ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology
Volume 1, Issue 5, July 2012

DATA SECURITY USING CRYPTOGRAPHY AND STEGNOGRAPHY

Epuru Madhavarao[1],ChikkalaJayaRaju[2],Pedasanaganti Divya[3],A.S.K.Ratnam[4]

Abstract— Steganography and Cryptography are two popular ways of sending vital information in a secret way. One hides the existence of the message and the other distorts the message itself. There are many cryptographic techniques available and among them AES is one of the most powerful techniques. The scenario of present day of information security system includes confidentiality, authenticity, integrity, non-repudiation. This present paper focus is enlightening the technique to secure data or message with authenticity and integrity. The security of communication is a crucial issue on World Wide Web (internet) and within organizations. It is about confidentiality, integrity and authentication during access or editing of confidential internal documents. We are using a nonconventional steganography to increase security, which uses the cryptography to encrypt confidential message with the public and private keys. These keys are generated differently. Then loss-less compression takes place, which makes possible to hide larger amounts of information and documents using steganography. This paper presents a technique for constructing and implementing new algorithm based on embedding efficiently a large amount of data with high quality of encryption techniques, together with steganography, providing authentication and electronic documents integrity.

Index Terms— Cryptography, Stegnography, Authentication, Integrity.

I INTRODUCTION

Cryptography [1] and Steganography [1] are well known and widely used techniques that manipulate information (messages) in order to cipher or hide their existence respectively. Steganography is the art and science of communicating in a way which hides the existence of the communication. Cryptography scrambles a message so it cannot be understood; the Steganography hides the message so it cannot be seen. In this paper we will focus to develop one system, which uses both cryptography and Steganography for better confidentiality and security. Presently we have very secure methods for both cryptography and Steganography - AES algorithm is a very secure technique for cryptography and the Steganography methods, which use frequency domain, are highly secured. Even if we combine these techniques straight forwardly, there is a chance that the intruder may detect the original message. Therefore, our idea is to apply both of them together with more security levels and to get a very highly secured system for data hiding. This paper mainly focuses on to develop a new system with extra security features where a meaningful piece of text message can be hidden by combining security techniques like

Manuscript received June, 2012.

Epuru Madhavarao, Pursuing M.Tech(CSE) from Vignan's Lara Institute of Technology & Science, Vadlamudi, Guntur, A.P., India. My research Interests are Data Security and computer networks.

Chikkala JayaRaju, Pursuing M.Tech(CSE) from Vignan's Lara Institute of Technology & Science, Vadlamudi, Guntur, A.P., India. My research Interests are Semantic Web and Mobile computing**.**

Pedasanaganti Divya, Pursuing M.Tech(CSE) from Vignan's Lara Institute of Technology & Science, Vadlamudi, Guntur, A.P., India. My research Interests are computer networks and Mobile computing.

A.S.K.Ratnam, Assoc.Professor & Head, Department of Computer Science Engineering at Vignan's Lara Institute of Technology & Science, Vadlamudi, Guntur, A.P., India..My research Interests includes Image Processing, Mobile Computing, Network Security.

Cryptography and Steganography.

Steganography is a technology that embeds a confidential message or image within a text, or a digital picture or digital videos or digital audios. It is sometime confused with cryptography, not in name but in the usage. The simple way to differentiate that steganography conceals not only the contents of the message but also the mere existence of a message from an observer so there is no chances of doubt of the existence of the message, where as in cryptography the purpose is to secure communication from hackers by converting confidential message into not understandable form. It is observed from previous experience that sending encrypted information may create suspicion while invisible information will not do so.

Steganalysis is a technology which determines the presence of a hidden message or image in cover image and attempt to disclose the actual contents of this message [1].A more sophisticated method of steganography is by combining the two technologies to produce more security to confidential data communication such that if hackers detect the presence of data even then digital pixels are represented by three colors: red, green and blue. These colors together form digital pictures or video. Each color of every pixel requires 1 byte of information, or 8 bits. Since the first bit is the "least significant" or carries the least amount of importance in the byte, this steganographic technique chooses to overwrite the first bit of successive bytes until the entire secret message is embedded into the original source file, or the cover data. Since we have only modified the least significant bits of a portion of the source file, the human eye should not be able to detect the degradation in the picture or video [2].

2.Cryptography:

What does the word *crypto* mean? It has its origins in the Greek word *kruptos*, which means *hidden*.Thus, the objective of cryptography is to hide information so that only the intended recipient(s) can "unhide" it. In crypto terms, the hiding of information is called *encryption*, and when the information is unhidden, it is called *decryption*.A cipher is used to accomplish the encryption and decryption. Merriam-Webster's Collegiate Dictionary defines *cipher* as "a method of transforming a text in order to conceal its meaning."The information that is being hidden is called *plaintext*; once it has been encrypted, it is called *ciphertext*.The ciphertext is transported, secure from prying eyes, to the intended recipient(s), where it is decrypted back into plaintext.

2.1Cryptography Techniques:

After applying the Huffman compression algorithm we need to know about the Cryptographic algorithms. There are some specific security requirements for cryptography, including Authentication, Privacy/confidentiality, and Integrity Nonrepudiation.

The three types of algorithms are described:

(i)Secret Key Cryptography (SKC): Uses a single key for both encryption and decryption

(ii)Public Key Cryptography (PKC): Uses one key for encryption and another for decryption

(iii)Hash Functions: Uses a mathematical transformation to irreversibly "encrypt" information.

2.2.Encryption key types:

Cryptography uses two types of keys: *symmetric* and *asymmetric*. Symmetric keys have been around the longest; they utilize a single key for both the encryption and decryption of the ciphertext.This type of key is called a *secret key*, because you must keep it secret. Otherwise, anyone in possession of the key can decrypt messages that have been encrypted with it.The algorithms used in symmetric key encryption have, for the most part, been around for many years and are well known, so the only thing that is secret is the key being used. Indeed, all of the really useful algorithms in use today are completely open to the public. A couple of problems immediately come to mind when you are using symmetric key encryption as the sole means of cryptography. First, how do you ensure that the sender and receiver each have the same key? Usually this requires the use of a courier service or some other trusted means of key transport.

Second, a problem exists if the recipient does not have the same key to decrypt the ciphertext from the sender. For example, take a situation where the symmetric key for a piece of crypto hardware is changed at 0400 every morning at both ends of a circuit.What happens if one end forgets to change the key (whether it is done with a strip tape, patch blocks, or some other method) at the appropriate time and sends ciphertext using the old key to another site that has properly changed to the new key? The end receiving the transmission will not be able to decrypt the ciphertext, since it is using the wrong key.This can create major problems in a time of crisis,

especially if the old key has been destroyed. This is an overly simple example, but it should provide a good idea of what can go wrong if the sender and receiver do not use the same secret key. Asymmetric cryptography is relatively new in the history of cryptography, and it is probably more recognizable to you under the synonymous term *public key cryptography*.Asymmetric algorithms use two different keys, one for encryption and one for decryption—a *public key* and a *private key*, respectively.Whitfield Diffie and Martin Hellman first publicly released public key cryptography in 1976 as a method of exchanging keys in a secret key system.Their algorithm, called the Diffie-Hellman (DH) algorithm, is examined later in the chapter. Even though it is commonly reported that public key cryptography was first invented by the duo, some reports state that the British Secret Service actually invented it a few years prior to the release by Diffie and Hellman. It is alleged, however, that the British Secret Service never actually did anything with their algorithm after they developed it.

3.Learning about Standard Cryptographic Algorithms:

Just why are there so many algorithms anyway? Why doesn't the world just standardize on one algorithm? Given the large number of algorithms found in the field today, these are valid questions with no simple answers. At the most basic level, it's a classic case of tradeoffs between security, speed, and ease of implementation. Here *security* indicates the likelihood of an algorithm to stand up to current and future attacks, *speed* refers to the processing power and time required to encrypt and decrypt a message, and *ease of implementation* refers to an algorithm's predisposition (if any) to hardware or software usage. Each algorithm has different strengths and drawbacks, and none of them is ideal in every way. In this chapter, we will look at the five most common algorithms that you will encounter: Data Encryption Standard (DES), AES [Rijndael], International Data Encryption Algorithm (IDEA), Diffie-Hellman, and Rivest, Shamir, Adleman (RSA). Be aware, though, that there are dozens more active in the field.

3.1.Symmetric Algorithms:

In this section, we will examine several of the most common symmetric algorithms in use: DES, its successor AES, and the European standard, IDEA. Keep in mind that the strength of symmetric algorithms lies primarily in the size of the keys used in the algorithm, as well as the number of cycles each algorithm employs.

A)DES Algorithm:

Among the oldest and most famous encryption algorithms is the Data Encryption Standard, which was developed by IBM and was the U.S. government standard from 1976 until about 2001. DES was based significantly on the Lucifer algorithm invented by Horst Feistel, which never saw widespread use. Essentially, DES uses a single 64-bit key—56 bits of data and 8 bits of parity—and operates on data in 64-bit chunks.This key is broken into 16 separate 48-bit subkeys, one for each round, which are called *Feistel cycles*. Figure 6.1 gives a schematic of how the DES encryption algorithm operates. Each round consists of a substitution phase, wherein the data is substituted with pieces of the key, and a permutation phase, wherein the substituted data is scrambled (re-ordered). Substitution operations, sometimes referred to as confusion operations, are said to occur within S-boxes. Similarly, permutation operations, sometimes called diffusion operations, are said to occur in P-boxes. Both of these operations occur in the "F Module" of the diagram.The security of DES lies mainly in the fact that since the substitution operations are non-linear, so the resulting ciphertext in no way resembles the original message.Thus, languagebased analysis techniques (discussed later in this chapter) used against the ciphertext reveal nothing.The permutation operations add another layer of security by scrambling the already partially encrypted message. Every five years from 1976 until 2001, the National Institute of Standards and Technology (NIST) reaffirmed DES as the encryption standard for the U.S. government. However, by the 1990s the aging algorithm had begun to show signs that it was nearing its end of life. New techniques that identified a shortcut method of attacking the DES cipher, such as differential cryptanalysis, were proposed as early as 1990, though it was still computationally unfeasible to do so.

DES is a block cipher which takes a fixed-length string of
plaintext bits and transforms it into cipher text bit string of
the same length. The key length of DES is 64 bits.

DES Algorithm uses Fiestal structure which performs the following operations.

1. *Expansion:* By duplicating some of the bits, 32-bit block is expanded into 48 bit block by using the expansion permutation.
2. *Key mixing:* Expanded block is mixed up with a substitution key by using an XOR operation. Sixteen 48-bit sub keys are derived from the main key as one key for each round.
3. *Substitution:* After Key Mixing is over, the block is further divided into 6-bit pieces of eight blocks. By following Non liner transformation, each of the Sboxes replaces its six input bits with four output bits. S-boxes plays a important role in determining the security of the algorithm and without them the algorithm becomes linear and easily breakable.
4. *Permutation*: Thus after substitution 32 outputs from the S-boxes is rearranged by using the concept of permutation

B)AES (Rijndael) Algorithm:

In 1997, as the fall of DES loomed ominously closer, NIST announced the search for the Advanced Encryption Standard, the successor to DES. Once the search began, most of the big name cryptography players submitted their own AES candidates.

Among the requirements of AES candidates were:

- AES would be a private key symmetric block cipher (similar to DES).
- AES needed to be stronger and faster then 3-DES.
- AES required a life expectancy of at least 20-30 years.
- AES would support key sizes of 128-bits, 192-bits, and 256-bits.
- AES would be available to all—royalty free, non-proprietary and unpatented.

Within months NIST had a total of 15 different entries, 6 of which were rejected almost immediately on grounds that they were considered incomplete.

By 1999, NIST had narrowed the candidates down to five finalists including MARS, RC6, Rijndael, Serpent, and Twofish.

Selecting the winner took approximately another year, as each of the candidates needed to be tested to determine how well they performed in a variety of environments. After all, applications of AES would range anywhere from portable smart cards to standard 32-bit desktop computers to high-end optimized 64-bit computers. Since all of the finalists were highly secure, the primary deciding factors were speed and ease of implementation (which in this case meant memory footprint).

Rijndael was ultimately announced as the winner in October of 2000 because of its high performance in both hardware and software implementations and its small memory requirement.The Rijndael algorithm, developed by Belgian cryptographers Dr. Joan Daemen and Dr.Vincent Rijmen, also seems resistant to power- and timing-based attacks. So how does AES/Rijndael work? Instead of using Feistel cycles in each round like DES, it uses iterative rounds like IDEA (discussed in the next section).

Data is operated on in 128-bit chunks, which are grouped into four groups of four bytes each.The number of rounds is also dependent on the key size, such that 128-bit keys have 9 rounds, 192-bit keys have 11 rounds and 256-bit keys require 13 rounds. Each round consists of a substitution step of one S-box per data bit followed by a pseudo-permutation step in which bits are shuffled between groups.Then each group is multiplied out in a matrix fashion and the results are added to the subkey for that round. How much faster is AES than 3-DES? It's difficult to say, because implementation speed varies widely depending on what type of processor is performing the encryption and whether or not the encryption is being performed in software or running on hardware specifically designed for encryption. However, in similar implementations, AES is always faster than its 3-DES counterpart. One test performed by Brian Gladman has shown that on a Pentium Pro 200 with optimized code written in C, AES (Rijndael) can encrypt and decrypt at an average speed of 70.2 Mbps, versus DES's speed of only 28 Mbps.

The European counterpart to the DES algorithm is the IDEA algorithm, and its existence proves that Americans certainly don't have a monopoly on strong cryptography. IDEA was first proposed under the name *Proposed Encryption Standard* (PES) in 1990 by cryptographers James Massey and Xuejia Lai as part of a combined research project between Ascom and the Swiss Federal Institute of Technology. Before it saw widespread use PES was updated in 1991 to increase its strength against differential cryptanalysis attacks and was renamed Improved PES (IPES). Finally, the name was changed to International Data Encryption Algorithm (IDEA) in 1992.

Not only is IDEA newer than DES, but IDEA is also considerably faster and more secure. IDEA's enhanced speed is due to the fact the each round consists of much simpler operations than the Fiestel cycle in DES.These operations (XOR, addition, and multiplication) are much simpler to implement in software than the substitution and permutation operations of DES. IDEA operates on 64-bit blocks with a 128-bit key, and the encryption/ decryption process uses 8 rounds with 6 16-bit subkeys per round.The IDEA algorithm is patented both in the US and in Europe, but free non-commercial use is permitted.

3.2.Asymmetric Algorithms:

Recall that unlike symmetric algorithms, asymmetric algorithms require more than one key, usually a *public* key and a *private* key (systems with more than two keys are possible). Instead of relying on the techniques of substitution and transposition, which symmetric key cryptography uses, asymmetric algorithms rely on the use of massively large integer mathematics problems. Many of these problems are simple to do in one direction but difficult to do in the opposite direction. For example, it's easy to multiply two numbers together, but it's more difficult to factor them back into the original numbers, especially if the integers you are using contain hundreds of digits.Thus, in general, the security of asymmetric algorithms is dependent not upon the feasibility of brute force attacks, but the feasibility of performing difficult mathematical inverse operations and advances in mathematical theory that may propose new "shortcut" techniques. In this section, we'll take a look at RSA and Diffie-Hellman, the two most popular asymmetric algorithms in use today.

C)Diffie-Hellman Algorithm:

In 1976, after voicing their disapproval of DES and the difficulty in handling secret keys,Whitfield Diffie and Martin Hellman published the Diffie-Hellman algorithm for key exchange.This was the first published use of public key cryptography, and arguably one of the cryptography field's greatest advances ever. Because of the inherent slowness of asymmetric cryptography, the Diffie-Hellman algorithm was not intended for use as a general encryption scheme—rather, its purpose was to transmit a private key for DES (or some similar symmetric algorithm) across an insecure medium. In most cases, Diffie-Hellman is not used for encrypting a complete message because it is 10 to 1000 times slower than DES, depending on implementation.

Prior to publication of the Diffie-Hellman algorithm, it was quite painful to share encrypted information with others because of the inherent key storage and transmission problems (as discussed later in this chapter). Most wire transmissions were insecure, since a message could travel between dozens of systems before reaching the intended recipient and any number of snoops along the way could uncover the key.With the Diffie-Hellman algorithm, the DES secret key (sent along with a DES-encrypted payload message) could be encrypted via Diffie- Hellman by one party and decrypted only by the intended recipient. In practice, this is how a key exchange using Diffie-Hellman works:

- The two parties agree on two numbers; one is a large prime number, the other is an integer smaller than the prime.They can do this in the open and it doesn't affect security.
- Each of the two parties separately generates another number, which they keep secret.This number is equivalent to a *private key*.A calculation is made involving the private key and the previous two public numbers. The result is sent to the other party.This result is effectively a *public key*.
- The two parties exchange their public keys.They then privately perform a calculation involving their own private key and the other party's public key.The resulting number is the *session key*. Each party will arrive at the same number.
- The session key can be used as a secret key for another cipher, such as DES. No third party monitoring the exchange can arrive at the same session key without knowing one of the private keys.
- The most difficult part of the Diffie-Hellman key exchange to understand is that there are actually two separate and independent encryption cycles happening. As far as Diffie-Hellman

Diffie-Hellman's greatest strength is that anyone can know either or both of the sender and recipient's public keys without compromising the security of the

message. Both the public and private keys are actually just very large integers.

The Diffie-Hellman algorithm takes advantage of complex mathematical functions known as *discrete logarithms*, which are easy to perform forwards but extremely difficult to find inverses for. Even though the patent on Diffie-Hellman has been expired for several years now, the algorithm is still in wide use, most notably in the IPSec protocol. IPSec uses the Diffie-Hellman algorithm in conjunction with RSA authentication to exchange a session key that is used for encrypting all traffic that crosses the IPSec tunnel.

D)RSA Algorithm:

In the year following the Diffie-Hellman proposal, Ron Rivest, Adi Shamir, and Leonard Adleman proposed another public key encryption system.Their proposal is now known as the RSA the mid-1990s. Now you are likely to encounter many programs making extensive use of RSA, such as PGP and Secure Shell (SSH).The RSA algorithm has been in the public domain since RSA Security placed it there two weeks before the patent expired in September 2000.Thus the RSA algorithm is now freely available for use by anyone, for any purpose.

4.Stegnography:

Steganography is a technique used to embed secret information into non-secret information, preventing the message from being detected by non-authorized people.The purpose of steganography is to hide the very presence of communication by embedding messages into innocuous-looking cover objects, such as digital images. To accommodate a secret message, the original cover image is slightly modified by the embedding algorithm to obtain the stego image. The embedding process usually incorporates a secret stego-key that governs the embedding process and it is also needed for the extraction of the hidden message .

There are three basic views behind hiding information. The first is capacity, which is the amount of information that can be embedded within the cover file. An information-hiding algorithm has to be able to compactly store a message within a file. Next is security, which refers to how a third-party can detect hidden information within a file. Intuitively, if a message is to be hidden, an ideal algorithm would store information in a way that was very hard to notice. High security layers have been proposed through three layers to make it difficult to break through the encryption of the input data and confuse steganalysis too. Various encryption techniques like cryptography,digital water marking.

algorithm, named for the last initials of the researchers. RSA shares many similarities with the Diffie-Hellman algorithm in that RSA is also based on multiplying and factoring large integers. However, RSA is significantly faster than Diffie-Hellman, leading to a split in the asymmetric cryptography field that refers to Diffie-Hellman and similar algorithms as Public Key Distribution Systems (PKDS) and RSA and similar algorithms as Public Key Encryption (PKE). PKDS systems are used as session-key exchange mechanisms, while PKE systems are generally considered fast enough to encrypt reasonably small messages. However, PKE systems like RSA are not considered fast enough to encrypt large amounts of data like entire filesystems or high-speed communications lines.

Because of the former patent restrictions on RSA, the algorithm saw only limited deployment, primarily only from products by RSA Security, until

5.Data Security Servicess:

1)confidentiality: information is available for reading only to authorized parties.

2)Authentiction:

- Data source authentication: the data is coming from an authorized party.
- Entity authentication: the entity is who it says it is.

3)**Integrity:**data was not modified from the source to the destination.

4)Non-repudiation:neither the sender, nor the receiver of a message are able to deny the transmission.

5)Acess Control: only authorized parties can use specific resources.

6)**Availability:** resources/services available to authorized parties.

6.conclusion:

The work accomplished during this paper can be summarized with the following points:

1.In this paper we have presented a new system for the combination of **Compression, cryptography and Steganography using three keys** which could be proven a highly secured method for data communication in near future.

2.Steganography especially combined with cryptography and compression is a powerful tool which enables people to communicate without possible eavesdroppers even knowing there is a form of communication in the first place.The proposed method provides acceptable image quality with very little distortion in the image.

7.Referencess:

[1] Domenico Daniele Bloisi , Luca Iocchi: Image based Steganography and cryptography, Computer Vision theory and applications volume 1 , pp. 127-134 .
[2] D.R. Stinson, Cryptography: Theory and Practice, Boca Raton, CRC Press, 1995. ISBN: 0849385210
[3] Mamta Sharma, S.L. Bawa D.A.V. college: Compression Using Huffman Coding, IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.5, May 2010
[4] Kharrazi, M., Sencar, H. T., and Memon, N. (2004). Image Steganography: Concepts and practice. In WSPC Lecture Notes Series
[5] DAVID A. HUFFMAN+, ASSOCIATE, A Method for the Construction of Minimum-Redundancy Codes, PROCEEDINGS OF THE IRE.
[6] Daemen, Joan; Rijmen, Vincent. AES Proposal: Rijndael. ijndael.pdf
[7] Provos, N. and Honeyman, P. (2003). Hide and seek: An introduction to steganography. IEEE SECURITY & PRIVACY
[8] Owens, M., "A discussion of covert channels and steganography", SANS Institute, 2002
[9] Chandramouli, R., Kharrazi, M. & Memon, N., "Image Steganography and steganalysis: Concepts and Practice",
Proceedings of the 2nd International Workshop on Digital Watermarking, October 2003
[10] Jamil, T., "Steganography: The art of hiding information is plain sight", IEEE Potentials, 18:01, 1999
[11] Stefan Katznbeisser, Fabien.A., P.Petitcolas editors, Information Hiding Techniques for Steganography and Digital Watermarking, Artech House, Boston. London, 2000.
[12] Wang, H & Wang, S, "Cyber warfare: Steganography vs. Steganalysis", Communications of the ACM, 47:10, October 2004
[13] Dunbar, B., "Steganography techniques and their use in an Open-Systems environment", SANS Institute, January 2002
[14] C.E., Shannon, (1949), Communication theory of secrecy systems, Bell System Technical Journal, 28, 656-715.
[15] Marvel, L.M., Boncelet Jr., C.G. & Retter, C., "Spread Spectrum Steganography", IEEE Transactions on image processing, 8:08, 1999
[16] N. F. Johnson and S. Katzenbeisser, .A survey of steganographic techniques., in S. Katzenbeisser and F. Peticolas (Eds.): Information Hiding, pp.43-78. Artech House, Norwood, MA, 2000.
[17] Currie, D.L. & Irvine, C.E.,"Surmounting the effects of lossy compression on Steganography", 19th National Information Systems Security Conference, 1996
[18] G., Derrick, (2001), Data watermarking Steganography and watermarking of digital data, Computer Law & Security Report, 17 (2), 101-104.
[19] Ross J. Anderson, Fabien A.P. Petitcolas, "On The Limits of Steganography", IEEE Journal of Selected Areas in Communications, 16(4): 474-481, May 1998. Special Issue on Copyright & Privacy Protection. ISSN 0733-8716.
[20]http://www.codeproject.com/KB/library/ArisFFTDFTLibrary.aspx

Authors:

Epuru Madhavarao, Pursuing M.Tech(CSE) from Vignan's Lara Institute of Technology & Science, Vadlamudi, Guntur, A.P., India. My research Interests are Data Security and computer networks.
Chikkala JayaRaju, Pursuing M.Tech(CSE) from Vignan's Lara Institute of Technology & Science, Vadlamudi, Guntur, A.P., India. My research Interests are Semantic Web and Mobile computing.
Pedasanaganti Divya, Pursuing M.Tech(CSE) from Vignan's Lara Institute of Technology & Science, Vadlamudi, Guntur, A.P., India. My research Interests are computer networks and Mobile computing.
A.S.K.Ratnam, Assoc.Professor & Head, Department of Computer Science Engineering at Vignan's Lara Institute of Technology & Science, Vadlamudi, Guntur, A.P., India..My research Interests includes Image Processing, Mobile Computing, Network Security.