ZAMBIA INFORMATION COMMUNICATION TECHNOLOGY (ICT) JOURNAL

Volume 7 (Issue 1) (2023) Pages 23-28

A Privacy-Preserving Scheme for Medical Diagnosis Records Based on Encrypted Image Steganography

Kaku Lishomwa[a] and Aaron Zimba[b]

a. Department of Computer Science & IT, Mulungushi University, Kabwe, Zambia, email: kakulishy@gmail.com

b. Department of Computer Science, ZCAS University, Lusaka Zambia, email: aaron.zimba@zcasu.edu.zm

Abstract— Healthcare records are essential as they establish perpetual reports on the health of patients. These records are in nature meant to be kept confidential. It is for this reason that the data should be secured before transmission. Steganography and cryptography are old concepts used to secure communications. Steganography is hiding data in a carrier, for instance, images, videos, texts, and files. It seeks to hide the existence of communications. Cryptography is communication in deciphering secret writings or ciphers. Steganography and cryptography can both work independently, the two techniques are combined to make communications more secure. In our privacy-preserving scheme for medical diagnosis records, a hybrid approach is used to ensure the maximum security of medical records before transmission. The proposed system utilizes AES 128 which is symmetric key encryption to scramble the data. As compared to asymmetric key encryption, symmetric key encryption is faster because both encryption and decryption utilizes a single key. The proposed system makes use of the Diffie Hellman key exchange algorithm for key agreement. The least significant bit (LSB) which involves the hiding of information in the most repetitive bits of each pixel of an image is implemented for steganography. The technique was chosen because the level of distortion made on the image is low hence the image quality change is minimal.

Keywords— Steganography, Cryptography, Diffie-Hellman, LSB, AES

I. Introduction

The Federal Chamber of Medicine through Resolution 1.638 [1] characterizes Healthcare records as an exceptional archive that is comprised of a lot of recorded data marked and pictures created dependent on realities, events, and circumstances of the soundness of the patient and the consideration given. Healthcare records give research help. Clinical specialists can similarly survey indications and signs as a result of further developing assurance and giving treatment [1].

Security refers to physical, innovative, or regulatory defenses or apparatus used to shield recognizable well-being information from undesirable access or exposure. The present age relies upon the organization and PC framework to furnish them with data, for example, health or clinical records. Most people have communicated worries over undesirable surveys of clinical records when data is shared between medical service providers [2]. Healthcare information that incorporates medical records and clinical pictures is traded between different places. The delicate data of this information requires greater security as they are communicated over the internet which expands the danger of a capture attempt. There is a need to secure healthcare records inside and out. This insurance can be conceded by utilizing security highlights, for example, Firewalls, Cryptography, and Virtue Private Networks (VPN), however, a portion of these strategies don't offer total security since they can undoubtedly be bypassed by hackers [3]. An efficient way of making health record transmission more secure is through the combination of Steganography and Cryptography, note that the two techniques can work independently to secure data. These techniques can therefore be combined to make data or the transmission of data more secure.

Steganography is a technique of hiding secret information in a carrier (image, video, audio, binary files, documents), it hides the presence of information. Steganography has the following components: the carrier, the message as well as the key [4]. Cryptography is communication in deciphering secret writings or ciphers. Information is scrambled in a way that makes it un-understandable to unintended users [5]. The proposed system uses cryptography to scramble Healthcare Records to make them un-understandable to un-intended users and steganography to embed the scrambled information into an image.

Interactive media objects, for example, messages, pictures, recordings, and sound are utilized as a cover medium to guarantee the sheltered transmission of private data or messages. The following are the types of steganography: Text Steganography where data is hidden in a text file, this type of steganography works by hiding the confidential data behind the nth letter of every word in the text message and uses several methods to hide the confidential data in the text file, these include: a method based on format, a method based on Random and statically, and methods based on linguistics [6]. Image Steganography is the most commonly used, it works by embedding confidential data or messages into an image and the result is a stego image that is then sent to the receiver. During the transmission, unauthorized persons don’t see the hidden information but rather only see the stego image[7] Audio Steganography is where data is hidden in an audio file which can be MP3, AU, or WAV sound files. This type of steganography uses several methods which include: Low Bit Encoding, Phase Coding, and Spread Spectrum [8]. Video Steganography is where data is hidden in video files. Video files are a combination of images and sounds. Videos are made up of images and audio hence techniques used in image and audio steganography apply to video steganography [9].

Cryptography is a technique in which information or messages are concealed in ways that make the information useless when landed on unintended users. Encryption and decryption are the fundamental functions of cryptography. Plain texts are converted into forms that cannot be read and understood, the converted plain text is called cipher texts (encryption), and the cipher texts are converted back to plaintexts (decryption). The following are the types of cryptography: Symmetric Key Cryptography which utilizes the same key for encryption and decryption. Instances of symmetric encryption algorithms include Advanced Encryption Standard (AES), Data Encryption Standard (DES), Internal Data Encryption Algorithm (IDEA), Rivets Cipher 4 (RC4), Rivets Cipher 5 (RC5), Rivets Cipher 6 (RC6) [10]. Asymmetric Key Cryptography is a type of cryptography that utilizes two distinct keys for encoding and decoding. A public key is utilized for encoding the information, and a private key is utilized for decoding the information. Examples of algorithms that utilize two distinct keys for encryption and decryption are Rivets Shamir Adelman (RSA), Diffie Hellman, and Elliptic Curve Cryptography (ECC) [10].

The rest of the paper is coordinated as follows; Section II presents the related works, and Section III brings forth the methodology and proposed framework. The results and analyses are presented in Section IV and the conclusion in Section V.

II. Related Works

Various Stenographic tools have been developed and the essential thought connected with these tools is something similar: to make software that can hide information in a medium. For this research paper, we review 5 stenographic tools namely: Xiao Steganography, CrptaPix, OurSecret, MP3Stego, and Masker.

Xiao Steganography [11] was developed by www.nakasoft.net. The software utilizes Bitmap (BMP) images and WAV files to hide the secret information. Various encryption algorithms are used: RC4, DES, 3DES 112, RC2, and hashing SHA, MD4, MD2, and MD5. The hidden information can only be extracted using this software. Xiao Steganography uses Least Significant bit (LSB) substitution as a stenographic algorithm.

CryptaPix [12] was developed by Kent Briggs. The software works by dividing the plaintext into 3 bits segments and uses AES 256 bits encryption. It is an encryption and file manager program for the Windows platform. The disadvantage is that it uses a 3-bit segment algorithm which is vulnerable to Robustness [14], this means that any image processing operation (contrast, brightness, etc.) performed on the stego image can destroy the secret information [15].

OurSecret [16] is software that shrouds text files, recordings, sounds, and pictures in a document. This software sends and hides confidential files and messages, information is encoded and covered up in documents,[16]. The software has a basic interface, the user loads the transporter record into the interface, the document conveys data and the user at that point creates a secret phrase to permit just approved clients to access the secret data. The favorable circumstances are that it embeds records in different documents consequently making it not perceptible to the users, the interface is easy to understand, and uses a secret key. The fact that it does not have a recovery option disadvantages the users in a case where the password for the carrier file is forgotten [17].

Mp3Stego [18] was developed by A.P Petitcolas. The software works by concealing secret data into MP3 records. The secret data is compressed, scrambled, and is currently covered up in the MP3 bit stream. The encoding cycle happens in the inward circle which quantizes the data info and increments the quantizer step size not until the quantized data is encoded in the accessible pieces [18]. The product utilizes a pseudo irregular piece generator dependent on SHA-1 and 3DES encryption [18]. The favorable position is that MP3Stego utilizes Audio documents that are bigger and utilizes MP3 design, P2P programming, and ties down communication plans to keep up a secret of the data to be sent in any event, when it goes through channels that are not made sure about. Notwithstanding the benefits of utilizing MP3 records, they likewise have their downsides. MP3 documents have dynamic matchless quality of the HAS over humans and the current information can without much of a stretch be perceived outwardly and permits just a specific measure of information to be stored [19]. The other burden of MP3Stego is that any rival can compress and recompress and accordingly the concealed data is deleted [20]

Masker is a program that utilizes steganography to veil mystery data in media records, for example, pictures and films. It empowers users to shroud secret data in records and is utilized to ship the information. Users select a document to convey the information and are moved to different users without a doubt. The program utilizes pictures, and recordings, The program gives users a few encryption methods (Blowfish, CASTS, DES, serpent-256, AES-256, 3DES, and Twofish) and the user will pick the algorithm that coordinates the user's security prerequisite and a secret password is entered to forestall unapproved access. The advantage is that the program doesn't have a cutoff for the measure of data to be put in the carrier [21]. The disadvantage is examining an AVI record with a modest quantity of installed content so the product utilizes EOF information injection. [22].

Table I summarizes the differences between our proposed model and existing approaches.

As seen in Table I, our proposed system has several advantages as compared to other systems. A secure steganography system requires that all three parameters detectability or imperceptibility which is the first parameter to be considered, followed by Robustness, and finally capacity are met [23]. Our system meets all three parameters hence making it more secure.

TABLE I. COMPARISON WITH OTHER WORKS

Application name	Carrier	Send text	Compression on	High Imperceptibility	High Robustness	High Capacity	Encryption
Xiao Steganography	Image/Audio	✓	×	×	×	✓	Symmetric
CryptaPix	Image	×	✓	✓	×	✓	Symmetric
OurSecret	File	✓	×	×	×	✓	Asymmetric
Mp3Stego	Audio	✓	✓	×	×	✓	Symmetric
Masker	File	×	✓	×	×	✓	Hybrid
Proposed system	Image	✓	✓	✓	✓	✓	Hybrid

III. METHODOLOGY AND PROPOSED FRAMEWORK

The proposed system has two modes, the client/server-side mode, and the normal mode. It aims at facilitating the transfer of healthcare records in a secure manner. The system implements a client/server that enables the agreement of a common key. The client/server implements a Diffie Hellman key exchange that allows the two parties to come to a ground of a common key. The system also implements AES 128 bits algorithm that is used for encrypting the data. The proposed system uses jpg and png images. The image extension was chosen because the majority of the images today are in those extensions thereby drawing fewer suspicions.

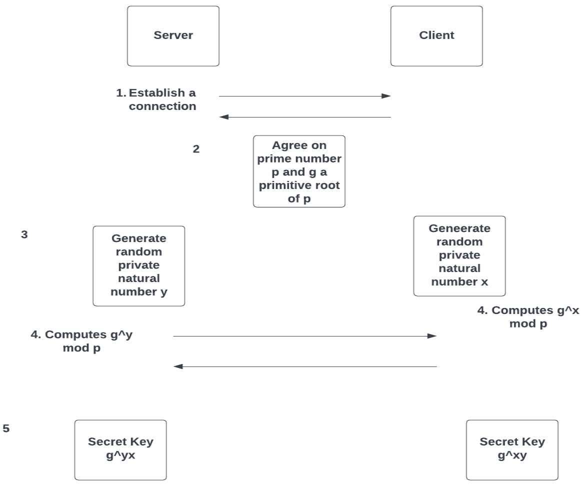

Server
Client
1. Establish a connection
2
Agree on prime number p and g a primitive root of p
3
Generate random private natural number y
Geneerate random private natural number x
4. Computes g^y mod p
4. Computes g^x mod p
5
Secret Key g^yx
Secret Key g^xy

Figure 1. Client/Server Mode

Figure 1 shows the system framework for the client/server mode of the proposed system. The proposed system utilizes java socket programming, and Diffie Hellman is utilized for key agreement. The server is first started on port 8088 and gets the IP address of the host machine. After the server starts, it waits for a client to connect on port 8088, immediately after the client is connected to the server the key agreement is initiated.

The figure below shows the basic system architecture for the normal mode. The user interacts with the system by providing the system with the cover image, key, and message as inputs. The key to be used is the agreed-upon key in the client/server mode.

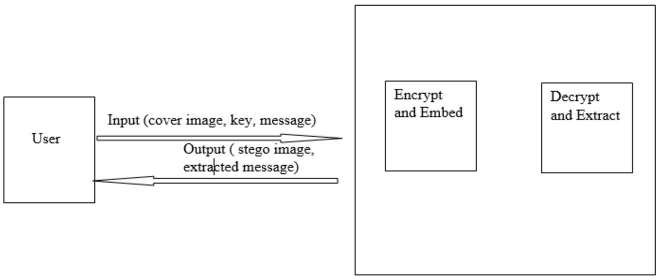

User
Input (cover image, key, message)
Output (stego image, extracted message)
Encrypt and Embed
Decrypt and Extract

Figure 2. Normal Mode

Algorithm 1: Diffie Hellman Key Exchange

Input: ***P****-Prime number, G-Primitive root modulo, S-Server, Y-Private natural number or Server, X-Private natural number for Client, C-Client*

Output: S-*Shared secret key*

1. S and C agree on a prime modulus number P and primitive root modulo G.
2. Verify that P and G are core primes
3. S generates Y, then calculates $A = G^Y \ mod \ P$
4. C generates X, then calculates B= G^X mod P
5. S sends C the value of A and C sends S the value of B.
6. *S computes* G^{YX}

```
7.  C computes $G^{XY}$
8.  End
```

Algorithm 2: Encryption and Encoding

Input: *N*-Cover image, S-Secret key, M-Secret message, L-Least significant bit (LSB), R-Binary bits, E-Base64Encoder
Output: X-*Cipher text, Z-Stego Image*

```
1.  Read N and M
2.  Convert M into X using E
3.  Add the value of X's length at the beginning of the text
    along with a '/' character (to mark the actual cipher
    length during decryption).
4.  Convert X into R to give M characters' bits
5.  Find L of the RGB pixels of N
6.  Embed R into L of the RGB pixels of N
7.  End
```

Algorithm 3: Decoding and Decryption

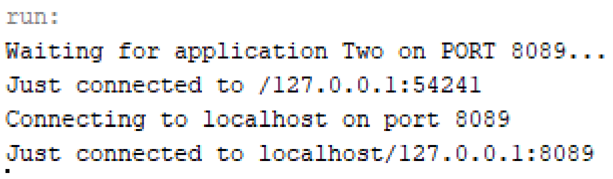
Input: S-Secret key, *Z-Stego Image, V-Binary String, W-String builder,* L-Least significant bit (LSB), D-Base64Decoder, M-Secret message
Output: M-Secret Message

```
1.  Read S and Z
2.  Retrieve  L of each RGB pixel of S
3.  Convert V found by every 8 RGD pixels of step 2 to a
    character and append the characters to a W.
4.  Upon finding the first '/' character from W, save its
    previous characters as text length and discard all
    characters till that index.
5.  Using S and D, decrypt X to get M
6.  End
```

IV. Results Analysis and Discussions

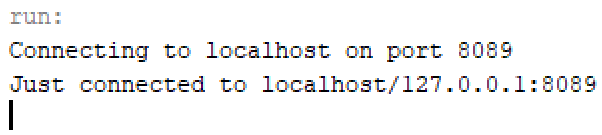
A. Description of Key Agreement

To apply the previously mentioned framework, we first start by running the application on the server's side on port 8089 and await for a client to connect.



```
run:
Waiting for application Two on PORT 8089...
Just connected to /127.0.0.1:54241
Connecting to localhost on port 8089
Just connected to localhost/127.0.0.1:8089
```

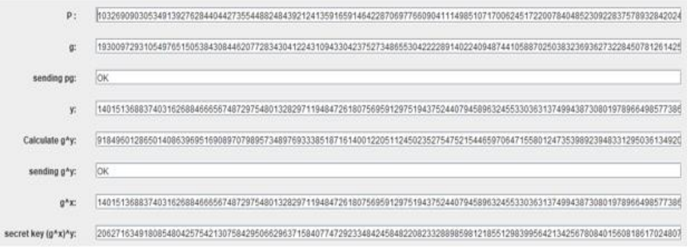
Figure 3. Server Console



```
run:
Connecting to localhost on port 8089
Just connected to localhost/127.0.0.1:8089
```

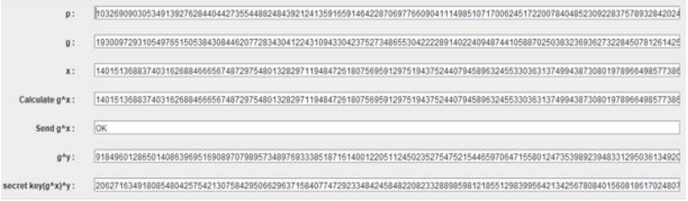
Figure 4. Client Console

Once the client has been connected to the server, the key agreement is initiated. Diffie Hellman is a key exchange algorithm that enables two or more computers to create a common key [24]. As seen in Figures 5 and 6 below, the server and the client agree on a prime number p and a primitive root modulo g. The server generates a random private natural number y, and the client generates a random private number x. The secret key is computed using the formulag^xy. The whole idea behind the Diffie Hellman key exchange is to enable the two parties communicating to end up with a common key without having to exchange it via any communication medium such as WhatsApp, telegram, etc. hence an additional security feature for our system.



P : 103269090305349139276284404427355448824843921241359165914642287069776609041114985107170062451722007840485230922837578932842024
g: 193009729310549765150538430844620772834304122431094330423752734865530422228914022409487441058870250383236936273228450781261425
sending pg: OK
y: 140151368837403162688466656748729754801328297119484726180756959129751943752440794589632455330363137499438730801978966498577386
Calculate g^y: 918496012865014086396951690897079895734897693338518716140012205112450235275475215446597064715580124735398923948331295036134920
sending g^y: OK
g^x: 140151368837403162688466656748729754801328297119484726180756959129751943752440794589632455330363137499438730801978966498577386
secret key (g^x)^y: 206271634918085480425754213075842950662963715840774729233484245848220823328898598121855129839956421342567808401560818617024807

Figure 5. Server's GUI after client connection.



p : 103269090305349139276284404427355448824843921241359165914642287069776609041114985107170062451722007840485230922837578932842024
g : 193009729310549765150538430844620772834304122431094330423752734865530422228914022409487441058870250383236936273228450781261425
x : 140151368837403162688466656748729754801328297119484726180756959129751943752440794589632455330363137499438730801978966498577386
Calculate g^x : 140151368837403162688466656748729754801328297119484726180756959129751943752440794589632455330363137499438730801978966498577386
Send g^x : OK
g^y : 918496012865014086396951690897079895734897693338518716140012205112450235275475215446597064715580124735398923948331295036134920
secret key(g^x)^y : 206271634918085480425754213075842950662963715840774729233484245848220823328898598121855129839956421342567808401560818617024807

Figure 6: Client's GUI after when connected to the server

B. Encryption and encoding

After the key has been agreed upon in the client/server mode, the first 16 characters of the secret key are utilized as the key during the encryption. The 16-character key is then changed to the required data type. If the user entered key is not of the length 16, the system pads the key with later "a" to reach the required key length. The instance of the AES cipher is taken and initialized with the ENCRYPT_MODE and given key. The message is then scrambled and the code string is produced. In an instance where the given key characters are longer than the required key size, the system encounters an error. After the message has been scrambled, the system checks if the bits of the cipher can be accommodated by the cover image. If the size of the image cannot accommodate the character bits of the cipher message, the cipher is not embedded into the image, if the cover image size is sufficient enough, the cipher is embedded into the cover image using the lowest significant bit (LSB). The cipher is only embedded into the last bits of the image hence the change in the image quality is barely seen.

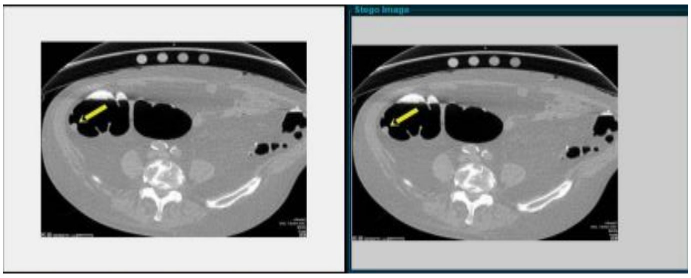
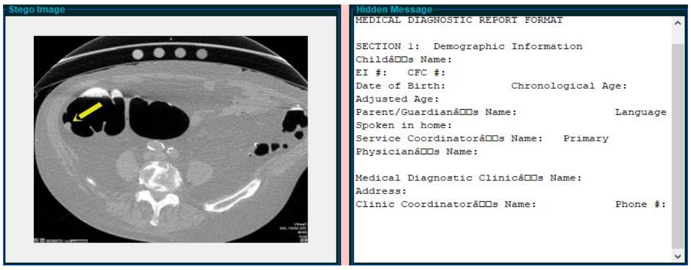


Figure 7: Data Encoding

The first image in the figure above shows the original image, and the second image is the resultant stego. The performance of steganography systems is measured using the following parameters: detectability, robustness, and capacity [23]. From figure 7, it is seen that the difference between the original and stego image is barely noticeable by the human eye thereby achieving imperceptibility. The format of the images used is portable network graphics (PNG). It has been proven that the PNG format is the best with regards to image steganography because it uses lossless compression so the substitution made during the entire process of LSB steganography is not lost and furthermore provides huge storing capacity[25].

C. *Decryption and decoding*

The proposed system utilizes a java API Base64 Decoder. A stego image is taken in as an input with the associated key. The scrambled message is first extracted from the received stego image. The first pixel of the RGB of every 8 bits is converted into characters. After the scrambled message has been extracted, it is decrypted by the AES 128-bits key. The entered key is converted into the required data type and utilized with DECRYPT_MODE and given a key by the AES instance. Figure 8 below shows the results on the receiver's side, the stego image on the left-hand side, and the extracted medical record on the right-hand side.

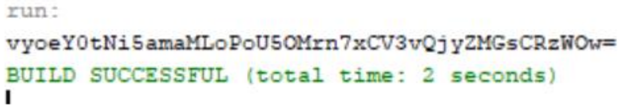


```
MEDICAL DIAGNOSTIC REPORT FORMAT

SECTION 1:  Demographic Information
Childâ□□s Name:
EI #:   CFC #:
Date of Birth:          Chronological Age:
Adjusted Age:
Parent/Guardianâ□□s Name:               Language
Spoken in home:
Service Coordinatorâ□□s Name:   Primary
Physicianâ□□s Name:

Medical Diagnostic Clinicâ□□s Name:
Address:
Clinic Coordinatorâ□□s Name:            Phone #:
```

Figure 8: Message Decoding

D. *Results Analysis and Discussions*

The first test was done on the encryption module. The total time of encryption has been recorded in seconds.

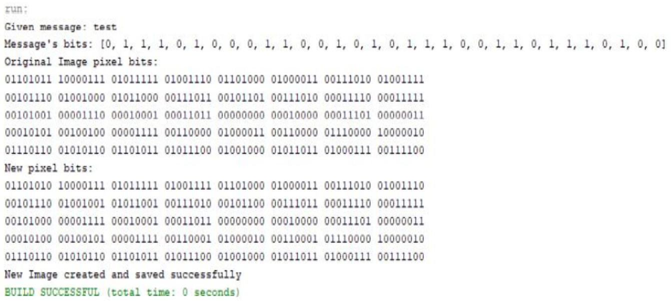


```
run:
vyoeY0tNi5amaMLoPoU5OMrn7xCV3vQjyZMGsCRzWOw=
BUILD SUCCESSFUL (total time: 2 seconds)
```

Figure 9: Cipher Text

On figure 9, the module was tested with a string, and the message was encrypted in 2 seconds. The time of execution was optimized because a symmetric encryption algorithm (AES) was used. The main advantage of symmetric encryption is that it is quicker and can also be used on massive data transmission[26].

The second test was done on the encoding module.

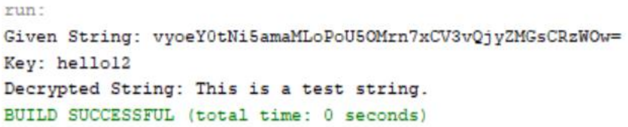


```
run:
Given message: test
Message's bits: [0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0]
Original Image pixel bits:
01101011 10000111 01011111 01001110 01101000 01000011 00111010 01001111
00101110 01001000 01011000 00111011 00101101 00111010 00011110 00011111
00101001 00001110 00010001 00011011 00000000 00010000 00011101 00000011
00010101 00100100 00001111 00110000 01000011 00110000 01110000 10000010
01110110 01010110 01101011 01011100 01001000 01011011 01000111 00111100
New pixel bits:
01101010 10000111 01011111 01001111 01101000 01000011 00111010 01001110
00101110 01001001 01011001 00111010 00101100 00111011 00011110 00011111
00101000 00001111 00010001 00011011 00000000 00010000 00011101 00000011
00010100 00100101 00001111 00110001 01000010 00110001 01110000 10000010
01110110 01010110 01101011 01011100 01001000 01011011 01000111 00111100
New Image created and saved successfully
BUILD SUCCESSFUL (total time: 0 seconds)
```

Figure 10: Message and image bits

On figure 10, the message bits were embedded into the lowest significant bits of the cover image and the message was successfully embedded.

The third test was done on the decryption module.

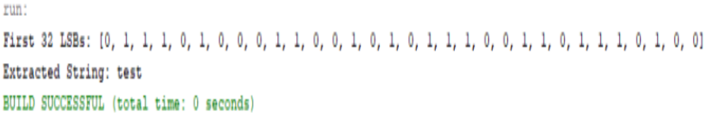


```
run:
Given String: vyoeY0tNi5amaMLoPoU5OMrn7xCV3vQjyZMGsCRzWOw=
Key: hello12
Decrypted String: This is a test string.
BUILD SUCCESSFUL (total time: 0 seconds)
```

Figure 11: Original message

On figure 11, the given string was taken as input and the associated key was used for decrypting the message.

The fourth test was done on the decoding module. The total time of execution was recorded in seconds.



```
run:
First 32 LSBs: [0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0]
Extracted String: test
BUILD SUCCESSFUL (total time: 0 seconds)
```

Figure 12: Extracted String

On figure 12, the first pixel of the RGB of every 8 bits was converted into characters, and the message was successfully extracted from the stego image.

V. Conclusions

In this paper, we presented a technique to secure the transmission of healthcare records. Apart from hiding the content of our messages, we must hide the existence of our communication. Other than using Steganography, we used

Cryptography to add another security feature to our system hence hiding both the content and existence of our communication. We also reviewed different systems in relation to our system and presented the comparisons. As compared to other systems, our system uses the AES encryption algorithm which is stronger and more secure compared to other symmetric algorithms such as 3DES, DES, RC2, and more. The proposed system implements the LSB algorithm for steganography which is considered the best technique for image steganography.

VI. References

[1] Pedro Luiz Côrtes, "HOSPITAL INFORMATION SYSTEMS: A STUDY OF ELECTRONIC PATIENT RECORDS," *J. Inf. Syst. Technol. Manag.*, vol. Vol. 8, No, 2011.

[2] M. V. P. P. H. J. M. L. S. J. W. Barker, "Individuals' Perceptions of the Privacy and Security of Medical Records," 2015.

[3] M. M. H. M. S. T. A. H. M. A. A. H. A. H. M. S. M. R. S. Islam, "Securing Medical Data Transmission Systems Based on Integrating Algorithm of Encryption and Steganography," 2019.

[4] N. Nabavian, "CPSC 350 Data Structures: Image Steganography," 2007.

[5] E. Cole and K. R. D., "Hiding in Plain Sight: Steganography and the Art of Covert Communication," 2001.

[6] M. J. S. A. S. B. J. D. S. N. Choudhary, "Steganography Techniques," vol. 5, 2017.

[7] B. A. Sheelu1, "An Overview of Steganography," *OSR J. Comput. Eng.*, vol. 11, 2019.

[8] D. V. Jasleen Kour, "Steganography Techniques –A Review Paper," *Int. J. Emerg. Res. Manag. &Technology*, vol. 3, 2014.

[9] P. V. K. Sagar S.Pawar1, "REVIEW ON STEGANOGRAPHY FOR HIDING DATA," vol. 3, 2014.

[10] V. K. and A. S. Mitali1, "A Survey on Various Cryptography Techniques," *Int. J. Emerg. Trends Technol. Comput. Sci.*, vol. 3, 2014.

[11] D. Kundu and A. Upreti, "Study of Various Steganography Tools," *2018 Int. Conf. Autom. Comput. Eng. ICACE 2018*, pp. 117–120, 2018, doi: 10.1109/ICACE.2018.8687092.

[12] H. A. and H. Hajjdiab, "A Comparison between Steganography Software Tools," 2017.

[13] "CryptaPix," 2020. http://www.resourcefill.com/36610/CryptaPix.html

[14] F. M. M. A. M. M. M. A. Shah, "Cryptography: A Comparative Analysis for Modern Techniques," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, 2017.

[15] N. K. S. Behal, "A Survey on various types of Steganography and Analysis of Hiding Techniques," vol. 11, 2014.

[16] OurSecret, "OurSecret," 2020. https://oursecret.software.informer.com/2.5/

[17] "FindMySoft Editor's Review," 2012. http://steganography.findmysoft.com/

[18] M. Noto, "MP3Stego: Hiding Text in MP3 Files," *SANS Inst. Inf. Secure. Read. Room*, vol. 1.2, 2001.

[19] M. N. V. M. V. K. Jain, "Audio Steganography – A Review," *Int. J. Adv. Res. Electron. Commun. Eng. (IJARECE*, vol. 2, 2013.

[20] Mp3Stego, "MP3STEGO," 2020. https://www.darknessgate.com/2014/12/25/mp3stego-2/

[21] "Masker," 2020. https://www.softpedia.com/get/Security/Encrypting/Masker.shtml

[22] Julio Hernandez, "Forensic analysis of video steganography tools," 2015.

[23] O. A. Nagham Hamid, Abid Yahya, R.Badlishah Ahmad, "Image Steganography Techniques: An Overview," 2012.

[24] S. Kallam, "Diffe-Hellman: Key Exchange and public key cryptosystems," 2015.

[25] M. B. Abha Sachdev, "Enhancing Cloud Computing Security using AES Algorithm," *Int. J. Comput. Appl. (0975 – 8887) Vol. 67– No.9, April 2013*, no. 67, 2013.

[26] H. Tyagi, "Symmetric Vs Asymmetric Encryption," 2020. https://www.codeitbro.com/symmetric-vs-asymmetric-encryptio