Blowfish algorithm and Huffman compression for data security application

IOP Conf. Series: Materials Science and Engineering **453** (2018) 012074 doi:10.1088/1757-899X/453/1/012074

Blowfish algorithm and Huffman compression for data security application

[1]Yaya Sudarya Triana, [2]Astari Retnowardhani

[1]Faculty of Computer Science, Universitas Mercu Buana, Jakarta, Indonesia
[2]Information Systems Management Department, BINUS Graduate Program - Master of Information Systems Management, Bina Nusantara University

E-mail: yaya.sudarya@mercubuana.ac.id

Abstract. In its development, information technology has become an important in people's everyday lives. Security and confidentiality of the data on computer networks today become a very important issue and continues to grow. Some of the cases relating to computer network security today become a job that requires handling fee and security has been tremendous. Vital systems, such as the defense system, the banking system, the hospital system, and other systems, requires such a high level of security. This is mainly due to the advancement of the field of computer networks with the concept of open system so that anyone, anywhere and at any time, have the opportunity to access these vital areas. To maintain the security and confidentiality of messages, data, or information in a computer network would require some encryption to create messages, data, or information that is not read or understood by any person, except for eligible recipients.

Keywords: Encryption, Decryption, Blowfish, Huffman

1. Introduction

The development of computer technology today has progressed very rapidly and has become a necessity. The higher the level of computer technology, the higher the threat level that threatens the security of computer users. One of the negative impacts of technological development is the theft of data. This crime is greatly feared by users of communication networks. With data theft, the data security aspects in data storage and information exchange are considered important. A remote data communication, not necessarily have a secure transmission line from wiretapping, as well as data storage is not necessarily safe from theft so that information security becomes an important part in the world of information itself.

To run the operational activities PT XYZ hospitals already use a good and integrated system, but the storage of administrative data is still stored in the shelves of bookkeeping. To keep doctors financial statements data is still manual. The use of files that are still manual or in the document for the management of financial statements data doctors be very important to be secured, because in the report there is information about rates and income doctors, if the file or document can be seen or changed contents by unauthenticated, then will have a negative impact on the person responsible for reporting, so that information can be stolen and will cause problems in the future.

Especially for the financial division that has many important documents such as doctors financial statements, the contents of the email if read by others who are not concerned will be able to harm the superiors and the finance. Therefore, a method that can maintain the confidentiality of PT XYZ

Hospitals information is required. The application of cryptography in this study is focused on securing stored document files, which can only be opened by parties with the authority to open them, not by irresponsible parties.

In general, there are two types of cryptography, namely classical cryptography and modern cryptography. Classical cryptography is character-based cryptography (encryption and decryption performed on every character). While modern cryptography is cryptography that operates in bit mode (expressed in 0 and 1). Therefore, the algorithm used is merging the two algorithms.

The symmetric cryptography algorithm to be used is the flow cipher Blowfish and Huffman compression methods to compress the file. The process begins with encrypting files with the two algorithms mentioned above by means of files that have been encrypted with one algorithm, then forwarded with another algorithm. This cryptographic technique is chosen because it is expected with this algorithm the process of data encryption-decryption can be done with a faster time.

2. Method

Cryptography is derived from the Greek word consisting of two words namely crypto and graphia. crypto means secret while graphia means writing. Cryptography was originally described as a science that learned how to hide messages. Cryptography is part of the mathematics that deals with information security such as secrecy, data integrity and authentication [3].

In general, cryptography is an information security technique that is done by processing the initial information or plaintext with a certain key using a certain encryption method, thus generating a new information or ciphertext that can not be read directly. The ciphertext can be returned to the original plaintext information through the decryption process.

Most of the history of cryptography is part of classical cryptography, which is a cryptographic method using paper and pencil or using simple mechanical aids. Classical cryptography is generally grouped into two categories, namely the transposition algorithm and substitution algorithm. Transposition algorithm is an algorithm that changes the order of letters in the message, while substitution algorithm is to replace each letter or group of letters with a letter or group of other letters.

Digital computer equipment produces modern cryptography. With digital computers, it would be possible to produce more complex and complicated ciphers. Classical cryptography generally uses characters per character (using a traditional alphabet), while modern cryptography operates on more complex biner cipher strings.

2.1. Cryptography Algorithm

Algorithm in cryptography is a set of rules (mathematical functions used) for encryption and decryption processes. In some cryptographic methods there is a difference between encryption function and decryption function. The underlying mathematical concept of the algorithm is the relationship between the set, ie the relation between the set containing the ciphertext elements. Encryption and decryption are functions that map the elements between the two sets.

Suppose the plaintext set is denoted P and the set of ciphertext element is denoted C, then the function E maps the set P to the set C.

$$\mathbf{E(P) = C}$$

And the decryption function maps the set C to the set P.

$$\mathbf{D(C) = P}$$

Since D decryption function returns set C to a set of P origin, the cryptographic algorithm must satisfy the equation.

$$\mathbf{D(E(P)) = P}$$

The security level of an algorithm in cryptography is often measured from the quantity of processes performed in a function, whether it is an encryption function or a decryption function. The process can also be linked to the required data sources, indicating the stronger the cryptographic algorithm.

In classic cryptography, cryptographic security lies in the confidentiality of its cryptographic algorithm. One example is an enigma machine issued by the German government during the second world war. However, this is a weak point when the algorithm leaks to the unauthorized party, thus requiring the preparation of a new algorithm without any concern about the leakage of such information, as the information can only be decrypted, ie those who have a private key.

Here are the terms that are often used in the cryptography field:

a. Plaintext is the message to send (contains original data).
b. Ciphertext is an encrypted (encrypted) message that is the result of encryption.
c. Encryption is the process of plaintext conversion into ciphertext.
d. Decryption is the opposite of encryption ie converting ciphertext into plaintext, so it is a preliminary or original data.
e. Keys are a secret number used in the encryption and decryption process.

In cryptography there are two main concepts namely encryption and decryption. Encryption is the process by which the information or data to be transmitted is converted to an almost unknown form as its initial information by using a specific algorithm. Decryption is the opposite of encryption that is to change the disguised form into preliminary information.

2.2. Symmetric Algorithm

The key used for the encryption and decryption process is the same key. In symmetrical key cryptography it can be assumed that both the receiver and the message sender have first shared the key before the message is sent. The security of this system lies in its confidentiality.

Generally included in this symmetric cryptography operates in block cipher, which is every time the encryption or decryption process is performed on a single data block (of a certain size), or operates in stream cipher mode, which is every time the encryption or decryption is performed on one bit or one byte of data.

The advantages of symmetric cryptography are The process of encryption or decryption symmetry cryptography takes a short time. Symmetrical key sizes are relatively shorter. Authentication of instant messaging is known from the accepted ciphertext, since keys are known only by the receiver and the sender only. Symmetrical cryptography deficiency is Symmetry locks should be sent over a secure communication channel, and both communicating entities should maintain key confidentiality.

2.3. Blowfish Algorithm

Blowfish (Open PGP Cipher 4) is an encryption that belongs to the class of symmetric crypto system, its encryption method is similar to Des Like Cipher (DES) created by a cryptanalyst named Bruce Schneier president of counterpane internet security firm Inc, a consulting firm on cryptography and computer security and published in 1994. Made for use on computers with large microprocessors 32 bits with very large data cache [8].

Blowfish is developed to meet the rapid design criteria in its implementation at optimum conditions of up to 26 clock cycles per byte, can run on less than 5 KB of memory, simple in algorithms so that it is easy to tell the error, and security variables where the key length varies (minimum 32 bits, maximum 448 bits, Multiple 8 bits, default 128 bits). Blowfish is optimized for a variety of applications where keys do not change frequently, such as on network communications or file encryption automatically. In application in computer microprocessor 32 bit with big data cache Pentium and power PC blowfish proved faster than DES. But blowfish does not match applications with frequent key changes or as one-way hast functions as in packet switching applications.

Blowfish is included in 64-bit encryption block cipher with key length which varies between 32 bit and 448 bits. Blowfish algorithm consists of two parts:

1) Key-Expansion
   Functioning to change the lock (Minimum 32 bits, Maximum 448 bits) into multiple subkey arrays (subkey) totalling 4168 bytes.

2) Data Encryption
It consists of simple Feistel Network iteration of 16 times. Each round consists of key dependent permissions, key substitutions, and dependent data. All operations are addition and XOR on variable 32 bit. Other additional operations are just four table lookup of indexed arrays for each round.

2.4. Permutation Box/Bleaching
Blowfish algorithm, used many subkey. These keys must be calculated or generated first before encryption or decryption of data. The purpose of this method is to randomize the sequence of bits in a block. This method differs from the substitution method of bit manipulation. The difference is that in this method a definite reference is used in the bit substitution. The reference does not have a special pattern, and in most cryptographic algorithms, the reference has been defined by the algorithm designer. It is called a permutation box, because it is a 2-dimensional box whose contents each have the bits of information must be moved to the order in the block. Blowfish utilizes permutation boxes in several processes. This permutation box aims to disrupt the sequence of bits, preventing cryptanalysts from attacking the algorithm using methods such as weak key methods. [5]

2.5. Encryption of Blowfish Algorithm
Blowfish uses a large subkey. The keys must be computed at the beginning, before computing the encryption and decrypting the data. The steps are as follows:

1) There is a permutation box (P-box) consisting of 18 pieces of 32 bits subkey: P1, P2, P3, ..., P18. This P-box has been set from the beginning, the first 4 P-boxes are as follows :
   P1 = 0x243f6a88
   P2 = 0x85a308d3
   P3 = 0x13198a2e
   P4 = 0x03707344
2) S-box forms of 4 pieces each worth 32 bits that have 256 inputs. Four 32-bit S-boxes each have 256 entries:
   S1,0,S1,1,.....,S1,255
   S2,0,S2,1,.....,S2,255
   S3,0,S3,1,.....,S3,255
   S4,0,S4,1,.....,S4,255.
3) The plaintext to be encrypted is assumed to be input, the Plaintext is taken as much as 64 bits, and if it is less than 64 bits we add the bit, so that the operation will be in accordance with the data.
4) The result is divided by 2, the first 32-bit is called XL, the second 32-bit is called XR.
5) Next do the operation XL = XL xor Pi and XR = F(XL) xor XR
6) The results of the above operations are converted XL to XR and XR to XL..
7) Do 16 times, repeat the 16th, do the XL and XR exchange process again.
8) In the 17th process do operations for XR = XR xor P17 and XL = XL xor P18.
9) The final process brings back XL and XR so that it becomes 64-bit again.

The keys used include, among others, 18 32-bit subkeys incorporated in the P-array (P1, P2, ..., P18). In addition, there are four 32-bit S-boxes each with 256 entries:

S1,0,S1,1,...,S1,255; S2,0, S2,1, ..., S2,255; S3,0, S3,1,..., S3,255; S4,0, S4,1, ..., S4,255.

In the feistel network, blowfish has 16 iterations, the input is a 64-bit data element, X. To perform the encryption process:

1) Divide X into two parts, each consisting of 32-bit: XL, XR.
2) *For i = 1 to 16:*
   XL = XL XOR Pi
   XR = F(XL) XOR XR
   Switch XL and XR
3) After the sixteenth iteration, exchange XL and XR again to undo the last account.

4) Then do it
   XR = XR XOR P17
   XL = XL XOR P1
5) Finally, reassemble XL and XR to get the cipherteks**.**

In the blowfish algorithm there is uniqueness in terms of the decryption process, ie the decryption process is done in the exact same sequence with the encryption process, only on the processdekripsi P1, P2, ..., P18 used in reverse order.

2.6. Decryption Blowfish Algorithm

Decryption is exactly the same as encryption, except P_1, P_2, ..., P_{18} is used in reverse order. except that P_1, P_2, ..., P_{18}. Decryption for Blowfish is forward-looking. Resulting decryption works in the direction of the same algorithm as encryption, but as the input is chipertext. However, as expected, the sub-keys are used in reverse order.

Subkeys are calculated using the blowfish algorithm, the method is as follows:

1) First initialize the P-array and then four S-boxes in sequence with a fixed string. This string consists of hexadecimal digits of pi.
2) XOR P1 with the first 32 bits of the key, XOR P2 with the second 32 bits of the key and so on for each bit of the key (until P18). Repeat to key bits until all P-arrays in XOR with key bits.
3) Encrypt all zeros with the blowfish algorithm using the subkey as described in steps (1) and (2).
4) Replace P1 and P2 with output of step (3)
5) Encrypt the output from step (3) with the blowfish algorithm with a modified subkey.
6) Replace P3 and P4 with the output of step (5).
7) Continue the process, replace all elements of the P-array, and then all four S-boxes in sequence, with continuously changing outputs of the Blowfish algorithm.

The method of Blowfish algorithm is done by reversing the existing 18 subkey. What we will do first is that this problem seems unreliable, because there are two XOR operations following the previous f-function, and only one previously used f-function. Even if we modify the algorithm so that the use of subkey 2 to 17 puts before the output of the XOR-function to the right of the block and done to the same data before the XOR, even though it means it is now on the right hand side of the block, since The XOR subkey has been moved before the swap (exchange) both halves of the block (exchange half of left block and half right block). We do not change anything because the same information is XORed to half the left block between each time, this information is used as the f-function input. In fact, we have the exact opposite of the decryption sequence.

2.7. Data Compression

Data compression techniques and technology are ever-evolving with new applications in image, speech, text, audio and video. This new edition includes all the latest developments in the field [9]. The field of data compression is one of the fundamental fields in information theory. Information Theory itself is a branch of mathematical science born in the late 1940s through the work of Claude Shannon at Bell's lab. Data compression becomes one of the branches of information theory because data compression is dabbling with the redundant problem of consuming extra bits to encode it and if the extra bits can be eliminated it means there has been a reduction in the size of the information.

World data compression itself can be divided into two, namely lossy data compression and lossles data compression. Lossy data compression is a compression technique that allows the loss of multiple data bits with the return reward of a fairly high level of compression. Examples of these compression techniques are widely implemented in the multimedia world, such as compression of image files (jpg, gif, png, etc.) and video files (mpg, avi, rmvb). Lossy algorithm can be enjoyed even if there are some missing bits, although of course there is a slight decrease in quality.

Meanwhile, lossles data compression algorithm is a data compression technique that does not allow data loss in both compression and decompression processes. Applications of data lossles compression today is very wide, as in archiving applications such as WinZip, WinRar, and WinAce.In general the

process of data compression consists of the process of taking the stream (stream) of the symbols and transform them into codes. If the compression level is effective enough, the output stream will have a smaller size of the input stream. The decision to output a code is based on a model. The model itself is a collection of data and rules that process input symbols and determine which code should be generated. The compression program uses a model to accurately define each symbol and code to be generated based on some probability.

2.8. Huffman Data Compression Algorithm

Huffman coding is based on the frequency of occurrence of a data item i.e. pixel in images. the technique is to use a lower number of bits to encode the data in to binary codes that occurs more frequently. It is used in JPEG files [10]. The Huffman algorithm is one of the oldest compression algorithms but still rated as one of the most powerful data compression algorithms. The Huffman algorithm was first published in 1952 by D.A. Huffman in his paper entitled "A Method for the Construction of Minimum Redundancy Codes". In general, this algorithm can provide savings of 20% -30%. The Huffman algorithm can be classified as a lossless data compression algorithm because no bits are lost during the compression or decompression process. While based on coding technique using symbol wise method. The Huffman algorithm is one of the algorithms used to compress text. The Huffman algorithm is complete:

1) Select two symbols with the least probability (in the example above symbols B and D). Both symbols are combined as parent nodes of symbols B and D so that the symbol BD with probability 1/7 + 1/7 = 2/7, the number of chances of both children.
2) Next, select the next two symbols, including the new symbol, which has the smallest chance.
3) Repeat steps 1 and 2 until all symbols are depleted.

To decompile previously encoded data with the Huffman algorithm, the following methods can be used:

1) Read the first bit of the binary string input.
2) Perform a traversal on the Huffman tree starting from the root according to the read bit. If the read bit is 0 then read the left child, but if the read bit is 1 then read the right child.
3) If the child of the tree is not a leaf (node without child) then read the next bit of the input binary string.
4) It is repeated (traversal) up to find leaves.
5) On the leaf the symbol is found and the decomposition process the code is complete.
6) The decoding process is done until the entire input binary string is processed

3. Results

3.1. Problem Analysis

Documents are very important data whether it be a personal document, company or organization and so forth. Therefore, a document should be kept confidential so as not to be abused by unauthorized persons. Often the security issue becomes second or even the last order in the list of things that are considered important. When disrupting system performance, these security issues are often reduced or even eliminated. One way to secure a document is to convert an original document into a document that can not be read by others or often called encryption. To implement encryption of documents requires encryption algorithm so that the document can be encrypted and then returned as normal or decrypted without experiencing changes. So it needs applications that can provide solutions to existing problems.

3.2. Problem solving

To solve the above problem, then made an application that can keep the confidentiality of a document or data. The application will be able to convert a document into a file whose contents can not be read and the document is kept confidential. Then restore the document to be the original without experiencing a flaw or a slight change. With this application is expected a document or important data can be stored and sent to the party who really authorized and not abused by the parties who are not responsible.

3.3. Program Design

The program design stage is done to find the optimal form and the program to be made by considering the problem factor and requirement that have been explained previously. Efforts are made to try to find a combination of the use of hardware (hardware) and software (software) is right so that obtained optimal results and easy to implement.

The program created consists of Login Form and Form Menu Utama. Form Main Menu consists of Form Home, Form Encryption, Form Decryption, Form Help and Logout.

To encrypt the file, the user can select the encryption menu. In this menu, the user is required to choose file PDF, DOC file, XLS file or Text file first, then do the encryption process. But file PDF, DOC files, XLS files or text files can not be larger than the file size that has been determined, then will display the output of information encrypted file.

Meanwhile, to restore the encrypted file to the original file, the user can also choose the decryption menu. And provide menuHelp to help users in using the program. Design of Encryption and Decryption Case Test

1 Case Test 1Test case 1 gives the test scenario the size of filePDF, fileDOC, XLS file or fileText by encrypting and decrypting using Blowfish and Huffman Compression algorithms.
2 Case Test 2Test case 2 provides a scenario comparing the time of the encryption process and decryption between the encrypted file and the decrypted file using the Blowfish and Huffman Compression algorithms.
3 Case Test 3Test case 3 provides a scenario where the file to be decrypted is a previously encrypted file using the Blowfish algorithm and Huffman Compression
4. Case Test 4Test case 4 provides a scenario testing the correctness of the use of the key (password) on the process of encryption and decryption. The test will work if the decrypted file has the same contents as the original file using the Blowfish and Huffman Compression algorithms.

3.4. Screen Design

Screen design is very important in making a program. Therefore, the design of the screen should be easy to understand and understand, so that in using the program users feel comfortable in using it. Thus, the design of the screen does not confuse a user and has no difficulty in using this application. In this program, will be drawn the design of the screen of each form, the design of the login form screen, main menu, form encryption, form decryption, and form help.

3.5. Program Algorithm

The following is an algorithm used in cryptographic applications using the Blowfish and Huffman Compression methods.

3.6. Main Menu Screen Design

Algorithm Login form describes the use before using this program the user must fill out the form login to bisamasuk into the main menu and use this program. For login user must enter username and password correctly**.**

1. Show Login Form
2. Input Username and Password
3. If Check Username and Password Then
4. Show Main Menu
5. Else
6. Show Error Message
7. Return to Line 2
8. End If

3.7. Decryption Process Algorithm

This decryption process algorithm explains how the process of restoring the encrypted data into the original file with the algorithm method is reversed from the encryption process. where to begin the decryption process, the first algorithm used is Huffman Compression followed by Blowfish.

1. Flowchart Algorithm System In this application consists of
2 algorithms. Both algorithms are the Blowfish algorithm and the Huffman Compression algorithm. Both algorithms are used for each process in encryption and decryption as discussed earlier.

4. Conclusion

This cryptographic application uses two algorithms namely Blowfish algorithm and Huffman Compression built to be able to secure the type of PDF files, DOC, XLS and text. Huffman Compression Algorithm is used to minimize size after encryption with Blowfish algorithm. An encrypted application can not be opened or restored as it was without a key inputted during encryption. Encryption process time is faster than decryption process time. With this cryptographic application, the storage process and information exchange become safer.

References

[1]. Adrisatria, Y. (2013). Penerapan Algoritma Huffman dalam dunia Kriptografi. Bandung: Institut Teknologi Bandung.
[2]. Andri,H.(2005). Implementasi Kriptografi Algoritma Blowfish dan Skipjack Pada Jaringan Client/ Server. Bandung: Universitas Komputer.
[3]. Sadikin, R.(2012). Kriptografi Untuk Keamanan Jaringan. Yogyakarta: Andi.
[4]. Sidik, Betha.(2012).Pemprograman Web PHP. Bandung: Informatika.
[5]. Trisnawati. (2008) "Sistem Keamanan Menggunakan Algoritma Blowfish Pada File Dan Folder Data. Palembang: Universitas Sriwijaya.
[6]. Wawan, I (2005).Implementasi Algoritma Run Legh, Half Byte Dan Huffman Untuk Kompresi File. Yogyakarta: Universitas Islam Indonesia.
[7]. Yudi, P. (2014). Implementasi Kriptografi Menggunakan Metode Kompresi Huffman. Jakarta: Universitas Budi Luhur.
[8]. Astari, R. &Yaya, .T. (2016).Classify interval range of crime forecasting for crime prevention decision making.Knowledge, Information and Creativity Support Systems (KICSS), 2016 11th International Conference on, Yogyakarta, Indonesia.
[9]. Khalid Sayood, (2017), Introduction to Data Compression, ISBN : 9780128094747, 5 Edition.
[10]. Mamta Sharma. (2010). Compression Using Huffman Coding. IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.5.