Enhancement IDEA Algorithm with Digital Image as Key Encryption and Decryption

Robbi Rahim\(^1\)*, Helmy Fauzi Siregar\(^2\), and Delima Sitanggang\(^3\)
\(^1\)Sekolah Tinggi Ilmu Manajemen Sukma, Medan, Indonesia
\(^2\)Department of Informatics, Universitas Asahan, Kisaran, Indonesia
\(^3\)Department of Informatics, Universitas Prima Indonesia, Medan, Indonesia

*usurobbi85@zoho.com

Abstract. Cryptography is one type of protection that is widely used to secure information, encryption, and decryption and the use of keys is a common process used to secure information; and also the cryptographic processes have a classic problem that is the distribution of keys that are vulnerable to interception when the sender sends the key to the receiver, the cryptographic protocol (Authentication) could be used to minimize key distribution problem because there is no need for key exchanges. Shamir's Three-Pass Protocol, Secret Splitting, Bit-Commitment Protocol and Blind Signature are few cryptographic protocols that can be used to help with key distribution issues, this article performs systematic approach to protocol cryptography for security level and the using other algorithms to combine with protocol.

1. Introduction

Data security [1]–[5] is still a very serious concern until today, with many technological developments and algorithms being undertaken to secure data from irresponsible parties [6]–[9]. Security can be done in various ways such as cryptography and steganography [10], [11], e.g. cryptography using AES, RSA, MMB, GOST[12] and IDEA algorithms [13], while LSB [14], EOF, Pixel Value Differencing for steganography [11]. Utilization of the algorithm can be done independently or can be combined, or also could be applied to certain processes such as on network communications.

International Data Encryption Algorithm or IDEA is a block cipher algorithm with 128 bit key length and 64 bit encryption block [13], [15], [16]. IDEA was originally a powerful and reliable algorithm for securing data, and like an algorithm if it has advantages it also has weaknesses and on IDEA algorithm the weakness lies in the key rotation especially linear factor [15], [17], this weakness can be overcome by modifying the key scheduling from IDEA algorithm [13], [17]. Another technique that can be used is a key that is used instead of a common text but an image that has the composition of pixel values [8], [18], [19], hashing values and also byte code so that have different variables to be key.

Digital image used as a key in the process of encryption and decryption is done by reading the value of hashing as the key [18], [20], this process would make it difficult for cryptanalyst to know the keys used. One of the reason because the keys that do not have a pattern, in addition to perform the decryption process also requires a digital image used. Digital image as a key is expected to be one of the solutions to improve the security of the IDEA algorithm.
2. Methodology

2.1 Cryptography

Cryptography can be defined as a science that studies mathematical techniques related to aspects of information security such as confidentiality, data integrity, sender/data receiver authentication, and data authentication. Cryptanalysis is the study of how to solve cryptographic mechanisms[21]–[24].

The IDEA algorithm used in this study has several cryptographic mathematical foundations used for the process of encryption and decryption[15], [16], [25], [26]:

a. Modular Arithmetic

Modular arithmetic is a mathematical operation that is widely implemented in cryptographic methods. In the IDEA method, the modular arithmetic operation used is the modulo $2^{16}$ addition operation and the modulo $2^{16} + 1$ multiplication operation. Modulo operation involves numbers 0 and 1 only so that it is identical to the bits on the computer. Example,

- $(65530 + 10) \mod 2^{16} = 65540 \mod 65536 = 4$
- $(32675 \times 4) \mod (2^{16} + 1) = 131060 \mod 65537 = 65523$

b. Invers Multiplication

Inverse multiplication used on the IDEA method is not like the inverse of the multiplication operation in mathematics. This multiplication cannot be explained mathematically, but by using the following algorithm:

```vbnet
Function InverseA As Double) As Double
n = 65537
G0 = n
G1 = A
V0 = 0
V1 = 1
While (G1 <> 0)
    Y = Int(G0 / G1)
    G2 = G0 - Y * G1
    G0 = G1
    G1 = G2
    V2 = V0 - Y * V1
    V0 = V1
    V1 = V2
Wend
If (V0 >= 0) Then
    Inverse = V0
Else
    Inverse = V0 + n
End If
End Function
```

c. Inverse Addition

Inverse addition in the IDEA method using the following algorithm:

- Inverse sum = $65536 - \text{pnBil}$
- Example: Inverse sum of 32654 is $65536 - 32654 = 32882$

d. XOR Operation

XOR is an Exclusive-OR operation denoted by a "$\oplus$" sign. The XOR operation will return a bit value of "0" (zero) if XOR two bits of the same value and will result in a "1" bit value if XOR two bits of a different bit value.

e. Permutation

Permutations in cryptography are often used to move the position of a number of bits to a predetermined position in the permutation table. Permutations in the IDEA algorithm are used as the process of establishing sub keys.
f. Bit Shift
   The bit shift is a shift operation against a sequence of bits as much as desired.

g. Bit Rotation
   Rotation of bits is the operation of rotation to a sequence of bits as much as desired, there are 2 operations
   that can be used left rotation and right rotation.

h. Modulo Multiplication
   Multiplication with zero always produces zero and has no inverse. Multiplication modulo \( n \) also has no
   inverse if the number multiplied is not relatively prime to \( n \). In IDEA, for multiplication operations, a 16 bit
   number consisting of zeros is all considered a number 65536, while other numbers remain in accordance
   with the unidentified numbers it represents, example:
   \[
   (32542 \times 10) \mod 65537 = 325420 \mod 65537 = 63272
   \]
   \[
   (3154 \times 25) \mod 65537 = 78850 \mod 65537 = 13313
   \]

2.2 Digital Image

Digital image is a two-dimensional image that is displayed on certain media as a set of digital values called pixels
[18], [27]-[29]. In a mathematical overview, the image is a continuous function of the intensity light in a two-di

Digital image is the image \( f(x, y) \) where discrete spatial coordinates are coordinated and quantization level
discretization. Digital image is a function of light intensity \( f(x, y) \), where \( x \) and \( y \) are spatial coordinates. The
function at each point \( (x, y) \) represents the brightness level of the image at that point. The digital image is a matrix
in which the row and column indices represent a point on the image and the matrix element (pixel element)
represents gray level at that point [28], [30], [31].

The pixel value form digital image used as key for encryption and decryption process has a 3x3 matrix size, see example of the following RGB image pixel.

![Figure 1. Pixel Image Sample](image)

The pixel value in Figure 1 will be the key to the encryption and decryption process performed on the IDEA
algorithm, the pixel value will be taken randomly for the encryption process and when the decryption process
also requires the same image.

3. Results and Discussion

Security analysis on IDEA algorithms using images as key requires several sets of rules that must be met as follows:

a. Maximized digital image size of 200x200
b. The key for encryption and decryption is obtained from randomly generated RGB image pixels by 3x3 pixel
   reading from left to right horizontally.
c. Key authentication is derived from digital images or not before encryption and decryption processes.
d. Encryption and decryption process is completed by using IDEA algorithm.

The key used from digital image with 3x3 matrix as follows:

**Table 1. 3x3 Matrix Key**

|   |   |   |
|---|---|---|
| 78 | 77 | 89 |
| 25 | 55 | 56 |
| 56 | 55 | 00 |

RGB Pixel value is then read from the left (78) to the right (00) horizontally and obtained the following result: 787782555656500. THOMPSON is a message to be secured using IDEA algorithm, the process as below:

**ROUND 1**

01) \( L#1 = (X1 \times K1) \mod (2^{16} + 1) = 0101010001001000 \times 0011011100111000 \mod (2^{16} + 1) = 1101010110010011 \)

02) \( L#2 = (X2 + K2) \mod 2^{16} = 0100111110100110 + 0011011110011011 \mod 2^{16} = 1000011010000100 \)

03) \( L#3 = (X3 + K3) \mod 2^{16} = 0101000001010011 + 0011100000111001 \mod 2^{16} = 1000100010001100 \)

04) \( L#4 = (X4 \times K4) \mod (2^{16} + 1) = 0100111110100110 \times 0011001000110101 \mod (2^{16} + 1) = 1001011110011001 \)

05) \( L#5 = L#1 \ XOR \ L#3 = 1101010110010011 \ XOR \ 1000100010011100 \ XOR \ 1000101100011111 \)

06) \( L#6 = L#2 \ XOR \ L#4 = 1001011110011001 \ XOR \ 1000100010001100 \ XOR \ 1000101100011101 \)

07) \( L#7 = (L#5 \times K5) \mod (2^{16} + 1) = 0101110100001111 \times 0011010100110101 \ mod (2^{16} + 1) = 1001111100010001 \)

08) \( L#8 = (L#6 + L#7) \mod 2^{16} = 0001000100011110 + 1001111100011001 \mod 2^{16} = 1011000000101110 \)

09) \( L#9 = (L#8 \times K6) \mod (2^{16} + 1) = 1011000000101110 \times 0011010100110110 \mod (2^{16} + 1) = 1000101100011010 \)

10) \( L#10 = (L#7 + L#9) \mod 2^{16} = 1001111110001001 \times 1000101100011010 \mod 2^{16} = 0010101000111111 \)

11) \( L#11 = L#1 \ XOR \ L#9 = 1101010110010011 \ XOR \ 1000101100011010 \ XOR \ 1001111100011111 \)

12) \( L#12 = L#2 \ XOR \ L#9 = 1000100010001100 \ XOR \ 1000101100011010 \ XOR \ 0001001111101101 \)

13) \( L#13 = L#2 \ XOR \ L#10 = 1000011010000100 \ XOR \ 0010101000110110 \ XOR \ 1010110010100011 \)

14) \( L#14 = L#4 \ XOR \ L#10 = 10010111110011001 \ XOR \ 0010101000110110 \ XOR \ 1011111111011111 \)

Next encryption rotation:

\( X1 = L#11 = 01011111010000101 \)
\( X2 = L#12 = 0000001111011010 \)
\( X3 = L#13 = 1010110010100011 \)
\( X4 = L#14 = 1011111101111110 \)
ROUND 2

01) L#1 = (X1 * K1) mod (2^16 + 1) = 0101010001001000 * 0011011100111000 mod (2^16 + 1) = 1101010110010011

02) L#2 = (X2 + K2) mod 2^16 = 10010111110011001 + 001101111011010011 mod 2^16 = 100001110100010010

03) L#3 = (X3 + K3) mod 2^16 = 010100000011011001 + 0011000000111001 mod 2^16 = 1001010001101011

04) L#4 = (X4 * K4) mod (2^16 + 1) = 10011011001110110 * 0011010000110101 mod (2^16 + 1) = 1001010011111001

05) L#5 = L#1 XOR L#3 = 1101010110010011 XOR 1001010011111001 = 0101110100011111

06) L#6 = L#2 XOR L#4 = 10010101001000000100 XOR 100010111100110010 = 0001000100011011

07) L#7 = (L#5 * K5) mod (2^16 + 1) = 01011110110011110 * 00110101011110101 mod (2^16 + 1) = 1001111110001001001

08) L#8 = (L#6 + L#7) mod 2^16 = 00010011010010110 + 100111111000101101 mod 2^16 = 1011000001011110

09) L#9 = (L#8 * K6) mod (2^16 + 1) = 101100001010110110 * 0011010011011100 mod (2^16 + 1) = 1000011011010110

10) L#10 = (L#7 + L#9) mod 2^16 = 1001111100010001001 + 1000011010001110 mod 2^16 = 000100111000100111

11) L#11 = L#1 XOR L#9 = 1101010110010011 XOR 10001011000011100 = 0101111010001101

12) L#12 = L#3 XOR L#9 = 10000100010001100 XOR 1001010011010110 = 0000000111001101

13) L#13 = L#2 XOR L#10 = 10000011010000100 XOR 0010101000111011 = 1010110101100011

14) L#14 = L#4 XOR L#10 = 1001011111010101001 XOR 0010101000111011 = 1011110110111110

Next encryption rotation:
X1 = L#11 = 0101111101000101
X2 = L#12 = 0000001111001101
X3 = L#13 = 1010110101000011
X4 = L#14 = 1011110110111110

ROUND 3

01) L#1 = (X1 * K1) mod (2^16 + 1) = 0101010001001000 * 0011011100111000 mod (2^16 + 1) = 1101010110010011

02) L#2 = (X2 + K2) mod 2^16 = 10010111110011001 + 001101111011010011 mod 2^16 = 100001110100010010

03) L#3 = (X3 + K3) mod 2^16 = 010100000011011001 + 0011000000111001 mod 2^16 = 1000100001100011

04) L#4 = (X4 * K4) mod (2^16 + 1) = 10011011001110110 * 00110010001110101 mod (2^16 + 1) = 1001010011111001

05) L#5 = L#1 XOR L#3 = 1101010110010011 XOR 1001010011111001 = 0101111010001101

06) L#6 = L#2 XOR L#4 = 10010101001000000100 XOR 100010111100110010 = 0001000100011011
07) \( L^#7 = (L^#5 \times K_5) \mod (2^{16} + 1) = 0101111010001111 \times 00110110110001 \mod (2^{16} + 1) = 1001111100010001 \)

08) \( L^#8 = (L^#6 + L^#7) \mod 2^{16} = 0001000100011101 + 10110110000100 \mod 2^{16} = 1011000000101110 \)

09) \( L^#9 = (L^#8 \times K_6) \mod (2^{16} + 1) = 1011000000101110 \times 00101010110101 \mod (2^{16} + 1) = 1000101100010110 \)

10) \( L^#10 = (L^#7 + L^#9) \mod 2^{16} = 0101111000100100 + 1000101100010110 \mod 2^{16} = 0010101001000111 \)

11) \( L^#11 = L^#1 \oplus L^#9 = 1000100010001100 \oplus 1000101100010110 = 0000001110011010 \)

12) \( L^#12 = L^#3 \oplus L^#9 = 1010110010100011 \oplus 1000101100010110 = 1010110010100011 \)

13) \( L^#13 = L^#2 \oplus L^#10 = 1000011010000100 \oplus 0010100100011111 = 1010110010100011 \)

14) \( L^#14 = L^#4 \oplus L^#10 = 1000101110001101 \oplus 0010101000011111 = 1010110110111110 \)

Next encryption rotation:
\( X_1 = L^#11 = 0101111010000101 \)
\( X_2 = L^#12 = 0000001110011010 \)
\( X_3 = L^#13 = 1010110010100011 \)
\( X_4 = L^#14 = 1011110101111110 \)

Encryption rotation will loop until eight rotation and the last result as below:

01) \( Y_1 = (X_1\times K_1) \mod (2^{16} + 1) = 0111011100010010 \times 11001100011000 \mod (2^{16} + 1) = 110111010111100 \)

02) \( Y_2 = (X_2 + K_2) \mod 2^{16} = 11010001000000100 + 00000011100001101 \mod 2^{16} = 1000111101000001 \)

03) \( Y_3 = (X_3 + K_3) \mod 2^{16} = 11001001001000100 + 1000110100011101 \mod 2^{16} = 0111110101011101 \)

04) \( Y_4 = (X_4 \times K_4) \mod (2^{16} + 1) = 0110101110011110 + 1000110101001101 \mod (2^{16} + 1) = 1001110000100000 \)

Encryption Result:
\( Y_1 = 11011101110000100 = ÚÅ \)
\( Y_2 = 1001110110011011 = Ÿ‰ \)
\( Y_3 = 0110101010100111 = e\)  
\( Y_4 = 1001110000010000 = œ \)

Cipher text = ÚÅŸ‰e œ

The chipertext result is then decrypted by following predefined rules, here is the decryption process for round 1:

01) \( L^#1 = (X_1 \times K_1) \mod (2^{16} + 1) = 1101101011000100 \times 01110110110001 \mod (2^{16} + 1) = 0111011011000100 \)

02) \( L^#2 = (X_2 + K_2) \mod 2^{16} = 1101101100011101 + 01110110110001 \mod 2^{16} = 110100100111101 \)

03) \( L^#3 = (X_3 + K_3) \mod 2^{16} = 0110101110011110 + 01110110110001 \mod 2^{16} = 110110110010000 \)
04) \( L#4 = (X4 \times K4) \mod (2^16 + 1) = 1001110000010000 \times 111010100110011 \mod (2^16 + 1) = 0110101101001110 \)

05) \( L#5 = L#1 \text{ XOR } L#3 = 011101110010010 \text{ XOR } 1101110001010000 = 101011111000010 \)

06) \( L#6 = L#2 \text{ XOR } L#4 = 1001000100111101 \text{ XOR } 0110101011001110 = 1111101001110011 \)

07) \( L#7 = (L#5 \times K5) \mod (2^16 + 1) = 1010111110100010 \times 110001101001110 \mod (2^16 + 1) = 0010100001101110 \)

08) \( L#8 = (L#6 + L#7) \mod 2^16 = 1111101001110011 + 0010100011000110 \mod 2^16 = 0010001110100001 \)

09) \( L#9 = (L#8 \times K6) \mod (2^16 + 1) = 001000011000001 \times 110001101001110 \mod (2^16 + 1) = 0000011011001011 \)

10) \( L#10 = (L#7 + L#9) \mod 2^16 = 0010100001100110 + 0000011011001011 \mod 2^16 = 0010111110011001 \)

11) \( L#11 = L#1 \text{ XOR } L#9 = 0111011100010010 \text{ XOR } 0000011011001011 = 0111000111101101 \)

12) \( L#12 = L#3 \text{ XOR } L#9 = 1101100001100001 \text{ XOR } 0000011011001011 = 1101111110101011 \)

13) \( L#13 = L#2 \text{ XOR } L#10 = 1001000100111011 \text{ XOR } 0010111110011001 = 1011111010100010 \)

14) \( L#14 = L#4 \text{ XOR } L#10 = 0110101100111010 \text{ XOR } 0010111110011001 = 010001001101111 \)

Next decryption rotation
\( X1 = L#11 = 01110001111101101 \)
\( X2 = L#12 = 1101111010011011 \)
\( X3 = L#13 = 1011111010100010 \)
\( X4 = L#14 = 0100010011010111 \)

Rotation is done until round 8 and produces the following output transformation:

01) \( Y1 = (X1 \times K1) \mod (2^16 + 1) = 110101010110010011 \times 1001100011010100 \mod (2^16 + 1) = 01010100011000100 \)

02) \( Y2 = (X2 + K2) \mod 2^16 = 10000110100000100 + 1100100011001001 \mod 2^16 = 01001111101001101 \)

03) \( Y3 = (X3 + K3) \mod 2^16 = 1000100010001100 + 1100011111000111 \mod 2^16 = 01010000010010111 \)

04) \( Y4 = (X4 \times K4) \mod (2^16 + 1) = 1001011110011001 \times 010010101100100010 \mod (2^16 + 1) = 01001111101000110 \)

Decryption result
\( Y1 = 010101000010000 = TH \)
\( Y2 = 0001001111011001 = OM \)
\( Y3 = 0101000001100110 = PS \)
\( Y4 = 01001111101001110 = ON \)

Plain text = THOMPSON
4. Conclusion

The experiment performed to improve the encryption and decryption of the IDEA algorithm use the random Pixel value of the image, and from the examination it is found that the process of decryption with different images cannot be done due to the different pixel images, in the experiment at key generator it requires a lot of time to read pixel images randomly, for future development can be done optimization by using a special algorithm image processing so that the combination of algorithms make the key generation much faster with maximum results.

REFERENCES

[1] A. Putera, U. Siahaan, and R. Rahim, “Dynamic Key Matrix of Hill Cipher Using Genetic Algorithm,” Int. J. Secur. Its Appl., vol. 10, no. 8, pp. 173–180, Aug. 2016.
[2] R. Rahim, “Man-in-the-middle-attack prevention using interlock protocol method,” ARPN J. Eng. Appl. Sci., vol. 12, no. 22, pp. 6483–6487, 2017.
[3] R. Rahim, M. Dahri, M. Syahlil, and B. Anwar, “Combination of the Blowfish and Lempel-Ziv-Welch algorithms for text compression,” World Trans. Eng. Technol. Educ., vol. 15, no. 3, pp. 292–297, 2017.
[4] H. Nurdiyanto, R. Rahim, and N. Wulan, “Symmetric Stream Cipher using Triple Transposition Key Method and Base64 Algorithm for Security Improvement,” J. Phys. Conf. Ser., vol. 930, no. 1, p. 012005, Dec. 2017.
[5] R. F. Marta, G. S. Achmad Daengs, A. Daniar, W. O. Seprina, and A. P. Menayang, “Information security risk and management in organizational network,” Int. J. Eng. Adv. Technol., vol. 8, no. 6 Special Issue 2, pp. 1152–1156, Aug. 2019.
[6] K. J. Fitzgerald, “Security and data integrity for LANs and WANs,” Int. J. Eng. Adv. Comput. Secur., vol. 3, no. 4, pp. 27–33, 1995.
[7] R. J. Boyle, C. D. Challa, and J. A. Clements, “Valuing Information Security: A Look at the Influence of User Engagement on Information Security Strength,” J. Inf. Priv. Secur., vol. 13, no. 3, pp. 137–156, Jul. 2017.
[8] Q. Kester, “A Visual Cryptographic Encryption Technique for Securing Medical Images,” arXiv Prepr. arXiv1307.7791, vol. 3, no. 6, pp. 3–7, 2013.
[9] R. Rahim, A. Indahingwati, I. K. Sudarsana, G. S. Achmad Daengs, and Y. Yuniningsih, “Caesar cipher and end of file combination algorithm for steganography,” J. Adv. Res. Dyn. Control Syst., vol. 10, no. 10 Special Issue, pp. 947–951, 2018.
[10] R. Rahim and A. Ikhwan, “Cryptography Technique with Modular Multiplication Block Cipher and Playfair Cipher,” Int. J. Sci. Res. Sci. Technol., vol. 2, no. 6, pp. 71–78, 2016.
[11] D. Nofriansyah and R. Rahim, “Combination of Pixel Value Differencing Algorithm with Caesar Cipher for Steganography,” Int. J. Res. Sci. Eng., vol. 2, no. 6, pp. 153–159, 2016.
[12] H. Nurdiyanto and R. Rahim, “Enhanced pixel value differencing steganography with government standard algorithm,” in 2017 3rd International Conference on Science in Information Technology (ICSITech), 2017, pp. 366–371.
[13] O. Almasri and H. M. Jani, “Introducing an Encryption Algorithm based on IDEA,” vol. 2, no. 9, pp. 334–339, 2013.
[14] B. S. Champakamala, K. Padmini, R. D. K. A. Professors, and D. Bosco, “Least Significant Bit algorithm for image steganography Overview of Steganography,” Int. J. Adv. Comput. Technol., vol. 3, no. 4, p. 5, 2014.
[15] J. Chen, D. Xue, and X. Lai, “An analysis of international data encryption algorithm(IDEA) security against differential cryptanalysis,” Wuhan Univ. J. Nat. Sci., vol. 13, no. 6, pp. 697–701, 2008.
[16] G. A. Ludwig and P. Alegre, “International Data Encryption Algorithm (IDEA),” vol. 8, no. 1, pp. 6–11, 2007.
[17] J. Daemen, R. Govaerts, and J. Vandewalle, “Weak Keys for IDEA,” in Advances in Cryptology
— CRYPTO ‘93, Berlin, Heidelberg: Springer Berlin Heidelberg, 1993, pp. 224–231.

[18] R. I. Al-Khalid, R. A. Al-Dallah, A. M. Al-Anani, R. M. Barham, and S. I. Hajir, “A Secure Visual Cryptography Scheme Using Private Key with Invariant Share Sizes,” J. Softw. Eng. Appl., vol. 10, no. 01, pp. 1–10, Jan. 2017.

[19] L. Sui, K. Duan, and J. Liang, “A secure double-image sharing scheme based on Shamir’s three-pass protocol and 2D Sine Logistic modulation map in discrete multiple-parameter fractional angular transform domain,” Opt. Lasers Eng., vol. 80, pp. 52–62, May 2016.

[20] Iswanto, O. Wahyunggoro, and A. I. Cahyadi, “3D object modeling using data fusion from laser sensor on quadrotor,” in AIP Conference Proceedings, 2016, vol. 1755.

[21] M. B. Pramanik, “Implementation of Cryptography Technique using Columnar Transposition,” Int. J. Comput. Appl., pp. 19–23, 2014.

[22] Q.-A. Kester, “A HYBRID CRYPTOSYSTEM BASED ON VIGENÈRE CIPHER AND COLUMNAR TRANSPOSITION CIPHER,” Int. J. Adv. Technol. Eng. Res., vol. 3, no. 1, pp. 141–147, 2013.

[23] S. Bruce, Applied cryptography. 1996.

[24] S. Manna and S. Dutta, “A Stream Cipher based Bit-Level Symmetric Key Cryptographic Technique using Chen Prime Number,” Int. J. Comput. Appl., vol. 107, no. 12, pp. 975–8887, 2014.

[25] H. P. Singh, S. Verma, and S. Mishra, “Secure-International Data Encryption Algorithm,” Int. J. Adv. Res. Electr. Electron. Instrum. Eng., vol. 2, no. 2, pp. 780–792, 2013.

[26] H. K. Sahu, V. Jadhav, S. Sonavane, and R. K. Sharma, “Cryptanalytic attacks on international data encryption algorithm block cipher,” Def. Sci. J., vol. 66, no. 6, pp. 582–589, 2016.

[27] E. Hariyanto and R. Rahim, “Arnold’s Cat Map Algorithm in Digital Image Encryption,” Int. J. Sci. Res., vol. 5, no. 10, pp. 1363–1365, Oct. 2016.

[28] X. Li, “Watermarking in secure image retrieval,” Pattern Recognit. Lett., vol. 24, no. 14, pp. 2431–2434, 2003.

[29] A. M. Bagade and S. N. Talbar, “Image morphing concept for secure transmission of image data contents over internet,” J. Comput. Sci., vol. 6, no. 9, pp. 987–992, 2010.

[30] R. Gonzalez and R. Woods, Digital image processing. 2002.

[31] A. Swaminathan, Y. Mao, and M. Wu, “Robust and secure image hashing,” IEEE Trans. Inf. Forensics Secur., vol. 1, no. 2, pp. 215–230, 2006.