International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 23 (2017) pp. 13589-13595
© Research India Publications. http://www.ripublication.com

A Survey on Data Compression and Cryptographic Algorithms

Kenang Eko Prasetyo[1], Tito Waluyo Purboyo[2] and Randy Erfa Saputra[3]

Faculty of Electrical Engineering, Telkom University, Bandung, Indonesia.

[1]Orcid: 0000-0002-8523-5882, [2]Orcid: 0000-0001-9817-3185, [3]Orcid: 0000-0002-8537-2086

Abstract

Data security and confidentiality issues are of paramount importance both within an organization in the form of a commercial (enterprise), a college, a government agency, as well as an individual (personal). Especially if the data is in a computer network connected/connected to the public network or the internet. The ability to access and provide information quickly and accurately will greatly affect an organization. In this paper, we will try to discuss the message/data delivery security system by using a form of encryption that aims to maintain the confidentiality of the data or the confidentiality of a message. So that messages/data that we send cannot be accessed or read by people who are not eligible. Because a lot of security systems are used by an organization or also personal, then here will use one of its methods that are by using a method of the security system using the Cryptographic algorithm.

Keywords: Data Encryption and Decryption, Data Compression, Cryptography Concept.

INTRODUCTION

To secure data, compression is used because it uses less disk space, more data can be transferred via the internet. This increases the speed of data transfer from disk to memory. The security objectives for data security are Secret, Authentication, Integrity, and Non-rejection. Data security provides data protection throughout the company. Information security is a growing issue among IT organizations of all sizes. Data compression is known for reducing storage space and communications. It involves transforming data from a specific format, called a source message to a data with a smaller format called a code word. Data encryption is known to protect information from tapping. It converts data from a specific format, called plaintext, to another format, called ciphertext, using an encryption key. Currently the method of compression and encryption is done separately. Cryptography before the modern era is effectively synonymous with encryption, the conversion of information from an easily readable state to nonsense. Modern cryptography is based largely on the mathematical theory and practice of computer science; The cryptographic algorithm is designed around the assumption of computational violence, so such an algorithm is difficult to solve in practice by any enemy. In theory it is possible to break such a system, but it is impossible to do so in a known practical way

CRYPTOGRAPHY

Cryptography is the art and science of securing messages. In the world of cryptography, messages are called plaintext or cleartext. The process of disguising the message in such a way as to hide its original content is called encryption. The encrypted message is called ciphertext. The process of returning a ciphertext to plaintext is called decryption. To implement the objective is designed a security system that serves to protect the information system. One of the efforts of information system security that can be done is cryptography. Cryptography is actually a study of mathematical techniques related to aspects security of an information system.

Cryptography can be defined as a science that studies mathematical techniques related to aspects of information security such as confidentiality, data integrity, sender / data receiver authentication, and data authentication. Cryptanalysis is the study of how to defeat (solve) Cryptographic mechanisms, and Cryptology derived from the word Kryptos and Logos (Greek) meaning hidden word, is the incorporation of the discipline of cryptography and cryptanalysis. (Jusuf Ir, 2002)

In Cryptography we will often find various terms or terminology. Some important terms to know:

- Plaintext (M) is the message to be sent (containing original data).
- Ciphertext (C) is an encrypted (encrypted) message that is the result of encryption.
- Encryption (function E) is the process of converting

plaintext into ciphertext.

- Decryption (function D) is the opposite of encryption ie change ciphertext becomes plaintext, so it is initial / original data.
- The key is an undisclosed number used in the encryption and decryption process.

According to [2], there are several demands related to data security issues

A. Confidentiality

Confidentiality is a service used to safeguard information from any unauthorized party to access it. This information will only be accessible to eligible parties only.

B. Authentication

Authentication is a service related to the identification of parties that want to access the information system (entity authentication) as well as the authenticity of the data from the information system itself (data origin authentication). At the time of sending or receiving information both parties need to know that the sender of the message is the actual person as claimed.

C. Data Integrity

Data integrity is a service that aims to prevent the occurrence of alteration of information by unauthorized parties. To ensure the integrity of this data must be ensured that the information system is able to detect the occurrence of data manipulation. Data manipulation herein includes the insertion, deletion, or replacement of data. This demand relates to the guarantee that every message sent must be received by the recipient without any portion of the message being replaced, duplicated, defaced, altered, and added.

D. Non-Repudiation

Non-repudiation prevents both senders and receivers from denying that they have sent or received a message / information. If a message is sent, the recipient can prove that the message was indeed sent by the sender listed. Conversely, if a message is received, the sender can prove that the message has been received by the intended party.

CRYPTOGRAPHY ALGORITHM

In this section, we describe two main types of cryptographic algorithms; Symmetric Key Cryptography and Asymmetric Key Cryptography [4] and also some examples in each type.

A. Symmetric Key Cryptography

Symmetric Cryptography is the Algorithm that it uses the key to the encryption process is the same as the key for the decryption process. Symmetric Cryptography Algorithm divided into 2 categories: Flow Algorithm (Stream Ciphers) and Block Algorithms (Block Ciphers).

The advantage of this Symmetric Cryptography Algorithm is the process time for encryption and decryption that is relatively fast. This is due to the efficiency that occurs in the key generator. Because the process is relatively fast then this algorithm is suitable for use in digital communication system in real time like GSM. But to send messages with different users it takes a different key than the previous one. so the number of buttons will be directly proportional to the number of users.

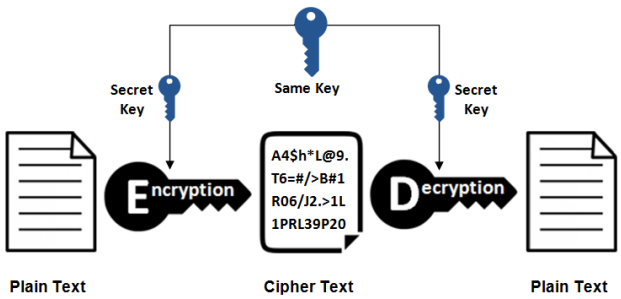

Figure 1: Symmetric Key Cryptography

The other advantages of using symmetric key algorithm is higher operating speed when compared with asymmetric key algorithm although directly proportional with the addition of file size i.e. speed of encryption and decryption process depends on file size; the bigger the file size the more time required for encryption and decryption. But there is also the disadvantage of this symmetric key algorithm: for each delivery of messages with different user required different keys, so there will be difficulties in key management [6].

Some algorithms using symmetric key algorithm:

1. DES (Data Encryption Algorithm) is a popular block cipher algorithm because it serves as a standard symmetry-encryption key algorithm, although it has

now been replaced by the new AES algorithm because DES has unsafe. DES belongs to a cryptographic system of symmetry and belongs to a block cipher type. DES operates on a 64-bit block size. DES describes 64 bits of plaintext to 64 bits of ciphertext by using 56 bits of internal key (internal key) or sub-key. The internal key is generated from an external key 64-bit length [7].

2. RC6 is a cryptographic algorithm that is included in a symmetric key. This algorithm is the development of the RC5 algorithm included in Advance Encrypt Standard (AES). RC6 uses the core of the data-dependent rotation. RC6 is an algorithm that uses block sizes up to 128 bits, with key sizes used varying between 128, 192 and 256 bits. RC6 divides the 128-bit block into four 32-bit blocks, and follows the six basic operating rules: number, subtraction, exclusive OR, integer multiplication, sliding bits to the right and bending bits to the left. The main process in this algorithm is the key scheduling and decryption of encryption.

B. Asymmetric Key Criptography

In the mid-70s Whitfield Diffie and Martin Hellman invented the Asymmetric Algorithm. This encryption technique that revolutionized the world of cryptography. An Asymmetric key is a pair of cryptographic keys used for the encryption process and the other for decryption. Anyone who gets a public key can use it to encrypt messages, while only one person has a certain secret in this case a private key to disassemble the password sent to him Figure 2.

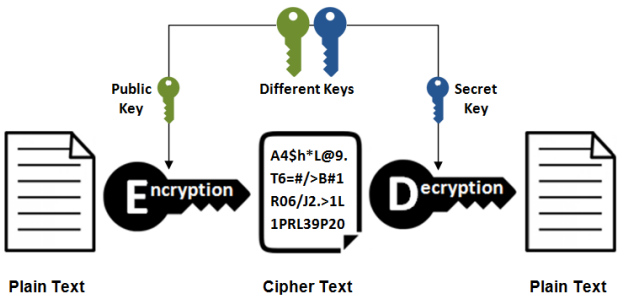

Figure 2: Asymmetric Key Cryptography

This Asymmetric Encryption technique has a weakness, ie the encryption process is much slower than the symmetric key. Therefore, usually the message is not encoded with an asymmetric key, but only symmetric keys are encrypted with an asymmetric key. When a message is sent after it is encoded with a symmetric key.

An example of algorithm using asymmetric key algorithm is Rivest-Shamir-Adleman (RSA) algorithm. RSA is the first algorithm suitable for digital signatures and encryption, and one of the most advanced in the field of public key cryptography since first described in 1977. The RSA algorithm involves some steps, such as key generation, encryption, and decryption [8].

DATA COMPRESSION

In this section, we describe specific examples types of Data Compression Algorithm**s:** Run Length Encoding (RLE), Huffman Coding, Arithmetic Coding, Punctured Elias Code, Goldbach Code.

A. Run Length Encoding (RLE)

Compression and decompression using Run length encoding is a method used to compress data containing repetitive characters. When the same characters are received consecutively four or more times (more than three), this algorithm compresses the data in a series of three characters. Run-Length Encoding is an algorithm that exploits repetitive characters in sequence in a data by encoding it with a string that consists of the sum of the number of character looping that occurs, followed by a repeating character.

Generally, RLE algorithm is used on files that have characters that tend to be homogeneous. Therefore, if the algorithm is used universally then it is necessary to do the grouping or transformation of similar characters / symbols. The steps of compressing data using Run Length Encoding are:

1. Techniques to reduce the size of data containing Repeating symbols.
2. Repeating symbols are designated: *run*
3. 1 *run* is usually encoded into a 2-byte data.
4. The first byte represents the number of symbols, called *run count*.
5. The second byte represents a repeating symbol, called *run value*.
6. RLE is suitable for compressing text data that contains many symbol repetitions

For example the data AAAAAABBBXXXXXT has the value of 15 Byte, With RLE, it can be encoded into = 6A 3B 5X 1T (8 Byte).

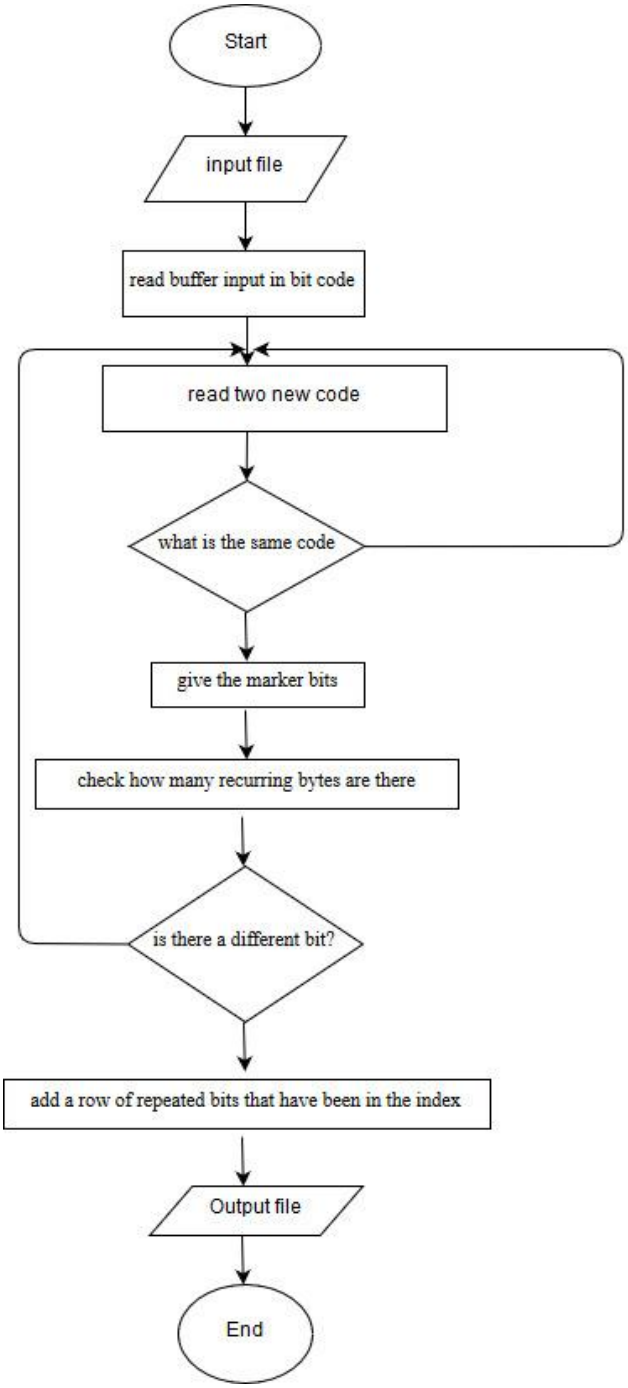

Figure 2: Flowchart of Run Length Encoding

B. Huffman Coding

Huffman Algorithm is one of the oldest compression algorithms compiled by David Huffman in 1952. The algorithm is used to create compression of the type of loss compression, ie data compression where not one byte is lost so that the data is intact and stored according to the original. The working principle of the Huffman algorithm is to encode each character into a bit representation. The representation of bits for each character differs from one another based on the frequency of appearance of a character. The more often the character appears, the shorter the length of its bit representation. Conversely, when the frequency of characters becomes less frequent, the longer the bit representation for that character. Huffman's algorithm compression technique can provide memory usage savings of up to 30%. The Huffman algorithm has a complexity of 0 (n log n) for the set with n characters.

C. Arithmetic Coding

In general, data compression algorithms are based on the selection of ways of replacing one or more of the same elements with a particular code. In contrast, Arithmetic Coding replaces a series of input symbols in a data file with a number using an arithmetic process. The longer and more complex the encoded message, the more bits it takes to compress and decompress the data.

The output of this Arithmetic coding is a number smaller than 1 and greater or equal to 0. This number can be uniquely decompressed to produce a sequence of symbols used to generate that number. To generate the output number, each symbol that will be compressed is assigned a set of probability values. The implementation of Arithmetic Coding should closely regard the capabilities of the encoder and decoder, which generally have a limited number of mantissa. This can cause errors if an Arithmetic Coding has code with a very long floating point.

The steps to compressing data using Arithmetic Encoding algorithm are:

1. Set Current Interval [0,1)
2. Divide the current interval into several sub intervals based on symbol probability; one sub interval represents one symbol.
3. Read the “i” symbol on the message, identify the sub interval belonging to that symbol.
4. Set Current Interval = sub interval symbol to “i”.
5. Repeat steps 2-4 until the last symbol on the message.

D. Punctured Elias Code

Punctured Elias Codes is a method designed by Peter Fenwick in an experiment to improve the performance of the Burrows-Wheeler transform. The term “Punctured” comes from Error Code Memory (ECC). ECC consists of the original data in addition to an array of numbers from check bits. Some check bits are omitted to shorten a series of codes

The steps to compress data using Punctured Elias Code algorithm are:

1. Take a binary number from “n”.
2. Reverse the bit and prepare a flag to show the number of bit that holds the value of 1 in “n”.
3. For every 1 bit in “n”, we ready a flag from 1 and ending with a flag with 0.
4. Combine the flag with the reversed binary number

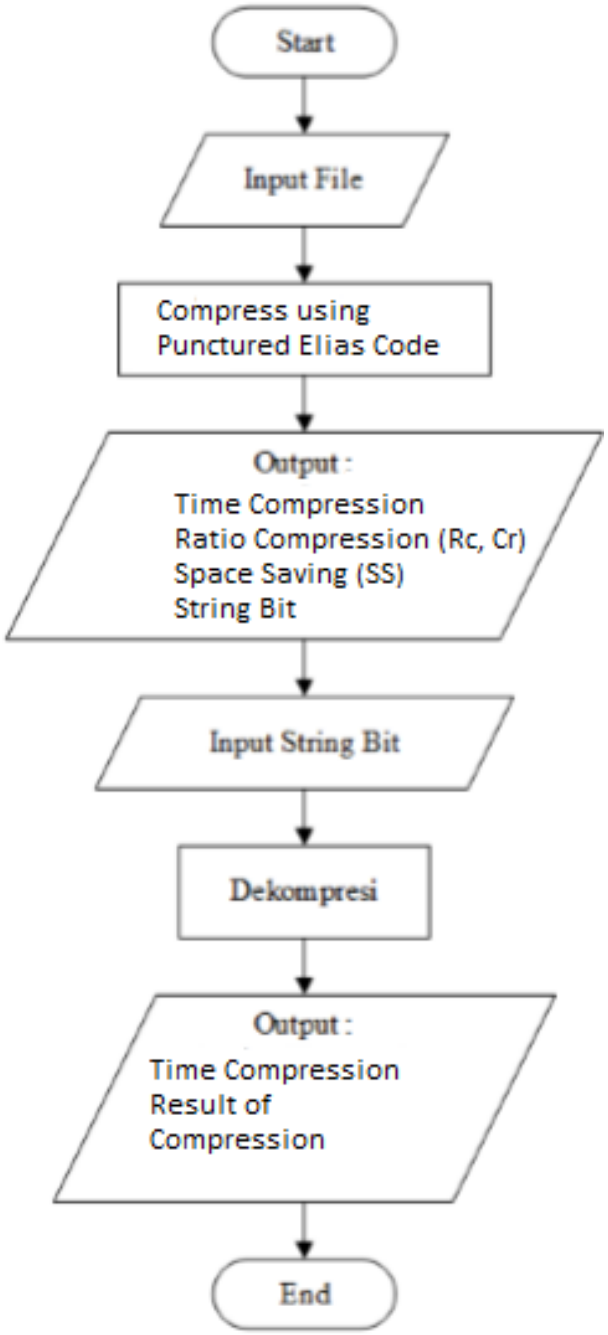

Figure 3: Flowchart of Punctured Elias Code Algorithm

E. Goldbach Code

The Goldbach Codes algorithm is an algorithm that is assumed to use the Goldbach Conjecture theory of "All positive even numbers greater than 2 are the sum of two prime numbers.

Goldbach Codes has three codes, the first Goldbach Code is called "G0". G0 encodes the positive integer "n" by converting it into an even positive integer with 2 (n + 3) and then writing the sum of the prime numbers in reverse.

The second Goldbach Codes are called "G1". In principle, G1 is to determine two primes; Pi and Pj (where $i \leq j$), whose sums produce integers n and encode pairs (i, j-i + 1) with gamma codes. While the third G2 or Goldbach Codes is an extension of the G1 Code with cases like the following:

1. The integers 1 and 2 are encoded to 110 and 111 (other than integers 1 and 2 no other encoding starts with 11 …)
2. Even integers are encoded the same as in G1 Code but with an exception; If n = Pi + Pj is determined, it will be encoded by the pair (i +1,j – i + 1) replacing (i,j – i + 1). Thus, if i = 1, it will be encoded to 010, the gamma code of 2. This will guarantee that even integers of G2 Code will not start from 1 and will always have the form 0 ...: 0 ...
3. If n is a prime Pi number, it will be encoded as a gamma code of (i+1)) followed by 1 single to produce 0 …:1.
4. If n is an odd number, then G2 code starts from a single 1, followed by a G2 Code from an even number n-1. Generating a gamma code with the form of 1:0…:0

RECENT RESEARCH

In Table 1, we summarize the recent research of data compression and cryptographic algorithm.

Table 1. Recent research comparison

Author	Method	Conclusion
Waghulde, R. et al. [17]	LZ77, LZW, 7ZIP, Huffman Coding	The proposed algorithm are best achieved on XML file format (.xml) with an average of 91% compression ratio.
Klinc, D. et al. [18]	Cipher Block Chaining (CBC), Electronic Code Book (ECB), Public-key encryption	The results, while still far from theoretical limits, indicate that considerable compression gains are practically attainable with block ciphers, and improved performance can be expected with future increase of block sizes.
Schonberg, D. et al. [19]	Low-Density Parity Check (LDPC)	The scheme presented is proven to achieve redundancy proportional to the inverse of the square root of the block length, and requires minimal feedback.

CONCLUSION

In this paper, we have discussed about data compression and cryptographic algorithm. Both of these algorithms have their respective objectives; the purpose of data compression is to reduce file size, while the purpose of cryptography is to secure a file in order to avoid data leakage.

.

REFERENCES

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

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

[3] Kandola, S., 2013, "A Survey of Cryptographic Algorithms," Master thesis. St. Lawrence University, Saint Lawrence County, NY.

[4] Mathur, H., and Alam, Prof. Z., 2015, "Analysis in Symmetric And Asymmetric Cryptology Algorithm," International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), 4(1), pp. 44-46.

[5] Chandra, S., and Bhattacharyya, S., 2014, "A Study and Analysis on Symmetric Cryptography," International Conference on Science Engineering and Management Research (ICSEMR), Chennai, India.

[6] Hercigonja, Z., 2016, "Comparative Analysis of Cryptographic Algorithms," International Journal of Digital Technology & Economy, 1(2), pp. 127-134.

[7] Soni, S., Agrawal, H., and Sharma, M., 2012, "Analysis and Comparison between AES and DES Cryptographic

Algorithm," International Journal of Engineering and Innovative Technology (IJEIT), 2(6), pp. 362-365.

[8] Mahajan, P., and Sachdeva A., 2013, "A Study of Encryption Algorithms AES, DES and RSA for Security," Global Journal of Computer Science and Technology Network, Web & Security, 13(15).

[9] Borda, M., 2011, "Fundamentals In Information Theory and Coding," Springer-Verlag Berlin Heidelberg, Romania, pp. 95.

[10] Shanmugasundaram, S., and Lourdusamy, R., 2011, "A Comparative Study of Text Compression Algorithms," International Journal of Wisdom Based Computing, 1(3), pp. 68-76.

[11] Altarawneh, H., and Altarawneh, M., 2011, "Data Compression Techniques on Text Files: A Comparison Study," International Journal of Computer Applications, 26(5).

[12] Antil, R., and Gupta, S., 2014, "Analysis and Comparison of Various Lossless Compression Techniques," International Journal for Research in Applied Science and Engineering Technology (IJRASET), 2(3), pp. 251-262.

[13] Bodden, E., Clasen, M., and Kneis J., 2004, "Arithmetic Coding Revealed," RWTH Aachen University, Aachen, Germany.

[14] Howard, P. G., and Vitter J. S., 1992, "Analysis Of Arithmetic Coding For Data Compression," Brown University, Providence, RI.

[15] Fenwick, P., 1996, "Punctured Elias Code for Variable-Length Coding of the Integers", Report no. 137, The University of Auckland, Auckland.

[16] Fenwick, P. 2002, "Variable-Length Integer Codes Based on the Goldbach Conjecture, and Other Additive Codes," IEEE Transactions on Information Theory, 48(8), pp. 2412-2417.

[17] Waghulde, R., Gurjar, H., Dholakia, V., and Bhole, G. P., 2014, "New Data Compression Algorithm and its Comparative Study with Existing Techniques", International Journal of Computer Applications, 102(7), pp. 35-38.

[18] Klinc, D. et al., 2012, "On Compression of Data Encrypted With Block Ciphers", IEEE Transactions on Information Theory, 58(11), pp. 6989-7001.

[19] Schonberg, D., Draper, S. C., and Ramchandran, K., 2005, "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, pp. 1538-1547.