Combination Base64 Algorithm and EOF Technique for Steganography

Robbi Rahim1*, Heri Nurdiyanto2, Rahmat Hidayat3, Ansari Saleh Ahmar4, Dodi Siregar5, Andysah Putera Utama Siahaan5, Ilham Faisal5, Sayuti Rahman5, Diana Suita6, Ahmad Zamsuri7, Darmawan Napitupulu9, Muhammad Ikhsan Setiawan10 and S Sriadhi11

1School of Computer and Communication Engineering, Universiti Malaysia Perlis, Malaysia
2Department of Informatics, STMIK Dharma Wacana, Metro Lampung, Indonesia
3Department of Information Technology, Politeknik Negeri Padang, Indonesia
4Department of Statistics, Universitas Negeri Makasar, Indonesia
5Department of Informatics Engineering, Universitas Harapan Medan, Indonesia
6Department of Informatics, Universitas Harapan Medan, Medan, Indonesia
7Department of Informatics, Universitas Lancang Kuning, Pekanbaru, Indonesia
8Department of Informatics, Universitas Malikussaleh, Aceh, Indonesia
9Research Center for Quality System and Testing Technology, Indonesian Institute of Sciences, Indonesia
10Department of Civil Engineering, Narotama University, Indonesia
11Department of Electrical Engineering, Universitas Negeri Medan, Indonesia

E-mail: *usurobbi85@zoho.com

Abstract. The steganography process combines mathematics and computer science. 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 main objective of the use of base64 method is to convert any file in order to achieve privacy. This paper discusses a steganography and encoding method using base64, which is a set of encoding schemes that convert 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. As an example, for the mechanisms a file is used to represent the texts, and by using the two methods together will increase the security level for protecting the data, this research aims to secure many types of files in a particular media with a good security and not to damage the stored files and coverage media that used.

1. Introduction

Data[1] is crucial for maintaining the confidentiality of information, especially vital information[2] and can only be known by certain parties only and the information are transmitted through the public network[1], [3], [4], if the data is not secured enough it will be easily tapped by others and the content easily known by users who do not have the authority[1], [5], [6]. One way to secure data is using a steganography technique[7], [8]. Steganography is the art and science to hide secret messages in other
media so that the existence of such secret messages is unknowable, steganography is concealed by covering up or hiding messages[9]–[11], hidden messages can be read or extract from a file by using a certain method in this paper EoF were a technique using for steganography and Base64 a method to convert any file to printable text before hidden to media.

Base64 is a coding schemes that encode binary data and translate into a base64 representation[12], [13], End of File (EOF) technique is a method that works by inserting data at the end of the file[14]. Singh and Supriya[12] explains the encoding and decoding process with the base64 algorithm can be performed for all types of files by converting into printable text, M. Ramalingam[15] conducts research for steganography on video types using 2 different methods for doing steganography and cryptography process with good result, another research was perform by Nurdiyanto[16] by using Pixel Value Differencing a message will be hidden in pixel image but the result from this made a distortion in pixel, it can happen when insert a large file or a lot of message to be hidden. Hidden message in pixel in maybe a good one but it can distort pixel, one of reason why combine Base64 and EoF for hidden information.

2. Methodology
Base64 algorithm[17], [18] and EOF technique are implemented by using Visual Basic.Net 2015 programming language, the experiment process is done step by step for each file to embedded into various media, the first step is to change the binary of each file into the form of text encoding using the Base64 algorithm, the result of text encoding is inserted into the media used as cover by using EOF technique, for base64 procedure encoding and decoding can be seen in the following function:

```vbnet
Function FileToBase64(ByVal xobject As object)
    Using ms As New MemoryStream()
        xobject.Save(ms, format)
        Dim objBytes As Byte() = ms.ToArray()
        Dim base64String As String = Convert.ToBase64String(objBytes)
    End Using
    Return base64String
End Function
```

```vbnet
Function Base64ToFile(ByVal base64Code As String)
    Dim objBytes As Byte() = Convert.FromBase64String(base64Code)
    Dim ms As New MemoryStream(objBytes, 0, objBytes.Length)
    Dim tmpObject As Object = objbytes.FromStream(ms, True)
    Return tmpObject
End Function
```

The above function is a process snippet code to get the base64 value of each object to be embedded, the procedure Base64[12], [17] are like below:

1. Look for the binary code of each object
2. Convert binary number 8 bits to the ASCII code
3. Combine the last 8 bits to 24 bits.
4. Then, broke a 24 bit earlier to 6 bits. It will produce four fractions.
5. Each fragment is converted into a decimal value.
6. Lastly, make value the decimal value to an index to choose a character constituent of base64 and the maximum is 63 or 64 to the index.

The index value of the base64 algorithm can be seen in the table below:

| Index | Value | Index | Value | Index | Value |
|-------|-------|-------|-------|-------|-------|
| 0     | A     | 28    | C     | 56    | 4     |
| 1     | B     | 29    | D     | 57    | 5     |
| 2     | C     | 30    | E     | 58    | 6     |
| 3     | D     | 31    | F     | 59    | 7     |
For example, Base64 algorithm encoding process performed on binary = 0100100101101110
0111010001101011100110011110100011010111010110111101101100 using the above procedure obtained the following encoding results in table 2 below:

| No | Binary 6 Bit | Decimal | Base64 Value | No | Binary 6 Bit | Decimal | Base64 Value |
|----|--------------|---------|--------------|----|--------------|---------|--------------|
| 1  | 010011       | 18      | S            | 10 | 010111       | 23      | X            |
| 2  | 010110       | 22      | W            | 11 | 010001       | 17      | R            |
| 3  | 111001       | 57      | 5            | 12 | 101001       | 41      | p            |
| 4  | 110100       | 52      | 0            | 13 | 011011       | 27      | b            |
| 5  | 011001       | 25      | Z            | 14 | 110110       | 54      | 2            |
| 6  | 010111       | 23      | X            | 15 | 111001       | 57      | 5            |
| 7  | 001001       | 9       | J            | 16 | 100001       | 33      | h            |
| 8  | 101110       | 46      | u            | 17 | 011011       | 27      | b            |
| 9  | 011000       | 24      | Y            | 18 | 000000       | 0       | A            |

From table 2 above obtained the encoding results SW50ZXJuYXRpb25hbA, then the encoding results are embedded into file by using EOF technique with the following functions:

FileOpen(1, filename, OpenMode.Binary)
Seek(1, LOF(1) + 1)
FilePut(1, StartData)
FilePut(1, Data)
FilePut(1, EndData)
FileClose(1)
The above functions make it possible to embed base64 encoding text into any file, so there is no limit how big the file size to be embedded to cover media.

3. Result and Discussion
Experiments with Base64 algorithms and EOF techniques are perform with various types of files to be embeded, table 3 below is the file information in the experiment.

| No | Media to be Secure          | Size in Kb | Cover Media | Size in Kb |
|----|----------------------------|------------|-------------|------------|
| 1  | IC_1.jpg                   | 109        | Lena.bmp    | 117        |
| 2  | IC_2.jpg                   | 115        | Lena.bmp    | 117        |
| 3  | IPS7-1.jpg                 | 831        | Lena.bmp    | 117        |
| 4  | Sertifikat IC Robbi.jpg    | 112        | Lena.bmp    | 117        |
| 5  | WhatsApp Image 2017-07-26 at 20.41.00.png | 299       | Sample.avi  | 539        |
| 6  | IJSRST.doc                 | 571        | Proceeding.pdf | 1326 |
| 7  | IJRTER.doc                 | 753        | Proceeding.pdf | 1326 |
| 8  | IJSRSET.PDF                | 412        | Proceeding.pdf | 1326 |

Table 3 shows the file name and size of the file to be inserted and also the size of media as the cover, the next process is to perform steganography the existing file, and the results as in table 4:

| No | Media to be Secure         | Cover Media | Result Steganography (Kb) |
|----|----------------------------|-------------|---------------------------|
| 1  | IC_1.jpg                   | Lena.bmp    | 262                       |
| 2  | IC_2.jpg                   | Lena.bmp    | 270                       |
| 3  | IPS7-1.jpg                 | Lena.bmp    | 1190                      |
| 4  | Sertifikat IC Robbi.jpg    | Lena.bmp    | 271                       |
| 5  | WhatsApp Image 2017-07-26 at 20.41.00.png | Sample.avi | 937                       |
| 6  | IJSRST.doc                 | Proceeding.pdf | 2167                      |
| 7  | IJRTER.doc                 | Proceeding.pdf | 2273                      |
| 8  | IJSRSET.PDF                | Proceeding.pdf | 1960                      |

Each of file will convert into printable text using base64 method and then insert into a cover media with EOF method, the size of cover media will increase when text are embeded, based on experiment process any file type can be well hide using the EoF method.

4. Conclusion
An experiments for all types can be done as well, before EoF process a file will be convert into printable text by using Base64 method, the steganography process with EoF techniques can perform will with Base64 encoding text. Next improvement a Base64 and EoF method can combine with compression algorithm, so the encoding base64 result compress by an algorithm and then embed to file by using EoF.

References
[1] R. Rahim and A. Ikhwan, “Study of Three Pass Protocol on Data Security,” Int. J. Sci. Res., vol. 5, no. 11, pp. 102–104, Nov. 2016.
[2] R. Rahim, M. Dahria, M. Syahril, and B. Anwar, “Combination of the Blowfish and Lempel-Ziv-Welch algorithms for text compression,” World Trans. Eng. Technol. Educ., vol. 15, no. 3, pp. 292–297, 2017.

[3] R. Rahim and A. Ikhwian, “Cryptography Technique with Modular Multiplication Block Cipher and Playfair Cipher,” Int. J. Sci. Res. Sci. Technol., vol. 2, no. 6, pp. 71–78, 2016.

[4] H. Nurdiyanto, R. Rahim, and N. Wulan, “Symmetric Stream Cipher using Triple Transposition Key Method and Base64 Algorithm for Security Improvement,” J. Phys. Conf. Ser., vol. 930, no. 1, p. 12005, Dec. 2017.

[5] E. Hariyanto and R. Rahim, “Arnold’s Cat Map Algorithm in Digital Image Encryption,” Int. J. Sci. Res., vol. 5, no. 10, pp. 1363–1365, Oct. 2016.

[6] A. Putera, U. Siahaan, and R. Rahim, “Dynamic Key Matrix of Hill Cipher Using Genetic Algorithm,” Int. J. Secur. Its Appl., vol. 10, no. 8, pp. 173–180, Aug. 2016.

[7] S. Islam, M. R. Modi, and P. Gupta, “Edge-based image steganography,” Eurasip J. Inf. Secur., vol. 2014, 2014.

[8] M. H. and M. Hussain, “A survey of image steganography techniques,” Int. J. Adv. Sci. Technol., vol. 54, no. 3, pp. 113–124, 2013.

[9] E. Zielińska, W. Mazurczyk, and K. Szczypiorski, “Trends in steganography,” Commun. ACM, vol. 57, no. 3, pp. 86–95, 2014.

[10] A. Binny and M. Koilakunta, “Hiding secret information using LSB based audio Steganography,” in Proceedings - 2014 International Conference on Soft Computing and Machine Intelligence, ISCI 2014, 2014, pp. 56–59.

[11] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, “Digital image steganography: Survey and analysis of current methods,” Signal Processing, vol. 90, no. 3, pp. 727–752, 2010.

[12] G. Singh and Supriya, “Modified vigenere encryption algorithm and its hybrid implementation with Base64 and AES,” in Proceedings - 2nd International Conference on Advanced Computing, Networking and Security, ADCONS 2013, 2013, pp. 232–237.

[13] L. Yu, Z. Wang, and W. Wang, “The Application of Hybrid Encryption Algorithm in Software Security,” Fourth Int. Conf. Comput. Intell. Commun. Networks, pp. 762–765, 2012.

[14] Y. Anggraini and D. V. S. Y. Sakti, “Penerapan Steganografi Metode End of File ( Eof ) Dan Enkripsi Metode Data Encryption Standard ( Des ) Pada Aplikasi Pengamanan Data Gambar Berbasis Java,” Konf. Nas. Sist. Informasi, STMIK Dipanegara Makassar, no. September 2016, pp. 1743–1753, 2014.

[15] M. Ramalingam and N. A. M. Isa, “A steganography approach over video images to improve security,” Indian J. Sci. Technol., vol. 8, no. 1, pp. 79–86, 2015.

[16] H. Nurdiyanto and R. Rahim, “Enhanced pixel value differencing steganography with government standard algorithm,” in 2017 3rd International Conference on Science in Information Technology (ICSIITech), 2017, pp. 366–371.

[17] K. Fiscus and D. Shinburg, “Base64 Can Get You Pwned,” 2011.

[18] Z. Liu, L. Liu, R. Hill, and Y. Zhan, “Base62x: An alternative approach to Base64 for non-alphanumeric characters,” in Proceedings - 2011 8th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2011, 2011, vol. 4, pp. 2667–2670.