COMPARATIVE ANALYSIS OF DIFFERENT CLASSICAL AND MODERN CIPHER FOR SECURE DATA COMMUNICATION

[1]NDUNAGU J.N AND [2]OLAWALE M. O
Computer Department, Faculty of Sciences,
National Open University of Nigeria, Jabi, Abuja
[1]jndunagu@noun.edu.ng, [2]funmitomary@gmail.com

ISSN: 2141 – 3290
www.wojast.com

ABSTRACT

Data communication is the process of using computing and communication technologies to transfer data from one place to another. The process encounters a lot of problems, which include virus attack, unauthorised access, delay and message distortion. This paper did a comparative analysis on different classical and modern cipher techniques used in data security with emphasis on speed of processing, area of application and strength. Categorising these cipher (encryption algorithm) into classical and modern cipher (symmetric key, asymmetric key and HASH functions) and nothing that these ciphers are unique, a methodology in which one cipher each was selected from the four classes namely; Ceaser, Advanced Encryption Standard (AES), Rivest–Shamir–Adleman (RSA) and SHA-512 was adopted.. The result of the experiment performed using a text size 14,028.8byte and JPEG image file sized 2020byte on an Intel (R) Core (TM) 1GHz, 64-bit Operating System, x64- based processor system showed that Ceaser, encrypted the text in 390 ms and could not encrypt the image, AES in 450 ms and the image in 44ms, RSA in 1100ms and the image in 79ms and SHA-512 in 1000ms and the image in 70ms. In conclusion, Ceaser cipher could only encrypt text document while AES, RSA and SHA-512 can encrypt both text and image documents. Ceaser is the fastest and less secure while RSA is the slowest and most secure

INTRODUCTION

Data communication according to Christopher (2016) is the transmission of digital messages to devices external to the message source. It enables the movement of digital data between two or more nodes, regardless of geographical location, technological medium or data contents. There have been several reported cases of documents which have been venerably tampered with by un-authorized persons and organizations because the documents / data were either not secured or the right security is not employed. As a result, losses were incurred by the affected individuals and organizations from which they may never recover, (Maureen & José, 2016).

In addition, other problems encountered in data communication are: virus attack, delay and message distortion. The major concern of this paper is on unauthorised access. There are various ways of securing data which include: hiding the content of a communication (encryption), hiding the parties to a communication – preventing identification, promoting anonymity and covert communication and hiding the fact that a communication takes place using data erasure and data masking. This paper is concerned with encryption. Encryption is the process of transforming information (referred to as plaintext) using an algorithm (called a cipher) to make it unreadable to anyone except those possessing special knowledge of the encrypting factor known as a key. The result of the process is cipher text.

The main tool used for encrypting messages is a cipher. A cipher is a well-defined procedure that states how to "hide" each character from the original text in the encrypted text and how the recipient should decrypt the cipher text to read the original message. Cryptography is the study of encryption. To accomplish encryption task, the original text, called plaintext, is "translated" into an encrypted version, called cipher text, which is sent to the intended recipient. The recipient decrypts the text to obtain the original message. Though several theories and concepts exist, each

varies with the amount of security it offers to the network channel. An important element which determines the type of cryptography is key distribution" (Uma *et al*, 2017).

Cryptanalysis can be described as the process of attempting to recover the plaintext and/or key from a cipher text. The method of cryptanalysis are: Brute-Force Attack, Exhaustive Search, Frequency Analysis Method, Genetic Algorithm Method, Simulated Annealing, Tabu search, Particle Swarm Optimization, Relaxation Algorithm and Index of Coincidence. There are lots of encryption algorithms and users do not usually know which algorithm is best at each situation, this paper highlights the factors that determine the choice of encryption algorithm from the pool of trusted ones and highlights and discuss the strengths and weaknesses of the chosen encryption algorithms. The knowledge of the strength and weaknesses of encryption algorithms also helps the users to make a choice of suitable algorithm.

Cipher is a mapping algorithm that is applied to a fixed number of characters at a time with intent of concealing the contents of the message. There are classical and modern ciphers (Figure 1).

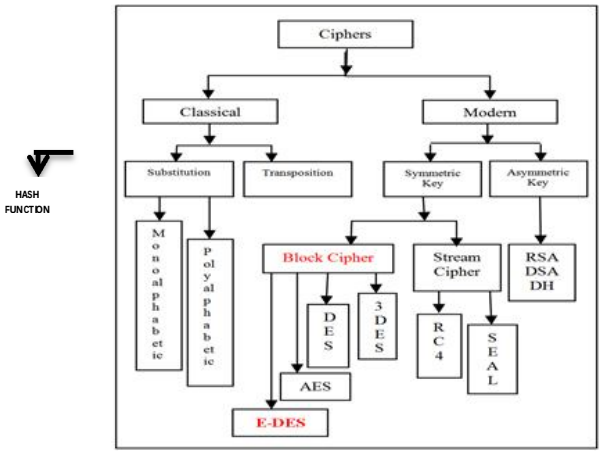


Figure 1: Classification of Encryption Algorithm (*Information Security and Computer Fraud, 2015)*

Classical Ciphers are earlier form of cipher while modern ciphers are resent ciphers whose algorithms have been improved to accommodate the flaws in classical ciphers. Classical ciphers can as well be classified as Substitution Cipher and Transposition Cipher. A substitution cipher is a method of encoding by which units of plaintext are replaced with cipher text, according to a fixed system; the "units" may be single letters (the most common), pairs of letters, triplets of letters, mixtures of the above, and so forth. The receiver deciphers the text by performing the inverse substitution. Substitution ciphers are the simplest ciphers used in cryptography. The most obvious substitution cipher is the Caesar cipher, which was in fact used by Julius Caesar to communicate with his army. The Caesar cipher is an example of the simplest class of substitution ciphers, the mono-alphabetic substitution ciphers. Preeti & Praveen (2016) states that sequence in which units of the plaintext appear is retained in the cipher text, but the units themselves are modified. A transposition cipher, also called a permutation cipher, is one for which applying *E* to plaintext produces cipher text with the same symbols as the plaintext, but rearranged in different positions. Classical ciphers can be broken because of the simplicity in their operation.

Modern Ciphers can be classified into **Symmetric (**or single-key or secret key or conventional encryption), **Asymmetric** (or two keys or private key encryption) and **Hash functions** (no key required). Modern ciphers use modern technology and more complex procedures to encrypt data ranging from exponentiation to factorization. Symmetric Key Ciphers: Both the sender and receiver use the same key to encrypt and decrypt the cipher text. Asymmetric Key Ciphers: It uses two keys. The sender uses the public key of the receiver to encrypt the text and the receiver uses a private key, which is only known to receiver, to decrypt the cipher text to plain text. Hash

functions, also called message digests and one-way encryption, are algorithms that, in some sense, use no key. The benefit of using ciphers is synonymous to the benefit derived from secure data communication, which are confidentiality, integrity, non-repudiation and authentication.

METHODOLOGY

The sample of this investigation consists of four ciphers covering the four classes of cipher. These were selected for experimental purpose based on the grounds of comparison established. They are Ceaser Cipher (substitution Cipher), Advanced Encryption Standard (AES) (Symmetric Key), Rivest–Shamir–Adleman (RSA) (Asymmetric Key) and SHA-512 (HASH Function). These ciphers were chosen because they are the most popular, commonly used and most secured in their group. Grembowski *et al.* (2016) states that Hardware implementations of SHA-384 and SHA-512 have exactly the same performance, so only one of them needs to be implemented for the purpose of comparative analysis. They are also similar in concept to majority of others in their group, experimenting them can give a broad view about others too.

Table1: Population, Sample size and Sampled Ciphers

S/N	CIPHER CLASS	POPULATION	SAMPLE SIZE	SAMPLED CIPHERS
1	Substitution Cipher	17	1	Caesar cipher
2	Symmetric Ciphers	130	1	AES
3	Asymmetric Ciphers	42	1	RSA
4	Hash Function Ciphers	34	1	SHA-512
	Total	**223**	**4**	

Data Source

The data source is from the user. The user either generates the data on the spot or copy from existing data. Open source software of the ciphers were downloaded online and used for the experiment. Information about the software is stated in the implementation procedure of each.

Encrypting and Decrypting Data with Ceaser Cipher

Implementation Procedure

User Interface- Qt
Programming Language- C++
Source- http://pdfsu.com/lib.php?q=read/sue-16/norma-jean-2006-redeemer&ref=raphael.chen.do

Run the Ceaser cipher program from the installation file. The user will be required to type in Shift value, shift represent the number of shift the software will make to replace the alphabet. For example, with a left shift of 3, D would be replaced by A, E would become B, and so on. Then at the plain text box, type in or copy the text to be encrypted, immediately, the counterpart cipher text will appear in the cipher text box. For Decrypting, type in the same shift value that was used in encrypting the text, the copy the text to be decrypted in the cipher text box, the plain text (Decrypted text) will automatically appear at the plain text box.

Algorithms

Algorithm for Encrypting
Start

1. Type in shift value
2. Type in or copy the text to be encrypted (plain text) in plain text box
3. Encrypted text (Cipher text) appears in Cipher text box

Stop

Algorithm for Decrypting
Start
1. Type in the shift value used for encryption

2. Type in or copy the text to be decrypted (cipher text) in cipher text box
3. Decrypted text (Plain text) appears in Plain text box

Stop

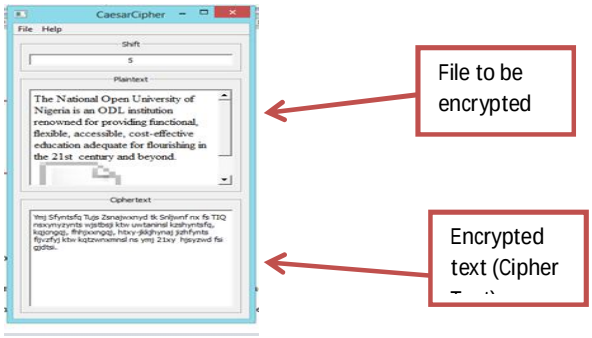


Figure 3 Sample Implementation Snapshot of Ceaser Cipher

Note that this cipher is good for text only, from figure 3, we could observe that Ceaser cipher did not encrypt the numeric number (21).

Encrypting and Decrypting Data with AES

Implementation Procedure

Source- https://download.cnet.com/AES-256-bit/3000-2092_4-10544070.html

Publisher- Telstar

Unzip the AES folder and run the application file, Run the AES program from the installation file. The user will be required to type in the password i.e. the key. Then type in or copy the text to be encrypted in the textbox and click Encrypt, immediately, the information in the textbox will be changed to cipher text.

For Decrypting, type in the same Key that was used in encrypting the text, then copy the text to be decrypted in the text box and click on decrypt button, the plain text (Decrypted text) will automatically appear in the text box.

Algorithms

Algorithm for Encrypting

Start

1. Type in password (key)
2. Type in or copy the text to be encrypted (plain text) into the text box
3. Click Encrypt button
4. Encrypted text (Cipher text) appears in text box

Stop

Algorithm for Decrypting

Start

1. Type in password (key) used in encrypting the text
2. Type in or copy the text to be decrypted (cipher text) in text box
3. Click Decrypt button
4. Decrypted text (Plain text) appears in Plain text box

Stop

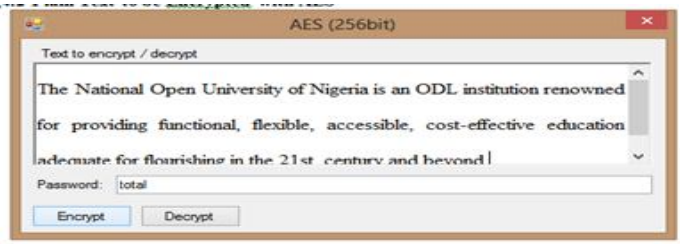


Figure 4: Files to be Encrypted with AES

qIZgXACcM6i+McHo66W42w==

Figure 5: Output of Encrypting with AES

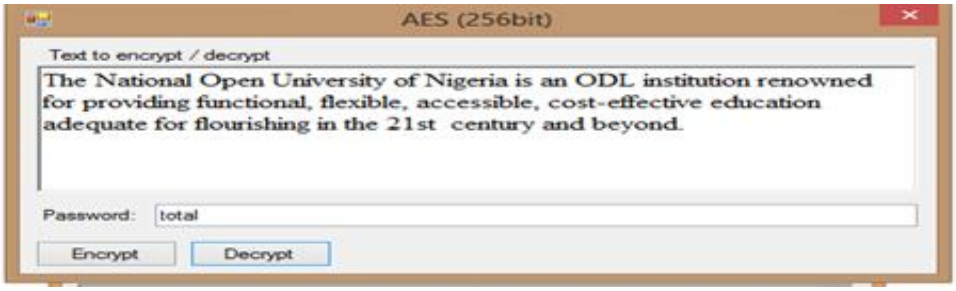


Figure 6: Decrypting Text with AES

Encrypting and Decrypting Data with RSA
Implementation Procedure
Source- https://download.cnet.com/Solid-RSA-Encryption/3055-2092_4-75805984.html?tag=pdl-redir

Unzip the folder contain the software and run the application file in the folder. Follow the instruction until installation is complete. An interface shows. If you have a specific key (Public key or password) to use click load key button and it will automatically generate the private key from the public key as discussed in item 3.4.5 alternatively, you can also allow the software to create public/private key pair for you.

Algorithm
Algorithm for Encrypting
Note: Ensure that the file to be encrypted is not open
Start
1. Select minimum Key Size in bit from the dropdown menu.
2. Click on Create Public/Private Key Pair button.
3. Click on Encrypt file button and navigate to the file to be encrypted.
4. Select location to save the encrypted file and the type the name for the encrypted file.
5. Wait for the software to finish encryption, watching the progress at the top.
6. Navigate to the location you saved the encrypted file and open the file with either word pad or note pad.
Stop

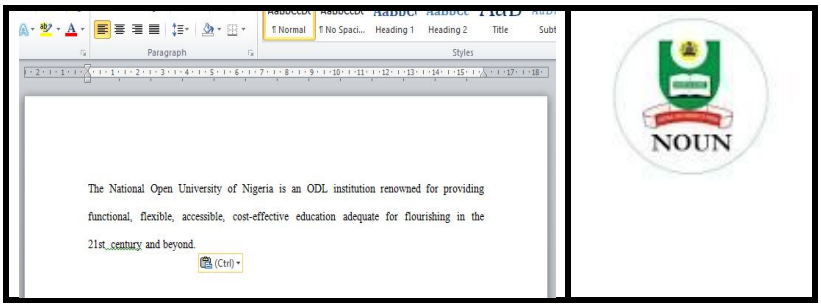


Figure 7 Files to be encrypted

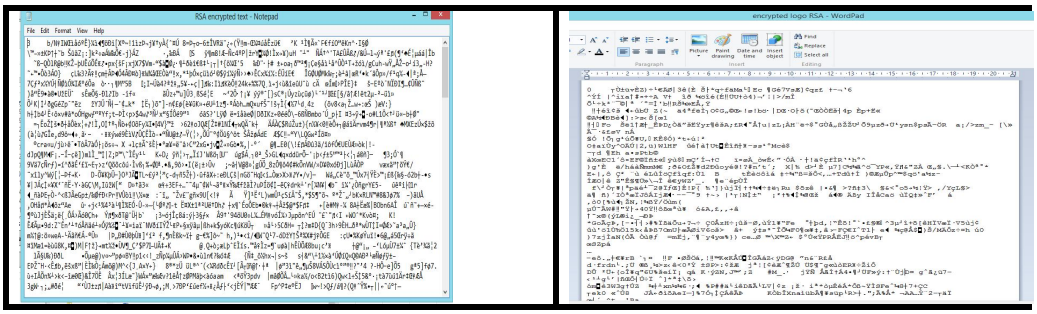


Figure 8 Output of Encrypting with RSA

Algorithm for Decryption

Start

1. Click on Decrypt file button and navigate to the file to be decrypted
2. Select location to save the decrypted file and the type the name for the decrypted file.
3. Wait for the software to finish encryption, watching the progress at the top.

Stop

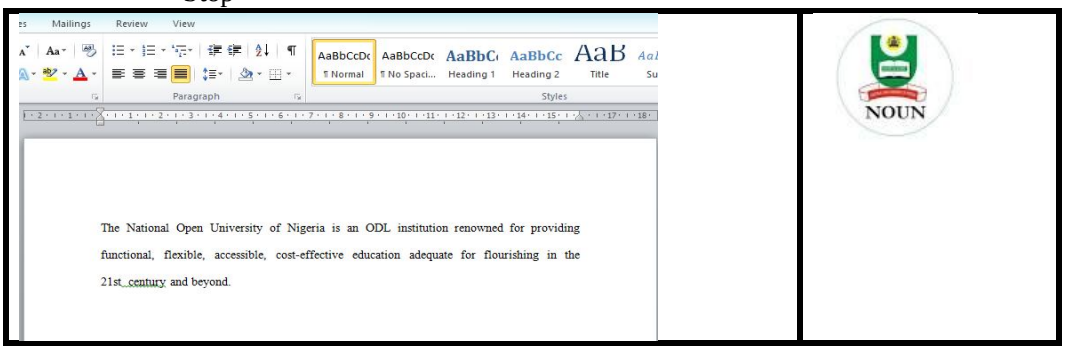


Figure 9 Output of Decryption with RSA

Encryption and Decryption of Data with HASH Functions

Implementation Procedure

Source- https://download.cnet.com/MD5-SHA-Checksum-Utility/3001-2092_4-10911445.html

Run the application file and the software will load

Encryption/ Generating Hash

Note: Ensure that the file to be encrypted is not open

Start

1. Select the kind of Hash value you want whether, MD5, SHA-1, SHA-256 or SHA512. Note: You can select only one or all or more they one.
2. Click on Browse button to the navigate to the file to be encrypted
3. The hash value will be generated automatically in the space in front of each HASH type.

Stop

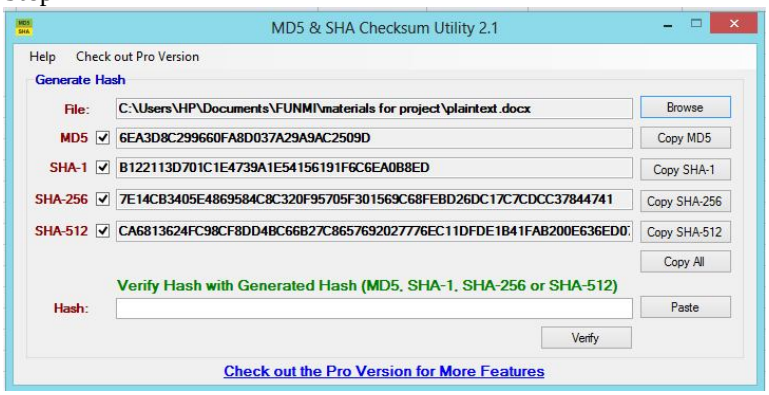


Figure 10 Generating HASH Value

Verify HASH Generated (Checksum)

Start

1. Click on the Copy button in front HASH type you need to verify
2. Click on Paste
3. Click on verify
4. Receive message

Stop

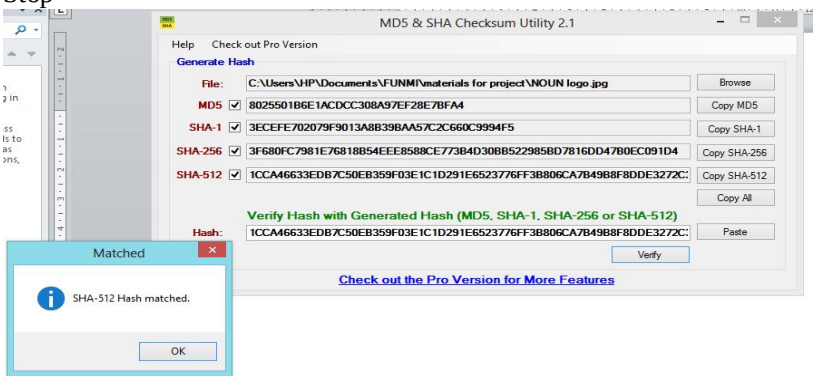


Fig 11 Verifying Generated HASH Value

The results from the encryption of a 14,028.8byte, 27 character text file using the four encryption ciphers on an Intel core 2.8GHz is given in the tables 2, 3 and 4.

DISCUSSION OF RESULTS

Ceaser Cipher, AES, RSA and SHA-512 were the algorithms considered. The ciphers were discussed based on the time taken to encrypt and decrypt, resources to encrypt, area of application and strengths and weaknesses using the information on the speed of processing, type of file they process and their key type.

- **Time Taken to Encrypt and Decrypt (Speed of Processing)**

The time taken to encrypt and decrypt is dependent on the configuration of the system used. The observation from the study showed that the document size 14,028.8byte was encrypted on On Intel (R) Core(TM) 1GHz. Ceaser Cipher encrypted the document in 390 ms, AES in 450 ms, RSA in 1100ms and SHA-512 in 1000ms. Based on the results of the experiment, it was found that Ceaser Cipher is the fastest and RSA is the slowest. It is also discovered that the file size of the encrypted file was larger than the original file after encryption in terms of no of characters and words.

Table 2: Nature of the cipher text generated after encryption

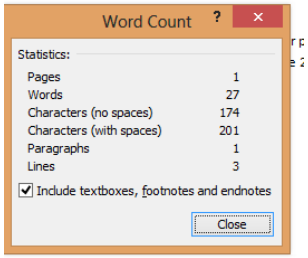
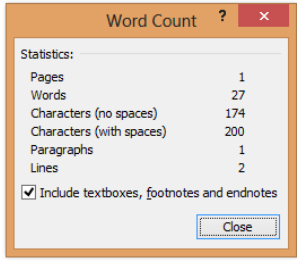
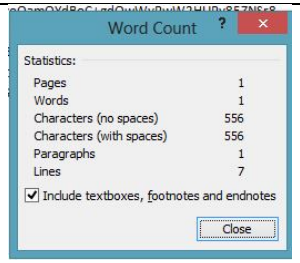
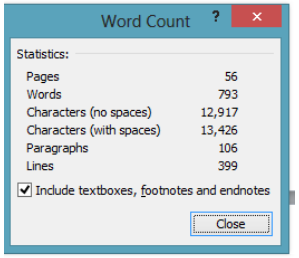
ORIGINAL TEXT	CEASER CIPHER	AES	RSA
			

Table 3: Comparative Analysis of the Four Experimented Ciphers Using Alphanumeric Text File

Plain Text	**Encrypted File Generated**	**Encrypted file characteristics (No of Words)**	**Speed of Processing**	**Status of Security**
The National Open University of Nigeria is an ODL institution renowned for providing functional, flexible, accessible, cost-effective education adequate for flourishing in the 21st century and beyond. Word count: 27	**Ceaser Cipher**			
	Ymj Sfyntsfq Tujs Zsnajwxnyd tk Snljwnf nx fs TIQ nsxynyzynts wjstbsji ktw uwtaninsl kzshyntsfq, kqjcngqj, fhhjxxngqj, htxy-jkkjhynaj jizhfynts fijvzfyj ktw kqtzwnxmnsl ns ymj 21xy hjsyzwd fsi gjdtsi.	27	390 µs	Less secure
	AES- Advanced Encryption Algorithm			
	KUJPInXlqZLXv31+94v758iP3UiEwt3JeV5vuMMpMOi9E0aPcPiLqHZldwF4e1dRumWyG2BgZ/WjKcBBAxmC1ZJiMbif/aUShl/TlVxOr0sRx8TRcq/ytgFjQfVZ70tk4x9T0X1vgKw6vjb7BYiqk7Tvl+gl4MKZLiZHNIBeCqkbrHIArVKW/0b1CGXRHZGLtT0EQ1KmR302fdsMRyoOamOYdBoC+gdOwWvRwW2HUPy85ZNSr8vQNEnZaI8kvXPlVrO7M0iC82ugPyn+4tuOF1y821bFhs3UNjFy/9O58yVI3U9g8Qv9U79aVQ+kPBnApcp6YBS5fk8Ers+nH522F8SZr8Cg9UDqnj7zi6mconYNLkJYYB/Slsx6digJ4s9w/kH7dmU1dq6l0T3CbiTBNqdiwRvpnZgG1OfXlSA9Pf3dbSVoQXLvYmYmS5EyHT5mXP/1Aq4WQTSOYuKiwdpcJC3NBsx6js8MbogjFbWoF2cZevLLLbj1C9EqsALDBiJGPgQ/JF7ncM4IkUJtJVKYS+1X+ZYyT3ih8BkWDQYbAPI=	1	450 µs	Very Secure
	RSA Rivest, Shamir, and Adleman			
	The text is too large to fit into the table. See Appendix1	793	1100 µs	Very secure
	HASH FUNCTION (SHA-512)			
	Generated a Hash Value-07B6C219C6F482FDA0230CF8B4C10289FD328051BFB5F6547B4DD6CEC5F029E1A111CBD7598E521A621ABE42D8C89C40CE96BE23C49A325A577C06C2691B321D		1000 µs	Very secure

Table 4: Comparative Analysis of the Four Experimented Ciphers Using Image File

Image	**Encrypted File Generated**	**Encrypted file size**	**Speed of Processing**
 File Size: 2020bytes	**Ceaser Cipher**		
	Ceaser Cipher could not encrypt the image file		
	AES- Advanced Encryption Algorithm		
	qIZgXACcM6i+McHo66W42w==	24 bytes	44 µs
	RSA Rivest, Shamir, and Adleman		
	The text is too large to fit into the table. See Appendix 2	2163 bytes	79 µs
	HASH FUNCTION –SHA-512		
	Generated a Hash Value- 1CCA46633EDB7C50EB359F03E1C1D291E6523776FF3B806CA7B49B8F8D DE3272 C3B2301AAA911D4337C173F645A025042DF475500828B82D0F3408B9C8F4C0E0		70 µs

Table 5 Further Analysis of the Four Experimented Ciphers

S/N	**Encryption Techniques**	**Developed by**	**Key Size**	**Speed of Processing 14,028.8byte text file**	**Limit of File Type it can process**	**Strength**	**Limitations**	**Comment**
1	Caesar cipher	Julius Caesar in 19 century	Fixed number. 25 possible keys	390 µs	Processed text only	One of the easiest methodologies used in cryptography. Simple substitution with alphabet. It permits multiple encryptions and decryption i.e. Encrypting cipher text and vice versa either with same or different shift key. This also enhance the effectiveness	Cracked by brute force attack because of 25 possible selections of key. Effective for text only	The security is not very strong but could be more secure by using the multiple encryption feature
2	Advanced Encryption Algorithm (AES)	Vincent Rijmen and Joan Daemen	128,192,256 bits	450 µs	Processed text/Image, image not well decrypted.	It uses higher length key sizes such as 128, 192 and 256 bits for encryption. Hence it makes AES algorithm more robust against hacking. It takes a lot of time to Bruce force attack about 3×10^{51} years. Multiple encryption and decryption possible.	It uses too simple algebraic structure. Every block is always encrypted in the same way. Hard to implement with software.	Secure if the key is kept or transmitted securely
3	RSA	Rivest, Shamir, and Adleman	>1024bits	1100 µs	Processed text and JPEG file	It is highly secure because it is difficult to produce the private key from the public key and modulus	The process of cryptography is quite slow	Secure. It supports multiple encryptions but not multiple decryptions. Not advisable to do second order encryption.
4	Secure Hash Algorithm (SHA) 512		Encrypted file cannot be decrypted	1000 µs	Text and image	Good for password hashing. It also ensures integrity of data.	Cannot recover encrypted file	Secure

Nadeem (2015) tested 20, 527 byte file on two different hardware platforms P-II 266 MHz and P-4 2.4 GHz to compare their performance. On P-II 266 MHz, AES processed a 20,527 byte file in 390 milliseconds and on the latter system, the same file was processed in 40 milliseconds. This showed that the speed of the processor of the hardware used also affects the speed of the encryption standard.

- **Resources Expended in Encrypting**

From the experiment, the original text consists of 1 page, 27 words, 174 character excluding the space, 1 paragraph and 3 lines. The result of the encrypted text using Ceaser cipher gave same result. However, the result differs using AES. AES produced a result with 1 page, 1 word (it removed all the space regarding them as characters to encrypt), 556 characters, 1 paragraph and 1 line. RSA produced a result of 56 pages, 793 words, 12,917 characters, 106 paragraphs and 399 lines. This shows that it is only Ceaser cipher that retains the characteristics of the text, in others, the number of text greatly increased.

- **Nature of the cipher text Generated**

It was observed that all the ciphers, namely AES, RSA and SHA-512 processed both text and JPEG files with the exception of Ceaser Cipher, this means that Ceaser Cipher can encrypt only text, it was discovered that even the number (21) within the text was not encrypted. When the need of the user is to encrypt text file only with a little processing time and resources, the user may choose Ceaser cipher. However if the text contains numbers and the number is expected to be encrypted, the user may opt for AES or RSA. AES is more applicable if there is a means for both the sender and receiver to securely communicate the key. The means could be a secure server that will authenticate the users before given access to the key. However if there is no such reliable means, RSA is preferable. If the text to be secured is a password, SHA-512 being an HASH function is the best. (Grembowski *et al*., 2016). The system does not keep an encrypted copy of the password but an HASH value of it so that when transmitted the receiver can verify whether the message sent is what is received and has not in any way been tampered with in the process of transmission. A practical example is ATM card password or Email account password, if forgotten; it cannot be recovered but can only be reset to the default one so the user can change it to a desirable one.

- **Strength and Weakness of the Ciphers (Level of Security)**

Ceaser Cipher was found to have a fixed key size ranging from 1-25 only because it is based on shift structure where each alphabet in the text to be encrypted is replaced with another based on the shift value provided and have only 26 alphabet letters in English. AES key size can be as large as 128bit, 192bit and 256 bit which make it stronger than Ceaser cipher because it becomes more difficult to discover the encryption key. RSA key can be as large as a value greater than 1024bit. SHA-512 does not use any key. Key size determines the strength of an algorithm. According to Preeti & Praveen (2016) the key size determines the strength of an algorithm. Another factor to be used to measure the security of an algorithm is the time it takes to Bruce force attack. Chadi & Pierre (2015) in their research work state that it takes a system 10^{19} years to Bruce force AES, this shows that practically, this may not be possible. By the time the system finishes the Bruce force attack and gain access to the information, the information may likely not be viable any more. Uma, *et al* (2017) noted that RSA is highly secure because attackers find it difficult to produce the public key and the modulus. AES is reliable but not as RSA, RSA is more secure because of its use of two keys. The intruder cannot easily deduce the private key from the public key, hence even if the public key used in encrypting is known, the intruder can still not easily break in, therefore RSA may be chosen for a better security but with a greater processing time and resources (Uma *et al* 2017). However, Decryption in java cryptography (2017) states that the most effective way to break AES is Bruce-force but RSA is not hard to break. Instead of Bruce forcing the keys in RSA, factorizing modulus into prime and deriving the keys is better, though uneasy but being a mathematical problem, it is believed that it can be solved without

taking as much time as AES. Though the two seems to be conflict but this is re-emphasizing that both are strong, both Bruce forcing for 10^{19} years and factorizing key from modulus is uneasy. This shows that both are very strong and reliable, it just depends on the users' need. SHA-512 is said to be more secure than AES. According Grembowski *et al* (2016), considered SHA-512 as the most secure algorithm of four investigated algorithm (SHA-512, SHA-1, AES and 3-DES).

CONCLUSION

The use of ciphers to encrypt data provides secure communication, integrity, authentication and confidentiality over an internet. Ceaser Cipher and AES permit multiple order encryption i.e. the cipher text generated can be re-encrypted, however, the user must endeavour to use same key used in encrypting to decrypt. The encrypted file is usually larger than the original file except in the case of Ceaser cipher. Ceaser Cipher is the fastest and less secure while RSA is the slowest and most secure. Encryption with AES is similar to that of Ceaser cipher though AES is a symmetric key encryption. It is also fast, very safe and reliable. SHA 512, a HASH function, whose focus is data integrity i.e. ensuring that the data being received is the same as the data sent by the sender and has not been altered in the process of transmission, only generate a HASH value to verify the integrity of the data received.

The algorithm for encryption can be selected based on the type of data being communicated and type of channel through which data is being communicated. All the three types of data encryption are useful, depending on the need of the user. When the utmost need of the user is confidentiality, Symmetric Key and Asymmetric key Ciphers are more useful. If the need is Integrity, Hash Function is the most appropriate. If the message to be encrypted contains alphabets, numbers and Images, RSA is more appropriate. If it is alphanumeric, AES is preferable. However, Ceaser Cipher processes only alphabets.

REFERENCES

AES Encryption Application Downloaded from https://download.cnet.com/AES-256-bit/3001-2092_4-10544070.html

Ceaser Cipher Encryption Application Downloaded from http://pdfsu.com/lib.php?q=read/sue-16/norma-jean-2006-redeemer&ref=raphael.chen.do

Chadi, R. & Pierre, E. (2015). "Comparative Analysis of Block Cipher-Based Encryption Algorithms: *A survey on information security and computer fraud*, 3(1) 1-7. doi: 10.12691/iscf-3-1-1.

Christopher, E. (2016) *Data Communication Basics*, [E-book]. Retrieved from https://www.camiresearch.com/Data_Com_Basics/data_com_tutorial.html

Grembowski, T., Lien, R., Gaj, K., Nguyen, N., Bellows, P., Flidr, J., Lehman, T., Schott, B. (2016) *Comparative analysis of the hardware implementations of hash functionsSHA-1 and SHA-512*. Retrieved from www.veracode.com/blog/research/encryption-and-decryption-java-cryptography

HASH Encryption Application Downloaded from https://download.cnet.com/MD5-SHA-Checksum-Utility/3001-2092_4-10911445.html

Information security and computer fraud (2015). *Science and Education Publishing. 3 (1), 1-7.*

Maureen, F. & José C. (2016) Unauthorized access: A growing problem with a straightforward fix [blog post]. Retrieved from https://www.lexology.com/library/detail.aspx?g=208533ed-ecbe-4410-9fd9-f159f135884c

Uma, K., Karthik, G. & Vishnu, R. (2017).*A comparative analysis of symmetric and asymmetric key cryptography.* Journal of Chemical and Pharmaceutical Sciences 324 JCPS 10 (1).

Preeti, P. & Praveen, K. (2016).*A comparative study of classical substitution ciphers.* International Journal of Computer Applications (0975 – 8887) 145 (10).

RSA Encryption Application Downloaded from https://download.cnet.com/Solid-RSA-Encryption/3055-2092_4-75805984.html?tag=pdl-redir