Security Assurance Framework for Intelligible Information Using a Customized Base64 Encryption Algorithm

[1]Logunleko K. B, [2]Logunleko A. M, [3]Akinwunmi O. O. & [4]Lawal O. O
[1,3]Department of Computer Science and Statistics, DS Adegbenro ICT Polytechnic, Itori, Ogun state, Nigeria.
[2]Department of Computer Science, Gateway ICT Polytechnic, Saapade, Ogun State, Nigeria.
[4]Department of Computer Engineering, Moshood Abiola Polytechnic, Abeokuta, Ogun State.
E-mails: [1]logunleko.kolawole@dsadegbenropoly.edu.ng, [2]adeyinkalogunleko@gmail.com
[3]akinwunmi.oluwafemi@dsadegbenropoly.edu.ng, [4]lawal.olanrewaju@mapoly.edu.ng

ABSTRACT

Data security is one of the most pressing issues facing by the individuals, groups and organizations today. Unauthorized access to confidential information is extremely disastrous. However, integrity and confidentiality are key aspect of an information system. The information sent is expected to be well received only by those who have the right to such information. So, such information will be rendered useless if during the time of transmission intercepted by an unauthorized person. Thus, security is imperative for an individual, groups and organization to maintain the integrity of the sensitive data and information in their domain at any given point. In this paper, a security assurance framework of a customized Base64 algorithm for both encryption and decryption were presented.

Keywords: Information, B64, Security, Encryption, Decryption, Cipher

iSTEAMS Proceedings Reference Format
Logunleko K. B, Logunleko A. M, Akinwunmi O. O. & Lawal O. O. (2019): Security Assurance Framework for Intelligible Information Using a Customized Base64 Encryption AlgorithmProceedings of the 17th iSTEAMS Multidisciplinary Research Nexus Conference, D.S. Adegbenro ICT Polytechnic, Itori-Ewekoro, Ogun State, Nigeria, 21st – 23rd July, 2019.Pp 85-93 . www.isteams.net - DOI Affix - https://doi.org/ 10.22624/AIMS/iSTEAMS-2019/V17N2P10

1. INTRODUCTION

Data security is the act of protecting digital data, such as those in a database from destructive forces and from the unwanted actions of unauthorized users such as a cyber attack or a data breach. Data security is an essential aspect of information technology organizations of every size and type. Therefore, data security is very crucial in maintaining the confidentiality of information especially containing sensitive information that should only be known by certain parties [7]. Base64 is a model used to encrypt the plaintext and decrypt what the plaintext turned to without the use of key mechanism. Base64 has a table pattern in the flow of its algorithm such that if the table pattern is known, one can easily decrypt such message [1].

This study describes the process of encryption and decryption using a reviewed process flow of Base64 algorithm. According to the study, the algorithm could not standalone because of the security threat thus it can either be hybridized with other cryptographic techniques or enhanced in order to strengthen the security capability of the algorithm.

2. LITERATURE REVIEW

A. A conceptual review of Base64 Algorithm

Base64 is an algorithm that uses a concept of modern encryption algorithms [6]. It is a block cipher algorithm that operates on a bit. Base64 algorithm encodes binary data and translates it into a representation of the base64. The term comes from the Base64 MIME (Multipurpose Internet Mail Extension) encoding specific content. This base64 algorithm is often used when there is a need to encode binary data that needs to be stored and transferred through media designed in form of textual data. This is to make sure that the data strictly remains intact without modification during shipping [2]. Therefore, Base64 can be used in different applications such as email through MIME and storage of complex data in XML. Hence, Base64 needs to be learned because the transformation of base64 is widely used on the Internet as a medium to transmit data format. Due to the result of the transformation of base64 to plain text, and then this value will be much more easily shipped, compared to the form of binary data format.

Common Use of Base64

The following are some common applications that make use of base64 algorithm:

a) Basic authentication to web sites: When this type of authentication is used, the username and the password are separated by a colon, concatenated and the results encoded using base64 [11].
b) Transfer of binary data via mediums such an email, as a replacement for uuencode [12].
c) Evasion of basic anti-spamming tools [10].
d) Encoding characters strings in LDAP LDIF or files [13].
e) Embedding binary data in an XML file.
f) Encoding binary files, such as images, within scripts or HTML to avoid depending on external files [16].
g) Communicating encrypted cookie information [15]

B. Review of Related Works

Dodi Siregar et,al., [3] carried out a study titled combination Base64 Algorithm and EOF Technique for Steganography. Steganography consists of a set of methods and techniques to embed the data into another media so that the contents are unreadable to anyone who does not have the authority to read these data. The authors discussed steganography and encoding techniques using base64, which is encoding scheme that converts the same binary data to the form of a series of ASCII code. Also, the EoF technique is used to embed encoding text performed by Base64. The authors further explained that the usage of the two methods together will definitely increase the security level for protecting such data. Hence, the research aimed to secure many types of files in a particular media with a good security and not to damage the stored files and coverage media being used.

Sumartono et, al,. [1] performed a research on Base64 Character Encoding and Decoding Modeling. The model transforms a textual data into cipher text by using Base64 encoding technique and transforms the cipher text back to plaintext. The Base64 algorithm is not often used for encryption like others actually, because of its inadequate security strength. The algorithm transforms characters to limited characters without the use of a key. Since Base64 algorithm does not have key, everyone can decrypt the message by knowing the table pattern unlike others encryption algorithms.

Guwalani et, al., [2] conducted a research on Image File Security using Base-64 Algorithm. The paper mainly focused on embedding the data from one format to another by designing a data conversion application which converts image file to text file and text file to image file. Usually, image loses its resolution after conversion of image is done. The authors proposed a method such that the image remains unchanged in its resolution as well in size.

3. METHODOLOGY

A. The transformation of Base64 is one of the algorithms for encoding and decoding data into ASCI (American Standard Code for Information Interchange) format, which is based on the number 64. The characters generated from Base64 consist of "A-Z", "a-z" and "0-9", and the last two characters are "/" and "+". The Base64 encoding technique is as follow.

B. Pseudo code for Base64 algorithm (B64)
The steps for implementing Base64 algorithm are divided into two; encryption and decryption process

B.I Pseudo code for Base64 Encryption is depicted below:

1. Look for the ASCII code of each text.
2. Convert each ASCII number to 8 bits Binary String
3. Combine the 8 bits to 24 bits.
4. Then, split a 24 bit earlier to 6 bits. It will produce four fractions.
5. Each fragment is converted into a decimal value.
6. Lastly, use the decimal value to choose a character constituent of Base64.

B.2 Pseudo code for Base64 Decryption is listed below:

1. Supply the encrypted text
2. Replace the each character of the text with its position in the base64 lookup table
3. Convert each character of the base64 index above to a 6-bits binary
4. Merge all the newly computed 6-bits Binary String
5. Split the Merged Binary String into each of 8-bits Binary String
6. Convert each 8-bits Binary String into Decimal
7. Get the equivalent character of each of the above decimal from the ASCII Table.

C Data Flow Diagram (DFD) for Base64 Algorithm (B64)

The data flow diagram for implementing Base64 algorithm is divided into two stages. In stage one, encryption was carried out as shown in figure 1 while in stage two, data flow diagram for decryption is shown in figure 2 respectively

C.I. Base64 Encryption

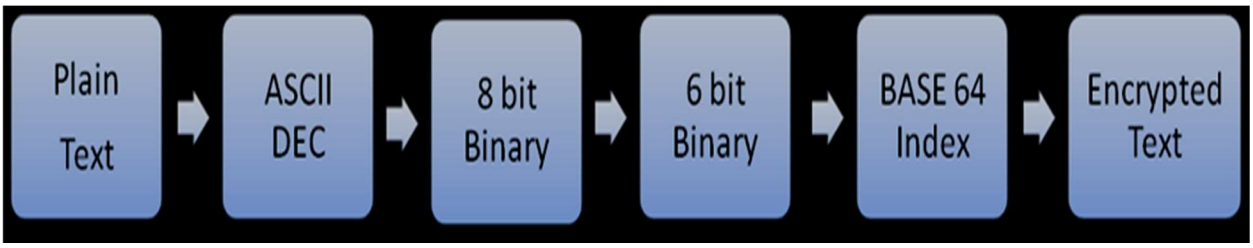

Figure 1

C.II. Base64 Decryption

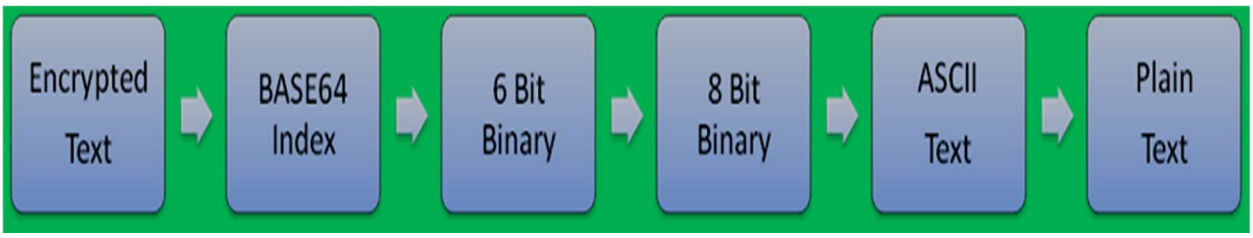

Figure 2

D. The Flow Chart of the Base64 (B64) Algorithm

The flow chart of the B64 algorithm is divided into two; encryption and decryption. Encryption process was shown below in figure 3 while decryption was shown in figure 4 respectively.

D.1. Flow Chart of the Base64 Encryption

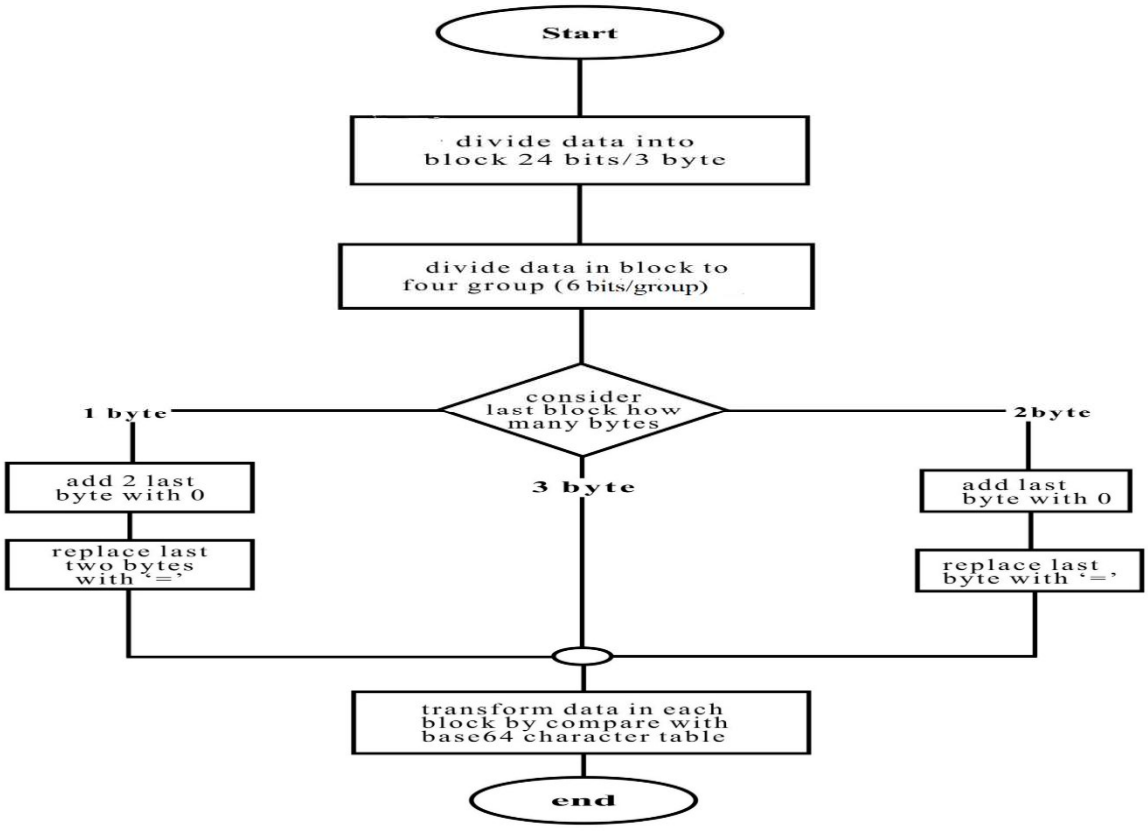

Figure 3

D.II. Flow Chart of the Base64 Decryption

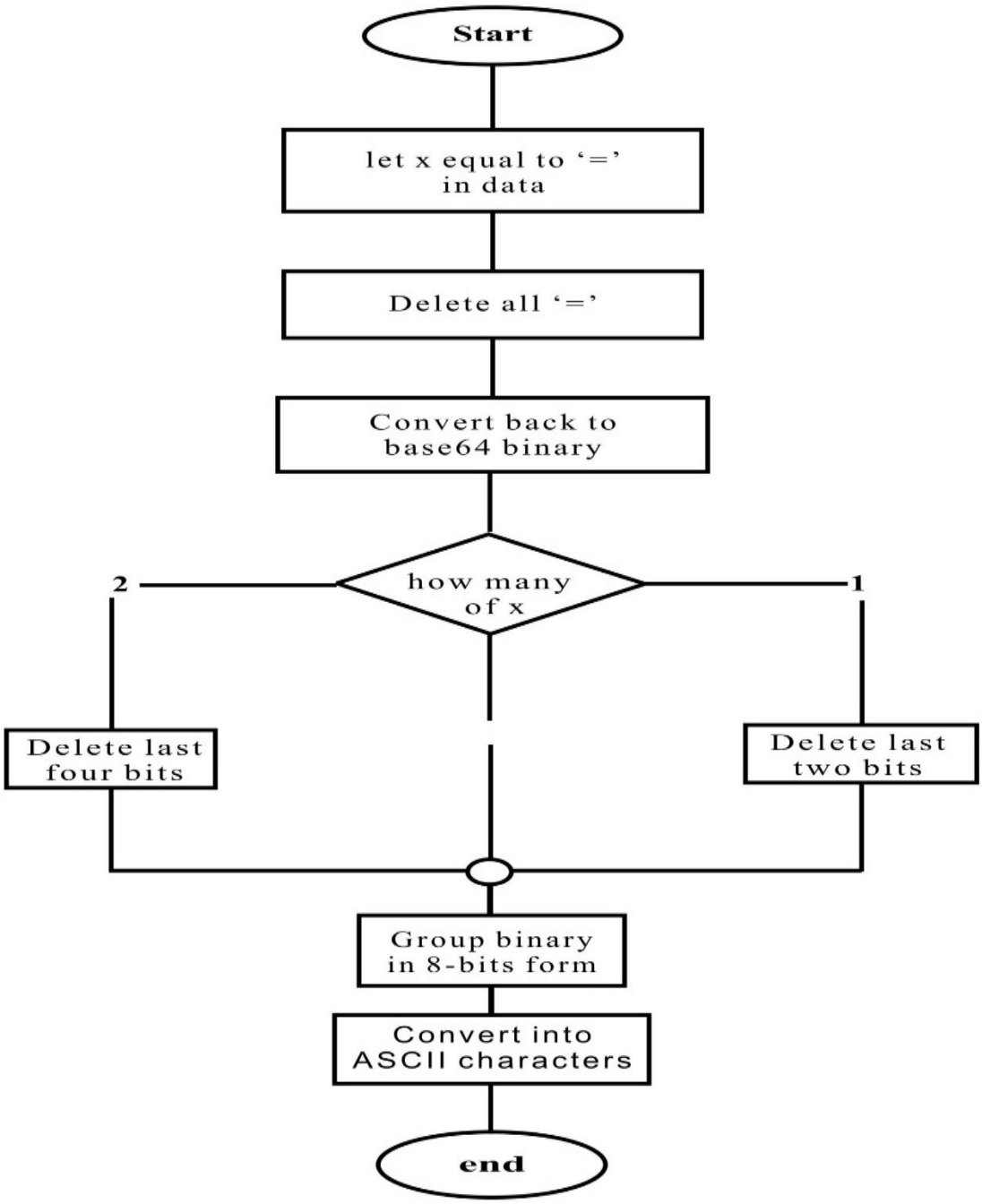

Figure 4

E. The Base64 Algorithm (B64)

The Base64 data flow diagram of each encryption and decryption turns to a separate algorithm for both encryption and decryption. This is as follows:

E.I Base64 Encryption Algorithm

Supply a plain text, s

$s = "a_0a_1a_2...a_n"$

Get ASCII number for each character

$asc = array[]$

$asc_i = ASCII(s_i)$

Convert each ASCII number to 8-bits binary string

$asc_i = Binary(asc_i)$

Merge all Binary Strings accordingly:

$s = "asc_0asc_1asc_2...asc_i"$

Split the merged string to each of 6-bits Binary String:

$split6 = array[]$

$split6_0 = "s_0s_1s_2…s_5"$

$split6_1 = "s_6s_7s_8…s_{11}"$

$split6_2 = "s_{12}s_{13}s_{14}…s_{17}"$

… … …

… … …

$split6_n = "s_{nx6}s_{nx6+1}s_{nx6+2}…s_{nx6+5}"$

convert each 6-bits Binary string to decimal:

$split6_n = Decimal\ (split6_n)$

Return the string found in the position $split6_n$ in base64 look up table:

$Const =$ "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"

$split6_n = Const\ [split6_n]$

If the length of split6 is not a multiple of 4, Add f number of equal to sign (=) to make the length of split6 a multiple of 4:

$Where\ f = 4 – (length\ (split6)(mod\ 4))$

$split6 = split6.add(=)_f$

converted split6 to text is the required result:

$result = Text(split6)$

E.II. The Base64 Decryption Algorithm

Strip of all '=' in the text:

$Text = Text.replace(\ "=","")$

Replace the text with its position in base64 lookup table:

$Const =$ "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"

$Text_i = Const.indexof(\ Text_i\ )$

Convert each of the decimal above to a 6 digit binary

$Text_i = Binary6(\ Text_i\ )$

$Text\_merged = Text_0Text_1Text_2…….Text_i$

Split merged data into each of 8 digit binary string:

$Text_n = "Text\_merged_{8n+0}Text\_merged_{8n+1}…Text\_merged_{8n+7}"$

Convert each of the 8 digit binary string above to decimal:
$Text_n = Decimal(Text_n)$
Lookup the character at the position of the Decimal number above:
$Text_n = ASCII[Text_n]$
Merge all the data in text into one single string.
$Merge \mathrel{+}= Text_n$
The merged string above is the plain text of the encrypted text.

4. RESULTS AND DISCUSSIONS

The developed system is a web based application that uses HTML5 and Java Script language to implement base64 algorithmic model. The developed system encrypts a plain text to a cipher text and the same cipher text was decrypted to a plain text which was shown in figure 5 and 6 respectively.

Encryption Model

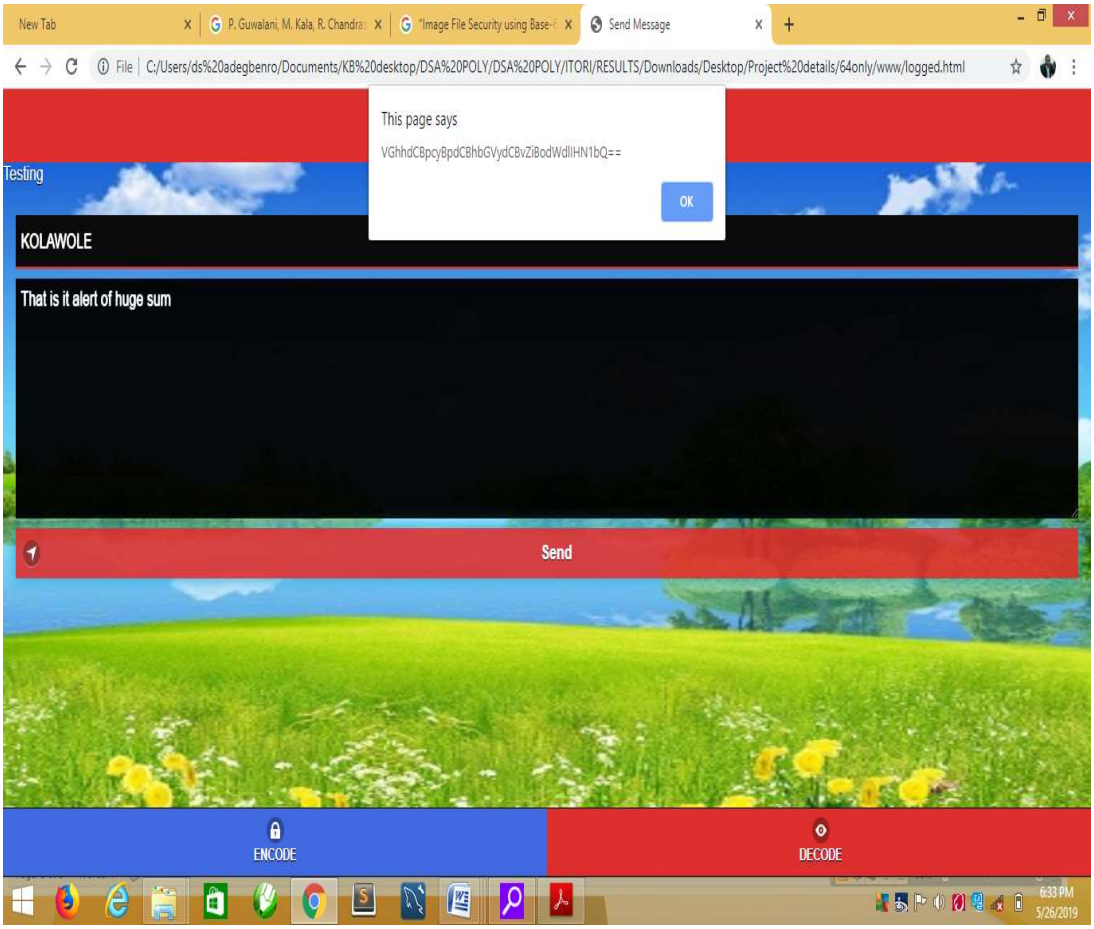

Figure 5

Decryption model

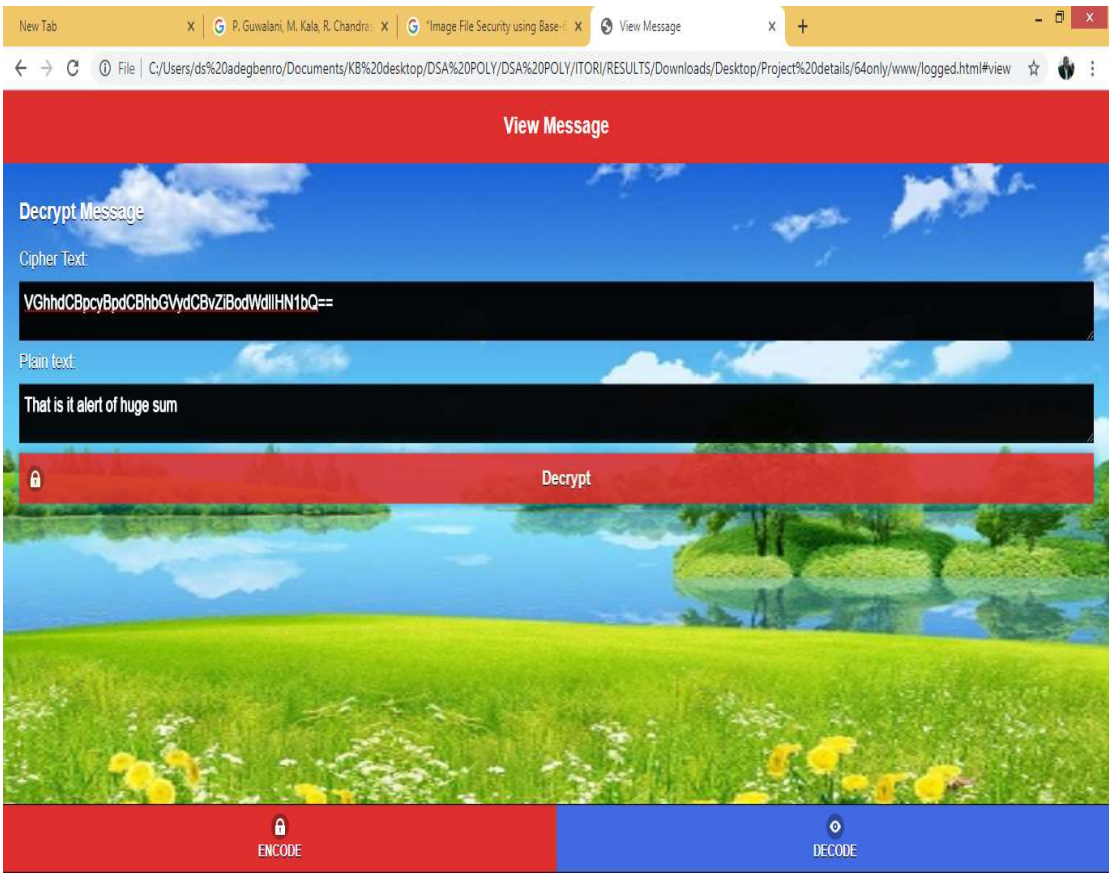

Figure 6

5. CONCLUSION

The study was successfully demonstrated using Base64 algorithm to encrypt and decrypt textual data. It is therefore served as an important information security model for both encryption and decryption process. The study transforms the inputted characters to limited characters as compared to others encryption algorithms without the use of key being introduced in the algorithmic process. Hence, the algorithm could be standardized by using key mechanism like others standardized key-based encryption and decryption algorithms.

REFERENCES

1. Sumartono U , Isnar S, Andysah P, and Arpan (2016), "Base64 Character Encoding and Decoding Modeling" International Journal of Recent Trends in Engineering & Research, Volume 02, Issue 12; [ISSN: 2455-1457], pp. 63-68.
2. Guwalani P, Kala M, Chandrashekar R, Shinde dan J and Mane D (2014), "Image File Security using Base-64 Algorithm," Pooja Guwalani et al, Int.J.Computer Technology & Applications, vol. 5, no. 6, pp. 1892-1895.
3. Dodi Siregar et al (2018), "Combination Base64 Algorithm and EOF Technique for Steganography". International Conference on Information and Communication Technology (IconICT): Journal of Physics: Conf. Series 1007.
4. Poonkuzhali S.M and Therasa M (2015), "Data Hiding Using Visual Cryptography for Secure Transmission,"International Journal of Advanced Research in Computer and Communication Engineering, vol. 4, no. 4, pp. 440-441.
5. Siahaan A. P (2016), "A Three-Layer Visual Hash Function Using Adler-32," International Journal of Computer Science and Software Engineering, vol. 5, no. 7, pp. 142-147.
6. [6] Hariyanto E and Rahim R (2016), "Arnold's Cat Map Algorithm in Digital Image Encryption," Int. J. Sci. Res., vol. 5, no. 10, pp. 1363–1365.
7. Putera A, Siahaan U, and Rahim R (2016), "Dynamic Key Matrix of Hill Cipher Using GeneticAlgorithm," Int. J. Secur. Its Appl., vol. 10, no. 8, pp. 173–180.
8. Shannon C (1998), "Communication Theory of Secrecy Systems", Bell Systems Technical Journal, MD Computing, vol. 15, pp. 57-64.
9. Al-hazaimeh and Obaida M (2014), "A novel encryption scheme for digital image-based on one dimensional logistic map." Computer and Information Science 7, no 4: page 65.
10. Craig. (2007). Filtering base64 encoded spam | Small Dropbear. Small Drop Bear . Retrieved August 16, 2011, from http://enc.com.au/2007/08/filteringbase64-encoded-spam
11. Franks. (2009). RFC2617 - HTTP Authentication. Internet Engineering Task Force. Retrieved August 16, 2011, from tools.ietf.org/html/rfc2117
12. Freed, N. (2006.). Multipurpose Internet Mail Extensions. Internet Engineering Task Force. Retrieved August 16, 2011, from tools.ietf.org/html/rfc2045
13. Good, G. (2010.). The LDAP Data Interchange Format (LDIF) - Technical Specifications. Internet Engineering Task Force. Retrieved August 16, 2011, from www.ietf.org/rfc/rfc2849.txt
14. Perera, H. (2011). History of Steganography. Hareendra's Blog. Retrieved August 16, 2011, from http://hareenlaks.blogspot.com/2011/04/history-ofsteganography.html
15. Prabhakar, A. (2011). the Digital me: Base 64 Encoding. the Digital me. Retrieved August 16, 2011, from http://digitalpbk.blogspot.com/2006/12/base-64-encoding.html
16. Coyier, C. (2010, March 25). DataÂ URIs | CSS-Tricks. CSS-Tricks. Retrieved August 16, 2011, from http://css-tricks.com/5970-data-uris