International Journal in IT & Engineering (IJITE)
Volume 11 Issue 04, April 2023 ISSN: 2321-1776 Impact Factor: 7.256
Journal Homepage: http://ijmr.net.in, Email: irjmss@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal

A HYBRID ALGORITHM DESIGN FOR XOR ENCRYPTION TECHNIQUES WITH A COMBINATION OF CBC - ECB 512-BITS BLOCK CIPHER MODES AND RSA ALGORITHMS

ANTONY
North Sumatra Islamic University
Medan, North Sumatra, Indonesia

ABSTRACT

This research was conducted by analyzing the combination of RSA algorithm with ECB and CBC block cipher modes with a 512-bit block length. The aim of this research is to obtain a cryptography algorithm that is fast, resource-efficient, and free from key distribution problems. The data used in the testing process was text data with message lengths ranging from 10 characters to 1 million characters. The results of this research indicate that the combination of RSA algorithm with CBC-ECB 512-bit block cipher mode is able to provide optimal security for the data. The resulting cipher text and cipher key will be very secure from exhaustive search or brute force attacks. It would take up to 1.80 x 10^257 years to decrypt the cipher text, while it would take 3.17 x 10^166 years to break the cipher key using exhaustive search or brute force techniques.

Keywords—RSA, Encryption, Cipher

1. INTRODUCTION.

Cryptography WILL keep information secret by encoding it into a form that the meaning can no longer be understood. Currently, many cryptographic algorithms have emerged which are continuously analyzed, tried, and refined to find algorithms that are considered to meet security standards. Each algorithm has a different level of security, as well as a different level of complexity. An algorithm with a high level of security and low complexity would be very good to implement, this is because the encryption and decryption process is much faster and consumes less resources. However, every cryptographic algorithm that has been found has its own weaknesses.

One of the fastest cryptographic algorithms is the cryptographic algorithm with the XOR technique. Even though it is very simple, the XOR technique is the choice for fast encryption process needs. The XOR technique will be very safe if it is set in the right process scheme, if the process scheme is not right, then the XOR technique will be very unsafe [1].

The XOR technique is included in the symmetric cryptographic algorithm section, where the key for the encryption and decryption processes is only the same key. Because in the process of encryption and decryption the XOR technique performs a one-to-one correspondence process, or one key bit will be XORed against one bit of plain text, the encryption and decryption process in the XOR technique is applied to the block cipher operating mode to increase the security of cipher text produced [2].

In block cipher operation, there are several modes that can be used, some of which are ECB (Electronic Code Book) and CBC (Cipher Block Chaining) operating modes. ECB and CBC are forms of operation of the block cipher mode in the symmetric algorithm. In the block cipher scheme, the plain text will first be divided into blocks of n-bits long and then each block will be operated on. The type of operation that can be used for this mode of operation is the XOR operation technique. The advantages of the XOR technique are its speed and simplicity in the process, but still promises high security for the resulting cipher text

[3].

The strength or security of the block cipher mode of operation lies mainly in the length of the block used, the longer the block used, the more secure the cipher text is. In this study, the proposed block length is 512-bits long. With a length of 512-bits, this algorithm has 2512 possible keys and possible variations of cipher text or around 1.34 x 10154 different key combinations [4].

This research tries to combine the block cipher Electronic Code Book (ECB) and Cipher Block Chaining (CBC) algorithms into one complete mode of operation to produce a super ciphertext that is much safer and stronger than the previous cipher text, supported by two secret keys. which have a key length of 512 bits each. With this combination, it will generate possible key combinations as long as 21024 which will increase the security of the lock without sacrificing the length of the processing block. As with other types of symmetric cryptographic algorithms, the main drawback of the XOR technique is the distribution key problem. where the sender of the message will have great difficulty in safely distributing the key to the recipient of the message so that the message can be decrypted back into the original message (plain text). If the key falls into the hands of eavesdroppers, the encryption process that has been carried out will be in vain [2].

To overcome the key distribution problem that occurs, an asymmetric algorithm is used to encrypt the generated key. The asymmetric algorithm used is the RSA algorithm designed by Ron Rivest, Adi Shamir, and Leonard Adleman, in the 1970s [5].

II. LITERATURE REVIEWS.

2.1. ECB (ELECTRONIC CODE BOOK)

Is one of the processing modes in the block cipher. ECB mode is suitable for encrypting randomly accessed files because each block of plain text is encrypted independently. Even if ECB mode is done with parallel processors, each processor can encrypt or decrypt different plain text blocks [3].

In the Electronic Code Book (ECB) operating mode, encryption is performed by XORing each block of plain text with a key that has the same bit length as each block, as shown in the following figure:

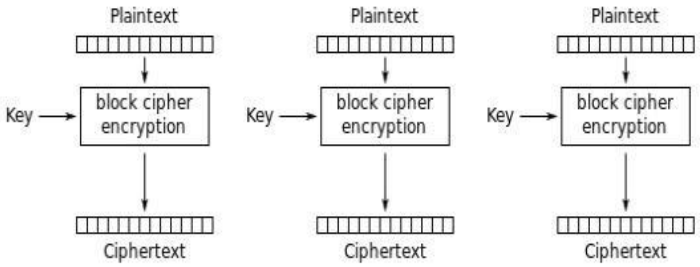

Figure 1.ECB Operation Mode Process Schematic

2.2. CBC (CIPHER BLOCK CHAINING)

The Cipher Block Chaining (CBC) mode implements a feedback mechanism on a block, in which case the results of the previous block encryption are fed back into the encryption block being processed. The trick is that the plain text block that is being processed is XORed first with the cipher text block previously encrypted, then the results of this XOR are entered into the encryption function [3].

In Cipher Block Chaining (CBC), encryption is done by XORing the first block of plain text with an IV (Initialization Vector) which consists of 0 bits along n-bits, then XORing it again with the key to

generate cipher text for the first block. This cipher text is used as the IV (Initialization Vector) for the next block encryption, and so on. As shown in the following figure:

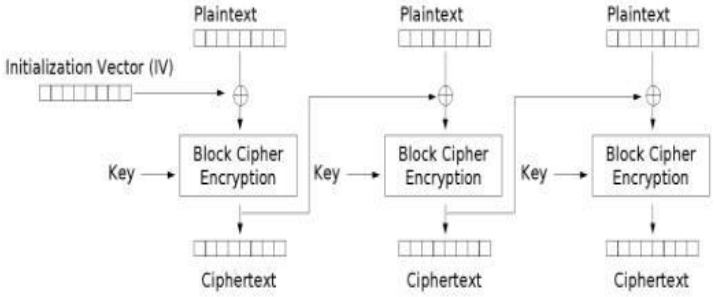

Figure 2.CBC Operation Mode Process Schematic

2.3. RSA

RSA is an asymmetric cryptographic algorithm having two keys, namely a public key and a private key designed by Ron Rivest, Adi Shamir, and Leonard Adleman. The strength of RSA is highly dependent on the complexity of factoring integers into two different prime numbers. RSA only uses exponential operations for encryption and decryption operations. RSA is an asymmetry algorithm [6].

The RSA algorithm consists of 3 processes, namely:[7]

1. Key Generator
   1) Choose two random prime numbers, p and q.
   2) Calculate the system modulus
      $$n = p * q$$
   3) Search Totient Φ(n)
      $$\Phi(n) = (p-1)(q-1)$$
   4) Choose encryption key e randomly
      Where $1 < e < \Phi(n)$, $\gcd(e, \Phi(n)) = 1$
   5) Determine the decryption key d with the following equation:
      $$d \equiv e\text{-}1 \pmod{\Phi(n)}$$
      Where the above equation is equivalent to:

$e * d \equiv 1 \pmod{\Phi(n)}$, where $0 \leq d \leq n$

Key generation result:

a. *Private keys*= (d,n)

is highly confidential, and only the recipient of the message may know it.

b. *Public keys* = (e,n)

It is not secret, and may be distributed freely.

2. Encryption

In general, the encryption process with RSA is done with the following formula:

$$Ci = Piemod\ n$$

3. Description

In general, the decryption process with RSA is carried out using the following formula:

$$Pi = Cid\ mod\ n$$

III. RESEARCH METHODS

The encryption scheme of the proposed hybrid algorithm can be seen in the following figure:

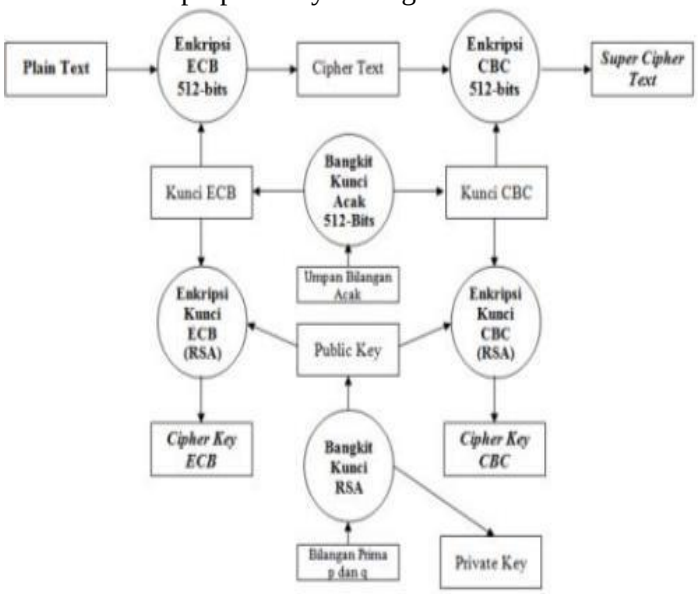

Figure 3. Hybrid Algorithm Encryption Scheme

The decryption process in the proposed Hybrid algorithm can be seen in the following flowchart:

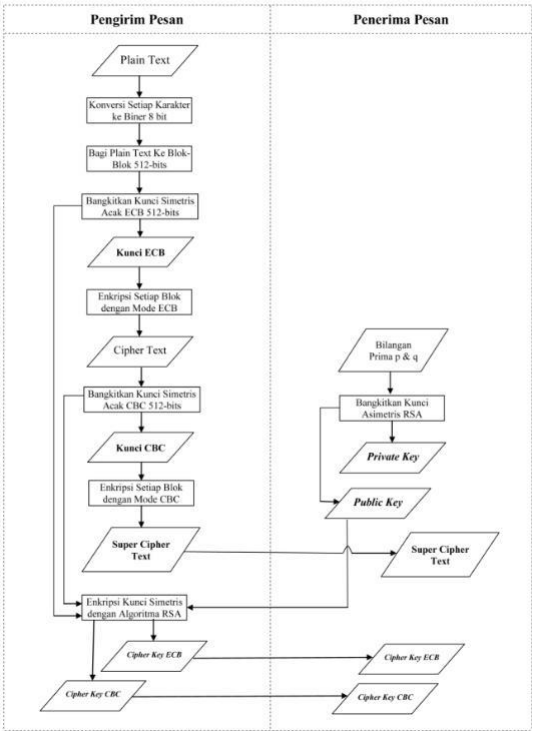

Figure 4. Flowchart of the Encryption Process from the Algorithm Hybrid

The decryption scheme of the proposed hybrid algorithm can be seen in the following figure:

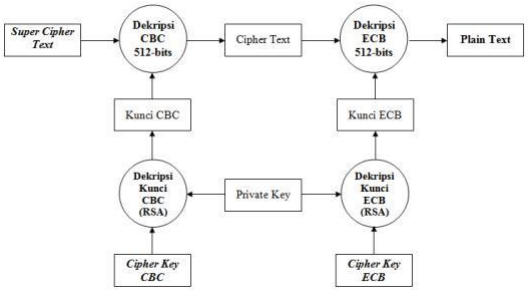

Figure 5. Schematic Decryption of the Hybrid Algorithm

The decryption process in the proposed Hybrid algorithm can be seen in the following flowchart:

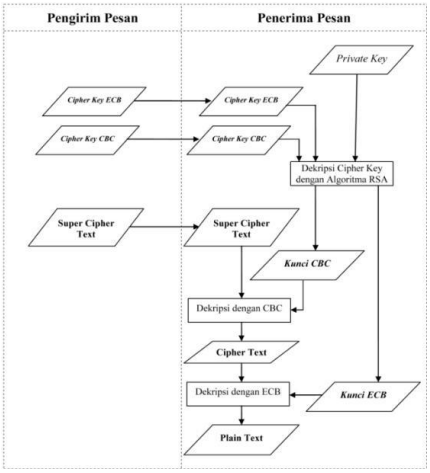

Figure 6. Flowchart of the Decryption Process of the Hybrid Algorithm

The process workflow in the proposed hybrid algorithm scheme follows the following flow:

1. First Phase (RSA Key Generation)
2. Second Stage (Encryption)
   a. ECB Key Generation
   b. CBC Key Generation
   c. Encryption Process with ECB Mode
   d. Encryption Process with CBC Mode
   e. ECB Key Encryption Process with RSA Algorithm
   f. CBC Key Encryption Process with RSA Algorithm
3. Third Stage (Decryption)
   a. CBC Cipher key decryption
   b. ECB Cipher key decryption

c. Super cipher text decryption with CBC Operation Mode

d. Cipher text decryption with ECB Operation Mode

VI. RESULTS AND DISCUSSION.

Testing is done against the schema

proposed hybrid algorithm. In this discussion, there is a plain text that will be tested, while the plaintext is:

Plain text : Medan City, Capital of North Sumatra Province, Medan City is the largest city outside Java in Indonesia.

Generate the RSA key to get a public key (3149, 2159869) and a private key (1010249, 2159869). Then generate ECB and CBC symmetric keys randomly 512-bits long. ECB 512-bits random key used as follows:

000101010110010111000111101110011000

010101101011000100111111101011011010011

011000100010001010100111001100011011001

010110010101100111010000010001111011111

011111110001101111100000101100110001011

101010110001110000100111111000110001110

011111001100101001100011001011011001100

010101111001010010011100100101111110001

111000101010010111100111000000111101000

111111000110011010100101101010110101000

111110110010101101011001111001110100011

011100010000000000010000001010011010001

010111100100000011110001110010110000100

00101101

CBC 512-bits random key used as follows:

001110100001110101000001101110010001

00000100001110001110111101100110110100 0

01110100010110110010000100001011111101 1

10011100111110101011110111100111110000 1

10101110100101010001000110111001110110 1

01101100011111101011001010001001100110 1

11101000100101011110011101100110000001 0

01110011010011101011010100000101010001 0

00101111101100000010000101010000001101 0

11100101000101011000111101011101010111 0

01000011011011010110000111000001101111 0

01111011010111000011111001011101100010 0

01010010110011010100001110001110011001 0

01000101

The length of the plain text to be encrypted is 106 characters or 848 bits. Convert plain text in binary form and then divide it into blocks of 512-bits length. If the last block is less than 512-bits long, then add the padding bit '1' until the last block is 512-bits long. The process of dividing plain text into 512-bit blocks is as follows:

Plain Block 0

01001011011011110111010001100001001000000100 11

01011001010110010001100001011011100010000001 00100

10110001001110101001000000100101101101111011 10100

01100001001000000101000001110010011011110111 01100

11010010110111001110011011010010010000001010 01101

11010101101101011000010111010001100101011100 10011

00001001000000101010101110100011000010111001 00110

00010010110000100000010010110110111101110100 01100

00100100000010011010110010101100100011000010 11011

10001000000100110101100101011100100111010101 11000

00110000101101011011000 01

Plain Block 1

01101110001000000100101101101111011101000110000

01001000000101010001100101011100100110001001100100

10111001101100001011100100010000001100100011010001

00100000010011000111010101100001011100100010000000

10010100110000101110111011000010010000000110010001

10100100100000010010010110111001100100011011110111

01110011001010111001101101001011000010010111011111

11111111111111111111111111111111111111111111111111

11111111111111111111111111111111111111111111111111

11111111111111111111111111111111111111111111111111

11111111111111111111111111

Perform XOR operations with 512-bits ECB mode as shown in the following table:

Table 1.0th Block Plain text Encryption with ECB Key

Plain text Block to – 0	ECB lock	Encryption Results
01001011011011110	00010101011001011	01011110000010101
11101000110000100	10001111011100110	01100111101100010
10000001001101011	00010101101011000	10010100100110011
00101011001000110	10011111110101101	10110100111101011
00010110111000100	10100110110001000	10110000001001100
00001001001011000	10001010100111001	10000011101100001
10011101010010111	10001101100100001	00010000110100101

10000	01011	11011
001001011011 01111	001010110011 10100	000011101000 11011
011101000110 00010	000100011110 11111	011001011000 11101
010000001010 00001	011111110001 10111	001111111011 10110
110010011011 11011	110000010110 01100	000010001101 10111
101100110100 10110	010111010101 10001	111011100001 00111
111001110011 01101	110000100111 11100	001001010100 10001
001001000000 10100	011000111001 11110	010001111001 01010
110111010101 10110	011001010011 00011	101110000110 10101
101100001011 10100	001011011001 10001	100111010010 00101
011001010111 00100	010111100101 00100	001110110010 00000
110000100100 00001	111001001011 11110	001001101111 11111
010101011101 00011	001111000101 01001	011010011000 01010
000010111001 00110	011110011100 00001	011100100101 00111
000100101100 00100	111010001111 11000	111110100011 11100
000010010110 11011	110011010100 10110	110001000010 01101
110111010001 10000	101011010100 01111	011100000101 11111
100100000010 01101	101100101011 01011	001000101001 00110

011001010110 01000	001111001110 10001	010110011000 11001
110000101101 11000	101110001000 00000	011110100101 11000
100000010011 01011	000100000010 10011	100100010001 11000
001010111001 00111	010001010111 10010	011011101110 10101
010101110000 01100	000001111000 11100	010100001000 10000
001011010110 11000	101100001000 01011	100111011110 10011
01	01	00

Table 2.1st Block Plain text Encryption with ECB Key

***plain text*block to – 1**	**ECB lock**	**Encryption Results**
011011110001 000000	000101010111 001011	011110110100 01011
10010110110 111101	10001111011 100110	000110011010 11011
110100011100 001001	000101011101 011000	110001000010 10001
000000101011 000110	100111111110 101101	100111010111 01011
01010111001 001100	10100110110 001000	111100011110 00100
010011100101 011100	100010101001 111001	110001100011 00101
11011000010 111001	10001101100 101011	010101011100 10010
000100000001 100100	001010110011 110100	001110110000 10000
01101001001 000000	00010001111 011111	011110001100 11111
100110001111 010101	011111111000 110111	111001111111 00010
10000101110 010001	11000001011 001100	010001001010 11101

000000010010100110	010111010101110001	01011111000010111
00010111011101100	11000010011111100	11010101000010000
001001000000011001	011000111100111110	01000111100100111
00011010010010000	01100101001100011	01111111011110011
001001001011101110	001011011001110001	00001001001011111
01100100011011110	01011110010100100	00111010001111010
110111001100010101	111001001011111110	00111000011101011
11001101101001011	00111100010101001	11110001111100010
000010010111101111	011110011100000001	01110000101101110
11111111111111111	11101000111111000	00010111000000111
111111111111111111	110011010100010110	00110010101101001
11111111111111111	10101101010001111	01010010101110000
111111111111111111	101100101011101011	01001101010010100
11111111111111111	00111100111010001	11000011000101110
111111111111111111	101110000100000000	01000111011111111
11111111111111111	00010000001010011	11101111110101100
111111111111111111	010001010111110010	10111010100001101
11111111111111111	00000111100011100	11111000011100011
111111111111111111	101100001000001011	01001111011110100
11	01	10

V. CONCLUSION.

By combining block operation mode*cipher*ECB and CBC are 512-bits long with the RSA algorithm, a fast, lightweight and secure cryptographic algorithm will be produced. Where the resulting cipher text and cipher key are very unlikely to be solved by exhaustive search. The cipher text and cipher key generated from the proposed algorithm are very safe from exhaustive search or gross force attacks. Where the time needed to solve the resulting cipher text takes 1.80 x 10^{257}Years, while for the cipher key it takes 3.17 x 10166 years. with the RSA algorithm, a cryptographic algorithm that is fast, lightweight, and secure will be generated. Where the resulting cipher text and

cipher key are very unlikely to be solved by exhaustive search. The cipher text and cipher key generated from the proposed algorithm are very safe from exhaustive search or gross force attacks. Where the time needed to solve the resulting cipher text takes 1.80 x 10257 years, while for the cipher key it takes 3.17 x 10166 years

REFERENCE

[1] Aryza, S., Irwanto, M., Khairunizam, W., Lubis, Z., Putri, M., Ramadhan, A., Hulu, F. N., Wibowo, P., Novalianda, S., & Rahim, R. (2018). An effect sensitivity harmonics of rotor induction motors based on fuzzy logic. *International Journal of Engineering and Technology(UAE)*, *7*(2.13 Special Issue 13), 418–420. https://doi.org/10.14419/ijet.v7i2.13.16936

[2] Wairya, S., Kumar. R., Nagaria., & Tiwari, (2012). Comparative Performance Analysis of XORXNOR Function Based High-Speed CMOS Full Adder Circuits For Low Voltage VLSI Design. International Journal of VLSI design & Communication Systems (VLSICS) Vol.3, No.2, April 2012

[3] Kumar, S., Suneetha, CH, & Chandrasekhar, A. (2011). A Block Cipher Using Rotation and Logical XOR Operations. IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 6, No 1, November 2011 ISSN (Online): 1694-0814

[4] Dashti, A., Kheradmand, HA, & Jazi, M. (2016). Comparison Of Three Modes Of Cryptography Operation For Providing Security and Privacy Based on Important *factors*. International Journal of Information Technology and Electrical Engineering Volume 5, Issue 3 ISSN: - 2306-708 X June 2016

[4] Sridevi. (2014). Construction of Stream Ciphers from Block Ciphers and their Security. IJCSMC, Vol. 3, Issues. 9, September 2014, pg. 703 – 714

[5] Munir, R. (2006). Cryptography. Informatics: Bandung.

[6] Kallam, RB (2011. An Enhanced RSA Public key Cryptographic Algorithm. International Journal of Advanced Research in Computer Science (IJARCS)

[7] Singh, S. (2013). A Performance Analysis of DES and RSA Cryptography. International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)