International Journal of Engineering & Technology, 7 (2.26) (2018) 109-118

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Nur Algorithm on Data Encryption and Decryption

Nur Aminudin[1], Andino Maseleno[1], Shankar K[2], S. Hemalatha[3], K. Sathesh kumar[2], Fauzi[1], Rita Irviani[1], Muhamad Muslihudin[1]

[1]Department of Information System, STMIK Pringsewu, Lampung, Indonesia
[2]School of Computing, Kalasalingam Academy of Research and Education, Krishnankoil, Tamilnadu, India
[3]Department of Computer Science, Karpagam Academy of Higher Education, Coimbatore, India.
**Corresponding author E-mail: andimaseleno@gmail.com*

Abstract

Security is a priority in information system, especially in the exchange of data that are important or confidential. The information to be given to the party entitled to the information must be properly safeguarded, don't fall into the other hands who have no right to such information. One way to maintain the security of information exchanged in a system can be done using cryptographic techniques. Cryptography is the art and science to hide information from third parties. In cryptography a person who has a private key can convert plaintext data into unique and unreadable data (ciphertext) and can convert existing ciphertext into plaintext form by using its private key. System development (System Development Lifecycle) can mean to construct a new system to replace old system, combined with prototyping technique to build a cryptographic system using Nur algorithm which is implemented using programming language used is assembly (MASM32). In Nur Aminuddin's Encryptor there are two data-reading techniques namely encryption technique (the technique of converting data from the original into unreadable code) and decryption technique (the techniques of reading unreadable codes become readable) Encryption technique is built by applying technique of modern cryptography which holds secrecy on the symmetric key, so the security of encryption depends only on the key and does not depend on whether the algorithm is known to people or not.

Keywords: *cryptography, private key, Nur algorithm, assembly, masm32*

1. Introduction

The progress of information systems has many advantages but also prone to negative things like information theft. For example in a company, some information that is confidential and should only be known by certain people in a company such as information about how to create products that are being developed. If the data which contain the information fall to the business counterpart, then the company will incur losses. To anticipate unwanted things like information theft, it takes a system to secure an information. Competitive information security is a very important issue for a company, an institution, a college or an individual. Systems for securing information on these issues can benefit from cryptography.

Secrecy and security are required in the science of cryptography. Cryptography itself has been known since the time of the Roman Empire when Julius Caesar led. At that time the way in conveying secret messages should not be known by any party especially to the enemy. The Caesarean method is the method used at the time which is known until now. The Caesarian method itself is a way of keeping message secret at that time by using shift techniques by shifting each letter to the right or left on the alphabetical order of some fruit. In the days of the second world war, cryptography was also needed to mess up the message. The media used at the time IS the machine to encrypt messages called Enigma. Enigma is the mainstay machine used by Germany, but the allies managed to solve it.

The cryptographic's objective that is to provide security services (also called security aspects) ie confidentiality, is a service intended to keep message unreadable by unauthorized parties. In cryptography, this service is realized by encoding the message into cypherteks. Data integrity (data integrity), is a service that ensures that the message is still original / intact or has never been manipulated during the delivery. In other words, this security aspect can be expressed as a statement: "Is the received message still original or unchanged (modified)?". To maintain data integrity, the system must have the ability to detect message manipulation by unauthorized parties, including insertion, deletion, and substitution of other data into actual messages. Authentication (authentication), is a service related to the identification, either identify the authenticity of the parties communicate (user authentication or entity authentication) or identify the truth of the source message (data origin authentication). Two parties who communicate with each other must be able to authenticate with each other so that they can confirm the source of the message. Messages sent over the communication channel also have to be authenticated originally. In other words, this security aspect can be expressed as a question: "Does the received message really come from the correct sender?". Message source authentication implicitly also provides data integrity certainty, because if the message has been modified it means the message source is incorrect. Therefore, data integrity services are always combined with the message source authentication service. Abuse (non-repudiation), is a service to prevent communicating entities from denial, ie sending a message deny sending or the recipient of the message denies receiving the message.

From the four fundamental objectives of the cryptography, the problem is limited only to the purpose of secrecy and integrity, two other objectives are set aside from this design and discussion. Confidentiality and security when exchanging data is crucial in data communications, both for shared security purposes, and for

individual privacy. Those who want the data is not known by the parties who are not concerned always trying to get around how to secure the information to be communicated. Protection against confidentiality increases, one way by encrypting data or encryption. Various techniques, logic, and cryptographic algorithms have emerged in the world [1] – [27]. Some require an understanding of the cryptographic mechanism itself and some require its use as commercial or shareware software. Understanding of cryptographic mechanisms allows one to take the time to study the mechanism or the cryptographic support software in question. On the other hand, the need to encode data may have been provisional.

Commercial requirement on some specific software makes people have to set aside funds to buy the product. Even if funds are available, a payment mechanism involving transactions between countries is still possible to be a constraint. For ordinary people and students who want to learn cryptographic technique from the start, complex cryptographic and plot mechanisms often become obstacles to learn from their origins. An introduction is needed to make that understanding accessible. This research uses two stages of research there are stage design and independent evaluation phase. The more difficult an algorithm, the more computation it will take, the more time it will take to execute it. While users never think about how difficult algorithms are used, the importance of secrecy of their data is guaranteed. In cryptography with modern security, users are often confronted with complicated rules in order to achieve the expected security system. These rules are often difficult to understand by basics of computers or people who only know the basics of computers. Based on the data above, the researcher concludes, how to make a technique and application program is simple but still can maintain data confidentiality. Therefore, it needs an encryption tool that is quite simple in its understanding and operation but it is reliable in security, and is able to provide understanding for those who want to learn it and develop it.

This research aims to present a design and a real application program that can be used to encrypt files and decrypt files that are safe enough and easy to use by a novice user, and enable encryption learning process to improve understanding of sustainable encryption techniques. The results of this study are expected to provide benefits that is for the general public, the results of this study can be used as a simple but reliable enough encryption tool without too many installation mechanisms to be done and understanding to be poured out. For the next researcher, the results of this study can provide an enrichment of understanding of the fundamental encryption technique so that it can be developed into a wider logic.

2. Nur Algorithm

The password algorithm is an algorithm that serves to perform cryptographic purposes. The algorithm must have the power to configure, from bright text so that it is difficult to reconstruct directly without using the decryption algorithm and difusion, from bright text so that the characteristics of the bright text are lost. So it can be used to secure information. In implementation a password algorithm should consider the quality of service / quality of service or QoS of the entire system in which it is implemented. A reliable password algorithm is a password algorithm whose strength lies in the key, not the secrecy of the algorithm itself. Techniques and methods for testing the reliability of password algorithms are cryptanalysis.

The underlying mathematical basis of the process of encryption and decryption is the relation between two sets, which contain a plaintext text element and cyphertext text element. Encryption and decryption are functions of transformation between the sets. If the bright text elements are denoted by P, the elements of the cipher text are denoted by C, while for the encryption process denoted by E, the decryption with the D notation.

encryption : $E(P) = C$

description : $D(C) = P$ or $D(E(P)) = P$

Generally based on the key of similarity, the password of algorithm is divided into symmetric-key keys, often called conventional password algorithms because they are generally applied to classic password algorithms and asymmetric / asymetric-key. Based on the direction of its implementation and its timing is divided into classic cryptography, classic password algorithm and modern cryptography, modern password algorithm. Based on the confidentiality of the key is distinguished into secret key secret-secret key algorithm and public-key public key-password algorithm.

In the symmetric key-scheme, a secret key is used to perform the encryption and decryption process. While the key-asymmetric system used a pair of different keys, commonly called the public key (public key) and private key (private key), used for the process of encryption and decryption process. When a light text element is encrypted using a private key, the resulting text element can only be decrypted by using its private key pair and vice versa, if the private key is used for the encryption process then the decryption process must use the partner's public key.

This study by understanding the principal above rules tries to offer a security alternative using its own design of encryption techniques. The design is named after the Nur method. In short, the logic of Nur adds plus = added in the binary / internal mechanism of the computer is implemented by rotating 0 - 255 as far as the desired distance (termed step). Example 0 step 2 then 2, times = by multiplying the result number from stage 1 as much as the desired multiplier (multiplier is 1 - 254), plus = 2nd stage result plus the desired value (termed step 2), formula step1-mul1-step2. Example if added 2 multiplied 2 plus again with 5 then the formula 2-2-5. The encryption scheme to be built on this research applies the technique to modern cryptography, which holds the secrecy of the symmetric key, so that the security of encryption depends only on the key and does not depend on whether the algorithm which is known to people or not.

Nur logic is built by developing existing techniques above that is addition. The addition of complexity is done by the mechanism of multiplication. To clarify the appearance of the multiplication argument and the mechanisms executed, the explanation will be detailed with the information below.

In the first step of information blurring, which is applied is the addition mechanism as illustrated above. In the addition mechanism (step 1) the byte data with the fixed byte-size addition because the addition of the value will make the value increase without changing the size as long as the resulting data lies between 00h (0 decimal) to FFh (255 decimal). In situations the value exceeds 255 (for example the initial data of 200 decimal (C8h) plus 100 (64h) will result in a value of 300 (12Ch). If reflected in machine code form, this mechanism will produce 1100 1000 b + 110 0100 b = 1 0010 1100 b. In machine language command the addition of two digits each of the byte sizes when passing 255 decimal will produce the final 8 bit result and the excess is stored in the carry register register, thus the result above (300 decimal or 1 0010 1100 b will be reflected in the data as 0010 1100 b (44 decimal) and carry value is 1). This means that data is only taken as 8 bits (or 1 byte) only and stored as encryption result of first step If the number is reversed (44 - 100) will result in a value of -56 (1100 1000 b with a register of borrows worth 1). This number equals the positive value 200 which also has a symbol of 100000 b, first step doesn't not get obstacles.

After data is scrambled with the mechanism above (addition mechanism), in the second stage the data is deemed necessary to be randomized. The only way to randomize the remainder of the multiplication remains. Why is that? The explanation is easy. If a temporary reduction of 3 has been added 5, this is tantamount to adding 2 (5 -3). Meanwhile, the division technique cannot be done because if the number is a primary number, it will produce fractions. In the hexadecimal symbol united bytes, it does not exist. The solution is to apply multiplication because it will definitely generate integers. However, there are consequences that

must be borne. If it only multiplies 5h x 5 h then the result is 25 decimal or 19h. the size of the data can still be stored in 1 byte. If multiplying 50h x 50h then the result is 1900h. The size of this data will not fit in the byte. Consequently the data should be stored in the word (2 bytes).

As with the provision of data storage in a computer, the multiplication mechanism requires transferring byte-sized data into word (2 bytes). The accommodation is necessary to keep the product intact. The consequence is that the maximum value of 1 byte of data is FFh (255 decimal) if multiplied by a maximum value of 1 byte (FFh or 255 decimal) then the result is FE01h or 65025 decimal. Meanwhile the 2 bytes capacity will be able to dump up to FFFFh or 65535 decimal. From the description it is clear that the file size folds to twice the original file size. Thus there are limitations of encryption processing if the files to be encrypted are larger than 1 gigabyte.

After the data is multiplied and stored in memory, the original byte-size data has been turned into 2 bytes. The randomization mechanism is re-enacted in this third step by adding a specific number (optional) into each data byte. Data becomes more complicated. This logic is the same as the first step of logic and the explanation does not need to be repeated again. If people want to crack the code without knowing how much is added at the beginning, then how many next multiplier and how many adders go next, then it will be very difficult to figure it out. This mechanism will be more complicated if the encrypted file is re-encrypted for the second time to three times and so on

Although it is recognized there is some weakness, the encrypted file will fold to twice the size of the previous file so that it will experience limitations when dealing with data sized above 1 gigabyte, this weakness can be solved by breaking the initial file first into sizes less than 1 gigabyte for each and then encrypt the fragments of the file. To increase the level of data confidentiality (although it has been encrypted), the sender / sender just sends the file and encode the code. The transmissions can be separate, sample files via email, password via sms. Thus the decryption engine can be designed to translate encrypted data by way of reversing ie minus 5, divided by 2, minus 2. To decrypt, the encrypted file should be decrypted and reassembled through the file merge utility.

Requirements and functions required by the user and must be implemented in the software are determined as follows:

1. Determining the programming language used. The programming language used is assembly.
2. Determining the Assembler used in this case selected MASM32 (http://www.masm32.com)
3. Determing coding rules and password disclosure. The coding rules will be explained in the Coding and Programming sub chapter.
4. Once the program is compiled / coded the program application is tested first by volunteers that will to test it. The absolute demand of the test result is that all source files must be encrypted and decrypted back completely without any defects. If there is a defect, the defect is addressed. If there is no defect the application program is deemed to have satisfied encoding and research needs.
5. Software is determined to meet the needs of most computer users that the software must be Microsoft Windows-based.
6. For convenience preferences, the software takes precedence over layout windows that can be executed directly and enjoyed by users. This means avoiding the process of typing the commands for the program to run properly. In this case the software will be enough to run by double-clicking its exe file and the program has been running and appearing in front of the user.
7. Software has a menu that is easy to understand. The use of the menu allows the user to select the desired functionality as usual of other programs they have known before.
8. Software has facilities to find files that will be encrypted and decrypted with the facilities owned by windows. This means the user no longer has to type the location of the file to be encrypted or decrypted. Typing will make it difficult for a novice user because typing allows small errors such as spaces or typos to prevent the program from finding the file in question. Thus, the software simply presents search / grab to retrieve the file in question.
9. Software has a function to store the result of the encrypted / decrypted file, so that the results that will be used as the result of encryption / decryption.
10. Software has the facility to determine the desired encryption code according to user preferences. The flexibility of choice is preferable so that only interested users can know and only interested users can decrypt it. Thus, the undisclosed information is not leaked to unauthorized parties.

2.1 Program Layout Design and User Interface

The design should capture and document an independent technology view of the process to automate as well as planning, system architects to ensure all designs are compatible and free of integrity issues and the technical architecture is feasible.

From the above function requirement, determined the desired layout and user interface that suits the needs of users as follows:

1. Layout is designed using graphical user interface (GUI) and there is menu to facilitate user to choose the desired function.
2. There is a button in the encryption / decryption window in the software to retrieve the desired file
3. There is a user interface function to store the result of an encrypted / decrypted file so that the result will be used as the result of encryption / decryption.
4. There is a window that has the facility to define your own desired encryption code according to user preferences.

2.2 Coding and Programming (Application of Nur Logic in Software)

After the initial needs and layouts are marked, Nur's logic is applicated in the form of software with programming languages. Some alternative programming languages can actually be chosen depending on the actor's familiarity with the programming language and the demands of the efficiency and effectiveness of the final program. For this purpose selected assembly language that can be obtained from Microsoft MASM32 (http://www.movsd.com). Application in a program which involves coding activity is possible to involve other parties to ensure the correct coding in accordance with the objective of the study ie to present an effective and efficient program.

Software is designed using the libraries (dll) that have been provided in Windows. Thus, according to the purpose of flexible programming and not dependent on other utilities, the program can be run directly elsewhere (another Windows based computer) simply by copying the file without any installation mechanism. Some rules for coding and programming are determined as follows:

1. Password is arranged with the rules of each byte taken to be processed by the rules of +, x, +. This means that the byte code is encrypted with step / shift of x, multiplied (mul) of y and added by z.
2. The program is arranged with the rules of reading the source file, taking the data byte by byte then processing it according to the fixed rules of provision. In the form of this technique the formula can be written with Y = ((x + step1) x Mul1) + step 2 with x is the source file byte and y is the word formation.
3. Encoding result will change value and size, value change according to processing code while size from byte become word (1 byte become s 2 bytes).
4. Encoding result is stored in the form of new files and new extensions (name and extension freely selected), the extention should be named .nur.

5. The encoding result is ready to be used / moved / submitted to the desired person.
6. The recipient returns an encrypted document with the rule: -: - , it means that any code (2 bytes) of the encoded document is reversed by the reverse step 1 rule of z divided (div) by y and the 2nd backward distep by x. In the form of this technique, it can be written with x = ((y - step backward 1) / div 1) - step backward 2. Thus writing the program code must include the option menu for decryption and in the window there is a choice of keys for description.

2.3 Software Testing (Self and Limited Testing)

Once the initial form of the program has been compiled, the program is used by the researcher to try to encrypt the file. Disabilities and defects are improved and refined. The next stage, this software is welcome to be tested by close friends who have competencies in the field of computers that understand the purpose of encryption and decryption. Input from them become material for improvement and correction either from side of algorithm, coding, display or other functions of optimization.

Some performance tests used to test Nur Encryptor are as follows:

1. Load testing: is the simplest form of performance testing. Load testing is usually raised to determine the behavior of a system in the specific expected load scope. Load can be done by implementing the program simultaneously on a continuous basis, to process transactions of a certain time duration. This test will answer the time required of all critical business transactions. In this case scope is limited only to software testing of Nur Encryptor. The purpose of this test is to mark where the blockage is generated by this program against the computer operating system.
2. Stress testing: normally used to determine the upper limit capacity of the system / program. This type of test is performed to determine the system's toughness in extreme load situation and helps the application administrator to determine whether the system will perform adequately if the load is above the expected maximum value.
3. Endurance testing (soak testing): This test is done by determining whether the system can persist in a certain load continuously. During this process, the use of memory is monitored to detect potential shrinkage. It is often reviewed the aspect of performance degradation to ensure that the response time remains good after a prolonged program period at the start of the program.
4. Configuration testing: This test is created to determine the effect of configuration changes on components on the performance and behavior of the system / program (Nur Encryptor).

Considering this system is used for the same operating system that is Windows XP upwards, then isolation testing is not used.

2.4 Software Evaluation by Evaluator

After the application software is tested by yourself and some competent parties, the software will be tested to independent volunteers who have the competence. These volunteers are parties who have no relationship and interest whatsoever with the researcher related to this research. Basically the type of testing is the same as in the self and limited testing stage. Thus, the results of their inputs are expected to reflect high independence and facts that reflect actual evaluation results. Based on the evaluation results, the software will be repaired if there are deficiencies that can lead the errors. If not, then the software is declared as the final research result for this purpose.

From the results of research on the application of encryption and description of Nur Aminuddin's Encryptor application, using Nur encryption algorithm obtained some results. The encryption and description process occurs when the file is entered into the application. Built-in encryption that applies techniques to modern cryptography, which holds secrecy on symmetric keys, so the security of encryption depends only on the key and does not depend on whether the algorithm is known to people or not.

Nur logic is built by developing existing techniques above that is addition. The addition of complexity is done by the mechanism of multiplication. To clarify the appearance of the multiplication argument and the mechanisms executed. In the first-stage of information blurring, which is applied is the addition mechanism as illustrated above. In the addition of mechanism (step 1) the byte data with the fixed byte-size addition because the addition of the value will make the value increase without changing the size as long as the resulting data lies between 00h (0 decimal) to FFh (255 decimal). In that situation the value exceeds 255 (for example the initial data of 200 decimal (C8h) plus 100 (64h) will result in a value of 300 (12Ch) .If it is reflected in machine code form, this mechanism will produce 1100 1000 b + 110 0100 b = 1 0010 1100 b. In machine language commands the addition of two digits each of the byte sizes when passing 255 decimal will produce the final 8 bit result and the excess is stored in the carry register register, thus the above result (300 decimal or 1 0010 1100 b will be reflected in the data as 0010 1100 b (44 decimal) and carry value is 1). This means that data is only taken as 8 bits (or 1 byte) and stored as encryption result of first step If the number is reversed (44 - 100) , it will result in a value of -56 (1100 1000 b with a register of borrows worth 1). This number equals the positive value 200 which also has a symbol of 100000 b, then first step does not get obstacle.

After data is randomized with the mechanism above (addition mechanism), in the second stage the data is deemed necessary to be randomized. The only way to randomize the remainder of the multiplication remains. Why is that? The explanation is easy. If a temporary reduction of 3 has been added 5, this is tantamount to adding 2 (5 -3). Meanwhile, the division technique can not be done because if the number is a primary number, it will produce fraction. In the hexadecimal symbol united bytes, it does not exist. The solution is to apply multiplication because it will definitely generate integers. However, there are consequences that must be borne. If it only multiplies 5h x 5 h then the result is 25 decimal or 19h. the size of the data can still be stored in 1 byte. If multiplying 50h x 50h then the result is 1900h. The size of this data will not fit in the byte. Consequently the data should be stored in the word (2 bytes).

As with the provision of data storage in a computer, the multiplication mechanism requires transferring byte-sized data into word (2 bytes). The accommodation is necessary to keep the product intact. The consequence is that the maximum value of 1 byte of data is FFh (255 decimal) if it is multiplied by a maximum value of 1 byte (FFh or 255 decimal) then the result is FE01h or 65025 decimal. Meanwhile the 2 byte capacity will be able to dump up to FFFFh or 65535 decimal. From the description it is clear that the file size folds to twice the original file size. Thus there are limitations of encryption processing if the files to be encrypted are larger than 1 gigabyte.

Once the data is multiplied and stored in memory, the original byte-size data has been turned into 2 bytes. The randomization mechanism is re-enacted in this third stage by adding a specific number (optional) into each data byte. Data becomes more complicated. This logic is the same as the first stage of logic and the explanation does not need to be repeated again. If people want to crack the code without knowing how much is added at the beginning, then how many next multiplier and how many adders go next, then it will be very difficult to figure it out. This mechanism will be more complicated if the encrypted file is re-encrypted for the second time to three times and so on.

Although it is recognized that the encrypted file will fold twice from the size of the previous file so that it will experience limitations when dealing with data sized above 1 gigabyte, this weakness can be solved by breaking the initial file first into size less than 1 gigabyte for each and then encrypt the fragments of the file. To increase the level of data confidentiality (although it has been encrypted), the sender / sender just sends the file and en-

codes the code. Transmission can be separate, the sampe of files can be sent via email, the password via sms. Thus the decryption engine can be designed to translate encrypted data by way of reversing ie minus 5, divided by 2, minus 2. To decrypt, the encrypted file should be decrypted and reassembled through the file merge utility.

2.5 Encryption and Decryption Process

In this process the application of Nur algorithm. The files that is entered by the user will be read by the application and then performed the encryption process. Encrypted poses have been performed on several different files to test and to find out the details of the difference between the initial conditions and the condition after the process. In the following results, there are not created decrypted table because the decryption process is the opposite of the encryption process to restore the file into its original form.
Security through the encryption process makes the contents of the message change so that it affects the size of the file that stores the message. Here is a Table 1 showing the difference in size of file encryption using Nur algorithm:

Table 1: The difference in size of file encryption using Nur algorithm

No.	File Name	Original Size	Encryption Size	Difference
1.	AMERIKA.PNG	114 KB	228 KB	114 KB
2.	AMPUNI DOSAKU.FLV	15,558 KB	31,116 KB	15,558 KB
3.	Bendera-Khifalah.Gif	77 KB	154 KB	77 KB
4.	Bendera-Khifalah.Psd	372 KB	744 KB	372 KB
5.	Bendera-Khifalah.Pxr	356 KB	712 KB	356 KB
6.	Berhak Uas.xlsx	14 KB	27 KB	13 KB
7.	chap06Corrected-T.ppt	1,185 KB	2,370 KB	1,185 KB
8.	CONTOH.txt	2 KB	4 KB	2 KB
9.	Isak Mengana Untu-ang.mp3	4,496	8,992 KB	4,496 KB
10.	Modul sistem pa-kar.docx	32 KB	64 KB	32 KB
11.	Nur Aminuddin.JPG	103 KB	205 KB	102 KB
12.	Surat Al-Fatiha.mp3	821 KB	1,641 KB	820 KB
13.	w02C01.pdf	38 KB	75 KB	37 KB

3. Results and Discussion

The Nur encryption mechanism and its actualization software can be implemented to encrypt all file types as long as the provisions given by the operating system allow it. The study of evidence is examined in the following description. To facilitate the study, it illustrates a simple overview of simple text using the ASCII code which reads "My Name is Nur Aminuddin." The text is typed in notepad.exe and stored as Nur Aminuddin.txt. The above illustration is shown in the Figure 1 below.

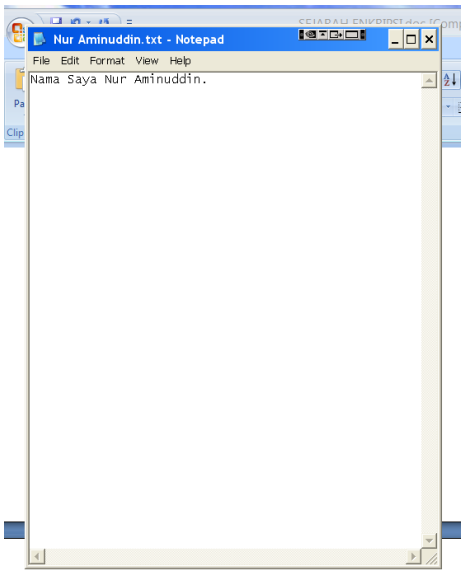

Fig. 1: ASCII Code Text Image

In binary form, the display of what is in the file can be seen in the following Figure 2.

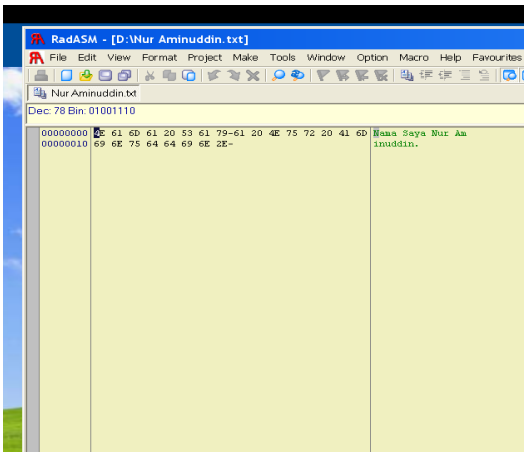

Fig. 2: The display of Nur Aminuddin.txt in Binary form

Nur Aminuddin.txt then encrypted with code of 10-10-5 with Nur Aminuddin's Encryptor software as shown in Figure 3.

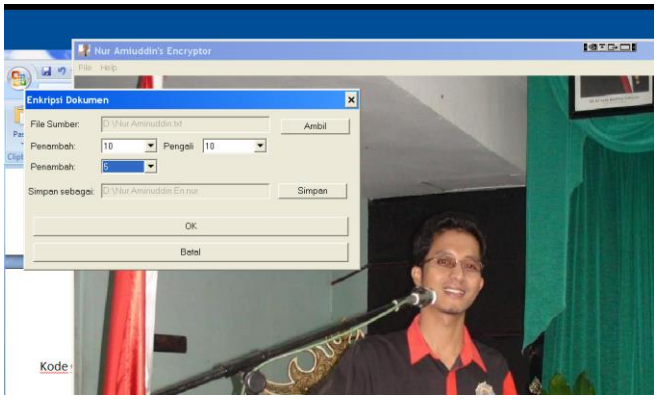

Fig. 3: Encrypted with code of 10-10-5 with Nur Aminudin's Encryptor software

After encrypted the data file can no longer be read through notepad.exe, and if seen in binary form will change as the following Figure 4.

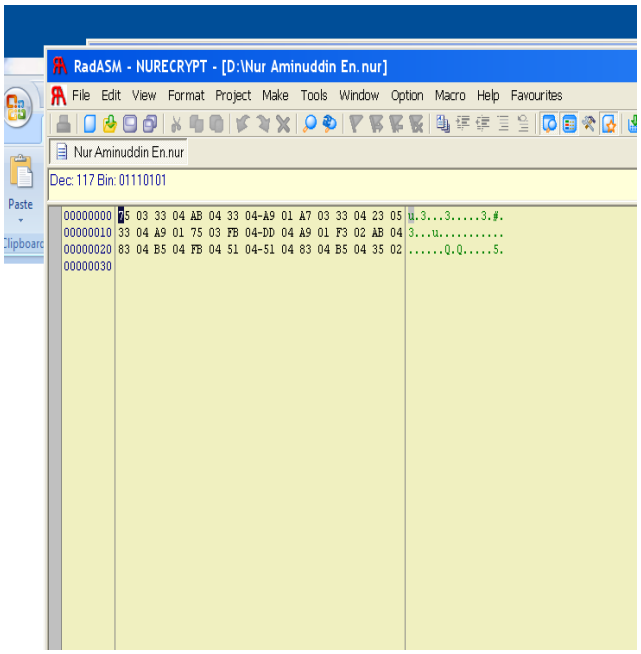

Fig. 4: The display of Nur Aminuddin.txt after 10-10-5 Encryption in Binary Form

Implementation of Nur Encryptor to encrypt an .exe-shaped file is poured in an experiment to encrypt notepad.exe files in Microsoft Windows®. Binary form of the original notepad.exe file can be seen in the Figure 5 below.

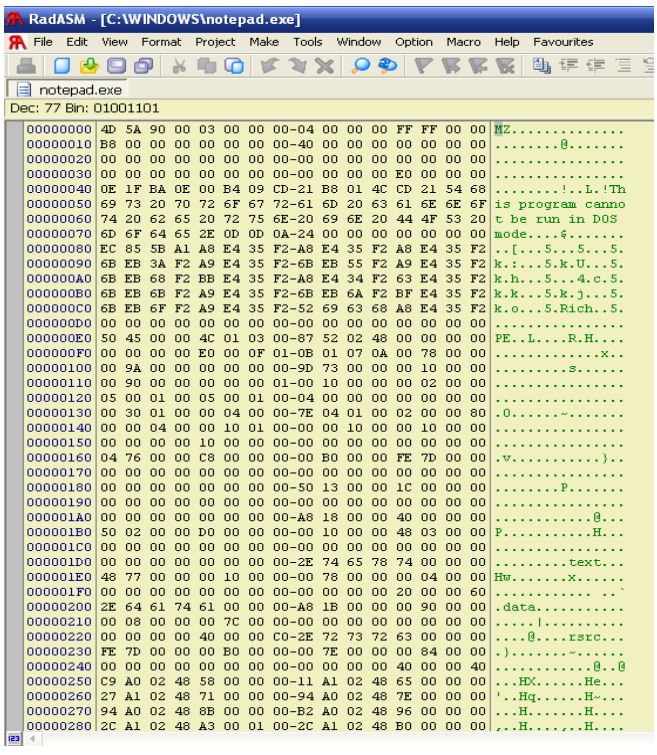

Fig. 5: The Display of notepad.exe before it is encrypted in Binary Form

After it is encrypted with code 10-10-5, the notepad.exe file has changed its code and can not be executed anymore. Display data in binary form can be seen in Figure 6 below.

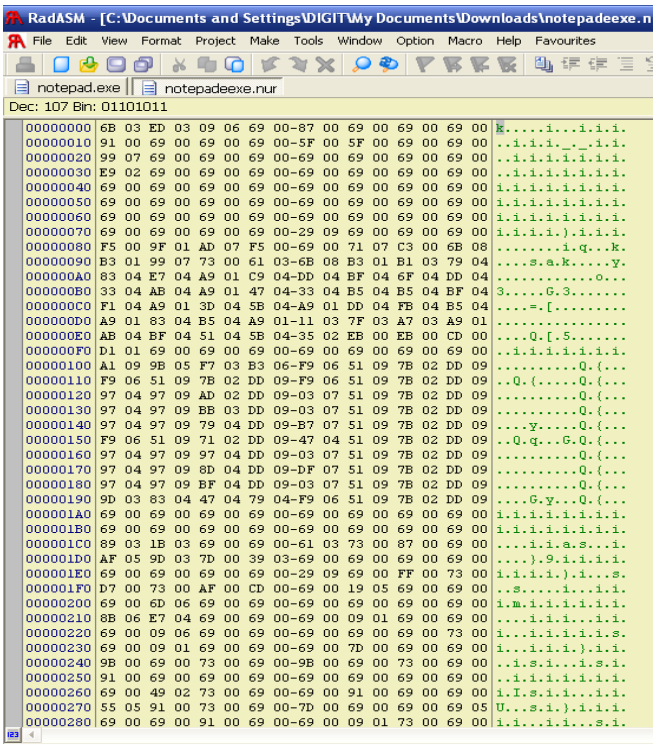

Fig. 6: The display of notepad.exe after 10-10-5 encryption in Binary form

The next experiment is to test Nur Encryptor which is used to encrypt JPG format files. The Figure 7 below is the original figure before it is encrypted.

Fig. 7: JPG file display before Encrypted

The above figure in binary form can be seen in the Figure 8 below.

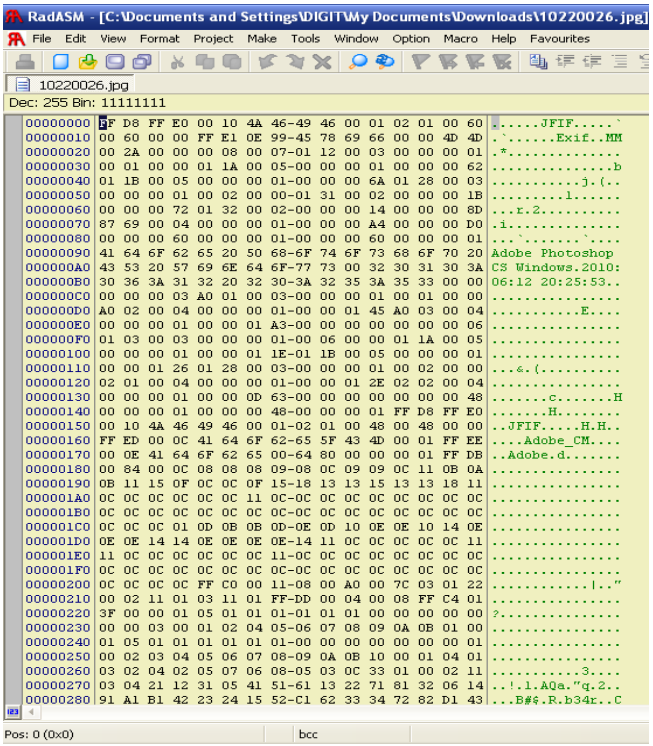

Fig. 8: The Display of the JPG File Binary Code before Encrypted

After it is encrypted the Figure code of Nur Aminuddin will be formed as this Figure 9 below.

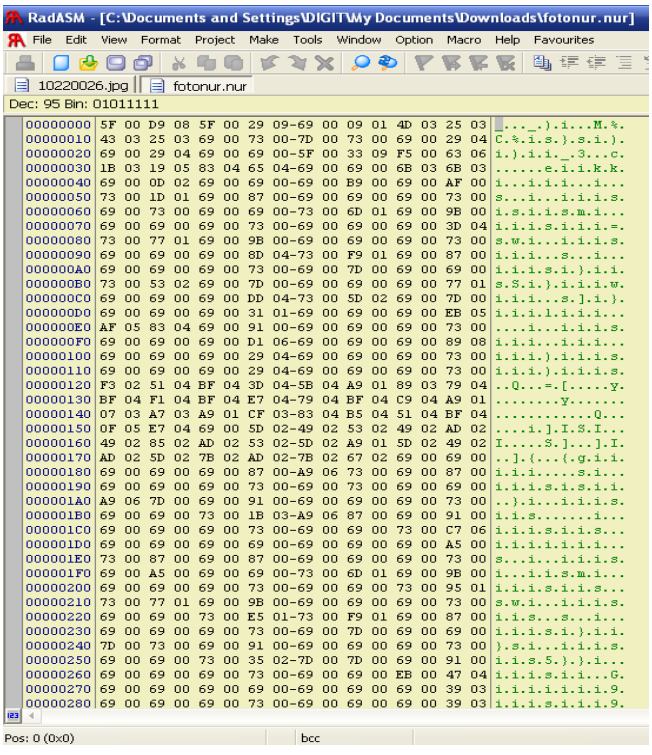

Fig. 9: The Display of Binary Code in JPG form after being Encrypted

The third experiment was continued by encrypting the song file (formed .mp3) The encrypting attempt was done by trying to encrypt the sound formed file with mp3 format with the file titled Surat Al-Fatiha.mp3. Before encrypted, the original binary form of the file titled Surat Al-Fatiha.mp3 is as the following Figure 10.

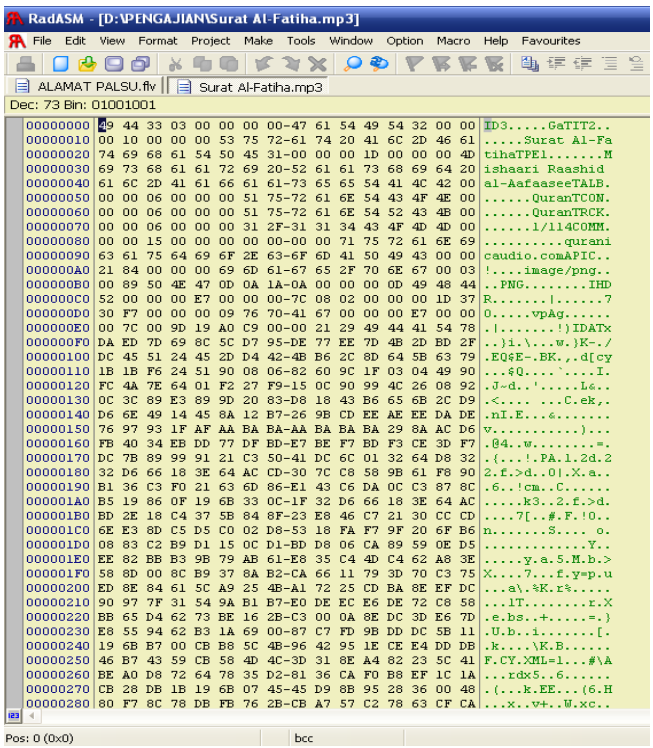

Fig. 10: The Display of Binary Code of Al-Fatiha.mp3 before being encrypted.

After it is encrypted with Nur Aminuddin's Encryptor, the binary code of the file changes to the following Figure 11.

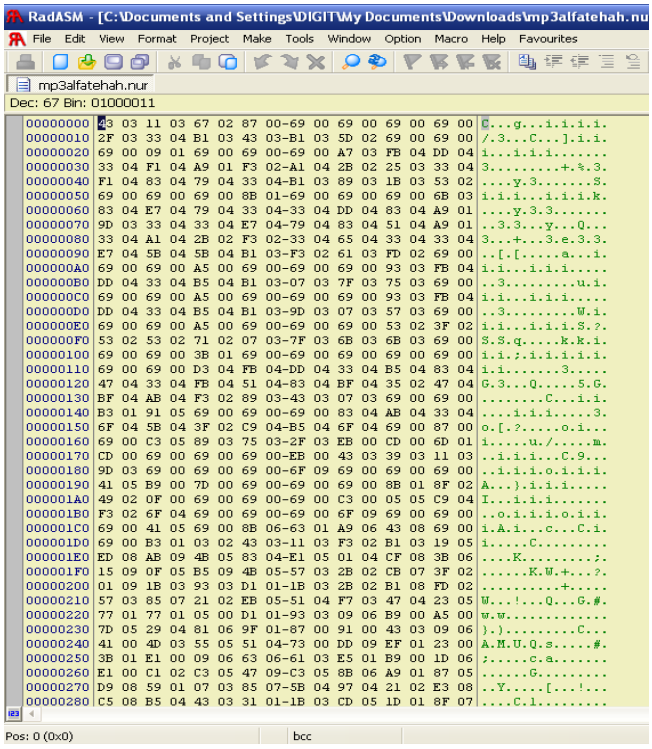

Fig. 11: The Display of binary code of Al-Fatiha.mp3 file after being Encrypted.

The next experiment followed by encrypting a video file in flv format with a file titled ADDRESS PALSU.flv. Before it is encrypted, the original binary form of the file titled FALSE ADDRESS.flv can be seen in the Figure 12 below.

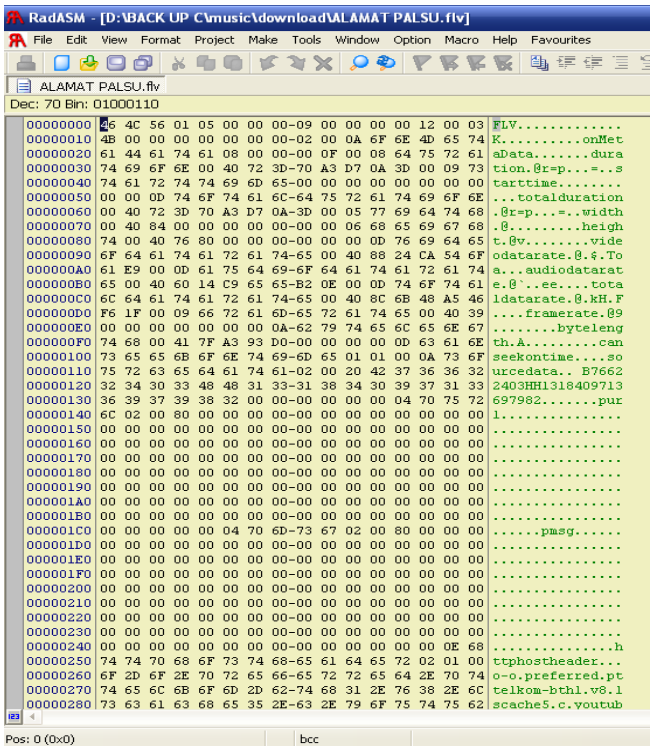

Fig. 12: The Display of Binary Code of ALAMAT PALSU.flv file before being encrypted

After it is encrypted with Nur Aminuddin's Encryptor, the binary code of the file turns into something like the Figure 13 below.

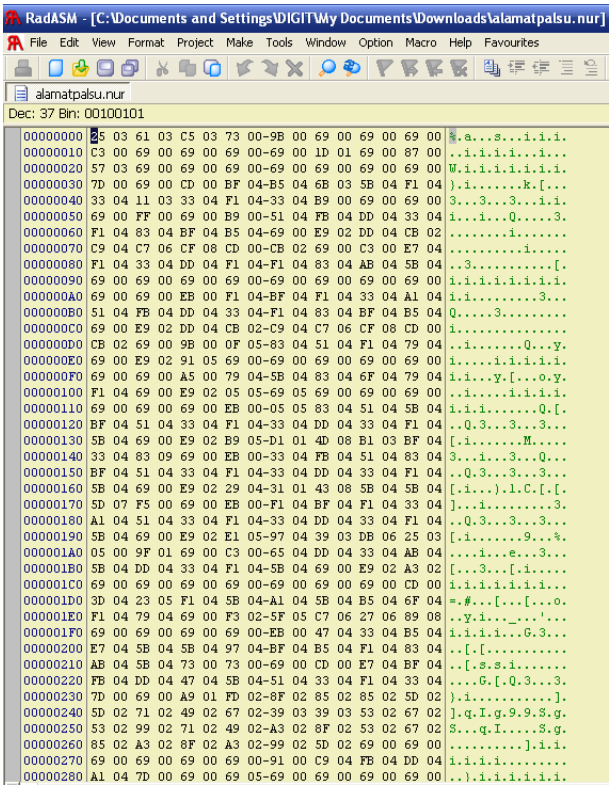

Fig. 13: The Display of Binary Code of ALAMAT PALSU.flv File After being Encrypted

3.1 System Review and Characteristics of Nur Encryptor

For convenience preference, the software takes precedence over layouting windows that can be executed directly and enjoyed by users. This means avoiding the process of typing the commands for the program to run properly. In this case the software will be enough to run by double-clicking its exe file and the program has been running and appearing in front of the user. Software has an easy to understand menu. The use of the menu allows the user to select the desired functionality as usual of other programs that they have known before.

Software has facilities to search files that will be encrypted and decrypted with facilities owned by windows. This means the user no longer has to type the location of the file to be encrypted or decrypted. Typing will make it difficult for a novice user because typing allows small errors such as spaces or typos to prevent the program from finding the file in question. Thus, the software simply presents the search / cap button to retrieve the file in question. Software has a function to store the results of the encrypted / decrypted file, so that the results that will be used as the result of encryption / decryption.

Software has the facility to define the desired encryption code according to user preferences. The flexibility of choice is preferable so that only interested users can know so that only interested users can decrypt it. Thus, the undisclosed information is not leaked to unauthorized parties.

From the speed aspect, given that the supporter software is built with assembly language (using MASM32 Projects) and directly touches the binary data to be encrypted, it works very fast almost without waiting time (average less than 1 second). This result is obtained from the evaluator's information evaluating the limited use of Nur Encryptor software. Processing speed and instantly get results so users can immediately enjoy the results and can perform actions according to the series of plans.

Thus from the aspect of ease, the user interface displays intuitively reflects the ease of use of the software. Support on the ease of use of this software is reinforced by the information of evaluators involved in the Nur Encryptor feasibility evaluation process that the software is very easy to use and does not require mechanisms and understanding of the use or logic and the theory of encryption in depth. Some evaluators say that this software can even be used directly by people who just started to know it after getting a brief explanation.

The most important aspect that needs to be studied is the security aspect of data confidentiality. All evaluators stated that the encrypted results have been very difficult to read, to understand or to reverse. In the general or standard user review, all users when they are prompted to solve (reversed) an encrypted file declare their inability to decrypt again without knowing the key that the encryptor has entered. Even with the use of Nur Encryptor's auxiliary software, volunteers remain unable to break back if they are not informed of the security lock. Thus, from the security aspect of data confidentiality, Nur Enryptor (as well as Nur's Encryption mechanism) is deemed to have adequate security well above the average. The following test results are presented from independent evaluators as shown in Table 2.

Table 2: Test results

Evaluator	Load testing	Stress testing	Endurance testing (soak testing)	Configuration testing
Evaluator 1	*****	**	****	*****
Evaluator 2	*****	***	*****	*****
Evaluator 3	****	**	*****	*****
Evaluator 4	*****	***	*****	*****
Evaluator 5	*****	**	*****	*****

Description: ***** Highest Criteria

The average test results show that this software is proper to use . There are exceptions to stress testing that this software, because of its design characteristics, can only be used to encrypt files under 1 gigabyte. To encrypt files above that number, the source file must be disparted first.

3.2 The Advantage of Nur Alghoritm/Logics/Technique

Some of the advantages of the techniques and actualization of Nur techniques can be described as follows, namely File actualization of the algorithm / mechanism Nur is very small (about 93.6 KB). This very small size makes it easy to copy, it can be carried around without worrying about the limited storage capacity of the media. In addition this file does not use links with the registry in Windows so that when it is copied and run on other computers based on Windows there is no constraint of use. This file is free to use without time limit for both commercial and non-commercial purposes. Some encryption files impose a fee (license purchase) for their use. This means an additional cost. By using Nur Encryptor, users are not required to pay in any form. Thus, the flexibility of use becomes very large, which means the aspect of usefulness is also expected to be very high. Nur Encryptor program works very fast. On average, encryption completion after the OK button is pressed takes less than 1 second. This speed makes time profits so that users are efficient in performing security tasks and other tasks. Nur's mechanism uses logic and an easy-to-understand algorithm. Only by determining the adder, multiplier, and adder that can be analogized with the lock code, the user can directly secure the data as well as remember the security key as a key to decrypt (by sorting in reverse the key).

3.3 Weakness of Nur Encryption Method

Some of the deficiencies of the Nur Encryption method can be marked as follows, ie in terms of super strict security review aspects, Nur's Encryption technique uses a symmetric encryption mechanism which means encrypting code is used to decrypt in reverse order. For super high security where many high level code revealers will try to solve it, these techniques and tools are not recommended for use. In accordance with the explanation in the previous section, this file will generate an encrypted file with a size exactly double the actual file. As with the provision of data storage in a computer, the multiplication mechanism requires transcribing a byte-size data into a word (2 bytes). The accommodation is necessary to keep the product intact. The consequence is that the maximum value of 1 byte of data is FFh (255 decimal) if it is multiplied by a maximum value of 1 byte (FFh or 255 decimal) then the result is FFFFh or 65535 decimal. From the description it is clear that the file size folds twice from the original file size. Thus there are limitations of encryption processing if the files to be encrypted are larger than 1 gigabyte. Although this weakness is recognized, this weakness can be overcome by splitting the initial file first to be less than 1 gigabyte in size for each fraction, then encrypt the file fragments. To decrypt, the live encrypted file is decrypted and reassembled through the file merge utility. Simple is often directly proportional to the security of data. Although it has been designed to be safe, this mechanism is still possible to be traced back. Nevertheless, the search process is not easy and even very difficult to do. To be more complicated and secure, the level of security can be increased by adding another level of enhancement or multiplier aspects in this Nur algorithm. Thus, the possibility to be traced back becomes very small. The logical consequence is that the file size may also be doubled again compared to the previous three levels of encryption. It should be emphasized that the possibility to be traced back is also encountered by various other encryption techniques. There is no perfect human work.

4. Conclusion

In Nur Aminuddin's Encryptor there are two data-reading techniques that are encryption technique (the technique of converting data from the original into unreadable code) and decryption technique (techniques to read unreadable codes become readable) Encryption techniques built, applying technique on cryptography modern, with secrecy on the symmetric key, so that the security of encryption depends only on the key and does not depend on whether the algorithm is known to people or not.Some ways to improve this quality can be achieved by increasing the level of security or randomization. Safety level can be achieved by re-enter the encrypted file into Nur Aminuddin's Encryptor for encryption again (rotate encryption). Rotate the sequence of data so it is not known where the initial data and so on so that the decryption process can only be known only by the right can also be used to improve quality. An encrypted file formed from a combination of techniques already displayed in Nur Aminuddin's Encryptor and byte rotation in the file will make it harder for people to crack the encoded files. The latter is to split the file into several parts so that if a data leak occurs then the party who accidentally find it or who fraudulently get it still can not decrypt the file because of lack of other data. To accommodate this mechanism, on the internet there are several free utilities available for splitting files and can be used as a complement to existing Nur Aminuddin's Encryptor software.

References

[1] Singh, P., & Kumar, S. (2017). Study & analysis of cryptography algorithms : RSA, AES, DES, T-DES, blowfish.*International Journal of Engineering & Technology, 7*(1.5), 221-225. doi:http://dx.doi.org/10.14419/ijet.v7i1.5.9150

[2] Santhosh Kumar, R., & Bharanidharan, R. (2017). Neighbor discovery-based security enhancement using threshold cryptography for IP address assigning in network. *International Journal of Engineering & Technology, 7*(1.1), 439-443. doi:http://dx.doi.org/10.14419/ijet.v7i1.1.11243

[3] Murali Krishna, B., Khan, H., & Madhumati, G. (2017). Reconfigurable pseudo biotic key encryption mechanism for cryptography applications. *International Journal of Engineering & Technology, 7*(1.5), 62-70. doi:http://dx.doi.org/10.14419/ijet.v7i1.5.9124

[4] P. George, J., & Varghese Kureethara, J. (2018). Anefficient 2-Step DNA symmetric cryptography algorithm based on dynamic data structures. *International Journal of Engineering & Technology, 7*(2.6), 141-146.
doi:http://dx.doi.org/10.14419/ijet.v7i2.6.10140

[5] Sri Lakshmi, M., & Srikanth, V. (2018). A Study on Light Weight Cryptography Algorithms for Data Security in IOT. *International Journal of Engineering & Technology, 7*(2.7), 887-890. doi:http://dx.doi.org/10.14419/ijet.v7i2.7.11088

[6] Noorbasha, F., & Suresh, K. (2018). FPGA implementation of RGB image encryption and decryption using DNA cryptography. *International Journal of Engineering & Technology, 7*(2.8), 397-403. doi:http://dx.doi.org/10.14419/ijet.v7i2.8.10469

[7] Bindu Swetha, P., Kishore Sonti, V., & Murali, A. (2017). VLSI design for efficient RSD-Based ECC processor using Karatsuba algorithm. *International Journal of Engineering & Technology, 7*(1.5), 164-169. doi:http://dx.doi.org/10.14419/ijet.v7i1.5.9140

[8] Kumar K, A., E A, N., S, D., & ., R. (2018). Secured cryptographic data model for cloud. *International Journal of Engineering & Technology, 7*(1.7), 128-131. doi:http://dx.doi.org/10.14419/ijet.v7i1.7.10632

[9] Choudhury, S., & Kirubanand, V. (2018). Data encryption in public cloud using multi-phase encryption model.*International Journal of Engineering & Technology, 7*(1), 223-227. doi:http://dx.doi.org/10.14419/ijet.v7i1.9309

[10] Kunchok, T., & Kirubanand V. B, P. (2018). A lightweight hybrid encryption technique to secure IoT data transmission. *International Journal of Engineering & Technology, 7*(2.6), 236-24. doi:http://dx.doi.org/10.14419/ijet.v7i2.6.10776

[11] Aswathy, R., & Malarvizhi, N. (2018). An investigation on cryptographic algorithms usage in IoT contexts.*International Journal*

of Engineering & Technology, *7*(1.7), 10-14. doi:http://dx.doi.org/10.14419/ijet.v7i1.7.9379

[12] K, L. (2018). Enforcing security in cloud environment using elliptic curve cryptography and third party auditing.*International Journal of Engineering & Technology*, *7*(1.7), 84-86. doi:http://dx.doi.org/10.14419/ijet.v7i1.7.9580

[13] Rajan, R., G. Murugaboopathi, D., & C.Parthasarathy, D. (2018). Analysis and assessment of various cryptographic techniques based on a variety of features. *International Journal of Engineering & Technology*, *7*(1.9), 28-33. doi:http://dx.doi.org/10.14419/ijet.v7i1.9.9730

[14] N K, M., D, M., & kumar K, A. (2018). An efficient cryptographic scheme for text message protection.*International Journal of Engineering & Technology*, *7*(1.7), 152-155. doi:http://dx.doi.org/10.14419/ijet.v7i1.7.10639

[15] Jerard, V., & Manimegalai, P. (2017). Content arrangement characteristic based encryption in cloud using public auditing for data management. *International Journal of Engineering & Technology*, *7*(1.1), 30-36. doi:http://dx.doi.org/10.14419/ijet.v7i1.1.8918

[16] R.K, N., R, S., P.G, S., & Sharma, G. (2018). Enhancing security for end users in cloud computing environment using hybrid encryption technique. *International Journal of Engineering & Technology*, *7*(1), 152-156. doi:http://dx.doi.org/10.14419/ijet.v7i1.8340

[17] K, S., & S.K, L. (2018). Optimal key based homomorphic encryption for color image security aid of ant lion optimization algorithm. *International Journal of Engineering & Technology*, *7*(1.9), 22-27. doi:http://dx.doi.org/10.14419/ijet.v7i1.9.9729

[18] Srinivasu, N., Sahil, M., Francis, J., & Pravallika, S. (2017). Security enhanced using honey encryption for private data sharing in cloud. *International Journal of Engineering & Technology*, *7*(1.1), 675-678. doi:http://dx.doi.org/10.14419/ijet.v7i1.1.10826

[19] Dhanasekaran, K., Anandan, P., & Manju, A. (2018). A Computational Approach of Highly Secure Hash Algorithm for Color Image Steganography Using Edge Detection and Honey Encryption Algorithm. *International Journal of Engineering & Technology*, *7*(2.24), 239-242. doi:http://dx.doi.org/10.14419/ijet.v7i2.24.12056

[20] Lalita Kumar Parvataneni, M., Sai Nath Adusumalli, E., & C, R. (2018). Secure and Efficient Query Processing Using Randomized Encryption and De-duplication on the Cloud. *International Journal of Engineering & Technology*, *7*(2.24), 311-315. doi:http://dx.doi.org/10.14419/ijet.v7i2.24.12071

[21] Bhardwaj, A., Som, S., & K. Muttoo, S. (2018). HS1-RIV: Improved Efficiency for Authenticated Encryption.*International Journal of Engineering & Technology*, *7*(2.7), 502-506. doi:http://dx.doi.org/10.14419/ijet.v7i2.7.10871

[22] Pillai, A., S. Vasanthi, M., Kadikar, R., & Amutha, B. (2018). Encryption analysis of AES-Cipher Block Chaining performance in Crypto-Wall Ransomware and SDN based mitigation. *International Journal of Engineering & Technology*, *7*(2.24), 47-54. doi:http://dx.doi.org/10.14419/ijet.v7i2.24.11997

[23] Buchade, S., & Devale, P. (2018). A study on searchable encryption schemes. *International Journal of Engineering & Technology*, *7*(2), 617-620. doi:http://dx.doi.org/10.14419/ijet.v7i2.10995

[24] Vurukonda, N., Trijan kumar, S., Rajasekhar Reddy, J., Adithya, A., & Babu Boddu, S. (2018). A secure attribute-Based encryption scheme in cloud computing. *International Journal of Engineering & Technology*, *7*(2.20), 90-92. doi:http://dx.doi.org/10.14419/ijet.v7i2.20.11761

[25] Noorbasha, F., & Suresh, K. (2018). FPGA implementation of RGB image encryption and decryption using DNA cryptography. *International Journal of Engineering & Technology*, *7*(2.8), 397-403. doi:http://dx.doi.org/10.14419/ijet.v7i2.8.10469

[26] Aradhyamath, S., & Paulose, J. (2018). Multi-key Modified Tiny Encryption Algorithm for HealthCare.*International Journal of Engineering & Technology*, *7*(2), 559-563. doi:http://dx.doi.org/10.14419/ijet.v7i2.9894

[27] Vasantha, R., & Satya Prasad, R. (2017). An identity encryption cloud scheme based on SMTP using advanced blow fish algorithm. *International Journal of Engineering & Technology*, *7*(1.5), 191-195. doi:http://dx.doi.org/10.14419/ijet.v7i1.5.9145

[28] Shankar, K., and P. Eswaran (2017). RGB based multiple share creation in visual cryptography with aid of elliptic curve cryptography. China Communications, 14(2), 118-130.