International Journal of Advanced Technology in Engineering and Science
Vol. No.3, Special Issue No. 01, August 2015
www.ijates.com
ijates
ISSN 2348 - 7550

ENHANCING SECURE COMMUNICATION USING GENERALIZED RSA AND COLOR COMPONENT REPLACEMENT STEGANOGRAPHY

Vishnu G. Nair[1], Sminesh C.N.[2]

[1]M.Tech Student, [2]Associate Professor, Department of Computer Science, Govt. Engineering College, Thrissur, Kerala (India)

ABSTRACT

Security protocols are essential for communication over digital media and Internet. To provide secure communication, sender and receiver should exercise an efficient technique to convert original plain text message to an unintelligible format to everyone except the intended receiver. Cryptography and Steganography are the two popular techniques to provide secure communication, where Cryptography distorts the message and Steganography hide the existence of the message. In this paper, the strength of Generalized RSA cryptosystem and Color component replacement steganography are utilized as a single system for enhancing security of secret information, and then comparing its performance with the existing systems. Generalized RSA is the generalization of RSA variants using Jordan's Totient function, which is the generalized function of classical Euler's function. RSA implementation can be speed up by using Jordan's Totient function instead of Euler's function. Color component replacement Steganography hides the secret data in color components of cover images. It is the enhanced version of Least Significant Bit replacement Steganography with minimum quality degradation of cover images. The proposed system hides the cipher text generated by Generalized RSA cryptosystem in color components of cover images using Color component replacement Steganography.

Keywords: ***Color Component Replacement, Cryptography, Generalized RSA, Jordan's Totient function, LSB***

I. INTRODUCTION

Security protocols are a must for the secret communication between two parties. Now a days we need secrecy in all the electronic communication areas like personal communication, military purposes, financial transactions, electronic banking, medical diagnosis etc. To attain security in these communications, the commonly used techniques are Cryptography and Steganography.

Cryptography ensure the security by encrypting the plain text into 'Cipher text' form by using cryptographic algorithms and secret keys. The cipher text is send from sender to receiver side. Unauthorized user cannot understand the actual plain text message from cipher text without knowing the secret keys. At the receiver's side, by using decryption algorithm and secret keys the receiver decrypts the cipher text and obtains the plain text/secret message. Steganography ensure the security of secrets by hiding them within the cover files. So the

messages cannot be seen by the unauthorized user. Steganographic algorithm embeds the plain text into the cover files and obtain the 'Stego files'. These stego files are sending from the sender to the receiver. The authorized receiver knows that the secret is present in the stego file and he can extract the actual message from stego file using proper steganographic algorithms and secret keys.

In this project, the features of Cryptography and Steganography are utilized as a single system. The main areas involved in this system are Cryptographic algorithms and Manipulation of cipher text. These two areas should be managed properly by an efficient cryptosystem.

One of the most popular and classical cryptosystem is RSA cryptosystem[5]. There are number of variants of RSA cryptosystem, we find out some of them and list out their properties and limitations. Finally most of these limitations can be resolved using an algorithm which uses Jordan's Totient function for the computations. Since the Jordan's Totient function is the generalization of Euler's Totient function, we call this algorithm as 'Generalized RSA Cryptosystem'[1].

By using Generalized RSA algorithm, plain text messages can be encrypted and decrypted securely. In conventional cryptosystem after encryption process, the generated cipher text is text format. The syntax and semantics of these text are completely different from conventional communication language. There is a possibility to an intruder to guess that there exist a secret within that text file. This limitation is overcame by hiding cipher text within the color components of color cover image files[2]. The proposed system uses the assumption that hiding data is better than sending it shown as encrypted. So we use color images are the medium for data hiding. In color component replacement steganography, data is hiding in the RGB (Red, Green, Blue) color components in different manner [9].

II. PROPOSED SYSTEMS AND DESIGN

By combining the strength Cryptography and Steganography, the security of secret data communication can be enhanced[3]. The Strength of Cryptography is mainly depends on the security of the cryptographic key and the time required to break the key. This can be ensured using very large size key for encryption and decryption. But it leads to large requirements to the algorithm and complex computational issues [4]. So, to an efficient cryptographic system, the requirements of the algorithm should be simple and time required for encryption and decryption should be less. Also these features do not affect the security aspects of the cryptosystem.

In the case of conventional RSA cryptosystem, the security is mainly depends on key size and factorization problem of large numbers. To attain these goals, requirements of RSA and computational issues of RSA are high [7]. Here we require very large prime numbers for very large size key generation and complex computations for encryption and decryption.

Use of Generalized RSA Cryptosystem [9] resolves these limitations. Instead of using the classical Euler's Totient function it uses Jordan's Totient function[6] for all its computations. For the same set of prime numbers Generalized RSA generate larger sized key than conventional RSA cryptosystem. When we use very large prime numbers, it can generate very large sized secret keys. By utilizing the features of Jordan's Totient function the complex exponential computations are reduced to multiplication operations[8]. So the computational requirements RSA are also reduced.

2.1 Proposed System Design

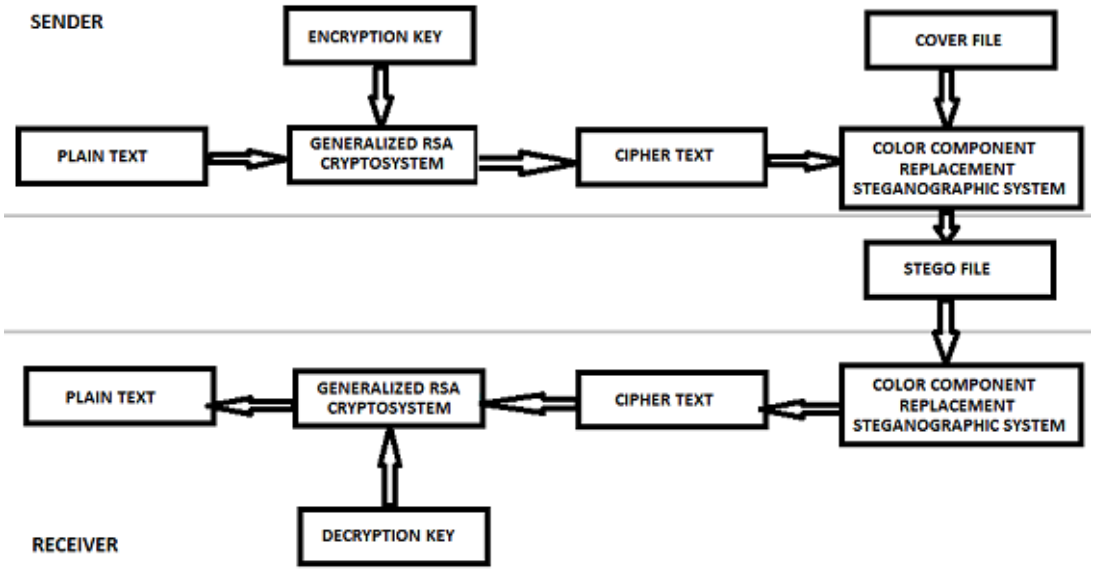

Fig. 1 System Design

The overall design of the proposed system can be depicted as shown in Fig. 1. Here plain text is encrypted using generalized RSA cryptosystm[1] and the obtained cipher text is processed with the steganographic module. Steganographic module takes cipher text and a color image file as inputs and embeds the cipher text into cover image using color components replacement steganography. Then, the obtained stego file is send to the receiver. Receiver extracts the stego file into cipher text and cover image file. The cipher text is decrypted using the decryption algorithm of generalized RSA cryptosystem and private key of the receiver. The entire system consists of Cryptographic module and Steganographic module.

2.2 Data Flow Diagram for Cryptographic Module

Fig. 2 shows the overall design of the Generalized RSA cryptosystem and it consists of following components.

- Jordan's Totient Function computing:

This module compute the value of Jordan's Totient function based on the user input[6]. In this stage user input are set of prime numbers and the generalizing index of Generalized RSA cryptosystem. The generated value is required for all the sub modules such as Key generator, Encryptor, and Decryptor.

$J_K(N) = N^K \prod_{P/n} (1 - P^{-K})$ Where K, N are positive integers 1

- Key Generator:

Key Generator is the module which generate the public key and private key for decryption and encryption.

- Select a random integer E such that, gcd $(E, J_K(N)) = 1$, where $1 < E < J_K(N)$, $E = M \bmod J_K(N)$
- Select integer D such that, $ED = 1 \pmod{J_K(N)}$ i.e., $D = E^{-1} \bmod J_K(N)$ where $1 < D < J_K(N)$

- Encryption:

Encryption module performs the encoding of plain text with public key. Output of encryption process is the cipher text. This cipher text is in unreadable form and Decryption process is required to make it readable. Given a public-key $(J_K(N);E)$ and a message M compute the cipher text

- $C = M * E \bmod J_K(N)$ 2

- Decryption:

Decryption module performs the decoding of cipher text with private key. Only the intended receiver can decrypt the cipher into readable plain text.

Given a private-key (J_K (N), D) and cipher text C, compute the message

- $M = C * D \bmod J_K (N)$ 3

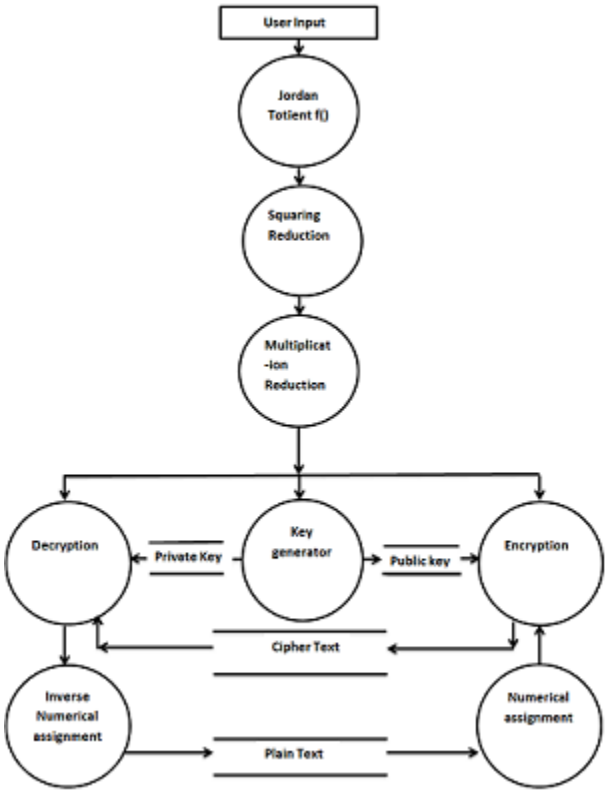

Fig. 2 Cryptographic Module

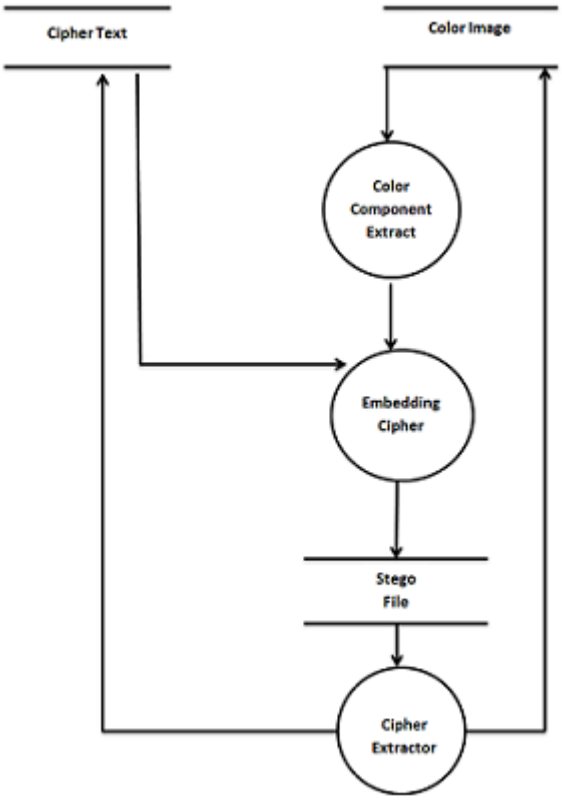

Fig. 3 Steganographic Module

- Numerical assignment:

This module assigns the numerical values to the characters. Since the public key cryptography is based on the mathematical functions, it is necessary to assign numerical values to characters before encryption.

- Inverse Numerical assignment:

This module performs the inverse mapping of numerical to character assignment. It is necessary after the decryption.

After the encryption process the generated cipher text is in text format[8]. Even though its syntax and semantics are differ from the natural languages, the intruder can assume the presence of something secrets in cipher text and he can try for the actual message. Here we uses the assumption that hiding data is better than sending it shown as encrypted. So we use Color Component Replacement Steganography for data hiding. This algorithm is an enhanced version of LSB technique[9][11]. Since the original LSB which is quite vulnerable, most common and well known method, hackers can easily try this method to retrieve the message[13]. So instead of hiding data within least significant bits of the pixel, data is hiding in the RGB(Red,Green,Blue) color components in different manner. The different methods are one color component complete replacement with one character of

cipher text, two color components of a pixel is replaced with 4 bits of a cipher text character in one component and remaining 4 bits in another color component, and at a time all color components altered with one cipher character's bit values.

The primary objective of steganography is to avoid drawing attention to the transmission of hidden information. If any attacker or hacker noted any changes in the stego file, then he can try for the inner contents. So the quality degradation of stego file should be prevent. The proposed color component replacement steganography is also a comparative study of, altering different color components, to ensure minimum quality degradation. The quality degradation of color images can be measured using the PSNR and here we maintain the PSNR above the allowable limit.

2.3 Data Flow Diagram for Steganographic Module

The above diagram (Fig. 3).represents the complete data flow of the steganographic module and it consists of following components.

- Color component extractor:

Color component replacement steganography is performed on color images by extracting color components of each pixels and replace these components with cipher text in different manner. The different methods are one color component complete replacement with one character of cipher text, two color components of a pixel is replaced with 4 bits of a cipher text character in one component and remaining 4 bits in another color component, and at a time all color components altered with one cipher character's bit values. A color component separator module is required to perform this function.

- Embedding Cipher:

This process embedded the cipher text character into the particular color components extracted by the color component extractor module. Red component of first pixel is replaced with binary of value of first cipher text character, Green component of second pixel is replaced with binary of second cipher text character, Blue component of third pixel is replaced with binary of third cipher text character and so on.

- Cipher extractor:

This module extracts the cipher text from the stego image. Input to this module is stego file and output from this module are cipher text and cover file. The technique used here is taking the cipher text, which is generated by the cryptographic module is taken as the input of steganographic module.

III. IMPLEMENTATION AND PERFORMANCE ANALYSIS

The proposed system design for cryptographic module and steganographic module are implemented in MATLAB. The performance of the proposed system can be analyzed using following parameters. In cryptographic module it can be measured using Key size, Message space size, Encryption time, and Decryption time etc. In Steganographic module it can be measured using cipher text capacity and Peak Signal to noise Ratio(PSNR). Since data is embedded into cover file is insertion of some noise into the actual cover, PSNR can measure amount of quality degradation of the cover file.

3.1 Cryptographic Module

3.1.1 Key Size

Key size determine the number of bits used to represent the encryption decryption keys. Strength of RSA mainly depends on Key size. For the same set of prime numbers Generalized RSA provide large key size than RSA. The largest and second largest prime numbers which can be represented using 2 bits are 3 and 2. By using these two prime numbers RSA generate the private key as 1 which is size of one bit. But for these prime numbers Generalized RSA with index value 2 generate the private key as 5 which is size of three bits. Below table (Table. 1) represents the key size generated from different size prime numbers. The values of table(Table. I)is plotted below(Fig. 4) which shows for the same set of prime numbers Generalized RSA provide large key size than RSA.

Table 1 Key Size

Key size				
	RSA		Gen. RSA	
Prime Size (bits)	Private Key	Key Size (bits)	Private Key	Key Size (bits)
2	1	1	5	3
3	5	3	461	9
4	103	7	7331	13
5	611	10	733091	20
6	2983	12	3698743	22
7	8811	14	82179491	27
8	17143	15	1407323077	31
9	170011	18	35754170531	36
10	890023	20	9.84025E+11	40
11	826613	20	6.22394E+12	43
12	4945177	23	1.05269E+13	44
13	47654443	26	2.54471E+15	52

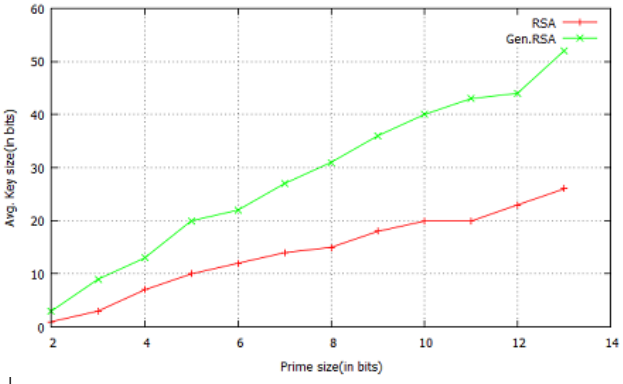

Fig.4 Avg. Key Size

3.1.2 Message space size:

It is the phi value generated and it is used for the modular operations. As the phi value increases size of message space also increases. Generalized RSA with indexing factor greater than one will give higher message space than RSA. The largest and second largest prime numbers which can be represented using 2 bits are 2 and 3.

For RSA, $\phi = (2 - 1) * (3 - 1) = 2$ Which can be represented using 2 bits

For Generalized RSA with index 2,

$\Phi = (2^2 - 1) * (3^2 - 1)$

$= 24$ Which requires 5 bits to represent

Below table (Table. 2) represents the phi value generated from different size prime numbers. One of the security aspects of RSA is factorization problem of larger numbers. Here with smaller requirements Generalized RSA provides larger values for phi(Fig. 5).

International Journal of Advanced Technology in Engineering and Science
Vol. No.3, Special Issue No. 01, August 2015
www.ijates.com
ijates
ISSN 2348 - 7550

Table 2 Message Space Size

Message Space size				
	RSA		Gen. RSA	
Prime Size (bits)	Phi value	Size of Phi (bits)	Phi value	Size of Phi (bits)
2	2	2	24	5
3	24	5	1152	11
4	120	7	20160	15
5	840	10	806400	20
6	3480	12	12945600	24
7	13216	14	1.81E+08	28
8	60000	16	3.66E+09	32
9	255016	18	6.55E+10	36
10	1038360	20	1.08E+12	40
11	4133064	22	1.71E+13	44
12	5844300	23	3.42E+13	45
13	66716220	26	4.45E+15	52

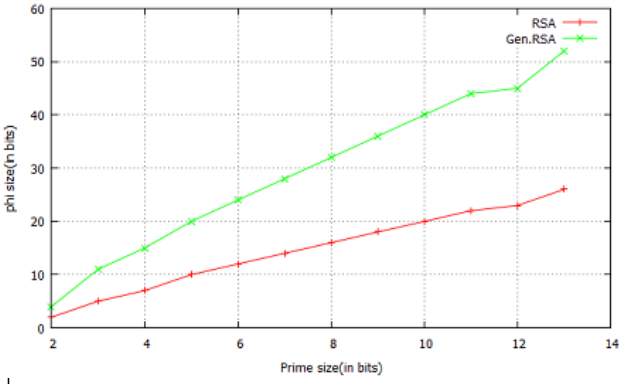

Fig. 5 Message Space Size

3.1.3 Encryption time:

Encryption time is the average time required to Encrypt the message into cipher text. The encryption and decryption time are measured in a computer system with following specifications:

- CPU Type: Intel Xenon E5
- Clock speed: 3.70 GHZ
- RAM Size: 16 GB
- Monitor Type: 15 inch color monitor

Generalized RSA is faster than RSA in the case of encryption (Fig. 6).

3.1.4 Decryption time:

Decryption time is the average time required to Decrypt the cipher text using private key. Generalized RSA require less time for decryption process (Fig. 7).

Table 3 Encryption And Decryption Time

	EncryptionTime (sec.)		Decryption Time (sec.)	
Data Size (KB)	Gen. RSA	RSA	Gen. RSA	RSA
1	0.0002	0.1406	0.0469	0.0938
3	0.0313	0.1563	0.0938	0.2031
5	0.0313	0.3125	0.1406	0.3594
9	0.0469	0.5469	0.3125	0.7031
14	0.0313	0.8438	0.4375	1.0156
18	0.0469	1.0625	0.5938	1.3125
36	0.125	2.0781	1.1719	2.5781
41	0.1406	2.25	1.2813	2.8125
81	0.25	4.6406	2.6563	5.875
121	0.3438	6.7656	3.7813	8.6406
202	0.625	11.1563	6.3438	14.25
242	0.7188	13.4844	7.5	17.3906

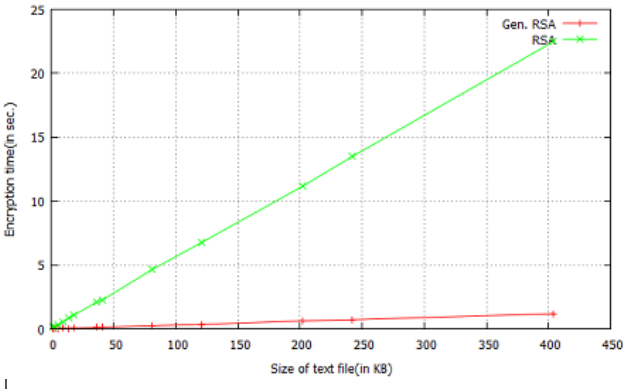

Fig. 6 Encryption Time

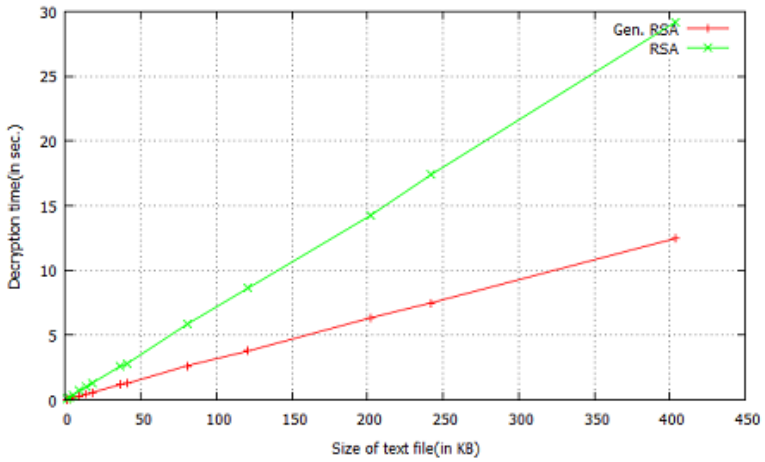

Fig.6 Decryption Time

3.2 Steganographic Module

3.2.1 PSNR

Peak Signal to Noise Ratio is used to measure the quality degradation of the stego image from the cover image. Since embedded cipher text is treated as noise to the image it measure the amount of noise added. It can be computed as shown below

$$PSNR = 20 \log_{10} (MAX_i) - 10 \log_{10} (MSE) \quad 4$$

MAX_i is the maximum possible pixel value of the image and MSE is the sum over all squared value differences divided by image size and by three. If PSNR(4) is greater than 60 decibel, human eye cannot detect the changes happened in the color images. So it is necessary to maintain the PSNR greater than 60 after embedding cipher to the cover image. In first case, One color component of one pixel is completely replaced with one character of cipher text. This improve the cipher capacity but it affect the quality of stego image, PSNR becomes less than 60 To solve this problem one character of the cipher text is embedded within two color components of a pixel by replacing 4 bits at a time. This method improves the quality of image and PSNR value. To get more better PSNR one character of cipher text is embedded within 3 color components of a pixel by replacing 2 bits of RED, 3 bits of GREEN, 3 bits of BLUE at a time. It improves the quality of stego image and maintain the capacity of 8 bits per pixel(Fig. 8)..

Table 4 PSNR Comparison

	PSNR Comparison		
Data Size (KB)	3Comp. Replacing (dB)	2Comp. Replacing (dB)	1Com. Replacing (dB)
2	80.957	76.78	65.8571
3	77.798	74.9606	64.71
5	74.6198	72.0389	61.33
9	71.6224	68.9249	58.243
14	69.933	67.259	56.8966
18	68.659	65.9363	55.4049
20	68.376	65.0249	54.0026
36	65.558	63.6843	52.3623
41	65.946	62.3306	51.9257
81	63.08	59.1425	48.9079
202	59	54.803	44.907
242	56.22	53.9224	44.1167

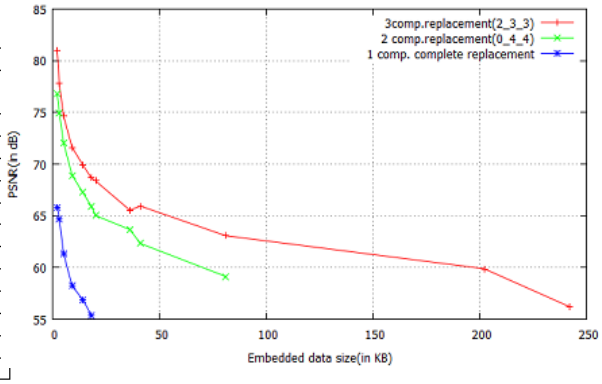

Fig. 7 PSNR Comparison

IV. CONCLUSION

Combining Cryptography and Steganography in communication can enhance the security. Generalized RSA reduces the requirements of RSA cryptosystem such as the requirement exponential computations for encryption and decryption. For the same set of prime numbers, Generalized RSA provide better key size and message space size than the conventional RSA. Generalized RSA is faster than RSA in the case of encryption and decryption. Color components replacement steganography gives more PSNR for very large size data if we replace less than four bits of a color component at a time. By combining Cryptography and Steganography and utilizing their features in the combined manner we can enhance the security of secret data communication. The work done in this project provides basis for future research in steganography and can be extended in several ways. One possible extension is to use cover video files for color component replacement steganography without degrading the quality of video. The possibility and the impact of such work needs to be investigated.

REFERENCES

[1] Vishnu G Nair and Sminesh C N, RSA Generalization using Jordan's Totient function, International Journal of Advanced Research Trends in Engineering and Technology(IJARTET), vol.2 issue X , March 2015, ISSN 2394-3777(Print), ISSN 2394-3785(Online)

[2] A Joseph Raphael and Dr. V sundaram, Cryptography and Steganography A Survey, International Journal for Computer Applications.Vol 2 (3), 626-630

[3] Khalil Challita and Hikmat Farhat, Combining Steganography and Cryptography: New Directions, International Journal on New Computer Architectures and Their Applications (IJNCAA) 1(1): 199-208, The Society of Digital Information and Wireless Communications, 2011 (ISSN 2220-9085)

[4] R.L. Rivest, A. Shamir, and L. Adleman, A Method for Obtaining Digital Signatures and Public-Key Cryptosystems,

[5] A.Shamir, How to check modular exponentiation, Eurocrypt 97, May 1997

[6] S. Thajoddin and S. Vangipuram, A Note On Jordans Totient function, Indian j. pure appl. Math December 1988

[7] Suresh K and Venkataramana.K, Study of Analysis on RSA and its Variants, International Journal of Computer Science Research & Technology (IJCSR),Vol. 1 Issue 4, September 2013

[8] E.Madhusudhana Reddy, B. Muneendra Nayak and M.Padmavathamma, Communication between two Parties using MJ2 -RSA Cryptosystem and Signature Scheme, IEEE CONECCT 2013

[9] Vipul Sharma and Sunny Kumar, A New Approach to Hide Text in Images Using Steganography, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 4, April 2013

[10] Vijay Kumar Sharma and Vishal Shrivastava, A Steganography algorithm for hiding image in image by improved LSB substitution by minimize detection, Journal of Theoretical and Applied Information Technology, 15th February 2012 Vol. 36 No.1

[11] Chi-Kwong Chan and L M Cheng, Hiding data in images by simple LSB Substitution, The Journal of the Pattern Recognition Society

[12] Abbas Cheddad, Joan Condell, Kevin Curran and Paul Mc Kevitt, Digital Image Steganography: Survey and Analysis of Current Methods, Signal Processing, Volume 90, Issue 3, March 2010

[13] H. Wu, H. Wang, C. Tsai and C. Wang, Reversible image steganographic scheme via predictive coding, Reversible image steganographic scheme via predictive coding 1 (2010), ISSN: 01419382, 35-43