A Stegnography Project-Data Hiding in Encrypted Images

Priyanka Pandey
B.Tech Student Department of CSE ITM GIDA, Gorakhpur

Pallavee Tiwari
B.Tech Student Department of CSE ITM GIDA, Gorakhpur

Shalini Mishra
B.Tech Student Department of CSE ITM GIDA, Gorakhpur

Ajay Kumar
Assistant Professor Department of CSE ITM GIDA, Gorakhpur

Abstract – In the present method the data is hidden behind the images, the intruders can easily acquire this information because it is not encrypted. In this paper we have developed a technique to hide data in Encrypted images. We have explored the limits of stegnography theory and practice and have arrived to this unique method of data hiding. This paper intends to give an overview of image stegnography and image encryption, its uses and techniques in the proposed system we are encrypting the data using the Rijndael algorithm which is also used by the U. S. Military forces. This algorithm hides the data in the image using the LSB technique and the image is again encrypted using the RSA algorithm. For this entire encryption process the user has a password. After the encryption process is complete the sender sends the file to the receiver. The receiver can extract the data only if he is authenticated in the process of login.

Index Terms – Data encryption, LSB technique, Rijndael algorithm, RSA algorithm

1. INTRODUCTION

One of the reasons that intruders can be successful is the most of the information they acquire from a system is in a form that they can read and comprehend. Intruders may reveal the information to others, modify it to misrepresent an individual or organization, or use it to launch an attack. One solution to this problem is, through the use of advanced stegnography. Advanced stegnography is a technique of hiding information in digital media and encrypting it. In contrast to cryptography, it is not to keep others from knowing the hidden information but it is to keep others from thinking that the information even exists. Advanced stegnography become more important as more people join the cyberspace revolution. Advanced stegnography is the art of concealing information in ways that prevents the detection of hidden messages. Advanced stegnography include an array of secret communication methods that hide the message from being seen or discovered.

Due to advances in ICT, most of information is kept electronically. Consequently, the security of information has become a fundamental issue. Besides cryptography, stegnography can be employed to secure information. In cryptography, the message or encrypted message is embedded in a digital host before passing it through the network, thus the existence of the message is unknown. Besides hiding data for confidentiality, this approach of information hiding can be extended to copyright protection for digital media: audio, video and images.

The growing possibilities of modern communications need the special means of security especially on computer network. The network security is becoming more important as the number of data being exchanged on the internet increases. Therefore, the confidentiality and data integrity are requires to protect against unauthorized access and use. This has resulted in an explosive growth of the field of information hiding. Information hiding is an emerging research area, which encompasses applications such as copyright protection for digital media, watermarking, fingerprinting, and stegnography. The goal of advanced stegnography is covert communication. So, a fundamental requirement of this advanced stegnography system is that the hider message carried by steno-media should not be sensible to human beings. The other goad of advanced stegnography is to avoid drawing suspicion to the existence of a hidden message. This approach of information hiding technique has recently become important in a number of application areas.

This project has following objectives:

1. To product security tool based on stegnography techniques.
2. To explore techniques of hiding data using encryption

module of this project.

3. To extract techniques of getting secret data using decryption module.

Advanced stegnography sometimes is used when encryption is not permitted. Or, more commonly, advanced stegnography is used to supplement encryption. An encrypted file may still hide information using stegnography, so even if the encrypted file is deciphered, the hidden message is not seen.

2. METHODOLOGY

User needs to run the 2 applications. In first application the user has two tab options – encrypt text and decrypt text. If user selects encrypt, application gives the screen to select image file, information file and option to save the image file. If user select decrypt, application gives the screen to select only image file and asks the path where the user wants to save the secrete file.

This project has four methods – Encrypt text, Decrypt text, Encrypt Image and Decrypt Image.

In encryption the secret information is hiding in with any type of image file and the image is encrypted after that. Decryption is getting the secret information from image file.

3. DESIGNING OF MODULES

Advanced stegnography system requires any type of image file and the information or message that is to be hidden. It has two modules encrypt and decrypt.

Microsoft .Net framework prepares a huge amount of tool and options for programmers that they simples programming. One of .Net tools for pictures and images is auto- converting most types of pictures to BMP format. I used this tool in this software called "Advanced stegnography" that is written in C#.Net language and you can use this software to hide your information in any type of pictures without any converting its format to BMP (software converts inside it) The algorithm used for Encryption and Decryption in this application provides using several layers lieu of using only LSB layer of image. Writing data starts from last layer (8st or LSB layer); because significant of this layer is least and every upper layer has doubled significant from its down layer. So every step we go to upper layer image quality decreases and image retouching transpires.

The encrypt module [1] is used to hide information into the image; no one can see that information or file. This module requires any type of image and message and gives the only one image file in destination. After hiding data in image the image is encrypted using Rijndael encryption algorithm the decrypt module is used to get the hidden information in an encrypted image file. It take the encrypted image file as an input, and give two file at destination folder, one is the same image file and another is the message file that is hidden it that.

Before encrypting file inside image we must save name and size of file in a definite place of image. We could save file name before file information in LSB layer and save file size and file name size in most right-down pixels of image. Writing this information is needed to retrieve file from encrypted image in decryption state.

The graphical representation of this system is as follows:

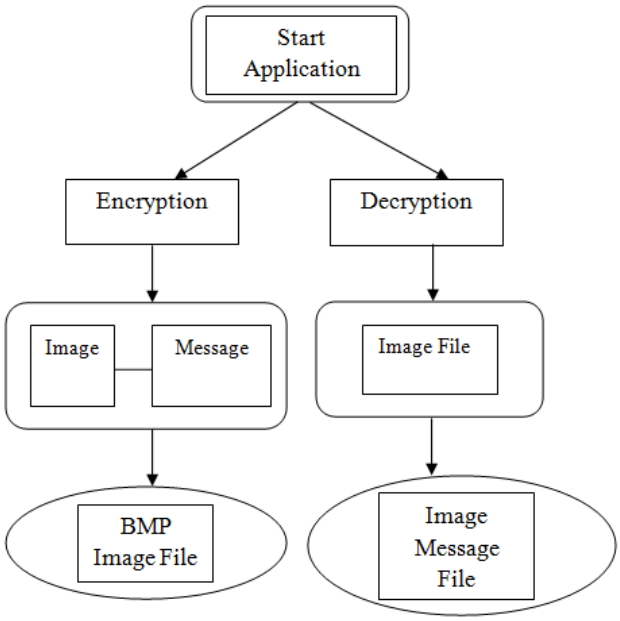

Figure 1. Process Model

4. ALGORITHM USED

RIJNDAEL

AES is based on the Rijndael cipher developed by two Belgian cryptographers, Joan Daemen and Vincen Rijmen, who submitted a proposal to NIST during the AES selection process. Rijndael is a family of ciphers with different key and block sizes.

For AES, NIST selected three members of the Rijndael family, each with a block size of 128 bits, but three different key lengths: 128, 192 and 256 bits.

AES [2, 3] is based on a design principle known as a substitution-permutation network, combination of both substitution and permutation, and is fast in both software and hardware. Unlike its predecessor DES, AES does not use a Feistel network. AES is a variant of Rijndael which has a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits. By contrast, the Rijndael specification *per se* is specified with block and key sizes that may be any multiple of 32 bits, both with a minimum of 128 and a maximum of 256 bits.

AES operates on a 4×4 column- major order matrix of bytes, termed the *state*, although some versions of Rijndael have a larger block size and have additional columns in the state. Most AES calculations are done in a special finite field.

The key size used for an AES cipher specifies the number of repetitions of transformation rounds that convert the input, called the plaintext, into the final output, called the cipher text. The number of cycles of repetition is as follows:

10 cycles of repetition for 128-bit keys.
12 cycles of repetition for 192-bit keys.
14 cycles of repetition for 256-bit keys.

Each round consists of several processing steps, each containing four similar but different stages, including one that depends on the encryption key itself. A set of reverse rounds are applied to transform cipher text back into the original plaintext using the same encryption key.

Encryption process

Key Expansion—round keys are derived from the cipher key using Rijndael key schedule. AES requires a separate 128-bit round key block for each round plus one more.

1. ***Initial Round***
    1. ***Add Round Key—each byte of the state is combined with a block of the round key using bitwise xor.***
2. ***Rounds***
    1. ***Sub Bytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.***
    2. ***Shift Rows—a transposition step where the last three rows of the state are shifted cyclically a certain number of steps.***
    3. ***Mix Columns—a mixing operation which operates on the columns of the state, combining the four bytes in each column.***
    4. ***Add Round Key***
3. ***Final Round (no Mix Columns)***
    1. ***Sub Bytes***
    2. ***Shift Rows***
    3. ***Add Round Key.***

Decryption process

Decryption process is entirely opposite of encryption process. It is shown in figure 2.

RSA

RSA [4] is a cryptosystem for public key encryption, and is widely used for securing sensitive data particularly when being sent over an insecure network such as the internet.RSA derives its security from the difficulty of factoring large integers that are the product of two large prime numbers. Multiplying these two numbers is easy, but determining the original prime numbers from the total factoring is considered infeasible due to the time it would take even using today's super computers. The public and the private key-generation algorithm [2] is the most complex part of RSA cryptography. Two large prime numbers, p and q, are generated using the Rabin-Miller primality test algorithm. A modulus n is calculated by multiplying p and q. This number is used by both the public and private keys and provides the link between them. Its length, usually expressed in bits, is called the key length. The public key consists of the modulus n, and a public exponent, e, which is normally set at 65537, as it's a prime number that is not too large. The e figure doesn't have to be a secretly selected prime number as the public key is shared with everyone. The private key consists of the modulus n and the private exponent d, which is calculated using the Extended Euclidean algorithm to find the multiplicative inverse with respect to the quotient of n.

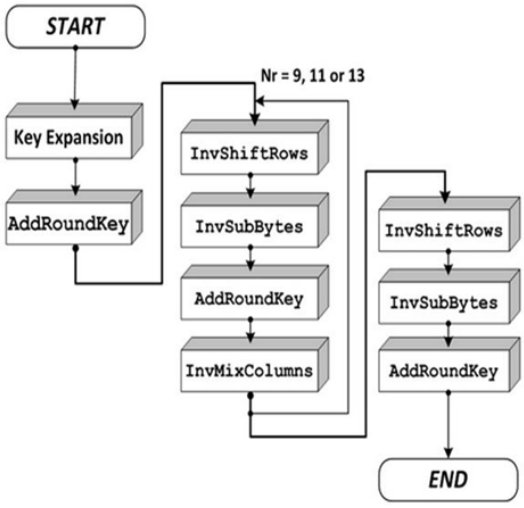

Figure 2. Decryption Process

Security of RSA

As discussed, the security of RSA relies on the computational difficulty of factoring large integers. As computing power increases and more efficient factoring algorithms are discovered, the ability to factor larger and larger numbers also increases. Encryption strength is directly tied to key size, and doubling key length delivers an exponential increase in strength, although it does impair performance. RSA keys are typically 1024- or 2048-bits. Long, but experts believe that 1024-bit keys could be broken in the near future, which is

why government and industry are moving to a minimum key length of 2048-bits. Barring an unforeseen breakthrough in quantum computing, it should be many years before longer keys are required, but elliptic curve cryptography is gaining favor with many security experts as an alternative to RSA for implementing public-key cryptography. It can create faster, smaller and more efficient cryptographic keys. Much of today's hardware and software is ECC-ready and its popularity is likely to grow as it can deliver equivalent security with lower computing power and battery resource usage, making it more suitable for mobile apps than RSA. Finally, a team of researchers which included Adi Shamir, a co- inventor of RSA [4], has successfully determined a 4096-bit RSA key using acoustic cryptanalysis, however any encryption algorithm is vulnerable to this type of attack.

5. CONCLUSION

The sender encrypts the data using a password and hides it behind the image and then the image is encrypted using the same password and sent to the receiver .If the receiver is authorized then he will authenticate through the login process and the decrypt the image and the data respectively using the same password .This provides high level of security y to the user.

REFERENCES

[1] C.Anuradha and S.Lavanya,"Computer Science Engineering,Bharath University, India" at International Journal of Advanced Research In Computer Science and Software Engineering.
[2] Srinivasan Nagaraj, Kishore Bhamidipat and G Apparao, Computer Science Engineering GMRIT University, India at International Journal of Computer Applications October 2010.
[3] [Online]. Available: http://wikipedia.org/
[4] [Online].Available:http://searchsecurity.techtarget.com/definition/RSA