

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/316175714>

# Design of Image Steganography based on RSA Algorithm and LSB Insertion for Android Smartphones

Article in International Journal of Computer Applications · April 2017

DOI: 10.5120/ijca2017913557

CITATIONS

24

READS

3,997

2 authors:



**Richard Apau**

African Union

18 PUBLICATIONS 135 CITATIONS

[SEE PROFILE](#)



**Clement Adomako**

Kwame Nkrumah University Of Science and Technology

1 PUBLICATION 24 CITATIONS

[SEE PROFILE](#)

# **Design of Image Steganography based on RSA Algorithm and LSB Insertion for Android Smartphones**

**Richard Apau**

Department Of Computer Science  
Kwame Nkrumah University of Science and  
Technology, Kumasi, Ghana

**Clement Adomako**

University Information Technology  
Services (UITS)  
Kwame Nkrumah University of Science and  
Technology, Kumasi, Ghana

## **ABSTRACT**

Modern advancement in communication technologies has resulted in the widely and increase in use of smartphones such as android, blackberry, iPhones and much more. The proliferation of smartphones raises much security issues. This is so because the security features of such devices are limited. The most novel approach to arrest the security challenges in the smartphone is cryptography and steganography. Cryptography concerns itself with the masking of the content of a secret message whereas steganography deals with the concealment or hiding of a secreted message from the unauthorized person. The system proposed in this study uses a cover object, image specifically to hide the message to be sent. Before a message is embedded in the image, the message is first encrypted using RSA encryption algorithm. After the message has been encrypted, the process of embedding or hiding the message in the image is carried on. Least Significant Bit (LSB) technique is used to embed the message into the video. The performance Analysis was carried out using Peak-Signal-to -Noise-Ratio (PSNR). The results show that high security and robustness is achieved in smartphones when cryptography is combined with steganography.

## **General Terms**

Cryptography, Steganography, Steganalysis, Data Communication, Security

## **Keywords**

Image Steganography, Smartphones, Android, Cryptography, LSB, RSA, PSNR.

## **1. INTRODUCTION**

Mobile phones in recent times have become much more powerful than previously. Increase in the memory capacities, higher performance of the processor, greater features like accelerometers, light sensors, greater camera pixels and much more have raised the bounds of the modern mobile phone. This advances in hardware capabilities have pushed the bounds to the software's developers can write for phones. The rate at which smartphones are currently being used can be attributed to social enterprises and networking and some instances faster way of sharing videos, photos, and text. According to [1], the functionality of mobile phones which is similar to computers which provide all- in- one portable device in terms of interconnectivity has made smartphones part and parcel of individuals living in this century. With the introduction of 4G technologies, there will be an improvement in the capabilities of smartphones which will, in turn, propel the rapid and usage of such phones. This presupposes that the popularity of smartphones will continue to go high exponentially. Undoubtedly, the most popular and widely use operating system for mobile phones in recent times is ANDROID [2]. Android is an open-source platform

developed by Google and the Open Handset Alliance on which interesting and powerful new applications can be quickly developed and distributed to many mobile device users" [3]. The flexibility, easiness and less complex nature of android has made it the most preferred operating system over windows and iOS. In as much as smartphone technologies have increased and it users enjoy the platform, there are seemingly threats that users' of such devices are exposed to [4]. The challenges that are currently being faced by smartphone users are similar to problems that computer users faced some past years. [4] Opined that there are limitation and underdevelopment of security resources in android based smartphones. However, the complication associated with smartphones continues to expand amidst threats of the number and type of network. These threats, therefore, make smartphones easy prone to attacks by crackers and malware than our normal desktop computers which is protected. [5] Opined that data communications is needless if there is no security of the data that is being transmitted. Steganography hides the existence of a message whiles cryptography masks the content of a message. In lieu of this, an approach is proposed in this study to further secure the communication of data in the android smartphone. This study is however designed to work in image steganography using RSA algorithm and LSB insertion for android based smartphones.

## **2. REVIEW OF LITERATURE**

### **2.1 Image Steganography**

The technique of hiding secret information or data in an image is called image steganography. Generally, pixel intensities are the methods used in hiding data in image steganography. According to [7], images are the most popular and widely use cover objects used in steganography. The degree of redundancy in images has made it the most sought for, in terms of steganography. Two categories of classification namely spatial -domain and transform domain based have been proposed in image steganography [6]. [8] Explained that spatial domain embeds the message directly into the pixels intensity whereas the transform domain also called the frequency domain transform the image before the message is embedded. Various file formats exist in image steganography. TIFF, JPEG, PNG, GIF and BMP can all be implementing in image steganography [9]. However, each of the file formats poses its own unique advantages and disadvantages. Because pixel intensities are used in image steganography, there is sometimes variation in the intensity of the original image and the stego image or the embedded image. The variation in intensity is so trivial or subtle in that it is not detectable or perceptible to the human eye [8].

## 2.2 Comparison of Symmetric and Asymmetric Cryptography

By far, the asymmetric cryptographic algorithm is the most secured type of cryptography [10] due to its mathematical functions [11]. Asymmetric cryptography addresses the problem of key distribution for encryption [12] still remains a major problem in symmetric cryptography. Asymmetric key cryptography implements a digital which allows a recipient of a message verify that indeed the message is coming from a particular sender [12]. The use of digital signature in asymmetric cryptographic algorithm also enables a receiver to find out if a message was altered in transit [13]. A digitally signed message cannot be modified without invalidating the signature. In cryptography, the higher the size of the key length, the more secure the algorithm is. This also brings a major advantage to asymmetric cryptographic algorithm since it has a longer key length and therefore, makes it attack resistant. Comparatively, speed is a major drawback in asymmetric cryptography due to the complexity of its mathematical computations. There is a trade-off between security and speed in asymmetric cryptographic algorithm [11]. This study, therefore, uses the asymmetric cryptographic algorithm due to its obvious advantage. So long as we continue to communicate in an untrusted medium like the internet, security remains the topmost priority.

## 2.3 Attacks on Steganographic Systems

Most steganographic systems designed for confidential communication has suffered some weaknesses. [14] opined that steganographic attacks comprise of detecting, extracting and destroying the hidden data within the covert media. Visual attacks and statistical attacks [15] are the two widely known attacks against steganography. Statistical attacks use steganalysis [14]. [16] Developed a steganalysis application that was successful in detecting a message embedded in an image. Statistical video steganalysis developed [17] was also successful in detecting a data hidden in a video whose algorithm was based on LSB. Because of the fear of terrorists using steganography to communicate over the internet, [18] came out with a steganalysis called the active warden approach that was capable of detecting embedded messages in images and videos. [19] Showed that the human eye is capable of detecting hidden messages due to distortion. From the attacks above, it is obvious that steganography itself is not an end to the security concern associated with data transfer or communication. In order to mitigate the attacks against steganography and to further strengthen data communication security, cryptography was introduced.

## 2.4 Android Smartphones and Devices

Modern advancement in communication technologies has resulted in the widely and increase in use of smartphones such as android, blackberry, iPhones and much more. Phones such as HTC, Nokia, Sony Ericson, Apple, Samsung, Motorola, and others are all smartphones produced by manufacturers in the technology industry. Android is a software bunch, comprising not only the operating system but also middleware and key applications. The flexibility, easiness and less complex nature of android has made it the most preferred operating system over windows and iOS. Because of the availability and popularity of smartphones, data sharing using such devices has also become popular. These have resulted in threats which make smartphones easy prone to attacks by crackers and malware. Nonetheless, the secrecy of data that is transmitted on android based smartphones can be achieved using steganography and cryptography [1].

## 2.5 Related Works

[20] Proposed a copyright protection for android smart devices. In their approach, an android phone was used to capture images; the images were uploaded onto the internet with the copyright information in it. The copyright information was automatically embedded into the pictures with watermark technology when the pictures are taken. [21] Developed an android based steganography. In this approach, an android smartphone was used to capture the image. An application developed in eclipse IDE was used to hide the file in the picture through the process of LSB insertion. The captured image by the camera is compressed and saved to the SD card of the phone for the steganography process to continue. [3] Came up with a new android smartphone based steganography. In their proposed method, an application named SmartSteg that works on android platforms were developed. the application hides and encrypt messages using digital images. An LSB insertion algorithm was used to hide the message whiles the messages were encrypted using symmetric key cryptography. In the end, the application achieved high processing speed. The application used LSB and BMP image files. [22] Proposed android steganographic based application that works in smart phones environment. In their method, an android phone was used to capture an image. A message was embedded in the image using LSB embedding algorithm. [23] Also proposed a novel MMS steganographic application for android smart phone devices. The application was developed to be imperceptible and robust to message loss. A sender of the messages chooses an image in which the message is to be embedded. [24] Proposed an LSB steganography for android smart devices. Perhaps, this application is the most closely related method to the approach proposed in this study. The authors combine the use of BPCS, RSA and LSB algorithm in their approach. [25] Proposed an implementation of steganography based on android smartphone platforms. The application was developed using Android Development Tools (ADT) with the Integrated Development Environment (IDE) provided by eclipse. The application combines both android and MATLAB since the two platforms are compatible the authors first encrypted the message before it was hidden in the cover image.

## 2.6 RSA Algorithm

RSA is an algorithm used by modern computers to encrypt and decrypt messages. It is an asymmetric cryptographic algorithm. RSA stands for Ron **Rivest**, Adi **Shamir**, and Leonard **Adleman**, who first publicly described it in 1978. RSA algorithm is an asymmetric cryptographic system that utilizes two set of keys to encrypt and decrypt messages to ensure the security of quality information. In its performance, the keys are generated through a process of complex mathematical computation. The two keys generated are called public key and private key. The public key is distributed to the sender of a message to encrypt the message whiles the receiver of a message keeps the private key secretly to decrypt the public key encrypted message.

The steps below are the processes in generating public and private keys using RSA

1. Pick two large prime numbers  $p$  and  $q$ ,  $p \neq q$ ;
2. Calculate  $n = p * q$ ;
3. Calculate  $A(n) = (p-1)(q-1)$ ;
4. Pick  $e$ , so that  $\gcd(e, A(n)) = 1, 1 < e < A(n)$ ;
5. Calculate  $d$ , so that  $d * e \bmod A(n) = 1$ , i.e.  $d$  is the

Multiplicative inverse of e in mod A (n);

6. Get public key as  $K_u = \{e, n\}$ ;

7. Get private key as  $K_r = \{d, n\}$ ;

## 2.6 LSB Algorithm

LSB stands for Least Significant Bit. There are basically two methods of concealing messages in an image: Least Significant Bits and Discrete Cosine Transform. LSB belongs to the spatial domain whereas the DCT falls in the category of a frequency domain. The simplest and easiest method to implement in image steganography is LSB. In LSB, there is the encoding of the data to be hidden since the individual pixels of the least significant bits of the image are modified. Using an image of 8bit, the Least Significant Bit, thus the last bit is the 8<sup>th</sup> number bit of each byte of the carrier image becomes the bit which is considered as the secreted message. For 24 bit image, the colors of the each component such as the Red, Green, and Blue (RGB) are changed.

For example: Assuming cover images has two- pixel values as (1010 0000 0010 0011 0100 0111) and (0101 1111 0011 1100 0111 1100). Let's also assume the secret bits are 110111<sub>2</sub>, immediately the secret bits are embedded, the pixel values also change. That pixel values are: (1010 0001 0010 0011 0100 0110) and (0101 1111 0011 1101 0111 1100). The underlined bits indicate the bits changed from the original value and only three bits in the carrier image get changed.

## 3. METHOD

The system proposed in this study uses a cover object to hide the message to be sent. To be precise, the use of an image is adopted as a means of providing secrecy to the secret file to be sent. Until the authorized recipient undergoes some required steps to reveal the contained message, the content of the carrier file remains unnoticeable. The fundamental difference that distinguishes this system from any other system is the ability of the proposed system to hide quality information from unauthorized persons. The proposed system is basically divided into two main categories: Cryptography and Steganography. Before a message is embedded in the image, the message is first encrypted using RSA encryption algorithm. Encryption is the first phase of the proposed system which involves the converting of a secret message into binary data. There are basically four processes involved in RSA algorithm. The processes are a key generation, key distribution, encryption and decryption. After the message has been encrypted, the process of embedding or hiding the message in the image is carried on. Least Significant Bit (LSB) technique is used to embed the message into the image. In this procedure, the appropriate frame is determined and selected based on the histogram values of the frames. The message is therefore embedded using the LSB method. This procedure ensures double security based on the assumption that, if an unauthorized individual extracts the message from the video, the message cannot be read.

### 3.1 Proposed Model

The complete process of the model consists of seven main steps. The first step as demonstrated by the model is the

generation of public and private keys for encryption. The second step is the encryption step, which involves employing standard encryption algorithm, in this case, RSA algorithm to encrypt the file into binary data. In the third step, the appropriate frame is selected and the process of embedding is carried out using LSB insertion to hide the file in the image as the fourth step. The fifth step involves the sending or transmission of the secret file to the intended recipient. The file is normally sent over a communication channel mostly the internet. In the sixth step, the file is extracted from the hidden frame and the seventh step applies the process of decryption using the private key to obtain the original file. Figure 1 depicts the proposed model.

### 3.2 Performance Analysis

The ability of the proposed system to withstand statistical attacks and to also illustrate the robustness and security of the application are determined by calculating the PSNR. The Peak-Signal-to-Noise-Ratio normally abbreviated as PSNR is a terminology in Engineering that measures the ratio between the maximum possible power of a signal and the power of the corrupting noise that affects the representation of the signal. The PSNR is the most appropriate and often used parameter to measure, calculate and calibrate the quality of the reconstructed image in steganographic applications. In this case, the signal is the original image, the noise is the introduction of error by the steganographic algorithms. The PSNR is often computed using the Mean Square Error (MSE). The MSE is for two M\*N monochrome images I and K where one of the images is considered noisy approximation of the other. The scientific formulae used in calculating PSNR and MSE are as follows:

#### COMPUTATION OF MSE

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2$$

#### COMPUTATION OF PSNR

$$\begin{aligned} PSNR &= 10 \cdot \log_{10} \left( \frac{MAX_I^2}{MSE} \right) \\ &= 20 \cdot \log_{10} \left( \frac{MAX_I}{\sqrt{MSE}} \right) \\ &= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE) \end{aligned}$$

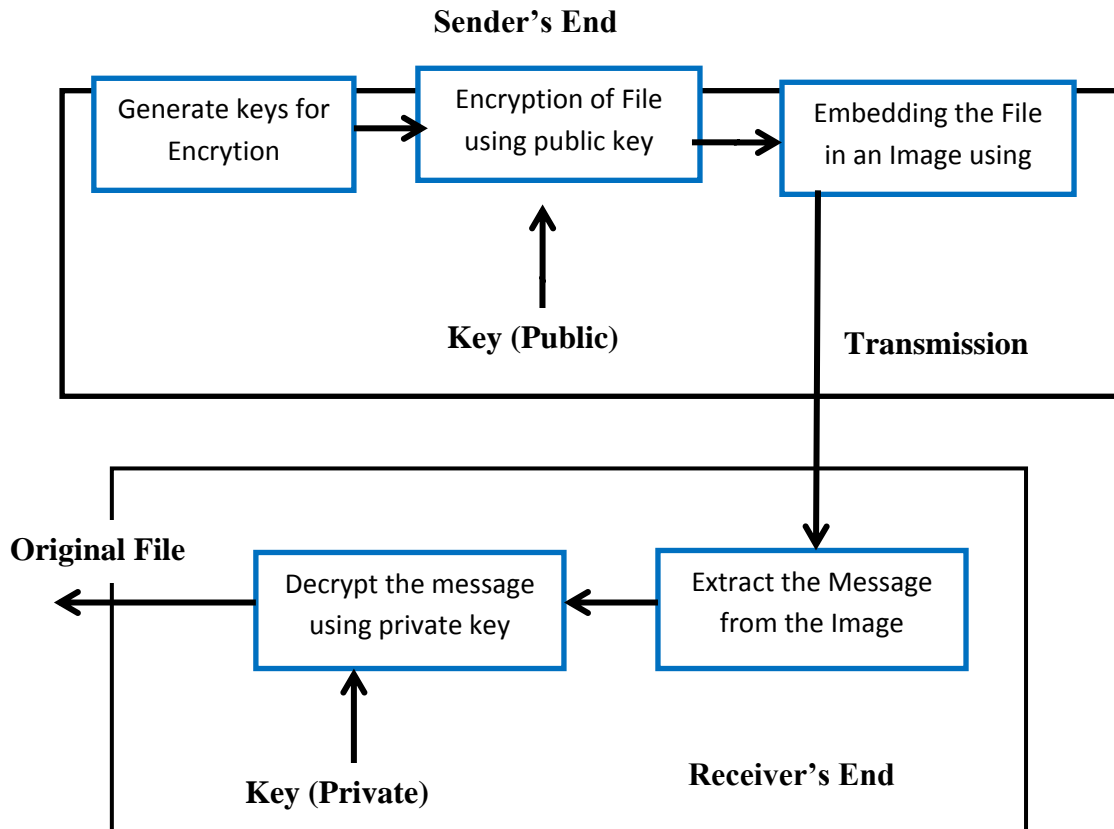


Figure 1: The Proposed Model

#### 4. DESIGN AND IMPLEMENTATION

In scientific and engineering applications such as developed in this research, engineering processes and procedures that are standard are followed. Approaches such as waterfall model, incremental model, and re-use oriented all exist. With respect to the development of test suite for this research, the engineering process adopted and chosen was the waterfall model. The process involves the development of Test Suite using Android in a computer lab in a step by step manner. Android currently is the most use operating system for the smartphone industry. The system proposed in this study was implemented using android SDK (Software Development Kit) and android studio IDE (Integrated Development Environment) as the main development tools. The fundamental principle underpinning the development of this application is to be made available on all devices that run the android operating system. As such the android SDK and the android studio provide such functions accordingly. The graphical user interface design was done using Java FX, which allows rich interface and graphical designs. The encryption and decryption algorithms are based on RSA algorithm whereas the messages were embedded using least significant bit (LSB) insertion algorithm. The source code for the application was developed using Android SDK supported by the Android NDK (Native Development Kit) and was run and compile on the Android Studio IDE. The minimum phone requirements to successfully run the application are: Minimum camera resolution is 2 megapixels, Minimum Processor speed is 512MHZ. In order to ensure that the system works with perfection, the test plan was carried on the android phone with the specification below:

MODEL	<b>ZTE AXON PRO</b>
INTERNAL MEMORY	<b>32/64 GB, 4 GB</b>
RAM	
ANDROID VERSION	<b>6.0</b>

#### 5. FUNCTIONAL REQUIREMENTS

The functional requirements specifications are the successful performance of the functions and the capabilities of the system. For the system proposed in this research, the functional requirements are:

- The system must allow a user take a picture with the camera or upload image from the phones or device gallery.
- The system should successfully allow a user to generate the keys for encryption.
- The system should allow a user perform the process of encryption using the public key generated.
- The system should allow a successful transfer of keys between the sender and the receiver.
- The application is expected to allow the embedding of the encrypted message in the image.
- The application should allow retrieval of information from the image at receivers end.
- The system should necessarily allow a user to decrypt encrypted message using the private key generated by the application itself.

#### 6. RESULTS AND DISCUSSIONS

The Fig 2 and Fig 3 illustrate the interface of the application after it has been launched. The three dots at the top right corner of Fig 2 represent the menu button of the App which when click gives menu items of the App (Fig 3). The menu items contains the steganography process, key generation process, sharing of embedded file, sharing of the app itself and also contain generation information about the application. Fig 3 enables users of the App to stay within the application to

do almost everything that needs to be done without closing the

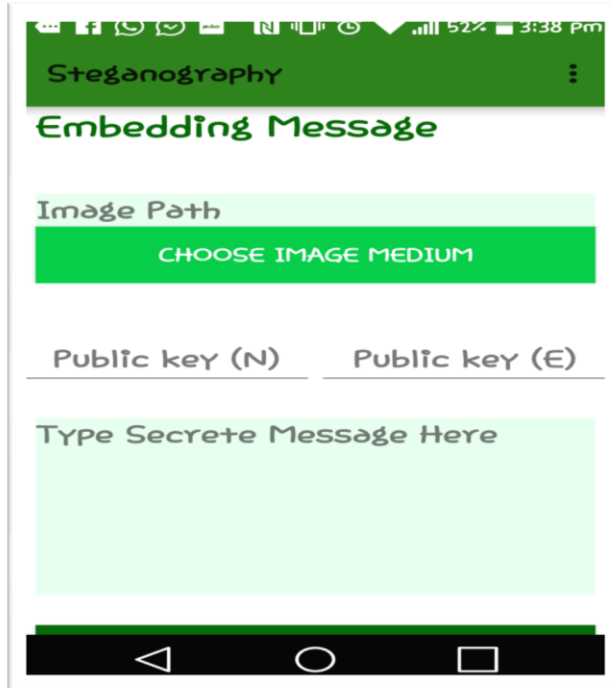


Figure 2: App Interface

### 6.1 RSA Key Generation

Two large prime numbers  $p$  and  $q$  are selected and with Fig 4, 17 and 13 were selected respectively and then the modulus  $n$  is calculated. According to RSA algorithm by multiplying  $p$  and  $q$  from the example in Fig 4 the modulus  $n$  which is 221 was obtained. The number  $n$  is used by both the public and the private keys (i.e. the encryption key is 221 and 7 and the decryption key is 221 and 55). The modulus  $n$  provides the link between them. Its length is usually expressed in bits and is called the key length. The private key  $d$  consist of the modulus of  $n$  and the private exponent  $d$  which is calculated using the Extended Euclidean algorithm to find the multiplicative inverse with respect to the totient of  $n$ . The RSA algorithm was used to masked the content of the file to be transmitted through the process of encryption.

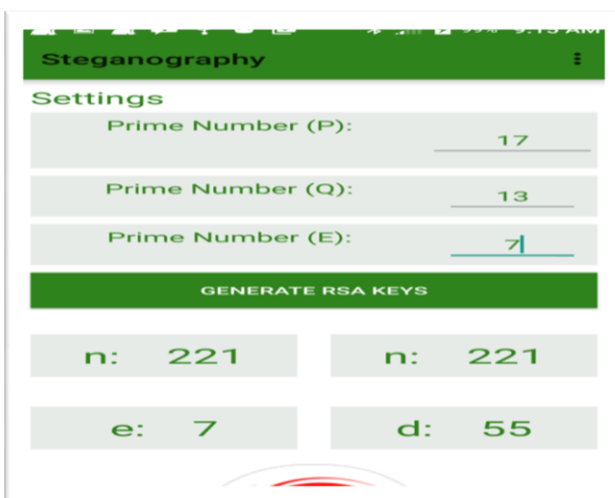


Figure 4: Key generation Page

application.

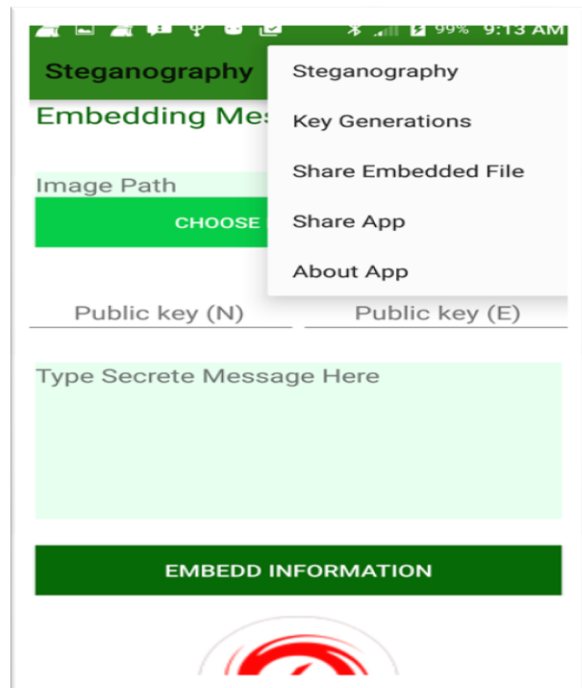


Figure 3: App Menu Items

### 6.2 Embedding and Encryption Process

During encryption, an image of a car was selected from the gallery as shown the Fig 5. Then the already generated public keys 221 and 7 were entered into their columns in the App, i.e. Public key (N) and Public key (E) respectively (Fig 6). After public keys entry, the supposed secret message to be sent: "the password for your account is 1234567890" was typed into space with inscription 'type secret message here'. A notification tab 'EMBEDD INFORMATION' which appears at the bottom of the secret message area is clicked for the embedding process to complete (Fig 7). After embedding process is complete, the encrypted file (image) can be shared to a recipient via e-mail or any social media platform on Android (Fig 8).

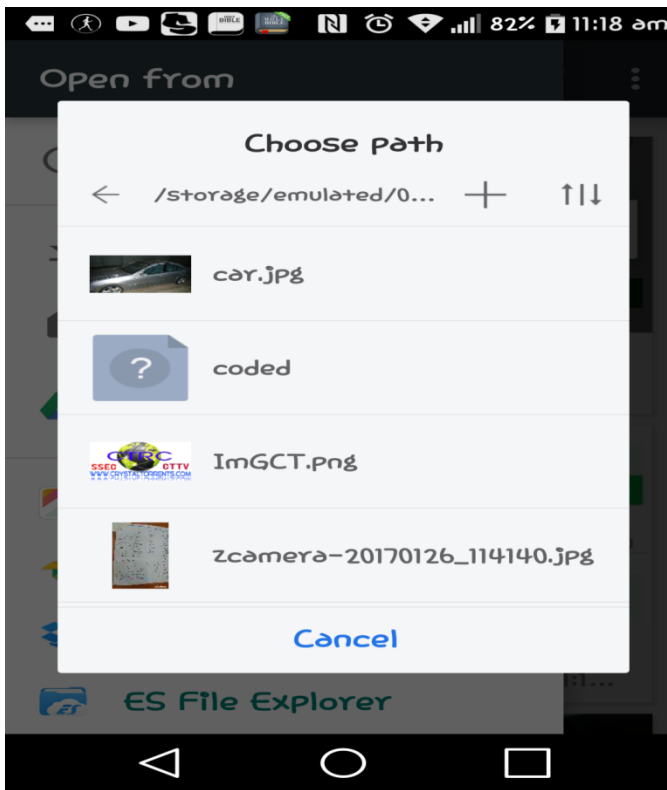


Figure 5: Image Selection for Embedding

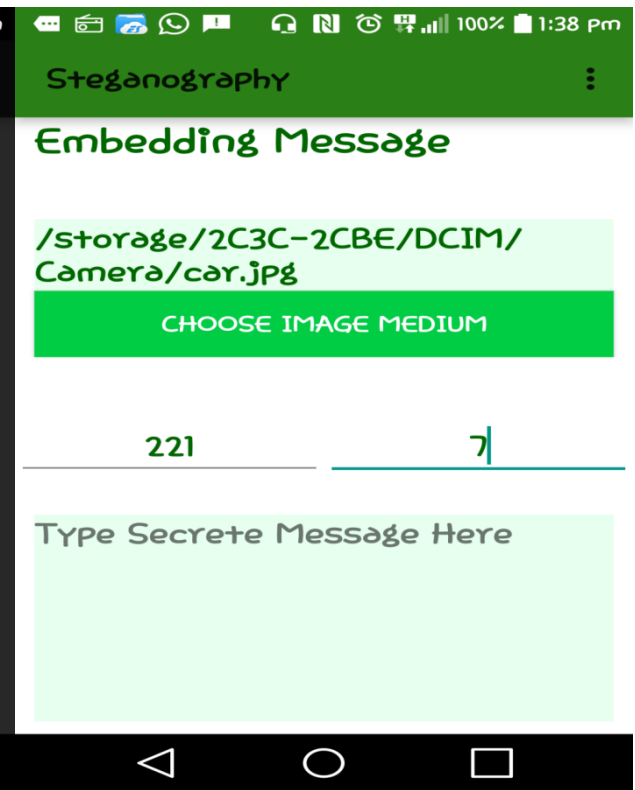


Figure 6: Public Keys Entry

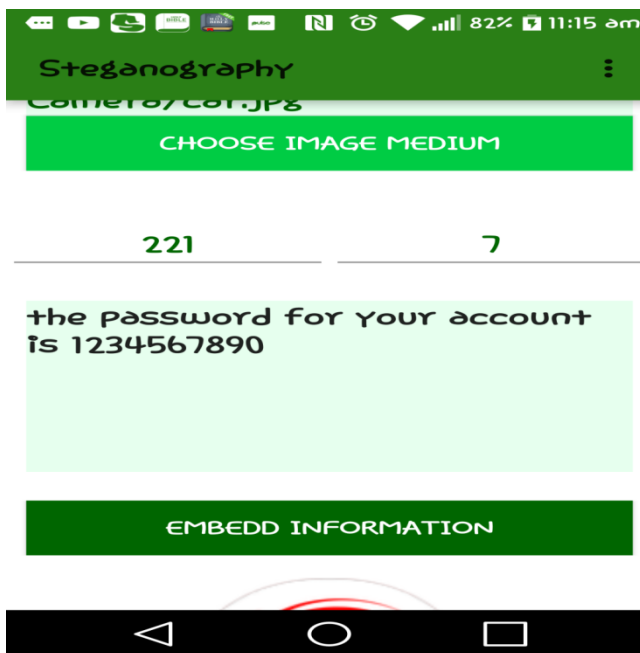


Figure 7: Embedding Information

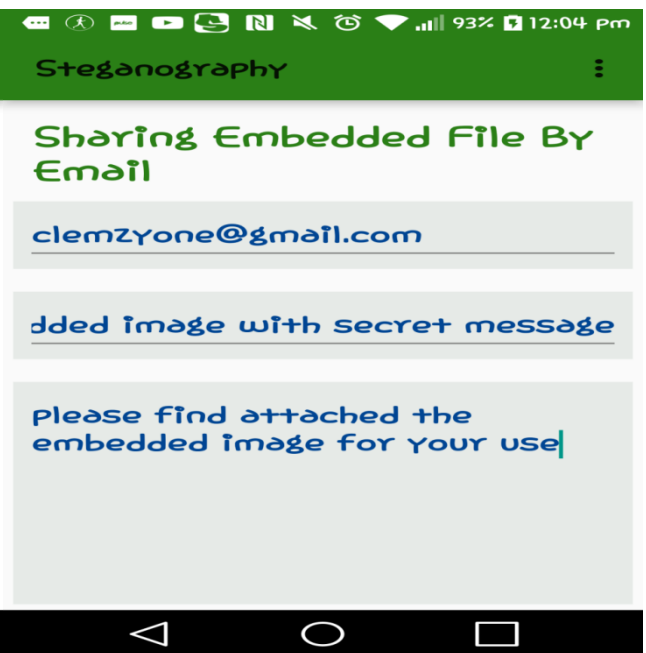


Figure 8: Sharing of Embedded File (image)

### 6.3 De-embedding and Decrypting Process

The recipient of the secret message first receives the image encrypted with the message on his/her android device via e-mail, WhatsApp, or any other social media platform. To retrieve the secret message, the receiver opens the steganography App on his Android device and scroll to the de-embedding section of the App (Fig 9) and click on the

'Image path EMBEDDED IMAGE' tab to select the file (image) from where it has been downloaded (download file) (Fig 11). After file selection, the private keys ( $N=221$ ,  $d=55$ ) are entered in their column (Fig 12) and the tab 'GET EMBEDDED MESSAGE' is clicked to enable de-embedding of the message from the image (Fig 13) to view the message at the recipient device (Fig 14).

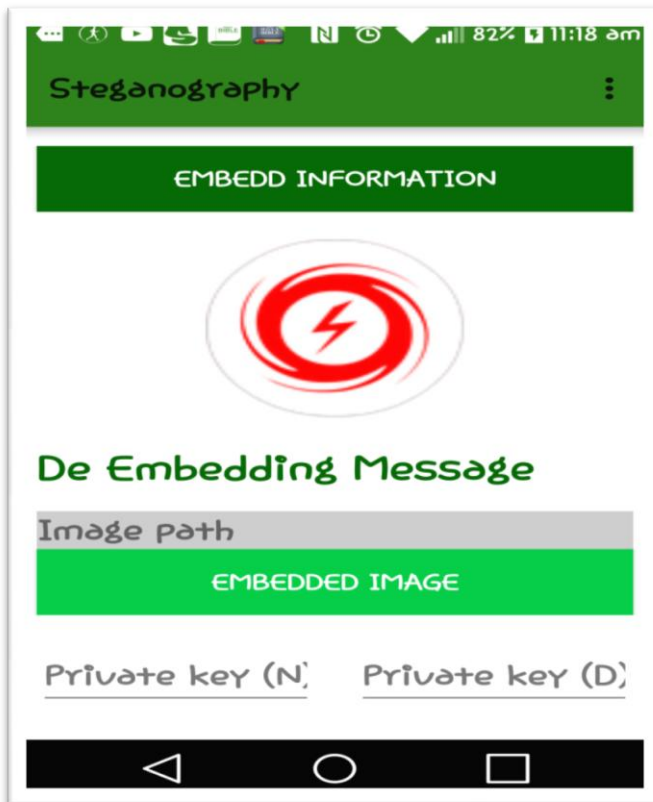


Figure 9: De-embedding Section

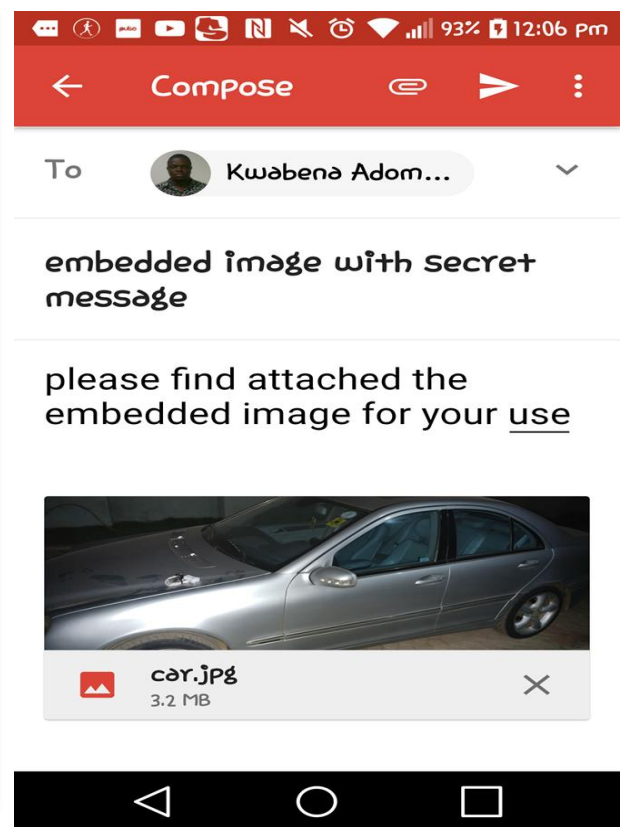


Figure 10: Recipient's inbox showing Image

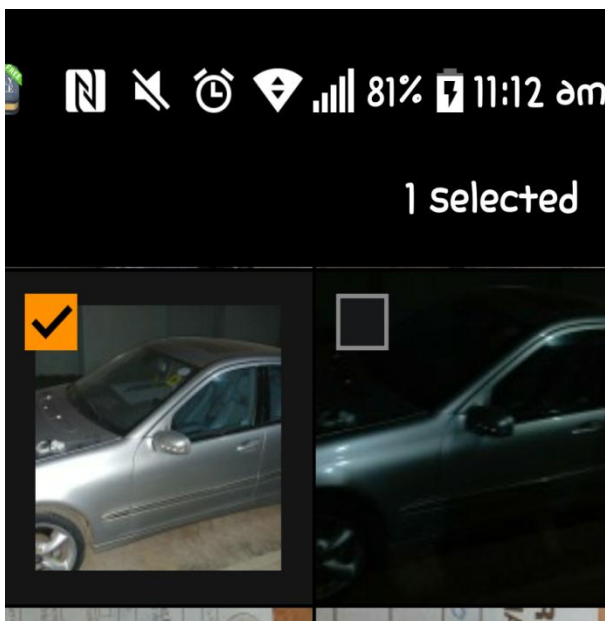


Figure 11: File Selection for De-embedding

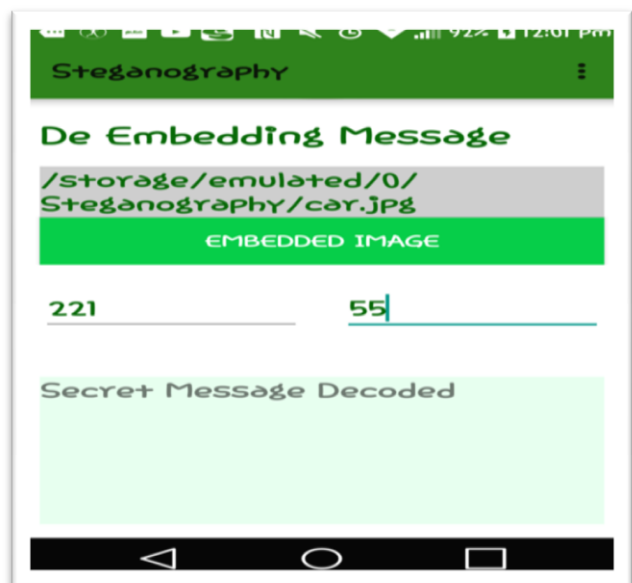


Figure 12: Private Key Entry



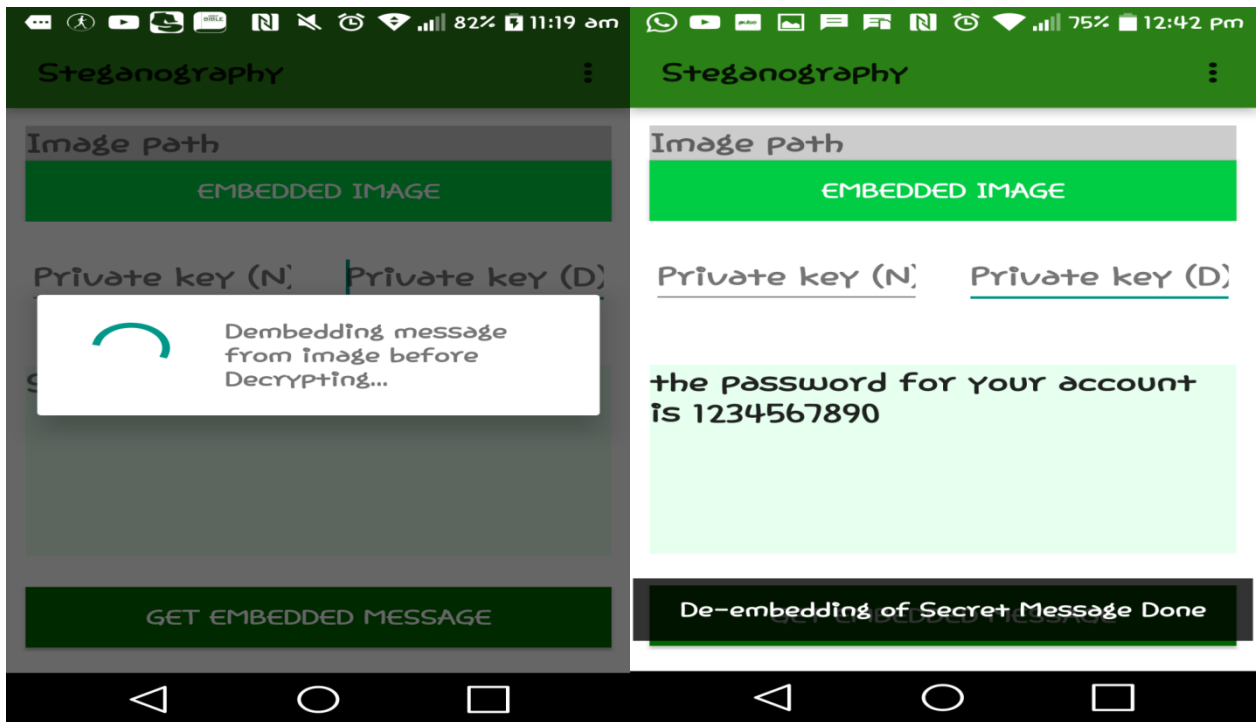


Figure 13: De-embedding Message from Image

Figure 14: App Showing Secret Message

#### 6.4 Validation Test

A key concern with the developed App for data transfer is whether an unauthorized person can gain access to data should he or she get access to the image with the encrypted message. The visual redundancy of an image is such that, the eyes normally do not care so much about the subtle change in color of some part of the image. The App is developed such that when the encrypted image is sent, until the authorized recipient follows the necessary steps and with the right private keys to reveal the contained message, the content of the carrier file remains unnoticeable. Hence, it is not possible for an intruder to finding the portions of the carrier image in which the data or the message is embedded without the

knowledge of private keys. To prove how authentic and secured the App could be, a series of trials with wrong keys necessarily close to the right private key to find out the result that will be displayed by the App. Since the right private key is 55, the developer chose the numbers 45 to decrypt to message (Fig 15). The message that was displayed by the App is an unreadable message that is widely different from the actual secret message embedded (Fig 16). The different and unreadable message display proves that a message or data transmission is secured with the App as a thorough knowledge of the private key is required for decrypting encrypted message or data.

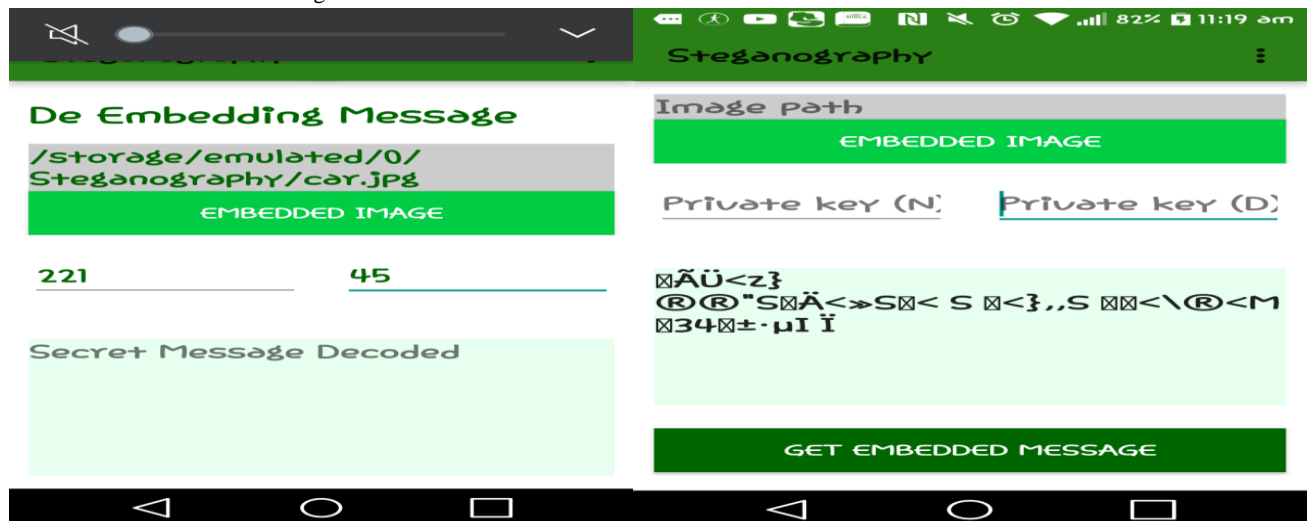


Figure 15: Entry of wrong private key (45)

Figure 16: Wrong Message Display





#### 6.5 Performance Evaluation

The study used three gray images which are 8-bit to verify the efficiency, security, and robustness of the proposed method. The three 8-bit images are **Koala.jpg**, **Lena.bmp**, **Rich.jpg**

and **Hannah.png**. The PSNR is calculated using file of a different number of characters (sizes). In calculating PSNR,  $MAX_i = 2^B - 1$ , where B is the bit per sample. So with 8-bit image as used in this study,  $MAX_i = 2^8 - 1 = 256 - 1 = 255$ , therefore PSNR can be rewritten as  $PSNR = 20 \cdot \log_{10}(255) -$

**10. Log<sub>10</sub>(MSE).** The MSE is calculated by plotting the pixel values of the cover image and the embedded image. After plotting, the regression line is drawn to obtain the linear regression equation and used to calculate the MSE. Table 1 indicates the properties or the characteristics of the images used for calculating the MSE and PSNR. Table 2 and Table 3 show the MSE variations and PSNR variations respectively against the various files sizes and images.

**Table 1: Characteristics of Images**

Images	Resolution	Image Type	Size
 <b>Koala</b>	1024*768	JPEG	762KB
 <b>Lena</b>	512*512	BMP	768KB
 <b>Rich</b>	640*960	JPEG	86.3KB
 <b>Hannah</b>	240*256	PNG	145KB

**Table 2: Variation of MSE with File Sizes**

Image Name	MSE values at File Sizes(Character)		
	1500 Chars	1050 Chars	850 Chars
<b>Koala</b> <b>(1024*768)</b>	0.0212	0.0197	0.0098
<b>Lena</b> <b>(512*512)</b>	0.0298	0.0286	0.0199
<b>Rich</b> <b>(640*960)</b>	0.0289	0.0279	0.0198
<b>Hannah</b> <b>(240*256)</b>	0.0899	0.0682	0.0399

**Table 3: Variation of PSNR with File Sizes**

Image Name	PSNR values at File Sizes(Character)		
	1500 Chars	1050 Chars	850 Chars
<b>Koala</b> <b>(1024*768)</b>	64.8674	65.1861	68.2185
<b>Lena</b> <b>(512*512)</b>	63.3886	63.5671	65.1423

<b>Rich</b> <b>(640*960)</b>	63.5218	63.6748	65.1642
<b>Hannah</b> <b>(240*256)</b>	58.5932	59.7930	62.1211

From table 2, the study achieved an average MSE of 0.0425 at file size of 1500 Chars, 0.0361 at 1050 Chars and 0.0221 at 850 Chars. The MSE average value for Koala is 0.0169, Lena is 0.0261, and Rich image achieved 0.0253 MSE whereas Hannah obtained 0.0660 average of MSE. The results showed higher MSE values files of larger size and low MSE values for images with high resolutions. In all the study achieved an average MSE value of 0.0336. Table 3 shows that the study achieved PSNR value of 62.5928dB at file size 1500 Chars, 63.0553dB at 1050 Chars and 65.615dB at 850 Chars. Koala obtained an average of 66.097dB of PSNR, Lena achieved 64.0327dB, Rich obtained 64.1203dB and Hannah obtained an average PSNR of 60.1691dB. The results show that, the larger the file size the lower the PSNR and vice versa. Images of high pixel resolution also achieved high PSNR values. The average PSNR value achieved in this study was 63.6032dB. Several steganographic applications have been proposed in the past. One of the most recent of such is the one developed by Thanikkal et al [1]. Thanikkal et al [1] used the method of LSB insertion and symmetric key cryptography with XOR algorithm and achieved an average PSNR value of 53.80dB. The 53.80dB is the calculated average value of all PSNR obtained in the study. An earlier one proposed by Gui et al [26] (as cited in Thannikall et al [1]) achieved PSNR average value of 40.53dB. The method proposed in this study used LSB for the embedding, RSA algorithm for encryption and achieved an average PSNR value of 63.6032dB.

## 7. CONCLUSION

Modern advancement in communication technologies has resulted in the widely and increase in use of smart phones such as android, blackberry, iPhones and much more. The proliferation of smartphones raises much security issues. This so because the security features of such devices is limited. The most novel approach to arrest the security challenges in the smartphone is cryptography and steganography. Cryptography concerns itself with the masking of the content of a secret message whereas steganography deals with the concealment or hiding of a secret message from an unauthorized person. The study draws our attention to the concept of efficiently utilizing steganography, which is image steganography and cryptography or asymmetric cryptography using RSA with LSB in android to protect the security of data. The choice of LSB embedding algorithm over any other existing algorithm is the capacity and the ability of LSB to ensure the security of smaller file size in steganography. From the obtained results, it can be concluded that high security and robustness is achieved in smart phones when RSA cryptographic algorithm is combined with LSB insertion algorithm in image steganography. The security of the proposed system is achieved by encrypting the message before embedding it in the image. Current steganographic applications in use hardly accept multiple image types; however, the results of the proposed system show that the system accepts different image file types. In the future, the version of the application would be multi-platform compliance, this will allow users of windows smart phones and iPhones to have access and use to the application.

## 8. REFERENCES

- [1] Thanikkal, J. G., Danish, M., & Sarwar, S. A. (October, 2014) New Android Based Steganography Application for Smartphone's. *Journal of Basic and Applied Engineering Research*. Print ISSN: 2350-0077; Online ISSN: 2350-0255; Volume 1, Number 8; pp. 32-35.
- [2] Savithri G, K.L.Sudha.( July 2014). Android Application for Secret Image Transmission and Reception Using Chaotic Steganography. *International Journal of Innovative Research in Computer and Communication Engineering* Vol. 2, Issue 7,
- [3] Bucerzan, D., Rațiu, C., & Manolescu, M. J. (2013). SmartSteg: A New Android Based Steganography Application. *International Journal of Computers, Communications & Control*, 8(5).
- [4] Jeon, W., J. Kim, Y. Lee, and D. Won, 2011. A Practical Analysis of Smartphone Security, In M.J. Smith, G. Salvendy (Eds.): *Human Interface*, Springer-Verlag Berlin Heidelberg, 311-320.
- [5] Apau, R., Hayfron-Acquah, J.B., and Twum, F. (June 2016). Enhancing Data Security using Video Steganography, RSA and Huffman Code Algorithms with LSB Insertion. *International Journal of Computer Applications*, 143 (4), 28-36.
- [6] Shelke, M. F. M., Dongre, M. A. A., & Soni, M. P. D. (2014). Comparison of different techniques for Steganography in images. *International Journal of Application or Innovation in Engineering & Management*, 3(2).
- [7] Morkel, T., Eloff, J. H., & Olivier, M. S. (2005, June). An overview of image steganography. In *ISSA* (pp. 1-11).
- [8] Eltyeb E. and Elgabar, A (2013) "Comparison of LSB Steganography in BMP and JPEG Images", *International Journal of Soft Computing and Engineering (IJSCE)* ISSN:2231-2307, Volume-3, Issue-5.
- [9] Al-Vahed, A., & Sahhavi, H. (2011). An overview of modern cryptography. *World Applied Programming*, 1(1), 3-8.
- [10] Kessler, G. C. (2015). An overview of cryptography. <http://www.garykessler.net/library/crypto.html#purpose> (accessed 2015 November 11)
- [11] Garg, N., Yadav, P. (2014) Comparison of Asymmetric Algorithms in Cryptography, *International Journal of Computer Science and Mobile Computing*, IJCSMC Vol.3 Issue.4, pg. 1190-1196.
- [12] Salomaa, A. (2013). *Public-key cryptography*. Springer Science & Business Media.
- [13] Das, S., Das, S., Bandyopadhyay, B., & Sanyal, S. (2011). Steganography and Steganalysis: different approaches. *arXiv preprint arXiv:1111.3758*.
- [14] Bateman, P., & Schaathun, H. G. (2008). Image steganography and steganalysis. Department Of Computing, Faculty of Engineering and Physical Sciences, University of Surrey, Guildford, Surrey, United Kingdom, 4th August.
- [15] Pevný, T., & Fridrich, J. (2008). Detection of double-compression in JPEG images for applications in steganography. *Information Forensics and Security, IEEE Transactions on*, 3(2), 247-258.
- [16] Budhia, U., Kundur, D., & Zourntos, T. (2006). Digital video steganalysis exploiting statistical visibility in the temporal domain. *Information Forensics and Security, IEEE Transactions on*, 1(4), 502-516.
- [17] Qi, Q. (2013). A Study on Countermeasures against Steganography: an Active Warden Approach (Doctoral dissertation, University of Nebraska).
- [18] Westfeld, A., & Pfitzmann, A. (2000). Attacks on steganographic systems breaking the steganographic utilities EzStego, Jsteg, Steganos, and S-Tools—and some lessons learned Lecture notes in computer science. vol. 1768.
- [19] Chen, Y. H., & Huang, H. C. (2011, October). A copyright information embedding system for android platform. In *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2011 Seventh International Conference on* (pp. 21-24). IEEE.
- [20] Rughani, P. H., & Pandya, H. N. (2012). Steganography on ANDROID based smart phones. *International Journal of Mobile & Adhoc Network*, 2(2), 150-152.
- [21] Sivakumar, S. and B.Rajesh (2014). Steganography on Android Based Smart Phones. *International Journal of Computer Science and Mobile Computing*. Vol. 3, Issue. 5, pg.1051 – 1054
- [22] Srinivasan, A., Wu, J., & Shi, J. (2015, May). Android-Stego: a novel service provider imperceptible MMS steganography technique robust to message loss. In *Proceedings of the 8th International Conference on Mobile Multimedia Communications* (pp. 205-212). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
- [23] Rajashri Ghare, Pruthvi Bansode, Sagar Bombale, Bilkis Chandargi (2015). LSB Steganography Using Android Phone. *International Journal of Computer Science and Information Technologies*, Vol. 6 (2), 1027-1029.
- [24] Khushali Pandit and Varsha Bhosale (2015). Implementation of Location based Steganography on mobile Smartphone using Android Platform. *International Journal of Computer Science and Information Technologies*, Vol. 6 (3), 2606-2609.
- [25] Gui, X., X.Li, B.Yang, 2014. A high capacity reversible data hiding scheme based on generalized prediction-error expansion and adaptive embedding. *Signal Processing*, 98: 370-380.