Probability Mapping of Encryption Algorithm

Issam Trrad

Faculty of Engineering, Department of Communication and Computer Engineering, Jadara University, Irbid, Jordan

Summary
Resource sharing on data communication network raises the importance of data security algorithms. Plenty of encryption algorithms are developed to ensure confidentiality of user data. This paper investigates the effect of statistical distribution of the keys and plain text on the security of the ciphered output. The developed algorithm utilizes diffusive encryption key blocks in hierarchical arrangement. This arrangement provides a variable encryption key statistical structure and confusion levels. The encryption key length and key generation functions provide different security levels whereas the hierarchical arrangement provides levels of confusion. However, the proposed algorithm maintains the parallelism of the encryption and decryption process. The effect of the encryption keys statistical distribution in consequence to the substitution box function on the breakability of the algorithm is experimentally verified. The results showed that the statistical distribution of the encryption keys could be selected according to the substitution box function to increase the robustness of the encryption against attack.

Key words:
DES, x-OR probability, binomial probability, encryption, decryption, keys.

1. Introduction

Data Encryption Standard (DES) was the main encryption standard in USA up until the introduction of the Advanced Encryption Standard (AES). DES is a symmetric 64 bit block cipher algorithm in which data blocks of 64 bits are ciphered at a time in contrast to stream cipher algorithms that is based on bit or byte ciphering process. The parallelism of the DES and the diffusion level provided through the permutation and substitution boxes are the advantages of the DES algorithm. However, the encryption key length of 56 bits is the main limiting factor for this standard, particularly, after the advent of high computational capability machines [1, 2-8].

DES was the result of a research project set up by International Business Machines (IBM) Corporation in the late 1960's which resulted in a cipher known as LUCIFER. In the early 1970's it was decided to commercialize LUCIFER and a number of significant changes were introduced. IBM was not the only one involved in these changes as they sought technical advice from the National Security Agency (NSA) with minor contribution of outside consultants. The altered version of LUCIFER was put forward as a proposal for the new national encryption standard requested by the National Bureau of Standards (NBS). It was finally adopted in 1977 as the Data Encryption Standard - DES [2, 4,10-18].

Some of the changes made to LUCIFER have been the subject of much controversy even to the present day. The most notable of these was the key size. LUCIFER used a key size of 128 bits however this was reduced to 56 bits for DES. Even though DES actually accepts a 64 bit key as input, the remaining eight bits are used for parity checking and have no effect on DES's security. Outsiders were convinced that the 56 bit key was an easy target for a brute force attack due to its extremely small size. The need for the parity checking scheme was also questioned without satisfying answers [16, 19-23].

Another controversial issue was that the S-boxes used were designed under classified conditions and no reasons for their particular design were ever given. This led people to assume that the NSA had introduced a "trapdoor" through which they could decrypt any data encrypted by DES even without knowledge of the key. One startling discovery was that the S-boxes appeared to be secure against an attack known as Differential Cryptanalysis which was only publicly discovered by Biham and Shamir in 1990's [23, 24].

This suggests that the NSA were aware of this attack in 1977; 13 years earlier. In fact the DES designers claimed that the reason they never made the design specifications for the S-boxes available was that they knew about a number of attacks that were not public. In 1994 NIST reaffirmed DES for government use for a further five years for use in areas other than "classified". DES of course is not the only symmetric cipher. There are many others, each with varying levels of complexity. Such ciphers include: IDEA, RC4, RC5, RC6 and the new Advanced Encryption Standard (AES). AES is an important algorithm and was originally meant to replace DES (and its more secure variant triple DES) as the standard algorithm for non-classified material. However as of 2003, AES with key sizes of 192 and 256 bits has been found to be secure enough to protect information up to top secret [24-30].

Since its creation, AES had underdone intense scrutiny as one would expect for an algorithm that is to be used as the standard. To date it has withstood all attacks but the search is still on and it remains to be seen whether or not this will last [18, 22, 30-36].

This work is directed to the analysis of the effect of the statistical distribution of the encryption keys on the

Manuscript received June 5, 2019
Manuscript revised June 20, 2019

encryption algorithm complexity. To perform this study, the substitution boxes (S-box) are modeled as a mapping function. The input to this mapping function is the result of XORing the text and the encryption keys. Encryption keys, in this algorithm, are selected from different sets of keys such that the distribution of the selected keys are uniform. This step in most of the block ciphering algorithms is considered the main building block of the algorithm and repeated in multiple steps to increase the robustness of the algorithm. A statistical modeling of the encryption keys and the substation boxes would help selecting the encryption keys distribution that would effectively increase the complexity of the encryption algorithm. This work considers single stage encryption, though, the results can be confidently expanded for multiple stages algorithms such as DES and its variants.

2. Mathematical Model

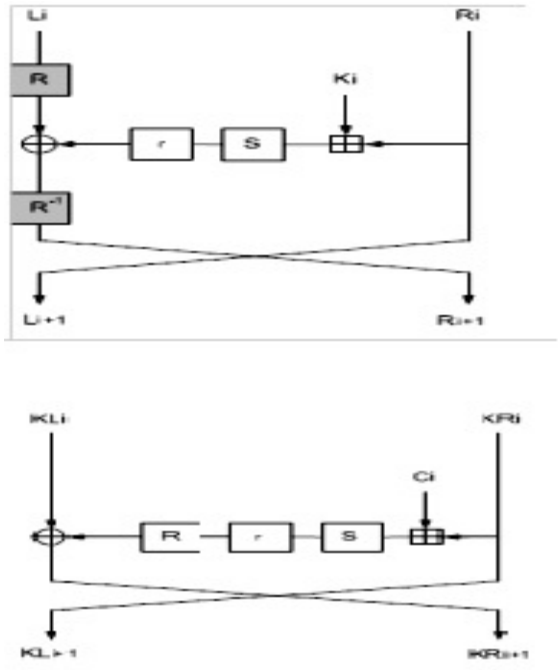

Fig. 1 Encryption Algorithm main round for encryption/ Decryption

The main block of the DES encryption algorithm as shown in Fig. 1 consists of rotation, XORing and mapping through the substitution box. This round is repeated N times in the encryption algorithm. In DES algorithm, this round is repeated for 16 times to ensure shuffling of the text and robust encryption. According to the number of rounds in encryption stage, the decryption follows the complement steps to decipher the encrypted text into the original text. As the S-Box and X-OR operations form the main blocks in DES like encryption algorithm, mainly in most of encryption algorithms, understanding the probability density of the output of every encryption stage requires modeling the probability mapping of the S-Box and X-OR operations. This modeling would help relating the distribution of the output of every stage to the distribution of its input.

2.1 S-Box Probability Mapping

Substitution boxes are considered as the main key of the robustness of encryption algorithms since they map the text bytes (bits) to another function domain. This mapping is random; therefore, it affects the randomness of the input text. In order to be able to decipher the encrypted text, the substitution boxes mapping function should be selected such that its inverse exists.

As the inputs to the substitution boxes are the result of XORing the input text with the randomly distributed encryption keys, the substitution boxes can be viewed, in probability perspective, as a probability transformation function. Thus, assuming the input to the substitution box as a continuous random variable, x, defined on the entire line R and the mapping function of the substitution box as a real valued strictly increasing function, $y = h(x)$, thus, $x = g(y)$, then the following relation can be formulated.

$$P(c \le Y \le d) = P(a \le X \le b) \quad (1)$$

Where

$b > a$ and $d > c$

This relation can be achieved by noting that

$$P(c \le Y \le d) = P(c \le h(X) \le d) \quad (2)$$

$$= P\left(g(c) \le g\big(h(X)\big) \le g(d)\right) \quad (3)$$

$$= P(a \le X \le b) \quad (4)$$

This Result shows the relation between the substitution box (S-Box) mapping function and the probability transformation applied on the random input x. Eq. (1) shows the relation between the substitution box input and output random variables probabilities, however, it does not provide a method of computing the probability density function of the output knowing the probability density function of the input and the substitution box mapping function. The following theorem shows that the output distribution function can be computed as a result of the input distribution function and the substitution mapping function.

Theorem 1: Suppose x is continuous with probability density function $f_X(x)$. Let $y = h(x)$ with h is a differentiable continuous strictly increasing function with inverse $y = h(x) \leftrightarrow x = g(y)$, then $Y = h(X)$ is continuous with probability density function $f_Y(y)$ computed as:

$$f_Y(y) = f_x\big(g(y)\big)g'(y) \quad (5)$$

This theorem shows that the output probability function is a direct result of multiplying the probability density function of the input by the derivative of the inverse mapping function. Noting that there is no specific known rule for selecting the S-Box mapping except that it is one to one mapping, that is, it maps each X-OR output to a unique image. Therefore, confidently it can be assumed that the S-Box obeys uniform distribution function in its entries. This uniform distribution relation, when imposed on a random distributed variable, it will only scale the random distribution function. Thus, to have full understanding of the probability distribution of the S-Box, investigating the X-OR probability mapping characteristics is eminent.

2.2 X-OR Probability Mapping

Similarly, the XORing of the encryption keys can be viewed as probability transformation function by noting that the decimal interpretation of the XORing function can be decimally represented in the following truth table considering the probability of 0 is px and for 1 is (1-px), and the same probabilities hold for the plain text bits, y:

Table 1: XOR truth table with probabilities

x	y	z	P(z)
0	0	0	P_xP_y
0	1	1	$P_x(1\text{-}P_y)$
1	0	1	$(1\text{-}P_x)P_y$
1	1	1	$(1\text{-}P_x)(1\text{-}P_y)$

thus, for tow binary numbers each of length n bits, lets define that:

m1 denotes the number of bits where x=1 and y =1
m0 denotes the number of bits where y=0 and x=0
k1 denotes the number of bits where x=1 and y=0
and finally
k2 denotes the number of bits where x=0 and y=1
accordingly the total number of bits in x or y is
n=k1+k2+m0+m1

In this notation the probability of the output of X-OR (variable z) can be written as:

$$P(z) = \sum_{r=0}^{m1+k1}\binom{m1+k1}{r}(-p_x)^{r+mo+k2} * \sum_{i=0}^{m1+k2}\binom{m1+k2}{i}(-p_y)^{i+mo+k1} \quad (6)$$

As a result of Eq. 6, the X-OR process transforms the probability of the operands into binomial probability of the bits which is the same result found in [37, 38]. That is, for a normal probability distributed operands, the result of the X-OR process would be binomial distributed according to the distance between the input plain text (variable y) and the keys (variable x). Moreover, since the plain text and the keys are independent random variables, the probability of the output (variable z) is the product of the probabilities of y and x. The significance of Eq. (5) is that it explicitly shows the effect of the distance function in determining the probability of X-OR output where the distance function is defined as:

$$D(x,y) = \sum_{j=0}^{n-1} a_i \quad (7)$$

where

$$a_i = \begin{cases} 1 & if \quad x_i \neq y_i \\ 0 & if \quad x_i = y_i \end{cases}$$

This distance function in Eq. 7 is related to Eq. 5 by noting that:

$$D(x,y) = k1 + k2 \quad (8)$$

This suggests that the distance between the generated random keys and the plain text plays a major role in identifying the probabilities of the X-OR output. However, this result emphasizes that the optimum set of keys that would produce more secure encryption is dependent on the distribution of the plain text. The X-OR output presented in Eq. 5, in general can be written as a contribution of the keys distribution and the plain text distribution:

$$P(z) = P(x) * P(y) \quad (9)$$

where both the keys and the plain text are independent random variables.

Thus, if the plain text is uniformly distributed, then the probability of the X-OR output is a function of the probability of the keys as follows:

$$P(z) = \frac{1}{2^n}\sum_{r=0}^{m1+k1}\binom{m1+k1}{r}(-p_x)^{r+mo+k2} \quad (10)$$

In order to have a uniform distribution for the X-OR output, the keys distribution should be maintained to be uniform, in this case the output probability would be given as:

$$P(z) = \frac{1}{2^{2n}} \quad (11)$$

However, the random nature of the plain text makes the realization of uniform distributed X-OR variable far from attainable. To solve this difficulty, in this work, the keys are generated in different sets where each set has a specific p_x. The plain text will be X-ORed with each set of the keys. Finally, the ciphered text would be selected such that the distribution of the X-OR output is closest to uniform. This scheme would increase the complexity of DES like encryption algorithms, however, reducing the complexity of such algorithms is and open topic for future research.

3. Encryption Algorithm Development

The developed block encryption/ decryption algorithm is similar to the DES as shown in Figures 2 and 3. According to the structure of the encryption algorithm, it is clear that it maintains parallelism of encryption which simplifies the algorithm and provide high efficiency in terms of computational complexity. The main distinction between the conventional DES algorithm and the developed algorithm is in the diffusion of encryption keys sets to control the probability distribution of the X-OR. This diffusion of the keys adds more complexity to the algorithm since it implements multiple steps of probability transformation to the keys within each stage of the algorithm. This diffusion is meant in this algorithm to alter the probability of the encryption toward the uniform distribution. Moreover, the level of the diffusivity in the developed algorithm is variable. This flexibility would suggest the suitability of the developed algorithm for different type of data encryption. The steps of encryption and decryption of the algorithm are explained in the flow charts shown in Figures 2 and 3. The developed encryption starts by generating a set of keys of n length. Each set obeys a particular probability of one (px). The plain text then is X-ORed with each set. The keys that have fixed distance $D(x, y)$ from the text is selected as the encryption keys. In the decryption algorithm, the sets keys are X-ORed with the ciphered text and the distance $D(x, y)$ is measured between the text and the keys to select the original text. Thus, the distance $D(x, y)$ is considered as one to one function.

4. Results

For a proof of concept, The experiments starts by evaluating the change of the binomial distribution properties versus the change of the probability of bits. For balanced bit probability where each bit has a probability of 0.5 to be one and, equivalently, 0.5 probability to be zero. On the other hand, if the number of ones in the bit stream is lower than the number of zeros, then, the probability of each bit to be one would be lower than 0.5. In this case the probability is denoted as imbalanced probability. In the imbalanced probability, the binomial distribution, mean and STD would be shifted accordingly as clarified in Figure 4. Thus, the probability of XOR output can be mitigated by changing the probability of the keys used for encryption. It is worthy to note that the probability of XOR results is in correlation with both the keys and the plain text probabilities. Therefore, the assumption of uniform probabilities for the plain text is valid and, hence, the probabilities of the X-OR output are affected mainly by the probabilities of the generated keys. More specifically, the distance between the generated encryption keys and the corresponding plain text.

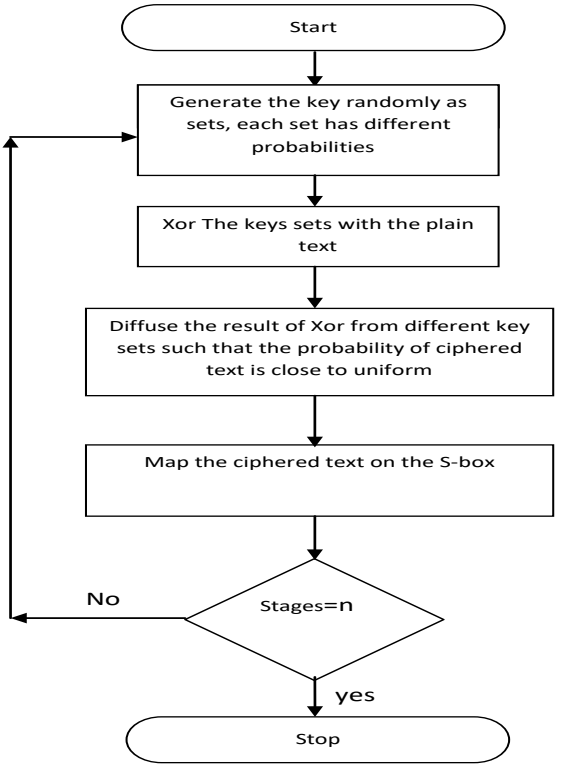

Fig. 2 Encryption Algorithm

the major factor affecting the probability of the outputs would be the distribution of keys.

The developed algorithm as extension of the regular DES encryption algorithm used a combination of keys that resulted by changing the probability of bits. This combination allowed the generation of keys with varieties of probability distributions. This scheme of keys may increase the robustness of the regular DES against hacking algorithms since it provides more complex combination of the keys that are considered as a summation of different distributions. The main concept in this analysis is to provide higher degree of robustness by using the regular simple encryption derived from the famous DES encryption/ decryption. The change of probabilities of the keys does not have any effect on the structure of the algorithm, therefore the simplicity features of fundamental DES is inherited in the developed algorithm.

The developed algorithm provided random encryption similar to the conventional DES as shown in Figures 2 and 3. This emphasizes that the mitigation of the keys probabilities could have direct effect on the complexity and robustness of the encryption algorithm whereas maintaining the low computational complexity of the conventional algorithm. This result facilitates the significance of statistical properties of the encryption keys

in the encryption/decryption algorithms and it may lead to the development of more simplified, yet robust, algorithms.

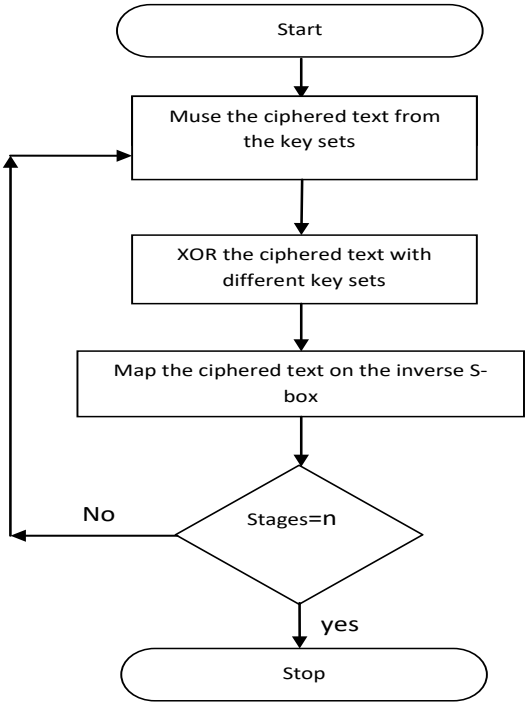

Fig. 3 Decryption Algorithm

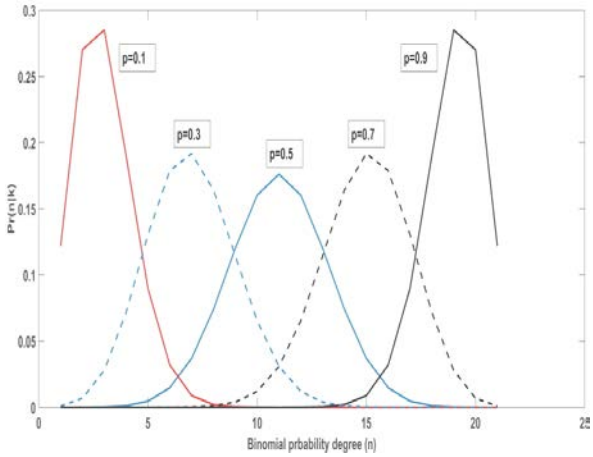

Fig. 4 Binomial Distribution for different probabilities (p)

5. conclusion

The main concept of encryption/ decryption is to provide high security in the data while maintaining the simple and low computational complexity. As XOR process is the core operation in most of the simple encryption/decryption algorithms, its input output statistical correlation is of high importance in elevating the robustness of the encryption/ decryption algorithms. Therefore, this paper, mainly, focused on providing a thorough study of the effect of keys distribution on the XOR output statistical properties. Experiments are provided to show that the distribution of the encryption keys for one stage of DES has direct effect on the XOR statistical properties. Thus, by controlling the encryption keys, the robustness of the encryption algorithm can be mitigated. This could be accomplished without altering the computational complexity of conventional encryption algorithms.

Further studies are needed to generate encryption keys obeying a varieties of probability distribution function such that the complexity of encryption algorithm is enhanced. Moreover, this scheme of analysis could, also, be applied on other newly developed algorithms and tested against differential crypt analysis hacking system.

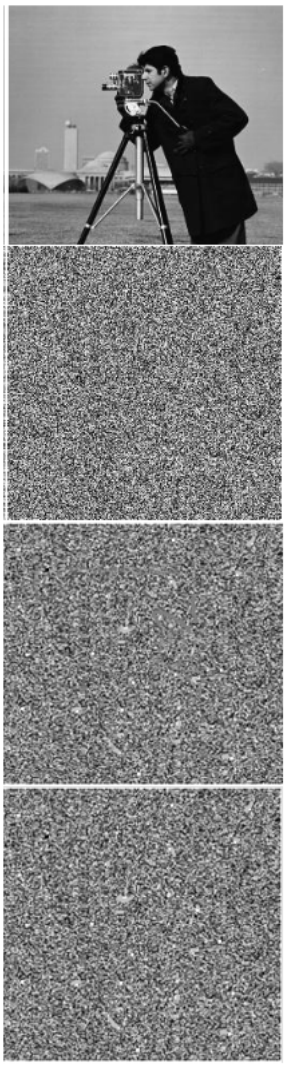

Fig. 5 Decryption Algorithm The camera man picture (up) and its encryption with first key set (second), encryption with second key set (third), and final encryption.

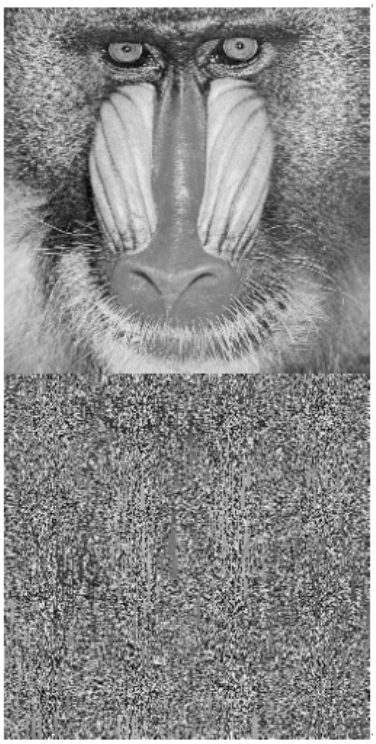

Fig. 6 Decryption Algorithm Monkey picture (up) and its final encryption (down)

References

[1] Stallings W., "Cryptography and Network Security 4th Ed," Prentice Hall, PP. 58-309, 2005.

[2] Brodney A, Asher J.,―Tales of the Encrypted‖; Availablefrom:http://library.thinkquest.org/28005/flashed/index2.shtml (2009).

[3] Deepak K. D. and Pawan D.,―Performance Comparison of Symmetric Data Encryption Techniques‖ ISSN: 2278 – 1323 International Journal of Advanced Research in Computer Engineering & Technology Volume 1, Issue 4 June 2012.

[4] Coppersmith D.,"The Data Encryption Standard (DES) Its Strength Against Attacks‖, IBM Journal of Research and Development, May 1994,pp. 243 -250.

[5] Chen J., Li X., Li W., Wan, W.,―An improved AES Encryption Algorithm‖, IET International Communication Conference on Wireless Mobile and Computing (CCWMC 2009).

[6] Bruce S., ―The Blowfish Encryption Algorithm Retrieved‖ http://www.schneier.com/ blowfish.html, 2008.

[7] Frank K. G., ‖Channel Attack secure Cryptographic Acceleration‖, 2006.

[8] Iana G. V., Anghelescu P., Serban G., ―RSA Encryption Algorithm Implemented on FPGA‖, International Conference on Applied Electronics, pp. 1-4, 2011.

[9] Rivest, R. L., Shamir, A., & Adleman, L.―Methods for Obtaining Digital Signatures and Public key cryptosystems‖, communication Of the

[10] ACM Vol. 21. pp. 120—126.1978.

[11] Frank kagan Giirkaynak,‖Channel Attack secure Cryptographic Acceleration,2006.

[12] DSA[Source:http://www.absoluteastronomy.com/topics/Digital_Signature

[13] DES[Source:http://dc532.4shared.com/joc/SrMkdWq3/preview.html]

[14] Blaze M., Diffie W., Schneider B., Shimomura T., Thompson E., and Wiener M., ―Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commercial Security‖, Report of Ad Hoc Panel of Cryptographers and Computer Scientists, 1996.

[15] Brodney A, Asher J., ―Tales of the Encrypt; available from:http://library.thinkquest.org/ 28005/flashed/index2.shtml,2009

[16] Chandramouli R., ―Battery power-aware encryption – ACM‖, Transactions on Information and System Security (TISSEC) Volume 9, Issue 2, May 2006.

[17] Deepak K. D. and Pawan D.,―Performance Comparison of Symmetric Data Encryption Techniques‖ ISSN: 2278 – 1323 International Journal of Advanced Research in Computer Engineering & Technology Volume 1, Issue 4 June 2012.

[18] J. Rodrigues, W. Puech, and A. Bors, "Selective encryption of human skin in jpeg images," in Image Processing, 2006 IEEE International Conference on. IEEE, 2006, pp. 1981–1984.

[19] A. Barbir, "Method and apparatus for performing simultaneous data compression and encryption," Sep. 19 2000, uS Patent 6,122,379.

[20] H. Chang and J. Liu, "A linear quadtree compression scheme for image encryption," Signal Processing: Image Communication, vol. 10, no. 4, pp. 279–290, 1997.

[21] S. Li, C. Li, K. Lo, and G. Chen, "Cryptanalysis of an image scrambling scheme without bandwidth expansion," Circuits and Systems for Video Technology, IEEE Transactions on, vol. 18, no. 3, pp. 338–349, 2008.

[22] T. Dan andW. Xiaojing, "Image encryption based on bivariate polynomials," in Computer Science and Software Engineering, 2008 International Conference on, vol. 6. IEEE, 2008, pp. 193–196.

[23] H. Ahmed, H. Kalash, and O. Allah, "Encryption efficiency analysis and security evaluation of rc6 block cipher for digital images," in Electrical Engineering, 2007. ICEE'07. International Conference on. IEEE, 2007, pp. 1–7.

[24] N. Flayh, R. Parveen, and S. Ahson, "Wavelet based partial image encryption," in Multimedia, Signal Processing and Communication Technologies, 2009. IMPACT'09. International. IEEE, 2009, pp. 32–35.

[25] F. Ahmed, M. Siyal, and V. Abbas, "A perceptually scalable and jpeg compression tolerant image encryption scheme," in Image and Video Technology (PSIVT), 2010 Fourth Pacific-Rim Symposium on. IEEE, 2010, pp. 232–238.

[26] I. Elashry, O. Allah, A. Abbas, S. El-Rabaie, and F. El-Samie, "Homomorphic image encryption," Journal of Electronic Imaging, vol. 18, p. 033002, 2009.

[27] N. El-Fishawy and O. Zaid, "Quality of encryption measurement of bitmap images with rc6, mrc6, and rijndael block cipher algorithms," International Journal of Network Security, vol. 5, no. 3, pp. 241–251, 2007.

[28] S. Kamali, R. Shakerian, M. Hedayati, and M. Rahmani, "A new modified version of advanced encryption standard based algorithm for image encryption," in Electronics and Information Engineering (ICEIE), 2010 International Conference On, vol. 1. IEEE, 2010, pp. V1–141.

[29] H. Elkamchouchi and M. Makar, "Measuring encryption quality for bitmap images encrypted with rijndael and kamkar block ciphers," in Radio Science Conference, 2005. NRSC 2005. Proceedings of the Twenty-Second National. IEEE, 2005, pp. 277–284.
[30] R. Gray, Entropy and information theory. Springer Verlag, 2010.
[31] Schindler, W., Lemke, K., & Paar, C. (2005, August). A stochastic model for differential side channel cryptanalysis. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 30-46). Springer, Berlin, Heidelberg.
[32] Qian, W., Riedel, M. D., Zhou, H., & Bruck, J. (2011). Transforming probabilities with combinational logic. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 30(9), 1279-1292.
[33] Rajashekarappa, K. M., & KA, S. D. (1963). comparative study on data encryption standard using differential cryptanalysis and linear cryptanalysis.
[34] Blackledge, J. M. (2011). Cryptography and Steganography: New Algorithms and Applications. Center for Advanced Studies Warsaw University of Technology.
[35] Hawkes, P., & O'Connor, L. (1999, May). XOR and non-XOR differential probabilities. In International Conference on the Theory and Applications of Cryptographic Techniques (pp. 272-285). Springer, Berlin, Heidelberg.
[36] Biham, E., & Shamir, A. (1991). Differential cryptanalysis of DES-like cryptosystems. Journal of CRYPTOLOGY, 4(1), 3-72.
[37] Qian, W., Riedel, M. D., Zhou, H., & Bruck, J. (2011). Transforming probabilities with combinational logic. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 30(9), 1279-1292.

Dr. Issam Traad received his PhD in Electrical and Communication Engineering from the Odessa National Academy of Communication in 2003. he is working as the dean of the faculty of Engineering in Jadara University/Jordan From 2014-2019. He is specialized in Radio Engineering and Television System, Image Processing, and Encryption algorithms.