Avrupa Bilim ve Teknoloji Dergisi
Sayı 14, S. 209-214, Aralık 2018

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
No. 14, pp. 209-214, December 2018

Research Article

Developing an End-to-End Secure Emailing Add-in using Steganography

Taner Sayim[1]*, Armagan Elibol[2]*
[1] Department of Mathematical Engineering, Yildiz Technical University, Davutpasa Campus, 34210, Istanbul-Turkey
[2] Department of Mathematical Engineering, Yildiz Technical University, Davutpasa Campus, 34210, Istanbul-Turkey

(First received 14 August 2018 and in final form 23 November 2018)

(DOI: 10.31590/ejosat.453530)

Abstract

Thanks to the emergence and spread of the internet, email is one of the main and fastest communication tools in both daily and business life. The amount and the importance of data exchanged in emailing have been continuously increasing. This brings the security-related issues. To be able to protect the security of the information, there has been a need to develop innovative ways. Although most email servers are already including some measures, there could some cases in which higher level security measures would be needed. In this paper, we propose an end-to-end secure emailing add-in using the steganographic method, which involves communicating secret data in an appropriate multimedia carrier, e.g., image, audio, and video files. The purpose of the system outlined in this paper is to implement the concept of "information confidentially" through both steganographic and cryptographic techniques. This system is integrated into one of the most commonly used commercial emailing software, Microsoft Outlook as an add-in.

Key words: Secure emailing, data hiding, steganography, software add-in

* Corresponding Author 1: (taner.sayim@intertech.com.tr) He is currently a Software Developer at Intertech Bilgi İşlem ve Paz.Tic. A.Ş. ORCID 0000-0003-4872-5994
* Corresponding Author 2: aelibol@jaist.ac.jp, He is currently affiliated with School of Information Science of Japan Advanced Institute of Science and Techonology. ORCID 0000-0003-0661-9536

1. Introduction

Considering today's technology, especially social media, and communication applications, it seems that there is security besides making improvements that are of utmost importance now. In this regard, there are important methods that have been developed by companies to keep their users from suffering information confidentiality. Examples include; cryptographic methods, Secure Socket Layer (SSL), End-to-End Encryption (E2EE), Point-to-Point Encryption (P2PE). In all these methods, the aim is to prevent the information from being understood and resolved by others, and communication is ensured through the transmission of information between the sender and receiver in a form that cannot be understood directly.

There are various methods (e.g., cryptography, hashing and many others) that have been developed for data hiding in order to protect the data from unauthorized access and tampering. Steganography, which is also one of those methods, aims to hide the existence of a message and create a "covert channel" [1]. Steganography can be performed in different ways, e.g., to hide the text into the text, hide the text into the picture, hide the picture into the audio file can be given. In this paper, our aim is to propose an end-to-end securing emailing system with steganography type of hiding text in an image focused on. One of the most important and preferred communication tools in daily business life is email. That's why security plays a crucial role due to the information exchange. Our aim is to use steganography to remove the visibility of the data (by hiding email body text into an image automatically and send the image as an attachment), and cryptographic techniques to make it difficult to resolve, even if the data were obtained. For this purpose, we developed an add-in for the Microsoft Outlook platform using steganographic and cryptographic techniques to ensure the security and confidentiality of the information. With the add-in developed, a safer communication environment is provided by hiding the data inside the image. Least Significant Bit (LSB) algorithm [2] is employed with random encoding and its corresponding decoding operations accordingly

The rest of the paper is organized as follows: The next section is to provide some information about Steganography while the third section explains the secure emailing system. The final section is to draw and discuss some conclusions.

2. Steganography Overview

The word Steganography has evolved from the Greek word, meaning *covered writing*. Steganography can be defined as concealing data within another, hence hiding the presence and existence of the communicated information [3,4]. In other words, Steganography is hiding a file, message, image or video within another file, message, image or video [4,5]. In steganography, the existence of the hidden message is known by the parties (the sender and the receiver), whereas in cryptography the existence of the hidden message is visible and known by everybody. Due to that, steganography also hides the existence of the hidden message and this provides the removal of the attention coming to the hidden message. For this reason, while the detection and decoding of steganography are complex, the analysis of cryptography is complex, but it is easy to detect [4-6]. The main difference between steganography and cryptography is that cryptographic methods aim to protect the content of the text, while steganography aims to hide the content [7,8,9]. There are three types of Steganographic protocols used namely, Pure, Secret Key and Public Key Steganography [7,8]. In this paper, Secret Key Steganography is used. This method relies on the prerequisite of exchange of a secret key (stego-key). Secret key Steganography uses stego-key while embedding the secret message. Only the ones who know the secret key can reverse the process and read the secret message.

An effective steganographic framework should have the following desired characteristics [5,8,10]:

Secrecy: Hidden data should not be extracted without knowing the secret key (stego-key).

Imperceptibility: Hidden data should cause major changes in the medium in which it is hidden. Hiding procedure should be done in such a way that it should cause any suspicion.

High capacity: It should be possible to hide data as long as possible.

Resistance: Hidden data should be preserved and resistant to the changes done in the host medium, for example by some lossy compression scheme.

Accurate extraction: Hidden data should be extracted both reliably and accurately when needed [5].

The pipeline of the hiding text into an image is given in Fig. 1 below.

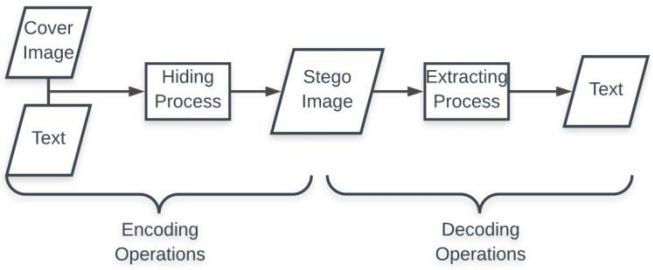

Fig. 1 The pipeline of the Steganography process

The cover image is the image where the text (or data) will be stored hidden. The text is the data to be hidden. Stego image is the image in which the text is already stored. The idea of steganography is the way in which the bits of text are hidden in the picture bits. The data to be stored in steganography is hidden in the pixel density of the image. The type of hiding of the stored data depends on the image type. For example, in RGB images, 1 pixel consists of 3 bytes (red, green, blue) and in 1-pixel 3-bit text data can be stored. This hiding can be done according to various algorithms. In this paper, the LSB insertion algorithm is used. In LSB, the bit is called the least significant bit is the last bit of a byte. If we consider 8-bit architecture, the contribution of the 8th bit can be maximum 1. The LSB algorithm works by replacing the bits of the text with the LSB bits of the host image. For example, RGB images have three LSB bits per pixel. This means that three bits data can be stored in one pixel's LSB bits. This algorithm can be used in sequential or random order. It is the most commonly used steganography algorithm because of its ease of use. Also, it is the easiest algorithm to decode. The sequential way of processing can be seen in Fig. 2.

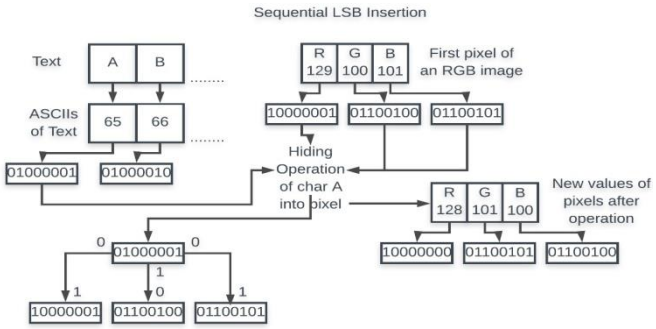

Figure 2 Sequential LSB Insertion

Let imagine that the letter A is to be hidden into a pixel value in Fig. 2. The letter A is first converted to ASCII code. The ASCII counterpart of A is 65 that is 01000001 in bits. The first element of this sequence, 0, will replace the LSB bit of the red value of the pixel. After this process is finished, the sequence goes to the second bit of A, 1, to process and continues. no change is made.

3. Secure Emailing Software Tool as Add-in

The goal is to provide a software tool which does hide email body text in an image selected by the user and email the stego image to the recipient as an encoding part and to extract the hidden message from the stego image as a decoding part. This tool developed as an add-in to facilitate and quicken its usage comparing to a desktop application. Our software maintains its own server in which user login and email transaction information are kept encrypted. Each time encoded email is sent by using the developed add-in, a single transaction data (from/to and time) is inserted into the database in the server and this information is also used as stego-key. This tool is developed as an add-in to Microsoft Outlook platform, which is commonly used as an emailing software. Visual Studio Tools for Office (VSTO) is used for development. Our tool is composed of different parts, Login and Membership, VSTO Ribbons and Task Pane to show hidden text.

Login and Membership: Our tool requires an additional login/sign-up a part different from the email account. This is to add an extra security level. The user information is kept in a database on the remote server. Exemplary screens are given in Fig. 3.

VISTO Ribbons: These are the main icons to use the proposed tool in this paper. An example is provided in Fig.3.

Data hiding (Encoding) process:

The pipeline of the process is outlined in Fig. 4. The input text can be either typed manually as email body text or a separate text file. In order to increase the security level, the text is first converted to Base64 representation as it is also commonly used in Email servers. To use the Base64 encryption algorithm, the input text must first be normalized. There are sixty-four characters in total in the Base64 encryption algorithm. Fifty-two

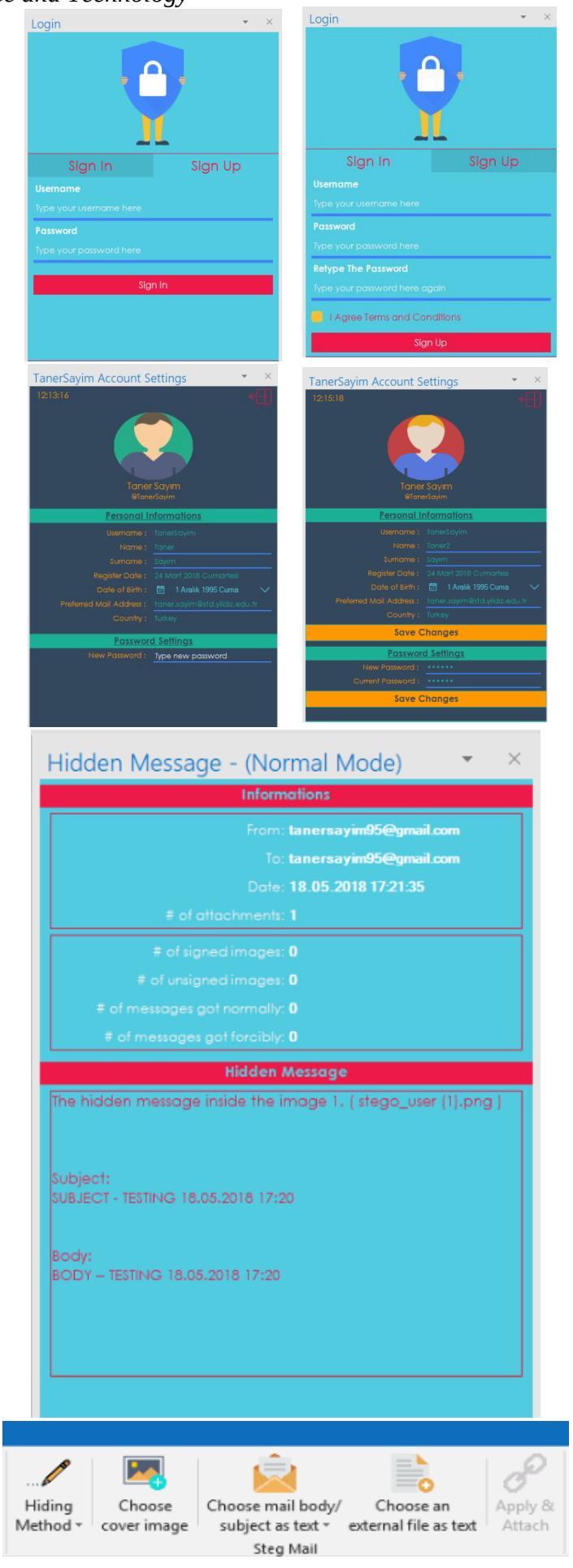

Fig. 3 Exemplary Screens from the add-in

of them are the lower and upper letters, ten of them are numbers and the remaining two are “+” and “/” characters. In order to encrypt any text with Base64, the characters of the text must be one of the 64 characters mentioned above. Each character of the given text is examined individually. If the current character is in the Base64 characters list, it is not normalized as it is already normal for Base64. The character in the normalization process is first found in the ASCII list. Then it is replaced between the “+/” and “/+” patterns. After concatenating all of these, the result will look like “+/…/+”. Here the “…” part is the ASCII counterpart of the character which is being normalized. As an example, the word “TÜRKİYE” is to be normalized. Here, the letters T, R, K, Y, and E are already in Base64; however letters Ü and İ must be normalized. After the normalization process, "TÜRKİYE" word turns into "T+/220/+RK+/304/+YE " which is suitable for Base64. After normalizing the text, the Base64 algorithm can be applied to encrypt the text. For example, "T+/220/+RK+/304/+YE" will turn into "VCsvMjIwLytSSysvMzA0LytZRQ==" under the UTF-8 charset. The next step is to parse and shuffle this Base64 string into bits. Let imagine a Base64 string like “abc”. If we consider bits representation of ASCII counterparts of each char, it would be; “011000010110001001100011”. This is the binary format to be shuffled. This binary text will be shuffled randomly with the seed value generated by the seed value generator function which can be seen in Figure 5.

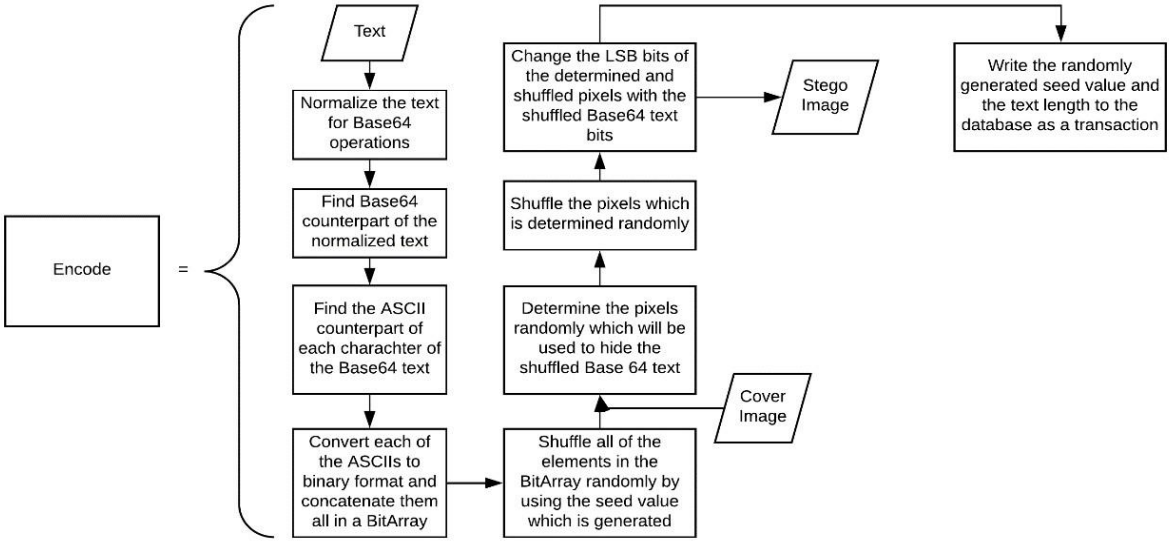

Fig. 4. The pipeline of the encoding process.

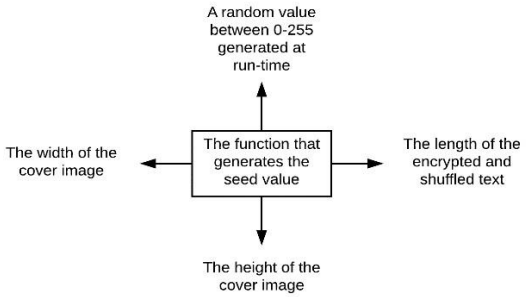

Fig.5. The seed generator function inputs

After shuffling of the location of each index by using the generated seed value, the new array of bits would be “010010100101110101111110”. Afterward, firstly octal blocks are reunited and three ASCII integers are obtained. Then finding the text counterpart of these ASCII integers, the output would be “J]~”. This is what is obtained at the end of the encryption process. Bit representation of this value will be hidden in the image. Now, the process of preparing the picture will be detailed: How many pixel values are needed to change is calculated before the hiding process starts. E.g., if the string of 27 characters is going to be hidden, this is 27 x 8 = 216 bits total. This means a total of 216/3 = 72 pixels in RGB images. Therefore, in the image, 72 pixels are randomly selected. Before selecting random pixels, the image is divided into parts. An equal number of pixels from each part is randomly selected. The reason for this is the more balanced distribution of a possible (color) change (due to the modification in RGB values) in the image. As in the text scenario, the seed value is used for the selection of pixel positions in each divided part. The pixel positions coming from different part of the image are shuffled again to ensure that bits of the string are randomly assigned in different parts of the image.

After all these operations, some information must be stored in the database stored in the server (which is different from the mail server) for decoding. These; the automatically assigned transaction identification number, the sender's username, the mail addresses of the sender and receiver, the transmission time of the mail, the randomly generated seed value, the length of the text and the conversation index unique to each mail.

Data Reading (Decoding) Process:

Decoding process checks first whether there is a transaction related to the current mail item or not. If it finds a corresponding record in the database, it retrieves the content of the record. With this information, the images in the mail attachments are tried to be decoded. To explain this process in more detail; the information obtained from the database is random seed value and text length information. The first thing to do is to divide the picture into parts and select an equal number of pixels from each region in which data is hidden during encoding part using the

random seed value and shuffling algorithm. Iteration is performed on the actual sequence obtained and all the bits required to be obtained according to the text length information are obtained. These bits are stored in a BitArray and then read into octal blocks and converted to ASCII. They are then converted to characters. This is the text shuffled Base64 string. What it is needed is to get the actual Base64 string from this shuffled Base64 string, just as it was done with pixels. Here is an algorithm similar to the algorithm that is employed for the pixel. The actual Base64 string is obtained through the function, which takes a random seed value as a parameter. First, this shuffled Base64 string is parsed into bits. And each one is stored in a one- dimensional array of BirArray. The next thing to do is to reorder the bits inside this BitArray by the generated random value. First, this shuffled Base64 string is parsed into bits. And each one is stored in a one-dimensional array of BitArray. The next thing to do is to reorder the bits inside this BitArray by the generated random value. The sorted bits are the text before the shuffle. And by reading these bits in octal blocks, we find the ASCII equivalents and then the character equivalents. At the moment we have the Base64 string. Now we will solve this with a decoding algorithm that decodes Base64. After decoding the Base64 string, the incoming string is the normalized version of the actual text. We normalize back this and we find the actual text.

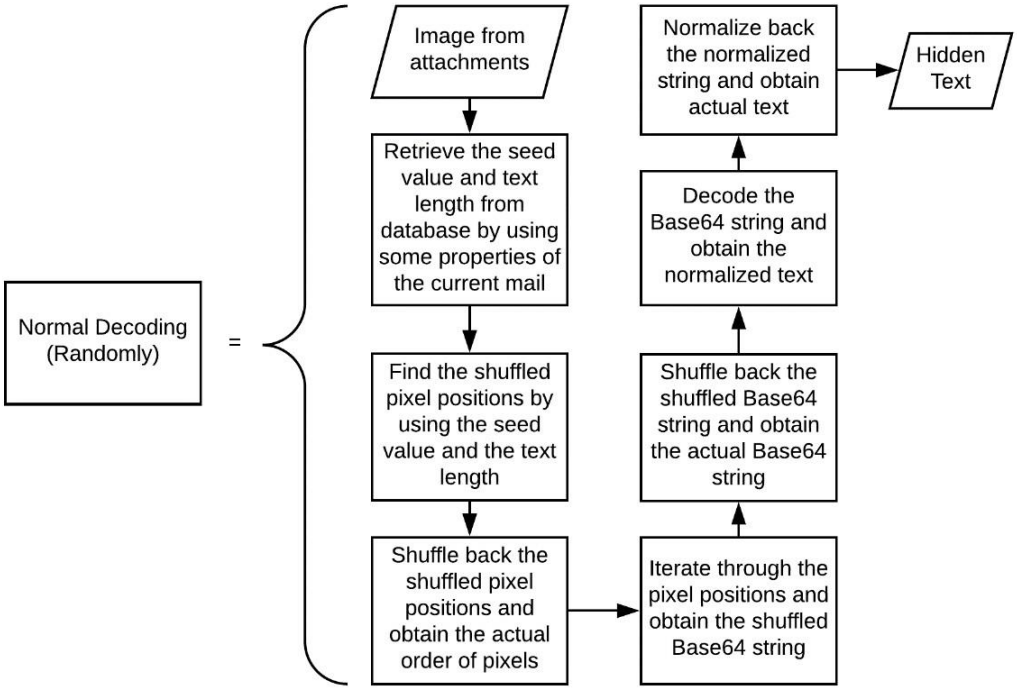

Figure 6. The pipeline of the decoding process

4. Conclusions and Future Work

Lately, the importance of preserving data has become a more important point than in the past. This is due to the widespread use of the internet and the increase in the size of the data. The data size expressed in kilobytes years ago is expressed in terabytes today. This leads people who are dealing with data security to produce new techniques and algorithms for the preservation of data. Steganography and cryptography are the best examples of these techniques. With the development of information technologies, significant developments have been made in these two areas. If we briefly compare these two; while the detection and decoding of steganography are complex, the analysis of cryptography is complex, but its detection is easy. This creates a possible suspicion for cryptography. This shows us that if steganography is used effectively, it would provide better results in terms of security. However, the combination of these two for more effective protection gives more effective results.

Recently, with the widespread use of cameras and the increasing use of camera phones, the number of pictures circulating on the internet has increased. This increase has also indirectly increased the use of image data. An example of this is image steganography. Images are used as cover data in the image steganography field. There are many algorithms and techniques developed in this area. One of these is the LSB insertion technique which is used in the project. This technique is the most preferred technique with ease of application. But the solving is also pretty easy. The LSB method used in this paper was implemented in randomly writing on PNG image files.

The add-in proposed in this paper was carried out in order to provide a more secure mail communication. With this add-in, users can send and receive steganographic images to each other. Our program allows users to encode and decode with a single keystroke. The add-in works on the Microsoft Outlook program. Due to the usage of randomness appear in both text reordering and pixels selection steps in the hiding stage,

The implementation of steganography in the paper consists of two different parts; encoding and decoding processes. Encoding and decoding processes apply randomly encoding-decoding to increase the security level.

There are a few areas where the software can be developed further. With the integration of the JPEG DCT algorithm, the

dependency on the PNG in the project will cease to exist, and users will be offered a new option. In addition, ready-made images can be integrated into the add-in. So the user can quickly select from ready-made images without having to select a specific image. In addition, the use of more than one image while encoding is done, and the subsequent merging of these images will also increase security. A new mapping algorithm can be developed cryptographically instead of Base64. In addition to the algorithmic enhancements, if the add-in is integrated into other platforms, users will be relieved of being dependent on the Microsoft Outlook program. For example, with an add-in to Microsoft Outlook Mobile, users will be able to do the same things they do on their smartphones.

Acknowledgments

The work in this paper was supported by the Scientific and Technological Research Council of Turkey (TUBITAK) with grant number 1919B011701791 entitled “Steganografik Tekniklerle Güvenli E-Posta İletişim Sistemi”

References

1. Kaushal, Singh V.P., Bamal R. (2013). Steganography: A Modern Day Art and Science For Data Hiding, International Journal of Latest Research in Science and Technology, 2(4), 9-14
2. Gupta, Shailender, Ankur Goyal, and Bharat Bhushan. "Information hiding using least significant bit steganography and cryptography." International Journal of Modern Education and Computer Science 4.6 (2012): 27.
3. Johnson N.F., Jajodia S. (1998). Exploring Steganography: Seeing the Unseen, IEEE, 31(2), 26-34

4. Zielińska, E., Mazurczyk, W., & Szczypiorski, K. (2014). Trends in steganography Communications of the ACM, 57(3), 86-95

5. Sheelu, Ahuja, B., (2013), “An Overview of Steganography”, IOSR Journal of Computer Engineering, 11(1), 15-19.

6. Rana M.S, Sangwan B.S., Jangir J.S. (2010). Art of Hiding: An Introduction to Steganography, International Journal of Engineering and Computer Science,1(1), 11- 22.

7. Jammi Ashok, Y. Raju, S. Munishankaraiah, K. Srinivas "Steganography: An Overview" in International Journal of Engineering Science and Technology, 2(10), 2010.
8. K.P.Adhiya and Swati A. Patil, Hiding Text in Audio Using LSB Based Steganography, Information and Knowledge Management, 2(3), 2012.
9. Pratap Chandra Mandal, Modern Steganographic technique: A survey, International Journal of Computer Science & Engineering Technology (IJCSET), 2012.
10. Rout, H., Mishra, B.K., Pros, and Cons of Cryptography, Steganography, and Perturbation techniques, IOSR Journal of Electronics and Communication Engineering, 76-81, 2014.

.