Blowfish Algorithm - An Efficient Data Encryption Technique to Ensure Data Confidentiality

Depavath Harinath[1] , Archana Patil[2], Prema Kirubakaran[3], M.V.Ramana Murthy[4]

Depavath Harinath[1]
[1]Dept of Computer Science, Ramnath Guljarilal Kedia College of Commerce, Hyderabad, Telangana, India.
Mail id : harinath.depavath@gmail.co
Archana Patil[2]
[2]Dept. of Computer Science & Engineering, Rishi MS Institute of Engineering and Technology for women, Hyderabad, Telanagana, India.
Mail id : archanbpatil@gmail.com
Prema Kirubakaran[3A4RF]
[3]Associate Professor & HoD, Department of IT & IS, Faculty of Computing , Nile University of Nigeria Nigeria,
mail id :premakirubakaran78@gmail.com
M.V.Ramana Murthy[4]
[4]Dept. of Mathematics and Computer Science, Osmania University, Hyderabad, Telangana, India.
mailid :- mv.rm50@gmail.com

Abstract*— As technology continues to advance, the importance of secure communication systems will only increase.** ***Due to quick growth of networks, information security becomes more important to protect commerce secrecy and privacy. Encryption algorithm plays important role in information security. The proposed technique i.e., Blowfish algorithm aims to provide a secure and confidential communication system by combining steganography and cryptography. The integration of these two technologies provides an extra layer of security to the data being transmitted, thus ensuring the confidentiality of sensitive information. This paper illustrates about the Blowfish algorithm which provides an efficient data encryption technique to ensure data confidentiality.

Keywords—Blowfish Algorithm, Encryption, Decryption, Information Security.

I. INTRODUCTION

Information security has become critical in the age of digitization and data-driven landscapes. The Blowfish algorithm, a symmetric key block cipher, is a cryptographic institution. Cryptology is the art of secret writing and is the method of securing secret information by converting plaintext into ciphertext. Encryption transforms plaintext into an unreadable format, called ciphertext, to hide its contents from unauthorized individuals. Decryption technique helps in converting the ciphertext back into its original plaintext form. Cryptography protect information from unauthorized access, even from those who can access the encrypted data.

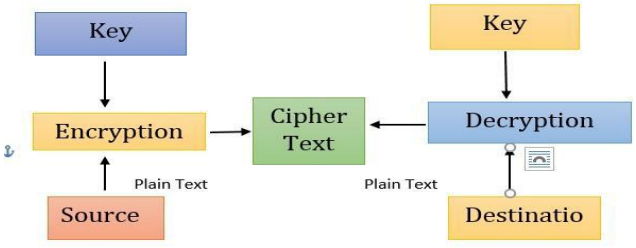

Figure (.1) Illustrates the process of cryptography

The purpose of cryptography is to allow for secure communication and storage of information over insecure networks. In order to solve specific problems various techniques and methods are designed; data encryption over networks to secure confidential information is a known key method. Not all encryption techniques have become popular for their demand, and Blowfish is no exception [1]. The Blowfish algorithm is a symmetric-key block cipher designed for secure data encryption and decryption. Blowfish, created in 1993 by Bruce Schneier, is a strong encryption method that ensures data confidentiality through a well-structured process. This paper illustrates about Blowfish algorithm an efficient data encryption technique to ensure data confidentiality.

II. RELATED WORK

Cryptography and steganography methods can be used to secure digital data during transmission over the internet. Cryptography involves the use of keys to encrypt and decrypt. data, making it difficult for unauthorized access. Steganography involves hiding data within an image, adding an additional layer of security. The combined approach of the cryptography using DES and steganography using DCT can be used to improve the data security by providing two levels of security. The experiment performed resulted with 58% of the success for securing the data, with 0.75 milliseconds/byte a computation time [3]. The process by which the data security can be improved over the internet depends on the cover image resolution which is being used. However, the time computed for securing the data over internet has a future scope in the research. Encrypted data sent over the computers connected in a network is the prime concern to protect the data from unauthorized access and simultaneously prevent accidental, deliberation or corruption of data. The communication lines are open to tapping among different terminals and other security risks, the protection of the data by means of computer system and to avoid any data loss or alteration of sensitive information transmitted over the network. The encryption methods used provide a solution to the various security concerns by converting the sensitive information into an unreadable format, to avoid unauthorized access by individuals for accessing or altering the data [4].

The Data Encryption Standard, an algorithm based on symmetric key encryption, uses a classic Feistel Network structure having 64 bits block size and 56 bits key size. However, the security level of DES is reduced by modifications, since the key size is reduced from 128 to 56 bits because of which the cipher was broken after 20 years. To address these weaknesses, an extended Feistel Network structure as a new variant called S-DES was proposed with which had a block size of 125 bits with 112 bit size of secret key. The improvement against the unauthorized access of the data over the network by using the attacks such as brute-force, differential, and linear cryptanalysis is enhanced by S-DES, compared to the original DES cipher scheme [5]. IBM in 1975 developed Data Encryption Standard (DES) algorithm and was one of the first widely used encryption algorithms for commercial data protection. Feistel structure takes 16 rounds by using 56-bit key to encrypt 64-bit blocks, utilizing 8 Sboxes and 16 sub keys of 48 bits each. The need of data transfer through DES has been phased out due to its small key size.

The encryption decryption keys used by DES, TDES, Blowfish, AES, and Twofish Symmetric encryption algorithms are same. The two different keys as public and private are used for encryption and decryption methods by asymmetric encryption methods. The key encryption plays an important role to maintain confidentiality of the data on the key, not the algorithm, which makes it secure even if the attacker is aware of the decryption algorithm. However, asymmetric encryption is slower than symmetric encryption which is widely used for key exchange and digital signatures [6].

The Blowfish Encryption Algorithm is a fast, compact, and simple encryption algorithm which uses a key for encryption and decryption of the data. It takes 16 rounds, each consisting of XOR operations with a function (F). Blowfish algorithm does not change the key because of which it is used for such applications where the key is not required, such as in communication links or files encryptors. However, Blowfish does not efficiently support frequent key changes or as a one-way hash function with reference to packet switching. The algorithm consists of a feistel network, which provides expansion of the key and encrypted data [7]. The expansion component divides into several sub key arrays for up to 448 bits, on the other hand the data encryption component has a simple function and iterates 16 times. Each round includes permutations and key operations which further consist of additions and XORs on 32-bit words. The enhanced Blowfish algorithm proposed by Agrawal and Mishra and the study utilizing image pixels to generate random numbers provide improved security and faster encryption and decryption times compared to the earlier Blowfish algorithm. The use of random numbers and image pixels in the encryption process makes it more difficult to predict and thus enhances the security level of the encrypted data. These studies provide potential solutions for enhancing the Blowfish algorithm and demonstrate the importance of continuous exploration and improvement in the field of cryptography. The modified Blowfish encryption algorithm by Mishra and Agrawal enrich the level of the security by reducing encryption and decryption time. The algorithm presented by [8] used variable key size of 448 bits and divides the image data into blocks for encryption. The modified Blowfish algorithm is as an exceptional standard encryption algorithm, as it gives efficient results as compared to earlier algorithm by increasing the number of rounds. The algorithm works efficiently and more securely when compared with algorithms like AES and DES which are symmetric encryption because of its capability of variable length key. Enhanced-Blowfish algorithm which works on random number generation on the image pixels is more secure compared to the original Blowfish algorithm due to its additional block switching method for scrambling data. The modified Blowfish algorithm is well-suited for protecting images and other data that require high security [9]. The various studies has been conducted for the improvement and upliftment in the security and performance of the Blowfish algorithm. Agrawal and Mishra improved the algorithm by using a random number generator to control the application of the F-function in each round. The overall execution time has been reduced by 14% by the modified Blowfish algorithm. The comparative study of Blowfish algorithm conducted by Ghorpade and Talwar and Sowbarnika et al. in comparison with other symmetric encryption algorithms concluded that Blowfish is the most capable encryption algorithm of variable length key and low memory consumption. Panda conducted a study for evaluating the computing resources consumed by various encryption algorithms including Blowfish and found that the performance in reference to throughput, encryption-decryption was better for AES [10]. These studies emphasized that as the number increased for processing rounds, the security for the algorithm was enhanced.

There are different parameters for evaluating the performance of various encryption algorithms depending on various factors like the size of the data being encrypted, the number of processing rounds, and the type of file being encrypted, among others.

Some studies have shown that AES provides better performance in reference of time for throughput as well as encryption-decryption, while others have concluded that Blowfish performs better, in reference of execution time and memory usage. These conclusions highlight the importance of considering different performance metrics and factors when evaluating the performance of encryption algorithms [11]. The various studies conducted for comparing the performance of Blowfish with other symmetric key algorithms such as AES and DES. The results of these studies are mixed, with some finding that Blowfish is superior in terms of execution time, required memory, and power consumption, while others find that AES provides better performance. Some studies also suggest that the optimal performance of Blowfish can be achieved when it is embedded in mobile devices with low power consumption and high throughput. This passage describes symmetric and asymmetric encryption algorithms, and focuses on some popular symmetric key encryption algorithms [12] as DES, DCT and blowfish. The key advantage in reference of the data confidentiality is with respect to the key and not the algorithm, thereby securing the data even if the attacker judge the decryption algorithm, the data cannot be decrypted without knowing the key. The encryption and decryption of data is done securely by two different keys used by asymmetric encryption as, private and public. The various asymmetric encryption algorithms which are popular are PGP, RSA, and SSH. The importance of protecting data transmitted over the internet is emphasized due to the increasing number of cases where confidential data is stolen by intruders [13].

III. BLOWFISH ALGORITHM

The Blowfish algorithm is a symmetric-key block cipher designed for secure data encryption and decryption. The Blowfish algorithm, a symmetric key block cipher, is a cryptographic institution. Blowfish, created in 1993 by Bruce Schneier, is a strong encryption method that ensures data confidentiality through a well-structured process.

A. Features of the Blowfish Algorithm

Here are some of the features of the Blowfish Algorithm.

1. Symmetric-Key Algorithm: Blowfish uses the same key for both encryption and decryption processes, making it a symmetric-key algorithm. This means that the party encrypting the data and the party decrypting it must possess the same secret key.

2. Block Cipher: Blowfish operates on fixed-size blocks of data. The standard block size is 64 bits, but it can work with smaller blocks as well. If the input data is not a multiple of the block size, padding is typically applied to the data before encryption.

3. Variable-Length Key: One of the unique features of Blowfish is its ability to accept variable-length encryption keys, making it adaptable to different security requirements. The key length can range from 32 to 448 bits, and it's expanded during encryption to generate a series of subkeys.

4. Feistel Network Structure: Blowfish employs a Feistel network structure in which data is divided into two halves, subjected to a series of rounds of operations, and then recombined. This structure allows for efficient encryption and decryption processes.

5. F-Function: The F-function is a core component of the Blowfish algorithm. It involves a combination of XOR (exclusive OR), substitution, and permutation operations, which contribute to the algorithm's strength and security.

6. Key Expansion: Before the actual encryption process, Blowfish generates a series of subkeys based on the provided key. These subkeys are used during the encryption and decryption rounds to introduce complexity and security.

7. Complexity and Security: Blowfish is designed to be highly secure against various cryptographic attacks. The complex F-function and key expansion process make it resistant to brute force and differential cryptanalysis.

B. Working of Blowfish Algorithm:

The Blowfish algorithm is renowned for its robust encryption and relatively simple structure. To truly understand its inner workings, let's dive into the encryption process step by step, shedding light on each intricate operation that contributes to its security.

1. Key Generation and Subkey Creation

The algorithm begins with a secret encryption key, which is used to generate a series of subkeys. Blowfish's subkey generation involves a complex process that enhances security. Here's how it works

Initialization of the P array and S boxes:

Blowfish uses a combination of pi (hexadecimal digits of π) and a series of S boxes (substitution boxes) to initialize its internal data structures.

Key Expansion:

The secret key is expanded using a key expansion routine. During this process, the key is used to modify the P array and S boxes. The subkeys derived from the key ensure that the encryption process remains secure and resistant to known attacks.

2. Data Encryption

Once the subkeys are generated, the algorithm proceeds with the encryption of the data block. The data block is divided into two 32 bit halves, L (left) and R (right). A series of rounds (typically 16) are performed on these halves to ensure strong encryption.

Feistel Network Rounds:

The algorithm employs a Feistel network structure, which involves applying a series of operations to the L and R halves in each round. These operations include XOR (exclusive OR) with the current subkey, applying the F function to R, and swapping L and R.

F function Operation

The F function takes the 32 bit R half and applies several steps Subkey XOR The current subkey is XORed with R. Substitution R is divided into four 8 bit quarters. Each quarter is used to index a specific S box, and the resulting values are combined permutation. The results from the S boxes are combined and transformed using the P array. This step introduces confusion and diffusion, crucial components of cryptographic security. Final Round After all rounds are executed, the resulting L and R halves are swapped one last time.

3. Data Decryption

The decryption process is essentially the reverse of encryption. The encrypted data block is divided into L and R halves, and the algorithm performs rounds in reverse order using the same subkeys

Feistel Network Rounds (Decryption)

Similar to encryption, rounds involve applying operations to L and R, but this time in reverse order using the corresponding subkey.

1. F function Operation (Decryption) The F function is applied in reverse, with the subkey XOR and S box steps inverted. This reverse operation successfully decrypts the data block.
2. Final Round (Decryption) After all decryption rounds, the decrypted L and R halves are combined to obtain the original data block.

For an even clearer understanding, let's revisit the Java code example provided earlier. The code demonstrates how to use the Java Cryptography Architecture to implement the Blowfish algorithm for encryption and decryption. This example showcases the actual process described above in a concise and practical manner.

Therefore, Blowfish algorithm which is an enhanced symmetric-key encryption algorithm and gives better results on block size of 64bits and a variable key length between 32-448 bits. Blowfish algorithm is a 16-

round Feistel cipher that uses large key-dependent S-boxes, similar to CAST-128 but with a different structure. Blowfish is fast, except during key changes, and is freely available to anyone, contributing to its popularity [14]. The Blowfish algorithm is also known for its simplicity and speed, making it a popular choice for certain applications. However, as mentioned, it is not suitable for use cases with frequent key changes and is limited to a maximum key length of 448 bits. It is fair enough if we consider the required specifications of a given application before selecting the best symmetric encryption algorithm. The symmetric-key encryption Blowfish algorithm uses the same key for encryption and decryption; block size of 64-bits and key ranging from 32-448 bits. It is to enhance the applications where frequent changes does not occur in the key, and is considered faster than other encryption algorithms while implemented on large data caches using 32-bit microprocessors. [15].

Choosing the right symmetric encryption algorithm depends on various factors such as encryption/decryption speed, key size, security, efficiency, and compatibility with the devices and software being used [16]. Additionally, the choice of symmetric encryption algorithm also depends on the type of data that is being encrypted and the level of security required. For instance, if security is a top priority, then algorithms such as AES or Twofish might be more suitable, while for faster encryption and decryption, algorithms such as Blowfish may be considered as a better option. It is must to weigh the trade-offs between security and efficiency when choosing a symmetric encryption algorithm [17].

The implementation of the Blowfish algorithm in C# using Visual Studio 2010 on Windows 7 involves using a 128-bit encryption key, either loaded from a file or generated, to generate the P-box and S-box arrays. The input message is encrypted using ECB mode of Blowfish encryption and divided into 64-bit data blocks which are encrypted with a specified number of rounds [18]. The decoding phase involves inverse wavelet transform on the stego image, extracting a bit stream, and using the decryption module of Blowfish to get the input plain text [19].

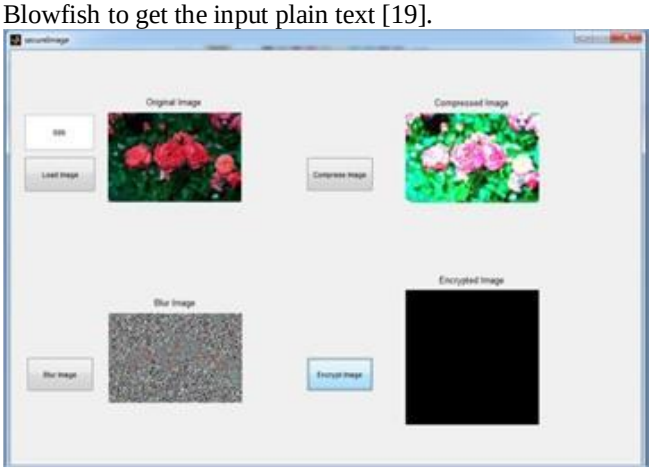

Figure 2: Embedded model showing compressed, blurred and encrypted images in steps to propose secured model and add the new image to the database

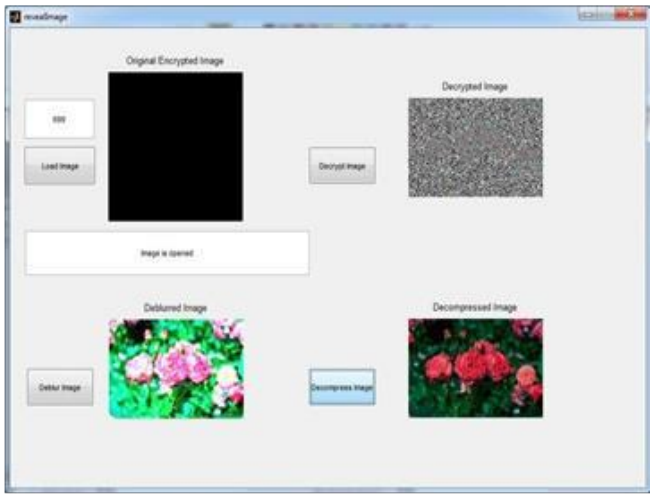

Figure 3: Embedded model using compressed, blurred and encrypted to propose secured system model for revealing the secured image.

Table 1: Table display the results of enhanced BLOWFISH Algorithm Implemented by using the dataset of 50 images **S. No.**	**File Size**		**Proposed Method**		**Existing Method**	
(Kb)	**Encryption Time (Seconds)**	**Decryption Time (Seconds)**	**Encryption Time (Seconds)**	**Decryption Time (Seconds)**		
1	1173.701	2.565684	0.646086	10.60768	8.28948	
2	711.6865	1.736618	0.625836	6.973018	6.549314	
3	589.6384	1.454034	0.62048	5.106201	4.66321	
4	565.7803	1.236578	0.626158	3.839002	3.249121	
5	730.1074	1.958067	0.623041	7.336028	5.407637	
6	1371.4	2.086155	0.622586	11.74675	10.0345	
7	434.9331	0.988519	0.626537	2.742176	2.296539	
8	1006.777	2.320674	0.62018	10.07432	8.25755	
9	236.466	0.72339	0.623044	2.458714	2.13564	
10	593.4033	1.656598	0.626732	6.34218	6.231172	
11	1147.318	2.722274	0.622052	11.73515	10.14766	
12	785.3652	2.083762	0.623586	9.15454	7.96725	
13	786.7613	2.075031	0.6237	8.25787	7.525522	
14	537.7011	1.760223	0.624206	6.473722	5.556662	
15	432.4409	1.333232	0.619566	5.37134	3.502222	
16	1170.136	2.660773	0.523029	10.66178	9.37332	

17	651.3251	1.818772	0.625462	7.544757	7.556526
18	1013.5	2.54001	0.623553	11.442226	8.7577
19	689.675	1.893676	0.62275	7.271263	7.295474
20	724.3528	1.740501	0.623433	7.226233	5.261662

IV. RESULT ANALYSIS

The performance with reference to the proposed algorithm using wavelet and contour let transforms was measured using three sets of image sizes: 256x256, 306x648, and 512x512, each containing ten colored images [20]. The performance is likely evaluated based on the level of protection provided to the data and the visual quality of the stego images. The reason for selecting image sizes of 256x256, 306x648, and 512x512 was to improve upon the results obtained in [21] where the image size used was 306 x 648. The input test images used as cover images are of varying levels of complexity and smoothness. To assess the level of distortion in the final stego image, the performance is measured using PSNR and MSE with the payload. The payload is calculated as the total number of bits that can be embedded into the number of bits of the input cover image and is expressed as a percentage [22]. PSNR is used as a measure of the quality of the image after processing, with a higher value of PSNR indicating better performance. PSNR is calculated from MSE, which measures the difference between two i x j images and is defined as follows: (1) MSE = (1/i * j) * SUM (Xi - Yi)^2 (2) PSNR = 10 * log10 (MAX^2 / MSE) where Xi and Yi are the pixel values of the two images and MAX is the maximum possible pixel value [23].

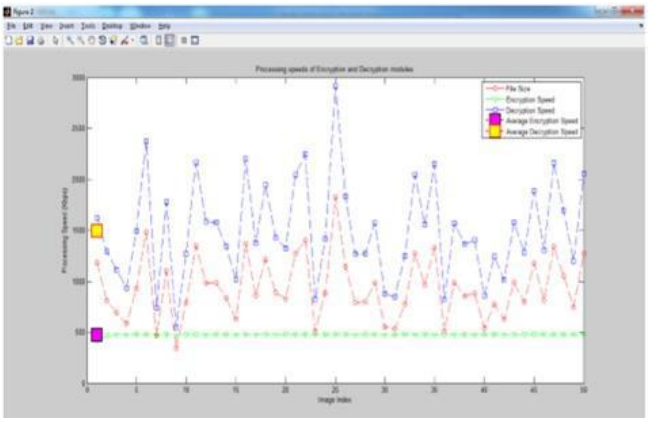

Figure 4: Encryption and Decryption graphs processing speeds using Blowfish Algorithm

The Blowfish algorithm uses wavelet and contour let transforms along with Blowfish algorithm for text data hiding in images. The performance of the algorithm was measured using three different image sizes, 256x256, 306x648, and 512x512, and the results were evaluated using PSNR and MSE values [24]. The experiments were performed with varying the correlation factor and input cover images and the results were compared. The algorithm achieved good results and improved upon previous work by Ali Al-Taby. It sounds like the experiments described were focused on evaluating the performance of a digital image steganography technique. Steganography is the practice of hiding information within another data object, in this case an image. The experiments involved varying the correlation factor (α) and input cover images, and using the Symlet 4 wavelet family for the image transform. The performance was evaluated by measuring the PSNR value, with a higher PSNR indicating that it is more difficult to detect the hidden message [25].

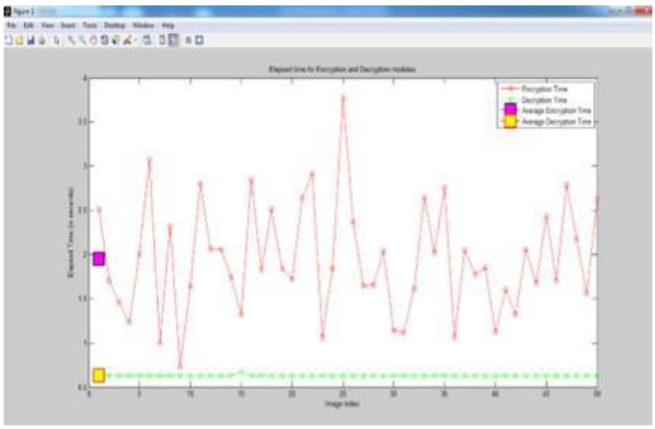

Figure 5: Encryption and Decryption graphs for time using Blowfish Algorithm

The results of the experiments indicate that the contourlet transform performs better than the wavelet transform when both algorithms are applied to the same image of the same size. This conclusion is illustrated in Figure. 3 and Figure.4. The experiments also show that the performance of the contour let transform remains consistent when the image size is increased, while the performance of the wavelet transform may be affected. Additionally, the results of the proposed algorithm were compared with the results of a previous study by Ali Al-Taby, and it was found that the contourlet transform performed better than the wavelet transform used in Al-Taby's work. The results presented in the table suggest that the proposed Blowfish encryption algorithm outperforms the existing AES encryption methods in terms of performance, particularly with regards to message lengths of 512 bytes or greater [25]. The fast implementation of the Blowfish algorithm, combined with its ability to fully utilize the multiple improved encryption patterns and validations for its initialized overhead, allows it to achieve near optimal performance. Additionally, the proposed algorithm has been configured with a 128-bit block size and a fixed s-box implementation, which has further improved its performance. The added validation method provides added flexibility and robustness to the proposed algorithm.

V. CONCLUSION

The Blowfish algorithm's security lies not only in its use of the Feistel network structure and the F function but also in its intricate subkey generation process. By meticulously expanding the original key into a series of subkeys and performing numerous rounds of operations, Blowfish ensures that the encrypted data remains secure and resistant to various attacks. Understanding the detailed encryption process allows developers and security professionals to appreciate the depth of thought and expertise behind this renowned cryptographic method. As technology continues to

advance, the importance of secure communication systems will only increase, making research and development in this area increasingly relevant and crucial. The use of the enhanced Blowfish encryption algorithm in combination with the Pixel Indicator technique provides an additional level of security to the secret messages being transmitted. The Blowfish algorithm is a symmetric-key block cipher that has been widely used for data encryption. Its enhanced version provides improved security compared to other encryption techniques. The Pixel Indicator technique is used to embed the encrypted text inside raw images, making it difficult for unauthorized users to access the secret messages. The combination of these two techniques provides a secure method of communication, making it more difficult for the transmitted information to be intercepted and decrypted by unauthorized users. The proposed technique aims to provide a secure and confidential communication system by combining steganography and cryptography. The integration of these two technologies provides an extra layer of security to the data being transmitted, thus ensuring the confidentiality of sensitive information.

The conclusion of the experiments is that the proposed systems outperformed previous work in the field of digital image steganography in terms of PSNR values and image imperceptibility. The experiments focused on comparing the performance of wavelet and contourlet transforms for colored images, which is an area where limited research has been done. The results showed that the proposed systems were able to maintain high image quality while operating at high PSNR levels. Currently, only one page of a Microsoft Word file is used as input information, but the algorithms could be modified to handle more pages or more characters. Additionally, the use of texture images could potentially lead to better results, as these images have many edges and contours that could be utilized. Further research in this area is needed to fully explore the potential of texture images for image steganography. The paper presents a new approach to digital security by combining three techniques: compression using DES and DCT, encryption using BLOWFISH. These techniques are intended to provide multiple layers of security for sensitive information. The proposed hybrid algorithm is aimed at providing a more secure and effective security system compared to traditional approaches. The proposed system could be useful for various applications where data privacy and security are a concern, such as financial transactions, medical records, and personal information.

REFERENCES

[1] David A. Huffman, ―A Method for the Construction of Minimum - Redundancy Codes‖, Proceedings of the I.R.E., September, pp.1098–1102, 1952.

[2] M. Kharrazi, H. T. Sencar, N. Memon, Image Steganography: Concepts and Practice, Lecture Note Series, Institute for Mathematical sciences, National University of Singapore, 2004.

[3] W. Pennebaker and J. Mitchell. ―JPEG STILL IMAGE DATA COMPRESSION STANDARD". van Nostrand Reinhold, 1993.

[4] L. Marvel ―Image Steganography for Hidden Communication". Ph.D. Dissertation, Univ. of Delaware, Dept of EE, 1999.

[5] Robert T. McKeon ―Strange Fourier Steganography in Movies‖, Proceedings of IEEE EIT 2007.

[6] R. Gonzalez and R. Woods, ―Digital Image Processing‖, Sec. Edition. Pp.373-374.

[7] S. Mallat, ―A theory for multiresolution signal decomposition: the wavelet representation‖, IEEE Pattern Anal. and Machine Intell., Vol.11, No.7, pp.674-693, 1989.

[8] Marcelloni, Francesco, and Massimo Vecchio. "A simple algorithm for data compression in wireless sensor networks." Communications Letters, IEEE, Vol.12, no.6, pp.411-413, 2008.

[9] Srisooksai, Tossaporn, Kamol Kaemarungsi, Poonlap Lamsrichan, and Kiyomichi Araki. "EnergyEfficient Data Compression in Clustered Wireless Sensor Networks using Adaptive Arithmetic Coding with Low Updating Cost."International Journal of Information and Electronics Engineering Vol.1, no.1, pp.85-93, 2011.

[10] Xi Xu et al. [2012]: Image Compression Using Radon Transform With DCT : Performance Analysis. International Journal of Scientific Engineering and Technology. Vol.2, Issue.8, pp.759- 765, 2012.

[11] Dong-U Lee et al. [2009]: Precision-Aware Self-Quantizing Hardware Architectures for the Discrete Wavelet Transform, IEEE Vol.21, 2009.

[12] Dolfus, Kirsten, and Torsten Braun. "An evaluation of compression schemes for wireless networks." In Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2010 International Congress on, IEEE, pp.1183-1188. 2010.

[13] A Survey on Data Compression in Wireless Sensor Network Naoto Kimura and Shahram Latifi Electrical and Computer Engineering, University of Nevada, Las Vegas{naoto, latifi}@egr.unlv.edu International Conference on Information Technology: Coding and Computing (ITCC'05) IEEE 2000.

[14] Medeiros, Henry Ponti, Marcos Costa Maciel, Richard Demo Souza, and Marcelo Eduardo Pellenz. "Lightweight Data Compression in Wireless Sensor Networks Using Huffman Coding." International Journal of Distributed Sensor Networks 2014.

[15] Liang, Yao. "Efficient temporal compression in wireless sensor networks." InLocal Computer Networks (LCN), 2011 IEEE 36th Conference on, IEEE, pp.466-474. 2011.

[16] Marcelloni, Francesco, and Massimo Vecchio. "A simple algorithm for data compression in wireless sensor networks." Communications Letters, IEEE Vol.**12**, Issue.6, pp.411-413, 2008.

[17] Paar, Christof. "Applied cryptography and data security." Lecture Notes), Ruhr-Universität Bochum (http://www. crypto. ruhr-uni-bochum. de), 2000.

[18] Hager, Creighton TR, Scott F. Midkiff, Jung-Min Park, and Thomas L. Martin. "Performance and energy efficiency of block ciphers in personal digital assistants." In Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on, IEEE, pp.127-136, 2005.

[19] Agarwal, Navita, and Himanshu Sharma. "An Efficient Pixel-shuffling Based Approach to Simultaneously Perform Image Compression, Encryption and Steganography." International Journal of Computer Science and Mobile Computing (IJCSMC) 2, no. 5, pp.376-385, 2013.

[20] M. P. R. Kamble, M. P. S. Waghamode, M. V. S. Gaikwad, and M. G. B. Hogade, "Steganography techniques: A review," International Journal of Engineering, Vol.2, no. 10, 2013.

[21] H. Inoue, A. Miyazaki, and T. Katsura, "An image watermarking method based on the wavelet transform," in Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348), IEEE, Vol.1, pp.296–300, 1999.

[22] K. Raja, K. Kumar, S. Kumar, M. Lakshmi, H. Preeti, K. Venugopal, and L. M. Patnaik, "Genetic algorithm based steganography using wavelets," in International Conference on Information Systems Security. Springer, pp. 51–63, 2007.

[23] C.-K. Chan and L.-M. Cheng, "Hiding data in images by simple lsb substitution," Pattern recognition, Vol.37, no.3, pp.469–474, 2004.

[24] R. El Safy, H. Zayed, and A. El Dessouki, "An adaptive steganographic technique based on integer wavelet transform," in 2009 International Conference on Networking and Media Convergence. IEEE, pp.111–117, 2009.

[25] A. M. Fard, M.-R. Akbarzadeh-T, and F. Varasteh-A, "A new genetic algorithm approach for secure jpeg steganography," in 2006 IEEE International Conference on Engineering of Intelligent Systems. IEEE, pp.1–6, 2006.

[26] D. R. ElShafie, N. Kharma, and R. Ward, "Parameter optimization of an embedded watermark using a genetic algorithm," in 2008 3rd International Symposium on Communications, Control and Signal Processing. IEEE, pp.1263–1267, 2008.

[27] Depavath Harinath,et.al,"A Review on Security Issues and Attacks in Distributed Systems," Journal of Advances in Information Technology(JAIT), California, USA, Vol. 8, No. 1, pp. 1-9, February, 2017. doi: 10.12720/jait.8.1.1-9

[28] Depavath Harinath, "Enhancing Data Security Using Elliptic Curve Cryptography in Cloud Computing", *International Journal of Science and Research (IJSR),* https://www.ijsr.net/archive/v5i7/v5i7.php, Volume 5 Issue 7, July 2016, 1884 - 1890, DOI: 10.21275/v5i7.ART2016624

[29] Depavath Harinath,et.al, "An Interplanetary File System Framework For Invocation and Machine Learning Model Training", Bulletin For Technology And History Journal , Volume 24, Issue 2, 2024, Page No:151-157, DOI:10.37326/bthnlv22.8/1513

[30] Depavath Harinath, et.al, "Enhancing Security and Malware Detection in IoT Devices using Random Forest Algorithm", Technische Sicherheit (Technical Security) Journal, Volume 24 Issue 3, 2024, ISSN:1434-9728 and Page No. 169 – 176, DOI:22.8342.TSJ.2024.V24.3.01304

Author Profile

Depavath Harinath, Assistant Professor, received Master of Computer Applications degree from Sreenidhi Institute of Science and Technology, an autonomous institution approved by UGC, Accredited by NAAC with 'A+' grade and accredited by NBA, AICTE, New Delhi – permanently affiliated to JNTU, Hyderabad, Telangana, India. Having more than twelve years of experience in teaching and already published 21 manuscripts in different international journals. Now working as Assistant Professor, Dept. of Computer Science, Ramnath Guljarilal Kedia College of Commerce, Hyderabad, Telangana, India. Research field includes Computer Networks, Network Security, Artificial Intelligence and Machine Learning.

Dr.Archana Patil B.E ,MTech(CSE) & Ph.D.(CSE)working as assistant professor in Rishi MS Institute of Engineering & Technology for Women, Hyderabad**.** She has 12+ years of teaching experience. She has published many papers in reputed International and National Journals and Conferences with good citation. She already published many books and holds more than 10 Patents on different area of Computer Science and engineering. Her area of interest includes Cloud Computing, Data structure, Computer Graphics, IOT, Mobile Adhoc Network, Cyber Security, and Machine learning

Dr.A.Prema Kirubakaran is an Associate Professor and Head of the Department of Information Technology & Information Systems. She has 21 years of Teaching Experience and 15 years of research experience in Indian universities and other foreign universities. She has published 51 articles in National and International Journals. She has received many awards for her research work, Teaching skills, and administration abilities. She is an article reviewer of many international journals, including Springer Journals, Taylor & Francis, CSI, ICT Research Journal, AJCIS Research, and Online reviewer for Shiksha higher studies. She holds 4 Indian patents and one UK patent.

Prof. M. V. Ramana Murthy, Professor in department of mathematics and computer science, Osmania University, since 1985. Obtained PhD degree from Osmania University in 1985 and visited many a countries across the globe in various capacities and participated in many academic programs. Research fields includes computational plasma, Artificial Neural Networks, and Network securities.