Volume 5, Issue 3, March 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Image Authentication and Confidentiality Using Cryptographic Techniques

Santhosh Kumar B. J
M.Tech, Amrita Vishwa Vidyapeetham,
Mysore Campus, Mysore, India

Abstract: ***The transmission of secure images has become a daily routine, especially over wireless for the purpose of communication. Email is one of the most widely used and regarded network services. The sensitive and secure images and data can be read or modified during their transmission via a non-controlled channel such as the internet. These images can be captured by an opponent during transmission using passive or active attacks. Hence it is necessary to make these data unreadable and indecipherable during their transfer. The data can be made unreadable by using symmetric or asymmetric cryptographic techniques. In this approach, asymmetric cryptographic technique RSA is used for key exchange and encryption. second technique is PGP, widely used de facto secure email standard. A software implementation has the advantage of being portable and low cost.PGP is an general purpose application to protect(encrypt and or sign) files. Advantage of using PGP is it can be used by corporations and as well as individuals. PGP is a personal high-security cryptographic software application that allows people to exchange messages or files with privacy, authentication, and convenience.***

Keywords: RSA,PGP,Active and passive attacks.

I. INTRODUCTION

Email is one of the most widely used and regarded network services for communication. Digital images are being used in every field. Therefore the need for distribution of digital images over networks has become an essential part of everyday life. Defense or Medical or organizational applications etc, often deal with sensitive information that are confidential and should only be accessible to authorized persons. Authentication and security of data, storing and sharing images secretly is a challenging task. It specifically includes ensuring the security of the confidential data during electronic transmission, and that access is limited only to authorized personnel. Therefore images are needed to be encrypted which is done using cryptographic techniques.

There are two types of cryptographic techniques symmetric key cryptography and asymmetric key cryptography. In symmetric key cryptosystems, the same key is used for the encryption or decryption and this key need to be secure and must be known to both the sender and the receiver. Asymmetric key cryptosystems rely on two keys: one key (public key) for encryption and a different but related key (private key) for decryption. With the receiver's public key, the sender encrypts the message and sends it to the receiver who decrypts the message with his private key.

The first and the most widely accepted public-key cryptosystem is the RSA whose security depends on the difficulty of discovering the private key in a reasonable time but not on the details of the algorithm.

Steganography is the art of concealing information in ways that prevent the detection of hidden messages. This is accomplished through hiding information in other information, thus hiding the existence of the communicated information.

ex: **Audio/Video Steganography** is the technology of embedding information in an audio/Video channel.

Watermarking is the technique which hides one piece of information [message] in another piece of information [carrier]

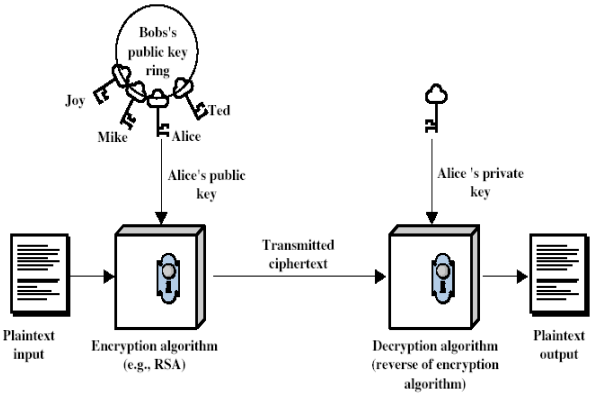

Fig-1 Asymmetric key cryptography

Here we are using LSB steganography for hiding the confidential details in the image.

RSA (Named after its three inventors-Rivest Shamir Adelman) widely used encryption technology which is based on a mathematical algorithm and utilizes both public and private keys for encryption/decryption.

It is based on two mathematical principles: factoring large integers and modular arithmetic. It is based on exponentiation in a finite (Galois) field over integers modulo a prime.

Exponentiation takes $O((\log n)^3)$ operations (easy)

The difficulty of factorizing a large integer number into two prime numbers makes this cryptosystem secure . Actually, keys of 1024 bits to 2048 bits are commonly used.

factorization takes $O(e^{\log n \log \log n})$ operations (hard)

Key Generation	
Select p, q	p and q both prime
Calculate n	$n = p \times q$
Select integer d	$\gcd(\phi(n), d) = 1; 1 < d < \phi(n)$
Calculate e	$e = d^{-1} \bmod \phi(n)$
Public Key	$KU = \{e, n\}$
Private Key	$KR = \{d, n\}$
Encryption	
Plaintext: $M < n$	
Ciphertext: $C = M^e \pmod n$	
Decryption	
Ciphertext: C	
Plaintext: $M = C^d \pmod n$	

Fig-2 RSA key generation, Encryption and Decryption

In the RSA, it contains the modulus n which is a large integer number, a product of two prime numbers p and q, whose bits length is the key size. These numbers p and q must be kept secret otherwise the private key can be calculated and the protocol will be broken

The public and private keys are two numbers e and d associated with n. e is random number between 2 and φ(n) and relatively prime with n. Where φ(n) is the Euler totient function which is defined as the number of positive integers less than n and relatively prime to n. For p, q prime, φ (pq) = (p-1) (q-1). Then d can be calculated as $d \equiv e^{-1} \bmod \varphi(n)$. The Extended Euclidean algorithm is used to instantly calculate the inverse d. The pair (e, n) is the public key and (d, n) is the private key.

RSA Encryption and decryption are of the following form, for some plaintext block M and cipher text block C:

The message M must be smaller than the modulus n otherwise it must be divided into blocks of length less than n. The plain text is encrypted using the public key (e, n) by computing the modular exponentiation to get the cipher text $C = M^e \bmod n$. Decryption is done using the private key (n, d) and computing the modular exponentiation to recover the plain text $M = C^d \bmod n = (M^e)^d \bmod n = M$. Since $e*d \bmod \varphi(n) = 1$.

II. LSB STEGANOGRAPHY

LSB steganography is simplest of all the image steganography techniques. To a computer, an image is a collection of numbers that constitute different light intensities in different areas of the image. This numeric representation forms a grid and the individual points are referred to as pixels. The number of bits in a colour scheme, called the bit depth, refers to the number of bits used for each pixel. The least significant bit (in other words, the 8th bit) of some or all of the bytes inside an image is changed to a bit of the secret message.

For example a grid for 3 pixels of a 24-bit image can be as follows:

(00101101 00011100 11011100)
(10100110 11000100 00001100)
(11010010 10101101 01100011)

When the number 200, represented by the binary equivalent of 11001000, is embedded into the least significant bits of this part of the image, the resulting grid is as follows:

(00101101 00011101 11011100)
(10100110 11000101 00001100)
(11010010 10101100 01100011)

Although the number was embedded into the first 8 bytes of the grid, only the 3 underlined bits needed to be changed according to the embedded message. Therefore On an average, only half of the bits in an image will need to be modified to hide a secret message using the maximum cover size [6]. Since there are 256 possible intensities of each primary colour, changing the LSB of a pixel results in small changes in the intensity of the colours. These changes cannot be perceived by the human eye - thus the message is successfully hidden.

III. SOFTWARE IMPLEMENTATION

The software implementation of the RSA encryption/decryption and the LSB steganography was done using the MATLAB and the symbolic math toolbox.

The implementation consists of five steps as shown in the Fig. 1.

A. Generation of public/private keys (n, e)/(n, d) consists of

1. Generating two random numbers p and q of N/2 bits, where N is the size in bits of the modulus n.
2. Calculating n and φ(n) using the equations
   $n = pxq$ and $\varphi(n) = (p-l)x(q-1)$.
3. Randomly generating e such that, e must be less than 64-bits to reduce the image encryption delay and e must be prime with φ(n).
4. Finally, d is calculated such that:
   $d = e^{-1} \bmod \varphi(n)$.

B. Hiding sensitive details inside the image

consists of

1. Concatenating the sensitive details into a single long message and converting that into binary data.

Hiding the sensitive details inside the medical image using LSB steganography.

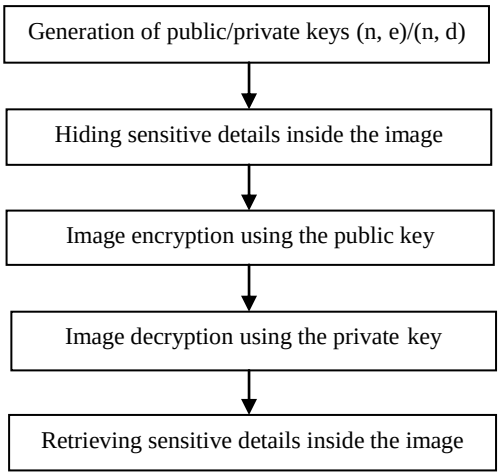

Fig-3 Steganography steps

C. Image encryption using the public key consists of

1. Reading the image to extract the pixel values which are converted into binary data which is then concatenated into a single message of numeric data.
2. Splitting the message into several smaller messages (M_i) of size of the modulus in bits.
3. Encryption of the split messages one by one by using the formula $C_i = M_i^e \bmod n$ to create a list of encrypted messages which are converted into pixels to view it as encrypted image.

D. Image decryption using the private key consists of

1. Reading the encrypted image to extract the pixel values which are converted into binary data which is then concatenated into a single message of numeric data.
   Splitting the message into several smaller
   messages (C_i) of size of the modulus in bits.
2. Decryption of the split messages one by one by using the formula $M_i = C_i^d \bmod n$ to create a list of decrypted messages which are converted into pixels to recover the original image.

E. Retrieving sensitive details hidden in the image consists of

1. Retrieving all the bits corresponding to information details hidden in the image.
2. Converting the binary data into meaningful data.

IV. ENCRYPTION/DECRYPTION OF IMAGES USING RSA

The software has been tested for the images with 256 gray levels and colour images. Monochrome and gray scale images use 8 bits for each pixel and are able to display 256 different colours or shades of grey. Digital colour images are typically stored in 24-bit files and use the RGB colour model, also known as true colour. All colour variations for the pixels of a 24-bit image are derived from three primary colours: red, green and blue, and each primary colour are represented by 8 bits. In Symbolic Math Toolbox, the operator "&^" uses the fast modular exponentiation algorithm. It was used for the calculation of the encryption and decryption operations of our software implementation of the RSA [7].

A. Image encryption

In image encryption the image in JPEG, BMP, TIF or any other format is converted to attain their pixel values which will be converted to binary values. These binary values are then translated to list of messages (M_i). Before

translation it is verified that the decimal numbers are represented using 8 bits if not zeros are appended to the MSB in order to represent in binary with 8 bits. Then these bits are concatenated into a single long numeric data which is then divided several blocks of size equal to the modulus size in bits. These blocks of data are encrypted to get a list of messages (C_i). If the last message in the list has lesser number of bits than the modulus then it is concatenated to the encrypted messages without being encrypted. If the encrypted messages have lesser number of bits than the modulus the zeros are appended to the MSB to make it equal to the size of modulus. This concatenated encrypted long single message is reconstructed as a matrix of pixels of 8 bits to be viewed as an image in MATLAB. Fig. 2(a) shows the image used for encryption and Fig. 2(b) shows the encrypted image using RSA with a key length of 1024 bits. The encrypted image will be of same size as the original image.

Communicating encrypted image over network using PGP

PGP is a personal high-security cryptographic software application that allows people to exchange messages or files with privacy, authentication, and convenience. PGP can be used to encrypt and digitally sign files and e-mail.

PGP provides five different services:-

1. Authentication
2. Confidentiality
3. Compression
4. E-mail compatibility
5. Segmentation and Reassembly.

PGP authentication procedure:

1. Sender creates message.
2. use SHA-1 to generate 160-bit hash of message
3. Signed hash with RSA using sender's private key,
and is attached to message.
4. Receiver uses RSA with sender's public key to
decrypt and recover hash code.
5. Receiver verifies received message using hash of it and compares with decrypted hash code

Message authentication is based on Digital signature(supported algorithms are RSA/SHA or DSS/SHA)

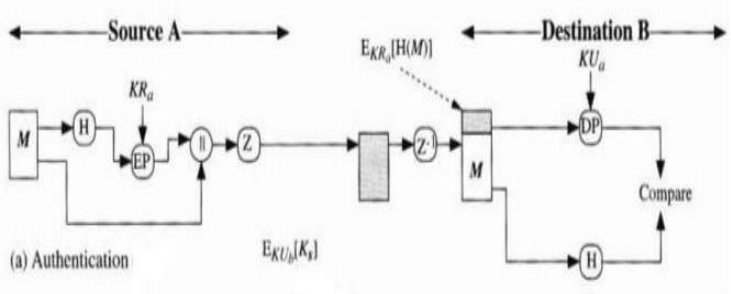

Fig-4 PGP Authentication and confidentiality procedure

The hash code of the message is created using SHA-1.This message digest is encrypted using DSS or RSA with the senders private key and included with the message.

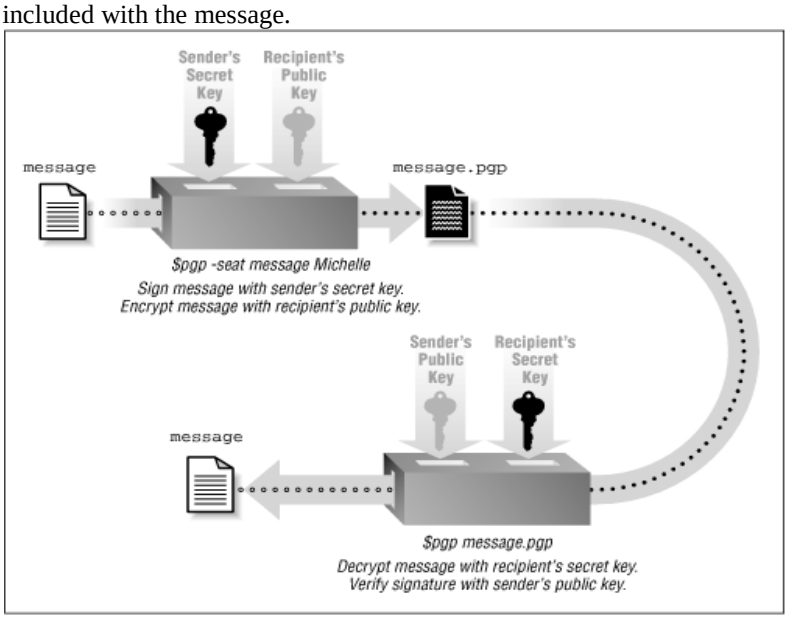

Fig-5

- Unlike earlier encryption methods, the security of PGP encryption lies entirely with the key.
- It is PGP's selection of the complex keys used to do an encryption that makes it next to impossible to crack.
- Sending plain text E-mail is little more secure than sending a postcard – PGP enables encryption
- PGP is useful for digitally signing material that is important (case of tutorials being cancelled)
- Enables secure transactions over E-mail.
- Pretty much unbreakable

Signing Messages

- Sender's private key is used to encrypt some or all of the message
- Public key of sender is widely available so verification of signature is easy for anyone

Fig. 6 a) Original image, b) Encrypted image using RSA with key size of 1024 bits.

B. Image decryption

In image decryption, the encrypted image is read in order to convert it to pixels whose decimal values are transformed to binary values of 8 bits, which is then converted to a large encrypted message which must be split into messages (C_i) equal to the size of the modulus in bits. Then these messages are decrypted one by one to get the original messages. As in encryption if the last message in the list does not have enough bits then it is concatenated to the decrypted messages without being decrypted. This decrypted long message is reconstructed as matrix of pixels of 8 bits which can be viewed as an image in the MATLAB. Fig. 3(b) shows the decrypted image which is same as the original image shown in Fig. 6(a).

V. HIDING SENSITIVE DETAILS INSIDE THE IMAGE

The sensitive details in the form of alphabets are converted into ASCII character which is then converted into binary data. Then each bit of the data is inserted into the LSB of image pixels according to the password set. The pixel who's LSB has to be exchanged with the data bit depends upon the password. At the receiver end the password is checked for authentication if it is correct then the message details is retrieved and displayed. If the password is wrong then error message is displayed. We can see that there is no visible difference in the image before hiding message details Fig. 4(a) and after hiding the important details Fig. 4(b).

a) Original image before hiding message details,

b) Image after hiding message details

Fig. 6

VI. CONCLUSIONS

In this paper the software implementation of the encryption/decryption using RSA and LSB steganography has been presented. This implementation has the facility that allows the user to generate keys of 128 to 2048 bits and encrypt or decrypt images with sizes ranging from (128xI28) pixels to (512x512) pixels while viewing them after the encryption or decryption. This also allows the hiding of the message details inside the image using LSB steganography. This implementation can be used for secure storage and transmission of sensitive images.

REFERENCES

[1] Juergen Seitz,DIgitalWtermarking for Digital Media,ISBN 151440519X,2005,Information Resources Press,Arlington,VA,USA

[2] Richard A. Mollin, An introduction to Cryptography (Discrete Mathematics and Its Applications), Chapman and Ha11/CRC; 2nd edition.

[3] William Stallings, Cryptography and Network Security Principles and Practices, Prentice Hall 4th Edition, November 16, 2005.

[4] R.L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems" Communications of the ACM, vol. 21, no. 2, pp. 120--126, Feb. 1978.

[5] Joppe w. Bos, Marcelo E. Kaihara, Thorsten Kleinjung, Arjen K. Lenstra and Peter L. Montgomery, "On the Security of 1024-bit RSA and 160-bit Elliptic Curve Cryptography", September 2009.

[6] T Morkel, JHP Eloff and MS Olivier, "An Overview of Image Steganography," Proceedings of the Fifth Annual Information Security South Africa Conference (ISSA2005), Sandton, South Africa, June/July 2005 (Published electronically). Richard E Klima, Neil Sigmon, Ernest Stitzinger, Applications of Abstract Algebra with Maple and MATLAB, Chapman and Hall/CRC; 2nd edition, July 12, 2006.
http://en.wikipedia.org/wiki/Steganography
http://www.math.ucsd.edu/~crypto/projects/MaxWeiss/stegano graphy.pdf

