IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Special Issue-16, May 2019 E-ISSN: 2347-2693

Analysis on Encryption and Compression Techniques for Information Security

Gowtham Mamidisetti[1*], Ramesh Makala[2], Ravi Teja K [3]

[1] Dept. of Computer Science and Engineering, Acharya Nagarjuna University, Guntur, India
[2]Dept. of Information Technology, RVR & JC College of Engineering, Andhra Pradesh, Guntur, India
[3] Dept. of Computer Sciences and Engineering, Shri Vishnu engineering college for women, Bhimavaram, India

[*]*Corresponding Author: mamidisetti.gowtham@gmail.com Tel.: +91 9951444611*

DOI: https://doi.org/10.26438/ijcse/v7si16.105112 | Available online at: www.ijcseonline.org

Abstract— This paper has investigated some of the encryption and compression techniques. The paper has also examined the implication of these techniques for the future of information security. Indeed, the threat environment has combined with corporate networks and made data security more complex. Therefore, an increasing number of data network users, who have turned to the cloud, have strived to secure their information systems beyond conventional virus scanners and firewalls. With an increasing demand for responsive information security systems, various techniques of data encryption and compression have evolved. Some of the devices, platforms, or information systems that have been targeted by these trends include websites, personal databases, and computers. Some of the specific algorithms that have been employed in data encryption include River-Shamir-Adleman (RSA) algorithm (in asymmetric encryption) and include Rivest Cipher (RC6) and Data Encryption Algorithm (DES) (in symmetric encryption). On the other hand, selected algorithms that have been used towards successful lossless data compression include lossless predictive coding, Arithmetic coding, Lempel-Ziv-Welch (LZW) compression, Huffman coding, and Run Length Encoding. These algorithms have gained application to situations involving platforms such as word processing files, tabular numbers, and executable codes. For lossy data compression, some of the algorithms that have gained increasing application include lossy predictive coding, wavelet coding, and transform coding. The implication for the future is that data encryption and compression techniques that will be responsive to the dynamic nature of the threat environment might prove successful in achieving the intended goals of data encryption and compression processes, hence assuring information system security.

Keywords—Compression,Encryption,RSAalgorithm,RC6,Huffmancoding.

I. INTRODUCTION

As an increasingly sophisticated threat environment combines with corporate networks that have continually grown in complexity, the majority of network users have had to ensure that their security systems are ramped up beyond conventional virus scanners and firewalls [1]. On a global scale, it is also worth indicating that data breaches have become increasingly serious and common. Thus, the need for information network security has proved more urgent. An exponential increase in data loss risks has been attributed further to the generation and consumption of most of the corporate data through mobile devices [2]. From the majority of the previous scholarly studies, information or data security entails protection approaches through which unauthorized access to platforms such as websites, personal databases, and computers is prevented [3-5]. Also, information security has been perceived as a process through which databases and the data are protected from malicious, deliberate, or accidental damages and modifications through unauthorized access [6]. With progress in the information security practice, encryption and compression techniques have emerged. Particularly, cryptography constitutes the hiding of messages whereby the target information's safety and security are assured [7]. Thus, data encryption involves the encoding of data to ensure that it can only be read by authorized users [2, 3]. On the other hand, compression constitutes the reduction of the number of bytes or bits required towards the representation of certain data sets – while ensuring easy storage and transmission of the information [7-9]. With the primary goal of information security observed to constitute access control, non-reproduction, integrity, authenticity, and confidentiality [8, 9], this paper examines some of the encryption and compression techniques characterizing the contemporary world of information security.

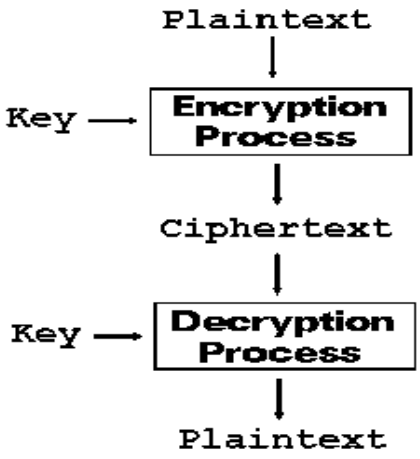


Figure 1: The data encryption process
Source: Mathur and Alam (2013)

II. COMPARING ENCRYPTION ALGORITHMS

Regarding data encryption, two major types of keys have been documented. They include asymmetric and symmetric keys. Indeed, symmetric keys rely on one key for decryption and encryption processes [9, 10]. The secret keys used exist in the form of Block Ciphers and Stream Ciphers. Particularly, stream ciphers operate in such a way that only one bit is processed at a time [7]. On the other hand, block ciphers ensure that blocks of beats are processed simultaneously [11]. Hence, symmetric encryption requires the receiver and the sender to have the same secret key. The figure below illustrates the symmetric key encryption.

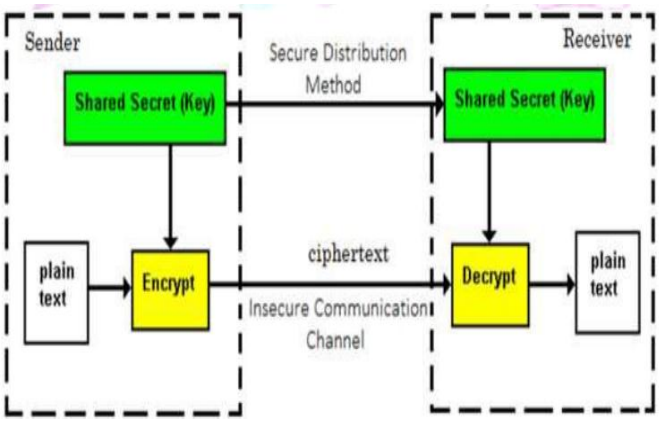
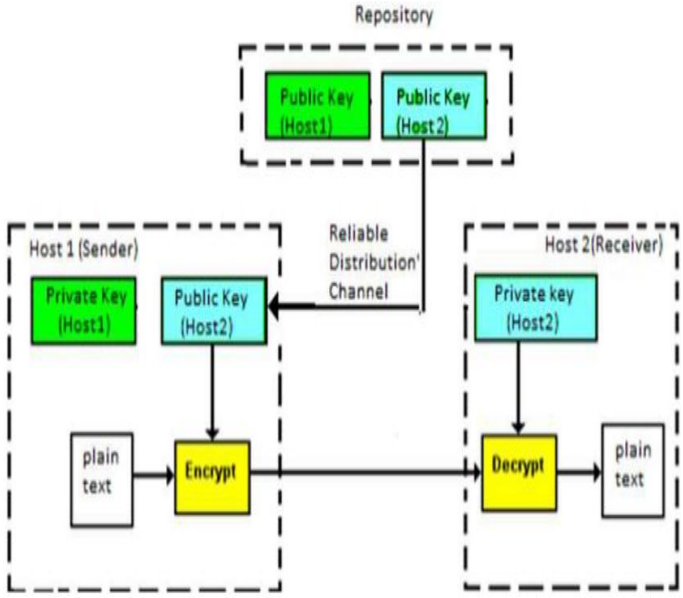


Figure 2: Symmetric key encryption
Source: Chandra and Bhattacharyya (2014)

For the case of asymmetric key encryption, it is worth noting that the technique reflects a two-key system whereby two separate keys exist [12]. Particularly, one of the keys is a public key while another one is a secret key. Therefore, the role of one of the keys involves decrypting the information while another key involves encrypting the data. When a public key is utilized, the implication is that computers that send the information encrypt messages to receivers via the use of specified or selected public keys of the target user, as well as the utilization of the private key of the sender. On the other hand, the receiver's computer aids in the decryption of the selected messages via the utilization of their private keys, before focusing on the public key of the sender [12, 13]. However, it is imperative to highlight that when the data encryption-decryption process involves asymmetric keys, it tends to be slower than the case in which symmetric keys are involved [14]; especially due to a large number of bits (or length of keys). The figure below illustrates the asymmetric encryption in use.



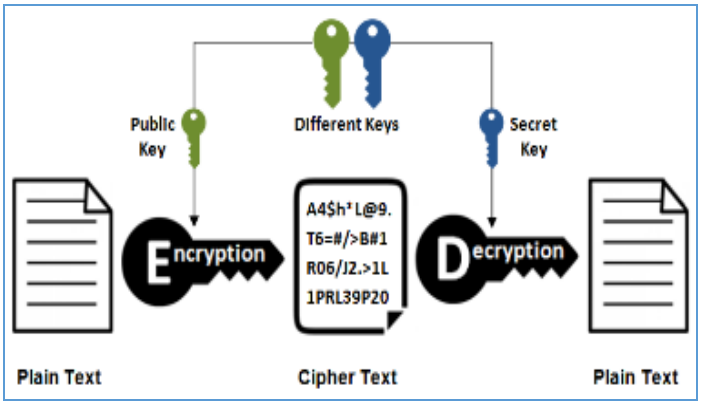


Figure 3: Asymmetric encryption highlighted
Source: Hercigonja (2016)

Given that asymmetric encryption is applied during the encryption and exchange of secret keys, it remains notable that its drawback is that if a receiver or sender loses the key

or that the secret key is intercepted, the communication is unlikely to occur securely because the system tends to be broken [12, 13].

For asymmetric encryption, one of the algorithms that have been employed is the case of River-Shamir-Adleman (RSA) algorithm. Notably, this algorithm applies to digital signatures and constitutes the decryption, encryption, and key generation processes [4, 6]. Regarding symmetric encryption, two major algorithms that have been employed include Rivest Cipher (RC6) and Data Encryption Algorithm (DES) [7]. Indeed, DES refers to a popular block cipher algorithm operating on 64-bit block sizes. For plaintext, DES algorithm offers descriptions to gain 64 ciphertext bits. Particularly, the algorithm employs 56 internal key or sub-key bits, with 64-bit-length external keys used to generate internal keys [5].

For RC6, the encryption algorithm applies to symmetric keys in which up to 128-bit block sizes are used. Imperatively the key sizes exhibit variations such as 256, 192, and 128 bits. Thus, the algorithm divides 128-bit blocks to obtain four 32-bit blocks. However, the algorithm conforms to six basic operating rules. These rules include bending bits to the left, sliding bits to the right, integer multiplication, exclusive OR, subtraction, and number [11-13]. Based on these insights, it is worth inferring that RC6 algorithm's primary process entails decryptions of encryptions, as well as key scheduling.

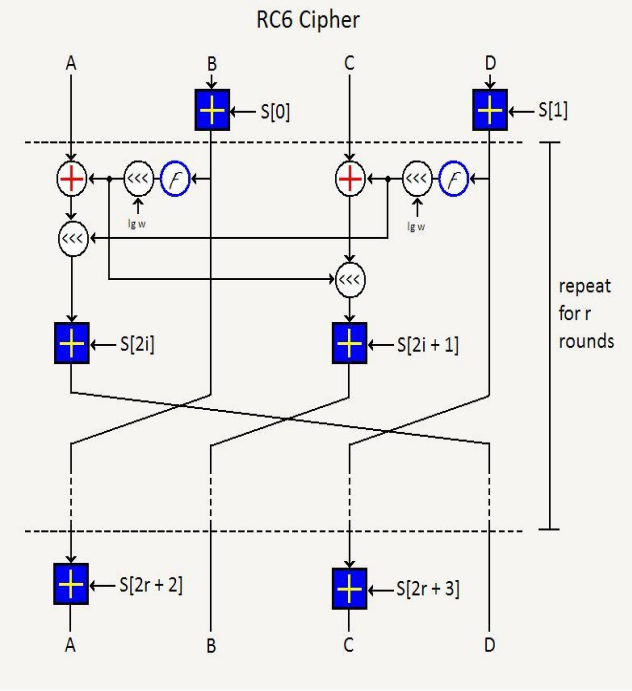


Figure 4: RC6 algorithm
Source: Mahajan and Sachdeva (2013) [23].

III. Comparing Compression Algorithms

Relative to the practice of data compression, it is important to note that the central motivation lies in the need to increase the transmission rate while reducing communication costs, having steered effective use of available bandwidth [14]. Thus, the role of compression algorithms lies in the reduction of the storage space in which data representation redundancy and the number of bits required for data representation are reduced significantly [14, 15]. This growing demand is informed by the contemporary era of technology and science whereby the volume of digital data transmission has grown tremendously; in terms of computer, video, audio, image, and text programs [14-16]. Indeed, the latter trend is attributed to the increasing number of people moving towards the cloud. Thus, compression techniques have emerged to ensure that the number of bits through which data can be represented is reduced, eventually reducing the storage space and also increasing the rate of data transmission [18].

From the current literature, compression techniques can be lossy or lossless. For lossless compression, the original data and the restored data are identical. From the previous scholarly findings, lossless compression applies to different forms of data. Examples include word processing files, tabular numbers, and executable codes in which the aim is to avoid losing any bit of information [13-15]. Major techniques for lossless compression include lossless predictive coding, Arithmetic coding, Lempel-Ziv-Welch (LZW) compression, Huffman coding, and Run Length Encoding [16, 17]. Regarding lossy compression, it is notable that the original data and the restored data are not identical, implying that when this technique is employed, some data tends to be lost [18]. Indeed, lossy compression has been affirmed to gain application to situations involving images, as well as other forms of application in which some degree of error can be tolerated [16, 19]. Examples of lossless compression techniques include lossy predictive coding, wavelet coding, and transform coding.

3.1 Run Length Encoding (RLE)

In Run Length Encoding (RLE), the target data constitutes repetitive characters [18]. As the same characters are received at least four times, the RLE algorithm compresses the information to form a three-character series [20]. Hence, the role of this algorithm is to utilize data characters in sequences by ensuring that the information is encoded with strings containing the number of total character looping that takes place, eventually following with repeating characters [21]. Based on this procedure, it is evident that the RLE algorithm gains application to files perceived to exhibit homogeneous characters. Major steps of the RLE algorithm as a data compression mechanism are highlighted below:

- Data sizes with repeating symbols are reduced

- "run" is used to designate repeating symbols
- Encoding 1 "run" to obtain 2-byte data
- One byte is the "run value" while another byte reflects the "run count"
- Data compression of text information associated with symbol repetitions

3.2 Huffman Coding

Emerging as one of the oldest techniques through which data compression is achieved, this algorithm ensures that the data is intact and that its storage abides by the original set. Indeed, this algorithm's working principle requires the encoding of the respective characters to obtain bit representations [22]. However, the manner in which the respective characters are represented varies in relation to the characters' frequency of appearance. In situations, where characters tend to appear more often, this algorithm yields shorter lengths of the characters' bit representation [23, 24]. Conversely, situations, where characters appear less frequently, the algorithm yields the characters' longer bit representations [24, 25]. From the majority of the previous scholarly investigations, the use of Huffman coding as a data compression algorithm tends to achieve up to 30 percent memory saving [19-22].

3.3 Arithmetic Coding

From the trend concerning data compression algorithms, the emerging theme is that their role is to select some of the methods through which at least one of the same elements can be replaced via the use of a certain code. Arithmetic Coding as a data compression algorithm operates in such a way that input symbol series are replaced with numbers via arithmetic processes, with the input symbols existing in certain data files [26]. The implication is that in situations, where the encoded messages are more complex and longer, more bits are required to achieve the data compression and decompression processes [25-27].

When this algorithm is employed towards data compression, the output exists in the form of numbers that are greater than or equal to 0 but smaller than 1. It is also worth noting that the numbers tend to be decompressed uniquely to achieve symbol sequences that might have been used towards generating the selected number [27, 28]. Indeed the algorithm operates in such a way that the generation of output numbers requires the respective symbols that need to be compressed to be assigned sets of probability values [28]. Imperatively, the capability of the decoders and encoders are worth considering while implementing Arithmetic Coding because the majority exhibit limited numbers of mantissa [29, 30], which are associated with errors [11]. During the use of Arithmetic Coding towards data compression, the steps involved include:

- Setting current interval (0,1)
- Ensuring that the current interval is divided into different sub-intervals relative to symbol probability; with the respective sub-intervals representing their associated symbols
- Ensuring that the message's "i" symbol is read to determine sub-intervals associated with the selected symbol
- Setting the current interval to be equal to the sub-interval symbol to "i"
- Ensuring that the second, third, and fourth steps are repeated to the extent that the message's last symbol is reached

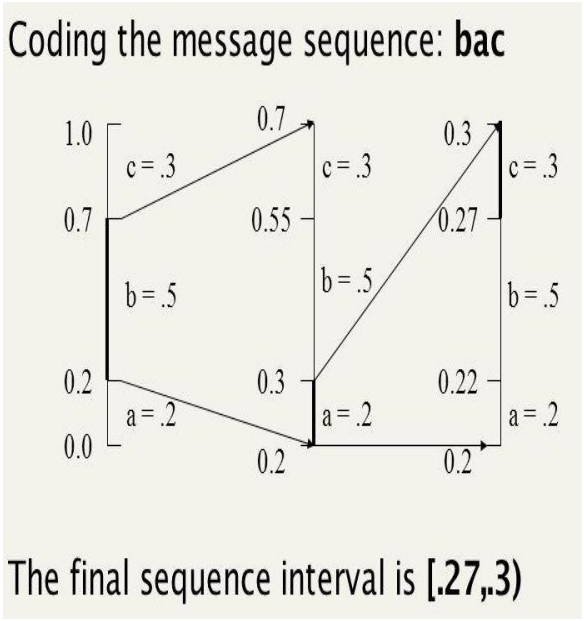


Figure 5: Sample Arithmetic Coding in practice
Source: Zhang, Xu and Zhou (2017)

3.4 Punctured Elias Code

When this algorithm is employed towards data compression, the initial step involves establishing binary numbers from "n." In turn, the bits are reversed before preparing flags through which the number of bits holding values of 1 in "n" can be shown [28]. The next step requires that for the respective 1 bit in the selected "n," flags from 1 are readied and, eventually, end with flags having 0 [29, 30]. With the reversed binary numbers combined with the flags, the algorithm's purpose of data compression is completed.

3.5 Lossless Compression Techniques (Lossy Predictive Coding, Wavelet Coding, and Transform Coding)

In transform coding, the main aim is to ensure that the data is converted to a form that makes it easier for compression [3]. This achievement ensures that the reduction of irrelevant data is facilitated [18] and that the correlation between pixels is also reduced [19].

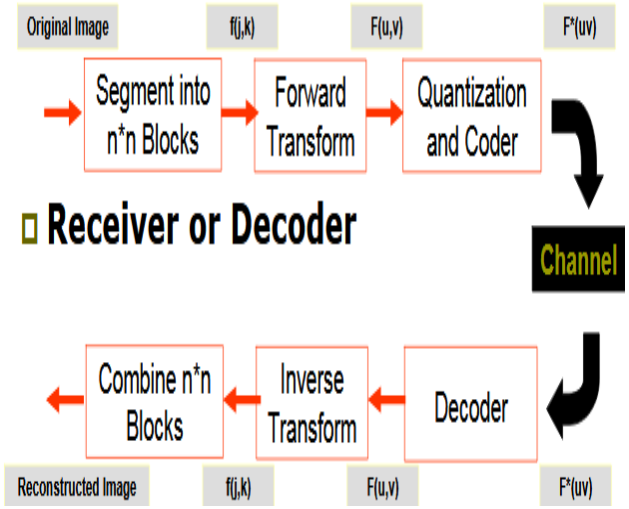


Figure 6: Transform coding
Source: Albano, Bruno and Carpentieri et al. (2014)

Regarding lossy predictive coding, this technique offers a data representation in which source symbols are represented via code words [11]. These symbols tend to deviate from the neighboring pixels' values or the predicted values and the role of the technique involves efficient reduction of interpreted redundancies [23, 27]. An example of a specific context where lossy predictive coding has proved feasible involves images exhibiting high-degree inter-pixel redundancies, especially when noise is present [2].

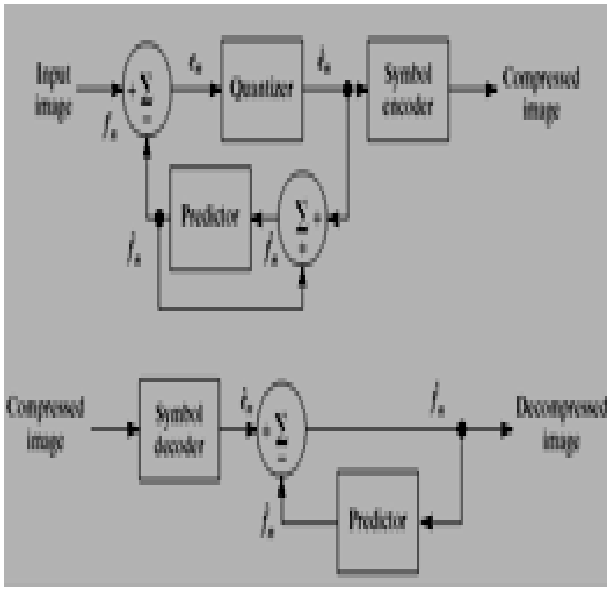


Figure 7: Lossy predictive coding
Source: Zhang, Xu and Zhou (2017)

In wavelet coding, a central assumption is that through linear coding (wavelet transform in this case), the resultant transform coefficients can be stored more efficiently compared to their associated pixels [10]. Given the computational efficiency of the wavelets, the original stage is not necessarily subdivided [3]. Compared to other approaches, wavelet coding has been deemed superior due to lower mean square errors [21].

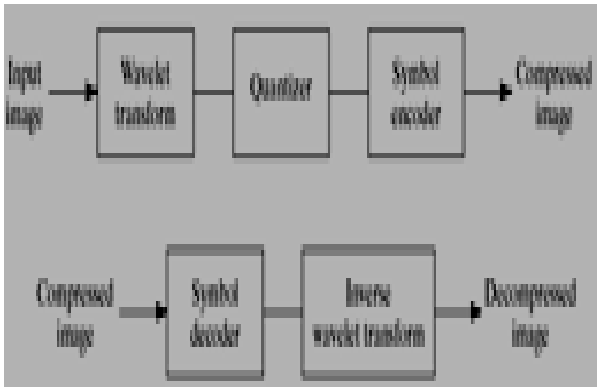


Figure 8: Wavelet coding
Source: Singh and Manimegalai (2012)

IV. INTEGRATION OF ENCRYPTION AND COMPRESSION

Integrating data encryption and compression forms another approach through which information security is increased. With huge amounts of data dominating the current and progressing world of computers, integration has been informed by the criticality of storing and transmitting huge data amounts securely [13]. Through the integration of data encryption and compression, most of the previous investigations hold that redundant character strings are removed in the given files to assure uniform character distribution [3-6]. The integration ensures that the shorter cipher text and plaintext save the time required towards file encryption, decryption, and transmission [8]. The integration also ensures that the plaintext's reduced redundancy hinders potential cryptanalytic attacks [11-14].

It is also worth noting that the integration of data encryption and compression processes has emerged at a time when there is a growing need for simultaneous securing of data while reducing the transmission time and memory space [8, 10], hence efficiency [16-18]. Notably, the role of data compression as a procedure encountered in the integration of data cryptography and compression lies in its capacity to remove redundant character strings [1]. Some of the specific issues to which the integrated approach has responded include the rapid progress felt in computing processes, the emergence of large data sizes (including multimedia information), and the vulnerability of Internet-based text data transmission [11, 14].

In the integrated encryption-compression technique, one of the most common combinations entails the use of lossless compression and symmetric cryptographic method [9].

Notably, this approach has gained wide application due to its emphasis on the security of the target images and texts, rather than the reduction of data size [11-13]. Particularly, integration via the use of lossless compression strives to ensure that the resultant data can be reversed to obtain the original information without compromising the compression ratio [2], as well as the reconstructed images and texts [7]. Hence, situations that emphasize data accuracy attract the integration approach that employs lossless compression, which is combined with the symmetric cryptographic method [22]. Some of these situations include legal data, biomedical image, and textual information [10].

V. The Need for Integrated Encryption and Compression Algorithms in Information Security

The need for integrated encryption and compression algorithms has been informed by the growing need for the prevention of data interception [2, 3]. Also, the need for these algorithms has been motivated by efforts to ensure that even in situations where data interception occurs, the information remains unreadable to intruders or interceptors [4]. With growing internet connections, it remains notable that data privacy remains a priority. Therefore, integrated data encryption and compression algorithms have evolved to ensure that the total communication over a given server is not accessed by unauthorized parties [7]. Other technological applications that have prompted the evolution and growing demand for integrated data encryption and compression algorithms include electronic mail transfer, Wi-Fi, SMSs, external hard drives, USB peripheral devices, laptops, and desktop PCs [9, 11]. Overall, these algorithms have been embraced due to their capacity to assure privacy in personal communications. Apart from offering a means through which information is secured, the algorithms also foster message authentication [14]. It is also worth indicating that the integrated encryption and compression algorithms are needed due to their capacity to save transmission time, CPU-time, and storage, eventually curbing possible redundancy.

The evolution of the hybrid approach combining the compression and encryption technique has been associated with the capacity to offer real data security while ensuring that the computational complexity remains low [23]. Due to this capacity, most of the previous studies avow that the hybrid technique increases the security and efficiency of information or data transmission [7-10]. Hence, the integration concept regarding data compression and encryption algorithms promises improvements in data security and transmission efficiency through their capacity to steer improvements in the performance of the respective cryptographic and compression techniques [13-17], positive outcomes that account for the recent increase in the popularity and incorporation of this hybrid concept into the field of information security [21-24].

VI. How the Integration of Encryption and Compression Helps For Security

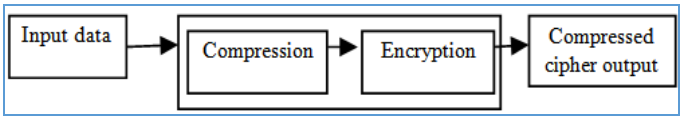
Indeed, the integration of data encryption and compression processes increases information security. The assurance is achieved in such a way that even after a malicious person deciphers some encrypted data, the reconstruction of the original data remains difficult for them [2], especially due to the absence of a transformation function [10]. Notably, the transformation function for the case of the integrated approach is kept as a secret key, having been encrypted and kept [13]. It is also worth indicating that the hybrid concept improves information security by combining excellent properties with which lossless and lossy compression techniques are associated [3-6] while ensuring that the downside associated with asymmetric and symmetric cryptographic techniques is offset [14-19], especially in relation to cipher key management [10].

In a hybrid approach where encryption is implemented before compression, information system security improves by providing room for the transmission of redundant data over bandwidth-constrained and insecure channels [6], having hidden the data from potential intruders [18]. In other hybrid situations where the data is compressed before being encrypted, this integrated approach has also been observed to steer improvements in information security. Particularly, the hybrid approach aids in decreasing the effectiveness of the majority of attacks (via the initial removal of redundant data) [15]. Additionally, the hybrid technique improves information security by making brute force attacks (which involve the use of different possible combinations of keys relative to the input data) to take longer [21-25]. Additional scholarly affirmations hold that the integrated approach's capacity to make brute force attacks take longer is achieved by prompting the attackers to decrypt the data prior to its decompression, steps that come before examining the feasibility of the output data [7]. Given that this process takes longer (courtesy of the integrated approach), the hybrid technique implies that the attack is unlikely to deduce the input [17-19]. Additional scholarly affirmations hold that the integrated approach improves information security by ensuring that the opponents are exposed to less data [8] and that the lesser the data or cipher text to which they are exposed (for analysis) the fewer the hints they tend to gain – relative to a sender's key and its associated cipher's internal state [24-27].

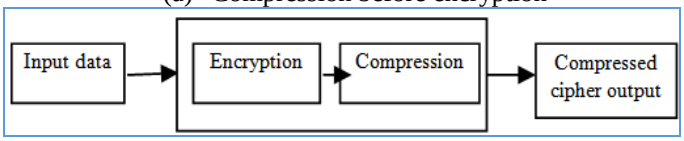
VII. How to Integrate Encryption and Compression Algorithms

When joint encryption and compression algorithms are employed, the two processes are combined to establish a single step. Indeed, two approaches are employed relative to the integrated encryption and compression processes. One of

the approaches involves encryption after compression [21]. Another approach entails encryption before compression [23, 24]. In the former approach, encrypting after data compression yields a two-fold merit. Particularly, the approach is associated with significant reductions in time and data size [26, 27]. Relative to the latter approach, the drawback is that it is time-consuming [14-16]. Despite the mixed outcomes regarding the application of the two approaches of integrated data encryption and compression, many studies avow that the joint algorithms consume less time when compared to situations where the encryption and compression algorithms are applied independently [7]. Also, it remains notable that the joint encryption and compression algorithms assure two levels of security [10-13].



(a) Compression before encryption



(b) Compression after encryption

Source: Singh and Manimegalai (2012)

Figure 9: Joint encryption and compression process

Based on the description of the hybrid approach above, it is evident that the stage involving data encryption ensures that data is transformed from the plaintext format to a cipher text format by using encryption keys [7]. On the other hand, the compression stage ensures that the source data existing in a given format is transformed into a codeword, which is smaller-sized [10]. Given that the current encryption and compression methods are constrained in terms of speed or a computer's required processing time, the two processes are integrated or combined to ensure that a pseudo random shuffle is introduced into the data compression procedure [4]. It is also worth indicating that some joint encryption and compression algorithms have been documented and affirmed to be faster due to their capacity to employ selective encryption techniques. Some of these algorithms include Real-time Video Encryption Algorithm (RVEA), Video Encryption Algorithm (VEA), and Secure Motion Picture Experts Group (SECMPEG) [31].

VIII. CONCLUSION

Due to the growing demand for responsive information security systems, data encryption and compression techniques have evolved and targeted systems or platforms such as websites, personal databases, and computers. Data encryption refers to a process through which information is encoded to ensure that only authorized users can read it. Regarding data compression, the process involves reducing the number of bytes or bits required to represent certain sets of data. Whereas encryption techniques fall into two broad categories involving symmetric and asymmetric keys, data compression can be lossy or lossless. Some of the specific algorithms that have been employed in data encryption include River-Shamir-Adleman (RSA) algorithm (in asymmetric encryption) and include Rivest Cipher (RC6) and Data Encryption Algorithm (DES) (in symmetric encryption). On the other hand, some of the algorithms that have been used towards successful lossless data compression include lossless predictive coding, Arithmetic coding, Lempel-Ziv-Welch (LZW) compression, Huffman coding, and Run Length Encoding. These algorithms have been employed in situations involving platforms such as word processing files, tabular numbers, and executable codes. For lossy data compression, some of the algorithms that have gained increasing application include lossy predictive coding, wavelet coding, and transform coding. In summary, an increasingly sophisticated threat environment continues to combine with corporate networks, which have also grown in complexity. This paper has demonstrated that when data encryption and data compression approaches are integrated, there is likely to be an enhanced state of information security. This enhancement, which is achieved by the joint encryption and compression algorithms, promotes efficient data storage and transmission. However, a problem that emerges involves determining whether compression should be conducted before encryption or vice versa. In the future, it remains inferable that successful data encryption and compression will be shaped by the degree of responsiveness of the algorithms that will be employed in different information systems.

REFERENCES

[1] Jay ram P, Ranganatha HR, Anupama HS. Information Hiding Using Audiostegnography: A Survey. *The International Journal of Multimedia & Its Applications (IJMA*) Vol.3, No.3, August 2011

[2] M. Baritha Begum and Y. Venkataramani. A New Compression Scheme for Secure Transmission, *International Journal of Automation and Computing*, 10(6), December 2013, 578-586, DOI: 10.1007/s11633-013-0756-3

[3] Ajit Singh and Rimple Gilhotra -Data Security Using Private Key Encryption System Based On Arithmetic Coding. *The International Journal of Network Security & Its Applications (IJNSA)*, Vol.3, No.3, May 2011

[4] Bobby Jasujaand and AbhishekPandya –Crypto. Compression System: An Integrated Approach using Stream Cipher Cryptography and Entropy Encoding. *International Journal of Computer Applications* (0975 –8887) Volume 116, No. 21, April 2015

[5] Kumari S. "A Research Paper on Cryptography Encryption and Compression Techniques," *International Journal of Engineering and Computer Science* 2017; 6(4), pp. 20915-20919

[6] Emami SS. "Security Analysis of Cryptographic Algorithms," Ph.D. thesis, Macquarie University, Sydney, New South Wales; 2013

[7] Kandola S. "A Survey of Cryptographic Algorithms," Master thesis. St. Lawrence University, Saint Lawrence County, NY. 2013
[8] Mathur H and Alam Z. "Analysis in Symmetric And Asymmetric Cryptology Algorithm." *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)* 2015; 4(1), pp. 44-46
[9] Chandra S and Bhattacharyya S. A Study and Analysis on Symmetric Cryptography," *International Conference on Science Engineering and Management Research (ICSEMR)*, Chennai, India, 2014
[10] Hercigonja Z. "Comparative Analysis of Cryptographic Algorithms." *International Journal of Digital Technology & Economy* 2016; 1(2), pp. 127-134.
[11] Soni S, Agrawal H and Sharma M. "Analysis and Comparison between AES and DES Cryptographic Algorithm." International Journal of Engineering and Innovative Technology (IJEIT) 2012; 2(6), pp. 362-365
[12] Mahajan P and Sachdeva A. "A Study of Encryption Algorithms AES, DES and RSA for Security." *Global Journal of Computer Science and Technology Network, Web & Security* 2013; 13(15)
[13] Borda M. "Fundamentals In Information Theory and Coding." Springer-Verlag Berlin Heidelberg, Romania, 2013; pp. 95
[14] Shanmugasundaram S and Lourdusamy R. "A Comparative Study of Text Compression Algorithms." *International Journal of Wisdom Based Computing* 2011; 1(3), pp. 68-76
[15] Altarawneh H and Altarawneh M. "Data Compression Techniques on Text Files: A Comparison Study." *International Journal of Computer Applications* 2011; 26(5)
[16] Antil R and Gupta S. "Analysis and Comparison of Various Lossless Compression Techniques." *International Journal for Research in Applied Science and Engineering Technology (IJRASET)* 2014; 2(3), pp. 251-262
[17] Bodden E, Clasen M and Kneis J. "Arithmetic Coding Revealed." RWTH Aachen University, Aachen, Germany. 2004
[18] R. Pizzolante and B. Carpentieri, "Lossless, low-complexity, compression of three-dimensional volumetric medical images via linear prediction." In *Proceedings of the 18th International Conference on Digital Signal Processing*, DSP '13, 2013
[19] Fenwick P. "Variable-Length Integer Codes Based on the Goldbach Conjecture, and Other Additive Codes." *IEEE Transactions on Information Theory* 2002; 48(8), pp. 2412-2417
[20] Waghulde R, Gurjar H, Dholakia V and Bhole GP. "New Data Compression Algorithm and its Comparative Study with Existing Techniques." *International Journal of Computer Applications* 2014; 102(7), pp. 35-38
[21] Klinc D. et al. "On Compression of Data Encrypted With Block Ciphers." *IEEE Transactions on Information Theory* 2012; 58(11), pp. 6989-7001
[22] Schonberg D, Draper SC and Ramchandran K. "On Blind Compression of Encrypted Correlated Data Approaching The Source Entropy Rate." *Proceedings of 43rd Annual Allerton Conference on Communication, Control and Computing, Allerton, IL* 2005; pp. 1538-1547
[23] C. P. Wu and C. C. J. Kuo, "Design of integrated multimedia compression and encryption systems." *IEEE Transactions on Multimedia* 2005; vol. 7, no. 5, pp. 828–839
[24] Y. Zhang, B. Xu, and N. Zhou, "A novel image compression–encryption hybrid algorithm based on the analysis sparse representation." *Optics Communications* 2017; vol. 392, pp. 223-233
[25] E. Setyaningsih and R. Wardoyo, "Review of image compression and encryption techniques." *International Journal of Advanced Computer Science and Applications* 2017; vol. 8, No. 2
[26] T. Sharma and S. Bollavarapu, "Data Security using Compression and Cryptography Techniques." *International Journal of Computer Applications* 2015; vol. 117, no. 14, pp. 15-18
[27] P. Albano, A. Bruno, B. Carpentieri et al., "Secure and distributed video surveillance via portable devices." *Journal of Ambient Intelligence and Humanized Computing* 2014; vol. 5, no. 2, pp. 205-213
[28] R. Pizzolante, B. Carpentieri, A. Castiglione, and G. De Maio, "The AVQ algorithm: Watermarking and compression performances." In *Proceedings of the 3rd IEEE International Conference on Intelligent Networking and Collaborative Systems, INCoS '11*; pp. 698-702. 2011
[29] U. Fiore, "Selective redundancy removal: a framework for data hiding." *Future Internet* 2010; vol. 2, no. 1, pp. 30–40, 2010
[30] M. R. Ogiela and L. Ogiela, "On using cognitive models in cryptography." In *Proceedings of the 30th IEEE International Conference on Advanced Information Networking and Applications, AINA '16*, pp. 1055–1058, IEEE, Crans-Montana, Switzerland, March 2016
[31] Singh KJ & Manimegalai R. A Survey of Joint Compression and Encryption Techniques for Video Data. *Journal of Computer Science* 2012; 8(5), 731-736

Authors Profile

Mr. Gowtham Mamidisetti is a research scholar of computer science and engineering department of Acharya Nagarjuna University, Guntur. His research area includes cryptography, information security, and cloud security. He has publications in various reputed journals and presented papers in international conferences. He did his masters degree in computer science from University campus, JNTU Kakinada.

Dr. Ramesh Makala is an associate professor in the Department of Information Technology at RVR & JC College of Engineering, Guntur, Andhra Pradesh, India, where he has been since 2003. He received his Ph.D. in Computer Science from Acharya Nagarjuna University in 2013 and M.Tech. from Andhra University in 2004. His research interests span both Image Processing and Information Security. Much of his work has been on security and data compression. He has published different novel works in IEEE Conference Proceedings(IEEE Xplore) and Scopus Indexed journals. He is in charge of the Research Center in the department. He also guides students who are working on Machine Learning and Bigdata Analytics problems.

Mr. Ravi Teja K is an assistant professor in Department of Computer Science Engineering,Shri vishnu engineering college for women, Bhimavaram. His research are includes Cryptography and network security.He got his Masters degree from JNTU Hyderabad.

