formation

In MSEA, following types of keys are used:

Swap Key (k): Swap key is variable in size. Its size is $\log_2 s$ bits, where s is the size of plain text message block. For example, 7-bit swap key is used for 128 bit plain text message block.

Master Key (K_M): Master key is used to generate the round keys for MSEA encryption and decryption process. Size of master key is 2s bits. For example, 256-bit master key is used for 128-bit plain text message block.

Round Keys ($K_1....K_r$): In MSEA, numbers of round keys are used on the basis of parameter r, where r represents the number of rounds in encryption and decryption process. For every round there is a unique round key. Size of each round key is same as of master key.

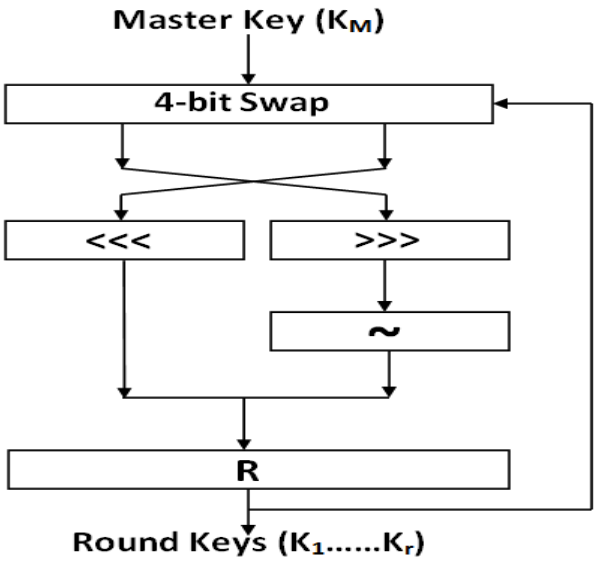


Figure 3.1: Round keys generation process in MSEA

Generation of Round Keys: Round keys generation is an iterative process in which each iteration gives a round key as its output. Number of iterations in round keys generation process are same as number of round functions in MSEA encryption process. Master Key K_M is the input for first iteration of generation process and the output of generation process is being served as the input for next iteration. As shown in figure 3.1, Master key K_M is been input and 4-bit swap applied on it. 4-bit swap function swaps each block of 4 consecutive bits from its next 4 consecutive bits. A heavy set of rotations are applied after 4-bit swap followed by complement. At last, data dependent rotation (R) is applied on resultant which is based on $\log_2$ s least significant bits of the resultant in which the first bit represents left or right rotations and

remaining bits represent number of rotations. The output of this iteration is a round key and also treated as the input for next iteration.

Figure 3.2 shows the cumulative key for MSEA which can be shared between sender and receiver using any key distribution scheme. It has four parts in which first part, 6-bit length, shows the number of round functions used in encryption and decryption process. In MSEA, minimum 1 and maximum 63 rounds can be possible. Second part, 12-bit length, represents the size of plain text block, which must be in multiple of 8, not smaller than 128 bits and not greater than 2048 bits. If plain text size is smaller than chosen block size, some padding bits are added into plain text to make its size same as chosen block size. Third part and fourth part, swap key and master key respectively, are described above in this section.

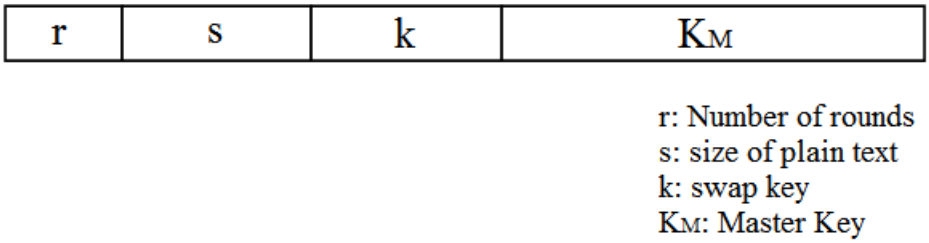


Figure 3.2: Cumulative key format

The size of swap keys, master keys and cumulative keys for some message blocks are shown in table 3.1.

Table 3.1: Cumulative key size for respective message block size (in bits)

MSEA Block Size	Swap Key Size	Master Key Size	Cumulative Key Size
128	7	256	281
192	7	384	409
216	7	432	457
256	8	512	538
384	8	768	794
512	9	1024	1051
728	9	1456	1483
936	9	1872	1899
1024	10	2048	2076
1384	10	2768	2796
1712	10	3424	3452
2048	11	4096	4125

3.2 MSEA Encryption

MSEA encryption is processed in three parts; firstly message expansion, then round functions, and finally two-phase swap. The round function is the core part of MSEA encryption. The overall scheme of MSEA encryption is shown in figure 3.3.

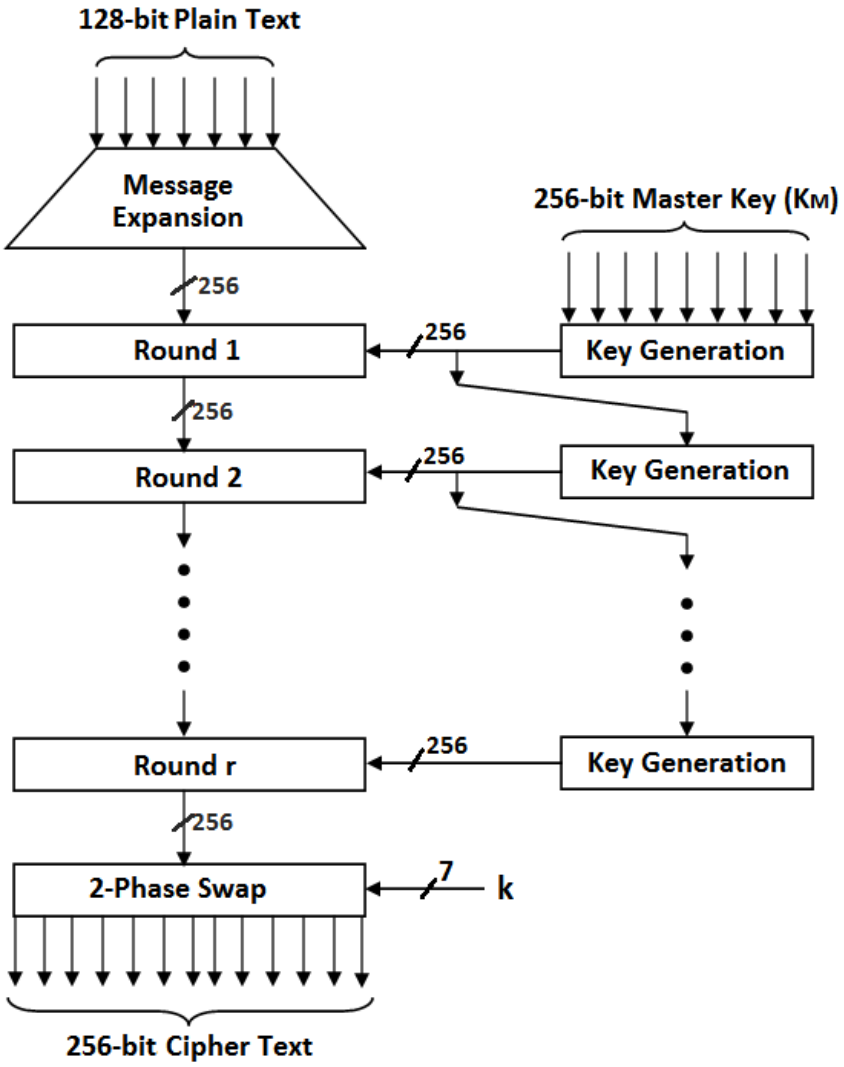


Figure 3.3: Block diagram of MSEA encryption for 128-bit plain text.

3.2.1 Message Expansion

In this phase, the plain text message block is expanded. Each 8-bit sub-block of input message block is expanded into 16-bit sub-block, rotated and

complemented. As shown in figure 3.4, 8-bit sub-block 10010011 is expanded, by appending 4 least significant bits at MSB side and appending 4 most significant bits at LSB side, into 16-bit sub-block 0011100100111001. As already describes that data dependent rotations are used in MSEA, the 16-bit sub-block is rotated in two phase. In first phase, rotation is performed according to last 4 bits in which MSB represents left or right rotation, 0 means right rotation and 1 means left rotation, and last 3 bits represent number of rotations. The remaining bits are rotated according to these 4 bits. In second phase, same process is performed but in it 4 starting bits are used instead of last 4 bits. For example in figure 2.4, the 16-bit sub-block is firstly rotated according to 1001, last 4 bits of sub-block, in which MSB is 1 means left rotation and remaining bits represent 1, in decimal, rotation then the resultant 16-bit sub-block after first phase rotations is 0111001001101001 after it the second phase rotation performed according to 0111, 4 starting bits of resultant sub-block, and the result is 01111101001101001. The result of second phase rotation is followed by complement operation. The above described process is also applied to all remaining 8-bit sub-blocks of input message block.

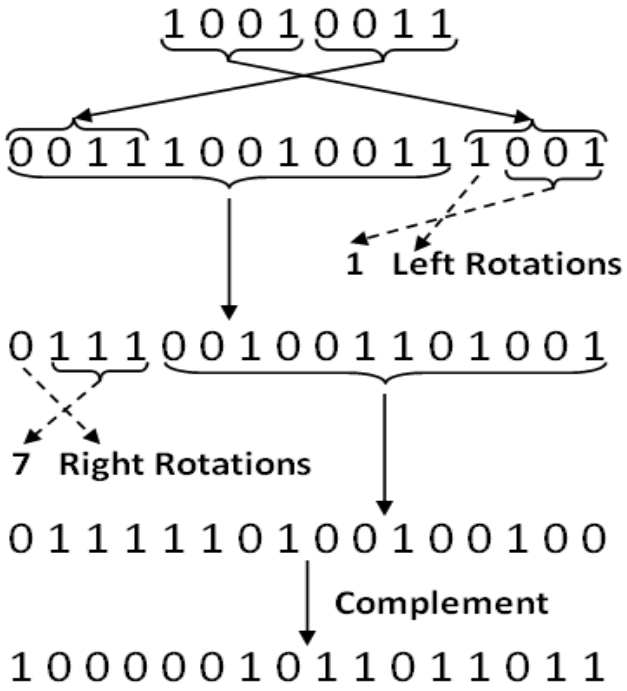


Figure 3.4: Message expansion phase in MSEA.

3.2.2 Round Function

The pseudo code for single round is given below:

1. divide the message block into four equal parts i.e. A, B, C, D
2. D:= C + D
3. C:= B + C
4. B:= A + B
5. A:= D + A
6. rotate A, B, C, D
7. divide Round Key into four equal parts i.e. K_{1a}, K_{1b}, K_{1c}, K_{1d}
8. A:= A $\oplus$ K_{1a}
9. B:= B $\oplus$ A $\oplus$ K_{1b}
10. C:= C $\oplus$ B $\oplus$ K_{1c}
11. D:= D $\oplus$ C $\oplus$ K_{1d}
12. concatenate C and A, let suppose E
13. concatenate D and B, let suppose F
14. F:= E $\oplus$ F
15. E:= E + F
16. concatenate F and E

The structure round function is partially same as of IDEA [10] and Akelarre [5][6]. Its input and output are double in size in comparison of plain text i.e. for 128-bit plain text block; its input and output have 256-bit. The message block is divided into four equal size sub-blocks and the modular addition scheme is applied in which a sub-block is added into other sub-block under a specified modulo. As shown in figure 3.5, sub-block C is added to D; B is added to C; A is added to B and resultant D is added to A. After it data dependant rotations are performed on each sub-block which are based on last $\log_2$ m bits of respective sub-keys, (i.e. K_{1a}, K_{1b}, K_{1c}, K_{1d}), where m is the number of bits in each sub-key. Two XOR operations are used for each sub-block besides first sub-block. As shown in figure 3.5, sub-block A is been XOR with K_{1a} but sub-block B is been XOR with resultant A and again with K_{1b}, sub-block C is been XOR with resultant B and again with K_{1c}, and finally sub-block D is been XOR with resultant C and again with K_{1d}. The resultant sub-blocks A, B, C, D are combined into two equal size sub-blocks, C attached into A and D attached into B. Second sub-block, let suppose F, is been XOR

with first sub-block, let suppose E, and resultant F is added to E then both sub-blocks are combined into one block as output of single round function.

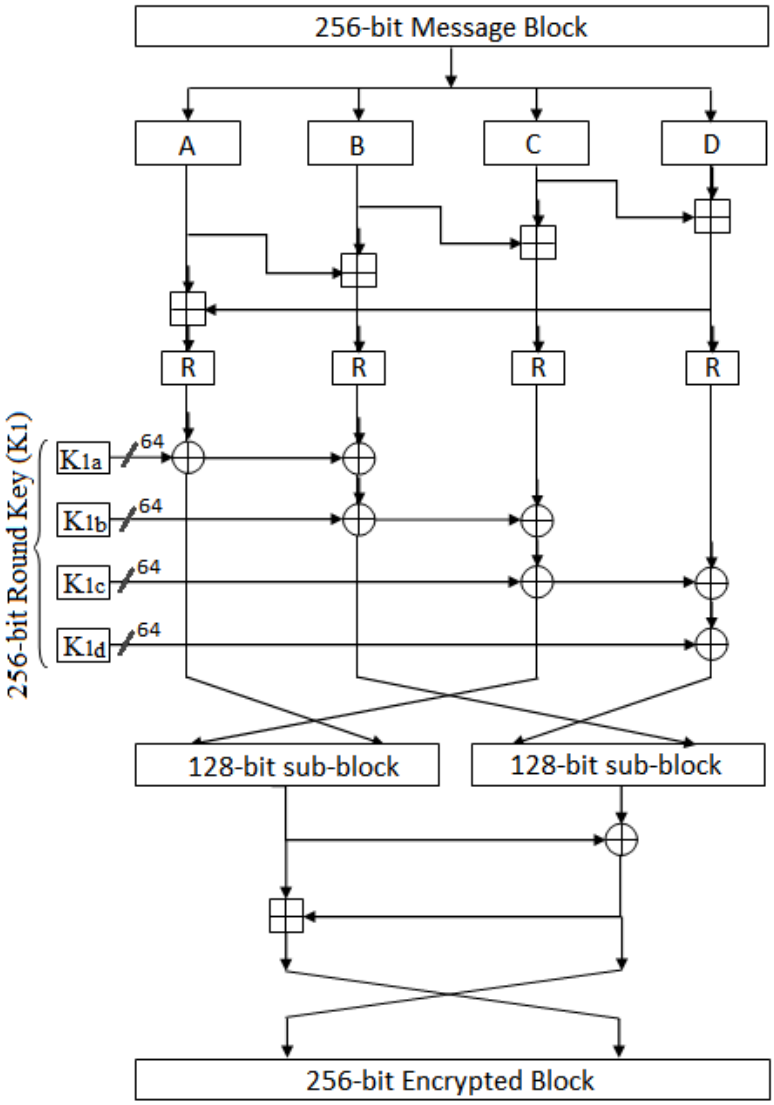


Figure 3.5: Single round of MSEA for 128-bit plain text.

3.2.3 Two- Phase Swap

In this function, two type of operations are used on the basis of swap key k, which is been input by user.

a. Swap every k^{th} bit of message block from 0 to 1 and from 1 to 0.

b. Swap every k bits of message block from next k bits.

As shown in figure 3.6, a 16-bit message block 1000110001111100 is been input to swap function with swap key 011, in decimal 3. In first process every 3rd bit of input is swap from 0 to 1 and 1 to 0. In second process each 3-bit sub-block of message is been swap with next 3-bits.

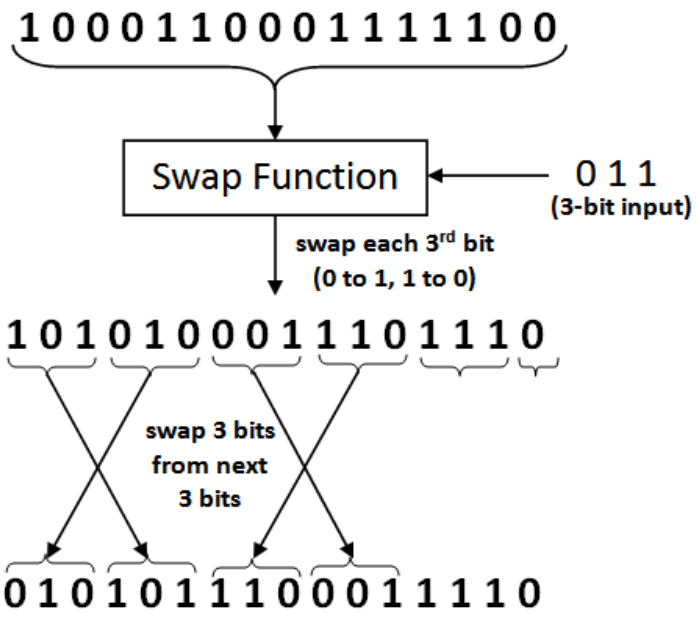


Figure 3.6: Swap function in MSEA

3.3 MSEA Decryption

The decryption routine can be derived directly from MSEA encryption. MSEA decryption is purely reverse function of its encryption process. Means firstly swap function is performed followed by reserve round function and after it message contraction is performed. The round keys are also is been input in reverse order. Thus, the complexity of decryption process is as same as encryption process.

4. Analysis

In this section, the proposed algorithm, MSEA is analyzed. The analysis is computed on 128-bit message block for which 281-bit cumulative key is used which produced 256-bit round keys. The analysis is taken place in following environment:

÷ Compiler: gcc version 4.5

÷ Simulated Processor: Intel Pentium (R) Dual CPU T3200 @ 2.00 GHz

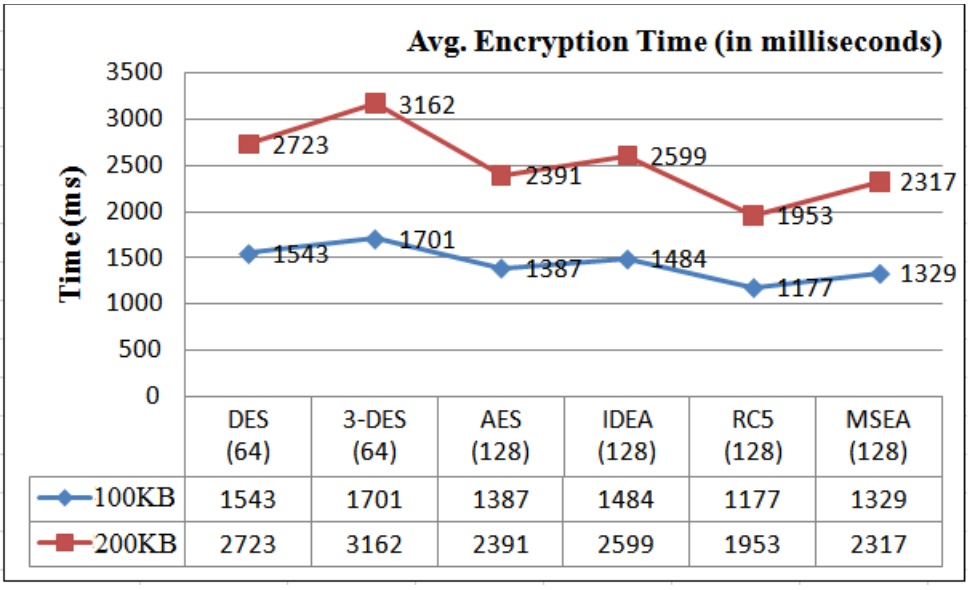
Due to use of ARX design, MSEA is simple in nature and secure form timing attacks. In MSEA, addition and data dependant rotations provides strong diffusion by creating the nonlinearity in message.

In MSEA variable rounds can be used. According to our results, 9 rounds are enough to produce strong avalanche effect. For one bit change in plain text, MSEA changes more than 50% bits of cipher text after 9 rounds.

MSEA rounds do not preserve the parity of input, thus the attacks which are based on parity relationship of input-output cannot be possible on MSEA. Thus MSEA is more secure than Akelarre.

Data dependent rotations, used in MSEA, create problems for linear and differential cryptanalysts because of rotating message bits randomly to random amount. According to our results, 12 rounds are sufficient to affect each and every bit of message.

For the sake of more security, we choose 18 as a default value for number of rounds when using 128-bit block of MSEA.



	DES (64)	3-DES (64)	AES (128)	IDEA (128)	RC5 (128)	MSEA (128)
100KB	1543	1701	1387	1484	1177	1329
200KB	2723	3162	2391	2599	1953	2317

Figure 3.7: Average Encryption Time of various algorithms for 100KB data.

MSEA encryption time is compared with DES, 3-DES, AES-128, IDEA-64 and RC5-128. Various 100KB and 200KB plain texts are taken and average encryption time is computed for all above listed algorithms. The average encryption time is measured in milliseconds (ms). As shown in figure 3.7, the encryption time of MSEA is less then DES, 3-DES, AES-128, IDEA-64 and more than RC5-128. It shows that MSEA is significantly fast.

5. Conclusion

We proposed a new symmetric key encryption algorithm, MSEA, which provides flexibility to user to choose plain text block size and number of rounds for encryption process. MSEA used elementary operations like modular addition, rotations, XORs and complement due to which structure of proposed algorithm is simple in nature. Data dependent rotation is the most vital feature of MSEA which create strong diffusion on plain text. Our analysis shows that MSEA is significantly faster than some existing encryption algorithms.

References

1. Biryukov, J. Nakahara Jr, B. Preneel, J. Vandewalle, “New Weak-Key Classes of IDEA”, *4th International Conference Information and Communications Security*, ICICS 2002, Lecture Notes in Computer Science 2513, Springer-Verlag, 2002, pp. 315-326.
2. D. Hong, J. K. Lee, D. C. Kim, D. Kwon, K. H. Ryu, and D. G. Lee, “LEA: A 128-Bit Block Cipher for Fast Encryption on Common Processors”, WISA 2013, LNCS 8267, Springer International Publishing Switzerland 2014.
3. D. Khovratovich, G. Leurent, and C. Rechberger, “Narrow-Bicliques: Cryptanalysis of Full IDEA”, *Advances in Cryptology*, EUROCRYPT 2012, LNCS 7237, Springer-Verlag, 2012, pp. 392–410.
4. E. Biham, O. Dunkelman, and N. Keller, "A New Attack on 6-Round IDEA", *Proceedings of Fast Software Encryption*, Lecture Notes in Computer Science, Springer-Verlag, 2007.

5. G. Álvarez, D. de la Guía, F. Montoya, and A. Peinado, “Akelarre: a new Block Cipher Algorithm”, *Third Annual Workshop on Selected Areas in Cryptography*, SAC 96, Kingston, Ontario, 15-16 August 1996, pp. 1-14.
6. G. Álvarez, D. de la Guía, F. Montoya, and A. Peinado, “Description of the new Block Cipher Algorithm Akelarre”, http://www.iec.csic.es/~fausto/papers/akelarre1.ps
7. N. Ferguson and B. Schneir, “Cryptanalysis of Akelarre”, 23 July 1997.
8. R. L. Rivest, “The RC5 Encryption Algorithm”, *Proceedings of the Second International Workshop on Fast Software Encryption*, 1994, pp. 86-96.
9. W. Diffie and M. Hellman, “New directions in cryptography”, *IEEE Transaction on Information Theory*, 1976, pp. 644–654.
10. X. Lai and J. Massey, “A Proposal for a New Block Encryption Standard”, *Advance in Cryptography*, EUROCRYPT 90, Springer Verlag, Berlin 1991, pp. 389-404.