Stecryption: Multiple Encryptions for Secure Image Steganography

Sapna Sinha
Assistant Professor, IT Dept.
ASET, Amity University, Sector-125, Noida

Vishal Bhatnagar
Associate Professor, CSE Dept.
AIACT & R, Geeta Colony Delhi

ABSTRACT
The art of hiding information behind the image is a popular concept. In this paper we have introduced a blend of multiple encryption techniques to hide information behind an image. In this paper we have discussed Steganography and encryption techniques that can be combined in several ways to strengthen information security. The effectiveness of the individual encryption techniques is pondered upon. The planned work flow to develop such an application is also discussed.

General Terms
Image Steganography, Information Security, Encryption.

Keywords
Steganography, AES, Caesar, DES, Desede, Ignore, Blowfish, At Bash, RC4, RSA.

1. INTRODUCTION
Data security during the transmission to various networks has become the major concern today. Different types of data get transmitted on these networks, out of which security of multimedia data is focal of concern. Existing encryption schemes are not found suitable for ensuring security of multimedia data, due to properties such as huge size and uncompressed data. Therefore, new and good image encryption schemes for multimedia data are needed [1].

The idea of camouflaging a message behind an image sounds secure enough and is difficult for the hackers to interpret and hack. Considering this fact and development in the same area, various security attacks and certain vulnerabilities are identified. Hence, there is a need to fortify the existing technique and further its security hold.

One of the ways of enhancing the security of the message to be passed across is to make the whole scheme a two way process. This is based on the simple process of increasing the number of levels through which a message would pass before reaching the authentic party, that is, encrypting the string even before it is provided to be hidden beneath a picture. Various encryption – decryption schemes are doing the rounds these days, we will use four such schemes in the venture. These are modern day encryption decryption schemes that are hard to crack even when they are used alone to encrypt the message.

Using these encryption-decryption schemes would undoubtedly increase the security of the message by many folds. Not only the user would be able to choose from these schemes, he may apply one or more encryption-decryption at the same time. Hence, the security concerns can be subsided simply by increasing the number of encryption process the message would have to undergo.

2. WHY STEGANOGRAPHY?
Steganography is the art and science of hiding communication; a steganographic system thus embeds hidden content in unremarkable cover media so as not to arouse an eavesdropper's suspicion. In the past, people used hidden tattoos or invisible ink to convey steganographic content. Today, computer and network technologies provide easy-to-use communication channels for steganography. Essentially, the information-hiding process in a steganographic system starts by identifying a cover medium's redundant bits (those that can be modified without destroying that medium's integrity). The embedding process creates a stego medium by replacing these redundant bits with data from the hidden message.

Modern steganography's goal is to keep its mere presence undetectable, but steganographic systems— because of their invasive nature—leave behind detectable traces in the cover medium. Even if secret content is not revealed, the existence of it is: modifying the cover medium changes its statistical properties, so eavesdroppers can detect the distortions in the resulting stego medium's statistical properties.

3. STEGANALYSIS: POSSIBILITY OF DETECTION
The goal of steganalysis is quite simple: Detect suspect data, determine whether information is hidden within the data, and recover that data.

For every measure, there is a countermeasure, and steganography is no different. In this case, the countermeasure is a technique known as steganalysis

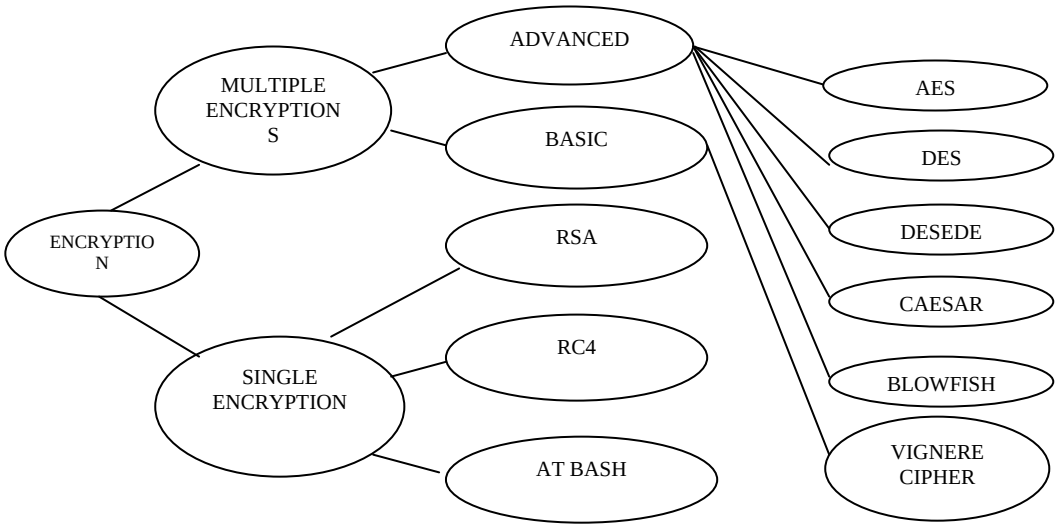

Fig. 1: Categorization of Encryption Algorithms

On the surface, this process may sound curiously like a companion technique known as cryptanalysis. But the reality is that the processes are not the same. With cryptanalysis, the fact that data is encrypted is obvious: You need only observe the data to determine whether it is encrypted. With steganalysis, the presence of data is only suspected, not confirmed.

A layer of complexity is required to be added with steganalysis techniques for the detection of hidden data. For this encryption can be combined before embedding the data.[2]

4. WORKING OF STECRYPTION

The heading of a section should be in Times New Roman 12-point bold in all-capitals flush left with additional 6-points of white space above the section head. Sections and subsequent sub- sections should be numbered and flush left. For a section head and a subsection head together (such as Section 3 and subsection 3.1), use no additional space above the subsection head.

These schemes can be merged with other encryption schemes of the same category or may be used over and over again. Vignere Cipher belongs to the Simple category while AES, DES, Desede, Blowfish, Caesar Cipher – they all belong to the Advanced category(refer Fig. 1). When finally the message is encrypted as per the wish, it is transmitted.

Proposed algorithm for Encryption:

Step 1 – Get the data to be hidden.
Step 2 – Open the image in which data has to be hidden
Step 3 – Convert the image into buffered image.
Step 4 – Encrypt the data using any encryption algorithm
Step 5 – Associate unique id for the encryption algorithm and append with round number as "1" and id at the end of the encrypted data.
Step 6 – repeat step 4 to 5 using any other encryption technique and increase round number by 1.
Step 7 – Check if the length of the encrypted text is less than the length of the image.
Step 8 – Convert the data into bytes
Step 9 – Convert the image into bytes.
Step 10- Loop through each bit of the text.
Step 11 – Use the exclusive or function to hide the data into the image by using least significant bit steganography.
Step 12 – compress the image
Step 13 – Transmit compress image.

Image is compressed to ensure faster transmission rate. The algorithm used for encryption will have associated secret code which gets appended with the information and at the next level gets encrypted with other encryption algorithm selected by user and the process continues till user compresses the image.

While decrypting, the user need not to remember which scheme(s) were used, the application would simply decrypt without any hassle. Also, the order of the use of the encryption algorithms will be encrypted and saved in the image. If someone intrudes and has a look at the encrypted message, then he would not be able to comprehend any aspect related to the encrypted message.

Proposed algorithm for Decryption:

Step 1 – user opens the image containing hidden information.
Step 2 – Convert the image into buffered image
Step 3 – convert the image into bytes.
Step 4 – Loop through each bit of the text.
Step 5 – user XOR to retrieve the data in the image by using least significant bit steganography
Step 6 - check the unique id of algorithm used and decrypt it using decryption algorithm
Step 7 – extract id and round no. if round no. is greater then "0" repeat step 6 to 7.
Step 8 – Display the hidden data

The concept of multiple encryption algorithms on the data to be hidden in the image is limited to certain image formats like – bmp, gif and .png. File formats like jpeg are lossy, so different encryption technique is required for image steganography.

Fig. 2.a

Fig. 2.b

Fig. 2a and 2b shows the images before and after 'strecrypting' the message. Apparently, there is no difference between the two images. Hence, it becomes impossible for any intruder to comprehend any kind of hiding of message.

Authorization and authentication can be used to add level of security for the use of application using login application which minimizes the risk of Intrusions.[9]

5. ANALYSING THE ENCRYPTION TECHNIQUES

In AES competition, Rijndael was the NIST finalist and was declared the new symmetric cipher that is expected to take place of DES. It was possible to break upto eight rounds of cipher with exhaustive key search, for 192 and 256 bit keys.[3].

Nine rounds can be attacked using related-key attacks, but this is still impractical. Four attacks were discovered to work against reduced-rounds versions of Rijndael: Square Attack, Improved Square Attack, Impossible Differential Attack and Reversed Key Schedule Attack.

A Square Attack breaks four rounds of Rijndael in 29 time, requiring 29 chosen plaintexts. An Improved Square Attack does that the same in 28 time. The same Square Attack when running on 5-rounds requires 240 times, with 211 chosen plaintexts. The Improved Square Attack does the same in 239 time. The Sqaure Attack on six rounds requires 272 time using 232 chosen plaintexts while the Improved Square Attack will take 271 times. The Impossible Differential Attack handles five rounds in 231 time using 229.5 chosen plaintexts, whereas a Reversed Key Schedule attack requires only 211 chosen plaintexts for the same job. An attack on six rounds can be done using the Reversed Key Schedule Attack in 263 time using 232 known plaintexts. Attacks on six rounds were also shown using 6*232 known plaintexts in 244 operations.

For seven rounds of Rijndael, attacks on a 192-bit key were shown using 19*232 known plaintexts in 2155 time. For a 256-bit key this attack would require 21*232 known plaintexts and 2172 time. These attacks are, of course, infeasible due to length of time required and due to the fact that Rijndael employs more than seven rounds when in practical use.[11][12][13]

Attacks on eight rounds will take the entire codebook of known plaintexts and 2188 time for a 192-bit key (2204 time for a 256-bit key) and are therefore totally inapplicable. A related-key attack on nine rounds of Rijndael with 256-bit key will require 277 plaintexts to be encrypted using 256 related keys and 2224 time. This attack is clearly infeasible.

The DES algorithm, other than its short key (which can be brute-forced quite easily), is known to be secure. Triple-DES was developed to overcome this limitation. An attack, on six rounds of DES, in a time of 254. This attack cannot be applied on more than eight rounds therefore it is not alarming in practice.[7]

A known plain-text attack, using S-box pairs, succeeded in cracking eight rounds of DES using 240 known plaintexts and with 240 operations work. This is not too alarming since it has almost no effect on the strength of the full sixteen round DES [10]. Extension of this particular attack to sixteen rounds will take more than the entire codebook of plaintexts and is therefore impractical. Yet, an improvement to this attack could crack full-DES (sixteen rounds) with a time of 250 for data collection plus a time of 250 for the attack itself. There is a tradeoff between success rate, time and amount of required plaintexts. The above mentioned figure presents the best "deal" in these terms. Alternatively, the attack can be performed to the extent it reveals 24 bits of the 56-bit key using 252 known plaintexts, with very little work (complexity). These results may be alarming for systems in which bulk data is encrypted with unchanged keys.

The DES algorithm suffers from Simple Relations in its keys. In DES, there is a simple relation in its key which is of complementary nature due to this complementary relationship between the resulting ciphers texts exists. This reduces the vulnerability of the algorithm.[8]

With regards to weak keys, DES has at least four of them. When encrypting using one of these weak keys, all sixteen rounds will be using the same sub-keys, making the algorithm as strong as a single round. Therefore, use of these keys must be avoided. In addition to these four keys, there are twelve more weak keys by which two rounds are running using the same sub-keys. In addition to these weak keys, DES also has keys that are defined as weak1 and keys that are defined as semi-weak2. All these keys should be avoided so as not to harm the strength of the implementation when using the algorithm. [4][5]

The key schedule that DES uses is not one-way. This results in the attacker being able to recover most of the master-key by

compromising the sub-keys of few rounds. This vulnerability is hardly practical since the round keys are not easily available. Yet, this feature does assist in optimizing differential attacks.

The DES algorithm is vulnerable to linear cryptanalysis attacks. By such an attack, the algorithm in its sixteen rounds can be broken using 243 known plaintexts. This vulnerability raises a notable risk when encrypting bulk data that may be predictable with keys that are constant.

Differential attack introduced by Eli Biham and Adi Shamir, by which key can be recovered in 237 time using 237 cipher texts taken from a pool after encrypting 247 chosen plaintexts, encrypted with different keys.[14]

This attack is hard to mount in most circumstances in practical. DES has several modes of operation, most commonly used is CBC. Yet, if such modes (other than ECB) are used, it must be verified that the IV (Initialization Vector) is either fixed or not transferred, which makes its implementation vulnerable in the presence of active attacker.[3]

Theoretically 3DES algorithm is vulnerable to meet-in-the-middle, linear and differential attacks. Costs of such attacks are not known.[6]

In general Triple DES with three independent keys (keying option 1) has a key length of 168 bits (three 56-bit DES keys), but due to the meet-in-the-middle attack the effective security it provides is only 112 bits. Keying option 2 reduces the key size to 112 bits. However, this option is susceptible to certain chosen-plaintext or known-plaintext attacks and thus it is designated by NIST to have only 80 bits of security.

The best attack known on keying option 1 requires around 232 known plaintexts, 2113 steps, 290 single DES encryptions, and 288 memory (the paper presents other tradeoffs between time and memory)[6]. This is not currently practical and NIST considers keying option 1 to be appropriate through 2030. If the attacker seeks to discover any one of many cryptographic keys, there is a memory-efficient attack which will discover one of 228 keys, given a handful of chosen plaintexts per key and around 284 encryption operations.

The main weakness in this algorithm is that due to a weak key-mixing phase, 1/256 of the keys belong to a class of weak keys. These keys are detectable. After detection of a key belonging to this class, it is fairly easy to reveal 16 bits of the key with a 13.8% probability. In any implementation of this algorithm, a test to assure these keys are not used must be performed.

Blowfish was written by Bruce Schneider, a well-known cryptographer. The algorithm is designed to resist all known attacks on block ciphers. Blowfish is known for its high security and is available in several common encryption products.

Blowfish has some classes of weak keys. For these weak keys, separate rounds end up using the same round-keys. Keys belonging to these classes can be detected only in reduced-rounds versions of the algorithm and not on the full blowfish. Blowfish is known to successfully make (almost) every bit of the key affect most of the round-keys. Blowfish is immune against differential related-key attacks because of the fact that every bit of the master key affects many round keys. The round-keys are highly independent, making related-key attacks very difficult or infeasible. Such independence is highly desirable.

6. CONCLUSION

Steganography is an old and incredibly versatile and effective technique for obscuring information that is actually in plain sight. Although methods for detecting hidden data do exist, they cannot be entirely relied upon, as none is 100 percent effective. Any attempt to detect and thwart this technique must combine technology and vigilance to blunt the effectiveness of the process. Remember, if done correctly, information could be hidden in any image, document, or other piece of data and never even be suspected.

7. REFERENCES

[1] T. Matsumoto, and J. Shikata, .Authenticated encryption and steganography in unconditional security setting, Theory and Practice in Information-Theoretic Security, 2005. IEEE Information Theory Workshop on, IEEE Conference Proceeding, 2005, pp.1-6.

[2] Sean-Philip Oriyeno (2009), 'Using Steganography to avoid observation', retrieved on 12th April, 2012, available at: http://www.ibm.com/ developerworks/web / library/wa-steganalysis/

[3] Limor Elbaz, Hagai Bar El, 'Strength Assessment of Encryption Algorithms' White Paper October 2000 Discretix Technologies Ltd.

[4] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid. (2007, Mar.) NIST Special Publication 800-57: Recommendation for Key Management — Part 1: General. [Online]. Available: http://csrc.nist.gov/ publications /nistpubs/800-57/sp800-57-Part1-revised2 Mar08-2007.pdf.

[5] W. Timothy Polk, Donna F. Dodson, and William E. Burr. Cryptographic algorithms and key sizes for personal identity verification. NIST Special Publication 800-78, National Institute of Standards and Technology, Gaithersburg, MD April 2005.

URL: http://csrc.ncsl.nist.gov/ publications /nistpubs/800-78/sp800-78-final.pdf.

[6] "The Cryptography Guide: Triple DES", Cryptography World, available at: http://www.cryptographyworld.com /des.htm.

[7] IBM, 'Triple DES Encryption', retrieved on 15th March 2010, Available at: http://publib. boulder.ibm.com /infocenter /zos/v1r9/ index.jsp?topic=/com.ibm.zos.r9. csfb400 /tdes1.htm

[8] Dworkin, Morris, 'Recommendation for Block Cipher Modes of Operation, Methods and Techniques', NIST Special Publication 800-38A 2001 Edition.

[9] Ralph Merkle, Martin Hellman: On the Security of Multiple Encryption , Communications of the ACM, Vol 24, No 7, pp 465–467, July 1981.

[10] Fauzan Mirza, 'Linear and S-Box Pair Cryptanalysis on DES', Third year undergraduate project. October 1996-April 1997.

[11] Niels Ferguson, John Kelsey, Stefan Lucks, Bruce Schneier Mike Stay, David Wagner, and Doug Whiting, 'Improved Cryptanalysis of Rijndael', Proceedings of Fast Software Encryption 2000, LNCS.

[12] Henry Gilbert, Marine Minier, 'A Collision Attack on Seven Rounds of Rijndael', available at: csrc.nist.gov/archive/aes/round2/conf3/papers/11-hgilbert.pdf.

[13] Eli Biham, Nathan Keller, 'Cryptanalysis on Reduced Rijndael', available at: http://csrc.nist.gov/ encryption/aes /round2/conf3/aes3papers.html.

[14] E. Biham and A. Shamir, "Differential cryptanalysis of DES- like cryptosystems," Journal of Cryptology, Vol. 4, No. 1, 1991, pp. 3–72.

[15] M. Sitaram Prasad, S. Naganjaneyulu, CH. Gopi Krishna, C. Nagaraju, A Novel Information Hiding Tecgnique For Security By Using Image Steganography, Journal of Theoretical and Applied Information Technology, Vol. 8, No. 1, 2009.